IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of:	Edward Chang	Attorney Docket No.: 60246-0005IP1
U.S. Patent No.:	7,349,448	
Issue Date:	Mar. 25, 2008	
Appl. Serial No.:	10/632,885	
Filing Date:	Aug. 1, 2003	
Title:	DISTRIBUTED MULTIPLEXING CIRCUIT WITH BUILT-IN REPEATER	

DECLARATION OF DR. ANDREW WOLFE IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF UNITED STATES PATENT NO. 7,349,448

I, Dr. Andrew Wolfe, declare as follows:

I. INTRODUCTION

1. My name is Andrew Wolfe. I have been retained by Fish & Richardson, P.C. on behalf of their client Infineon Technologies Americas Corp. (which I will refer to as Infineon or Petitioner) to investigate and opine on the technical subject matter of this declaration, including the patentability of claims 1-21 of U.S. Patent No. 7,349,448 (the '448 patent).

2. My investigation and opinions are based on my years of education, research, and experience, as well as my review and study of relevant materials. The materials I have reviewed for this declaration include the exhibits I understand are being submitted with Infineon's petition regarding the '448 patent. I may rely on these materials, my knowledge and experience, and any materials submitted by MOSAID (whom I understand claims to own the '448 patent). I may also investigate and consider additional materials, and may further refine my opinions, after reviewing any expert testimony or briefing submitted by MOSAID or to address any questions or concerns posed by the adjudicatory body in this matter.

3. I am being compensated for my time at my standard consulting rate. I am also being reimbursed for any expenses I may incur during the course of this work, including (for example) the cost of traveling to attend deposition. Neither my compensation nor any reimbursements are contingent upon the results of my

study and analysis, the substance of my opinions, or the outcome of any proceeding involving the '448 patent. I am unaware of any financial interest in the outcome of this matter nor in any litigation involving the '448 patent.

II. BACKGROUND AND EXPERIENCE

4. As noted above, the opinions I offer in this declaration are, in part, based on my years of education and experience. I therefore detail that education and experience below.

5. I currently hold a Ph.D. in Computer Engineering from Carnegie Mellon University. My doctoral dissertation proposed a new approach for the architecture of a computer processor (namely, a variable instruction stream computer architecture). Prior to obtaining my Ph.D., I earned a B.S.E.E. degree in Electrical Engineering and Computer Science from The Johns Hopkins University (in 1985); and a M.S. degree in Electrical and Computer Engineering from Carnegie Mellon University (in 1992).

6. I have more than 35 years of experience as a computer architect, computer system designer, personal computer graphics designer, educator, and executive in the electronics industry.

7. In 1983, I began designing touch sensors, microprocessor-based computer systems, and I/O (input/output) cards for personal computers as a senior design engineer for Touch Technology, Inc. During the course of my design projects

with Touch Technology, I designed I/O cards for PC-compatible computer systems, including the IBM PC-AT, to interface with interactive touch-based computer terminals that I designed for use in public information systems. I continued designing and developing related technology as a consultant to the Carroll Touch division of AMP, Inc., where in 1986, I designed one of the first custom touch-screen integrated circuits. I designed the touch/pen input system for the Linus WriteTop, which many believe to be the first commercial tablet computer.

8. From 1986 through 1987, I designed and built a high-performance computer system as a student at Carnegie Mellon University. From 1986 through early 1988, I also developed the curriculum and supervised the teaching laboratory for processor design courses.

9. In the latter part of 1989, I worked as a senior design engineer for ESL- TRW Advanced Technology Division. While at ESL-TRW, I designed and built a bus interface and memory controller for a workstation-based computer system, and also worked on the design of a multiprocessor system.

10. At the end of 1989, I (along with some partners) reacquired the rights to the technology I had developed at Touch Technology and at AMP and founded The Graphics Technology Company. Over the next seven years, as an officer and a

consultant for The Graphics Technology Company, I managed the company's engineering development activities and personally developed dozens of touch screen sensors, controllers, and interactive touch-based computer systems.

11. I have consulted, formally and informally, for a number of fabless semiconductor companies. In particular, I have served on the technical advisory boards for two processor design companies: BOPS, Inc., where I chaired the board; and Siroyan Ltd., where I served in a similar role for three networking chip companies—Intellon, Inc., Comsilica, Inc., and Entridia, Inc.—and one 3D game accelerator company, Ageia, Inc.

12. I have also served as a technology advisor to Motorola and to several venture capital funds in the U.S. and Europe. Currently, I am a director of Turtle Beach Corporation, providing guidance in its development of premium audio peripheral devices for a variety of commercial electronic products.

13. From 1991 through 1997, I served on the Faculty of Princeton University as an Assistant Professor of Electrical Engineering. At Princeton, I taught undergraduate and graduate-level courses in Computer Architecture, Advanced Computer Architecture, Display Technology, and Microprocessor Systems, and conducted sponsored research in the area of computer systems and related topics. I was also a principal investigator for DOD research in video technology and a principal investigator for the New Jersey Center for Multimedia Research. From 1999

through 2002, while a Consulting Professor, I taught a Computer Architecture course to both undergraduate and graduate students at Stanford University. At Princeton, I received several teaching awards, both from students and from the School of Engineering. I have also taught advanced microprocessor architecture to industry professionals in seminars sponsored by the Institute of Electrical and Electronics Engineers ("IEEE") and the Association for Computing Machinery teaching courses on Microprocessor Systems, Integrated Circuit ("IC") Design, PCB Design, Real-Time Computing, and Mechatronics("ACM"). I am currently an Assistant Teaching Professor at Santa Clara University.

14. From 1997 through 2002, I held a variety of executive positions at a publicly-held fabless semiconductor company originally called S3, Inc. and later called SonicBlue Inc. I held the positions of Chief Technology Officer, Vice President of Systems Integration Products, Senior Vice President of Business Development, and Director of Technology, among others. At the time I joined S3, the company supplied graphics accelerators for more than 50% of the PCs sold in the United States. At S3, I supervised the design of several PC graphics accelerators. During my time at SonicBlue we launched more than 30 new consumer electronics products including devices to support copy-protected video and many of the first commercial products to support copy-protected internet audio content. These in-

cluded some of the first consumer products to support playback of encrypted content using digital rights management (DRM) based security such as the Rio 600 portable music player. I also worked with Universal Music Group and Sony Entertainment on the development of systems for distribution and management of secure, encrypted content including user-specific licensed downloads and streaming. I also worked on the development of handheld PDA devices, portable wireless tablets, portable CD players, and handheld video players.

15. I have published more than fifty peer-reviewed papers in computer architecture and computer systems and IC design. I also have chaired IEEE and ACM conferences in microarchitecture and integrated circuit design and served as an associate editor for IEEE and ACM journals. I served on the IEEE Computer Society Awards committee. I am an IEEE Fellow, an IEEE Computer Society Distinguished Contributor, and a Member of ACM. I am a named inventor on at least fifty-seven U.S. patents and thirty-seven foreign patents, which are listed in my curriculum vitae. Some of these patents relate to encryption systems.

16. In 2002, I was the invited keynote speaker at the ACM/IEEE International Symposium on Microarchitecture and at the International Conference on Multimedia. From 1990 through 2005, I have also been an invited speaker on various aspects of technology and the PC industry at numerous industry events includ-

ing the Intel Developer's Forum, Microsoft Windows Hardware Engineering Conference, Microprocessor Forum, Embedded Systems Conference, Comdex, and Consumer Electronics Show, as well as at the Harvard Business School and the University of Illinois Law School. I have been interviewed on subjects related to computer graphics and video technology and the electronics industry by publications such as the Wall Street Journal, New York Times, Los Angeles Times, Time, Newsweek, Forbes, and Fortune as well as on CNN, NPR, and the BBC. I have also spoken at dozens of universities including MIT, Stanford, University of Texas, Carnegie Mellon University, UCLA, University of Michigan, Rice University, and Duke University.

17. I also have specific experience in the design of multiplexors. Multiplexers are used in a variety of circuits within processors and memory systems. I have utilized, designed, analyzed, implemented, and taught about multiplexers for decades, both at the chip and system level. I have published peer-reviewed papers about multiplexers, and they are discussed in some of my issued patents.

18. I have not attempted to detail every aspect of my education and my experience—I only provide a representative narrative sufficient to qualify myself as an expert in the field of electrical engineering and, in particular, the subject matter of the '448 patent. I have attached my full CV to this declaration.

III. MY UNDERSTANDING OF THE RELEVANT LEGAL PRINCIPLES

19. In forming my analysis and conclusions expressed in this Declaration, I have applied the legal principles described in the following paragraphs, which were provided to me by counsel for the Petitioner.

A. Anticipation

20. It is my understanding that invalidation by "anticipation" exists only if a single alleged prior art reference discloses each and every limitation of the claim at issue, either expressly or inherently. In other words, every limitation of the claim must appear in a single prior art reference for the reference to anticipate that claim. I also understand that all elements of the claim must be disclosed in the reference as they are arranged in the claim. I also understand that to be considered anticipatory, the prior art reference must be enabling and must describe the patentee's claimed invention sufficiently to have placed it in the possession of a person of ordinary skill in the field of invention. Under the principles of inherency, I understand that if the prior art necessarily functions in accordance with or includes the claimed limitations, it anticipates.

21. I have been informed that a claim is invalid if the claimed invention was known or used by others in the U.S., or was patented or published anywhere, before the Applicant's invention. I further have been informed that a claim is invalid if the invention was patented or published anywhere, or was in public use, on

sale, or offered for sale in this country, more than one year prior to the filing date of the patent application (critical date). I have also been informed that a claim is invalid if an invention described by that claim was described in a U.S. patent granted on an application for a patent (or in a published application for a U.S. patent) that was filed by another in the U.S. before the date of invention for such a claim.

22. I understand that a dependent claim contains all of the limitations of the independent claim from which it depends. Thus, to establish anticipation, every limitation set forth in a dependent claim as well as the independent claim from which it depends must be found in an anticipatory reference.

B. Obviousness

23. I have been informed that a patent claim is invalid as "obvious" in light of one or more prior art references if it would have been obvious to a person of ordinary skill in the art at the time of the alleged invention ("POSITA"), taking into account (1) the scope and content of the prior art, (2) the differences between the prior art and the claims, (3) the level of ordinary skill in the art, and (4) any so called "secondary considerations" of non-obviousness, which include: (i) "long felt need" for the claimed invention, (ii) commercial success attributable to the claimed invention, (iii) unexpected results of the claimed invention, and (iv) "copying" of the claimed invention by others. I understand that a patent cannot be properly granted for subject matter that would have been obvious to a POSITA at the time

of the alleged invention, and that a patent claim directed to such obvious subject matter is invalid under 35 U.S.C. § 103. It is also my understanding that in assessing the obviousness of claimed subject matter one should evaluate obviousness over the prior art from the perspective of a POSITA at the time the invention was made (and not from the perspective of either a layman or a genius in that art).

24. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. To be obvious in light of a single prior art reference or multiple prior art references, there must be a reason that would have prompted (or "motivated") a POSITA to modify the single prior art reference, or combine two or more references, in a manner that provides the elements of the claimed invention. This reason may come from a teaching, suggestion, or motivation to combine, or may come from the reference(s) themselves, the knowledge or "common sense" of a POSITA, or from the nature of the problem to be solved, and this reason may be explicit or implicit from the prior art as a whole. I have been informed that, under the law, the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. I also understand it is improper to rely on hindsight in making the obviousness determination.

25. I understand that an obviousness determination also requires that a POSITA would have had a reasonable expectation of success. This concept has

been explained to me as relating to the relative likelihood or predictability—from the perspective of a POSITA—of successfully obtaining a result within the scope of the challenged claim(s) based on the combined, modified, and/or suggested prior art teachings at hand. I understand that the expectation of success need only be "reasonable" and, thus, does not require the absolute certainty gleaned from physically creating the prior art teachings.

C. Claim Construction

26. I understand that, for purposes of my analysis in this IPR proceeding, the terms appearing in the patent claims should be interpreted according to their "ordinary and customary meaning." In determining the ordinary and customary meaning, the words of a claim are first given their plain meaning that those words would have had to a POSITA. I understand that the structure of the claims, the specification, and the file history may be used to better understand a claim. Moreover, treatises and dictionaries may be used, albeit under limited circumstances, to determine the meaning attributed by a person of ordinary skill in the art to a claim term at the relevant time of the challenged patent (i.e., the Critical Date). I have followed this approach in my analysis.

IV. BACKGROUND KNOWLEDGE OF A POSITA

27. The '448 patent discloses and claims a purported invention in the field of digital circuit design. Therefore, before diving into a detailed discussion of the

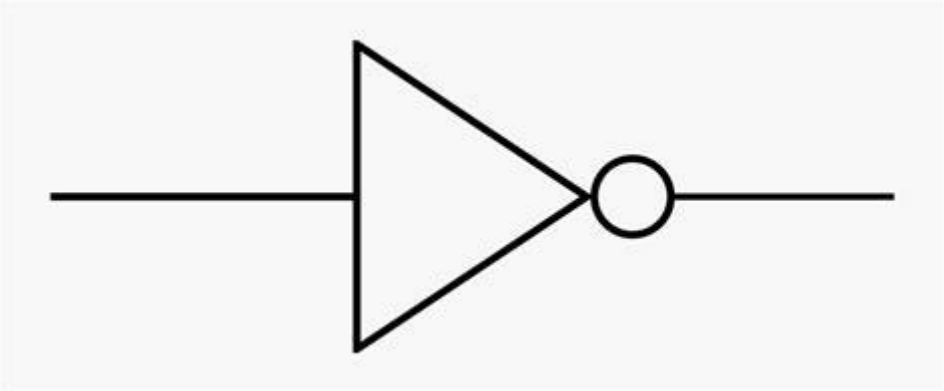
’448 patent and/or any prior art to the ’448 patent, I provide some brief background regarding digital circuit design, including some of the components discussed in the ’448 patent and the prior art.

28. A “digital circuit” is a circuit that processes and manipulates “digital” signals. The difference between an analog circuit and a digital circuit relates generally to the type of signal being processed and/or manipulated. While an analog signal (for example, the audio signal sent along a wire to a set of speakers or headphones) may have infinite natures and forms; a digital signal only has a finite set of values (and technically is only valid at discrete times). For example, and most typically, a digital signal will have two independent values – one value representing “0” and another value representing “1.”

29. The field of digital circuit design was well-developed at the time of the ’448 patent. Indeed, throughout this declaration, I cite to two textbooks (including one ***second edition***) that were widely available and used to educate engineers at the time of the ’448 patent. According to one of those textbooks—one designed for undergraduate study, I might add—“[h]aving an understanding of digital integrated circuits (at least to a moderate degree) is becoming almost essential for an engineer doing system design” because “practically every modern system contains digital integrated circuits as critical subcomponents.” EX1010 at 1.

30. The basic building block of digital circuit design is the transistor. A transistor is a three-terminal semiconductor device that with three terminals and serves as the physical building block for most digital circuits. Transistors can be arranged to create different components that can manipulate the current or voltage passed through them in order to perform logic functions.

31. Transistors and other physical components (such as resistors) can be arranged in certain ways to perform certain arithmetic functions. Digital circuits are often shown not in terms of the physical design of the transistors used to build them but in terms of the logic performed on the signals. For example, shown below is a symbol for an inverter – a digital circuit design component that will "invert" the signal:



If the input into this component is a "1" – then the output will be a "zero." If the input into this component is a "0," then the output will be a "1". *E.g.* EX1010, 2. Additional basic logic functions might include an AND gate (which outputs a 1 when all of its inputs are one; and outputs a 0 if any of the inputs are zero); an OR

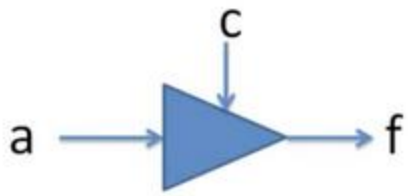
gate (which outputs a 1 if any of the inputs or one), and other similar gates to represent the basic logic functions.

32. These logical functions are also illustrated by truth-tables, which indicate an output based on an input or set of inputs. For example, a truth table for an "AND" gate is shown below:

A (Input 1)	B (Input 2)	X = (A.B)
0	0	0
0	1	0
1	0	0
1	1	1

As discussed in the last paragraph, an AND gate only outputs a "1" when both inputs are "1." This is shown in the truth table.

33. Components that could perform more complex logic functions were also well-known. One such component discussed in the '448 patent and the prior art cited herein is a "tri-state driver" (also called a "tri-state buffer"). The typical symbol used to indicate a tri-state buffer, as well as a truth table associated with a tri-state buffer, are excerpted below:



c	a	f
0	0	Z
0	1	Z
1	0	0
1	1	1

In the above truth table, the "Z" represents a high-impedance state which means little to no current will pass through the tri-state driver. In other words, in addition to the traditional outputs of 1 and 0, a tri-state driver can be turned, in essence, off given a certain input—hence the term "tri-state."

34. In addition to logic gates, memory components (i.e. storage elements) were also well-known in the field of digital circuit design. Indeed, in order to design a synchronous circuit—i.e. one that works according to a clock—memory components are necessary to ensure that operations occur synchronously within the circuit.

35. The most basic storage element in digital circuit design is a "latch"—a component discussed both in the '448 patent and in the prior art. "A latch consists of a single memory location with some access circuitry." EX1010, 284. Multiple

latches can be arranged together to create more complex storage circuits. For example, "[a] flip-flop in an IC typically consists of two cascaded latches and perhaps some additional input logic."[1] EX1010, 284.

36. In my opinion, a POSITA relating to the subject matter of the '448 patent as of the critical date of the '448 patent (August 1, 2003) would have possessed at least a bachelor's degree in electrical engineering (or equivalent degree) and three years of work experience in the field of digital circuit design including the use of multiplexors. Additional work experience can make up for less education; and vice versa. And, as discussed above, this POSITA would necessarily have a significant background in the field of digital design, including of each of the components used in digital design (and their connections) discussed above. However, my goal in this section is not to provide a fulsome inventory of everything that would have been known to a POSITA, nor provide anything close to an introductory education concerning digital circuit design. Rather, I hope to introduce a few topics and some background that will provide context and allow greater understanding of the discussion that follows concerning the '448 patent and the prior art.

[1] Other forms of flip flops were known at the time but for purposes of this inquiry are equivalent.

V. THE '448 PATENT

A. Critical Date

37. I am informed that the critical date for the '448 patent is August 1, 2003 – the day the application that led to the '448 patent was filed. I further understand that, to qualify as prior art, the patents and printed publications I discuss herein must have been filed, patented, and/or published before that date. Based on my review of those materials and, as I am informed by counsel, each of the prior art references I discuss in this declaration quality as prior art to the '448 patent.

B. Brief Description

38. The '448 Patent generally relates to "systems for distributing multiplexor functionality." EX1001, 1:52-53.

39. As explained in the patent, a "standard" multiplexor ("MUX") "may comprise eight data connections that transmit 128-bit data words." EX1001, 1:6-12. "Typically, each of the 128-bit data words may be routed to multiplexor logic on the [integrated circuit] via a connection that comprises 128 routing wires, e.g., a wire for each bit that is transmitted." EX1001, 1:12-15. This results in 1024 routing wires on a single integrated circuit (IC) and can lead "to routing congestion and the inherent problems associated therewith, e.g., difficulty in designing and fabricating." EX1001, 1:16-23. Referring to the "conventional MUX circuit 100," the '448 Patent describes that "[t]he multiplexor logic 110 receives the data words and, at any given time, selects one of the data words on one of the connections 112-115

based on a control signal from MUX control logic 109." EX1001, 1:30-33. The significant number of data wires on the single IC only increases as the number of selected data words are added. EX1001, 1:35-38.

40. In order to address these alleged problems, the '448 patent purports to provide "a distributed multiplexor circuit select between multiple data connections." EX1001, 2:33-34. One example embodiment is illustrated in FIG. 2 (reproduced below). EX1001, 2:65-67. In FIG. 2, the distributed multiplexor logic (DML) "comprises a plurality of stages 204-207, preferably corresponding to the number of data connections, from which a selected data word is chosen and transmitted via connection 226." EX1001, 3:8-11. In the example of FIG. 2, each data word is connected to a corresponding DML stage. *See* EX1001, 3:21-35. Each DML stage is then connected to select logic 202 that "indicate which of the data words from data connection 214a-214d is the 'selected data word' that is to be output from the DML 201 and transmitted, via connection 226, to a system bus 224." EX1001, 3:39-43.

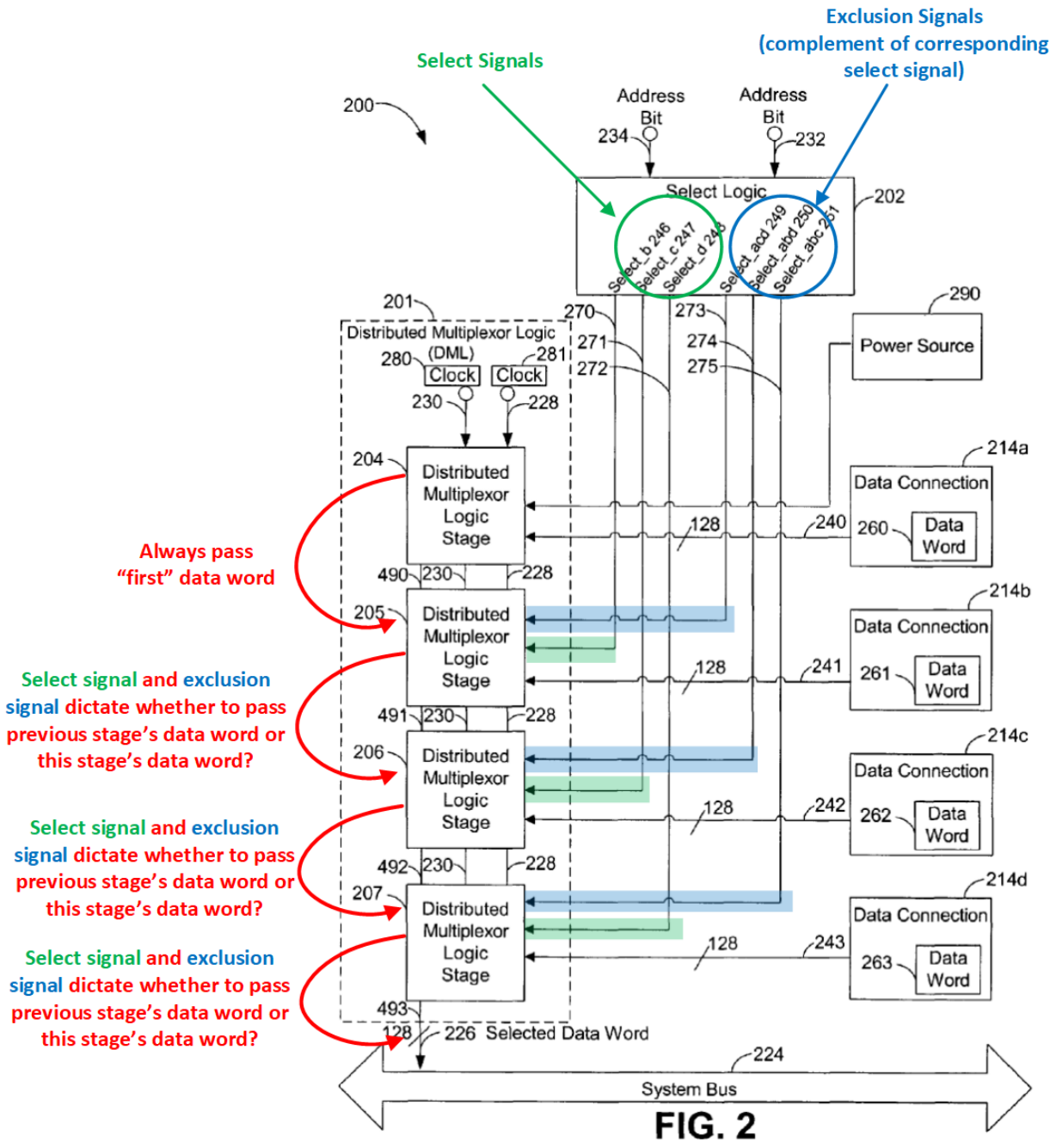


FIG. 2

Annotated FIG. 2 of '448 Patent

41. A subset of the select logic is connected to each DML stage other than the first (*e.g.*, DML stage 204 in FIG. 2), and indicates whether the word associated with that DML stage is being selected or not. *See* EX1001, 3:58-4:5. Each

DML stage other than the first receives a "select signal" and an "exclusion signal." EX1001, 3:64-4:9. When the select logic is driven to select a data word corresponding to a DML stage, that DML stage's select bit is asserted while its exclusion signal is deasserted, and all other DML stage's select bits are deasserted while their exclusion signal is asserted. *See generally* EX1001, 3:64-4:23, 4:66-5:22. Thus, in the example of FIG. 2, the exclusion signal for each DML stage is the complement (*i.e.*, opposite) of the select signal of that stage. *See id.*

> For example, if address bits 234 and 232 indicate that the data word associated with data connection 214b is desired, then the select logic 202 asserts the select signal select b 246, which is received by stage 205 of the DML 201, e.g., the stage associated with data connection 214b. Further, select logic 202 deasserts signals select c 247 and select d 248, which are received by stage 206 and stage 207, respectively. Select logic 202 also asserts select abd 250 and select abc 251, which are transmitted via connections 274 and 275 to stage 206 and stage 207. Further, select logic 202 deasserts data connection exclusion signals select acd 249, which is transmitted via connection 273 to stage 205.

EX1001, 4:24-35. Based on the value of the select signal and exclusion signal it receives, each DML stage other than the first "selects from a data word 260-263 transmitted by the previous stage 204-206 and the data word 261-263 received via data connections 240-243." *See* EX1001, 5:10-23.

42. FIG. 4 (reproduced below), shows in more detail the components of each DML stage. *See* EX1001, 7:15-20. "Each DML stage 204-207, preferably comprises data connection selection tristate drivers 408-411, each of which receives and outputs its associated data word 260-263 to its associated connection 490-493, respectively, based upon select signals 246-251." EX1001, 7:16-20. Except for the first DML stage, the other DML stages include two tristate drivers: a data connection selection tristate driver and a data connection exclusion tristate driver. *See* EX1001, 7:21-48.

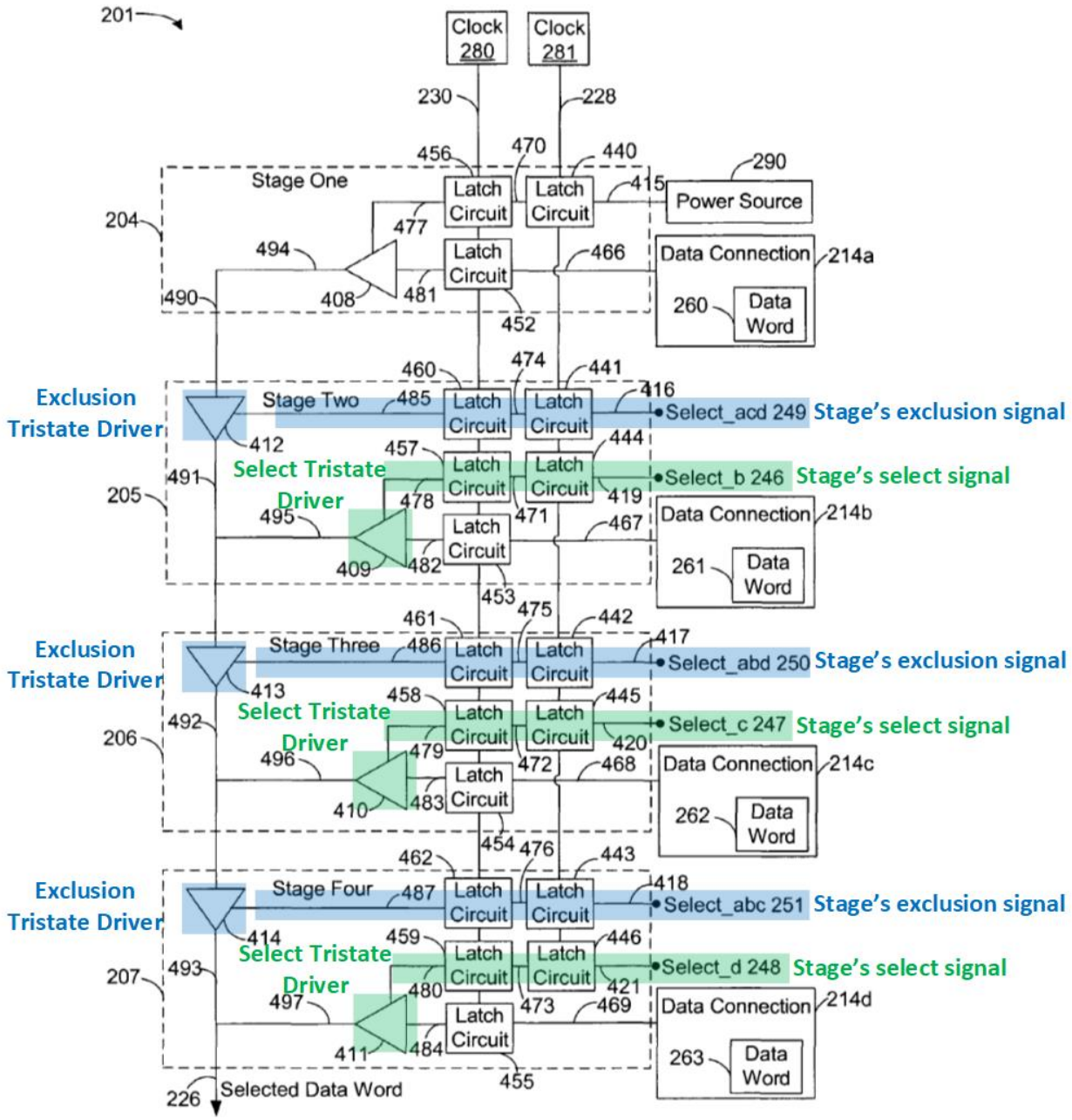


FIG. 4

Annotated FIG. 4 of '448 Patent

43. When the select logic is driven to select a data word corresponding to a DML stage, that DML stage's data connection selection tristate driver is activated by the select signal corresponding to the DML stage and the data connection

exclusion tristate driver is deactivated by the exclusion signal corresponding to the DML stage. *See* EX1001, 7:49-62. The other two-tristate-driver DML stages' data connection selection tristate drivers are deactivated by the corresponding select signal and the data connection exclusion tristate drivers are activated by the corresponding exclusion signal. *See* EX1001, 7:49-62. "In propagating the data words 260-263 from the first stage to the final stage of the circuit 201, the tristate drivers 412-414 located between each stage 204-207 serve to increase the signal strength of the data word propagated from the previous stage, thereby acting as repeaters." EX1001, 9:63-10:3.

44. As shown in FIG. 4 (above), each stage includes latch circuits that are driven by out-of-phase clock signals. *See generally* EX1001, 8:1-54. The latch circuits support "[c]oncurrent enablement and disablement of the tristate drivers 408-411 and 412-414" in order to "limit the amount of 'drive fight,' e.g., the time that more than one driver is attempting to drive a connection 490-493 to a different data word 260-263." EX1001, 10:4-15.

45. The '448 Patent identifies at least two prior art solutions to the same problems addressed by its distributed MUX. EX1001, 1:39-43. According to the '448 Patent, the first is a "dynamic MUX," which allegedly "requires more power, has smaller noise margins, and takes more time to design correctly." EX1001, 1:39-43. The second prior art solution is "usage of tristate multiplexors," which

are "acceptable when the distances between the data connections 102-105 and the multiplexor logic 110 are small." EX1001, 1:43-48. While the '448 Patent suggests that, for tristate multiplexors, "speed and noise degradation typically increases as the distances between the data connections increase" (EX1001, 1:46-48), I do not discern anything in the claims of the '448 Patent that would specify the distance between data connections. Moreover, the DML stages of the '448 Patent rely upon tristate drivers—which were asserted as the point of novelty. And, as discussed further below, the Challenged Claims are broad enough to cover solutions based on tristate multiplexors, which were replete in the prior art.

C. Prosecution History

46. In addition to the specification, I have also reviewed the prosecution history of the '448 patent. I summarize below.

47. Originally filed claims 1-3 and 14-17 were initially rejected in a first office action based on U.S. Pat. No. 7,023,844 (Wolrich). EX1002, p. 49. Claims 4-7 and 18-20 were indicated as allowable, because the prior art reviewed by the Examiner allegedly "fail[ed] to teach the Multiplexer circuit comprising first and second tri state driver that enable if a data connection exclusion signal is asserted." EX1002, p. 50.

48. In response to the first office action, the Applicant amended rejected independent claim 1, in part, to recite: "the second stage logic having a connection,

a second tristate driver, and a third tristate driver, the second tristate driver having an output electrically coupled to an output of the third tristate driver, the second tri-state driver configured to receive the first data word from the first tristate driver and to drive the connection of the second stage logic with the first data word if the first data word is selected for transmission by the second stage logic, the third tri-state driver configured to receive the second data word and to drive the connection of the second stage logic with the second data word if the second data word is selected for transmission by the second stage logic." EX1002, pp. 35-36. With respect to the Examiner's rejection, Applicant argued that "Wolrich appears to disclose a distributed arrangement of switches," but "fails to disclose at least an arrangement of 'tristate drivers,' as recited by claim 1." EX1002, p. 36. In other words, according to the Applicant, the use of 'tri-state drivers' was what made its invention novel (though—as noted above, they were well known).

49. Applicant also amended rejected independent claim 15 (issued independent claim 14), in part, to recite: "driving a first connection with the first data word via the second tristate driver if the first data word is selected by the selecting; and driving the first connection with the second data word via a third tristate driver if the second data word is selected by the selecting." Similar to independent claim

1, Applicant argued that "Wolrich appears to disclose a method of using a distributed arrangement of switches," but "fails to disclose at least that 'tristate drivers' are used to drive a connection as recited by claim 15." EX1002, p. 37.

50. The Examiner subsequently issued a Notice of Allowance without providing any explanation regarding the reasons for allowance. *See* EX1002, pp. 12-23. However, in allowing the claims, the Office never considered Pixley, which is the basis for Ground 1 and which, as described below, discloses the features that led to allowance. EX1002, *generally*.

VI. OVERVIEW OF PRIOR ART

A. Pixley

51. Pixley relates to "tri-state multiplexers in a circuit design." EX1004, 1:8-10. "A tri-state multiplexer provides faster performance and consumes less chip area and power, over a gate-implemented multiplexer." EX1004, 2:63-65. "However, a circuit with a tri-state multiplexer may generate an undesirable situation if not properly designed." EX1004, 2:65-67. For example, "[i]f two or more select lines are enabled with data inputs driving different logic values, the circuit may show unintended functional behavior, consume excessive power, and may be damaged." EX1004, 2:67-3:3. Accordingly, Pixley "provides a method and apparatus to perform checking of tri-state multiplexers to ensure that the tri-state multiplexers are operating in a proper manner." EX1004, 2:60-63.

52. To demonstrate the functionality of its invention, Pixley provides FIGS. 5-11, which are diagrams of "typical" tri-state multiplexer (MUX) circuits that "could be located in the circuit model 37 of FIG. 4," and checked for proper operation by Pixley's tri-state multiplexer checker 38. *See* EX1004, 2:24-45, 4:41-65. For example, "FIG. 9 and FIG. 10 illustrate how nested multiplexers are processed." EX1004, 7:7-8.

