

Substitute for form 1449/PTO				Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Use as many sheets as necessary)</i>				Application Number	Not yet assigned
				Filing Date	Concurrently Herewith
				First Named Inventor	Guy Larri
				Art Unit	Not yet assigned
				Examiner Name	Not yet assigned
Sheet	1	of	1	Attorney Docket Number	M0025.0369-C

U.S. PATENT DOCUMENTS					
Examiner Initials*	Cite No. ¹	Document Number	Publication Date MM-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number-Kind Code ² (if known)			
		5,881,312	03-1999	Dulong	
		7,328,153	02-2008	Wells et al.	
		2004/0111264 A1	06-2004	Wang et al.	
		2003/0061046 A1	03-2003	Zhao et al.	
		6,374,220	04-2002	Kao	
		9,076,441	07-2015	Larri et al.	
		8,352,262	01-2013	Larri et al.	
		7,979,277	07-2011	Larri et al.	
		8,036,890	10-2011	Catchpole	
		8,612,227	12-2013	Kato	

FOREIGN PATENT DOCUMENTS						
Examiner Initials*	Cite No. ¹	Foreign Patent Document	Publication Date MM-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear	T ⁶
		Country Code ³ -Number ⁴ -Kind Code ⁵ (if known)				
		GB 2 391 679 A	02-2004			
		EP 1 178 466 A2	02-2002			
		WO 01/75862 A2	10-2001			
		WO 01/48737 A2	07-2001			
		GB 2 333 172 A	07-1999			
		WO 03/067572 A2	08-2003			

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. ¹ Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. ³ Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). ⁴ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁵ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁶ Applicant is to place a check mark here if English language Translation is attached.

NON PATENT LITERATURE DOCUMENTS				
Examiner Initials	Cite No. ¹	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.		T ²
		Glinski et al., "Spoken Language Recognition on DSP Array Processor" IEEE Transactions on Parallel and Distributed Systems, Vol. 5, No. 7, pp. 697-703, July 1, 1994.		
		S. Chatterjee et al., "Connected Speech Recognition on a Multiple Processor Pipeline." ICASSP 89, pp. 774-777, May 23, 1989.		

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

Examiner Signature		Date Considered	
--------------------	--	-----------------	--

DSMDB-3348573 v1

Electronic Patent Application Fee Transmittal				
Application Number:				
Filing Date:				
Title of Invention:		A SPEECH RECOGNITION CIRCUIT AND METHOD		
First Named Inventor/Applicant Name:		Guy Larri		
Filer:		Stephen A. Soffen/Madeline Baker		
Attorney Docket Number:		M0025.0369-c		
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Utility filing Fee (Electronic filing)	4011	1	70	70
Utility Search Fee	2111	1	300	300
Utility Examination Fee	2311	1	360	360
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				730

Electronic Acknowledgement Receipt	
EFS ID:	22793233
Application Number:	14788164
International Application Number:	
Confirmation Number:	1250
Title of Invention:	A SPEECH RECOGNITION CIRCUIT AND METHOD
First Named Inventor/Applicant Name:	Guy Larri
Customer Number:	24998
Filer:	Stephen A. Soffen/Madeline Baker
Filer Authorized By:	Stephen A. Soffen
Attorney Docket Number:	M0025.0369-c
Receipt Date:	30-JUN-2015
Filing Date:	
Time Stamp:	17:57:34
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	\$730
RAM confirmation Number	6125
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		M0369c.pdf	4739441	yes	107
			f5ab446257e08ebfb8722eb57db09d1d899e732f		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Transmittal of New Application		1	1	
	Application Data Sheet		2	6	
	Appendix to the Specification		7	70	
	Claims		71	71	
	Abstract		72	72	
	Drawings-only black and white line drawings		73	99	
	Oath or Declaration filed		100	103	
	Power of Attorney		104	104	
	Transmittal Letter		105	106	
	Information Disclosure Statement (IDS) Form (SB08)		107	107	
	Warnings:				
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	35187	no	2
			172c6a028f9412e4020ed957ba8bb8ca324bbbbb4		
Warnings:					
Information:					
Total Files Size (in bytes):			4774628		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

POWER OF ATTORNEY TO PROSECUTE APPLICATIONS BEFORE THE USPTO

As the Applicant and/or Assignee I hereby appoint:



Practitioners associated with the Customer Number:

24998

OR



Practitioner(s) named below (if more than ten patent practitioners are to be named, then a customer number must be used):

Name	Registration Number	Name	Registration Number

as attorney(s) or agent(s) to represent the Applicant and/or Assignee before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned to the Applicant and/or Assignee, under an obligation to be assigned to the Applicant and/or Assignee, or for which the Applicant and/or Assignee has a sufficient proprietary interest.

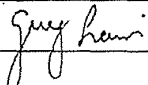
Assignee and/or Applicant Name and Address:

Zentian Limited
Paddock Row
Elsworth, Cambridge
Cambridgeshire CB23 4JG
United Kingdom

The practitioners appointed in this form are authorized to act on behalf of the assignee and/or applicant to transact all business in the United States Patent and Trademark Office.

SIGNATURE of Assignee and/or Applicant

The individual whose signature and title is supplied below is authorized to act on behalf of the assignee and/or applicant

Signature		Date	28 July 2014
Name	GUY LARRÉ	Telephone	07768424838 in UK
Title and Company	Director, Zentian Limited		

DSMDB-3259765 v1

Docket No.: M0025.0369-C
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Letters Patent of:
Guy Larri et al.

Confirmation No.: Not yet assigned

Application No.: Not yet assigned

Group Art Unit: Not yet assigned

Issued: Concurrently Herewith

Examiner: Not yet assigned

For: A SPEECH RECOGNITION CIRCUIT AND
METHOD

INFORMATION DISCLOSURE STATEMENT (IDS)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Madam:

Pursuant to 37 CFR 1.56, 1.97 and 1.98, the attention of the Patent and Trademark Office is hereby directed to the references listed on the attached PTO/SB/08. It is respectfully requested that the information be expressly considered during the prosecution of this application, and that the references be made of record therein and appear among the "References Cited" on any patent to issue therefrom.

This Information Disclosure Statement accompanies the continuation patent application submitted herewith. Those documents cited in the attached form PTO/SB/08 that were previously cited by or submitted to the Office in prior application number 13/735,091, filed July 7, 2013 and/or the parent applications thereof, and relied upon in this application for an earlier filing date under 35 U.S.C. 120, are not supplied.

DSMDB-3348571 v1

Application No.: Not yet assigned

Docket No.: M0025.0369-C

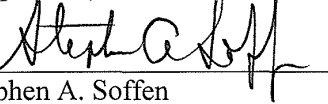
In accordance with 37 CFR 1.98(a)(2)(ii), Applicant has not submitted copies of U.S. patents and U.S. patent applications.

In accordance with 37 CFR 1.97(g), the filing of this Information Disclosure Statement shall not be construed to mean that a search has been made or that no other material information as defined in 37 CFR 1.56(b) exists. In accordance with 37 CFR 1.97(h), the filing of this Information Disclosure Statement shall not be construed to be an admission that any patent, publication or other information referred to therein is "prior art" for this invention unless specifically designated as such.

It is submitted that the Information Disclosure Statement is in compliance with 37 CFR 1.98 and the Examiner is respectfully requested to consider the listed references.

Dated: June 30, 2015

Respectfully submitted,

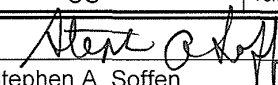
By 

Stephen A. Soffen

Registration No.: 31,063
DICKSTEIN SHAPIRO LLP
1825 Eye Street, NW
Washington, DC 20006-5403
(202) 420-2200
Attorney for Applicant

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL <small>(ONLY FOR NEW NONPROVISIONAL APPLICATIONS UNDER 37 CFR 1.53(B))</small>		Attorney Docket No. M0025.0369-C	
		First Inventor Guy Larri	
		Title A SPEECH RECOGNITION CIRCUIT AND METHOD	
		Express Mail Label No.	

APPLICATION ELEMENTS <small>See MPEP chapter 600 concerning utility patent application contents.</small>	ADDRESS TO: Commissioner for Patents P.O. Box 1450 Alexandria, VA 22313-1450
1. ☐ Fee Transmittal Form. <small>(PTO/SB/17 or equivalent)</small> 2. ☒ Applicant claims small entity status. <small>See 37 CFR 1.27.</small> 3. ☒ Specification. [Total Pages <u>66</u>] <small>Both the claims and abstract must start on a new page (For information on the preferred arrangement, see MPEP § 608.01(a))</small> 4. ☒ Drawing(s). (35 U.S.C. 113) [Total Sheets <u>27</u>] 5. Inventor's Oath or Declaration. [Total Sheets <u>4</u>] <small>(including substitute statements under 37 CFR 1.64 and assignments serving as an oath or declaration under 37 CFR 1.63(e))</small> a. ☒ Newly executed (original or copy) b. ☐ A copy from a prior application (37 CFR 1.63(d)) 6. ☒ Application Data Sheet. *See Note below. <small>See 37 CFR 1.76 (PTO/AIA/14 or equivalent)</small> 7. ☐ CD-ROM or CD-R. <small>In duplicate, large table or Computer Program (Appendix)</small> ☐ Landscape Table on CD 8. Nucleotide and/or Amino Acid Sequence Submission. <small>(if applicable, items a. – c. are required)</small> a. ☐ Computer Readable Form (CRF) b. Specification Sequence Listing on: i. ☐ CD-ROM or CD-R (2 copies); or ii. ☐ Paper c. ☐ Statements verifying identity of above copies	ACCOMPANYING APPLICATION PARTS 9. ☐ Assignment Papers. (cover sheet & document(s)) <div style="border: 1px solid black; height: 20px; margin-top: 5px;"></div> 10. ☐ 37 CFR 3.73(c) Statement. ☒ Power of Attorney. <small>(when there is an assignee)</small> 11. ☐ English Translation Document. (if applicable) 12. ☒ Information Disclosure Statement. <small>(PTO/SB/08 or PTO-1449)</small> ☐ Copies of citations attached 13. ☐ Preliminary Amendment. 14. ☐ Return Receipt Postcard. <small>(MPEP § 503) (Should be specifically itemized)</small> 15. ☐ Certified Copy of Priority Document(s). <small>(if foreign priority is claimed)</small> 16. ☐ Nonpublication Request. <small>Under 35 U.S.C. 122 (b)(2)(B)(i). Applicant must attach form PTO/SB/35 or equivalent.</small> 17. ☐ Other: <div style="border: 1px solid black; height: 30px; width: 100%;"></div>
<p>*Note: (1) Benefit claims under 37 CFR 1.78 and foreign priority claims under 1.55 must be included in an Application Data Sheet (ADS). (2) For applications filed under 35 U.S.C. 111, the application must contain an ADS specifying the applicant if the applicant is an assignee, person to whom the inventor is under an obligation to assign, or person who otherwise shows sufficient proprietary interest in the matter. See 37 CFR 1.46(b).</p>	
19. CORRESPONDENCE ADDRESS	
☒ The address associated with Customer Number: <u>24998</u> OR ☐ Correspondence address below	
Name	Stephen A. Soffen DICKSTEIN SHAPIRO LLP
Address	1825 Eye Street, NW
City	Washington
State	DC
Zip Code	20006-5403
Country	US
Telephone	(202) 420-2200
Email	
Signature	
Date	June 30, 2015
Name (Print/Type)	Stephen A. Soffen
Registration No. (Attorney/Agent)	31,063

DSMDB-3348578 v1

Application Data Sheet

Inventor Information

Inventor Number:: 1
Given Name:: Guy
Family Name:: Larri
City of Residence:: Cambridge
Country of Residence:: United Kingdom
Street of mailing address:: 1 Cutter Ferry Lane
City of mailing address:: Cambridge
Country of mailing address:: United Kingdom
Postal or Zip Code of mailing address:: CB4 1JR

Inventor Number:: 2
Given Name:: Mark
Family Name:: Catchpole
City of Residence:: Prickwillow
Country of Residence:: United Kingdom
Street of mailing address:: Malahax, Ely Road
City of mailing address:: Prickwillow
Country of mailing address:: United Kingdom
Postal or Zip Code of mailing address:: CB7 4UJ

Inventor Number:: 3
Given Name:: Damian
Middle Name:: Kelly
Family Name:: Harris-Dowsett
City of Residence:: Coventry
Country of Residence:: United Kingdom
Street of mailing address:: 22 Hollis Road
City of mailing address:: Coventry
Country of mailing address:: United Kingdom
Postal or Zip Code of mailing address:: CV3 1AL

Inventor Number:: 4
Given Name:: Timothy
Middle Name:: Brian
Family Name:: Reynolds
City of Residence:: Cambridge
Country of Residence:: United Kingdom
Street of mailing address:: 22 Camside
City of mailing address:: Cambridge
Country of mailing address:: United Kingdom
Postal or Zip Code of mailing address:: CB4 1PQ

Correspondence Information

Correspondence Customer Number:: 24998

Application Information

Application Type:: Regular

Subject Matter:: Utility

CD-ROM or CD-R?: None

Sequence submission?: None

Computer Readable Form (CRF)?:: No

Title:: A SPEECH RECOGNITION CIRCUIT AND
METHOD

Attorney Docket Number:: M0025.0369-C

Request for Early Publication?: No

Request for Non-Publication?: No

Total Drawing Sheets:: 27

Small Entity?: Yes

Petition included?: No

Secrecy Order in Parent Appl.?: No

Authorization to Permit Access:: No

Representative Information

Representative Customer Number:: 24998

Domestic Priority Information

Application::	Continuity Type::	Parent Application::	Parent Filing Date::	Status
This Application	is a continuation of	13/735,091	01/07/2013	Pat. 9,076,441
13/735,091	is a continuation of	13/162,128	06/16/2011	Pat. 8,352,262
13/162,128	Is a divisional of	11/662,704	03/14/2007	Pat. 7,979,277
11/662,704	Is a national stage under 35 USC 371 of	PCT/GB2005/003554	09/14/2005	

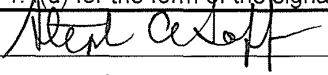
Foreign Priority Information

Country::	Application number::	Filing Date::	Priority Claimed::
United Kingdom	0420464.0	09/14/2004	Yes

Applicant Information

Applicant Number:: 1
Applicant Type:: Assignee
Organization Name:: Zentian Limited
Street of mailing address:: Paddock Row, Elsworth
City of mailing address:: Cambridge
Country of mailing address:: United Kingdom
Postal or Zip Code of mailing address:: CB23 4JG

Signature:

A signature of the applicant or representative is required in accordance with 37 CFR 1.33 and 10.18. Please see 37 CFR 1.4(d) for the form of the signature.			
Signature		Date	June 30, 2015
Name (Print/Type)	Stephen A. Soffen	Registration No. (Attorney/Agent)	31,063

A SPEECH RECOGNITION CIRCUIT AND METHOD

This is a continuation of application Serial No. 13/735,091, filed January 7, 2013, now Pat. 9,076,441, which is a continuation of application Serial No. 13/162,128, filed June 16, 2011, now Pat. 8,352,262, which is a divisional of application Serial No. 11/662,704, filed March 14, 2007, now Pat. 7,979,277, which is a 371 of PCT/GB2005/003554, filed September 14, 2005, which claims priority to UK Application No. 0420464.0, filed September 14, 2004, the disclosures of which are hereby incorporated by reference in their entireties.

BACKGROUND OF THE INVENTION

1. Field of the Invention:

The present invention relates to speech recognition circuits and methods. These circuits and methods have wide applicability, particularly for devices such as mobile electronic devices.

2. Description of the Related Art:

There is growing consumer demand for embedded speech recognition in mobile electronic devices, such as mobile phones, dictation machines, PDAs (personal digital assistants), mobile games consoles, etc. For example, email and text message dictation, note taking, form filling, and command and control applications are all potential applications of embedded speech recognition.

However, when a medium to large vocabulary is required, effective speech recognition for mobile electronic devices has many difficulties not associated with speech recognition systems in hardware systems such as personal computers or workstations. Firstly, the available power in mobile systems is often supplied by battery, and may be severely limited. Secondly, mobile electronic devices are frequently designed to be as small as practically possible. Thus, the memory and resources of such mobile embedded systems tends to be very limited, due to power and space restrictions. The cost of providing extra memory and resources in a mobile electronic device is typically much higher than that for a less portable device without this space restriction. Thirdly, the

DSMDB-#3348563-v1

mobile hardware may be typically used in a noisier environment than that of a fixed computer, e.g. on public transport, near a busy road, etc. Thus, a more complex speech model and more intensive computation may be required to obtain adequate speech recognition results.

These restrictions have made it difficult to implement effective speech recognition in mobile devices, other than with very limited vocabularies.

Some prior art schemes have been proposed to increase the efficiency of speech recognition systems, in an attempt to make them more suitable for use in mobile technology.

In an article entitled “A low-power accelerator for the SPHINX 3 speech recognition system”, in University of Utah, International conference on Compilers, Architectures and Synthesis for Embedded Systems, Nov 2003, Davis et al have proposed the idea of using a special purpose co-processor for up-front calculation of the computationally expensive Gaussian output probabilities of audio frames corresponding to particular states in the acoustic model.

In an article entitled “Hardware Speech Recognition in Low Cost, Low Power Devices”, University of California, Berkeley, CS252 Class Project, Spring 2003, Sukun Kim et al describe using special purpose processing elements for each of the nodes in the network to be searched. This effectively implies having a single processing element for each phone in the network. An alternative suggested by Sukun Kim et al is to provide a processor for each state in the network.

In an article entitled “Dynamic Programming Search for Continuous Speech Recognition” in IEEE Signal Processing Magazine, Sept 1999, Ney et al discuss language model lookahead. Language model lookahead involves computation of a language model factor for each node (i.e. phone) in the lexical tree. This technique is also known as smearing. Each phone instance in the search network can be given a language model factor when it is used in the lexical tree search. Ney et al show that for an example bigram language model, the average number of states per 10 ms frame can

DSMDB-#3348563-v1

be reduced from around 168,000 states with no language model lookahead to around 8,000 states when language model lookahead is used. They also show that bigram language model lookahead requires about a quarter of the states compared with unigram language model lookahead.

Although these prior art documents provide improvements to speech recognition in embedded mobile technology, further improvement is still needed to provide a larger vocabulary and better accuracy.

SUMMARY OF THE INVENTION

One aspect of the present invention provides a speech recognition circuit including a circuit for providing state identifiers which identify states corresponding to nodes or groups of adjacent nodes in a lexical tree, and for providing scores corresponding to said state identifiers. The lexical tree includes a model of words. The speech recognition circuit also has a memory structure for receiving and storing state identifiers identified by a node identifier identifying nodes or groups of adjacent nodes, the memory structure being adapted to allow lookup to identify particular state identifiers, reading of the scores corresponding to the state identifiers, and writing back of the scores to the memory structure after modification of the scores. An accumulator is provided for receiving score updates corresponding to particular state identifiers from a score update generating circuit which generates the score updates using audio input, for receiving scores from the memory structure, and for modifying said scores by adding said score updates to said scores. A selector circuit is used for selecting at least one node or group of nodes of the lexical tree according to said scores.

One suitable type of hardware for the memory structure includes a content addressable memory (CAM). A CAM is a memory unit which stores a series of data items using a series of addresses. However, the memory is accessed by specifying a data item, such that the CAM returns the corresponding address. This contrasts with a random access memory (RAM) in which the memory is accessed by specifying an address, such that the RAM returns the corresponding data item.

DSMDB-#3348563-v1

However, the memory structure is not limited to including a CAM. Other types of hardware are also possible, to provide this functionality. For example, a single chip which operates in the same way as a CAM and RAM may be used instead.

Embodiments of the present invention provide a solution to the problem of how to map a lexical tree search to a CAM system architecture. The realisation by the present inventors that certain speech recognition data structures can be mapped into the CAMs allows a lexical tree search to be performed using a CAM system architecture.

Further embodiments of the invention include a counter for sequentially generating state identifiers, and using said generated state identifiers to sequentially lookup said states in the memory structure.

The node identifier may comprise a direct reference to the lexical tree. However, in some embodiments, the node identifier for at least some of the states includes a pointer to a node identifier for another state. For example, a state corresponding to the furthest part of the search path in the lexical tree may be referenced by a node identifier which directly links to a particular node or group of nodes in the lexical tree. In a lexical tree comprising phones, using a state model of triphones, the node identifier may indicate the position of a triphone in the lexical tree.

However, in this example, for states occurring further back in the search path, instead of supplying a node identifier linking directly to the lexical tree, instead a pointer to a node identifier of another state may be supplied. E.g. a triphone instance may have a pointer to another triphone instance, which has a pointer to another triphone instance, which has a pointer to a node or group of nodes in the lexical tree. Chains of reference may be set up in this way, where only the last state in the chain has a direct pointer to the lexical tree.

There may not be a one-to-one correspondence between the nodes of the lexical tree and the node identifiers. This will occur for a branched lexical tree, where the nodes represent monophones, but the acoustic model states represent triphones, i.e. groups of three adjacent monophones. Then, paths of three monophones will have unique

DSMDB-#3348563-v1

identifiers to be stored in the memory structure, rather than single monophones having unique identifiers.

Phone instance numbers may be generated, and used to uniquely label each phone instance. They can be generated sequentially, using a counter. The phone instance numbers may be used as pointers between phone instances to assist in node identification. It is thus not essential to provide a direct node identifier for each phone instance to directly indicate a location in the lexical tree. The dynamic network of phone instances provided in the memory structure may thus include both direct and relative references to the lexical tree.

The memory structure may be divided into one part which stores phone instance identifiers and direct references to the lexical tree, and a second part which stores phone instance identifiers and corresponding states. This can speed up the processing, by only storing the phone instances which are furthest on in the lexical tree in the first part of the memory structure.

The memory structure may also be divided into separately accessible units, to reduce the amount of data in each unit, thereby decreasing the chance of finding the same two states identifiers in different phone instances in any single memory unit, and increasing the chance of some state identifiers being completely absent from any single memory unit. This makes it easier to deal with the situation when the same two state identifiers are found, because a spare time slot is available for processing when a state identifier is not present.

A further aspect of the invention provides a distance calculation engine within a speech recognition system. The distance calculation engine may be included within an accelerator. The accelerator may include logic to interface with other parts of a speech recognition circuit, in addition to the distance engine, although this is not essential. For example, the accelerator may include one or more results memories for storing distances calculated by the distance calculation engine. The accelerator may also include at least one of a memory for storing one or more acoustic models, a decompressor for decompressing acoustic data that has been stored in a compressed format, a memory for

DSMDB-#3348563-v1

storing feature vectors, a checksum or data signature calculation means, buffers for data storage, and data registers. The accelerator may be implemented in software or in hardware, or in a combination. It may be physically separate to the rest of the speech recognition circuit, although this is not essential.

The distance calculation engine may calculate one or more of a wide range of distance metrics and probability distributions. The distances may represent the likely correspondance of feature vectors to states in an acoustic model. In other words, the distances can indicate the similarity of an audio data frame to each possible state in an acoustic model

There are a wide variety of probability distributions that can be used for the distance calculation stage of a speech recogniser, and a wide variety of distance metrics used. These are widely documented in the literature. A point is a simple example of a probability distribution.

A common choice is to use Gaussian Distributions and correspondingly the Mahalanobis Distance metric. The Gaussian probability distribution is then defined by a mean vector, which defines centre point in the N-dimensional space, and a Covariance matrix which defines the shape of the probability distribution. It is common to restrict the Covariance matrix to be a diagonal matrix (only N non-zero values along the diagonal of the NxN matrix) which significantly lowers the implementation cost by reducing the number of arithmetic operations.

In particular embodiments, the distance calculated is a Mahalanobis distance. Particular examples of this are described later in the specification.

In one embodiment, the distance engine autonomously computes all of the distances associated with a given feature vector. This may comprise computing distances for every state in the lexicon. The distance engine may operate in a pipelined manner with other stages of the recognition process. In this context a distance is an indication of the probability or likelihood that a feature vector corresponds to a particular state. An

important class of distance computation in speech recognition is the calculation of output state probabilities in recognisers using Hidden Markov Models. Another possible use is in recognisers using Neural Networks.

The distance engine reads data from the acoustic models to use as parameters in the calculation. The acoustic models may be optionally stored in a compressed format. The distance engine may read and de-compress the acoustic models one (or more) times for each feature vector processed. Each reading of the acoustic models may require reading the entire acoustic model, or various optimisations may be implemented to avoid reading parts of the acoustic model that are not required to be used for calculations with the current feature vector. The distance engine may use a de-compression method where the de-compression is sign or zero extension or may otherwise convert data of narrow or variable width to a wider data format. The distance engine may use a de-compression method where the de-compression is sign or zero extension or may otherwise convert data of narrow or variable width to IEEE standard single or double precision floating point format. The distance engine may use a decompression method where decompression is a codebook decompression of a binary bitstream, where the codebook is stored as part of the acoustic model data. The distance engine may use a decompression method where the decompression is decompression of a Huffman or Lempel-Ziv compressed stream. The distance engine may use a decompression method where decompression is decompression of run length encoded data. The distance engine may use a decompression method where decompression is decompression of difference encoded data. The distance engine may use a decompression method using any well known lossy or lossless compression scheme. The distance engine may use a decompression method using subspace distribution clustering. The distance engine may use a decompression method comprising any combination of the above described decompression types.

The distance engine may read the acoustic models from a dedicated on-chip memory. The distance engine may read the acoustic models from a dedicated off-chip memory. The distance engine may read the acoustic models from a shared on-chip memory. The distance engine may read the acoustic models from a shared off-chip memory. Any of these acoustic models may be compressed.

The distance engine may compute a CRC or checksum or similar signature as it reads in the acoustic model and compares this to a stored CRC, checksum, or signature, in order to check that the acoustic model has not been corrupted, and signals an error condition if such corruption is detected. The stored CRC, Checksum, or signature may have been pre-computed and stored in the model data, or it may be computed at the time the model data is loaded into the Acoustic Model Memory. It may be held in Acoustic Model Memory, or it may be loaded into a register or another memory from where it can be accessed and compared when the CRC/checksum/signature is computed each time the Acoustic Model is loaded.

The distance engine may support the pass-through of data from the front-end to the search-stage. The data to be passed through will be supplied to the distance engine as an adjunct to the feature vector, and the distance engine will pass it to the search stage as an adjunct to the distance results. This provides a simple mechanism for passing frame-specific data that is not involved in the distance calculation through to the search stage, and keeping it associated with the correct frame, which may otherwise be complex in pipelined systems with multiple processors. The data passed through may be for any purpose. Examples might include silence detected, end-of-audio-stream detected, a frame number, information that some intervening frames have been dropped, or data from another input device such as a button or keyboard in a multi-modal interface.

The distance engine may be implemented in hardware, software, or a combination. Other stages may be implemented in hardware, software, or a combination. The distance engine may be implemented with any number representation format including fixed point or floating point arithmetic, or any mixture of number representation formats.

In particular, the other stages may be implemented on a CPU, or on a DSP and CPU. The "DSP" and "CPU" may each be implemented as software programmable devices.

The distance engine may implement one or more additional pipeline stages to overcome delays introduced by low bandwidth, high latency, or conflicted bus interfaces. The distance engine may also implement additional pipeline stages to maintain the same

DSMDB-#3348563-v1

throughput while allowing more time for each distance calculation. Particular embodiments of the invention may include one or more of the above aspects.

A further aspect of the invention comprises a speech recognition circuit, comprising: an audio front end for calculating a feature vector from an audio signal, wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame; calculating circuit for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words; a buffer memory between the calculating circuit and the search stage, for receiving data passing from the calculating circuit to the search stage, wherein a processor in the search stage has higher bandwidth and/or lower latency access to the buffer compared to the bandwidth and/or latency of direct transfer between the calculating circuit and the search stage. The data transfer from the calculating circuit to the buffer memory and/or from the buffer memory to the search stage may be performed as one or more sequential bursts. The data transfer to the buffer memory may be performed in parallel with data transfer to the calculating circuit and/or in parallel with data transfer to the search stage. A second buffer memory may be provided between the audio front end and the calculating circuit.

A further aspect of the invention comprises a speech recognition circuit, comprising: an audio front end for calculating a feature vector from an audio signal, wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame; calculating circuit for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words; comprising an elastic buffer between at least one of the front end and calculating circuit, or the calculating circuit and search stage, and/or for buffering said audio signal.

A further aspect of the invention comprises an accelerator for a speech recognition circuit, the accelerator comprising: calculating means for calculating a distance

DSMDB-#3348563-v1

indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model, wherein the feature vector comprises a plurality of extracted and/or derived quantities from an audio signal during a defined audio time frame; means for comparing a first version of a stored checksum of data representing said acoustic model and a second version of said stored checksum, wherein the second version is obtained from an updated measurement and calculation of the checksum; and means for indicating an error status if the checksums do not match.

A further aspect of the invention comprises an accelerator for a speech recognition circuit, the accelerator comprising: calculating means for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model, wherein the feature vector comprises a plurality of extracted and/or derived quantities from an audio signal during a defined audio time frame; wherein said accelerator is configured to autonomously compute distances for every acoustic state defined by the acoustic model.

A further aspect of the invention comprises an accelerator for calculating distances for a speech recognition circuit, the accelerator comprising: calculating circuit for calculating distances indicating the similarity between a feature vector and a plurality of predetermined acoustic states of an acoustic model, wherein the feature vector comprises a plurality of extracted and/or derived quantities from an audio signal during a defined audio time frame; first and second storage circuit, which may be referred to as result memories, each for storing calculated distances for at least one said audio time frame, and for making said stored distances available for use by another part of the speech recognition circuit; control circuit for controlling read and write access to the first and second storage circuit, said control means being configured to allow writing to one said storage means while the other said storage means is available for reading, to allow first calculated distances for one audio time frame to be written to one said storage means while second calculated distances for an earlier audio time frame are made available for reading from the other said storage means.

Embodiments of the invention may comprise means for generating a checksum or computed signature for the acoustic model data stored in the memory, and means for comparing checksums or computed signatures that have been calculated at different

DSMDB-#3348563-v1

times, to indicate an error status if the checksums do not match, one possible cause of such mismatch being that the acoustic model data has been overwritten by said other data and said error status being used to indicate that the acoustic model should be re-loaded into the said memory.

A further aspect of the invention comprises a speech recognition circuit comprising: lexical memory containing lexical data for word recognition, said lexical data comprising a lexical tree data structure comprising a model of words; means for accessing a state model corresponding to each phone or each group of phones in the lexical tree; a content addressable memory for storing content addressable data for each phone or group of phones, including states corresponding to said phone or group of phones, and for storing an address value for each said phone or group of phones; a RAM configured to store accumulated scores for each said phone or group of phones, the accumulated scores being addressable by said address value for each said phone or group of phones; means to obtain scores that each of a plurality of frames of an audio signals corresponds to each of a plurality of said states; a counter to sequentially search for each said state in the content addressable memory, to obtain the corresponding address value if the state is found in the content addressable memory; means to use said address value to access an accumulated likelihood and an accumulator to add said likelihood to the accumulated likelihood; means to use the phones or groups of phones with the highest accumulated scores to obtain a plurality of next phones from the lexical tree which correspond to the next phone; and output means for outputting a lexical tree path of highest likelihood.

A further aspect of the invention comprises speech recognition apparatus comprising: a lexical tree having a corresponding state model; means for obtaining scores of an audio input corresponding to each of a plurality of states in said state model; a content addressable memory for storing a marker indicating a part of the lexical tree, and one or more states associated with said part of the lexical tree; a random access memory addressable by the CAM output, to output accumulated scores for states corresponding to said parts of the lexical tree; adder means for adding likelihood to said accumulated likelihood, to be stored back in the RAM.

A further aspect of the invention comprises speech recognition apparatus, comprising: a CAM-RAM arrangement for storing records including pointers to a lexical tree, and accumulative scores for states within the lexical tree; input means for obtaining scores that an audio frame corresponds to a particular state in the lexical tree; an accumulator for calculating the updated scores and modifying the records in the CAM-RAM accordingly; output means for outputting a path of highest likelihood in the lexical tree.

A further aspect of the invention comprises a speech recognition method comprising: storing state identifiers which identify states corresponding to nodes or groups of adjacent nodes in a lexical tree, and scores corresponding to said state identifiers in a memory structure, the lexical tree comprising a model of words and the memory structure being adapted to allow lookup to identify particular state identifiers, reading of the scores corresponding to the state identifiers, and writing back of the scores to the memory structure after modification of the scores; repeating the following sequence of steps for each of a plurality of incoming frames of an audio signal; obtaining score updates corresponding to the likelihoods that said frame of the audio signal corresponds to each of a plurality of said states; accessing said memory structure to obtain scores, updating the scores by adding score updates to the scores, and writing back the updated scores to the memory structure; determining if scores for states furthest on in the lexical tree correspond to a significant likelihood, and if so, then accessing the lexical tree to determine the next set of possible states; and storing the next set of possible states and said scores of significant likelihood in the memory structure.

A further aspect of the invention comprises speech recognition circuit comprising: a circuit for providing state identifiers which identify states corresponding to phones or groups of adjacent phones in a lexical tree, and for providing scores corresponding to said state identifiers, the lexical tree comprising a model of words; a memory structure for receiving and storing state identifiers and phone instance identifiers uniquely identifying instances of phones or groups of phones in the lexical tree; said memory structure being adapted to allow lookup to identify particular state identifiers, reading of the scores corresponding to the state identifiers, and writing back of the scores to the memory structure after modification of the scores; an accumulator for receiving score updates corresponding to particular state identifiers from a score update generating

DSMDB-#3348563-v1

circuit which generates the score updates using audio input, for receiving scores from the memory structure, and for modifying said scores by adding said score updates to said scores; and a selector circuit for selecting at least one phone instance identifier according to said scores.

A speech recognition apparatus according to the invention may be embedded in or included in a mobile electronic device such as a mobile telephone, PDA (personal digital assistant), etc.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figures 1A-1B are a block diagram of the system architecture for a speech recognition apparatus according to an embodiment of the invention;

Figures 2A-2B are a block diagram showing the main data structures used in the speech recognition apparatus of Figures 1A-1B;

Figures 3A-3B show a block diagram of the Single Phone Instance Network Engine (PINE) of Figure 1B;

Figure 4 is a block diagram showing the Wave Front Manager architecture;

Figure 5 shows the use of link nodes to permit path merging;

Figure 6 shows a block diagram of a Path Merge Unit in the Phone Level;

Figure 7 shows the Phone Book architecture;

Figures 8A-8B show Lexical tree storage;

DSMDB-#3348563-v1

Figure 9 shows the data structure for the word link record;

Figure 10 shows a block diagram of Word Engine;

Figure 11 shows a Language Model interface architecture;

Figure 12 shows an example of a simple Loop Grammar;

Figure 13 shows a block diagram of grammar engine;

Figure 14 shows a single frame of the Audio Codec link protocol; and

Figure 15 illustrates the system timing which may be used in the embodiment of Figures 1A-1B;

Figure 16 is a block diagram showing a speech recognition circuit according to an embodiment of the invention, and illustrating data flow between parts of the circuit;

Figure 17 is a block diagram showing a distance calculation engine according to an embodiment of the invention;

Figure 18 shows ideal data flow over several time steps within a speech recognition circuit having separate front end and search stages, in an embodiment of the invention;

Figure 19 shows an example of actual data flow over several time steps within a speech recognition circuit having separate front end and search stages, in a further embodiment of the invention;

Figure 20 shows ideal data flow over several time steps within a speech recognition circuit having a shared processor for front end and search stages, in an embodiment of the invention;

Figure 21 shows an example of actual data flow over several time steps within a speech recognition circuit having a shared processor for front end and search stages, in an embodiment of the invention;

Figure 22 shows ideal data flow in a speech accelerator pipeline with two data transfer stages, according to an embodiment of the invention; and

Figure 23 shows an example of actual data flow in a speech accelerator pipeline with two data transfer stages according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figures 1A-1B show a speech recognition system architecture for an embodiment of the invention. The speech recognition system is divided into several levels of operation: a state level, a phone level, a word level, a grammar level, and a system level. Each level is shown in Figures 1A-1B as a, interconnected to other levels of the system.

The information passed between different levels or subcircuits of Figures 1A-1B is indicated on Figures 1A-1B using the following identifiers.

Identifier	Definition
grammar_id	Identifies which grammar model is currently being used
grammar_node_id	Identifies a node in the grammar model
word_id	Uniquely identifies a word
wordlat_node_id	Identifies a node in the word lattice
LT_id	Lexical tree identifier. Identifies the lexical tree to be used
LT_branch_id	Identifies a unique branch in a lexical tree. A branch is defined as all of the monophones subsequent to and including the first monophone in a tree.
LT_node_id	Identifies a node in a lexical tree
monophone_id	Identifies a monophone
transvec_id	Identifies which transition vector to use
state_id	Identifies the state
phone_no	Unique phone instance identifier
prev_phone_no	Phone number of previous phone in network
triphone_id	Triphone identifier
LML_factor	Language model lookahead factor. The smeared language model score that must be added

DSMDB-#3348563-v1

The system level includes a general purpose processor and cache, a system controller, a DMA (direct memory access) controller, clocks, PLL (phased locked loop), GPIO (general purpose I/O interface), timer watch dog, memory interfaces to interface to main memory and non volatile storage, serial, host, debug and display interfaces. The system level also includes a model adaptation engine, a language model factor engine, a speech recognition system controller, a language model interface, and a recognition output interface.

The processor organises the transfer of the various models from the non volatile storage to the various model memories. It handles the language model factor requests. Language model factors are used to by the state level to apply the language model score. The language model (LM) interface provides the mechanism for obtaining LM factors as and when they are needed. The processor keeps a copy of the wave front in memory which is updated simultaneously with that in the wave front manager. Using its copy of the lexical tree, the processor can determine which language model scores may be needed in the coming frames. Once calculated, the language model factors can be kept in memory for use in subsequent frames. The system level is also used for forming speech model adaptation, and acting as an overflow for the various buffers used by the architecture.

The system level includes the audio codec interface and the audio front end interface, connected to the processor via a APB (advanced peripheral bus) to AHB (advanced high performance bus) bridge.

The grammar level includes a grammar engine and grammar model(s). The grammar level sets out the rules of the particular recognition task at the highest level. During system initialisation a number of grammar models might be loaded, for example a command grammar and a dictation grammar. Command mode grammar is a restricted grammar which includes only a very limited lexical tree with command specific words. Some branches of this lexical tree may include multiple words together, where a command has such a structure, e.g. “open file menu”. Typically, only a single instance of the lexical tree will be needed. Command mode is used because it can obtain a higher accuracy, due to the lexical tree constraint. In dictation mode grammar, a full lexical tree structure is used.

DSMDB-#3348563-v1

The grammar may be selected by a user during the recognition process. For example, the command grammar may be selected when the user presses a command button. The architecture supports fast swapping of grammar models. For example, a user may be in the middle of dictating a message when a “punctuation button” is pressed and the phrase “question mark” uttered. At this point, a punctuation grammar could be loaded, in order to increase recognition accuracy.

The word level comprises a word engine, which includes a word lattice and a word graph generator. The word level has two main functions, of keeping a record of the search path at the word level using a word lattice, and interfacing to the language model via a cache. The word lattice is a record of all of the words that have been searched up to the current frame. As new words are searched, they are appended into the lattice at the appropriate place. After the final frame in the current sentence, the n-best word lattice node identifiers from the n-best scoring tokens are passed back to the word lattice. This enables the n-best sentences to be traced back through the lattice. The word lattice can also be post processed to produce a word graph which is essentially a compact version of the word lattice.

At the word level, a record is kept of all tokens that have passed through a word end. This results in a lattice of words. When the final feature vector has been processed, the best scoring token is identified and used to trace back the search path back to the start of the search.

The phone level includes a “phone book” for generating new triphone instances, a wave front manager for sending requests to the phone book to generate new phone instances, a phone request arbiter, and a path merge unit.

The phone book includes a lexical tree, which comprises a vocabulary of words, sounds and/or phrases organised according to the phones or monophones making up the sound. A lexical tree is commonly used in speech recognition rather than a linear lexicon, because it greatly reduces the amount of storage space required to store a word vocabulary, by avoiding duplication of common word beginnings. It is also easier to search when the word beginning is known.

DSMDB-#3348563-v1

Although it is possible to search a lexical tree based on monophones, it is well known that better results can be achieved in speech recognition by basing the lexical tree search on groups of more than one monophone, since the sound of one syllable commonly affects the pronunciation of the next syllable in the same word, or the first syllable in the next word. In this embodiment, groups of three monophones are used, and these are known as triphones. It is alternatively possible to use a different number of monophones for each group. The phone generator in the “phone book” uses data from the lexical tree to generate triphones.

In one embodiment of the invention, approximately 45 different cued mono-phones are considered. This gives of order 60,000 tri-phones. Each triphone in this embodiment is modelled by a Hidden Markov Model (HMM) with three states. There is no correspondence between individual states and individual phones of the triphone. Alternatively, it is possible to model each triphone by a single state, or by any other number of states.

The phone level is responsible for generating the search space according to the word structure specified in the grammar model and the phone structure specified by the lexical tree. The search space can be envisaged as a wave moving through an acousto-linguistic search space. In order that the wave can be dynamically created, a record of the wave front phone instances must be maintained. This is done in a wave front phone CAM/RAM structure, within the wave front manager. When a token occurs in the final state of a phone instance in the PINE array at the front of the wave, a new phone instance must be generated at the phone level and passed to the PINE array.

The phone instance includes a phone_no, which is a phone instance identifier to identify the phone uniquely in the current search. The phone instance also includes a word lattice node identifier (labelled “word_lat_node_id”) to identify the phone’s position in the word lattice, state identifiers (labelled “state_ids”) for the HMM, storage to hold the tokens as they move through the HMM and LM_factor which is used to by the PINE to apply the language model score. To handle the connectivity between phone instances, a phone instance points to a previous phone instance using the phone_no of the previous phone. This allows tokens to be passed between phones. A “token” is a hypothesized

DSMDB-#3348563-v1

search path, represented by a non-zero score at a particular state, and tokens are propagated through the HMM with each time frame.

The phone instance network can be visualised as a wave with new phone instances being added to the front of the wave, and empty phone instances being removed from the back of the wave. In some circumstances, the back of the wave may be in the same position as the wave front. The back of the wave moves on through the lexical tree as tokens are either moved on or pruned.

Tokens, each representing a particular hypothesis, hop through the states in the HMMs, each hop consuming a feature vector. Each token explores all possible next states which typically involves placing a token in the next state and keeping one in the current state. When there is more than one token in a state, the least likely token can be removed because both have the same word history. When a token reaches the last state of a phone instance at the front of the wave, the next phone instance(s) are requested from the phone level. The phone level uses the lexical tree to create the new phones. As well as looking up the state identification numbers for each HMM state, it also incorporates the correct language model factor.

The language model factor engine is used to smear the language model scores across the lexical tree. This operates in two ways depending on whether the factor is needed for a word start phone(s) or word internal phones. For word internal phones, LM factors can be computed as needed, provided that the LM scores are available in cache. This is because in general the storage requirements for the LM scores are less than the storage of the LM factors for all of the factors in the branch. However when a token is in a word end phone there may be a significant number of word start phones e.g. up to (monophones)² requiring a large amount of computation. To avoid this the LM engine will pre compute word start LM factors when it seems highly likely that a token will be passed to the word starts. These word start LM factors will be held in the LM factors cache. The phone level is responsible for generating the search space according to the word structure specified in the grammar model and the phone structure specified by the lexical tree.

The state level shown in Figure 1B includes a phone instance network engine (PINE) array, a MHD (Mahalanobis distance) engine, an audio front end, a feature vector buffer for model adaptation, and compressed acoustic models. The state level is the lowest level, in which a network of phones is dynamically created.

The audio input for speech recognition is input to the front end unit of the state level. This audio input may be digital audio, or it may be analog audio which is converted to digital audio using an analog to digital converter. The audio input is divided into time frames, each time frame typically being of the order of 10ms.

For each audio input time frame, the audio signal is converted into a feature vector. This may be done by splitting the audio signal into spectral components. For example, in the present embodiment, the audio input signal is split into 12 frequency bands and one energy band, and a feature vector is constructed using these 13 components, plus their first and second derivatives, a total of 39 components.

The feature vector is passed to the MHD (Mahalanobis distance) engine. This computes all possible state output likelihoods for the current frame as the state mean and covariance data is read from the acoustic model memory.

The likelihood of each state is determined by the distance between the feature vector and each state. To reduce the number of states, a technique known as "state tying" may be used. Multiple states are grouped together in 39D space, according to their proximity to one another. An average or central value from each group is used to represent the entire group. Thus, the effective number of states may be reduced from 40,000 to 8,000. The accuracy may be improved using a language model in addition to the word recognition lexical tree. A language model is based on the probability of sequential words. The above figure of 8,000 states gives best recognition results when used with a language model.

The MHD (Mahalanobis Distance) is a distance between two N-dimensional points, scaled by the statistical variation in each component. In embodiments of the invention,

the Mahalanobis distance between the feature vector and each state is calculated, to determine similarity of the feature vector to each state.

The Feature Vector represents a point in an N-dimensional space. Different embodiments of the invention may use different values of N. Generally we see values of N in the range of 20 to 39, although values outside this range are not excluded.

The distance calculator stage of the recognition process computes a probability or likelihood that a feature vector corresponds to a particular state. Each state has an associated probability distribution in the N-dimensional space. The distance calculator computes the distance in the N-dimensional space from the Feature Vector to the probability distribution for each state. In some embodiments, the calculation is performed only when the search stage algorithms determine that the distance for that particular state is needed by the search stage. In some embodiments, the distances for all of the states are calculated and made available to the search stage, which allows pipelined operation. Embodiments that do the calculations on demand are preferable.

The Mahalanobis Distance (MHD) is extensively described in the literature. It may be calculated using the formula

$$\text{MHD value} = \ln(P(x)) = -0.5[N \cdot \ln(2 \cdot \pi) + \ln|\text{Cov}|] + (-0.5 \cdot [(x-u)' \cdot \text{inverse}(\text{Cov}) \cdot (x-u)])$$

where

N= number of dimensions in the space

x=feature vector, a point in the N dimensional space

u=mean vector

Cov=covariance matrix

|Cov|=determinant of covariance matrix

(a)'= transpose of matrix a

Different embodiments may use variations on this equation, for example the base for the logarithm is not always the natural base e that is used in the above example. The log of the probability is preferably calculated, rather than the probability, as it simplifies the implementation. Not all embodiments do this. In alternative embodiments, the likelihoods rather than probabilities are computed as it leads to simpler implementations of computations in the search stage.

The feature vector is used to calculate 8,000 MHD distances for each time frame, i.e. one distance for each of the 8,000 states. It is beneficial, although not essential, to perform the MHD calculations separately in a pre-processor, known as a MHD engine, because MHD calculation is a computationally intensive process.

Due to the 10ms frame length, a feature vector arrives at the MHD engine of the state level every 10ms. The MHD engine calculates the likelihood of each feature vector corresponding to each of a number of states derived from an acoustic model. Each state is also a 39 dimensional vector, having the same spectral components as the feature vector. The MHD engine calculates the difference in 39-D space between the feature vector and each state to obtain the likelihood for that state. A parallel MHD or a single MHD arrangement may be used.

The feature vectors may be sent to a feature vector buffer for model adaptation, which determines whether the observed features fit well with the acoustic model, and may initiate an adaptation of the model if the fit is not sufficiently good.

Scores indicating the state likelihoods or probabilities are output from the MHD engine to the PINE array. It is preferable to use likelihoods rather than probabilities, because likelihoods can be added together to give a total likelihood. Probabilities must be multiplied, which can give rise to rounding errors. In this embodiment, eight separate pines are used, each of 4 kilobytes, thus storing 4000 phones per pine.

During recognition, the Phone Instance Network Engine (PINE) Array will contain a network of phone instances. Tokens representing search hypotheses in the form of a probability(likelihood) will exist in some of the HMM states of the phone instances. The

DSMDB-#3348563-v1

tokens are propagated into all available states and updated with the corresponding state output probabilities. Maximum model, word end and beam pruning are applied to the tokens to ensure the search space is kept bounded. When a token is in the final state of a phone model and has avoided being pruned, the PINE makes a next phone request to the Phone level. The PINE array may receive a number of new phone models from the phone level which must be appended to the appropriate existing phone.

Figure 2B shows the data structures stored in the PINE array, and other main data structures used during the acoustic search. Each PINE includes a phone instance CAM and a phone instance RAM. The phone instance CAM stores the phone instance number (labelled “phone_no”) which uniquely identifies each phone instance. The phone instance CAM also stores the state IDs of the three HMM states used to model the phone, and flags indicating the position of the phone. The “back” flag indicates if the phone is at the back of the wavefront. The “front” flag indicates if the phone is at the front of the wavefront. The “empty” flag indicates if there are no tokens in any of the three states for that phone. The “end” flag indicates if the phone is at the end of a word.

The phone instance RAM stores the token scores for each of the three states corresponding to each phone. The st_dum field indicates a dummy state, which is used to represent a state prior to the first state of the first phone. The “tmp” field is used for passing a token from a previous phone in the lexical tree to the current phone. This will be explained in more detail later. The prev_phone_no field is for storing a pointer to the previous phone instance in the network. The wordlat_node_id identifies the position of the phone instance on the word lattice node, and the LM_factor stores the language model factor for that phone instance.

At the phone level a record of all phones on the wave front must be maintained. If a phone instance is on the wave front, the phone_no stored in the phone instance CAM will point to the details in the wave front phone CAM and RAM.

The wave front phone CAM and RAM keeps pointers into the grammar and lexical tree. When a token in a wave front phone reaches the last state, the PINE array makes a request to the phone level by passing back the phone_id. The phone_id is used as a key

DSMDB-#3348563-v1

to determining the token's `grammer_node_id` and `LT_node_id` within the wave front CAM/RAM structure. If the token is at a word end the grammar model is checked. A grammar identifier points to the type of grammar.

At the word level, figures 2A-2B show an example word lattice structure, with silence having a `word_id` of 0, and a `wordlat_node_id` of 6. This is shown as connected to two word nodes – “which” has a `word_id` of 238 and a `wordlat_node_id` of 4, and “neither” has a `word_id` of 3556 and a `wordlat_node_id` of 5.

At the highest level, i.e. the grammar level, a simple grammar model is shown which will allow the phrase ‘silence which house silence’ or ‘silence neither house silence’ to be spoken. Whenever a word end is reached during the search, the grammar model must be consulted to determine the next possible word(s) or lexical tree(s). This allows the word lattice at the word level, consisting of a network of words, to be assembled. Whenever a word end is reached, the word is added to the lattice and the `wordlat_node_id` updated accordingly.

Figures 3A-3B show a single phone instance network engine (PINE). At system power up the state transition RAM will be loaded with the appropriate state transition vector but the CAM and RAM memories will be empty. When the system is initialised an initial set of phone model(s) will be instantiated with a token in the dummy state and a path score of zero. The phone models are received from the phone level.

When the audio data is processed, a stream of all possible state output probabilities will be received from the MHD engine. A counter that is synchronous to the stream of state output probabilities from the MHD produces a concurrent stream of `state_ids`. The job of the CAM is to perform a search to see if the `state_id` is instantiated anywhere in the CAM `state_id` fields. If it exists and the location doesn't have its empty flag set then this indicates that a token may need to be updated. The match signal from the CAM must be encoded into an address for the RAM. The corresponding RAM address can then be read. Any token that will end up in the identified state can then have its path score updated and including any transition vector likelihood.

DSMDB-#3348563-v1

It is possible that a single state_id generates multiple CAM matches. To handle this situation the CAM and RAM are effectively decoupled by means of a match fifo and repeat count fifo. The state_id and corresponding state_output_probability are stored in the repeat count fifo. During a subsequent frame when there is no match, this state_id can be revisited, ignoring the matches already dealt with.

All searched phone instances corresponding to words in the current possible lexical tree are stored in the CAM-RAM. This includes phone instances at the beginning of the word, middle of the word and end of the word. These phone instances are connected by the “previous phone instance” field, which allows traceback through an entire word within the CAM-RAM.

A mechanism is thus required to pass tokens from one phone instance to the next. This mechanism is provided by the tmp field in the RAM.

The tmp field in the RAM holds tokens during token passing.

At the start of each time frame, the audio input is converted into a feature vector, the MHD values calculated, and the MHD values used to update the token scores. After update of all tokens, 2 passes are made through the CAM-RAM, firstly for pruning, and then for token passing. The pruning in this example removes tokens of below a threshold score, although other types of pruning are also possible. The pruning may involve removing a predetermined proportion of the tokens, chosen as those of lowest likelihood. A third possible method of pruning is removing all tokens with scores below a first predetermined value, but at word ends removing all tokens with scores below a second predetermined value, the second predetermined value being greater than the first predetermined value.

The token passing works as follows. Each phone instance points back to a previous phone, due to the prev_phone_no field stored in the RAM. During the token passing, the third token of a previous phone is grabbed and put into the tmp field of the current phone.

In other words, if a token is on the third state, and it is not pruned, it must enter the next state of the next phone. To do this, the tokens that lie in the third state are moved across to the tmp field.

For example, phone no. 345 needs to look at phone no. 320 to see if any tokens are in the third state. If yes, it takes these tokens and puts them in the tmp field.

One field in the CAM is labelled “front”. This indicates whether the phone is on the wave front. If the phone is not on the wave front, then another phone will have picked up the token in the third state. However, if the phone is on the wave front, then no other phone will have picked up the token on the third state. Therefore, if there is a token in the third state, and the phone is on the wavefront, it is necessary to generate a next phone request.

The current phone number, word lattice identifier and word end flag are sent to the wave front manager, to request a next phone. If the word end flag indicates a word end, then the score is also passed to the wave front manager. The score can be passed in any case, but it is only required at a word end.

The phone level generates the next phone or phones. A new phone instance starts, and all of state1, state2 and state3 have scores pre-set to zero. The old score is provided in the tmp field.

The old phone is still within the wave, but it is unlikely to be at the back of the wave at this stage. When it gets to the back of the wave, it is deleted. Until it gets there, it remains in the CAM-RAM as before. The third token remains where it was, since tokens can either stay in the third state or move on.

Note that it is possible to end up with more than one token in each state. If this occurs, then the token with the poorer score is deleted. This does not affect the result of the lexical tree search, because both tokens on the same state have exactly the same word history.

Figure 4 shows a block diagram of the wave front manager. The wave front manager is responsible for maintaining a record of the set of phone models that exist at the front of the acoustic search space. This set of phone models is referred to as the “wave front.” The key requirement for this block is to minimise the time between receiving a “next phone request” and generating the appropriate responses. The wave front manager receives the next phone requests from the PINE array and generates appropriate responses for the Phone Book, Word Engine, Path Merge Unit and Grammar engine. The wave front manager also receives new wave front phone models from the Phone Book and updates the wave front.

The wave front manager receives the next phone request from the PINE array, via the phone request arbiter. The next phone request consists of a phone_no, word_end_flag and wordlat_node_id. If this phone is at the end of a word, the word_end_flag will be set.

The wave front manager uses the phone_no as input to the wave front phone CAM. As each phone_no is unique, it is guaranteed that only a single match will be generated. The wave front manager uses the CAM generated address to access the RAM. It then sends retrieved data from the RAM to the phone book, grammar engine, Path Merge Unit and word engine.

The wave front manager removes phone instance from CAM and RAM by writing 0's into CAM and RAM, and increments the empty count by one.

Sometime later, the wave front manager receives new phone instances from the phone book and place them in a fifo. The phone instances are taken from the fifo and written into the CAM and RAM at the next_empty_address. The empty_count is decremented by one. A match is run on the “occupy” flag, and the address is placed into next_empty_address register.

Both the phone level and the state level make use of a path merge unit to reduce the number of possible paths, by path merging. Figure 5 shows the use of link nodes to permit path merging. In the present embodiment, all word end phone instances are

DSMDB-#3348563-v1

linked to word start triphones with a link node. During each frame the path merge unit looks for *word end* tokens that can be recombined. This is done by looking for word end phones that have the same word_id and word history. When they are found a link is added using the path merge unit in the PINE array.

Figure 6 shows a diagram of the architecture at the Phone level, showing the path merge unit. The path merge unit includes a path merge CAM and a path merge RAM. The path merge CAM stores word_id, word history for two words, and an “occupy” field. The path merge RAM stores a link_node_id. A CAM RAM occupancy manager keeps track of the empty addresses in the CAM and RAM. The path merge unit is only passed information from the wave front manager if the phone is a *word end phone*. For each frame, there is a check to see if there is an identical word with the same history, and if there is, the link_node_id is passed to the path merge unit. If there isn’t an identical word with the same history, a new record is written into the Path merge CAM-RAM.

Figure 7 is a block diagram showing the Phone Book architecture. The Phone Book is tasked with generating the next phone models when given a pointer into the lexical tree. The key requirement for this block as to minimise the time between receiving a “lexical tree address” and generating the appropriate responses, in particular the next phone model instance(s).

The lexical tree is used to build the next triphones based on the current context.

Current triphone	Next triphone	comments
Left context = a Central context = b Right context = c	discarded Left context = b Central context = c Right context = ?	next_leftmph_id stored in wave front manager current LT_node_id stored in wave front manager must be looked up. Large (< 45) for new word start

The LT_node_id, next_left_mph_id and prev_phone_no are received at the phone generator from the wavefront manager. The LT_node_id is then used to address the lexical tree store and return the central monophone_id. One or more further accesses to memory are performed using the microcoded instructions, each memory access returning the right monophone_id.

A two stage lookup is accessed using the three monophone_ids (giving a triphone), the associated state_ids. The phone model is assembled, and returned to the PINE array. The wavefront manager is then updated by sending it relevant parts of the phone model.

The lexical tree store is a micro-coded ROM with both instruction and data fields. The lexical tree store holds a number of lexical trees. Each lexical tree may hold single words or multiple words. Each lexical tree comprises lexical tree elements, and each element has three fields. For now, the fields will be integers and will be converted to std_logic_vector later. The following is an example of the lexical tree store, with the LT data in word_id order.

ZERO	z i y r o w	19 8 12 11 0 0 0 0	sp 0
ONE	w a h n	18 2 10 0 0 0 0 0	sil 1
TWO	t u w	14 16 0 0 0 0 0 0	ah 2
THREE	th r i y	15 12 8 0 0 0 0 0	ay 3
FOUR	f o w r	6 11 12 0 0 0 0 0	eh 4
FIVE	f a y v	6 3 17 0 0 0 0 0	ey 5
SIX	s i h k s	13 7 9 13 0 0 0 0	f 6
SEVEN	s e h v i h n	13 4 17 7 10 0 0 0	ih 7
EIGHT	e y t	5 14 0 0 0 0 0 0	iy 8
NINE	n a y n	10 3 10 0 0 0 0 0	k 9
OH	o w	11 0 0 0 0 0 0 0	n 10
			ow 11
			r 12
			s 13
			t 14
EIGHT	e y t	5 14 0 0 0 0 0 0	th 15
FIVE	f a y v	6 3 17 0 0 0 0 0	uw 16
FOUR	f o w r	6 11 12 0 0 0 0 0	v 17
NINE	n a y n	10 3 10 0 0 0 0 0	w 18
OH	o w	11 0 0 0 0 0 0 0	z 19
SEVEN	s e h v i h n	13 4 17 7 10 0 0 0	
SIX	s i h k s	13 7 9 13 0 0 0 0	
TWO	t u w	14 16 0 0 0 0 0 0	
THREE	th r i y	15 12 8 0 0 0 0 0	
ONE	w a h n	18 2 10 0 0 0 0 0	
ZERO	z i y r o w	19 8 12 11 0 0 0 0	

The following is an example of the three fields of a lexical tree element. The first field is an instruction field, and the second and third fields are data fields.

Fields:	1st	2nd	3rd	
	instr	data	data	
	2b	6b	6b	
	00	MP	MP	- raw data field.
	01	br	br	- branch address.
	10	br	br	- branch address.
	11	word_id		- double field.

Note that there might be no difference between "01" and "10"

To build the "ey" tree, we only have to consider "EIGHT", because this is the only word in the lexical tree which starts with "ey". EIGHT is represented as "ey t", with phone_nos 5 14 0 0 0 0 0. The first lexical tree element stores the raw data specifying the phone_nos, the second lexical tree element is an end field, and the subsequent lexical tree element is set to zero.

```
0 00 5 14 - raw data
1 11 8 0 - end field
2 00 0 0
```

To build the "f" tree, we have:

```
FIVE f ay v 6 3 17 0 0 0 0 0
FOUR f ow r 6 11 12 0 0 0 0 0

0 01 4 0 -- branch forward four places as well
1 00 6 3 -- as the implied other brach of next (one).
2 00 3 17
3 11 5 0 -- all addresses are relative offsets.
4 00 6 11
5 00 11 12
6 11 4 0
7 00 0 0
```

To build the "n" tree, we have:

```
NINE n ay n 10 3 10 0 0 0 0 0

0 00 10 3
1 00 3 10
2 11 9 0
3 00 0 0
```

To build the "ow" tree, we have "ow" in both the first and second data fields, as no tree is going to be a single monophone long.

```
OH          ow          11 0 0 0 0 0 0 0
0 11 11 0
1 00 0 0
```

To build the "s" tree, we have:

```
SEVEN      s eh v ih n    13 4 17 7 10 0 0 0
SIX        s ih k s       13 7 9 13 0 0 0 0
```

```
0 01 6 0 -- branch forward six places.
1 00 13 4
2 00 4 17
3 00 17 7
4 00 7 10
5 11 7 0
6 00 13 7
7 00 7 9
8 00 9 13
9 11 6 0
10 00 0 0
```

To build the "t" tree, we have:

```
TWO   t uw      14 16 0 0 0 0 0 0
0 00 14 16
1 11 2 0
2 00 0 0
```

To build the "th" tree, we have:

```
THREE   th r iy  15 12 8 0 0 0 0 0
0 00 15 12
1 00 12 8
2 11 3 0
3 00 0 0
```

To build the "w" tree, we have:

```
ONE    w ah n    18 2 10 0 0 0 0 0
0 00 18 12
1 00 12 10
2 11 1 0
3 00 0 0
```

To build the "z" tree, we have:

```
ZERO ziyrow 19 8 12 11 0 0 0 0

0 00 19 8
1 00 8 12
2 00 12 11
3 11 0 0
4 00 0 0
```

Once all of the branches have been built, the next stage is to concatenate the tree. It appears that four elements make a page, and each element is two bytes.

		page
ey:	00 5 14 11 8 0 00 0 0 00 0 0	5
f:	01 4 0 00 6 3 00 3 17 11 5 0	6
	00 6 11 00 11 12 11 4 0 00 0 0	7
n:	00 10 3 00 3 10 11 9 0 00 0 0	8
ow:	11 11 0 00 0 0 00 0 0 00 0 0	9
s:	01 6 0 00 13 4 00 4 17 00 17 7	10
	00 7 10 11 7 0 00 13 7 00 7 9	11
	00 9 13 11 6 0 00 0 0 00 0 0	12
t:	00 14 16 11 2 0 00 0 0 00 0 0	13
th:	00 15 12 00 12 8 11 3 0 00 0 0	14
w:	00 18 12 00 12 10 11 1 0 00 0 0	15
z:	00 19 8 00 8 12 00 12 11 11 0 0	16

Then, the start_lookup header is defined. Each row (page)is 8 bytes and hence the header takes 3 rows. The end_lookup header is two rows. Therefore, the lexical tree store proper starts at the 6th (#5) page.

```
sp sil ah ay eh ey f ih      - MP
0 1 2 3 4 5 6 7              - MP_id
3 0 0 0 5 0 6 0              - data stored.

iy k n ow r s t th
8 9 10 11 12 13 14 15
0 0 8 9 0 10 13 14

uw v w z . . . .
16 17 18 19 . . . .
0 0 15 16 0 0 0 0
```

The end_lookup header is defined. 20 is out of bounds.

```
ey f n ow s t th w
5 6 10 11 13 14 15 18
```

```
z . . . . .
19 20 20 20 20 20 20 20
```

To put the above information all together:

```
START      sp:ih    3 0    0 0    5 0    6 0    0
LOOKUP     iy:th    0 0    8 9    0 10   13 14   1
           uw: v    0 0    15 16   0 0    0 0    2

END        ey:w     5 6    10 11   13 14   15 18   3
LOOKUP     z       19 20   20 20   20 20   20 20   4

LEXICAL    ey:      00 5 14  11 8 0  00 0 0  00 0 0   5
TREE       f:       01 4 0  00 6 3  00 3 17  11 5 0   6
STORE      00 6 11  00 11 12  11 4 0  00 0 0   7
           n:       00 10 3  00 3 10  11 9 0  00 0 0   8
           ow:      11 11 0  00 0 0  00 0 0  00 0 0   9
           s:       01 6 0  00 13 4  00 4 17  00 17 7  10
                  00 7 10  11 7 0  00 13 7  00 7 9   11
                  00 9 13  11 6 0  00 0 0  00 0 0   12
           t:       00 14 16  11 2 0  00 0 0  00 0 0   13
           th:      00 15 12  00 12 8  11 3 0  00 0 0   14
           w:       00 18 12  00 12 10  11 1 0  00 0 0   15
           z:       00 19 8  00 8 12  00 12 11  11 0 0   16
```

```
3 0    0 0    5 0    6 0
0 0    8 9    0 10   13 14
0 0    15 16   0 0    0 0
5 6    10 11   13 14   15 18
19 20   20 20   20 20   20 20
00 5 14  11 8 0  00 0 0  00 0 0
01 4 0  00 6 3  00 3 17  11 5 0
00 6 11  00 11 12  11 4 0  00 0 0
00 10 3  00 3 10  11 9 0  00 0 0
11 11 0  00 0 0  00 0 0  00 0 0
01 6 0  00 13 4  00 4 17  00 17 7
00 7 10  11 7 0  00 13 7  00 7 9
00 9 13  11 6 0  00 0 0  00 0 0
00 14 16  11 2 0  00 0 0  00 0 0
00 15 12  00 12 8  11 3 0  00 0 0
00 18 12  00 12 10  11 1 0  00 0 0
00 19 8  00 8 12  00 12 11  11 0 0
```

This information is converted to hexadecimal. The first five are byte fields.

