

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: HakJune Oh et al. Attorney Docket No.: 60246-0003IP1
U.S. Patent No.: 9,972,381
Issue Date: May 15, 2018
Appl. Serial No.: 15/868,219
Filing Date: January 11, 2018
Title: MEMORY WITH OUTPUT CONTROL

DECLARATION OF DR. CARL SECHEN

I declare that all statements made herein on my own knowledge are true and that all statements made on information and belief are believed to be true, and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable under Section 1001 of Title 18 of the United States Code.

Date: 29 August 2025

By: Carl Sechen
Carl Sechen, Ph.D.

Table of Contents

I.	QUALIFICATIONS AND BACKGROUND INFORMATION.....	4
II.	LEGAL PRINCIPLES.....	9
	A. Anticipation.....	10
	B. Obviousness	10
III.	OVERVIEW OF CONCLUSIONS FORMED	13
IV.	BACKGROUND KNOWLEDGE ONE OF SKILL IN THE ART WOULD HAVE HAD PRIOR TO THE PRIORITY DATE OF THE '381 PATENT	14
V.	INTERPRETATIONS OF THE '381 PATENT CLAIMS AT ISSUE.....	16
VI.	THE '381 PATENT.....	16
	A. Overview of the '381 Patent.....	16
	B. File History	22
VII.	OVERVIEW AND COMBINATIONS OF PRIOR ART REFERENCES ..	23
	A. Overview of Grundy	24
	B. Overview of Kilbuck	29
	C. Combination of Grundy and Kilbuck	34
VIII.	MANNER IN WHICH THE PRIOR ART REFERENCES GRUNDY AND KILBUCK (GROUND 1) RENDER THE '381 CLAIMS UNPATENTABLE.....	39
	A. Claim 1.....	39
	B. Claim 2.....	72
	C. Claim 3.....	77
	D. Claim 4.....	79
	E. Claim 5.....	83
	F. Claim 6.....	84
	G. Claim 7.....	87
	H. Claim 8.....	88
	I. Claim 9.....	90
	J. Claim 10.....	91
	K. Claim 11.....	92
	L. Claim 12.....	95
	M. Claim 13.....	97
	N. Claim 14.....	103
	O. Claim 15.....	105
	P. Claim 16.....	108
	Q. Claim 17.....	111
	R. Claim 18.....	119

S. Claim 19.....	121
T. Claim 20.....	122
U. Claim 21.....	122
IX. CONCLUSION.....	123

DECLARATION OF DR. CARL SECHEN

I, Carl Sechen, of Dallas, TX, declare that:

I. QUALIFICATIONS AND BACKGROUND INFORMATION

1. My name is Carl Sechen and I have been retained as an expert witness by Petitioner Infineon Technologies Americas Corp. (“Infineon”) to provide assistance regarding U.S. Patent No. 9,972,381 (“the ’381 Patent”). Specifically, I have been asked to consider the validity of claims 1-21 of the ’381 Patent (the “Challenged Claims”).

2. I have been retained by Petitioners, as an independent expert in this proceeding before the United States Patent and Trademark Office. Although I am being compensated at my usual and customary rate, no part of my compensation depends on the outcome of this proceeding, and I have no other interest in this proceeding.

3. I understand that this proceeding involves the ’381 Patent, and I have been asked to consider the validity of certain claims of the ’381 Patent based on certain prior art references. I have also been asked to consider the state of the art and prior art available as of September 30, 2005. Based on the prior art discussed in this declaration, it is my opinion that claims 1-21 of the ’381 Patent are unpatentable for the reasons provided below.

4. I believe that I am well qualified to serve as a technical expert in this matter based upon my educational and work experience, and specifically, in the field of integrated circuits (ICs), clock signaling, and memory devices. I understand that my curriculum vitae (“CV”) is attached as Exhibit INFINEON-1004 of the IPR petition.

5. Of note, as shown in my CV, I have extensive experience in the design and implementation of integrated circuits involving clock signals and logic elements.

6. I received a B.S. degree in Electrical Engineering in 1975, an M.S. degree in Electrical Engineering in 1977, and a Ph.D. degree in Electrical Engineering in 1986 from the University of California, Berkeley. I received several academic honors, including my B.S. degree with honors, a best graduate student award and a best teaching assistant award during my M.S. study, and a best paper award from a major international conference for reporting results from my Ph.D. thesis.

7. I have been a professor of Electrical Engineering for 39 years. After receiving my Ph.D., I joined the faculty at Yale University as an Assistant Professor of Electrical Engineering in July 1986. In 1990, I was promoted to an Associate Professor of Electrical Engineering, a position I held until June 1992. From July 1992 to August 2005, I was first an Associate Professor and then a Professor of Electrical Engineering at the University of Washington. Since August 2005 to the

present, I have been a Professor of Electrical and Computer Engineering at the University of Texas at Dallas. Over these years, my research has focused on the design and computer-aided design of digital and analog integrated circuits. I have taught numerous students how to design digital, analog, and mixed signal integrated circuits and printed circuit boards, as part of the design courses that I have offered.

8. As a professor, I have developed and taught numerous courses, For example, several courses that teach VLSI integrated circuit design in great detail. I have taught these courses continuously for the past 30 years. I have taught Verilog design to undergraduate and graduate students at the University of Washington and at the University of Texas at Dallas since 2001.

9. I have also been involved in numerous research projects on digital and analog VLSI design. I have taught numerous graduate researchers how to design digital and analog integrated circuits. My research has been funded by many companies over the years, including Intel, AMD, Cadence, LSI Logic, and Texas Instruments. For example, Texas Instruments has funded my work on memory cell research and VLSI design. My VLSI and memory design research has also been supported by governmental agencies such as DARPA and the National Science Foundation. I have authored or coauthored over 200 papers, one book, and four patents, the majority of which concern digital and analog integrated circuit design,

as well as computer-aided design. I have also designed, fabricated and tested numerous integrated circuits.

10. I was elected a Fellow of the IEEE in 2002 for contributions to placement and routing of ASICs. Since then I have been promoted to Life Fellow. IEEE stands for Institute of Electrical and Electronics Engineers, which is the leading professional association for electrical engineers. The Board of Directors of the IEEE awards the rank of “Fellow” to individuals with an extraordinary record of accomplishments in any of the IEEE fields of interest. The total number of IEEE members who can be named Fellows in any one year cannot exceed one-tenth of one percent of the total voting IEEE membership.

11. I also received a Distinguished Teaching Award from the University of Texas at Dallas in 2008. I also received a second Distinguished Teaching Award from the University of Texas at Dallas in 2014. In addition, I received the *Graduate Teaching Excellence Award* from the ECE Department at UT Dallas, May 2024. In 2002, I also received the Outstanding Research Advisor Award from the Department of Electrical engineering at the University of Washington and the Best Project Award from the NSF Center for the Design of Digital and Analog ICs. I also received the Semiconductor Research Corporation's Inventor's Recognition Award in 2001 for the development of output prediction logic. I received the 1994 SRC Technical Excellence Award, and I also received the Semiconductor Research Corporation's

1988 SRC Inventor's Award. Additional honors and awards are set forth in my CV.
See INFINEON-1004.

12. I have supervised numerous graduate students, and indeed 37 Ph.D. students have graduated under my guidance. A number of doctoral theses that I have supervised relate specifically to integrated circuits, logic design, clock signaling and other circuit technologies. My students and I have published a number of peer-reviewed articles. Several of these articles received accolades in the field. For example, in 2017, we received a best paper award at the prestigious IEEE PhD Research in Microelectronics and Electronics Conference (PRIME).

13. My CV includes a list of publications on which I am a named author. It contains further details regarding my experience, education, publications, and other qualifications to render an expert opinion in connection with this proceeding.

14. In writing this Declaration, I have considered the following: my own knowledge and experience, including my work experience in mobile systems and networks; my experience in teaching those subjects; and my experience in working with others involved in those fields. In addition, I have analyzed the following publications and materials, in addition to other materials I cite in my declaration:

- U.S. Patent 9,972,381 (INFINEON-1001)
- File history of U.S. Patent 9,972,381 (INFINEON-1002)
- U.S. Publication No. 2004/0148482 ("Grundy") (INFINEON-1005)

- U.S. Publication No. 2005/0204091 (“Kilbuck”) (INFINEON-1006)
- Wiley Electrical and Electronics Engineering Dictionary, 2004 (excerpts) (INFINEON-1007)
- T. Tanaka *et al.*, “A quick intelligent page-programming architecture and a shielded bitline sensing method for 3 V-only NAND flash memory,” in IEEE Journal of Solid-State Circuits, vol. 29, no. 11, pp. 1366-1373, Nov. 1994, doi: 10.1109/4.328638 (INFINEON-1008)
- P. Cappalletti *et al.*, “Flash Memories,” Kluwer Academic Publishers, 1999 (INFINEON-1009)

II. LEGAL PRINCIPLES

15. In forming my opinions and considering the subject matter of the '381 Patent and its claims in light of the prior art, I am relying on certain legal principles that counsel in this case explained to me. My understanding of these concepts is summarized below.

16. I understand that the claims define the invention. I also understand that an unpatentability analysis is a two-step process. First, the claims of the patent are construed to determine their meaning and scope. Second, after the claims are construed, the content of the prior art is compared to the construed claims.

17. I understand that a claimed invention is only patentable when it is new, useful, and non-obvious in light of the “prior art.” That is, the invention, as defined

by the claims of the patent, must not be anticipated or rendered obvious by the prior art.

A. Anticipation

18. I understand that under 35 U.S.C. § 102, a patent claim is invalid if its subject matter was patented or described in a printed publication before the effective filing date of the claimed invention. I have been told that this is referred to as invalidity by anticipation. I have been told that a patent claim is anticipated under § 102 if a single prior art reference discloses all limitations of the claimed invention.

B. Obviousness

19. I understand that a patent claim is invalid as obvious if the claimed invention would have been obvious to a person of ordinary skill in the art (“POSITA”) at the time the claimed invention was made. This means that even if all of the elements of the claim cannot be found in a single prior art reference that would anticipate the claim, a person of ordinary skill in the field who knew about all the prior art would have come up with the claimed invention. I understand that in an obviousness determination, the person of ordinary skill in the art is presumed to have knowledge of all material prior art. I understand that whether a claim is obvious is based upon the determination of several factual issues.

20. I understand that obviousness is a determination of law based on underlying determinations of fact. I understand that these factual determinations

include the scope and content of the prior art, the level of ordinary skill in the art, the differences between the claimed invention and the prior art, and secondary considerations of non-obviousness.

21. In considering obviousness, I understand that one must determine the scope and content of the prior art. I understand that, in order to be considered as prior art to a patent being considered, a prior art reference must be reasonably related to the claimed invention of that patent. A reference is reasonably related if it is in the same field as the claimed invention or is from another field to which a person of ordinary skill in the art would look to solve a known problem.

22. I understand that one must determine what differences, if any, existed between the claimed invention and the prior art.

23. I understand that a patent claim composed of several elements is not proved obvious merely by demonstrating that each of its elements was independently known in the prior art. In evaluating whether such a claim would have been obvious, one may consider whether a reason has been identified that would have prompted a person of ordinary skill in the art to combine the elements or concepts from the prior art in the same way as in the claimed invention. There is no single way to define the line between true inventiveness on the one hand (which is patentable) and the application of common sense and ordinary skill to solve a problem on the other hand

(which is not patentable). For example, market forces or other design incentives may be what precipitated a change, rather than true inventiveness.

24. I understand that whether a prior art reference renders a patent claim unpatentable as obvious is determined from the perspective of a person of ordinary skill in the art at the time of the alleged invention. I have been told that there is no requirement that the prior art contain an express suggestion to combine known elements to achieve the claimed invention, but a suggestion to combine known elements to achieve the claimed invention may come from the prior art, as filtered through the knowledge of one skilled in the art. In addition, I have been told that the inferences and creative steps a person of ordinary skill in the art would employ are also relevant to the determination of obviousness.

25. I understand that there is no rigid rule that a reference or combination of references must contain a “teaching, suggestion, or motivation” to combine references. But I also understand that the “teaching, suggestion, or motivation” test can be a useful guide in establishing a rationale for combining elements of the prior art. I have been told that this test poses the question as to whether there is an express or implied teaching, suggestion, or motivation to combine prior art elements in a way that realizes the claimed invention, and that it seeks to counter impermissible hindsight analysis.

26. I understand that one may consider, e.g., whether (1) the change was merely the predictable result of using prior art elements according to their known functions, or whether it was the result of true inventiveness; (2) there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent; (3) the claimed innovation applies a known technique that had been used to improve a similar device or method in a similar way; (4) the claimed invention would have been obvious to try, meaning that the claimed innovation was one of a relatively small number of possible approaches to the problem with a reasonable expectation of success by those skilled in the art; (5) the invention merely substituted one known element for another known element in order to obtain predictable results; (6) the invention merely applies a known technique to a known device, method, or product to yield predictable results; or (7) known work in the field may have prompted variations of use of the same inventions in the same or different fields due to market forces or design incentives that would have been predictable to a person of ordinary skill in the art.

27. I understand that any assertion of secondary considerations of non-obviousness must be accompanied by a nexus between the merits of the invention and the evidence offered.

III. OVERVIEW OF CONCLUSIONS FORMED

28. This expert Declaration explains the conclusions that I have formed based on my analysis. To summarize those conclusions, based upon my knowledge and experience and my review of the prior art publications listed above, it is my opinion that Claims 1-21 of the '381 Patent are obvious over Grundy in view of Kilbuck.

**IV. BACKGROUND KNOWLEDGE ONE OF SKILL IN THE ART
WOULD HAVE HAD PRIOR TO THE PRIORITY DATE OF THE
'381 PATENT**

29. I am informed that patentability must be analyzed from the perspective of “one of ordinary skill in the art” in the same field as the '381 Patent at the time of the invention. As previously discussed, the relevant time of invention is the patent’s priority date, which is no earlier than September 30, 2005. Moreover, I understand that whether or not a patent application provides “written description” support for patent claims must also be assessed from the perspective of a person of ordinary skill in the art.

30. I am also informed that several factors are considered in assessing the level of ordinary skill in the art, including (1) the types of problems encountered in the art; (2) the prior art solutions to those problems; (3) the rapidity with which innovations are made; (4) the sophistication of the technology; and (5) the education level of active workers in the field.

31. Based on the foregoing and upon my experience in this area, a person of ordinary skill in the art (“POSITA”) relating to the subject matter of the ’381 Patent by the Critical Date (September 30, 2005) would have had a bachelor of science degree in electrical engineering, computer engineering, or a related field, and at least three years of experience in the research, design, or development of electronic memory circuits. Additional education could substitute for experience and vice versa.

32. Based on my experiences, I have a good understanding of the capabilities of a POSITA as I was such an individual at the time of the Critical Date. Moreover, I have taught, participated in organizations, and worked closely with many such persons over the course of my career.

33. Although I surpass this definition of one of ordinary skill in the art now and at the priority date of the ’381 Patent, my analysis regarding the ’381 Patent has been based on the perspective of one ordinary skill in the art as of the priority date of the ’381 Patent.

34. I am also familiar with the knowledge of the person of ordinary skill in the art as of the priority date of the ’381 Patent. I am able to opine on how the person of ordinary skill in the art would have understood the disclosure and claims of the ’381 Patent, the disclosures of the prior art, the motivation to combine the prior

art, and what combinations would have been obvious to one of ordinary skill in the art.

V. INTERPRETATIONS OF THE '381 PATENT CLAIMS AT ISSUE

35. For purposes of my analysis in this IPR proceeding, I understand that the terms that appear in the claims of the '381 Patent should be interpreted according to their “ordinary and customary meaning,” which is sometimes called the *Phillips* standard. In determining the ordinary and customary meaning, I understand that the words of a claim are first given their plain meaning that those words would have had to a POSITA at the time. I also understand that the structure of the claims, the specification, and the file history also may be used to better construe a claim insofar as the plain meaning of the claims cannot be understood. Moreover, I understand that even treatises and dictionaries may be used, albeit under limited circumstances, to determine the meaning attributed by a POSITA to a claim term at the time of filing. I have followed this approach in interpreting terms of the Challenged Claims herein.

VI. THE '381 PATENT

A. Overview of the '381 Patent

36. The '381 Patent, entitled “MEMORY WITH OUTPUT CONTROL,” issued on May 15, 2018 from U.S. Patent Application No. 15/868,219 (“the '219 application”), which was filed on January 11, 2018 and claims priority to September 30, 2005. INFINEON-1001, Cover. The '381 Patent relates to memory systems including data storage and retrieval thereof. Under the “FIELD OF THE

INVENTION” heading, the ’381 Patent states that its invention “relates to a memory architecture for improving the speed and/or capacity of semiconductor Flash memory devices.” *Id.*, 1:43-46.

37. According to the ’381 Patent, a purported problem of memory devices is cross talk. *Id.*, 2:41-53. Cross talk is “the unwanted coupling of signals on wires that are in close proximity” such as among input/output (I/O) pins of a memory device. *Id.* Additionally, cross talk “becomes more of a problem as the operating speed of the memory device increases.” *Id.* The ’381 Patent discloses that “there is a need in the art for memory modules, for use in mobile electronic devices, and solid-state drive applications that have increased memory capacities and/or operating speeds while minimizing the number input pins and wires required to access the memory module.” *Id.*, 2:54-59.

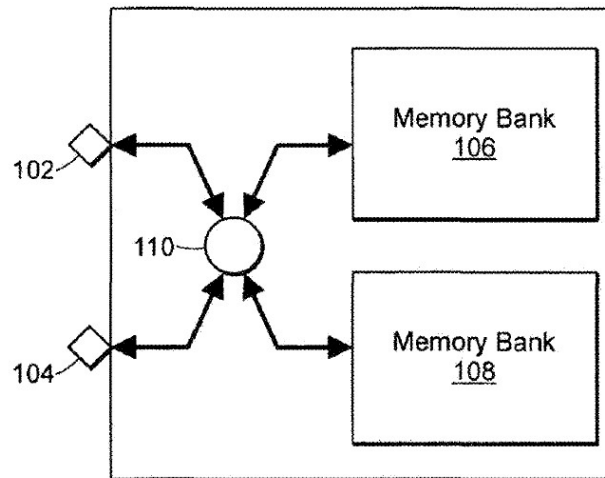
38. The ’381 Patent discloses memory devices with serial data interfaces and explains that an “advantage of a serial interface is a low-pin-count device.” *Id.*, 5:1-20. The ’381 Patent discloses flash memory devices having single or multiple serial data links, memory banks, and control input ports that enable the memory device to transfer serial data to a serial data output port of the memory device:

An apparatus, system, and method for controlling data transfer to an output port of a serial data link interface in a semiconductor memory is disclosed. In one example, a flash memory device may have multiple serial data links, multiple memory banks and control input ports that enable the memory device to transfer the serial data to a serial data output port of the memory device. In another example, a flash memory

device may have a single serial data link, a single memory bank, a serial data input port, a control input port for receiving output enable signals. The flash memory devices may be cascaded in a daisy-chain configuration using echo signal lines to serially communicate between memory devices.

Id., Abstract.

39. The '381 Patent shows in Fig. 1A a high-level diagram of the disclosed memory device in one configuration.



INFINEON-1001, Fig. 1A. As depicted above, the memory device's serial data link interfaces 102 and 104 "can transfer data to and from any of the memory banks 106 and 108."

FIGS. 1A and 1B are high level diagrams showing illustrative memory devices that support concurrent operations, in accordance with various aspects of the invention. FIG. 1A shows a memory device having multiple serial data link interfaces 102 and 104 and multiple memory banks 106 and 108. The presently shown arrangement is referred to herein as a dual port configuration. Each serial data link interface has an associated input/output pin and data input and data output circuitry,

which will be described in further detail with respect to FIG. 2A. Data transferred through a serial data link interface passes through in a serial fashion (e.g., as a single-bit-wide stream of data.) Each of the data link interfaces 102 and 104 in the memory device are independent and can transfer data to and from any of the memory banks 106 and 108. For example, serial data link 102 can transfer data to and from memory bank 106 or memory bank 108. Similarly, serial data link 104 can transfer data to and from memory bank 106 and memory bank 108. Since the two serial data link interfaces shown are independent, they can concurrently transfer data to and from separate memory banks. Link, as used herein, refers to the circuitry that provides a path for, and controls the transfer of, data to and from one or more memory banks. A control module 110 is configurable with commands to control the exchange of data between each serial data link interface 102 and 104 and each memory bank 106 and 108. For example, control module 110 can be configured to allow serial data link interface 102 to read data from memory bank 106 at the same time that serial data link interface 104 is writing data to memory bank 108. This feature provides enhanced flexibility for system design and enhanced device utilization (e.g., bus utilization and core utilization). As will be shown later, control module 110 can include control circuits, registers and switch circuits.

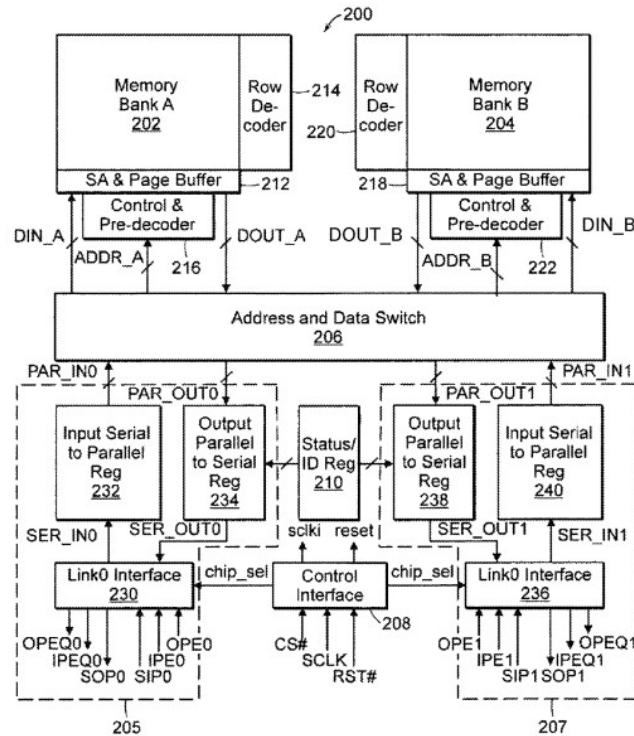
INFINEON-1001, 5:21-54; *see also* 6:35-54.

40. Further, with reference to Fig. 2A (reproduced below), the '381 Patent discloses:

Memory device 200 includes a multiplicity of identical memory banks with their respective data, control and addressing circuits, such as memory bank A 202 and memory bank B 204, an address and data path switch circuit 206 connected to both memory banks 202 and 204, and identical interface circuits 205 and 207, associated with each memory bank for providing data to and for receiving data from the switch circuit

206. Memory banks 202 and 204 are *preferably non-volatile memory, such as Flash memory*, for example.

INFINEON-1001, 6:55-64.