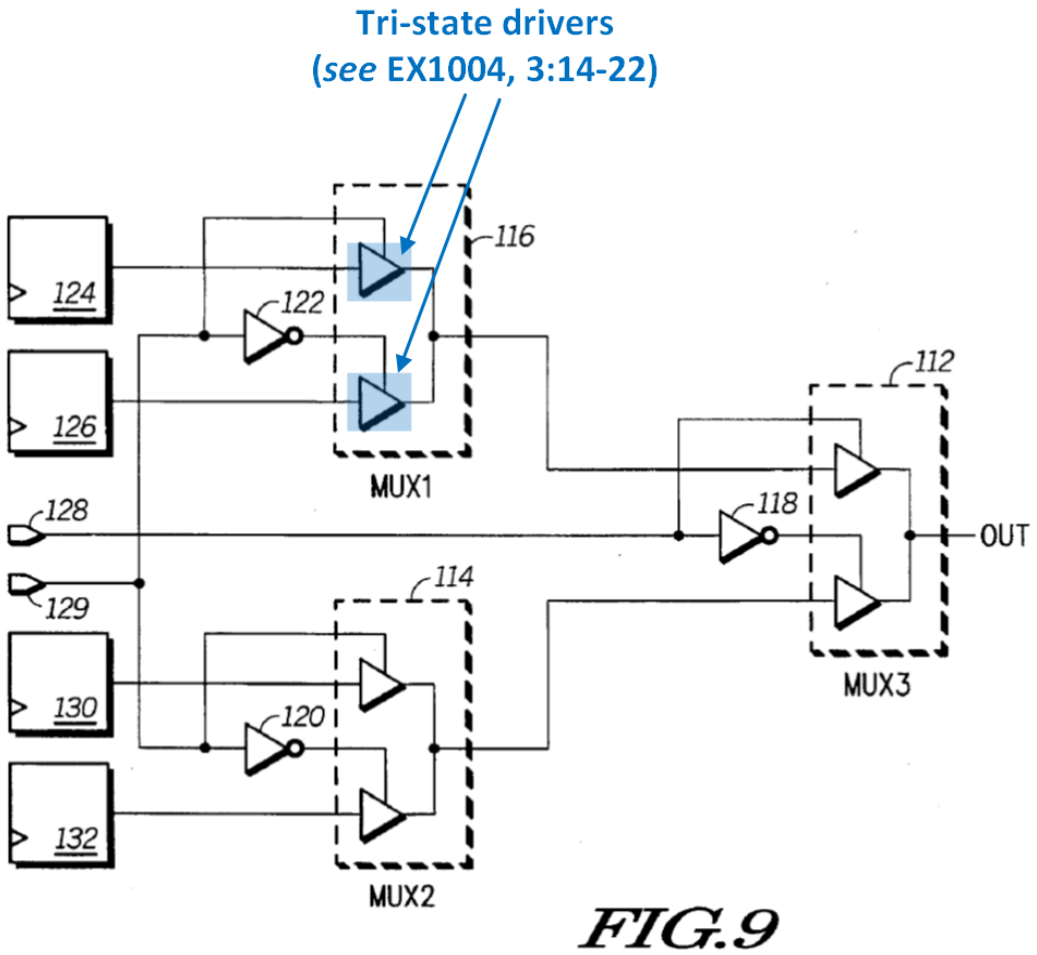


FIG.9

Annotated FIG. 9 of Pixley

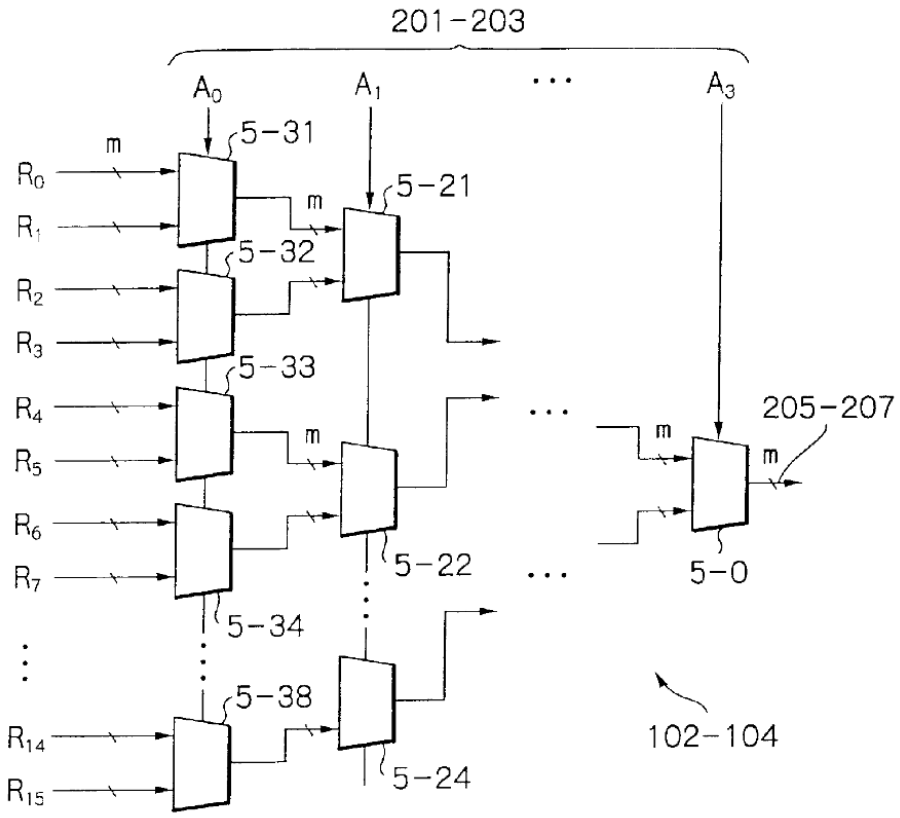
53. "FIG. 9 illustrates a circuit having tri-state MUXes 112, 114, and 116." EX1004, 7:8-9. Each of these MUXes includes two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexers). "Each driver [] is enabled only if the corresponding select line (sel-#) is enabled." EX1004, 3:18-19. "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22. In the example of FIG. 9, the select line of the upper tri-state driver of MUX 116 is driven by input terminal/pin 129 and the select line of the lower tri-state driver of MUX 116 is driven by the complement (*i.e.*, inverse) of the input terminal/pin 129. *See* EX1004, FIG. 9, 7:7-14. According to Pixley, the tri-state multiplexer checker 38 would verify the example tri-state MUX design of FIG. 9, because it meets the required conditions. *See* EX1004, 7:14-16.

B. Watanabe

54. Watanabe relates generally to "a register read circuit and a microprocess, which is applicable to, for example, reading out data from a plurality of registers of a register bank." EX1005, [2]. Watanabe claims and discloses improving these registers by "provid[ing] a register read circuit with a structure suitable for implementation, and easy for installation, on an integrated circuit chip." EX1005, [20].

55. To disclose this improvement, though, Watanabe also includes a discussion concerning the then state-of-the-art concerning multiplexors. For example, Watanabe explains that it was known that "[a] 2-to-1 multiplexor of m-bit length is composed of m 2-1 multiplexers of one-bit length." EX1005, [18]. Watanabe also discusses a known "method in which one 16-to-1 multiplexors is used for one operand" and includes a discussion of the known design for such a multiplexors, such as illustrated in Figure 3 from Watanabe, excerpted below:

Fig. 3 PRIOR ART



EX1005, [33], Fig. 3. As illustrated in Figure 3 and explained by Watanabe, the disclosed 16-1 multiplexor is comprised of eight 2-to1 multiplexors 5-31 through 5-38 arranged in a nested pattern. EX1005, [36]-[37], Fig. 3.

C. Additional Prior Art

56. In addition to Pixley and Watanabe, I cite additional prior art to the '448 patent. These additional materials are not cited or discussed to fill any gaps in the disclosures of the disclosed references. Rather, these additional references are cited for the purposes of illustrating how a skilled artisan would understand the teachings of Pixley and/or Watanabe; as well as evidencing the knowledge, motivations, and skill of a POSITA.

VII. ANALYSIS OF CLAIMS IN LIGHT OF THE PRIOR ART

1. Claim 1

[1.pre] A circuit for multiplexing a plurality of data words including a first data word and a second data word, comprising:

57. In my opinion, a combination of Pixley and Watanabe (hereinafter "Pixley-Watanabe") renders the preamble[2] of claim 1 obvious.

58. Pixley relates to "tri-state multiplexers in a circuit design." EX1004, 1:8-10. Pixley includes several "typical" tri-state multiplexer (MUX) circuits in order to demonstrate the operation of its tri-state multiplexer checker. *See*

[2] For this and all claims in the '448 patent, I have not been asked to analyze whether the preambles are limiting.

EX1004, 2:24-45, 4:41-65. For example, "FIG. 9 and FIG. 10 illustrate how nested multiplexers are processed." EX1004, 7:7-8. FIG. 9 includes three 2-to-1 MUXs 112, 114, and 116. *See* EX1004, FIG. 9, 7:11-14. That is, each multiplexer receives two data inputs (*e.g.*, the output of D flip-flops 124 and 126 provide data inputs to tri-state multiplexer 116) and selects one of the two data inputs for its output. *See* EX1004, FIG. 9, 7:11-14.

59. In my opinion, Pixley's disclosure is alone sufficient to teach multiplexing "data words" as that term is used in the '448 patent. The '448 Patent describes that "[e]ach data connection is preferably capable of transmitting an n-bit data word, where 'n' refers to the number of bits within the data word." EX1001, 2:36-38. The '448 Patent goes on to state that "[a] set of bits, e.g. 128 bits simultaneously transmitted from the same data connection, is hereinafter referred to as a 'data word.'" EX1001, 2:41-43. The '448 Patent does not preclude a "data word" from being one bit.

60. Nonetheless, even if it did, a POSITA would have at least found it obvious that Pixley's multiplexer would operate with respect to multi-bit data words. For example, a POSITA would have understood or at least found obvious that the 2-to-1 multiplexers would have been configured to multiplex m-bit data words. Indeed, at the time of the '448 patent, most processors were larger than 1-bit processors, and any multiplexing circuits used with such processors would need to be

able to process multi-bit data words. Moreover, it was known that "[a] 2-to-1 multiplexer of m-bit length is composed of m 2-to-1 multiplexers of one-bit length." *See, e.g.*, EX1005, [0018].

61. Further, a POSITA would know that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037]. Because it would have been obvious that each of the connections in such a configuration of Pixley's multiplexor would have thus transmitted a "set of bits," those connections would be transmitting "data words," even if "data words" precludes 1-bit data words. *See* EX1001, 2:41-43. One known use case of such a configuration was the selection and transmission of an operand—which Watanabe describes as a "word"—from a selected register in a register bank for transmission to and use in an arithmetic and logical unit (ALU). *See* EX1005, FIG. 4, [0006] ("This is because the instruction word is also 32 bits or 64 bits long on such a processor and this word length is long enough for specifying the operands"), [0043].

62. Thus, a POSITA would have understood or at least found obvious that, in m-bit processing applications in which m-bit data must, for example, be read from registers or memory, each of Pixley's tri-state multiplexors 112, 114, and 116 would have been configured as an m-bit multiplexer composed of m one-

bit multiplexers, one for each bit of data—effectively stacking a set of the multiplexer circuits illustrated in FIG. 9. A POSITA would have been motivated to use m-bit multiplexors in m-bit processing applications as such applications demand the ability to process m-bit data words.

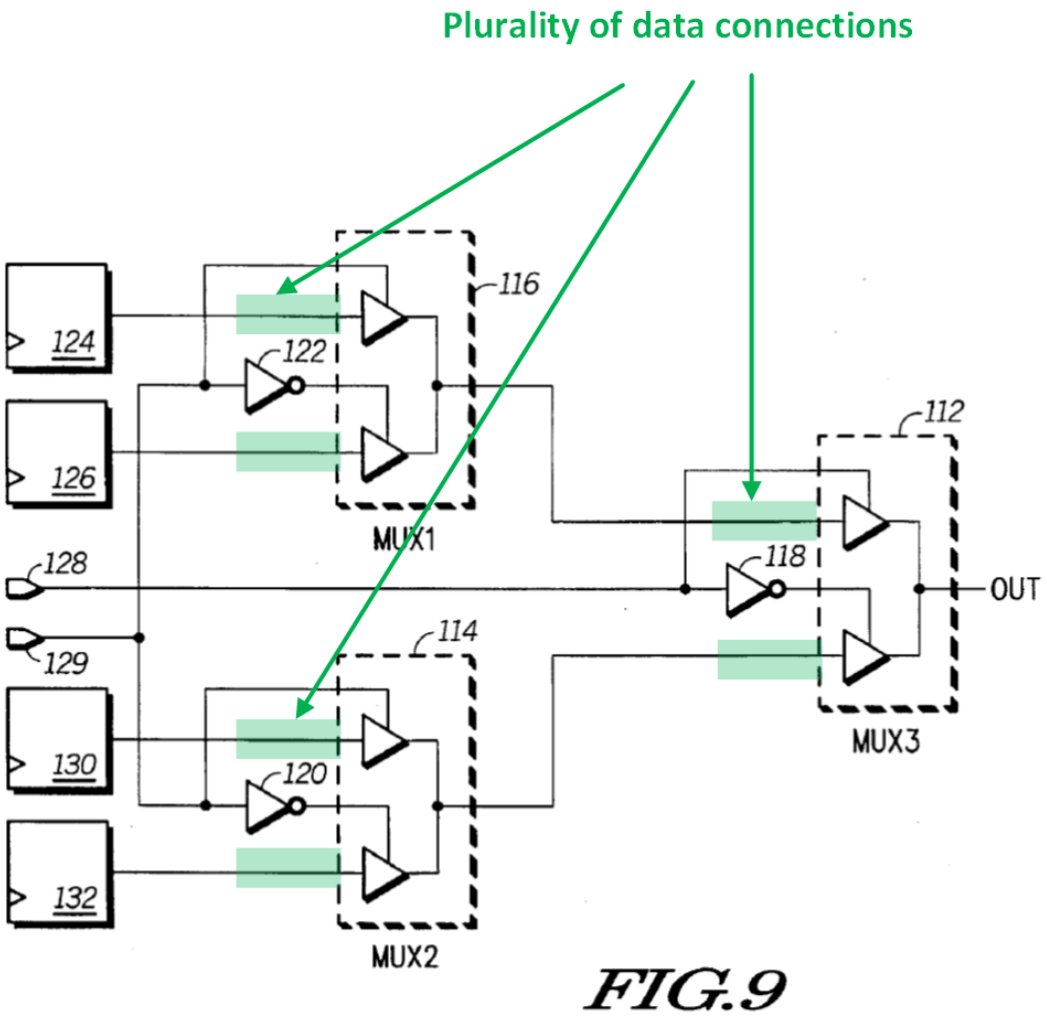
63. Implemented in this way, a POSITA would have understood and found obvious that Pixley-Watanabe provides ***a circuit*** [*e.g.*, the nested multiplexer circuit illustrated in Pixley's FIG. 9] ***for multiplexing a plurality of data words*** [*e.g.*, the m-bit data words input into each of MUXs 112, 114, and 116] ***including a first data word*** [*e.g.*, data input 124 to tri-state MUX 116] ***and a second data word*** [*e.g.*, the m-bit data word output by tri-state MUX 114 and input into tri-state MUX 112], as recited in [1.pre].

[1.a] a plurality of data connections;

64. In my opinion, Pixley-Watanabe renders obvious Element [1.a].

65. The term "data connection" is used broadly in the '448 patent to refer to any connection that carries data. For example, the '448 patent labels the lines within the multiplexor as "connections" (e.g., connections 490-493, *see* '494 patent, 7:15-20); and there can be no disputes that these connections carry data. Moreover, as explained in the '448 patent, data connections comprise "e.g., data ports, etc." – implying a broad definition that encompasses connections that carry data. *Id.*, 1:24-27.

66. As to Pixley, with respect to FIG. 9, Pixely describes that "[d]ata inputs and control/select signals to the MUXes 112, 114, and 116 are provided via four D flip-flops 124, 126, 130, and 132 and via two input terminals/pins 128 and 129." EX1004, 7:11-14. Further, as illustrated in FIG. 9, the outputs of tri-state MUXs 114 and 116 are inputs to tri-state MUX 112, connecting tri-state MUX 112 to each of data inputs 124, 126, 130, and 132 via tri-state MUXs 114 and 116. *See* EX1004, FIG. 9.



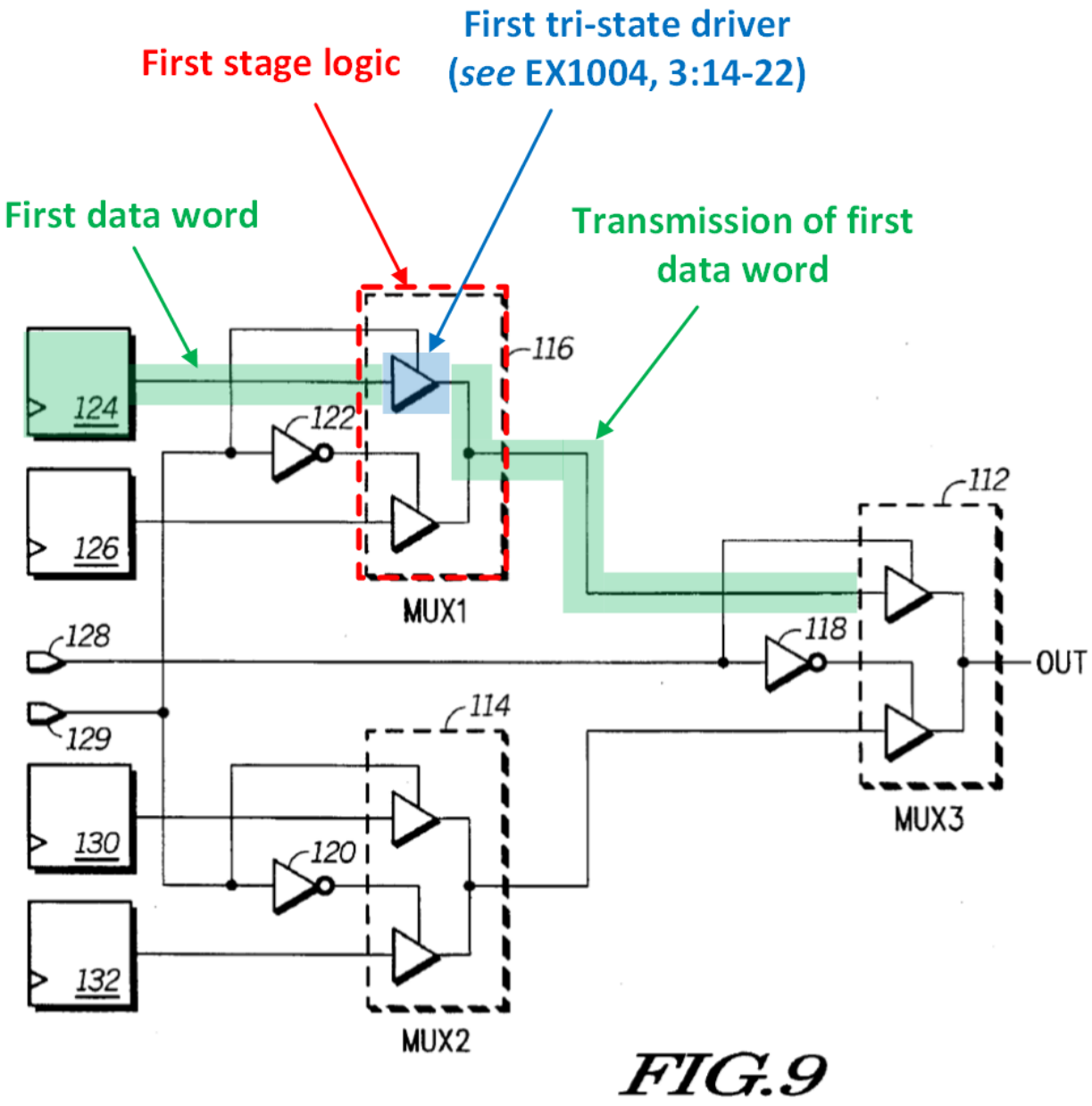
Annotated FIG. 9 of Pixley

67. Accordingly, a POSITA would have understood and found obvious that Pixley-Watanabe's nested multiplexers of FIG. 9 included ***a plurality of data connections*** [*e.g.*, the data inputs to each of tri-state MUXs 112, 114, and 116], as recited in [1.a].

[1.b] first stage logic having a first tristate driver, the first tristate driver configured to receive the first data word from one of the connections and to transmit the first data word received;

68. In my opinion, Pixley-Watanabe renders obvious Element [1.b].

69. "FIG. 9 illustrates a circuit having tri-state MUXes 112, 114, and 116." EX1004, 7:8-9. Each of these MUXes includes two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "Each driver [] is enabled only if the corresponding select line (sel-#) is enabled." EX1004, 3:18-19. "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.



Annotated FIG. 9 of Pixley

70. In the example of FIG. 9, when the select signal input on terminal 129 is enabled (*i.e.*, a logic high), the first/upper tri-state driver of tri-state MUX 116 (highlighted above in blue) is enabled, and the data word input to the first/upper tri-state driver is coupled to the output of tri-state MUX 116. *See* EX1004, FIG. 9,

3:14-22. Correspondingly, the lower tri-state driver of tri-state MUX 116 receives a complement of the select signal input on terminal 129 from inverter 122 and therefore is disabled. *See id.*

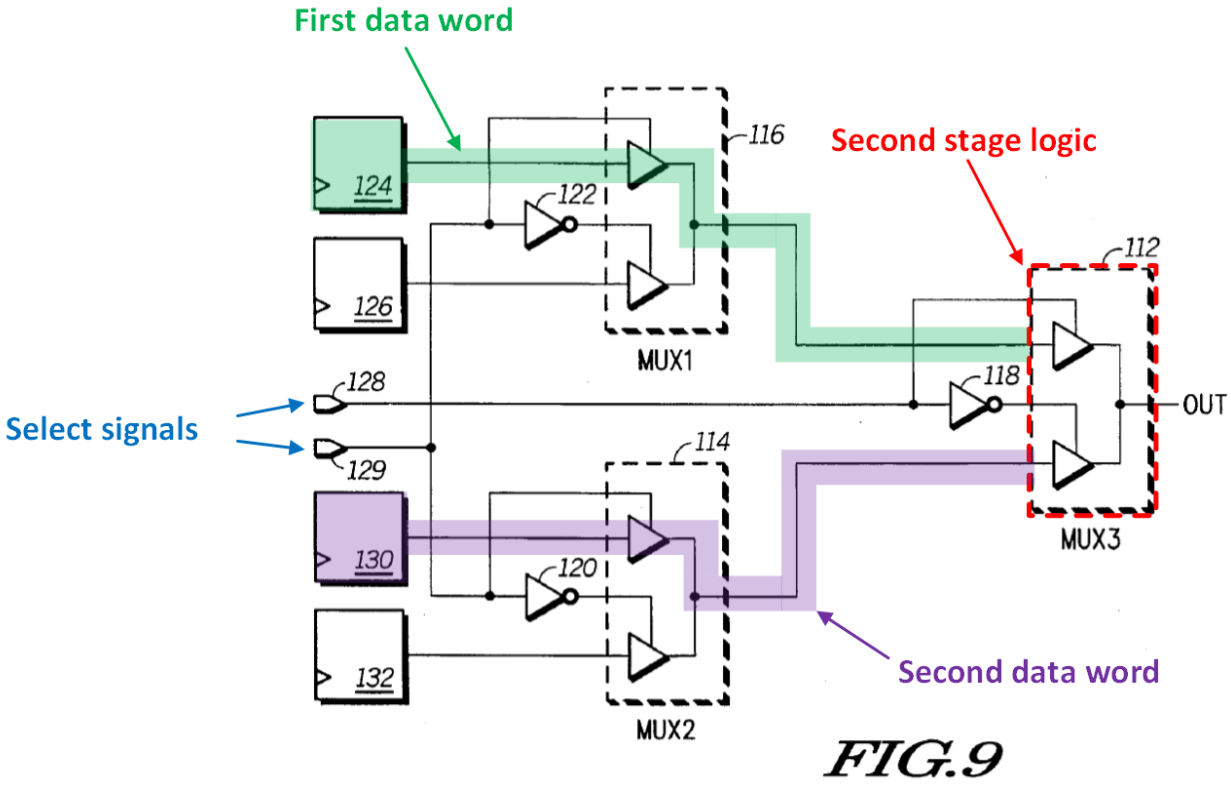
71. Accordingly, a POSITA would have understood and found obvious that Pixley-Watanabe's nested multiplexers of FIG. 9 included ***first stage logic*** [*e.g.*, tri-state MUX 116] ***having a first tristate driver*** [*e.g.*, the upper tri-state driver of tri-state MUX 116, highlighted in blue above]***, the first tristate driver configured to receive the first data word from one of the connections*** [*e.g.*, the m-bit data word received from the m D flip-flops 124] ***and to transmit the first data word received*** [*e.g.*, the upper tri-state driver transmits the m-bit data word received from the m D flip-flops 124 when the select signal input on terminal 129 is enabled], as recited in [1.b].

[1.c.i] second stage logic configured to receive the first data word from the first stage logic and to receive the second data word from another of the plurality of data connections, the second stage logic configured to select for transmission one of the first data word and the second data word based upon a set of select signals, the second stage logic configured to transmit the data word selected for transmission by the second stage logic

72. In my opinion, Pixley-Watanabe renders obvious Element [1.c.i].

73. As described with respect to Element [1.b], *supra,* each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "When the select line labeled sel-N (wherein N is an integer

between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.



Annotated FIG. 9 of Pixley

74. In the example of FIG. 9, when the select signal input on terminal 129 is enabled (*i.e.*, a logic high), the upper tri-state driver of tri-state MUX 114 is enabled, and the second data word (highlighted in purple above) input to the upper tri-state driver is coupled to the output of tri-state MUX 114 and correspondingly to one of the inputs of tri-state MUX 112. *See* EX1004, FIG. 9, 3:14-22. And, as described with respect to Element [1.b], *supra*, when the select signal input on terminal 129 is enabled (*i.e.*, a logic high), the first/upper tri-state driver of tri-state

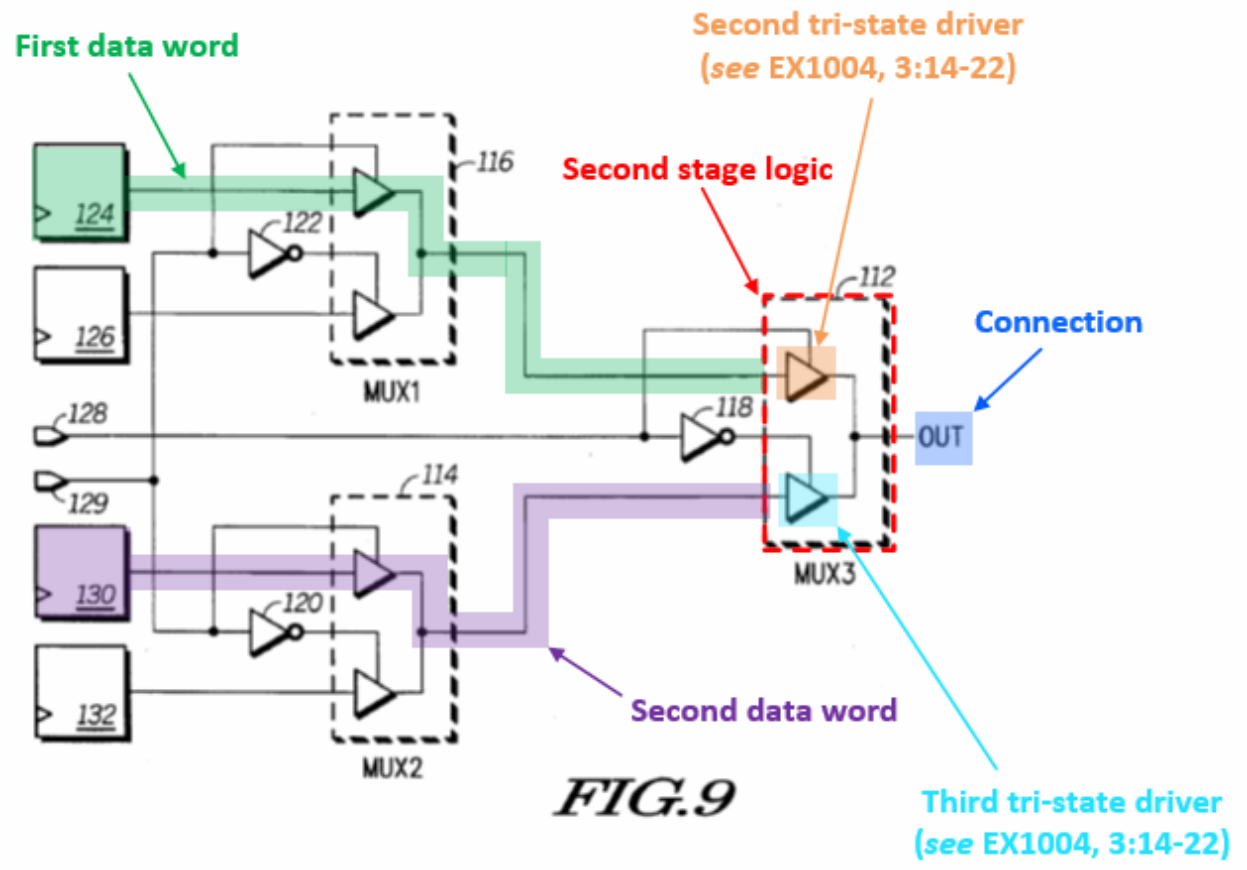
MUX 116 is enabled, and the first data word (highlighted in green above) input to the first/upper tri-state driver is coupled to the output of tri-state MUX 116 and correspondingly to the other input of tri-state MUX 112. *See id.*

75. Accordingly, a POSITA would have understood and found obvious that Pixley-Watanabe's nested multiplexers of FIG. 9 included ***second stage logic*** [*e.g.*, tri-state MUX 112] ***configured to receive the first data word from the first stage logic*** [*e.g.*, the tri-state MUX 112 receives the m-bit data word of the m D flip-flops 124 from the tri-state MUX 116 when select signal input on terminal 129 is enabled] ***and to receive the second data word from another of the plurality of data connections*** [*e.g.*, the tri-state MUX 112 receives the m-bit data word of the m D flip-flops 130 from the tri-state MUX 114 when select signal input on terminal 129 is enabled]***, the second stage logic configured to select for transmission one of the first data word and the second data word based upon a set of select signals*** [*e.g.*, the tri-state MUX 112 selects between the data words output by each of tri-state MUX 114 and tri-state MUX 116 based on two select signals: the signal input on terminal/pin 128 input to the upper tri-state driver and the signal output by inverter 118 and input into the lower tristate driver]***, the second stage logic configured to transmit the data word selected for transmission by the second stage logic*** [*e.g.*, "OUT" of tri-state MUX 112 transmits the selected data word], as recited in [1.c.i].

[1.c.ii] the second stage logic having a connection, a second tristate driver, and a third tristate driver, the second tristate driver having an output electrically coupled to an output of the third tristate driver, the second tristate driver configured to receive the first data word from the first tristate driver and to drive the connection of the second stage logic with the first data word if the first data word is selected for transmission by the second stage logic, the third tristate driver configured to receive the second data word and to drive the connection of the second stage logic with the second data word if the second data word is selected for transmission by the second stage logic.

76. In my opinion, Pixley-Watanabe renders obvious Element [1.c.ii].

77. As described with respect to Element [1.b], *supra*, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.



Annotated FIG. 9 of Pixley

78. In the example of FIG. 9, when the select signal input on terminal 128 is enabled (*i.e.*, a logic high), the second/upper tri-state driver of tri-state MUX 112 is enabled, and the first data word (highlighted in green above) input to the upper tri-state driver is coupled to the output OUT of tri-state MUX 112. *See* EX1004, FIG. 9, 3:14-22. Correspondingly, the third/lower tri-state driver of tri-state MUX 112 receives a complement of the select signal input on terminal 128 via inverter 118, such that when the select signal input on terminal 128 is enabled, the output of inverter 118 is disabled (*i.e.*, a logic low) and the third/lower tri-state driver of tri-

state MUX 112 is disabled. *See id.* On the other hand, when the select signal input on terminal 128 is disabled (*i.e.*, a logic low), the second/upper tri-state driver of tri-state MUX 112 is disabled. *See id.* And, correspondingly, the third/lower tri-state driver of tri-state MUX 112 receives a complement of the select signal input on terminal 128 via inverter 118, so when the select signal input on terminal 128 is disabled, the output of inverter 118 is enabled (*i.e.*, a logic high) and the third/lower tri-state driver of tri-state MUX 112 is enabled such that the second data word (highlighted in purple above) input to the lower tri-state driver is coupled to the output OUT of tri-state MUX 112. *See id.*

79. Accordingly, a POSITA would have understood and found obvious that Pixley-Watanabe's tri-state MUX 112 [the second stage logic] included ***a connection*** [*e.g.*, output OUT of tri-state MUX 112]***, a second tristate driver*** [*e.g.*, the upper tri-state driver of tri-state MUX 112, highlighted in orange above]***, and a third tristate driver*** [*e.g.*, the lower tri-state driver of tri-state MUX 112, highlighted in light blue above]***, the second tristate driver having an output electrically coupled to an output of the third tristate driver*** [*e.g.*, the output of the upper and lower tri-state drivers of tri-state MUX 112 are coupled together at OUT]***, the second tristate driver configured to receive the first data word from the first tri-state driver*** [*e.g.*, the upper tri-state driver of tri-state MUX 112 receives the first

data word from the upper tri-state driver of tri-state MUX 116 when the select signal input on terminal/pin 129 is enabled] ***and to drive the connection of the second stage logic with the first data word if the first data word is selected for transmission by the second stage logic*** [*e.g.*, the data word received from the tri-state MUX 116 to the upper tri-state driver of tri-state MUX 112 is coupled to OUT for transmission when the select signal input on terminal/pin 128 is enabled]***, the third tristate driver configured to receive the second data word and to drive the connection of the second stage logic with the second data word if the second data word is selected for transmission by the second stage logic*** [*e.g.*, the data word received from the tri-state MUX 114 to the lower tri-state driver of tri-state MUX 112 is coupled to OUT for transmission when the select signal output from inverter 118 is enabled, which is when the select signal input on terminal/pin 128 is disabled], as recited in [1.c.ii].

2. Claim 2

[2] The circuit of claim 1, wherein the second stage logic is further configured to transmit, to a third stage logic, the data word selected for transmission by the second stage logic.

80. In my opinion, Pixley-Watanabe renders obvious claim 2.

81. As described with respect to Element [1.b], *supra*, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-

14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.