The rest are:

2bit - uCode.

6bit - data.

2bit - null.

6bit - data

03 00	00 00	05 00	06 00
00 00	08 09	00 0A	0D 0E
00 00	0F 10	00 00	00 00
05 06	0A 0B	0D 0E	0F 12
13 14	14 14	14 14	14 14
05 0E	C8 00	00 00	00 00
84 00	06 03	03 11	C5 00
06 0B	0B 0C	C4 00	00 00
0A 03	03 0A	C9 09	00 00
CB 00	00 00	00 00	00 00
86 00	0D 04	04 11	11 07
07 0A	C7 00	0D 07	07 09
00 0D	C6 00	00 00	00 00
0E 10	C2 00	00 00	00 00
0F 0C	0C 08	C3 00	00 00
12 0C	0C 0A	C1 00	00 00
13 08	08 0C	0C 0B	C0 00

The above is the micro code for the digit lexical tree. It allows branching, and it holds the word_id. It knows which lexical trees it contains and where they are housed. It also stores the possible end monophone list.

Figures 8A-8B show a further example of the Lexical Tree RAM, giving examples of different types of branching in the lexical tree, and how these are represented in the RAM.

Figure 9 is a block diagram showing the data structure that is supported for the word link record, to enable word n-best decoding. The word engine maintains a record of the words that have been currently searched by the PINE array. It is used to determine the word history and allow trace back during or at the end of search. The word history is needed to determine the language model score and whether particular paths can be recombined or merged.

Figure 10 shows a block diagram of the word engine. This is designed to allow fast access to the word history (thus allowing rapid phone creation at the phone level), and to allow reasonably fast trace back. The Word Engine is primarily a memory store that fills incrementally. The decoupling of the wordlat_node_id and the physical address will allow the possibility of being able to keep the WLRs for a given word in contiguous addresses, thus avoiding the need for the pointer field.

The language model interface is designed to allow the Phone Book to quickly obtain the appropriate LM factor for the phone model that it is currently constructing. Based on the phones in the wave front, all possible next LM factors can be determined. This will be done in software together with the language model factor engine. The ARM 9 will maintain its own record of the wavefront using the same next phone requests received by the wave front manager and its own copy of the lexical tree.

Figure 11 shows the architecture of the LM interface. This allows the Phone book to quickly access the LM factor for the phone that is currently being generated. Prior to the phone generation phase LM_factors are computed for all possible phones that may be generated in the next frame.

The grammar engine is designed to give the Phone Book a quick response to requests for the next grammar node(s). In the case of a task grammar where the lexicon may be limited to a few words(e.g. yes, no), it may be desirable to use the same lexical tree. In this case it will be necessary for the grammar, to provide lexical tree pointers such that the phone generator can jump to the appropriate part of the lexical tree.

For a speech recognition system it is also necessary to specify the grammar. Again we will use HTK's Standard Lattice Format(SLF) to specify the grammar. In practice it will be convenient to use HTK's HParse format to specify the grammar using a high level grammar notation. This can then be converted into using the HParse tool. Consider having to specify a grammar for a digit recognition task. The HParse format for this is as follows:

```

$digit = one | two | three | four | five | six | seven | eight | nine | zero
$digit_sp = $digit sp
$noise = lipsmack | breath | sil
( [sil] < $digit_sp | $noise > [sil] )

```

where | denotes alternatives, < > denotes one or more repetitions, { } denotes zero or more repetitions, [] denotes that item is optional, and () contains the grammar.

Figure 12 shows the network that is defined by this grammar. The network starts and ends with a “sil” phone, representing silence. Between the start and end, a number of possible routes correspond to further silence, a breath, a lipsmack sound, or a detected word.

Figure 13 shows the grammar engine architecture. When a next phone request arrives, the next grammar node(s) must be retrieved from memory. In the case of a task grammar, the phone book should be given the lexical tree memory addresses to allow the phone to be determined.

Figure 14 shows a single frame of the Audio Codec (AC) link protocol used in some embodiments of the invention. The Advanced Audio CODEC Interface (AACI) is an ARM Primecell that provides an interface to off-chip AC97 audio codecs. AC97 is Intel’s audio codec specification, initially created for the PC market. The AACI is an AMBA slave block that connects to the Advanced Peripheral bus (APB). The AACI connects to the off-chip codec using the AC-link protocol.

The AC-link protocol is a bi-directional, fixed clock rate, serial digital stream. AC-link handles multiple input and output PCM audio streams as well as register control accesses. It employs a time division multiplexed (TDM) scheme that divides each audio frame into 12 outgoing and 12 incoming data streams each with up to 20 bit sample resolution. The tag slots are used to indicate which of the subsequent slots contain valid data. Add and dat are the control register address and data slots for read and writes. The adc slots contain the outputs from the analog to digital converters. Assuming that the

ADC sample rate is set to 16 kHz then the AC-link controller will receive 16 bits of data every third codec frame.

The AACI has 4 channels that can be operated simultaneously. They are accessed through DMA.

Figure 15 shows the system timing. Each frame is divided into three sections:

- Token update (MHD wash)
- First pass - inter phone token passing
- Second pass – garbage collection, max model pruning

The horizontal direction of the figure indicates elapsing time, and various subcircuits and processes are indicated along the vertical direction of the figure. The top part of the diagram shows frame count, and push to dictate button, and the lower part of the diagram shows various subcircuits and processes, within a single time frame, whilst the push to dictate button is engaged. These subcircuits and processes are language model interface, word engine, phone book, PINE array, MHD engine, mean & covar reg, and feature vector reg.

At the start of the time frame, the feature vector is generated, and the MHD values are calculated. The MHD values are then passed to the PINE array, which engages in beam pruning and token score updating. The phone book receives next phone requests and returns next phone responses. The word engine adds word link records, and the language model interface supplies language model factors.

In the second part of the frame, pruning thresholds are calculated for the PINE array, for the next frame, and the PINE array engages in token passing. The language model interface transfers language model factors for the next frame.

While the “push to dictate” button remains activated, the above process repeats in a sequence of frames.

The embodiments of the invention which are now described relate to a distance engine for calculation of Mahalanobis Distances.

Figure 16 shows a block diagram of a speech recognition system according to an embodiment of the invention. The system includes a microphone 100, which outputs an audio signal to a combined AGC (automatic gain control) and ADC (analogue to digital converter) 101. In alternative embodiments, the AGC and ADC may be separate. In this embodiment, the incoming data comes from a microphone, and is then converted to digital format. However, it is also possible that the digital audio data may be received from a network, or may be read from storage means such as a memory, an optical disk, or a hard disk drive.

The digital audio data output from the ADC is input to an audio data buffer 102, and may be held here until required. The data is then output from the audio data buffer 102 to a front end circuit 103. The incoming audio data is divided into frames, as described above. In cases where the system is extremely busy, it is possible to drop some audio frames, to allow the system to catch up with any backlog.

In the audio front end circuit 103, the audio is split into spectral components to generate a feature vector, as described above. For example, the feature vector may have a number of frequency bands, and an energy band, plus their first and second derivatives. The feature vector is then output from the front end 103 to the distance calculation engine 104.

The distance calculation engine 104 converts feature vectors into Mahalanobis distances. The acoustic model may be chosen from a plurality of acoustic models, e.g. according to the language used, and may be stored in a storage means 105 such as a memory, a disk drive, etc. The distances calculated by the distance calculation engine are then transferred to the search stage 106 of the speech processing circuit, which uses models such as one or more word models and/or language models to generate and output recognised text. Such word and language models may be stored in a separate storage means 107.

In one embodiment, the distance calculation engine uses Mahalanobis Distance (MHD) calculations to determine distance values for output to the search stage 106. In further embodiments, different types of distance calculations may be used, e.g as described above.

In some embodiments of the invention, the distance calculation engine is designed as a speech accelerator, to operate as a loosely bound co-processor for a CPU running speech recognition software. This has the advantage of reducing computational load on the CPU, and reducing memory bandwidth load for the CPU.

In some embodiments, the accelerator interface is memory-mapped. Other types of interface are also possible to support different system configurations, for example removable accelerator cards with SDIO (secure digital input/output), CardBus, or USB interfaces.

The speech recognition apparatus may also operate with a CPU and DSP (digital signal processor), where the DSP implements the "front-end" signal processing to produce a feature vector, and the CPU implements the search stage, reading the distance results from the distance calculation engine.

The system may be implemented in any combination of hardware or software.

In some embodiments of the invention the search stage 106 provides the distance calculation engine 104 with an active state list 108 that indicates the subset of states for which the search stage requires distances to be calculated by the distance calculation engine 104. This is an optimization that may reduce computation time and/or power consumption.

In preferred embodiments of the invention, the front end 103, distance calculation engine 104 and the search stage 106 operate in a pipelined manner. When operating in a pipelined manner it is unlikely that the search stage 106 will be able to provide the active state list 108 early enough for the distance calculation engine 104 to implement the optimization of computing only the distances that will be required by the search

DSMDB-#3348563-v1

stage 106. The distance calculation circuit 104 may calculate the MHD values for every state in the lexicon, per frame, whether it is subsequently required by the search stage or not. This allows the accelerator and software system to operate in a concurrent pipelined manner, which maximizes the system throughput.

The distance calculation engine 104 may have a plurality of memory areas for storing feature vectors, to allow a feature vector for one frame to be transferred to one memory area in the distance calculation engine 104, while another feature vector for another frame is being read from a second memory area, for calculation of a distance. The memory areas used for reading and writing may alternate with subsequent frames.

Figure 17 shows a block diagram of a distance calculation accelerator 200, used in an embodiment of the invention. A feature vector is received at input 212 of the accelerator 200. The feature vector is loaded into feature vector (FV) memory A 208 via a bus 211. In reality, there may be more than a single bus, but for simplicity, only one bus 211 is shown in the figure.

The registers 203 are discussed further at the end of this description. The registers 203 may include, for example, an accelerator status register, for indicating whether the acoustic model data is correctly stored in memory, as described in more detail below.

The FV registers 209 hold the feature vector whose distances are currently being computed by the distance calculation engine 204. If the distance calculation engine 204 is implemented in software, the FV registers might be implemented as memory attached to the processor running the software. If the distance calculation engine 204 is implemented in hardware the FV registers might be implemented as logic or as a memory block. The FV registers 209 are loaded from the FV memory 208 when the Distance Calculation Engine 204 has completed distance calculations for the current feature vector and a subsequent feature vector is available in the FV memory 208. In some embodiments there is an optional FIFO associated with FV registers 209 so that the FIFO is loaded from the FV memory 208 when there is a subsequent feature vector in the FV memory 208 and there is a space in the FIFO. In these embodiments the FV Registers 209 are loaded from the associated optional FIFO 209 when the Distance

DSMDB-#3348563-v1

Calculation Engine 204 has completed calculations for the current feature vector and there is a subsequent feature vector in the associated optional FIFO 209. The presence of the FIFO 209 provides for a looser coupling between the Front End 103 and the accelerator 200 which improves performance in some system configurations.

Each feature vector is transferred to a distance calculation engine circuit 204, to obtain distances for each state of the acoustic model. The acoustic model is stored in an acoustic model memory 205, which is connected to the bus 211, for control of reading/writing to the memory. In some embodiments, this also allows acoustic models to be changed or updated via the bus. The acoustic model may be stored in compressed form in the acoustic model memory 205. A decompressor 206 is provided on the data output of the acoustic model memory. The decompressor 206 can decompress the results as they are needed, and thus allows a much smaller size of acoustic memory to be used, than with uncompressed data. From the decompressor 206, the decompressed acoustic model states are sent to the distance calculation engine 204.

The distance calculation engine 204 converts feature vectors and acoustic model states to distances. The accelerator 200 includes two separate results memories for storing these calculated distances, labelled as “result memory A” 201 and “result memory B” 202.

The control, sequencing, and address generation 210 is logic, state machines, and/or software programs that provide overall control of the other elements of the accelerator, taking inputs from the registers 203, and all the other elements and interfaces to determine when to start and stop the Distance Calculation Engine 204. The control 210 also produces addresses and control signals to control the reading and writing of the memories 201, 202, 205 and 208, to control the flow of data between the various elements of the accelerator 200, the operation of the Decompressor 206, CRC check 207 and Distance Calculation Engine 204, and to control the setting or clearing or status bits in the Registers 203 and to control signalling to the front end 103 and search stage 106. The control 210 is implemented using any of the widely known and used methods of controlling and sequencing in digital systems including logic gates, state machines, and embedded software systems. The control 210 might be a centralized control, or it might

DSMDB-#3348563-v1

be distributed in some or all of the other elements of the accelerator 200, or some combination of these.

Each time a feature vector is loaded into the accelerator, the accelerator computes the distances for all states for that feature vector, and stores the results alternately in the A or B Results Memory. The results are written to one of the two memories 201, 202, while the other memory is available for reading. Thus, while the distance calculation engine 204 is storing the distances calculated for one audio frame, the other result memory may make available the distances calculated for an earlier audio frame, to the search stage of the speech recognition system. For non-real time processing, “earlier” audio frame may refer to an audio frame that has been processed earlier, rather than being limited to an audio frame that corresponds to speech at an earlier time.

Thus, the search stage software reads results alternatively from the A or B result memory, while the accelerator writes results into the other result memory, allowing concurrent pipelined operation of the accelerator and the CPU [and the DSP if present].

In some implementations the A and B results memories may be exposed to the software interface as separate memories appearing at different locations in the CPU and/or DSP memory map. Another implementation is to map each memory alternatively to the same range of addresses in the CPU and/or DSP memory map, so that the software always accesses results at the same location, and the memory containing the set of results the software currently needs to access is the one mapped there.

The accelerator may autonomously load and decompress an acoustic model, which is stored in a compressed format in the acoustic model memory 205. The Acoustic Model may be loaded into the Acoustic Model Memory by software running on the CPU prior to the first use of the Accelerator. The Acoustic Model Memory may be an on-chip memory, to maximize bandwidth and minimize power consumption of reading in the entire Acoustic Model every frame, which means 50 to 100 times per second, depending on the system design.

The Acoustic Model Memory may be a dedicated memory attached to, or incorporated within, the accelerator. Alternatively the Acoustic Model Memory may be a shared resource that is used by other system functions or applications when speech recognition is not running. To allow detection of situations where the acoustic model has been overwritten by another application, a pre-calculated CRC (cyclic redundancy check) signature may be stored with the acoustic model. Each time the acoustic model is read into the accelerator, a CRC may be computed and compared with the signature. If the comparison fails, an error bit is set in the accelerator status register.

Some embodiments of the invention relate to speech recognition systems comprising separate front end 103 and search stage 106 processors. These separate processors perform the front-end and search stages of the recognition process, with the accelerator 200 performing the distance calculations. Typically the front-end processing would be performed on a DSP and search stage would be performed on a CPU, although this description holds for any type of processors, even dedicated hardware implementations. As shown in figure 17, the front end 103 supplies the distance accelerator with feature vectors, and the search stage 106 uses distances calculated by the accelerator.

The accelerator is designed to operate in a concurrent pipelined manner with the other processors, allowing a high level of parallelism in the system.

Figure 18 shows how this pipelined operation works in an embodiment of the invention with ideal data flow, i.e. where each of the three recognition stages require exactly the same time to process each data frame, and the processors are never delayed, stalled, or diverted to other tasks.

The time axis is divided into equal “pipeline step times”. If the step time is the same as a frame time then the system is processing audio in real time. If the step time is short than a frame time, the system is operating faster than real time, which may be valuable for batch processing of audio data, or catching up after some type of delay. If the step time is longer than a frame time, the system is operating slower than real time, which may be acceptable in some applications, for example where the person dictating pauses occasionally allowing the system to “catch up”.

DSMDB-#3348563-v1

Once the pipeline has filled (which takes two step times), the DSP, the Accelerator, and the CPU are all operating in parallel. In this idealized example, the total time available for processing a frame is three step times – one step time in the DSP, one step time in the accelerator, and one step time in the search stage. There is a latency of three step times from the time the audio data enters the front end to the completion of the search stage for that frame, and the processing of one frame finishes each step time. If the system were not pipelined, then the processing of all stages for one frame would have to complete in one step time in order to achieve the same overall throughput of frames per second. Thus pipelining in this way can achieve three times the throughput with the same processing elements, compared to a non-pipelined system. A non-pipelined system would be one in which the processing of frame $n+1$ would not start until all stages of the processing of frame n was already complete.

If, for example, in a system running in real time with a frame time and step time of $1/100^{\text{th}}$ of a second, which is 10ms, each of the three processing elements has the entire 10ms available to complete the processing of one stage of the recognition task, giving 30ms of processing time overall for each frame. If the system were not pipelined, real time operation would require that all three processing elements would be complete in 10ms, allowing only 3.3ms for each of the three processing elements.

At the end of each pipeline step time, the DSP provides a feature vector to the Accelerator, and the accelerator provides a set of distance results to the CPU. When the CPU is performing the search stage for frame n , the Accelerator is computing the distances for frame $n+1$, and the DSP is performing the front-end stage for frame $n+2$. During each pipeline step, the front end processing consumes one frame of digital audio data (although this is not indicated explicitly on the diagram).

Real world systems are more complex than the idealized system in many ways. The processing time for a frame is often highly variable and data dependent, especially in the search stage, and the processing time for each processing stage is likely to be quite different. The DSP and CPU are likely from time to time to be diverted to run other software tasks, either to service interrupts, or under control of an operating system

DSMDB-#3348563-v1

scheduling regime. Communications between the processing elements may be stalled due to competing bus traffic, for example for DMA transfers.

Recognition software may implement “frame dropping” and other advanced techniques, and as a result not all frames are processed through all of the recognition stages. Processing elements cannot respond instantly to the availability of data from the preceding processing stage, resulting in delays.

Finite buffer space for information storage between stages means that an earlier stage may be stalled waiting for a space to store its output data when following processing stages are “falling behind”.

The accelerator has been carefully designed to mitigate these problems. Figure 19 shows a pipelined system with real world complications added. The following text describes the interactions shown in the diagram. The time for the processing of various stages has been varied to show difference examples of buffering of data between stages and stalling of processing elements. The numbering corresponds to the numbered bubbles on the diagram. The text also makes reference to memories used to pass information into and out of the accelerator, which are shown in figure 17.

The front end has computed the feature vector for the first frame, FV1. It is loaded into the accelerator, causing the accelerator to commence computation of the distances for the frame.

The feature vector for the second frame, FV2, is passed to the accelerator. The accelerator is still busy computing the distances for the first frame, so this feature vector is held in the accelerators FV memory.

The front end has computed the feature vector for the third frame, FV3, but it cannot be delivered to the accelerator as the FV memory is still full, holding FV2. The time period labelled “stall delivering FV3” is the period between FV3 being computed and when it can first be delivered. Of course the DSP could be diverted to some other useful task, or it could buffer FV3 in its own memory and commence computation of FV4.

DSMDB-#3348563-v1

The arrow labelled “FV3 earliest” shows the earliest time that the feature vector FV3 can be delivered to the accelerator, which is immediately after the accelerator loads FV2 into its internal registers and frees the memory.

The arrow labelled “FV3 latest” shows the latest time that the feature vector FV3 can be delivered to the accelerator without causing the accelerator to stall for lack of a feature vector to work on. This example shows that there is a wide tolerance in the system to when FV3 can be delivered without affecting system performance – an entire accelerator frame processing time. The accelerator provides an interrupt and a status bit that can be polled to indicate the state of the FV memory. Regardless of which control method the DSP uses, the system tolerates long latency without reducing performance.

The accelerator signals to the search stage that the distances for frame 1 are available in result memory A. The search stage commences processing for frame 1, reading distances from result memory A.

The search stage has finished its processing for frame 1 before the accelerator has completed computing the distances for frame 2, so the search stage stalls waiting for the accelerator. Of course the CPU could be diverted to other useful tasks, including any search stage tasks that can be completed before distances are available. Again, an interrupt and status bits are provided by the accelerator to control this interaction between the accelerator and the CPU. .

The accelerator completes the distance calculation for frame 2, and signals to the CPU that it can commence reading the distances from result memory B.

The accelerator has completed the computation of distances for frame 3 before the search stage has finished processing frame 2. As soon as the search stage signals to the accelerator that it has completed frame 2 by releasing memory B, the accelerator signals that the distances for frame 3 are available, allowing the search stage to immediately move on to processing stage 3.

The accelerator is stalled waiting for feature vector FV4, due to the long processing time in the front end. In real systems this may be caused by a number of factors, including the DSP being diverted to another task for a period of time, or simply due to the DSP performance being insufficient to keep up with the accelerator's throughput.

The accelerator restarts as soon as FV4 is delivered by the front end.

The search stage is stalled waiting for distances for frame 4, a follow on consequence of the late delivery of FV4 to the accelerator.

FV5 is delivered to the accelerator while it is still computing the distances for the previous frame, allowing it to move onto processing frame 5 without stalling when it completes frame 4.

The accelerator signals that the distances for frame 4 are available and the search stage commences processing frame 4.

FV6 is delivered to the accelerator while it is still computing the distances for frame 5.

The availability of distances for frame 5 is signalled by the accelerator to the CPU.

Even though the accelerator has already received FV6, it is stalled waiting for a free result memory it can write the distances into.

The search stage releases result memory B at the end of its processing of frame 4. This allows the accelerator to start computing distances for frame 6.

The accelerator completes computing distances for frame 6 and signals to the CPU that the results are ready.

Further embodiments of the invention relate to systems with a Front End and Search Stage on a single processor. This single processor performs the front-end and search

stages of the recognition process, with the accelerator performing the distance calculations. This configuration is likely to be the most common configuration where the accelerator is added to an embedded system such as a PDA or mobile phone that has a high-performance embedded application processor (but no DSP) available to run speech recognition software.

In this configuration the accelerator still operates in a concurrent pipelined manner with the front-end and search stages; it is just that the front-end and search stages are running on a single processor.

CPUs in embedded systems are generally relatively simple compared to desktop PCs and servers. Embedded CPUs in mobile phones, PDAs, in-car navigation and telematics systems, and other consumer electronics devices are typically “simple uni-processors” by which we mean not multi-processors and not supporting hardware multi-threading support – i.e. they can only execute a single thread of instructions (although they can be diverted to a different thread under the control of interrupts, exceptions, or an operating system). For this discussion, except where explicitly stated otherwise, we are talking about embedded processors that are “simple uni-processors”. We are also using the term “CPU” and “processor” interchangeably.

Figure 20 shows how the system can be pipelined with a single embedded processor working with the accelerator, in an idealized system. In this idealized system, the time taken for the processor to perform the front-end and search stage processing is identical to the time the accelerator takes to perform the distance calculations, and the processor is never delayed, stalled, or diverted to other tasks.

The time axis is divided into equal “pipeline step” times. If the step time is the same as a frame time then the system is processing audio in real time. If the step time is short than a frame time, the system is operating faster than real time, which may be valuable for batch processing of audio data, or catching up after some type of delay. If the step time is longer than a frame time, the system is operating slower than real time, which may be acceptable in some applications, for example where the person dictating pauses occasionally allowing the system to “catch up”.

DSMDB-#3348563-v1

The first two pipeline step times are used to fill the pipeline. By the third pipeline step time, the pipeline is full, and all three stages of the recognition process are processed every pipeline step time, with each stage working on the processing of a different data frame.

During the first pipeline step time, the CPU completes the front-end processing for frame 1, and the feature vector passed to the accelerator so that distance calculations can start in the accelerator.

During the second pipeline step time, the front-end processing for frame 2 is completed by the CPU, in parallel with the accelerator computing the distances for frame 1.

During the third pipeline step time, the CPU runs the search stage for frame 1 reading distance results from the accelerator, and then runs the front-end processing for frame 3. The accelerator runs in parallel computing the distances for frame 2 using the feature vectors computed in the previous pipeline step.

Each subsequent pipeline step time is similar to the third pipeline step, with the frame numbers incremented. So for pipeline step $n = 4, 5, 6$, etc. the CPU runs the search stage for frame $n-2$ and then runs the front-end for frame n . In parallel the accelerator computes the distances for frame $n-1$.

Note that with this situation the production of feature vectors and the consumptions of results run in lock step - one feature vector is produced every time the search stage consumes one frame of distance results. This considerably reduces the need for elastic buffering between the distance accelerator and the software system.

Again, real world systems are more complex than the idealized system in many ways. For example, the processing time for a frame is often highly variable and data dependent, especially in the search stage, and the processing time for each processing stage is likely to be quite different. The CPU is likely from time to time to be diverted to run other software tasks, either to service interrupts, or under control of an operating

DSMDB-#3348563-v1

system scheduling régime. Communications between the CPU and accelerator may be stalled due to competing bus traffic, for example for DMA transfers.

Recognition software may implement “frame dropping” and other advanced techniques, and as a result not all frames are processed through all of the recognition stages. Processing elements cannot respond instantly to the availability of data from the preceding processing stage, resulting in delays.

Finite buffer space for information storage between stages means that an earlier stage may be stalled waiting for a space to store its output data when following processing stages are “falling behind”.

The accelerator has been carefully designed to mitigate these problems. Figure 21 shows a pipelined system with real world complications added. The following text describes the interactions shown in the diagram. The time for the processing of various stages has been varied to show difference examples of buffering of data between stages and stalling of processing elements. The numbering corresponds to the numbered bubbles on the diagram. The text also makes reference to memories used to pass information into and out of the accelerator, which are shown in figure 21.

At the end of front-end processing for frame 1, the CPU passes the feature vectors for frame 1 to the accelerator, allowing distance calculations to start.

To fill the pipeline, the CPU immediately performs the front-end processing for frame 2 and passes the feature vectors to the accelerator. They are stored in the accelerator so that the accelerator can load them into its internal engine for processing as soon as the distance calculations for frame 1 are complete.

The CPU is now ready to start the search stage for frame 1, however the distance computation in the accelerator has not completed. The box labeled “Prep search stage 1” indicates a period where the CPU could do any preparatory search stage processing that does not require distances. Alternatively the CPU could be diverted to another task, or it can stall waiting for the distance results.

DSMDB-#3348563-v1

The accelerator signals that the distance results for frame 1 are now available. The search stage can commence on the CPU, reading distance results from the accelerator's Result Memory A. The accelerator automatically starts processing the feature vector for frame 2 that was delivered earlier.

The accelerator signals to the CPU that it has completed the distance calculations for frame 2. The CPU is still busy running the search stage for frame 1, and has still to run the front end for frame 3, so the completion signal is ignored until the CPU has need of it, when it starts running the search stage for frame 2. As the accelerator has not received the feature vector for frame 3, it has no task it can perform and so it stalls.

The CPU delivers the feature vector for frame 3, restarting the accelerator, which writes results into Result Memory A. Note that Result Memory A was previously "released" at the end of the processing of search stage 1, informing the accelerator that it can overwrite the old results held in that memory with new results. The CPU then runs the search stage for frame 2, reading distance results from the accelerator's Result Memory B.

The accelerator signals to the CPU that the distance results for frame 3 are complete in Result Memory A. Again, the CPU ignores this signal until it is ready to commence the search stage for frame 3, and the accelerator stalls waiting for the next feature vector.

The CPU delivers the feature vector for frame 4, restarting the accelerator. It then starts search stage 3 reading distances from the accelerator's result memory A.

Search stage 3 and front end 5 were completed quickly, so that the feature vector for frame 5 is delivered to the accelerator while it is still running. The CPU performs all the preparatory work it can for the search stage, and then stalls waiting for distances to be available.

The accelerator signals that the distances for frame 4 are available. The CPU starts search stage 4, reading the distances from the accelerator's result memory B. The

DSMDB-#3348563-v1

accelerator automatically starts computing distances for frame 5 using the feature vector delivered earlier.

Note that with the pipelining scheme just presented which alternates front-end and search stage processing, the accelerator can never stall waiting for a result memory to be released because the feature vector the accelerator needs as an input is not produced until after the memory has been cleared. The CPU task schedule could be described with the shorthand F1 F2 S1 F3 S2 F4 S3 F5 S4.

It can be seen readily from figure 21 and the accompanying description above, that with this CPU task schedule, that the operation of the CPU and accelerator are tightly coupled and whenever the processing time for a pipeline step on the CPU is unequal to the processing time on the accelerator, then it quickly results in one of them stalling.

In a system where either the CPU or the accelerator is always (or at least in most instances) slower than the other in completing its processing task, stalling is inevitable and throughput may not be improved by adding additional buffering (e.g. additional result memories or FIFO depth on the feature vector interface). [Unless there are other difficulties to mitigate such as very low-bandwidth buses linking the CPU to the accelerator, or competing bus traffic such as DMA transfers that can significantly reduce performance.

If the processing time for a pipeline step is highly variable on either the CPU or Accelerator, such that there are regular changes in which one of them takes the longer time, then significant enhancements in throughput may be obtained by adding additional buffering between them. This may have the disadvantage of adding additional latency through the system. This buffering may be implemented by adding additional depth to the Feature Vector FIFO, and/or adding additional Result Memories. Alternatively it may be by implementing elastic buffers in software.

If the accelerator has sufficient buffering on the feature vector interface, e.g. by including the optional FIFO shown in figure 21, then increasing the pipeline depth by computing extra front end frames up front will improve throughput at a cost of extra

DSMDB-#3348563-v1

latency. This will result in a CPU task schedule of F1 F2 F3 S1 F4 S2 F5 S3 F6 S4 or F1 F2 F3 F4 S1 F5 S2 F6 S3 F7 S4, etc.

Of course it would also be possible to dynamically schedule whether the CPU should run the front-end or search stage code based on the availability, or not, of distance results and/or the space available to store more feature vectors and distance results. Given the fundamental relationship that one set of distance results is produced by the accelerator for every feature vector it consumes, and the restriction stated at the start of this section that we are discussing single-threaded CPUs, dynamic scheduling is likely to add little performance improvement while being complex to implement.

On a CPU supporting multi-threaded operation (which is a technique for improving performance in the presence of high memory system latency), or on a multi-processor, the highest system performance is likely to be achieved by running the front-end and search stages concurrently as separate threads, rather than alternating between the front-end and search stages. The system operation is then very similar to that described above, if each thread is thought of as a separate virtual processor. In some embodiments of the invention, elastic buffers may be used between stages, to accommodate varying time delays between the processing of frames at each stage. Elastic buffers in the interfaces between the three recognition stages may also significantly enhance performance of such a system.

Operation of the accelerator and the software system is not constrained to a rigid repetitive frame time. Instead the timing of the system is controlled dynamically by the availability of feature vectors into the accelerator, and the consumption of results out of the accelerator.

This allows the recognition process to fall behind real-time operation and to "catch up" again. It also allows the system to be robust in complex systems where interrupts, DMA bus traffic, memory caches, operating system task switching, and competing real-time software tasks all make guaranteeing rigid timing constraints impractical.

Elastic buffers at the major data transfer points between processing elements maximize system performance by allowing one element to continue useful processing whenever it can, regardless of whether other processing elements have been stalled or diverted to other tasks. Elastic buffers are useful for the following applications:

- (1) For buffering raw audio data arriving in real time from the microphone(s). The data would typically be moved into the buffer by a DMA and/or high priority interrupt driven software process to ensure that hard real time constraints are met.
- (2) For buffering feature vectors output from the front end. This is especially true if the front-end is implemented on a DSP or other processing element independent of the search stage.
- (3) For buffering distance results from the accelerator. Again this is especially true if the front-end and search stages are implemented on independent processing elements. This buffering is also especially valuable if the interface between the search stage processor and the accelerator is very slow, has high latency, or is subject to competing bus traffic.

The Feature Vector and Results interfaces on the accelerator already provide a level of elasticity that should be sufficient in systems where the Front End and Search stages are running on the same CPU, and the accelerator has a high-bandwidth interface to the CPU with low level of competition from other bus traffic such as DMA.

Other embodiments may have deeper FIFOs. In one embodiment, the two result memories are an implementation of a FIFO where the implementation is exposed across the interface. It would be possible to have additional memories and use them in a round-robin fashion, or to use another implementation of a FIFO that hides the FIFO depth by only exposing one result memory at a time to the interface. An example of this would be to map each memory in turn to a memory range that is always used to read results, so that at any one time the “current” result memory is mapped to that location. This would hide the FIFO depth behind the interface so that the pipeline depth can be changed without changing the software (or hardware) that accesses the interface.

DSMDB-#3348563-v1

Slow or Conflicted Bus Interfaces may occur. In the system configurations described above, the search stage reads distances directly from the Accelerator's Result Memories. The result memories allow the search stage to randomly access the distance results, as they are needed. This will only give a high performance system if the interface or bus between the CPU and Accelerator provides the CPU with low latency, high bandwidth access to the Accelerator Result Memories.

In some systems it is not practical to provide such a low latency, high bandwidth interface between the CPU and the Accelerator. This may be for many reasons, including:

- The interface is a bus shared with other devices and/or processors that generate bus traffic that keeps the bus busy such that the CPU access to the accelerator is delayed waiting for the bus to be free. DMA transfers are one example of conflicting bus traffic.
- The interface is implemented as a serial bus, or narrow parallel bus, for cost or compatibility reasons, yielding low bandwidth and high latency
- The interface crosses an industry standard interface such as PCI, USB, SDIO, Cardbus, Firewire, PCIExpress etc. that introduces protocol overheads and/or latency as would be the case where the accelerator is in a plug-in peripheral to a system containing a CPU (or CPU and DSP).

In these situations it is possible to add another pipeline stage during which the result data is copied from the Accelerator into another memory (which we will call a "buffer memory") to which the CPU does have high bandwidth, low latency access. This copying can be performed by a DMA controller, or by software running as an interrupt handler, or a separate software thread on the CPU, or a software thread on another processor, or by other commonly known means. The device controlling the transfer may be on the CPU side of the interface or on the accelerator side of the interface, depending on the capabilities of the interface. This extra pipeline stage then allows an entire pipeline step time to copy the data from the accelerator to the other memory. The data copy can optionally be performed as a sequential burst, or a number of sequential bursts, to optimize transfers across the interface between the Accelerator and the buffer

DSMDB-#3348563-v1

memory. The search stage, running on the CPU, can then randomly access the results in the buffer memory with low latency.

Similarly an extra pipeline stage can be added for transferring the Feature Vectors to the Accelerator, although given that the feature vectors are a much smaller object than the distance results, and that they are not accessed randomly across the interface but delivered once to the Accelerator for each frame, there are fewer situations where the performance benefit from this extra pipeline stage is as significant.

Figures 22 and 23 show idealized systems with the two extra pipeline stages for transfers added. The first shows a system with a CPU, DSP, and Accelerator with the two extra pipeline stages. The second shows a system with a CPU and DSP with the two pipeline stages added.

Other embodiments may add additional pipeline stages so that the distance computations are performed over two or more pipeline stages rather than being restricted to a single pipeline stage. This allows more time for the computation of the distance results for each frame, while maintaining the same throughput of distance results. This is similar in concept to the well-known pipelining techniques used to implement RISC microprocessors.

The Acoustic Models may occupy many megabytes of storage space. In embedded systems they will typically be held in a non-volatile memory such as FLASH memory, or other types of non-volatile memory as is the appropriate currently economic technology. The Accelerator may read the acoustic model directly from the non-volatile memory, or more likely the acoustic model will be copied from the non-volatile memory into a higher bandwidth RAM memory of some type (e.g. SRAM or DRAM, SDRAM, etc.). The RAM memory may be part of the same integrated circuit as the Accelerator (i.e. “on chip”), or attached to the integrated circuit containing the Accelerator. (“off chip”). The having the RAM memory on-chip is particularly advantageous, as this configuration would give the lowest power consumption and highest bandwidth. An alternative solution would be for the acoustic model to be stored on a disk drive, or another device accessible via a wired or wireless bus or network, and for the acoustic

DSMDB-#3348563-v1

model to be loaded into the RAM memory from the disk drive, or over the wired or wireless bus or network.

On-chip RAM memory is however likely to come at a cost-premium to off-chip commodity memory devices. Many embedded systems face extreme cost pressures in the market place, especially high volume devices supplied by many manufacturers such as mobile phones. For a speech recognizer to be incorporated into such systems, the overall recognizer solution must be delivered at very low cost. There is therefore a significant economic benefit from reducing the cost of the RAM memory and the non-volatile memory. One way to reduce the cost of these memories is to compress the acoustic models and store the compressed model in the memory.

The Accelerator computes all of the distances associated with each Feature Vector that is delivered to it by the front end. This means that the accelerator must use the data from the entire Acoustic Model each time a new Feature Vector is presented to the Accelerator (with the exception of some specific optimizations that may determine that some data values need not be used in the distance computations). It is therefore possible to arrange for the accelerator to read through the acoustic data sequentially once for each Feature Vector, using each data value as it read from the Acoustic Model.

Reading through the entire Acoustic Model sequentially is ideally suited to placing a decompressor between the distance calculation engine and the RAM memory, because many compression schemes depend on the decompressor reading sequentially through the compressed data, and do not readily support random access into the compressed data.

Thus this configuration allows storing compressed data in the RAM memory (and the non-volatile memory), which can therefore be smaller and lower cost. There is no requirement for a memory that is large enough to hold the entire decompressed acoustic model, as the decompressed data is consumed as it is decompressed.

The decompression scheme could be any combination of:

DSMDB-#3348563-v1

- sign or zero extension or otherwise conversion of narrow or variable width data to a wider data format.
- sign or zero extension or otherwise conversion of narrow or variable width data to IEEE standard single or double precision floating point format
- codebook decompression of a binary bitstream, where the codebook is stored as part of the acoustic model data
- decompression of a Huffman or Lempel-Ziv compressed stream
- decompression of run length encoded data
- decompression of difference encoded data
- any well known compression scheme.

Additionally the model size can be reduced by using *subspace distribution clustering* and the decompressor and distance calculation engine designed to operate with such models.

The following description provides further information on the visible registers in some embodiments of the invention.

The register bank is designed to allow simple connection with an 8-bit data bus. Accesses to the register bank should be made with byte-read and byte-write accesses. To facilitate this, only the least significant 8 bits of the data bus are used.

The address decoder should respond to byte accesses to byte 0 or byte 3 within the designated word. This way the same code can be used on an ARM processor regardless of whether the processor is configured for little endian or big endian operation.

REGBASE=Base address of register bank

Register 0 Product Version

8-bit Product version ID

This register value is changed when the functionality of the hardware changes in a way that changes the hardware/software interface.

Address = REGBASE (and REGBASE+3 for big endian)

Read only. Reads with value 0x01

Writes ignored

Register 1 Implementation Version

8-bit Implementation version ID

This register value is changed for each implementation to facilitate bug tracking and version control.

Address = REGBASE+4 (and REGBASE+4+3 for big endian)

Read only. Reads with value 0x01

Writes ignored

Register 2 Interrupt Enable

8-bit Interrupt Enable register

Address = REGBASE+4*2 (and REGBASE+4*2+3 for big endian)

Readable and writeable.

Value following reset is 0 for bits 0,1,2,3

Bit 0 I 1 = Enable FV memory empty interrupt on Int0 output.

Bit 1 J 1 = Enable Results complete interrupt on Int0 output.

Bit 2 K 1 = Enable FV memory empty interrupt on Int1 output.

Bit 3 L 1 = Enable Results complete interrupt on Int1 output.

Bit 4 Future use. Read value is not defined. Writes ignored.

Bit 5 Future use. Read value is not defined. Writes ignored.

Bit 6 Future use. Read value is not defined. Writes ignored.

Bit 7 Future use. Read value is not defined. Writes ignored.

Software should use a read-modify-write code sequence when setting this register so that future use bits are written with the value read from them.

Register 3 Status

Address = REGBASE+4*3 (and REGBASE+4*3+3 for big endian)

Read-only register. Writes are ignored.

Value following reset is 0 for bits 0,1,2,3

Bit 0 FVMB Feature Vector Memory is Busy

0 = OK to write new value to FV memory (FV memory is empty)

1 = Don't write new value to FV memory (FV memory is busy)

Bit 1 CRM Current Result Memory

0 = Result memory A

1 = Result memory B

This bit toggles every time the Release Result Memory (RRM) command is issued to the accelerator (register 3). It indicates which memory software should read results from, and which result memory the RRM command should release next time it is issued. Note: Software should also check that results are available in the indicated result memory by checking the value of the MA and MB bits.

Bit 2 MA 1=Results are available is result memory A

The MA bit is set to 1 when the compute engine completes writing all of the distances for a frame into result memory A. This value is set to zero when the Release Result Memory command is issued and CRM==0.

Bit 3 MB 1= Results are available is result memory B

The MB bit is set to 1 when the compute engine completes writing all of the distances for a frame into result memory B. This value is set to zero when the Release Result Memory command is issued and CRM==1.

Bit 4 Dfault 1=Accoustic data CRC fault. Indicates that the CRC value at the end of the acoustic data did not match the CRC value computed while the acoustic data was read in.

Bit 5 Future use. Value undefined.

Bit 6 Future use. Value undefined.

Bit 7 Future use. Value undefined.

Register 4 Command

Address = REGBASE+4*4 (and REGBASE+4*4+3 for big endian)

Write-only register.

Software should perform byte writes to this register with a single bit set to 1 and all other bits set to zero to issue a command to the accelerator.

Behaviour is not defined if more than one command is issued within a single write (i.e. if more than 1 bit is a 1).

Value on read is not defined.

Bit 0 LFV Load Feature Vector from the FV memory.

Tells the accelerator that a new feature vector has been placed into FV memory ready for use by the accelerator. The FV will be loaded into the compute engine when the compute engine completes its current task (or immediately if the compute engine is idle). Computation of distances for this FV will commence when the FV is loaded AND there is a released result memory. (At reset, both result memories are released). The results for the first, third, fifth etc. feature vectors loaded are written to Result Memory A. The results for the second, fourth, sixth etc. feature vectors loaded are written to Result Memory B.

Bit 1 RRM Release Result memory. Indicates that the results in the current result memory can now be overwritten with new results. The current result memory is indicated by the CRM bit in the status register.

Bit 2 CFVI Clear the FV memory empty interrupt

Bit 3 CRI Clear the Results Complete interrupt

Bit 4 Future use – write 0

Bit 5 Future use – write 0

Bit 6 Future use – write 0

Bit 7 Future use – write 0

The present invention may be included in a mobile phone, or in another consumer electronics device, a PC, Laptop or tablet PC, in-car electronics system, or plug-in card for an electronic device.

The methods and apparatus according to the invention are not limited to embedded mobile systems, but may be used with any type of computing device, including PCs, workstations, laptop computers, notebook computers, handheld computing devices, dictation machines, PDAs (personal digital assistants), personal organisers, mobile phones, games consoles, video players, etc. Embedded computers according to the present invention may be embedded in a wide range of apparatus and machinery, such as electronic entertainment products, domestic appliances, factory machinery, vehicles, etc.

Embodiments of the present invention may be implemented as an IP core in a System on Chip (SOC) or a discrete device. Multiple chips or bare die may be connected to provide a larger vocabulary system.

Although the apparatus may include all or some of an input for accepting audio input, a circuit for digitising the audio input, a circuit for dividing the audio input into a plurality of time-frames, and a circuit for analysing the spectral components of each time frame to characterise the audio in that frame, this is not essential. The apparatus may also include a circuit for calculating a score indicating the probability or likelihood that each audio frame corresponds to each of a plurality of different possible states, but again, this is not essential. Instead, the apparatus may include an input for receiving the pre-calculated scores.

In the described embodiments, n-gram language models are used. For example, the Bigram language model is based on outputting the probability that a given second word directly follows a given first word. However, it is possible to use alternative language models instead. It is possible to use a mixture of different models. For example, it is possible to run a word-dependent n-best decode applying a bigram language model, and the resulting word link record or word graph could then be rescored using a trigram language model.

DSMDB-#3348563-v1

The Viterbi beam search algorithm may be used in embodiments of the invention, to perform the search. A word level search may be performed. Dummy states may be used for the start and end of an utterance

The grammar level model may be used to add distance elements into the likelihood calculation. This is not essential, but tends to give much better results.

The HMM models described in the embodiments comprise self connections to each states and single connections to a neighbouring state. However, the present invention is not limited to such HMM models, and for example, HMM models with additional connections between states may also be used. Alternatively, embodiments of the invention may use alternative types of acoustic models, rather than HMM models.

The system may comprise a plurality of language models (e.g. English, French). A model of words may comprise actual words, phrases, silence, lipsmack sounds, and/or other sounds for recognition by the apparatus.

As previously discussed, the phone level has been found to be the best level for lexical tree searching, rather than the state level or the word level. However, this invention is not limited solely to phone level searching.

Pipelining may comprises processing data in different parts of a circuit at the same time, i.e. parallel processing.

In the following claims, the calculation means may comprise the distance calculation engine described above.

Further embodiments of the invention include the methods carried out in the accelerator, and the methods carried out in the speech recognition circuit.

While the invention has been described in terms of what are at present its preferred embodiments, it will be apparent to those skilled in the art that various changes can be

DSMDB-#3348563-v1

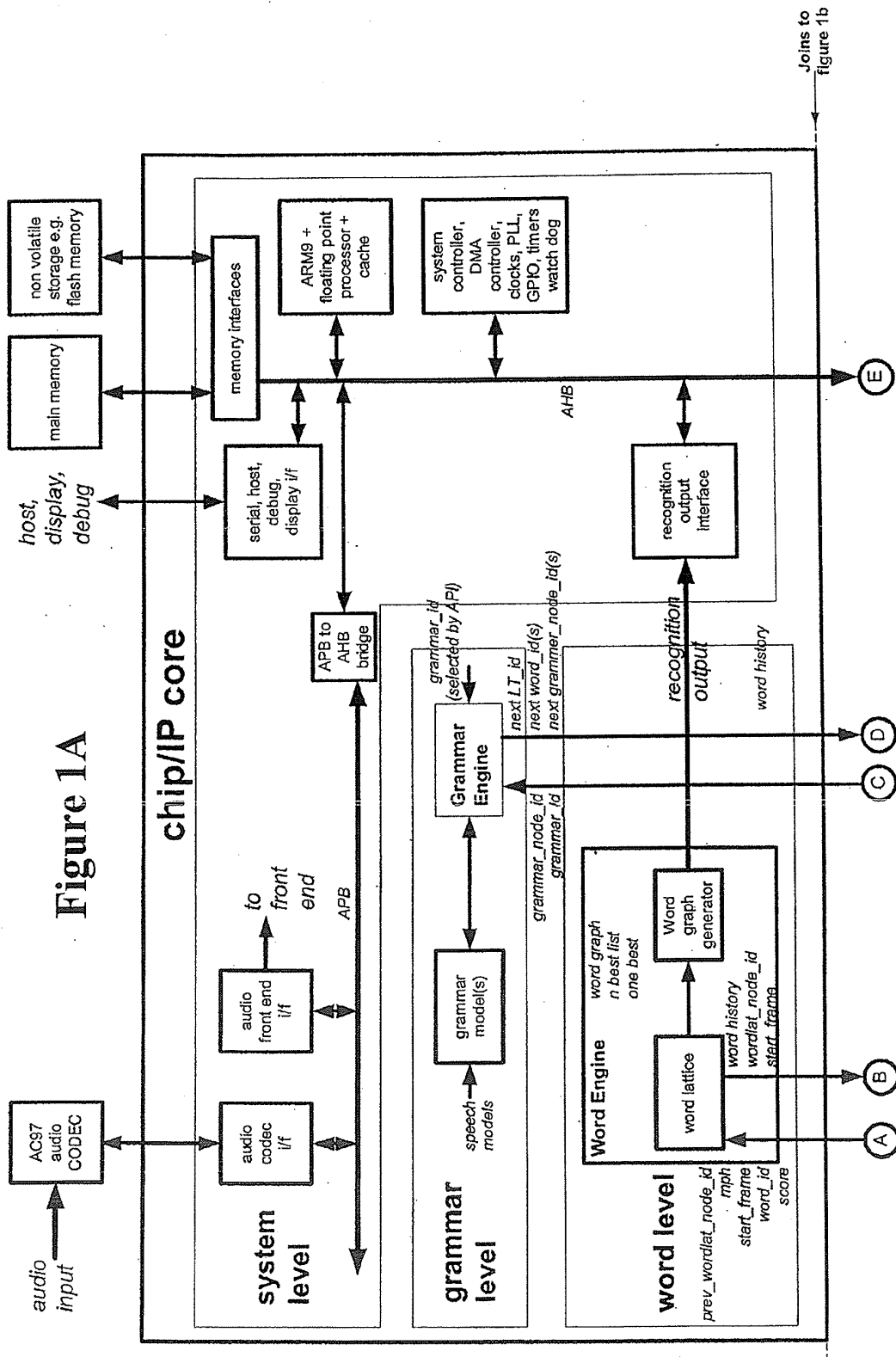
made to the preferred embodiments without departing from the scope of the invention,
which is defined by the claims.

WHAT IS CLAIMED IS:

1. A speech recognition circuit, comprising:
 - an audio front end for calculating a feature vector from an audio signal, wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame;
 - a calculating circuit for calculating distances indicating the similarity between a feature vector and a plurality of predetermined acoustic states of an acoustic model; and
 - a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words;wherein said audio front end and said search stage are implemented using a first processor, and said calculating circuit is implemented using a second processor, and wherein data is pipelined from the front end to the calculating circuit to the search stage.

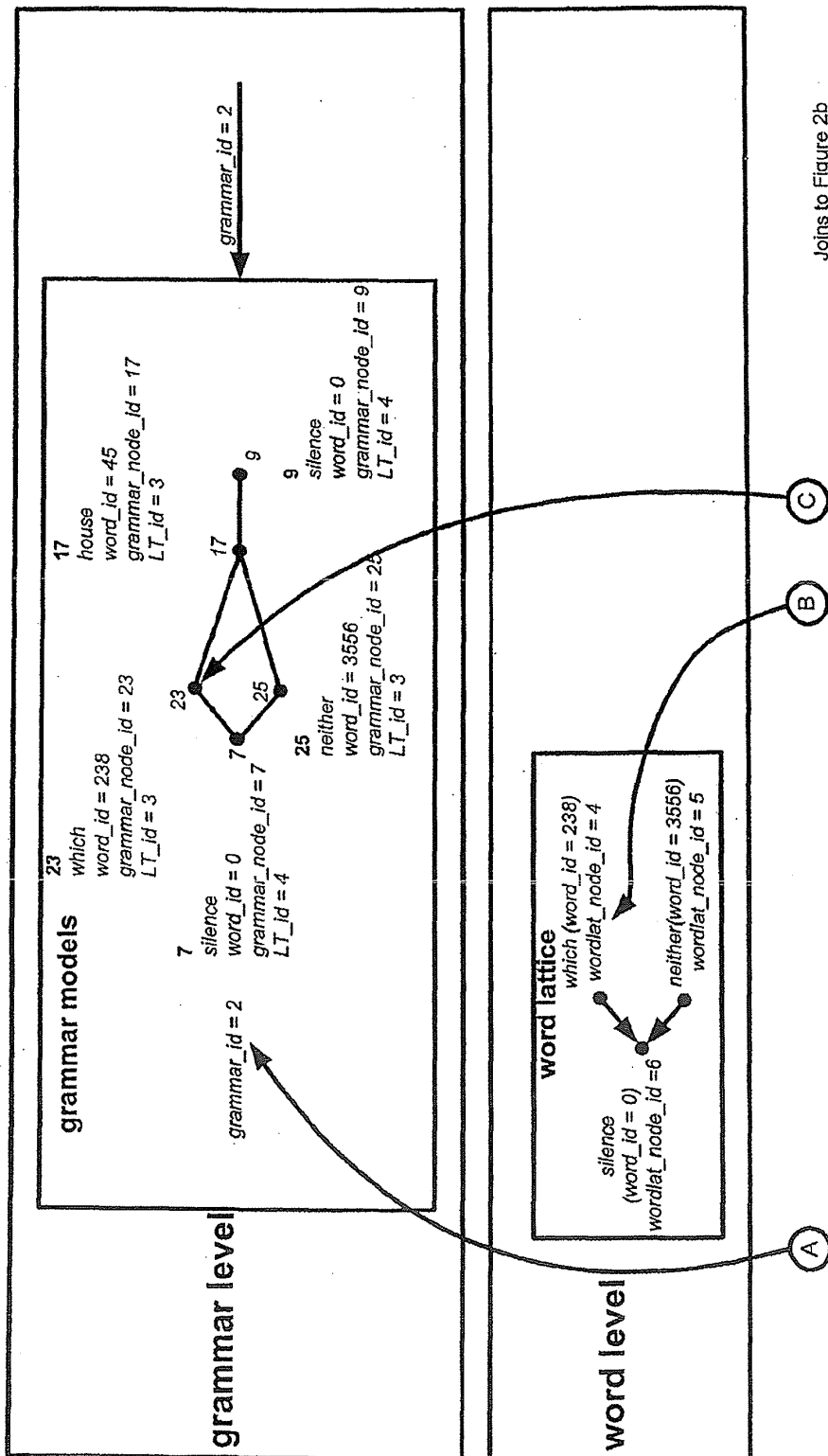
ABSTRACT

A speech recognition circuit comprising a circuit for providing state identifiers which identify states corresponding to nodes or groups of adjacent nodes in a lexical tree, and for providing scores corresponding to said state identifiers, the lexical tree comprising a model of words. The circuit includes: a memory structure for receiving and storing state identifiers identified by a node identifier identifying a node or group of adjacent nodes, the memory structure being adapted to allow lookup to identify particular state identifiers, reading of the scores corresponding to the state identifiers, and writing back of the scores to the memory structure after modification of the scores; an accumulator for receiving score updates corresponding to particular state identifiers from a score update generating circuit which generates the score updates using audio input, for receiving scores from the memory structure, and for modifying said scores by adding said score updates to said scores; and a selector circuit for selecting at least one node or group of adjacent nodes of the lexical tree according to said scores.



Joins to figure 1a





Joins to Figure 2b

Figure 2A

Figure 3A

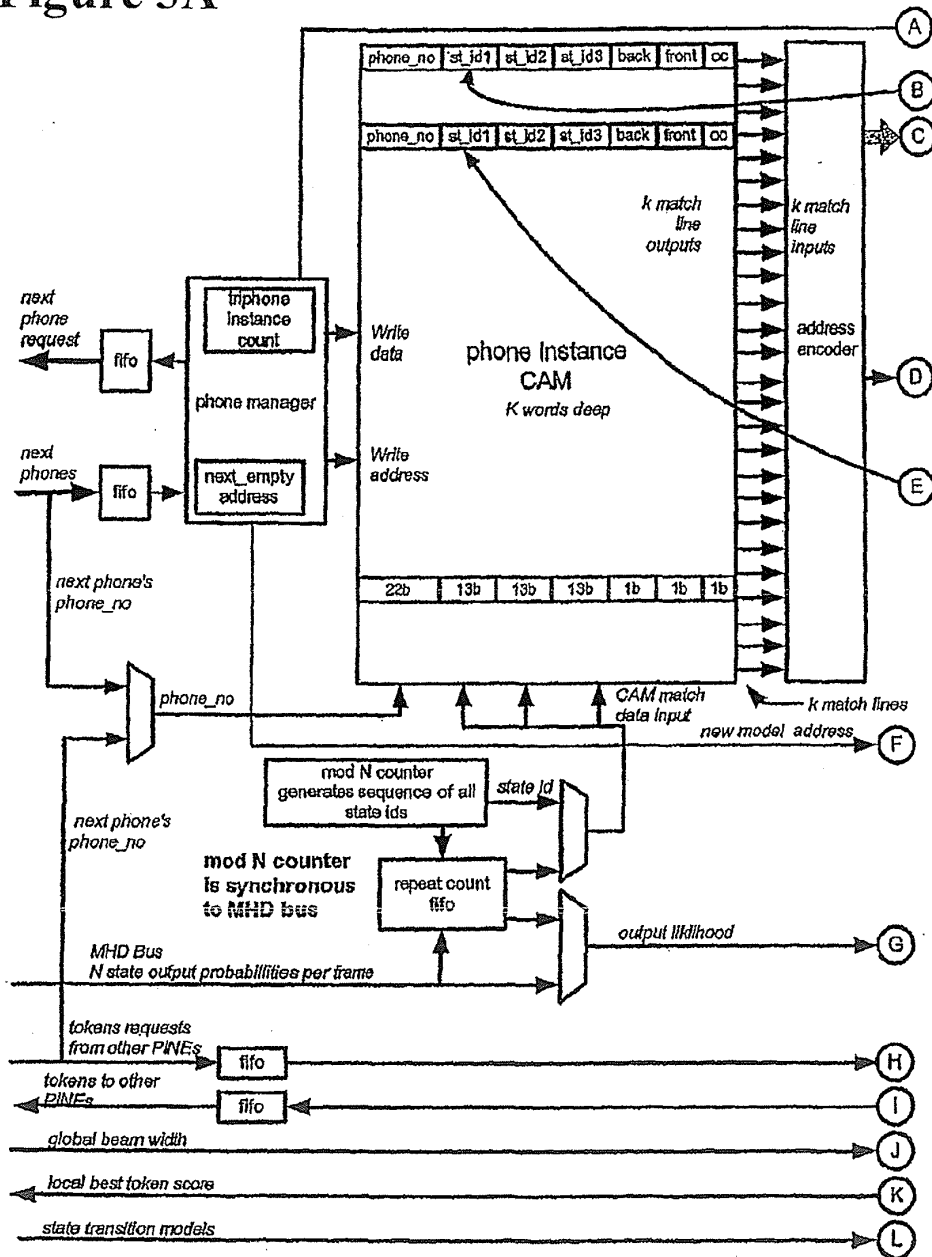
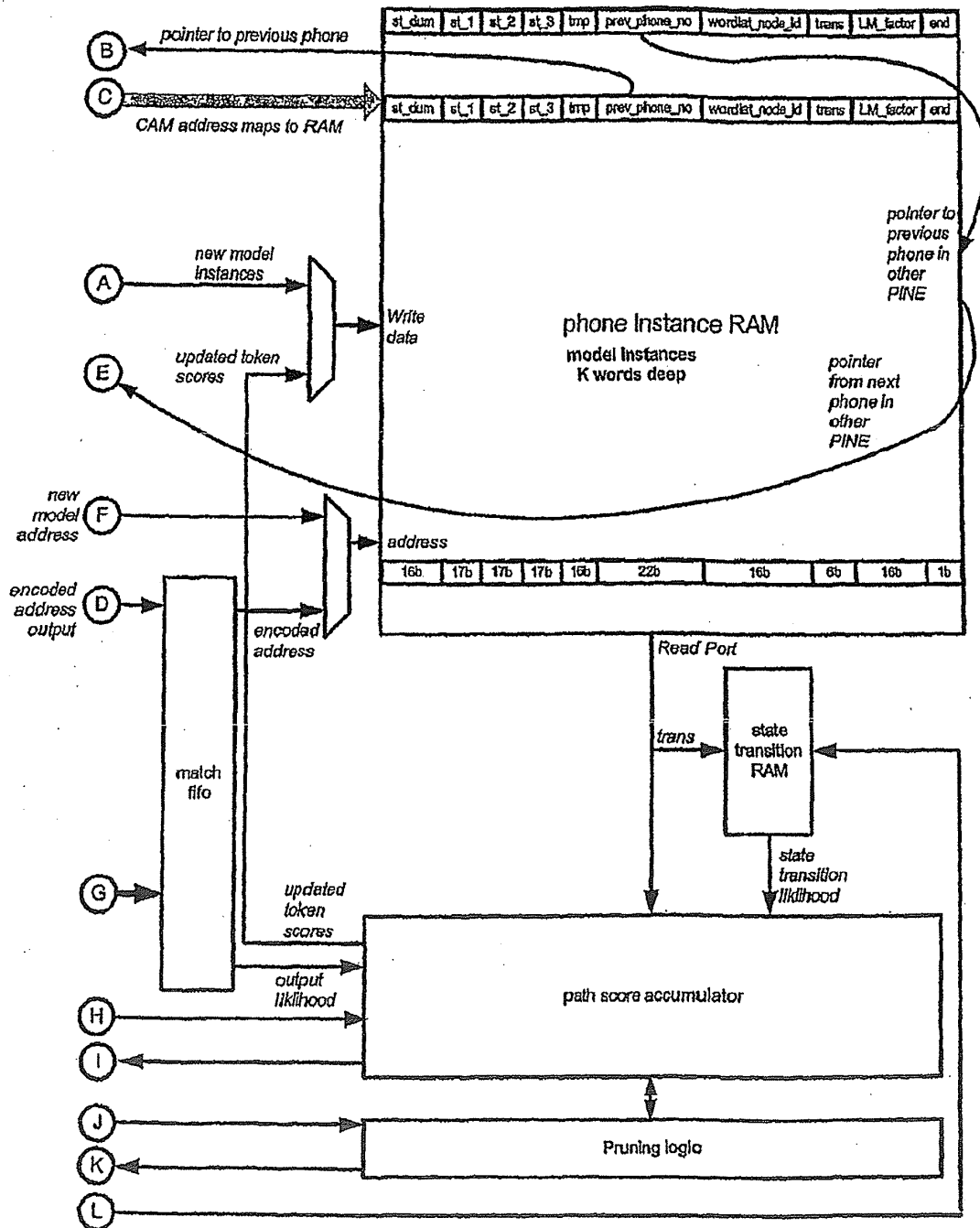