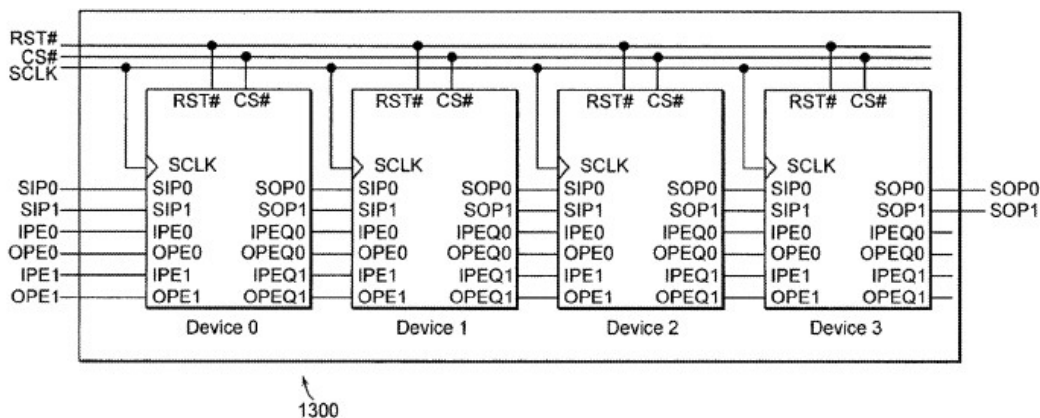
INFINEON-1001, Fig. 2A. The '381 Patent discloses that

Each interface circuit 205/207 receives access data in a serial data stream, where the access data can include a command, address information and input data for programming operations, for example. In a read operation, the interface circuit will provide output data as a serial data stream in response to a read command and address data.

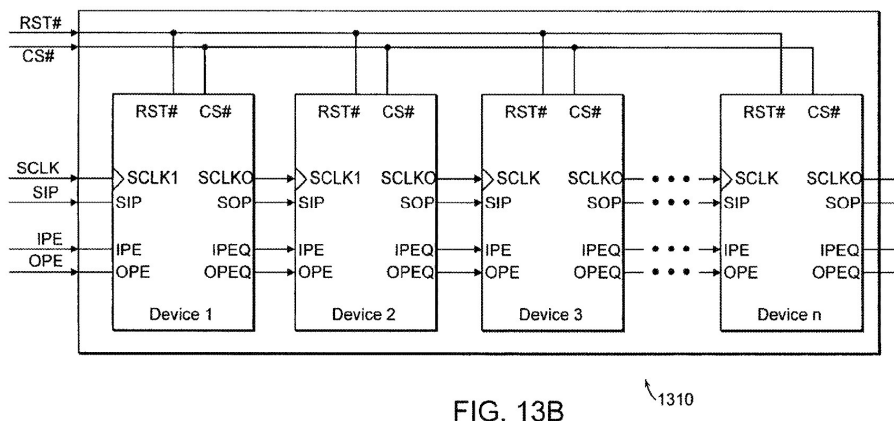
INFINEON-1001, 7:4-9.

41. The '381 Patent also describes that a memory device includes a status register. *Id.*, 7:9-15, 12:46-67, 15:48-16:13, 17:31-51, 18:3-37, Figs. 2A, 2E.

42. The '381 Patent shows in Figs. 13A and 13B examples of a daisy-chain configuration, which serially connects multiple memory devices.



INFINEON-1001, Fig. 13A.



INFINEON-1001, Fig. 13B; *see also* 22:44-23:2, 24:27-42.

43. The '381 Patent describes that each memory device includes additional inputs such as a reset, chip select, and clock input port which is used for synchronization as well as outputs.

- “Those of ordinary skill in the art will understand that serial link interface 205 may further include other device ports such as a chip

select (CS#) pin, a clock signal input (SCLK), or a reset select (RST#) pin” (INFINEON-1001, 13:31-34),

- “system clock (SCLK) signal 304 is an input into the memory device” (14:58-61), and
- “In this cascade configuration a serial clock output port (SCLKO) is added in every device. And serial clock input port name is changed to SCLKI in order to distinguish it from serial clock output port (SCLKO).” (24:35-39).

The ’381 Patent also describes that “in a double data rate (DDR) implementation, both the rising and falling edges of the SCLK clock signal may be used to latch information.” *Id.*, 14:66-15:2.

44. The ’381 Patent includes 21 claims, of which claims 1 and 17 are independent. Independent claim 1 is directed to a flash memory device, whereas claim 17 is directed to a method of operating a flash memory device.

B. File History

45. As part of my preparation of this declaration, I reviewed the file history of the ’381 Patent (INFINEON-1002). I noticed that the Examiner never issued any rejections of the claims of the ’381 Patent based on prior art references, and the claims were promptly allowed.

46. In the reasons for allowance, the Examiner provided the following:

The following is an examiner's statement of reasons for allowance:

The prior art made of record and considered pertinent to the applicant's disclosure, especially when these limitations are considered within the specific combination claimed, does not teach or fairly disclose a flash memory device having the structure as claimed, comprising:

a flash memory comprising a plurality of erasable blocks, each erasable block comprising a plurality of pages, each page comprising a plurality of flash memory cells; a clock input port configured to receive a clock signal;

at least one common data interface configured to transfer command data, address data, input data and output data, wherein at least one of command data, address data, input data and output data is transferred in synchronization with both rising and falling edges of the clock signal when the flash memory device is in a double data rate configuration;

a control input port configured to receive a control signal, wherein a transition of the control signal from an inactive state to an active state indicates a beginning of command data being received at the at least one common data interface;

a control circuitry configured to execute a page program operation to store the input data memory cells in accordance to the command data and address data received at the at least one common data interface; and

a status register configured to indicate a status of the flash memory device.

Independent claim 17 claiming a method for the device similar limitation of independent claim 1 and therefore is allowed for similar reasons.

Dependent claims depend from allowable independent claims and inherently include limitation therein and therefore are allowed as well.

INFINEON-1002, pp. 29-45 (Notice of Allowance). The file history contains no indication that the Examiner had ever considered the prior art combinations advanced in the grounds of this Petition before the '219 application was allowed. As I discuss herein, each of the claimed limitations were well-known in the art as demonstrated by references like Grundy and Kilbuck.

VII. OVERVIEW AND COMBINATIONS OF PRIOR ART REFERENCES

A. Overview of Grundy

47. Grundy, entitled “MEMORY CHAIN,” was filed on January 13, 2004 and published on July 29, 2004. INFINEON-1005, Cover. Under the “FIELD OF THE INVENTION” heading, Grundy states that its invention “relates generally to the field of data processing, and more particularly to high-speed memory systems and components of high-speed memory systems.” INFINEON-1005, [0002]. Both Grundy and the ’381 Patent relate to memory systems including data storage/retrieval thereof. The abstract of Grundy discloses:

A memory system having a plurality of memory devices and a memory controller. The memory devices are coupled to one another in a chain. The memory controller is coupled to the chain and configured to output a memory access command that is received by each of the memory devices in the chain and that selects a set of two or more of the memory devices to be accessed.

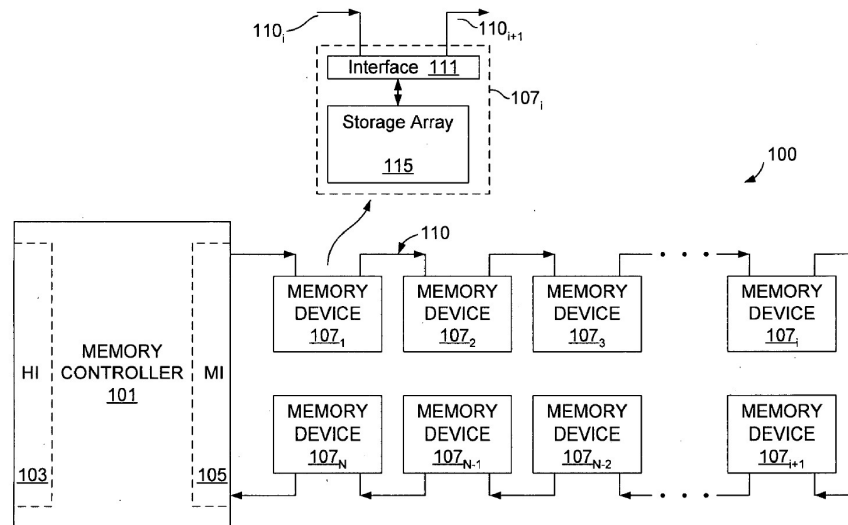
INFINEON-1005, Abstract.

48. Grundy describes a memory system that includes a memory controller and memory devices coupled in a chain configuration, and the memory devices can include flash memory.

FIG. 3 illustrates a memory system 100 according to an embodiment of the invention. *The memory system 100 includes a set of memory devices 107₁-107_N coupled one to another in a chain, with initial and final devices of the chain (107₁ and 107_N, respectively) being coupled to a memory controller 101.* Referring to the detail view of memory device 107_i, *The memory devices 107 each include a signaling interface 111 and a storage array 115, the signaling interface 111*

being used to receive and transmit data and commands propagating through the chain of memory devices, and the storage array 115 (which may include multiple storage arrays) being used for data storage. While the memory devices 107 are generally described below as having **semiconductor storage arrays 115** (e.g., including, but not limited to, volatile storage arrays such as static random access memory arrays (static RAM) and dynamic RAM arrays; **nonvolatile storage arrays** such as programmable read only memory (PROM), erasable PROM (EPROM), electrically erasable or alterable EPROM (EEPROM), **flash EEPROM**, thyristor-based memory devices (so-called negative-device-resistance storage devices) and so forth), the storage arrays 115 may alternatively be formed using other media including, without limitation, various forms of magnetic and optical media.

INFINEON-1005, [0035].



INFINEON-1005, Fig. 3. Grundy’s semiconductor storage arrays 115 can include “nonvolatile storage arrays such as programmable read only memory (PROM), erasable PROM (EPROM), electrically erasable or alterable EPROM (EEPROM), **flash EEPROM**.” *Id.*, Abstract, [0035].

49. A POSITA would have recognized flash EEPROM as being a type of flash memory. Flash EEPROM was an initial name for flash memory upon its invention. In addition to defining flash memory, a technical dictionary also defines “flash EEPROM” as “flash memory utilized as a programmable ROM chip.”

flash EEPROM Flash memory utilized as a programmable ROM chip. It is similar to other programmable ROM chips, such as EEPROMs, but do not permit bit by bit writing and erasing. Also called **flash EPROM**.

flash memory 1. A memory chip that retains its content without power, and which is written and erased in blocks of a fixed size. It is similar to programmable ROM chips, such as EEPROMs, but do not permit bit by bit writing and erasing. Used, for instance, for storage in portable or handheld computers such as notebooks and PDAs, in cellular phones, MP3 players, or to store the BIOS of a system. Also called **flash-memory chip**. 2. The content stored in **flash memory** (1).

INFINEON-1007, pp. 288-289 (see “flash EEPROM” and “flash memory” terms).

50. Grundy further explains that the memory controller and memory devices are coupled in a ring by point-to-point signaling links, where each link includes a clock line, valid line, command line, and a set of N multi-purpose lines:

FIG. 5A illustrates a chained memory system 150 according to an embodiment of the invention. *The memory system includes a memory controller 151 and eight chained memory devices 155₁-155₈ coupled in a ring by point-to-point signaling links 153.* Each of the point-to-point signaling links 153 includes *a clock line 152, valid line 154, command line 156 and a set of N multi-purpose lines 158.* The clock line 152, valid line 154 and command line 156 are used to conduct a clock signal, valid signal and command signal, respectively. The clock signal is used to time receipt of the valid signal, command signal and

signals on the multipurpose lines 158. The valid signal and command signal are protocol control signals. In one embodiment, the valid signal is asserted (e.g., raised to a logic high level) by an upstream device to indicate that a valid command or data value is being transmitted on the multipurpose lines 158. The command signal is asserted when the value transmitted on the multipurpose lines is a command (which may include address information), and deasserted otherwise.

INFINEON-1005, [0043]. Grundy further describes:

The multi-purpose lines 158 are used at alternate times to transfer N-bit wide data values and command values and are referred to collectively herein as a bus. The term “bus” is used herein to mean a signaling structure for parallel transfer of bits of a multi-bit value. A bus may be a point-to-point link as in FIG. 5A, or a multi-drop structure. Also, while any or all of the signal lines within a given point-to-point signaling link 153 may be single-ended, each of the individual signal lines are preferentially differential (i.e., each formed by a pair of signal lines) to enable faster signaling rates.

INFINEON-1005, [0044].

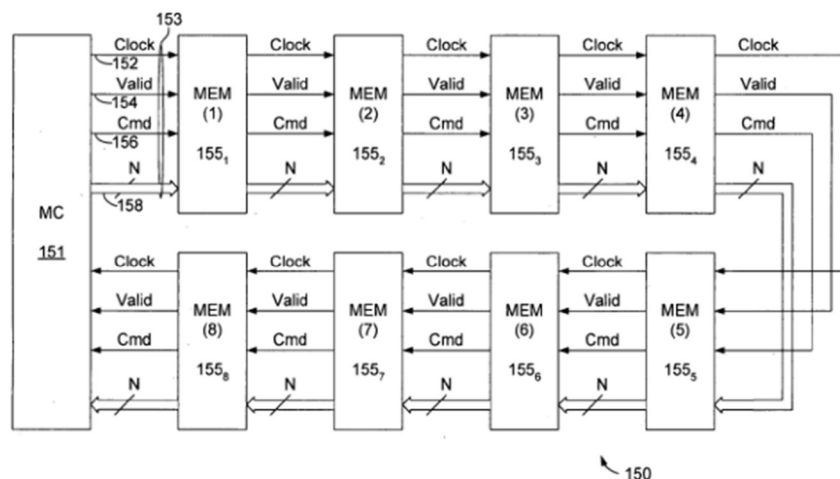


FIG. 5A

INFINEON-1005, [0084].



INFINEON-1005, Fig. 17.

B. Overview of Kilbuck

52. Kilbuck, entitled “NON-VOLATILE MEMORY WITH SYNCHRONOUS DRAM INTERFACE,” was filed on March 11, 2004 and published on September 15, 2005. INFINEON-1006, Cover. Under the “FIELD OF THE INVENTION” heading, Kilbuck’s “invention relates generally to memory devices and in particular the present invention relates to non-volatile memory devices with synchronous interfaces.” INFINEON-1006, [0001]. Both Kilbuck and the ’381 Patent relate to memory systems including data storage/retrieval thereof.

The abstract of Kilbuck discloses:

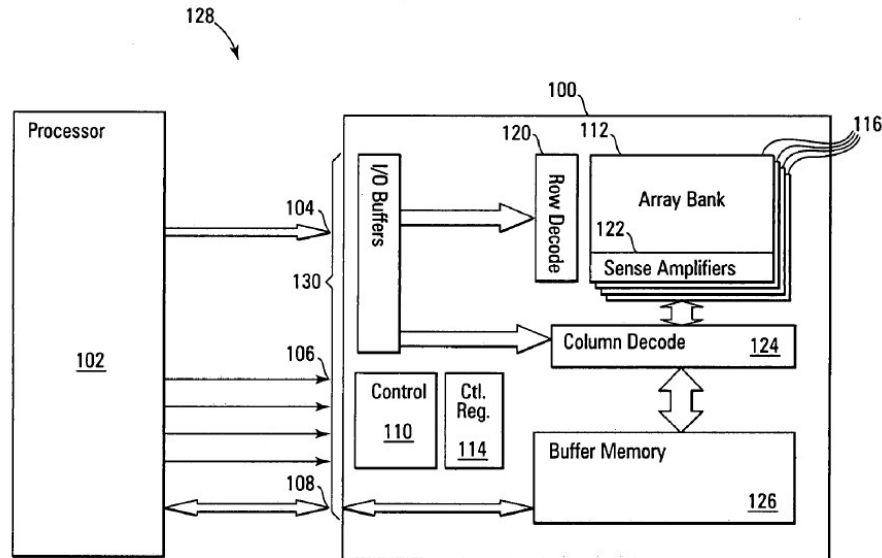
A high density non-volatile memory system, card, and device is described that incorporates a synchronous interface. This is accomplished through use of an external or embedded controller and/or memory buffer to manage the high density non-volatile memory device(s) to present it as a conventional memory device having a synchronous interface that is accessible by row and column address. This allows the high density non-volatile memory to support in-place code execution and allows it to be booted from. Additionally, this incorporation eliminates the overhead of drivers and/or operating system support required to utilize and present conventional high density non-volatile memory devices and moves it internal to the memory device. This simplifies the use and design effort in the overhead and specialized interfacing of high density non-volatile memories and in particular, NAND architecture Flash memories, while reducing the production cost through use of less expensive high density non-volatile memory.

INFINEON-1006, Abstract.

53. Kilbuck describes a memory system that includes a memory controller and one or more memory devices coupled together, and the memory devices can include flash memory.

FIG. 1 shows a simplified diagram of a system 128 incorporating a NAND architecture Flash memory device 100 of the present invention coupled to a host 102, which is typically a processing device or memory controller. It is noted that memory device 100 embodiments of the present invention incorporating other non-volatile memory arrays 112 of differing technology and architecture types (including, but not limited to, Polymer Memory, FeRAM, OUM, MRAM, Molecular Memory, and Carbon Nanotube Memory) are also possible and should be apparent to those skilled in the art with the benefit of the present disclosure. The NAND architecture Flash memory device 100 has a synchronous interface 130 that contains an address interface 104, control interface 106, and data interface 108 that are each coupled to the processing device 102 to allow synchronous memory read and write accesses. Internal to the NAND architecture Flash memory device, an internal memory controller 110 directs the internal operation; managing the Flash memory array 112 and updating RAM control registers and non-volatile erase block management registers 114. The RAM control registers and tables 114 are utilized by the internal memory controller 110 during operation of the Flash memory device 100. The NAND architecture Flash memory array 112 contains a sequence of memory banks or segments 116. Each bank 116 is organized logically into a series of erase blocks (not shown).

INFINEON-1006, [0030].



INFINEON-1006, Fig. 1. Kilbuck provides additional descriptions of system diagrams in Figs. 2, 3A, and 3B. *Id.*, [0038]-[0042].

54. Kilbuck discloses basic properties of flash memories and their attributes. Flash memories comprise memory cells which store data (bits); the memories are arranged into blocks (e.g., each block comprising 16 pages), and further into pages (e.g., each page comprising 64 bytes of data, or 512 bits of data).

Yet another type of non-volatile memory is a Flash memory. A ***Flash memory is a type of EEPROM that can be erased and reprogrammed in blocks instead of one byte at a time.*** A typical Flash memory comprises a memory array, which includes a large number of ***memory cells***. Each of the memory cells includes a floating gate field-effect transistor capable of holding a charge. The data in a cell is determined by the presence or absence of the charge in the floating gate. ***The cells are usually grouped into sections called “erase blocks.”*** The memory cells of a Flash memory array are typically arranged into a “NOR” architecture (each cell directly coupled to a bitline) or a “NAND”

architecture (cells coupled into “strings” of cells, such that each cell is coupled indirectly to a bitline and requires activating the other cells of the string for access, but allowing for a higher cell density). Each of the cells within an erase block can be electrically programmed in a random basis by charging the floating gate. The charge can be removed from the floating gate by a block erase operation, wherein all floating gate memory cells in the erase block are erased in a single operation.

INFINEON-1006, [0004]. *See also*, generally, INFINEON-1008. Kilbuck discloses different types of flash memory cells, including a cell that stores one bit and a cell that stores multiple bits.

This ECC generation in internal firmware/hardware is beneficial, in particular, where the NAND memory array includes multi-level memory cells (MLCs) that store multiple data bits in a single cell, further increasing the memory array 112 density but also increasing the ECC overhead to require the use of dedicated ECC generation in either the memory controller 110 firmware or in specialized ECC hardware.

INFINEON-1006, [0031]; *see also* [0004], [0005]. A cell that stores one bit is referred to as a single-level cell (SLC). A cell that stores multiple bits is referred to as a multi-level cell (MLC). *See also* INFINEON-1006, [0031]; INFINEON-1009, pp. 253-255.

55. Kilbuck discloses flash memory architectures including NAND and NOR type architectures:

As stated above, *the two common types of Flash memory array architectures are the “NAND” and “NOR” architectures*, so called for the similarity each basic memory cell configuration has to the corresponding logic gate design. Both NAND and NOR Flash memory devices have memory cells that are typically arranged in an array of

rows and columns. In NAND Flash memory devices and some NOR Flash memory devices, a row (page) is accessed and then memory cells can be randomly accessed on the page by providing column addresses. This access mode is referred to as page mode access. To read or write to multiple column locations on a page requires the external application of multiple column addresses.

INFINEON-1006, [0024]; *see also* [0004]. Kilbuck also describes how read and writes/programs take place on the memory devices:

Memory access addresses are received on the address interface 104 of the NAND architecture Flash memory device 100 and divided into a row and column address portions. ***On a read access*** the row address is latched and decoded by row decode circuit 120, which selects and activates a row page (not shown) of memory cells across a selected memory bank, the memory cells of the strings associated with the selected row page are also activated in a pass-through mode. The bit values encoded in the output of the selected row of memory cells are coupled to a local bitline (not shown) and a global bitline (not shown) and are detected by sense amplifiers 122 associated with the memory bank. The column address of the access is latched and decoded by the column decode circuit 124. The output of the column decode circuit 124 selects the desired column data from the internal data bus (not shown) that is coupled to the outputs of the individual read sense amplifiers 122 and couples them to an internal buffer memory 126 for transfer from the memory device 100 through the data interface 108. ***On a write access*** the row decode circuit 120 selects the row page and column decode circuit 124 selects write sense amplifiers 122. Data values to be written are coupled from the buffer 126 via the internal data bus to the write sense amplifiers 122 selected by the column decode circuit 124 and written to the selected floating gate memory cells (not shown) of the memory array 112. The written cells are then reselected by the row and column decode circuits 120, 124 and sense amplifiers 122 so that they can be read to verify that the correct values have been programmed into the selected memory cells.

INFINEON-1006, [0030]; *see also* [0039].

56. Kilbuck also teaches a status register configured to indicate a status of the flash memory device.

In one embodiment of the present invention, a special purpose “ready/busy” external connection pin or internal *status register*/flag of the NAND architecture Flash memory device 100 may be monitored to indicate the status of the memory. This allows the memory device 100 to signal that it is busy with a read or write access (due to read or write latency, etc.), and causes the host 102 to wait, halting data transfers when active or set.

INFINEON-1006, [0033]; *see also* [0034].

C. Combination of Grundy and Kilbuck

57. As I discuss below, the Grundy-Kilbuck combination incorporates Kilbuck’s teachings about flash memory, including associated circuitry, into Grundy’s memory system. A POSITA would have been motivated to incorporate the teachings of Kilbuck into Grundy’s system with a reasonable expectation of success. Indeed, Grundy describes that flash memory is a type of non-volatile memory that can be used in Grundy’s system. Kilbuck provides additional details of flash memory systems.

58. I note, as an initial matter, that Grundy and Kilbuck are in the same field of endeavor. Under their respective FIELD OF INVENTION headings, Grundy and Kilbuck both disclose that they relate to memory systems including data storage and retrieval thereof. INFINEON-1005, [0002] (“The present invention relates

generally to the field of data processing, and more particularly to high-speed memory systems and components of high-speed memory systems”); INFINEON-1006, [0001] (“The present invention relates generally to memory devices and in particular the present invention relates to non-volatile memory devices with synchronous interfaces”). Given its similarity to Grundy, a reference like Kilbuck would have been one of the top choices for the POSITA. As I discussed in Sections VII.A and VII.B, Grundy and Kilbuck both describe the use of non-volatile memory, including flash, in their respective memory systems. *See, e.g.*, INFINEON-1005, [0035], [0062]; INFINEON-1006, [0004], [0030]. It is my opinion that a POSITA would have been motivated to pursue the Grundy-Kilbuck combination for multiple reasons.

