



US 20060098520A1

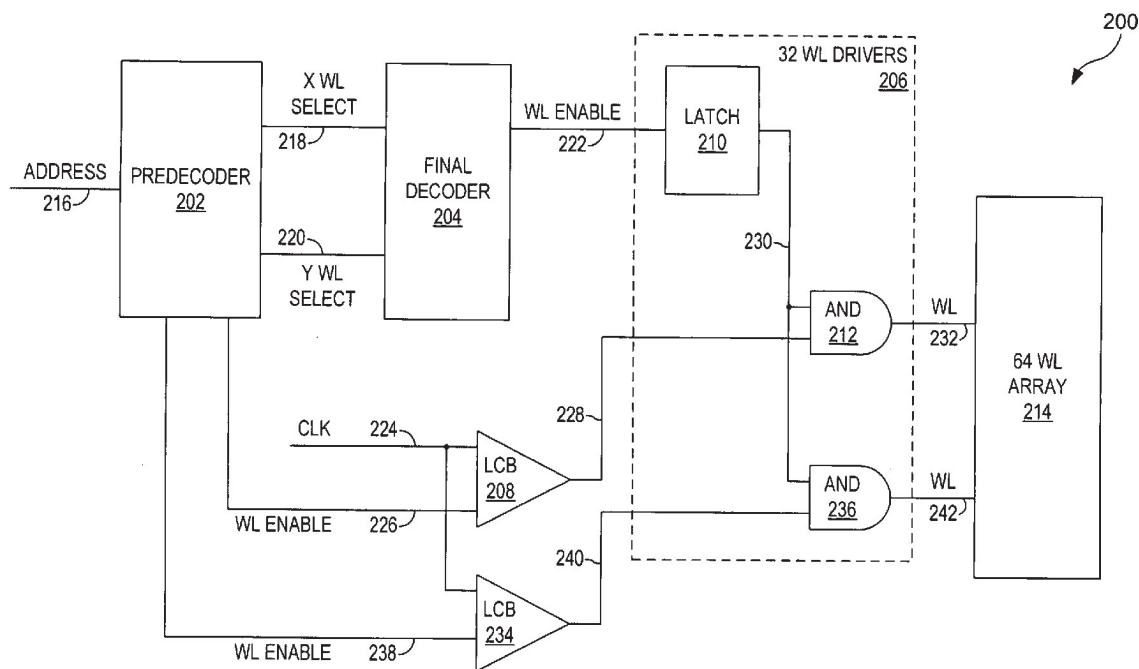
(19) **United States**(12) **Patent Application Publication** (10) **Pub. No.: US 2006/0098520 A1****Asano et al.**(43) **Pub. Date: May 11, 2006**(54) **APPARATUS AND METHOD OF WORD LINE
DECODING FOR DEEP PIPELINED
MEMORY**(21) Appl. No.: **10/982,109**(22) Filed: **Nov. 5, 2004**(75) Inventors: **Toru Asano**, Omihachinan-shi (JP);
Sang Hoo Dhong, Austin, TX (US);
Takaaki Nakazato, Austin, TX (US);
Osamu Takahashi, Round Rock, TX
(US)**Publication Classification**(51) **Int. Cl.****G11C 8/00** (2006.01)**G11C 7/10** (2006.01)(52) **U.S. Cl.** **365/230.06; 365/233; 365/189.05**