Figure 3B



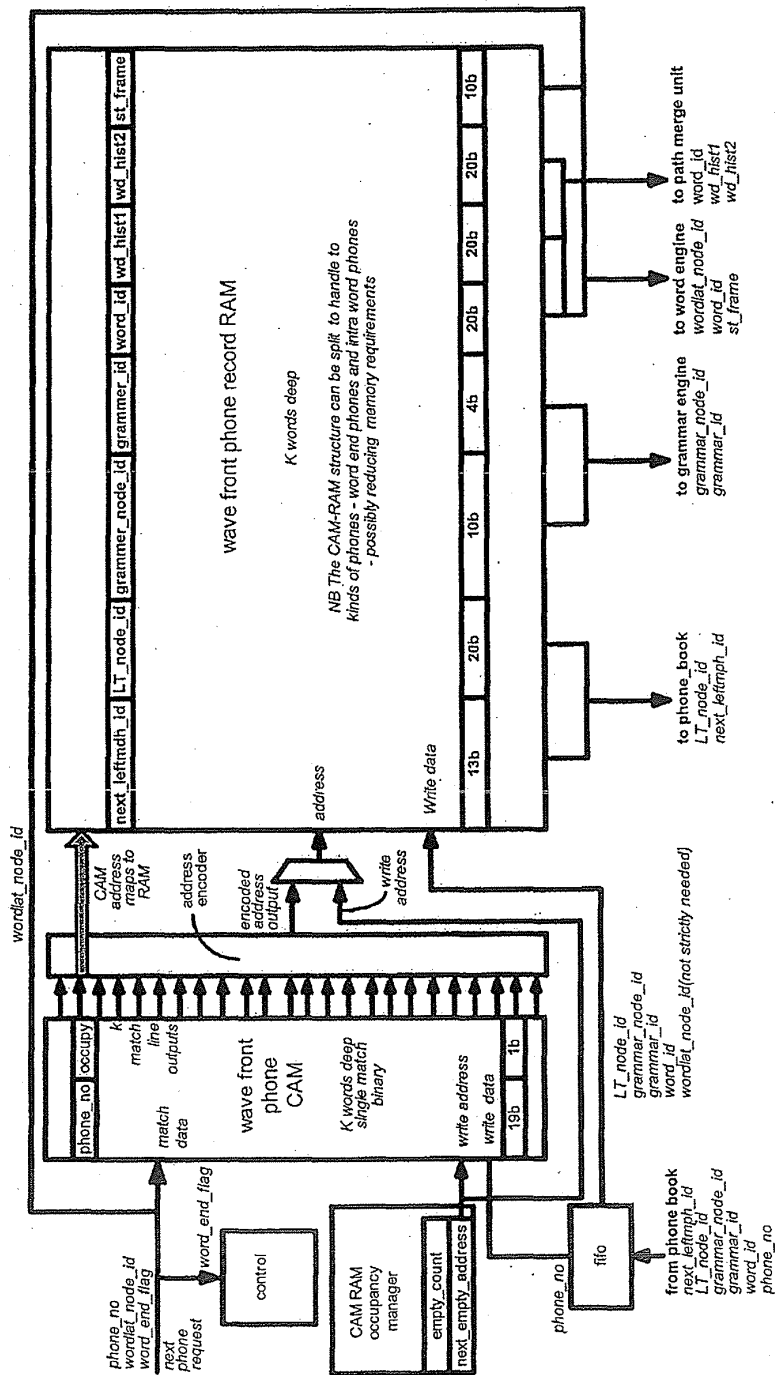


Figure 4

Figure 5

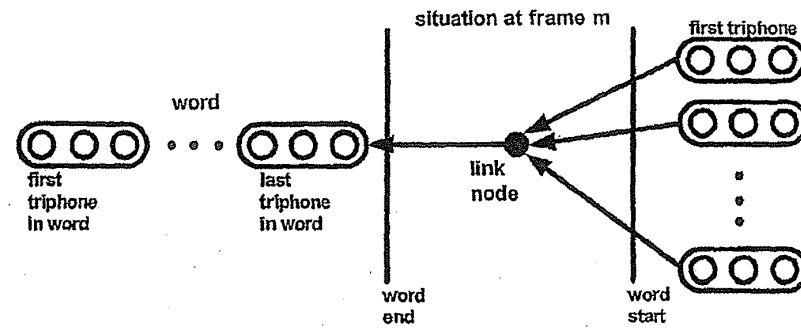
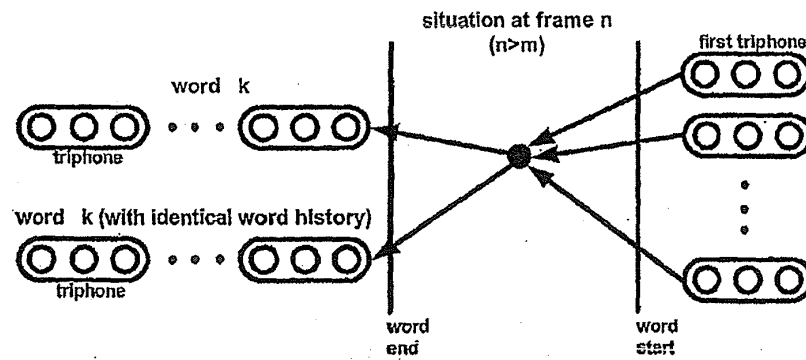
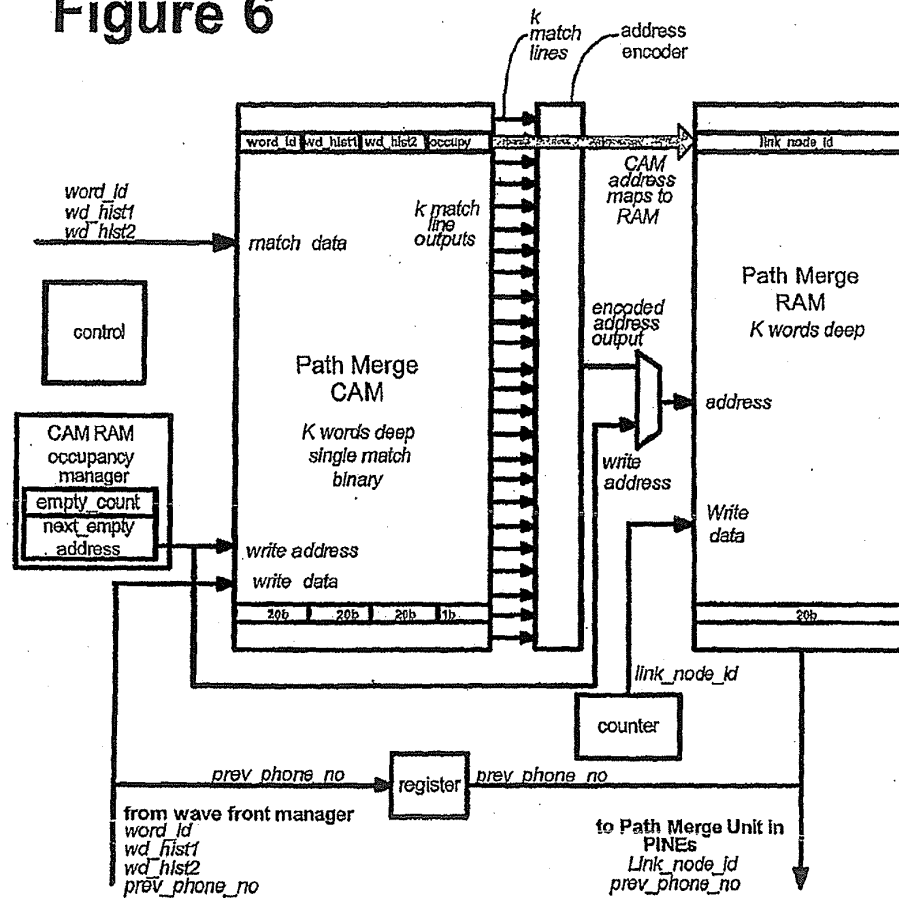


Figure 6



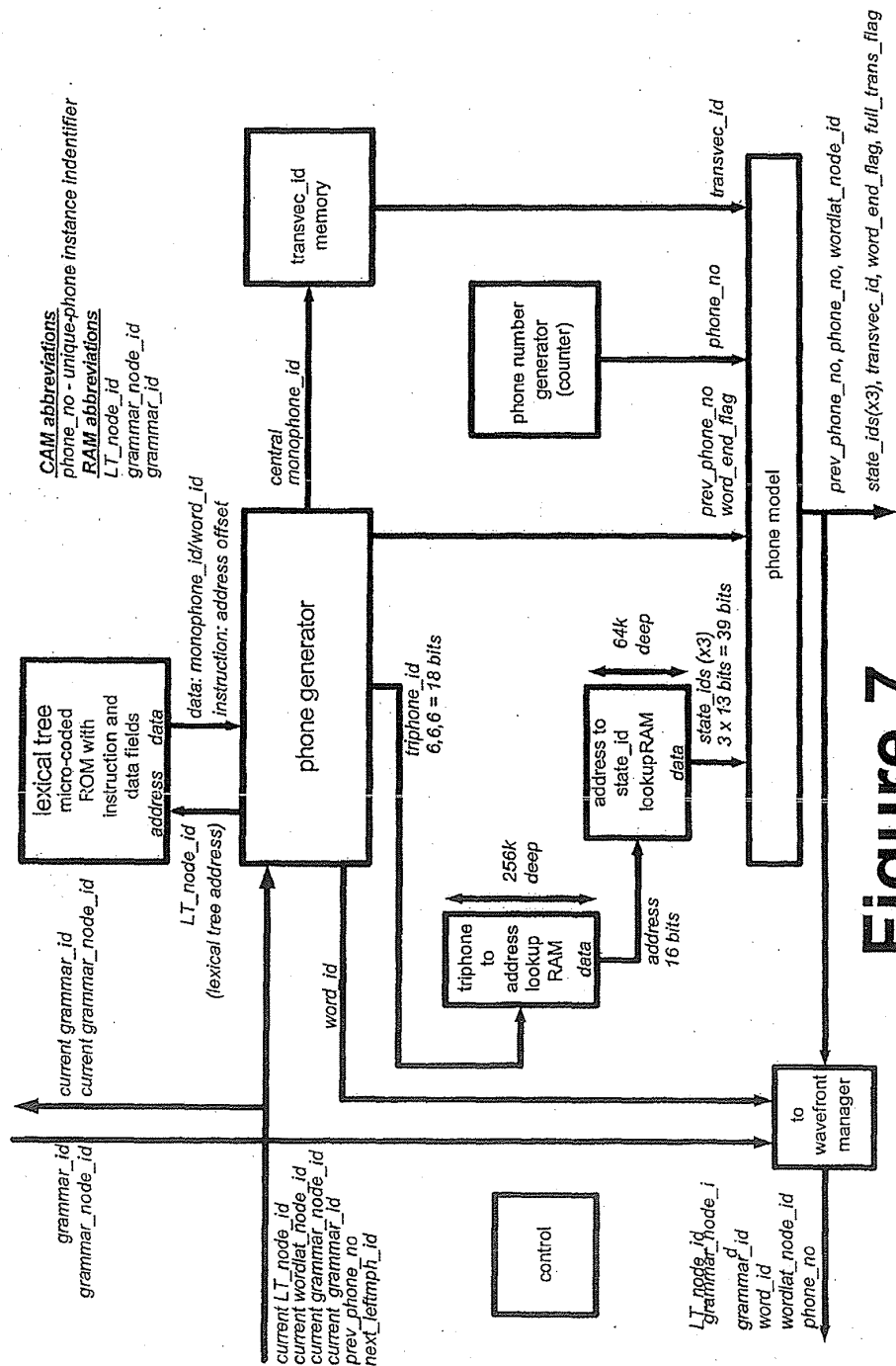
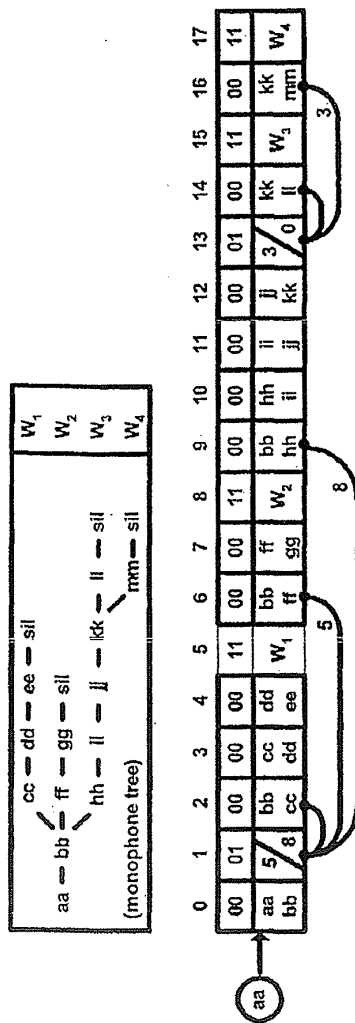
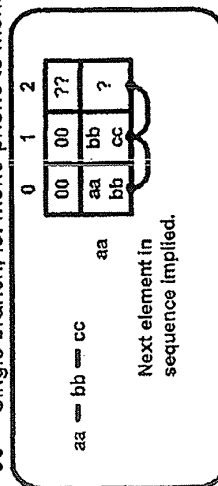


Figure 7

Figure 8A



"00" - Single branch, ie. mono-phone to mono-phone.



"01" - two branches.

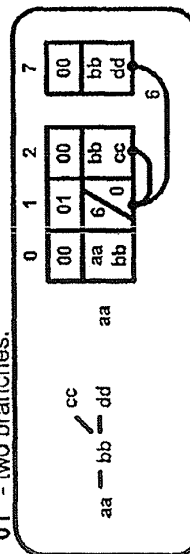


Figure 8B

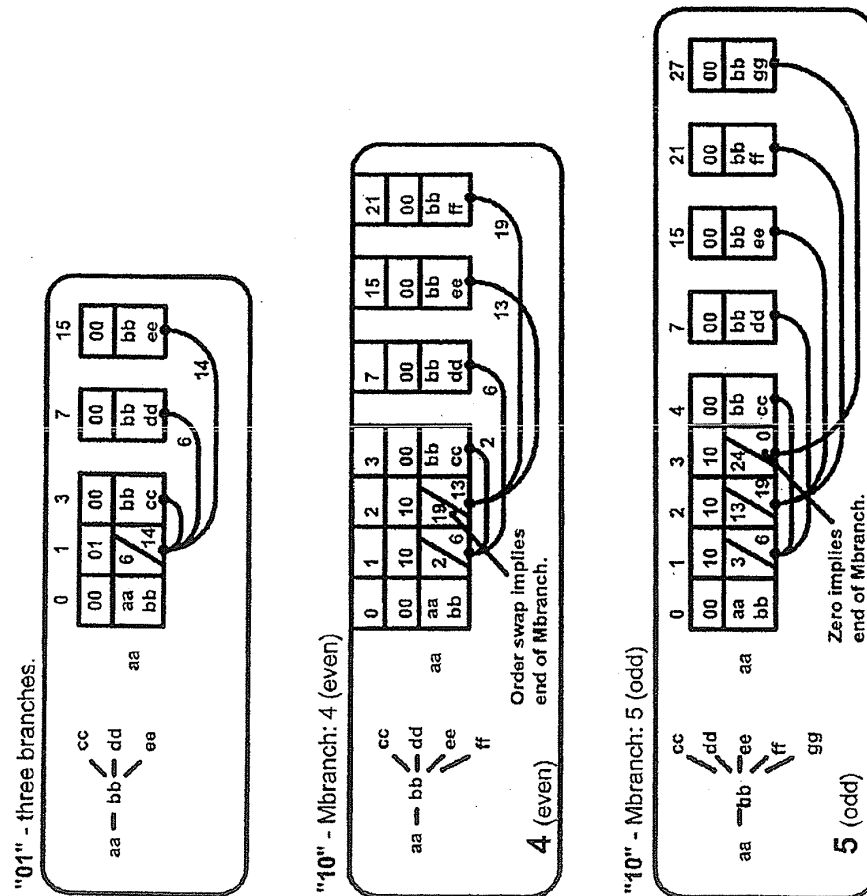


Figure 9

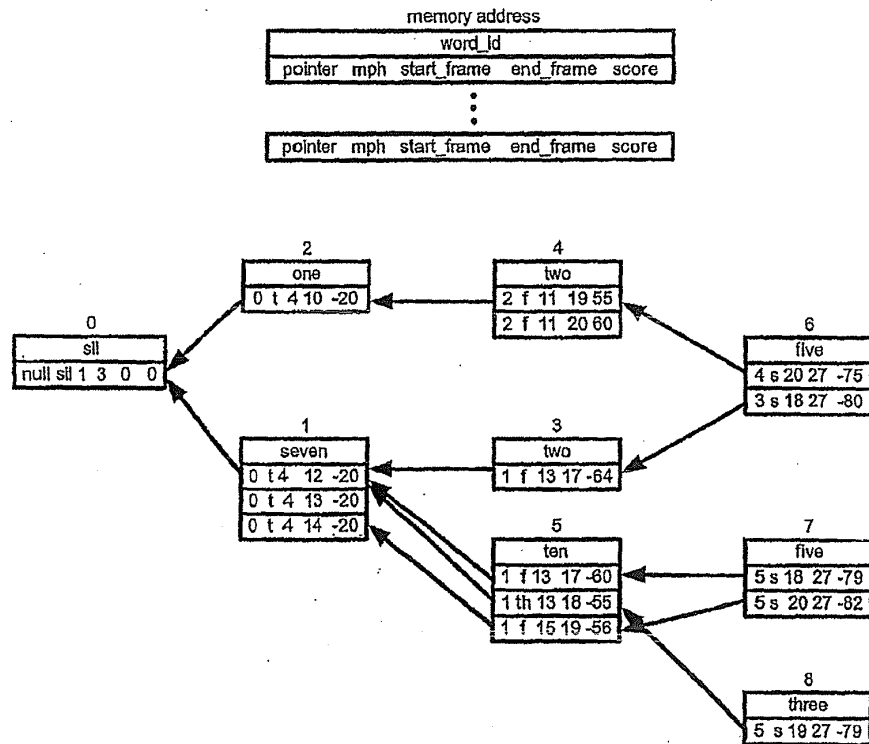
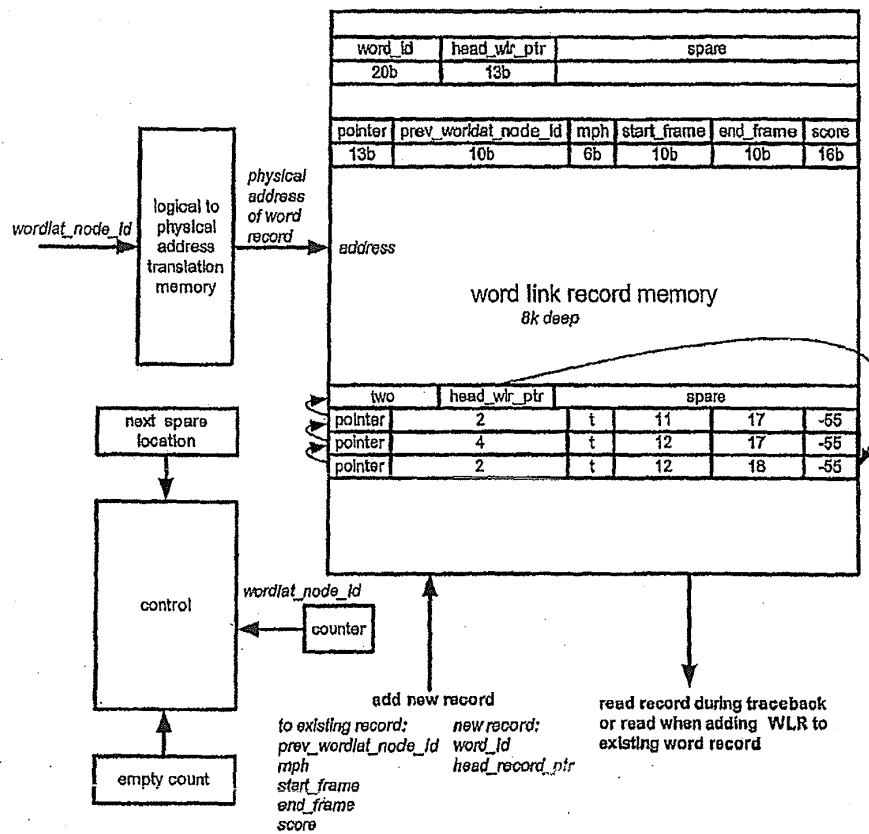


Figure 10



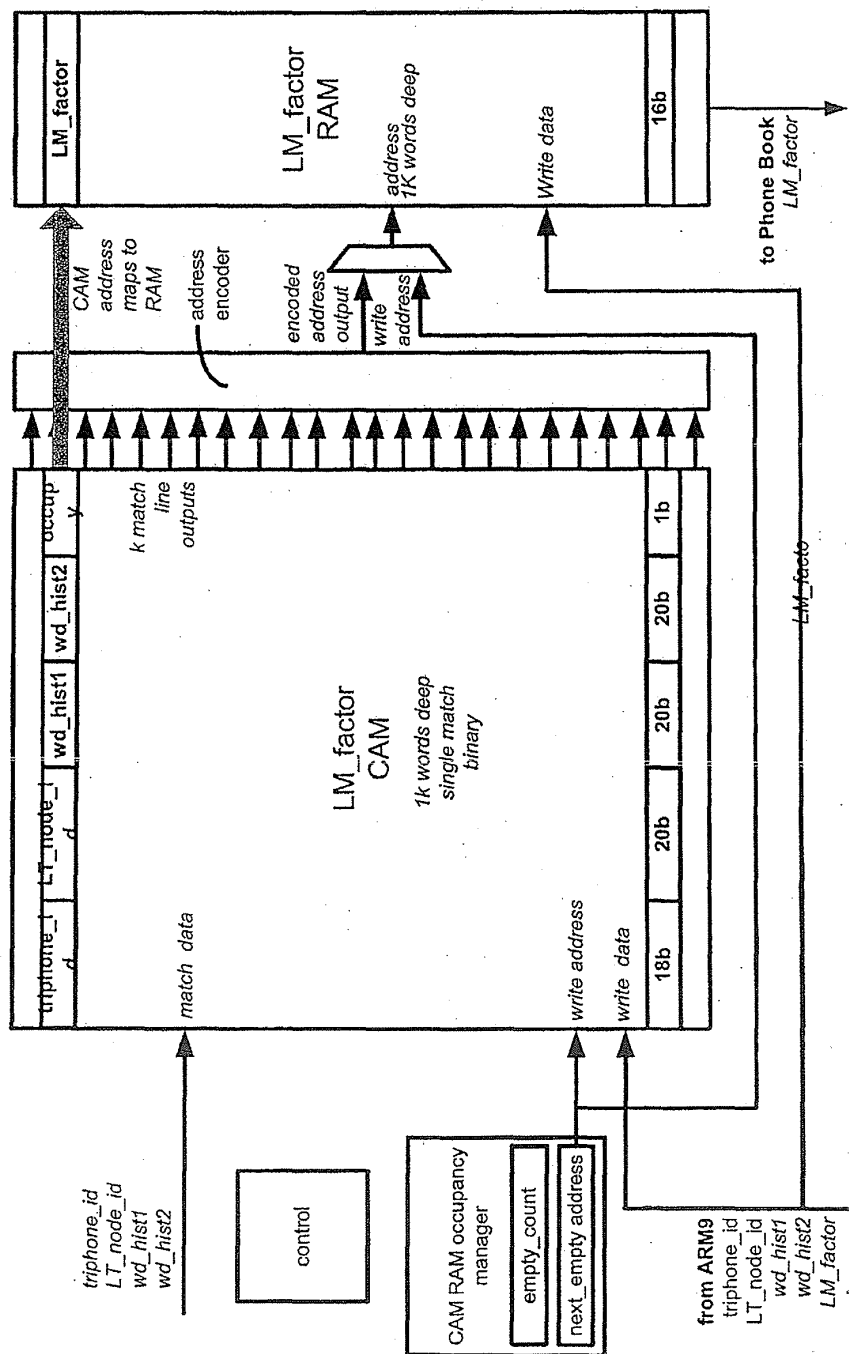


Figure 11

Figure 12

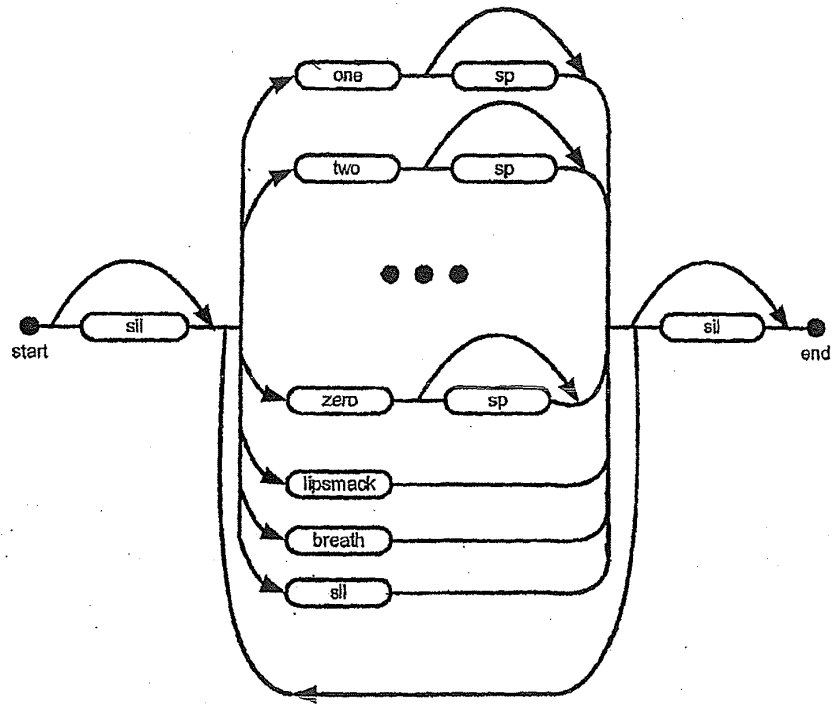
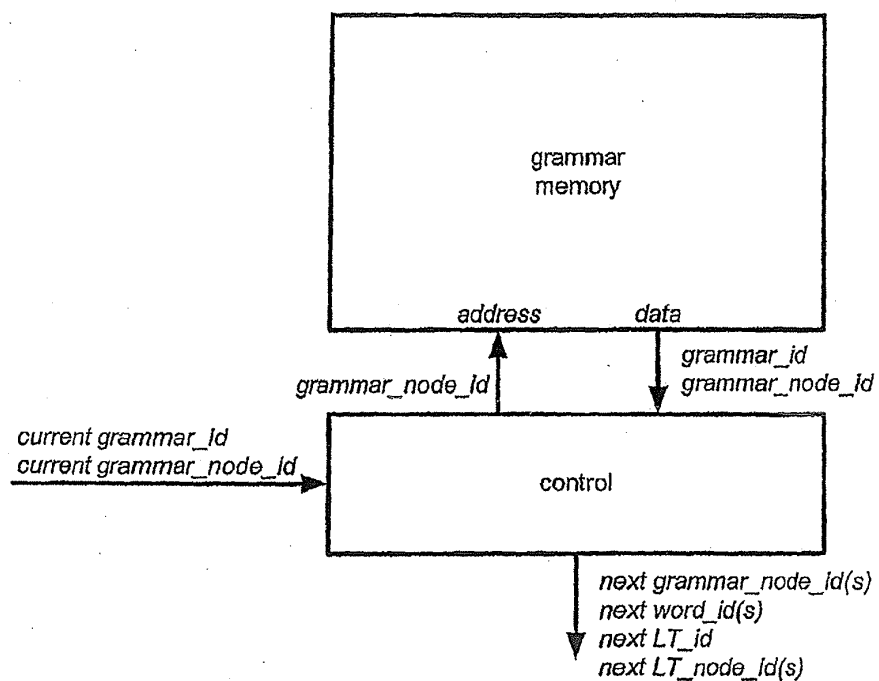


Figure 13



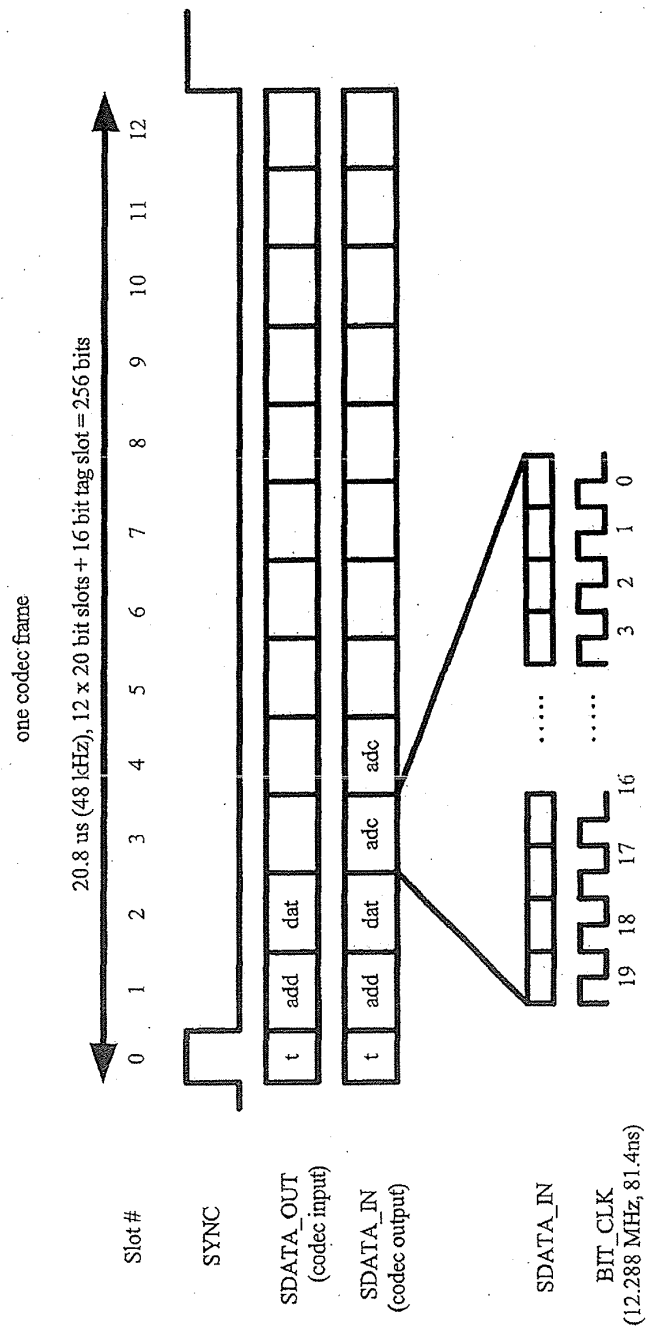
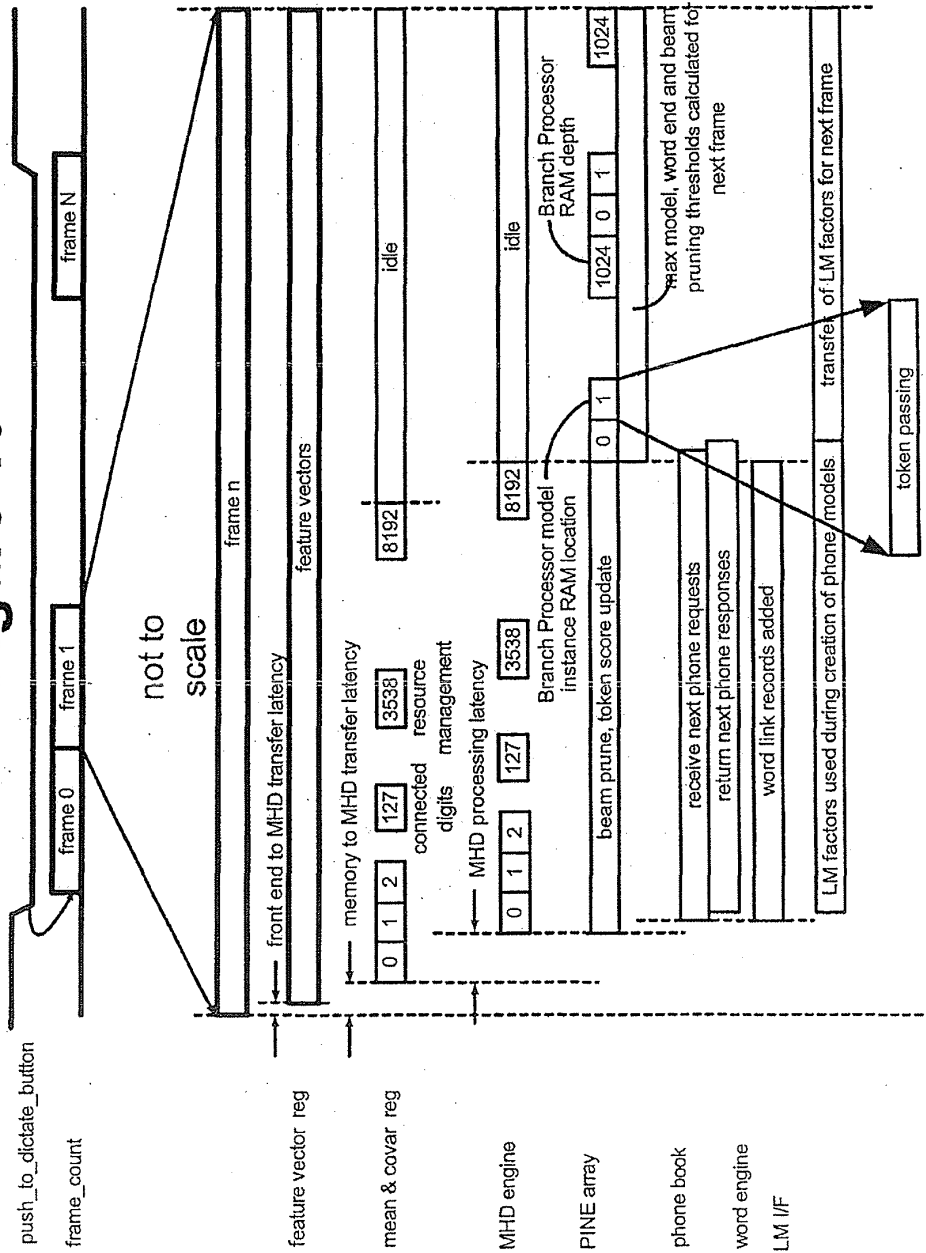


Figure 14

Figure 15



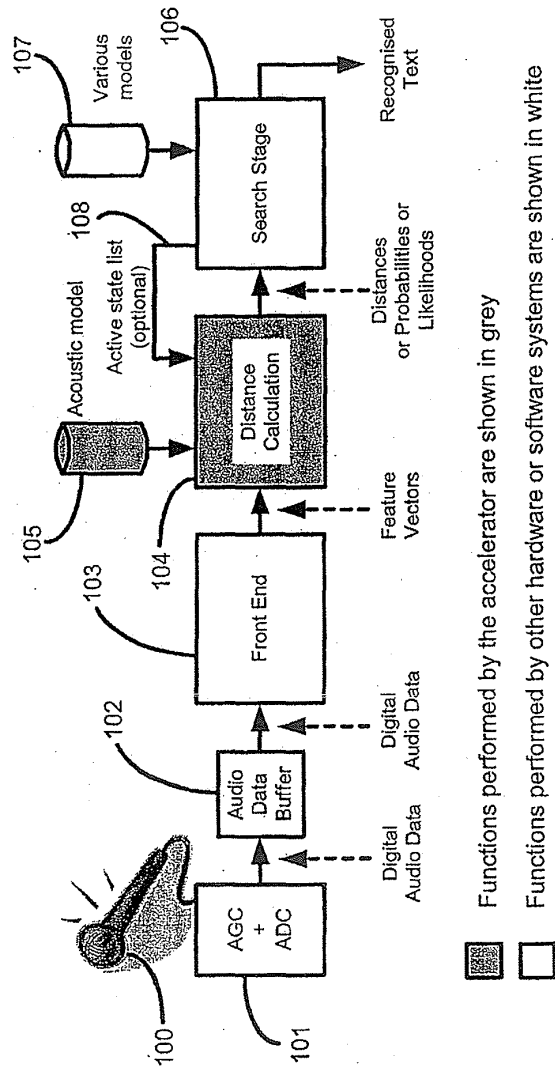


Figure 16

The diagram illustrates the internal architecture of the Distance Accelerator Subsystem (200). A dashed line (211) separates the subsystem from the external environment. An external interface (212) is shown at the top. Inside the subsystem, the following components are present:

- Registers (201)**: Connected to the external interface (212) and the internal bus (211).
- Result Memory A (202)**: Connected to the external interface (212) and the internal bus (211).
- Result Memory B (202)**: Connected to the internal bus (211).
- FV Memory A (208)**: Connected to the internal bus (211) and the FV Registers and optional FIFO (209).
- FV Registers and optional FIFO (209)**: Connected to FV Memory A (208) and the Distance Calculation Engine (204).
- Control, Sequencing and Address Generation (210)**: Connected to the internal bus (211) and the Distance Calculation Engine (204).
- Distance Calculation Engine (204)**: The central processing unit, connected to Result Memory A (202), Result Memory B (202), FV Registers (209), and the Decompressor (206).
- CRC Check (207)**: Connected to the Distance Calculation Engine (204).
- Decompressor (206)**: Connected to the Distance Calculation Engine (204) and the Acoustic Model Memory (205).
- Acoustic Model Memory (205)**: The base memory for the system, connected to the Decompressor (206).

The internal bus (211) facilitates data flow between the Registers, Result Memory A, Result Memory B, FV Memory A, and the Control, Sequencing and Address Generation block. The external interface (212) provides a connection to the system's input/output.

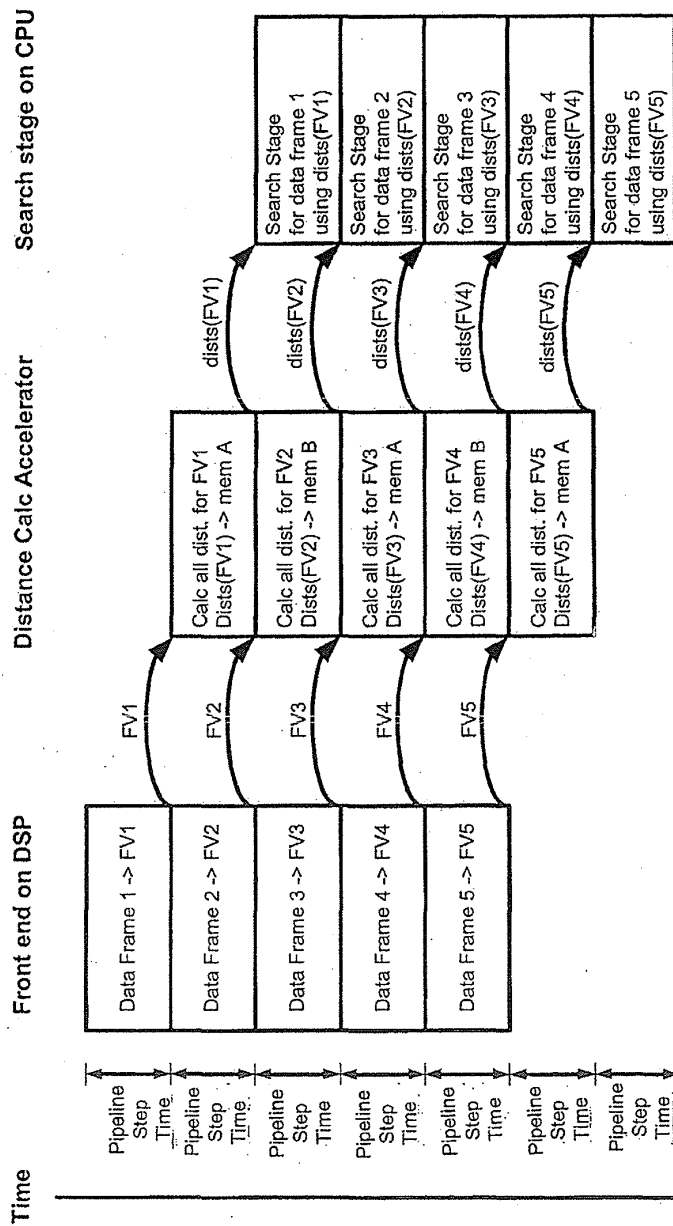


Figure 18

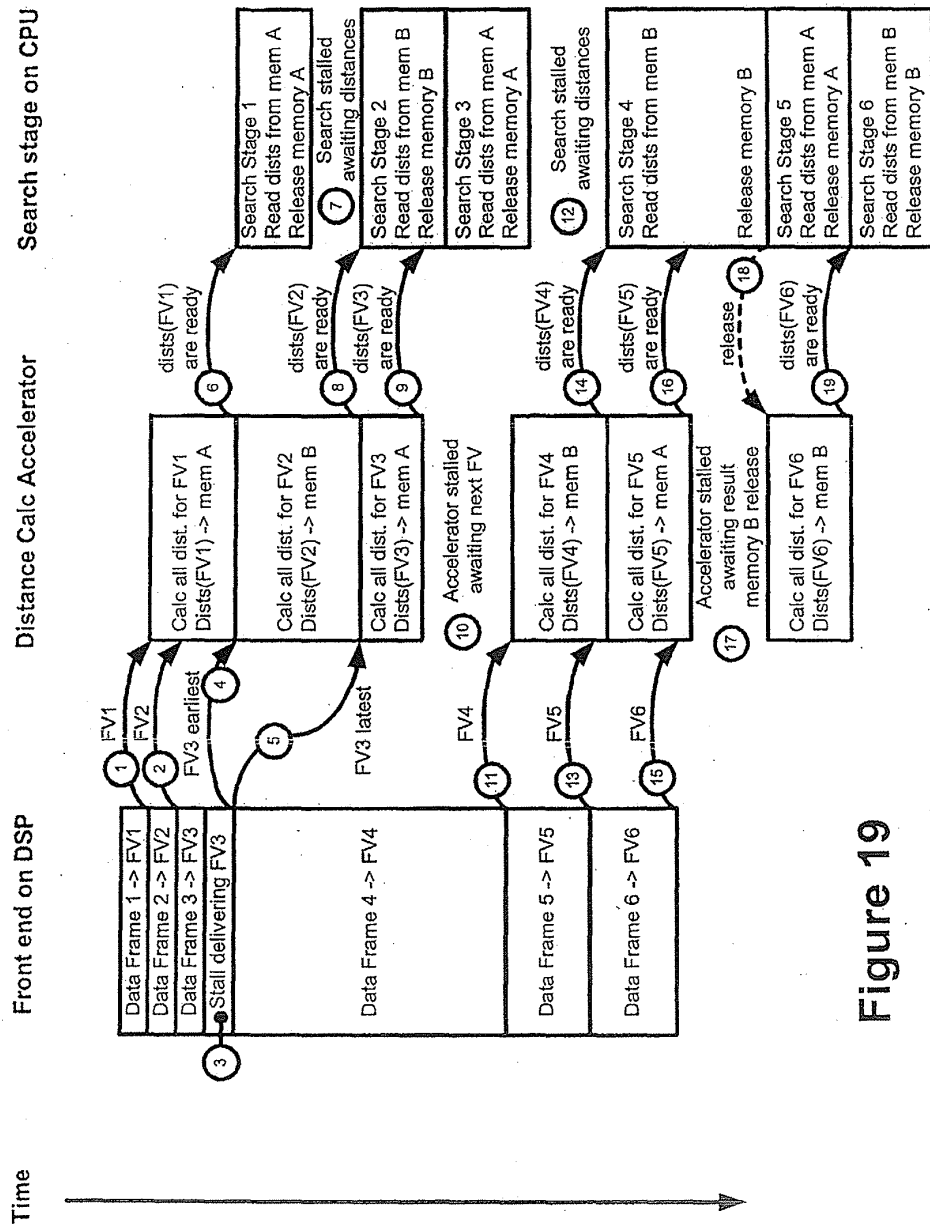


Figure 19

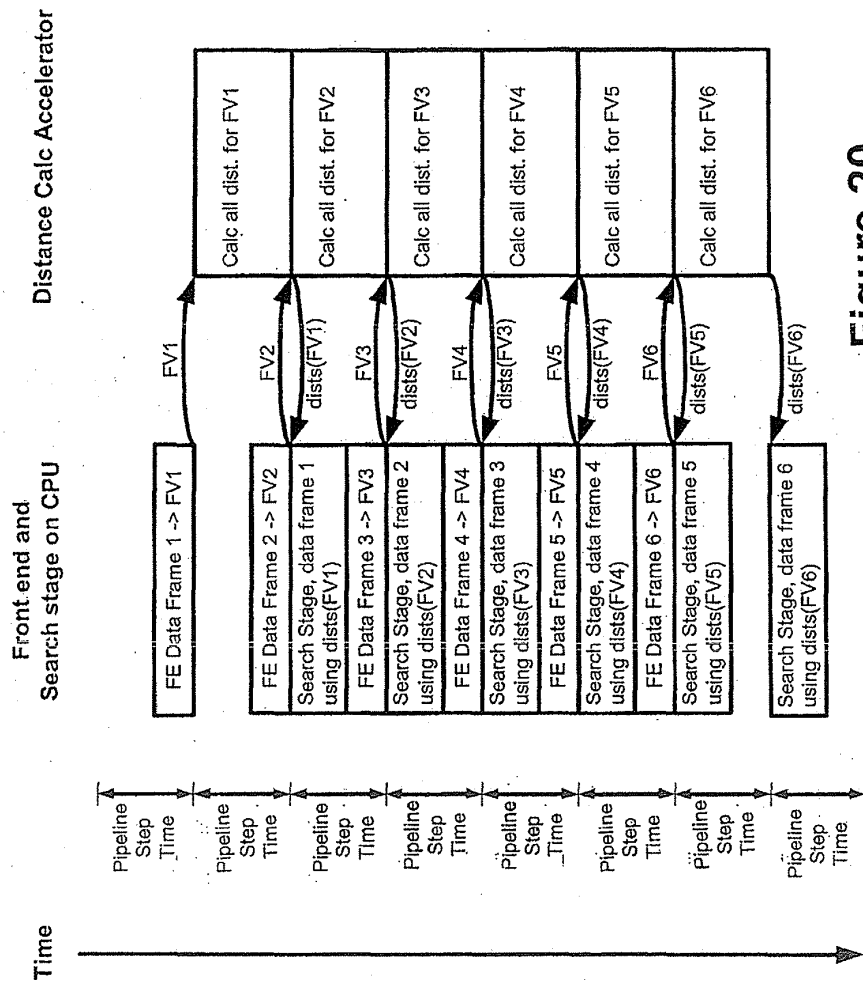
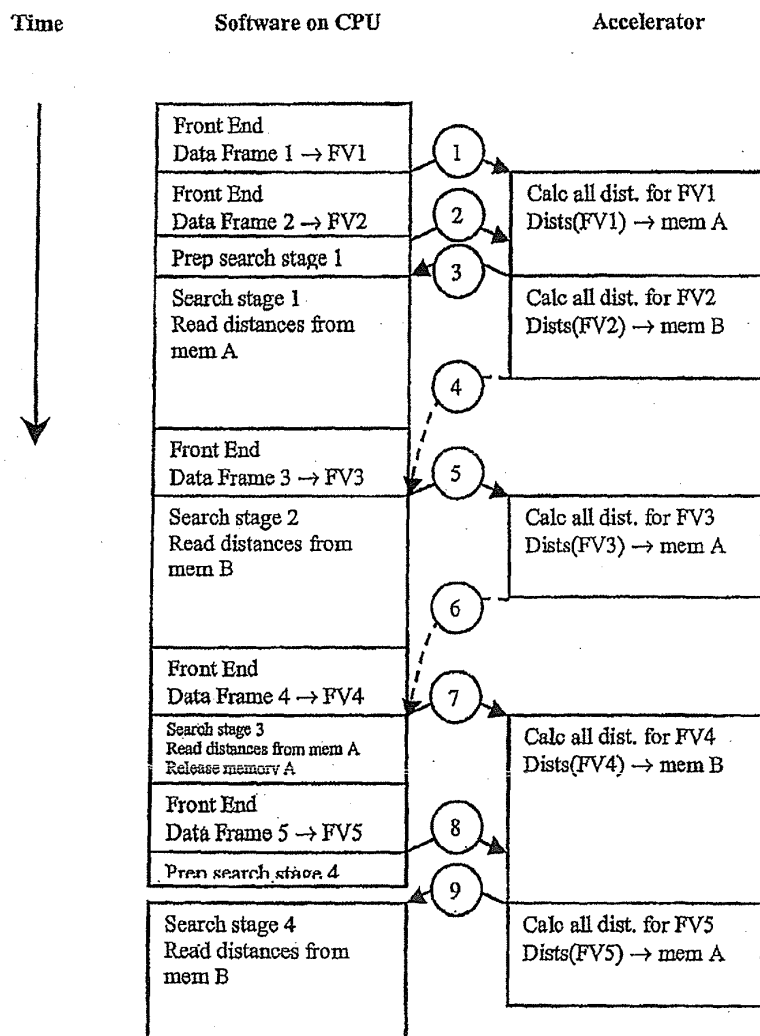


Figure 20

FIGURE 21



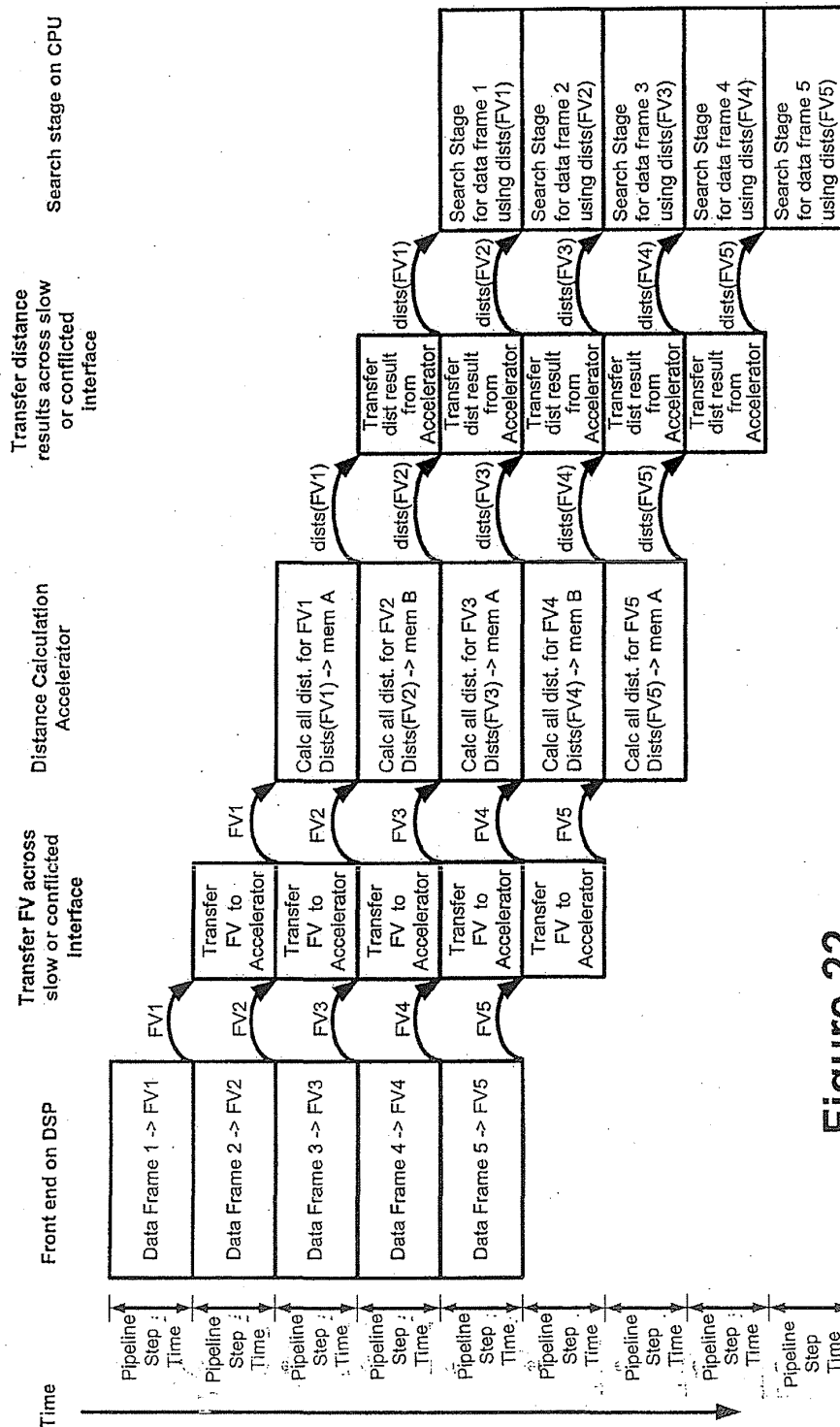


Figure 22

**DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN
APPLICATION DATA SHEET (37 CFR 1.76)**

**Title of
Invention**

A SPEECH RECOGNITION CIRCUIT AND METHOD

As the below named inventor, I declare that:

This declaration ☒ The attached application, or
is directed to:

☐ United States application or PCT international application number _____
filed on _____

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I have reviewed and understand the contents of the above-identified application, including the claims.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in 37 CFR 1.56, including for continuation-in-part applications, material information which became available between the filing date of the prior application and the national or PCT International filing date of the continuation-in-part application.

All statements made herein of my/our own knowledge are true, all statements made herein on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like are punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both, and may jeopardize the validity of the application or any patent issuing thereon.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: _____ Guy Larri _____ Date (Optional): _____

**Signature: _____

Guy Larri

** An electronic S-signature is permitted. The S-signature must consist only of letters, or Arabic numerals, or both, with appropriate spaces and commas, periods, apostrophes, or hyphens for punctuation, and the person signing the correspondence must insert his or her own S-signature with a first single forward slash mark before and a second single forward slash mark after the S-signature (e.g., /Dr. James T. Jones, Jr./).

DSMDB-3348559 v1

**DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN
APPLICATION DATA SHEET (37 CFR 1.76)**

**Title of
Invention**

A SPEECH RECOGNITION CIRCUIT AND METHOD

As the below named inventor, I declare that:

This declaration ☒ The attached application, or
is directed to:

☐ United States application or PCT international application number _____
filed on _____.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I have reviewed and understand the contents of the above-identified application, including the claims.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in 37 CFR 1.56, including for continuation-in-part applications, material information which became available between the filing date of the prior application and the national or PCT International filing date of the continuation-in-part application.

All statements made herein of my/our own knowledge are true, all statements made herein on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like are punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both, and may jeopardize the validity of the application or any patent issuing thereon.

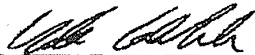
WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: _____ Mark Catchpole _____ Date (Optional): _____

**Signature: _____



** An electronic S-signature is permitted. The S-signature must consist only of letters, or Arabic numerals, or both, with appropriate spaces and commas, periods, apostrophes, or hyphens for punctuation, and the person signing the correspondence must insert his or her own S-signature with a first single forward slash mark before and a second single forward slash mark after the S-signature (e.g., /Dr. James T. Jones, Jr./).

**DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN
APPLICATION DATA SHEET (37 CFR 1.76)**

**Title of
Invention**

A SPEECH RECOGNITION CIRCUIT AND METHOD

As the below named inventor, I declare that:

This declaration ☒ The attached application, or
is directed to:

☐ United States application or PCT international application number _____
filed on _____.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I have reviewed and understand the contents of the above-identified application, including the claims.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in 37 CFR 1.56, including for continuation-in-part applications, material information which became available between the filing date of the prior application and the national or PCT International filing date of the continuation-in-part application.

All statements made herein of my/our own knowledge are true, all statements made herein on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like are punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both, and may jeopardize the validity of the application or any patent issuing thereon.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Damian Kelly Harris-Dowsett Date (Optional): _____

**Signature: 

** An electronic S-signature is permitted. The S-signature must consist only of letters, or Arabic numerals, or both, with appropriate spaces and commas, periods, apostrophes, or hyphens for punctuation, and the person signing the correspondence must insert his or her own S-signature with a first single forward slash mark before and a second single forward slash mark after the S-signature (e.g., /Dr. James T. Jones, Jr./).

**DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN
APPLICATION DATA SHEET (37 CFR 1.76)**

**Title of
Invention**

A SPEECH RECOGNITION CIRCUIT AND METHOD

As the below named inventor, I declare that:

This declaration ☒ The attached application, or
is directed to:

☐ United States application or PCT international application number _____
filed on _____

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I have reviewed and understand the contents of the above-identified application, including the claims.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in 37 CFR 1.56, including for continuation-in-part applications, material information which became available between the filing date of the prior application and the national or PCT International filing date of the continuation-in-part application.

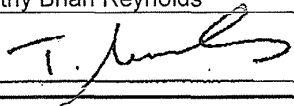
All statements made herein of my/our own knowledge are true, all statements made herein on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like are punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both, and may jeopardize the validity of the application or any patent issuing thereon.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: _____ Timothy Brian Reynolds _____ Date (Optional): _____

**Signature: _____  _____

** An electronic S-signature is permitted. The S-signature must consist only of letters, or Arabic numerals, or both, with appropriate spaces and commas, periods, apostrophes, or hyphens for punctuation, and the person signing the correspondence must insert his or her own S-signature with a first single forward slash mark before and a second single forward slash mark after the S-signature (e.g., /Dr. James T. Jones, Jr./).



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
14/788,164	06/30/2015		730	M0025.0369-C	1	1

CONFIRMATION NO. 1250

FILING RECEIPT



CC000000078271103

24998
DICKSTEIN SHAPIRO LLP
1825 EYE STREET NW
Washington, DC 20006-5403

Date Mailed: 07/16/2015

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Guy Larri, Cambridge, UNITED KINGDOM;
Mark Catchpole, Prickwillow, UNITED KINGDOM;
Damian Kelly Harris-Dowsett, Coventry, UNITED KINGDOM;
Timothy Brian Reynolds, Cambridge, UNITED KINGDOM;

Applicant(s)

Zentian Limited, Cambridge, UNITED KINGDOM;

Power of Attorney: The patent practitioners associated with Customer Number 24998

Domestic Priority data as claimed by applicant

This application is a CON of 13/735,091 01/07/2013 PAT 9076441
which is a CON of 13/162,128 06/16/2011 PAT 8352262
which is a DIV of 11/662,704 03/14/2007 PAT 7979277
which is a 371 of PCT/GB2005/003554 09/14/2005

Foreign Applications (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.)
UNITED KINGDOM 0420464.0 09/14/2004 No Access Code Provided

If Required, Foreign Filing License Granted: 07/14/2015

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 14/788,164**

Projected Publication Date: To Be Determined - pending completion of Corrected Papers

Non-Publication Request: No

Early Publication Request: No

**** SMALL ENTITY ****

Title

SPEECH RECOGNITION CIRCUIT AND METHOD

Preliminary Class

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER
Title 35, United States Code, Section 184
Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit <http://www.SelectUSA.gov> or call +1-202-482-6800.

PATENT APPLICATION FEE DETERMINATION RECORD						Application or Docket Number 14/788,164			
Substitute for Form PTO-875									
APPLICATION AS FILED - PART I									
(Column 1)		(Column 2)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY			
FOR	NUMBER FILED	NUMBER EXTRA		RATE(\$)	FEE(\$)	RATE(\$)	FEE(\$)		
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A		N/A	70	N/A			
SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A		N/A	300	N/A			
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A		N/A	360	N/A			
TOTAL CLAIMS (37 CFR 1.16(i))	1	minus 20 = *		x 40 =	0.00	OR			
INDEPENDENT CLAIMS (37 CFR 1.16(h))	1	minus 3 = *		x 210 =	0.00				
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).				0.00				
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))					0.00				
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL	730	TOTAL			
APPLICATION AS AMENDED - PART II									
AMENDMENT A	(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY	OR OTHER THAN SMALL ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT	MINUS	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	x	=	OR	x	=
	Independent (37 CFR 1.16(h))	*	Minus	***	x	=	OR	x	=
	Application Size Fee (37 CFR 1.16(s))						OR		
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						OR		
				TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE		
AMENDMENT B	(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY	OR OTHER THAN SMALL ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT	MINUS	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	x	=	OR	x	=
	Independent (37 CFR 1.16(h))	*	Minus	***	x	=	OR	x	=
	Application Size Fee (37 CFR 1.16(s))						OR		
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						OR		
				TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE		
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3. ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20". *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3". The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.									



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
14/788,164	06/30/2015	Guy Larri	M0025.0369-C

CONFIRMATION NO. 1250

FORMALITIES LETTER



Date Mailed: 07/16/2015

24998
DICKSTEIN SHAPIRO LLP
1825 EYE STREET NW
Washington, DC 20006-5403

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Filing Date Granted

An application number and filing date have been accorded to this application. The application is informal since it does not comply with the regulations for the reason(s) indicated below. Applicant is given TWO MONTHS from the date of this Notice within which to correct the informalities indicated below. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

The required item(s) identified below must be timely submitted to avoid abandonment:

- A substitute specification in compliance with 37 CFR 1.52, 1.121(b)(3), and 1.125, is required. The substitute specification must be submitted with markings and be accompanied by a clean version (without markings) as set forth in 37 CFR 1.125(c) and a statement that the substitute specification contains no new matter (see 37 CFR 1.125(b)). The specification, claims, and/or abstract page(s) submitted is not acceptable and cannot be scanned or properly stored because:
 - The line spacing on the specification, claims, and/or abstract is not 1½ or double spaced (see 37 CFR 1.52(b)).
- Replacement drawings in compliance with 37 CFR 1.84 and 37 CFR 1.121(d) are required. The drawings submitted are not acceptable because:
 - Numbers, letters, and reference characters on the drawings must measure at least 0.32 cm (1/8 inch) in height. See Figure(s) 1B, 2B, 4, and 6.
 - The drawings submitted to the Office are not electronically reproducible because portions of figures 3A and 3B are missing and/or blurry.

Applicant is cautioned that correction of the above items may cause the specification and drawings page count to exceed 100 pages. If the specification and drawings exceed 100 pages, applicant will need to submit the required application size fee.

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web, including a copy of this Notice and selecting the document description "Applicant response to Pre-Exam Formalities Notice".
<https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html>

For more information about EFS-Web please call the USPTO Electronic Business Center at 1-866-217-9197 or visit our website at <http://www.uspto.gov/ebc>.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/mbelay/

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Guy Larri et al.

Application No.: 14/788,164

Confirmation No.: 1250

Filed: June 30, 2015

Art Unit: Not Yet Assigned

For: A SPEECH RECOGNITION CIRCUIT AND
METHOD

Examiner: Not Yet Assigned

RESPONSE TO NOTICE TO FILE CORRECTED APPLICATION PAPERS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

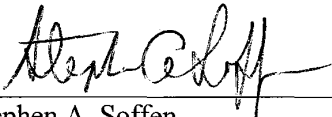
Dear Madam:

In response to the Notice to File Corrected Application Papers dated July 16, 2015, applicants submit six (6) sheets of replacement drawings (Figs. 1B, 2B, 3A, 3B, 4, and 6).

The Notice to File Corrected Application Papers calls for a substitute specification with correct line spacing, but does not identify any pages of the specification that contain incorrect line spacing. Applicant submits that the specification as originally filed has the proper 1.5 line spacing and is in compliance with the rules.