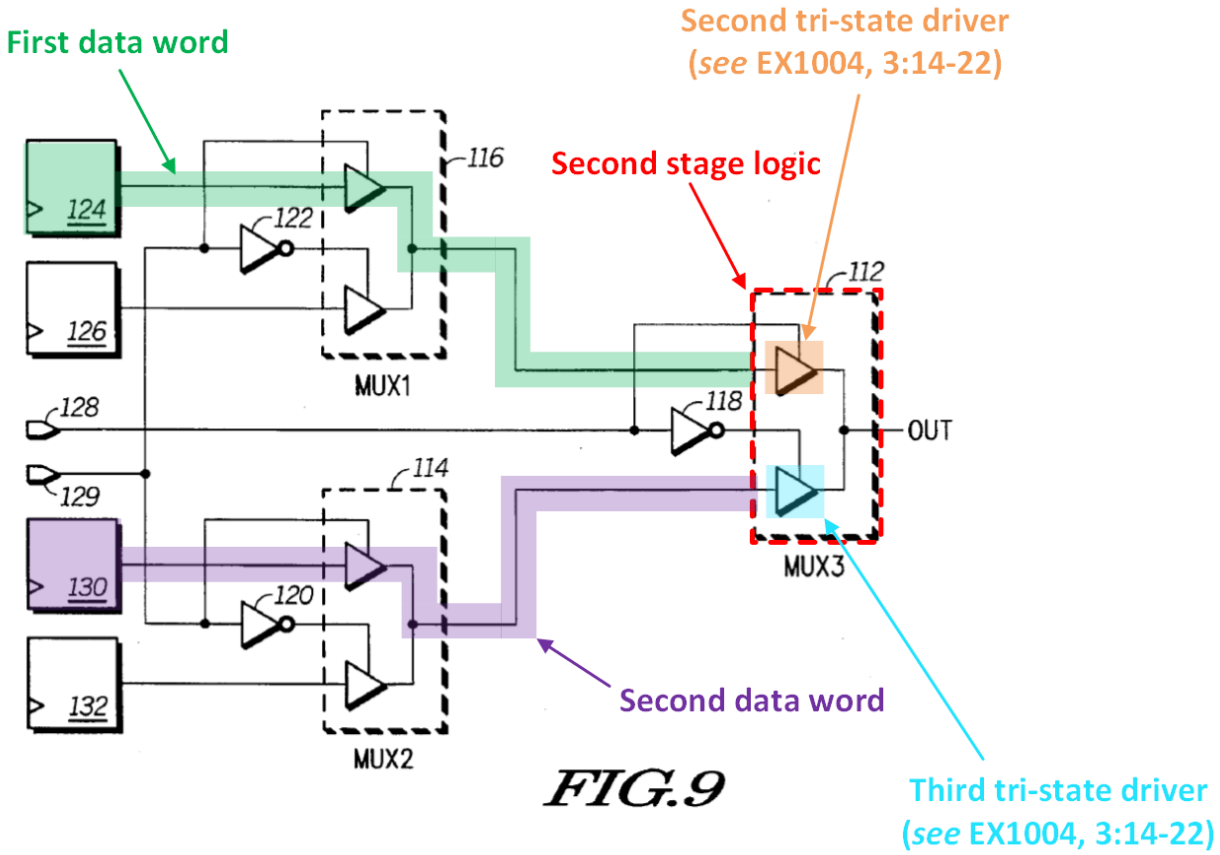
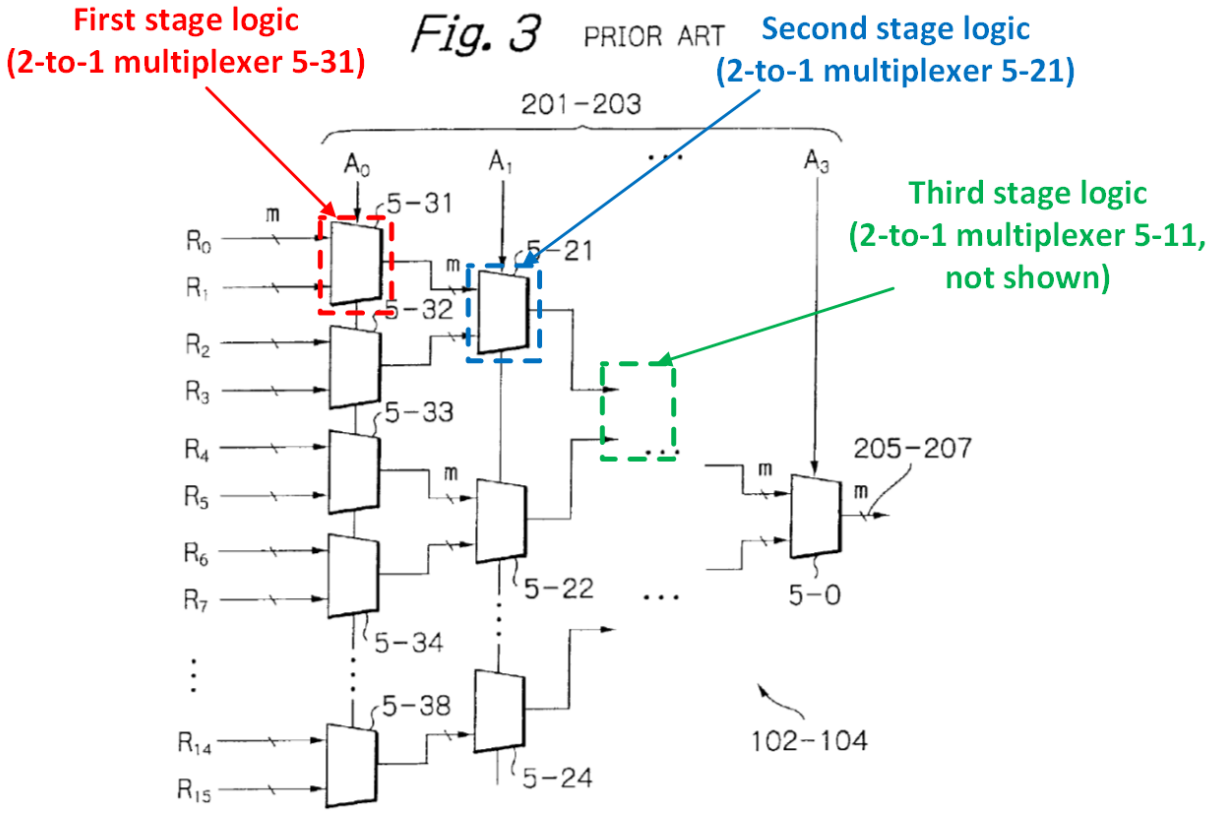


FIG.9

Annotated FIG. 9 of Pixley

82. Pixley does not limit to where the output of the tri-state MUX 112 is transmitted. *See generally* EX1004, 7:7-8:3. As described with respect to Element [1.pre], *supra*, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See, e.g.*, EX1005, FIG. 3,

[0035]-[0037]. In one example of nested multiplexers described by Watanabe, the output of a second stage MUX (*e.g.*, 2-to-1 multiplexer 5-21) is transmitted to the input of a third stage MUX (*e.g.*, 2-to-1 multiplexer 5-11). *See* EX1005, FIG. 3, [0036]-[0037]. This configuration was useful when the nested/cascading multiplexers were required to select between more than four data inputs. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037].



Annotated FIG. 3 of Watanabe

83. Accordingly, a POSITA would have understood and found obvious that, in Pixley-Watanabe's nested multiplexers of FIG. 9, ***the second stage logic*** [*e.g.*, multiplexer 112] ***is further configured to transmit, to a third stage logic***

[*e.g.*, a subsequent stage tri-state multiplexer in a larger nested multiplexer circuit with more than four data inputs]***, the data word selected for transmission by the second stage logic*** [*e.g.*, the output of tri-state multiplexer would have been connected to the input of the subsequent stage tri-state multiplexer], as recited in Claim [2].

3. Claim 3

[3] The circuit of claim 1, wherein the second stage logic is further configured to transmit, to a system bus, the data word selected for transmission by the second stage logic.

84. In my opinion, Pixley-Watanabe renders obvious Claim [3].

85. As described with respect to [1.c.ii], *supra*, the tri-state MUX 112 is configured to output the data word received from either tri-state MUX 114 or tri-state MUX 116, depending on the state of the select signals output by terminal/PIN 128 and inverter 118. *See* EX1004, FIG. 9, 7:7-14, 3:14-22.

86. Further, as described with respect to [1.pre], *supra*, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037]. One known use case of such a configuration was the output of an operand from a selected register in a register bank for transmission to and use in an arithmetic and logical unit (ALU). *See* EX1005, FIG. 4, [0043]. In this example use case, "[t]he operand out-

put . . . that has been readout is interconnected to be sent, for example, to the arithmetic and logical unit (ALU) via a system bus . . . ." EX1005, [0043]; *see also* EX1006, FIG. 9, [0021], [0028]-[0029], [0034] (illustrating and describing "4-to-1 Multiplexer (MUX) formed with cascaded buses" and tri-state drivers).

87. Accordingly, a POSITA would have at least found it obvious that, in Pixley-Watanabe's nested multiplexers of FIG. 9, ***the second stage logic*** [*e.g.*, tri-state MUX 112] ***is further configured to transmit, to a system bus*** [*e.g.*, a system bus connecting the output OUT of tri-state MUX 112 to an arithmetic and logical unit (ALU)]***, the data word selected for transmission by the second stage logic*** [*e.g.*, either the data word output by tri-state MUX 114 or the data word output by tri-state MUX 116, depending on the state of the select signals output by terminal/PIN 128 and inverter 118], as recited in Claim [3].

4. Claim 4

[4.pre] A multiplexor circuit, comprising:

88. Element [4.pre] is substantially the same as at least a portion of Element [1.pre]. In my opinion, Pixley-Watanabe renders obvious Element [4.pre] for at least the same reasons described above with respect to Element [1.pre]. *See* Element [1.pre], *supra*.

[4.a] a plurality of data connections;

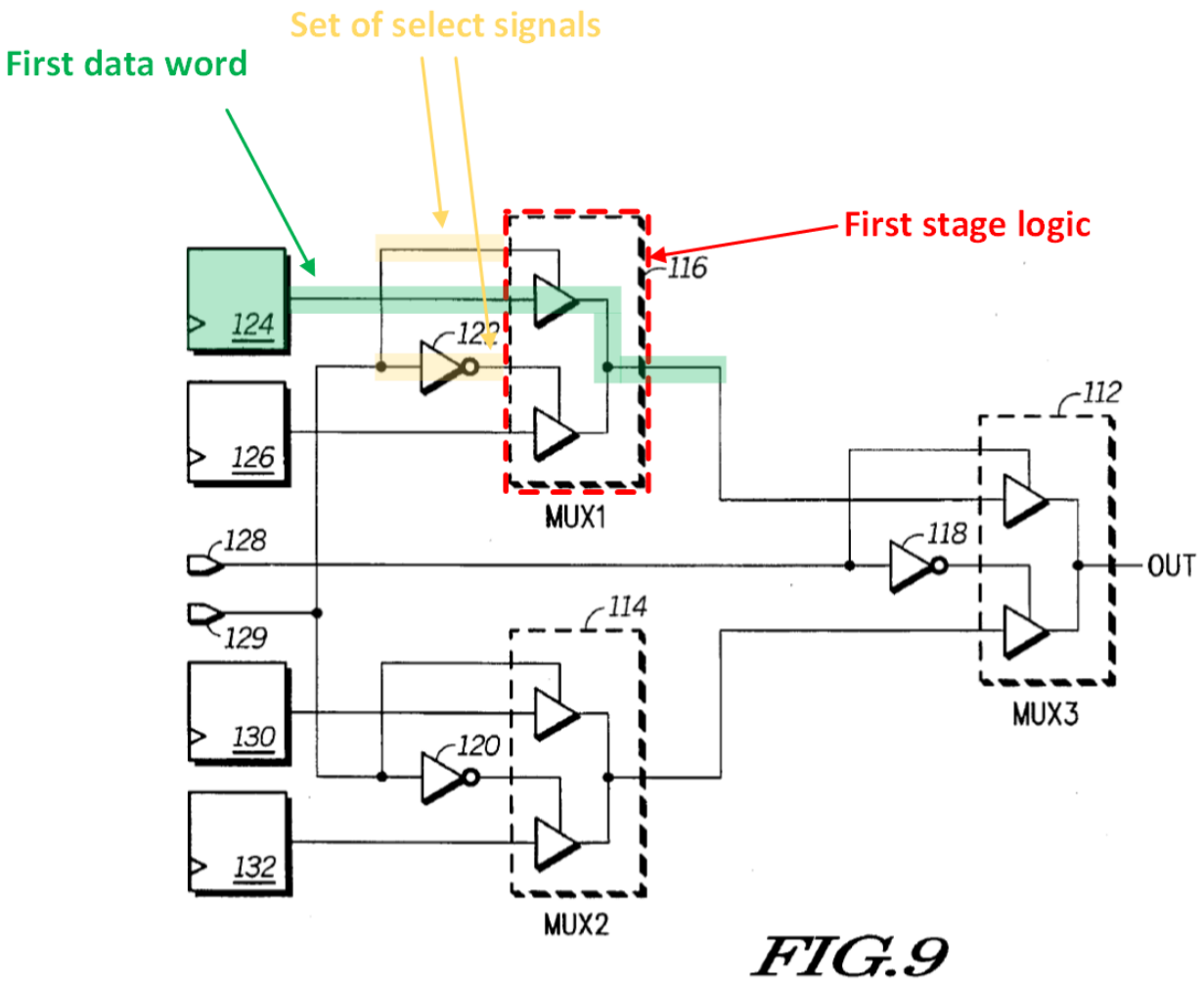
89. Element [4.a] is substantially the same as at least a portion of Element [1.a]. In my opinion, Pixley-Watanabe renders obvious Element [4.a] for at least

the same reasons described above with respect to Element [1.a]. *See* Element [1.a], *supra*.

[4.b] first stage logic configured to receive a first data word from one of the connections and to transmit the first data word received based on a set of select signals; and

90. In my opinion, Pixley-Watanabe renders obvious Element [4.b].

91. "FIG. 9 illustrates a circuit having tri-state MUXes 112, 114, and 116." EX1004, 7:8-9. Each of these MUXes include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "Each driver 30-32 is enabled only if the corresponding select line (sel-#) is enabled." EX1004, 3:18-19. "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.



Annotated FIG. 9 of Pixley

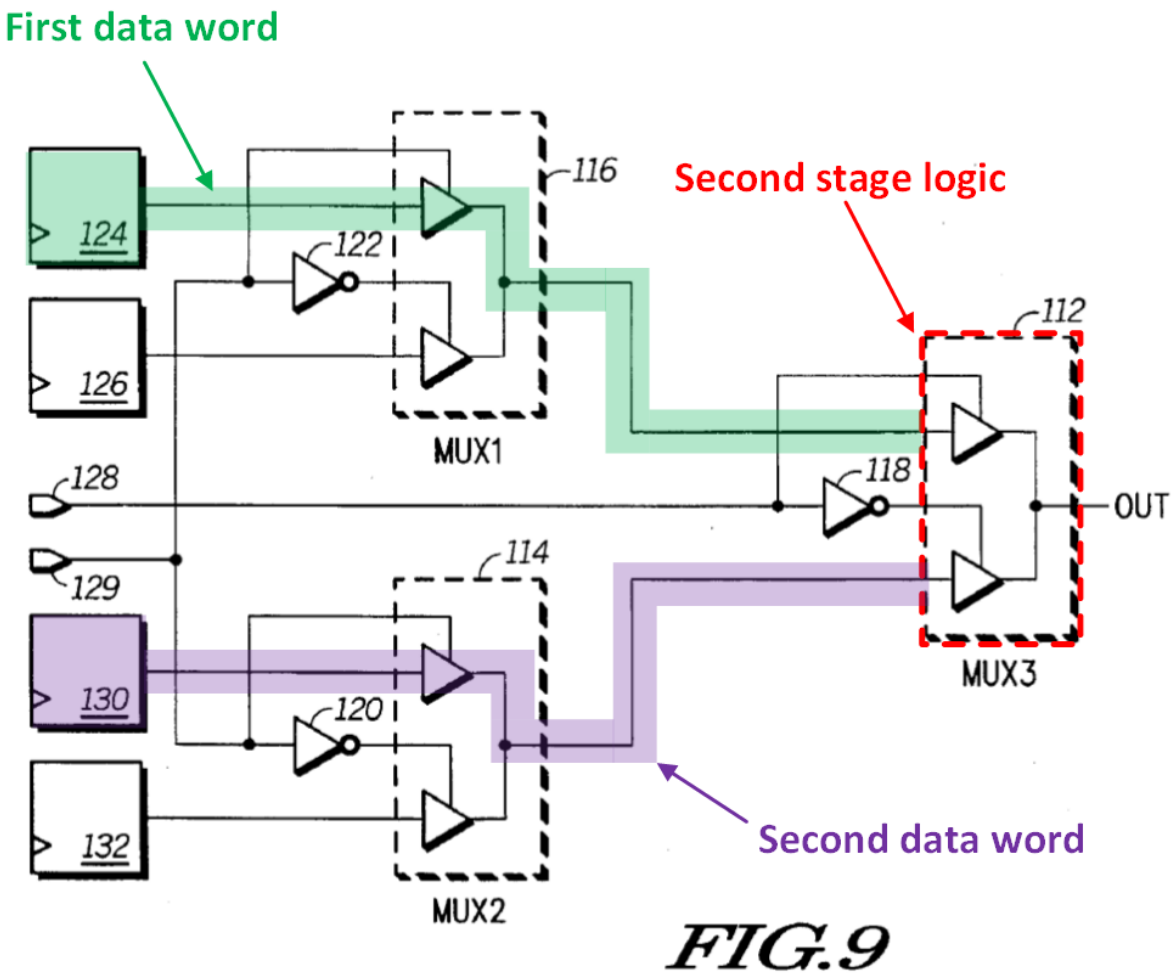
92. In the example of FIG. 9, when the select signal input on terminal 129 is enabled (*i.e.*, a logic high), the first/upper tri-state driver of tri-state MUX 116 is enabled, and the data word input to the first/upper tri-state driver from D flip-flops 124 is coupled to the output of tri-state MUX 116. *See* EX1004, FIG. 9, 3:14-22. Correspondingly, the lower tri-state driver of tri-state MUX 116 receives a complement of the select signal input on terminal 129 from inverter 122, and therefore is disabled. *See id.*

93. Accordingly, a POSITA would have understood and found obvious that Pixley-Watanabe's nested multiplexers of FIG. 9 included ***first stage logic*** [*e.g.*, tri-state MUX 116] ***configured to receive the first data word from one of the connections*** [*e.g.*, the m-bit data word received from the m D flip-flops 124] ***and to transmit the first data word received*** [*e.g.*, the upper tri-state driver transmits the m-bit data word received from the m D flip-flops 124 when the select signal input on terminal 129 is enabled] ***based on a set of select signals*** [*e.g.*, the select signal input on terminal 129 and the complement output by inverter 122], as recited in [1.b].

[4.c.i] second stage logic configured to receive the first data word from the first stage logic and to select a selected data word between the first data word and a second data word received from another of the plurality of data connections based upon a set of select signals,

94. In my opinion, Pixley-Watanabe renders obvious Element [1.c.i].

95. As described with respect to Element [1.b], *supra*, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.



Annotated FIG. 9 of Pixley

96. In the example of FIG. 9, when the select signal input on terminal 129 is enabled (*i.e.*, a logic high), the upper tri-state driver of tri-state MUX 114 is enabled, and the second data word (highlighted in purple above) input to the upper tri-state driver is coupled to the output of tri-state MUX 114 and correspondingly to one of the inputs of tri-state MUX 112. *See* EX1004, FIG. 9, 3:14-22. And, as described with respect to Element [1.b], *supra*, when the select signal input on terminal 129 is enabled (*i.e.*, a logic high), the first/upper tri-state driver of tri-state

MUX 116 is enabled, and the first data word (highlighted in green above) input to the first/upper tri-state driver is coupled to the output of tri-state MUX 116 and correspondingly to the other input of tri-state MUX 112. *See id.*

97. Accordingly, a POSITA would have understood and found obvious that Pixley-Watanabe's nested multiplexers of FIG. 9 included ***second stage logic*** [*e.g.*, tri-state MUX 112] ***configured to receive the first data word from the first stage logic*** [*e.g.*, the tri-state MUX 112 receives the m-bit data word of the m D flip-flops 124 from the tri-state MUX 116 when select signal input on terminal 129 is enabled] ***and to select a selected data word between the first data word*** [*e.g.*, the m-bit data word received from the m D flip-flops 124 and output by tri-state MUX 116 when select signal s129 is enabled] ***and a second data word received from another of the plurality of data connections*** [*e.g.*, the tri-state MUX 112 receives the m-bit data word of the m D flip-flops 130 from the tri-state MUX 114 when select signal input on terminal 129 is enabled] ***based upon a set of select signals*** [*e.g.*, the tri-state MUX 112 selects between the data words output by each of tri-state MUX 114 and tri-state MUX 116 based on two select signals: the signal input on terminal/pin 128 input to the upper tri-state driver and the signal output by inverter 118 and input into the lower tristate driver], as recited in [1.c.i].

[4.c.ii] the second stage logic configured to transmit the selected data word to a third stage logic and to increase a signal strength of the first data word when the first data word is transmitted, if the first data word is selected.

98. In my opinion, Pixley-Watanabe renders obvious Element [1.c.ii].

99. Pixley describes that "[a]n output of MUX 112 is labeled as OUT." EX1004, 7:10-11. As described with respect to Element [1.b], *supra*, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). When the select line is enabled, then the input value of the tri-state driver "is coupled to the output node OUT." EX1004, 3:19-22.

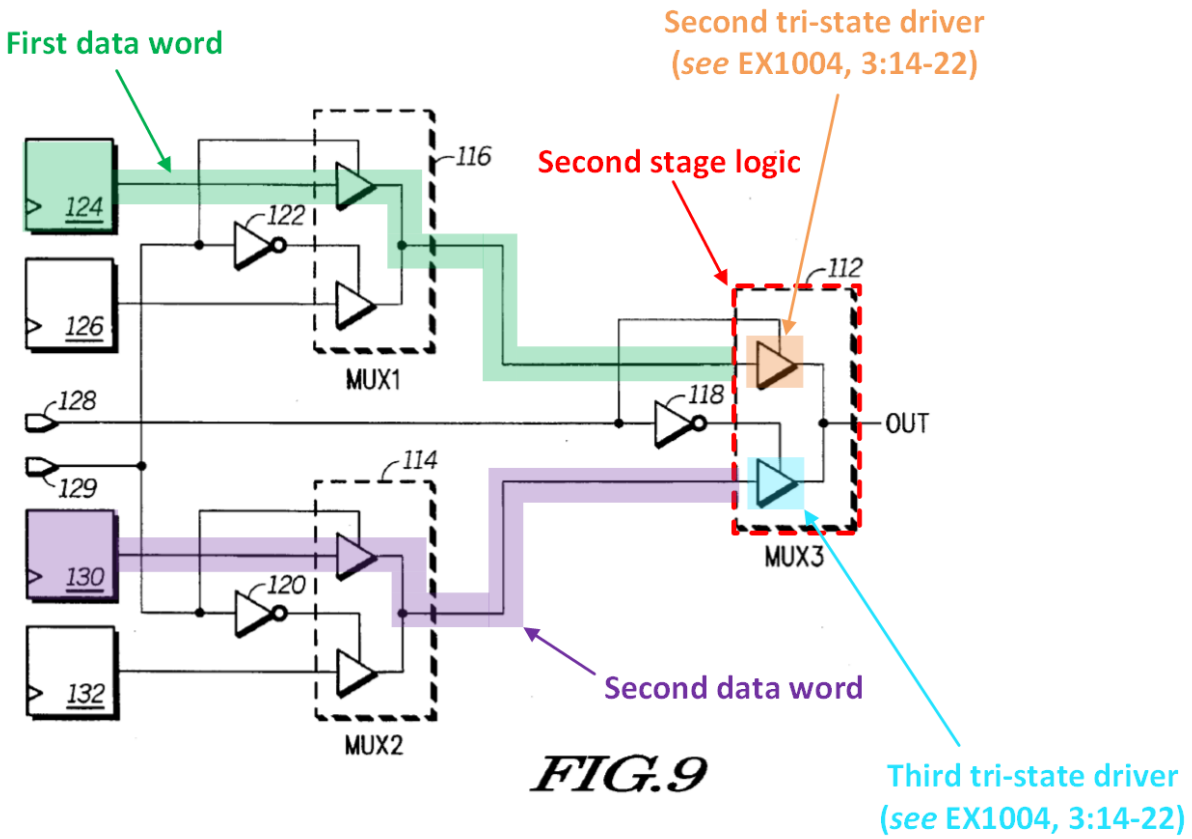


FIG.9

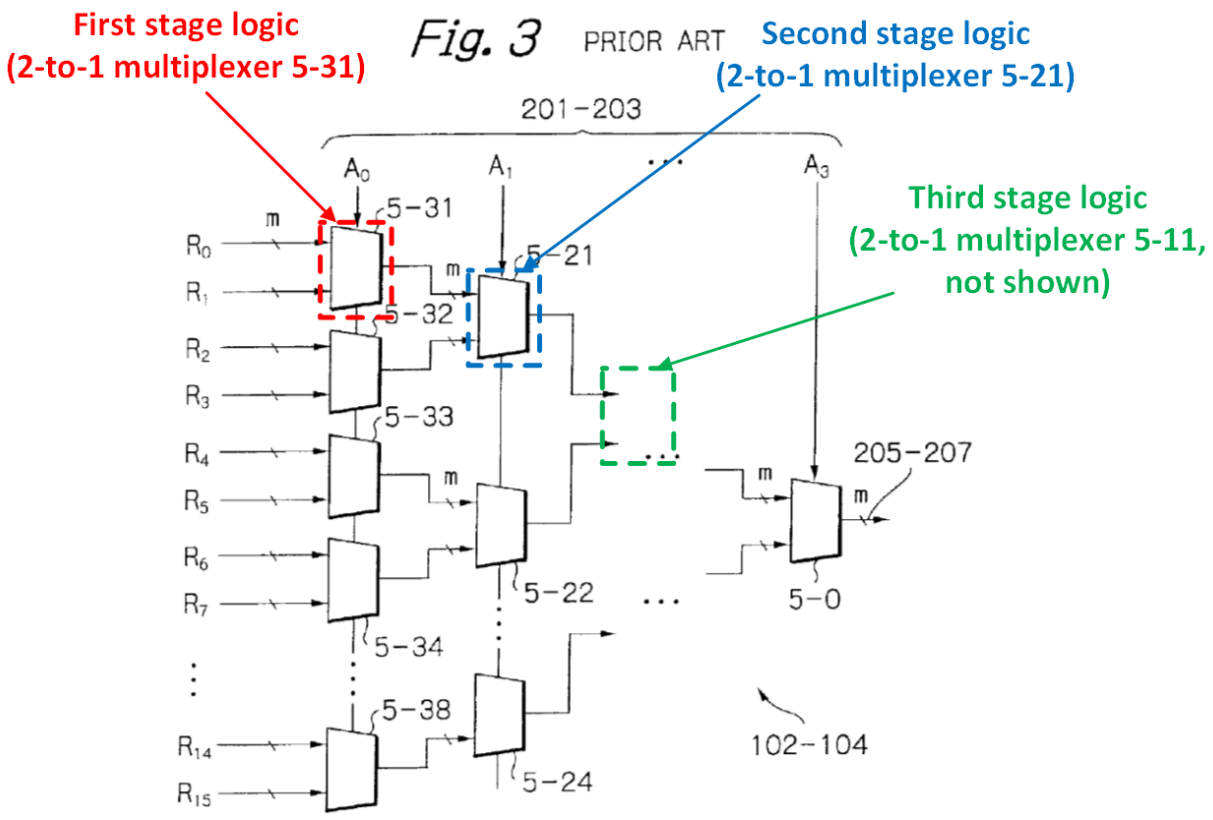
Annotated FIG. 9 of Pixley

100. As described with respect to Element [1.c.i], *supra*, and shown in FIG. 9, the select signals for the tri-state MUX 112 are the input terminal/pin 128 and its complement as output by inverter 118. *See* EX1004, FIG. 9, 7:7-14. When the input terminal/pin 128 is enabled (*i.e.*, logic level high), the upper tri-state driver of MUX 112 is enabled, the lower tri-state driver of MUX 112 is disabled, and the MUX 112 passes the first data word from MUX 116 to output OUT via the upper tri-state driver. *See* EX1004, FIG. 9, 3:14-22, 7:7-14. Conversely, when input terminal/pin 128 is disabled (*i.e.*, logic level low), the upper tri-state driver of MUX 112 is disabled, the lower tri-state driver of MUX 112 is enabled, and the MUX 112 passes the second data word from MUX 114 to output OUT via the lower tri-state driver. *See id.*

101. A POSITA would have understood or found obvious that the tri-state drivers described by Pixley would have increased a signal strength of a data word when they were enabled and coupled the data word to the output of the corresponding tri-state multiplexer. It was known that tri-state drivers were programmable "to be either off (high output impedence [sic]) or on (able to pass and amplify the applied data input signal)." EX1007, [0074]. Indeed, one of the primary purposes of a "driver" was generally known to be amplification for the purpose of signal transmission. *See, e.g.*, EX1008, 1:14-23 ("One of the circuit designs used in MOS LSI

devices is a tristate driver circuit, i.e., a circuit which . . . interfaces with and functions to drive output loads, normally external to the MOS LSI, in response to low power signal sources within the MOS LSI"); EX1009, 1:64-2:5 (describing "an improved CMOS off-chip driver circuit which interfaces between a first circuit with a given supply voltage and a second circuit with a supply voltage having a magnitude greater than that of the given supply voltage").

102. Pixley does not limit to where the output of the tri-state MUX 112 is transmitted. *See generally* EX1004, 7:7-8:3. As described with respect to Element [1.pre] and Claim [3], *supra*, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See*, *e.g.*, EX1005, FIG. 3, [0035]-[0037]. In one example of nested multiplexers described by Watanabe, the output of a second stage MUX (*e.g.*, 2-to-1 multiplexer 5-21) is transmitted to the input of a third stage MUX (*e.g.*, 2-to-1 multiplexer 5-11). *See* EX1005, FIG. 3, [0036]-[0037]. This configuration was useful when the nested/cascading multiplexers were required to select between more than four data inputs. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037].



Annotated FIG. 3 of Watanabe

103. Accordingly, a POSITA would have understood and found obvious that, in Pixley-Watanabe's nested multiplexers of FIG. 9, ***the second stage logic*** [*e.g.*, the tri-state MUX 112] ***[was] configured to transmit the selected data word to a third stage logic*** [*e.g.*, in a larger nested multiplexer, the output of tri-state MUX 112 would be connected to a subsequent stage multiplexer] ***and to increase a signal strength of the first data word when the first data word is transmitted, if the first data word is selected*** [*e.g.*, when the first/upper tri-state driver is enabled and couples its input with the first data word to OUT, the tristate driver amplifies the first data word], as recited in Element [4.c.ii].

5. Claim 5

[5.pre] A multiplexor circuit, comprising:

104. Element [5.pre] is substantially the same as at least a portion of Element [1.pre]. In my opinion, Pixley-Watanabe renders obvious Element [5.pre] for at least the same reasons described above with respect to Element [1.pre]. *See* Element [1.pre], *supra*.

[5.a] a plurality of data connections;

105. Element [5.a] is substantially the same as at least a portion of Element [1.a]. In my opinion, Pixley-Watanabe renders obvious Element [5.a] for at least the same reasons described above with respect to Element [1.a]. *See* Element [1.a], *supra*.

[5.b] first stage logic configured to receive a first data word from one of the connections and to transmit the first data word received; and

106. Element [5.b] is substantially the same as at least a portion of Element [4.b]. In my opinion, Pixley-Watanabe renders obvious Element [5.b] for at least the same reasons described above with respect to Element [4.b]. *See* Element [4.b], *supra*.

[5.c] second stage logic configured to receive the first data word from the first stage logic and to select a selected data word between the first data word and a second data word received from another of the plurality of data connections based upon a set of select signals, the second stage logic configured to transmit the selected data word to a third stage logic,

107. Element [5.c] is substantially the same as Element [4.c.i] and at least a portion of Element [4.c.ii]. In my opinion, Pixley-Watanabe renders obvious Element [5.c] for at least the same reasons described above with respect to Elements [4.c.i] and [4.c.ii]. *See* Elements [4.c.i], [4.c.ii], *supra*.

[5.d] wherein the second stage logic comprises: a first tristate driver configured to drive a connection with the first data word from the first stage logic, the first tristate driver enabled if a data connection exclusion signal is asserted and disabled if the data connection exclusion signal is deasserted; and wherein the second stage logic comprises: . . . a second tristate driver configured to drive the connection with the second data word, the second tristate driver enabled if a data connection select signal is asserted and disabled if the data connection select signal is deasserted.

108. In my opinion, Pixley-Watanabe renders obvious Element [5.d].

109. As described with respect to Element [1.b], *supra*, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.

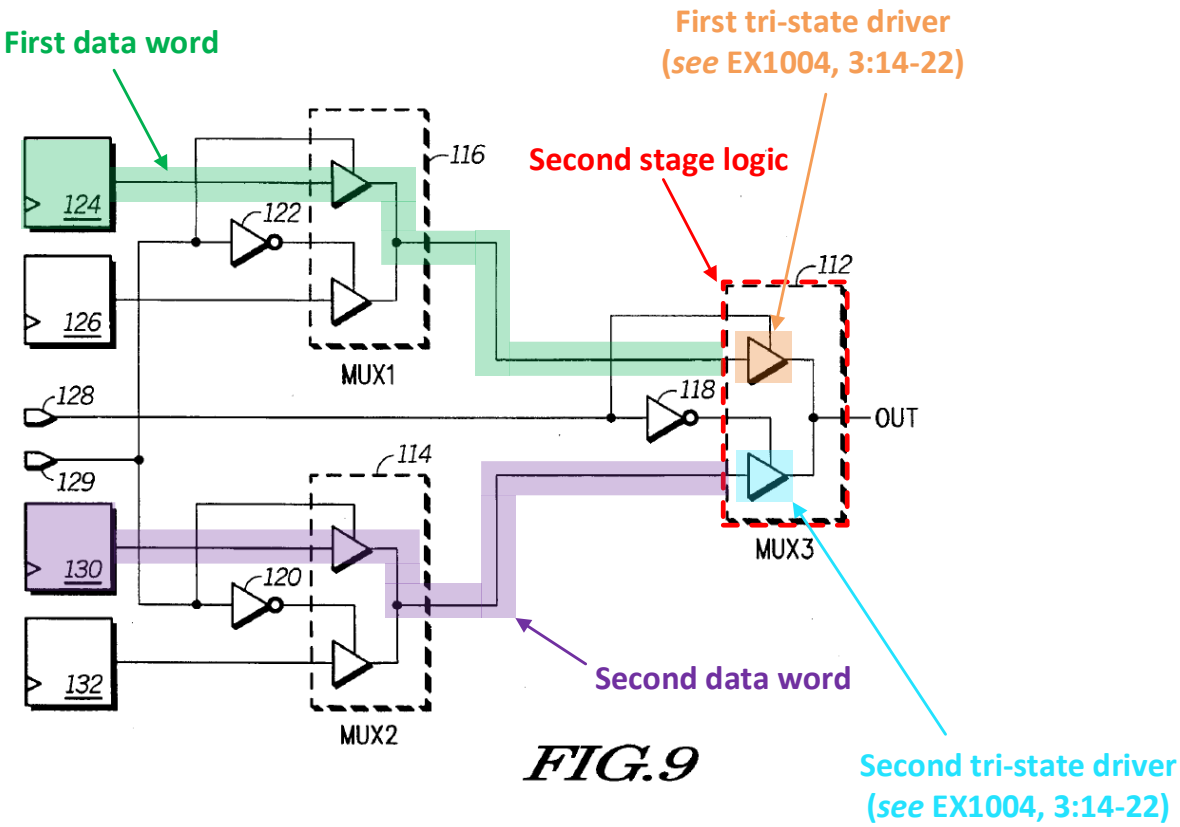


FIG.9

Annotated FIG. 9 of Pixley

110. In the example of FIG. 9, when the select signal input on terminal 128 is asserted (*i.e.*, a logic high), the first/upper tri-state driver of tri-state MUX 112 is enabled, and the first data word (highlighted in green above) input to the upper tri-state driver is coupled to the output OUT of tri-state MUX 112. *See* EX1004, FIG. 9, 3:14-22. Correspondingly, the second/lower tri-state driver of tri-state MUX 112 receives a complement of the select signal input on terminal 128 via inverter 118, such that when the select signal input on terminal 128 is asserted, the output of inverter 118 is deasserted (*i.e.*, a logic low) and the second/lower tri-state driver

of tri-state MUX 112 is disabled. *See id.* In this regard, the assertion of select signal input on terminal 128 is a data connection exclusion signal with respect to second data word, at least because when select signal input on terminal 128 is asserted, the second data word is excluded from being connected to the output OUT of the tri-state MUX 112 as a result of the second/lower tri-state driver being disabled. *See id.* On the other hand, when the select signal input on terminal 128 is deasserted (*i.e.*, a logic low), the first/upper tri-state driver of tri-state MUX 112 is disabled. *See id.*

111. Accordingly, a POSITA would have understood and found obvious that Pixley-Watanabe's tri-state MUX 112 [the second stage logic] included ***a first tristate driver*** [*e.g.*, the first/upper tri-state driver of tri-state MUX 112] ***configured to drive a connection with the first data word from the first stage logic*** [*e.g.*, the connection of the output of the tri-state MUX 116 to the input of the first/upper tri-state driver of tri-state MUX 112]***, the first tristate driver enabled if a data connection exclusion signal is asserted*** [*e.g.*, when the select signal input on terminal 128 is enabled/asserted, the first/upper tri-state driver couples its input with the first data word to the output OUT of the tri-state MUX 112] ***and disabled if the data connection exclusion signal is deasserted*** [e.g., when the select signal input on terminal 128 is disabled/deasserted, the first/upper tri-state driver is disabled], as recited in [5.d].