59. **Motivations:** A POSITA would have sought out teachings from references like Kilbuck to provide additional details for implementing flash memory that Grundy leaves to the discretion of a POSITA. Grundy discloses several types of memory suitable for Grundy’s memory devices including “nonvolatile storage arrays such as ... flash EEPROM.” INFINEON-1005, [0035]. As I have discussed above, flash EEPROM is a type of flash memory.

60. A POSITA implementing Grundy’s memory devices using flash memory would turn to references like Kilbuck to provide additional details about flash memory based systems. As I discussed in Section VII.B, Kilbuck describes

flash memory cells as either single-level or multi-level cells that are arranged into blocks and further arranged into pages. INFINEON-1006, [0004], [0005], [0024]-[0031]. Kilbuck also discloses different types of flash memory architectures such as NAND and NOR type flash memory architectures. INFINEON-1006, [0004], [0024]. Indeed, these and other flash memory attributes were already well known to POSITA, which was acknowledged by the '381 Patent in its background section.

... flash memory is popular because of its high density, non-volatility, and small size relative to hard disk drives. Flash memory technology is based on EPROM and EEPROM technologies. The term “flash” was chosen because a large number of memory cells could be erased at one time as distinguished from EEPROMs, where each byte was erased individually. The advent of multi-level cells (MLC) further increases Flash memory density relative to single level cells. Those of skill in the art will understand that Flash memory can be configured as NOR Flash or NAND Flash, with NAND Flash having higher density per given area due to its more compact memory array structure. For the purpose of further discussion, references to Flash memory should be understood as being either NOR or NAND or other type Flash memory.

INFINEON-1001, 1:62-2:9 (“BACKGROUND OF THE INVENTION”).

61. Further, a POSITA would have sought out teachings from references like Kilbuck to incorporate supportive flash memory circuitry in Grundy’s system based on the teachings and suggestions within Grundy. Kilbuck for example discloses circuitry to “verify that the correct values have been programmed into the selected memory cells.” INFINEON-1006, [0030], [0039]. This verification step increases flash memory’s reliability by ensuring that the memory stores the correct

values in the first place. Kilbuck also provides an internal memory controller to evaluate and generate error correction codes (ECC) to protect data stored in a flash memory. INFINEON-1006, [0031], [0005]. The use of ECC increases the reliability of flash memory because the controller can detect and correct errors. A POSITA would have naturally sought to bolster Grundy's teachings with the additional circuitry details Kilbuck provides.

62. In addition to teachings and suggestions within Grundy regarding flash memory, a POSITA would have sought to incorporate specific features of conventional flash memory devices as provided in Kilbuck to achieve predictable advantages that were known to flow from these features. Under the "BACKGROUND OF THE INVENTION" heading, the '381 Patent discloses that "flash memory is popular because of its high density, non-volatility, and small size relative to hard disk drives." INFINEON-1001, 1:62-64. The '381 Patent acknowledges that known and predictable advantages such as high density, non-volatility, and small size of flash memory were already known to a POSITA. The application of Kilbuck's teachings to Grundy's system would have better allowed Grundy to achieve the known and predictable advantages of flash memory in the system of the resulting combination.

63. A POSITA would have been led to the Grundy-Kilbuck combination because it merely involves the application of a known technique (Kilbuck's

techniques for implementing flash memory) to a known system (Grundy's memory system) to achieve predictable results (a memory system with flash memory and supportive circuitry). Each element in the Grundy-Kilbuck combination merely performs the same function as it does separately.

64. **Reasonable Expectation of Success:** As I discussed in Sections VII.A and VII.B, Grundy and Kilbuck both describe techniques and systems related to non-volatile memory including flash memory. Grundy discloses multiple examples of non-volatile memory including flash that can be used as the storage arrays of Grundy's memory system. INFINEON-1005, [0035], [0062]. Thus, it would have been straightforward from a POSITA's perspective to turn to a reference such as Kilbuck to provide additional details of a flash memory based system. The Grundy-Kilbuck combination represents a combination of known elements each performing the functions it had been known to perform to produce the predictable result of storing and retrieving data. In sum, a POSITA would have reasonably expected a successful outcome from combining Kilbuck's teachings on flash memory devices with Grundy's description of a memory system.

VIII. MANNER IN WHICH THE PRIOR ART REFERENCES GRUNDY AND KILBUCK (GROUND 1) RENDER THE '381 CLAIMS UNPATENTABLE

A. Claim 1

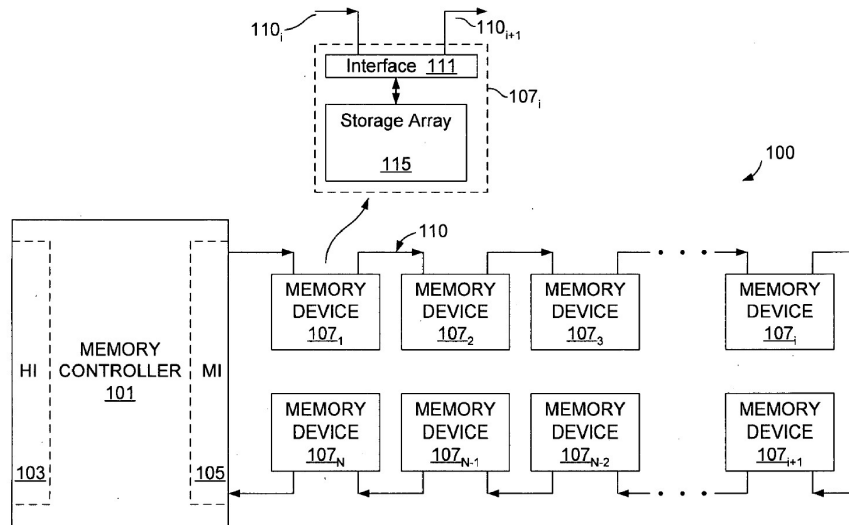
[1pre] A flash memory device comprising:

65. It is my opinion that the Grundy-Kilbuck combination renders [1pre] obvious even to the extent that the preamble is considered a limitation.

66. Grundy describes a memory system that includes a memory controller and memory devices coupled in a chain configuration, and that the memory devices can include flash memory.

FIG. 3 illustrates a memory system 100 according to an embodiment of the invention. *The memory system 100 includes a set of memory devices 107₁-107_N coupled one to another in a chain, with initial and final devices of the chain (107₁ and 107_N, respectively) being coupled to a memory controller 101.* Referring to the detail view of memory device 107_i, The memory devices 107 each include a signaling interface 111 and a storage array 115, the signaling interface 111 being used to receive and transmit data and commands propagating through the chain of memory devices, and the storage array 115 (which may include multiple storage arrays) being used for data storage. While the memory devices 107 are generally described below as having *semiconductor storage arrays 115* (e.g., including, but not limited to, volatile storage arrays such as static random access memory arrays (static RAM) and dynamic RAM arrays; *nonvolatile storage arrays* such as programmable read only memory (PROM), erasable PROM (EPROM), electrically erasable or alterable EPROM (EEPROM), *flash EEPROM*, thyristor-based memory devices (so-called negative-device-resistance storage devices) and so forth), the storage arrays 115 may alternatively be formed using other media including, without limitation, various forms of magnetic and optical media.

INFINEON-1005, [0035].



INFINEON-1005, Fig. 3; *see also* [0043] and Fig. 5A. As I discussed above, flash EEPROM is a type of flash memory. *See also* INFINEON-1007, p. 288 (“flash EEPROM” is “flash memory utilized as a programmable ROM chip...”)

67. Additionally, Kilbuck discloses a flash memory system that includes one or more flash memory devices. INFINEON-1006, [0004], [0030], [0038], [0040]-[0042], Figs. 1, 2, 3A, and 3B.

[1a] a flash memory comprising a plurality of erasable blocks, each erasable block comprising a plurality of pages, each page comprising a plurality of flash memory cells;

68. It is my opinion that the Grundy-Kilbuck combination renders [1a] obvious.

69. Grundy discloses a flash memory comprising a plurality of erasable blocks.

FIG. 3 illustrates a memory system 100 according to an embodiment of the invention. *The memory system 100 includes a set of memory devices 107₁-107_N coupled one to another in a chain, with initial and final devices of the chain (107₁ and 107_N, respectively) being coupled to a memory controller 101.* Referring to the detail view of memory device 107_i, The memory devices 107 each include a signaling interface 111 and a storage array 115, the signaling interface 111 being used to receive and transmit data and commands propagating through the chain of memory devices, and the storage array 115 (which may include multiple storage arrays) being used for data storage. While the memory devices 107 are generally described below as having *semiconductor storage arrays 115* (e.g., including, but not limited to, volatile storage arrays such as static random access memory arrays (static RAM) and dynamic RAM arrays; *nonvolatile storage arrays* such as programmable read only memory (PROM), erasable PROM (EPROM), electrically erasable or alterable EPROM (EEPROM), *flash EEPROM*, thyristor-based memory devices (so-called negative-device-resistance storage devices) and so forth), the storage arrays 115 may alternatively be formed using other media including, without limitation, various forms of magnetic and optical media.

INFINEON-1005, [0035]; *see also* [0062]. As I have discussed above, a POSITA would have recognized flash EEPROM as being a type of flash memory. INFINEON-1007, p. 288 (“flash EEPROM” is “flash memory utilized as a programmable ROM chip...”).

70. Grundy describes that each memory device includes blocks which are erasable:

Any number of additional parameters may be included within the device parameter packet 201 in alternative embodiments including, without limitation, parameters specific to the nature of the storage media of the memory device (e.g., *block size information in a flash*

EEPROM) and the storage array architecture (e.g., number of memory banks, architecture-specific access restrictions, etc.).

INFINEON-1005, [0062].

Once initialized with device identifiers, the memory devices monitor commands that propagate through the memory chain to determine if the commands require responsive action (e.g., data read, write and **erase actions**, parameter query response, status response, etc.).

INFINEON-1005, [0032]. Grundy further describes actions performed on these blocks which include **erase** actions. As such, a POSITA would have recognized that Grundy's system includes erasable blocks. This is consistent with the definition of flash memory. *See, e.g.,* INFINEON-1007, pp. 288-289 ("flash memory" is "[a] memory chip ... which is written and erased in blocks of a fixed size").

71. Kilbuck also describes that flash memory includes a plurality of erasable blocks. Kilbuck for example discloses:

Yet another type of non-volatile memory is a Flash memory. A **Flash memory is a type of EEPROM that can be erased and reprogrammed in blocks instead of one byte at a time**. A typical Flash memory comprises a memory array, which includes a large number of memory cells. Each of the memory cells includes a floating gate field-effect transistor capable of holding a charge. The data in a cell is determined by the presence or absence of the charge in the floating gate. **The cells are usually grouped into sections called "erase blocks."** The memory cells of a Flash memory array are typically arranged into a "NOR" architecture (each cell directly coupled to a bitline) or a "NAND" architecture (cells coupled into "strings" of cells, such that each cell is coupled indirectly to a bitline and requires activating the other cells of the string for access, but allowing for a higher cell density). Each of the

cells within an erase block can be electrically programmed in a random basis by charging the floating gate. The charge can be removed from the floating gate by a block erase operation, wherein all floating gate memory cells in the erase block are erased in a single operation.

INFINEON-1006, [0004]. Kilbuck also discloses: “Because all the cells in an erase block of a Flash memory device must be erased at once...” (INFINEON-1006, [0028]), and “Each bank 116 is organized logically into a series of erase blocks” ([0030]).

72. Grundy further discloses that each erasable block comprises a plurality of pages. Grundy discloses various page-related aspects of a memory device:

- “The write data packet may be transferred to a page buffer” (INFINEON-1005, [0093]),
- “the memory page stored in the page buffer” ([0101]),
- “the drive protocol engine 458 maintains an open page list that indicates the open *pages* within individual memory devices of the memory chain” ([0109]),
- “memory devices having a page buffer architecture” ([0062]).

Further, it was well-known that flash memory is arrangeable in blocks, and further arrangeable in pages. As such, a POSITA would have understood that each of Grundy’s erasable blocks (discussed above) includes a plurality of pages.

73. Kilbuck also discloses that each erasable block comprises a plurality of pages. As I discussed above, Kilbuck discloses that flash memory has erasable blocks. INFINEON-1006, [0004], [0028], [0030]. Kilbuck further describes that

As stated above, the two common types of Flash memory array architectures are the “NAND” and “NOR” architectures, so called for the similarity each basic memory cell configuration has to the corresponding logic gate design. Both NAND and NOR Flash memory devices have memory cells that are typically arranged in an array of rows and columns. In NAND Flash memory devices and some NOR Flash memory devices, *a row (page) is accessed and then memory cells can be randomly accessed on the page by providing column addresses. This access mode is referred to as page mode access.* To read or write to multiple column locations on a page requires the external application of multiple column addresses.

INFINEON-1006, [0024]; *see also* [0025], [0027], [0030]. A POSITA would have understood that each of Kilbuck’s erasable blocks (discussed above) includes pages.

74. Grundy further discloses that each page comprises a plurality of flash memory cells. As I discussed above, Grundy discloses flash memory. A POSITA would have understood that flash memory has flash memory cells. A cell is the basic building block of flash memory, and where a single-level cell stores one bit of information per cell. Indeed, even the background section of the ’381 Patent describes flash memory as having cells:

... flash memory is popular because of its high density, non-volatility, and small size relative to hard disk drives. Flash memory technology is based on EPROM and EEPROM technologies. The term “flash” was chosen because a large number of memory cells could be erased at one time as distinguished from EEPROMs, where each byte was erased individually. The advent of multi-level cells (MLC) further increases Flash memory density relative to single level cells.

INFINEON-1001, 1:62-2:3. A POSITA would have also understood that each page includes flash memory cells because flash memory cells are the basic building block of flash memory and those cells are organized into pages. INFINEON-1005, [0109], [0101], [0062]. Thus, it would have been apparent or obvious to a POSITA that Grundy's flash memory disclosure teaches a plurality of flash memory cells.

75. Kilbuck also discloses that each page comprises a plurality of flash memory cells.

As stated above, the two common types of Flash memory array architectures are the "NAND" and "NOR" architectures, so called for the similarity each basic memory cell configuration has to the corresponding logic gate design. Both NAND and NOR Flash memory devices have *memory cells that are typically arranged in an array of rows and columns*. In NAND Flash memory devices and some NOR Flash memory devices, *a row (page) is accessed and then memory cells can be randomly accessed on the page by providing column addresses. This access mode is referred to as page mode access*. To read or write to multiple column locations on a page requires the external application of multiple column addresses.

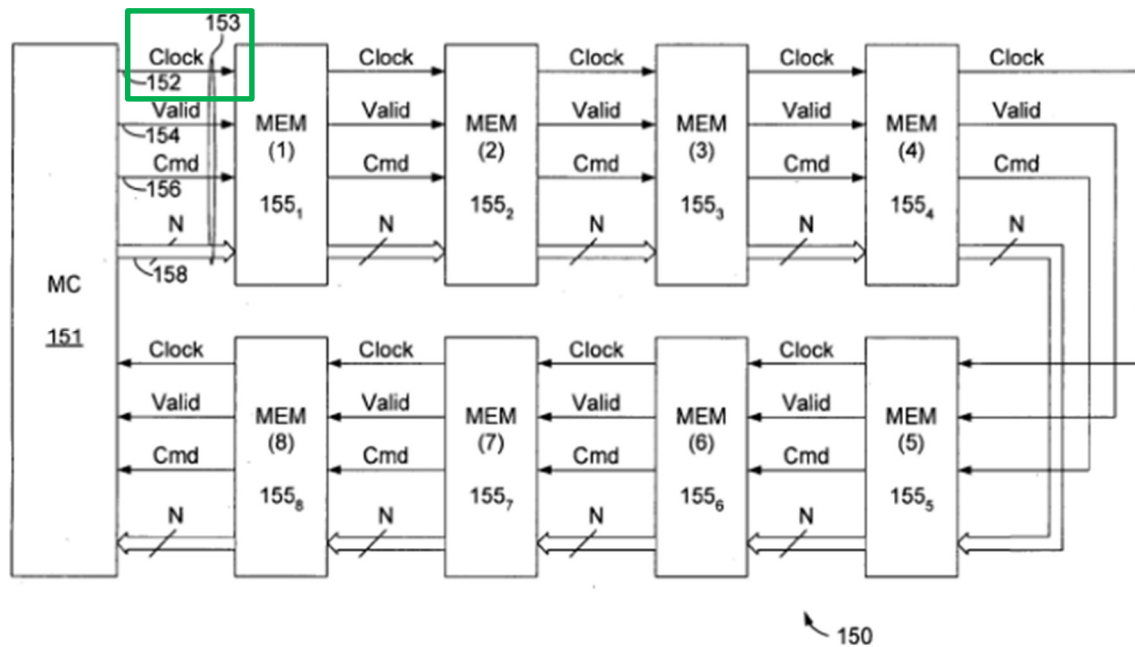
INFINEON-1006, [0024]; *see also* [0025], [0027], [0030], [0004]. A POSITA would have recognized that each of Kilbuck's pages includes flash memory cells.

[1b] a clock input port configured to receive a clock signal;

76. It is my opinion that the Grundy-Kilbuck combination renders [1b] obvious. Grundy's memory device includes a clock input port configured to receive a clock signal. Grundy discloses:

FIG. 5A illustrates a chained memory system 150 according to an embodiment of the invention. The memory system includes a memory controller 151 and eight chained memory devices 155₁-155₈ coupled in a ring by point-to-point signaling links 153. Each of the point-to-point signaling links 153 includes a **clock line** 152, valid line 154, command line 156 and a set of N multi-purpose lines 158. The **clock line** 152, valid line 154 and command line 156 are used to conduct a **clock signal**, valid signal and command signal, respectively. The **clock signal is used to time receipt of the valid signal, command signal and signals on the multipurpose lines 158.**

INFINEON-1005, [0043]. Each link 153 includes a “clock line 152” which is “used to conduct a **clock signal**.” *Id.* This is shown in Fig. 5A of Grundy:



INFINEON-1005, Fig. 5A (annotated). For simplicity, I have added emphasis to show the clock input port of the initial memory device. However, each memory device in Grundy’s chain includes a clock input port. Grundy further describes that “the memory controller 151 outputs a free-running clock signal 172₁ on the clock

line 152 coupled between the memory controller 151 and the initial memory device 155₁.” *Id.*, [0047]; *see also* Fig. 6.

77. Grundy’s Fig. 17 shows a block diagram of an example of Grundy’s memory device which includes a “high-speed interface” that “operates at the clock rate provided in the *incoming link clock signal*.”

FIG. 17 is a block diagram of a memory device 155 according to an embodiment of the invention. The memory device 155 is divided into two distinct sections: *a high-speed interface and a memory storage structure*. The *high-speed interface is designed to work seamlessly with the incoming and outgoing point-to-point links and operates at the clock rate provided in the incoming link clock signal*. The memory storage structure includes the actual storage array 307 (which may include multiple sub-arrays) and protocol processing.

INFINEON-1005, [0084].

The high-speed interface is formed by I/O circuitry 301, I/O controller 303 and protocol engine 305. The I/O circuitry 301 includes a bank of differential receivers 310 coupled to a signal-input interface 308 of the memory device 155, an interface register 311 coupled to outputs of the differential receivers 310, a select circuit 313 coupled to outputs of the interface register 311, and a bank of differential output drivers 320 having inputs coupled to outputs of the select circuit 313 and outputs coupled to a signal-output interface 309 of the memory device. The *signal-input interface 308 includes interconnection structures (e.g., pins or other contacts and corresponding signal path segments) to couple inputs of the differential receivers 310 to the differential signal lines 152, 154 and 156* that carry the incoming *clock*, valid and command signals, respectively, and the differential signal lines 158₀-158_{N-1} that constituent the input-side bus.

INFINEON-1005, [0085].

[illegible]

78. A POSITA would have recognized from Grundy's disclosure that its memory device includes *a clock input port* (see, e.g., "Clock" / "Clock In") which is configured to receive a clock signal. This is consistent with the '381 Patent which describes "a clock signal input (SCLK)" as a "device port" and as "an input into the memory device."

- 48

- “In this cascade configuration a serial clock output port (SCLKO) is added in every device. And serial clock input port name is changed to SCLKI in order to distinguish it from serial clock output port (SCLKO).” (24:35-39),
- Memory circuit and timing diagrams (Figs. 2A, 2B, 2C, 13A, 13B).

This is also consistent with a technical dictionary which defines a port as “a point or opening which provides an input, output, or interface” and “an input, output, or other point where a signal or energy can be provided to, or taken from, a circuit or system.”

port 1. A point or opening which provides an input, output, or interface. For example, an I/O port, a speaker port, or an input to an electric network. 2. An input, output, or other point where a signal or energy can be provided to, or taken from, a circuit or system. For instance, a point where a meter may be placed for readings. 3. An input, output, or interface between a computer and a peripheral, another computer, or a network. For instance, a data port. 4. An input, output, or interface between a CPU and a peripheral. For example a USB port, or a serial port. Also called **I/O port**. 5. To modify a program so that it can function properly within a different system or architecture. 6. A carefully-dimensioned opening in a loudspeaker enclosure. It is employed for various purposes, but it is mainly used to increase and extend the reproduction of low frequencies. Also called **speaker port**, **duct (5)**, **ducted port**, or **vent (3)**. 7. An opening in a waveguide through which energy can be fed to, or taken from. For example, an opening utilized to take measurements. Also called **waveguide port**.

INFINEON-1007, p. 591.

79. As the discussion above shows, Grundy discloses a clock input port because Grundy’s memory device is coupled with the clock line via, for example one or more “pins or other contacts” of the device which receives Grundy’s clock signal as an input. INFINEON-1005, [0043], [0047], [0084]-[0085], Figs. 5A, 17.

[1c-i] at least one common data interface configured to transfer command data, address data, input data and output data,

80. It is my opinion that the Grundy-Kilbuck combination renders [1c-i] obvious. Grundy's memory device includes at least one common data interface configured to transfer command data, address data, input data and output data.

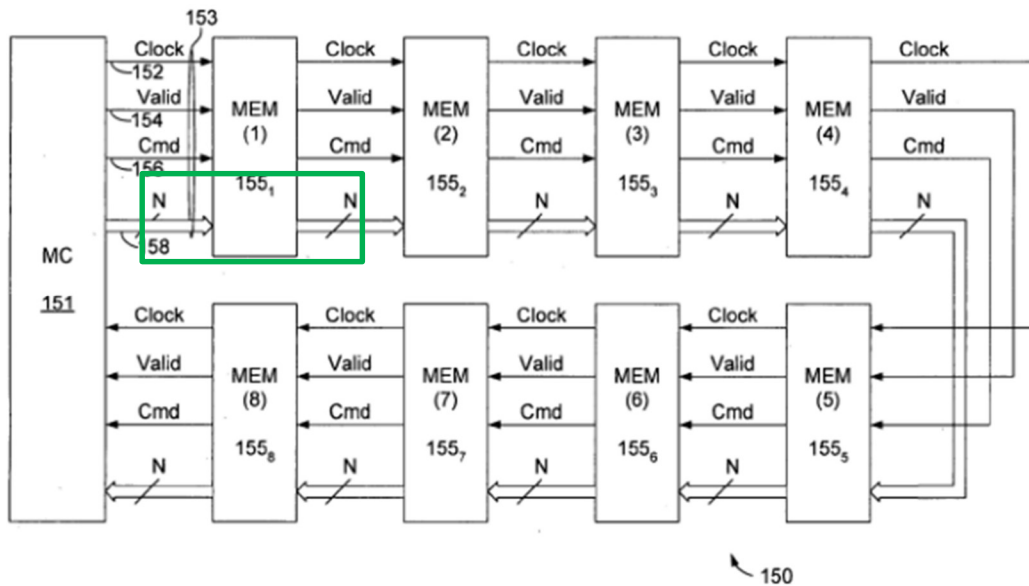
Grundy, for example, discloses:

FIG. 5A illustrates a chained memory system 150 according to an embodiment of the invention. The memory system includes a memory controller 151 and eight chained memory devices 155₁-155₈ coupled in a ring by *point-to-point signaling links 153*. Each of the point-to-point signaling links 153 includes a clock line 152, valid line 154, command line 156 and *a set of N multi-purpose lines 158*. The clock line 152, valid line 154 and command line 156 are used to conduct a clock signal, valid signal and command signal, respectively. The clock signal is used to time receipt of the valid signal, command signal and signals on the multipurpose lines 158.

INFINEON-1005, [0043].

The *multi-purpose lines 158 are used at alternate times to transfer N-bit wide data values and command values* and are referred to collectively herein as a bus. The term "bus" is used herein to mean a signaling structure for parallel transfer of bits of a multi-bit value. A bus may be a point-to-point link as in FIG. 5A, or a multi-drop structure.

INFINEON-1005, [0044]. Fig. 5A, reproduced below, shows these multi-purpose lines 158.



INFINEON-1005, Fig. 5A (annotated). For simplicity, I have added emphasis to show the at least one common data interface of the initial memory device. However, each memory device in Grundy's chain includes one or more common data interfaces.

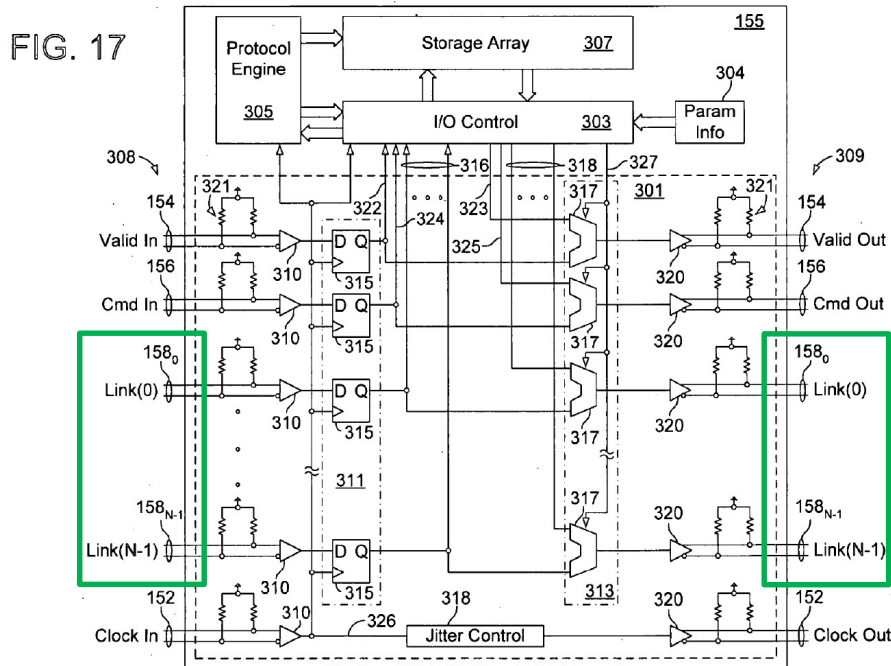
81. I also note that the signaling paths can be bi-directional. Grundy discloses that "the signaling paths 110 may be bi-directional rather than unidirectional, allowing data and/or control signals to travel through the memory chain in either direction." INFINEON-1005, [0041].

82. Fig. 17 of Grundy shows a block diagram of an example of Grundy's memory device. INFINEON-1005, [0021], [0084]. Grundy discloses:

FIG. 17 is a block diagram of a memory device 155 according to an embodiment of the invention. The memory device 155 is divided into two distinct sections: a *high-speed interface* and a memory storage structure. The *high-speed interface is designed to work seamlessly with the incoming and outgoing point-to-point links* and operates at

the clock rate provided in the incoming link clock signal. The memory storage structure includes the actual storage array 307 (which may include multiple sub-arrays) and protocol processing.

INFINEON-1005, [0084]. As I discussed above, this point-to-point links are carried via Grundy's multi-purpose lines 158.



INFINEON-1005, Fig. 17 (annotated). Grundy further discloses:

The high-speed interface is formed by I/O circuitry 301, I/O controller 303 and protocol engine 305. The I/O circuitry 301 includes a bank of differential receivers 310 coupled to a **signal-input interface 308 of the memory device 155**, an interface register 311 coupled to outputs of the differential receivers 310, a select circuit 313 coupled to outputs of the interface register 311, and a bank of differential output drivers 320 having inputs coupled to outputs of the select circuit 313 and outputs coupled to a signal-output interface 309 of the memory device. The **signal-input interface 308 includes interconnection structures (e.g., pins or other contacts and corresponding signal path segments) to couple inputs of the differential receivers 310 to the differential signal lines 152, 154 and 156 that carry the incoming clock, valid and command signals, respectively, and the differential signal lines 158₀-**

158_{N-1} that constitute the input-side bus. The *signal-output interface 309* similarly includes interconnection structures to couple the differential output drivers 320 to the differential signal lines 152, 154 and 156 that carry the outgoing clock, valid and command signals, respectively, and the different signal lines 158_0 - 158_{N-1} that constitute the output-side bus.

INFINEON-1005, [0085].

83. A POSITA would have recognized that Grundy's memory device has a common data interface because it has an interface that transfers command data, address data, input data, and output data via multi-purpose lines 158. For example, Grundy discloses that the multi-purpose lines 158 "transfer N-bit wide *data values and command values*." INFINEON-1005, [0044]. Moreover, Grundy's *high-speed interfaces* (including interfaces 308, 309) are coupled to these multi-purpose lines. INFINEON-1005, [0084]. This is consistent with the description of interfaces in the '381 Patent. INFINEON-1001, 6:55-7:16, 7:61-67, 8:1-9:16, Figs. 2A, 2B. I note that Fig. 2A of the '381 Patent shows distinct serial input ports and serial output ports for each link interface. This is also consistent with the definition of an interface in a technical dictionary.

interface 1. The point or points where two entities meet. For instance, the connection between devices exchanging information, the boundary between two phases of matter such as that between an electrode and an electrolyte, or between semiconductor regions. 2. A device which serves to connect a computer to peripheral devices or a network. Such interfaces include SCSI, RS-232, and network adapters. Also called **computer interface** (2). 3. Any **interface** (1) used in association with a computer, such as those between programs, between hardware and software, between hardware and a user, or between software and a user. Also called **computer interface** (3).

INFINEON-1007, p. 382 (definition of “interface”).

84. In addition to command data, input data, and output data, Grundy’s multi-purpose lines 158 also carry address data: “The command signal is asserted when the value transmitted on the multipurpose lines is a command (which may include address information), and deasserted otherwise.” INFINEON-1005, [0043]. Indeed, commands such as “read and write commands” have associated address data because these commands instruct a memory device to read from or write to select memory addresses. In more detail, a read command will result in output data being transferred via Grundy’s multi-purpose lines 158. A write command will cause input data transferred via Grundy’s multi-purpose lines 158 to be written to memory. *See* generally, INFINEON-1005, [0033], [0044], [0072], [0079]-[0080].

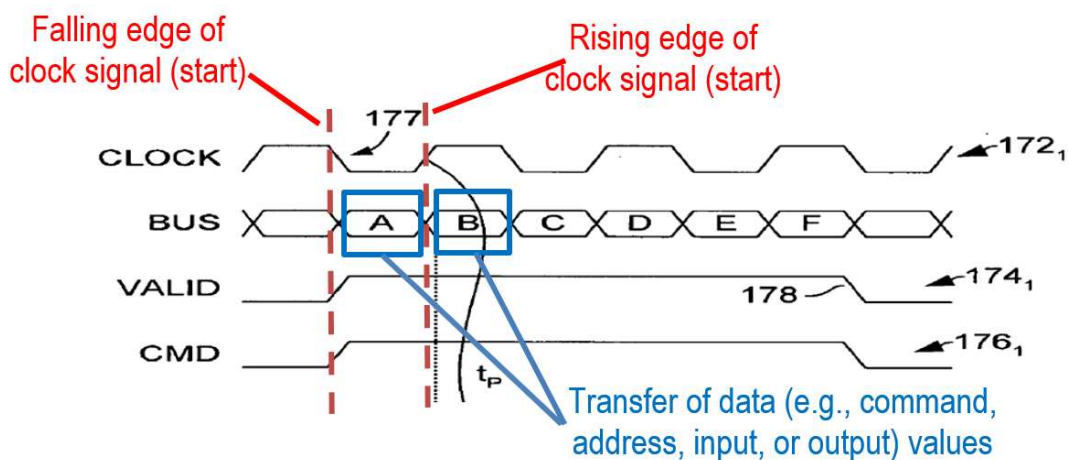
[1c-ii]wherein at least one of command data, address data, input data and output data is transferred in synchronization with both rising and falling edges of the clock signal when the flash memory device is in a double data rate configuration;

85. It is my opinion that the Grundy-Kilbuck combination renders [1c-ii] obvious. Grundy discloses:

When command or data values are to be transmitted on the signaling queue (i.e., the overall transmission path formed by the point-to-point links 153 and memory devices 155), the memory controller 151 simultaneously asserts a valid signal 174₁ on the valid line 154 (and a command signal 176₁ on command line 156 if a command is being transmitted) and drives a first value, A, onto the bus 158, both in response to an *edge [177] of the clock signal 172₁*. The memory controller 151 continues to assert the valid signal 174₁ (and command signal if appropriate) as subsequent values B, C, D, E and F are transmitted onto the bus 158 *in successive transmit intervals*. In the embodiment of FIGS. 5 and 6, *the transmit interval is a half cycle of the clock signal 172₁ such that a new command or data value is transmitted on the bus in response to each rising and falling edge of the clock signal 172₁ following edge [177]*.

INFINEON-1005, [0047]. After reviewing [0047] and Fig. 6 of Grundy, I noticed that in its description of Fig. 6 (which depicts an edge with “177”), Grundy’s specification refers to “edge 167.” A POSITA would have understood that this should be edge 177 because Fig. 6 shows a clock edge with 177.

86. I have reproduced Fig. 6 of Grundy below with annotations to show how Grundy’s data transfer is synchronized with both rising and falling edges of the clock signal. Here, value A is transmitted in response to the falling edge of the clock signal, whereas value B is transmitted in response to the rising edge.



INFINEON-1005, Fig. 6 (excerpted, annotated). I note that, in Fig. 6, Grundy's command signal is asserted, which indicates command data on the bus. The timing relations between the clock and bus apply to other types of data, and in general, the values A, B, C, D, E, and F can be command, address, input, or output. INFINEON-1005, [0043], [0047].

87. Grundy further discloses that a memory device can be in a double data rate configuration:

In one embodiment, a logic level clock signal 326 is generated by the signal receiver 310 coupled to the input-side clock line 152 and is supplied to the protocol engine 305, I/O controller 303 and interface register 315. The clock signal 326 is additionally provided to the output driver 320 coupled to the output-side clock line 152 to drive the outgoing clock signal. A jitter control circuit 318 may optionally be provided to reduce jitter accumulation on the clock signal as it propagates from memory device to memory device. In one embodiment, the interface register 311 includes edge-triggered storage elements 315 (e.g., D flip-flops) coupled to outputs of the differential receivers 310 and having strobe inputs coupled to receive the clock

signal 326 to capture signals generated by the differential receivers 310 at each rising edge and falling edge of the clock signal 326. In one implementation, *such double-data rate capture is achieved by providing a first set of storage elements 315 to capture outputs of the differential receivers 310 in response to each rising edge of the clock signal 326, and a second set of storage elements 315 to capture outputs of the differential receivers 310 in response to a falling edge of the clock signal 326 (or in response to a rising edge of a complement of the clock signal 326).* Other circuits may be used to capture data on both edges of the clock signal in alternative embodiments, and, as discussed above, more or fewer than two signal samples per clock cycle may be captured in the interface register.

INFINEON-1005, [0088]. Grundy describes that data can be transferred in synchronization with both rising and falling edges of the clock signal when the flash memory device is in a double data rate configuration. In addition to Grundy's specification, Grundy explicitly claims that "the interface register within each of the memory devices is configured to store a value present at the buffer input *in synchronism with each rising edge transition of the clock signal*" (claim 12) and "the interface register within each of the memory devices is further configured to store a value present at the buffer input *in synchronism with each falling edge transition of the clock signal*" (claim 13).

88. A POSITA would have recognized that Grundy discloses a double data rate configuration because two values are transmitted for each clock cycle (in synchronization with both rising and falling edges of the clock signal) rather than one. This is further consistent with the '381 Patent, which describes that "[i]n a

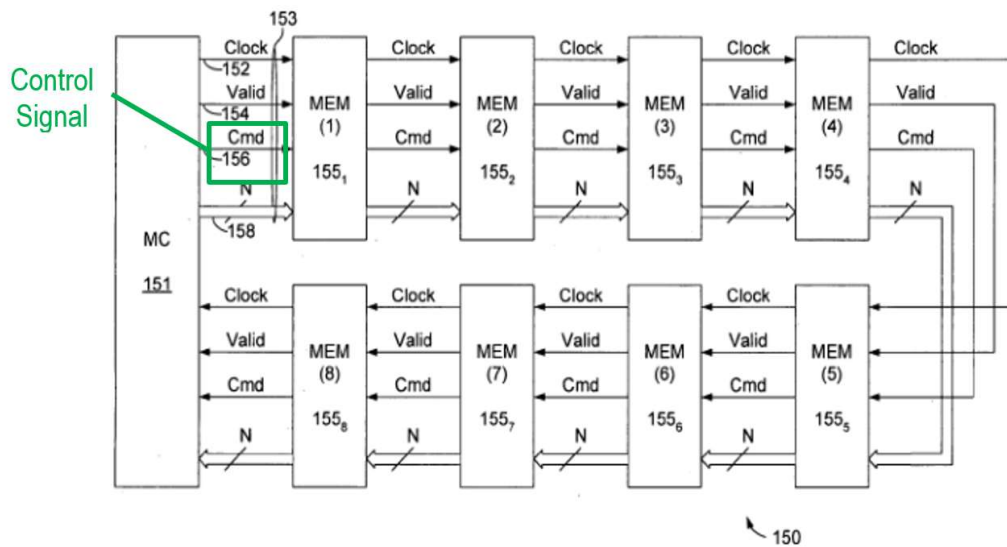
double data rate (DDR) implementation, both the rising and falling edges of the SCLK clock signal may be used to latch information.” INFINEON-1001, 14:66-15:5.

[1d] a control input port configured to receive a control signal, wherein a transition of the control signal from an inactive state to an active state indicates a beginning of command data being received at the at least one common data interface;

89. It is my opinion that the Grundy-Kilbuck combination renders [1d] obvious. Grundy’s memory device includes a control input port configured to receive a control signal, which Grundy refers to as a command signal.

FIG. 5A illustrates a chained memory system 150 according to an embodiment of the invention. The memory system includes a memory controller 151 and eight chained memory devices 155₁-155₈ coupled in a ring by point-to-point signaling links 153. Each of the point-to-point signaling links 153 includes a clock line 152, valid line 154, ***command line 156*** and a set of N multi-purpose lines 158. The clock line 152, valid line 154 and ***command line 156*** are used to conduct a clock signal, valid signal and ***command signal***, respectively. The clock signal is used to time receipt of the valid signal, command signal and signals on the multipurpose lines 158. ***The valid signal and command signal are protocol control signals.*** In one embodiment, the valid signal is asserted (e.g., raised to a logic high level) by an upstream device to indicate that a valid command or data value is being transmitted on the multipurpose lines 158. ***The command signal is asserted when the value transmitted on the multipurpose lines is a command (which may include address information), and deasserted otherwise.***

INFINEON-1005, [0043].



INFINEON-1005, Fig. 5A (annotated). For simplicity, I have added emphasis to show the control input port of the initial memory device. However, each memory device in Grundy's chain includes a control input port.

90. Grundy describes that “[t]he command signal is asserted when the value transmitted on the multipurpose lines is a command (which may include address information), and deasserted otherwise.” INFINEON-1005, [0043]. As such, a POSITA would have recognized that Grundy’s command signal is a *control signal* because Grundy’s command signal, when asserted, *indicates a beginning of command data being received at the at least one common data interface* (e.g., Grundy’s set of N multi-purpose lines 158 as I discussed above for [1c-i]).

91. Further, a POSITA would have recognized that Grundy’s memory device includes a *control input port configured to receive the control signal* (e.g.,

“Cmd” / “Cmd In” in Figs. 5A and 17) because Grundy’s memory device is coupled with “command line 156” which is “used to conduct a ... command signal.” INFINEON-1005, [0043], Fig. 5A. Additionally, Grundy’s Fig. 17 shows circuitry inside an example of Grundy’s memory device that receives the *control signal* (“Cmd In”):

The high-speed interface is formed by I/O circuitry 301, I/O controller 303 and protocol engine 305. The I/O circuitry 301 includes a bank of differential receivers 310 coupled to a ***signal-input interface 308 of the memory device 155***, an interface register 311 coupled to outputs of the differential receivers 310, a select circuit 313 coupled to outputs of the interface register 311, and a bank of differential output drivers 320 having inputs coupled to outputs of the select circuit 313 and outputs coupled to a signal-output interface 309 of the memory device. The ***signal-input interface 308 includes interconnection structures (e.g.,***

pins or other contacts and corresponding signal path segments) to couple inputs of the differential receivers 310 to the differential signal lines 152, 154 and 156 that carry the incoming clock, valid and command signals, respectively, and the differential signal lines 158₀-158_{N-1} that constituent the input-side bus.

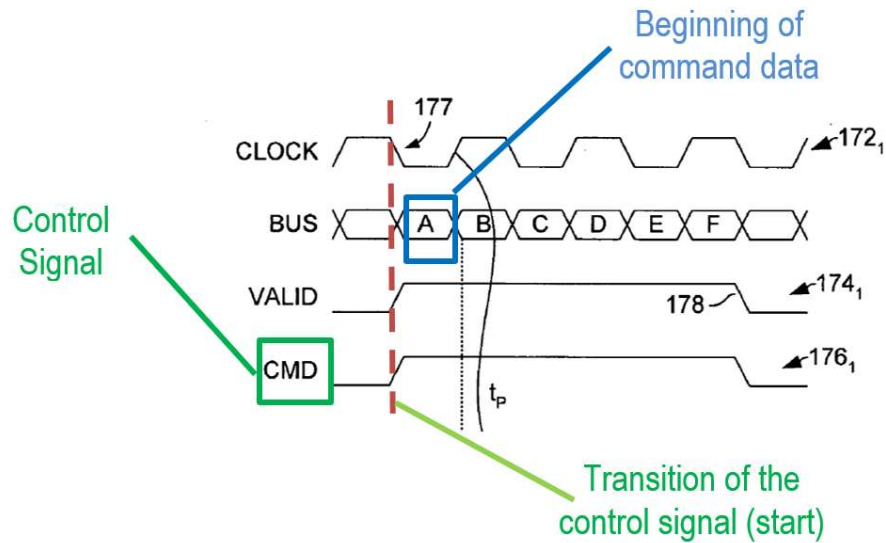
INFINEON-1005, [0085].

92. Grundy further describes a transition of the control signal from an inactive state to an active state indicates a beginning of command data being received at the at least one common data interface. Grundy discloses:

When command or data values are to be transmitted on the signaling queue (i.e., the overall transmission path formed by the point-to-point links 153 and memory devices 155), the memory controller 151 simultaneously asserts a valid signal 174₁ on the valid line 154 (*and a command signal 176₁ on command line 156 if a command is being transmitted*) and drives a first value, A, onto the bus 158, both in response to an edge [177] of the clock signal 172₁. The memory controller 151 continues to assert the valid signal 174₁ (*and command signal if appropriate*) as subsequent values B, C, D, E and F are transmitted onto the bus 158 in successive transmit intervals. In the embodiment of FIGS. 5 and 6, the transmit interval is a half cycle of the clock signal 172₁ such that a new command or data value is transmitted on the bus in response to each rising and falling edge of the clock signal 172₁ following edge [177].

INFINEON-1005, [0047]. As I noted above, “edge 167” should be “edge [177]” to be consistent with Fig. 6 (which depicts an edge with “177”).

93. I have reproduced Fig. 6 of Grundy below with annotations to show how the transition of the control signal (CMD) corresponds to a beginning of command data.



INFINEON-1005, Fig. 6 (excerpted, annotated). Grundy explains that the control signal (CMD / command signal) “is asserted when the value transmitted on the multipurpose lines is a command (which may include address information), and deasserted otherwise.” INFINEON-1005, [0043].

94. Grundy explains that “[a] signal line is said to be ‘activated’ when a signal is asserted on the signal line, and ‘deactivated’ when the signal is deasserted.” INFINEON-1005, [0029]. Grundy explains that a signal is asserted when it is “raised to a logic high level.” INFINEON-1005, [0043]. As shown in Fig. 6, a command signal is asserted on Grundy’s command line, and as such, the command line is activated. *See also* INFINEON-1005, [0047]. A POSITA would have understood that asserting a signal corresponds to a transition of a signal from an inactive state to an active state. Thus, it is my opinion that a POSITA would have recognized that

Grundy discloses a transition of the control signal (“CMD”) from an inactive state to an active state to indicate a beginning of command data (e.g., A, B, C, etc.) being received at the at least one common data interface (interface for “BUS” / Grundy’s set of N multi-purpose lines 158).

[1e] a control circuitry configured to execute a page program operation to store the input data on a selected page, and to execute a read operation to retrieve the output data from the flash memory cells in accordance to the command data and address data received at the at least one common data interface; and

95. It is my opinion that the Grundy-Kilbuck combination renders [1e] obvious.

96. Grundy discloses control circuitry. Each of Grundy’s memory devices includes control circuitry configured to execute operations associated with various commands such as read and page program (write) commands.

- “the memory devices monitor commands that propagate through the memory chain to determine if the commands require responsive action (e.g., data read, write and erase actions, parameter query response, status response, etc.)” (INFINEON-1005, [0032]),
- “Commands sent to target memory devices may include data read and write commands, erase commands, parameter query commands, status requests and may command or request any other action indicated by the device parameter information to be within the capability of the target memory devices” ([0033]),

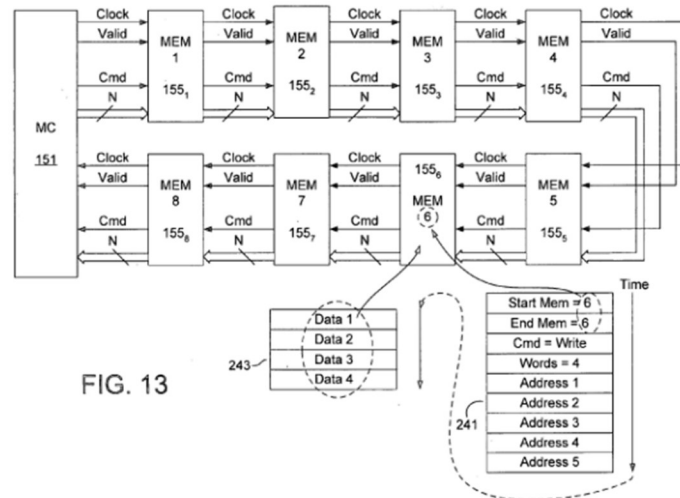
- “The memory system includes a memory controller 151 and eight chained memory devices 155₁-155₈ coupled in a ring by point-to-point signaling links 153” ([0043]),
- Figs., 5A, 9, 10, 17, 19-21.

Additionally, Grundy’s memory controller includes control circuitry configured to execute operations associated with various commands such as read and write commands. INFINEON-1005, [0032], [0033], [0036], [0043].

97. Grundy’s memory device includes control circuitry (discussed above) configured to execute a page program operation to store the input data on a selected page. Grundy describes a “command packet 241 for a write command” which is handled by a memory device:

FIG. 13 illustrates an exemplary *command packet 241 for a write command* directed to memory device 155₆. As shown, the command packet 241 is a long-form command packet in which the device selectors specify device ID ‘6’, *the command specifier indicates a write operation, the number of data words to be written is four (WORDS=4) and in which a set of address words specify the starting write address. A data packet 243 containing the four words to be written is transmitted back-to-back with the command packet* (i.e., transmitted in the first available transmit interval following transmission of the command packet). Though not shown in FIG. 13, error checking values may be included in both the command packet 241 and data packet 243 (e.g., at the end of each) or a single error checking value may be transmitted at the end of the data packet 243 to cover the contents of both the data and command packets.

INFINEON-1005, [0072].



INFINEON-1005, Fig. 13. Grundy also discloses another example of a command packet which is directed to multiple memory devices. INFINEON-1005, [0073], Fig. 14.

98. A POSITA would have understood, or at least would have found it obvious, that Grundy provides a page program operation (e.g., INFINEON-1005, [0072], [0073]) because Grundy's write command packet causes data to be written to memory. This is consistent with the '381 Patent which describes that a page program causes a write: "A 'page program' command may then be used to cause the internal control logic to automatically write the page data to the target address." INFINEON-1001, 17:64-66.

99. A POSITA would have also understood that Grundy's data packet contains input data to be stored on a selected page. First, Grundy discloses various page-related aspects of a memory device:

- “The write data packet may be transferred to a page buffer” ([0093]),
- “the memory page stored in the page buffer” ([0101]),
- “memory devices having a page buffer architecture” ([0062]).

Second, Fig. 13 shows Grundy’s write command packet which includes a set of address words specify the starting write address, and data packet which includes input data 1, 2, 3, 4. INFINEON-1005, [0072]. This set of address words specify the starting write address and corresponds to a page that has been selected for the page program operation.

100. Kilbuck also discloses control circuitry configured to execute a page program operation to store the input data on a selected page. Kilbuck describes for example how a page is selected in response to a write access.

On a *write access* the row decode circuit 120 *selects the row page* and column decode circuit 124 selects write sense amplifiers 122. Data values to be written are coupled from the buffer 126 via the internal data bus to the write sense amplifiers 122 selected by the column decode circuit 124 and written to the selected floating gate memory cells (not shown) of the memory array 112. The written cells are then reselected by the row and column decode circuits 120, 124 and sense amplifiers 122 so that they can be read to verify that the correct values have been programmed into the selected memory cells.

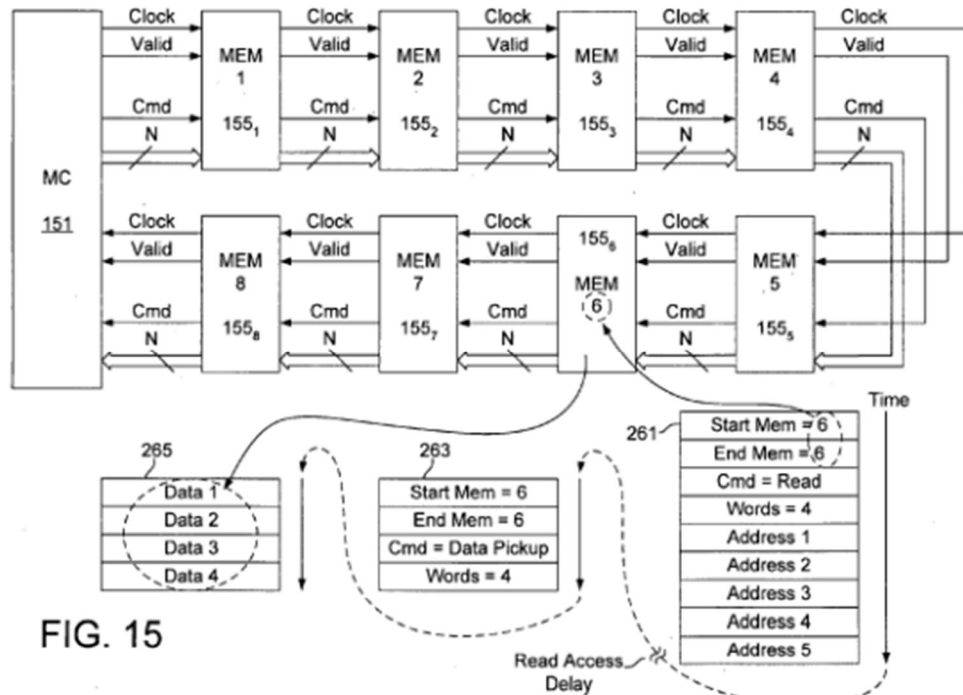
INFINEON-1006, [0030]; *see also* [0032], [0038]-[0042], Figs. 1, 2, 3A, and 3B.

101. Grundy’s memory device includes control circuitry (discussed above) configured to execute a read operation to retrieve the output data from the flash memory cells in accordance to the command data and address data received at the at

least one common data interface. As I discussed for [1a], the Grundy-Kilbuck combination provides that these storage locations are flash memory cells. As I discussed for [1c-i], Grundy's memory device has at least one common data interface which receives command and address data. Grundy further describes a read command:

FIG. 15 illustrates an exemplary data read operation directed to memory device 155₆ within the memory system 150 of FIG. 5A. Initially, the memory controller 151 outputs a ***memory read command*** 261 having device selectors that specify device 155₆, a command specifier that indicates a ***data read operation***, a word count that specifies the number of data words to be read (four in this example, though more or fewer words may be read, depending on the maximum burst length of the memory device), and an ***address value*** (specified by five address words in this example) that specifies a starting memory address within the storage array of memory device 155₆. Memory device 155₆ determines itself to be targeted by the read command, and ***responds by retrieving data words from the storage locations indicated collectively by the starting memory address and the word count***, and by buffering the retrieved data words within an output buffer. The data retrieval operation requires a minimum access time as specified in the parameter information for the target memory device.

INFINEON-1005, [0079].



INFINEON-1005, Fig. 15. Grundy also discloses another example of a read command packet which is directed to multiple memory devices. INFINEON-1005, [0081]-[0083], Fig. 16.

102. Grundy discloses that the retrieved data words are subsequently transmitted within a data packet via the signaling queue which includes Grundy's set of N multi-purpose lines.

After transmitting the read command 261, the memory controller 151 delays for a period referred to herein as the read access delay (according to the minimum access time information stored in the parameter database for the target memory device), then issues a data shuttle command 263 indicating that four data words are to be output onto the signaling queue by memory device 155₆. The memory device responds to the data shuttle command 263 by transmitting a data packet 265 that includes the retrieved data words (i.e., the data words within the output buffer) onto the signaling queue immediately following the re-

transmission of the data shuttle command. The data shuttle command with appended data packet propagates through the remaining memory devices of the memory chain (i.e., memory devices 155₇ and 155₈) to the memory controller, thus concluding the read operation initiated by transmission of the read command 261.

INFINEON-1005, [0080].

103. Kilbuck also discloses control circuitry configured to execute a read operation to retrieve the output data from the flash memory cells in accordance to the command data and address data received at the at least one common data interface.

The NAND architecture Flash memory array 112 contains a sequence of memory banks or segments 116. Each bank 116 is organized logically into a series of erase blocks (not shown). Memory access addresses are received on the address interface 104 of the NAND architecture Flash memory device 100 and divided into a row and column address portions. On a read access the row address is latched and decoded by row decode circuit 120, which selects and activates a row page (not shown) of memory cells across a selected memory bank, the memory cells of the strings associated with the selected row page are also activated in a pass-through mode. The bit values encoded in the output of the selected row of memory cells are coupled to a local bitline (not shown) and a global bitline (not shown) and are detected by sense amplifiers 122 associated with the memory bank. The column address of the access is latched and decoded by the column decode circuit 124. The output of the column decode circuit 124 selects the desired column data from the internal data bus (not shown) that is coupled to the outputs of the individual read sense amplifiers 122 and couples them to an internal buffer memory 126 for transfer from the memory device 100 through the data interface 108.

INFINEON-1006, [0030]; *see also* [0032], [0038]-[0042], Figs. 1, 2, 3A, and 3B.

[1f] a status register configured to indicate a status of the flash memory device.

104. It is my opinion that the Grundy-Kilbuck combination renders [1f] obvious. Grundy, for example, describes status requests and responses to/from a memory device:

- “Commands sent to target memory devices may include ... status requests” (INFINEON-1005, [0033]),
- “the memory devices monitor commands that propagate through the memory chain to determine if the commands require responsive action (e.g., data read, write and erase actions, parameter query response, status response, etc.)” ([0032]),
- “read, write and erase commands issued in short-form command packets 203 are interpreted by the target memory devices as status read, configuration write (e.g., to a device configuration register) and configuration and/or status reset commands, respectively” ([0068]), and
- “memory controller 101 also includes a host interface 103 (HI) to receive memory access requests, addresses and write data from, and to transmit read data and status information to, one or more host devices (not shown) such as a processor, application-specific integrated circuit (ASIC), video controller, direct memory access controller and/or other device capable of issuing memory access requests” ([0036]).

A POSITA would have understood or at least found obvious that data representative of Grundy’s status response would be available from memory such as a register to

ensure the memory device could obtain the status response data to return in response to a status request. Indeed, it was well known that registers can be used to store data such as status including status of a flash memory device.

register 1. A high-speed storage area within the CPU, which is utilized to hold data for specific purposes. Before data may be processed, it must be represented in a register. There are various types, including control registers, index registers, arithmetic registers, and shift registers. Also called **data register**, or **storage register** (1). 2. A device which records and/or stores a value or number. Also called **storage register** (2). 3. To indicate. For example, for an instrument to register a given value. Also, that which is so indicated. 4. To align, or be in proper alignment. Said, for instance of superimposed images.

INFINEON-1007, p. 648 (definition of “register”). A textbook written by Cappalletti discloses that “The user or the external microcontroller can check the status of the operation using either the R/B# pin or the status on DQ0-DQ7 (the former can be issued either in ‘Data Polling - Toggle Bits’ or ‘Status Register’ fashion).”

INFINEON-1009, p. 256.

105. Kilbuck also teaches a status register configured to indicate a status of the flash memory device.

In one embodiment of the present invention, a special purpose “ready/busy” external connection pin or internal **status register**/flag of the NAND architecture Flash memory device 100 may be monitored to indicate the status of the memory. This allows the memory device 100 to signal that it is busy with a read or write access (due to read or write latency, etc.), and causes the host 102 to wait, halting data transfers when active or set.

INFINEON-1006, [0033]; *see also* [0034]. Further, Kilbuck in general discloses the use of registers to store information. *Id.*, [0030], [0033], [0034], [0039]. In my opinion, a POSITA would have found it obvious to apply Kilbuck's teachings regarding details of a flash memory device, including details of a status register of a flash memory device, in combination with Grundy. In addition to the reasons I provided in Section VII.C, it would have been obvious to include in Grundy's memory device a status register as taught in Kilbuck to provide a reliable instrument for storing data representative of Grundy's memory device status that could be readily accessed to provide a status response in reply to a status request as contemplated in Grundy. INFINEON-1005, [0032]-[0033], [0036], [0068]. Indeed, registers were well known and commonly implemented before the Critical Date of the '381 Patent for the purpose of storing information such as status data, especially because registers provide high-speed storage that ensures data can be quickly and efficiently accessed for processing, reporting, and the like.

B. Claim 2

[2] The flash memory device as claimed in claim 1, wherein the command data is latched at the first rising edge of the clock signal after the transition of the control signal.

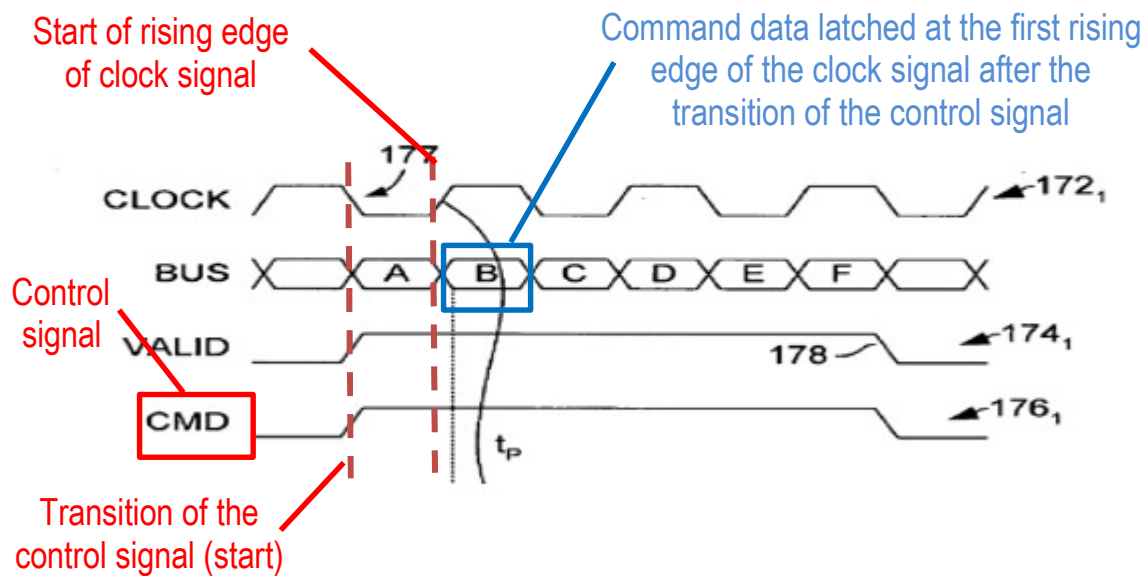
106. It is my opinion that the Grundy-Kilbuck combination renders [2] obvious. As I discussed for claim 1, Grundy discloses the command data, clock signal, control signal, and transition as claimed. Grundy further discloses that the

command data is latched at the first rising edge of the clock signal after the transition of the control signal (which corresponds to Grundy's command "CMD" signal). For example, Grundy discloses:

When command or data values are to be transmitted on the signaling queue (i.e., the overall transmission path formed by the point-to-point links 153 and memory devices 155), the memory controller 151 simultaneously asserts a valid signal 174₁ on the valid line 154 (***and a command signal 176₁ on command line 156 if a command is being transmitted***) and drives a first value, A, onto the bus 158, both in response to an ***edge [177] of the clock signal 172₁***. The memory controller 151 continues to assert the valid signal 174₁ (and command signal if appropriate) as subsequent values B, C, D, E and F are transmitted onto the bus 158 in successive transmit intervals. In the embodiment of FIGS. 5 and 6, ***the transmit interval is a half cycle of the clock signal 172₁ such that a new command or data value is transmitted on the bus in response to each rising and falling edge of the clock signal 172₁ following edge [177]***.

INFINEON-1005, [0047].

107. I have reproduced Fig. 6 of Grundy below with annotations to show when the command data is latched at the first rising edge of the clock signal after the transition of the control signal.



INFINEON-1005, Fig. 6 (excerpted, annotated). When a command is transmitted, Grundy's control signal (command "CMD" signal) is asserted for the transmission of value A and can remain asserted for subsequent values B, C, D, E, and F. Grundy's command can span multiple values (A, B, etc.). INFINEON-1005, [0047], [0065]-[0071], and Figs. 6, 9, 10. As shown in Fig. 6, value A is transmitted in response to the falling edge of the clock signal, whereas value B is transmitted in response to the rising edge. After the transition of the control signal (command "CMD" signal), command data are transmitted on Grundy's bus. Transmission of the B value occurs contemporaneously with the rising edge of the clock signal.

108. Grundy's memory device latches command data based on the clock signal. I have reproduced Fig. 19 of Grundy below with a box around Grundy's command buffer 351, which stores command data.

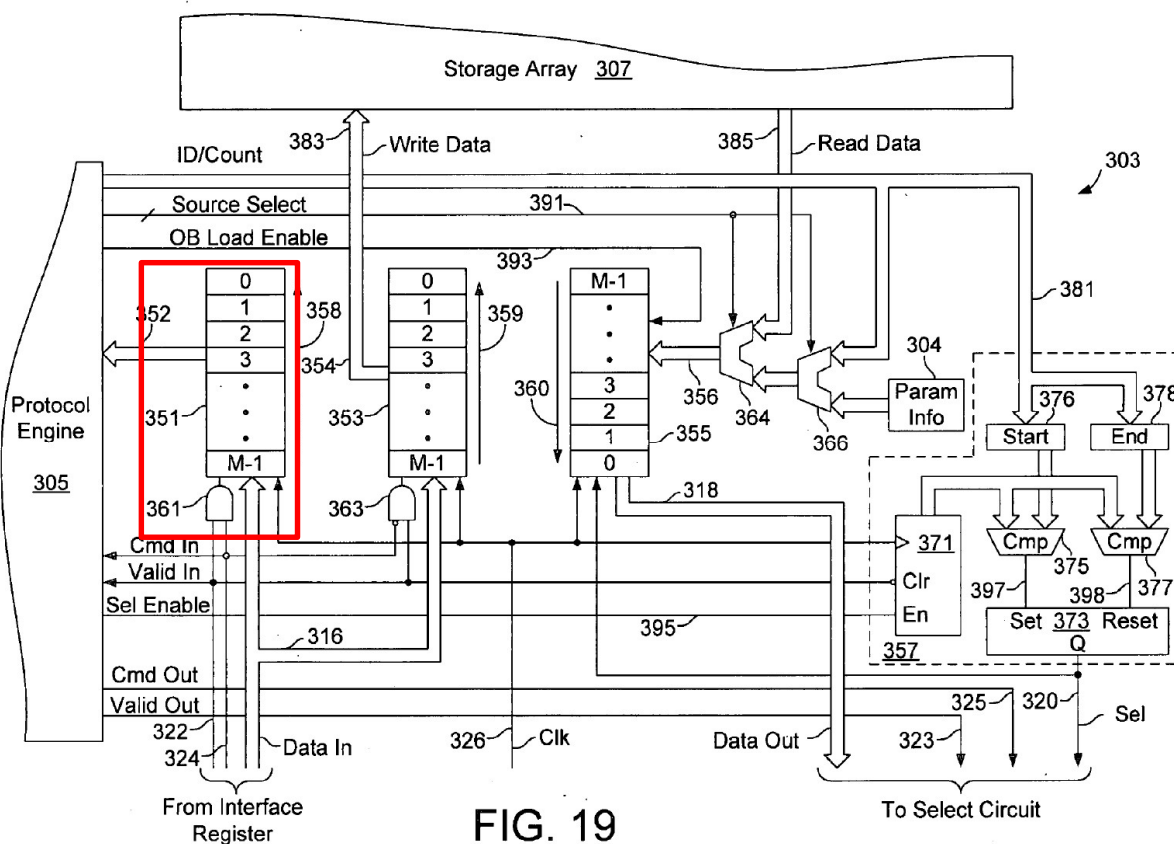


FIG. 19

INFINEON-1005, Fig. 19 (annotated).

FIG. 19 illustrates the I/O controller 303 of FIG. 17 and its interconnection to the protocol engine 305, storage array 307 and I/O circuit 301 according to an embodiment of the invention. The I/O controller 303 includes a **command buffer 351**, input data buffer 353 and output buffer 355, as well as logic AND gates 361 and 363, multiplexers 364 and 366, and select logic 357. The input data buffer and command buffer are first-in-first-out (FIFO) storage structures used to buffer data packets and command packets, respectively. Data packets and command packets are loaded from the interface register into the

input data buffer 353 and **command buffer 351**, respectively, one word at time in response to transitions of the clock signal 326. In one embodiment, the command buffer size is fixed according to the command protocol (e.g., at least deep enough to hold a long-form command packet), and the input data buffer size is determined according to the number of data packets the memory device 155 is designed to store in response to one or more write requests. Alternatively, the command buffer 351, input data buffer 353 and/or output buffer 355 may be formed by a shared storage array, with storage elements allocated to the different buffers according to commands received within the memory device 155. The logic AND gates 361 and 363 are coupled to receive the valid-in and command-in signals 322 and 324 and are used to ***selectively enable*** data words and ***command words to be loaded into the*** input data buffer 353 and ***command buffer 351***, respectively. More specifically, ***when the valid-in signal 322 and command-in signal 324 both go high (or remain high), the output of logic AND gate 361 goes high to enable a corresponding command word to be loaded into command buffer 351 (i.e., via the data-in signal lines 316)***, shifting all other command words forward within the command buffer 351 in the direction shown by arrow 358.

INFINEON-1005, [0092]; *see also* [0097], Fig. 20. A POSITA would have recognized that Grundy discloses latching of command data because a command word is loaded into a command buffer. This is consistent with the description of latching in the '381 Patent. *See, e.g.*, INFINEON-1001, 14:66-15:5.

109. A POSITA would have further recognized that Grundy's clock signal drives this latching of command data. As shown by Fig. 19, Grundy's clock signal 326 is coupled with the command buffer 351. Grundy's clock signal controls when command packets are loaded into the command buffer, e.g., Grundy's device loads

a word of a command packet into the buffer at each transition of the clock signal, one on a rising edge and one on falling edge. Grundy for example discloses:

- “a logic level clock signal 326 is generated by the signal receiver 310 coupled to the input-side clock line 152 and is supplied to the protocol engine 305, I/O controller 303 and interface register 315” ([0088]),
- “data packets and command packets are loaded from the interface register into the input data buffer 353 and command buffer 351, respectively, one word at time in response to transitions of the clock signal 326” ([0092]).

In sum, Grundy discloses that the command data is latched at the first rising edge of the clock signal after the transition of the control signal.

C. Claim 3

[3] The flash memory device as claimed in claim 1, wherein the status register is further configured to provide a result of the page program operation after the page program operation is completed.

110. It is my opinion that the Grundy-Kilbuck combination renders [3] obvious. As I discussed for [1f] and [1e], Grundy-Kilbuck provides for a status register and for the execution of the page program operation. As I explained, Grundy’s memory devices can monitor for a status request and provide a status response. INFINEON-1005, [0032], [0033], [0036], [0068]. A POSITA would have recognized that Grundy provides a result of the page program operation after the page program operation is completed. Additionally, a POSITA would have further recognized that a result of the page program operation is stored after the page

program operation is completed such that Grundy's memory device can provide a response to a status request. A POSITA would have understood or at least found obvious that data representative of Grundy's status response would be available from memory such as a register to ensure the memory device could obtain the status response data such as a result of the page program operation to return in response to a status request.

111. As I explained above, it was well known that registers can be used to store data. *See also* INFINEON-1007, p. 648 (definition of "register"). In fact, Kilbuck discloses the use of registers to store information. INFINEON-1006, [0030], [0033], [0034], [0039]. Kilbuck for example discloses a status register in a flash memory device:

In one embodiment of the present invention, a special purpose "ready/busy" external connection pin or internal *status register*/flag of the NAND architecture Flash memory device 100 may be monitored to indicate the status of the memory. This allows the memory device 100 to signal that it is busy with a read or write access (due to read or write latency, etc.), and causes the host 102 to wait, halting data transfers when active or set.

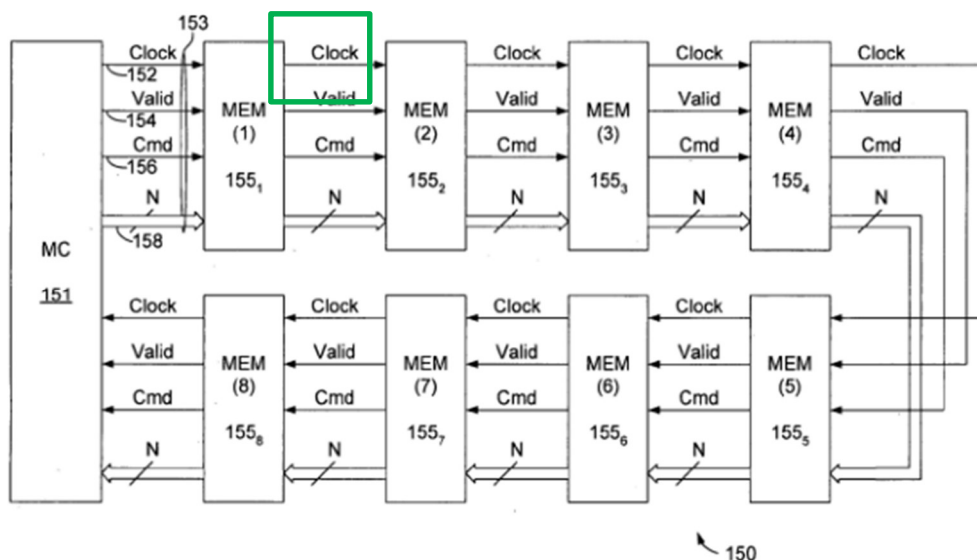
INFINEON-1006, [0033]. A POSITA would have found it obvious to apply Kilbuck's teachings regarding details of a flash memory device, including details of a register such as a status register of a flash memory device, in combination with Grundy for the reasons I provided in Section VII.C. It further would have been obvious to include in Grundy's memory device a status register as taught in Kilbuck

to provide a reliable instrument for storing data representative of Grundy's memory device status that could be readily accessed to provide a status response such as a response indicating a result of the page program operation to return in response to a status request. INFINEON-1005, [0032]-[0033], [0036], [0068].

D. Claim 4

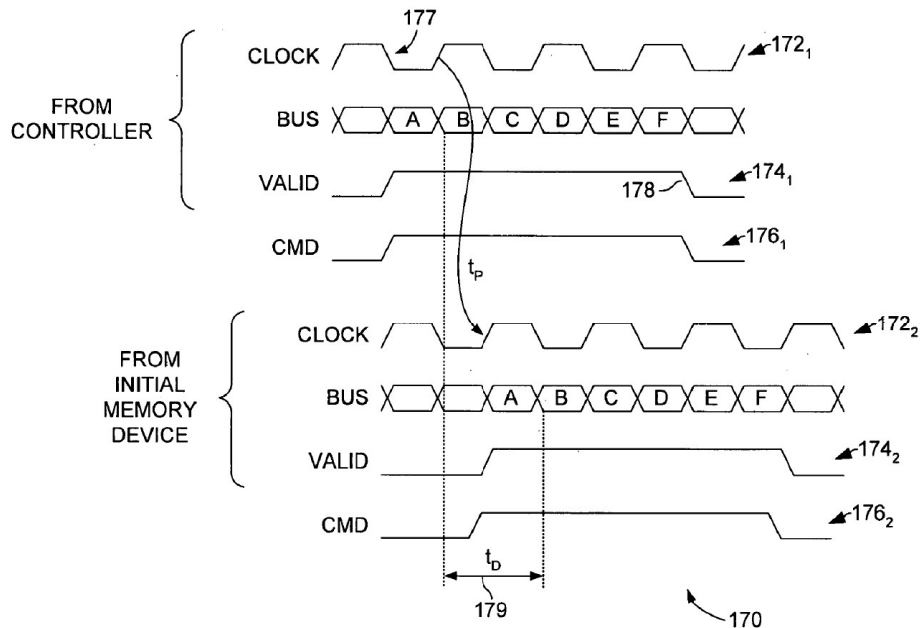
[4] The flash memory device as claimed in claim 1, further comprising a clock output port configured to transmit a clock output signal synchronized with the output data when the flash memory device is in the double data rate configuration.

112. It is my opinion that the Grundy-Kilbuck combination renders [4] obvious. Each of Grundy's memory devices retransmits a clock signal via a clock output port.



INFINEON-1005, Fig. 5A (annotated). For simplicity, I have added emphasis to show the clock output port of the initial memory device. However, each memory device in Grundy's chain includes a clock output port. Grundy discloses:

- “Each of the point-to-point signaling links 153 includes a clock line 152, valid line 154, command line 156 and a set of N multi-purpose lines 158” (INFINEON-1005, [0043]),
- “FIG. 6 is an exemplary timing diagram 170 showing the relative timing of signals transmitted by the memory controller 151 of FIG. 5A on a first point-to-point link 153, and then *retransmitted* by the initial memory device 155₁ on a second point-to-point link 153” ([0047]),
- “Referring to the clock signal 172₂, valid signal 174₂, command signal 176₂ and bus signals retransmitted by the initial memory device 155₁, it can be seen that *the clock signal 172₂ is retransmitted with a propagation delay, t_p , relative to the clock signal 172₁ transmitted by the memory controller 151*” ([0048]).



INFINEON-1005, Fig. 6. Grundy's Fig. 6, for example, shows how a clock output signal is synchronized with the output data.

113. Grundy's Fig. 17 shows a block diagram of an example of a memory device and depicts a clock output port.

FIG. 17 is a block diagram of a memory device 155 according to an embodiment of the invention. The memory device 155 is divided into two distinct sections: *a high-speed interface and a memory storage structure*. The *high-speed interface is designed to work seamlessly with the incoming and outgoing point-to-point links and operates at the clock rate provided in the incoming link clock signal*. The memory storage structure includes the actual storage array 307 (which may include multiple sub-arrays) and protocol processing.

INFINEON-1005, [0084].

The signal-output interface 309 similarly includes interconnection structures to couple the differential output drivers 320 to the differential signal lines **152**, 154 and 156 that carry the *outgoing clock*, valid and command signals, respectively, and the different signal lines 158₀-158_{N-1} that constitute the output-side bus.

INFINEON-1005, [0085].

In one embodiment, a logic level clock signal 326 is generated by the signal receiver 310 coupled to the input-side clock line 152 and is supplied to the protocol engine 305, I/O controller 303 and interface register 315. *The clock signal 326 is additionally provided to the output driver 320 coupled to the output-side clock line 152 to drive the outgoing clock signal*. A jitter control circuit 318 may optionally be provided to reduce jitter accumulation on the clock signal as it propagates from memory device to memory device.

INFINEON-1005, [0088].

signal and the output data would rely on the data to be synchronized with the clock signal to ensure the data could be properly processed according to rising and falling edges of the clock signal.

E. Claim 5

[5] The flash memory device as claimed in claim 4, further comprising a delay lock loop (DLL) to synchronize the clock output signal with the clock signal.

115. It is my opinion that the Grundy-Kilbuck combination renders [5] obvious. Grundy discloses a clock output signal, which I have discussed for claim 4.

Grundy discloses a delay lock loop (DLL):

Referring again to FIG. 5A, it should be noted that different clocking schemes may be used to control the propagation of signals through the signaling queue in alternative embodiments. For example, rather than having each memory device issue a clock signal to its downstream neighbor, a clock signal may be supplied from a single source to all memory devices in the memory system and/or to the memory controller. The memory devices may include *delay-locked loops (DLLs)*, phase-locked loops (PLLs) or other clock-phase adjustment circuitry *to adjust the phase of the clock signals as necessary* to sample incoming signals at a desired sampling instant.

INFINEON-1005, [0051].

In yet other alternative embodiments, the interface register may be omitted from the memory devices 155 to effect a stateless retransmission of commands and data through the signaling queue. In such a system, clock recovery circuitry (e.g., clock data recovery circuitry, phase-locked loop circuitry, *delay-locked loop circuitry*, etc.) *may be used to generate or recover a clock signal having a precisely*

controlled phase, thereby enabling reliable reception and retransmission of signals at each hop of the signaling queue.

INFINEON-1005, [0053].

In one embodiment, a logic level clock signal 326 is generated by the signal receiver 310 coupled to the input-side clock line 152 and is supplied to the protocol engine 305, I/O controller 303 and interface register 315. ***The clock signal 326 is additionally provided to the output driver 320 coupled to the output-side clock line 152 to drive the outgoing clock signal.*** A jitter control circuit 318 may optionally be provided to reduce jitter accumulation on the clock signal as it propagates from memory device to memory device.

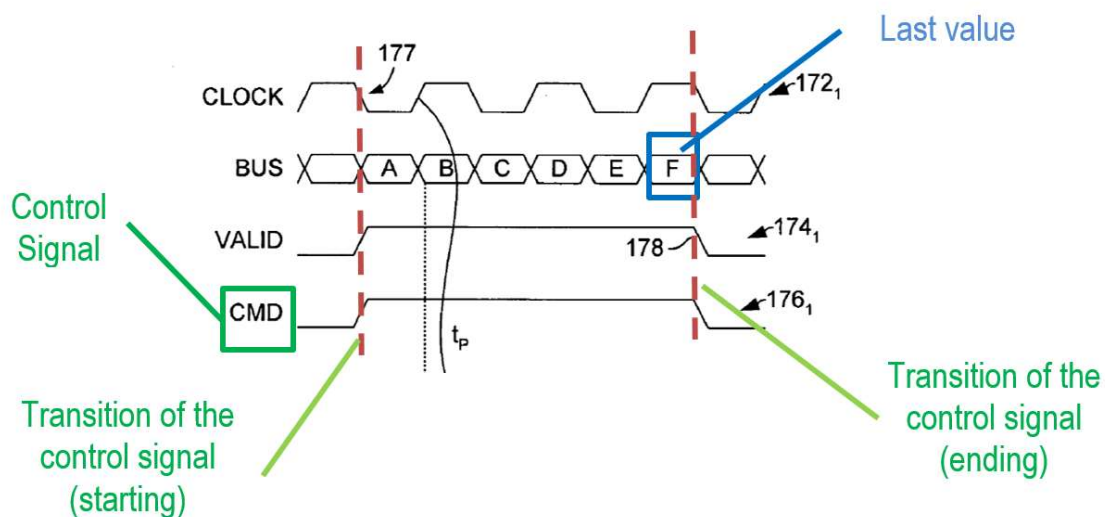
INFINEON-1005, [0088].

116. As I have discussed for claims 1 and 4, each of Grundy's devices retransmits the clock signal, *i.e.*, a device's clock input signal is retransmitted as a clock output signal. The clock output signal becomes the clock input signal to the next device on Grundy's chain. Grundy's delay-locked loop circuitry "may be used to generate or recover a clock signal having a precisely controlled phase, thereby enabling ***reliable reception and retransmission of signals*** at each hop of the signaling queue." INFINEON-1005, [0053]; *see also* [0043], [0047], [0051], [0088], Figs. 5A, 6, 17. As such, a POSITA would have recognized that Grundy's DLL is used to synchronize the clock output signal with the clock signal.

F. Claim 6

[6] The flash memory device as claimed in claim 1, wherein a transition of the control signal from the active state to the inactive state indicates an end of the transfer of the at least one of command data, address data, input data and output data.

117. It is my opinion that the Grundy-Kilbuck combination renders [6] obvious. As I discussed for claim 1, Grundy discloses a control signal (command “CMD” signal) and a transition of the control signal from an inactive state to an active state indicates a beginning of command data being received at the at least one common data interface. Grundy discloses a transition of a command signal that indicates an end of the transfer of command data in Fig. 6 (reproduced below).



INFINEON-1005, Fig. 6 (excerpted, annotated). As is shown in the righthand-side of Fig. 6, the command signal is deasserted after the final value in the command sequence. Grundy discloses:

The valid signal and command signal are protocol control signals. In one embodiment, the valid signal is asserted (e.g., raised to a logic high level) by an upstream device to indicate that a valid command or data

value is being transmitted on the multipurpose lines 158. The ***command signal is*** asserted when the value transmitted on the multipurpose lines is a command (which may include address information), and ***deasserted otherwise***.

INFINEON-1005, [0043].

After the final value in the command or data sequence (value F in this example) has been transmitted, and if no immediately succeeding command or data value is to be transmitted, the memory controller 151 deasserts the valid signal 174₁ as shown at 178.

INFINEON-1005, [0047].

118. Grundy's transition of the control signal from the active state to the inactive state corresponds to Grundy's command signal transitioning from asserted to deasserted. Grundy describes that a signal line is 'deactivated' when the signal is deasserted (e.g., not raised to a logic high level):

- "A signal line is said to be 'activated' when a signal is asserted on the signal line, and ***'deactivated' when the signal is deasserted***" (INFINEON-1005, [0029]),
- "In one embodiment, the valid signal is ***asserted*** (e.g., ***raised to a logic high level***) by an upstream device to indicate that a valid command or data value is being transmitted on the multipurpose lines 158. The command signal is asserted when the value transmitted on the multipurpose lines is a command (which may include address information), and ***deasserted otherwise***" ([0043]).

Deassertion of the command signal on Grundy's command line (e.g., Fig. 6) indicates that the command line is deactivated. A POSITA would have understood that deasserting a signal corresponds to a transition of a signal from an active state to an inactive state. In sum, a POSITA would have recognized that Grundy discloses a transition of the control signal (command "CMD" signal) from the active state to the inactive state indicates an end of the transfer of the at least one of command data, address data, input data and output data.

G. Claim 7

[7] The flash memory device as claimed in claim 1, wherein the flash memory device is a NAND flash memory device.

119. It is my opinion that the Grundy-Kilbuck combination renders [7] obvious. Kilbuck describes a NAND flash memory device in its specification and claims:

- "The memory cells of a Flash memory array are typically arranged into a "NOR" architecture (each cell directly coupled to a bitline) or a "***NAND***" *architecture* (cells coupled into "strings" of cells, such that each cell is coupled indirectly to a bitline and requires activating the other cells of the string for access, but allowing for a higher cell density)" (INFINEON-1006, [0004]),
- "two common types of Flash memory array architectures are the 'NAND' and '***NOR***' *architectures*, so called for the similarity each basic memory cell configuration has to the corresponding logic gate design" ([0024]),

- Claim 10 recites “The non-volatile memory device of claim 1, wherein the non-volatile memory device is one of a *NAND architecture Flash*, a NOR architecture Flash, ...” (Claim 10).

In addition to the reasons I provided above in Section VII.C, it would have been obvious to try NAND type flash memory in the Grundy-Kilbuck combination. There were only a finite number of identified predictable solutions pertaining to the type of flash memory architecture. INFINEON-1006, [0004], [0024]. The background section of the '381 Patent also acknowledges the finite number of types of flash memory architectures. INFINEON-1001, 1:62-2:9 (“[t]hose of skill in the art will understand that Flash memory can be configured as NOR Flash or NAND Flash”; “[f]or the purpose of further discussion, references to Flash memory should be understood as being either NOR or NAND or other type Flash memory”). A POSITA would have selected among one of the finite number of existing flash memory architectures such as NAND or NOR. Moreover, a POSITA would have reasonably expected a successful outcome from combining a NAND type architecture with Grundy’s description of a memory system because the Grundy-Kilbuck combination represents a combination of known elements each performing the functions it had been known to perform to produce the predictable result of storing/retrieving data.

H. Claim 8

[8] The flash memory device as claimed in claim 1, wherein the flash memory device is a NOR flash memory device.

120. It is my opinion that the Grundy-Kilbuck combination renders [8] obvious. Kilbuck describes a NOR flash memory device in its specification and claims:

- “The memory cells of a Flash memory array are typically arranged into a “***NOR***” *architecture* (each cell directly coupled to a bitline) or a “NAND” architecture (cells coupled into “strings” of cells, such that each cell is coupled indirectly to a bitline and requires activating the other cells of the string for access, but allowing for a higher cell density)” (INFINEON-1006, [0004]),
- “two common types of Flash memory array architectures are the ‘NAND’ and ‘***NOR***’ *architectures*, so called for the similarity each basic memory cell configuration has to the corresponding logic gate design” ([0024]),
- Claim 10 recites “The non-volatile memory device of claim 1, wherein the non-volatile memory device is one of a NAND architecture Flash, a ***NOR architecture Flash***, ...”).

In addition to the reasons I provided above in Section VII.C, it would have been obvious to try NOR type flash memory in the Grundy-Kilbuck combination. There were only a finite number of identified predictable solutions pertaining to the type of flash memory architecture. INFINEON-1006, [0004], [0024]. The background section of the '381 Patent also acknowledges the finite number of types of flash memory architectures. INFINEON-1001, 1:62-2:9 (“[t]hose of skill in the art will

understand that Flash memory can be configured as NOR Flash or NAND Flash”; “[f]or the purpose of further discussion, references to Flash memory should be understood as being either NOR or NAND or other type Flash memory”). A POSITA would have selected among one of the finite number of existing flash memory architectures such as NAND or NOR. Moreover, a POSITA would have reasonably expected a successful outcome from combining a NOR type architecture with Grundy’s description of a memory system because the Grundy-Kilbuck combination represents a combination of known elements each performing the functions it had been known to perform to produce the predictable result of storing/retrieving data.

I. Claim 9

[9] The flash memory device as claimed in claim 1, wherein the flash memory cell is a single-level-cell.

121. It is my opinion that the Grundy-Kilbuck combination renders [9] obvious. Kilbuck discloses that the flash memory cell is a single-level-cell.

Yet another type of non-volatile memory is a Flash memory. A Flash memory is a type of EEPROM that can be erased and reprogrammed in blocks instead of one byte at a time. *A typical Flash memory comprises a memory array, which includes a large number of memory cells. Each of the memory cells includes a floating gate field-effect transistor capable of holding a charge. The data in a cell is determined by the presence or absence of the charge in the floating gate.* The cells are usually grouped into sections called “erase blocks.” The memory cells of a Flash memory array are typically arranged into a “NOR” architecture (each cell directly coupled to a bitline) or a

“NAND” architecture (cells coupled into “strings” of cells, such that each cell is coupled indirectly to a bitline and requires activating the other cells of the string for access, but allowing for a higher cell density). Each of the cells within an erase block can be electrically programmed in a random basis by charging the floating gate. The charge can be removed from the floating gate by a block erase operation, wherein all floating gate memory cells in the erase block are erased in a single operation.

INFINEON-1006, [0004]. A POSITA would have recognized that Kilbuck’s disclosure describes a single-level-cell because a single-level-cell stores a single bit of information in which one voltage threshold level is used to distinguish between the presence or absence of the charge in the cell’s floating gate. Furthermore, it was well-known that the original and most basic type of a flash memory cell is a single-level-cell.

122. A single-level-cell stores one bit using a single charge level which is in contrast to the more complex multi-level-cell which store multiple bits. Kilbuck describes that “multi-level memory cells (MLCs) [] store multiple data bits in a single cell.” INFINEON-1006, [0024]. Similarly, the ’381 Patent, in its background section, discloses that “[t]he advent of multi-level cells (MLC) further increases Flash memory density relative to single level cells.” INFINEON-1001, 1:62-2:3.

J. Claim 10

[10] The flash memory device as claimed in claim 1, wherein the flash memory cell is a multi-level-cell.

123. It is my opinion that the Grundy-Kilbuck combination renders [10] obvious. Kilbuck discloses a NAND based flash memory device which can include multi-level memory cells (MLCs).

This ECC generation in internal firmware/hardware is beneficial, in particular, where the NAND memory array includes *multi-level memory cells (MLCs)* that store multiple data bits in a single cell, further increasing the memory array 112 density but also increasing the ECC overhead to require the use of dedicated ECC generation in either the memory controller 110 firmware or in specialized ECC hardware

INFINEON-1006, [0031]; *see also* [0005]. In addition to the reasons I have already provided above, a POSITA would have been motivated to combine Kilbuck's teachings of a MLC with Grundy's memory device to increase the device's storage capacity because storing more bits for cell translates into more storage capacity for the same number of cells.

K. Claim 11

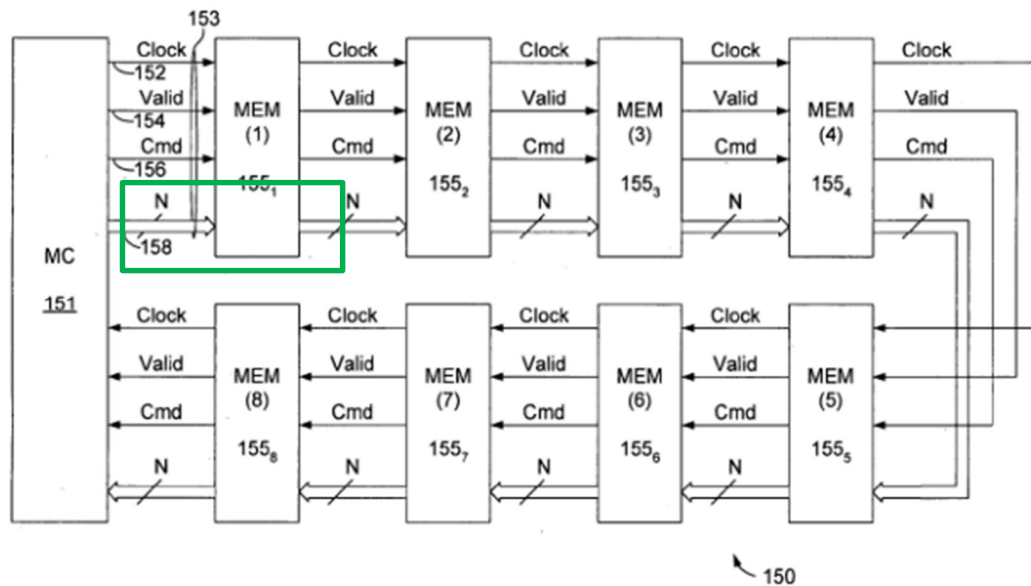
[11] The flash memory device as claimed in claim 1, wherein the at least one common data interface comprises a single-bit serial data input and output.

124. It is my opinion that the Grundy-Kilbuck combination renders [11] obvious. As I discussed for claim 1, Grundy describes a common data interface (e.g., a set of N multi-purpose lines 158). Grundy discloses that "[t]he multi-purpose lines 158 are used at alternate times to transfer N-bit wide data values and command values."

FIG. 5A illustrates a chained memory system 150 according to an embodiment of the invention. The memory system includes a memory controller 151 and eight chained memory devices 155₁-155₈ coupled in a ring by point-to-point signaling links 153. Each of the point-to-point signaling links 153 includes a clock line 152, valid line 154, command line 156 and *a set of N multi-purpose lines 158*. The clock line 152, valid line 154 and command line 156 are used to conduct a clock signal, valid signal and command signal, respectively. The clock signal is used to time receipt of the valid signal, command signal and signals on the multipurpose lines 158. The valid signal and command signal are protocol control signals. In one embodiment, the valid signal is asserted (e.g., raised to a logic high level) by an upstream device to indicate that a valid command or data value is being transmitted on the multipurpose lines 158. The command signal is asserted when the value transmitted on the *multi-purpose lines* is a command (which may include address information), and deasserted otherwise.

The multi-purpose lines 158 are used at alternate times to transfer N-bit wide data values and command values and are referred to collectively herein as a bus. The term “bus” is used herein to mean a signaling structure for parallel transfer of bits of a multi-bit value. A bus may be a point-to-point link as in FIG. 5A, or a multi-drop structure. Also, while any or all of the signal lines within a given point-to-point signaling link 153 may be single-ended, each of the individual signal lines are preferentially differential (i.e., each formed by a pair of signal lines) to enable faster signaling rates.

INFINEON-1005, [0043]-[0044].



INFINEON-1005, Fig. 5A.

125. Grundy further describes that the signals on the multi-purpose lines can be time-multiplexed onto a smaller number of signal lines. In particular, Grundy describes that a serializer circuit can be provided “to serialize an N-bit transmission onto a *single* differential signal line.”

In other embodiments, *signals transmitted on the N signal lines that form the bus may alternatively be time-multiplexed onto a smaller number of signal lines by the upstream device*, then restored to N distinct signal conductors by the recipient, downstream device. For example, in one embodiment, a serializer circuit is provided at the output of the memory controller 151 and each memory device 155 to serialize an N-bit transmission *onto a single differential signal line*, and a deserializer circuit is provided at the input of each memory device 155 and the memory controller 151 to deserialize the N-bit transmission to recover the original N-bit value (i.e., thus effecting a SerDes operation)

INFINEON-1005, [0051].

126. A POSITA would have understood that under at least this implementation, each of Grundy's memory devices would have a single differential signal line as an input and as an output to respectively carry data input and output, and thus would satisfy the single-bit serial data input and output claim language. This is further consistent with the '381 Patent, which discloses a "memory device fully serializes a single set of serial input and output pins which are SIP (Serial Input Port) and SOP (Serial Output Port)." INFINEON-1001, 3:38-43.

L. Claim 12

[12] The flash memory device as claimed in claim 1, wherein the at least one common data interface comprises multi-bit serial data input and output.

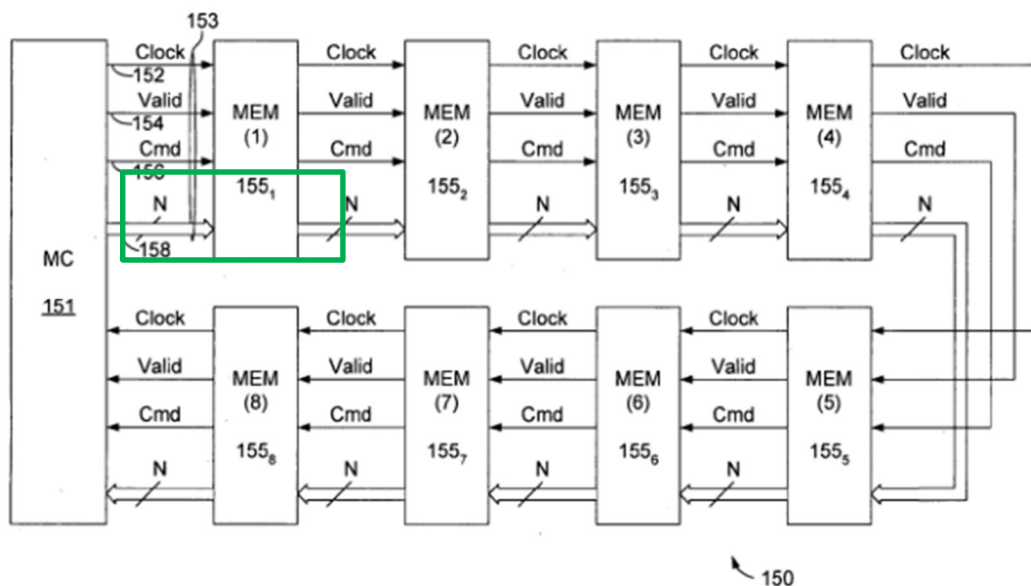
127. It is my opinion that the Grundy-Kilbuck combination renders [12] obvious. As I discussed for claim 1, Grundy describes a common data interface (e.g., a set of N multi-purpose lines 158). Grundy further describes that the signaling links includes "a set of N multi-purpose lines 158" which "are used at alternate times to transfer N-bit wide data values and command values."

FIG. 5A illustrates a chained memory system 150 according to an embodiment of the invention. The memory system includes a memory controller 151 and eight chained memory devices 155₁-155₈ coupled in a ring by point-to-point signaling links 153. Each of the point-to-point signaling links 153 includes a clock line 152, valid line 154, command line 156 and *a set of N multi-purpose lines 158*. The clock line 152, valid line 154 and command line 156 are used to conduct a clock signal, valid signal and command signal, respectively. The clock signal is used to time receipt of the valid signal, command signal and signals on the multipurpose lines 158. The valid signal and command signal are

protocol control signals. In one embodiment, the valid signal is asserted (e.g., raised to a logic high level) by an upstream device to indicate that a valid command or data value is being transmitted on the multipurpose lines 158. The command signal is asserted when the value transmitted on the *multipurpose lines* is a command (which may include address information), and deasserted otherwise.

The multi-purpose lines 158 are used at alternate times to transfer N-bit wide data values and command values and are referred to collectively herein as a bus. The term “bus” is used herein to mean a signaling structure for parallel transfer of bits of a multi-bit value. A bus may be a point-to-point link as in FIG. 5A, or a multi-drop structure. Also, while any or all of the signal lines within a given point-to-point signaling link 153 may be single-ended, each of the individual signal lines are preferentially differential (i.e., each formed by a pair of signal lines) to enable faster signaling rates.

INFINEON-1005, [0043]-[0044].



INFINEON-1005, Fig. 5A (annotated). A POSITA would have understood that

Grundy discloses multi-bit lines because Grundy discloses implementations having

multiple lines, *i.e.*, $N > 1$. A POSITA would have understood that each of Grundy's memory devices, under at least this implementation, would have a set of lines as an input and as an output to respectively carry data input and output, and thus would satisfy the multi-bit serial data input and output claim language.

M. Claim 13

[13] The flash memory device as claimed in claim 1, wherein the control circuitry is configured to perform the read operation during a time period that overlaps with a time period for the page program operation.

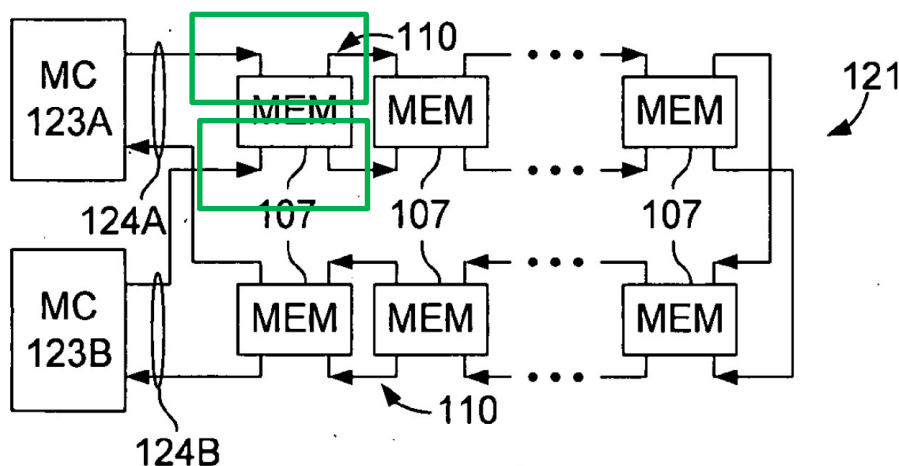
128. It is my opinion that the Grundy-Kilbuck combination renders [13] obvious.

129. As I discussed for claim 1, Grundy's memory device can receive and process commands including read and page program (*i.e.*, write) commands:

- “the memory devices monitor commands that propagate through the memory chain to determine if the commands require responsive action (e.g., data read, write and erase actions, parameter query response, status response, etc.)” (INFINEON-1005, [0032]),
- “Commands sent to target memory devices may include data read and write commands, erase commands, parameter query commands, status requests and may command or request any other action indicated by the device parameter information to be within the capability of the target memory devices” ([0033]).

130. Grundy describes that a memory device can have multiple signaling paths, e.g., multiple links (“point-to-point signaling links 153 includes a clock line 152, valid line 154, command line 156 and a set of N multi-purpose lines 158”).

INFINEON-1005, [0041], [0043], Figs. 4A, 4B, 4C, 4D, 5A. Read and page program commands can be issued on each signaling path. Fig. 4A of Grundy shows an example of a memory device having multiple signaling paths.



INFINEON-1005, Fig. 4A (annotated). I annotated Fig. 4A to show how each memory device has two signaling paths, one depicted on top and another depicted on the bottom.

131. It would have been obvious to a POSITA to modify Grundy's memory device such that it can perform overlapping read and page program operations since the dual signaling paths enable such overlapping of read and page program operations. In my opinion, it would have been obvious to a POSITA to modify Grundy's memory device based on Kilbuck's flash memory disclosures such that Grundy's system can perform overlapping read and page program operations. In addition to the reasons I provided in Section VII.C, a POSITA would have been further motivated to make this modification to increase operational speed and/or

provide “fast random read/write and/or burst capabilities.” INFINEON-1006, [0040]-[0042].

132. Kilbuck describes systems, depicted in Fig. 3A (reproduced below) and 3B, that includes a memory controller and two or more flash memory devices:

The *memory controller 312 is coupled to and controls one or more NAND architecture Flash memory devices 314* via an internal bus 320; controlling the specialized Flash memory interfacing and interaction with the one or more NAND architecture Flash memory devices 314.

The memory controller 312 includes or is coupled to an *internal buffer memory 318 for use in buffering data accesses to and from the NAND architecture Flash memory devices 314*. The internal buffer memory 318 operates in concert with the memory controller 312 buffering data reads and writes, allowing the NAND architecture Flash memory subsystem 304 to be presented as a synchronous memory device that has fast random read/write and/or burst capabilities. This allows read/write buffering to the slow NAND architecture Flash memory devices 314 and allows for in-place code execution from the NAND architecture Flash memory subsystem 304.

INFINEON-1006, [0040].

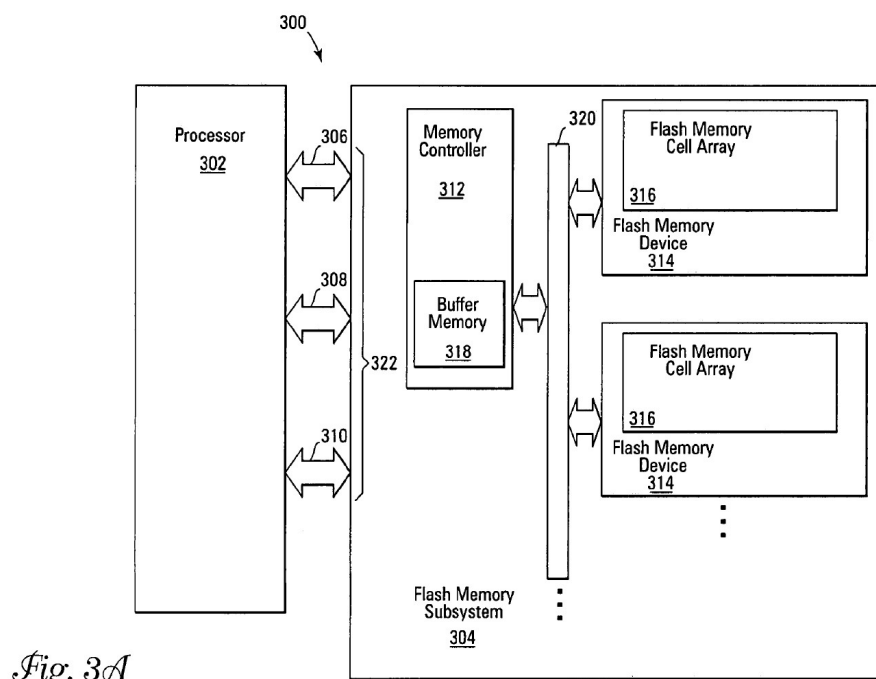


Fig. 3A

INFINEON-1006, Fig. 3A; *see also* [0041]-[0042] and Fig. 3B. Kilbuck’s data accesses, *e.g.*, “data reads and writes” ([0040]), cause read and page write operations, respectively. For the reasons I noted above, it would have been obvious to a POSITA to modify Grundy’s memory device based on Kilbuck’s flash memory architectures such as the one shown by Kilbuck’s Figs. 3A or 3B.

133. A POSITA would have incorporated two or more of Kilbuck’s flash memory devices 314 in Grundy’s memory device such that Grundy’s memory device can perform overlapping operations in response to receiving commands on respective signaling paths (see annotated version of Grundy at Fig. 4A, which I provided above).

134. A POSITA would have understood that each of Kilbuck's flash memory devices 314 is capable of performing read operations and page program (write) operations because each has its own circuitry to access (read/program) its own flash memory array such that one of the memory devices 314 can perform a read operation during a time period that overlaps with a time period for a page program operation which is performed by another one of the memory devices 314. INFINEON-1006, [0030], [0040]-[0042], Figs. 1, 2, 3A, and 3B. Indeed, each of the memory devices 314 has its own buffer, controller, and row/column decode to perform overlapping operations. INFINEON-1006, [0030].

135. Kilbuck's disclosure of data accesses including buffering data reads and writes indicates that read and page program operations may overlap in their time periods. INFINEON-1006, [0032] ("internal buffer memory 126 operates in concert with the memory controller 110 buffering data reads and writes"), [0038], [0040], [0042] ("rapidly changing multiple reads and writes of the memory system occur (such as variable updating by a program)"). This is consistent with a POSITA's knowledge of how a processor typically interacts with a memory (e.g., reading and writing data to/from memory).

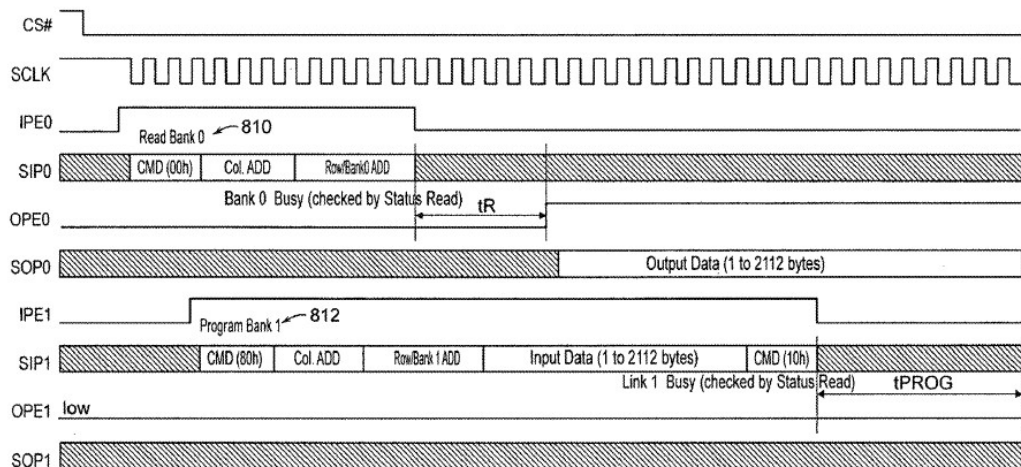
136. The above-discussed Grundy-Kilbuck combination is further consistent with the disclosures and drawings of the '381 Patent which provides timing diagrams for "concurrent operations using dual independent serial data links 230 and 236."

FIGS. 8A, 8B, and 8C illustrate timing diagrams for a *memory device in accordance with aspects of the invention being used to perform concurrent operations using dual independent serial data links 230 and 236*. Some concurrent operations performed by a memory device in accordance with aspects of the invention include, but are not limited to, concurrent read, concurrent program, concurrent erase, read while program, read while erase, and program while erase. FIG. 8A illustrates a concurrent “page read” operation being performed on bank A (bank 202) and bank B (bank 204). In FIG. 8A, bank A is represented as “bank 0” while bank B is represented as “bank 1”. FIG. 8B. Other concurrent operations will become apparent to one skilled in the art upon review of the entire disclosure herein.

INFINEON-1001, 19:1-14.

Furthermore, in accordance with various aspects of the invention, *FIG. 8B illustrates a timing diagram of a “page read” command 810 and a “page program” command 812 directed at different memory banks in a memory device 200 being performed concurrently*. In this example, a read operation (“page read” 810) is being performed in one of the plurality of memory banks (e.g., memory bank 202) through serial data link interface 230. Meanwhile, simultaneously, a write operation (“page program” 812) is being performed in another of the plurality of memory banks (e.g. memory bank 204) through serial data link interface 236. In accordance with various aspects of the invention, each link in the memory device 200 may access any of the memory banks and operate independently.

INFINEON-1001, 19:50-63.



INFINEON-1001, Fig. 8B. For example, Fig. 8B is a timing diagram of concurrent page read and page program commands being performed concurrently. The page read command 810 and page program command 812 are respectively communicated via two dual independent serial data links 230 and 236 and carried out using two separate memory banks 202 and 204. These links and banks are shown in Fig. 2A, which I previously discussed in Section VI.A.

N. Claim 14

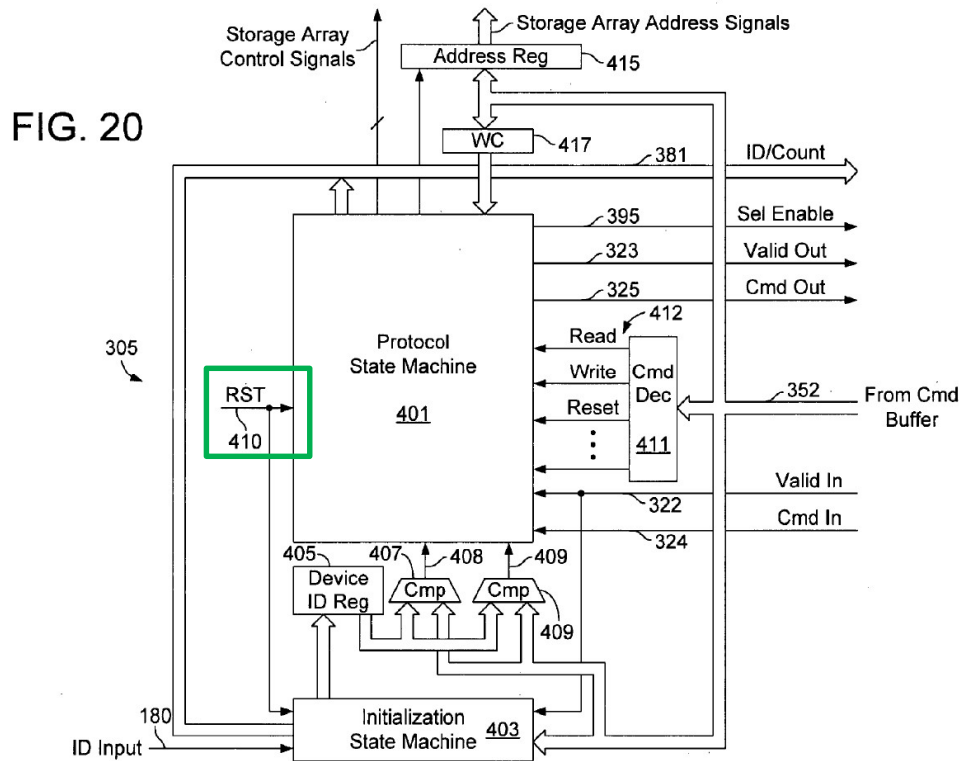
[14] The flash memory device as claimed in claim 1, further comprising a second control input port configured to receive a reset signal, wherein the status register is configured to be reset when the reset signal is received at the second control input port.

137. It is my opinion that the Grundy-Kilbuck combination renders [14] obvious.

138. Grundy's memory device includes protocol engine 305 as shown in Fig. 20 includes an initialization state machine 403.

The initialization state machine 403 is coupled to the command buffer (i.e., element 351 of FIG. 19) via command path 352 and is coupled *to receive a reset signal 410 (RST)*, the device ID input 180 and, as discussed, the valid-in signal 322. The initialization state machine 403 is additionally coupled to the output buffer (i.e., element 355 of FIG. 19) via load path 381 and the multiplexers 366 and 364 described in reference to FIG. 19. When the reset signal 410 is asserted (e.g., during device power-up), the initialization state machine 403 *is reset to an initial state* and remains in the initial state until the valid-in signal 322 is asserted.

INFINEON-1005, [0098]; *see also* [0097], [0099]. The initialization state machine 403 “receive[s] a reset signal 410 (RST).” INFINEON-1005, [0098].



INFINEON-1005, Fig. 20 (annotated). A POSITA would have recognized that Grundy discloses a second control input port configured to receive a reset signal

(“reset signal 410 (RST)”). Indeed, the depiction of the input arrow in Fig. 20 is consistent with the common representation of ports in digital circuit diagrams. *See also* INFINEON-1007, p. 591 (definition of “port”). Additionally, a POSITA would have understood or at least found it obvious that the status register is configured to be reset when the reset signal (“reset signal 410 (RST)”) is received at the second control input port. Grundy’s “reset to an initial state” step would include resetting status, *e.g.*, resetting a value stored in a status register. INFINEON-1005, [0098]. It would have been obvious to reset the status register in the Grundy-Kilbuck memory device when the reset signal is received because the status register needs to provide accurate status. Indeed, a status before the reset would no longer be valid after the reset.

O. Claim 15

[15] The flash memory device as claimed in claim 1, further comprising an error correcting code (ECC) configured to be utilized during the page program operation.

139. It is my opinion that the Grundy-Kilbuck combination renders [15] obvious.

140. In regard to a page program (write) command, Grundy discloses the use of error checking values.

FIG. 13 illustrates an exemplary command packet 241 for a write command directed to memory device 155₆. As shown, the command packet 241 is a long-form command packet in which the device selectors specify device ID ‘6’, the command specifier indicates a write

operation, the number of data words to be written is four (WORDS=4) and in which a set of address words specify the starting write address. A data packet 243 containing the four words to be written is transmitted back-to-back with the command packet (i.e., transmitted in the first available transmit interval following transmission of the command packet). Though not shown in FIG. 13, ***error checking values may be included in both the command packet 241 and data packet 243 (e.g., at the end of each) or a single error checking value may be transmitted at the end of the data packet 243 to cover the contents of both the data and command packets.***

INFINEON-1005, [0072]. A POSITA would have found it obvious to include an ECC configured to be utilized during the page program operation to increase reliability of flash memory storage. Not only would an ECC detect errors (i.e., serve as error checking values), an ECC would also correct them to the extent that the errors do not exceed the capabilities of the ECC.

141. Kilbuck further discloses that flash memory “typically utilizes error correction codes.”

A problem with NAND Flash memory is that with its higher density and configuration it requires specialized interfacing and interaction to utilize; many available processors and memory controllers do not directly support NAND Flash memory whereas NOR Flash memory support is more commonly available. Typically this means that a majority of the control and management of a NAND Flash memory device is external from the memory device and in expensive dedicated hardware or a software driver that is executed on a coupled memory controller or processor. In addition, as multiple floating gate memory cell transistors are coupled together in the strings of a NAND Flash memory array, individual errors in the array are more difficult to correct for than in NOR Flash memory. This issue becomes even more of a

problem when multi-level memory cells (MLCs), which store multiple memory states in each cell, are utilized. To compensate for this and to take advantage of the inherent higher array density, ***NAND Flash memory typically utilizes error correction codes (ECC)*** and/or are interfaced to and presented as a mass storage device, such as a magnetic disk. In this manner the errors of a NAND Flash memory device can be addressed by the operating system/host/driver/firmware and/or the file system that the Flash device is formatted with. In addition, wear leveling routines are also typically incorporated in the interface/drivers for NAND Flash memory device(s) to distribute writes and help prevent early wear out of the device(s). However, NAND architecture Flash memory devices are also generally less expensive and of larger capacity than a corresponding NOR architecture Flash memory device, making them desirable for system designers.

INFINEON-1006, [0005]. Kilbuck's "internal memory controller 110 allows ECC codes to be evaluated and generated."

As stated above, the internal memory controller 110 controls operation of the NAND architecture Flash memory device 100 and through the synchronous interface 130 presents the NAND architecture Flash memory device 100 as a conventional synchronous memory (SDRAM, DDR, DDR 2, GDDR, RDRAM, etc.). The internal memory controller 110 also controls the specialized Flash memory interfacing and interaction with the memory array 112, including, but not limited to, accessing the memory array 112, address abstraction, mapping bad blocks, ***generating and evaluating ECCs***, erase block management, wear leveling, block erasure, and writing data and/or files to the memory array 112. This allows the host 102 to interface to the NAND architecture Flash memory device 100 as if it was a conventional synchronous memory device and not require the use of external memory drivers or expensive dedicated external hardware. ***Additionally, the internal memory controller 110 allows ECC codes to be evaluated and generated in a fast and efficient manner via firmware run on the internal memory controller 110 or on dedicated***

internal ECC hardware, offloading the task from the host. This ECC generation in internal firmware/hardware is beneficial, in particular, where the NAND memory array includes multi-level memory cells (MLCs) that store multiple data bits in a single cell, further increasing the memory array 112 density but also increasing the ECC overhead to require the use of dedicated ECC generation in either the memory controller 110 firmware or in specialized ECC hardware.

INFINEON-1006, [0031]. A POSITA would have understood that Kilbuck's ECC is utilized during a page program operation because data written to flash memory during a page program operation would be protected by an ECC that is generated and stored during the operation. Additionally, a POSITA would have found it obvious and would have been motivated to implement in Grundy's memory device an ECC to be utilized during the page program operation to achieve the benefits contemplated by Kilbuck as discussed above such as protecting data written during the page program operation.

P. Claim 16

[16] The flash memory device as claimed in claim 1, wherein the control circuitry is configured to automate the page program operation including an internal verification.

142. It is my opinion that the Grundy-Kilbuck combination renders [16] obvious. A POSITA would have found it obvious to include an internal verification step as part of the page program operation in the Grundy-Kilbuck combination because a data storage system is expected to store data reliably. As I discussed in

Section VII.C, a verification step increases flash memory's reliability by ensuring that the memory stores the correct values in the first place.

143. Kilbuck also discloses control circuitry that is configured to automate the page program operation including an internal verification.

Internal to the NAND architecture Flash memory device, an *internal memory controller 110 directs the internal operation; managing the Flash memory array 112 and updating RAM control registers and non-volatile erase block management registers 114*. The RAM control registers and tables 114 are utilized by the internal memory controller 110 during operation of the Flash memory device 100. The NAND architecture Flash memory array 112 contains a sequence of memory banks or segments 116. Each bank 116 is organized logically into a series of erase blocks (not shown). Memory access addresses are received on the address interface 104 of the NAND architecture Flash memory device 100 and divided into a row and column address portions.

On a write access the row decode circuit 120 selects the row page and column decode circuit 124 selects write sense amplifiers 122. Data values to be written are coupled from the buffer 126 via the internal data bus to the write sense amplifiers 122 selected by the column decode circuit 124 and written to the selected floating gate memory cells (not shown) of the memory array 112. The *written cells are then reselected by the row and column decode circuits 120, 124 and sense amplifiers 122 so that they can be read to verify that the correct values have been programmed into the selected memory cells*.

INFINEON-1006, [0030]; *see also* [0039]. Kilbuck's internal memory controller directs the internal operations of the memory device. In doing so, Kilbuck's controller automates the page program operation because the controller causes

associated functions to be performed by itself. For example, Kilbuck's verification step occurs automatically after writing data to cells. Kilbuck's verification step is internal because it occurs in Kilbuck's memory device. Indeed, Kilbuck's components that perform verification such as "memory controller 110" and "row and column decode circuits 120, 124 and sense amplifiers 122" are internal components of the memory device.

144. In sum, a POSITA would have understood that the Kilbuck's verification is internal, and Kilbuck's page program operation, including its verification step, are performed automatically by Kilbuck's flash memory device. This is also consistent with the '381 Patent which discloses "[a]n on-chip [sic: on-chip] write controller may be used to automate all program and erase functions including pulse repetition, where used, and internal verification and margining of data." INFINEON-1001, 27:25-27.

Q. Claim 17

[17pre] A method of operating a flash memory device, the flash memory device comprising a clock input port, a control input port, at least one common data interface, a control circuitry, a status register, and a plurality of flash memory arrays, each flash memory array comprising a plurality of erasable blocks, each erasable block comprising a plurality of pages, each page comprising a plurality of flash memory cells, the method comprising:

145. Claim 17 is an independent method claim that parallels the apparatus claim recited in claim 1. The limitations of claims 1 and 17 are substantially similar and are therefore rendered obvious by the Grundy-Kilbuck combination for the same reasons as I discussed above for claim 1. Any minor differences in claim language between claims 1 and 17 are addressed below.

146. It is my opinion that the Grundy-Kilbuck combination renders [17pre] obvious. See Section VIII.A, [1pre], [1a], [1b], [1c-i], [1c-ii], [1d], [1e], and [1f]. I note that claim 1 recites “a flash memory comprising a plurality of erasable blocks” whereas [17pre] recites “a plurality of flash memory arrays, each flash memory array comprising a plurality of erasable blocks.” The Grundy-Kilbuck combination further discloses a plurality of flash memory arrays.

147. Grundy describes that its memory devices can include storage arrays of flash memory.

The memory devices 107 each include a signaling interface 111 and a storage array 115, the signaling interface 111 being used to receive and transmit data and commands propagating through the chain of memory devices, and the storage array 115 (which may include multiple storage arrays) being used for data storage. While the memory devices 107 are

generally described below as having ***semiconductor storage arrays 115*** (e.g., including, but not limited to, volatile storage arrays such as static random access memory arrays (static RAM) and dynamic RAM arrays; ***nonvolatile storage arrays*** such as programmable read only memory (PROM), erasable PROM (EPROM), electrically erasable or alterable EPROM (EEPROM), ***flash EEPROM***, thyristor-based memory devices (so-called negative-device-resistance storage devices) and so forth), the storage arrays 115 may alternatively be formed using other media including, without limitation, various forms of magnetic and optical media.

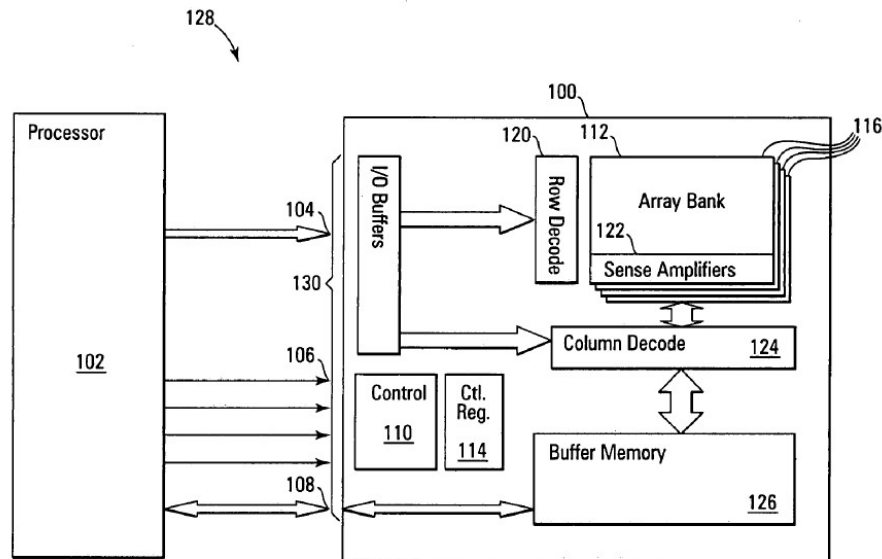
Id., [0035]; *see also* Fig. 3.

148. Kilbuck also describes that its memory devices can include storage arrays of flash memory.

FIG. 1 shows a simplified diagram of a system 128 incorporating a NAND architecture ***Flash memory device 100*** of the present invention coupled to a host 102, which is typically a processing device or memory controller. It is noted that memory device 100 embodiments of the present invention incorporating other ***non-volatile memory arrays 112*** of differing technology and architecture types (including, but not limited to, Polymer Memory, FeRAM, OUM, MRAM, Molecular Memory, and Carbon Nanotube Memory) are also possible and should be apparent to those skilled in the art with the benefit of the present disclosure. The NAND architecture Flash memory device 100 has a synchronous interface 130 that contains an address interface 104, control interface 106, and data interface 108 that are each coupled to the processing device 102 to allow synchronous memory read and write accesses. Internal to the NAND architecture Flash memory device, an internal memory controller 110 directs the internal operation; managing the Flash memory array 112 and updating RAM control registers and non-volatile erase block management registers 114. The RAM control registers and tables 114 are utilized by the internal memory controller 110 during operation of the Flash memory device 100. ***The NAND***

architecture Flash memory array 112 contains a sequence of memory banks or segments 116. Each bank 116 is organized logically into a series of erase blocks (not shown).

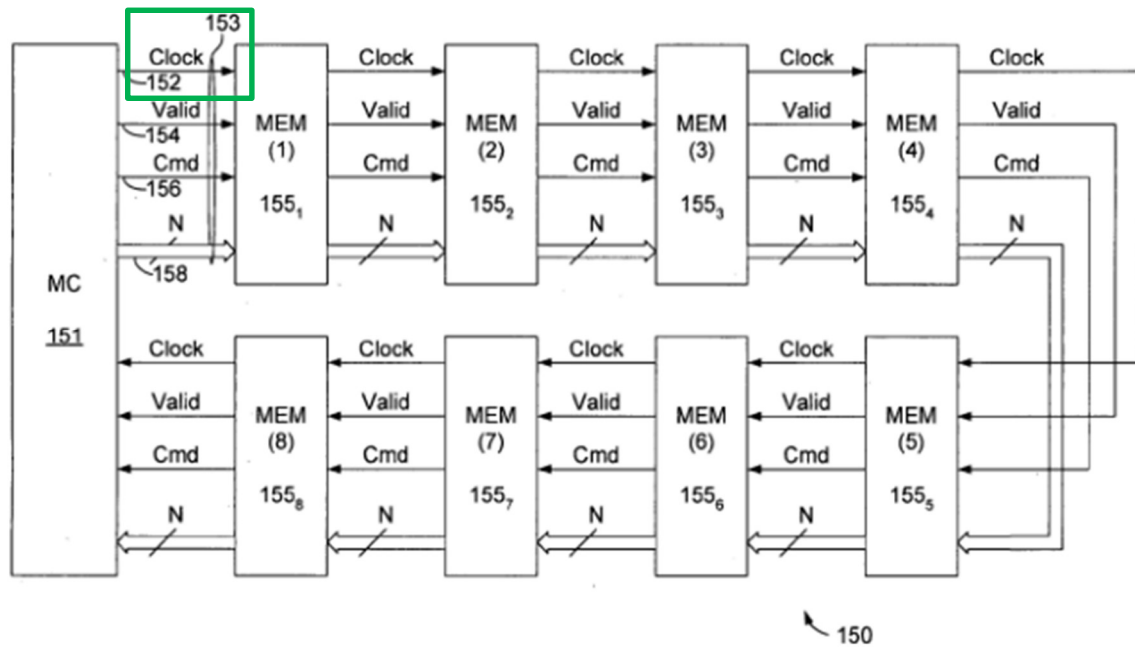
INFINEON-1006, [0030].



INFINEON-1006, Fig. 1.

[17a] providing a clock signal to the clock input port;

149. It is my opinion that the Grundy-Kilbuck combination renders [17a] obvious. *See* Section VIII.A, [1b]. Further, Grundy's memory controller provides the clock signal, *e.g.*, clock signal 172₁ on clock line 152, to the clock input port Grundy's memory device.



INFINEON-1005, Fig. 5A (annotated); *see also* [0043], [0047], Fig. 6. For simplicity, I have added emphasis to show the clock input port of the initial memory device. However, each memory device in Grundy's chain includes a clock input port. Additionally, each of Grundy's memory devices provides a clock signal to the next memory device or to the controller (if it is the last device on the chain).

[17b-i] transferring command data, address data, input data and output data to and from the at least one common data interface,

150. It is my opinion that the Grundy-Kilbuck combination renders [17b-i] obvious. As I have discussed above, Grundy discloses the at least one common data interface. *See* Section VIII.A, [1c-i]. Grundy discloses:

FIG. 5A illustrates a chained memory system 150 according to an embodiment of the invention. The memory system includes a memory controller 151 and eight chained memory devices 155₁-155₈ coupled in a ring by *point-to-point signaling links 153*. Each of the point-to-point

signaling links 153 includes a clock line 152, valid line 154, command line 156 and *a set of N multi-purpose lines 158*. The clock line 152, valid line 154 and command line 156 are used to conduct a clock signal, valid signal and command signal, respectively. The clock signal is used to time receipt of the valid signal, command signal and signals on the multipurpose lines 158.

INFINEON-1005, [0043].

The *multi-purpose lines 158* are used at alternate times to transfer *N-bit wide data values and command values* and are referred to collectively herein as a bus. The term “bus” is used herein to mean a signaling structure for parallel transfer of bits of a multi-bit value. A bus may be a point-to-point link as in FIG. 5A, or a multi-drop structure.

INFINEON-1005, [0044].

151. Grundy’s multi-purpose lines are highlighted in green in the following figure.



FIG. 5A

INFINEON-1005, Fig. 5A (annotated); *see also* Fig. 3. I also note that the signaling paths can be bi-directional. Grundy discloses that “the signaling paths 110 may be bi-directional rather than unidirectional, allowing data and/or control signals to travel through the memory chain in either direction.” INFINEON-1005, [0041].

152. A POSITA would have recognized that Grundy’s memory controller and memory devices transfer command data, address data, input data, and output data to and from the at least one common data interface (e.g., Grundy’s multi-purpose lines which I have highlighted in green in the above-provided figure). As I discussed for [1c-i] and [1e], Grundy’s memory controller transfers command data, address data, and input data to the initial memory device (e.g., 155₁ in Fig. 5A) on the chain. INFINEON-1005, [0043]-[0044]. The memory device can further transfer the data (including output data if responding to a read command) to the next memory device or to the controller (if it is the last device on the chain). Grundy’s memory controller also transfers output data from the last memory device on the chain. Additionally, each memory device transfers data from a previous device on the chain.

[17b-ii] wherein at least one of command data, address data, input data and output data is transferred in synchronization with both rising and falling edges of the clock signal when the flash memory device is in a double data rate configuration;

153. It is my opinion that the Grundy-Kilbuck combination renders [17b-ii] obvious. *See* Section VIII.A, [1c-ii].

[17c] providing a control signal to the control input port, wherein a first transition of the control signal from an inactive state to an active state indicates a beginning of command data being provided at the at least one common data interface;

154. It is my opinion that the Grundy-Kilbuck combination renders [17c] obvious. *See* Section VIII.A, [1d].

[17d] providing a page program command to the control circuitry to perform a page program operation to store the input data on a selected page of the flash memory device;

155. It is my opinion that the Grundy-Kilbuck combination renders [17d] obvious. *See* Section VIII.A, [1e].

[17e] providing a read command to retrieve the output data from the flash memory cells; and

156. It is my opinion that the Grundy-Kilbuck combination renders [17e] obvious. *See* Section VIII.A, [1e].

[17f] providing a read status command to access the status register and determine a status of the flash memory device.

157. It is my opinion that the Grundy-Kilbuck combination renders [17f] obvious. *See* Sections VIII.A, [1f], [17pre].

158. As discussed for [1f], Grundy-Kilbuck provides a status register. Grundy further describes a read status command to access the status register and determine a status of the flash memory device.

- “Commands sent to target memory devices may include data read and write commands, erase commands, parameter query commands, status

requests and may command or request any other action indicated by the device parameter information to be within the capability of the target memory devices” (INFINEON-1005, [0033]),

- “the memory devices monitor commands that propagate through the memory chain to determine if the commands require responsive action (e.g., data read, write and erase actions, parameter query response, *status response*, etc.)” ([0032]).

A POSITA would have recognized Grundy’s status request as a read status command.

159. Kilbuck also describes a read status command to access the status register and determine a status of the flash memory device.

In one embodiment of the present invention, a special purpose “ready/busy” external connection pin or internal *status register*/flag of the NAND architecture Flash memory device 100 *may be monitored* to indicate the status of the memory. This allows the memory device 100 to signal that it is busy with a read or write access (due to read or write latency, etc.), and causes the host 102 to wait, halting data transfers when active or set.

INFINEON-1006, [0033]; *see also* [0030], [0034], [0039]. A POSITA would have recognized that Kilbuck provides a read status command to access Kilbuck’s internal status register and determine a status of the flash memory device.

R. Claim 18

[18] The method of claim 17, further comprising receiving a result of the page program operation after the page program operation is completed.

160. It is my opinion that the Grundy-Kilbuck combination renders [18] obvious. See Sections VIII.Q, VIII.C. As I discussed for claim 3, the Grundy-Kilbuck combination provides a result of the page program operation after the page program operation is completed.

161. Grundy describes status requests and responses to/from a memory device:

- “[c]ommands sent to target memory devices may include ... status requests” (INFINEON-1005, [0033]),
- “the memory devices monitor commands that propagate through the memory chain to determine if the commands require responsive action (e.g., data read, write and erase actions, parameter query response, status response, etc.)” ([0032]),
- “read, write and erase commands issued in short-form command packets 203 are interpreted by the target memory devices as status read, configuration write (e.g., to a device configuration register) and configuration and/or status reset commands, respectively” ([0068]), and
- “memory controller 101 also includes a host interface 103 (HI) to receive memory access requests, addresses and write data from, and to transmit read data and status information to, one or more host devices (not shown) such as a processor, application-specific integrated circuit

(ASIC), video controller, direct memory access controller and/or other device capable of issuing memory access requests” ([0036]).

A POSITA would have recognized that a response to a status request regarding a page program operation would have been received by Grundy’s memory controller and/or host interface.

162. Kilbuck discloses a status register which may be monitored to indicate the status of the memory.

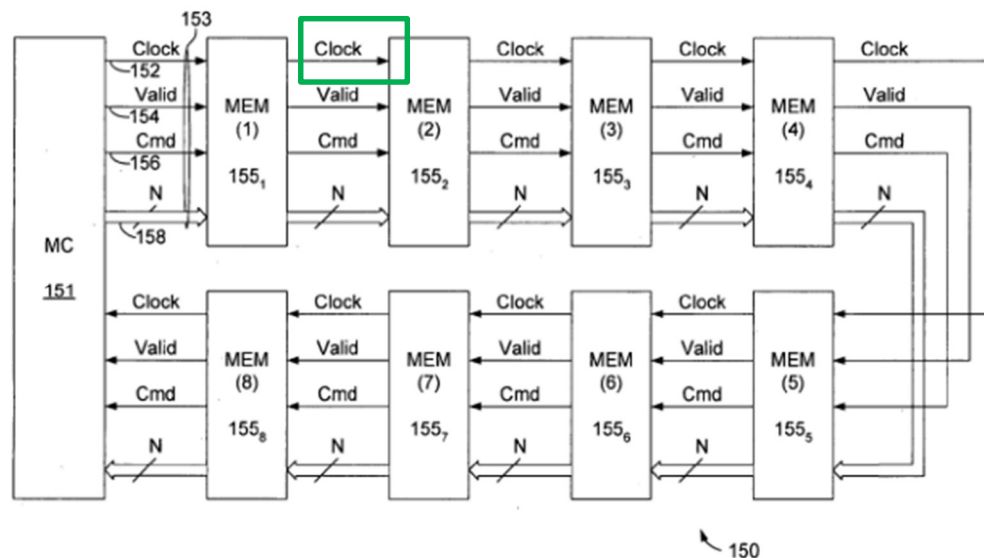
In one embodiment of the present invention, a special purpose “ready/busy” external connection pin or internal *status register*/flag of the NAND architecture Flash memory device 100 *may be monitored* to indicate the status of the memory. This allows the memory device 100 to signal that it is busy with a read or write access (due to read or write latency, etc.), and causes the host 102 to wait, halting data transfers when active or set.

INFINEON-1006, [0033]; *see also* [0030], [0034], [0039]. As I discussed above, a POSITA would have understood that Kilbuck’s status register as included in the Grundy-Kilbuck combination provides a result of the page program operation after the page program operation is completed. Further, a POSITA would have recognized that Kilbuck’s monitoring of a status register would include receiving a value of the status register. The value would be a result of an operation stored in the register.

S. Claim 19

[19] The method of claim 17, further comprising, receiving a clock output signal from a clock output port of the flash memory device in synchronization with the output data when the flash memory device is in the double data rate configuration.

163. It is my opinion that the Grundy-Kilbuck combination renders [19] obvious. *See* Sections VIII.Q, VIII.D. As I discussed for claim 4, each of Grundy's memory devices outputs a clock output signal from a clock output port. Further, a subsequent memory device or controller on Grundy's chain receives the clock output signal. INFINEON-1005, [0043]-[0044]. This is shown in Fig. 5A of Grundy.



INFINEON-1005, Fig. 5A (annotated). For simplicity, I have added emphasis to show the clock input port of Grundy's second memory device on the chain which receives a clock output signal from Grundy's first memory device. However, each memory device in Grundy's chain includes a clock input port.

T. Claim 20

[20] The method of claim 17, wherein a second transition of the control signal from the active state to the inactive state indicates an end of the transfer of the at least one of command data, address data, input data and output data.

164. It is my opinion that the Grundy-Kilbuck combination renders [20] obvious. *See* Sections VIII.Q, VIII.F.

U. Claim 21

[21] The method of claim 17, wherein the providing a read command occurs during a time period that overlaps with a time period for the page program operation.

165. It is my opinion that the Grundy-Kilbuck combination renders [21] obvious. *See* Sections VIII.Q, VIII.M.

IX. CONCLUSION

166. For all the reasons I have noted in the foregoing paragraphs, claims 1-21 of the '381 Patent are obvious in view of the references discussed above.

167. I currently hold the opinions set expressed in this declaration. But my analysis may continue, and I may acquire additional information and/or attain supplemental insights that may result in added observations.