Dated: September 16, 2015

Respectfully submitted,

By 

Stephen A. Soffen

Registration No.: 31,063
DICKSTEIN SHAPIRO LLP
1825 Eye Street, NW
Washington, DC 20006-5403
(202) 420-2200
Attorney for Applicant

DSMDB-3365302 v1

Figure 1B

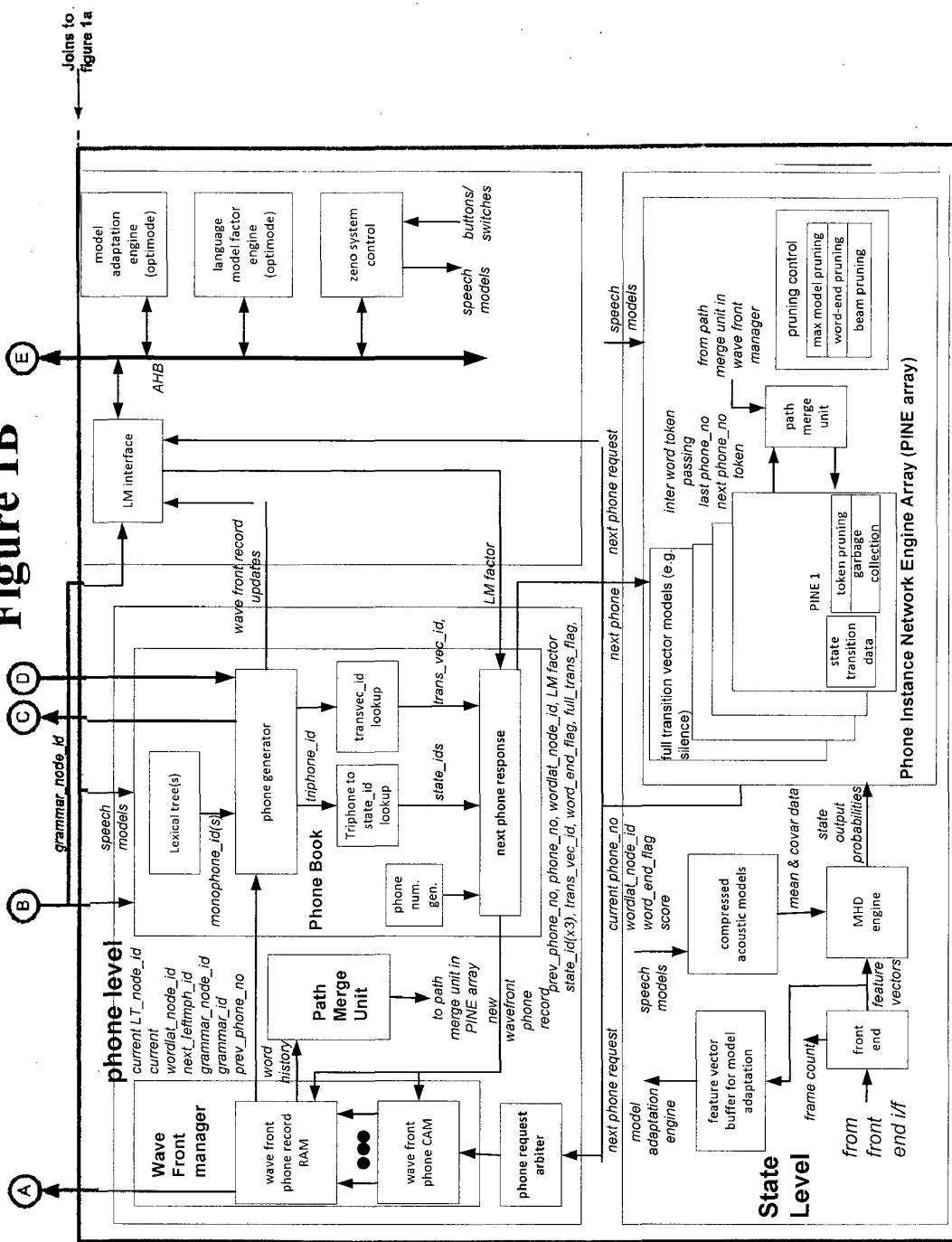


Figure 2B

Joins to Figure 2a

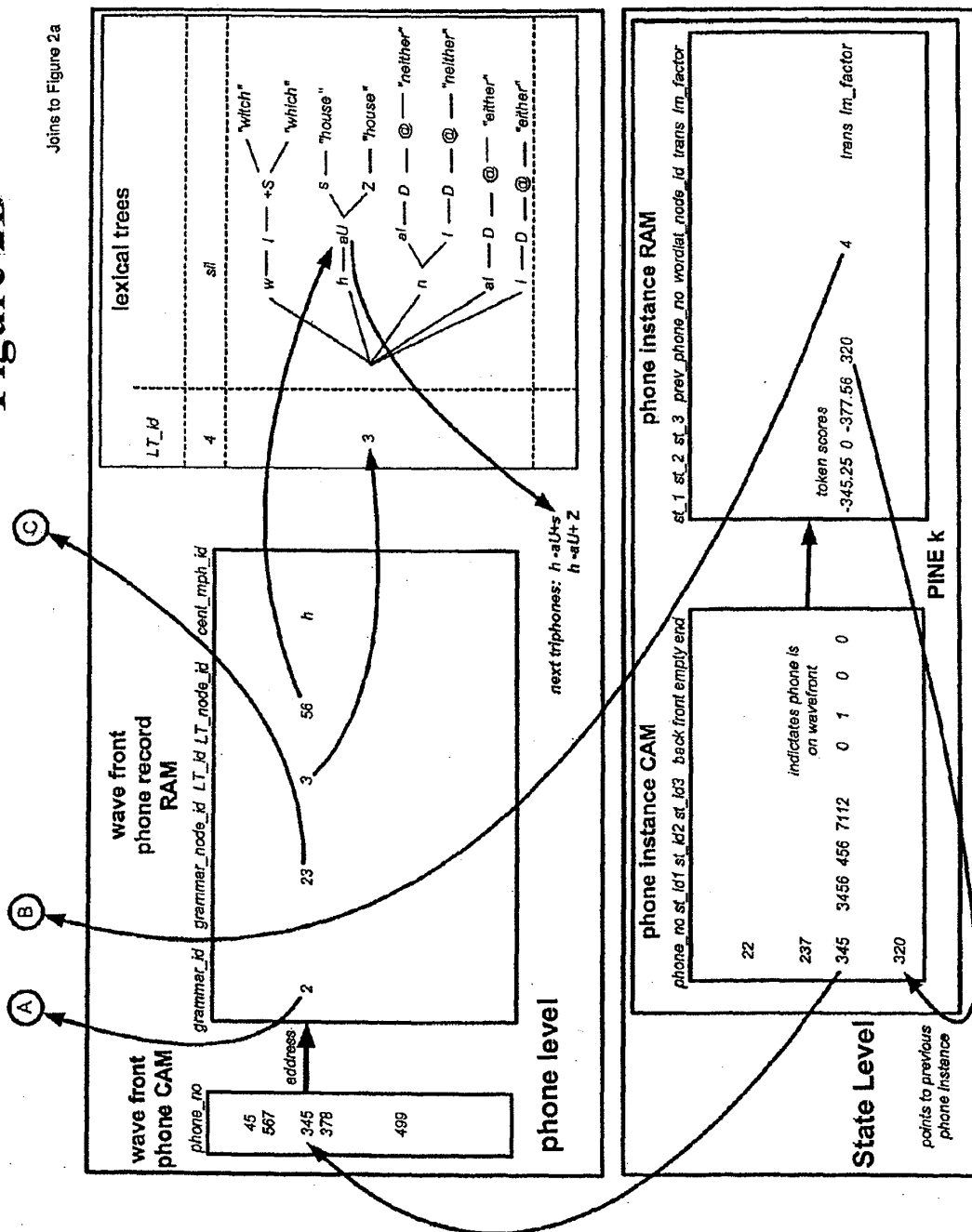


Figure 3A

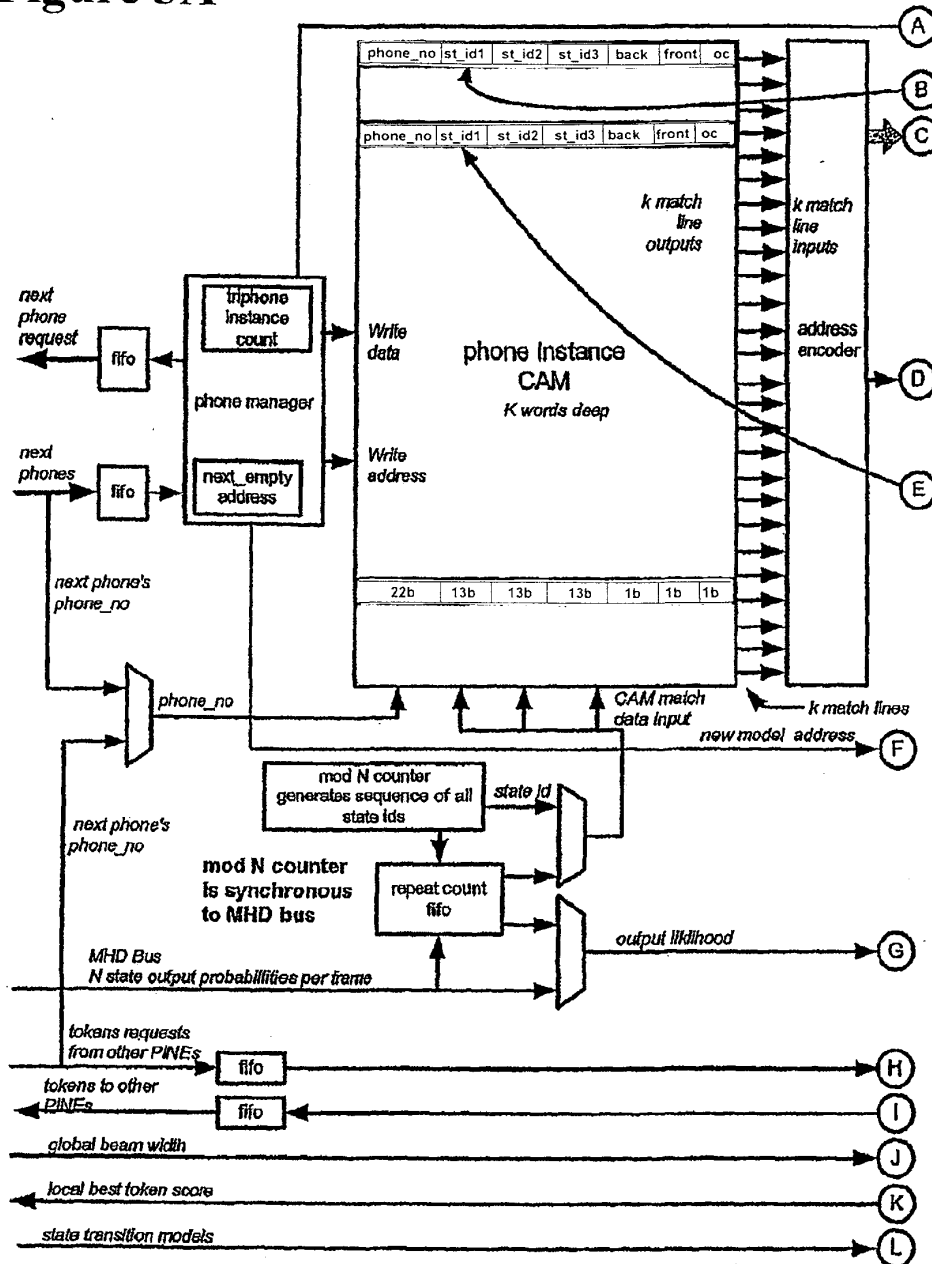
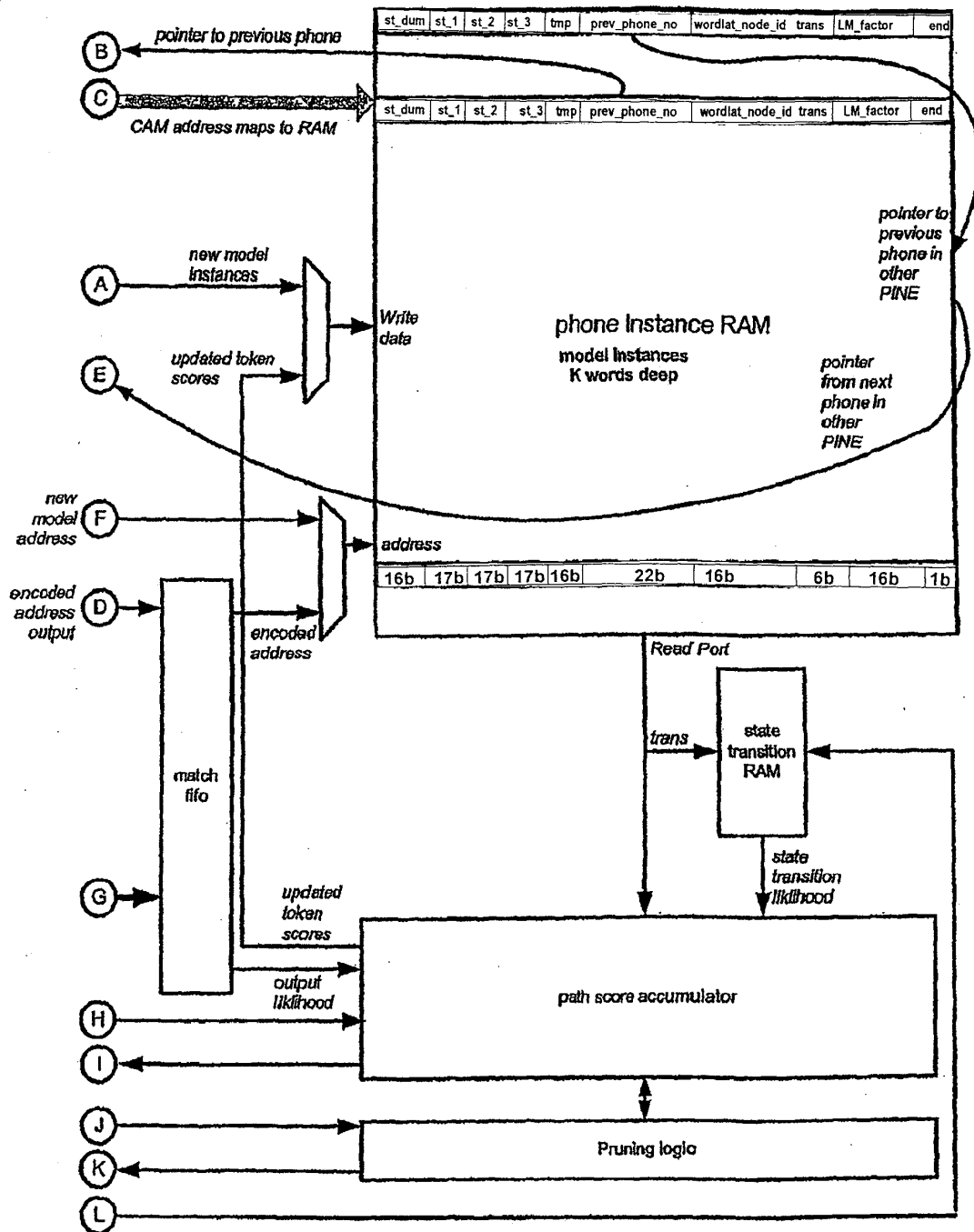


Figure 3B



REPLACEMENT SHEET

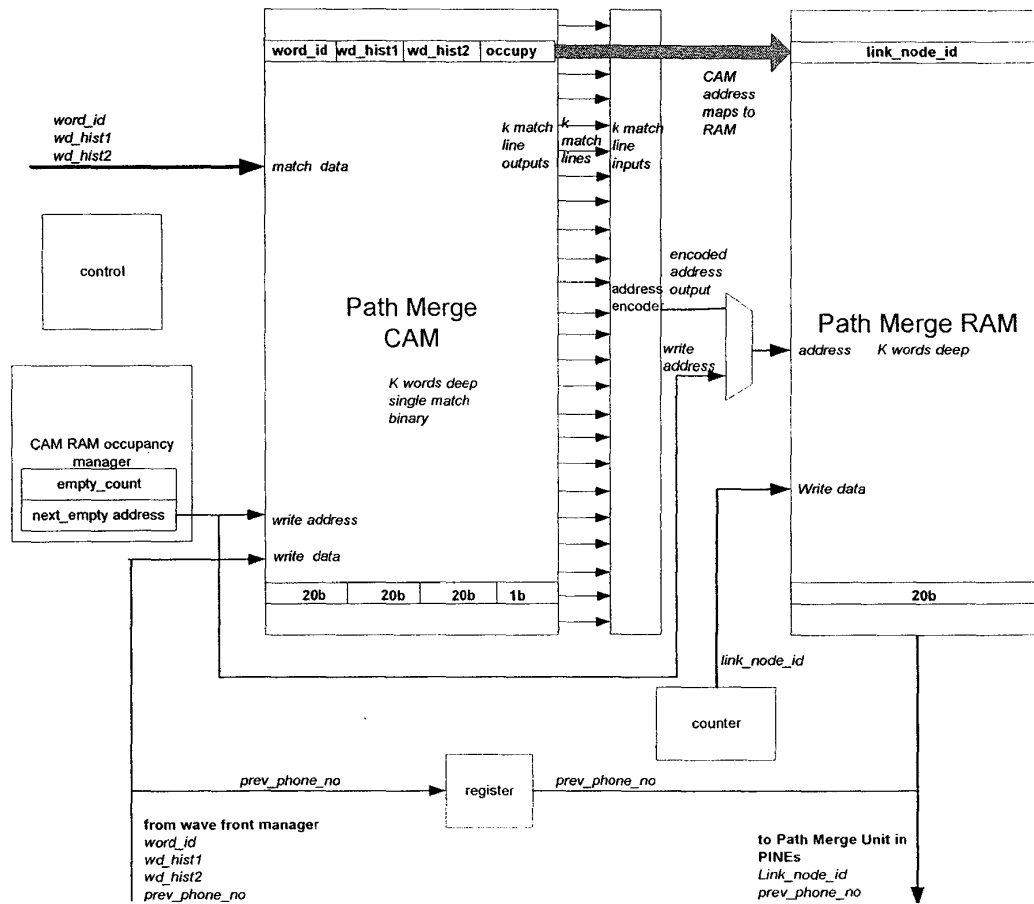


Figure 6

Electronic Acknowledgement Receipt	
EFS ID:	23514660
Application Number:	14788164
International Application Number:	
Confirmation Number:	1250
Title of Invention:	SPEECH RECOGNITION CIRCUIT AND METHOD
First Named Inventor/Applicant Name:	Guy Larri
Customer Number:	24998
Filer:	Stephen A. Soffen/Kanessa Howard
Filer Authorized By:	Stephen A. Soffen
Attorney Docket Number:	M0025.0369-C
Receipt Date:	16-SEP-2015
Filing Date:	30-JUN-2015
Time Stamp:	17:35:25
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment		no			
File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		M0025_0369_C_Resp_NTFCAP.pdf	216281 57c8774b404040f73db1d3d33e5623932d2204aa	yes	7

	Multipart Description/PDF files in .zip description		
	Document Description	Start	End
	Applicant Response to Pre-Exam Formalities Notice	1	1
	Drawings-only black and white line drawings	2	7
Warnings:			
Information:			
Total Files Size (in bytes):		216281	
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>			



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
14/788,164	06/30/2015	Guy Larri	M0025.0369/P369-C

CONFIRMATION NO. 1250

FORMALITIES LETTER



OC000000077625149*

24998
DICKSTEIN SHAPIRO LLP
1825 EYE STREET NW
Washington, DC 20006-5403

Date Mailed: 09/24/2015

NOTICE OF INCOMPLETE REPLY (NONPROVISIONAL)

Filing Date Granted

The U.S. Patent and Trademark Office has received your reply on 09/16/2015 to the Notice to File Missing Parts (Notice) mailed 07/16/2015 and it has been entered into the nonprovisional application. The reply, however, does not include the following items required in the Notice. A complete reply must be timely filed to prevent ABANDONMENT of the above-identified application. Replies should be mailed to: Mail Stop Missing Parts, Commissioner for Patents, P.O. Box 1450, Alexandria VA 22313-1450.

Applicant is given **TWO MONTHS** from the date of the Notice to File Missing Parts (Notice) mailed 07/16/2015 within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

Items Required to Avoid Abandonment:

The required items noted below SHOULD be filed along with any items required above. The filing date of this nonprovisional application will be the date of receipt of the items required above.

The application is informal since it does not comply with the regulations for the reason(s) indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- A substitute specification in compliance with 37 CFR 1.52, 1.121(b)(3), and 1.125, is required. The substitute specification must be submitted with markings and be accompanied by a clean version (without markings) as set forth in 37 CFR 1.125(c) and a statement that the substitute specification contains no new matter (see 37 CFR 1.125(b)). The specification, claims, and/or abstract page(s) submitted is not acceptable and cannot be scanned or properly stored because:
 - The line spacing on the specification, claims, and/or abstract is not 1½ or double spaced (see 37 CFR 1.52(b)).

Applicant is cautioned that correction of the above items may cause the specification and drawings page count to exceed 100 pages. If the specification and drawings exceed 100 pages, applicant will need to submit the required application size fee.

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web, including a copy of this Notice and selecting the document description "Applicant response to Pre-Exam Formalities Notice".
<https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html>

For more information about EFS-Web please call the USPTO Electronic Business Center at 1-866-217-9197 or visit our website at <http://www.uspto.gov/ebc>.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/tlulu/

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875						Application or Docket Number 14/788,164		
APPLICATION AS FILED - PART I								
(Column 1)		(Column 2)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY		
FOR	NUMBER FILED	NUMBER EXTRA	RATE(\$)	FEE(\$)		RATE(\$)	FEE(\$)	
BASIC FEE <small>(37 CFR 1.16(a), (b), or (c))</small>	N/A	N/A	N/A	70		N/A		
SEARCH FEE <small>(37 CFR 1.16(k), (l), or (m))</small>	N/A	N/A	N/A	300		N/A		
EXAMINATION FEE <small>(37 CFR 1.16(o), (p), or (q))</small>	N/A	N/A	N/A	360		N/A		
TOTAL CLAIMS <small>(37 CFR 1.16(j))</small>	1	minus 20 = *	x 40 =	0.00	OR			
INDEPENDENT CLAIMS <small>(37 CFR 1.16(h))</small>	1	minus 3 = *	x 210 =	0.00				
APPLICATION SIZE FEE <small>(37 CFR 1.16(s))</small>	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			0.00				
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				0.00				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	730		TOTAL		
APPLICATION AS AMENDED - PART II								
AMENDMENT A	(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY	OR OTHER THAN SMALL ENTITY
		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA		RATE(\$)	ADDITIONAL FEE(\$)
	Total <small>(37 CFR 1.16(i))</small>	*	Minus	**	=		x	=
	Independent <small>(37 CFR 1.16(h))</small>	*	Minus	***	=		x	=
	Application Size Fee (37 CFR 1.16(s))							
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							
						TOTAL ADD'L FEE		TOTAL ADD'L FEE
AMENDMENT B	(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY	OR OTHER THAN SMALL ENTITY
		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA		RATE(\$)	ADDITIONAL FEE(\$)
	Total <small>(37 CFR 1.16(i))</small>	*	Minus	**	=		x	=
	Independent <small>(37 CFR 1.16(h))</small>	*	Minus	***	=		x	=
	Application Size Fee (37 CFR 1.16(s))							
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							
						TOTAL ADD'L FEE		TOTAL ADD'L FEE
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3. ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20". *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3". The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.								

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Guy Larri et al.

Application No.: 14/788,164

Confirmation No.: 1250

Filed: June 30, 2015

Art Unit: Not Yet Assigned

For: A SPEECH RECOGNITION CIRCUIT AND
METHOD

Examiner: Not Yet Assigned

RESPONSE TO NOTICE OF INCOMPLETE REPLY

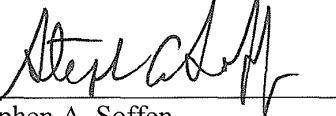
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Madam:

In response to the Notice of Incomplete Reply dated September 24, 2015, applicants hereby submit a substitute specification with proper line spacing. A marked up copy of the specification is attached, identifying the portions in which the line spacing has been changed from 1.0 (single line) to 1.5 (space and a half). The substitute specification contains no new matter.

Dated: September 25, 2015

Respectfully submitted,

By 

Stephen A. Soffen

Registration No.: 31,063
DICKSTEIN SHAPIRO LLP
1825 Eye Street, NW
Washington, DC 20006-5403
(202) 420-2200
Attorney for Applicant

A SPEECH RECOGNITION CIRCUIT AND METHOD

This is a continuation of application Serial No. 13/735,091, filed January 7, 2013, now Pat. 9,076,441, which is a continuation of application Serial No. 13/162,128, filed June 16, 2011, now Pat. 8,352,262, which is a divisional of application Serial No. 11/662,704, filed March 14, 2007, now Pat. 7,979,277, which is a 371 of PCT/GB2005/003554, filed September 14, 2005, which claims priority to UK Application No. 0420464.0, filed September 14, 2004, the disclosures of which are hereby incorporated by reference in their entireties.

BACKGROUND OF THE INVENTION

1. Field of the Invention:

The present invention relates to speech recognition circuits and methods. These circuits and methods have wide applicability, particularly for devices such as mobile electronic devices.

2. Description of the Related Art:

There is growing consumer demand for embedded speech recognition in mobile electronic devices, such as mobile phones, dictation machines, PDAs (personal digital assistants), mobile games consoles, etc. For example, email and text message dictation, note taking, form filling, and command and control applications are all potential applications of embedded speech recognition.

However, when a medium to large vocabulary is required, effective speech recognition for mobile electronic devices has many difficulties not associated with speech recognition systems in hardware systems such as personal computers or workstations. Firstly, the available power in mobile systems is often supplied by battery, and may be severely limited. Secondly, mobile electronic devices are frequently designed to be as small as practically possible. Thus, the memory and resources of such mobile embedded systems tends to be very limited, due to power and space restrictions. The cost of providing extra memory and resources in a mobile electronic device is typically much higher than that for a less portable device without this space restriction. Thirdly, the

DSMDB-#3348563-v1

mobile hardware may be typically used in a noisier environment than that of a fixed computer, e.g. on public transport, near a busy road, etc. Thus, a more complex speech model and more intensive computation may be required to obtain adequate speech recognition results.

These restrictions have made it difficult to implement effective speech recognition in mobile devices, other than with very limited vocabularies.

Some prior art schemes have been proposed to increase the efficiency of speech recognition systems, in an attempt to make them more suitable for use in mobile technology.

In an article entitled “A low-power accelerator for the SPHINX 3 speech recognition system”, in University of Utah, International conference on Compilers, Architectures and Synthesis for Embedded Systems, Nov 2003, Davis et al have proposed the idea of using a special purpose co-processor for up-front calculation of the computationally expensive Gaussian output probabilities of audio frames corresponding to particular states in the acoustic model.

In an article entitled “Hardware Speech Recognition in Low Cost, Low Power Devices”, University of California, Berkeley, CS252 Class Project, Spring 2003, Sukun Kim et al describe using special purpose processing elements for each of the nodes in the network to be searched. This effectively implies having a single processing element for each phone in the network. An alternative suggested by Sukun Kim et al is to provide a processor for each state in the network.

In an article entitled “Dynamic Programming Search for Continuous Speech Recognition” in IEEE Signal Processing Magazine, Sept 1999, Ney et al discuss language model lookahead. Language model lookahead involves computation of a language model factor for each node (i.e. phone) in the lexical tree. This technique is also known as smearing. Each phone instance in the search network can be given a language model factor when it is used in the lexical tree search. Ney et al show that for an example bigram language model, the average number of states per 10 ms frame can

DSMDB-#3348563-v1

be reduced from around 168,000 states with no language model lookahead to around 8,000 states when language model lookahead is used. They also show that bigram language model lookahead requires about a quarter of the states compared with unigram language model lookahead.

Although these prior art documents provide improvements to speech recognition in embedded mobile technology, further improvement is still needed to provide a larger vocabulary and better accuracy.

SUMMARY OF THE INVENTION

One aspect of the present invention provides a speech recognition circuit including a circuit for providing state identifiers which identify states corresponding to nodes or groups of adjacent nodes in a lexical tree, and for providing scores corresponding to said state identifiers. The lexical tree includes a model of words. The speech recognition circuit also has a memory structure for receiving and storing state identifiers identified by a node identifier identifying nodes or groups of adjacent nodes, the memory structure being adapted to allow lookup to identify particular state identifiers, reading of the scores corresponding to the state identifiers, and writing back of the scores to the memory structure after modification of the scores. An accumulator is provided for receiving score updates corresponding to particular state identifiers from a score update generating circuit which generates the score updates using audio input, for receiving scores from the memory structure, and for modifying said scores by adding said score updates to said scores. A selector circuit is used for selecting at least one node or group of nodes of the lexical tree according to said scores.

One suitable type of hardware for the memory structure includes a content addressable memory (CAM). A CAM is a memory unit which stores a series of data items using a series of addresses. However, the memory is accessed by specifying a data item, such that the CAM returns the corresponding address. This contrasts with a random access memory (RAM) in which the memory is accessed by specifying an address, such that the RAM returns the corresponding data item.

DSMDB-#3348563-v1

However, the memory structure is not limited to including a CAM. Other types of hardware are also possible, to provide this functionality. For example, a single chip which operates in the same way as a CAM and RAM may be used instead.

Embodiments of the present invention provide a solution to the problem of how to map a lexical tree search to a CAM system architecture. The realisation by the present inventors that certain speech recognition data structures can be mapped into the CAMs allows a lexical tree search to be performed using a CAM system architecture.

Further embodiments of the invention include a counter for sequentially generating state identifiers, and using said generated state identifiers to sequentially lookup said states in the memory structure.

The node identifier may comprise a direct reference to the lexical tree. However, in some embodiments, the node identifier for at least some of the states includes a pointer to a node identifier for another state. For example, a state corresponding to the furthest part of the search path in the lexical tree may be referenced by a node identifier which directly links to a particular node or group of nodes in the lexical tree. In a lexical tree comprising phones, using a state model of triphones, the node identifier may indicate the position of a triphone in the lexical tree.

However, in this example, for states occurring further back in the search path, instead of supplying a node identifier linking directly to the lexical tree, instead a pointer to a node identifier of another state may be supplied. E.g. a triphone instance may have a pointer to another triphone instance, which has a pointer to another triphone instance, which has a pointer to a node or group of nodes in the lexical tree. Chains of reference may be set up in this way, where only the last state in the chain has a direct pointer to the lexical tree.

There may not be a one-to-one correspondence between the nodes of the lexical tree and the node identifiers. This will occur for a branched lexical tree, where the nodes represent monophones, but the acoustic model states represent triphones, i.e. groups of three adjacent monophones. Then, paths of three monophones will have unique

DSMDB-#3348563-v1

identifiers to be stored in the memory structure, rather than single monophones having unique identifiers.

Phone instance numbers may be generated, and used to uniquely label each phone instance. They can be generated sequentially, using a counter. The phone instance numbers may be used as pointers between phone instances to assist in node identification. It is thus not essential to provide a direct node identifier for each phone instance to directly indicate a location in the lexical tree. The dynamic network of phone instances provided in the memory structure may thus include both direct and relative references to the lexical tree.

The memory structure may be divided into one part which stores phone instance identifiers and direct references to the lexical tree, and a second part which stores phone instance identifiers and corresponding states. This can speed up the processing, by only storing the phone instances which are furthest on in the lexical tree in the first part of the memory structure.

The memory structure may also be divided into separately accessible units, to reduce the amount of data in each unit, thereby decreasing the chance of finding the same two states identifiers in different phone instances in any single memory unit, and increasing the chance of some state identifiers being completely absent from any single memory unit. This makes it easier to deal with the situation when the same two state identifiers are found, because a spare time slot is available for processing when a state identifier is not present.

A further aspect of the invention provides a distance calculation engine within a speech recognition system. The distance calculation engine may be included within an accelerator. The accelerator may include logic to interface with other parts of a speech recognition circuit, in addition to the distance engine, although this is not essential. For example, the accelerator may include one or more results memories for storing distances calculated by the distance calculation engine. The accelerator may also include at least one of a memory for storing one or more acoustic models, a decompressor for decompressing acoustic data that has been stored in a compressed format, a memory for

DSMDB-#3348563-v1

storing feature vectors, a checksum or data signature calculation means, buffers for data storage, and data registers. The accelerator may be implemented in software or in hardware, or in a combination. It may be physically separate to the rest of the speech recognition circuit, although this is not essential.

The distance calculation engine may calculate one or more of a wide range of distance metrics and probability distributions. The distances may represent the likely correspondance of feature vectors to states in an acoustic model. In other words, the distances can indicate the similarity of an audio data frame to each possible state in an acoustic model

There are a wide variety of probability distributions that can be used for the distance calculation stage of a speech recogniser, and a wide variety of distance metrics used. These are widely documented in the literature. A point is a simple example of a probability distribution.

A common choice is to use Gaussian Distributions and correspondingly the Mahalanobis Distance metric. The Gaussian probability distribution is then defined by a mean vector, which defines centre point in the N-dimensional space, and a Covariance matrix which defines the shape of the probability distribution. It is common to restrict the Covariance matrix to be a diagonal matrix (only N non-zero values along the diagonal of the NxN matrix) which significantly lowers the implementation cost by reducing the number of arithmetic operations.

In particular embodiments, the distance calculated is a Mahalanobis distance. Particular examples of this are described later in the specification.

In one embodiment, the distance engine autonomously computes all of the distances associated with a given feature vector. This may comprise computing distances for every state in the lexicon. The distance engine may operate in a pipelined manner with other stages of the recognition process. In this context a distance is an indication of the probability or likelihood that a feature vector corresponds to a particular state. An

important class of distance computation in speech recognition is the calculation of output state probabilities in recognisers using Hidden Markov Models. Another possible use is in recognisers using Neural Networks.

The distance engine reads data from the acoustic models to use as parameters in the calculation. The acoustic models may be optionally stored in a compressed format. The distance engine may read and de-compress the acoustic models one (or more) times for each feature vector processed. Each reading of the acoustic models may require reading the entire acoustic model, or various optimisations may be implemented to avoid reading parts of the acoustic model that are not required to be used for calculations with the current feature vector. The distance engine may use a de-compression method where the de-compression is sign or zero extension or may otherwise convert data of narrow or variable width to a wider data format. The distance engine may use a de-compression method where the de-compression is sign or zero extension or may otherwise convert data of narrow or variable width to IEEE standard single or double precision floating point format. The distance engine may use a decompression method where decompression is a codebook decompression of a binary bitstream, where the codebook is stored as part of the acoustic model data. The distance engine may use a decompression method where the decompression is decompression of a Huffman or Lempel-Ziv compressed stream. The distance engine may use a decompression method where decompression is decompression of run length encoded data. The distance engine may use a decompression method where decompression is decompression of difference encoded data. The distance engine may use a decompression method using any well known lossy or lossless compression scheme. The distance engine may use a decompression method using subspace distribution clustering. The distance engine may use a decompression method comprising any combination of the above described decompression types.

The distance engine may read the acoustic models from a dedicated on-chip memory. The distance engine may read the acoustic models from a dedicated off-chip memory. The distance engine may read the acoustic models from a shared on-chip memory. The distance engine may read the acoustic models from a shared off-chip memory. Any of these acoustic models may be compressed.

The distance engine may compute a CRC or checksum or similar signature as it reads in the acoustic model and compares this to a stored CRC, checksum, or signature, in order to check that the acoustic model has not been corrupted, and signals an error condition if such corruption is detected. The stored CRC, Checksum, or signature may have been pre-computed and stored in the model data, or it may be computed at the time the model data is loaded into the Acoustic Model Memory. It may be held in Acoustic Model Memory, or it may be loaded into a register or another memory from where it can be accessed and compared when the CRC/checksum/signature is computed each time the Acoustic Model is loaded.

The distance engine may support the pass-through of data from the front-end to the search-stage. The data to be passed through will be supplied to the distance engine as an adjunct to the feature vector, and the distance engine will pass it to the search stage as an adjunct to the distance results. This provides a simple mechanism for passing frame-specific data that is not involved in the distance calculation through to the search stage, and keeping it associated with the correct frame, which may otherwise be complex in pipelined systems with multiple processors. The data passed through may be for any purpose. Examples might include silence detected, end-of-audio-stream detected, a frame number, information that some intervening frames have been dropped, or data from another input device such as a button or keyboard in a multi-modal interface.

The distance engine may be implemented in hardware, software, or a combination. Other stages may be implemented in hardware, software, or a combination. The distance engine may be implemented with any number representation format including fixed point or floating point arithmetic, or any mixture of number representation formats.

In particular, the other stages may be implemented on a CPU, or on a DSP and CPU. The "DSP" and "CPU" may each be implemented as software programmable devices.

The distance engine may implement one or more additional pipeline stages to overcome delays introduced by low bandwidth, high latency, or conflicted bus interfaces. The distance engine may also implement additional pipeline stages to maintain the same

DSMDB-#3348563-v1

throughput while allowing more time for each distance calculation. Particular embodiments of the invention may include one or more of the above aspects.

A further aspect of the invention comprises a speech recognition circuit, comprising: an audio front end for calculating a feature vector from an audio signal, wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame; calculating circuit for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words; a buffer memory between the calculating circuit and the search stage, for receiving data passing from the calculating circuit to the search stage, wherein a processor in the search stage has higher bandwidth and/or lower latency access to the buffer compared to the bandwidth and/or latency of direct transfer between the calculating circuit and the search stage. The data transfer from the calculating circuit to the buffer memory and/or from the buffer memory to the search stage may be performed as one or more sequential bursts. The data transfer to the buffer memory may be performed in parallel with data transfer to the calculating circuit and/or in parallel with data transfer to the search stage. A second buffer memory may be provided between the audio front end and the calculating circuit.

A further aspect of the invention comprises a speech recognition circuit, comprising: an audio front end for calculating a feature vector from an audio signal, wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame; calculating circuit for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words; comprising an elastic buffer between at least one of the front end and calculating circuit, or the calculating circuit and search stage, and/or for buffering said audio signal.

A further aspect of the invention comprises an accelerator for a speech recognition circuit, the accelerator comprising: calculating means for calculating a distance

DSMDB-#3348563-v1

indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model, wherein the feature vector comprises a plurality of extracted and/or derived quantities from an audio signal during a defined audio time frame; means for comparing a first version of a stored checksum of data representing said acoustic model and a second version of said stored checksum, wherein the second version is obtained from an updated measurement and calculation of the checksum; and means for indicating an error status if the checksums do not match.

A further aspect of the invention comprises an accelerator for a speech recognition circuit, the accelerator comprising: calculating means for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model, wherein the feature vector comprises a plurality of extracted and/or derived quantities from an audio signal during a defined audio time frame; wherein said accelerator is configured to autonomously compute distances for every acoustic state defined by the acoustic model.

A further aspect of the invention comprises an accelerator for calculating distances for a speech recognition circuit, the accelerator comprising: calculating circuit for calculating distances indicating the similarity between a feature vector and a plurality of predetermined acoustic states of an acoustic model, wherein the feature vector comprises a plurality of extracted and/or derived quantities from an audio signal during a defined audio time frame; first and second storage circuit, which may be referred to as result memories, each for storing calculated distances for at least one said audio time frame, and for making said stored distances available for use by another part of the speech recognition circuit; control circuit for controlling read and write access to the first and second storage circuit, said control means being configured to allow writing to one said storage means while the other said storage means is available for reading, to allow first calculated distances for one audio time frame to be written to one said storage means while second calculated distances for an earlier audio time frame are made available for reading from the other said storage means.

Embodiments of the invention may comprise means for generating a checksum or computed signature for the acoustic model data stored in the memory, and means for comparing checksums or computed signatures that have been calculated at different

DSMDB-#3348563-v1

times, to indicate an error status if the checksums do not match, one possible cause of such mismatch being that the acoustic model data has been overwritten by said other data and said error status being used to indicate that the acoustic model should be re-loaded into the said memory.

A further aspect of the invention comprises a speech recognition circuit comprising: lexical memory containing lexical data for word recognition, said lexical data comprising a lexical tree data structure comprising a model of words; means for accessing a state model corresponding to each phone or each group of phones in the lexical tree; a content addressable memory for storing content addressable data for each phone or group of phones, including states corresponding to said phone or group of phones, and for storing an address value for each said phone or group of phones; a RAM configured to store accumulated scores for each said phone or group of phones, the accumulated scores being addressable by said address value for each said phone or group of phones; means to obtain scores that each of a plurality of frames of an audio signals corresponds to each of a plurality of said states; a counter to sequentially search for each said state in the content addressable memory, to obtain the corresponding address value if the state is found in the content addressable memory; means to use said address value to access an accumulated likelihood and an accumulator to add said likelihood to the accumulated likelihood; means to use the phones or groups of phones with the highest accumulated scores to obtain a plurality of next phones from the lexical tree which correspond to the next phone; and output means for outputting a lexical tree path of highest likelihood.

A further aspect of the invention comprises speech recognition apparatus comprising: a lexical tree having a corresponding state model; means for obtaining scores of an audio input corresponding to each of a plurality of states in said state model; a content addressable memory for storing a marker indicating a part of the lexical tree, and one or more states associated with said part of the lexical tree; a random access memory addressable by the CAM output, to output accumulated scores for states corresponding to said parts of the lexical tree; adder means for adding likelihood to said accumulated likelihood, to be stored back in the RAM.

A further aspect of the invention comprises speech recognition apparatus, comprising: a CAM-RAM arrangement for storing records including pointers to a lexical tree, and accumulative scores for states within the lexical tree; input means for obtaining scores that an audio frame corresponds to a particular state in the lexical tree; an accumulator for calculating the updated scores and modifying the records in the CAM-RAM accordingly; output means for outputting a path of highest likelihood in the lexical tree.

A further aspect of the invention comprises a speech recognition method comprising: storing state identifiers which identify states corresponding to nodes or groups of adjacent nodes in a lexical tree, and scores corresponding to said state identifiers in a memory structure, the lexical tree comprising a model of words and the memory structure being adapted to allow lookup to identify particular state identifiers, reading of the scores corresponding to the state identifiers, and writing back of the scores to the memory structure after modification of the scores; repeating the following sequence of steps for each of a plurality of incoming frames of an audio signal; obtaining score updates corresponding to the likelihoods that said frame of the audio signal corresponds to each of a plurality of said states; accessing said memory structure to obtain scores, updating the scores by adding score updates to the scores, and writing back the updated scores to the memory structure; determining if scores for states furthest on in the lexical tree correspond to a significant likelihood, and if so, then accessing the lexical tree to determine the next set of possible states; and storing the next set of possible states and said scores of significant likelihood in the memory structure.

A further aspect of the invention comprises speech recognition circuit comprising: a circuit for providing state identifiers which identify states corresponding to phones or groups of adjacent phones in a lexical tree, and for providing scores corresponding to said state identifiers, the lexical tree comprising a model of words; a memory structure for receiving and storing state identifiers and phone instance identifiers uniquely identifying instances of phones or groups of phones in the lexical tree; said memory structure being adapted to allow lookup to identify particular state identifiers, reading of the scores corresponding to the state identifiers, and writing back of the scores to the memory structure after modification of the scores; an accumulator for receiving score updates corresponding to particular state identifiers from a score update generating

DSMDB-#3348563-v1

circuit which generates the score updates using audio input, for receiving scores from the memory structure, and for modifying said scores by adding said score updates to said scores; and a selector circuit for selecting at least one phone instance identifier according to said scores.

A speech recognition apparatus according to the invention may be embedded in or included in a mobile electronic device such as a mobile telephone, PDA (personal digital assistant), etc.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figures 1A-1B are a block diagram of the system architecture for a speech recognition apparatus according to an embodiment of the invention;

Figures 2A-2B are a block diagram showing the main data structures used in the speech recognition apparatus of Figures 1A-1B;

Figures 3A-3B show a block diagram of the Single Phone Instance Network Engine (PINE) of Figure 1B;

Figure 4 is a block diagram showing the Wave Front Manager architecture;

Figure 5 shows the use of link nodes to permit path merging;

Figure 6 shows a block diagram of a Path Merge Unit in the Phone Level;

Figure 7 shows the Phone Book architecture;

Figures 8A-8B show Lexical tree storage;

DSMDB-#3348563-v1

Figure 9 shows the data structure for the word link record;

Figure 10 shows a block diagram of Word Engine;

Figure 11 shows a Language Model interface architecture;

Figure 12 shows an example of a simple Loop Grammar;

Figure 13 shows a block diagram of grammar engine;

Figure 14 shows a single frame of the Audio Codec link protocol; and

Figure 15 illustrates the system timing which may be used in the embodiment of Figures 1A-1B;

Figure 16 is a block diagram showing a speech recognition circuit according to an embodiment of the invention, and illustrating data flow between parts of the circuit;

Figure 17 is a block diagram showing a distance calculation engine according to an embodiment of the invention;

Figure 18 shows ideal data flow over several time steps within a speech recognition circuit having separate front end and search stages, in an embodiment of the invention;

Figure 19 shows an example of actual data flow over several time steps within a speech recognition circuit having separate front end and search stages, in a further embodiment of the invention;

Figure 20 shows ideal data flow over several time steps within a speech recognition circuit having a shared processor for front end and search stages, in an embodiment of the invention;

Figure 21 shows an example of actual data flow over several time steps within a speech recognition circuit having a shared processor for front end and search stages, in an embodiment of the invention;

Figure 22 shows ideal data flow in a speech accelerator pipeline with two data transfer stages, according to an embodiment of the invention; and

Figure 23 shows an example of actual data flow in a speech accelerator pipeline with two data transfer stages according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figures 1A-1B show a speech recognition system architecture for an embodiment of the invention. The speech recognition system is divided into several levels of operation: a state level, a phone level, a word level, a grammar level, and a system level. Each level is shown in Figures 1A-1B as a, interconnected to other levels of the system.

The information passed between different levels or subcircuits of Figures 1A-1B is indicated on Figures 1A-1B using the following identifiers.

Identifier	Definition
grammar_id	Identifies which grammar model is currently being used
grammar_node_id	Identifies a node in the grammar model
word_id	Uniquely identifies a word
wordlat_node_id	Identifies a node in the word lattice
LT_id	Lexical tree identifier. Identifies the lexical tree to be used
LT_branch_id	Identifies a unique branch in a lexical tree. A branch is defined as all of the monophones subsequent to and including the first monophone in a tree.
LT_node_id	Identifies a node in a lexical tree
monophone_id	Identifies a monophone
transvec_id	Identifies which transition vector to use
state_id	Identifies the state
phone_no	Unique phone instance identifier
prev_phone_no	Phone number of previous phone in network
triphone_id	Triphone identifier

DSMDB-#3348563-v1

LML_factor	Language model lookahead factor. The smeared language model score that must be added
------------	--

The system level includes a general purpose processor and cache, a system controller, a DMA (direct memory access) controller, clocks, PLL (phased locked loop), GPIO (general purpose I/O interface), timer watch dog, memory interfaces to interface to main memory and non volatile storage, serial, host, debug and display interfaces. The system level also includes a model adaptation engine, a language model factor engine, a speech recognition system controller, a language model interface, and a recognition output interface.

The processor organises the transfer of the various models from the non volatile storage to the various model memories. It handles the language model factor requests. Language model factors are used to by the state level to apply the language model score. The language model (LM) interface provides the mechanism for obtaining LM factors as and when they are needed. The processor keeps a copy of the wave front in memory which is updated simultaneously with that in the wave front manager. Using its copy of the lexical tree, the processor can determine which language model scores may be needed in the coming frames. Once calculated, the language model factors can be kept in memory for use in subsequent frames. The system level is also used for forming speech model adaptation, and acting as an overflow for the various buffers used by the architecture.

The system level includes the audio codec interface and the audio front end interface, connected to the processor via a APB (advanced peripheral bus) to AHB (advanced high performance bus) bridge.

The grammar level includes a grammar engine and grammar model(s). The grammar level sets out the rules of the particular recognition task at the highest level. During system initialisation a number of grammar models might be loaded, for example a command grammar and a dictation grammar. Command mode grammar is a restricted grammar which includes only a very limited lexical tree with command specific words. Some branches of this lexical tree may include multiple words together, where a

DSMDB-#3348563-v1

command has such a structure, e.g. “open file menu”. Typically, only a single instance of the lexical tree will be needed. Command mode is used because it can obtain a higher accuracy, due to the lexical tree constraint. In dictation mode grammar, a full lexical tree structure is used.

The grammar may be selected by a user during the recognition process. For example, the command grammar may be selected when the user presses a command button. The architecture supports fast swapping of grammar models. For example, a user may be in the middle of dictating a message when a “punctuation button” is pressed and the phrase “question mark” uttered. At this point, a punctuation grammar could be loaded, in order to increase recognition accuracy.

The word level comprises a word engine, which includes a word lattice and a word graph generator. The word level has two main functions, of keeping a record of the search path at the word level using a word lattice, and interfacing to the language model via a cache. The word lattice is a record of all of the words that have been searched up to the current frame. As new words are searched, they are appended into the lattice at the appropriate place. After the final frame in the current sentence, the n-best word lattice node identifiers from the n-best scoring tokens are passed back to the word lattice. This enables the n-best sentences to be traced back through the lattice. The word lattice can also be post processed to produce a word graph which is essentially a compact version of the word lattice.

At the word level, a record is kept of all tokens that have passed through a word end. This results in a lattice of words. When the final feature vector has been processed, the best scoring token is identified and used to trace back the search path back to the start of the search.

The phone level includes a “phone book” for generating new triphone instances, a wave front manager for sending requests to the phone book to generate new phone instances, a phone request arbiter, and a path merge unit.

The phone book includes a lexical tree, which comprises a vocabulary of words, sounds and/or phrases organised according to the phones or monophones making up the sound.

DSMDB-#3348563-v1

A lexical tree is commonly used in speech recognition rather than a linear lexicon, because it greatly reduces the amount of storage space required to store a word vocabulary, by avoiding duplication of common word beginnings. It is also easier to search when the word beginning is known.

Although it is possible to search a lexical tree based on monophones, it is well known that better results can be achieved in speech recognition by basing the lexical tree search on groups of more than one monophone, since the sound of one syllable commonly affects the pronunciation of the next syllable in the same word, or the first syllable in the next word. In this embodiment, groups of three monophones are used, and these are known as triphones. It is alternatively possible to use a different number of monophones for each group. The phone generator in the “phone book” uses data from the lexical tree to generate triphones.

In one embodiment of the invention, approximately 45 different cued mono-phones are considered. This gives of order 60,000 tri-phones. Each triphone in this embodiment is modelled by a Hidden Markov Model (HMM) with three states. There is no correspondence between individual states and individual phones of the triphone. Alternatively, it is possible to model each triphone by a single state, or by any other number of states.

The phone level is responsible for generating the search space according to the word structure specified in the grammar model and the phone structure specified by the lexical tree. The search space can be envisaged as a wave moving through an acousto-linguistic search space. In order that the wave can be dynamically created, a record of the wave front phone instances must be maintained. This is done in a wave front phone CAM/RAM structure, within the wave front manager. When a token occurs in the final state of a phone instance in the PINE array at the front of the wave, a new phone instance must be generated at the phone level and passed to the PINE array.

The phone instance includes a phone_no, which is a phone instance identifier to identify the phone uniquely in the current search. The phone instance also includes a word lattice node identifier (labelled “word_lat_node_id”) to identify the phone’s position in the word lattice, state identifiers (labelled “state_ids”) for the HMM, storage to hold the

DSMDB-#3348563-v1

tokens as they move through the HMM and LM_factor which is used to by the PINE to apply the language model score. To handle the connectivity between phone instances, a phone instance points to a previous phone instance using the phone_no of the previous phone. This allows tokens to be passed between phones. A “token” is a hypothesized search path, represented by a non-zero score at a particular state, and tokens are propagated through the HMM with each time frame.

The phone instance network can be visualised as a wave with new phone instances being added to the front of the wave, and empty phone instances being removed from the back of the wave. In some circumstances, the back of the wave may be in the same position as the wave front. The back of the wave moves on through the lexical tree as tokens are either moved on or pruned.

Tokens, each representing a particular hypothesis, hop through the states in the HMMs, each hop consuming a feature vector. Each token explores all possible next states which typically involves placing a token in the next state and keeping one in the current state. When there is more than one token in a state, the least likely token can be removed because both have the same word history. When a token reaches the last state of a phone instance at the front of the wave, the next phone instance(s) are requested from the phone level. The phone level uses the lexical tree to create the new phones. As well as looking up the state identification numbers for each HMM state, it also incorporates the correct language model factor.

The language model factor engine is used to smear the language model scores across the lexical tree. This operates in two ways depending on whether the factor is needed for a word start phone(s) or word internal phones. For word internal phones, LM factors can be computed as needed, provided that the LM scores are available in cache. This is because in general the storage requirements for the LM scores are less than the storage of the LM factors for all of the factors in the branch. However when a token is in a word end phone there may a significant number of word start phones e.g. up to (monophones)² requiring a large amount of computation. To avoid this the LM engine will pre compute word start LM factors when it seems highly likely that a token will be

DSMDB-#3348563-v1

passed to the word starts. These word start LM factors will be held in the LM factors cache. The phone level is responsible for generating the search space according to the word structure specified in the grammar model and the phone structure specified by the lexical tree.

The state level shown in Figure 1B includes a phone instance network engine (PINE) array, a MHD (Mahalanobis distance) engine, an audio front end, a feature vector buffer for model adaptation, and compressed acoustic models. The state level is the lowest level, in which a network of phones is dynamically created.

The audio input for speech recognition is input to the front end unit of the state level. This audio input may be digital audio, or it may be analog audio which is converted to digital audio using an analog to digital converter. The audio input is divided into time frames, each time frame typically being of the order of 10ms.

For each audio input time frame, the audio signal is converted into a feature vector. This may be done by splitting the audio signal into spectral components. For example, in the present embodiment, the audio input signal is split into 12 frequency bands and one energy band, and a feature vector is constructed using these 13 components, plus their first and second derivatives, a total of 39 components.

The feature vector is passed to the MHD (Mahalanobis distance) engine. This computes all possible state output likelihoods for the current frame as the state mean and covariance data is read from the acoustic model memory.

The likelihood of each state is determined by the distance between the feature vector and each state. To reduce the number of states, a technique known as “state tying” may be used. Multiple states are grouped together in 39D space, according to their proximity to one another. An average or central value from each group is used to represent the entire group. Thus, the effective number of states may be reduced from 40,000 to 8,000. The accuracy may be improved using a language model in addition to the word recognition lexical tree. A language model is based on the probability of sequential

words. The above figure of 8,000 states gives best recognition results when used with a language model.

The MHD (Mahalanobis Distance) is a distance between two N-dimensional points, scaled by the statistical variation in each component. In embodiments of the invention, the Mahalanobis distance between the feature vector and each state is calculated, to determine similarity of the feature vector to each state.

The Feature Vector represents a point in an N-dimensional space. Different embodiments of the invention may use different values of N. Generally we see values of N in the range of 20 to 39, although values outside this range are not excluded.

The distance calculator stage of the recognition process computes a probability or likelihood that a feature vector corresponds to a particular state. Each state has an associated probability distribution in the N-dimensional space. The distance calculator computes the distance in the N-dimensional space from the Feature Vector to the probability distribution for each state. In some embodiments, the calculation is performed only when the search stage algorithms determine that the distance for that particular state is needed by the search stage. In some embodiments, the distances for all of the states are calculated and made available to the search stage, which allows pipelined operation. Embodiments that do the calculations on demand are preferable.

The Mahalanobis Distance (MHD) is extensively described in the literature. It may be calculated using the formula

$$\text{MHD value} = \ln(P(x)) = -0.5[N \cdot \ln(2 \cdot \pi) + \ln|\text{Cov}|] + (-0.5 \cdot [(x-u)' \cdot \text{inverse}(\text{Cov}) \cdot (x-u)])$$

where

N= number of dimensions in the space

x=feature vector, a point in the N dimensional space

u=mean vector

Cov=covariance matrix

|Cov|=determinant of covariance matrix

(a)'= transpose of matrix a

DSMDB-#3348563-v1

Different embodiments may use variations on this equation, for example the base for the logarithm is not always the natural base e that is used in the above example. The log of the probability is preferably calculated, rather than the probability, as it simplifies the implementation. Not all embodiments do this. In alternative embodiments, the likelihoods rather than probabilities are computed as it leads to simpler implementations of computations in the search stage.

The feature vector is used to calculate 8,000 MHD distances for each time frame, i.e. one distance for each of the 8,000 states. It is beneficial, although not essential, to perform the MHD calculations separately in a pre-processor, known as a MHD engine, because MHD calculation is a computationally intensive process.

Due to the 10ms frame length, a feature vector arrives at the MHD engine of the state level every 10ms. The MHD engine calculates the likelihood of each feature vector corresponding to each of a number of states derived from an acoustic model. Each state is also a 39 dimensional vector, having the same spectral components as the feature vector. The MHD engine calculates the difference in 39-D space between the feature vector and each state to obtain the likelihood for that state. A parallel MHD or a single MHD arrangement may be used.

The feature vectors may be sent to a feature vector buffer for model adaptation, which determines whether the observed features fit well with the acoustic model, and may initiate an adaptation of the model if the fit is not sufficiently good.

Scores indicating the state likelihoods or probabilities are output from the MHD engine to the PINE array. It is preferable to use likelihoods rather than probabilities, because likelihoods can be added together to give a total likelihood. Probabilities must be multiplied, which can give rise to rounding errors. In this embodiment, eight separate pines are used, each of 4 kilobytes, thus storing 4000 phones per pine.

During recognition, the Phone Instance Network Engine (PINE) Array will contain a network of phone instances. Tokens representing search hypotheses in the form of a

DSMDB-#3348563-v1

probability(likelihood) will exist in some of the HMM states of the phone instances. The tokens are propagated into all available states and updated with the corresponding state output probabilities. Maximum model, word end and beam pruning are applied to the tokens to ensure the search space is kept bounded. When a token is in the final state of a phone model and has avoided being pruned, the PINE makes a next phone request to the Phone level. The PINE array may receive a number of new phone models from the phone level which must be appended to the appropriate existing phone.

Figure 2B shows the data structures stored in the PINE array, and other main data structures used during the acoustic search. Each PINE includes a phone instance CAM and a phone instance RAM. The phone instance CAM stores the phone instance number (labelled “phone_no”) which uniquely identifies each phone instance. The phone instance CAM also stores the state IDs of the three HMM states used to model the phone, and flags indicating the position of the phone. The “back” flag indicates if the phone is at the back of the wavefront. The “front” flag indicates if the phone is at the front of the wavefront. The “empty” flag indicates if there are no tokens in any of the three states for that phone. The “end” flag indicates if the phone is at the end of a word.

The phone instance RAM stores the token scores for each of the three states corresponding to each phone. The st_dum field indicates a dummy state, which is used to represent a state prior to the first state of the first phone. The “tmp” field is used for passing a token from a previous phone in the lexical tree to the current phone. This will be explained in more detail later. The prev_phone_no field is for storing a pointer to the previous phone instance in the network. The wordlat_node_id identifies the position of the phone instance on the word lattice node, and the LM_factor stores the language model factor for that phone instance.

At the phone level a record of all phones on the wave front must be maintained. If a phone instance is on the wave front, the phone_no stored in the phone instance CAM will point to the details in the wave front phone CAM and RAM.

The wave front phone CAM and RAM keeps pointers into the grammar and lexical tree. When a token in a wave front phone reaches the last state, the PINE array makes a

DSMDB-#3348563-v1

request to the phone level by passing back the phone_id. The phone_id is used as a key to determining the token's grammar_node_id and LT_node_id within the wave front CAM/RAM structure. If the token is at a word end the grammar model is checked. A grammar identifier points to the type of grammar.

At the word level, figures 2A-2B show an example word lattice structure, with silence having a word_id of 0, and a wordlat_node_id of 6. This is shown as connected to two word nodes – “which” has a word_id of 238 and a wordlat_node_id of 4, and “neither” has a word_id of 3556 and a wordlat_node_id of 5.

At the highest level, i.e. the grammar level, a simple grammar model is shown which will allow the phrase ‘silence which house silence’ or ‘silence neither house silence’ to be spoken. Whenever a word end is reached during the search, the grammar model must be consulted to determine the next possible word(s) or lexical tree(s). This allows the word lattice at the word level, consisting of a network of words, to be assembled. Whenever a word end is reached, the word is added to the lattice and the wordlat_node_id updated accordingly.

Figures 3A-3B show a single phone instance network engine (PINE). At system power up the state transition RAM will be loaded with the appropriate state transition vector but the CAM and RAM memories will be empty. When the system is initialised an initial set of phone model(s) will be instantiated with a token in the dummy state and a path score of zero. The phone models are received from the phone level.

When the audio data is processed, a stream of all possible state output probabilities will be received from the MHD engine. A counter that is synchronous to the stream of state output probabilities from the MHD produces a concurrent stream of state_ids. The job of the CAM is to perform a search to see if the state_id is instantiated anywhere in the CAM state_id fields. If it exists and the location doesn't have its empty flag set then this indicates that a token may need to be updated. The match signal from the CAM must be encoded into an address for the RAM. The corresponding RAM address can then be read. Any token that will end up in the identified state can then have its path score updated and including any transition vector likelihood.

DSMDB-#3348563-v1

It is possible that a single state_id generates multiple CAM matches. To handle this situation the CAM and RAM are effectively decoupled by means of a match fifo and repeat count fifo. The state_id and corresponding state_output_probability are stored in the repeat count fifo. During a subsequent frame when there is no match, this state_id can be revisited, ignoring the matches already dealt with.

All searched phone instances corresponding to words in the current possible lexical tree are stored in the CAM-RAM. This includes phone instances at the beginning of the word, middle of the word and end of the word. These phone instances are connected by the “previous phone instance” field, which allows traceback through an entire word within the CAM-RAM.

A mechanism is thus required to pass tokens from one phone instance to the next. This mechanism is provided by the tmp field in the RAM.

The tmp field in the RAM holds tokens during token passing.

At the start of each time frame, the audio input is converted into a feature vector, the MHD values calculated, and the MHD values used to update the token scores. After update of all tokens, 2 passes are made through the CAM-RAM, firstly for pruning, and then for token passing. The pruning in this example removes tokens of below a threshold score, although other types of pruning are also possible. The pruning may involve removing a predetermined proportion of the tokens, chosen as those of lowest likelihood. A third possible method of pruning is removing all tokens with scores below a first predetermined value, but at word ends removing all tokens with scores below a second predetermined value, the second predetermined value being greater than the first predetermined value.

The token passing works as follows. Each phone instance points back to a previous phone, due to the prev_phone_no field stored in the RAM. During the token passing, the third token of a previous phone is grabbed and put into the tmp field of the current phone.

In other words, if a token is on the third state, and it is not pruned, it must enter the next state of the next phone. To do this, the tokens that lie in the third state are moved across to the tmp field.

For example, phone no. 345 needs to look at phone no. 320 to see if any tokens are in the third state. If yes, it takes these tokens and puts them in the tmp field.

One field in the CAM is labelled “front”. This indicates whether the phone is on the wave front. If the phone is not on the wave front, then another phone will have picked up the token in the third state. However, if the phone is on the wave front, then no other phone will have picked up the token on the third state. Therefore, if there is a token in the third state, and the phone is on the wavefront, it is necessary to generate a next phone request.

The current phone number, word lattice identifier and word end flag are sent to the wave front manager, to request a next phone. If the word end flag indicates a word end, then the score is also passed to the wave front manager. The score can be passed in any case, but it is only required at a word end.

The phone level generates the next phone or phones. A new phone instance starts, and all of state1, state2 and state3 have scores pre-set to zero. The old score is provided in the tmp field.

The old phone is still within the wave, but it is unlikely to be at the back of the wave at this stage. When it gets to the back of the wave, it is deleted. Until it gets there, it remains in the CAM-RAM as before. The third token remains where it was, since tokens can either stay in the third state or move on.

Note that it is possible to end up with more than one token in each state. If this occurs, then the token with the poorer score is deleted. This does not affect the result of the lexical tree search, because both tokens on the same state have exactly the same word history.

Figure 4 shows a block diagram of the wave front manager. The wave front manager is responsible for maintaining a record of the set of phone models that exist at the front of the acoustic search space. This set of phone models is referred to as the “wave front.” The key requirement for this block is to minimise the time between receiving a “next phone request” and generating the appropriate responses. The wave front manager receives the next phone requests from the PINE array and generates appropriate responses for the Phone Book, Word Engine, Path Merge Unit and Grammar engine. The wave front manager also receives new wave front phone models from the Phone Book and updates the wave front.

The wave front manager receives the next phone request from the PINE array, via the phone request arbiter. The next phone request consists of a `phone_no`, `word_end_flag` and `wordlat_node_id`. If this phone is at the end of a word, the `word_end_flag` will be set.

The wave front manager uses the `phone_no` as input to the wave front phone CAM. As each `phone_no` is unique, it is guaranteed that only a single match will be generated. The wave front manager uses the CAM generated address to access the RAM. It then sends retrieved data from the RAM to the phone book, grammar engine, Path Merge Unit and word engine.

The wave front manager removes phone instance from CAM and RAM by writing 0's into CAM and RAM, and increments the empty count by one.

Sometime later, the wave front manager receives new phone instances from the phone book and place them in a fifo. The phone instances are taken from the fifo and written into the CAM and RAM at the `next_empty_address`. The `empty_count` is decremented by one. A match is run on the “occupy” flag, and the address is placed into `next_empty_address` register.

Both the phone level and the state level make use of a path merge unit to reduce the number of possible paths, by path merging. Figure 5 shows the use of link nodes to permit path merging. In the present embodiment, all word end phone instances are

DSMDB-#3348563-v1

linked to word start triphones with a link node. During each frame the path merge unit looks for *word end* tokens that can be recombined. This is done by looking for word end phones that have the same word_id and word history. When they are found a link is added using the path merge unit in the PINE array.

Figure 6 shows a diagram of the architecture at the Phone level, showing the path merge unit. The path merge unit includes a path merge CAM and a path merge RAM. The path merge CAM stores word_id, word history for two words, and an “occupy” field. The path merge RAM stores a link_node_id. A CAM RAM occupancy manager keeps track of the empty addresses in the CAM and RAM. The path merge unit is only passed information from the wave front manager if the phone is a *word end phone*. For each frame, there is a check to see if there is an identical word with the same history, and if there is, the link_node_id is passed to the path merge unit. If there isn’t an identical word with the same history, a new record is written into the Path merge CAM-RAM.

Figure 7 is a block diagram showing the Phone Book architecture. The Phone Book is tasked with generating the next phone models when given a pointer into the lexical tree. The key requirement for this block as to minimise the time between receiving a “lexical tree address” and generating the appropriate responses, in particular the next phone model instance(s).

The lexical tree is used to build the next triphones based on the current context.

Current triphone	Next triphone	comments
Left context = a	discarded	
Central context = b	Left context = b	next_leftmph_id stored in wave front manager
Right context = c	Central context = c	current LT_node_id stored in wave front manager
	Right context = ?	must be looked up. Large (< 45) for new word start

The LT_node_id, next_left_mph_id and prev_phone_no are received at the phone generator from the wavefront manager. The LT_node_id is then used to address the lexical tree store and return the central monophone_id. One or more further accesses to

memory are performed using the microcoded instructions, each memory access returning the right monophone_id.

A two stage lookup is accessed using the three monophone_ids (giving a triphone), the associated state_ids. The phone model is assembled, and returned to the PINE array. The wavefront manager is then updated by sending it relevant parts of the phone model.

The lexical tree store is a micro-coded ROM with both instruction and data fields. The lexical tree store holds a number of lexical trees. Each lexical tree may hold single words or multiple words. Each lexical tree comprises lexical tree elements, and each element has three fields. For now, the fields will be integers and will be converted to std_logic_vector later. The following is an example of the lexical tree store, with the LT data in word_id order.

ZERO	z i y r o w	19 8 12 11 0 0 0 0	sp 0
ONE	w a h n	18 2 10 0 0 0 0 0	sil 1
TWO	t u w	14 16 0 0 0 0 0 0	ah 2
THREE	th r i y	15 12 8 0 0 0 0 0	ay 3
FOUR	f o w r	6 11 12 0 0 0 0 0	eh 4
FIVE	f a y v	6 3 17 0 0 0 0 0	ey 5
SIX	s i h k s	13 7 9 13 0 0 0 0	f 6
SEVEN	s e h v i h n	13 4 17 7 10 0 0 0	ih 7
EIGHT	e y t	5 14 0 0 0 0 0 0	iy 8
NINE	n a y n	10 3 10 0 0 0 0 0	k 9
OH	o w	11 0 0 0 0 0 0 0	n 10
			ow 11

The following shows the LT data in MP (monophone) start order.

EIGHT	e y t	5 14 0 0 0 0 0 0	th 15
FIVE	f a y v	6 3 17 0 0 0 0 0	uw 16
FOUR	f o w r	6 11 12 0 0 0 0 0	v 17
NINE	n a y n	10 3 10 0 0 0 0 0	w 18
OH	o w	11 0 0 0 0 0 0 0	z 19
SEVEN	s e h v i h n	13 4 17 7 10 0 0 0	
SIX	s i h k s	13 7 9 13 0 0 0 0	
TWO	t u w	14 16 0 0 0 0 0 0	

DSMDB-#3348563-v1

THREE	th r i y	15 12 8 0 0 0 0 0
ONE	w a h n	18 2 10 0 0 0 0 0
ZERO	z i y r o w	19 8 12 11 0 0 0 0

The following is an example of the three fields of a lexical tree element. The first field is an instruction field, and the second and third fields are data fields.