112. As described with respect to Element [1.b], *supra*, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.

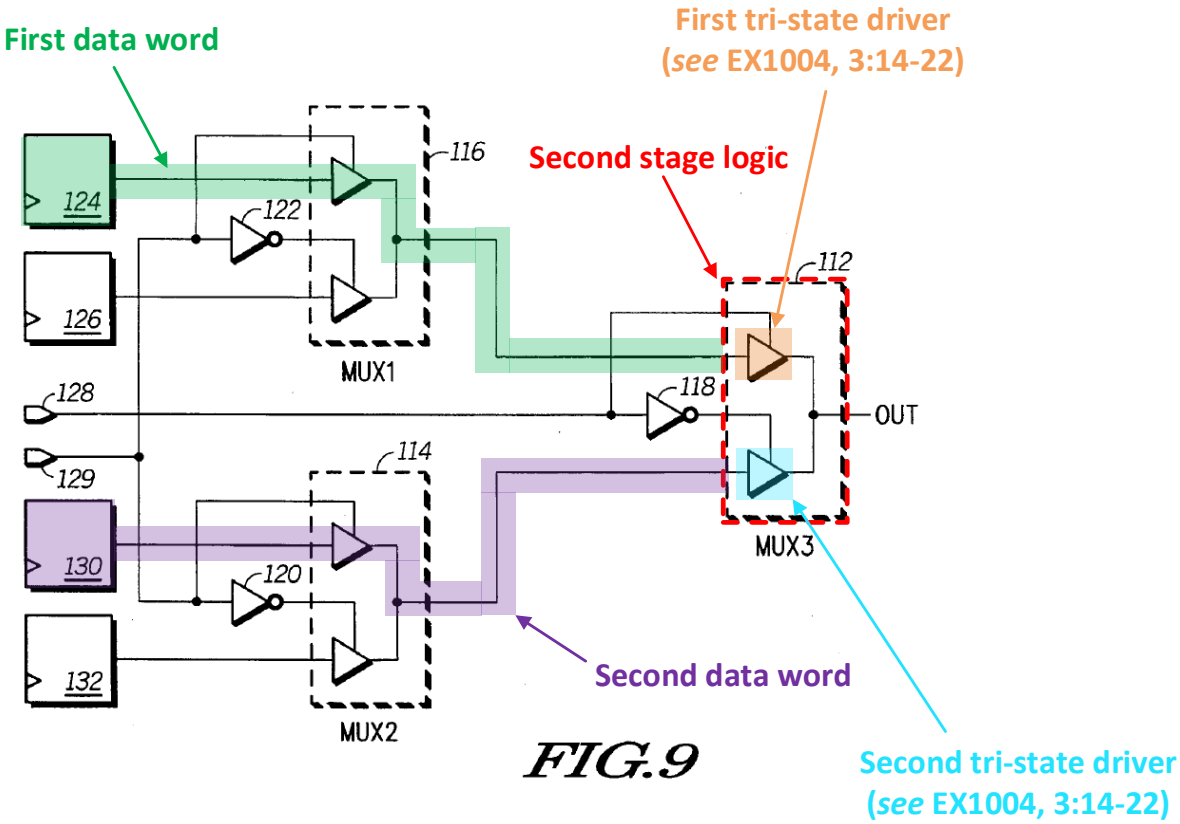


FIG.9

Annotated FIG. 9 of Pixley

113. In the example of FIG. 9, when the select signal output by the inverter 118 is asserted (*i.e.*, a logic high), the second/lower tri-state driver of tri-state MUX 112 is enabled, and the second data word (highlighted in purple above) input

to the lower tri-state driver is coupled to the output OUT of tri-state MUX 112. *See* EX1004, FIG. 9, 3:14-22. In this regard, the assertion of the select signal output by inverter 118 is a data connection select signal with respect to second data word, at least because when select signal output by inverter 118 is asserted, the second data word is selected to be connected to the output OUT of the tri-state MUX 112 as a result of the second/lower tri-state driver being enabled. *See id.* On the other hand, when the select signal output by inverter 118 is deasserted (*i.e.*, a logic low), the second/lower tri-state driver of tri-state MUX 112 is disabled. *See id.*

114. Accordingly, a POSITA would have understood and found obvious that Pixley-Watanabe's tri-state MUX 112 [the second stage logic] included ***a second tristate driver*** [*e.g.*, the second/lower tri-state driver of tri-state MUX 112] ***configured to drive a connection with the second data word*** [*e.g.*, the connection of the output of the tri-state MUX 114 to the input of the second/lower tri-state driver of tri-state MUX 112]***, the second tristate driver enabled if a data connection select signal is asserted*** [*e.g.*, when the select signal output by inverter 118 is enabled/asserted, the second/lower tri-state driver couples its input with the second data word to the output OUT of the tri-state MUX 112] ***and disabled if the data***

connection select signal is deasserted [*e.g.*, when the select signal output by inverter 118 is disabled/deasserted, the second/lower tri-state driver is disabled], as recited in [5.d].

6. Claim 6

[6] The circuit of claim 5 wherein the first tristate driver increases the signal strength of the first data word when the selected data word is the first data word.

115. In my opinion, Pixley-Watanabe renders obvious Claim [6].

116. As described with respect to Element [4.c.ii], *supra*, a POSITA would have understood or found obvious that the tri-state drivers described by Pixley would have increased a signal strength of a data word when they were enabled and coupled the data word to the output of the corresponding tri-state multiplexer. It was known that tri-state drivers were programmable "to be either off (high output impedence) or on (able to pass and amplify the applied data input signal)." EX1007, [0074]. Indeed, one of the primary purposes of a "driver" was generally known to be amplification for the purpose of signal transmission. *See, e.g.*, EX1008, 1:14-23 ("One of the circuit designs used in MOS LSI devices is a tristate driver circuit, i.e., a circuit which . . . interfaces with and functions to drive output loads, normally external to the MOS LSI, in response to low power signal sources within the MOS LSI"); EX1009, 1:64-2:5 (describing "an improved CMOS off-

chip driver circuit which interfaces between a first circuit with a given supply voltage and a second circuit with a supply voltage having a magnitude greater than that of the given supply voltage").

117. Accordingly, a POSITA would have understood and found obvious that, in Pixley-Watanabe's nested multiplexers of FIG. 9, ***the first tristate driver*** [*e.g.*, the first/upper tri-state driver of tri-state MUX 112] ***increases the signal strength of the first data word when the selected data word is the first data word*** [*e.g.*, when the first/upper tri-state driver is enabled and couples its input with the first data word to OUT, the tristate driver amplifies the first data word], as recited in Claim [6].

7. Claim 7

[7.a] The circuit of claim 5, further comprising: a first latch circuit configured to receive the data connection exclusion signal, the first latch circuit configured to latch the value indicative of the data connection exclusion signal at an output of the first latch circuit, the output of the first latch circuit configured to enable the first tristate driver if the data connection exclusion signal is asserted;

118. In my opinion, Pixley-Watanabe renders obvious element [7.a].

119. Pixley explicitly discloses multiplexing circuits that operate according to a system clock. *See* EX1004, Claim 7 ("The method of claim 1 wherein the circuit model uses a circuit clock signal wherein operations are performed in response to active edges of the circuit clock signal…"); *see also, e.g., id.* at 4:21-40 (dis-

cussing the system clock). Pixley's system therefore is both sequential (*i.e.*, it recites a machine that performs steps in a particular order) and synchronous (*i.e.*, a machine that performs those sequential steps according to a system clock). *See, e.g.*, EX1011, 420 (explaining that machines that use a "system clock" are called "synchronous sequential machines," which are machines where "input data are introduced into the machine and are processed sequentially according to some algorithm, and outputs are generated—all regulated by a system clock"). At minimum, a POSITA would have understood that Pixley's system could be operated according to a system clock as so implementing it allows it to receive, process, and output data in an orderly sequential fashion so as to work with other components.

120. A POSITA would have further understood that, in order to operate a machine in a synchronous, sequential fashion, memory elements must be added to a circuit. EX1011, 419. This is because devices that "lack true memory . . . lack the ability to perform sequential operations." EX1011, 419. Such a memory element "will remember a logic 1 or a logic 0 indefinitely or until it is directed to change to the other value." EX1011, ¶428. Thus, these memory elements allow for synchronous and sequential operation of a circuit because, rather than operating whenever an input is received, these memory elements hold their value, allowing operations to be performed at a time specified by the system clock.

121. A POSITA would have understood "[a] digital latch is the simplest digital storage element." EX1010, 285. As such, absent other considerations, a POSITA would have been strongly motivated to use a latch to avoid adding unnecessary complexity to a circuit. Alternatively, depending on design considerations, a POSITA may have also considered the use of a flip-flop—i.e. "the traditional storage element used to realize synchronous logic circuit." EX1010, 294. Flip-flops "save the state of the machine from one clock cycle to the next," and thus a POSITA would have also been strongly motivated to use a flip-flop in order to implement a synchronous sequential system. And, among the different type of flip-flops, "D flip-flops are the preferred type for integrated circuit applications"—as a POSITA would have also understood. EX1010, 295, A D flip-flop, though, is commonly comprised of two latches connected in a master-slave configuration. EX1010, 295-296.

122. In order to implement its disclosed multiplexors according to a system clock, Pixley discloses that "[d]ata inputs and control/select signals to the MUXes 112, 114, and 116 are provided via four D flip-flops 124, 126, 130, and 132..." EX1004, 7:10-13; *see also id.* Claim 10. As such, a POSITA would understand it obvious to use D flip-flops, each comprising two latches, for both the data inputs and the control/select signals. Even in the absence of this teaching, a POSITA

would have understood that, in order to implement Pixley's multiplexor with a system clock, memory elements would have been required and would have been motivated to use the "preferred type" of memory element—a D flip-flop comprising two latches—in order to accomplish that goal. EX1010, 295. Indeed, such implementation details well-known and easy to implement by a skilled artisan.

123. Accordingly, in my opinion, Pixley-Watanabe renders obvious a using a ***first latch circuit*** [*e.g.,* one of the latches in a D-flip flop] to ***receive*** and ***latch*** the value of the data connection exclusion signal, and ***output*** that value to the first tri-state driver according to a system clock.

[7.b] a second latch circuit configured to receive the data connection selection signal, the second latch circuit configured to latch the value indicative of the data connection selection signal at an output of the second latch circuit, the output of the second latch circuit configured to enable the second tristate driver if the data connection selection signal is asserted;

124. In my opinion, Pixley-Watanabe renders obvious element [7.b]

125. As discussed above, Pixley teaches a synchronous and sequential system that operates according to a system clock. EX1004, Claim 7 ("The method of claim 1 wherein the circuit model uses a circuit clock signal wherein operations are performed in response to active edges of the circuit clock signal…"); *see also, e.g., id.* at 4:21-40 (discussing the system clock); *see also* Element [7.a], *supra*. As also discussed above, Pixley teaches the use of D flip-flops as memory elements to en-

sure synchronous operation according to that clock. EX1004, 7:10-13 ("[d]ata inputs and control/select signals to the MUXes 112, 114, and 116 are provided via four D flip-flops 124, 126, 130, and 132..."); *see also id.* Claim 10. A POSITA would have been strongly motivated to use memory elements, and preferably D flip-flops, in order to ensure synchronous operation of the Pixley's multiplexors, including for both the control and data inputs. EX1010, 295; EX1011, 419-420. Accordingly, In my opinion, Pixley-Watanabe renders obvious using a ***second latch circuit*** [*e.g.*, one of the latches of a D-flip flop] to ***receive*** and ***latch*** the value of the data connection selection signal, and ***output*** that value to the second tristate driver according to a system clock.

[7.c] a third latch circuit configured to receive the second data word, the third latch circuit configured to latch the second data word at an output of the third latch circuit, the output of the third latch circuit configured for receipt by the second tristate driver;

126. In my opinion, Pixley-Watanabe renders obvious element [7.c].

127. As discussed above, Pixley teaches a synchronous and sequential system that operates according to a system clock. EX1004, Claim 7 ("The method of claim 1 wherein the circuit model uses a circuit clock signal wherein operations are performed in response to active edges of the circuit clock signal…"); *see also, e.g., id.* at 4:21-40 (discussing the system clock); *see also* Element [7.a], *supra*. As also discussed above, Pixley teaches the use of D flip-flops as memory elements to en-

sure synchronous operation according to that clock. EX1004, 7:10-13 ("[d]ata inputs and control/select signals to the MUXes 112, 114, and 116 are provided via four D flip-flops 124, 126, 130, and 132..."); *see also id.* Claim 10. A POSITA would have been strongly motivated to use memory elements, and preferably D flip-flops, in order to ensure synchronous operation of the Pixley's multiplexors, including for both the control and data inputs. EX1010, 295; EX1011, 419-420. Accordingly, In my opinion, Pixley-Watanabe renders obvious using a ***third latch circuit*** [*e.g.*, one of the latches of a D-flip flop] to ***receive*** and ***latch*** the second data word, and ***output*** that data word to the second tristate driver according to a system clock.

[7.d] a first clock configured to transmit a clock signal to the first, second, and third latch circuits, the clock signal activating latching of the second data word, the data connection selection signal and the data connection exclusion signal at the outputs of the third, second and first latch circuits, respectively.

128. In my opinion, Pixley-Watanabe renders obvious element [7.d.]

129. As discussed above, Pixley teaches a synchronous and sequential system that operates according to a system clock. EX1004, Claim 7 ("The method of claim 1 wherein the circuit model uses a circuit clock signal wherein operations are performed in response to active edges of the circuit clock signal…"); *see also, e.g., id.* at 4:21-40 (discussing the system clock); *see also* Element [7.a], <u>*supra*</u>. Accordingly, In my opinion, Pixley-Watanabe renders obvious using a ***first clock***

configured to transmit a clock signal to each of the components of the system, including the three latches discussed above.

8. Claim 8

[8.pre] A circuit comprising

130. Element [8.pre] is substantially the same as at least a portion of Element [1.pre]. In my opinion, Pixley-Watanabe renders obvious Element [8.pre] for at least the same reasons described above with respect to Element [1.pre]. *See* Element [1.pre], *supra*.

[8.a] a plurality of data connections;

131. Element [8.a] is substantially the same as at least a portion of Element [1.a]. In my opinion, Pixley-Watanabe renders obvious Element [8.a] for at least the same reasons described above with respect to Element [1.a]. *See* Element [1.a], *supra*.

[8.b] logic associated with a first data connection of the plurality of data connections, the logic associated with the first data connection configured to transmit a first data word corresponding to the first data connection;

132. In my opinion, Pixley-Watanabe renders obvious Element [8.b].

133. With respect to FIG. 9, Pixley describes that "[d]ata inputs and control/select signals to the MUXes 112, 114, and 116 are provided via four D flip-flops 124, 126, 130, and 132 . . . ." EX1004, 7:11-14. A POSITA would have understood that flip-flops are a form of "cyclic logic circuits," and are therefore a form of logic. EX1010, 5.

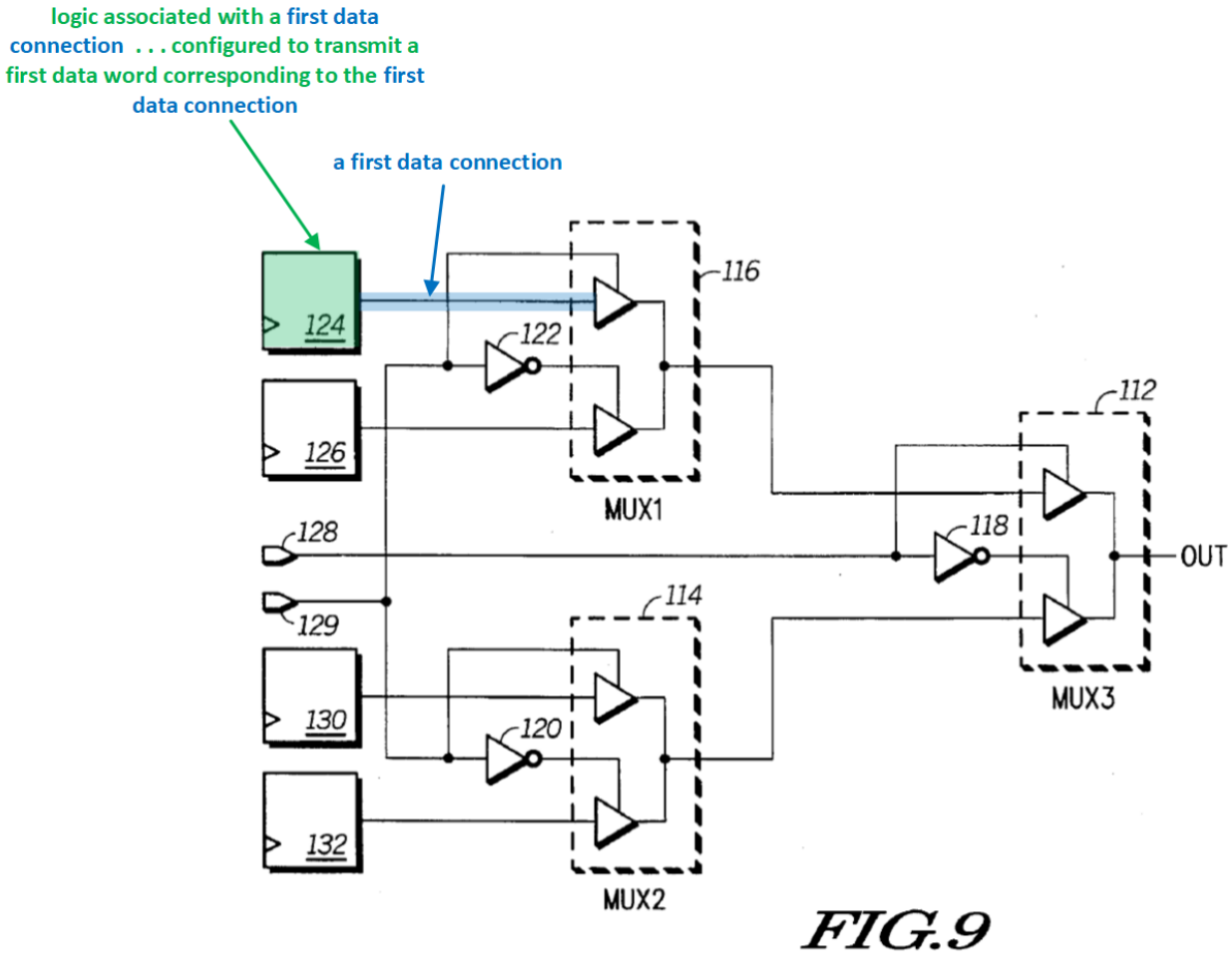


FIG.9

Annotated FIG. 9 of Pixley

134. As described with respect to Element [1.pre], *supra*, a POSITA would have understood or at least found obvious that, for certain known use cases, the 2-to-1 multiplexers described by Pixler would have been configured to multiplex m-bit data words. Specifically, it was known that "[a] 2-to-1 multiplexer of m-bit length is composed of m 2-to-1 multiplexers of one-bit length." *See, e.g.*, EX1005, [0018]. Further, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See, e.g.*, EX1005,

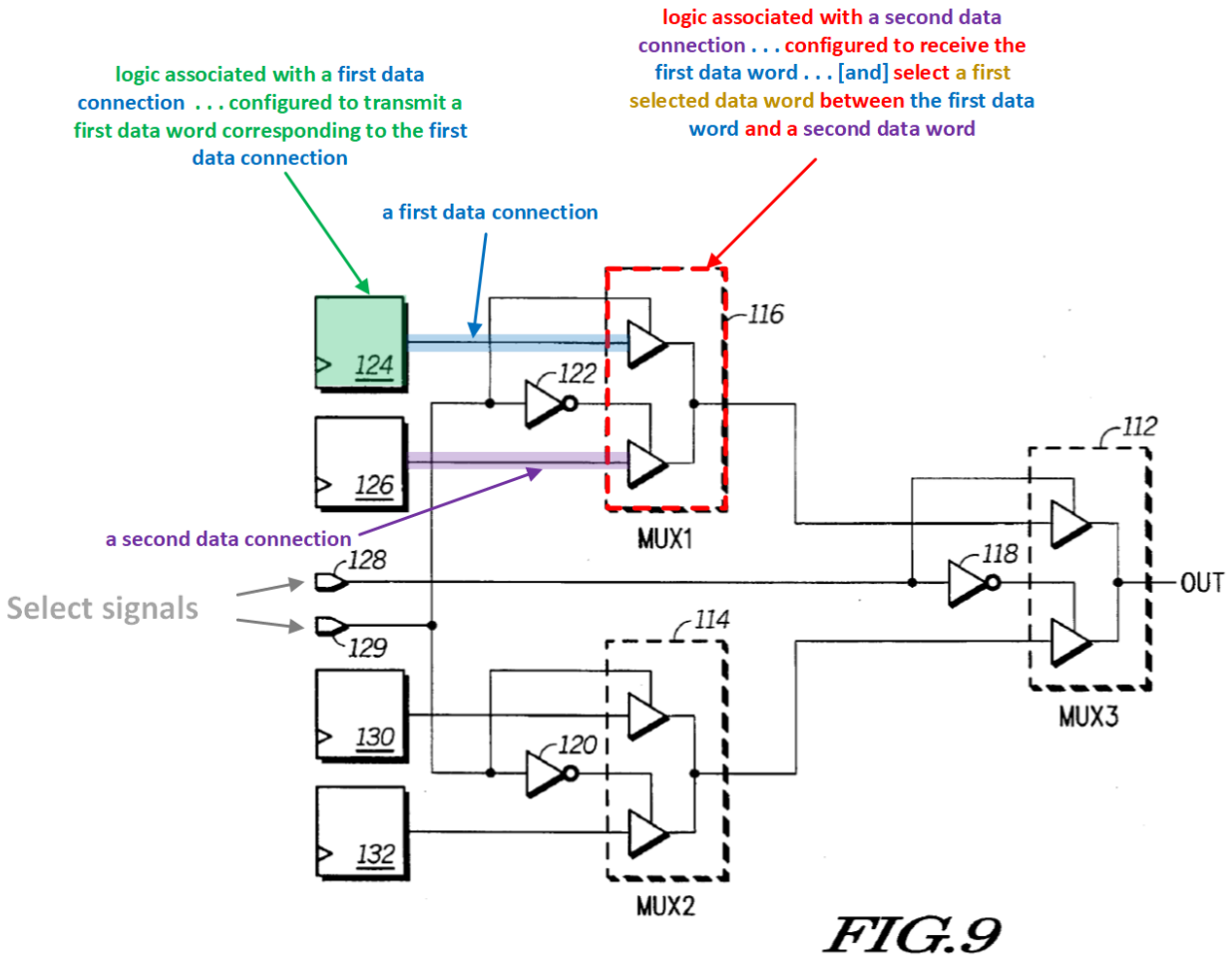
FIG. 3, [0035]-[0037]. One known use case of such a configuration was the output of an operand from a selected register in a register bank for transmission to and use in an arithmetic and logical unit (ALU). *See* EX1005, FIG. 4, [0043]. Thus, a POSITA would have understood or at least found obvious that, in m-bit processing applications in which m-bit data must, for example, be read from registers or memory, each of Pixley's tri-state multiplexors 112, 114, and 116 would have been configured as an m-bit multiplexer composed of m one-bit multiplexers, one for each bit of data—effectively stacking a set of the multiplexer circuits illustrated in FIG. 9. In the context of Pixley's FIG. 9 circuit, the stack of one-bit multiplexer circuits would receive input from a corresponding set of flip-flops 124. *See* EX1004, 7:11-14. Indeed, a register may have been implemented with a set of flip-flops—one for each stored bit.

135. Accordingly, a POSITA would have understood and found obvious that Pixley's nested multiplexers of FIG. 9 included ***logic associated with a first data connection of the plurality of data connections*** [*e.g.*, the set of flip-flops 124]***, the logic associated with the first data connection configured to transmit a first data word corresponding to the first data connection*** [*e.g.*, the set of flip-flops 124 transmits a first data word to the tri-state MUX 116], as recited in [8.b].

[8.c] logic associated with a second data connection of the plurality of data connections configured to receive the first data word corresponding to the first data connection from the logic associated with the first of the plurality of data connections, the logic associated with the second data connection configured to select a first selected data word between the first data word and a second data word corresponding to the second data connection based upon a set of select signals; and

136. In my opinion, Pixley-Watanabe renders obvious Element [8.c].

137. "FIG. 9 illustrates a circuit having tri-state MUXes 112, 114, and 116." EX1004, 7:8-9. Each of these MUXes include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "Each driver 30-32 is enabled only if the corresponding select line (sel-#) is enabled." EX1004, 3:18-19. "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.



Annotated FIG. 9 of Pixley

138. In the example of FIG. 9, when the select signal input on terminal 129 is enabled (*i.e.*, a logic high), the upper tri-state driver of tri-state MUX 116 is enabled, and the first data word (highlighted in blue above) input to the upper tri-state driver is coupled to the output of tri-state MUX 116. *See* EX1004, FIG. 9, 3:14-22. Correspondingly, the lower tri-state driver of tri-state MUX 116 receives a complement of the select signal input on terminal 129 via inverter 122, such that when the select signal input on terminal 129 is enabled, the output of inverter 122

is disabled (*i.e.*, a logic low) and the lower tri-state driver of tri-state MUX 116 is disabled. *See id.* On the other hand, when the select signal input on terminal 129 is disabled (*i.e.*, a logic low), the upper tri-state driver of tri-state MUX 116 is disabled. *See id.* And, correspondingly, the lower tri-state driver of tri-state MUX 116 receives a complement of the select signal input on terminal 129 via inverter 122, so when the select signal input on terminal 129 is disabled, the output of inverter 122 is enabled (*i.e.*, a logic high) and the lower tri-state driver of tri-state MUX 116 is enabled such that the second data word (highlighted in purple above) input to the lower tri-state driver is coupled to the output of tri-state MUX 116. *See id.*

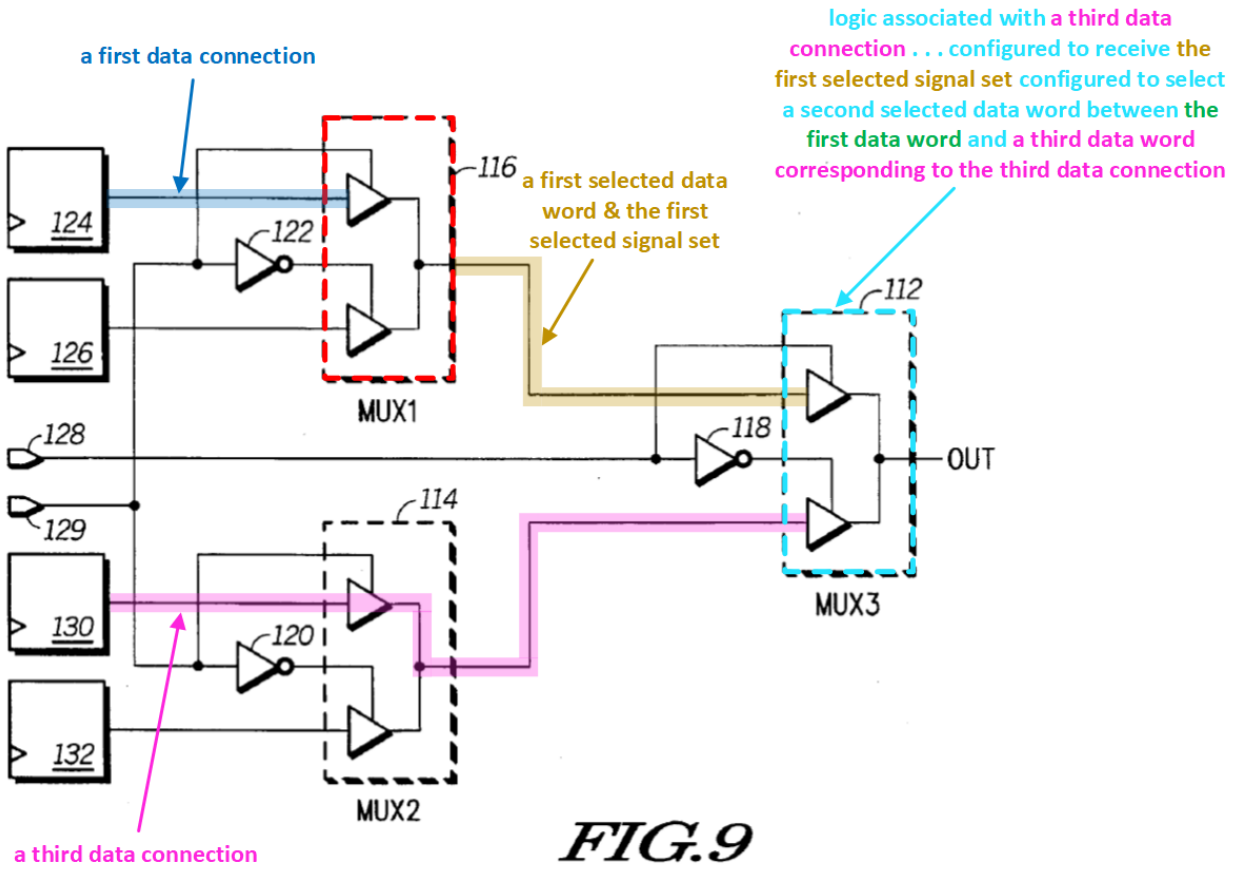
139. Accordingly, a POSITA would have understood and found obvious that Pixley's nested multiplexers of FIG. 9 included ***logic associated with a second data connection of the plurality of data connections*** [*e.g.*, tri-state MUX 116, which is connected to a second data connection output by the set of flip-flops 126] ***configured to receive the first data word corresponding to the first data connection from the logic associated with the first of the plurality of data connections*** [*e.g.*, one input of tri-state MUX 116 is the output of the set of flip-flops 124, which transmit a first data word]***, the logic associated with the second data connection configured to select a first selected data word between the first data word and a second data word corresponding to the second data connection based upon a set of select signals*** [*e.g.*, the tri-state MUX 116 selects between the first data

word received from the set of flip-flops 124 and the second data word received from the set of flip-flops 126 based on the select signal of input terminal 129 and the select signal output by inverter 122], as recited in [8.c].

[8.d] logic associated with a third data connection of the plurality of data connections configured to receive the first selected signal set, the logic associated with the third data connection configured to select a second selected data word between the first data word and a third data word corresponding to the third data connection based upon a set of select signals, the logic associated with the third data connection further configured transmit the second selected data word to a system bus.

140. In my opinion, Pixley-Watanabe renders obvious Element [8.d].

141. "FIG. 9 illustrates a circuit having tri-state MUXes 112, 114, and 116." EX1004, 7:8-9. Each of these MUXes include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "Each driver 30-32 is enabled only if the corresponding select line (sel-#) is enabled." EX1004, 3:18-19. "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.



Annotated FIG. 9 of Pixley

142. In the example of FIG. 9, when the select signal of terminal 129 is enabled, the upper tri-state driver of each of tri-state multiplexers 114 and 116 is also enabled, connecting the first and third data words received from the sets of flip-flops 124 and 130 to be connected to the inputs of tri-state multiplexer 112. *See* EX1004, FIG. 9, 3:14-22, 7:7-14. The tri-state multiplexer 112 then selects between the first and third data words output by the tri-state multiplexers 114 and 116, respectively, based on the select signal of terminal 129 and the output of inverter 118. *See id.* When the select signal input on terminal 128 is enabled (*i.e.*, a

logic high), the upper tri-state driver of tri-state MUX 112 is enabled, and the first data word (highlighted in blue above) input to the upper tri-state driver is coupled to the output OUT of tri-state MUX 112. *See id.* And when the select signal output by the inverter 118 is enabled (*i.e.*, a logic high), the second/lower tri-state driver of tri-state MUX 112 is enabled, and the second data word (highlighted in purple above) input to the lower tri-state driver is coupled to the output OUT of tri-state MUX 112.

143. Further, as described with respect to [1.pre], *supra*, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037]. One known use case of such a configuration was the output of an operand from a selected register in a register bank for transmission to and use in an arithmetic and logical unit (ALU). *See* EX1005, FIG. 4, [0043]. In this example use case, "[t]he operand output . . . that has been readout is interconnected to be sent, for example, to the arithmetic and logical unit (ALU) via a system bus . . . ." EX1005, [0043]; *see also* EX1006, FIG. 9, [0021], [0028]-[0029], [0034] (illustrating and describing "4-to-1 Multiplexer (MUX) formed with cascaded buses" and tri-state drivers).

144. Accordingly, a POSITA would have understood and found obvious that Pixley's nested multiplexers of FIG. 9 included ***logic associated with a third data connection of the plurality of data connections*** [*e.g.*, tri-state MUX 112,

which is connected to a third data connection output by the set of flip-flops 130 via tri-state MUX 114] ***configured to receive the first selected signal set*** [*e.g.*, the upper tri-state driver of tri-state MUX 112 receives the set of signals of the data word selected by tri-state MUX 116]***, the logic associated with the third data connection configured to select a second selected data word*** [*e.g.*, the data word output on OUT by tri-state MUX 112] ***between the first data word*** [*e.g.*, the data word associated with the set of flip-flops 124 received by tri-state MUX 112 via tri-state MUX 116] ***and a third data word corresponding to the third data connection*** [*e.g.*, the data word associated with the set of flip-flops 130 received by tri-state MUX 112 via tri-state MUX 114] ***based upon a set of select signals*** [*e.g.*, the select signal of input terminal 128 and the select signal output by inverter 118]***, the logic associated with the third data connection further configured transmit the second selected data word to a system bus*** [*e.g.*, a system bus connecting the output OUT of tri-state MUX 112 to an arithmetic and logical unit (ALU)], as recited in [8.d].

9. Claim 9

[9.pre] A multiplexor circuit, comprising:

145. Element [9.pre] is substantially the same as at least a portion of Element [1.pre]. In my opinion, Pixley-Watanabe renders obvious Element [9.pre] for

at least the same reasons described above with respect to Element [1.pre]. *See* Element [1.pre], *supra*.

[9.a] a first tristate driver comprising a first data input, a first data output, and a first enable input, the first tristate driver adapted to receive a first data word via the first data input representative of a first data connection;

146. In my opinion, Pixley-Watanabe renders obvious Element [9.a].

147. "FIG. 9 illustrates a circuit having tri-state MUXes 112, 114, and 116." EX1004, 7:8-9. Each of these MUXes include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "Each driver 30-32 is enabled only if the corresponding select line (sel-#) is enabled." EX1004, 3:18-19. "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.

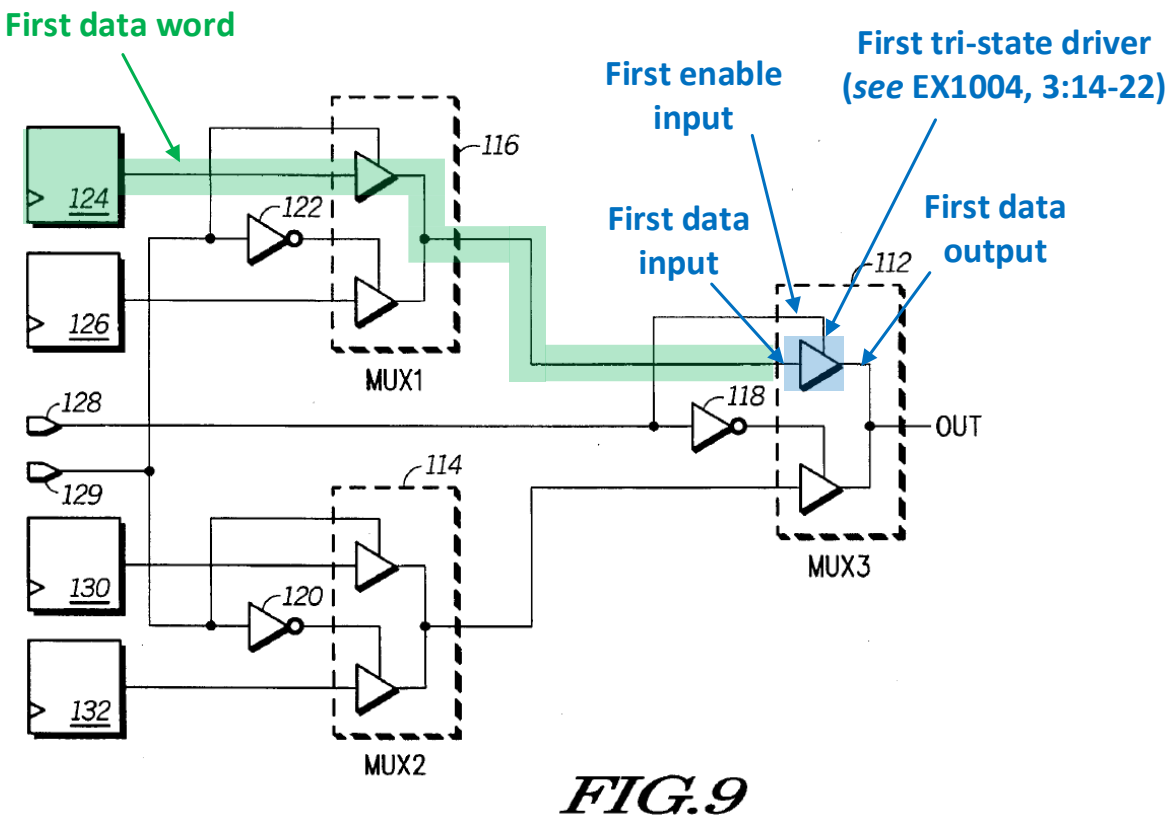


FIG.9

Annotated FIG. 9 of Pixley

148. As described with respect to Element [1.pre], *supra*, a POSITA would have understood or at least found obvious that, for certain known use cases, the 2-to-1 multiplexers described by Pixler would have been configured to multiplex m-bit data words. Specifically, it was known that "[a] 2-to-1 multiplexer of m-bit length is composed of m 2-to-1 multiplexers of one-bit length." *See, e.g.*, EX1005, [0018]. Further, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037]. One known use case of such a configuration was the output of an operand from a selected register in a register bank for transmission to and use

in an arithmetic and logical unit (ALU). *See* EX1005, FIG. 4, [0043]. Thus, a POSITA would have understood or at least found obvious that, in m-bit processing applications in which m-bit data must, for example, be read from registers or memory, each of Pixley's tri-state multiplexors 112, 114, and 116 would have been configured as an m-bit multiplexer composed of m one-bit multiplexers, one for each bit of data—effectively stacking a set of the multiplexer circuits illustrated in FIG. 9. In the context of Pixley's FIG. 9 circuit, the stacks of one-bit multiplexer circuits would receive inputs from corresponding sets of m flip-flops 124, 126, 130, 132. *See* EX1004, 7:11-14. Indeed, a register may have been implemented with a set of flip-flops—one for each stored bit.

149. When the circuit of FIG. 9 was implemented in this way, the upper tri-state driver of tri-state MUX 112 received data input from the set of flip-flops 124 via the tri-state MUX 116. *See* EX1004, FIG. 9, 7:11-14. The upper tri-state driver of tri-state MUX 112 received a select signal from terminal 128, and coupled the input data to the output of tri-state MUX 112 when the select signal of terminal 128 was enabled. *See* EX1004, FIG. 9, 3:14-22, 7:7-14.

150. Accordingly, a POSITA would have understood and found obvious that Pixely's nested multiplexers of FIG. 9 included ***a first tristate driver comprising a first data input, a first data output, and a first enable input*** [*e.g.*, the upper tri-state driver of tri-state MUX 112]***, the first tristate driver adapted to receive a***

first data word via the first data input representative of a first data connection

[*e.g.*, the upper tri-state driver of tri-state MUX 112 receiving the data word of flip-flops 124 via tri-state MUX 116], as recited in [9.a].

[9.b] a second tristate driver comprising a second data input, a second data output, and a second enable input, the second tristate driver adapted to receive a second data word representative of a second data connection;

151. In my opinion, Pixley-Watanabe renders obvious Element [9.b].

152. As described with respect to Element [9.a], *supra*, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.

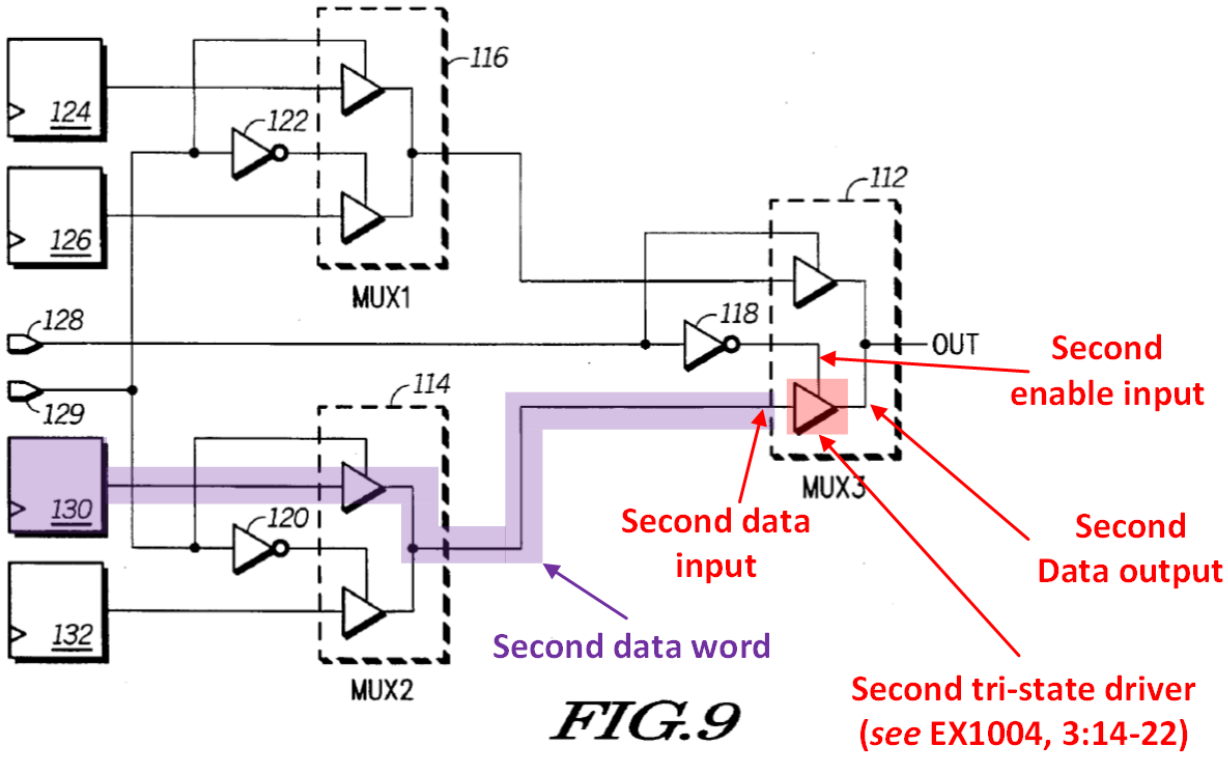


FIG.9

Annotated FIG. 9 of Pixley

153. As described with respect to [9.a], *supra*, a POSITA would have understood or at least found obvious that, in m-bit processing applications in which m-bit data must, for example, be read from registers or memory, each of Pixley's tri-state multiplexors 112, 114, and 116 would have been configured as an m-bit multiplexer composed of m one-bit multiplexers and these multiplexors would receive inputs from corresponding sets of m flip-flops 124, 126, 130, 132. *See, e.g.*, EX1005, FIGS. 3, 4, [0018], [0035]-[0037], [0043]; EX1004, 7:11-14. When the circuit of FIG. 9 was implemented in this way, the lower tristate driver of tri-state MUX 112 received data input from the set of flip-flops 130 via the tri-state MUX

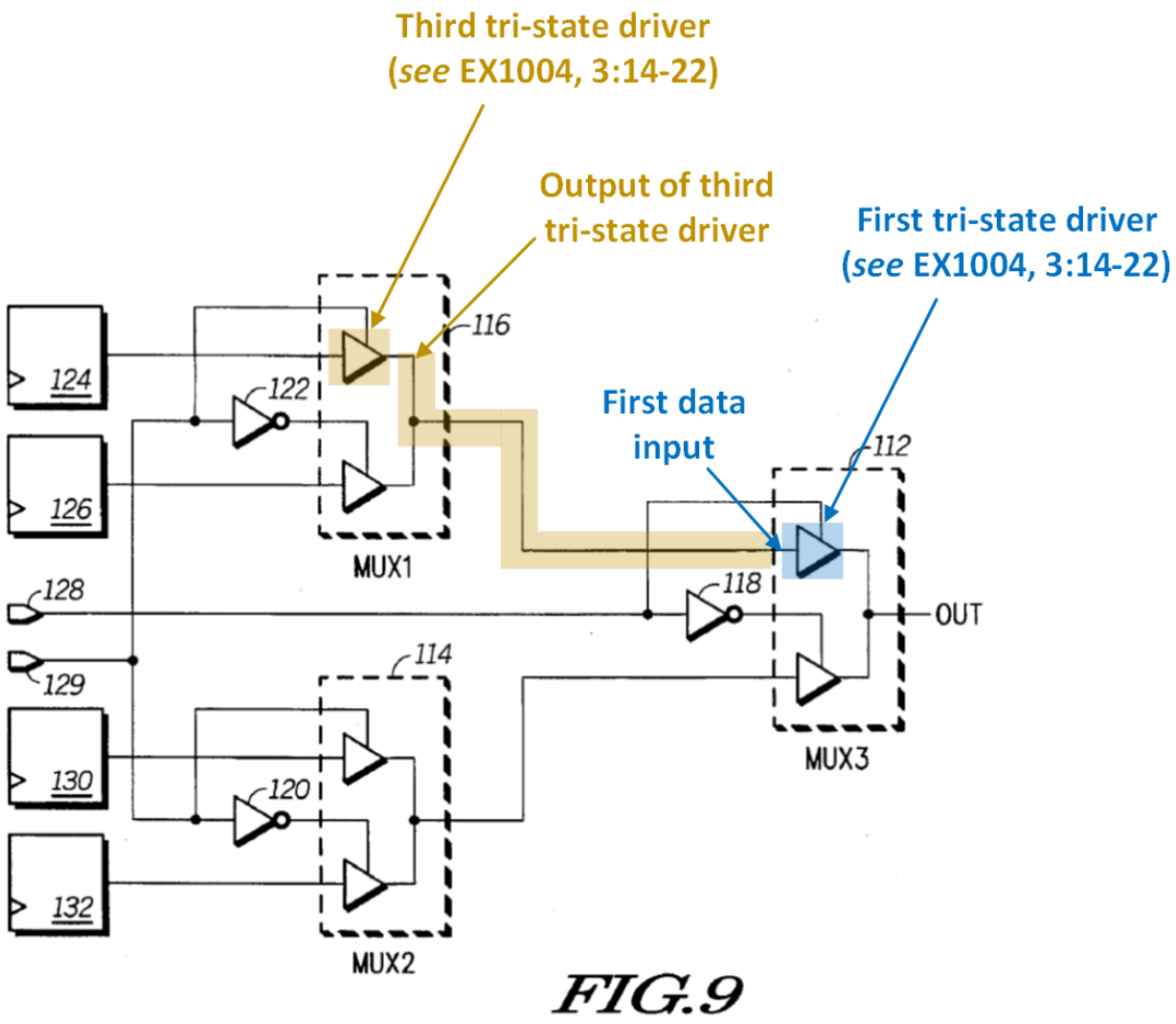
114. *See* EX1004, FIG. 9, 7:11-14. The lower tri-state driver of tri-state MUX 112 received a select signal from inverter 118, and coupled the input data to the output of tri-state MUX 112 when the select signal output by inverter 118 was enabled. *See* EX1004, FIG. 9, 3:14-22, 7:7-14.

154. Accordingly, a POSITA would have understood and found obvious that Pixely's nested multiplexers of FIG. 9 included ***a second tristate driver comprising a second data input, a second data output, and a second enable input*** [*e.g.*, the lower tri-state driver of tri-state MUX 112]***, the second tristate driver adapted to receive a second data word representative of a second data connection*** [*e.g.*, the lower tri-state driver of tri-state MUX 112 receiving the data word of flip-flops 130 via tri-state MUX 114], as recited in [9.b].

[9.c] a third tristate driver having an output coupled to the first data input of the first tristate driver; and

155. In my opinion, Pixley-Watanabe renders obvious Element [9.c].

156. As described with respect to Element [9.a], *supra*, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors).



Annotated FIG. 9 of Pixley

157. As shown above in FIG. 9, the upper tri-state driver of tri-state MUX 116 has an output connected to the data input of the upper tri-state driver of tri-state MUX 112. *See* EX1004, FIG. 9, 7:11-14.

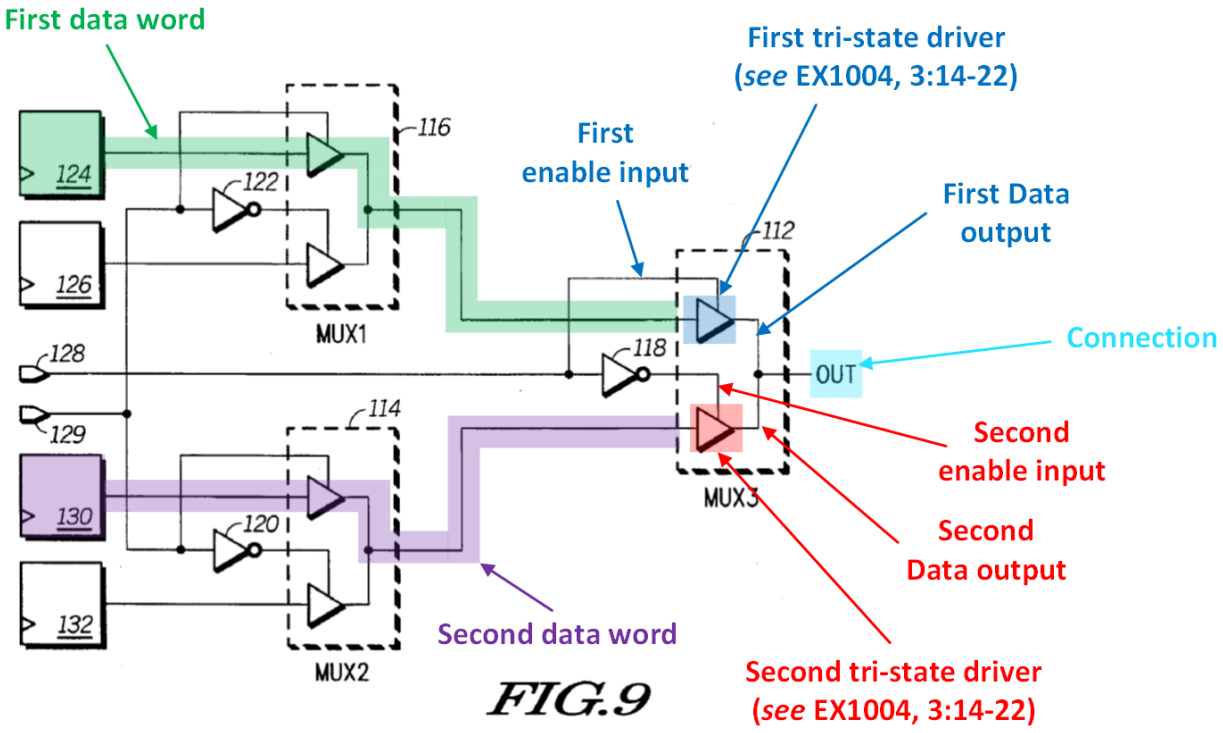
158. Accordingly, a POSITA would have understood and found obvious that Pixely's nested multiplexers of FIG. 9 included ***a third tristate driver*** [*e.g.*, the upper tri-state driver of tri-state MUX 116] ***having an output coupled to the first***

data input of the first tristate driver [*e.g.*, the upper tri-state driver of tri-state MUX 112], as recited in [9.c].

[9.d] a connection adapted to receive, via the data output of the first tristate driver, the first data word, if a first signal is asserted at the first enable input of the first tristate driver, the connection further adapted to receive, via the second data output of the second tristate driver, the second data word, if a second signal is asserted at the enable input of the second tristate driver.

159. In my opinion, Pixley-Watanabe renders obvious Element [9.d].

160. Pixley describes that "[w]hen the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.



Annotated FIG. 9 of Pixley

161. In the example of FIG. 9, when the select signal of terminal 129 is enabled, the upper tri-state driver of each of tri-state multiplexers 114 and 116 is also enabled, connecting the first and second data words received from the sets of flip-flops 124 and 130 to be connected to the inputs of tri-state multiplexer 112. *See* EX1004, FIG. 9, 3:14-22, 7:7-14. The tri-state multiplexer 112 then selects between the first and second data words output by the tri-state multiplexers 114 and 116, respectively, based on the select signal of terminal 129 and the output of inverter 118. *See id.* When the select signal input on terminal 128 is enabled (*i.e.*, a logic high), the upper tri-state driver of tri-state MUX 112 is enabled, and the first data word (highlighted in green above) input to the upper tri-state driver is coupled to the output OUT of tri-state MUX 112. *See id.* And when the select signal output by the inverter 118 is enabled (*i.e.*, a logic high), the second/lower tri-state driver of tri-state MUX 112 is enabled, and the second data word (highlighted in purple above) input to the lower tri-state driver is coupled to the output OUT of tri-state MUX 112. *See id.*

162. Accordingly, a POSITA would have understood and found obvious that Pixely's nested multiplexers of FIG. 9 included ***a connection*** [*e.g.*, the output OUT of the tri-state multiplexer 112] ***adapted to receive, via the data output of the first tristate driver*** [*e.g.*, the upper tri-state driver of tri-state multiplexer 112]***, the first data word*** [*e.g.*, the data word input to the upper tri-state driver of the tri-state

multiplexer 112 from the flip-flops 124 via tri-state multiplexer 116]***, if a first signal is asserted at the first enable input of the first tristate driver*** [*e.g.*, the input of the upper tri-state driver is coupled to the output OUT of the tri-state multiplexer 112 when the select signal of terminal 128 is enabled]***, the connection further adapted to receive, via the second data output of the second tristate driver*** [*e.g.*, the lower tri-state driver of tri-state multiplexer 112]***, the second data word*** [*e.g.*, the data word input to the lower tri-state driver of the tri-state multiplexer 112 from the flip-flops 130 via tri-state multiplexer 114]***, if a second signal is asserted at the enable input of the second tristate driver*** [*e.g.*, the input of the lower tri-state driver is coupled to the output OUT of the tri-state multiplexer 112 when the select signal output by inverter 118 is enabled], as recited in [9.d].

10. Claim 10

[10] The multiplexor circuit of claim 9, further comprising select logic configured to assert the first signal if an application program desires the first data word representative of the first data connection and configured to assert the second signal if the application program desires the second data word representative of the second data connection.

163. In my opinion, Pixley-Watanabe renders obvious Claim [10].

164. Pixley describes that "control/select signals to the MUXes 112, 114, and 116 are provided . . . via two input terminals/pins 128 and 129." EX1004, 7:11-14. Pixley describes that these input terminals/pins select the output of the tri-state multiplexers 112, 114, and 116. *See* EX1004, 3:16-22.

165. A POSITA would have understood or at least found obvious that the select signals asserted on input terminals/pins 128 and 129—which control which data words are selected by the tri-state multiplexers of FIG. 9—would have been asserted by select logic controlled by an application program. Pixley does not limit from where the signals asserted on input terminals/pins 128 and 129 are received. *See generally* EX1004, 7:7-8:3. As described with respect to Element [1.pre], *supra*, one known use case of a configuration of nested/cascading multiplexers like that shown in Pixley's FIG. 9 was the selection of an operand from a selected register in a register bank for transmission to and use in an arithmetic and logical unit (ALU). *See* EX1005, FIG. 4, [0043]. In this use case, it was further known that the select signal indicating the register number corresponding to the selected operand was received by logic (*e.g.*, an instruction register/decoder) controlled by an application program of a microprocessor. *See* EX1005, [0004]-[0006] (describing that a user "develop[s] program sequences" for a microprocessor that includes the nested/cascading multiplexers used to read the registers), [0035]-[0037], [0040] (describing that an "instruction register/decoder 11 is adapted to retain and decode an instruction received from other circuit components . . . of the microprocessor 10"), [0043] (describing that the nested/cascading multiplexers are "adapted to read out the contents, or register value, of a register, whose number is specified by

the operand selection inputs 301, 302 and 303 (register number) output by the instruction register/decoder 11"), [0050]. Indeed, a POSITA would have understood and found obvious that, in a case that the nested/cascading multiplexers are part of a microprocessor, the select signals would be provided by logic controlled by one or more application programs, because microprocessors were well known to be instructed by application programs.

166. Accordingly, a POSITA would have understood and found obvious that Pixely's nested multiplexers of FIG. 9 included ***select logic*** [e.g., an instruction register/decoder] ***configured to assert the first signal if an application program desires the first data word representative of the first data connection*** [*e.g.*, an application directing a microprocessor that includes the nested/cascading multiplexers of Pixley's FIG. 9 asserts the select signal of input terminal 128, enabling the upper tri-state driver] ***and configured to assert the second signal if the application program desires the second data word representative of the second data connection*** [*e.g.,* an application directing a microprocessor that includes the nested/cascading multiplexers of Pixley's FIG. 9 deasserts the select signal of input terminal 128, resulting in an assertion of the output of inverter 118, enabling the lower tri-state driver], as recited in Claim [10].

11. Claim 11

[11] The multiplexor circuit of claim 10, wherein the select logic deasserts the first signal if the application program does not desire the first data word and de-asserts the second signal if the application program does not desire the second data word.

167. In my opinion, Pixley-Watanabe renders obvious Claim [11].

168. Pixley describes that "control/select signals to the MUXes 112, 114, and 116 are provided . . . via two input terminals/pins 128 and 129." EX1004, 7:11-14. Pixley further describes that these input terminals/pins select the output of the tri-state multiplexers 112, 114, and 116. *See* EX1004, 3:16-22. And, as described with respect to Claim [10], *supra*, a POSITA would have understood or at least found obvious that the select signals asserted on input terminals/pins 128 and 129—which control which data words are selected by the tri-state multiplexers of FIG. 9—would have been asserted by select logic controlled by an application program.

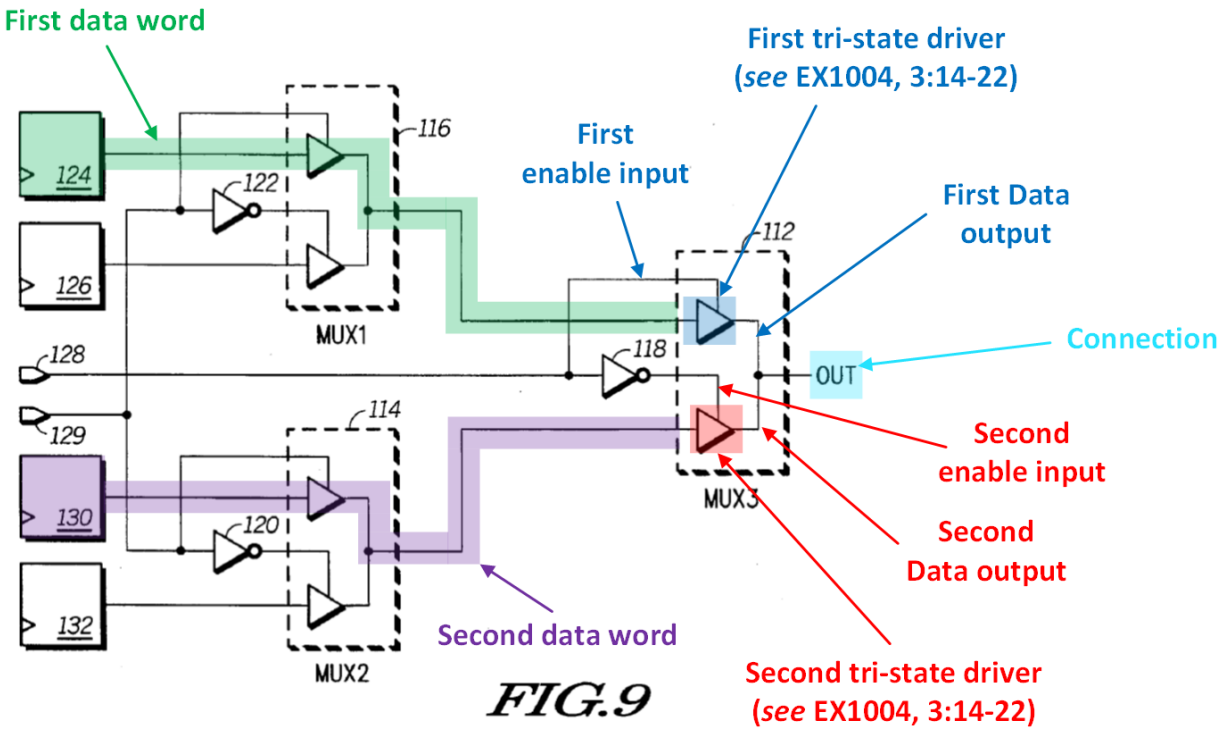


FIG.9

Annotated FIG. 9 of Pixley

169. In the example of FIG. 9, when the select signal of terminal 129 is enabled, the upper tri-state driver of each of tri-state multiplexers 114 and 116 is also enabled, connecting the first and second data words received from the sets of flip-flops 124 and 130 to be connected to the inputs of tri-state multiplexer 112. *See* EX1004, FIG. 9, 3:14-22, 7:7-14. The tri-state multiplexer 112 then selects between the first and second data words output by the tri-state multiplexers 114 and 116, respectively, based on the select signal of terminal 129 and the output of inverter 118. *See id.* When the select signal input on terminal 128 is disabled (*i.e.*, a logic low), the upper tri-state driver of tri-state MUX 112 is disabled, and the first

data word (highlighted in green above) input to the upper tri-state driver is not selected for output. *See id.* And when the select signal output by the inverter 118 is disabled (*i.e.*, a logic low), the second/lower tri-state driver of tri-state MUX 112 is disabled, and the second data word (highlighted in purple above) input to the lower tri-state driver is not selected for output. *See id.*

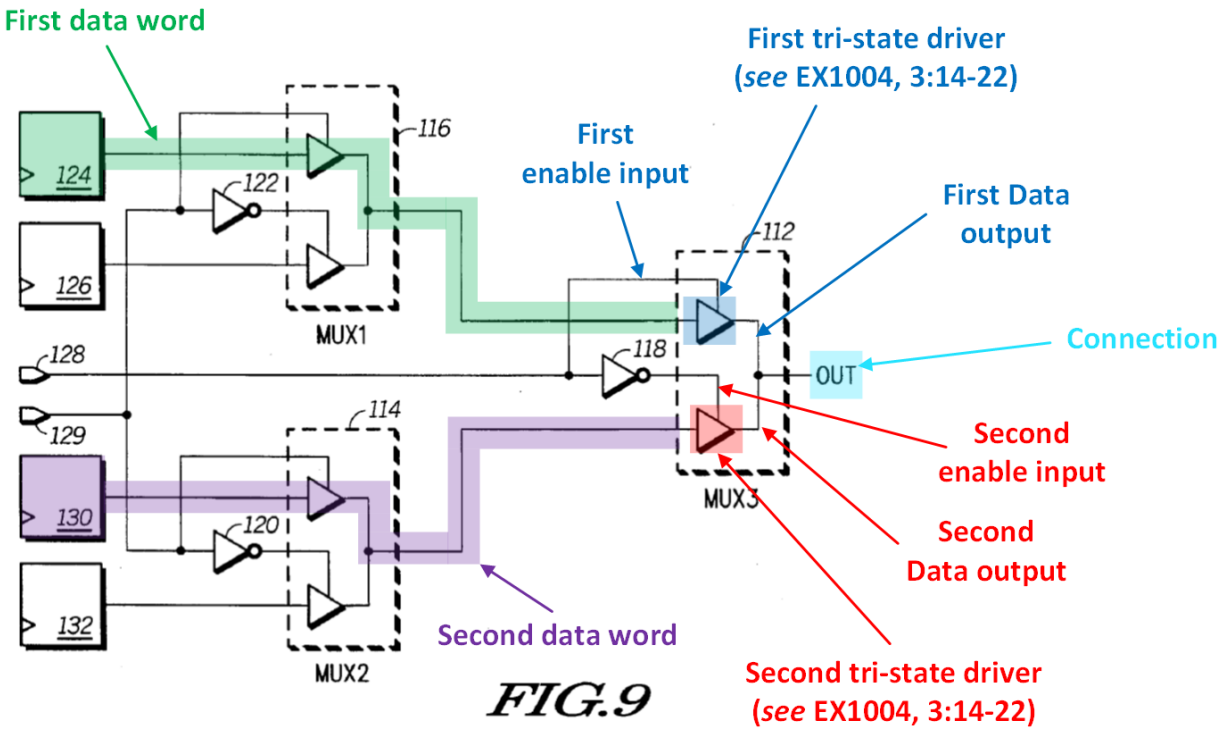
170. Accordingly, a POSITA would have understood and found obvious that, in Pixely's nested multiplexers of FIG. 9, ***the select logic deasserts the first signal if the application program does not desire the first data word*** [*e.g.*, the select logic deasserts the select signal input on terminal 128] ***and deasserts the second signal if the application program does not desire the second data word*** [*e.g.*, the select logic asserts the select signal input on terminal 128, which results in the output of inverter 118 being deasserted], as recited in claim [11].

12. Claim 12

[12] The multiplexor circuit of claim 11, wherein the first signal and the second signal enable and disable the first and second tristate drivers simultaneously.

171. In my opinion, Pixley-Watanabe renders obvious Claim [12].

172. Pixley describes that "control/select signals to the MUXes 112, 114, and 116 are provided . . . via two input terminals/pins 128 and 129." EX1004, 7:11-14. Pixley further describes that these input terminals/pins select the output of the tri-state multiplexers 112, 114, and 116. *See* EX1004, 3:16-22.



Annotated FIG. 9 of Pixley

173. In the example of FIG. 9, when the select signal of terminal 129 is enabled, the output of inverter 118 is disabled, because the select signal of terminal 129 is the input of inverter 118 and therefore the two select signals are complements of one another. *See* EX1004, FIG. 9, 7:7-14. As described by Pixley, "a tri-state multiplexer should be carefully checked during design to ensure: (1) that no two select lines which have different input logic values are enabled at or around a critical period of time; and (2) that at least one select line is enabled during all critical time periods." EX1004, 4:21-26. Pixley further teaches that the nested multiplexer circuit illustrated in FIG. 9 meets these two conditions. *See* EX1004, 7:14-16.

174. Accordingly, a POSITA would have understood and found obvious that, in Pixely's nested multiplexers of FIG. 9, ***the first signal and the second signal enable and disable the first and second tristate drivers simultaneously*** [*e.g.*, when the select signal of input terminal/pin 128 is asserted, the output of inverter 118 is simultaneously deasserted, and when the select signal of input terminal/pin 128 is deasserted, the output of inverter 118 is simultaneously asserted], as recited in [12].

13. Claim 13

[13.pre] The multiplexor circuit of claim 12, further comprising:

[13.a] a first latch circuit configured to transmit the first signal to the first tristate driver upon a leading edge of a clock signal; and

175. Element [13.a] is substantially similar to at least part of element [7.a]. It would have been obvious for the same reason. *See* Element [7.a], *supra*.

[13.b] a second latch circuit configured to transmit the second signal to the second tristate driver upon the leading edge of the clock signal.

176. Element [13.b] is substantially similar to at least part of element [7.a]. It would have been obvious for the same reason. *See* Element [7.b], *supra*.

14. Claim 14

[14.pre] A method for distributing multiplexing, comprising:

177. In my opinion, Pixley-Watanabe renders obvious Element [14.pre].

178. For example, Pixley "provides a method and apparatus to perform checking of tri-state multiplexers to ensure that the tri-state multiplexers are operating in a proper manner." EX1004, 2:60-63. Further, Pixley describes the method by which the tri-state multiplexers illustrated in FIGS. 1-11 are operated through the input of data at data inputs and the assertion of select signals supplied to the tri-state multiplexers. *See* EX1004, 3:16-22, 7:7-14.

179. Accordingly, a POSITA would have understood or at least found obvious that Pixley teaches a ***method for distributing multiplexing***, as recited in [14.pre].

[14.a] transmitting a first data word from a first tristate driver to a second tristate driver;

180. Element [14.a] is substantially similar to at least a portion of Elements [1.b] and [1.c.ii]. In my opinion, Pixley-Watanabe renders obvious Element [14.a] for at least the same reasons described above with respect to Elements [1.b] and [1.c.ii]. *See* Elements [1.b], [1.c.ii], *supra*.

181. For example, Pixley describes that each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.

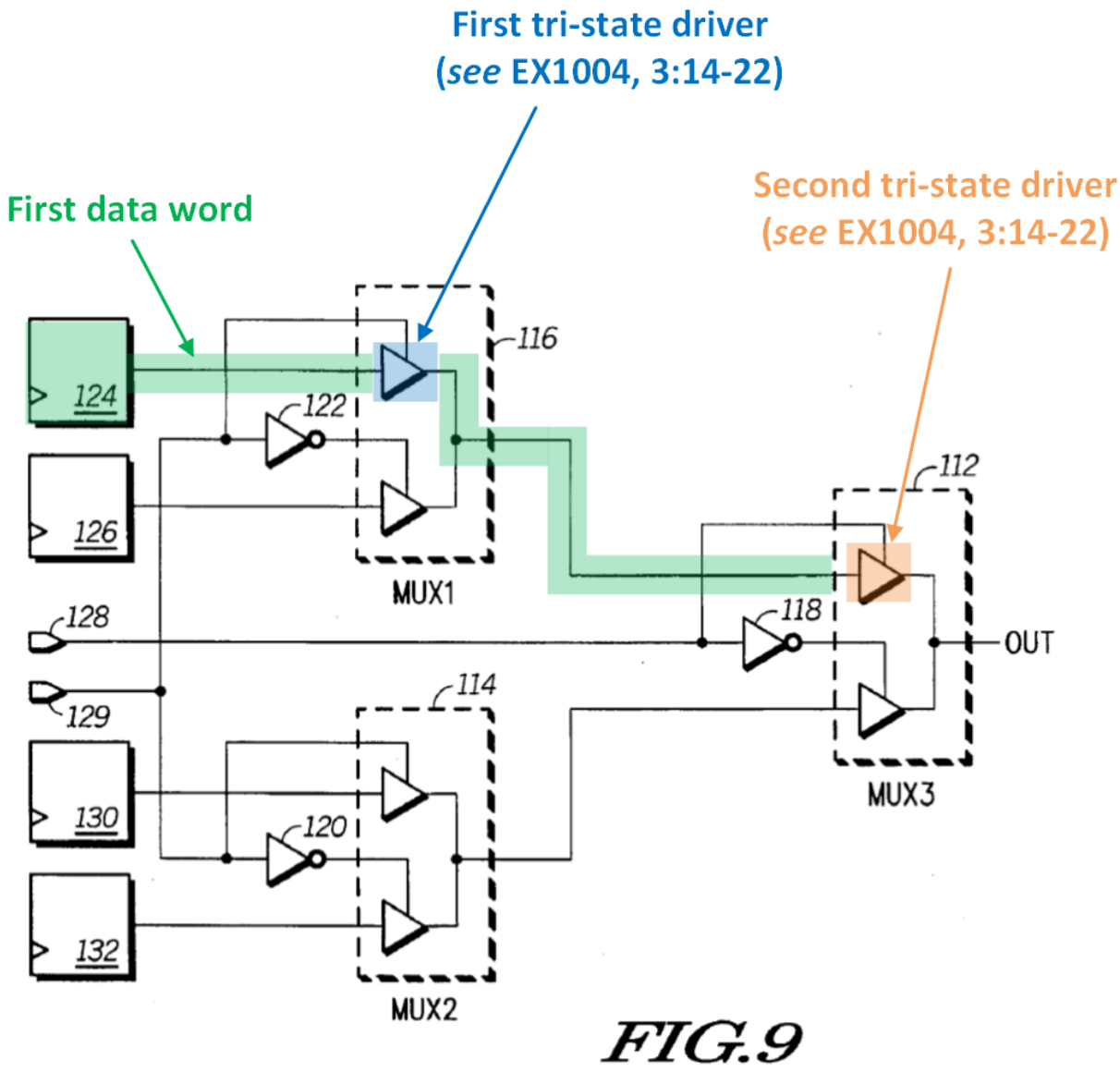


FIG. 9

Annotated FIG. 9 of Pixley

182. In the example of FIG. 9, when the select signal input on terminal/pin 129 is asserted, the upper tri-state driver of tri-state MUX 116 couples the first data word received from flip-flops 124 to the output of tri-state MUX 116, which is input into the upper tri-state driver of tri-state MUX 112. *See* EX1004, FIG. 9, 7:7-14.

183. Accordingly, a POSITA would have understood and found obvious that Pixely's nested multiplexers of FIG. 9 ***transmit[s] a first data word*** [*e.g.*, the data word provided by flip-flops 124] ***from a first tristate driver*** [*e.g.*, the upper tri-state driver of tri-state MUX 116] ***to a second tristate driver*** [*e.g.*, the upper tri-state driver of tri-state MUX 116], as recited in [14.a].

[14.b] receiving a second data word; and

184. Element [14.b] is substantially similar to at least a portion of Element [1.c.i]. In my opinion, Pixley-Watanabe renders obvious Element [14.b] for at least the same reasons described above with respect to Element [1.c.i]. *See* Element [1.c.i], *supra*.

185. With respect to FIG. 9, Pixely describes that "[d]ata inputs and control/select signals to the MUXes 112, 114, and 116 are provided via four D flip-flops 124, 126, 130, and 132 and via two input terminals/pins 128 and 129." EX1004, 7:11-14. Further, as illustrated in FIG. 9, the outputs of tri-state MUXs 114 and 116 are inputs to tri-state MUX 112, connecting tri-state MUX 112 to each of data inputs 124, 126, 130, and 132 via tri-state MUXs 114 and 116. *See* EX1004, FIG. 9.

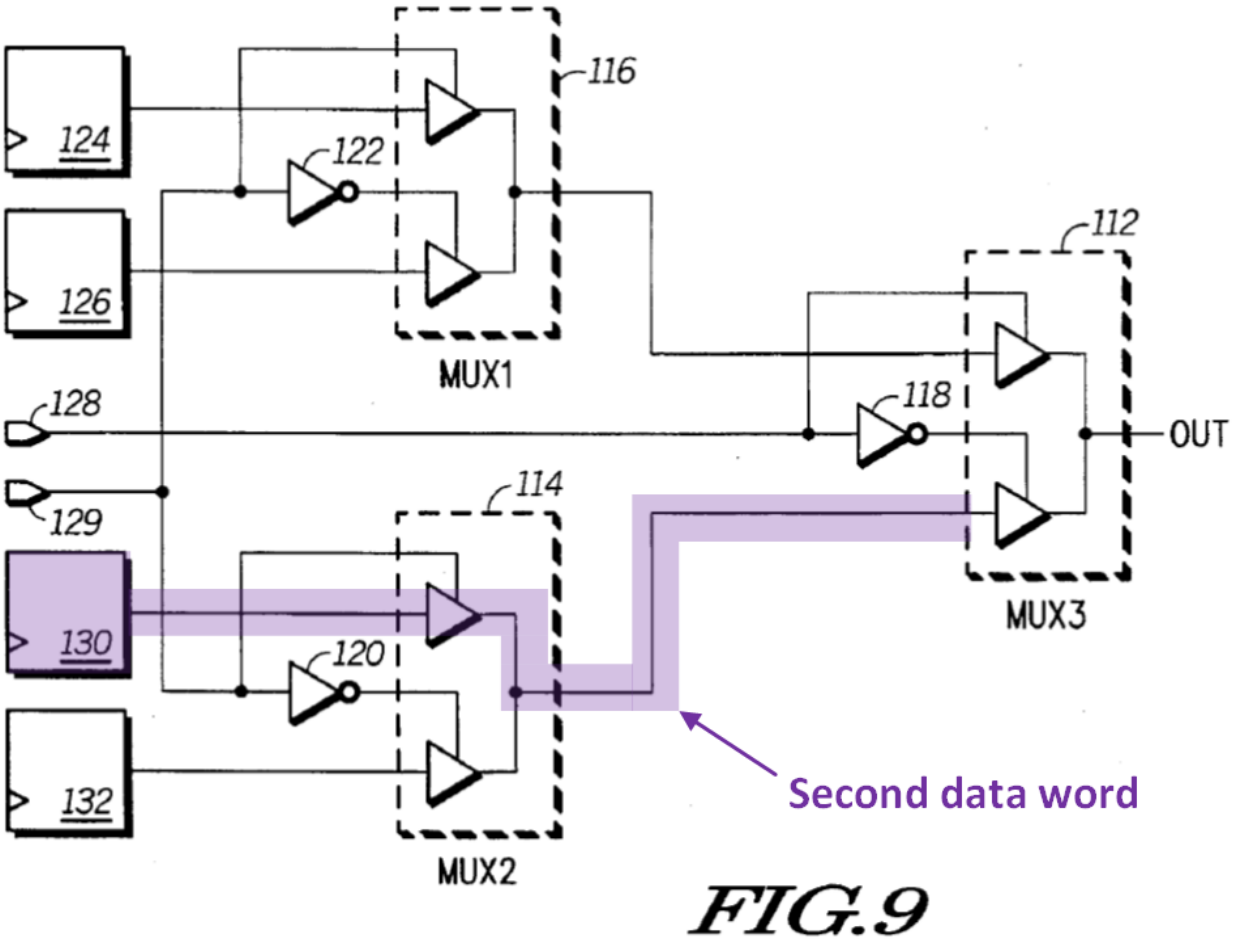


FIG. 9

Annotated FIG. 9 of Pixley

186. In the example of FIG. 9, when the select signal input on terminal 129 is enabled (*i.e.*, a logic high), the upper tri-state driver of tri-state MUX 114 is enabled, and the second data word (highlighted in purple above) input to the upper tri-state driver is coupled to the output of tri-state MUX 114 and correspondingly to one of the inputs of tri-state MUX 112. *See* EX1004, FIG. 9, 3:14-22.

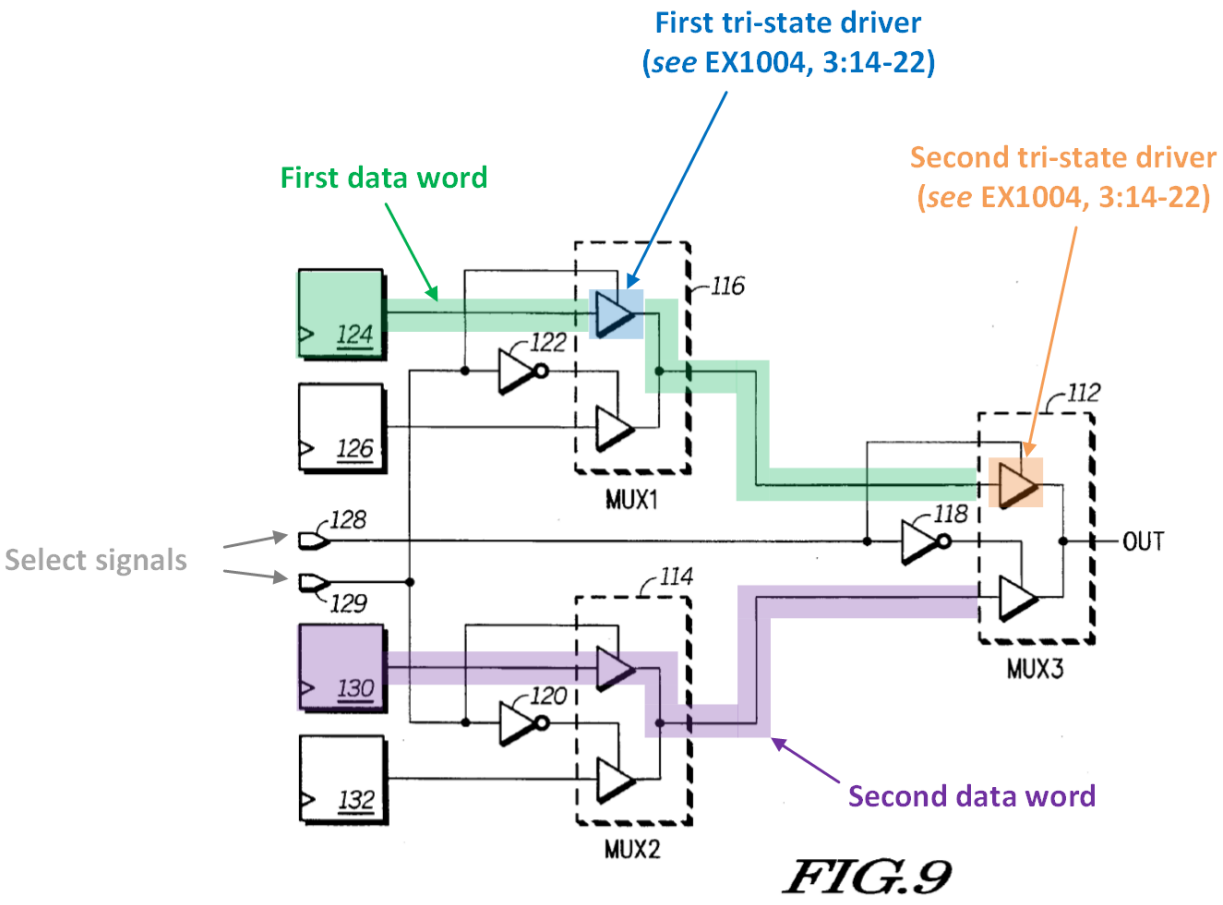
187. Accordingly, a POSITA would have understood and found obvious that Pixely's nested multiplexers of FIG. 9 ***receiv[es] a second data word*** [*e.g.*, the

data word received at an input of the tri-state MUX 112 from the flip-flops 130 via tri-state MUX 114], as recited in [14.b].

[14.c] selecting for transmission between the first data word and the second data word based upon a set of select signals;

188. Element [14.c] is substantially similar to at least a portion of Element [1.c.i]. In my opinion, Pixley-Watanabe renders obvious Element [14.c] for at least the same reasons described above with respect to Element [1.c.i]. *See* Element [1.c.i], *supra*.

189. As described with respect to Element [1.b], *supra*, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.



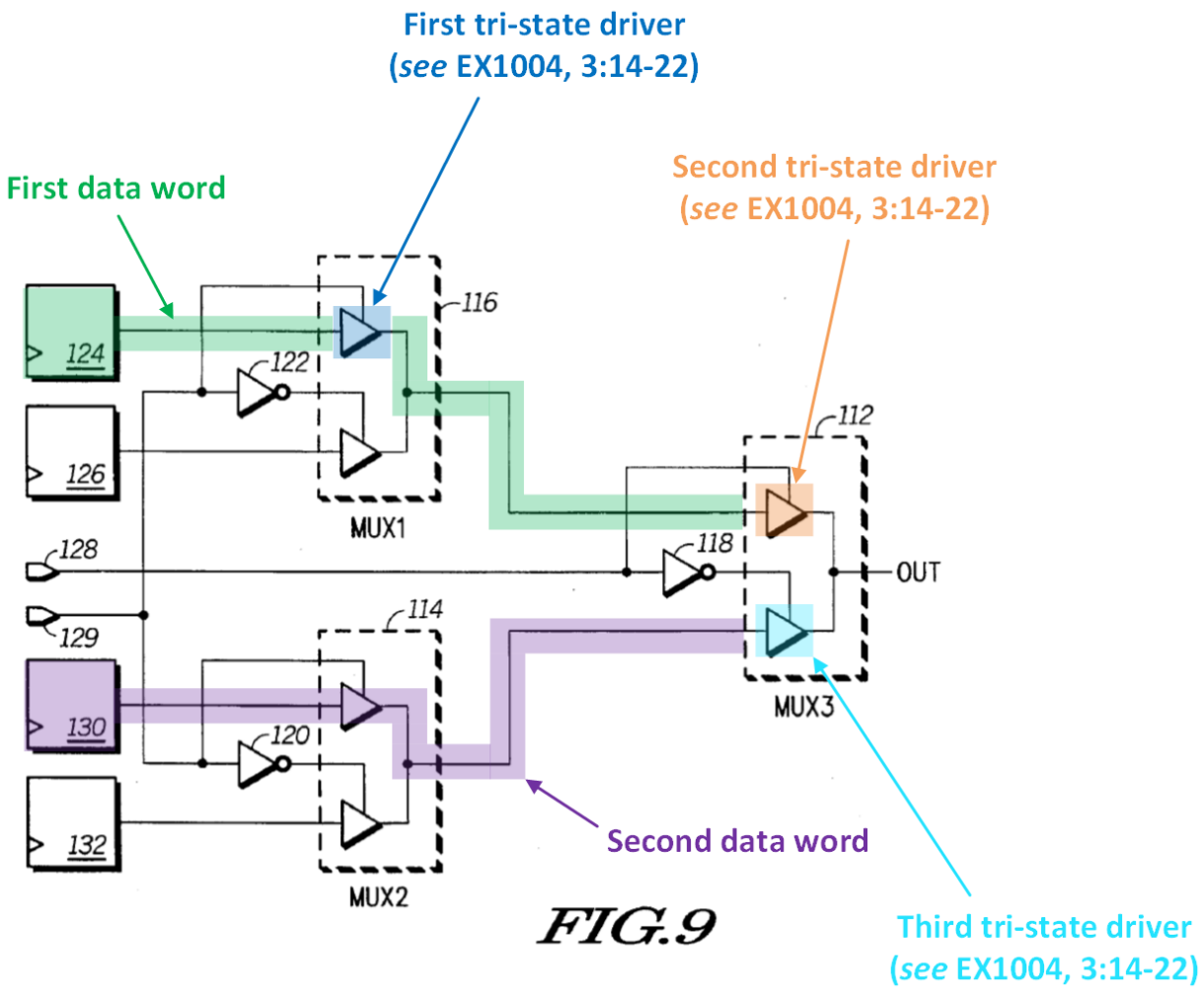
Annotated FIG. 9 of Pixley

190. In the example of FIG. 9, when the select signal of terminal 129 is enabled, the upper tri-state driver of each of tri-state multiplexers 114 and 116 is also enabled, connecting the first and second data words received from the sets of flip-flops 124 and 130 to be connected to the inputs of tri-state multiplexer 112. *See* EX1004, FIG. 9, 3:14-22, 7:7-14. The tri-state multiplexer 112 then selects between the first and second data words output by the tri-state multiplexers 114 and 116, respectively, based on the select signal of terminal 129 and the output of inverter 118. *See id.*

191. Accordingly, a POSITA would have understood and found obvious that Pixely's nested multiplexers of FIG. 9 ***select[s] for transmission between the first data word and the second data word based upon a set of select signals*** [*e.g.*, the tri-state MUX 112 select between its data inputs based on the select signal of terminal 129 and the output of inverter 118], as recited in Element [14.c].

[14.d] driving a first connection with the first data word via the second tristate driver if the first data word is selected by the selecting; and

192. Element [14.d] is substantially similar to at least a portion of Element [1.c.ii]. In my opinion, Pixley-Watanabe renders obvious Element [14.d] for at least the same reasons described above with respect to Element [1.c.ii]. *See* Element [1.c.ii], *supra*.



Annotated FIG. 9 of Pixley

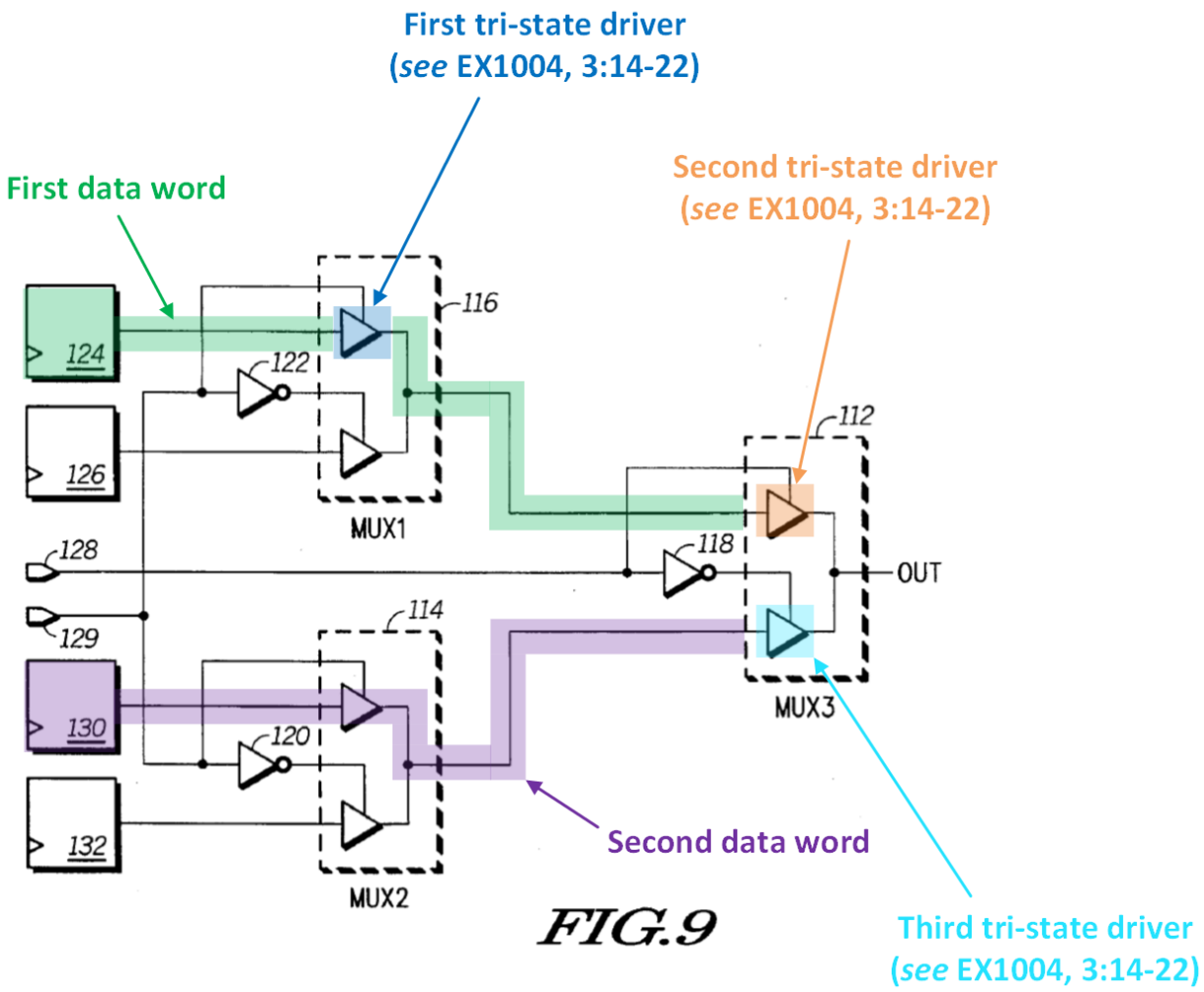
193. In the example of FIG. 9, when the select signal input on terminal 129 is enabled (*i.e.*, a logic high), the upper tri-state driver of tri-state MUX 114 is enabled, and the second data word (highlighted in purple above) input to the upper tri-state driver is coupled to the output OUT of tri-state MUX 114 and correspondingly to one of the inputs of tri-state MUX 112. *See* EX1004, FIG. 9, 3:14-22.

194. Accordingly, a POSITA would have understood and found obvious that Pixely's nested multiplexers of FIG. 9 ***driv[es] a first connection*** [*e.g.*, the

output OUT of tri-state MUX 112] ***with the first data word via the second tristate driver*** [*e.g.*, the upper tri-state driver of tri-state MUX 112] ***if the first data word is selected by the selecting*** [*e.g.*, if the input of terminal 128 is asserted], as recited in [14.d].

[14.e] driving the first connection with the second data word via a third tristate driver if the second data word is selected by the selecting.

195. Element [14.e] is substantially similar to at least a portion of Element [1.c.ii]. In my opinion, Pixley-Watanabe renders obvious Element [14.e] for at least the same reasons described above with respect to Element [1.c.ii]. *See* Element [1.c.ii], *supra*.



Annotated FIG. 9 of Pixley

196. In the example of FIG. 9, when the select signal input on terminal 129 is enabled (*i.e.*, a logic high), the first/upper tri-state driver of tri-state MUX 116 is enabled, and the first data word (highlighted in green above) input to the first/upper tri-state driver is coupled to the output of tri-state MUX 116 and correspondingly to the other input of tri-state MUX 112. *See* EX1004, FIG. 9, 3:14-22, 7:7-14.

197. Accordingly, a POSITA would have understood and found obvious that Pixely's nested multiplexers of FIG. 9 ***driv[es] the first connection*** [*e.g.*, the output OUT of tri-state MUX 112] ***with the second data word via a third tristate driver*** [*e.g.*, the lower tri-state driver of tri-state MUX 112] ***if the second data word is selected by the selecting*** [*e.g.*, if the output of inverter 118 is asserted], as recited in [14.e].

15. Claim 15

[15] The method of claim 14, wherein the selected data word is transmitted to a discrete integrated circuit (IC) component.

198. In my opinion, Pixley-Watanabe renders obvious Claim 15.

199. As described with respect to [1.pre], *supra*, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037]. One known use case of such a configuration was the output of an operand from a selected register in a register bank for transmission to and use in an arithmetic and logical unit (ALU). *See* EX1005, FIG. 4, [0043]. In this example use case, "[t]he operand output . . . that has been readout is interconnected to be sent, for example, to the arithmetic and logical unit (ALU) via a system bus . . . ." EX1005, [0043]; *see also* EX1006, FIG. 9, [0021], [0028]-[0029], [0034] (illustrating and describing "4-to-1 Multiplexer (MUX) formed with cascaded buses" and tri-state drivers).

Indeed, FIG. 9 depicts an embodiment of the circuit model with nested multiplexors and an output. EX1004, 2:36-38, 7:7-10.

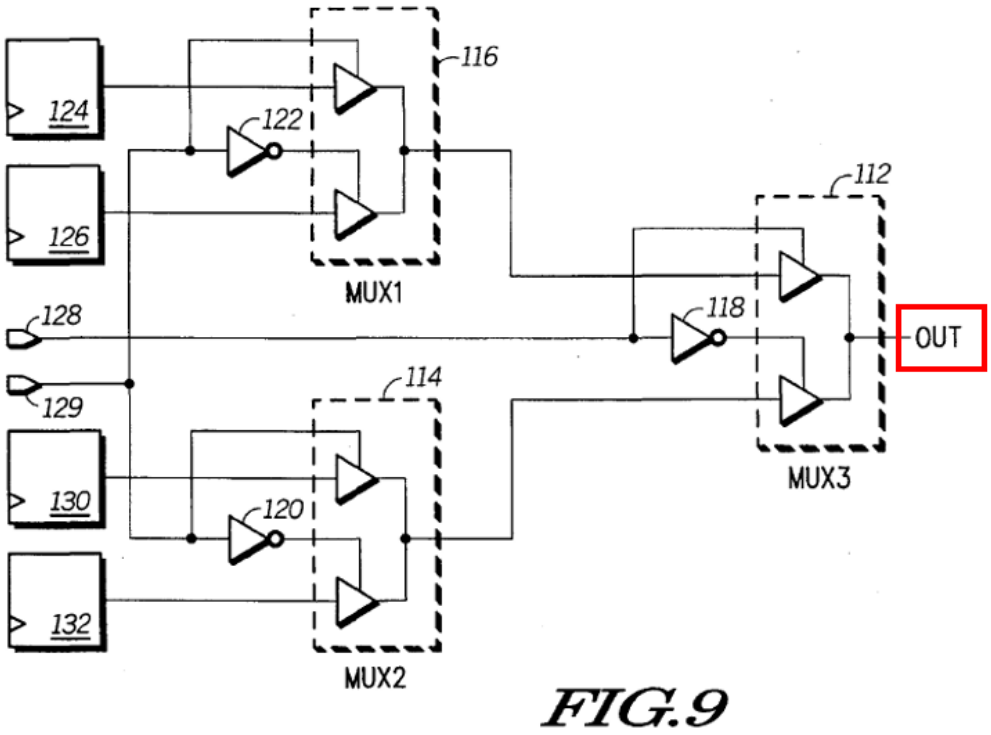


FIG.9

Annotated FIG. 9 of Pixley

200. Applying the MUX circuit shown in FIG. 9 to the example use case described by Watanabe, a POSITA would have at least found it obvious that the output OUT of tri-state MUX 112 would have been connected to a system bus to provide data words read out from memory, such as registers, for use in an integrated circuit (IC) component, such as an ALU. *See* EX1005, FIGS. 3, 9, [0021], [0028]-[0029], [0034]-[0037], [0043].

201. Moreover, the ’448 Patent equates transmitting to a discrete IC component to generic outputting. *See* EX1001, 2:44-48 (“select logic then indicates which data word from the multiple data connections is to be output (e.g. transmitted to a discrete IC component”). As illustrated in Figure 9, excerpted above, the data word is output (i.e. “OUT), which, as explained by the ’448 patent, equates to transmission to a discrete IC component.

202. Accordingly, a POSITA would have understood and found obvious that the output of FIG. 9 [*e.g.*, ***the selected data word***] is transmitted to a discrete integrated circuit component on the CPU.

16. Claim 16

[16] The method of claim 15, wherein the select signals comprise a data connection selection signal indicative of which data word is to be selected and a data connection exclusion signal indicative of which data word is to be excluded.

203. Pixley-Watanabe renders Claim 16 obvious for at least the same reasons discussed above in Elements [5.d.i] and [5.d.ii]. *See* Element [5.d.i], [5.di.ii], *supra*.

17. Claim 17

[17.pre] A method for distributing multiplexing, comprising:

204. Pixley-Watanabe renders Element [17.pre] obvious for at least the same reasons discussed above in Element [15.pre]. *See* Element [15.pre], *supra*.

[17.a] receiving a first data[j] word corresponding to a first data connection;

205. Pixley-Watanabe renders Element [17.a] obvious.

206. As an initial matter, I note that though claim 17 uses the same terminology as earlier claims, the "first" data word and similar numbered terminology of claim 17 does not correspond to the "first" data word claimed in other claims—as is apparent from context of the claim. For example, in claim 1, a "second" stage logic selects between a "first" data word received from the "first" stage logic and another data word received from another of the plurality of data connections. In claim 17, a "first data word" is transmitted by "a" tristate driver to a "first logic," and is driven by a "first" tristate driver—i.e. not "a" tristate driver.

207. With respect to FIG. 9, Pixley describes that "[d]ata inputs and control/select signals to the MUXes 112, 114, and 116 are provided via four D flip-flops 124, 126, 130, and 132 . . . ." EX1004, 7:11-14.

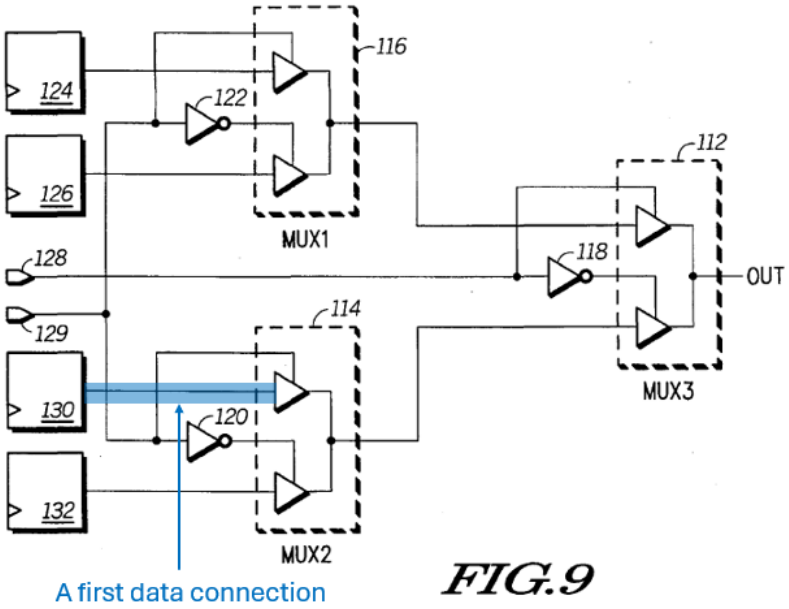


FIG.9

Annotated FIG. 9 of Pixley

208. As described with respect to Element [1.pre], *supra*, a POSITA would have understood or at least found obvious that, for certain known use cases, the 2-to-1 multiplexers described by Pixler would have been configured to multiplex m-bit data words. Specifically, it was known that "[a] 2-to-1 multiplexer of m-bit length is composed of m 2-to-1 multiplexers of one-bit length." *See, e.g.*, EX1005, [0018]. Further, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037]. Thus, a POSITA would have understood or at least found obvious that, in m-bit processing applications in which m-bit data must, for example, be read from registers or memory, each of Pixley's tri-state multiplexors 112, 114, and 116 would have been configured as an m-bit multiplexer composed of m one-bit multiplexers, one for each bit of data—effectively stacking a set of the multiplexer circuits illustrated in FIG. 9. In the context of Pixley's FIG. 9 circuit, the stack of one-bit multiplexer circuits would receive input from a corresponding set of flip-flops 132. *See* EX1004, 7:11-14. Indeed, a register may have been implemented with a set of flip-flops—one for each stored bit.

209. Accordingly, a POSITA would have understood and found obvious that Pixley's nested multiplexers of FIG. 9 included ***receives a first data word corresponding to a first data connection*** [*e.g.*, the connection between the set of flip-

flops 130 and tri-state MUX 114, where MUX 114 receives a first data word from flip-flops 130], as recited in [17.a].

[17.b] transmitting, via a tristate driver, the first data word to first logic associated with a second data connection;

210. In my opinion, Pixley-Watanabe renders obvious Element [17.b].

211. "FIG. 9 illustrates a circuit having tri-state MUXes 112, 114, and 116." EX1004, 7:8-9. Each of these MUXes include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors).

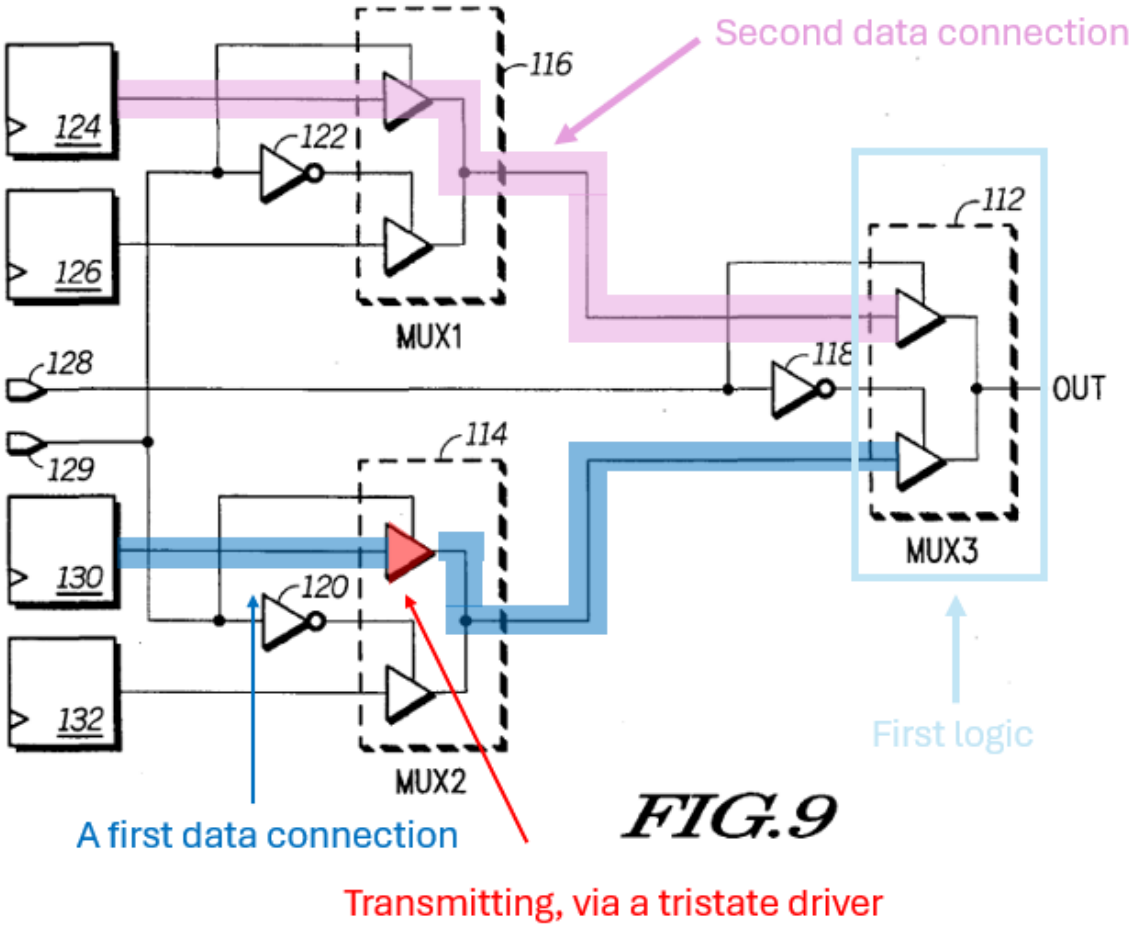


FIG. 9

Annotated FIG. 9 of Pixley

212. As described with respect to Element [1.pre], *supra*, a POSITA would have understood or at least found obvious that, for certain known use cases, the 2-to-1 multiplexers described by Pixler would have been configured to multiplex m-bit data words. And as described with respect to Element [17.a], MUX1 114 receives a first word via the first data connection. Next, upper tri-state driver of MUX2 114 (annotated in red above) transmits the first word. *See* 1004, 7:7-14,

FIG. 9. When MUX1 114 receives an enable signal via input signal 129, the upper tri-state driver is coupled to the output OUT of MUX2 114, and the first data word (highlighted in blue above) is sent [e.g., transmitted] to MUX3 112. *See id*. A POSITA would have recognized or found obvious that MUX3 112 is logic. For example, when the select signal input on terminal 128 is enabled (*i.e.*, a logic high), the upper tri-state driver of tri-state MUX 112 is enabled, and the first data word (highlighted in blue above) input to the upper tri-state driver is coupled to the output OUT of tri-state MUX 112. *See* EX1004, FIG. 9, 3:14-22, 7:7-14. And when the select signal output by the inverter 118 is enabled (*i.e.*, a logic high), the second/lower tri-state driver of tri-state MUX 112 is enabled, and the second data word (highlighted in pink above) input to the lower tri-state driver, via the second data connection is coupled to the output OUT of tri-state MUX 112. *See* EX1004, FIG. 9, 3:14-22, 7:7-14. Finally, a POSITA would have recognized or found obvious that the first logic (MUX3 112) is associated with the second data connection (highlighted in pink above), since the second data connection data to MUX3 112.

213. Accordingly, a POSITA would have understood and found obvious that Pixley's nested multiplexers of FIG. 9 ***transmit, via a tristate driver*** [*e.g.*, the upper tri-state driver of MUX1 114], ***the first data word*** [*e.g.*, the data word associated with the set of flip-flops 130 received by tri-state MUX 112 via tri-state MUX 114] ***to first logic associated with a second data connection*** [*e.g.*, tri-state

MUX 112, which is connected to a second data connection through its lower tri-state driver], as recited in [17.b].

[17.c] selecting a selected data word via the first logic between the first data word and a second data word transmitted from the second data connection based upon a set of select signals; and

214. In my opinion, Pixley-Watanabe renders obvious Element [17.c].

215. "FIG. 9 illustrates a circuit having tri-state MUXes 112, 114, and 116." EX1004, 7:8-9. Each of these MUXes include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "Each driver 30-32 is enabled only if the correspond-ing select line (sel-#) is enabled." EX1004, 3:18-19. "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.

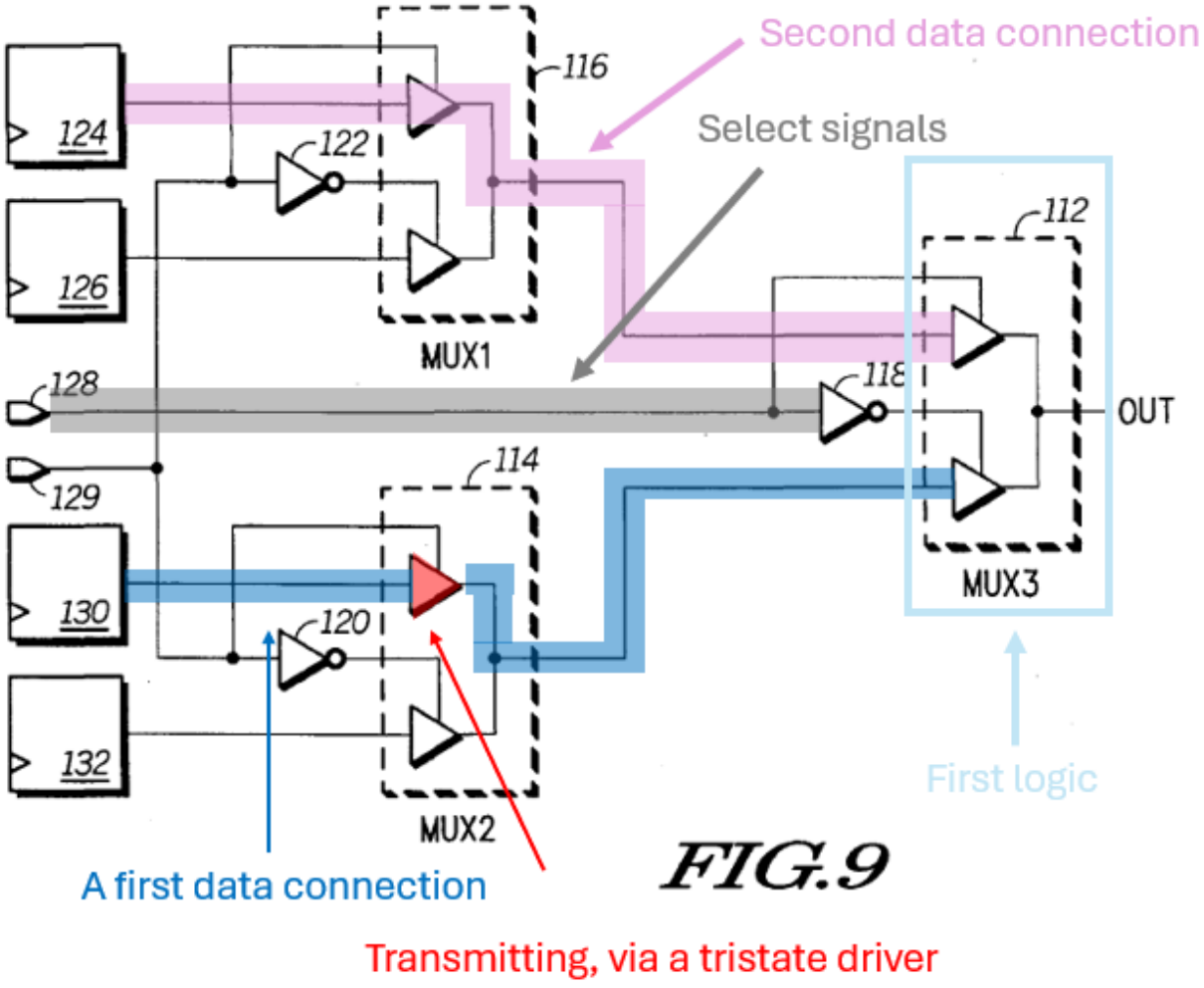


FIG. 9

Annotated FIG. 9 of Pixley

216. As described with respect to Element [1.pre], *supra*, a POSITA would have understood or at least found obvious that, for certain known use cases, the 2-to-1 multiplexers described by Pixler would have been configured to multiplex m-bit data words. And as described with respect to Element [17.a], MUX1 114 receives a first word via the first data connection from flip-flop 130.

217. In the example of FIG. 9, when the select signal input on terminal 128 is enabled (*i.e.*, a logic high), the upper tri-state driver of tri-state MUX 112 is enabled, and the second data word (highlighted in pink above) input to the upper tri-state driver is coupled to the output of tri-state MUX 112, or "***selected***." *See* EX1004, FIG. 9, 3:14-22. Correspondingly, the lower tri-state driver of tri-state MUX 112 receives a complement of the select signal input on terminal 128 via inverter 118, such that when the select signal input on terminal 128 is enabled, the output of inverter 118 is disabled (*i.e.*, a logic low) and the lower tri-state driver of tri-state MUX 112 is disabled. *See id.* On the other hand, when the select signal input on terminal 128 is disabled (*i.e.*, a logic low), the upper tri-state driver of tri-state MUX 112 is disabled. *See id.* And, correspondingly, the lower tri-state driver of tri-state MUX 112 receives a complement of the select signal input on terminal 128 via inverter 118, so when the select signal input on terminal 128 is disabled, the output of inverter 118 is enabled (*i.e.*, a logic high) and the lower tri-state driver of tri-state MUX 112 is enabled such that the first data word (highlighted in blue above) input to the lower tri-state driver is coupled to the output of tri-state MUX 112, or "*selected*." *See id.*

218. Accordingly, a POSITA would have understood and found obvious that Pixley's nested multiplexers of FIG. 9 selects a selected data word via the first

logic between the first data word and a second data word transmitted from the second data connection based upon a set of select signals [e.g., the tri-state MUX 112 selects between the first data word received from the set of flip-flops 130 and the second data word received from the set of flip-flops 124 based on the select signal of input terminal 128 and the select signal output by inverter 118], as recited in [17.c].

[17.d] transmitting the selected data word,

219. In my opinion, Pixley-Watanabe renders obvious Element [17.d].

220. As described with respect to [1.pre], supra, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See*, *e.g.*, EX1005, FIG. 3, [0035]-[0037]. One known use case of such a configuration was the output of an operand from a selected register in a register bank for transmission to and use in an arithmetic and logical unit (ALU). *See* EX1005, FIG. 4, [0043]. In this example use case, "[t]he operand output . . . that has been readout is interconnected to be sent, for example, to the arithmetic and logical unit (ALU) via a system bus . . . ." EX1005, [0043]; *see also* EX1006, FIG. 9, [0021], [0028]-[0029], [0034] (illustrating and describing "4-to-1 Multiplexer (MUX) formed with cascaded buses" and tri-state drivers).

221. Accordingly, a POSITA would have understood and found obvious that Pixley's nested multiplexers of FIG. 9 ***transmits the selected data word*** [*e.g.*, a

system bus connecting the output OUT of tri-state MUX 112 to an arithmetic and logical unit (ALU)], as recited in [17.d].

[17.e] wherein the selecting comprises: driving the second data word to a connection via a first tristate driver if the data connection selection signal is asserted; and disabling a second tristate driver configured to receive the first data word and configured to drive the first data word to the connection if the data connection exclusion signal is asserted.

222. In my opinion, Pixley-Watanabe renders obvious Element [17.e].

223. As described with respect to Element [1.b], *supra*, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multi-plexors). "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.

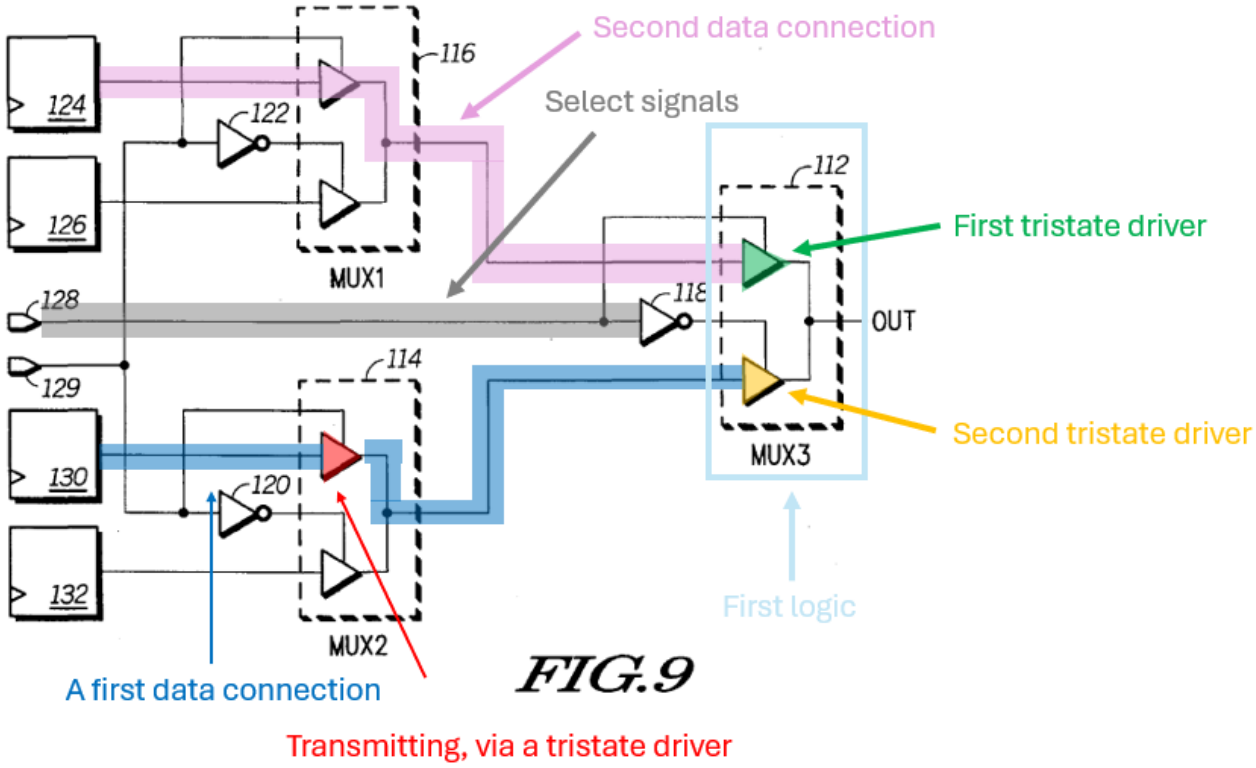


FIG. 9

Annotated FIG. 9 of Pixley

224. In the example of FIG. 9, when the select signal input on terminal 128 is enabled (*i.e.*, a logic high), the first/upper tri-state driver of tri-state MUX 112 is enabled, and the second data word (highlighted in pink above) input to the upper tri-state driver is coupled to the output OUT of tri-state MUX 112. *See* EX1004, FIG. 9, 3:14-22. Correspondingly, the second/lower tri-state driver of tri-state MUX 112 receives a complement of the select signal input on terminal 128 via inverter 118, such that when the select signal input on terminal 128 is enabled, the output of inverter 118 is disabled (*i.e.*, a logic low) and the second/lower tri-state driver of tri-state MUX 112 is disabled. *See id.* In this regard, the enablement of

select signal input on terminal 128 is a data connection selection signal with respect to second data word, at least because when select signal input on terminal 128 is enabled, the second data word is connected to the output OUT of the tri-state MUX 112 as a result of the first/upper tri-state driver being enabled. *See id.*

225. Accordingly, a POSITA would have understood and found obvious that Pixley's nested multiplexers of FIG. 9 **drives the second data word to a connection via a first tri-state driver if the data connection selection signal is asserted** [e.g., when the select signal input on terminal 128 is enabled/asserted, the first/upper tri-state driver couples its input with the second data word to the output OUT of the tri-state MUX 112], as recited in [17.e].

226. As described with respect to Element [1.b], *supra*, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. See EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multi-plexors). "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.

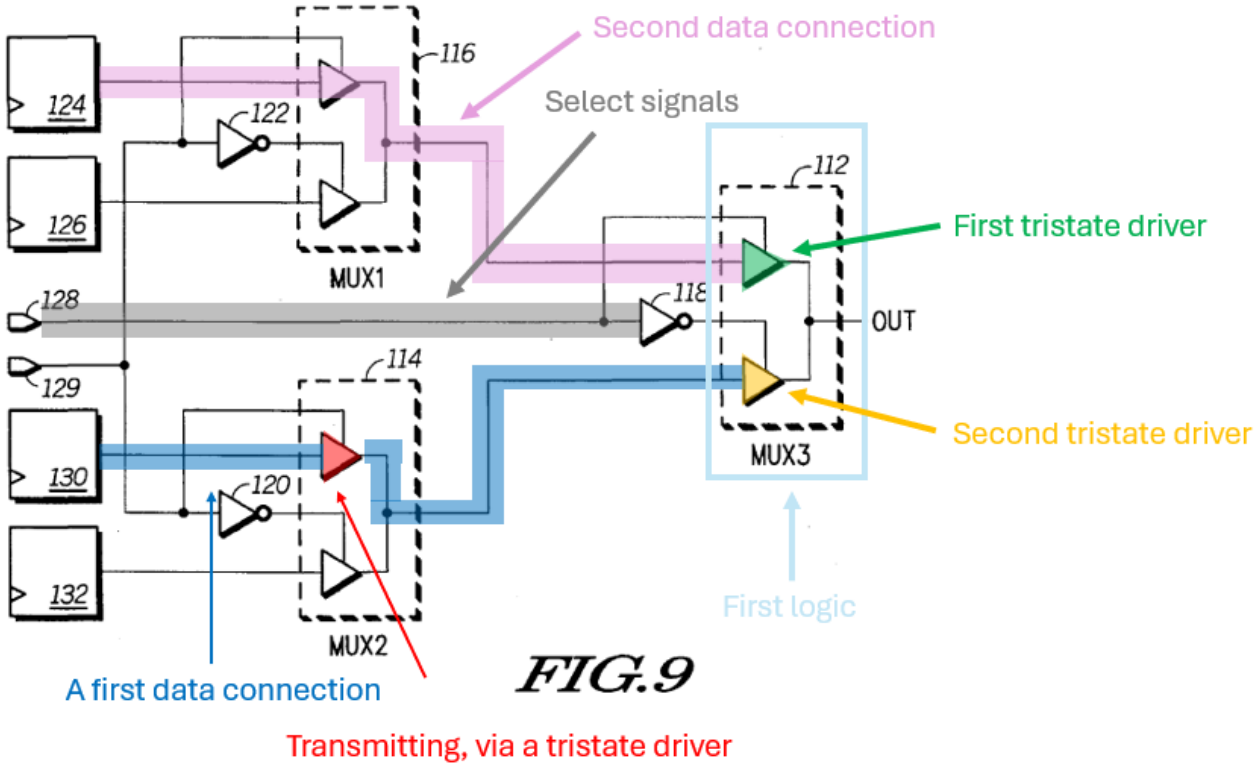


FIG. 9

Annotated FIG. 9 of Pixley

227. In the example of FIG. 9, when the select signal input on terminal 128 is enabled (*i.e.*, a logic high), the first/upper tri-state driver of tri-state MUX 112 is enabled, and the second data word (highlighted in pink above) input to the upper tri-state driver is coupled to the output OUT of tri-state MUX 112. *See* EX1004, FIG. 9, 3:14-22. Correspondingly, the second/lower tri-state driver of tri-state MUX 112 receives a complement of the select signal input on terminal 128 via inverter 118, such that when the select signal input on terminal 128 is enabled, the output of inverter 118 is disabled (*i.e.*, a logic low) and the second/lower tri-state driver of tri-state MUX 112 is disabled. *See id.* In this regard, the enablement of

select signal input on terminal 128 is a data connection exclusion signal with respect to first data word, at least because when select signal input on terminal 128 is enabled, the first data word is excluded from being connected to the output OUT of the tri-state MUX 112 as a result of the second/lower tri-state driver being disabled. *See id*. On the other hand, when the select signal input on terminal 128 is disabled (*i.e.*, a logic low), the first/upper tri-state driver of tri-state MUX 112 is disabled. *See id*.

228. Accordingly, a POSITA would have understood and found obvious that Pixley's nested multiplexers of FIG. 9 ***disables a second tristate driver configured to receive the first data word*** [*e.g.*, when the select signal input on terminal 128 is enabled/asserted, the second/lower tri-state driver is disabled]***, and configured to drive the first data word to the connection if the data connection exclusion signal is asserted*** [*e.g.*, when the select signal input on terminal 128 is disabled/deasserted, the second/lower tri-state driver couples its input with the first data word to the output OUT of the tri-state MUX 112], as recited in [17.d].

18. Claim 18

[18] The method of claim 17, wherein the selecting step further comprises: driving the first data word to the connection via the second tristate driver if the data connection exclusion signal is asserted; and disabling the first tristate driver if the data connection selection signal is deasserted.

229. In my opinion, Pixley-Watanabe renders obvious Element Claim 18.

230. As described with respect to Element [1.b], supra, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the de-scribed multi-plexors). "When the select line labeled sel-N (wherein N is an inte-ger between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.

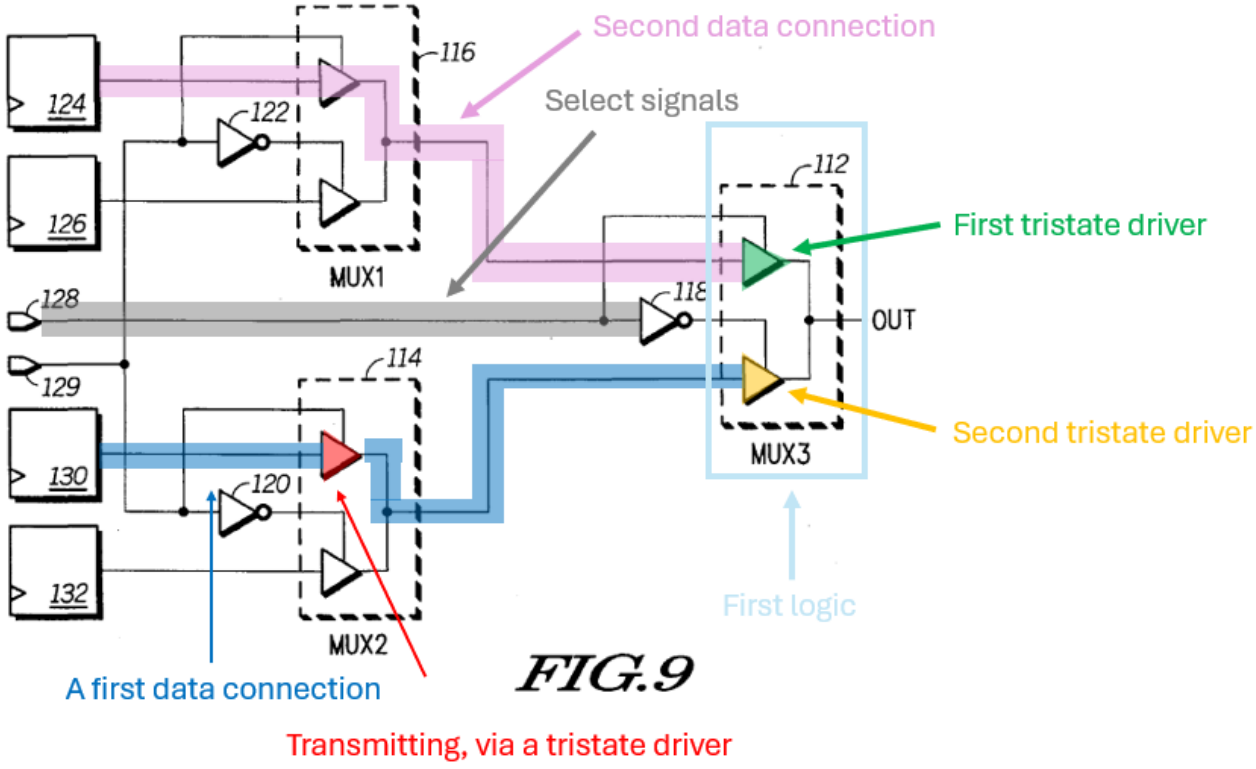


FIG.9

Annotated FIG. 9 of Pixley

231. In the example of FIG. 9, when the select signal input on terminal 128 is enabled (*i.e.*, a logic high), the first/upper tri-state driver of tri-state MUX 112 is enabled, and the second data word (highlighted in pink above) input to the upper

tri-state driver is coupled to the output OUT of tri-state MUX 112. *See* EX1004, FIG. 9, 3:14-22. Correspondingly, the second/lower tri-state driver of tri-state MUX 112 receives a complement of the select signal input on terminal 128 via inverter 118, such that when the select signal input on terminal 128 is enabled, the output of inverter 118 is disabled (*i.e.*, a logic low) and the second/lower tri-state driver of tri-state MUX 112 is disabled. *See id.* In this regard, the enablement of select signal input on terminal 128 is a data connection exclusion signal with respect to first data word, at least because when select signal input on terminal 128 is enabled, the first data word is excluded from being connected to the output OUT of the tri-state MUX 112 as a result of the second/lower tri-state driver being disabled. *See id*. On the other hand, when the select signal input on terminal 128 is disabled (*i.e.*, a logic low), the first/upper tri-state driver of tri-state MUX 112 is disabled. *See id*.

232. Accordingly, a POSITA would have understood and found obvious that Pixley's nested multiplexers of FIG. 9 drives the first data word to the connection via the second tristate driver if the data connection exclusion signal is asserted [e.g., when the select signal input on terminal 128 is disabled/deasserted, the second/lower tri-state driver couples its input with the first data word to the output OUT of the tri-state MUX 112], and disables the first tristate driver if the data connection selection signal is deasserted [e.g., when the select signal input on terminal

128 is disabled/deasserted, the first/upper tristate driver is disabled], as recited in [17.d].

19. Claim 19

[19] The method of claim 18, wherein the driving the first data word to the connection further comprises increasing signal strength of the first data word.

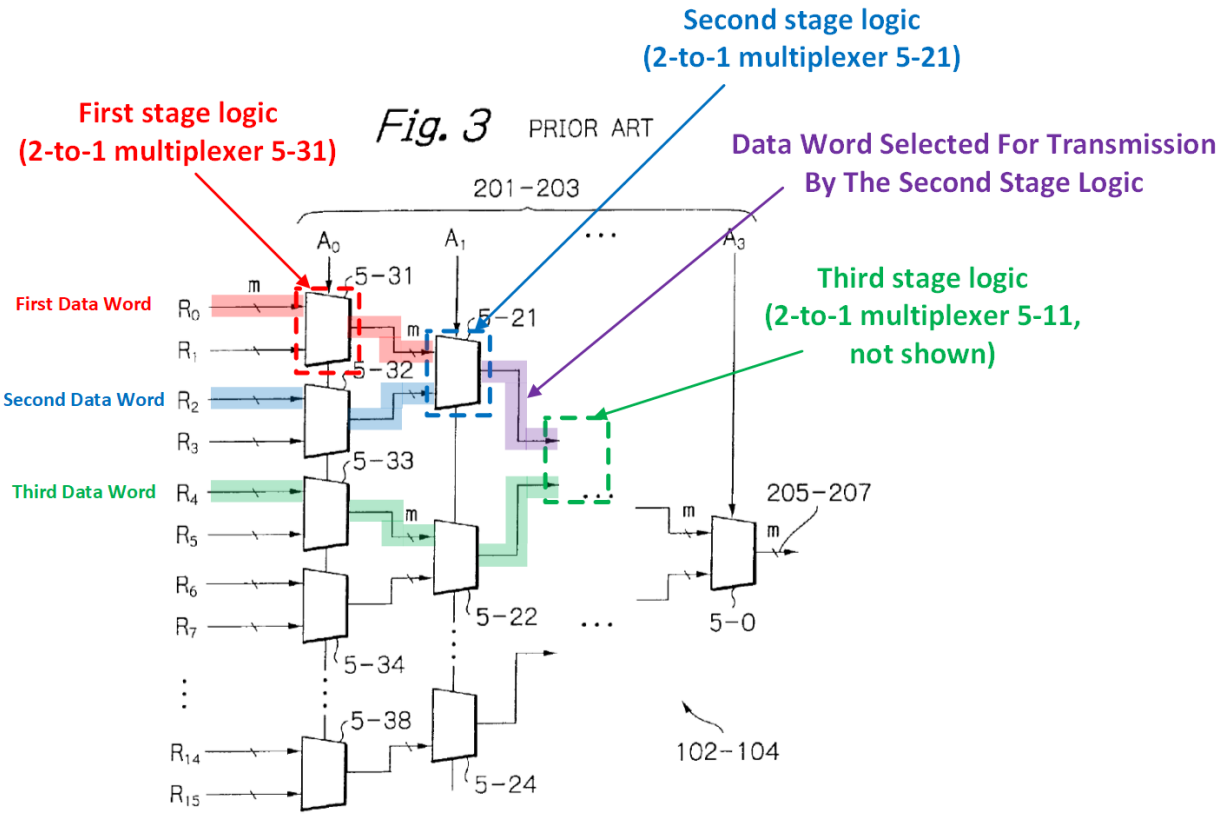
233. Pixley-Watanabe renders Claim 19 obvious for at least the same reasons discussed above in Claim 6. *See* Claim 6, *supra*.

20. Claim 20

[20.a] The circuit of claim 1, wherein the plurality of data words includes a third data word,

234. Pixley-Watanabe renders Element [20.a] obvious.

235. As described with respect to Element [1.pre] and Claim [3], *supra*, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037]. In one example of nested multiplexers described by Watanabe, the output of a second stage MUX (*e.g.*, 2-to-1 multiplexer 5-21) is transmitted to the input of a third stage MUX (*e.g.*, 2-to-1 multiplexer 5-11). *See* EX1005, FIG. 3, [0036]-[0037]. This configuration was useful when the nested/cascading multiplexers were required to select between more than four data inputs. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037].



Annotated FIG. 3 of Watanabe

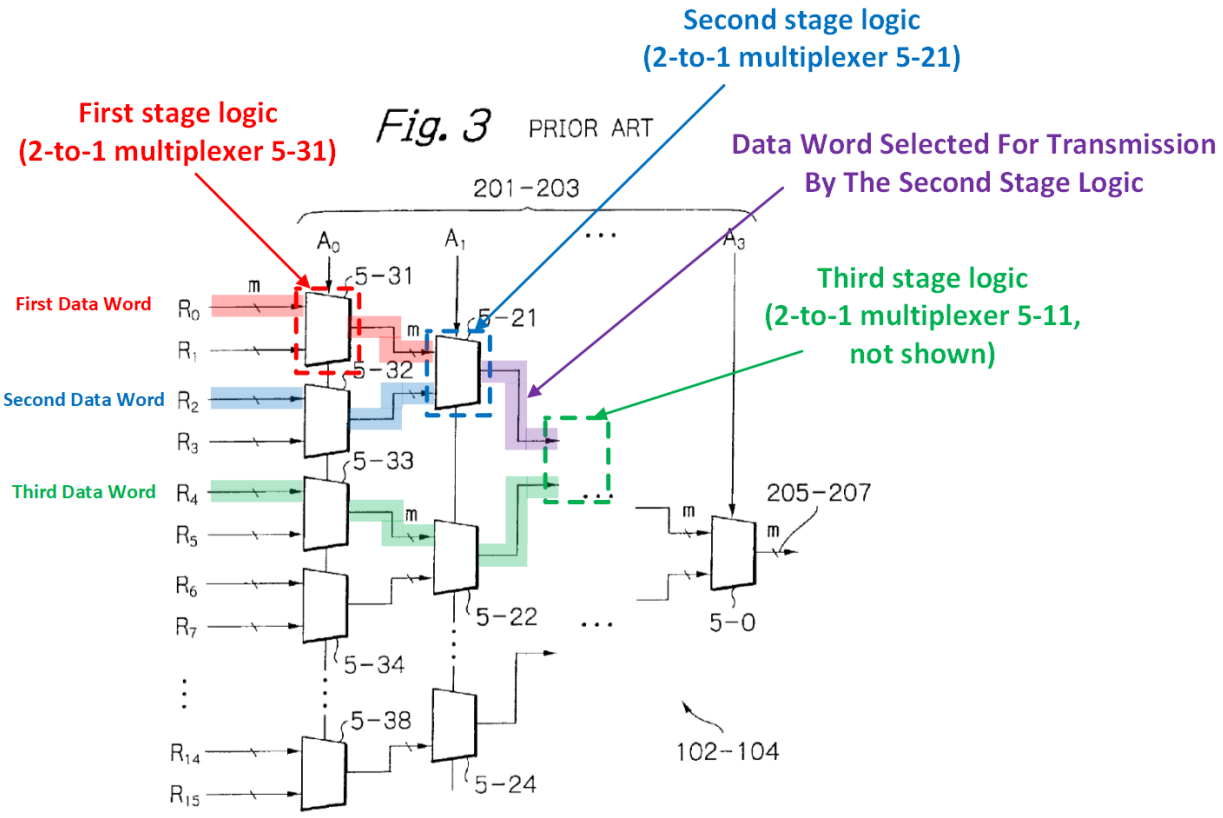
236. With respect to FIG. 9, Pixley describes that "[d]ata inputs and control/select signals to the MUXes 112, 114, and 116 are provided via four D flip-flops 124, 126, 130, and 132 . . . ." EX1004, 7:11-14.

237. Accordingly, a POSITA would have understood and found obvious that, in Pixely's nested multiplexers of FIG. 9, the plurality of data words includes a third data word [*e.g.*, the m-bit data word received from the m D flip-flops, highlighted in green above] as recited in Element [20.a].

[20.b] the circuit further comprising third stage logic configured to receive the third data word and the data word selected for transmission by the second stage logic,

238. Pixley-Watanabe renders Element [20.b] obvious.

239. As described with respect to Element [1.pre], *supra*, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037]. In one example of nested multiplexers described by Watanabe, the output of a second stage MUX *(e.g.*, 2-to-1 multiplexer 5-21) is transmitted to the input of a third stage MUX (*e.g.*, 2-to-1 multiplexer 5-11). *See* EX1005, FIG. 3, [0036]-[0037]. This configuration was useful when the nested/cascading multiplexers were required to select between more than four data inputs. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037].



Annotated FIG. 3 of Watanabe

240. As described with respect to Element [1.pre], *supra*, a POSITA would have understood or at least found obvious that, for certain known use cases, the 2-to-1 multiplexers described by Pixler would have been configured to multiplex m-bit data words. And as described with respect to Element [20.a], the nested multiplexor embodiment shown in FIG. 3 of Watanabe receives a third data word. A POSITA would have further recognized or found obvious that the third stage logic, named aptly because of the two prior stages of logic, receives the third data word (highlighted in green above) and the data word selected for transmission by the

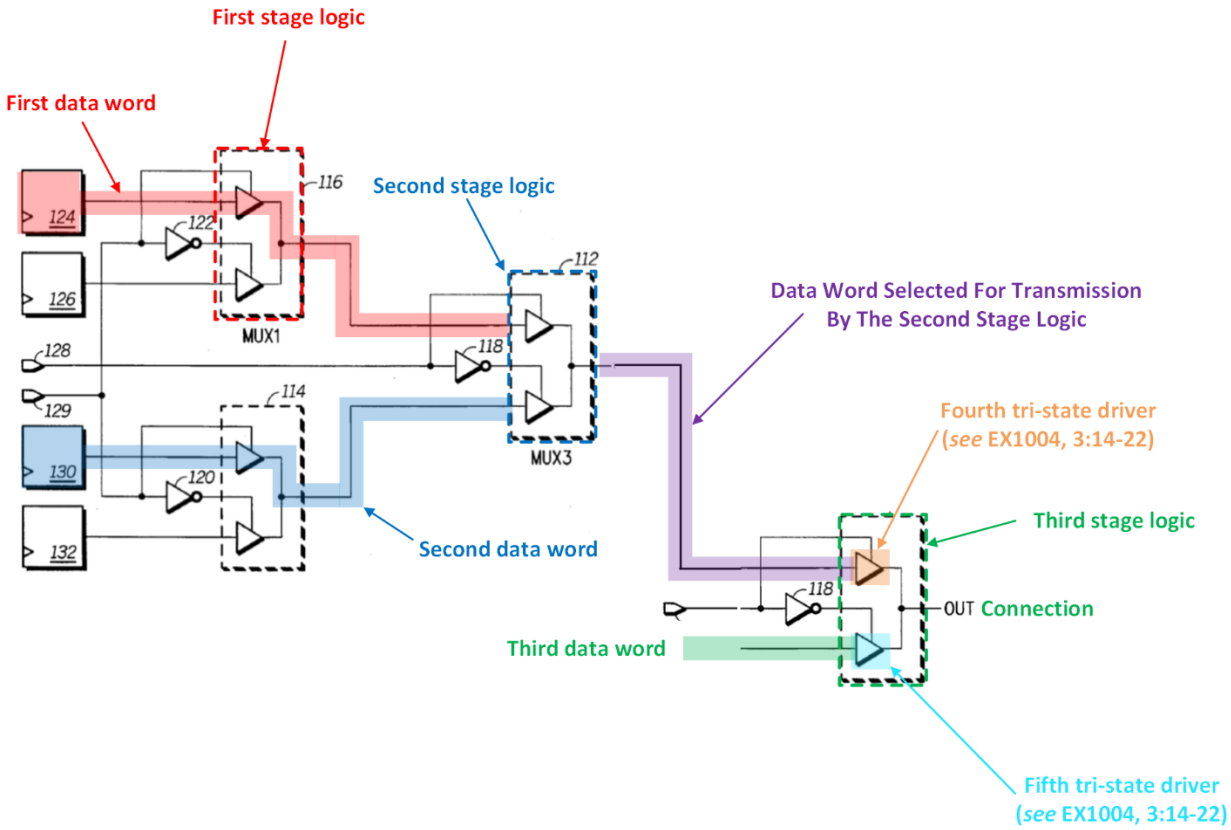
second stage logic (highlighted in purple above). A POSITA would have also recognized or found obvious that a MUX is logic. Using first stage logic 5-11 as an example, when the select signal input is enabled (*i.e.*, a logic high), the upper tri-state driver of tri-state MUX is enabled, and the data word selected for transmission by the second stage logic (highlighted in purple above) input to the upper tri-state driver is coupled to the output OUT of tri-state MUX. *See* EX1004, FIG. 9, 3:14-22, 7:7-14. And when the select signal output by the inverter is enabled (*i.e.*, a logic high), the second/lower tri-state driver of tri-state MUX is enabled, and the third data word (highlighted in green above) input to the lower tri-state driver is coupled to the output OUT of tri-state MUX. *See* EX1004, FIG. 9, 3:14-22, 7:7-14.

241. Accordingly, a POSITA would have understood and found obvious that Pixley's nested multiplexers of FIG. 9 further comprises ***third stage logic*** [*e.g.*, a subsequent stage tri-state multiplexer in a larger nested multiplexer circuit with more than four data inputs, shown in green above], ***configured to receive the third data word*** [*e.g.*, the output of tri-state multiplexer would have been connected to the input of the subsequent stage tri-state multiplexer]***and the data word selected for transmission by the second stage logic*** [*e.g.*, the output of tri-state multiplexer would have been connected to the input of the subsequent stage tri-state multiplexer], as recited in [20.b].

[20.c] the third stage logic configured to select for transmission one of the third data word and the data word selected for transmission by the second stage logic based on a set of select signal,

242. Pixley-Watanabe renders Element [20.c] obvious.

243. As described with respect to Element [1.pre], *supra*, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037]. In one example of nested multiplexers described by Watanabe, the output of a second stage MUX *(e.g.*, 2-to-1 multiplexer 5-21) is transmitted to the input of a third stage MUX (*e.g.*, 2-to-1 multiplexer 5-11). *See* EX1005, FIG. 3, [0036]-[0037]. This configuration was useful when the nested/cascading multiplexers were required to select between more than four data inputs. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037]. As shown below, a POSITA would have recognized that the third stage logic would have the same structure as the other MUXes in FIG. 9 of Pixley, including the tri-state driver structure and selection signals.



Modified FIG. 9 of Pixley

244. "FIG. 9 illustrates a circuit having tri-state MUXes 112, 114, and 116." EX1004, 7:8-9. Each of these MUXes include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "Each driver 30-32 is enabled only if the corresponding select line (sel-#) is enabled." EX1004, 3:18-19. "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22.

245. In the example of modified FIG. 9, a third logic (outlined by a green box) selects between the third data word and the data word selected for transmission by the second stage logic based on a select signal (received in the same manner as the other logics) and the output of inverter (inverted in the same manner as the other logics). *See id.* When the select signal input is enabled (*i.e.*, a logic high), the upper tri-state driver (orange) is enabled, and the data word selected for transmission by the second stage logic (highlighted in purple above) input to the upper tri-state driver is coupled to the output OUT. *See id.* And when the select signal output by the inverter is enabled (*i.e.*, a logic high), the second/lower tri-state driver (light blue) is enabled, and the third data word (highlighted in green above) input to the lower tri-state driver is coupled to the output OUT.

246. Accordingly, a POSITA would have understood and found obvious that Pixley's nested multiplexers of modified FIG. 9 included ***the third stage logic*** [*e.g.*, tri-state MUX outlined in green, which is connected to the third data word and the data word selected for transmission by the second stage logic] ***configured to select for transmission one of the third data word*** [*e.g.*, the data output by tri-state MUX] ***and the data word selected for transmission by the second stage logic*** [*e.g.*, the data word highlighted in green above] ***based on a set of select signal*** [*e.g.*, the select signal of input terminal and the select signal output by the inverter], as recited in [20.c].

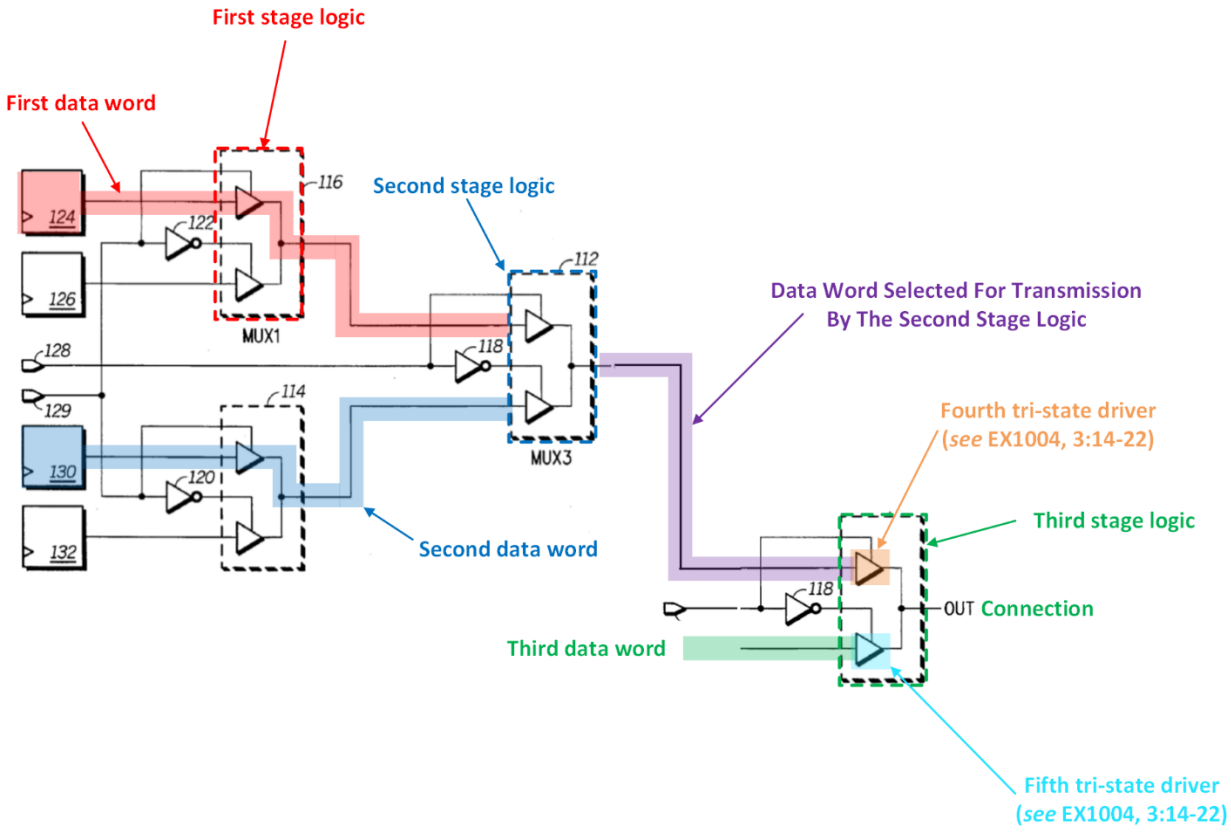
[20.d] the third stage logic having a connection, a fourth tristate driver, and a fifth tristate driver, the fourth tristate driver having an output electrically coupled to an output of the fifth tristate driver, the fourth tristate driver configured to receive the data word selected for transmission by the second stage logic and to drive the connection of the third stage logic with the data word selected for transmission by the second stage logic if the data word selected for transmission by the second stage logic is selected for transmission by the third stage logic, the fifth tristate driver configured to receive the third data word and to drive the connection of the third stage logic with the third data word if the third data word is selected for transmission by the third stage logic.

247. In my opinion, Pixley-Watanabe renders obvious Element [20.d.i].

248. As described with respect to Element [1.pre], *supra*, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037]. In one example of nested multiplexers described by Watanabe, the output of a second stage MUX *(e.g.*, 2-to-1 multiplexer 5-21) is transmitted to the input of a third stage MUX (*e.g.*, 2-to-1 multiplexer 5-11). *See* EX1005, FIG. 3, [0036]-[0037]. This configuration was useful when the nested/cascading multiplexers were required to select between more than four data inputs. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037].

249. As described with respect to Element [1.b], *supra*, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "When the select line labeled sel-N (wherein N is an integer

between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22. A POSITA would have recognized and found obvious that the third stage logic had the same structure as the prior stages.



Modified FIG. 9 of Pixley

250. In the example of modified FIG. 9, when the select signal input is enabled (*i.e.*, a logic high), the fourth/upper tri-state driver is enabled, and the data word selected for transmission by the second stage logic (highlighted in purple above) input to the upper tri-state driver is coupled to the output OUT. *See* EX1004, FIG. 9, 3:14-22. Correspondingly, the fifth/lower tri-state driver of tri-state MUX 110 receives a complement of the select signal input on a terminal via

an inverter—similar to what has been described for the other logics—such that when the select signal input is enabled, the output of the inverter is disabled (*i.e.*, a logic low) and the fifth/lower tri-state driver (light blue) is disabled. *See id.* On the other hand, when the select signal input is disabled (*i.e.*, a logic low), the fourth/upper tri-state driver (orange) is disabled. *See id.* And, correspondingly, the fifth/lower tri-state (light blue) receives a complement of the select signal input via an inverter, so when the select signal input is disabled, the output of inverter is enabled (*i.e.*, a logic high) and the fifth/lower tri-state driver (light blue) is enabled such that the third data word (highlighted in green above) input to the lower tri-state driver is coupled to the output OUT of tri-state MUX 110. *See id.*

251. Accordingly, a POSITA would have understood and found obvious that Pixely's tri-state MUX [the third stage logic] in modified FIG. 9 included ***a connection*** [*e.g.*, output OUT of tri-state MUX], ***a fourth tristate driver*** [*e.g.*, the upper tri-state driver, highlighted in orange above], ***and a fifth tristate driver*** [*e.g.*, the lower tri-state driver, highlighted in light blue above], ***the fourth tristate driver having an output electrically coupled to an output of the fifth tristate driver*** [*e.g.*, the output of the upper and lower tri-state drivers of tri-state MUX are coupled together at OUT], ***the fourth tristate driver configured to receive the data word selected for transmission by the second stage logic and*** [*e.g.*, the upper tri-state driver (orange) receives the data word selected from tri-state MUX 112] ***and to***

drive the connection of the third stage logic with the data word selected for transmission by the second stage logic if the data word selected for transmission by the second stage logic is selected for transmission by the third stage logic [*e.g.*, the data word received from the tri-state MUX 112 to the upper tri-state driver (orange) is coupled to OUT for transmission when the select signal input is enabled]**,** ***the fifth tristate driver configured to receive the third data word and to drive the connection of the second stage logic with the third data word if the second data word is selected for transmission by the third stage logic*** [*e.g.*, lower tri-state driver (light blue) is coupled to OUT for transmission when the select signal output is enabled, which is when the select signal input is disabled], as recited in [20.d].

21. Claim 21

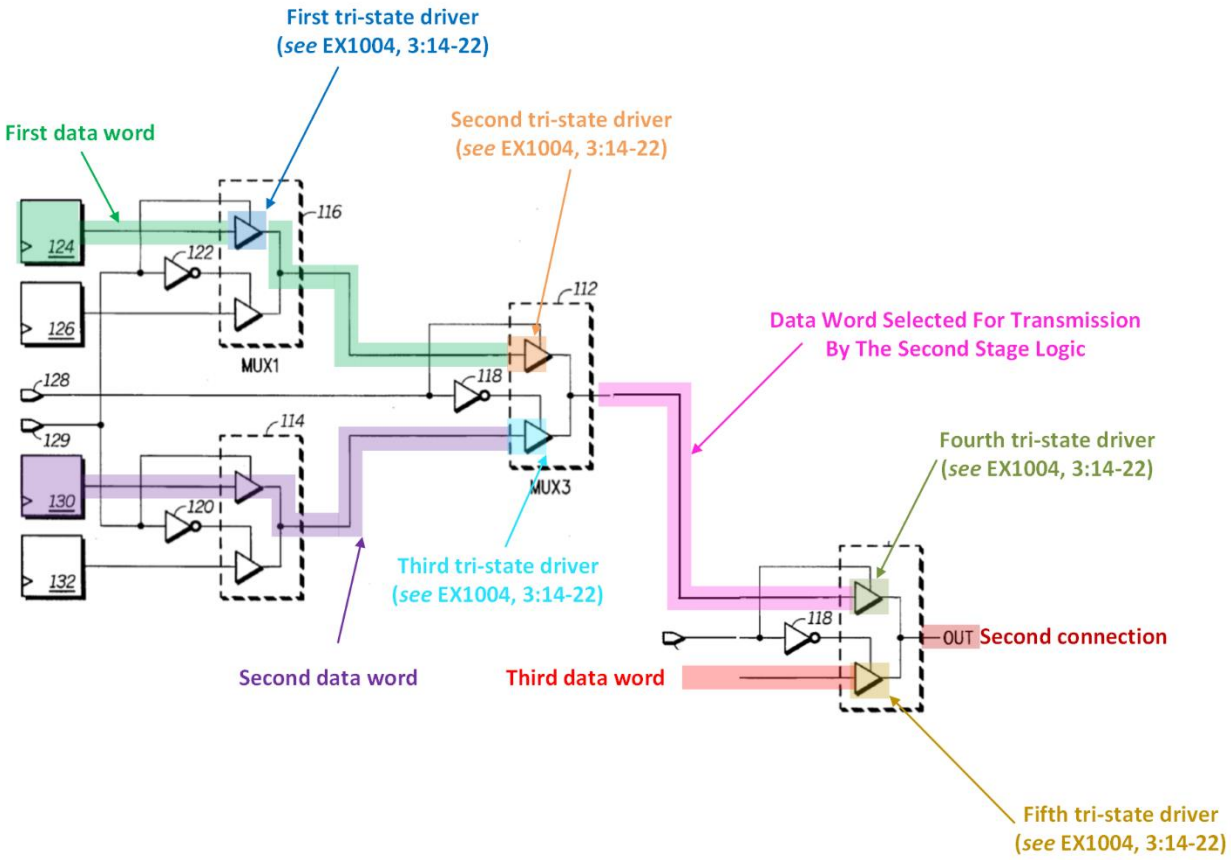
[21] The method of claim 14, further comprising: receiving at a fourth tristate driver the data word selected by the selecting; selecting for transmission between the received data word and a third data word based upon a set of select signals; driving a second connection with the received data word via the fourth tristate driver if the received data word is selected by the selecting for transmission between the received data word and the third data word; and driving the second connection with the third data word if the third data word is selected by the selecting for transmission between the received data word and the third data word.

252. In my opinion, Pixley-Watanabe renders obvious Claim [21].

253. As described with respect to Element [1.pre], *supra*, it was known that several m-bit 2-to-1 multiplexers could be nested, similar to the configuration shown in Pixley's FIG. 9. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037]. In one example of nested multiplexers described by Watanabe, the output of a second stage

MUX *(e.g.*, 2-to-1 multiplexer 5-21) is transmitted to the input of a third stage MUX (*e.g.*, 2-to-1 multiplexer 5-11). *See* EX1005, FIG. 3, [0036]-[0037]. This configuration was useful when the nested/cascading multiplexers were required to select between more than four data inputs. *See, e.g.*, EX1005, FIG. 3, [0035]-[0037].

254. As described with respect to Element [1.b], *supra*, each of the tri-state MUXs 112, 114, and 116 include two tri-state drivers. *See* EX1004, FIG. 9, 7:7-14, 3:14-22 (describing the operation of the tri-state drivers included in the described multiplexors). "When the select line labeled sel-N (wherein N is an integer between 0 and 2 for FIG. 3) is enabled then the [input] value is coupled to the output node OUT." EX1004, 3:19-22. A POSITA would have recognized and found obvious that the third stage logic had the same structure as the prior stages.



255. In the example of modified FIG. 9, the fourth/upper tri-state driver (olive green) receives a data word selected for transmission by the second stage logic. *See* Element [1.c.ii]. Additionally, the second stage logic selects which input to send based upon selection signal 128. *See* EX1004, FIG. 9, 3:14-22. However, when the select signal input is enabled (*i.e.*, a logic high), the fourth/upper tri-state driver (olive green) is enabled, and the data word received from the second stage logic (highlighted in pink above) input to the upper tri-state driver is coupled to the output OUT, or second connection, and subsequently transmitted. *See* EX1004, FIG. 9, 3:14-22. Correspondingly, the fifth/lower tri-state driver (gold)

receives a complement of the select signal input via an inverter, such that when the select signal input is enabled, the output of the inverter is disabled (*i.e.*, a logic low) and the fifth/lower tri-state driver (gold) is disabled. *See id.* On the other hand, when the select signal input is disabled (*i.e.*, a logic low), the fourth/upper tri-state driver (olive gren) is disabled. *See id.* And, correspondingly, the fifth/lower tri-state driver (gold) receives a complement of the select signal input via an inverter, so when the select signal input is disabled, the output of the inverter is enabled (*i.e.*, a logic high) and the fifth/lower tri-state driver (gold) is enabled such that the third data word (highlighted in red above) input to the lower tri-state driver is coupled to the output OUT and transmitted. *See id.*

256. Accordingly, a POSITA would have understood and found obvious Pixley's nested multiplexers of modified FIG. 9 ***receives at a fourth tristate driver the data word selected by the selecting*** [*e.g.*, the fourth/upper tri-state driver (olive green) receives a first or second data word from the second tri-state MUX 112 depending on selection signal 128], ***selects for transmission between the received data word and a third data word based upon a set of select signals;*** [*e.g.*, selects between the data word received from tri-state MUX 112 (shown in pink above) and the third data word (shown in red above) based on the selection signal], ***drives a second connection with the received data word via the fourth tristate driver if the***

received data word is selected by the selecting for transmission between the received data word and the third data word [*e.g.*, the data word received from the tri-state MUX 112 to the upper tri-state driver (olive green) is coupled to OUT (second connection) and transmitted when the select signal input is enabled], ***and drives the second connection with the third data word if the third data word is selected by the selecting for transmission between the received data word and the third data word.*** [*e.g.*, the third data word from the tri-state MUX 112 to the lower tri-state driver (gold) is coupled to OUT (second connection) and transmitted when the select signal output from the inverter is enabled, which is when the select signal input is disabled], as recited in [21].

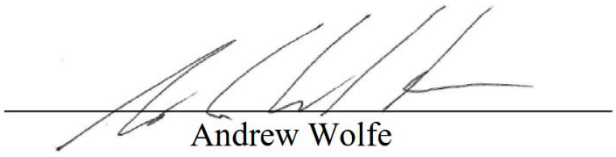
VIII. SECONDARY CONSIDERATIONS

257. I am not aware of any relevant information related to secondary considerations of obviousness. For example, I am not aware of any attempt to commercialize the claimed technology in a product or through licensing. To the extent Patent Owner comes forward with any such information, I will consider it.

* * *

I, Andrew Wolfe, do hereby declare and state 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 these statements were made with the knowledge that willful false statements and the like are punishable by fine or imprisonment under Section 1001 of Title 18 of the United States Code.

Executed on: August 29, 2025



Andrew Wolfe

Appendix A

Andrew Wolfe Ph.D.

2005 De La Cruz Blvd STE 142
Santa Clara, CA 95050
(408) 394-1096 (mobile)
Email: awolfe@awolfe.org

Education:

Ph.D. in Electrical and Computer Engineering, Carnegie Mellon University, 1992
Visiting Graduate Student, Center for Reliable Computing, Stanford University, 1988-1989
M.S. in Electrical and Computer Engineering, Carnegie Mellon University, 1987
B.S.E.E. in Electrical Engineering and Computer Science, The Johns Hopkins University, 1985

Recent Employment:

Lecturer, Asst. Teaching Professor [September 2013-present]
Santa Clara University

Teaching graduate and undergraduate courses on embedded computing, mechatronics, real-time systems, computer architecture, design, and community service. Faculty Senate President 2023-24.

Consultant, [October 2002-present]
Wolfe Consulting

Consultant on processor technology, computer systems, consumer electronics, software, design tools, and intellectual property issues. Testifying and consulting expert for IP and other technology-related litigation matters.

Sample clients include:

Google	Apple	Samsung
Sony	Huawei	HTC
AT&T	Verizon Mobile	T-Mobile
IBM	Motorola/Lenovo	Nintendo
Activision	AMD	Western Digital
Sonos	Qualcomm	LG
Netflix	Roku	Tivo
Synaptics	Sawstop	3M
Medigus	TCL	ZTE
Egnyte	Acer	Waymo

Chief Technical Officer, [1999-2002]; Sr. VP of Business Development, [2001-2002]; VP, Systems Integration, S3 Fellow , [1998 – 1999]; Director of Technology, S3 Fellow , [1997 - 1998]
SONIC|blue, Inc, Santa Clara, CA (formerly S3 Inc.)

Strategic Business Development:
Developed and implemented strategy to reposition S3 from PC graphics into the leading networked consumer electronics company.

- Acquired Diamond Multimedia and coordinated integration of communications, Rio digital music, and workstation graphics divisions into S3.
- Identified and negotiated acquisitions to grow digital media businesses including Empeg, ReplayTV, and Sensory Science.
- Identified and negotiated strategic investments including Comsilica, Intellon, KBGear Interactive, Entridia, DataPlay and others.

- Developed strategy for integrated graphics/core-logic products and established a joint venture with Via Technologies to design and market these products.
- Negotiated divestiture of graphics chip business to Via and the workstation graphics division to ATI.

Product Planning and Development:
- Drove roadmap development within SONICblue product divisions.
- Managed Business Development for all product lines.
- Led New Product Development and Corporate Vision processes.
- Acting co-General Manager of Rio digital music business in 2nd half of 2001. Responsible for all areas of product development, business development, and cost management.
- Managed development of the Savage/MX and Savage/IX mobile 3D graphics accelerators and Savage/NB system logic products.

Public Relations, Public Policy and Investor Relations:
- Present company products and strategy at industry events such as CES, Comdex, and Microprocessor Forum.
- Discuss new products and initiatives with the press.
- Promote issues of interest to SONICblue to industry groups and in Washington.
- Brief analysts, and investors on company progress. Participate in quarterly conference calls.

IP Management and Licensing:
- Negotiated and managed partnership agreements including a critical cross-licensing agreement with Intel.
- Renegotiated technology-licensing agreements with IBM for workstation graphics products.
- Evaluated outside technology opportunities, managed video research and development, and managed corporate IP strategy with legal staff including patent filings, cross licensing, and litigation.

Consulting Professor , [1999-2002]
Stanford University, Stanford, CA

Teaching computer architecture and microprocessor design.

Assistant Professor [1991 - 1997]
Princeton University, Princeton, NJ

Teaching and research in the Electrical Engineering department. Research in embedded computing systems, multimedia, video signal processors, compiler optimization, and high performance computer architecture. Principal investigator or project manager for ~$6M in funded research.

Visiting Assistant Professor , [1992]
Carnegie Mellon University, Pittsburgh, PA

Research and preparation of teaching materials on advanced microprocessor designs including new superscalar and superpipelined processor architectures.

Founder and Vice President and Consultant, [1989 - 1995]
The Graphics Technology Company, Inc., Austin, TX

Founded company to develop touch-sensitive components and systems for the first generation of PDA devices and interactive public systems. Obtained financing from Gunze Corp., Osaka, Japan. Company is now part of 3M.

Senior Electrical Engineer, [1989]
ESL - TRW, Advanced Technology Division, Sunnyvale, CA

Designed the architecture for an Intel i860-based multiple-processor digital signal processing system for advanced military applications. Designed several FPGA interface chips for VME-bus systems.

Design Consultant, [1986 -1987]
Carroll Touch Division, AMP Inc., Round Rock, TX

Developed several new technologies for touch-screen systems. Designed the first ASIC produced for AMP, a mixed-signal interface chip for controlling touch-screen sensors. Developed the system electronics, system firmware, and customer utility software for numerous products including those based on the new ASIC.

Senior Design Engineer, [1983 -1985]
Touch Technology Inc., Annapolis, MD

Advisory Boards:

Director, Turtle Beach Corporation (NASDAQ:HEAR) (formerly Parametric Sound Corporation), KBGear Interactive, Inc., Comsilica, Inc., Rioport.com, various S3 subsidiaries.

Technical Advisory Boards, Ageia, Inc., Intellon, Inc., Comsilica, Inc., Entridia, Inc., Siroyan, Ltd., BOPS, Inc, Quester Venture Funds

Carnegie Mellon University Silicon Valley Advisory Board; Johns Hopkins University Tech Transfer Advisory Board

Awards:

IEEE Fellow - for contributions in hardware code compression of embedded software, power consumption analysis, and optimization, 2022
IEEE Computer Society Distinguished Contributor - 2021
Micro Test-of-Time Award (in recognition of one of the ten most influential papers of the first 25 years of the symposium), 2014
Business 2.0 "20 Young Executives You Need to Know", 2002
Walter C. Johnson Prize for Teaching Excellence, 1997.
Princeton University Engineering Council Excellence in Teaching Award, Spring 1996
AT&T/Lucent Foundation Research Award, 1996.
Walter C. Johnson Prize for Teaching Excellence, 1995
IEEE Certificate of Appreciation, 1995, 2001.
AT&T Foundation Research Award, 1993.
Semiconductor Research Corporation Fellow, 1986 - 1991.
Burroughs Corporation Fellowship in Engineering, 1985 - 1986.

Professional Activities:

Program Chair: Micro-24, 1991, Hot Chips 13, 2001.

General Chair: Micro-26, 1993, Micro-33, 2000.

Associate Editor: IEEE Computer Architecture Letters; ACM Transactions in Embedded Computing Systems

Speaker at CES, WinHec, Comdex, Intel Dev. Forum, Digital Media Summit, Microprocessor Forum, etc.

Keynote speaker at Micro-34, ICME 2002

IEEE B. Ramakrishna Rau Award committee – 2012-2016

IEEE Computer Society Awards Committee – 2015

Director – IEEE Consultants Network of Silicon Valley – 2022-present

CES Awards Judge – 2016

Entrepreneurship Mentor – Draper University

Over 50 refereed publications.

Publications since January 2006:

Wolfe, A., "Retrospective on Code Compression and a Fresh Approach to Embedded Systems", IEEE MICRO, July/Aug. 2016, Invited paper.

Michael Kreienkamp, Olivia McConaghy, Sally L. Wood, Andrew Wolfe, Julia A. Scott, 2024 NYC Neuromodulation Conference, Transcranial Photobiomodulation Control System Compatible with EEG, abstract and poster. June 2024 https://neuromodec.org/nyc-neuromodulation-2024/poster-list.html#A83D

Patents:

U.S. Pat. 5,041,701 – *Edge Linearization Device for a Contact Input System*, Aug. 20, 1991.
U.S. Pat. 5,438,168 – *Touch Panel*, Aug. 1, 1995.
U.S. Pat. 5,736,688 – *Curvilinear Linearization Device for Touch Systems*, Apr. 7, 1998.
U.S. Pat. 6,037,930 – *Multimodal touch sensitive peripheral device*, March 14, 2000.
U.S. Pat. 6,408,421 – *High-speed asynchronous decoder circuit for variable-length coded data*, June 18, 2002.
U.S. Pat. 6,865,668 – *Variable-length, high-speed, asynchronous decoder circuit*, March 8, 2005
U.S. Pat. 7,079,133 – *Superscalar 3D Graphics Engine*, July 18, 2006
EP 1 661 131 B1 – *PORTABLE ENTERTAINMENT APPARATUS*, Jan. 21, 2009
U.S. Pat. 7,555,006 – *Method and system for adaptive transcoding and transrating in a video network*, June 30, 2009
U.S. Pat. 7,996,595 – *Interrupt Arbitration for Multiprocessors*, Aug. 9, 2011
EP 2 241 979 B1 – *Interrupt Arbitration for Multiprocessors*, Oct. 10, 2011
U.S. Pat. 8,131,970 – *Compiler Based Cache Allocation*, March 6, 2012
U.S. Pat. 8,180,963 – *Hierarchical read-combining local memories*, May 15, 2012
U.S. Pat. 8,193,941 – *Snoring Treatment*, June 5, 2012
U.S. Pat. 8,203,541 – *OLED display and sensor*, June 19, 2012
U.S. Pat. 8,243,045 – *Touch-sensitive display device and method*, August 14, 2012
U.S. Pat. 8,244,982 – *Allocating processor cores with cache memory associativity*, August 14, 2012
U.S. Pat. 8,260,996 – *Interrupt Optimization for Multiprocessors*, Sept. 4, 2012
101185761 (KR) – *Noise Cancellation for Phone Conversation,* Sept. 19, 2012
101200740 (KR) – *OLED display and sensor*, November 7, 2012
101200741 (KR) – T*ouch-sensitive display device and method*, November 7, 2012
U.S. Pat. 8,321,614 – *Dynamic scheduling interrupt controller for multiprocessors*, Nov. 27, 2012
U.S. Pat. 8,352,679 – *Selectively securing data and/or erasing secure data caches responsive to security compromising conditions*, Jan. 8, 2013
U.S. Pat. 8,355,541 – *Texture Sensing*, Jan. 15, 2013
U.S. Pat. 8,370,307 – *Cloud Data Backup Storage Manager*, Feb. 5, 2013
U.S. Pat. 8,398,451 – *Tactile Input Interaction*, March. 19, 2013
JP 5241032 B2 – *Routing Across Multicore Network Using Real World or Modeled Data*, April 13, 2013
ZL201010124820.3 – *Interrupt Optimization for Multiprocessors,* April 17, 2013
U.S. Pat. 8,428,438 – *Apparatus for Viewing Television with Pause Capability*, April 23, 2013
JP 5266197 B2 – *Data Centers Task Mapping,* May 10, 2013
U.S. Pat. 8,508,498 – *Direction and Force Sensing Input Device*, August 13, 2013
U.S. Pat. 8,547,457 – *Camera Flash Mitigation*, October 1, 2013
U.S. Pat. 8,549,339 – *Processor core communication in multi-core processor*, October 1, 2013
101319048 (KR) – *Camera Flash Mitigation*, October 10, 2013
U.S. Pat. 8,628,478 – *Microphone for remote health sensing*, January 14, 2014
101362017 (KR) – *Thread Shift: Allocating Threads to Cores*, Feb. 5, 2014
101361928 (KR) – *Cache Prefill on Thread Migration*, Feb. 5, 2014
101361945 (KR) – *Mapping Of Computer Threads onto Heterogeneous Resources*, Feb. 5, 2014
JP 5487307 B2 – *Mapping Of Computer Threads onto Heterogeneous Resources*, Feb. 28, 2014
JP 5484580 B2 – *Task Scheduling Based on Financial Impact,* Feb. 28, 2014
JP 5487306 B2 – *Cache Prefill on Thread Migration*, Feb. 28, 2014
101372623 (KR) – *Power Management for Processor*, March. 4, 2014
101373925 (KR) – *Allocating Processor Cores with Cache Memory Associativity*, March 6, 2014
U.S. Pat. 8,676,668 – *Method for the determination of a time, location, and quantity of goods to be made available based on mapped population activity*, March 18, 2014
U.S. Pat. 8,687,533 – *Energy Reservation in Power Limited Networks*, April 1, 2014
101388735 (KR) – *Routing Across Multicore Networks Using Real World or Modeled Data*, April 17, 2014
U.S. Pat. 8,725,697 – *Cloud Data Backup Storage*, May 13, 2014
U.S. Pat. 8,726,043 – *Securing Backing Storage Data Passed Through a Network*, May 13, 2014
ZL201010124826.0 – *Dynamic scheduling interrupt controller for multiprocessors*, May 14, 2014
JP 5547820 B2 – *Processor core communication in multi-core processor*, May 23, 2014
U.S. Pat. 8,738,949 – *Power Management for Processor*, May 27, 2014

U.S. Pat. 8,751,854 – *Processor Core Clock Rate Selection*, June 10, 2014
JP 5559891 B2 – *Thermal Management in Multi-Core Processor*, June 13, 2014
101414033 (KR) – *Dynamic Computation Allocation*, June 25, 2014
JP 5571184 B2 – *Dynamic Computation Allocation*, July 4, 2014
101426341 (KR) – *Processor core communication in multi-core processor*, May 23, 2014
U.S. Pat. 8,799,671 – *Techniques for Detecting Encrypted Data*, Aug 5, 2014
101433485 (KR) – *Task Scheduling Based on Financial Impact,* Aug. 18, 2014
U.S. Pat. 8,824,666 – *Noise Cancellation for Phone Conversation,* Sept. 2, 2014
U.S. Pat. 8,836,516 – *Snoring Treatment*, Sept. 16, 2014
U.S. Pat. 8,838,370 – *Traffic flow model to provide traffic flow information*, Sept. 16, 2014
U.S. Pat. 8,838,797 – *Dynamic Computation Allocation*, Sept. 16, 2014
U.S. Pat. 8,854,379 – *Routing Across Multicore Networks Using Real World or Modeled Data*, Oct. 7, 2014
JP 5615361 B2 – *Thread Shift: Allocating Threads to Cores*, Oct. 15, 2014
U.S. Pat. 8,866,621 – *Sudden infant death prevention clothing*, Oct. 21, 2014
U.S. Pat. 8,881,157 – *Allocating threads to cores based on threads falling behind threads*, Nov. 4, 2014
ZL201080024755.5 – *Camera Flash Mitigation*, Nov 5, 2014
U.S. Pat. 8,882,677 – *Microphone for remote health sensing*, Nov. 11, 2014
U.S. Pat. 8,924,743 – *Securing Data Cache through Encryption*, December 30, 2014
U.S. Pat. 8,994,857 – *Camera Flash Mitigation*, March 31, 2015
JP 5699140 B2 – *Camera Flash Mitigation*, April 8, 2015
ZL201080035189.8 – *Thread Shift: Allocating Threads to Cores*, June 10, 2015
ZL201180005030.6 – *Processor core communication in multi-core processor*, June 10, 2015
U.S. Pat. 9,143,814 – *Method and system for adaptive transcoding and transrating in a video network*, Sept 22, 2015
ZL201080035177.5 – *Mapping Of Computer Threads onto Heterogeneous Resources*, Oct. 14, 2015
U.S. Pat. 9,178,694 – *Securing Backing Storage Data Passed Through a Network*, November 3, 2015
U.S. Pat. 9,189,282 – *Thread-to-core mapping based on thread deadline, thread demand, and hardware characteristics data collected by a performance counter*, November 17, 2015
U.S. Pat. 9,189,448 – *Routing image data across on-chip networks*, November 17, 2015
U.S. Pat. 9,208,093 – *Allocation of memory space to individual processor cores*, December 8, 2015
U.S. Pat. 9,239,994 – *Data Centers Task Mapping,* January 19, 2016
ZL201080036611.1 – *Allocating Processor Cores with Cache Memory Associativity*, January 20, 2016
EP2228779 B1 – *Traffic flow model to provide traffic flow information*, Jan. 27, 2016
U.S. Pat. 9,262,628 – *Operating System Sandbox,* February 16, 2016
GB2485682 – *Mapping Of Computer Threads onto Heterogeneous Resources*, Sept. 28, 2016
U.S. Pat. 9,330,137 – *Cloud Data Backup Storage Manager*, May. 3, 2016
ZL201080035185.X – *Cache Prefill on Thread Migration*, Aug. 24, 2016
U.S. Pat. 9,519,305 – *Processor Core Clock Rate Selection*, December 13, 2016
U.S. Pat. 9,569,270 – *Mapping thread phases onto heterogeneous cores based on execution characteristics and cache line eviction count*, February 14, 2017
GB2485683 – *Thread Shift: Allocating Threads to Cores*, Oct. 18, 2017
U.S. Pat. 9,852,435 – *Telemetrics based location and tracking*, December 26, 2017.
U.S. Pat. 9,915,994 – *Power management for processor*, March 13, 2018
U.S. Pat. 9,927,254 – *Traffic flow model to provide traffic flow information*, March 27, 2018
EP2254048 B1 – *Thread Mapping in Multi-Core Processors*, August 29, 2018
U.S. Pat. 10,860,432 – *Cloud Data Backup Storage Manager*, December 8, 2020

Expert testimony by deposition or at trial – January 1, 2019 - -present

Case	Venue	Case Number
HTC Corporation v. Telefonaktiebolaget LM Ericsson	E.D. Texas	6: 18-cv-00243-JRG
Seven Networks, LLC v ZTE (USA) Inc and ZTE Corporation	N. D. Texas - Dallas	3:17-CV-1495
AGIS Software Development, LLC v. HTC Corporation	E. D. Texas	2:17-cv-514
Barbaro Technologies, LLC v. Niantic, Inc	N.D. CA	3:18-cv-02955-RS
Immersion Corp. v. Samsung Electronics America, Inc. et al	E.D. Texas	2:17-cv-00572
INTER PARTES REVIEW OF U.S. PATENT NO. 7,171,526 (Northstar Innovations – Micron)	PTAB	IPR2018-01004 IPR2018-01005
CISCO SYSTEMS, INC., vs. UNILOC USA, INC., UNILOC 2017 LLC and UNILOC LICENSING USA LLC	N.D. CA	3:18-cv-04991-SI
INTER PARTES REVIEW OF U.S. PATENT NO. 8,020,014 (Intel/VLSI)	PTAB	IPR2018-01661 IPR2018-01312
INTER PARTES REVIEW OF U.S. PATENT NO Patent 9,294,799 (Comcast/Rovi)	PTAB	IPR2019-00299
Inter Partes Review of U.S. Patent No 7720929 (Unified Patents v Datascape)	PTAB	IPR2019-01115
Solas OLED Ltd,, v. Samsung Display Co,, Ltd,, Samsung Electronics Co, Ltd,, and Samsung Electronics America, Inc,,	PTAB	IPR2019-01668
INTER PARTES REVIEW OF U.S. PATENT NO Patent 8,973,069 (Comcast/Rovi)	PTAB	IPR2019-01434
U.S. Patent No. 8,448,215 IPR (Comcast v Rovi)	PTAB	IPR2019-01353
U.S. PATENT NO. 8,847,898 IPR filing (Samsung v Neodron)	PTAB	IPR2020-00234
U.S. PATENT NO. 8,610,009 IPR filing (Samsung v Neodron)	PTAB	IPR2020-00225
U.S. PATENT NO. 10,365,747 IPR filing (Samsung v Neodron)	PTAB	IPR2020-00308
AGIS Software Development LLC v. Google LLC, AGIS Software Development LLC v. WAZE Mobile Limited,and AGIS Software Development LLC v. Samsung Elecs. Co., Ltd. et al	E.D. Texas	2:19-cv-00361-JRG 2:19-cv-00359-JRG 2:19-cv-00362-JRG
Inter Partes Review of U.S. Patent No 8,112,670 (Sony)	PTAB	IPR2020-00726
U.S. Patent 8,819,505 IPR (Intel v PACT)	PTAB	IPR2020-00525
Inter Partes Review of U.S. Patent No 8,078,540 (Sony)	PTAB	IPR2020-00922
Inter Partes Review of U.S. Patent No 9,037,807 (Intel v PACT)	PTAB	IPR2020-00540

HONG KONG UCLOUDLINK NETWORK TECHNOLOGY LIMITED AND UCLOUDLINK (AMERICA), LTD., vs. SIMO HOLDINGS INC. AND SKYROAM, INC.,	N.D. CA	3:18-cv-05031-EMC
Joe Andrew Salazar v AT&T Mobility, LLC et al.	E.D. Texas	2:20-cv-00004-JRG
Innovative Memory Solutions Inc. v Micron, Inc	Delaware	14-1480 RGA
Inter Partes Review of U.S. Patent No. 6,411,941	PTAB	IPR2021-01338 IPR2021-01406
Maxell, Ltd. et al v. VIZIO, Inc	C.D. CA	2-21-cv-06758
Inter Partes Review of U.S. Patent No. 7,619,912	PTAB	IPR2022-00615
Inter Partes Review of U.S. Patent Nos. 11,016,918 and 11,232,054	PTAB	IPR2022-00996 IPR2022-00999
In the Matter of Certain Smart Televisions, - Maxell, Ltd. et al v. VIZIO, Inc	ITC	337-TA-1338
INTER PARTES REVIEW OF U.S. PATENT NOS. 8,787,060 and 9,318,160	PTAB	IPR2022-01427 IPR2022-01428
Sweat Equity Partners, LLC vs. Swoop Search, LLC, et al.,	San Francisco Superior Court	CGC-21-591702
Walter Kidde Portable Equipment, Inc. v. First Alert, Inc. and BRK Brands	W.D. Texas	6:22-cv-566-ADA
INTER PARTES REVIEW OF U.S. PATENT NO. 11,093,417	PTAB	IPR2023-00454
INTER PARTES REVIEW OF U.S. PATENT NO. 9,858,215	PTAB	IPR2023-00455
INTER PARTES REVIEW OF U.S. PATENTS NOS. 9,369,081 and 9,941,830	PTAB	IPR2023-00992 IPR2023-00993
FlatFrog Laboratories AB v. Chemtronics Co., Ltd.	E.D. Texas	2:23-cv-00306
INTER PARTES REVIEW OF U.S. PATENT NOS. 8,624,844	PTAB	IPR2024-00404
Topia Technology, Inc. v. Egnyte, Inc.	District of Delaware	1-21-cv-01821-CJB
IPR of U.S. Patent No. 11,540,216	PTAB	IPR2024-00857
IPR of U.S. Patent No. 9,330,593	PTAB	IPR2024-00620
Cutting Edge Vision LLC v. T-Mobile US Inc and T-Mobile USA	W.D. Tx	6:24-cv-270
Inter Partes review of U.S. Patent No. 10,690,336	PTAB	IPR2024-01230
Universal Connectivity Technologies Inc. v. HP Inc.	ND CA	4:24-04097
Mullen Industries LLC v. Samsung Electronics Co., Ltd. et al,	E.D. Texas	2:24-cv-00049-JRG