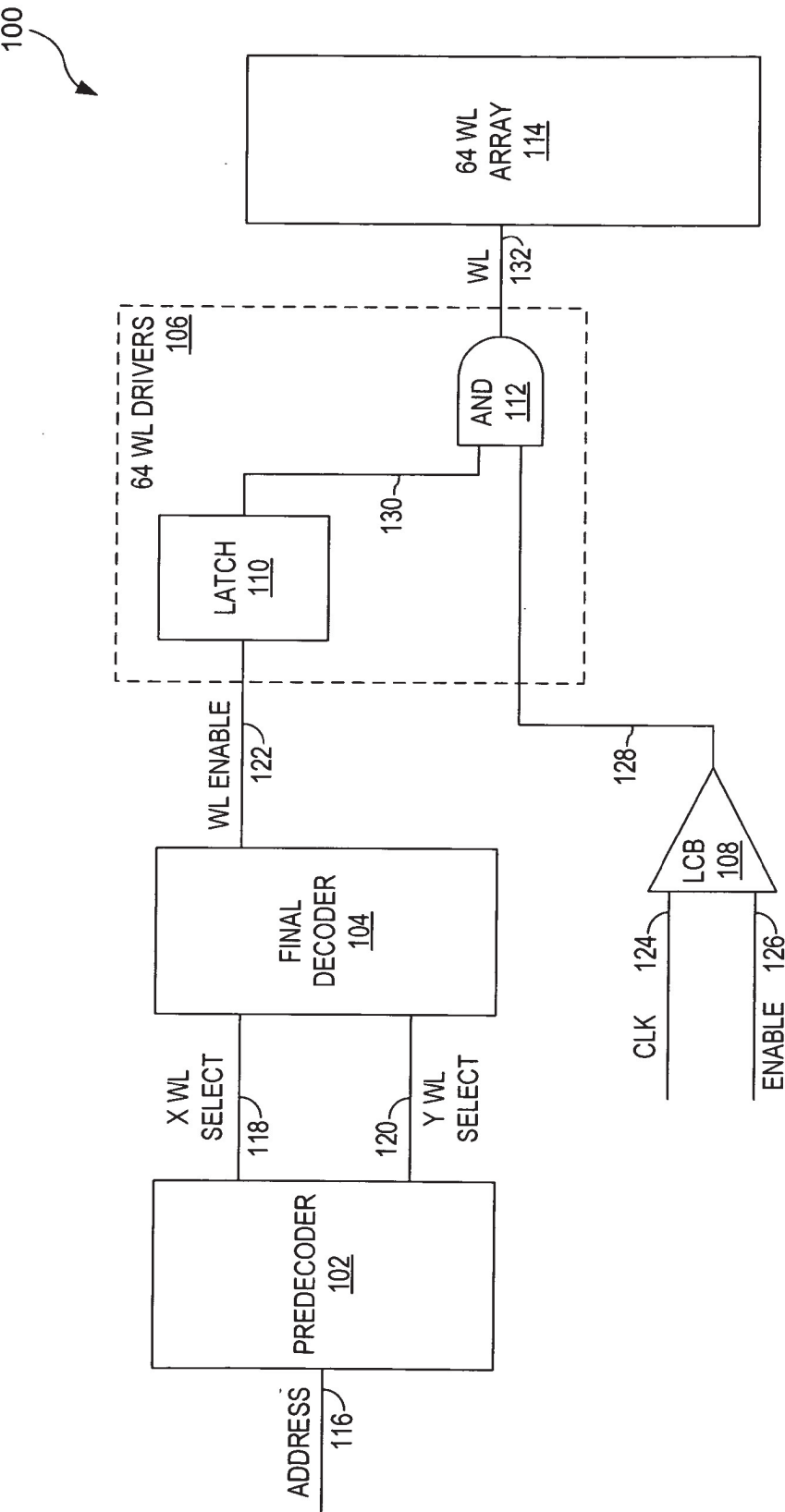
(57)

ABSTRACT

A method, an apparatus, and a computer program are provided to reduce the number of required latches in a deep pipeline wordline (WL) decoder. Traditionally, a signal local clock buffer (LCB) has been responsible for providing a driving signal to a WL driver. However, with this configuration, a large number of latches are utilized. To reduce this latch usage, a number of LCBs are employed, such that one latch can enable an increased number of WLs. Hence, the overall area occupied by latches is reduced and power consumption is reduced.

Correspondence Address:

IBM CORPORATION (CS)**C/O CARR LLP****670 FOUNDERS SQUARE****900 JACKSON STREET****DALLAS, TX 75202 (US)**(73) Assignees: **International Business Machines Corporation**, Armonk, NY; **Toshiba America Electronic Components, Inc.**, Irvine, CA; **Kabushiki Kaisha Toshiba**, Tokyo (JP)



PRIOR ART

FIG. 1

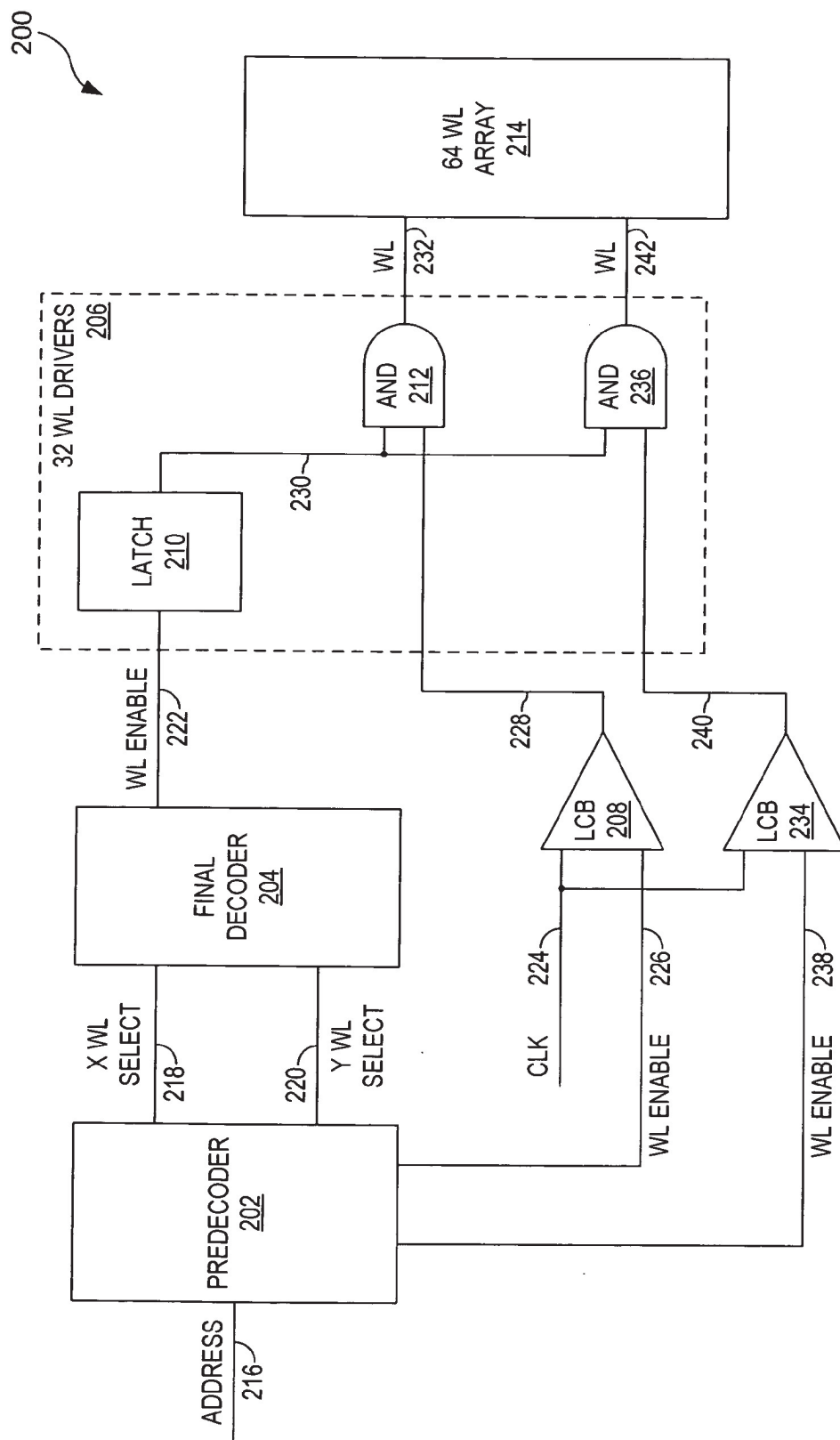


FIG. 2

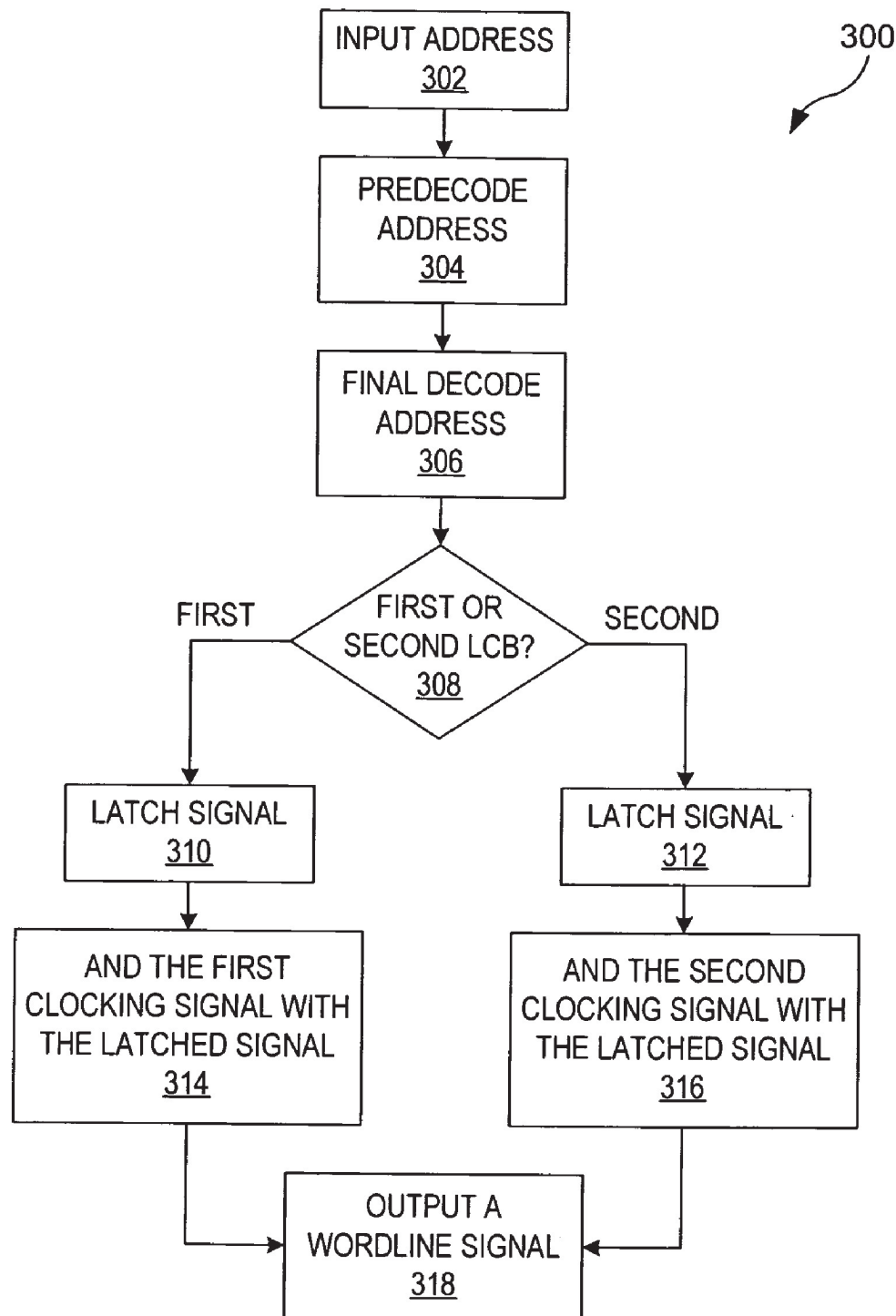


FIG. 3

APPARATUS AND METHOD OF WORD LINE DECODING FOR DEEP PIPELINED MEMORY

FIELD OF THE INVENTION

[0001] The present invention relates generally to memory arrays, and more particularly, to wordline decoding for memory arrays.

DESCRIPTION OF THE RELATED ART

[0002] In conventional memory arrays, the pipeline is becoming increasingly deep. Additionally, the performance of memory arrays is becoming increasingly important to assist in high speed computations and computer performance. However, in deep pipelined high performance memory, a wordline driver has a cycle bound that starts the access cycle. To utilize a cycle bound to initiate the access cycle, wordline drivers typically employ latches. Each latch employed then consumes power.

[0003] Referring to **FIG. 1** of the drawings, the reference numeral **100** generally designates conventional memory. The memory **100** comprises a predecoder **102**, a final decoder **104**, 64 wordline (WL) drivers **106**, a local clock buffer (LCB) **108**, and a 64 wordline array **114**.

[0004] To begin the access cycle for the memory **100**, an address is first received at the predecoder **102** through a first communication channel **116**. Typically, the address is 6 bits long, and from those 6 bits, the predecoder derives two distinct wordline select signals, an X wordline select signal and a Y wordline select signal. The X wordline select signal is 8 bits long and is output to the final decoder **104** through a second communication channel **118**. The Y wordline select signal is output to the final decoder **104** through a third communication channel **120** and is 8 bits long.

[0005] Once the X wordline select signal and the Y wordline select signal have been transmitted to the final decoder **104**, the final decoder **104** determines which of the 64 wordline drivers **106** are to be enabled. The wordline enable signals are communicated to the wordline drivers **106** through a fourth communication channel **122**. The LCB **108** provides a clocking signal to the wordline drivers **106** through a fifth communication channel **128**. The clocking signal from the LCB **108** is usually based on two inputs, a clock input and an enable input, which are provided to the LCB **108** through a sixth communication channel **124** and a seventh communication channel **126**, respectively.

[0006] Each of the wordlines within the array **114** has an associated driver. Each driver comprises a latch and an AND gate, so that for the 64 wordline array **114**, there are 64 drivers. For the sake of illustration, a single latch **110** and an AND gate **112** are depicted. To function, the latch **110** receives a wordline enable signal through the fourth communication channel **122**, where the signal is latched. The latch **110** then outputs a signal to the AND gate **112** through an eighth communication channel **130**. The AND gate **112** also received the clocking signal from the LCB **108** through the fifth communication channel **128**. The AND gate **112** then outputs a wordline signal to a wordline within the 64 wordline array **114** through a ninth communication channel **132**.

[0007] These conventional memories, such as the memory **100**, can, however, have several drawbacks. For example,

clock load for the wordline timing signal can be high. Because of the large number of latches, there is a substantial risk of soft errors, and more latches require more clock power. Therefore, there is a need for a method and/or apparatus for storing data that addresses at least some of the problems associated with conventional memories.

SUMMARY OF THE INVENTION

[0008] The present invention provides a wordline (WL) driver method, apparatus, and computer program for reducing required latches in a WL decode path for deep pipelined memory and for use in a WL decode scheme. As with many systems, a plurality of timing signals are generated. A WL driver then receives a WL enable data signal. Once received, a plurality of WL signals are generated based on the plurality of timing signals and the WL enable data signal.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following descriptions taken in conjunction with the accompanying drawings, in which:

[0010] **FIG. 1** is a block diagram depicting conventional memory;

[0011] **FIG. 2** is a block diagram depicting modified memory; and

[0012] **FIG. 3** is a flow chart depicting the operation of the modified memory.

DETAILED DESCRIPTION

[0013] In the following discussion, numerous specific details are set forth to provide a thorough understanding of the present invention. However, those skilled in the art will appreciate that the present invention may be practiced without such specific details. In other instances, well-known elements have been illustrated in schematic or block diagram form in order not to obscure the present invention in unnecessary detail. Additionally, for the most part, details concerning network communications, electro-magnetic signaling techniques, and the like, have been omitted inasmuch as such details are not considered necessary to obtain a complete understanding of the present invention, and are considered to be within the understanding of persons of ordinary skill in the relevant art.

[0014] It is further noted that, unless indicated otherwise, all functions described herein may be performed in either hardware or software, or some combinations thereof. In a preferred embodiment, however, the functions are performed by a processor such as a computer or an electronic data processor in accordance with code such as computer program code, software, and/or integrated circuits that are coded to perform such functions, unless indicated otherwise.

[0015] Referring to **FIGS. 2 and 3** of the drawings, the reference numerals **200** and **300** generally designate modified memory and the operation of the modified memory. The memory **200** comprises a predecoder **202**, a final decoder **204**, 32 wordline drivers **206**, a first LCB **208**, a second LCB **234**, and a 64 wordline array **214**.

[0016] To begin the access cycle for the memory **200**, an address is first received in step **302** at the predecoder **202**

through a first communication channel 216. Typically, the address is 6 bits long, and from those 6 bits, the predecoder derives a wordline enable signal and two wordline select signals in step 304, an X wordline select signal and a Y wordline select signal. The X wordline select signal is 8 bits long and is output to the final decoder 204 through a second communication channel 218. The Y wordline select signal is output to the final decoder 204 through a third communication channel 220 and is 4 bits long.

[0017] Once the X wordline select signal and the Y wordline select signal have been transmitted to the final decoder 204, the final decoder 204 in step 306 determines which of the 32 wordline drivers 206 are to be enabled. The “true final decode,” though, is done at wordline drivers 206 by enabling and selectively activating clock signals. The wordline enable signals are communicated to the wordline drivers 206 through a fourth communication channel 222. The first LCB 208 and the second LCB 234 also provide clocking signals to the wordline drivers 206 through a fifth communication channel 228 and a sixth communication 240.

[0018] The clocking signal from each of the LCBs 208 and 234 are based on two inputs, a clock input and a select signal. Each of the LCBs 208 and 234 receive a clocking signal through a seventh communication channel 224, and the predecoder 202 generates additional selection signals for the LCBs 208 and 234 in step 308. A selection signal for the first LCB 208 and for the second LCB 234 are provided by the predecoder 202 through an eighth communication channel 226 and a ninth communication channel 238, respectively. By providing selection signals to the LCBs, the last decoding can be delayed until the wordline driver stage. Also, AND gates can be replaced by NAND gates, NOR gates, or OR gates depending upon the circuit type which receives the wordlines.

[0019] The significance of the late last decoding to the wordline driver stage is that the number of latches can be reduced. Within the modified memory 200, every two of the wordlines within the array 214 has an associated driver. Each driver comprises a latch and two AND gates, so that for the 64 wordline array 214, there are 32 drivers. For the sake of illustration, a single latch 210, first AND gate 212, and a second AND gate 236 are depicted. To function, the latch 210 receives a wordline enable signal through the fourth communication channel 222, where the signal is latched in step 310 and 312. The latch 210 then outputs a signal to the first AND gate 212 and the second AND gate 236 through a tenth communication channel 230. The first AND gate 212 receives a clocking signal from the first LCB 208 through the fifth communication channel 228, while the second AND gate 236 receives a clocking signal from the second LCB 234 through the sixth communication channel 240. Depending on the most significant bit of the address signal that is input into the predecoder 202, either the first AND gate 212 or the second AND gate 236 is selected, wherein the clocking signal is ANDed with the output of the latch 210 in steps 314 and 316. One of the respective AND gates 212 and 236 can then output a wordline signal in step 318 to a wordline within the 64 wordline array 214 through an eleventh communication channel 232 or a twelfth communication channel 242, respectively.

[0020] By having the late last decoding, area and power consumption can be reduced. Because each of the LCBs

only provide one-half the power, the drive ability of the LCBs are reduced. The impact, though, of the reduction of drive ability is negated by the fact that the number of LCBs is doubled. However, the area of the final decoder can be reduced by one-half and the number of latches can be reduced by one-half. The reduction of the number of latches, therefore, reduces power consumption and area. And, it also lowers the risk of soft errors.

[0021] Additionally, for the purposes of illustration, 1 bit has been utilized for LCB selections. It is possible to have 2 or more LCB selections up to N bits. In each case, there will be 2^N LCBs each with a reduced load of 2^{-N} . Also, the number of latches can be reduced 2^{-N} , and the area of the final decoder can be reduced by 2^{-N} .

[0022] It is understood that the present invention can take many forms and embodiments. Accordingly, several variations may be made in the foregoing without departing from the spirit or the scope of the invention. The capabilities outlined herein allow for the possibility of a variety of programming models. This disclosure should not be read as preferring any particular programming model, but is instead directed to the underlying mechanisms on which these programming models can be built.

[0023] Having thus described the present invention by reference to certain of its preferred embodiments, it is noted that the embodiments disclosed are illustrative rather than limiting in nature and that a wide range of variations, modifications, changes, and substitutions are contemplated in the foregoing disclosure and, in some instances, some features of the present invention may be employed without a corresponding use of the other features. Many such variations and modifications may be considered desirable by those skilled in the art based upon a review of the foregoing description of preferred embodiments. Accordingly, it is appropriate that the appended claims be construed broadly and in a manner consistent with the scope of the invention.

1. A wordline (WL) driver method for reducing required latches in a WL decode path for deep pipelined memory and for use in a WL decode scheme, comprising:

generating a plurality of timing signals;

receiving in a WL driver a WL enable data signal; and

generating a plurality of WL signals from the plurality of timing signals and from the WL enable data signal.

2. The method of claim 1, wherein the method further comprises further comprises generating at least one local clock buffer signal based on at least one WL signal and at least one timing signal.

3. The method of claim 1, wherein the method further comprises alternatively enabling a plurality of local clock buffers that propagate the plurality of timing signals.

4. The method of claim 3, wherein the step of enabling further comprises enabling a fraction of a total number of WLs.

5. The method of claim 1, wherein the step of generating the plurality of WL signals further comprises logically combining a local clock buffer signal with a latch signal based on the WL enable signal.

6. The method of claim 5, wherein the step of logically combining further comprises ANDing the local clock buffer signal with the latch signal.

7. An apparatus for reducing required latches in a WL decode path for deep pipelined memory in a WL decode scheme, comprising comprising:

a plurality of local clock buffers (LCBs), wherein each LCB of the plurality of the plurality of LCBs is at least configured to propagate clocking signals if enabled; and

a plurality of logic gates within a WL driver, wherein each logic gate of the plurality of logic gates are at least configured to propagate a WL signal from a latch when a signal from at least one LCB is received.

8. The apparatus of claim 7, wherein each LCB of the plurality of LCBs is configured to alternatively receive an enable signal.

9. The apparatus of claim 7, wherein the latch is at least configured to receive an WL enable signal.

10. The apparatus of claim 7, wherein the plurality of logic gates further comprise a plurality of AND gates.

11. The apparatus of claim 7, wherein the only one LCB of the plurality of LCBs are enabled at one time.

12. The apparatus of claim 7, wherein more than one LCB of the plurality of LCBs are enabled at one time.

13. A computer program product for reducing required latches in a WL decode path for deep pipelined memory and for use in a WL decode scheme, the computer program product having a medium with a computer program embodied thereon, the computer program comprising:

computer code for generating a plurality of timing signals;

computer code for receiving in a WL driver a WL enable data signal; and

computer code for generating a plurality of WL signals from the plurality of timing signals and from the WL enable data signal.

14. The computer program product of claim 13, wherein the computer program product further comprises further comprises computer code for generating at least one local clock buffer signal based on at least one WL signal and at least one timing signal.

15. The computer program product of claim 13, wherein the computer program product further comprises computer code for alternatively enabling a plurality of local clock buffers that propagate the plurality of timing signals.

16. The computer program product of claim 15, wherein the computer code for enabling further comprises computer code for enabling a fraction of a total number of WLs.

17. The computer program product of claim 13, wherein the computer code for generating the plurality of WL signals further comprises computer code for logically combining a local clock buffer signal with a latch signal based on the WL enable signal.

18. The computer program product of claim 17, wherein the computer code for logically combining further comprises computer code for ANDing the local clock buffer signal with the latch signal.

* * * * *