Fields:	1st	2nd	3rd	
	instr	data	data	
	2b	6b	6b	
	00	MP	MP	- raw data field.
	01	br	br	- branch address.
	10	br	br	- branch address.
	11	word_id		- double field.

Note that there might be no difference between "01" and "10"

To build the "ey" tree, we only have to consider "EIGHT", because this is the only word in the lexical tree which starts with "ey". EIGHT is represented as "ey t", with phone_nos 5 14 0 0 0 0 0 0. The first lexical tree element stores the raw data specifying the phone_nos, the second lexical tree element is an end field, and the subsequent lexical tree element is set to zero.

```
0 00 5 14 - raw data
1 11 8 0 - end field
2 00 0 0
```

To build the "f" tree, we have:

FIVE	f a y v	6 3 17 0 0 0 0 0
FOUR	f o w r	6 11 12 0 0 0 0 0

```
0 01 4 0 -- branch forward four places as well
1 00 6 3 -- as the implied other brach of next (one).
2 00 3 17
```

3 11 5 0 -- all addresses are relative offsets.

4 00 6 11

5 00 11 12

6 11 4 0

7 00 0 0

To build the "n" tree, we have:

NINE n ay n 10 3 10 0 0 0 0 0

0 00 10 3

1 00 3 10

2 11 9 0

3 00 0 0

To build the "ow" tree, we have "ow" in both the first and second data fields, as no tree is going to be a single monophone long.

OH ow 11 0 0 0 0 0 0 0

0 11 11 0

1 00 0 0

To build the "s" tree, we have:

SEVEN s eh v ih n 13 4 17 7 10 0 0 0

SIX sih k s 13 7 9 13 0 0 0 0

0 01 6 0 -- branch forward six places.

1 00 13 4

2 00 4 17

3 00 17 7

4 00 7 10

5 11 7 0

6 00 13 7

7 00 7 9

8 00 9 13

9 11 6 0

10 00 0 0

DSMDB-#3348563-v1

To build the "t" tree, we have:

TWO t u w 14 16 0 0 0 0 0 0

0 00 14 16

1 11 2 0

2 00 0 0

To build the "th" tree, we have:

THREE th r i y 15 12 8 0 0 0 0 0

0 00 15 12

1 00 12 8

2 11 3 0

3 00 0 0

To build the "w" tree, we have:

ONE w a h n 18 2 10 0 0 0 0 0

0 00 18 12

1 00 12 10

2 11 1 0

3 00 0 0

To build the "z" tree, we have:

ZERO z i y r o w 19 8 12 11 0 0 0 0

0 00 19 8

1 00 8 12

2 00 12 11

3 11 0 0

4 00 0 0

Once all of the branches have been built, the next stage is to concatenate the tree. It appears that four elements make a page, and each element is two bytes.

		page
ey:	00 5 14 11 8 0 00 0 0 00 0 0	5
f:	01 4 0 00 6 3 00 3 17 11 5 0	6
	00 6 11 00 11 12 11 4 0 00 0 0	7
n:	00 10 3 00 3 10 11 9 0 00 0 0	8
ow:	11 11 0 00 0 0 00 0 0 00 0 0	9
s:	01 6 0 00 13 4 00 4 17 00 17 7	10
	00 7 10 11 7 0 00 13 7 00 7 9	11
	00 9 13 11 6 0 00 0 0 00 0 0	12
t:	00 14 16 11 2 0 00 0 0 00 0 0	13
th:	00 15 12 00 12 8 11 3 0 00 0 0	14
w:	00 18 12 00 12 10 11 1 0 00 0 0	15
z:	00 19 8 00 8 12 00 12 11 11 0 0	16

Then, the start_lookup header is defined. Each row (page) is 8 bytes and hence the header takes 3 rows. The end_lookup header is two rows. Therefore, the lexical tree store proper starts at the 6th (#5) page.

```

sp sil ah ay eh ey f ih      - MP
0 1 2 3 4 5 6 7              - MP_id
3 0 0 0 5 0 6 0              - data stored.

```

```

iy k n ow r s t th
8 9 10 11 12 13 14 15
0 0 8 9 0 10 13 14

```

```

uw v w z . . . .
16 17 18 19 . . . .
0 0 15 16 0 0 0 0

```


The end_lookup header is defined. 20 is out of bounds.

ey f n ow s t th w
5 6 10 11 13 14 15 18

z
19 20 20 20 20 20 20 20

To put the above information all together:

START	sp:ih	3 0	0 0	5 0	6 0	0
LOOKUP	iy:th	0 0	8 9	0 10	13 14	1
	uw: v	0 0	15 16	0 0	0 0	2
END	ey:w	5 6	10 11	13 14	15 18	3
LOOKUP	z	19 20	20 20	20 20	20 20	4
LEXICAL	ey:	00 5 14	11 8 0	00 0 0	00 0 0	5
TREE	f:	01 4 0	00 6 3	00 3 17	11 5 0	6
STORE		00 6 11	00 11 12	11 4 0	00 0 0	7
	n:	00 10 3	00 3 10	11 9 0	00 0 0	8
	ow:	11 11 0	00 0 0	00 0 0	00 0 0	9
	s:	01 6 0	00 13 4	00 4 17	00 17 7	10
		00 7 10	11 7 0	00 13 7	00 7 9	11
		00 9 13	11 6 0	00 0 0	00 0 0	12
	t:	00 14 16	11 2 0	00 0 0	00 0 0	13
	th:	00 15 12	00 12 8	11 3 0	00 0 0	14
	w:	00 18 12	00 12 10	11 1 0	00 0 0	15
	z:	00 19 8	00 8 12	00 12 11	11 0 0	16

3 0 0 0 5 0 6 0
0 0 8 9 0 10 13 14
0 0 15 16 0 0 0 0
5 6 10 11 13 14 15 18
19 20 20 20 20 20 20 20
00 5 14 11 8 0 00 0 0 00 0 0
01 4 0 00 6 3 00 3 17 11 5 0

DSMDB-#3348563-v1

```

00 6 11 00 11 12 11 4 0 00 0 0
00 10 3 00 3 10 11 9 0 00 0 0
11 11 0 00 0 0 00 0 0 00 0 0
01 6 0 00 13 4 00 4 17 00 17 7
00 7 10 11 7 0 00 13 7 00 7 9
00 9 13 11 6 0 00 0 0 00 0 0
00 14 16 11 2 0 00 0 0 00 0 0
00 15 12 00 12 8 11 3 0 00 0 0
00 18 12 00 12 10 11 1 0 00 0 0
00 19 8 00 8 12 00 12 11 11 0 0

```

This information is converted to hexadecimal. The first five are byte fields.

The rest are:

2bit - uCode.

6bit - data.

2bit - null.

6bit - data

```

03 00 00 00 05 00 06 00
00 00 08 09 00 0A 0D 0E
00 00 0F 10 00 00 00 00
05 06 0A 0B 0D 0E 0F 12
13 14 14 14 14 14 14 14
05 0E C8 00 00 00 00 00
84 00 06 03 03 11 C5 00
06 0B 0B 0C C4 00 00 00
0A 03 03 0A C9 09 00 00
CB 00 00 00 00 00 00 00
86 00 0D 04 04 11 11 07
07 0A C7 00 0D 07 07 09
00 0D C6 00 00 00 00 00
0E 10 C2 00 00 00 00 00
0F 0C 0C 08 C3 00 00 00
12 0C 0C 0A C1 00 00 00
13 08 08 0C 0C 0B C0 00

```

DSMDB-#3348563-v1

The above is the micro code for the digit lexical tree. It allows branching, and it holds the word_id. It knows which lexical trees it contains and where they are housed. It also stores the possible end monophone list.

Figures 8A-8B show a further example of the Lexical Tree RAM, giving examples of different types of branching in the lexical tree, and how these are represented in the RAM.

Figure 9 is a block diagram showing the data structure that is supported for the word link record, to enable word n-best decoding. The word engine maintains a record of the words that have been currently searched by the PINE array. It is used to determine the word history and allow trace back during or at the end of search. The word history is needed to determine the language model score and whether particular paths can be recombined or merged.

Figure 10 shows a block diagram of the word engine. This is designed to allow fast access to the word history (thus allowing rapid phone creation at the phone level), and to allow reasonably fast trace back. The Word Engine is primarily a memory store that fills incrementally. The decoupling of the wordlat_node_id and the physical address will allow the possibility of being able to keep the WLRs for a given word in contiguous addresses, thus avoiding the need for the pointer field.

The language model interface is designed to allow the Phone Book to quickly obtain the appropriate LM factor for the phone model that it is currently constructing. Based on the phones in the wave front, all possible next LM factors can be determined. This will be done in software together with the language model factor engine. The ARM 9 will maintain its own record of the wavefront using the same next phone requests received by the wave front manager and its own copy of the lexical tree.

Figure 11 shows the architecture of the LM interface. This allows the Phone book to quickly access the LM factor for the phone that is currently being generated. Prior to the phone generation phase LM_factors are computed for all possible phones that may be generated in the next frame.

DSMDB-#3348563-v1

The grammar engine is designed to give the Phone Book a quick response to requests for the next grammar node(s). In the case of a task grammar where the lexicon may be limited to a few words(e.g. yes, no), it may be desirable to use the same lexical tree. In this case it will be necessary for the grammar, to provide lexical tree pointers such that the phone generator can jump to the appropriate part of the lexical tree.

For a speech recognition system it is also necessary to specify the grammar. Again we will use HTK's Standard Lattice Format(SLF) to specify the grammar. In practice it will be convenient to use HTK's HParse format to specify the grammar using a high level grammar notation. This can then be converted into using the HParse tool. Consider having to specify a grammar for a digit recognition task. The HParse format for this is as follows:

```
$digit = one | two | three | four | five | six | seven | eight | nine | zero
$digit_sp = $digit sp
$noise = lipsmack | breath | sil
( [sil] < $digit_sp | $noise > [sil] )
```

where | denotes alternatives, < > denotes one or more repetitions, { } denotes zero or more repetitions, [] denotes that item is optional, and () contains the grammar.

Figure 12 shows the network that is defined by this grammar. The network starts and ends with a "sil" phone, representing silence. Between the start and end, a number of possible routes correspond to further silence, a breath, a lipsmack sound, or a detected word.

Figure 13 shows the grammar engine architecture. When a next phone request arrives, the next grammar node(s) must be retrieved from memory. In the case of a task grammar, the phone book should be given the lexical tree memory addresses to allow the phone to be determined.

Figure 14 shows a single frame of the Audio Codec (AC) link protocol used in some embodiments of the invention. The Advanced Audio CODEC Interface (AACI) is an

DSMDB-#3348563-v1

ARM Primecell that provides an interface to off-chip AC97 audio codecs. AC97 is Intel's audio codec specification, initially created for the PC market. The AACI is an AMBA slave block that connects to the Advanced Peripheral bus (APB). The AACI connects to the off-chip codec using the AC-link protocol.

The AC-link protocol is a bi-directional, fixed clock rate, serial digital stream. AC-link handles multiple input and output PCM audio streams as well as register control accesses. It employs a time division multiplexed (TDM) scheme that divides each audio frame into 12 outgoing and 12 incoming data streams each with up to 20 bit sample resolution. The tag slots are used to indicate which of the subsequent slots contain valid data. Add and dat are the control register address and data slots for read and writes. The adc slots contain the outputs from the analog to digital converters. Assuming that the ADC sample rate is set to 16 kHz then the AC-link controller will receive 16 bits of data every third codec frame.

The AACI has 4 channels that can be operated simultaneously. They are accessed through DMA.

Figure 15 shows the system timing. Each frame is divided into three sections:

- Token update (MHD wash)
- First pass - inter phone token passing
- Second pass – garbage collection, max model pruning

The horizontal direction of the figure indicates elapsing time, and various subcircuits and processes are indicated along the vertical direction of the figure. The top part of the diagram shows frame count, and push to dictate button, and the lower part of the diagram shows various subcircuits and processes, within a single time frame, whilst the push to dictate button is engaged. These subcircuits and processes are language model interface, word engine, phone book, PINE array, MHD engine, mean & covar reg, and feature vector reg.

At the start of the time frame, the feature vector is generated, and the MHD values are calculated. The MHD values are then passed to the PINE array, which engages in beam

DSMDB-#3348563-v1

pruning and token score updating. The phone book receives next phone requests and returns next phone responses. The word engine adds word link records, and the language model interface supplies language model factors.

In the second part of the frame, pruning thresholds are calculated for the PINE array, for the next frame, and the PINE array engages in token passing. The language model interface transfers language model factors for the next frame.

While the “push to dictate” button remains activated, the above process repeats in a sequence of frames.

The embodiments of the invention which are now described relate to a distance engine for calculation of Mahalanobis Distances.

Figure 16 shows a block diagram of a speech recognition system according to an embodiment of the invention. The system includes a microphone 100, which outputs an audio signal to a combined AGC (automatic gain control) and ADC (analogue to digital converter) 101. In alternative embodiments, the AGC and ADC may be separate. In this embodiment, the incoming data comes from a microphone, and is then converted to digital format. However, it is also possible that the digital audio data may be received from a network, or may be read from storage means such as a memory, an optical disk, or a hard disk drive.

The digital audio data output from the ADC is input to an audio data buffer 102, and may be held here until required. The data is then output from the audio data buffer 102 to a front end circuit 103. The incoming audio data is divided into frames, as described above. In cases where the system is extremely busy, it is possible to drop some audio frames, to allow the system to catch up with any backlog.

In the audio front end circuit 103, the audio is split into spectral components to generate a feature vector, as described above. For example, the feature vector may have a number of frequency bands, and an energy band, plus their first and second derivatives. The

feature vector is then output from the front end 103 to the distance calculation engine 104.

The distance calculation engine 104 converts feature vectors into Mahalanobis distances. The acoustic model may be chosen from a plurality of acoustic models, e.g. according to the language used, and may be stored in a storage means 105 such as a memory, a disk drive, etc. The distances calculated by the distance calculation engine are then transferred to the search stage 106 of the speech processing circuit, which uses models such as one or more word models and/or language models to generate and output recognised text. Such word and language models may be stored in a separate storage means 107.

In one embodiment, the distance calculation engine uses Mahalanobis Distance (MHD) calculations to determine distance values for output to the search stage 106. In further embodiments, different types of distance calculations may be used, e.g. as described above.

In some embodiments of the invention, the distance calculation engine is designed as a speech accelerator, to operate as a loosely bound co-processor for a CPU running speech recognition software. This has the advantage of reducing computational load on the CPU, and reducing memory bandwidth load for the CPU.

In some embodiments, the accelerator interface is memory-mapped. Other types of interface are also possible to support different system configurations, for example removable accelerator cards with SDIO (secure digital input/output), CardBus, or USB interfaces.

The speech recognition apparatus may also operate with a CPU and DSP (digital signal processor), where the DSP implements the "front-end" signal processing to produce a feature vector, and the CPU implements the search stage, reading the distance results from the distance calculation engine.

The system may be implemented in any combination of hardware or software.

DSMDB-#3348563-v1

In some embodiments of the invention the search stage 106 provides the distance calculation engine 104 with an active state list 108 that indicates the subset of states for which the search stage requires distances to be calculated by the distance calculation engine 104. This is an optimization that may reduce computation time and/or power consumption.

In preferred embodiments of the invention, the front end 103, distance calculation engine 104 and the search stage 106 operate in a pipelined manner. When operating in a pipelined manner it is unlikely that the search stage 106 will be able to provide the active state list 108 early enough for the distance calculation engine 104 to implement the optimization of computing only the distances that will be required by the search stage 106. The distance calculation circuit 104 may calculate the MHD values for every state in the lexicon, per frame, whether it is subsequently required by the search stage or not. This allows the accelerator and software system to operate in a concurrent pipelined manner, which maximizes the system throughput.

The distance calculation engine 104 may have a plurality of memory areas for storing feature vectors, to allow a feature vector for one frame to be transferred to one memory area in the distance calculation engine 104, while another feature vector for another frame is being read from a second memory area, for calculation of a distance. The memory areas used for reading and writing may alternate with subsequent frames.

Figure 17 shows a block diagram of a distance calculation accelerator 200, used in an embodiment of the invention. A feature vector is received at input 212 of the accelerator 200. The feature vector is loaded into feature vector (FV) memory A 208 via a bus 211. In reality, there may be more than a single bus, but for simplicity, only one bus 211 is shown in the figure.

The registers 203 are discussed further at the end of this description. The registers 203 may include, for example, an accelerator status register, for indicating whether the acoustic model data is correctly stored in memory, as described in more detail below.

The FV registers 209 hold the feature vector whose distances are currently being computed by the distance calculation engine 204. If the distance calculation engine 204 is implemented in software, the FV registers might be implemented as memory attached to the processor running the software. If the distance calculation engine 204 is implemented in hardware the FV registers might be implemented as logic or as a memory block. The FV registers 209 are loaded from the FV memory 208 when the Distance Calculation Engine 204 has completed distance calculations for the current feature vector and a subsequent feature vector is available in the FV memory 208. In some embodiments there is an optional FIFO associated with FV registers 209 so that the FIFO is loaded from the FV memory 208 when there is a subsequent feature vector in the FV memory 208 and there is a space in the FIFO. In these embodiments the FV Registers 209 are loaded from the associated optional FIFO 209 when the Distance Calculation Engine 204 has completed calculations for the current feature vector and there is a subsequent feature vector in the associated optional FIFO 209. The presence of the FIFO 209 provides for a looser coupling between the Front End 103 and the accelerator 200 which improves performance in some system configurations.

Each feature vector is transferred to a distance calculation engine circuit 204, to obtain distances for each state of the acoustic model. The acoustic model is stored in an acoustic model memory 205, which is connected to the bus 211, for control of reading/writing to the memory. In some embodiments, this also allows acoustic models to be changed or updated via the bus. The acoustic model may be stored in compressed form in the acoustic model memory 205. A decompressor 206 is provided on the data output of the acoustic model memory. The decompressor 206 can decompress the results as they are needed, and thus allows a much smaller size of acoustic memory to be used, than with uncompressed data. From the decompressor 206, the decompressed acoustic model states are sent to the distance calculation engine 204.

The distance calculation engine 204 converts feature vectors and acoustic model states to distances. The accelerator 200 includes two separate results memories for storing these calculated distances, labelled as “result memory A” 201 and “result memory B” 202.

The control, sequencing, and address generation 210 is logic, state machines, and/or software programs that provide overall control of the other elements of the accelerator, taking inputs from the registers 203, and all the other elements and interfaces to determine when to start and stop the Distance Calculation Engine 204. The control 210 also produces addresses and control signals to control the reading and writing of the memories 201, 202, 205 and 208, to control the flow of data between the various elements of the accelerator 200, the operation of the Decompressor 206, CRC check 207 and Distance Calculation Engine 204, and to control the setting or clearing or status bits in the Registers 203 and to control signalling to the front end 103 and search stage 106. The control 210 is implemented using any of the widely known and used methods of controlling and sequencing in digital systems including logic gates, state machines, and embedded software systems. The control 210 might be a centralized control, or it might be distributed in some or all of the other elements of the accelerator 200, or some combination of these.

Each time a feature vector is loaded into the accelerator, the accelerator computes the distances for all states for that feature vector, and stores the results alternately in the A or B Results Memory. The results are written to one of the two memories 201, 202, while the other memory is available for reading. Thus, while the distance calculation engine 204 is storing the distances calculated for one audio frame, the other result memory may make available the distances calculated for an earlier audio frame, to the search stage of the speech recognition system. For non-real time processing, “earlier” audio frame may refer to an audio frame that has been processed earlier, rather than being limited to an audio frame that corresponds to speech at an earlier time.

Thus, the search stage software reads results alternatively from the A or B result memory, while the accelerator writes results into the other result memory, allowing concurrent pipelined operation of the accelerator and the CPU [and the DSP if present].

In some implementations the A and B results memories may be exposed to the software interface as separate memories appearing at different locations in the CPU and/or DSP memory map. Another implementation is to map each memory alternatively to the same range of addresses in the CPU and/or DSP memory map, so that the software always

DSMDB-#3348563-v1

accesses results at the same location, and the memory containing the set of results the software currently needs to access is the one mapped there.

The accelerator may autonomously load and decompress an acoustic model, which is stored in a compressed format in the acoustic model memory 205. The Acoustic Model may be loaded into the Acoustic Model Memory by software running on the CPU prior to the first use of the Accelerator. The Acoustic Model Memory may be an on-chip memory, to maximize bandwidth and minimize power consumption of reading in the entire Acoustic Model every frame, which means 50 to 100 times per second, depending on the system design.

The Acoustic Model Memory may be a dedicated memory attached to, or incorporated within, the accelerator. Alternatively the Acoustic Model Memory may be a shared resource that is used by other system functions or applications when speech recognition is not running. To allow detection of situations where the acoustic model has been overwritten by another application, a pre-calculated CRC (cyclic redundancy check) signature may be stored with the acoustic model. Each time the acoustic model is read into the accelerator, a CRC may be computed and compared with the signature. If the comparison fails, an error bit is set in the accelerator status register.

Some embodiments of the invention relate to speech recognition systems comprising separate front end 103 and search stage 106 processors. These separate processors perform the front-end and search stages of the recognition process, with the accelerator 200 performing the distance calculations. Typically the front-end processing would be performed on a DSP and search stage would be performed on a CPU, although this description holds for any type of processors, even dedicated hardware implementations. As shown in figure 17, the front end 103 supplies the distance accelerator with feature vectors, and the search stage 106 uses distances calculated by the accelerator.

The accelerator is designed to operate in a concurrent pipelined manner with the other processors, allowing a high level of parallelism in the system.

Figure 18 shows how this pipelined operation works in an embodiment of the invention with ideal data flow, i.e. where each of the three recognition stages require exactly the same time to process each data frame, and the processors are never delayed, stalled, or diverted to other tasks.

The time axis is divided into equal “pipeline step times”. If the step time is the same as a frame time then the system is processing audio in real time. If the step time is short than a frame time, the system is operating faster than real time, which may be valuable for batch processing of audio data, or catching up after some type of delay. If the step time is longer than a frame time, the system is operating slower than real time, which may be acceptable in some applications, for example where the person dictating pauses occasionally allowing the system to “catch up”.

Once the pipeline has filled (which takes two step times), the DSP, the Accelerator, and the CPU are all operating in parallel. In this idealized example, the total time available for processing a frame is three step times – one step time in the DSP, one step time in the accelerator, and one step time in the search stage. There is a latency of three step times from the time the audio data enters the front end to the completion of the search stage for that frame, and the processing of one frame finishes each step time. If the system were not pipelined, then the processing of all stages for one frame would have to complete in one step time in order to achieve the same overall throughput of frames per second. Thus pipelining in this way can achieve three times the throughput with the same processing elements, compared to a non-pipelined system. A non-pipelined system would be one in which the processing of frame $n+1$ would not start until all stages of the processing of frame n was already complete.

If, for example, in a system running in real time with a frame time and step time of $1/100^{\text{th}}$ of a second, which is 10ms, each of the three processing elements has the entire 10ms available to complete the processing of one stage of the recognition task, giving 30ms of processing time overall for each frame. If the system were not pipelined, real time operation would require that all three processing elements would be complete in 10ms, allowing only 3.3ms for each of the three processing elements.

DSMDB-#3348563-v1

At the end of each pipeline step time, the DSP provides a feature vector to the Accelerator, and the accelerator provides a set of distance results to the CPU. When the CPU is performing the search stage for frame n , the Accelerator is computing the distances for frame $n+1$, and the DSP is performing the front-end stage for frame $n+2$. During each pipeline step, the front end processing consumes one frame of digital audio data (although this is not indicated explicitly on the diagram).

Real world systems are more complex than the idealized system in many ways. The processing time for a frame is often highly variable and data dependent, especially in the search stage, and the processing time for each processing stage is likely to be quite different. The DSP and CPU are likely from time to time to be diverted to run other software tasks, either to service interrupts, or under control of an operating system scheduling regime. Communications between the processing elements may be stalled due to competing bus traffic, for example for DMA transfers.

Recognition software may implement “frame dropping” and other advanced techniques, and as a result not all frames are processed through all of the recognition stages. Processing elements cannot respond instantly to the availability of data from the preceding processing stage, resulting in delays.

Finite buffer space for information storage between stages means that an earlier stage may be stalled waiting for a space to store its output data when following processing stages are “falling behind”.

The accelerator has been carefully designed to mitigate these problems. Figure 19 shows a pipelined system with real world complications added. The following text describes the interactions shown in the diagram. The time for the processing of various stages has been varied to show difference examples of buffering of data between stages and stalling of processing elements. The numbering corresponds to the numbered bubbles on the diagram. The text also makes reference to memories used to pass information into and out of the accelerator, which are shown in figure 17.

The front end has computed the feature vector for the first frame, FV1. It is loaded into the accelerator, causing the accelerator to commence computation of the distances for the frame.

The feature vector for the second frame, FV2, is passed to the accelerator. The accelerator is still busy computing the distances for the first frame, so this feature vector is held in the accelerators FV memory.

The front end has computed the feature vector for the third frame, FV3, but it cannot be delivered to the accelerator as the FV memory is still full, holding FV2. The time period labelled “stall delivering FV3” is the period between FV3 being computed and when it can first be delivered. Of course the DSP could be diverted to some other useful task, or it could buffer FV3 in its own memory and commence computation of FV4.

The arrow labelled “FV3 earliest” shows the earliest time that the feature vector FV3 can be delivered to the accelerator, which is immediately after the accelerator loads FV2 into its internal registers and frees the memory.

The arrow labelled “FV3 latest” shows the latest time that the feature vector FV3 can be delivered to the accelerator without causing the accelerator to stall for lack of a feature vector to work on. This example shows that there is a wide tolerance in the system to when FV3 can be delivered without affecting system performance – an entire accelerator frame processing time. The accelerator provides an interrupt and a status bit that can be polled to indicate the state of the FV memory. Regardless of which control method the DSP uses, the system tolerates long latency without reducing performance.

The accelerator signals to the search stage that the distances for frame 1 are available in result memory A. The search stage commences processing for frame 1, reading distances from result memory A.

The search stage has finished its processing for frame 1 before the accelerator has completed computing the distances for frame 2, so the search stage stalls waiting for the accelerator. Of course the CPU could be diverted to other useful tasks, including any

DSMDB-#3348563-v1

search stage tasks that can be completed before distances are available. Again, an interrupt and status bits are provided by the accelerator to control this interaction between the accelerator and the CPU. .

The accelerator completes the distance calculation for frame 2, and signals to the CPU that it can commence reading the distances from result memory B.

The accelerator has completed the computation of distances for frame 3 before the search stage has finished processing frame 2. As soon as the search stage signals to the accelerator that it has completed frame 2 by releasing memory B, the accelerator signals that the distances for frame 3 are available, allowing the search stage to immediately move on to processing stage 3.

The accelerator is stalled waiting for feature vector FV4, due to the long processing time in the front end. In real systems this may be caused by a number of factors, including the DSP being diverted to another task for a period of time, or simply due to the DSP performance being insufficient to keep up with the accelerator's throughput.

The accelerator restarts as soon as FV4 is delivered by the front end.

The search stage is stalled waiting for distances for frame 4, a follow on consequence of the late delivery of FV4 to the accelerator.

FV5 is delivered to the accelerator while it is still computing the distances for the previous frame, allowing it to move onto processing frame 5 without stalling when it completes frame 4.

The accelerator signals that the distances for frame 4 are available and the search stage commences processing frame 4.

FV6 is delivered to the accelerator while it is still computing the distances for frame 5.

The availability of distances for frame 5 is signalled by the accelerator to the CPU.

DSMDB-#3348563-v1

Even though the accelerator has already received FV6, it is stalled waiting for a free result memory it can write the distances into.

The search stage releases result memory B at the end of its processing of frame 4. This allows the accelerator to start computing distances for frame 6.

The accelerator completes computing distances for frame 6 and signals to the CPU that the results are ready.

Further embodiments of the invention relate to systems with a Front End and Search Stage on a single processor. This single processor performs the front-end and search stages of the recognition process, with the accelerator performing the distance calculations. This configuration is likely to be the most common configuration where the accelerator is added to an embedded system such as a PDA or mobile phone that has a high-performance embedded application processor (but no DSP) available to run speech recognition software.

In this configuration the accelerator still operates in a concurrent pipelined manner with the front-end and search stages; it is just that the front-end and search stages are running on a single processor.

CPUs in embedded systems are generally relatively simple compared to desktop PCs and servers. Embedded CPUs in mobile phones, PDAs, in-car navigation and telematics systems, and other consumer electronics devices are typically “simple uni-processors” by which we mean not multi-processors and not supporting hardware multi-threading support – i.e. they can only execute a single thread of instructions (although they can be diverted to a different thread under the control of interrupts, exceptions, or an operating system). For this discussion, except where explicitly stated otherwise, we are talking about embedded processors that are “simple uni-processors”. We are also using the term “CPU” and “processor” interchangeably.

Figure 20 shows how the system can be pipelined with a single embedded processor working with the accelerator, in an idealized system. In this idealized system, the time taken for the processor to perform the front-end and search stage processing is identical to the time the accelerator takes to perform the distance calculations, and the processor is never delayed, stalled, or diverted to other tasks.

The time axis is divided into equal “pipeline step” times. If the step time is the same as a frame time then the system is processing audio in real time. If the step time is short than a frame time, the system is operating faster than real time, which may be valuable for batch processing of audio data, or catching up after some type of delay. If the step time is longer than a frame time, the system is operating slower than real time, which may be acceptable in some applications, for example where the person dictating pauses occasionally allowing the system to “catch up”.

The first two pipeline step times are used to fill the pipeline. By the third pipeline step time, the pipeline is full, and all three stages of the recognition process are processed every pipeline step time, with each stage working on the processing of a different data frame.

During the first pipeline step time, the CPU completes the front-end processing for frame 1, and the feature vector passed to the accelerator so that distance calculations can start in the accelerator.

During the second pipeline step time, the front-end processing for frame 2 is completed by the CPU, in parallel with the accelerator computing the distances for frame 1.

During the third pipeline step time, the CPU runs the search stage for frame 1 reading distance results from the accelerator, and then runs the front-end processing for frame 3. The accelerator runs in parallel computing the distances for frame 2 using the feature vectors computed in the previous pipeline step.

Each subsequent pipeline step time is similar to the third pipeline step, with the frame numbers incremented. So for pipeline step $n = 4, 5, 6$, etc. the CPU runs the search stage

DSMDB-#3348563-v1

for frame $n-2$ and then runs the front-end for frame n . In parallel the accelerator computes the distances for frame $n-1$.

Note that with this situation the production of feature vectors and the consumptions of results run in lock step - one feature vector is produced every time the search stage consumes one frame of distance results. This considerably reduces the need for elastic buffering between the distance accelerator and the software system.

Again, real world systems are more complex than the idealized system in many ways. For example, the processing time for a frame is often highly variable and data dependent, especially in the search stage, and the processing time for each processing stage is likely to be quite different. The CPU is likely from time to time to be diverted to run other software tasks, either to service interrupts, or under control of an operating system scheduling régime. Communications between the CPU and accelerator may be stalled due to competing bus traffic, for example for DMA transfers.

Recognition software may implement “frame dropping” and other advanced techniques, and as a result not all frames are processed through all of the recognition stages. Processing elements cannot respond instantly to the availability of data from the preceding processing stage, resulting in delays.

Finite buffer space for information storage between stages means that an earlier stage may be stalled waiting for a space to store its output data when following processing stages are “falling behind”.

The accelerator has been carefully designed to mitigate these problems. Figure 21 shows a pipelined system with real world complications added. The following text describes the interactions shown in the diagram. The time for the processing of various stages has been varied to show difference examples of buffering of data between stages and stalling of processing elements. The numbering corresponds to the numbered bubbles on the diagram. The text also makes reference to memories used to pass information into and out of the accelerator, which are shown in figure 21.

DSMDB-#3348563-v1

At the end of front-end processing for frame 1, the CPU passes the feature vectors for frame 1 to the accelerator, allowing distance calculations to start.

To fill the pipeline, the CPU immediately performs the front-end processing for frame 2 and passes the feature vectors to the accelerator. They are stored in the accelerator so that the accelerator can load them into its internal engine for processing as soon as the distance calculations for frame 1 are complete.

The CPU is now ready to start the search stage for frame 1, however the distance computation in the accelerator has not completed. The box labeled “Prep search stage 1” indicates a period where the CPU could do any preparatory search stage processing that does not require distances. Alternatively the CPU could be diverted to another task, or it can stall waiting for the distance results.

The accelerator signals that the distance results for frame 1 are now available. The search stage can commence on the CPU, reading distance results from the accelerator’s Result Memory A. The accelerator automatically starts processing the feature vector for frame 2 that was delivered earlier.

The accelerator signals to the CPU that it has completed the distance calculations for frame 2. The CPU is still busy running the search stage for frame 1, and has still to run the front end for frame 3, so the completion signal is ignored until the CPU has need of it, when it starts running the search stage for frame 2. As the accelerator has not received the feature vector for frame 3, it has no task it can perform and so it stalls.

The CPU delivers the feature vector for frame 3, restarting the accelerator, which writes results into Result Memory A. Note that Result Memory A was previously “released” at the end of the processing of search stage 1, informing the accelerator that it can overwrite the old results held in that memory with new results. The CPU then runs the search stage for frame 2, reading distance results from the accelerator’s Result Memory B.

The accelerator signals to the CPU that the distance results for frame 3 are complete in Result Memory A. Again, the CPU ignores this signal until it is ready to commence the search stage for frame 3, and the accelerator stalls waiting for the next feature vector.

The CPU delivers the feature vector for frame 4, restarting the accelerator. It then starts search stage 3 reading distances from the accelerator's result memory A.

Search stage 3 and front end 5 were completed quickly, so that the feature vector for frame 5 is delivered to the accelerator while it is still running. The CPU performs all the preparatory work it can for the search stage, and then stalls waiting for distances to be available.

The accelerator signals that the distances for frame 4 are available. The CPU starts search stage 4, reading the distances from the accelerator's result memory B. The accelerator automatically starts computing distances for frame 5 using the feature vector delivered earlier.

Note that with the pipelining scheme just presented which alternates front-end and search stage processing, the accelerator can never stall waiting for a result memory to be released because the feature vector the accelerator needs as an input is not produced until after the memory has been cleared. The CPU task schedule could be described with the shorthand F1 F2 S1 F3 S2 F4 S3 F5 S4.

It can be seen readily from figure 21 and the accompanying description above, that with this CPU task schedule, that the operation of the CPU and accelerator are tightly coupled and whenever the processing time for a pipeline step on the CPU is unequal to the processing time on the accelerator, then it quickly results in one of them stalling.

In a system where either the CPU or the accelerator is always (or at least in most instances) slower than the other in completing its processing task, stalling is inevitable and throughput may not be improved by adding additional buffering (e.g. additional result memories or FIFO depth on the feature vector interface). [Unless there are other difficulties to mitigate such as very low-bandwidth buses linking the CPU to the

DSMDB-#3348563-v1

accelerator, or competing bus traffic such as DMA transfers that can significantly reduce performance.

If the processing time for a pipeline step is highly variable on either the CPU or Accelerator, such that there are regular changes in which one of them takes the longer time, then significant enhancements in throughput may be obtained by adding additional buffering between them. This may have the disadvantage of adding additional latency through the system. This buffering may be implemented by adding additional depth to the Feature Vector FIFO, and/or adding additional Result Memories. Alternatively it may be by implementing elastic buffers in software.

If the accelerator has sufficient buffering on the feature vector interface, e.g. by including the optional FIFO shown in figure 21, then increasing the pipeline depth by computing extra front end frames up front will improve throughput at a cost of extra latency. This will result in a CPU task schedule of F1 F2 F3 S1 F4 S2 F5 S3 F6 S4 or F1 F2 F3 F4 S1 F5 S2 F6 S3 F7 S4, etc.

Of course it would also be possible to dynamically schedule whether the CPU should run the front-end or search stage code based on the availability, or not, of distance results and/or the space available to store more feature vectors and distance results. Given the fundamental relationship that one set of distance results is produced by the accelerator for every feature vector it consumes, and the restriction stated at the start of this section that we are discussing single-threaded CPUs, dynamic scheduling is likely to add little performance improvement while being complex to implement.

On a CPU supporting multi-threaded operation (which is a technique for improving performance in the presence of high memory system latency), or on a multi-processor, the highest system performance is likely to be achieved by running the front-end and search stages concurrently as separate threads, rather than alternating between the front-end and search stages. The system operation is then very similar to that described above, if each thread is thought of as a separate virtual processor. In some embodiments of the invention, elastic buffers may be used between stages, to accommodate varying time delays between the processing of frames at each stage. Elastic buffers in the interfaces

DSMDB-#3348563-v1

between the three recognition stages may also significantly enhance performance of such a system.

Operation of the accelerator and the software system is not constrained to a rigid repetitive frame time. Instead the timing of the system is controlled dynamically by the availability of feature vectors into the accelerator, and the consumption of results out of the accelerator.

This allows the recognition process to fall behind real-time operation and to "catch up" again. It also allows the system to be robust in complex systems where interrupts, DMA bus traffic, memory caches, operating system task switching, and competing real-time software tasks all make guaranteeing rigid timing constraints impractical.

Elastic buffers at the major data transfer points between processing elements maximize system performance by allowing one element to continue useful processing whenever it can, regardless of whether other processing elements have been stalled or diverted to other tasks. Elastic buffers are useful for the following applications:

(1) For buffering raw audio data arriving in real time from the microphone(s) The data would typically be moved into the buffer by a DMA and/or high priority interrupt driven software process to ensure that hard real time constraints are met.

(2) For buffering feature vectors output from the front end. This is especially true if the front-end is implemented on a DSP or other processing element independent of the search stage.

(3) For buffering distance results from the accelerator. Again this is especially true if the front-end and search stages are implemented on independent processing elements. This buffering is also especially valuable if the interface between the search stage processor and the accelerator is very slow, has high latency, or is subject to competing bus traffic.

The Feature Vector and Results interfaces on the accelerator already provide a level of elasticity that should be sufficient in systems where the Front End and Search stages are running on the same CPU, and the accelerator has a high-bandwidth interface to the CPU with low level of competition from other bus traffic such as DMA.

Other embodiments may have deeper FIFOs. In one embodiment, the two result memories are an implementation of a FIFO where the implementation is exposed across the interface. It would be possible to have additional memories and use them in a round-robin fashion, or to use another implementation of a FIFO that hides the FIFO depth by only exposing one result memory at a time to the interface. An example of this would be to map each memory in turn to a memory range that is always used to read results, so that at any one time the “current” result memory is mapped to that location. This would hide the FIFO depth behind the interface so that the pipeline depth can be changed without changing the software (or hardware) that accesses the interface.

Slow or Conflicted Bus Interfaces may occur. In the system configurations described above, the search stage reads distances directly from the Accelerator’s Result Memories. The result memories allow the search stage to randomly access the distance results, as they are needed. This will only give a high performance system if the interface or bus between the CPU and Accelerator provides the CPU with low latency, high bandwidth access to the Accelerator Result Memories.

In some systems it is not practical to provide such a low latency, high bandwidth interface between the CPU and the Accelerator. This may be for many reasons, including:

- The interface is a bus shared with other devices and/or processors that generate bus traffic that keeps the bus busy such that the CPU access to the accelerator is delayed waiting for the bus to be free. DMA transfers are one example of conflicting bus traffic.
- The interface is implemented as a serial bus, or narrow parallel bus, for cost or compatibility reasons, yielding low bandwidth and high latency
- The interface crosses an industry standard interface such as PCI, USB, SDIO, Cardbus, Firewire, PCIeExpress etc. that introduces protocol overheads and/or

DSMDB-#3348563-v1

latency as would be the case where the accelerator is in a plug-in peripheral to a system containing a CPU (or CPU and DSP).

In these situations it is possible to add another pipeline stage during which the result data is copied from the Accelerator into another memory (which we will call a “buffer memory”) to which the CPU does have high bandwidth, low latency access. This copying can be performed by a DMA controller, or by software running as an interrupt handler, or a separate software thread on the CPU, or a software thread on another processor, or by other commonly known means. The device controlling the transfer may be on the CPU side of the interface or on the accelerator side of the interface, depending on the capabilities of the interface. This extra pipeline stage then allows an entire pipeline step time to copy the data from the accelerator to the other memory. The data copy can optionally be performed as a sequential burst, or a number of sequential bursts, to optimize transfers across the interface between the Accelerator and the buffer memory. The search stage, running on the CPU, can then randomly access the results in the buffer memory with low latency.

Similarly an extra pipeline stage can be added for transferring the Feature Vectors to the Accelerator, although given that the feature vectors are a much smaller object than the distance results, and that they are not accessed randomly across the interface but delivered once to the Accelerator for each frame, there are fewer situations where the performance benefit from this extra pipeline stage is as significant.

Figures 22 and 23 show idealized systems with the two extra pipeline stages for transfers added. The first shows a system with a CPU, DSP, and Accelerator with the two extra pipeline stages. The second shows a system with a CPU and DSP with the two pipeline stages added.

Other embodiments may add additional pipeline stages so that the distance computations are performed over two or more pipeline stages rather than being restricted to a single pipeline stage. This allows more time for the computation of the distance results for each frame, while maintaining the same throughput of distance

results. This is similar in concept to the well-known pipelining techniques used to implement RISC microprocessors.

The Acoustic Models may occupy many megabytes of storage space. In embedded systems they will typically be held in a non-volatile memory such as FLASH memory, or other types of non-volatile memory as is the appropriate currently economic technology. The Accelerator may read the acoustic model directly from the non-volatile memory, or more likely the acoustic model will be copied from the non-volatile memory into a higher bandwidth RAM memory of some type (e.g. SRAM or DRAM, SDRAM, etc.). The RAM memory may be part of the same integrated circuit as the Accelerator (i.e. “on chip”), or attached to the integrated circuit containing the Accelerator. (“off chip”). The having the RAM memory on-chip is particularly advantageous, as this configuration would give the lowest power consumption and highest bandwidth. An alternative solution would be for the acoustic model to be stored on a disk drive, or on another device accessible via a wired or wireless bus or network, and for the acoustic model to be loaded into the RAM memory from the disk drive, or over the wired or wireless bus or network.

On-chip RAM memory is however likely to come at a cost-premium to off-chip commodity memory devices. Many embedded systems face extreme cost pressures in the market place, especially high volume devices supplied by many manufacturers such as mobile phones. For a speech recognizer to be incorporated into such systems, the overall recognizer solution must be delivered at very low cost. There is therefore a significant economic benefit from reducing the cost of the RAM memory and the non-volatile memory. One way to reduce the cost of these memories is to compress the acoustic models and store the compressed model in the memory.

The Accelerator computes all of the distances associated with each Feature Vector that is delivered to it by the front end. This means that the accelerator must use the data from the entire Acoustic Model each time a new Feature Vector is presented to the Accelerator (with the exception of some specific optimizations that may determine that some data values need not be used in the distance computations). It is therefore possible

DSMDB-#3348563-v1

to arrange for the accelerator to read through the acoustic data sequentially once for each Feature Vector, using each data value as it read from the Acoustic Model.

Reading through the entire Acoustic Model sequentially is ideally suited to placing a decompressor between the distance calculation engine and the RAM memory, because many compression schemes depend on the decompressor reading sequentially through the compressed data, and do not readily support random access into the compressed data.

Thus this configuration allows storing compressed data in the RAM memory (and the non-volatile memory), which can therefore be smaller and lower cost. There is no requirement for a memory that is large enough to hold the entire decompressed acoustic model, as the decompressed data is consumed as it is decompressed.

The decompression scheme could be any combination of:

- sign or zero extension or otherwise conversion of narrow or variable width data to a wider data format.
- sign or zero extension or otherwise conversion of narrow or variable width data to IEEE standard single or double precision floating point format
- codebook decompression of a binary bitstream, where the codebook is stored as part of the acoustic model data
- decompression of a Huffman or Lempel-Ziv compressed stream
- decompression of run length encoded data
- decompression of difference encoded data
- any well known compression scheme.

Additionally the model size can be reduced by using *subspace distribution clustering* and the decompressor and distance calculation engine designed to operate with such models.

The following description provides further information on the visible registers in some embodiments of the invention.

The register bank is designed to allow simple connection with an 8-bit data bus. Accesses to the register bank should be made with byte-read and byte-write accesses. To facilitate this, only the least significant 8 bits of the data bus are used.

The address decoder should respond to byte accesses to byte 0 or byte 3 within the designated word. This way the same code can be used on an ARM processor regardless of whether the processor is configured for little endian or big endian operation.

REGBASE=Base address of register bank

Register 0 Product Version

8-bit Product version ID

This register value is changed when the functionality of the hardware changes in a way that changes the hardware/software interface.

Address = REGBASE (and REGBASE+3 for big endian)

Read only. Reads with value 0x01

Writes ignored

Register 1 Implementation Version

8-bit Implementation version ID

This register value is changed for each implementation to facilitate bug tracking and version control.

Address = REGBASE+4 (and REGBASE+4+3 for big endian)

Read only. Reads with value 0x01

Writes ignored

Register 2 Interrupt Enable

8-bit Interrupt Enable register

Address = REGBASE+4*2 (and REGBASE+4*2+3 for big endian)

Readable and writeable.

Value following reset is 0 for bits 0,1,2,3

DSMDB-#3348563-v1

Bit 0 I 1 = Enable FV memory empty interrupt on Int0 output.
 Bit 1 J 1 = Enable Results complete interrupt on Int0 output.
 Bit 2 K 1 = Enable FV memory empty interrupt on Int1 output.
 Bit 3 L 1 = Enable Results complete interrupt on Int1 output.
 Bit 4 Future use. Read value is not defined. Writes ignored.
 Bit 5 Future use. Read value is not defined. Writes ignored.
 Bit 6 Future use. Read value is not defined. Writes ignored.
 Bit 7 Future use. Read value is not defined. Writes ignored.

Software should use a read-modify-write code sequence when setting this register so that future use bits are written with the value read from them.

Register 3 Status

Address = REGBASE+4*3 (and REGBASE+4*3+3 for big endian)

Read-only register. Writes are ignored.

Value following reset is 0 for bits 0,1,2,3

Bit 0 FVMB Feature Vector Memory is Busy

0 = OK to write new value to FV memory (FV memory is empty)

1 = Don't write new value to FV memory (FV memory is busy)

Bit 1 CRM Current Result Memory

0 = Result memory A

1 = Result memory B

This bit toggles every time the Release Result Memory (RRM) command is issued to the accelerator (register 3). It indicates which memory software should read results from, and which result memory the RRM command should release next time it is issued. Note: Software should also check that results are available in the indicated result memory by checking the value of the MA and MB bits.

Bit 2 MA 1=Results are available is result memory A

The MA bit is set to 1 when the compute engine completes writing all of the distances for a frame into result memory A. This value is set to zero when the Release Result Memory command is issued and CRM==0.

Bit 3 MB 1= Results are available in result memory B

The MB bit is set to 1 when the compute engine completes writing all of the distances for a frame into result memory B. This value is set to zero when the Release Result Memory command is issued and CRM==1.

Bit 4 Dfault 1=Acoustic data CRC fault. Indicates that the CRC value at the end of the acoustic data did not match the CRC value computed while the acoustic data was read in.

Bit 5 Future use. Value undefined.

Bit 6 Future use. Value undefined.

Bit 7 Future use. Value undefined.

Register 4 Command

Address = REGBASE+4*4 (and REGBASE+4*4+3 for big endian)

Write-only register.

Software should perform byte writes to this register with a single bit set to 1 and all other bits set to zero to issue a command to the accelerator.

Behaviour is not defined if more than one command is issued within a single write (i.e. if more than 1 bit is a 1).

Value on read is not defined.

Bit 0 LfV Load Feature Vector from the FV memory.

Tells the accelerator that a new feature vector has been placed into FV memory ready for use by the accelerator. The FV will be loaded into the compute engine when the compute engine completes its current task (or immediately if the compute engine is idle). Computation of distances for this FV will commence when the FV is loaded AND there is a released result memory. (At reset, both result memories are released). The results for the first, third, fifth etc. feature vectors loaded are written to Result Memory

A. The results for the second, fourth, sixth etc. feature vectors loaded are written to Result Memory B.

Bit 1 RRM Release Result memory. Indicates that the results in the current result memory can now be overwritten with new results. The current result memory is indicated by the CRM bit in the status register.

Bit 2 CFVI Clear the FV memory empty interrupt

Bit 3 CRI Clear the Results Complete interrupt

Bit 4 Future use – write 0

Bit 5 Future use – write 0

Bit 6 Future use – write 0

Bit 7 Future use – write 0

The present invention may be included in a mobile phone, or in another consumer electronics device, a PC, Laptop or tablet PC, in-car electronics system, or plug-in card for an electronic device.

The methods and apparatus according to the invention are not limited to embedded mobile systems, but may be used with any type of computing device, including PCs, workstations, laptop computers, notebook computers, handheld computing devices, dictation machines, PDAs (personal digital assistants), personal organisers, mobile phones, games consoles, video players, etc. Embedded computers according to the present invention may be embedded in a wide range of apparatus and machinery, such as electronic entertainment products, domestic appliances, factory machinery, vehicles, etc.

Embodiments of the present invention may be implemented as an IP core in a System on Chip (SOC) or a discrete device. Multiple chips or bare die may be connected to provide a larger vocabulary system.

Although the apparatus may include all or some of an input for accepting audio input, a circuit for digitising the audio input, a circuit for dividing the audio input into a plurality of time-frames, and a circuit for analysing the spectral components of each time frame

DSMDB-#3348563-v1

to characterise the audio in that frame, this is not essential. The apparatus may also include a circuit for calculating a score indicating the probability or likelihood that each audio frame corresponds to each of a plurality of different possible states, but again, this is not essential. Instead, the apparatus may include an input for receiving the pre-calculated scores.

In the described embodiments, n-gram language models are used. For example, the Bigram language model is based on outputting the probability that a given second word directly follows a given first word. However, it is possible to use alternative language models instead. It is possible to use a mixture of different models. For example, it is possible to run a word-dependent n-best decode applying a bigram language model, and the resulting word link record or word graph could then be rescored using a trigram language model.

The Viterbi beam search algorithm may be used in embodiments of the invention, to perform the search. A word level search may be performed. Dummy states may be used for the start and end of an utterance

The grammar level model may be used to add distance elements into the likelihood calculation. This is not essential, but tends to give much better results.

The HMM models described in the embodiments comprise self connections to each states and single connections to a neighbouring state. However, the present invention is not limited to such HMM models, and for example, HMM models with additional connections between states may also be used. Alternatively, embodiments of the invention may use alternative types of acoustic models, rather than HMM models.

The system may comprise a plurality of language models (e.g. English, French). A model of words may comprise actual words, phrases, silence, lipsmack sounds, and/or other sounds for recognition by the apparatus.

As previously discussed, the phone level has been found to be the best level for lexical tree searching, rather than the state level or the word level. However, this invention is not limited solely to phone level searching.

Pipelining may comprises processing data in different parts of a circuit at the same time, i.e. parallel processing.

In the following claims, the calculation means may comprise the distance calculation engine described above.

Further embodiments of the invention include the methods carried out in the accelerator, and the methods carried out in the speech recognition circuit.

While the invention has been described in terms of what are at present its preferred embodiments, it will be apparent to those skilled in the art that various changes can be made to the preferred embodiments without departing from the scope of the invention, which is defined by the claims.

A SPEECH RECOGNITION CIRCUIT AND METHOD

This is a continuation of application Serial No. 13/735,091, filed January 7, 2013, now Pat. 9,076,441, which is a continuation of application Serial No. 13/162,128, filed June 16, 2011, now Pat. 8,352,262, which is a divisional of application Serial No. 11/662,704, filed March 14, 2007, now Pat. 7,979,277, which is a 371 of PCT/GB2005/003554, filed September 14, 2005, which claims priority to UK Application No. 0420464.0, filed September 14, 2004, the disclosures of which are hereby incorporated by reference in their entireties.

BACKGROUND OF THE INVENTION

1. Field of the Invention:

The present invention relates to speech recognition circuits and methods. These circuits and methods have wide applicability, particularly for devices such as mobile electronic devices.

2. Description of the Related Art:

There is growing consumer demand for embedded speech recognition in mobile electronic devices, such as mobile phones, dictation machines, PDAs (personal digital assistants), mobile games consoles, etc. For example, email and text message dictation, note taking, form filling, and command and control applications are all potential applications of embedded speech recognition.

However, when a medium to large vocabulary is required, effective speech recognition for mobile electronic devices has many difficulties not associated with speech recognition systems in hardware systems such as personal computers or workstations. Firstly, the available power in mobile systems is often supplied by battery, and may be severely limited. Secondly, mobile electronic devices are frequently designed to be as small as practically possible. Thus, the memory and resources of such mobile embedded systems tends to be very limited, due to power and space restrictions. The cost of providing extra memory and resources in a mobile electronic device is typically much higher than that for a less portable device without this space restriction. Thirdly, the

DSMDB-#3348563-v1

mobile hardware may be typically used in a noisier environment than that of a fixed computer, e.g. on public transport, near a busy road, etc. Thus, a more complex speech model and more intensive computation may be required to obtain adequate speech recognition results.

These restrictions have made it difficult to implement effective speech recognition in mobile devices, other than with very limited vocabularies.

Some prior art schemes have been proposed to increase the efficiency of speech recognition systems, in an attempt to make them more suitable for use in mobile technology.

In an article entitled “A low-power accelerator for the SPHINX 3 speech recognition system”, in University of Utah, International conference on Compilers, Architectures and Synthesis for Embedded Systems, Nov 2003, Davis et al have proposed the idea of using a special purpose co-processor for up-front calculation of the computationally expensive Gaussian output probabilities of audio frames corresponding to particular states in the acoustic model.

In an article entitled “Hardware Speech Recognition in Low Cost, Low Power Devices”, University of California, Berkeley, CS252 Class Project, Spring 2003, Sukun Kim et al describe using special purpose processing elements for each of the nodes in the network to be searched. This effectively implies having a single processing element for each phone in the network. An alternative suggested by Sukun Kim et al is to provide a processor for each state in the network.

In an article entitled “Dynamic Programming Search for Continuous Speech Recognition” in IEEE Signal Processing Magazine, Sept 1999, Ney et al discuss language model lookahead. Language model lookahead involves computation of a language model factor for each node (i.e. phone) in the lexical tree. This technique is also known as smearing. Each phone instance in the search network can be given a language model factor when it is used in the lexical tree search. Ney et al show that for an example bigram language model, the average number of states per 10 ms frame can

DSMDB-#3348563-v1

be reduced from around 168,000 states with no language model lookahead to around 8,000 states when language model lookahead is used. They also show that bigram language model lookahead requires about a quarter of the states compared with unigram language model lookahead.

Although these prior art documents provide improvements to speech recognition in embedded mobile technology, further improvement is still needed to provide a larger vocabulary and better accuracy.

SUMMARY OF THE INVENTION

One aspect of the present invention provides a speech recognition circuit including a circuit for providing state identifiers which identify states corresponding to nodes or groups of adjacent nodes in a lexical tree, and for providing scores corresponding to said state identifiers. The lexical tree includes a model of words. The speech recognition circuit also has a memory structure for receiving and storing state identifiers identified by a node identifier identifying nodes or groups of adjacent nodes, the memory structure being adapted to allow lookup to identify particular state identifiers, reading of the scores corresponding to the state identifiers, and writing back of the scores to the memory structure after modification of the scores. An accumulator is provided for receiving score updates corresponding to particular state identifiers from a score update generating circuit which generates the score updates using audio input, for receiving scores from the memory structure, and for modifying said scores by adding said score updates to said scores. A selector circuit is used for selecting at least one node or group of nodes of the lexical tree according to said scores.

One suitable type of hardware for the memory structure includes a content addressable memory (CAM). A CAM is a memory unit which stores a series of data items using a series of addresses. However, the memory is accessed by specifying a data item, such that the CAM returns the corresponding address. This contrasts with a random access memory (RAM) in which the memory is accessed by specifying an address, such that the RAM returns the corresponding data item.

DSMDB-#3348563-v1

However, the memory structure is not limited to including a CAM. Other types of hardware are also possible, to provide this functionality. For example, a single chip which operates in the same way as a CAM and RAM may be used instead.

Embodiments of the present invention provide a solution to the problem of how to map a lexical tree search to a CAM system architecture. The realisation by the present inventors that certain speech recognition data structures can be mapped into the CAMs allows a lexical tree search to be performed using a CAM system architecture.

Further embodiments of the invention include a counter for sequentially generating state identifiers, and using said generated state identifiers to sequentially lookup said states in the memory structure.

The node identifier may comprise a direct reference to the lexical tree. However, in some embodiments, the node identifier for at least some of the states includes a pointer to a node identifier for another state. For example, a state corresponding to the furthest part of the search path in the lexical tree may be referenced by a node identifier which directly links to a particular node or group of nodes in the lexical tree. In a lexical tree comprising phones, using a state model of triphones, the node identifier may indicate the position of a triphone in the lexical tree.

However, in this example, for states occurring further back in the search path, instead of supplying a node identifier linking directly to the lexical tree, instead a pointer to a node identifier of another state may be supplied. E.g. a triphone instance may have a pointer to another triphone instance, which has a pointer to another triphone instance, which has a pointer to a node or group of nodes in the lexical tree. Chains of reference may be set up in this way, where only the last state in the chain has a direct pointer to the lexical tree.

There may not be a one-to-one correspondence between the nodes of the lexical tree and the node identifiers. This will occur for a branched lexical tree, where the nodes represent monophones, but the acoustic model states represent triphones, i.e. groups of three adjacent monophones. Then, paths of three monophones will have unique

DSMDB-#3348563-v1

identifiers to be stored in the memory structure, rather than single monophones having unique identifiers.

Phone instance numbers may be generated, and used to uniquely label each phone instance. They can be generated sequentially, using a counter. The phone instance numbers may be used as pointers between phone instances to assist in node identification. It is thus not essential to provide a direct node identifier for each phone instance to directly indicate a location in the lexical tree. The dynamic network of phone instances provided in the memory structure may thus include both direct and relative references to the lexical tree.

The memory structure may be divided into one part which stores phone instance identifiers and direct references to the lexical tree, and a second part which stores phone instance identifiers and corresponding states. This can speed up the processing, by only storing the phone instances which are furthest on in the lexical tree in the first part of the memory structure.

The memory structure may also be divided into separately accessible units, to reduce the amount of data in each unit, thereby decreasing the chance of finding the same two states identifiers in different phone instances in any single memory unit, and increasing the chance of some state identifiers being completely absent from any single memory unit. This makes it easier to deal with the situation when the same two state identifiers are found, because a spare time slot is available for processing when a state identifier is not present.

A further aspect of the invention provides a distance calculation engine within a speech recognition system. The distance calculation engine may be included within an accelerator. The accelerator may include logic to interface with other parts of a speech recognition circuit, in addition to the distance engine, although this is not essential. For example, the accelerator may include one or more results memories for storing distances calculated by the distance calculation engine. The accelerator may also include at least one of a memory for storing one or more acoustic models, a decompressor for decompressing acoustic data that has been stored in a compressed format, a memory for

DSMDB-#3348563-v1

storing feature vectors, a checksum or data signature calculation means, buffers for data storage, and data registers. The accelerator may be implemented in software or in hardware, or in a combination. It may be physically separate to the rest of the speech recognition circuit, although this is not essential.

The distance calculation engine may calculate one or more of a wide range of distance metrics and probability distributions. The distances may represent the likely correspondance of feature vectors to states in an acoustic model. In other words, the distances can indicate the similarity of an audio data frame to each possible state in an acoustic model

There are a wide variety of probability distributions that can be used for the distance calculation stage of a speech recogniser, and a wide variety of distance metrics used. These are widely documented in the literature. A point is a simple example of a probability distribution.

A common choice is to use Gaussian Distributions and correspondingly the Mahalanobis Distance metric. The Gaussian probability distribution is then defined by a mean vector, which defines centre point in the N-dimensional space, and a Covariance matrix which defines the shape of the probability distribution. It is common to restrict the Covariance matrix to be a diagonal matrix (only N non-zero values along the diagonal of the NxN matrix) which significantly lowers the implementation cost by reducing the number of arithmetic operations.

In particular embodiments, the distance calculated is a Mahalanobis distance. Particular examples of this are described later in the specification.

In one embodiment, the distance engine autonomously computes all of the distances associated with a given feature vector. This may comprise computing distances for every state in the lexicon. The distance engine may operate in a pipelined manner with other stages of the recognition process. In this context a distance is an indication of the probability or likelihood that a feature vector corresponds to a particular state. An

DSMDB-#3348563-v1

important class of distance computation in speech recognition is the calculation of output state probabilities in recognisers using Hidden Markov Models. Another possible use is in recognisers using Neural Networks.

The distance engine reads data from the acoustic models to use as parameters in the calculation. The acoustic models may be optionally stored in a compressed format. The distance engine may read and de-compress the acoustic models one (or more) times for each feature vector processed. Each reading of the acoustic models may require reading the entire acoustic model, or various optimisations may be implemented to avoid reading parts of the acoustic model that are not required to be used for calculations with the current feature vector. The distance engine may use a de-compression method where the de-compression is sign or zero extension or may otherwise convert data of narrow or variable width to a wider data format. The distance engine may use a de-compression method where the de-compression is sign or zero extension or may otherwise convert data of narrow or variable width to IEEE standard single or double precision floating point format. The distance engine may use a decompression method where decompression is a codebook decompression of a binary bitstream, where the codebook is stored as part of the acoustic model data. The distance engine may use a decompression method where the decompression is decompression of a Huffman or Lempel-Ziv compressed stream. The distance engine may use a decompression method where decompression is decompression of run length encoded data. The distance engine may use a decompression method where decompression is decompression of difference encoded data. The distance engine may use a decompression method using any well known lossy or lossless compression scheme. The distance engine may use a decompression method using subspace distribution clustering. The distance engine may use a decompression method comprising any combination of the above described decompression types.

The distance engine may read the acoustic models from a dedicated on-chip memory. The distance engine may read the acoustic models from a dedicated off-chip memory. The distance engine may read the acoustic models from a shared on-chip memory. The distance engine may read the acoustic models from a shared off-chip memory. Any of these acoustic models may be compressed.

The distance engine may compute a CRC or checksum or similar signature as it reads in the acoustic model and compares this to a stored CRC, checksum, or signature, in order to check that the acoustic model has not been corrupted, and signals an error condition if such corruption is detected. The stored CRC, Checksum, or signature may have been pre-computed and stored in the model data, or it may be computed at the time the model data is loaded into the Acoustic Model Memory. It may be held in Acoustic Model Memory, or it may be loaded into a register or another memory from where it can be accessed and compared when the CRC/checksum/signature is computed each time the Acoustic Model is loaded.

The distance engine may support the pass-through of data from the front-end to the search-stage. The data to be passed through will be supplied to the distance engine as an adjunct to the feature vector, and the distance engine will pass it to the search stage as an adjunct to the distance results. This provides a simple mechanism for passing frame-specific data that is not involved in the distance calculation through to the search stage, and keeping it associated with the correct frame, which may otherwise be complex in pipelined systems with multiple processors. The data passed through may be for any purpose. Examples might include silence detected, end-of-audio-stream detected, a frame number, information that some intervening frames have been dropped, or data from another input device such as a button or keyboard in a multi-modal interface.

The distance engine may be implemented in hardware, software, or a combination. Other stages may be implemented in hardware, software, or a combination. The distance engine may be implemented with any number representation format including fixed point or floating point arithmetic, or any mixture of number representation formats.

In particular, the other stages may be implemented on a CPU, or on a DSP and CPU. The "DSP" and "CPU" may each be implemented as software programmable devices.

The distance engine may implement one or more additional pipeline stages to overcome delays introduced by low bandwidth, high latency, or conflicted bus interfaces. The distance engine may also implement additional pipeline stages to maintain the same

DSMDB-#3348563-v1

throughput while allowing more time for each distance calculation. Particular embodiments of the invention may include one or more of the above aspects.

A further aspect of the invention comprises a speech recognition circuit, comprising: an audio front end for calculating a feature vector from an audio signal, wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame; calculating circuit for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words; a buffer memory between the calculating circuit and the search stage, for receiving data passing from the calculating circuit to the search stage, wherein a processor in the search stage has higher bandwidth and/or lower latency access to the buffer compared to the bandwidth and/or latency of direct transfer between the calculating circuit and the search stage. The data transfer from the calculating circuit to the buffer memory and/or from the buffer memory to the search stage may be performed as one or more sequential bursts. The data transfer to the buffer memory may be performed in parallel with data transfer to the calculating circuit and/or in parallel with data transfer to the search stage. A second buffer memory may be provided between the audio front end and the calculating circuit.

A further aspect of the invention comprises a speech recognition circuit, comprising: an audio front end for calculating a feature vector from an audio signal, wherein the feature vector comprises a plurality of extracted and/or derived quantities from said audio signal during a defined audio time frame; calculating circuit for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model; and a search stage for using said calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words; comprising an elastic buffer between at least one of the front end and calculating circuit, or the calculating circuit and search stage, and/or for buffering said audio signal.

A further aspect of the invention comprises an accelerator for a speech recognition circuit, the accelerator comprising: calculating means for calculating a distance

DSMDB-#3348563-v1

indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model, wherein the feature vector comprises a plurality of extracted and/or derived quantities from an audio signal during a defined audio time frame; means for comparing a first version of a stored checksum of data representing said acoustic model and a second version of said stored checksum, wherein the second version is obtained from an updated measurement and calculation of the checksum; and means for indicating an error status if the checksums do not match.

A further aspect of the invention comprises an accelerator for a speech recognition circuit, the accelerator comprising: calculating means for calculating a distance indicating the similarity between a feature vector and a predetermined acoustic state of an acoustic model, wherein the feature vector comprises a plurality of extracted and/or derived quantities from an audio signal during a defined audio time frame; wherein said accelerator is configured to autonomously compute distances for every acoustic state defined by the acoustic model.

A further aspect of the invention comprises an accelerator for calculating distances for a speech recognition circuit, the accelerator comprising: calculating circuit for calculating distances indicating the similarity between a feature vector and a plurality of predetermined acoustic states of an acoustic model, wherein the feature vector comprises a plurality of extracted and/or derived quantities from an audio signal during a defined audio time frame; first and second storage circuit, which may be referred to as result memories, each for storing calculated distances for at least one said audio time frame, and for making said stored distances available for use by another part of the speech recognition circuit; control circuit for controlling read and write access to the first and second storage circuit, said control means being configured to allow writing to one said storage means while the other said storage means is available for reading, to allow first calculated distances for one audio time frame to be written to one said storage means while second calculated distances for an earlier audio time frame are made available for reading from the other said storage means.

Embodiments of the invention may comprise means for generating a checksum or computed signature for the acoustic model data stored in the memory, and means for comparing checksums or computed signatures that have been calculated at different

DSMDB-#3348563-v1

times, to indicate an error status if the checksums do not match, one possible cause of such mismatch being that the acoustic model data has been overwritten by said other data and said error status being used to indicate that the acoustic model should be re-loaded into the said memory.

A further aspect of the invention comprises a speech recognition circuit comprising: lexical memory containing lexical data for word recognition, said lexical data comprising a lexical tree data structure comprising a model of words; means for accessing a state model corresponding to each phone or each group of phones in the lexical tree; a content addressable memory for storing content addressable data for each phone or group of phones, including states corresponding to said phone or group of phones, and for storing an address value for each said phone or group of phones; a RAM configured to store accumulated scores for each said phone or group of phones, the accumulated scores being addressable by said address value for each said phone or group of phones; means to obtain scores that each of a plurality of frames of an audio signals corresponds to each of a plurality of said states; a counter to sequentially search for each said state in the content addressable memory, to obtain the corresponding address value if the state is found in the content addressable memory; means to use said address value to access an accumulated likelihood and an accumulator to add said likelihood to the accumulated likelihood; means to use the phones or groups of phones with the highest accumulated scores to obtain a plurality of next phones from the lexical tree which correspond to the next phone; and output means for outputting a lexical tree path of highest likelihood.

A further aspect of the invention comprises speech recognition apparatus comprising: a lexical tree having a corresponding state model; means for obtaining scores of an audio input corresponding to each of a plurality of states in said state model; a content addressable memory for storing a marker indicating a part of the lexical tree, and one or more states associated with said part of the lexical tree; a random access memory addressable by the CAM output, to output accumulated scores for states corresponding to said parts of the lexical tree; adder means for adding likelihood to said accumulated likelihood, to be stored back in the RAM.

A further aspect of the invention comprises speech recognition apparatus, comprising: a CAM-RAM arrangement for storing records including pointers to a lexical tree, and accumulative scores for states within the lexical tree; input means for obtaining scores that an audio frame corresponds to a particular state in the lexical tree; an accumulator for calculating the updated scores and modifying the records in the CAM-RAM accordingly; output means for outputting a path of highest likelihood in the lexical tree.

A further aspect of the invention comprises a speech recognition method comprising: storing state identifiers which identify states corresponding to nodes or groups of adjacent nodes in a lexical tree, and scores corresponding to said state identifiers in a memory structure, the lexical tree comprising a model of words and the memory structure being adapted to allow lookup to identify particular state identifiers, reading of the scores corresponding to the state identifiers, and writing back of the scores to the memory structure after modification of the scores; repeating the following sequence of steps for each of a plurality of incoming frames of an audio signal; obtaining score updates corresponding to the likelihoods that said frame of the audio signal corresponds to each of a plurality of said states; accessing said memory structure to obtain scores, updating the scores by adding score updates to the scores, and writing back the updated scores to the memory structure; determining if scores for states furthest on in the lexical tree correspond to a significant likelihood, and if so, then accessing the lexical tree to determine the next set of possible states; and storing the next set of possible states and said scores of significant likelihood in the memory structure.

A further aspect of the invention comprises speech recognition circuit comprising: a circuit for providing state identifiers which identify states corresponding to phones or groups of adjacent phones in a lexical tree, and for providing scores corresponding to said state identifiers, the lexical tree comprising a model of words; a memory structure for receiving and storing state identifiers and phone instance identifiers uniquely identifying instances of phones or groups of phones in the lexical tree; said memory structure being adapted to allow lookup to identify particular state identifiers, reading of the scores corresponding to the state identifiers, and writing back of the scores to the memory structure after modification of the scores; an accumulator for receiving score updates corresponding to particular state identifiers from a score update generating

DSMDB-#3348563-v1

circuit which generates the score updates using audio input, for receiving scores from the memory structure, and for modifying said scores by adding said score updates to said scores; and a selector circuit for selecting at least one phone instance identifier according to said scores.

A speech recognition apparatus according to the invention may be embedded in or included in a mobile electronic device such as a mobile telephone, PDA (personal digital assistant), etc.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figures 1A-1B are a block diagram of the system architecture for a speech recognition apparatus according to an embodiment of the invention;

Figures 2A-2B are a block diagram showing the main data structures used in the speech recognition apparatus of Figures 1A-1B;

Figures 3A-3B show a block diagram of the Single Phone Instance Network Engine (PINE) of Figure 1B;

Figure 4 is a block diagram showing the Wave Front Manager architecture;

Figure 5 shows the use of link nodes to permit path merging;

Figure 6 shows a block diagram of a Path Merge Unit in the Phone Level;

Figure 7 shows the Phone Book architecture;

Figures 8A-8B show Lexical tree storage;

DSMDB-#3348563-v1

Figure 9 shows the data structure for the word link record;

Figure 10 shows a block diagram of Word Engine;

Figure 11 shows a Language Model interface architecture;

Figure 12 shows an example of a simple Loop Grammar;

Figure 13 shows a block diagram of grammar engine;

Figure 14 shows a single frame of the Audio Codec link protocol; and

Figure 15 illustrates the system timing which may be used in the embodiment of Figures 1A-1B;

Figure 16 is a block diagram showing a speech recognition circuit according to an embodiment of the invention, and illustrating data flow between parts of the circuit;

Figure 17 is a block diagram showing a distance calculation engine according to an embodiment of the invention;

Figure 18 shows ideal data flow over several time steps within a speech recognition circuit having separate front end and search stages, in an embodiment of the invention;

Figure 19 shows an example of actual data flow over several time steps within a speech recognition circuit having separate front end and search stages, in a further embodiment of the invention;

Figure 20 shows ideal data flow over several time steps within a speech recognition circuit having a shared processor for front end and search stages, in an embodiment of the invention;

Figure 21 shows an example of actual data flow over several time steps within a speech recognition circuit having a shared processor for front end and search stages, in an embodiment of the invention;

Figure 22 shows ideal data flow in a speech accelerator pipeline with two data transfer stages, according to an embodiment of the invention; and

Figure 23 shows an example of actual data flow in a speech accelerator pipeline with two data transfer stages according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figures 1A-1B show a speech recognition system architecture for an embodiment of the invention. The speech recognition system is divided into several levels of operation: a state level, a phone level, a word level, a grammar level, and a system level. Each level is shown in Figures 1A-1B as a, interconnected to other levels of the system.

The information passed between different levels or subcircuits of Figures 1A-1B is indicated on Figures 1A-1B using the following identifiers.

Identifier	Definition
grammar_id	Identifies which grammar model is currently being used
grammar_node_id	Identifies a node in the grammar model
word_id	Uniquely identifies a word
wordlat_node_id	Identifies a node in the word lattice
LT_id	Lexical tree identifier. Identifies the lexical tree to be used
LT_branch_id	Identifies a unique branch in a lexical tree. A branch is defined as all of the monophones subsequent to and including the first monophone in a tree.
LT_node_id	Identifies a node in a lexical tree
monophone_id	Identifies a monophone
transvec_id	Identifies which transition vector to use
state_id	Identifies the state
phone_no	Unique phone instance identifier
prev_phone_no	Phone number of previous phone in network
triphone_id	Triphone identifier
LML_factor	Language model lookahead factor. The smeared language model score that must be added

DSMDB-#3348563-v1

The system level includes a general purpose processor and cache, a system controller, a DMA (direct memory access) controller, clocks, PLL (phased locked loop), GPIO (general purpose I/O interface), timer watch dog, memory interfaces to interface to main memory and non volatile storage, serial, host, debug and display interfaces. The system level also includes a model adaptation engine, a language model factor engine, a speech recognition system controller, a language model interface, and a recognition output interface.

The processor organises the transfer of the various models from the non volatile storage to the various model memories. It handles the language model factor requests. Language model factors are used to by the state level to apply the language model score. The language model (LM) interface provides the mechanism for obtaining LM factors as and when they are needed. The processor keeps a copy of the wave front in memory which is updated simultaneously with that in the wave front manager. Using its copy of the lexical tree, the processor can determine which language model scores may be needed in the coming frames. Once calculated, the language model factors can be kept in memory for use in subsequent frames. The system level is also used for forming speech model adaptation, and acting as an overflow for the various buffers used by the architecture.

The system level includes the audio codec interface and the audio front end interface, connected to the processor via a APB (advanced peripheral bus) to AHB (advanced high performance bus) bridge.

The grammar level includes a grammar engine and grammar model(s). The grammar level sets out the rules of the particular recognition task at the highest level. During system initialisation a number of grammar models might be loaded, for example a command grammar and a dictation grammar. Command mode grammar is a restricted grammar which includes only a very limited lexical tree with command specific words. Some branches of this lexical tree may include multiple words together, where a command has such a structure, e.g. "open file menu". Typically, only a single instance of the lexical tree will be needed. Command mode is used because it can obtain a higher accuracy, due to the lexical tree constraint. In dictation mode grammar, a full lexical tree structure is used.

DSMDB-#3348563-v1

The grammar may be selected by a user during the recognition process. For example, the command grammar may be selected when the user presses a command button. The architecture supports fast swapping of grammar models. For example, a user may be in the middle of dictating a message when a “punctuation button” is pressed and the phrase “question mark” uttered. At this point, a punctuation grammar could be loaded, in order to increase recognition accuracy.

The word level comprises a word engine, which includes a word lattice and a word graph generator. The word level has two main functions, of keeping a record of the search path at the word level using a word lattice, and interfacing to the language model via a cache. The word lattice is a record of all of the words that have been searched up to the current frame. As new words are searched, they are appended into the lattice at the appropriate place. After the final frame in the current sentence, the n-best word lattice node identifiers from the n-best scoring tokens are passed back to the word lattice. This enables the n-best sentences to be traced back through the lattice. The word lattice can also be post processed to produce a word graph which is essentially a compact version of the word lattice.

At the word level, a record is kept of all tokens that have passed through a word end. This results in a lattice of words. When the final feature vector has been processed, the best scoring token is identified and used to trace back the search path back to the start of the search.

The phone level includes a “phone book” for generating new triphone instances, a wave front manager for sending requests to the phone book to generate new phone instances, a phone request arbiter, and a path merge unit.

The phone book includes a lexical tree, which comprises a vocabulary of words, sounds and/or phrases organised according to the phones or monophones making up the sound. A lexical tree is commonly used in speech recognition rather than a linear lexicon, because it greatly reduces the amount of storage space required to store a word vocabulary, by avoiding duplication of common word beginnings. It is also easier to search when the word beginning is known.

DSMDB-#3348563-v1

Although it is possible to search a lexical tree based on monophones, it is well known that better results can be achieved in speech recognition by basing the lexical tree search on groups of more than one monophone, since the sound of one syllable commonly affects the pronunciation of the next syllable in the same word, or the first syllable in the next word. In this embodiment, groups of three monophones are used, and these are known as triphones. It is alternatively possible to use a different number of monophones for each group. The phone generator in the “phone book” uses data from the lexical tree to generate triphones.

In one embodiment of the invention, approximately 45 different cued mono-phones are considered. This gives of order 60,000 tri-phones. Each triphone in this embodiment is modelled by a Hidden Markov Model (HMM) with three states. There is no correspondence between individual states and individual phones of the triphone. Alternatively, it is possible to model each triphone by a single state, or by any other number of states.

The phone level is responsible for generating the search space according to the word structure specified in the grammar model and the phone structure specified by the lexical tree. The search space can be envisaged as a wave moving through an acousto-linguistic search space. In order that the wave can be dynamically created, a record of the wave front phone instances must be maintained. This is done in a wave front phone CAM/RAM structure, within the wave front manager. When a token occurs in the final state of a phone instance in the PINE array at the front of the wave, a new phone instance must be generated at the phone level and passed to the PINE array.

The phone instance includes a phone_no, which is a phone instance identifier to identify the phone uniquely in the current search. The phone instance also includes a word lattice node identifier (labelled “word_lat_node_id”) to identify the phone’s position in the word lattice, state identifiers (labelled “state_ids”) for the HMM, storage to hold the tokens as they move through the HMM and LM_factor which is used to by the PINE to apply the language model score. To handle the connectivity between phone instances, a phone instance points to a previous phone instance using the phone_no of the previous phone. This allows tokens to be passed between phones. A “token” is a hypothesized

DSMDB-#3348563-v1

search path, represented by a non-zero score at a particular state, and tokens are propagated through the HMM with each time frame.

The phone instance network can be visualised as a wave with new phone instances being added to the front of the wave, and empty phone instances being removed from the back of the wave. In some circumstances, the back of the wave may be in the same position as the wave front. The back of the wave moves on through the lexical tree as tokens are either moved on or pruned.

Tokens, each representing a particular hypothesis, hop through the states in the HMMs, each hop consuming a feature vector. Each token explores all possible next states which typically involves placing a token in the next state and keeping one in the current state. When there is more than one token in a state, the least likely token can be removed because both have the same word history. When a token reaches the last state of a phone instance at the front of the wave, the next phone instance(s) are requested from the phone level. The phone level uses the lexical tree to create the new phones. As well as looking up the state identification numbers for each HMM state, it also incorporates the correct language model factor.

The language model factor engine is used to smear the language model scores across the lexical tree. This operates in two ways depending on whether the factor is needed for a word start phone(s) or word internal phones. For word internal phones, LM factors can be computed as needed, provided that the LM scores are available in cache. This is because in general the storage requirements for the LM scores are less than the storage of the LM factors for all of the factors in the branch. However when a token is in a word end phone there may be a significant number of word start phones e.g. up to (monophones)² requiring a large amount of computation. To avoid this the LM engine will pre compute word start LM factors when it seems highly likely that a token will be passed to the word starts. These word start LM factors will be held in the LM factors cache. The phone level is responsible for generating the search space according to the word structure specified in the grammar model and the phone structure specified by the lexical tree.

DSMDB-#3348563-v1

The state level shown in Figure 1B includes a phone instance network engine (PINE) array, a MHD (Mahalanobis distance) engine, an audio front end, a feature vector buffer for model adaptation, and compressed acoustic models. The state level is the lowest level, in which a network of phones is dynamically created.

The audio input for speech recognition is input to the front end unit of the state level. This audio input may be digital audio, or it may be analog audio which is converted to digital audio using an analog to digital converter. The audio input is divided into time frames, each time frame typically being of the order of 10ms.

For each audio input time frame, the audio signal is converted into a feature vector. This may be done by splitting the audio signal into spectral components. For example, in the present embodiment, the audio input signal is split into 12 frequency bands and one energy band, and a feature vector is constructed using these 13 components, plus their first and second derivatives, a total of 39 components.

The feature vector is passed to the MHD (Mahalanobis distance) engine. This computes all possible state output likelihoods for the current frame as the state mean and covariance data is read from the acoustic model memory.

The likelihood of each state is determined by the distance between the feature vector and each state. To reduce the number of states, a technique known as “state tying” may be used. Multiple states are grouped together in 39D space, according to their proximity to one another. An average or central value from each group is used to represent the entire group. Thus, the effective number of states may be reduced from 40,000 to 8,000. The accuracy may be improved using a language model in addition to the word recognition lexical tree. A language model is based on the probability of sequential words. The above figure of 8,000 states gives best recognition results when used with a language model.

The MHD (Mahalanobis Distance) is a distance between two N-dimensional points, scaled by the statistical variation in each component. In embodiments of the invention,

DSMDB-#3348563-v1

the Mahalanobis distance between the feature vector and each state is calculated, to determine similarity of the feature vector to each state.

The Feature Vector represents a point in an N-dimensional space. Different embodiments of the invention may use different values of N. Generally we see values of N in the range of 20 to 39, although values outside this range are not excluded.

The distance calculator stage of the recognition process computes a probability or likelihood that a feature vector corresponds to a particular state. Each state has an associated probability distribution in the N-dimensional space. The distance calculator computes the distance in the N-dimensional space from the Feature Vector to the probability distribution for each state. In some embodiments, the calculation is performed only when the search stage algorithms determine that the distance for that particular state is needed by the search stage. In some embodiments, the distances for all of the states are calculated and made available to the search stage, which allows pipelined operation. Embodiments that do the calculations on demand are preferable.

The Mahalanobis Distance (MHD) is extensively described in the literature. It may be calculated using the formula

$$\text{MHD value} = \ln(P(x)) = -0.5[N \cdot \ln(2 \cdot \pi) + \ln|\text{Cov}|] + (-0.5 \cdot [(x-u)' \cdot \text{inverse}(\text{Cov}) \cdot (x-u)])$$

where

N= number of dimensions in the space

x=feature vector, a point in the N dimensional space

u=mean vector

Cov=covariance matrix

|Cov|=determinant of covariance matrix

(a)'= transpose of matrix a

Different embodiments may use variations on this equation, for example the base for the logarithm is not always the natural base e that is used in the above example. The log of the probability is preferably calculated, rather than the probability, as it simplifies the implementation. Not all embodiments do this. In alternative embodiments, the likelihoods rather than probabilities are computed as it leads to simpler implementations of computations in the search stage.

The feature vector is used to calculate 8,000 MHD distances for each time frame, i.e. one distance for each of the 8,000 states. It is beneficial, although not essential, to perform the MHD calculations separately in a pre-processor, known as a MHD engine, because MHD calculation is a computationally intensive process.

Due to the 10ms frame length, a feature vector arrives at the MHD engine of the state level every 10ms. The MHD engine calculates the likelihood of each feature vector corresponding to each of a number of states derived from an acoustic model. Each state is also a 39 dimensional vector, having the same spectral components as the feature vector. The MHD engine calculates the difference in 39-D space between the feature vector and each state to obtain the likelihood for that state. A parallel MHD or a single MHD arrangement may be used.

The feature vectors may be sent to a feature vector buffer for model adaptation, which determines whether the observed features fit well with the acoustic model, and may initiate an adaptation of the model if the fit is not sufficiently good.

Scores indicating the state likelihoods or probabilities are output from the MHD engine to the PINE array. It is preferable to use likelihoods rather than probabilities, because likelihoods can be added together to give a total likelihood. Probabilities must be multiplied, which can give rise to rounding errors. In this embodiment, eight separate pines are used, each of 4 kilobytes, thus storing 4000 phones per pine.

During recognition, the Phone Instance Network Engine (PINE) Array will contain a network of phone instances. Tokens representing search hypotheses in the form of a probability(likelihood) will exist in some of the HMM states of the phone instances. The

DSMDB-#3348563-v1

tokens are propagated into all available states and updated with the corresponding state output probabilities. Maximum model, word end and beam pruning are applied to the tokens to ensure the search space is kept bounded. When a token is in the final state of a phone model and has avoided being pruned, the PINE makes a next phone request to the Phone level. The PINE array may receive a number of new phone models from the phone level which must be appended to the appropriate existing phone.

Figure 2B shows the data structures stored in the PINE array, and other main data structures used during the acoustic search. Each PINE includes a phone instance CAM and a phone instance RAM. The phone instance CAM stores the phone instance number (labelled “phone_no”) which uniquely identifies each phone instance. The phone instance CAM also stores the state IDs of the three HMM states used to model the phone, and flags indicating the position of the phone. The “back” flag indicates if the phone is at the back of the wavefront. The “front” flag indicates if the phone is at the front of the wavefront. The “empty” flag indicates if there are no tokens in any of the three states for that phone. The “end” flag indicates if the phone is at the end of a word.

The phone instance RAM stores the token scores for each of the three states corresponding to each phone. The st_dum field indicates a dummy state, which is used to represent a state prior to the first state of the first phone. The “tmp” field is used for passing a token from a previous phone in the lexical tree to the current phone. This will be explained in more detail later. The prev_phone_no field is for storing a pointer to the previous phone instance in the network. The wordlat_node_id identifies the position of the phone instance on the word lattice node, and the LM_factor stores the language model factor for that phone instance.

At the phone level a record of all phones on the wave front must be maintained. If a phone instance is on the wave front, the phone_no stored in the phone instance CAM will point to the details in the wave front phone CAM and RAM.

The wave front phone CAM and RAM keeps pointers into the grammar and lexical tree. When a token in a wave front phone reaches the last state, the PINE array makes a request to the phone level by passing back the phone_id. The phone_id is used as a key

DSMDB-#3348563-v1

to determining the token's `grammar_node_id` and `LT_node_id` within the wave front CAM/RAM structure. If the token is at a word end the grammar model is checked. A grammar identifier points to the type of grammar.

At the word level, figures 2A-2B show an example word lattice structure, with silence having a `word_id` of 0, and a `wordlat_node_id` of 6. This is shown as connected to two word nodes – “which” has a `word_id` of 238 and a `wordlat_node_id` of 4, and “neither” has a `word_id` of 3556 and a `wordlat_node_id` of 5.

At the highest level, i.e. the grammar level, a simple grammar model is shown which will allow the phrase ‘silence which house silence’ or ‘silence neither house silence’ to be spoken. Whenever a word end is reached during the search, the grammar model must be consulted to determine the next possible word(s) or lexical tree(s). This allows the word lattice at the word level, consisting of a network of words, to be assembled. Whenever a word end is reached, the word is added to the lattice and the `wordlat_node_id` updated accordingly.

Figures 3A-3B show a single phone instance network engine (PINE). At system power up the state transition RAM will be loaded with the appropriate state transition vector but the CAM and RAM memories will be empty. When the system is initialised an initial set of phone model(s) will be instantiated with a token in the dummy state and a path score of zero. The phone models are received from the phone level.

When the audio data is processed, a stream of all possible state output probabilities will be received from the MHD engine. A counter that is synchronous to the stream of state output probabilities from the MHD produces a concurrent stream of `state_ids`. The job of the CAM is to perform a search to see if the `state_id` is instantiated anywhere in the CAM `state_id` fields. If it exists and the location doesn't have its empty flag set then this indicates that a token may need to be updated. The match signal from the CAM must be encoded into an address for the RAM. The corresponding RAM address can then be read. Any token that will end up in the identified state can then have its path score updated and including any transition vector likelihood.

DSMDB-#3348563-v1

It is possible that a single state_id generates multiple CAM matches. To handle this situation the CAM and RAM are effectively decoupled by means of a match fifo and repeat count fifo. The state_id and corresponding state_output_probability are stored in the repeat count fifo. During a subsequent frame when there is no match, this state_id can be revisited, ignoring the matches already dealt with.

All searched phone instances corresponding to words in the current possible lexical tree are stored in the CAM-RAM. This includes phone instances at the beginning of the word, middle of the word and end of the word. These phone instances are connected by the “previous phone instance” field, which allows traceback through an entire word within the CAM-RAM.

A mechanism is thus required to pass tokens from one phone instance to the next. This mechanism is provided by the tmp field in the RAM.

The tmp field in the RAM holds tokens during token passing.

At the start of each time frame, the audio input is converted into a feature vector, the MHD values calculated, and the MHD values used to update the token scores. After update of all tokens, 2 passes are made through the CAM-RAM, firstly for pruning, and then for token passing. The pruning in this example removes tokens of below a threshold score, although other types of pruning are also possible. The pruning may involve removing a predetermined proportion of the tokens, chosen as those of lowest likelihood. A third possible method of pruning is removing all tokens with scores below a first predetermined value, but at word ends removing all tokens with scores below a second predetermined value, the second predetermined value being greater than the first predetermined value.

The token passing works as follows. Each phone instance points back to a previous phone, due to the prev_phone_no field stored in the RAM. During the token passing, the third token of a previous phone is grabbed and put into the tmp field of the current phone.

In other words, if a token is on the third state, and it is not pruned, it must enter the next state of the next phone. To do this, the tokens that lie in the third state are moved across to the tmp field.

For example, phone no. 345 needs to look at phone no. 320 to see if any tokens are in the third state. If yes, it takes these tokens and puts them in the tmp field.

One field in the CAM is labelled “front”. This indicates whether the phone is on the wave front. If the phone is not on the wave front, then another phone will have picked up the token in the third state. However, if the phone is on the wave front, then no other phone will have picked up the token on the third state. Therefore, if there is a token in the third state, and the phone is on the wavefront, it is necessary to generate a next phone request.

The current phone number, word lattice identifier and word end flag are sent to the wave front manager, to request a next phone. If the word end flag indicates a word end, then the score is also passed to the wave front manager. The score can be passed in any case, but it is only required at a word end.

The phone level generates the next phone or phones. A new phone instance starts, and all of state1, state2 and state3 have scores pre-set to zero. The old score is provided in the tmp field.

The old phone is still within the wave, but it is unlikely to be at the back of the wave at this stage. When it gets to the back of the wave, it is deleted. Until it gets there, it remains in the CAM-RAM as before. The third token remains where it was, since tokens can either stay in the third state or move on.

Note that it is possible to end up with more than one token in each state. If this occurs, then the token with the poorer score is deleted. This does not affect the result of the lexical tree search, because both tokens on the same state have exactly the same word history.

Figure 4 shows a block diagram of the wave front manager. The wave front manager is responsible for maintaining a record of the set of phone models that exist at the front of the acoustic search space. This set of phone models is referred to as the “wave front.” The key requirement for this block is to minimise the time between receiving a “next phone request” and generating the appropriate responses. The wave front manager receives the next phone requests from the PINE array and generates appropriate responses for the Phone Book, Word Engine, Path Merge Unit and Grammar engine. The wave front manager also receives new wave front phone models from the Phone Book and updates the wave front.

The wave front manager receives the next phone request from the PINE array, via the phone request arbiter. The next phone request consists of a `phone_no`, `word_end_flag` and `wordlat_node_id`. If this phone is at the end of a word, the `word_end_flag` will be set.

The wave front manager uses the `phone_no` as input to the wave front phone CAM. As each `phone_no` is unique, it is guaranteed that only a single match will be generated. The wave front manager uses the CAM generated address to access the RAM. It then sends retrieved data from the RAM to the phone book, grammar engine, Path Merge Unit and word engine.

The wave front manager removes phone instance from CAM and RAM by writing 0's into CAM and RAM, and increments the empty count by one.

Sometime later, the wave front manager receives new phone instances from the phone book and place them in a fifo. The phone instances are taken from the fifo and written into the CAM and RAM at the `next_empty_address`. The `empty_count` is decremented by one. A match is run on the “occupy” flag, and the address is placed into `next_empty_address` register.

Both the phone level and the state level make use of a path merge unit to reduce the number of possible paths, by path merging. Figure 5 shows the use of link nodes to permit path merging. In the present embodiment, all word end phone instances are

DSMDB-#3348563-v1

linked to word start triphones with a link node. During each frame the path merge unit looks for *word end* tokens that can be recombined. This is done by looking for word end phones that have the same word_id and word history. When they are found a link is added using the path merge unit in the PINE array.

Figure 6 shows a diagram of the architecture at the Phone level, showing the path merge unit. The path merge unit includes a path merge CAM and a path merge RAM. The path merge CAM stores word_id, word history for two words, and an “occupy” field. The path merge RAM stores a link_node_id. A CAM RAM occupancy manager keeps track of the empty addresses in the CAM and RAM. The path merge unit is only passed information from the wave front manager if the phone is a *word end phone*. For each frame, there is a check to see if there is an identical word with the same history, and if there is, the link_node_id is passed to the path merge unit. If there isn’t an identical word with the same history, a new record is written into the Path merge CAM-RAM.

Figure 7 is a block diagram showing the Phone Book architecture. The Phone Book is tasked with generating the next phone models when given a pointer into the lexical tree. The key requirement for this block as to minimise the time between receiving a “lexical tree address” and generating the appropriate responses, in particular the next phone model instance(s).

The lexical tree is used to build the next triphones based on the current context.

change
1.5

Current triphone	Next triphone	comments
Left context = a Central context = b Right context = c	discarded Left context = b Central context = c Right context = ?	next_leftmph_id stored in wave front manager current LT_node_id stored in wave front manager must be looked up. Large (< 45) for new word start

The LT_node_id, next_left_mph_id and prev_phone_no are received at the phone generator from the wavefront manager. The LT_node_id is then used to address the lexical tree store and return the central monophone_id. One or more further accesses to memory are performed using the microcoded instructions, each memory access returning the right monophone_id.

A two stage lookup is accessed using the three monophone_ids (giving a triphone), the associated state_ids. The phone model is assembled, and returned to the PINE array. The wavefront manager is then updated by sending it relevant parts of the phone model.

The lexical tree store is a micro-coded ROM with both instruction and data fields. The lexical tree store holds a number of lexical trees. Each lexical tree may hold single words or multiple words. Each lexical tree comprises lexical tree elements, and each element has three fields. For now, the fields will be integers and will be converted to std_logic_vector later. The following is an example of the lexical tree store, with the LT data in word_id order.

*change
1.5*

ZERO	z i y r o w	19 8 12 11 0 0 0 0	sp 0
ONE	w a h n	18 2 10 0 0 0 0 0	sil 1
TWO	t u w	14 16 0 0 0 0 0 0	ah 2
THREE	th r i y	15 12 8 0 0 0 0 0	ay 3
FOUR	f o w r	6 11 12 0 0 0 0 0	eh 4
FIVE	f a y v	6 3 17 0 0 0 0 0	ey 5
SIX	s i h k s	13 7 9 13 0 0 0 0	f 6
SEVEN	s e h v i h n	13 4 17 7 10 0 0 0	ih 7
EIGHT	e y t	5 14 0 0 0 0 0 0	iy 8
NINE	n a y n	10 3 10 0 0 0 0 0	k 9
OH	o w	11 0 0 0 0 0 0 0	n 10
			ow 11
			r 12
			s 13
			t 14
EIGHT	e y t	5 14 0 0 0 0 0 0	th 15
FIVE	f a y v	6 3 17 0 0 0 0 0	uw 16
FOUR	f o w r	6 11 12 0 0 0 0 0	v 17
NINE	n a y n	10 3 10 0 0 0 0 0	w 18
OH	o w	11 0 0 0 0 0 0 0	z 19
SEVEN	s e h v i h n	13 4 17 7 10 0 0 0	
SIX	s i h k s	13 7 9 13 0 0 0 0	
TWO	t u w	14 16 0 0 0 0 0 0	
THREE	th r i y	15 12 8 0 0 0 0 0	
ONE	w a h n	18 2 10 0 0 0 0 0	
ZERO	z i y r o w	19 8 12 11 0 0 0 0	

The following is an example of the three fields of a lexical tree element. The first field is an instruction field, and the second and third fields are data fields.

*change
1.5*

Fields:	1st instr 2b	2nd data 6b	3rd data 6b	
	00	MP	MP	- raw data field.
	01	br	br	- branch address.
	10	br	br	- branch address.
	11	word_id		- double field.

Note that there might be no difference between "01" and "10"

To build the "ey" tree, we only have to consider "EIGHT", because this is the only word in the lexical tree which starts with "ey". EIGHT is represented as "ey t", with phone_nos 5 14 0 0 0 0 0. The first lexical tree element stores the raw data specifying the phone_nos, the second lexical tree element is an end field, and the subsequent lexical tree element is set to zero.

0 00 5 14 - raw data
1 11 8 0 - end field
2 00 0 0

To build the "f" tree, we have:

FIVE f ay v 6 3 17 0 0 0 0 0
FOUR f ow r 6 11 12 0 0 0 0 0

0 01 4 0 -- branch forward four places as well
1 00 6 3 -- as the implied other brach of next (one).
2 00 3 17
3 11 5 0 -- all addresses are relative offsets.
4 00 6 11
5 00 11 12
6 11 4 0
7 00 0 0

*change
1.5*

To build the "n" tree, we have:

NINE n ay n 10 3 10 0 0 0 0 0

0 00 10 3
1 00 3 10
2 11 9 0
3 00 0 0

To build the "ow" tree, we have "ow" in both the first and second data fields, as no tree is going to be a single monophone long.

OH ow 11 0 0 0 0 0 0 0
 0 11 11 0
 1 00 0 0

To build the "s" tree, we have:

SEVEN s eh v ih n 13 4 17 7 10 0 0 0
 SIX s ih k s 13 7 9 13 0 0 0 0

0 01 6 0 -- branch forward six places.
 1 00 13 4
 2 00 4 17
 3 00 17 7
 4 00 7 10
 5 11 7 0
 6 00 13 7
 7 00 7 9
 8 00 9 13
 9 11 6 0
 10 00 0 0

*change
1.5*

To build the "t" tree, we have:

TWO t uw 14 16 0 0 0 0 0 0
 0 00 14 16
 1 11 2 0
 2 00 0 0

To build the "th" tree, we have:

THREE th r iy 15 12 8 0 0 0 0 0
 0 00 15 12
 1 00 12 8
 2 11 3 0
 3 00 0 0

To build the "w" tree, we have:

ONE w ah n 18 2 10 0 0 0 0 0
 0 00 18 12
 1 00 12 10
 2 11 1 0
 3 00 0 0

DSMDB-#3348563-v1

To build the "z" tree, we have:

ZERO z i y r o w 19 8 12 11 0 0 0 0

0 00 19 8
1 00 8 12
2 00 12 11
3 11 0 0
4 00 0 0

Once all of the branches have been built, the next stage is to concatenate the tree. It appears that four elements make a page, and each element is two bytes.

ey: 00 5 14 11 8 0 00 0 0 00 0 0 page 5
f: 01 4 0 00 6 3 00 3 17 11 5 0 6
00 6 11 00 11 12 11 4 0 00 0 0 7
n: 00 10 3 00 3 10 11 9 0 00 0 0 8
ow: 11 11 0 00 0 0 00 0 0 00 0 0 9
s: 01 6 0 00 13 4 00 4 17 00 17 7 10
00 7 10 11 7 0 00 13 7 00 7 9 11
00 9 13 11 6 0 00 0 0 00 0 0 12
t: 00 14 16 11 2 0 00 0 0 00 0 0 13
th: 00 15 12 00 12 8 11 3 0 00 0 0 14
w: 00 18 12 00 12 10 11 1 0 00 0 0 15
z: 00 19 8 00 8 12 00 12 11 11 0 0 16

Then, the start_lookup header is defined. Each row (page) is 8 bytes and hence the header takes 3 rows. The end_lookup header is two rows. Therefore, the lexical tree store proper starts at the 6th (#5) page.

sp sil ah ay eh ey f ih - MP
0 1 2 3 4 5 6 7 - MP_id
3 0 0 0 5 0 6 0 - data stored.

iy k n ow r s t th
8 9 10 11 12 13 14 15
0 0 8 9 0 10 13 14

uw v w z
16 17 18 19
0 0 15 16 0 0 0 0

The end_lookup header is defined. 20 is out of bounds.

ey f n ow s t th w
5 6 10 11 13 14 15 18

z
19 20 20 20 20 20 20 20

To put the above information all together:

START	sp:ih	3 0	0 0	5 0	6 0	0
LOOKUP	iy:th	0 0	8 9	0 10	13 14	1
	uw: v	0 0	15 16	0 0	0 0	2
END	ey:w	5 6	10 11	13 14	15 18	3
LOOKUP	z	19 20	20 20	20 20	20 20	4
LEXICAL	ey:	00 5 14	11 8 0	00 0 0	00 0 0	5
TREE	f:	01 4 0	00 6 3	00 3 17	11 5 0	6
STORE		00 6 11	00 11 12	11 4 0	00 0 0	7
	n:	00 10 3	00 3 10	11 9 0	00 0 0	8
	ow:	11 11 0	00 0 0	00 0 0	00 0 0	9
	s:	01 6 0	00 13 4	00 4 17	00 17 7	10
		00 7 10	11 7 0	00 13 7	00 7 9	11
		00 9 13	11 6 0	00 0 0	00 0 0	12
	t:	00 14 16	11 2 0	00 0 0	00 0 0	13
	th:	00 15 12	00 12 8	11 3 0	00 0 0	14
	w:	00 18 12	00 12 10	11 1 0	00 0 0	15
	z:	00 19 8	00 8 12	00 12 11	11 0 0	16

3 0 0 0 5 0 6 0
0 0 8 9 0 10 13 14
0 0 15 16 0 0 0 0
5 6 10 11 13 14 15 18
19 20 20 20 20 20 20 20
00 5 14 11 8 0 00 0 0 00 0 0
01 4 0 00 6 3 00 3 17 11 5 0
00 6 11 00 11 12 11 4 0 00 0 0
00 10 3 00 3 10 11 9 0 00 0 0
11 11 0 00 0 0 00 0 0 00 0 0
01 6 0 00 13 4 00 4 17 00 17 7
00 7 10 11 7 0 00 13 7 00 7 9
00 9 13 11 6 0 00 0 0 00 0 0
00 14 16 11 2 0 00 0 0 00 0 0
00 15 12 00 12 8 11 3 0 00 0 0
00 18 12 00 12 10 11 1 0 00 0 0
00 19 8 00 8 12 00 12 11 11 0 0

*change
1.5*

This information is converted to hexadecimal. The first five are byte fields.

The rest are:

2bit - uCode.
6bit - data.
2bit - null.
6bit - data

03 00	00 00	05 00	06 00
00 00	08 09	00 0A	0D 0E
00 00	0F 10	00 00	00 00
05 06	0A 0B	0D 0E	0F 12
13 14	14 14	14 14	14 14
05 0E	C8 00	00 00	00 00
84 00	06 03	03 11	C5 00
06 0B	0B 0C	C4 00	00 00
0A 03	03 0A	C9 09	00 00
CB 00	00 00	00 00	00 00
86 00	0D 04	04 11	11 07
07 0A	C7 00	0D 07	07 09
00 0D	C6 00	00 00	00 00
0E 10	C2 00	00 00	00 00
0F 0C	0C 08	C3 00	00 00
12 0C	0C 0A	C1 00	00 00
13 08	08 0C	0C 0B	C0 00

*change
1.5*

The above is the micro code for the digit lexical tree. It allows branching, and it holds the word_id. It knows which lexical trees it contains and where they are housed. It also stores the possible end monophone list.

Figures 8A-8B show a further example of the Lexical Tree RAM, giving examples of different types of branching in the lexical tree, and how these are represented in the RAM.

Figure 9 is a block diagram showing the data structure that is supported for the word link record, to enable word n-best decoding. The word engine maintains a record of the words that have been currently searched by the PINE array. It is used to determine the word history and allow trace back during or at the end of search. The word history is needed to determine the language model score and whether particular paths can be recombined or merged.

Figure 10 shows a block diagram of the word engine. This is designed to allow fast access to the word history (thus allowing rapid phone creation at the phone level), and to allow reasonably fast trace back. The Word Engine is primarily a memory store that fills incrementally. The decoupling of the `wordlat_node_id` and the physical address will allow the possibility of being able to keep the WLRs for a given word in contiguous addresses, thus avoiding the need for the pointer field.

The language model interface is designed to allow the Phone Book to quickly obtain the appropriate LM factor for the phone model that it is currently constructing. Based on the phones in the wave front, all possible next LM factors can be determined. This will be done in software together with the language model factor engine. The ARM 9 will maintain its own record of the wavefront using the same next phone requests received by the wave front manager and its own copy of the lexical tree.

Figure 11 shows the architecture of the LM interface. This allows the Phone book to quickly access the LM factor for the phone that is currently being generated. Prior to the phone generation phase LM_factors are computed for all possible phones that may be generated in the next frame.

The grammar engine is designed to give the Phone Book a quick response to requests for the next grammar node(s). In the case of a task grammar where the lexicon may be limited to a few words(e.g. yes, no), it may be desirable to use the same lexical tree. In this case it will be necessary for the grammar, to provide lexical tree pointers such that the phone generator can jump to the appropriate part of the lexical tree.

For a speech recognition system it is also necessary to specify the grammar. Again we will use HTK's Standard Lattice Format(SLF) to specify the grammar. In practice it will be convenient to use HTK's HParse format to specify the grammar using a high level grammar notation. This can then be converted into using the HParse tool. Consider having to specify a grammar for a digit recognition task. The HParse format for this is as follows:

```

$digit = one | two | three | four | five | six | seven | eight | nine | zero
$digit_sp = $digit sp
$noise = lipsmack | breath | sil
( [sil] < $digit_sp | $noise > [sil] )

```

where | denotes alternatives, < > denotes one or more repetitions, { } denotes zero or more repetitions, [] denotes that item is optional, and () contains the grammar.

Figure 12 shows the network that is defined by this grammar. The network starts and ends with a “sil” phone, representing silence. Between the start and end, a number of possible routes correspond to further silence, a breath, a lipsmack sound, or a detected word.

Figure 13 shows the grammar engine architecture. When a next phone request arrives, the next grammar node(s) must be retrieved from memory. In the case of a task grammar, the phone book should be given the lexical tree memory addresses to allow the phone to be determined.

Figure 14 shows a single frame of the Audio Codec (AC) link protocol used in some embodiments of the invention. The Advanced Audio CODEC Interface (AACI) is an ARM Primecell that provides an interface to off-chip AC97 audio codecs. AC97 is Intel’s audio codec specification, initially created for the PC market. The AACI is an AMBA slave block that connects to the Advanced Peripheral bus (APB). The AACI connects to the off-chip codec using the AC-link protocol.

The AC-link protocol is a bi-directional, fixed clock rate, serial digital stream. AC-link handles multiple input and output PCM audio streams as well as register control accesses. It employs a time division multiplexed (TDM) scheme that divides each audio frame into 12 outgoing and 12 incoming data streams each with up to 20 bit sample resolution. The tag slots are used to indicate which of the subsequent slots contain valid data. Add and dat are the control register address and data slots for read and writes. The adc slots contain the outputs from the analog to digital converters. Assuming that the

DSMDB-#3348563-v1

ADC sample rate is set to 16 kHz then the AC-link controller will receive 16 bits of data every third codec frame.

The AACI has 4 channels that can be operated simultaneously. They are accessed through DMA.

Figure 15 shows the system timing. Each frame is divided into three sections:

- Token update (MHD wash)
- First pass - inter phone token passing
- Second pass – garbage collection, max model pruning

The horizontal direction of the figure indicates elapsing time, and various subcircuits and processes are indicated along the vertical direction of the figure. The top part of the diagram shows frame count, and push to dictate button, and the lower part of the diagram shows various subcircuits and processes, within a single time frame, whilst the push to dictate button is engaged. These subcircuits and processes are language model interface, word engine, phone book, PINE array, MHD engine, mean & covar reg, and feature vector reg.

At the start of the time frame, the feature vector is generated, and the MHD values are calculated. The MHD values are then passed to the PINE array, which engages in beam pruning and token score updating. The phone book receives next phone requests and returns next phone responses. The word engine adds word link records, and the language model interface supplies language model factors.

In the second part of the frame, pruning thresholds are calculated for the PINE array, for the next frame, and the PINE array engages in token passing. The language model interface transfers language model factors for the next frame.

While the “push to dictate” button remains activated, the above process repeats in a sequence of frames.

The embodiments of the invention which are now described relate to a distance engine for calculation of Mahalanobis Distances.

Figure 16 shows a block diagram of a speech recognition system according to an embodiment of the invention. The system includes a microphone 100, which outputs an audio signal to a combined AGC (automatic gain control) and ADC (analogue to digital converter) 101. In alternative embodiments, the AGC and ADC may be separate. In this embodiment, the incoming data comes from a microphone, and is then converted to digital format. However, it is also possible that the digital audio data may be received from a network, or may be read from storage means such as a memory, an optical disk, or a hard disk drive.

The digital audio data output from the ADC is input to an audio data buffer 102, and may be held here until required. The data is then output from the audio data buffer 102 to a front end circuit 103. The incoming audio data is divided into frames, as described above. In cases where the system is extremely busy, it is possible to drop some audio frames, to allow the system to catch up with any backlog.

In the audio front end circuit 103, the audio is split into spectral components to generate a feature vector, as described above. For example, the feature vector may have a number of frequency bands, and an energy band, plus their first and second derivatives. The feature vector is then output from the front end 103 to the distance calculation engine 104.

The distance calculation engine 104 converts feature vectors into Mahalanobis distances. The acoustic model may be chosen from a plurality of acoustic models, e.g. according to the language used, and may be stored in a storage means 105 such as a memory, a disk drive, etc. The distances calculated by the distance calculation engine are then transferred to the search stage 106 of the speech processing circuit, which uses models such as one or more word models and/or language models to generate and output recognised text. Such word and language models may be stored in a separate storage means 107.

DSMDB-#3348563-v1

In one embodiment, the distance calculation engine uses Mahalanobis Distance (MHD) calculations to determine distance values for output to the search stage 106. In further embodiments, different types of distance calculations may be used, e.g as described above.

In some embodiments of the invention, the distance calculation engine is designed as a speech accelerator, to operate as a loosely bound co-processor for a CPU running speech recognition software. This has the advantage of reducing computational load on the CPU, and reducing memory bandwidth load for the CPU.

In some embodiments, the accelerator interface is memory-mapped. Other types of interface are also possible to support different system configurations, for example removable accelerator cards with SDIO (secure digital input/output), CardBus, or USB interfaces.

The speech recognition apparatus may also operate with a CPU and DSP (digital signal processor), where the DSP implements the "front-end" signal processing to produce a feature vector, and the CPU implements the search stage, reading the distance results from the distance calculation engine.

The system may be implemented in any combination of hardware or software.

In some embodiments of the invention the search stage 106 provides the distance calculation engine 104 with an active state list 108 that indicates the subset of states for which the search stage requires distances to be calculated by the distance calculation engine 104. This is an optimization that may reduce computation time and/or power consumption.

In preferred embodiments of the invention, the front end 103, distance calculation engine 104 and the search stage 106 operate in a pipelined manner. When operating in a pipelined manner it is unlikely that the search stage 106 will be able to provide the active state list 108 early enough for the distance calculation engine 104 to implement the optimization of computing only the distances that will be required by the search

DSMDB-#3348563-v1

stage 106. The distance calculation circuit 104 may calculate the MHD values for every state in the lexicon, per frame, whether it is subsequently required by the search stage or not. This allows the accelerator and software system to operate in a concurrent pipelined manner, which maximizes the system throughput.

The distance calculation engine 104 may have a plurality of memory areas for storing feature vectors, to allow a feature vector for one frame to be transferred to one memory area in the distance calculation engine 104, while another feature vector for another frame is being read from a second memory area, for calculation of a distance. The memory areas used for reading and writing may alternate with subsequent frames.

Figure 17 shows a block diagram of a distance calculation accelerator 200, used in an embodiment of the invention. A feature vector is received at input 212 of the accelerator 200. The feature vector is loaded into feature vector (FV) memory A 208 via a bus 211. In reality, there may be more than a single bus, but for simplicity, only one bus 211 is shown in the figure.

The registers 203 are discussed further at the end of this description. The registers 203 may include, for example, an accelerator status register, for indicating whether the acoustic model data is correctly stored in memory, as described in more detail below.

The FV registers 209 hold the feature vector whose distances are currently being computed by the distance calculation engine 204. If the distance calculation engine 204 is implemented in software, the FV registers might be implemented as memory attached to the processor running the software. If the distance calculation engine 204 is implemented in hardware the FV registers might be implemented as logic or as a memory block. The FV registers 209 are loaded from the FV memory 208 when the Distance Calculation Engine 204 has completed distance calculations for the current feature vector and a subsequent feature vector is available in the FV memory 208. In some embodiments there is an optional FIFO associated with FV registers 209 so that the FIFO is loaded from the FV memory 208 when there is a subsequent feature vector in the FV memory 208 and there is a space in the FIFO. In these embodiments the FV Registers 209 are loaded from the associated optional FIFO 209 when the Distance

DSMDB-#3348563-v1

Calculation Engine 204 has completed calculations for the current feature vector and there is a subsequent feature vector in the associated optional FIFO 209. The presence of the FIFO 209 provides for a looser coupling between the Front End 103 and the accelerator 200 which improves performance in some system configurations.

Each feature vector is transferred to a distance calculation engine circuit 204, to obtain distances for each state of the acoustic model. The acoustic model is stored in an acoustic model memory 205, which is connected to the bus 211, for control of reading/writing to the memory. In some embodiments, this also allows acoustic models to be changed or updated via the bus. The acoustic model may be stored in compressed form in the acoustic model memory 205. A decompressor 206 is provided on the data output of the acoustic model memory. The decompressor 206 can decompress the results as they are needed, and thus allows a much smaller size of acoustic memory to be used, than with uncompressed data. From the decompressor 206, the decompressed acoustic model states are sent to the distance calculation engine 204.

The distance calculation engine 204 converts feature vectors and acoustic model states to distances. The accelerator 200 includes two separate results memories for storing these calculated distances, labelled as “result memory A” 201 and “result memory B” 202.

The control, sequencing, and address generation 210 is logic, state machines, and/or software programs that provide overall control of the other elements of the accelerator, taking inputs from the registers 203, and all the other elements and interfaces to determine when to start and stop the Distance Calculation Engine 204. The control 210 also produces addresses and control signals to control the reading and writing of the memories 201, 202, 205 and 208, to control the flow of data between the various elements of the accelerator 200, the operation of the Decompressor 206, CRC check 207 and Distance Calculation Engine 204, and to control the setting or clearing or status bits in the Registers 203 and to control signalling to the front end 103 and search stage 106. The control 210 is implemented using any of the widely known and used methods of controlling and sequencing in digital systems including logic gates, state machines, and embedded software systems. The control 210 might be a centralized control, or it might

DSMDB-#3348563-v1

be distributed in some or all of the other elements of the accelerator 200, or some combination of these.

Each time a feature vector is loaded into the accelerator, the accelerator computes the distances for all states for that feature vector, and stores the results alternately in the A or B Results Memory. The results are written to one of the two memories 201, 202, while the other memory is available for reading. Thus, while the distance calculation engine 204 is storing the distances calculated for one audio frame, the other result memory may make available the distances calculated for an earlier audio frame, to the search stage of the speech recognition system. For non-real time processing, "earlier" audio frame may refer to an audio frame that has been processed earlier, rather than being limited to an audio frame that corresponds to speech at an earlier time.

Thus, the search stage software reads results alternatively from the A or B result memory, while the accelerator writes results into the other result memory, allowing concurrent pipelined operation of the accelerator and the CPU [and the DSP if present].

In some implementations the A and B results memories may be exposed to the software interface as separate memories appearing at different locations in the CPU and/or DSP memory map. Another implementation is to map each memory alternatively to the same range of addresses in the CPU and/or DSP memory map, so that the software always accesses results at the same location, and the memory containing the set of results the software currently needs to access is the one mapped there.

The accelerator may autonomously load and decompress an acoustic model, which is stored in a compressed format in the acoustic model memory 205. The Acoustic Model may be loaded into the Acoustic Model Memory by software running on the CPU prior to the first use of the Accelerator. The Acoustic Model Memory may be an on-chip memory, to maximize bandwidth and minimize power consumption of reading in the entire Acoustic Model every frame, which means 50 to 100 times per second, depending on the system design.

The Acoustic Model Memory may be a dedicated memory attached to, or incorporated within, the accelerator. Alternatively the Acoustic Model Memory may be a shared resource that is used by other system functions or applications when speech recognition is not running. To allow detection of situations where the acoustic model has been overwritten by another application, a pre-calculated CRC (cyclic redundancy check) signature may be stored with the acoustic model. Each time the acoustic model is read into the accelerator, a CRC may be computed and compared with the signature. If the comparison fails, an error bit is set in the accelerator status register.

Some embodiments of the invention relate to speech recognition systems comprising separate front end 103 and search stage 106 processors. These separate processors perform the front-end and search stages of the recognition process, with the accelerator 200 performing the distance calculations. Typically the front-end processing would be performed on a DSP and search stage would be performed on a CPU, although this description holds for any type of processors, even dedicated hardware implementations. As shown in figure 17, the front end 103 supplies the distance accelerator with feature vectors, and the search stage 106 uses distances calculated by the accelerator.

The accelerator is designed to operate in a concurrent pipelined manner with the other processors, allowing a high level of parallelism in the system.

Figure 18 shows how this pipelined operation works in an embodiment of the invention with ideal data flow, i.e. where each of the three recognition stages require exactly the same time to process each data frame, and the processors are never delayed, stalled, or diverted to other tasks.

The time axis is divided into equal “pipeline step times”. If the step time is the same as a frame time then the system is processing audio in real time. If the step time is short than a frame time, the system is operating faster than real time, which may be valuable for batch processing of audio data, or catching up after some type of delay. If the step time is longer than a frame time, the system is operating slower than real time, which may be acceptable in some applications, for example where the person dictating pauses occasionally allowing the system to “catch up”.

DSMDB-#3348563-v1

Once the pipeline has filled (which takes two step times), the DSP, the Accelerator, and the CPU are all operating in parallel. In this idealized example, the total time available for processing a frame is three step times – one step time in the DSP, one step time in the accelerator, and one step time in the search stage. There is a latency of three step times from the time the audio data enters the front end to the completion of the search stage for that frame, and the processing of one frame finishes each step time. If the system were not pipelined, then the processing of all stages for one frame would have to complete in one step time in order to achieve the same overall throughput of frames per second. Thus pipelining in this way can achieve three times the throughput with the same processing elements, compared to a non-pipelined system. A non-pipelined system would be one in which the processing of frame $n+1$ would not start until all stages of the processing of frame n was already complete.

If, for example, in a system running in real time with a frame time and step time of $1/100^{\text{th}}$ of a second, which is 10ms, each of the three processing elements has the entire 10ms available to complete the processing of one stage of the recognition task, giving 30ms of processing time overall for each frame. If the system were not pipelined, real time operation would require that all three processing elements would be complete in 10ms, allowing only 3.3ms for each of the three processing elements.

At the end of each pipeline step time, the DSP provides a feature vector to the Accelerator, and the accelerator provides a set of distance results to the CPU. When the CPU is performing the search stage for frame n , the Accelerator is computing the distances for frame $n+1$, and the DSP is performing the front-end stage for frame $n+2$. During each pipeline step, the front end processing consumes one frame of digital audio data (although this is not indicated explicitly on the diagram).

Real world systems are more complex than the idealized system in many ways. The processing time for a frame is often highly variable and data dependent, especially in the search stage, and the processing time for each processing stage is likely to be quite different. The DSP and CPU are likely from time to time to be diverted to run other software tasks, either to service interrupts, or under control of an operating system

DSMDB-#3348563-v1

scheduling regime. Communications between the processing elements may be stalled due to competing bus traffic, for example for DMA transfers.

Recognition software may implement “frame dropping” and other advanced techniques, and as a result not all frames are processed through all of the recognition stages. Processing elements cannot respond instantly to the availability of data from the preceding processing stage, resulting in delays.

Finite buffer space for information storage between stages means that an earlier stage may be stalled waiting for a space to store its output data when following processing stages are “falling behind”.

The accelerator has been carefully designed to mitigate these problems. Figure 19 shows a pipelined system with real world complications added. The following text describes the interactions shown in the diagram. The time for the processing of various stages has been varied to show difference examples of buffering of data between stages and stalling of processing elements. The numbering corresponds to the numbered bubbles on the diagram. The text also makes reference to memories used to pass information into and out of the accelerator, which are shown in figure 17.

The front end has computed the feature vector for the first frame, FV1. It is loaded into the accelerator, causing the accelerator to commence computation of the distances for the frame.

The feature vector for the second frame, FV2, is passed to the accelerator. The accelerator is still busy computing the distances for the first frame, so this feature vector is held in the accelerators FV memory.

The front end has computed the feature vector for the third frame, FV3, but it cannot be delivered to the accelerator as the FV memory is still full, holding FV2. The time period labelled “stall delivering FV3” is the period between FV3 being computed and when it can first be delivered. Of course the DSP could be diverted to some other useful task, or it could buffer FV3 in its own memory and commence computation of FV4.

DSMDB-#3348563-v1

The arrow labelled “FV3 earliest” shows the earliest time that the feature vector FV3 can be delivered to the accelerator, which is immediately after the accelerator loads FV2 into its internal registers and frees the memory.

The arrow labelled “FV3 latest” shows the latest time that the feature vector FV3 can be delivered to the accelerator without causing the accelerator to stall for lack of a feature vector to work on. This example shows that there is a wide tolerance in the system to when FV3 can be delivered without affecting system performance – an entire accelerator frame processing time. The accelerator provides an interrupt and a status bit that can be polled to indicate the state of the FV memory. Regardless of which control method the DSP uses, the system tolerates long latency without reducing performance.

The accelerator signals to the search stage that the distances for frame 1 are available in result memory A. The search stage commences processing for frame 1, reading distances from result memory A.

The search stage has finished its processing for frame 1 before the accelerator has completed computing the distances for frame 2, so the search stage stalls waiting for the accelerator. Of course the CPU could be diverted to other useful tasks, including any search stage tasks that can be completed before distances are available. Again, an interrupt and status bits are provided by the accelerator to control this interaction between the accelerator and the CPU. .

The accelerator completes the distance calculation for frame 2, and signals to the CPU that it can commence reading the distances from result memory B.

The accelerator has completed the computation of distances for frame 3 before the search stage has finished processing frame 2. As soon as the search stage signals to the accelerator that it has completed frame 2 by releasing memory B, the accelerator signals that the distances for frame 3 are available, allowing the search stage to immediately move on to processing stage 3.

The accelerator is stalled waiting for feature vector FV4, due to the long processing time in the front end. In real systems this may be caused by a number of factors, including the DSP being diverted to another task for a period of time, or simply due to the DSP performance being insufficient to keep up with the accelerator's throughput.

The accelerator restarts as soon as FV4 is delivered by the front end.

The search stage is stalled waiting for distances for frame 4, a follow on consequence of the late delivery of FV4 to the accelerator.

FV5 is delivered to the accelerator while it is still computing the distances for the previous frame, allowing it to move onto processing frame 5 without stalling when it completes frame 4.

The accelerator signals that the distances for frame 4 are available and the search stage commences processing frame 4.

FV6 is delivered to the accelerator while it is still computing the distances for frame 5.

The availability of distances for frame 5 is signalled by the accelerator to the CPU.

Even though the accelerator has already received FV6, it is stalled waiting for a free result memory it can write the distances into.

The search stage releases result memory B at the end of its processing of frame 4. This allows the accelerator to start computing distances for frame 6.

The accelerator completes computing distances for frame 6 and signals to the CPU that the results are ready.

Further embodiments of the invention relate to systems with a Front End and Search Stage on a single processor. This single processor performs the front-end and search

stages of the recognition process, with the accelerator performing the distance calculations. This configuration is likely to be the most common configuration where the accelerator is added to an embedded system such as a PDA or mobile phone that has a high-performance embedded application processor (but no DSP) available to run speech recognition software.

In this configuration the accelerator still operates in a concurrent pipelined manner with the front-end and search stages; it is just that the front-end and search stages are running on a single processor.

CPUs in embedded systems are generally relatively simple compared to desktop PCs and servers. Embedded CPUs in mobile phones, PDAs, in-car navigation and telematics systems, and other consumer electronics devices are typically “simple uni-processors” by which we mean not multi-processors and not supporting hardware multi-threading support – i.e. they can only execute a single thread of instructions (although they can be diverted to a different thread under the control of interrupts, exceptions, or an operating system). For this discussion, except where explicitly stated otherwise, we are talking about embedded processors that are “simple uni-processors”. We are also using the term “CPU” and “processor” interchangeably.

Figure 20 shows how the system can be pipelined with a single embedded processor working with the accelerator, in an idealized system. In this idealized system, the time taken for the processor to perform the front-end and search stage processing is identical to the time the accelerator takes to perform the distance calculations, and the processor is never delayed, stalled, or diverted to other tasks.

The time axis is divided into equal “pipeline step” times. If the step time is the same as a frame time then the system is processing audio in real time. If the step time is short than a frame time, the system is operating faster than real time, which may be valuable for batch processing of audio data, or catching up after some type of delay. If the step time is longer than a frame time, the system is operating slower than real time, which may be acceptable in some applications, for example where the person dictating pauses occasionally allowing the system to “catch up”.

DSMDB-#3348563-v1

The first two pipeline step times are used to fill the pipeline. By the third pipeline step time, the pipeline is full, and all three stages of the recognition process are processed every pipeline step time, with each stage working on the processing of a different data frame.

During the first pipeline step time, the CPU completes the front-end processing for frame 1, and the feature vector passed to the accelerator so that distance calculations can start in the accelerator.

During the second pipeline step time, the front-end processing for frame 2 is completed by the CPU, in parallel with the accelerator computing the distances for frame 1.

During the third pipeline step time, the CPU runs the search stage for frame 1 reading distance results from the accelerator, and then runs the front-end processing for frame 3. The accelerator runs in parallel computing the distances for frame 2 using the feature vectors computed in the previous pipeline step.

Each subsequent pipeline step time is similar to the third pipeline step, with the frame numbers incremented. So for pipeline step $n = 4, 5, 6$, etc. the CPU runs the search stage for frame $n-2$ and then runs the front-end for frame n . In parallel the accelerator computes the distances for frame $n-1$.

Note that with this situation the production of feature vectors and the consumptions of results run in lock step - one feature vector is produced every time the search stage consumes one frame of distance results. This considerably reduces the need for elastic buffering between the distance accelerator and the software system.

Again, real world systems are more complex than the idealized system in many ways. For example, the processing time for a frame is often highly variable and data dependent, especially in the search stage, and the processing time for each processing stage is likely to be quite different. The CPU is likely from time to time to be diverted to run other software tasks, either to service interrupts, or under control of an operating

DSMDB-#3348563-v1

system scheduling régime. Communications between the CPU and accelerator may be stalled due to competing bus traffic, for example for DMA transfers.

Recognition software may implement “frame dropping” and other advanced techniques, and as a result not all frames are processed through all of the recognition stages. Processing elements cannot respond instantly to the availability of data from the preceding processing stage, resulting in delays.

Finite buffer space for information storage between stages means that an earlier stage may be stalled waiting for a space to store its output data when following processing stages are “falling behind”.

The accelerator has been carefully designed to mitigate these problems. Figure 21 shows a pipelined system with real world complications added. The following text describes the interactions shown in the diagram. The time for the processing of various stages has been varied to show difference examples of buffering of data between stages and stalling of processing elements. The numbering corresponds to the numbered bubbles on the diagram. The text also makes reference to memories used to pass information into and out of the accelerator, which are shown in figure 21.

At the end of front-end processing for frame 1, the CPU passes the feature vectors for frame 1 to the accelerator, allowing distance calculations to start.

To fill the pipeline, the CPU immediately performs the front-end processing for frame 2 and passes the feature vectors to the accelerator. They are stored in the accelerator so that the accelerator can load them into its internal engine for processing as soon as the distance calculations for frame 1 are complete.

The CPU is now ready to start the search stage for frame 1, however the distance computation in the accelerator has not completed. The box labeled “Prep search stage 1” indicates a period where the CPU could do any preparatory search stage processing that does not require distances. Alternatively the CPU could be diverted to another task, or it can stall waiting for the distance results.

DSMDB-#3348563-v1

The accelerator signals that the distance results for frame 1 are now available. The search stage can commence on the CPU, reading distance results from the accelerator's Result Memory A. The accelerator automatically starts processing the feature vector for frame 2 that was delivered earlier.

The accelerator signals to the CPU that it has completed the distance calculations for frame 2. The CPU is still busy running the search stage for frame 1, and has still to run the front end for frame 3, so the completion signal is ignored until the CPU has need of it, when it starts running the search stage for frame 2. As the accelerator has not received the feature vector for frame 3, it has no task it can perform and so it stalls.

The CPU delivers the feature vector for frame 3, restarting the accelerator, which writes results into Result Memory A. Note that Result Memory A was previously "released" at the end of the processing of search stage 1, informing the accelerator that it can overwrite the old results held in that memory with new results. The CPU then runs the search stage for frame 2, reading distance results from the accelerator's Result Memory B.

The accelerator signals to the CPU that the distance results for frame 3 are complete in Result Memory A. Again, the CPU ignores this signal until it is ready to commence the search stage for frame 3, and the accelerator stalls waiting for the next feature vector.

The CPU delivers the feature vector for frame 4, restarting the accelerator. It then starts search stage 3 reading distances from the accelerator's result memory A.

Search stage 3 and front end 5 were completed quickly, so that the feature vector for frame 5 is delivered to the accelerator while it is still running. The CPU performs all the preparatory work it can for the search stage, and then stalls waiting for distances to be available.

The accelerator signals that the distances for frame 4 are available. The CPU starts search stage 4, reading the distances from the accelerator's result memory B. The

DSMDB-#3348563-v1

accelerator automatically starts computing distances for frame 5 using the feature vector delivered earlier.

Note that with the pipelining scheme just presented which alternates front-end and search stage processing, the accelerator can never stall waiting for a result memory to be released because the feature vector the accelerator needs as an input is not produced until after the memory has been cleared. The CPU task schedule could be described with the shorthand F1 F2 S1 F3 S2 F4 S3 F5 S4.

It can be seen readily from figure 21 and the accompanying description above, that with this CPU task schedule, that the operation of the CPU and accelerator are tightly coupled and whenever the processing time for a pipeline step on the CPU is unequal to the processing time on the accelerator, then it quickly results in one of them stalling.

In a system where either the CPU or the accelerator is always (or at least in most instances) slower than the other in completing its processing task, stalling is inevitable and throughput may not be improved by adding additional buffering (e.g. additional result memories or FIFO depth on the feature vector interface). [Unless there are other difficulties to mitigate such as very low-bandwidth buses linking the CPU to the accelerator, or competing bus traffic such as DMA transfers that can significantly reduce performance.

If the processing time for a pipeline step is highly variable on either the CPU or Accelerator, such that there are regular changes in which one of them takes the longer time, then significant enhancements in throughput may be obtained by adding additional buffering between them. This may have the disadvantage of adding additional latency through the system. This buffering may be implemented by adding additional depth to the Feature Vector FIFO, and/or adding additional Result Memories. Alternatively it may be by implementing elastic buffers in software.

If the accelerator has sufficient buffering on the feature vector interface, e.g. by including the optional FIFO shown in figure 21, then increasing the pipeline depth by computing extra front end frames up front will improve throughput at a cost of extra

DSMDB-#3348563-v1

latency. This will result in a CPU task schedule of F1 F2 F3 S1 F4 S2 F5 S3 F6 S4 or F1 F2 F3 F4 S1 F5 S2 F6 S3 F7 S4, etc.

Of course it would also be possible to dynamically schedule whether the CPU should run the front-end or search stage code based on the availability, or not, of distance results and/or the space available to store more feature vectors and distance results. Given the fundamental relationship that one set of distance results is produced by the accelerator for every feature vector it consumes, and the restriction stated at the start of this section that we are discussing single-threaded CPUs, dynamic scheduling is likely to add little performance improvement while being complex to implement.

On a CPU supporting multi-threaded operation (which is a technique for improving performance in the presence of high memory system latency), or on a multi-processor, the highest system performance is likely to be achieved by running the front-end and search stages concurrently as separate threads, rather than alternating between the front-end and search stages. The system operation is then very similar to that described above, if each thread is thought of as a separate virtual processor. In some embodiments of the invention, elastic buffers may be used between stages, to accommodate varying time delays between the processing of frames at each stage. Elastic buffers in the interfaces between the three recognition stages may also significantly enhance performance of such a system.

Operation of the accelerator and the software system is not constrained to a rigid repetitive frame time. Instead the timing of the system is controlled dynamically by the availability of feature vectors into the accelerator, and the consumption of results out of the accelerator.

This allows the recognition process to fall behind real-time operation and to "catch up" again. It also allows the system to be robust in complex systems where interrupts, DMA bus traffic, memory caches, operating system task switching, and competing real-time software tasks all make guaranteeing rigid timing constraints impractical.

Elastic buffers at the major data transfer points between processing elements maximize system performance by allowing one element to continue useful processing whenever it can, regardless of whether other processing elements have been stalled or diverted to other tasks. Elastic buffers are useful for the following applications:

- (1) For buffering raw audio data arriving in real time from the microphone(s) The data would typically be moved into the buffer by a DMA and/or high priority interrupt driven software process to ensure that hard real time constraints are met.
- (2) For buffering feature vectors output from the front end. This is especially true if the front-end is implemented on a DSP or other processing element independent of the search stage.
- (3) For buffering distance results from the accelerator. Again this is especially true if the front-end and search stages are implemented on independent processing elements. This buffering is also especially valuable if the interface between the search stage processor and the accelerator is very slow, has high latency, or is subject to competing bus traffic.

The Feature Vector and Results interfaces on the accelerator already provide a level of elasticity that should be sufficient in systems where the Front End and Search stages are running on the same CPU, and the accelerator has a high-bandwidth interface to the CPU with low level of competition from other bus traffic such as DMA.

Other embodiments may have deeper FIFOs. In one embodiment, the two result memories are an implementation of a FIFO where the implementation is exposed across the interface. It would be possible to have additional memories and use them in a round-robin fashion, or to use another implementation of a FIFO that hides the FIFO depth by only exposing one result memory at a time to the interface. An example of this would be to map each memory in turn to a memory range that is always used to read results, so that at any one time the “current” result memory is mapped to that location. This would hide the FIFO depth behind the interface so that the pipeline depth can be changed without changing the software (or hardware) that accesses the interface.

DSMDB-#3348563-v1

Slow or Conflicted Bus Interfaces may occur. In the system configurations described above, the search stage reads distances directly from the Accelerator's Result Memories. The result memories allow the search stage to randomly access the distance results, as they are needed. This will only give a high performance system if the interface or bus between the CPU and Accelerator provides the CPU with low latency, high bandwidth access to the Accelerator Result Memories.

In some systems it is not practical to provide such a low latency, high bandwidth interface between the CPU and the Accelerator. This may be for many reasons, including:

- The interface is a bus shared with other devices and/or processors that generate bus traffic that keeps the bus busy such that the CPU access to the accelerator is delayed waiting for the bus to be free. DMA transfers are one example of conflicting bus traffic.
- The interface is implemented as a serial bus, or narrow parallel bus, for cost or compatibility reasons, yielding low bandwidth and high latency
- The interface crosses an industry standard interface such as PCI, USB, SDIO, Cardbus, Firewire, PCIExpress etc. that introduces protocol overheads and/or latency as would be the case where the accelerator is in a plug-in peripheral to a system containing a CPU (or CPU and DSP).

In these situations it is possible to add another pipeline stage during which the result data is copied from the Accelerator into another memory (which we will call a "buffer memory") to which the CPU does have high bandwidth, low latency access. This copying can be performed by a DMA controller, or by software running as an interrupt handler, or a separate software thread on the CPU, or a software thread on another processor, or by other commonly known means. The device controlling the transfer may be on the CPU side of the interface or on the accelerator side of the interface, depending on the capabilities of the interface. This extra pipeline stage then allows an entire pipeline step time to copy the data from the accelerator to the other memory. The data copy can optionally be performed as a sequential burst, or a number of sequential bursts, to optimize transfers across the interface between the Accelerator and the buffer

DSMDB-#3348563-v1

memory. The search stage, running on the CPU, can then randomly access the results in the buffer memory with low latency.

Similarly an extra pipeline stage can be added for transferring the Feature Vectors to the Accelerator, although given that the feature vectors are a much smaller object than the distance results, and that they are not accessed randomly across the interface but delivered once to the Accelerator for each frame, there are fewer situations where the performance benefit from this extra pipeline stage is as significant.

Figures 22 and 23 show idealized systems with the two extra pipeline stages for transfers added. The first shows a system with a CPU, DSP, and Accelerator with the two extra pipeline stages. The second shows a system with a CPU and DSP with the two pipeline stages added.

Other embodiments may add additional pipeline stages so that the distance computations are performed over two or more pipeline stages rather than being restricted to a single pipeline stage. This allows more time for the computation of the distance results for each frame, while maintaining the same throughput of distance results. This is similar in concept to the well-known pipelining techniques used to implement RISC microprocessors.

The Acoustic Models may occupy many megabytes of storage space. In embedded systems they will typically be held in a non-volatile memory such as FLASH memory, or other types of non-volatile memory as is the appropriate currently economic technology. The Accelerator may read the acoustic model directly from the non-volatile memory, or more likely the acoustic model will be copied from the non-volatile memory into a higher bandwidth RAM memory of some type (e.g. SRAM or DRAM, SDRAM, etc.). The RAM memory may be part of the same integrated circuit as the Accelerator (i.e. “on chip”), or attached to the integrated circuit containing the Accelerator. (“off chip”). Having the RAM memory on-chip is particularly advantageous, as this configuration would give the lowest power consumption and highest bandwidth. An alternative solution would be for the acoustic model to be stored on a disk drive, or on another device accessible via a wired or wireless bus or network, and for the acoustic

DSMDB-#3348563-v1

model to be loaded into the RAM memory from the disk drive, or over the wired or wireless bus or network.

On-chip RAM memory is however likely to come at a cost-premium to off-chip commodity memory devices. Many embedded systems face extreme cost pressures in the market place, especially high volume devices supplied by many manufacturers such as mobile phones. For a speech recognizer to be incorporated into such systems, the overall recognizer solution must be delivered at very low cost. There is therefore a significant economic benefit from reducing the cost of the RAM memory and the non-volatile memory. One way to reduce the cost of these memories is to compress the acoustic models and store the compressed model in the memory.

The Accelerator computes all of the distances associated with each Feature Vector that is delivered to it by the front end. This means that the accelerator must use the data from the entire Acoustic Model each time a new Feature Vector is presented to the Accelerator (with the exception of some specific optimizations that may determine that some data values need not be used in the distance computations). It is therefore possible to arrange for the accelerator to read through the acoustic data sequentially once for each Feature Vector, using each data value as it read from the Acoustic Model.

Reading through the entire Acoustic Model sequentially is ideally suited to placing a decompressor between the distance calculation engine and the RAM memory, because many compression schemes depend on the decompressor reading sequentially through the compressed data, and do not readily support random access into the compressed data.

Thus this configuration allows storing compressed data in the RAM memory (and the non-volatile memory), which can therefore be smaller and lower cost. There is no requirement for a memory that is large enough to hold the entire decompressed acoustic model, as the decompressed data is consumed as it is decompressed.

The decompression scheme could be any combination of:

DSMDB-#3348563-v1

- sign or zero extension or otherwise conversion of narrow or variable width data to a wider data format.
- sign or zero extension or otherwise conversion of narrow or variable width data to IEEE standard single or double precision floating point format
- codebook decompression of a binary bitstream, where the codebook is stored as part of the acoustic model data
- decompression of a Huffman or Lempel-Ziv compressed stream
- decompression of run length encoded data
- decompression of difference encoded data
- any well known compression scheme.

Additionally the model size can be reduced by using *subspace distribution clustering* and the decompressor and distance calculation engine designed to operate with such models.

The following description provides further information on the visible registers in some embodiments of the invention.

The register bank is designed to allow simple connection with an 8-bit data bus. Accesses to the register bank should be made with byte-read and byte-write accesses. To facilitate this, only the least significant 8 bits of the data bus are used.

The address decoder should respond to byte accesses to byte 0 or byte 3 within the designated word. This way the same code can be used on an ARM processor regardless of whether the processor is configured for little endian or big endian operation.

REGBASE=Base address of register bank

Register 0 Product Version

8-bit Product version ID

This register value is changed when the functionality of the hardware changes in a way that changes the hardware/software interface.

Address = REGBASE (and REGBASE+3 for big endian)

Read only. Reads with value 0x01

Writes ignored

Register 1 Implementation Version

8-bit Implementation version ID

This register value is changed for each implementation to facilitate bug tracking and version control.

Address = REGBASE+4 (and REGBASE+4+3 for big endian)

Read only. Reads with value 0x01

Writes ignored

Register 2 Interrupt Enable

8-bit Interrupt Enable register

Address = REGBASE+4*2 (and REGBASE+4*2+3 for big endian)

Readable and writeable.

Value following reset is 0 for bits 0,1,2,3

Bit 0 I 1 = Enable FV memory empty interrupt on Int0 output.

Bit 1 J 1 = Enable Results complete interrupt on Int0 output.

Bit 2 K 1 = Enable FV memory empty interrupt on Int1 output.

Bit 3 L 1 = Enable Results complete interrupt on Int1 output.

Bit 4 Future use. Read value is not defined. Writes ignored.

Bit 5 Future use. Read value is not defined. Writes ignored.

Bit 6 Future use. Read value is not defined. Writes ignored.

Bit 7 Future use. Read value is not defined. Writes ignored.

Software should use a read-modify-write code sequence when setting this register so that future use bits are written with the value read from them.

Register 3 Status

Address = REGBASE+4*3 (and REGBASE+4*3+3 for big endian)

Read-only register. Writes are ignored.

Value following reset is 0 for bits 0,1,2,3

Bit 0 FVMB Feature Vector Memory is Busy

0 = OK to write new value to FV memory (FV memory is empty)

1 = Don't write new value to FV memory (FV memory is busy)

Bit 1 CRM Current Result Memory

0 = Result memory A

1 = Result memory B

This bit toggles every time the Release Result Memory (RRM) command is issued to the accelerator (register 3). It indicates which memory software should read results from, and which result memory the RRM command should release next time it is issued. Note: Software should also check that results are available in the indicated result memory by checking the value of the MA and MB bits.

Bit 2 MA 1=Results are available is result memory A

The MA bit is set to 1 when the compute engine completes writing all of the distances for a frame into result memory A. This value is set to zero when the Release Result Memory command is issued and CRM==0.

Bit 3 MB 1= Results are available is result memory B

The MB bit is set to 1 when the compute engine completes writing all of the distances for a frame into result memory B. This value is set to zero when the Release Result Memory command is issued and CRM==1.

Bit 4 Dfault 1=Accoustic data CRC fault. Indicates that the CRC value at the end of the acoustic data did not match the CRC value computed while the acoustic data was read in.

Bit 5 Future use. Value undefined.

Bit 6 Future use. Value undefined.

Bit 7 Future use. Value undefined.

Register 4 Command

Address = REGBASE+4*4 (and REGBASE+4*4+3 for big endian)

Write-only register.

Software should perform byte writes to this register with a single bit set to 1 and all other bits set to zero to issue a command to the accelerator.

Behaviour is not defined if more than one command is issued within a single write (i.e. if more than 1 bit is a 1).

Value on read is not defined.

Bit 0 LFM Load Feature Vector from the FV memory.

Tells the accelerator that a new feature vector has been placed into FV memory ready for use by the accelerator. The FV will be loaded into the compute engine when the compute engine completes its current task (or immediately if the compute engine is idle). Computation of distances for this FV will commence when the FV is loaded AND there is a released result memory. (At reset, both result memories are released). The results for the first, third, fifth etc. feature vectors loaded are written to Result Memory A. The results for the second, fourth, sixth etc. feature vectors loaded are written to Result Memory B.

Bit 1 RRM Release Result memory. Indicates that the results in the current result memory can now be overwritten with new results. The current result memory is indicated by the CRM bit in the status register.

Bit 2 CFVI Clear the FV memory empty interrupt

Bit 3 CRI Clear the Results Complete interrupt

Bit 4 Future use – write 0

Bit 5 Future use – write 0

Bit 6 Future use – write 0

Bit 7 Future use – write 0

The present invention may be included in a mobile phone, or in another consumer electronics device, a PC, Laptop or tablet PC, in-car electronics system, or plug-in card for an electronic device.

The methods and apparatus according to the invention are not limited to embedded mobile systems, but may be used with any type of computing device, including PCs, workstations, laptop computers, notebook computers, handheld computing devices, dictation machines, PDAs (personal digital assistants), personal organisers, mobile phones, games consoles, video players, etc. Embedded computers according to the present invention may be embedded in a wide range of apparatus and machinery, such as electronic entertainment products, domestic appliances, factory machinery, vehicles, etc.

Embodiments of the present invention may be implemented as an IP core in a System on Chip (SOC) or a discrete device. Multiple chips or bare die may be connected to provide a larger vocabulary system.

Although the apparatus may include all or some of an input for accepting audio input, a circuit for digitising the audio input, a circuit for dividing the audio input into a plurality of time-frames, and a circuit for analysing the spectral components of each time frame to characterise the audio in that frame, this is not essential. The apparatus may also include a circuit for calculating a score indicating the probability or likelihood that each audio frame corresponds to each of a plurality of different possible states, but again, this is not essential. Instead, the apparatus may include an input for receiving the pre-calculated scores.

In the described embodiments, n-gram language models are used. For example, the Bigram language model is based on outputting the probability that a given second word directly follows a given first word. However, it is possible to use alternative language models instead. It is possible to use a mixture of different models. For example, it is possible to run a word-dependent n-best decode applying a bigram language model, and the resulting word link record or word graph could then be rescored using a trigram language model.

DSMDB-#3348563-v1

The Viterbi beam search algorithm may be used in embodiments of the invention, to perform the search. A word level search may be performed. Dummy states may be used for the start and end of an utterance

The grammar level model may be used to add distance elements into the likelihood calculation. This is not essential, but tends to give much better results.

The HMM models described in the embodiments comprise self connections to each states and single connections to a neighbouring state. However, the present invention is not limited to such HMM models, and for example, HMM models with additional connections between states may also be used. Alternatively, embodiments of the invention may use alternative types of acoustic models, rather than HMM models.

The system may comprise a plurality of language models (e.g. English, French). A model of words may comprise actual words, phrases, silence, lipsmack sounds, and/or other sounds for recognition by the apparatus.

As previously discussed, the phone level has been found to be the best level for lexical tree searching, rather than the state level or the word level. However, this invention is not limited solely to phone level searching.

Pipelining may comprises processing data in different parts of a circuit at the same time, i.e. parallel processing.

In the following claims, the calculation means may comprise the distance calculation engine described above.

Further embodiments of the invention include the methods carried out in the accelerator, and the methods carried out in the speech recognition circuit.

While the invention has been described in terms of what are at present its preferred embodiments, it will be apparent to those skilled in the art that various changes can be

DSMDB-#3348563-v1

made to the preferred embodiments without departing from the scope of the invention,
which is defined by the claims.

Electronic Patent Application Fee Transmittal				
Application Number:		14788164		
Filing Date:		30-Jun-2015		
Title of Invention:		SPEECH RECOGNITION CIRCUIT AND METHOD		
First Named Inventor/Applicant Name:		Guy Larri		
Filer:		Stephen A. Soffen/Kanessa Howard		
Attorney Docket Number:		M0025.0369/P369-C		
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

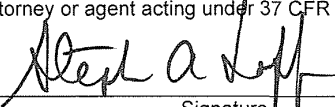
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension - 1 month with \$0 paid	2251	1	100	100
Miscellaneous:				
Total in USD (\$)				100

Electronic Acknowledgement Receipt	
EFS ID:	23610097
Application Number:	14788164
International Application Number:	
Confirmation Number:	1250
Title of Invention:	SPEECH RECOGNITION CIRCUIT AND METHOD
First Named Inventor/Applicant Name:	Guy Larri
Customer Number:	24998
Filer:	Stephen A. Soffen/Kanessa Howard
Filer Authorized By:	Stephen A. Soffen
Attorney Docket Number:	M0025.0369/P369-C
Receipt Date:	25-SEP-2015
Filing Date:	30-JUN-2015
Time Stamp:	16:34:51
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	\$ 100
RAM confirmation Number	3634
Deposit Account	041073
Authorized User	SOFFEN, STEPHEN A.
<p>The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:</p> <p>Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)</p> <p>Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)</p>	

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)					
Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)					
Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)					
File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		M0025_0369_C__Resp_Notice_of_Incomplete_Reply.pdf	6154139 2afedd6e82c51ad5e324c8d5214538f42e87512c	yes	131
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Extension of Time		1	1	
	Applicant Response to Pre-Exam Formalities Notice		2	2	
	Specification		3	67	
	Specification		68	131	
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30636 dde1fc03d1e64c695288222781c3d27f82c4ee4a	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			6184775		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a)		Docket Number (Optional) M0025.0369/P369-C			
Application Number	14/788,164-Conf. #1250	Filed	June 30, 2015		
For A SPEECH RECOGNITION CIRCUIT AND METHOD					
Art Unit	N/A	Examiner	Not Yet Assigned		
This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above-identified application. The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):					
		Fee	Small Entity Fee	Micro Entity Fee	
☒	One month (37 CFR 1.17(a)(1))	\$200	\$100	\$50	\$ 100.00
☐	Two months (37 CFR 1.17(a)(2))	\$600	\$300	\$150	\$
☐	Three months (37 CFR 1.17(a)(3))	\$1,400	\$700	\$350	\$
☐	Four months (37 CFR 1.17(a)(4))	\$2,200	\$1,100	\$550	\$
☐	Five months (37 CFR 1.17(a)(5))	\$3,000	\$1,500	\$750	\$
☒	Applicant asserts small entity status. See 37 CFR 1.27.				
☐	Applicant certifies micro entity status. See 37 CFR 1.29. Form PTO/SB/15A or B or equivalent must either be enclosed or have been submitted previously.				
☐	A check in the amount of the fee is enclosed.				
☒	Payment by credit card. Form PTO-2038 is attached.				
☐	The Director has already been authorized to charge fees in this application to a Deposit Account.				
☒	The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 04-1073				
☐	Payment made via EFS-Web.				
WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.					
I am the					
☐	applicant.				
☒	attorney or agent of record. Registration number 31,063				
☐	attorney or agent acting under 37 CFR 1.34. Registration number				
		September 25, 2015			
Signature		Date			
Stephen A. Soffen		(202) 420-4879			
Typed or printed name		Telephone Number			
NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. Submit multiple forms if more than one signature is required, see below*.					
☐	* Total of 1 forms are submitted.				

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875						Application or Docket Number 14/788,164		
APPLICATION AS FILED - PART I								
(Column 1)		(Column 2)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY		
FOR	NUMBER FILED	NUMBER EXTRA	RATE(\$)	FEE(\$)		RATE(\$)	FEE(\$)	
BASIC FEE <small>(37 CFR 1.16(a), (b), or (c))</small>	N/A	N/A	N/A	70		N/A		
SEARCH FEE <small>(37 CFR 1.16(k), (l), or (m))</small>	N/A	N/A	N/A	300		N/A		
EXAMINATION FEE <small>(37 CFR 1.16(o), (p), or (q))</small>	N/A	N/A	N/A	360		N/A		
TOTAL CLAIMS <small>(37 CFR 1.16(j))</small>	1	minus 20 = *	x 40 =	0.00	OR			
INDEPENDENT CLAIMS <small>(37 CFR 1.16(h))</small>	1	minus 3 = *	x 210 =	0.00				
APPLICATION SIZE FEE <small>(37 CFR 1.16(s))</small>	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			0.00				
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				0.00				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	730		TOTAL		
APPLICATION AS AMENDED - PART II								
AMENDMENT A	(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY	OR OTHER THAN SMALL ENTITY
		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA		RATE(\$)	ADDITIONAL FEE(\$)
	Total <small>(37 CFR 1.16(i))</small>	*	Minus	**	=	x	=	OR x
	Independent <small>(37 CFR 1.16(h))</small>	*	Minus	***	=	x	=	OR x
	Application Size Fee (37 CFR 1.16(s))							OR
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR
					TOTAL ADD'L FEE		OR TOTAL ADD'L FEE	
AMENDMENT B	(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY	OR OTHER THAN SMALL ENTITY
		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA		RATE(\$)	ADDITIONAL FEE(\$)
	Total <small>(37 CFR 1.16(i))</small>	*	Minus	**	=	x	=	OR x
	Independent <small>(37 CFR 1.16(h))</small>	*	Minus	***	=	x	=	OR x
	Application Size Fee (37 CFR 1.16(s))							OR
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR
					TOTAL ADD'L FEE		OR TOTAL ADD'L FEE	
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3. ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20". *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3". The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.								



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO.	TOT CLAIMS	IND CLAIMS
14/788,164	06/30/2015		730	M0025.0369/P369-C	1	1

CONFIRMATION NO. 1250

UPDATED FILING RECEIPT



CC000000077818956

24998
DICKSTEIN SHAPIRO LLP
1825 EYE STREET NW
Washington, DC 20006-5403

Date Mailed: 10/02/2015

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Guy Larri, Cambridge, UNITED KINGDOM;
Mark Catchpole, Prickwillow, UNITED KINGDOM;
Damian Kelly Harris-Dowsett, Coventry, UNITED KINGDOM;
Timothy Brian Reynolds, Cambridge, UNITED KINGDOM;

Applicant(s)

Zentian Limited, Cambridge, UNITED KINGDOM;

Power of Attorney: The patent practitioners associated with Customer Number 24998

Domestic Priority data as claimed by applicant

This application is a CON of 13/735,091 01/07/2013 PAT 9076441
which is a CON of 13/162,128 06/16/2011 PAT 8352262
which is a DIV of 11/662,704 03/14/2007 PAT 7979277
which is a 371 of PCT/GB2005/003554 09/14/2005

Foreign Applications (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.)
UNITED KINGDOM 0420464.0 09/14/2004 No Access Code Provided

If Required, Foreign Filing License Granted: 07/14/2015

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 14/788,164**

Projected Publication Date: 01/07/2016

Non-Publication Request: No

Early Publication Request: No

**** SMALL ENTITY ****

Title

SPEECH RECOGNITION CIRCUIT AND METHOD

Preliminary Class

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER
Title 35, United States Code, Section 184
Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit <http://www.SelectUSA.gov> or call +1-202-482-6800.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
14/788,164	06/30/2015	Guy Larri	M0025.0369/P369-C

CONFIRMATION NO. 1250

PUBLICATION NOTICE



OC000000079762670*

24998
DICKSTEIN SHAPIRO LLP
1825 EYE STREET NW
Washington, DC 20006-5403

Title:SPEECH RECOGNITION CIRCUIT AND METHOD

Publication No.US-2016-0005397-A1

Publication Date:01/07/2016

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publicly available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently <http://www.uspto.gov/patft/>.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Office of Public Records. The Office of Public Records can be reached by telephone at (703) 308-9726 or (800) 972-6382, by facsimile at (703) 305-8759, by mail addressed to the United States Patent and Trademark Office, Office of Public Records, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently <http://pair.uspto.gov/>. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/788,164	06/30/2015	Guy Larri	M0025.0369/P369-C	1250
24998	7590	09/30/2016		
Blank Rome LLP 1825 EYE STREET NW Washington, DC 20006-5403			EXAMINER MCFADDEN, SUSAN IRIS	
			ART UNIT 2658	PAPER NUMBER
			NOTIFICATION DATE 09/30/2016	DELIVERY MODE ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

#IPDocketing-DC@BlankRome.com

Office Action Summary	Application No. 14/788,164	Applicant(s) LARRI ET AL.	
	Examiner SUSAN MCFADDEN	Art Unit 2658	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 6-30-15.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on ____.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on ____; the restriction requirement and election have been incorporated into this action.
- 4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

- 5) ☒ Claim(s) 1 is/are pending in the application.
5a) Of the above claim(s) ____ is/are withdrawn from consideration.
- 6) ☐ Claim(s) ____ is/are allowed.
- 7) ☒ Claim(s) 1 is/are rejected.
- 8) ☐ Claim(s) ____ is/are objected to.
- 9) ☐ Claim(s) ____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

- 10) ☒ The specification is objected to by the Examiner.
- 11) ☒ The drawing(s) filed on 6-30-15 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

- a) ☒ All b) ☐ Some** c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
2. ☒ Certified copies of the priority documents have been received in Application No. 11/662,704.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) ☐ Notice of References Cited (PTO-892) | 3) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. ____. |
| 2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b)
Paper No(s)/Mail Date ____. | 4) ☐ Other: ____. |

DETAILED ACTION

1. The present application is being examined under the pre-AIA first to invent provisions.

Specification

1. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Double Patenting

1. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

2. Claim 1 is rejected on the ground of nonstatutory double patenting as being unpatentable over claim 1 of U.S. Patent No. 7,279,277. Although the claims at issue are not identical, they are not patentably distinct from each other because they both claim a speech recognition circuit which includes and audio front end, a calculating circuit, a search stage using lexical trees.

Claim 1 is similar to claim 1 of U.S. Patent 7279277.

Allowable Subject Matter

Claim 1 would be allowed if a terminal disclaimer is submitted.

The following is a statement of reasons for the indication of allowable subject matter: The prior art of record, Larri et al (9,076,441 and (8,352,262) show it is well known to have speech recognition circuits with calculating circuits including identifying words by using lexical trees. In regard to claim 1, the prior art of record does not show or suggest the combination of an audio front end, the calculating circuit, a search stage,

wherein the audio front end and search stage are implemented using a first processor, and a calculating circuit is implemented using a second processor, and wherein data is pipelined from the front end to the calculating circuit to the search stage.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SUSAN MCFADDEN whose telephone number is (571)272-7621. The examiner can normally be reached on Monday-Friday, 9:30-6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Richemond Dorvil can be reached on 571-272-7602. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/SUSAN MCFADDEN/
Primary Examiner, Art Unit 2658
September 26, 2016

Search Notes 	Application/Control No. 14788164	Applicant(s)/Patent Under Reexamination LARRI ET AL.
	Examiner SUSAN MCFADDEN	Art Unit 2658

CPC- SEARCHED		
Symbol	Date	Examiner
G10L 15/10	9-26-16	SM

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
704	238,251	9-26-16	SM

SEARCH NOTES		
Search Notes	Date	Examiner
Searched WEST	9-26-16	SM

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

--	--

Substitute for form 1449/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Use as many sheets as necessary)</i>				Complete if Known	
				Application Number	Not yet assigned 14788164 - GAU: 2658
				Filing Date	Concurrently Herewith
				First Named Inventor	Guy Larri
				Art Unit	Not yet assigned
				Examiner Name	Not yet assigned
Sheet	1	of	1	Attorney Docket Number	M0025.0369-C

U.S. PATENT DOCUMENTS					
Examiner Initials*	Cite No. ¹	Document Number	Publication Date MM-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number-Kind Code ² (if known)			
		5,881,312	03-1999	Dulong	
		7,328,153	02-2008	Wells et al.	
		2004/0111264 A1	06-2004	Wang et al.	
		2003/0061046 A1	03-2003	Zhao et al.	
		6,374,220	04-2002	Kao	
		9,076,441	07-2015	Larri et al.	
		8,352,262	01-2013	Larri et al.	
		7,979,277	07-2011	Larri et al.	
		8,036,890	10-2011	Catchpole	
		8,612,227	12-2013	Kato	

FOREIGN PATENT DOCUMENTS						
Examiner Initials*	Cite No. ¹	Foreign Patent Document	Publication Date MM-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear	T ⁶
		Country Code ³ -Number ⁴ -Kind Code ⁵ (if known)				
		GB 2 391 679 A	02-2004			
		EP 1 178 466 A2	02-2002			
		WO 01/75862 A2	10-2001			
		WO 01/48737 A2	07-2001			
		GB 2 333 172 A	07-1999			
		WO 03/067572 A2	08-2003			

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. ¹ Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. ³ Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). ⁴ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁵ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁶ Applicant is to place a check mark here if English language Translation is attached.

NON PATENT LITERATURE DOCUMENTS				
Examiner Initials	Cite No. ¹	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ²	
		Glinski et al., "Spoken Language Recognition on DSP Array Processor" IEEE Transactions on Parallel and Distributed Systems, Vol. 5, No. 7, pp. 697-703, July 1, 1994.		
		S. Chatterjee et al., "Connected Speech Recognition on a Multiple Processor Pipeline." ICASSP 89, pp. 774-777, May 23, 1989.		

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

Examiner Signature	/Susan McFadden/	Date Considered	09/26/2016
--------------------	------------------	-----------------	------------

DSMDB-3348573 v1

<p align="center"><i>Index of Claims</i></p> 	Application/Control No. 14788164	Applicant(s)/Patent Under Reexamination LARRI ET AL.
	Examiner SUSAN MCFADDEN	Art Unit 2658

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

☐ Claims renumbered in the same order as presented by applicant										☐ CPA										☐ T.D.										☐ R.1.47									
CLAIM										DATE																													
Final	Original	09/26/2016																																					
	1	✓																																					

WEST Search History for Application 14788164

Creation Date: 2016092617:33

Prior Art Searches

Query	DB	Hits	Op.	Plur.	Thes.	Date
larri.in.	USPT	26	OR	YES		09-19-2016
larri.in. and audio and calculating	USPT	6	OR	YES		09-19-2016
larri.in. and audio and calculating near5 distances	USPT	3	OR	YES		09-19-2016
larri.in. and audio and calculating near5 distances and similarity	USPT	3	OR	YES		09-19-2016
larri.in. and audio and calculating near5 distances and similarity and tree	USPT	3	OR	YES		09-19-2016
larri.in. and audio near front and calculating near5 distances and similarity and tree	USPT	3	OR	YES		09-19-2016
larri.in. and audio near front near end and calculating near5 distances and similarity and tree	USPT	3	OR	YES		09-19-2016
larri.in.	USPT	26	OR	YES		09-26-2016
larri.in. and audio near front	USPT	3	OR	YES		09-26-2016
larri.in. and audio near front and calculating	USPT	3	OR	YES		09-26-2016
larri.in. and audio near front and calculating and search	USPT	3	OR	YES		09-26-2016
larri.in. and audio near front and calculating and search and distance	USPT	3	OR	YES		09-26-2016
larri.in. and audio near front and calculating and search and distance and lexical	USPT	3	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical	USPT	3	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical near tree	USPT	3	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical near tree and word	USPT	3	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical near tree and word near identifying	USPT	0	OR	YES		09-26-2016

audio near front and calculating and search and distance and lexical near tree and identifying near4 word	USPT	0	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical near tree and identify near4 word	USPT	3	OR	YES		09-26-2016



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
 United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 1250

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.		
14/788,164	06/30/2015	704	2658	M0025.0369/P369-C		
APPLICANTS Zentian Limited, Cambridge, UNITED KINGDOM; INVENTORS Guy Larri, Cambridge, UNITED KINGDOM; Mark Catchpole, Prickwillow, UNITED KINGDOM; Damian Kelly Harris-Dowsett, Coventry, UNITED KINGDOM; Timothy Brian Reynolds, Cambridge, UNITED KINGDOM; ** CONTINUING DATA ***** This application is a CON of 13/735,091 01/07/2013 PAT 9076441 which is a CON of 13/162,128 06/16/2011 PAT 8352262 which is a DIV of 11/662,704 03/14/2007 PAT 7979277 which is a 371 of PCT/GB2005/003554 09/14/2005 ** FOREIGN APPLICATIONS ***** UNITED KINGDOM 0420464.0 09/14/2004 ** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** * SMALL ENTITY ** 07/14/2015						
Foreign Priority claimed ☒ Yes ☐ No 35 USC 119(a-d) conditions met ☒ Yes ☐ No Verified and /SUSAN IRIS Acknowledged MCFADDEN/ Examiner's Signature		☐ Met after Allowance Initials	STATE OR COUNTRY UNITED KINGDOM	SHEETS DRAWINGS 27	TOTAL CLAIMS 1	INDEPENDENT CLAIMS 1
ADDRESS Blank Rome LLP 1825 EYE STREET NW Washington, DC 20006-5403 UNITED STATES						
TITLE SPEECH RECOGNITION CIRCUIT AND METHOD						
FILING FEE RECEIVED 730	FEES: Authority has been given in Paper No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:			☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit		

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Guy Larri et al.

Application No.: 14/788,164

Confirmation No.: 1250

Filed: June 30, 2015

Art Unit: 2658

For: SPEECH RECOGNITION CIRCUIT WITH
ACOUSTIC COPROCESSOR (AS
AMENDED)

Examiner: S. I. McFadden

AMENDMENT IN RESPONSE TO NON-FINAL OFFICE ACTION

MS Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Madam:

INTRODUCTORY COMMENTS

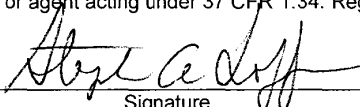
In response to the Office Action dated September 30, 2016, please amend the above-identified U.S. patent application as follows:

Amendments to the Title begin on page 2 of this paper.

Amendments to the Claims are reflected in the listing of claims which begins on page 3 of this paper.

Remarks/Arguments begin on page 7 of this paper.

200494.00239/105170228

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a)		Docket Number (Optional) M0025.0369/P369-C	
Application Number 14/788,164-Conf. #1250		Filed June 30, 2015	
For A SPEECH RECOGNITION CIRCUIT AND METHOD			
Art Unit 2658		Examiner S. I. McFadden	
This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above-identified application.			
The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):			
	<u>Fee</u>	<u>Small Entity Fee</u>	<u>Micro Entity Fee</u>
☐ One month (37 CFR 1.17(a)(1))	\$200	\$100	\$50
☒ Two months (37 CFR 1.17(a)(2))	\$600	\$300	\$150
☐ Three months (37 CFR 1.17(a)(3))	\$1,400	\$700	\$350
☐ Four months (37 CFR 1.17(a)(4))	\$2,200	\$1,100	\$550
☐ Five months (37 CFR 1.17(a)(5))	\$3,000	\$1,500	\$750
☒ Applicant asserts small entity status. See 37 CFR 1.27. ☐ Applicant certifies micro entity status. See 37 CFR 1.29. Form PTO/SB/15A or B or equivalent must either be enclosed or have been submitted previously. ☐ A check in the amount of the fee is enclosed. ☒ Payment by credit card. Form PTO-2038 is attached. ☐ The Director has already been authorized to charge fees in this application to a Deposit Account. ☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number <u>23-2185</u> . ☐ Payment made via EFS-Web.			
WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.			
I am the			
☐ applicant.			
☒ attorney or agent of record. Registration number <u>31,063</u> .			
☐ attorney or agent acting under 37 CFR 1.34. Registration number _____.			
 _____ Signature		<u>February 28, 2017</u> _____ Date	
<u>Stephen A. Soffen</u> _____ Typed or printed name		<u>(202) 420-4879</u> _____ Telephone Number	
NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. Submit multiple forms if more than one signature is required, see below*.			
☐ * Total of <u>1</u> forms are submitted.			

Electronic Patent Application Fee Transmittal				
Application Number:		14788164		
Filing Date:		30-Jun-2015		
Title of Invention:		SPEECH RECOGNITION CIRCUIT AND METHOD		
First Named Inventor/Applicant Name:		Guy Larri		
Filer:		Stephen A. Soffen/Cheryl de Lopez		
Attorney Docket Number:		M0025.0369/P369-C		
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension - 2 months with \$0 paid	2252	1	300	300
Miscellaneous:				
Total in USD (\$)				300

Electronic Acknowledgement Receipt	
EFS ID:	28485875
Application Number:	14788164
International Application Number:	
Confirmation Number:	1250
Title of Invention:	SPEECH RECOGNITION CIRCUIT AND METHOD
First Named Inventor/Applicant Name:	Guy Larri
Customer Number:	24998
Filer:	Stephen A. Soffen/Cheryl de Lopez
Filer Authorized By:	Stephen A. Soffen
Attorney Docket Number:	M0025.0369/P369-C
Receipt Date:	28-FEB-2017
Filing Date:	30-JUN-2015
Time Stamp:	14:37:59
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$ 300
RAM confirmation Number	030117INTEFSW14383700
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		amendp369c.pdf	225471	yes	8
			d6bd27fe9dc5963debcaf910b20a7b5a52a664b9		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Applicant Arguments/Remarks Made in an Amendment		8	8	
	Claims		4	7	
	Specification		3	3	
	Amendment/Req. Reconsideration-After Non-Final Reject		2	2	
	Extension of Time		1	1	
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30643	no	2
			d49c23eef534a31c8d199ecf1cba1a3e706eda59		
Warnings:					
Information:					
Total Files Size (in bytes):			256114		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

REMARKS

Claims 2-17 are pending in the present application. Claim 1 has been canceled. Claims 2-17 have been added.

The title has been amended to be more descriptive of the invention to which the claims are directed.

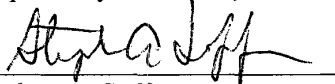
Claim 1 is rejected on grounds of obviousness-type double patenting as obvious over claim 1 of U.S. Patent No. 7,279,277. In view of the cancellation of claim 1, this rejection is moot.

New claims 2-17 are submitted to be novel and non-obvious over the prior art. In view of the above amendment, applicant believes the pending application is in condition for allowance.

The Director is hereby authorized to charge any deficiency or credit any overpayment in the fees filed, asserted to be filed or which should have been filed herewith (or with any paper hereafter filed in this application by this firm) to our Deposit Account No. 23-2185, under Order No. M0025.0496.

Dated: February 28, 2017

Respectfully submitted,

By 

Stephen A. Soffen

Registration No.: 31,063

BLANK ROME LLP

1825 Eye Street, NW

Washington, DC 20006-5403

(202) 420-2200

Attorney for Applicant

AMENDMENTS TO THE CLAIMS

1. (Canceled)
2. (New) An acoustic coprocessor, comprising:
an interface for receiving at least one feature vector;
a calculating apparatus for calculating distances indicating the similarity between said at least one feature vector and at least one acoustic state of an acoustic model read from an acoustic model memory; and
an interface for sending at least one distance calculated by the calculating apparatus;
wherein said calculating apparatus and said acoustic model memory are fabricated on a single integrated circuit.
3. (New) The acoustic coprocessor of claim 2 wherein said acoustic model contains Gaussian parameters and the distances are probabilities.
4. (New) The acoustic coprocessor of claim 3 wherein said acoustic model memory is a FLASH memory.
5. (New) The acoustic coprocessor of claim 4 wherein said interface for sending distances is a serial bus.
6. (New) The acoustic coprocessor of claim 5 wherein the distances sent over the serial bus are Mahalanobis Distances.

7. (New) The acoustic coprocessor of claim 5 wherein the acoustic coprocessor in response to receiving a feature vector autonomously calculates said distance for every acoustic state of said acoustic model.

8. (New) The acoustic coprocessor of claim 7 further comprising one or more result memories and wherein said distance for every acoustic state of said acoustic model are stored in said one or more result memories.

9. (New) The acoustic coprocessor of claim 8 wherein said one or more result memories are configured so that distances calculated from a first feature vector may be sent over the interface for sending while at least one distance calculated from a second feature vector is being stored in the said one or more result memories.

10. (New) A speech recognition system comprising the acoustic coprocessor of claim 9 and further comprising at least one CPU wherein said CPU executes software which performs or initiates a speech recognition word search using one or more distances calculated by said acoustic coprocessor in response to receiving a first feature vector and wherein the time period taken for said speech recognition word search overlaps the time period taken for said acoustic coprocessor to calculate one or more distances in response to receiving a second feature vector.

11. (New) A speech recognition system comprising:

a first programmable device programmed to calculate a feature vector from a digital audio stream, wherein the feature vector comprises a plurality of extracted and/or derived quantities from said digital audio stream during a defined audio time frame;

a second programmable device programmed to calculate distances indicating the similarity between a feature vector and a plurality of acoustic states of an acoustic model wherein said feature

vector is received by the second programmable device after it is calculated by the first programmable device;

a third programmable device programmed to identify spoken words in said digital audio stream using Hidden Markov Models and/or Neural Networks wherein said word identification uses one or more distances that were calculated by the second programmable device;

wherein said identification of spoken words uses one or more distances calculated from a first feature vector.

12. (New) The speech recognition system of claim 11 wherein the time period taken for said identification of spoken words overlaps the time period taken for said second programmable device to calculate one or more distances from a second feature vector.

13. (New) The speech recognition system of claim 12 wherein the first programmable device, the second programmable device, and third programmable device are physically distinct processing circuits.

14. (New) The speech recognition system of claim 13 wherein said feature vectors and said distances derived from a sequence of audio time frames of the digital audio stream flow in a pipelined manner from said first programmable device to said second programmable device, and from said second programmable device to said third programmable device respectively.

15. (New) The speech recognition system of claim 12 wherein the first programmable device and the second programmable device comprise one physical uni-processor or multi-processor, and the second programmable device is a physically distinct processing circuit.

16. (New) The speech recognition system of claim 13 wherein said physically distinct processing circuits are all formed on a single integrated circuit.

17. (New) The speech recognition system of claim 15 wherein said one physical uni-processor or multi-processor and said second programmable device are all formed on a single integrated circuit.

AMENDMENTS TO THE TITLE

Please amend the title as follows:

[[A]] SPEECH RECOGNITION CIRCUIT WITH ACOUSTIC CO-PROCESSOR AND
~~METHOD~~

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875				Application or Docket Number 14/788,164		Filing Date 06/30/2015		☐ To be Mailed	
ENTITY: ☐ LARGE ☒ SMALL ☐ MICRO									
APPLICATION AS FILED – PART I									
(Column 1)		(Column 2)							
FOR	NUMBER FILED	NUMBER EXTRA				RATE (\$)	FEE (\$)		
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A				N/A			
☐ SEARCH FEE (37 CFR 1.16(k), (j), or (m))	N/A	N/A				N/A			
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A				N/A			
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =		*				X \$	=	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =		*				X \$	=	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).								
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))									
* If the difference in column 1 is less than zero, enter "0" in column 2.						TOTAL			
APPLICATION AS AMENDED – PART II									
(Column 1)		(Column 2)		(Column 3)					
AMENDMENT	02/28/2017	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)		ADDITIONAL FEE (\$)	
	Total (37 CFR 1.16(i))	* 16	Minus	** 20	= 0	X \$40 =		0	
	Independent (37 CFR 1.16(h))	* 2	Minus	***3	= 0	X \$210 =		0	
	☐ Application Size Fee (37 CFR 1.16(s))								
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
						TOTAL ADD'L FEE		0	
(Column 1)		(Column 2)		(Column 3)					
AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)		ADDITIONAL FEE (\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	=	X \$			
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X \$			
	☐ Application Size Fee (37 CFR 1.16(s))								
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
						TOTAL ADD'L FEE			
<p>* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.</p> <p>** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".</p> <p>*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".</p> <p>The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.</p>									

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/788,164	06/30/2015	Guy Larri	M0025.0369/P369-C	1250
24998 7590 05/09/2017 Blank Rome LLP 1825 EYE STREET NW Washington, DC 20006-5403			EXAMINER MCFADDEN, SUSAN IRIS	
			ART UNIT 2658	PAPER NUMBER
			NOTIFICATION DATE 05/09/2017	DELIVERY MODE ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

#IPDocketing-DC@BlankRome.com

Office Action Summary	Application No. 14/788,164	Applicant(s) LARRI ET AL.	
	Examiner SUSAN MCFADDEN	Art Unit 2658	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 2 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 2-28-17.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on ____.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on ____; the restriction requirement and election have been incorporated into this action.
- 4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

- 5) ☒ Claim(s) 2-17 is/are pending in the application.
5a) Of the above claim(s) ____ is/are withdrawn from consideration.
- 6) ☐ Claim(s) ____ is/are allowed.
- 7) ☐ Claim(s) ____ is/are rejected.
- 8) ☐ Claim(s) ____ is/are objected to.
- 9) ☒ Claim(s) 2-17 are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

- 10) ☐ The specification is objected to by the Examiner.
- 11) ☐ The drawing(s) filed on ____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

- a) ☐ All b) ☐ Some** c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. ____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) ☐ Notice of References Cited (PTO-892) | 3) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. ____. |
| 2) ☐ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b)
Paper No(s)/Mail Date ____. | 4) ☐ Other: ____. |

RESTRICTION

The present application is being examined under the pre-AIA first to invent provisions

Election/Restrictions

1. Restriction to one of the following inventions is required under 35 U.S.C. 121:
 - I. Claims 2-10, drawn to searching circuits, classified in G10L 15/08.
 - II. Claims 11-17, drawn to a speech recognition system using HMMs, classified in G10L 15/142.

The inventions are distinct, each from the other because of the following reasons:

2. Inventions I and II are unrelated. Inventions are unrelated if it can be shown that they are not disclosed as capable of use together and they have different designs, modes of operation, and effects (MPEP § 802.01 and § 806.06). In the instant case, the different inventions Group 2 is not required for Group 1.
3. Restriction for examination purposes as indicated is proper because all the inventions listed in this action are independent or distinct for the reasons given above and there would be a serious search and/or examination burden if restriction were not required because one or more of the following reasons apply:

They have different classifications.

Applicant is advised that the reply to this requirement to be complete must include (i) an election of a invention to be examined even though the requirement may be traversed (37 CFR 1.143) and (ii) identification of the claims encompassing the elected invention.

The election of an invention may be made with or without traverse. To reserve a right to petition, the election must be made with traverse. If the reply does not distinctly and specifically point out supposed errors in the restriction requirement, the election shall be treated as an election without traverse. Traversal must be presented at the time of election in order to be considered timely. Failure to timely traverse the requirement will result in the loss of right to petition under 37 CFR 1.144. If claims are added after the election, applicant must indicate which of these claims are readable upon the elected invention.

Should applicant traverse on the ground that the inventions are not patentably distinct, applicant should submit evidence or identify such evidence now of record showing the inventions to be obvious variants or clearly admit on the record that this is the case. In either instance, if the examiner finds one of the inventions unpatentable over the prior art, the evidence or admission may be used in a rejection under 35 U.S.C. 103 or pre-AIA 35 U.S.C. 103(a) of the other invention.

4. A telephone call was made to Stephen Soffen on May 1, 2017 to request an oral election to the above restriction requirement, but did not result in an election being made.

5. Applicant is reminded that upon the cancellation of claims to a non-elected invention, the inventorship must be corrected in compliance with 37 CFR 1.48(a) if one or more of the currently named inventors is no longer an inventor of at least one claim remaining in the application. A request to correct inventorship under 37 CFR 1.48(a) must be accompanied by an application data sheet in accordance with 37 CFR 1.76 that

identifies each inventor by his or her legal name and by the processing fee required under 37 CFR 1.17(i).

Claim Objections

6. Claims 2-17 are further objected to because of the following informalities: “acoustic coprocessor”, “first, second, and third programmable device” are not found in the Specification. Applicant is advised to show where in the Specification support for the new claims are. Appropriate correction is required.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SUSAN MCFADDEN whose telephone number is (571)272-7621. The examiner can normally be reached on Monday-Friday, 9:30-6:00.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Richemond Dorvil can be reached on 571-272-7602. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/SUSAN MCFADDEN/
Primary Examiner, Art Unit 2658
May 3, 2017

Search Notes 	Application/Control No. 14788164	Applicant(s)/Patent Under Reexamination LARRI ET AL.
	Examiner SUSAN MCFADDEN	Art Unit 2658

CPC- SEARCHED		
Symbol	Date	Examiner
G10L 15/10	9-26-16	SM

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
704	238,251	9-26-16	SM

SEARCH NOTES		
Search Notes	Date	Examiner
Searched WEST	9-26-16	SM
Consulted with SPE Rich Dorvil about cancellation of claims and new claims added	5-1-17	SM

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

--	--

<i>Index of Claims</i> 	Application/Control No. 14788164	Applicant(s)/Patent Under Reexamination LARRI ET AL.
	Examiner SUSAN MCFADDEN	Art Unit 2658

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

☐ Claims renumbered in the same order as presented by applicant										☐ CPA										☐ T.D.										☐ R.1.47									
CLAIM		DATE																																					
Final	Original	09/26/2016	05/01/2017																																				
	1	✓	-																																				
	2		÷																																				
	3		÷																																				
	4		÷																																				
	5		÷																																				
	6		÷																																				
	7		÷																																				
	8		÷																																				
	9		÷																																				
	10		÷																																				
	11		÷																																				
	12		÷																																				
	13		÷																																				
	14		÷																																				
	15		÷																																				
	16		÷																																				
	17		÷																																				

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Guy Larri et al.

Application No.: 14/788,164

Confirmation No.: 1250

Filed: June 30, 2015

Art Unit: 2658

For: SPEECH RECOGNITION CIRCUIT WITH
ACOUSTIC COPROCESSOR (AS AMENDED)

Examiner: S. I. McFadden

RESPONSE TO RESTRICTION REQUIREMENT

MS Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Restriction Requirement set forth in the Office Action mailed May 9, 2017 (Paper No. 20170501), applicants hereby elects Group II, claims 11-17, for continued examination, without traverse.

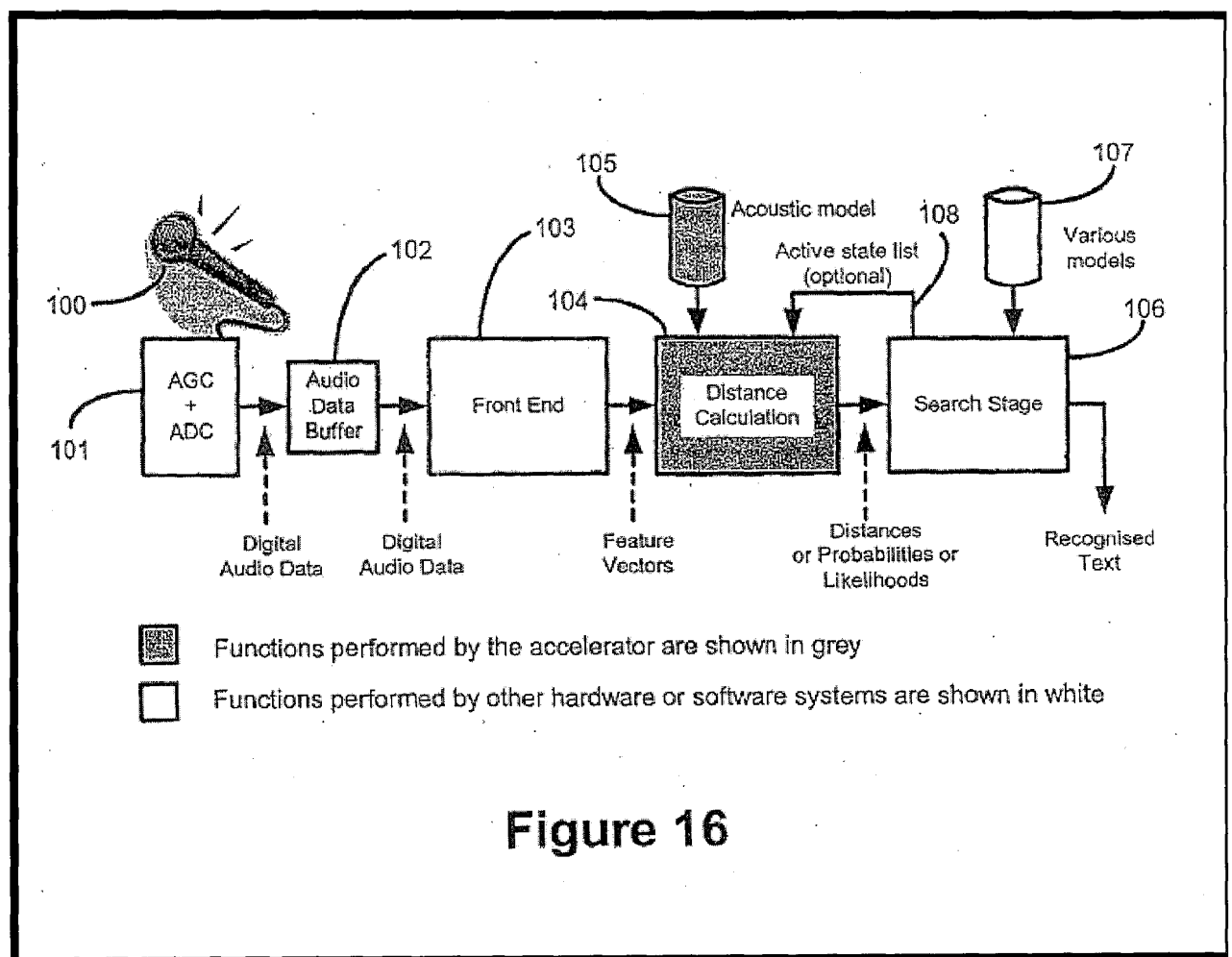
Claims 2-17 are objected to on the ground that the term "acoustic coprocessor," and the "first, second and third programmable device" are not found in the specification. The Examiner advises applicant to identify the support in the specification for these terms.

In response to the objection that "acoustic coprocessor" is not found in the specification, applicant submits that this objection is now moot as the term appears in claims 2-10, that are now withdrawn from consideration (and will be canceled) by election of the second group, claims 11-17. Applicant reserves the right to address the objection, should claims 2-10 be presented in a future continuation application or divisional application.

200494.00239/106036222

In response to the objection that "first, second, and third programmable device" are not found in the specification, the applicant indicates below the portions of the specification that support the use in the claims 11-17 of "a first programmable device," "a second programmable device," and "a third programmable device."

Fig. 16 shows the three stages "Front End", "Distance Calculation", and "Search Stage":



Paragraph [0024] of the published application states: "The distance calculation engine may be included within an accelerator" ... "The accelerator may be implemented in software or in hardware, or in a combination."

Paragraph [0034] of the published application states: "The distance engine may be implemented in hardware, software, or a combination. Other stages may be implemented in hardware, software, or a combination."

The above quotes from paragraphs [0024] and [0034] of the published application which indicate that the "distance engine may be included within an accelerator" and "The accelerator may be implemented in software or in hardware, or in a combination" clearly indicate that the accelerator may include a programmable engine.

Paragraph [0035] of the published application states: "In particular, the other stages may be implemented on a CPU, or on a DSP and CPU. The "DSP" and "CPU" may each be implemented as software programmable devices."

Note that the first part of the sentence quoted above in paragraph [0035], "the other stages may be implemented on a CPU," supports dependent claim 15, while the later part of the sentence "or on a DSP and CPU", in combination with the statement quoted from paragraph [0034] above, supports the recitation in independent claim 11 of three separate programmable devices for the three stages.

See also the headings of Fig. 22 which include "Front end on DSP", "Distance Calculation Accelerator", and "Search stage on CPU". Given that a DSP is a well-known programmable device, and a CPU is a well-known programmable device, and the quotes from the specification given above indicate that the "distance calculation engine may be included within an accelerator" and "the accelerator may be implemented in software or in hardware, or in a combination," it is clear that

each of "Front end on DSP", "Distance Calculation Accelerator", and "Search stage on CPU" can be (in some embodiments) three different programmable devices.

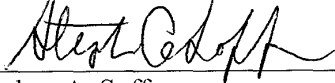
The headings in Figure 23 of the present application support claim 15, in particular "Front end and search stage on CPU," which is describing an embodiment when those two stages are implemented on a single programmable device.

See also paragraph [0195] of the published application: "The system may be implemented in any combination of hardware or software."

As demonstrated above, the specification and drawings of the present application support the terms "a first programmable device," "a second programmable device," and "a third programmable device."

Dated: August 9, 2017

Respectfully submitted,

By 

Stephen A. Soffen

Registration No.: 31,063

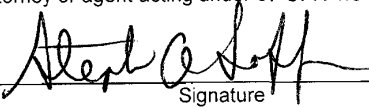
BLANK ROME LLP

1825 Eye Street, NW

Washington, DC 20006-5403

(202) 420-2200

Attorney for Applicants

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a)		Docket Number (Optional) M0025.0369/P369-C			
Application Number	14/788,164-Conf. #1250	Filed	June 30, 2015		
For SPEECH RECOGNITION CIRCUIT WITH ACOUSTIC COPROCESSOR (AS AMENDED)					
Art Unit	2658	Examiner	S. I. McFadden		
This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above-identified application. The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):					
		Fee	Small Entity Fee	Micro Entity Fee	
☒	One month (37 CFR 1.17(a)(1))	\$200	\$100	\$50	\$ 100.00
☐	Two months (37 CFR 1.17(a)(2))	\$600	\$300	\$150	\$
☐	Three months (37 CFR 1.17(a)(3))	\$1,400	\$700	\$350	\$
☐	Four months (37 CFR 1.17(a)(4))	\$2,200	\$1,100	\$550	\$
☐	Five months (37 CFR 1.17(a)(5))	\$3,000	\$1,500	\$750	\$
☒	Applicant asserts small entity status. See 37 CFR 1.27.				
☐	Applicant certifies micro entity status. See 37 CFR 1.29. Form PTO/SB/15A or B or equivalent must either be enclosed or have been submitted previously.				
☐	A check in the amount of the fee is enclosed.				
☒	Payment by credit card. Form PTO-2038 is attached.				
☐	The Director has already been authorized to charge fees in this application to a Deposit Account.				
☒	The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 23-2185				
☐	Payment made via EFS-Web.				
WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.					
I am the					
☐	applicant.				
☒	attorney or agent of record. Registration number 31,063				
☐	attorney or agent acting under 37 CFR 1.34. Registration number				
		August 9, 2017			
Signature		Date			
Stephen A. Soffen		(202) 420-4879			
Typed or printed name		Telephone Number			
NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. Submit multiple forms if more than one signature is required, see below*.					
☐	* Total of 1 forms are submitted.				

Electronic Patent Application Fee Transmittal				
Application Number:		14788164		
Filing Date:		30-Jun-2015		
Title of Invention:		SPEECH RECOGNITION CIRCUIT AND METHOD		
First Named Inventor/Applicant Name:		Guy Larri		
Filer:		Stephen A. Soffen/Cheryl de Lopez		
Attorney Docket Number:		M0025.0369/P369-C		
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension - 1 month with \$0 paid	2251	1	100	100
Miscellaneous:				
Total in USD (\$)				100

Electronic Acknowledgement Receipt	
EFS ID:	30025766
Application Number:	14788164
International Application Number:	
Confirmation Number:	1250
Title of Invention:	SPEECH RECOGNITION CIRCUIT AND METHOD
First Named Inventor/Applicant Name:	Guy Larri
Customer Number:	24998
Filer:	Stephen A. Soffen/Cheryl de Lopez
Filer Authorized By:	Stephen A. Soffen
Attorney Docket Number:	M0025.0369/P369-C
Receipt Date:	09-AUG-2017
Filing Date:	30-JUN-2015
Time Stamp:	10:21:56
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$ 100
RAM confirmation Number	080917INTEFSW10223600
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		restresp369c.pdf	211594	yes	5
			1c34d10ae129ae2ec6253357c5a427ad76123a6c		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Response to Election / Restriction Filed		2	5	
	Extension of Time		1	1	
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30641	no	2
			924fcd7bb7986be1890a6e9e8880ee7d88719f7a		
Warnings:					
Information:					
Total Files Size (in bytes):			242235		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/788,164	06/30/2015	Guy Larri	M0025.0369/P369-C	1250
24998	7590	09/28/2017		
Blank Rome LLP 1825 EYE STREET NW Washington, DC 20006-5403			EXAMINER MCFADDEN, SUSAN IRIS	
			ART UNIT 2658	PAPER NUMBER
			NOTIFICATION DATE 09/28/2017	DELIVERY MODE ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

WashingtonDocketing@blankrome.com

Office Action Summary	Application No. 14/788,164	Applicant(s) LARRI ET AL.	
	Examiner SUSAN MCFADDEN	Art Unit 2658	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on 8-9-17.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on ____.

2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.

3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on ____; the restriction requirement and election have been incorporated into this action.

4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5) ☒ Claim(s) 2-17 is/are pending in the application.
5a) Of the above claim(s) 2-10 is/are withdrawn from consideration.

6) ☐ Claim(s) ____ is/are allowed.

7) ☒ Claim(s) 11-17 is/are rejected.

8) ☐ Claim(s) ____ is/are objected to.

9) ☐ Claim(s) ____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10) ☒ The specification is objected to by the Examiner.

11) ☒ The drawing(s) filed on 9-16-15 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a) ☒ All b) ☐ Some** c) ☐ None of the:

1. ☒ Certified copies of the priority documents have been received.

2. ☐ Certified copies of the priority documents have been received in Application No. ____.

3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b)
Paper No(s)/Mail Date ____.

3) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. ____.

4) ☐ Other: ____.

DETAILED ACTION

1. The present application is being examined under the pre-AIA first to invent provisions.

Priority

Receipt is acknowledged of certified copies of papers required by 37 CFR 1.55.

Specification

The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Claim Rejections - 35 USC § 101

1. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 11-17 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter because the claim(s) as a whole, considering all claim elements both individually and in combination, do not amount to significantly more than an abstract idea. In regard to claims 11-17, the claim(s) is/are directed to "calculating feature vectors", "calculating distances" and "identifying spoken words" using distance measures. The abstract idea per se amount(s) to no more than: data manipulation using an algorithm. Viewed as a whole, these additional claim element(s) do not provide meaningful limitation(s) to transform the abstract idea into a patent eligible application of the abstract idea such that the claim(s) amounts to significantly

more than the abstract idea itself. Therefore, the claim(s) are rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter.

In regard to Alice, a two part test is performed. Are the claims related to an Abstract idea? Yes, there are just "calculating and identifying". The second part is "Are there any elements sufficient to ensure that the claims amount to significantly more than an abstract idea itself?" In this case, the answer is No. Courts have determined that organizing information through mathematical calculations is non-statutory subject matter. See *Digitech*.

Double Patenting

1. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422

F.2d 438, 164 USPQ 619 (CCPA 1970); and In re Thorington, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

1. Claims 11,13, and 15 are rejected on the ground of nonstatutory double patenting as being unpatentable over claims 14 and 15 of Patent No. 7,977,277.

Although the claims at issue are not identical, they are not patentably distinct from each other because they both claim a speech recognition system which uses distances to identify words.

Claims 11 and 13 are similar to claim 14 of US Patent 7,977,277. The Markov or neural network being similar to a tree.

Claim 15 is similar to claim 1 of US Patent 7,977,277. Two distinct circuits or programmable devices.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SUSAN MCFADDEN whose telephone number is (571)272-7621. The examiner can normally be reached on Monday-Friday, 9:30-6:00.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Richemond Dorvil can be reached on 571-272-7602. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/SUSAN MCFADDEN/
Primary Examiner, Art Unit 2658
September 7, 2017

Search Notes 	Application/Control No. 14788164	Applicant(s)/Patent Under Reexamination LARRI ET AL.
	Examiner SUSAN MCFADDEN	Art Unit 2658

CPC- SEARCHED		
Symbol	Date	Examiner
G10L 15/10	9-26-16	SM

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
704	238,251	9-26-16	SM

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

SEARCH NOTES		
Search Notes	Date	Examiner
Searched WEST	9-26-16	SM
Consulted with SPE Rich Dorvil about cancellation of claims and new claims added	5-1-17	SM
Updated Search	9-7-17	SM

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

--	--

WEST Search History for Application 14788164

Creation Date: 2017090715:09

Prior Art Searches

Query	DB	Hits	Op.	Plur.	Thes.	Date
larri.in.	USPT	26	OR	YES		09-19-2016
larri.in. and audio and calculating	USPT	6	OR	YES		09-19-2016
larri.in. and audio and calculating near5 distances	USPT	3	OR	YES		09-19-2016
larri.in. and audio and calculating near5 distances and similarity	USPT	3	OR	YES		09-19-2016
larri.in. and audio and calculating near5 distances and similarity and tree	USPT	3	OR	YES		09-19-2016
larri.in. and audio near front and calculating near5 distances and similarity and tree	USPT	3	OR	YES		09-19-2016
larri.in. and audio near front near end and calculating near5 distances and similarity and tree	USPT	3	OR	YES		09-19-2016
larri.in.	USPT	26	OR	YES		09-26-2016
larri.in. and audio near front	USPT	3	OR	YES		09-26-2016
larri.in. and audio near front and calculating	USPT	3	OR	YES		09-26-2016
larri.in. and audio near front and calculating and search	USPT	3	OR	YES		09-26-2016
larri.in. and audio near front and calculating and search and distance	USPT	3	OR	YES		09-26-2016
larri.in. and audio near front and calculating and search and distance and lexical	USPT	3	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical	USPT	3	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical near tree	USPT	3	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical near tree and word	USPT	3	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical near tree and word near identifying	USPT	0	OR	YES		09-26-2016

audio near front and calculating and search and distance and lexical near tree and identifying near4 word	USPT	0	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical near tree and identify near4 word	USPT	3	OR	YES		09-26-2016
acoustic near4 coprocessor	USPT	10	OR	YES		04-25-2017
acoustic near4 coprocessor and similarity	USPT	0	OR	YES		04-25-2017
acoustic near4 coprocessor and distance	USPT	10	OR	YES		04-25-2017
acoustic near4 coprocessor and distance	PGPB	2	OR	YES		04-25-2017
audio near front near end and distances	PGPB	43	OR	YES		04-25-2017
audio near front near end and distances and similarity	PGPB	5	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic	PGPB	5	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic near coprocessor	PGPB	0	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic near5 coprocessor	PGPB	0	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic and Hidden near markov near models	PGPB	4	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic and Hidden near markov near models and first near programmable near device	PGPB	0	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic and Hidden near markov near models and neural near network	PGPB	4	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic and Hidden near markov near models and neural near network and integrated near circuit	PGPB	4	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic and Hidden near markov near models and neural near network and integrated near circuit and speech near recognition	PGPB	4	OR	YES		04-25-2017
distances and similarity and acoustic and Hidden	USPT	76	OR	YES		04-25-2017

near markov near models and neural near network and integrated near circuit and speech near recognition						
(distances and similarity and acoustic and Hidden near markov near models and neural near network and integrated near circuit and speech near recognition).clm.	USPT	0	OR	YES		04-25-2017
(distances and similarity and acoustic and (Hidden near markov near models or neural near network) and integrated near circuit and speech near recognition).clm.	USPT	0	OR	YES		04-25-2017
(distances and similarity and acoustic and (Hidden near markov near models or neural near network) and speech near recognition).clm.	USPT	3	OR	YES		04-25-2017
(distances and similarity and acoustic and (Hidden near markov near models or neural near network) and speech near recognition).clm. and word	USPT	3	OR	YES		04-25-2017
(distances and similarity and acoustic and (Hidden near markov near models or neural near network) and speech near recognition).clm. and spoken near4 word	USPT	1	OR	YES		04-25-2017
larri.in.	USPT	26	OR	YES		09-05-2017
larri.in. and recognition	USPT	6	OR	YES		09-05-2017
larri.in. and recognition and words	USPT	3	OR	YES		09-05-2017
larri.in. and recognition and words and (HMM or markov or neural near network)	USPT	3	OR	YES		09-05-2017
larri.in. and recognition and words and (HMM or markov or neural near network) and distances	USPT	3	OR	YES		09-05-2017
larri.in. and (recognition or identification) and words and (HMM or markov or neural near network) and distances	USPT	3	OR	YES		09-05-2017
larri.in. and ((recognition or identification) and words and (HMM or markov or neural near network or tree) and distances).clm.	USPT	1	OR	YES		09-05-2017
larri.in. and ((recognition or identification) and words and (HMM or markov or neural near network or tree) and distances).clm. and (time or period)	USPT	1	OR	YES		09-05-2017
larri.in. and ((recognition or identification) and words and (HMM or markov or neural near network or tree) and distances).clm. and (time or	PGPB	2	OR	YES		09-05-2017

period)						
(recognition or identification)	USPT	918457	OR	YES		09-07-2017
(recognition or identification) and word	USPT	368704	OR	YES		09-07-2017
(recognition or identification) and word and distances	USPT	143265	OR	YES		09-07-2017
(recognition or identification) and word and distances and similarity	USPT	24993	OR	YES		09-07-2017
(recognition or identification) and word and distances and similarity and (markov or neural)	USPT	7307	OR	YES		09-07-2017
(recognition or identification) and word and distances and similarity and (markov or neural) and audio	USPT	2538	OR	YES		09-07-2017
(recognition or identification) and word and distances and similarity and (markov or neural) and audio and feature near vector	USPT	715	OR	YES		09-07-2017
((recognition or identification) and word and distances and similarity and (markov or neural) and audio and feature near vector).clm.	USPT	0	OR	YES		09-07-2017
((recognition or identification) and word and (distances or similarity) and (markov or neural) and audio and feature near vector).clm.	USPT	2	OR	YES		09-07-2017
((recognition or identification) and word and (distances or similarity) and (markov or neural) and audio and feature near vector).clm.	PGPB	4	OR	YES		09-07-2017
7972777.pn. and audio and similarity and recognition	USPT	0	OR	YES		09-07-2017
7979277.pn. and audio and similarity and recognition	USPT	1	OR	YES		09-07-2017
7979277.pn. and audio and similarity and recognition and circuit	USPT	1	OR	YES		09-07-2017
larri.in. and audio and similarity and recognition and circuit	USPT	3	OR	YES		09-07-2017
larri.in. and audio and similarity and recognition and circuit and (markov or neural or tree)	USPT	3	OR	YES		09-07-2017
larri.in. and audio and similarity and recognition and circuit and (markov or neural or tree) and word	USPT	3	OR	YES		09-07-2017
larri.in. and audio and similarity and recognition and circuit and (markov or neural or tree) and	PGPB	4	OR	YES		09-07-2017

word						
larri.in. and audio and similarity and recognition and circuit and (markov or neural or tree) and word and integrated near circuit	PGPB	4	OR	YES		09-07-2017
larri.in. and audio and similarity and recognition and circuit and (markov or neural or tree) and word and integrated near circuit and uni-processor	PGPB	4	OR	YES		09-07-2017

Substitute for form 1449/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(Use as many sheets as necessary)</i>				Complete if Known	
				Application Number	Not yet assigned
				Filing Date	Concurrently Herewith
				First Named Inventor	Guy Larri
				Art Unit	Not yet assigned
				Examiner Name	Not yet assigned
Sheet	1	of	1	Attorney Docket Number	M0025.0369-C

U.S. PATENT DOCUMENTS					
Examiner Initials*	Cite No. ¹	Document Number	Publication Date MM-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number-Kind Code ² (if known)			
		5,881,312	03-1999	Dulong	
		7,328,153	02-2008	Wells et al.	
		2004/0111264 A1	06-2004	Wang et al.	
		2003/0061046 A1	03-2003	Zhao et al.	
		6,374,220	04-2002	Kao	
		9,076,441	07-2015	Larri et al.	
		8,352,262	01-2013	Larri et al.	
		7,979,277	07-2011	Larri et al.	
		8,036,890	10-2011	Catchpole	
		8,612,227	12-2013	Kato	

FOREIGN PATENT DOCUMENTS						
Examiner Initials*	Cite No. ¹	Foreign Patent Document	Publication Date	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear	T ⁶
		Country Code ³ -Number ⁴ -Kind Code ⁵ (if known)	MM-YYYY			
		GB 2 391 679 A	02-2004			
		EP 1 178 466 A2	02-2002			
		WO 01/75862 A2	10-2001			
		WO 01/48737 A2	07-2001			
		GB 2 333 172 A	07-1999			
		WO 03/067572 A2	08-2003			

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. ¹ Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. ³ Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). ⁴ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁵ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁶ Applicant is to place a check mark here if English language Translation is attached.

NON PATENT LITERATURE DOCUMENTS				
Examiner Initials	Cite No. ¹	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.		T ²
		Glinski et al., "Spoken Language Recognition on DSP Array Processor" IEEE Transactions on Parallel and Distributed Systems, Vol. 5, No. 7, pp. 697-703, July 1, 1994.		
		S. Chatterjee et al., "Connected Speech Recognition on a Multiple Processor Pipeline." ICASSP 89, pp. 774-777, May 23, 1989.		

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

Examiner Signature	/SUSAN I MCFADDEN/ (09/06/2017)	Date Considered	09/06/2017
--------------------	---------------------------------	-----------------	------------

DSMDB-3348573 v1

<i>Index of Claims</i> 	Application/Control No. 14788164	Applicant(s)/Patent Under Reexamination LARRI ET AL.
	Examiner SUSAN MCFADDEN	Art Unit 2658

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

☐ Claims renumbered in the same order as presented by applicant										☐ CPA										☐ T.D.										☐ R.1.47									
CLAIM		DATE																																					
Final	Original	09/26/2016	05/01/2017	09/07/2017																																			
	1	✓	-	-																																			
	2		÷	N																																			
	3		÷	N																																			
	4		÷	N																																			
	5		÷	N																																			
	6		÷	N																																			
	7		÷	N																																			
	8		÷	N																																			
	9		÷	N																																			
	10		÷	N																																			
	11		÷	✓																																			
	12		÷	✓																																			
	13		÷	✓																																			
	14		÷	✓																																			
	15		÷	✓																																			
	16		÷	✓																																			
	17		÷	✓																																			



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/788,164	06/30/2015	Guy Larri	M0025.0369/P369-C	1250
24998	7590	10/31/2017		
Blank Rome LLP 1825 EYE STREET NW Washington, DC 20006-5403			EXAMINER MCFADDEN, SUSAN IRIS	
			ART UNIT 2658	PAPER NUMBER
			NOTIFICATION DATE 10/31/2017	DELIVERY MODE ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

WashingtonDocketing@blankrome.com

<i>Applicant-Initiated Interview Summary</i>	Application No. 14/788,164	Applicant(s) LARRI ET AL.	
	Examiner SUSAN MCFADDEN	Art Unit 2658	

All participants (applicant, applicant's representative, PTO personnel):

(1) SUSAN MCFADDEN. (3) ____.

(2) Jon Grossman. (4) ____.

Date of Interview: 25 October 2017.

Type: ☒ Telephonic ☐ Video Conference
 ☐ Personal [copy given to: ☐ applicant ☐ applicant's representative]

Exhibit shown or demonstration conducted: ☐ Yes ☐ No.
 If Yes, brief description: ____.

Issues Discussed ☒ 101 ☐ 112 ☐ 102 ☐ 103 ☐ Others
 (For each of the checked box(es) above, please describe below the issue and detailed description of the discussion)

Claim(s) discussed: 11.

Identification of prior art discussed: ____.

Substance of Interview
 (For each issue discussed, provide a detailed description and indicate if agreement was reached. Some topics may include: identification or clarification of a reference or a portion thereof, claim interpretation, proposed amendments, arguments of any applied references etc...)

Discussed adding features like the "search stage that identifies words using a lexical tree" to overcome the 101 rejection. Applicant will consider changes and then submit a formal amendment. A further search will be conducted on the new claims.

Applicant recordation instructions: The formal written reply to the last Office action must include the substance of the interview. (See MPEP section 713.04). If a reply to the last Office action has already been filed, applicant is given a non-extendable period of the longer of one month or thirty days from this interview date, or the mailing date of this interview summary form, whichever is later, to file a statement of the substance of the interview

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

☒ Attachment

/SUSAN MCFADDEN/ Primary Examiner, Art Unit 2658	
---	--

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record

A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews Paragraph (b)

In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.

All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:

- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:

- 1) A brief description of the nature of any exhibit shown or any demonstration conducted,
- 2) an identification of the claims discussed,
- 3) an identification of the specific prior art discussed,
- 4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
- 5) a brief identification of the general thrust of the principal arguments presented to the examiner,
(The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
- 6) a general indication of any other pertinent matters discussed, and
- 7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

**UNITED STATES DEPARTMENT OF COMMERCE****U.S. Patent and Trademark Office**

Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPLICATION NO./ CONTROL NO.	FILING DATE	FIRST NAMED INVENTOR / PATENT IN REEXAMINATION	ATTORNEY DOCKET NO.
14/788,164	30 June, 2015	LARRI ET AL.	M0025.0369/P369-C

Blank Rome LLP 1825 EYE STREET NW Washington, DC 20006-5403		EXAMINER	
		SUSAN MCFADDEN	
		ART UNIT	PAPER
		2658	20171025

DATE MAILED:

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner for Patents

Attached is the Interview Summary from 10-25-17.

/SUSAN MCFADDEN/
Primary Examiner, Art Unit 2658

PTO-90C (Rev.04-03)

<i>Applicant-Initiated Interview Summary</i>	Application No. 14/788,164	Applicant(s) LARRI ET AL.	
	Examiner SUSAN MCFADDEN	Art Unit 2658	

All participants (applicant, applicant's representative, PTO personnel):

(1) SUSAN MCFADDEN. (3) ____.

(2) Jon Grossman. (4) ____.

Date of Interview: 25 October 2017.

Type: ☒ Telephonic ☐ Video Conference
 ☐ Personal [copy given to: ☐ applicant ☐ applicant's representative]

Exhibit shown or demonstration conducted: ☐ Yes ☐ No.
 If Yes, brief description: ____.

Issues Discussed ☒ 101 ☐ 112 ☐ 102 ☐ 103 ☐ Others
 (For each of the checked box(es) above, please describe below the issue and detailed description of the discussion)

Claim(s) discussed: 11.

Identification of prior art discussed: ____.

Substance of Interview
 (For each issue discussed, provide a detailed description and indicate if agreement was reached. Some topics may include: identification or clarification of a reference or a portion thereof, claim interpretation, proposed amendments, arguments of any applied references etc...)

Discussed adding features like the "search stage that identifies words using a lexical tree" to overcome the 101 rejection. Applicant will consider changes and then submit a formal amendment. A further search will be conducted on the new claims.

Applicant recordation instructions: The formal written reply to the last Office action must include the substance of the interview. (See MPEP section 713.04). If a reply to the last Office action has already been filed, applicant is given a non-extendable period of the longer of one month or thirty days from this interview date, or the mailing date of this interview summary form, whichever is later, to file a statement of the substance of the interview

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

☒ Attachment

/SUSAN MCFADDEN/ Primary Examiner, Art Unit 2658	
---	--

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record

A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews Paragraph (b)

In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.

All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:

- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:

- 1) A brief description of the nature of any exhibit shown or any demonstration conducted,
- 2) an identification of the claims discussed,
- 3) an identification of the specific prior art discussed,
- 4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
- 5) a brief identification of the general thrust of the principal arguments presented to the examiner,
(The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
- 6) a general indication of any other pertinent matters discussed, and
- 7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

Doc Code: DIST.E.FILE Document Description: Electronic Terminal Disclaimer - Filed		PTO/SB/26 U.S. Patent and Trademark Office Department of Commerce	
Electronic Petition Request	TERMINAL DISCLAIMER TO OBVIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT		
Application Number	14788164		
Filing Date	30-Jun-2015		
First Named Inventor	Guy Larri		
Attorney Docket Number	M0025.0369/P369-C		
Title of Invention	SPEECH RECOGNITION CIRCUIT AND METHOD		
☒ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action ☒ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.			
Owner		Percent Interest	
Zentian Limited		100%	
<p>The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)</p> <p>7979277</p> <p>as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.</p> <p>In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:</p> <ul style="list-style-type: none"> - expires for failure to pay a maintenance fee; - is held unenforceable; - is found invalid by a court of competent jurisdiction; - is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321; - has all claims canceled by a reexamination certificate; - is reissued; or - is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer. <p>☒ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.</p>			

☐ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.	
Applicant claims the following fee status:	
☒ Small Entity ☐ Micro Entity ☐ Regular Undiscounted	
I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.	
THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES	
I certify, in accordance with 37 CFR 1.4(d)(4) that I am:	
☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application Registration Number <u> 31063 </u>	
☐ A sole inventor	
☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application	
☐ A joint inventor; all of whom are signing this request	
Signature	/Stephen A. Soffen/
Name	Stephen A. Soffen

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
 Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

Electronic Patent Application Fee Transmittal				
Application Number:		14788164		
Filing Date:		30-Jun-2015		
Title of Invention:		SPEECH RECOGNITION CIRCUIT AND METHOD		
First Named Inventor/Applicant Name:		Guy Larri		
Filer:		Stephen A. Soffen/Cheryl de Lopez		
Attorney Docket Number:		M0025.0369/P369-C		
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
STATUTORY OR TERMINAL DISCLAIMER	2814	1	160	160
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				160

Doc Code: DISQ.E.FILE
Document Description: Electronic Terminal Disclaimer – Approved

Application No.: 14788164

Filing Date: 30-Jun-2015

Applicant/Patent under Reexamination: Larri

Electronic Terminal Disclaimer filed on February 28, 2018

☒ APPROVED

This patent is subject to a terminal disclaimer

☐ DISAPPROVED

Approved/Disapproved by: Electronic Terminal Disclaimer automatically approved by EFS-Web

U.S. Patent and Trademark Office

Electronic Acknowledgement Receipt	
EFS ID:	31910680
Application Number:	14788164
International Application Number:	
Confirmation Number:	1250
Title of Invention:	SPEECH RECOGNITION CIRCUIT AND METHOD
First Named Inventor/Applicant Name:	Guy Larri
Customer Number:	24998
Filer:	Stephen A. Soffen/Cheryl de Lopez
Filer Authorized By:	Stephen A. Soffen
Attorney Docket Number:	M0025.0369/P369-C
Receipt Date:	28-FEB-2018
Filing Date:	30-JUN-2015
Time Stamp:	10:09:52
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$160
RAM confirmation Number	022818INTEFSW10095000
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Terminal Disclaimer-Filed (Electronic)	eTerminal-Disclaimer.pdf	33391	no	2
			d4a2b8c0b24dd25dad2f0881c1a0e79792d ccbf		
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30462	no	2
			76aa23adeee4feeb3c4396a0c7295acf7e8e c38b		
Warnings:					
Information:					
Total Files Size (in bytes):			63853		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Guy Larri et al.

Application No.: 14/788,164

Confirmation No.: 1250

Filed: June 30, 2015

Art Unit: 2658

For: DISTRIBUTED PIPELINED PARALLEL
SPEECH RECOGNITION SYSTEM (AS
AMENDED)

Examiner: S. I. McFadden

AMENDMENT IN RESPONSE TO NON-FINAL OFFICE ACTION

MS Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

INTRODUCTORY COMMENTS

In response to the Office Action dated September 28, 2017, please amend the above-identified U.S. patent application as follows:

Amendments to the Specification begin on page 2 of this paper.

Amendments to the Claims are reflected in the listing of claims which begins on page 3 of this paper.

Remarks/Arguments begin on page 5 of this paper.

200494.00239/106667761

AMENDMENTS TO THE SPECIFICATION

Please amend the Title as follows:

~~SPEECH RECOGNITION CIRCUIT WITH ACOUSTIC COPROCESSOR~~
DISTRIBUTED PIPELINED PARALLEL SPEECH RECOGNITION SYSTEM

AMENDMENTS TO THE CLAIMS

Claims 1-10. (Canceled)

11. (Currently amended) A speech recognition system comprising:

a first programmable device programmed to calculate a feature vector from a digital audio stream, wherein the feature vector comprises a plurality of extracted and/or derived quantities from said digital audio stream during a defined audio time frame;

a second programmable device programmed to calculate distances indicating the similarity between a feature vector and a plurality of acoustic states of an acoustic model wherein said feature vector is received by the second programmable device after it is calculated by the first programmable device; and

a third programmable device programmed to identify spoken words in said digital audio stream using Hidden Markov Models and/or Neural Networks wherein said word identification uses one or more distances that were calculated by the second programmable device[[:]], wherein said identification of spoken words uses one or more distances calculated from a first feature vector[[:]]; and

a search stage for using the calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words.

12. (Previously presented) The speech recognition system of claim 11 wherein the time period taken for said identification of spoken words overlaps the time period taken for said second programmable device to calculate one or more distances from a second feature vector.

13. (Previously presented) The speech recognition system of claim 12 wherein the first programmable device, the second programmable device, and third programmable device are physically distinct processing circuits.

14. (Canceled)

15. (Previously presented) The speech recognition system of claim 12 wherein the first programmable device and the second programmable device comprise one physical uni-processor or multi-processor, and the second programmable device is a physically distinct processing circuit.

16. (Previously presented) The speech recognition system of claim 13 wherein said physically distinct processing circuits are all formed on a single integrated circuit.

17. (Previously presented) The speech recognition system of claim 15 wherein said one physical uni-processor or multi-processor and said second programmable device are all formed on a single integrated circuit.

REMARKS

Previously elected claims (11-17) stand rejected in the Office Action mailed on September 28, 2017. In response, claim 11 is hereby amended and claims 1-10 and 14 are cancelled without prejudice. Accordingly, claims 11-13 and 15-17 remain pending. For the reasons set forth below, the outstanding claims overcome all grounds of rejection and are in condition for allowance.

At the outset, applicant acknowledges with appreciation the telephonic interview conducted with the Examiner on October 25, 2017 with respect to the abstract idea rejection discussed below.

Claims 11-17 were rejected in the Office Action as constituting non-statutory subject matter under 35 U.S.C. § 101 as being drawn to an “abstract idea.” During the interview, applicant presented a proposed amendment to independent claim 11 and was acknowledged as being sufficient to render claim 11 patent eligible. Claim 11 has been amended as follows to include the terms proposed during the interview: “a search stage for using the calculated distances to identify words within a lexical tree, the lexical tree comprising a model of words.”

As amended, the claim 11 as a whole recites sufficient system elements that render it non-abstract under the two-part Alice test. In Alice Corporation Pty. Ltd. v. CLS Bank International 134 S. Ct. 2347 (2014), the Supreme Court set forth the test for patent eligibility as follows: step one, known as the “filter step,” tests whether the claims at issue are directed to a patent- ineligible concept, such as an abstract idea. If the claims are deemed abstract in step one, then the inquiry passes to step two, which tests whether the elements of the claim contain an inventive concept sufficient to transform the abstract idea determined in filter step one into a patent- eligible invention. To determine step two, all individual claim limitations or the ordered combination of claim limitations are looked at to test whether there is “something more” than the performance of well- understood routine and conventional activities previously performed manually or already known in the industry.

As amended, claim 11 recites a non-abstract speech recognition system including, among other things: a first programmable device programmed to calculate a feature vector from a digital audio stream, a second programmable device programmed to calculate distances indicating the similarity between a feature vector and a plurality of acoustic states of an acoustic model, a third programmable device programmed to identify spoken words in said digital audio stream, and the search stage quoted above. Claim 11 recites a structure that operates as a whole as a non-abstract system for several reasons. First, the claim recites system elements without using the programmable devices as “mere tools” *See Visual Memory LLC v. NVIDIA Corp.*, 867 F.3d 1253 (Fed. Cir. 2017). Second, the speech recognition system presents a new approach that overcomes prior art deficiencies by providing an efficient speech recognition system for use in mobile devices. *See Thales Visionix Inc., v. United States*, 850 F.3d 1343 (Fed. Cir. 2017). Third, claim 11 does not rely on routine or conventional activities to achieve its recited solution. *See Berkheimer v. HP Inc.*, No. 2017-1437, 2018 U.S. App. LEXIS 3040 (Fed. Cir. Feb. 8, 2018)(applied to second step of the two-part Alice test). Accordingly, for at least the above reasons, claim 11 is not abstract. Under *Alice*, step 2 of the test is not necessary since claim 11 is not abstract. Accordingly, claim 11 as well as dependent claims 12-13 and 15-17 are patent eligible under 35 U.S.C. § 101.

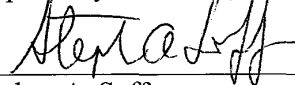
Claims 11, 13 and 15 stand rejected on the ground of non-statutory double patenting as being obvious over claims 14 and 15 of U.S. Patent No. 7,979,277 (the Office Action erroneously references this parent patent as U.S. Patent No. 7,977,277). To obviate this obviousness-type double patenting rejection, applicant has filed an e-terminal disclaimer concurrently herewith.

In view of the above amendment and at least for the reasons set forth herein, Applicant believes the pending application is in condition for allowance.

The Director is hereby authorized to charge any deficiency or credit any overpayment in the fees filed, asserted to be filed or which should have been filed herewith (or with any paper hereafter filed in this application by this firm) to our Deposit Account No. 23-2185, under Order No. M0025.0369/P369-C.

Dated: February 28, 2018

Respectfully submitted,

By 

Stephen A. Soffen

Registration No.: 31,063
BLANK ROME LLP
1825 Eye Street, NW
Washington, DC 20006-5403
(202) 420-2200
Attorney for Applicant

Electronic Patent Application Fee Transmittal				
Application Number:		14788164		
Filing Date:		30-Jun-2015		
Title of Invention:		SPEECH RECOGNITION CIRCUIT AND METHOD		
First Named Inventor/Applicant Name:		Guy Larri		
Filer:		Stephen A. Soffen/Cheryl de Lopez		
Attorney Docket Number:		M0025.0369/P369-C		
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension - 2 months with \$0 paid	2252	1	300	300
Miscellaneous:				
Total in USD (\$)				300

Electronic Acknowledgement Receipt	
EFS ID:	31918585
Application Number:	14788164
International Application Number:	
Confirmation Number:	1250
Title of Invention:	SPEECH RECOGNITION CIRCUIT AND METHOD
First Named Inventor/Applicant Name:	Guy Larri
Customer Number:	24998
Filer:	Stephen A. Soffen/Cheryl de Lopez
Filer Authorized By:	Stephen A. Soffen
Attorney Docket Number:	M0025.0369/P369-C
Receipt Date:	28-FEB-2018
Filing Date:	30-JUN-2015
Time Stamp:	15:23:50
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	\$ 300
RAM confirmation Number	030118INTEFSW00002164232185
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		amendp369c.pdf	256161	yes	8
			595f31b96fdb3210c6c48948b3eca1bd2edd8a28		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Extension of Time		1	1	
	Amendment/Req. Reconsideration-After Non-Final Reject		2	2	
	Specification		3	3	
	Claims		4	5	
	Applicant Arguments/Remarks Made in an Amendment		6	8	
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30644	no	2
			25916896cddbdc9127263d50ce1fa0a637d917ff		
Warnings:					
Information:					
Total Files Size (in bytes):			286805		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

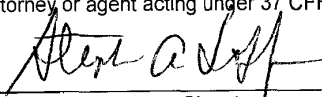
National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a)		Docket Number (Optional) M0025.0369/P369-C																															
Application Number 14/788,164-Conf. #1250		Filed June 30, 2015																															
For DISTRIBUTED PIPELINED PARALLEL SPEECH RECOGNITION SYSTEM (AS AMENDED)																																	
Art Unit 2658		Examiner S. I. McFadden																															
<p>This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above-identified application.</p> <p>The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;"></th> <th style="text-align: center;"><u>Fee</u></th> <th style="text-align: center;"><u>Small Entity Fee</u></th> <th style="text-align: center;"><u>Micro Entity Fee</u></th> <th style="text-align: center;"></th> </tr> </thead> <tbody> <tr> <td>☐ One month (37 CFR 1.17(a)(1))</td> <td style="text-align: center;">\$200</td> <td style="text-align: center;">\$100</td> <td style="text-align: center;">\$50</td> <td style="text-align: center;">\$ _____</td> </tr> <tr> <td>☒ Two months (37 CFR 1.17(a)(2))</td> <td style="text-align: center;">\$600</td> <td style="text-align: center;">\$300</td> <td style="text-align: center;">\$150</td> <td style="text-align: center;">\$ 300.00</td> </tr> <tr> <td>☐ Three months (37 CFR 1.17(a)(3))</td> <td style="text-align: center;">\$1,400</td> <td style="text-align: center;">\$700</td> <td style="text-align: center;">\$350</td> <td style="text-align: center;">\$ _____</td> </tr> <tr> <td>☐ Four months (37 CFR 1.17(a)(4))</td> <td style="text-align: center;">\$2,200</td> <td style="text-align: center;">\$1,100</td> <td style="text-align: center;">\$550</td> <td style="text-align: center;">\$ _____</td> </tr> <tr> <td>☐ Five months (37 CFR 1.17(a)(5))</td> <td style="text-align: center;">\$3,000</td> <td style="text-align: center;">\$1,500</td> <td style="text-align: center;">\$750</td> <td style="text-align: center;">\$ _____</td> </tr> </tbody> </table> <p>☒ Applicant asserts small entity status. See 37 CFR 1.27.</p> <p>☐ Applicant certifies micro entity status. See 37 CFR 1.29. Form PTO/SB/15A or B or equivalent must either be enclosed or have been submitted previously.</p> <p>☐ A check in the amount of the fee is enclosed.</p> <p>☒ Payment by credit card. Form PTO-2038 is attached.</p> <p>☐ The Director has already been authorized to charge fees in this application to a Deposit Account.</p> <p>☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 23-2185</p> <p>☒ Payment made via EFS-Web.</p> <p>WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.</p> <p>I am the</p> <p>☐ applicant.</p> <p>☒ attorney or agent of record. Registration number 31,063</p> <p>☐ attorney or agent acting under 37 CFR 1.34. Registration number _____</p> <div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div style="text-align: center;">  _____ Signature </div> <div style="text-align: center;"> February 28, 2018 _____ Date </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="text-align: center;"> Stephen A. Soffen _____ Typed or printed name </div> <div style="text-align: center;"> (202) 420-4879 _____ Telephone Number </div> </div> <p>NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. Submit multiple forms if more than one signature is required, see below*.</p>					<u>Fee</u>	<u>Small Entity Fee</u>	<u>Micro Entity Fee</u>		☐ One month (37 CFR 1.17(a)(1))	\$200	\$100	\$50	\$ _____	☒ Two months (37 CFR 1.17(a)(2))	\$600	\$300	\$150	\$ 300.00	☐ Three months (37 CFR 1.17(a)(3))	\$1,400	\$700	\$350	\$ _____	☐ Four months (37 CFR 1.17(a)(4))	\$2,200	\$1,100	\$550	\$ _____	☐ Five months (37 CFR 1.17(a)(5))	\$3,000	\$1,500	\$750	\$ _____
	<u>Fee</u>	<u>Small Entity Fee</u>	<u>Micro Entity Fee</u>																														
☐ One month (37 CFR 1.17(a)(1))	\$200	\$100	\$50	\$ _____																													
☒ Two months (37 CFR 1.17(a)(2))	\$600	\$300	\$150	\$ 300.00																													
☐ Three months (37 CFR 1.17(a)(3))	\$1,400	\$700	\$350	\$ _____																													
☐ Four months (37 CFR 1.17(a)(4))	\$2,200	\$1,100	\$550	\$ _____																													
☐ Five months (37 CFR 1.17(a)(5))	\$3,000	\$1,500	\$750	\$ _____																													
☒ * Total of 1 forms are submitted.																																	

200494.00239/106667756

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875				Application or Docket Number 14/788,164		Filing Date 06/30/2015		☐ To be Mailed	
ENTITY: ☐ LARGE ☒ SMALL ☐ MICRO									
APPLICATION AS FILED – PART I									
(Column 1)		(Column 2)							
FOR		NUMBER FILED		NUMBER EXTRA		RATE (\$)		FEE (\$)	
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))		N/A		N/A		N/A			
☐ SEARCH FEE (37 CFR 1.16(k), (j), or (m))		N/A		N/A		N/A			
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))		N/A		N/A		N/A			
TOTAL CLAIMS (37 CFR 1.16(i))		minus 20 =		*		X \$ =			
INDEPENDENT CLAIMS (37 CFR 1.16(h))		minus 3 =		*		X \$ =			
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))		If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).							
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))									
* If the difference in column 1 is less than zero, enter "0" in column 2.						TOTAL			
APPLICATION AS AMENDED – PART II									
(Column 1)		(Column 2)		(Column 3)					
AMENDMENT	02/28/2018		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)		ADDITIONAL FEE (\$)
	Total (37 CFR 1.16(i))		* 18	Minus	** 20	= 0	X \$50 =		0
	Independent (37 CFR 1.16(h))		* 1	Minus	***3	= 0	X \$230 =		0
	☐ Application Size Fee (37 CFR 1.16(s))								
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
						TOTAL ADD'L FEE		0	
(Column 1)		(Column 2)		(Column 3)					
AMENDMENT			CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)		ADDITIONAL FEE (\$)
	Total (37 CFR 1.16(i))		*	Minus	**	=	X \$ =		
	Independent (37 CFR 1.16(h))		*	Minus	***	=	X \$ =		
	☐ Application Size Fee (37 CFR 1.16(s))								
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
						TOTAL ADD'L FEE			
<p>* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.</p> <p>** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".</p> <p>*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".</p> <p>The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.</p>									

LIE
ANDREA FREEMAN

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

24998 7590 04/23/2018
Blank Rome LLP
1825 EYE STREET NW
Washington, DC 20006-5403

EXAMINER

MCFADDEN, SUSAN IRIS

ART UNIT PAPER NUMBER

2658

DATE MAILED: 04/23/2018

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/788,164	06/30/2015	Guy Larri	M0025.0369/P369-C	1250

TITLE OF INVENTION: SPEECH RECOGNITION CIRCUIT AND METHOD

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	\$500	\$0	\$0	\$500	07/23/2018

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 1/2 the amount of undiscounted fees, and micro entity fees are 1/2 the amount of small entity fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Maintenance fees are due in utility patents issuing on applications filed on or after Dec. 12, 1980. It is patentee's responsibility to ensure timely payment of maintenance fees when due. More information is available at www.uspto.gov/PatentMaintenanceFees.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

24998 7590 04/23/2018
Blank Rome LLP
1825 EYE STREET NW
Washington, DC 20006-5403

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/788,164	06/30/2015	Guy Larri	M0025.0369/P369-C	1250

TITLE OF INVENTION: SPEECH RECOGNITION CIRCUIT AND METHOD

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	\$500	\$0	\$0	\$500	07/23/2018

EXAMINER	ART UNIT	CLASS-SUBCLASS
MCFADDEN, SUSAN IRIS	2658	704-251000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
 (1) The names of up to 3 registered patent attorneys or agents OR, alternatively, 1 _____
 (2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed. 2 _____
 3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)
 PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.
 (A) NAME OF ASSIGNEE _____ (B) RESIDENCE: (CITY and STATE OR COUNTRY) _____

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
☐ Issue Fee
☐ Publication Fee (No small entity discount permitted)
☐ Advance Order - # of Copies _____
 4b. Payment of Fee(s): (**Please first reapply any previously paid issue fee shown above**)
☐ A check is enclosed.
☐ Payment by credit card. Form PTO-2038 is attached.
☐ The director is hereby authorized to charge the required fee(s), any deficiency, or credits any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
☐ Applicant certifying micro entity status. See 37 CFR 1.29
☐ Applicant asserting small entity status. See 37 CFR 1.27
☐ Applicant changing to regular undiscounted fee status.
 NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.
 NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.
 NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature _____ Date _____
 Typed or printed name _____ Registration No. _____



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/788,164	06/30/2015	Guy Larri	M0025.0369/P369-C	1250
24998	7590	04/23/2018	EXAMINER	
Blank Rome LLP 1825 EYE STREET NW Washington, DC 20006-5403			MCFADDEN, SUSAN IRIS	
			ART UNIT	PAPER NUMBER
			2658	
DATE MAILED: 04/23/2018				

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)

(Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. **DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.** Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No. 14/788,164	Applicant(s) LARRI ET AL.	
	Examiner SUSAN MCFADDEN	Art Unit 2658	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to Terminal Disclaimer filed 2-28-18.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.
2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.
3. ☒ The allowed claim(s) is/are 11-13 and 15-17. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.
4. ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
Certified copies:
a) ☒ All b) ☐ Some *c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
2. ☒ Certified copies of the priority documents have been received in Application No. 11/662704.
3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).
* Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.
Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s) 1. ☐ Notice of References Cited (PTO-892) 2. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____ 3. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material 4. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ . 5. ☐ Examiner's Amendment/Comment 6. ☐ Examiner's Statement of Reasons for Allowance 7. ☐ Other _____.	
---	--

/SUSAN MCFADDEN/ Primary Examiner, Art Unit 2658	
---	--

Search Notes 	Application/Control No. 14788164	Applicant(s)/Patent Under Reexamination LARRI ET AL.
	Examiner SUSAN MCFADDEN	Art Unit 2658

CPC- SEARCHED		
Symbol	Date	Examiner
G10L 15/10	9-26-16	SM

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
704	238,251	9-26-16	SM

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

SEARCH NOTES		
Search Notes	Date	Examiner
Searched WEST	9-26-16	SM
Consulted with SPE Rich Dorvil about cancellation of claims and new claims added	5-1-17	SM
Updated Search	9-7-17	SM
Updated Search	4-5-18	SM

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner
G10L	15/10	4-5-18	SM

--	--

WEST Search History for Application 14788164

Creation Date: 2018040515:30

Interference Searches

Query	DB	Hits	Op.	Plur.	Thes.	Date
(speech near4 recognition and distance and (hmm or markov or neural or lexical) and identify near5 word).clm.	PGPB	2	OR	YES		04-05-2018
(speech near4 recognition and distance and (hmm or markov or neural or lexical) and identify near5 word).clm.	USPT	1	OR	YES		04-05-2018

Prior Art Searches

Query	DB	Hits	Op.	Plur.	Thes.	Date
larri.in.	USPT	26	OR	YES		09-19-2016
larri.in. and audio and calculating	USPT	6	OR	YES		09-19-2016
larri.in. and audio and calculating near5 distances	USPT	3	OR	YES		09-19-2016
larri.in. and audio and calculating near5 distances and similarity	USPT	3	OR	YES		09-19-2016
larri.in. and audio and calculating near5 distances and similarity and tree	USPT	3	OR	YES		09-19-2016
larri.in. and audio near front and calculating near5 distances and similarity and tree	USPT	3	OR	YES		09-19-2016
larri.in. and audio near front near end and calculating near5 distances and similarity and tree	USPT	3	OR	YES		09-19-2016
larri.in.	USPT	26	OR	YES		09-26-2016
larri.in. and audio near front	USPT	3	OR	YES		09-26-2016
larri.in. and audio near front and calculating	USPT	3	OR	YES		09-26-2016
larri.in. and audio near front and calculating and search	USPT	3	OR	YES		09-26-2016
larri.in. and audio near front and calculating and search and distance	USPT	3	OR	YES		09-26-2016
larri.in. and audio near front and calculating and search and distance and lexical	USPT	3	OR	YES		09-26-2016

audio near front and calculating and search and distance and lexical	USPT	3	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical near tree	USPT	3	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical near tree and word	USPT	3	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical near tree and word near identifying	USPT	0	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical near tree and identifying near4 word	USPT	0	OR	YES		09-26-2016
audio near front and calculating and search and distance and lexical near tree and identify near4 word	USPT	3	OR	YES		09-26-2016
acoustic near4 coprocessor	USPT	10	OR	YES		04-25-2017
acoustic near4 coprocessor and similarity	USPT	0	OR	YES		04-25-2017
acoustic near4 coprocessor and distance	USPT	10	OR	YES		04-25-2017
acoustic near4 coprocessor and distance	PGPB	2	OR	YES		04-25-2017
audio near front near end and distances	PGPB	43	OR	YES		04-25-2017
audio near front near end and distances and similarity	PGPB	5	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic	PGPB	5	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic near coprocessor	PGPB	0	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic near5 coprocessor	PGPB	0	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic and Hidden near markov near models	PGPB	4	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic and Hidden near markov near models and first near programmable near device	PGPB	0	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic and Hidden near markov near models and neural near network	PGPB	4	OR	YES		04-25-2017

audio near front near end and distances and similarity and acoustic and Hidden near markov near models and neural near network and integrated near circuit	PGPB	4	OR	YES		04-25-2017
audio near front near end and distances and similarity and acoustic and Hidden near markov near models and neural near network and integrated near circuit and speech near recognition	PGPB	4	OR	YES		04-25-2017
distances and similarity and acoustic and Hidden near markov near models and neural near network and integrated near circuit and speech near recognition	USPT	76	OR	YES		04-25-2017
(distances and similarity and acoustic and Hidden near markov near models and neural near network and integrated near circuit and speech near recognition).clm.	USPT	0	OR	YES		04-25-2017
(distances and similarity and acoustic and (Hidden near markov near models or neural near network) and integrated near circuit and speech near recognition).clm.	USPT	0	OR	YES		04-25-2017
(distances and similarity and acoustic and (Hidden near markov near models or neural near network) and speech near recognition).clm.	USPT	3	OR	YES		04-25-2017
(distances and similarity and acoustic and (Hidden near markov near models or neural near network) and speech near recognition).clm. and word	USPT	3	OR	YES		04-25-2017
(distances and similarity and acoustic and (Hidden near markov near models or neural near network) and speech near recognition).clm. and spoken near4 word	USPT	1	OR	YES		04-25-2017
larri.in.	USPT	26	OR	YES		09-05-2017
larri.in. and recognition	USPT	6	OR	YES		09-05-2017
larri.in. and recognition and words	USPT	3	OR	YES		09-05-2017
larri.in. and recognition and words and (HMM or markov or neural near network)	USPT	3	OR	YES		09-05-2017
larri.in. and recognition and words and (HMM or markov or neural near network) and distances	USPT	3	OR	YES		09-05-2017
larri.in. and (recognition or identification) and words and (HMM or markov or neural near network) and distances	USPT	3	OR	YES		09-05-2017

larri.in. and ((recognition or identification) and words and (HMM or markov or neural near network or tree) and distances).clm.	USPT	1	OR	YES		09-05-2017
larri.in. and ((recognition or identification) and words and (HMM or markov or neural near network or tree) and distances).clm. and (time or period)	USPT	1	OR	YES		09-05-2017
larri.in. and ((recognition or identification) and words and (HMM or markov or neural near network or tree) and distances).clm. and (time or period)	PGPB	2	OR	YES		09-05-2017
(recognition or identification)	USPT	918457	OR	YES		09-07-2017
(recognition or identification) and word	USPT	368704	OR	YES		09-07-2017
(recognition or identification) and word and distances	USPT	143265	OR	YES		09-07-2017
(recognition or identification) and word and distances and similarity	USPT	24993	OR	YES		09-07-2017
(recognition or identification) and word and distances and similarity and (markov or neural)	USPT	7307	OR	YES		09-07-2017
(recognition or identification) and word and distances and similarity and (markov or neural) and audio	USPT	2538	OR	YES		09-07-2017
(recognition or identification) and word and distances and similarity and (markov or neural) and audio and feature near vector	USPT	715	OR	YES		09-07-2017
((recognition or identification) and word and distances and similarity and (markov or neural) and audio and feature near vector).clm.	USPT	0	OR	YES		09-07-2017
((recognition or identification) and word and (distances or similarity) and (markov or neural) and audio and feature near vector).clm.	USPT	2	OR	YES		09-07-2017
((recognition or identification) and word and (distances or similarity) and (markov or neural) and audio and feature near vector).clm.	PGPB	4	OR	YES		09-07-2017
7972777.pn. and audio and similarity and recognition	USPT	0	OR	YES		09-07-2017
7979277.pn. and audio and similarity and recognition	USPT	1	OR	YES		09-07-2017
7979277.pn. and audio and similarity and recognition and circuit	USPT	1	OR	YES		09-07-2017

larri.in. and audio and similarity and recognition and circuit	USPT	3	OR	YES		09-07-2017
larri.in. and audio and similarity and recognition and circuit and (markov or neural or tree)	USPT	3	OR	YES		09-07-2017
larri.in. and audio and similarity and recognition and circuit and (markov or neural or tree) and word	USPT	3	OR	YES		09-07-2017
larri.in. and audio and similarity and recognition and circuit and (markov or neural or tree) and word	PGPB	4	OR	YES		09-07-2017
larri.in. and audio and similarity and recognition and circuit and (markov or neural or tree) and word and integrated near circuit	PGPB	4	OR	YES		09-07-2017
larri.in. and audio and similarity and recognition and circuit and (markov or neural or tree) and word and integrated near circuit and uni-processor	PGPB	4	OR	YES		09-07-2017
front near end and accelerator and search	USPT	3803	OR	YES		10-25-2017
front near end and accelerator and search and distance	USPT	1567	OR	YES		10-25-2017
front near end and accelerator and search and distance and word near identification	USPT	5	OR	YES		10-25-2017
front near end and accelerator and search and distance and word near identification and spoken near words	USPT	3	OR	YES		10-25-2017
front near end and accelerator and search and distance and word near identification and spoken near words and (markov or neural near4 network)	USPT	1	OR	YES		10-25-2017
((front near end and accelerator and search) or programmable) and distance and word near identification and spoken near words and (markov or neural near4 network)	USPT	6	OR	YES		10-25-2017
9076441.pn.	USPT	1	OR	YES		10-25-2017
9076441.pn. and identification or recognition	USPT	373496	OR	YES		10-25-2017
9076441.pn. and (identification or recognition)	USPT	1	OR	YES		10-25-2017
9076441.pn. and (identification or recognition) and markov	USPT	1	OR	YES		10-25-2017
9076441.pn. and (identification or recognition)	PGPB	0	OR	YES		10-25-2017

and markov						
larri.in. and (identification or recognition) and markov	PGPB	4	OR	YES		10-25-2017
larri.in.	USPT	29	OR	YES		04-05-2018
larri.in. and speech near4 recognition	USPT	3	OR	YES		04-05-2018
larri.in. and speech near4 recognition and distance	USPT	3	OR	YES		04-05-2018
larri.in. and speech near4 recognition and distance and (hmm or markov or neural)	USPT	3	OR	YES		04-05-2018
larri.in. and speech near4 recognition and distance and (hmm or markov or neural) and word	USPT	3	OR	YES		04-05-2018
larri.in. and (speech near4 recognition and distance and (hmm or markov or neural) and word).clm.	USPT	0	OR	YES		04-05-2018
larri.in. and (speech near4 recognition and distance and (hmm or markov or neural) and word).clm.	PGPB	0	OR	YES		04-05-2018
larri.in. and (speech near4 recognition and distance and (hmm or markov or neural) and words).clm.	PGPB	0	OR	YES		04-05-2018
larri.in. and (speech near4 recognition and distance and (hmm or markov or neural)).clm.	PGPB	0	OR	YES		04-05-2018
larri.in. and (speech near4 recognition and distance and (hmm or markov or neural)).clm.	USPT	0	OR	YES		04-05-2018
larri.in. and (speech near4 recognition and distance and (hmm or markov or neural)).clm.	PGPB	0	OR	YES		04-05-2018
larri.in. and (speech near4 recognition and distance and (hmm or markov or neural or lexical)).clm.	PGPB	2	OR	YES		04-05-2018
larri.in. and (speech near4 recognition and distance and (hmm or markov or neural or lexical)).clm.	USPT	1	OR	YES		04-05-2018
(speech near4 recognition and distance and (hmm or markov or neural or lexical)).clm.	USPT	80	OR	YES		04-05-2018
(speech near4 recognition and distance and (hmm or markov or neural or lexical) and word).clm.	USPT	43	OR	YES		04-05-2018
(speech near4 recognition and distance and (hmm or markov or neural or lexical) and identify near5 word).clm.	USPT	1	OR	YES		04-05-2018

<i>Index of Claims</i> 	Application/Control No. 14788164	Applicant(s)/Patent Under Reexamination LARRI ET AL.
	Examiner SUSAN MCFADDEN	Art Unit 2658

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

☐ Claims renumbered in the same order as presented by applicant										☐ CPA										☒ T.D.										☐ R.1.47									
CLAIM		DATE																																					
Final	Original	09/26/2016	05/01/2017	09/07/2017	04/06/2018																																		
	1	✓	-	-	-																																		
	2		÷	N	-																																		
	3		÷	N	-																																		
	4		÷	N	-																																		
	5		÷	N	-																																		
	6		÷	N	-																																		
	7		÷	N	-																																		
	8		÷	N	-																																		
	9		÷	N	-																																		
	10		÷	N	-																																		
1	11		÷	✓	=																																		
2	12		÷	✓	=																																		
3	13		÷	✓	=																																		
	14		÷	✓	-																																		
5	15		÷	✓	=																																		
4	16		÷	✓	=																																		
6	17		÷	✓	=																																		

Issue Classification 	Application/Control No. 14788164	Applicant(s)/Patent Under Reexamination LARRI ET AL.
	Examiner SUSAN MCFADDEN	Art Unit 2658

US ORIGINAL CLASSIFICATION						INTERNATIONAL CLASSIFICATION													
CLASS			SUBCLASS			CLAIMED					NON-CLAIMED								
704			238			G	1	0	L	15 / 10 (2006.01.01)									
CROSS REFERENCE(S)																			
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)																		

NONE		Total Claims Allowed:	
		6	
(Assistant Examiner)	(Date)		
/SUSAN MCFADDEN/ Primary Examiner. Art Unit 2658	4-6-18	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	16

Issue Classification 	Application/Control No. 14788164	Applicant(s)/Patent Under Reexamination LARRI ET AL.
	Examiner SUSAN MCFADDEN	Art Unit 2658

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☒ T.D. ☐ R.1.47															
Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	11														
2	12														
3	13														
	14														
5	15														
4	16														
6	17														

NONE		Total Claims Allowed:	
		6	
(Assistant Examiner)	(Date)		
/SUSAN MCFADDEN/ Primary Examiner. Art Unit 2658	4-6-18	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	16

Docket No.: M0025.0369/P369-C
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Guy Larri et al.

Application No.: 14/788,164

Confirmation No.: 1250

Filed: June 30, 2015

Art Unit: 2658

For: DISTRIBUTED PIPELINED PARALLEL
SPEECH RECOGNITION SYSTEM (As
Amended)

Examiner: S. I. McFadden

AMENDMENT AFTER ALLOWANCE UNDER 37 C.F.R. § 1.312

MS Issue Fee
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

INTRODUCTORY COMMENTS

Prior to issuance of the patent, applicant respectfully requests entry on this amendment under 37 C.F.R. 1.312 for the above-captioned patent application.

Amendments to the Title begin on page 2 of this paper.

Remarks/Arguments begin on page 3 of this paper.

200494.00239/107142677

AMENDMENTS TO THE TITLE

Please amend the Title as follows:

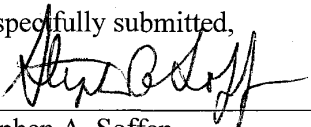
~~SPEECH RECOGNITION CIRCUIT AND METHOD~~ DISTRIBUTED PIPELINED
PARALLEL SPEECH RECOGNITION SYSTEM

REMARKS

The foregoing amendment to the title was included in the Amendment filed on February 28, 2018 in response to the Examiner's statement in the Office Action mailed on September 28, 2017, that the title is not descriptive of the invention. However, it appears from the Notice of Allowance that the title has not been amended in the records of the U.S. Patent and Trademark Office. Accordingly, entry of this Rule 312 Amendment is proper, and is requested.

Dated: April 23, 2018

Respectfully submitted,

By 

Stephen A. Soffen

Registration No.: 31,063

BLANK ROME LLP

1825 Eye Street, NW

Washington, DC 20006-5403

(202) 772-5800

Attorney for Applicant

Electronic Acknowledgement Receipt	
EFS ID:	32417196
Application Number:	14788164
International Application Number:	
Confirmation Number:	1250
Title of Invention:	SPEECH RECOGNITION CIRCUIT AND METHOD
First Named Inventor/Applicant Name:	Guy Larri
Customer Number:	24998
Filer:	Stephen A. Soffen/Cheryl de Lopez
Filer Authorized By:	Stephen A. Soffen
Attorney Docket Number:	M0025.0369/P369-C
Receipt Date:	23-APR-2018
Filing Date:	30-JUN-2015
Time Stamp:	16:40:19
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment		no			
File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		rule312.pdf	48463	yes	3
			5fe6b9c9653841c45f14ddd23d527b72758649c3		

	Multipart Description/PDF files in .zip description		
	Document Description	Start	End
	Amendment after Notice of Allowance (Rule 312)	1	1
	Specification	2	2
	Applicant Arguments/Remarks Made in an Amendment	3	3
Warnings:			
Information:			
Total Files Size (in bytes):		48463	
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>			



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
14/788,164	06/30/2015	2658	730	M0025.0369/P369-C	1	1

CONFIRMATION NO. 1250
CORRECTED FILING RECEIPT



24998
Blank Rome LLP
1825 EYE STREET NW
Washington, DC 20006-5403

Date Mailed: 04/30/2018

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Guy Larri, Cambridge, UNITED KINGDOM;
Mark Catchpole, Prickwillow, UNITED KINGDOM;
Damian Kelly Harris-Dowsett, Coventry, UNITED KINGDOM;
Timothy Brian Reynolds, Cambridge, UNITED KINGDOM;

Applicant(s)

Zentian Limited, Cambridge, UNITED KINGDOM;

Power of Attorney: The patent practitioners associated with Customer Number 24998

Domestic Priority data as claimed by applicant

This application is a CON of 13/735,091 01/07/2013 PAT 9076441
which is a CON of 13/162,128 06/16/2011 PAT 8352262
which is a DIV of 11/662,704 03/14/2007 PAT 7979277
which is a 371 of PCT/GB2005/003554 09/14/2005

Foreign Applications (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.)
UNITED KINGDOM 0420464.0 09/14/2004 No Access Code Provided

Permission to Access Application via Priority Document Exchange: No

Permission to Access Search Results: No

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 07/14/2015

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 14/788,164**

Projected Publication Date: Not Applicable

Non-Publication Request: No

Early Publication Request: No

**** SMALL ENTITY ****

Title

DISTRIBUTED PIPELINED PARALLEL SPEECH RECOGNITION SYSTEM

Preliminary Class

704

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER
Title 35, United States Code, Section 184
Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit <http://www.SelectUSA.gov> or call +1-202-482-6800.

PART B-FEE(S) TRANSMITTAL

Complete and send this form, together with the applicable fee(s), to: Mail

Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22314-1450
(571)-273-2885

or Fax

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note Use Block 1 for any change of address)

24998
BLANK ROME LLP
1825 Eye Street, NW
Washington, DC 20006-5403

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, EFS-Web transmitted, or facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's Name)
(S-signature)
(Date)

APPLICATION NO	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO	CONFIRMATION NO
14/788,164	06/30/2015	Guy Larri	M0025.0369/P369-C	1250

TITLE OF INVENTION: DISTRIBUTED PIPELINED PARALLEL SPEECH RECOGNITION SYSTEM

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	\$500.00			\$500.00	07/23/2018

EXAMINER	ART UNIT	CLASS-SUBCLASS
S. I. McFadden	2658	

1. Change of correspondence address or indication of "Fee Address" (37CFR 1.363).

- ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. Use of a Customer Number is required.

2. For printing on the patent front page, list

- (1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered attorney or agent. If no name is listed, no name will be printed

1. Blank Rome LLP

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

1a. NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Zentian Limited

Cambridge, United Kingdom

Please check the appropriate assignee categories (will not be printed on the patent):

☐ Individual ☒ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:

- ☒ Issue Fee
☐ Publication Fee (No small entity discount permitted)
☐ Advance Order - # of Copies

4b. Payment of Fees(s): (Please first reapply and previously paid fee shown above)

- ☐ A check is enclosed
☒ Payment by credit card. Form PTO-2038 is attached
☒ The director is hereby authorized to charge the required fee(s), any deficiency, or credits any overpayment, to Deposit Account Number 23-2185 (enclose an extra copy of this form)

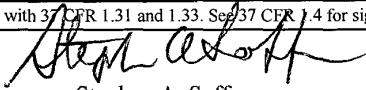
5. Change of Entity Status (from status indicated above)

- ☐ Applicant certifying micro entity status. See 37 CFR 1.29
☐ Applicant asserting small entity status. See 37 CFR 1.27
☐ Applicant changing to regular undiscounted fee status.

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.
NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.
NOTE: Checking this box will be taken to be notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature



Date July 23, 2018

Typed or printed name

Stephen A. Soffen

Registration No. 31,063

PTOL-85 Part B (06-17) Approved for use through 1/31/2020

OMB 0651-0033

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

200494.00239/109692418

Electronic Patent Application Fee Transmittal				
Application Number:		14788164		
Filing Date:		30-Jun-2015		
Title of Invention:		DISTRIBUTED PIPELINED PARALLEL SPEECH RECOGNITION SYSTEM		
First Named Inventor/Applicant Name:		Guy Larri		
Filer:		Stephen A. Soffen/Cheryl de Lopez		
Attorney Docket Number:		M0025.0369/P369-C		
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
UTILITY APPL ISSUE FEE	2501	1	500	500

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				500

Electronic Acknowledgement Receipt	
EFS ID:	33248013
Application Number:	14788164
International Application Number:	
Confirmation Number:	1250
Title of Invention:	DISTRIBUTED PIPELINED PARALLEL SPEECH RECOGNITION SYSTEM
First Named Inventor/Applicant Name:	Guy Larri
Customer Number:	24998
Filer:	Stephen A. Soffen/Cheryl de Lopez
Filer Authorized By:	Stephen A. Soffen
Attorney Docket Number:	M0025.0369/P369-C
Receipt Date:	23-JUL-2018
Filing Date:	30-JUN-2015
Time Stamp:	09:59:28
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$ 500
RAM confirmation Number	072318INTEFSW10002601
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	issuefeep369C.pdf	85838	no	1
			7db5b721c357dbe0b5f8f1a2362bb30c001c66e7		
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30472	no	2
			780e0a469c442270baf04189d0cc07e12ef85212		
Warnings:					
Information:					
Total Files Size (in bytes):			116310		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/788,164	08/28/2018	10062377	M0025.0369/P369-C	1250

24998 7590 08/08/2018

Blank Rome LLP
1825 EYE STREET NW
Washington, DC 20006-5403

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b) (application filed on or after May 29, 2000)

The Patent Term Adjustment is 0 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (<http://pair.uspto.gov>).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site <http://pair.uspto.gov> for additional applicants):

Guy Larri, Cambridge, UNITED KINGDOM;
Zentian Limited, Cambridge, UNITED KINGDOM;
Mark Catchpole, Prickwillow, UNITED KINGDOM;
Damian Kelly Harris-Dowsett, Coventry, UNITED KINGDOM;
Timothy Brian Reynolds, Cambridge, UNITED KINGDOM;

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage and facilitate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re U.S. Patent No.: 10,062,377	)	
	)	
Inventors: Guy LARRI et al.	)	
	)	
Issued Date: August 28, 2018	)	Confirmation No.: 1250
	)	
Application No.: 14/788,164	)	
	)	
For: DISTRIBUTED PIPELINED	)	
PARALLEL SPEECH RECOGNITION	)	
SYSTEM	)	

Commissioner for Patents
Office of Data Management Attention: Certificates of Correction Branch
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

REQUEST FOR CERTIFICATE OF CORRECTION

Pursuant to 35 U.S.C. § 255, and 37 C.F.R. § 1.323, this is a request for a Certificate of Correction in the above-identified patent.

The mistake identified in the appended Form is of a clerical or typographical nature, or of minor character, and resulted from an error made in good faith by patentee(s). The typographical error is that claim 5 recites “second” instead of “third.”

This typographical error made claim 5 nonsensical. Specifically, issued claim 5 recites both (a) that the “first” and “second” devices are “one physical” processor, and (b) that the “second” device is “physically distinct.” Both cannot be true. This error also made dependent claim 6 nonsensical.

The appended Form corrects claim 5 to recite that the “first” and “third” devices are “one physical” processor. This makes sensical the recitation in claim 5 that the “second” device is “physically distinct.” This also makes sensical the further recitations in claim 6.

The fee of \$150, as set forth in 37 C.F.R § 1.20(a), is submitted herewith. Please charge any additional required fees to Deposit Account No. 06-0916.

The complete Certificate of Correction involves one (1) page. Issuance of the Certificate of Correction containing the correction is earnestly requested.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: June 25, 2020

By: /Wei Yuan/
Wei Yuan
Reg. No. 71,772
(202) 408-4103

**UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION**

Page 1 of 1

PATENT NO. : 10,062,377
APPLICATION NO. : 14/788,164
ISSUE DATE : August 28, 2018
INVENTOR(S) : Guy LARRI et al.

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Claim 5, column 39, line 22, "the second programmable device" should read --the third programmable device--.

MAILING ADDRESS OF SENDER (Please do not use Customer Number below):

1

Error! Unknown document property name.

Error! Unknown document property name.

Electronic Acknowledgement Receipt	
EFS ID:	39824217
Application Number:	14788164
International Application Number:	
Confirmation Number:	1250
Title of Invention:	DISTRIBUTED PIPELINED PARALLEL SPEECH RECOGNITION SYSTEM
First Named Inventor/Applicant Name:	Guy Larri
Customer Number:	24998
Filer:	Wei Yuan/Maria Poulakos
Filer Authorized By:	Wei Yuan
Attorney Docket Number:	M0025.0369/P369-C
Receipt Date:	25-JUN-2020
Filing Date:	30-JUN-2015
Time Stamp:	11:21:33
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment		no			
File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Certificate of Correction	15372_0003_As_filed_Request_for_Cert_of_Correction.pdf	16484	no	2
			3ca8f22386e38d6b515b19d6ea556a5844f40d6e		
Warnings:					

Information:					
2	Request for Certificate of Correction	15372_0003_As_filed_Certificate_of_Correction.pdf	22124	no	1
			06bc78a21c2da7e5b00d3e21defda497421ddfb8		
Warnings:					
Information:					
Total Files Size (in bytes):			38608		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 10,062,377 B2
APPLICATION NO. : 14/788164
DATED : August 28, 2018
INVENTOR(S) : Guy Larri et al.

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In the Claims

Claim 5, Column 39, Line 22, "the second programmable device" should read --the third programmable device--.

Signed and Sealed this
Twenty-first Day of July, 2020



Andrei Iancu
Director of the United States Patent and Trademark Office

AO 120 (Rev. 08/10)

TO: Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK
---	---

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court _____ for the Western District of Texas _____ on the following

☐ Trademarks or ☒ Patents. (☐ the patent action involves 35 U.S.C. § 292.);

DOCKET NO. 6:22-cv-00123	DATE FILED 2/2/2022	U.S. DISTRICT COURT for the Western District of Texas
PLAINTIFF Zentian Ltd.		DEFENDANT Amazon.com, Inc.
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,979,277	7/12/2011	Zentian Ltd.
2 7,587,319	9/8/2009	Zentian Ltd.
3 10,971,140	4/6/2021	Zentian Ltd.
4 10,062,377	8/28/2018	Zentian Ltd.
5 10,839,789	11/17/2020	Zentian Ltd.

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY ☐ Amendment ☐ Answer ☐ Cross Bill ☐ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT		
CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
 Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK
---	---

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court _____ for the Western District of Texas _____ on the following

☐ Trademarks or ☒ Patents. (☐ the patent action involves 35 U.S.C. § 292.);

DOCKET NO. 6:22-cv-00122	DATE FILED 2/2/2022	U.S. DISTRICT COURT for the Western District of Texas
PLAINTIFF Zentian Ltd.		DEFENDANT Apple, Inc.
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,979,277	7/12/2011	Zentian Ltd.
2 7,587,319	9/8/2009	Zentian Ltd.
3 10,971,140	4/6/2021	Zentian Ltd.
4 10,062,377	8/28/2018	Zentian Ltd.
5 10,839,789	11/17/2020	Zentian Ltd.

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY ☐ Amendment ☐ Answer ☐ Cross Bill ☐ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT		
CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
 Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy