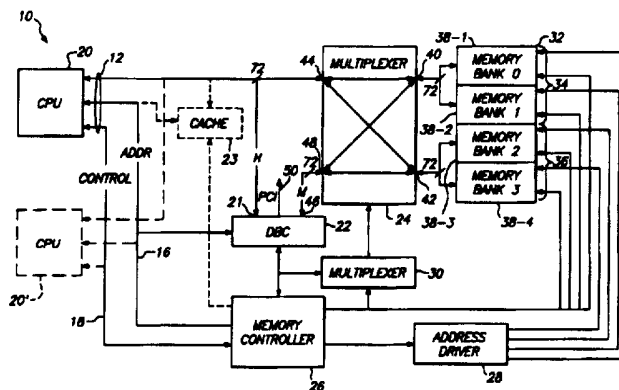




## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>6</sup> : <b>G06F 12/00</b>	<b>A1</b>	(11) International Publication Number: <b>WO 96/30837</b> (43) International Publication Date: 3 October 1996 (03.10.96)
(21) International Application Number: PCT/US96/04352 (22) International Filing Date: 28 March 1996 (28.03.96) (30) Priority Data: 08/414,118                      31 March 1995 (31.03.95)                      US (71) Applicant: SHABLAMM! COMPUTER INC. [US/US]; Suite 201, 21040 Homestead Road, Cupertino, CA 95014 (US). (72) Inventor: NORTH, Tom; 27890 Via Feliz, Los Altos, CA 94022 (US). (74) Agents: WELLER, Edward, B. et al.; Fenwick & West L.L.P., Suite 600, Two Palo Alto Square, Palo Alto, CA 94306 (US).		(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).  <b>Published</b> <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

(54) Title: UPGRADABLE, CPU PORTABLE MULTI-PROCESSOR CROSSBAR INTERLEAVED COMPUTER



## (57) Abstract

A computer (10) includes at least one central processing unit (20). A system bus (12) is coupled to the at least one CPU. A data bus controller (22) is coupled to the system bus and communicates with the local bus. A multiplexer (24) communicates signals between a first portion of a plurality of memories (32) and the system bus responsive to a first control signal, communicates signals between the first portion of a plurality of memories and data bus controller responsive to a second control signal, communicates signals between a second portion of the plurality of memories and the system bus responsive to a third control signal and communicates signals between the second portion of the plurality of memories and data bus controller responsive to a fourth control signal. A multiplexer controller (30) is coupled to the bus and to the plurality of memories for providing addressing signals to the plurality of memories.

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgyzstan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	KZ	Kazakhstan	SG	Singapore
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovakia
CM	Cameroon	LR	Liberia	SN	Senegal
CN	China	LT	Lithuania	SZ	Swaziland
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	LV	Latvia	TG	Togo
DE	Germany	MC	Monaco	TJ	Tajikistan
DK	Denmark	MD	Republic of Moldova	TT	Trinidad and Tobago
EE	Estonia	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	UG	Uganda
FI	Finland	MN	Mongolia	US	United States of America
FR	France	MR	Mauritania	UZ	Uzbekistan
GA	Gabon			VN	Viet Nam

Upgradeable. CPU Portable Multi-processor Crossbar Interleaved ComputerCross Reference Related to Application

This application is a continuation-in part of applicant's copending  
5 application Serial No. 08/215,144, filed March 15, 1994, which application is pending,  
the subject matter of which is incorporated herein by reference.

Field Of The Invention

This invention relates to multiple processor computers and more  
10 particularly to crossbar interleaving between processors, memory, and local bus.

Background of the Invention

Workstation and other computers suffer from performance limitations due  
to either bottlenecks which restrict their bandwidth, or the need for expensive cache  
15 RAMs and cache coherency controllers. Users of these computers frequently desire to  
upgrade the processor of their computers while retaining as much as possible the  
hardware and the software of the computer. The computer may be upgraded by  
physically removing the processor and cache modules, which are expensive modules,  
and replacing them with another cache and processor module. If the replacement  
20 processor module has a different operation system, instruction set, or other operational  
characteristics than the original processor module, the memory controller of the  
computer may need to be modified or replaced. This additional replacement adds cost  
to the upgrade.

Both shared and unichache designs bottleneck at the cache or data bus with  
25 each processor fighting the other for the cache or data bus resource. Conventional  
multi-cache designs require separate caches for each processor and cache control logic  
to maintain the cache coherency between each processor's primary and secondary  
cache, main memory, and other processors. This requirement is expensive. Typical  
architectures are also limited in their upgradeability because the tight coupling of the  
30 processor to its cache control logic. The upgrade of a computer often requires replacing  
a module with both the processor and the cache which is expensive because it has  
many expensive components.

Summary Of The Invention

35 In accordance with the present invention, a computer includes at least one  
central processing unit. A system bus is coupled to the at least one central processing  
unit. A data bus controller is coupled to the system bus and communicates with the  
local bus.

A multiplexer has a first bi-directional terminal coupled to a first portion of a plurality of memories, has a second bi-directional terminal coupled to a second portion of the plurality of memories, has a third bi-directional terminal coupled to the system bus, has a fourth bi-directional terminal coupled to the data bus controller, and  
5 has a first input for receiving first, second, third, and fourth control signals. The multiplexer communicates signals between the first and third terminals responsive to the first control signal, communicates signals between the first and fourth terminals responsive to the second control signal, communicates signals between the second and third terminals responsive to the third control signal, and communicates signals  
10 between the second and fourth terminals responsive to the fourth control signal. The multiplexer has a negligible delay for communicating signals between its terminals.

A multiplexer controller is coupled to the first input of the multiplexer and provides the first, second, third, and fourth control signals. A memory controller is coupled to the bus and to the plurality of memories for providing addressing signals to  
15 the plurality of memories.

Also in accordance with the present invention, a computer system includes a first bus having a data bus, having a control bus, and having an address bus. A plurality of memories is coupled to the data bus. A processor is coupled to the first bus, and has a central processing unit of a particular configuration and has a memory  
20 for storing an initialization program representative of the CPU of the particular configuration. A programmable memory controller is coupled to the address bus, to the control bus, and to the plurality of memories for providing addressing signals to the plurality of memories and for receiving the initialization program.

The computer system may further include a data bus controller for  
25 communicating with a second bus. A multiplexer has a first bi-directional terminal coupled to a first portion of the plurality of memories, has a second bi-directional terminal coupled to a second portion of the plurality of memories, has a third bi-directional terminal coupled to the bus, has a fourth bi-directional terminal coupled to the data bus controller, and has a first input for receiving first, second, third, and fourth  
30 control signals. The multiplexer communicates signals between the first and third terminals responsive to the first control signal, communicates signals between the first and fourth terminals responsive to the second control signal, communicates signals between the second and third terminals responsive to the third control signal, and communicates signals between the second and fourth terminals responsive to the  
35 fourth control signal. A multiplexer controller coupled to the first input of the multiplexer provides the first, second, third, and fourth control signals to the multiplexer.

Brief Description of the Drawings

FIG. 1 is a block diagram illustrating a multi-processor multiplexer interleaved computer in accordance with a first embodiment of the present invention.

5 FIG. 2 is a block diagram illustrating a multi-processor multiplexer interleaved computer in accordance with a second embodiment of the present invention.

FIG. 3 is a block diagram illustrating a multi-processor multiplexer interleaved computer in accordance with a third embodiment of the present invention.

10 FIG. 4 is a block diagram illustrating a multi-processor multiplexer interleaved computer in accordance with a fourth embodiment of the present invention.

FIG. 5 is a flowchart illustrating the initialization of the computer system of FIG. 4

15 FIG. 6 is a block diagram illustrating a multi-processor multiplexer interleaved computer in accordance with a fifth embodiment of the present invention.

FIG. 7 is a block diagram illustrating a multi-processor multiplexer interleaved computer in accordance with a sixth embodiment of the present invention.

20 FIG. 8 is a schematic diagram illustrating the interconnections of the multiplexer of FIG. 1.

FIG. 9 is a schematic diagram illustrating the interconnections of the multiplexer of FIG. 4.

FIG. 10 is a block diagram illustrating a conventional interconnection of a single in line module.

25 FIG. 11 is a block diagram illustrating the memory of the computer of FIG. 4.

FIGs 12a -12h are timing diagrams illustrating a read and a page hit with prefetch.

30 FIGs. 13a-13h are timing diagrams illustrating a read with page hit column miss.

FIGs. 14a-14h are timing diagrams illustrating a read with page miss.

FIGs. 15a-15j are timing diagrams illustrating an interleaved posted write and an interleaved write.

35 FIGs. 16a, 16b ,and 16c are schematic diagrams illustrating a conventional non-interleaved memory, a page interleaved architecture memory, and a cache line size architecture memory, respectively.

FIG. 17 is a schematic diagram illustrating the interconnections of the multiplexer of FIG. 6.

FIG. 18 is a flowchart illustrating the operation of the interleaving of the memory of FIG. 11.

### Description of the Preferred Embodiments

5

Referring to FIG. 1, there is shown a block diagram illustrating a computer system 10 according to a first embodiment of the present invention. The computer system 10 includes a system bus 12 having a system data bus 14, a system address bus 16, and a system control bus 18. The system data bus 14 interconnects a central processing unit (CPU) 20, a host terminal 21 of a data bus controller (DBC) 22, a cache 23, and a multiplexer 24 for communicating data. Each of these elements is well known in the art. The CPU 20 may be, for example, a 80486 processor or a Pentium processor that are each commercially available from Intel Corporation (Santa Clara, California). Additional CPUs 20' may be coupled to the system bus 12. For clarity, only one additional CPU 20' is shown and is indicated by dashed lines. The data bus controller 22 may be, for example, a model KS82C533 manufactured by Intel Corporation (Santa Clara, California). The multiplexer 24 may be, for example, a model QS3B481 manufactured by Quality Semiconductor (Santa Clara, California). The cache 23 may be, for example, a model COAST cache module manufactured by Cypress Semiconductor (Santa Clara, California)).

The system address bus 16 interconnects the CPU 20, the data bus controller 22, the cache 23, and a memory controller 26 for communicating address signals. The memory controller 26 may be, for example, a burst EDO chipset, such as model KS82C531 manufactured by Samsung Electronics (San Jose, California).

The system control bus 18 interconnects the CPU 20 and the memory controller 26 for communicating control signals. The memory controller 26 provides control signals to a row address driver 28 for supplying the row address to a main memory 32 during an address cycle responsive to such signals. The row address driver 28 may include, for example, three model 163344 address drivers manufactured by Integrated Device Technology (Santa Clara, California) or six model 163244 address drivers also manufactured by Integrated Device Technology (Santa Clara, California). A multiplexer controller 30, the data bus controller 22, and the memory controller 26 are coupled for communicating control and status signals. The memory controller 26 provides control signals, such as write enable, row enable, column enable, and output enable, to the main memory 32 and to the multiplexer controller 32, which provides control signals to the multiplexer 24 for selecting the communication connections therein. The memory controller 26 also provides control and addressing signals to the cache 23.

The main memory 32 is organized as a first memory bank 34 and a second memory bank 36. Each memory bank 34, 36 is preferably conventional burst extended data out (EDO) memory. Alternatively, each memory bank 34, 36 may be, for example, a conventional dynamic random-access memory (DRAM) or may be enhanced DRAMs (EDRAM) memory chips from Ramtron, Inc. (Colorado Springs, Colorado). The memory banks 34, 36 may be interleaved, such as described later in conjunction with FIG. 11, to provide substantially twice the performance normally provided by the memory devices.

Each memory bank 34, 36 may include a pair of single in line memory modules 38-1, 38-2 and 38-3, 38-4, respectively. Each single in line memory module 38 also has two memory subbanks (not shown) and is designed to allow interleaving between them for additional performance. The configuration and operation of the single in line memory module is described in U.S. patent application S/N 08/215,144, filed March 15, 1994, the subject matter of which is incorporated herein by reference. The memory 32 preferably has sufficient addressing to provide up to 512 megabytes of memory for a memory having 64 megabit memory chips.

The multiplexer 24 selectively couples the memory 32 to the system data bus 14 and to the data bus controller 22. The memory banks 34, 36 are coupled to respective bi-directional memory terminals 40, 42 of the multiplexer 24. The system data bus 14 is coupled to a bi-directional data terminal 44 of the multiplexer 24. A memory terminal 46 of the data bus controller 22 is coupled to a bi-directional data terminal 48 of the multiplexer 24. The multiplexer 24 selectively couples the first memory bank 34 to either the system data bus 14 for CPU reads or to the data bus controller 22 for CPU writes to memory which are posted or input/output reads or writes to memory responsive to control signals from the memory controller 26. The multiplexer 24 selectively couples the second memory bank 36 to either the system data bus 14 for CPU reads or to the data bus controller 46 for CPU writes to memory which are posted or input/output reads or writes to memory responsive to control signals from the memory controller 26. The multiplexer 24 has a negligible delay for data transfers between the memory terminals 40, 42 and the data terminals 44, 48. In this way, the memory 32 may be coupled directly to the CPU 20 without intervening buffering to thereby reduce the delay and the number of clock cycles for completing a memory access. The multiplexer 24 preferably comprises pass transistors. The multiplexer 24 has a bus hold capability for holding an output signal of the multiplexer 24 at the value the signal reaches after the coupling has switched and the output signal is no longer being driven.

The multiplexer 24 provides an interface to allow the CPU 20 and the memory banks 34, 36 to operate at different voltages, such as a 5 volt to 3.3 volt interface.

The data bus controller 22 controls the transfer of data between the memory 32 and a local bus 50 for input/output reads or writes and between the memory 32 and the system data bus 14. The local bus 50 may be, for example, a high speed bus, such as a Video Electronics Standards Association (VESA) bus or Peripheral Component Interconnect (PCI) bus.

During memory read accesses by the CPU 20, data is communicated directly from the selected memory bank 38 through the corresponding memory terminal 40, 42 through the multiplexer 24 to the CPU 20. This path avoids the delay of passing through the data bus controller 22 to thereby save one cycle.

The CPU 20 writes data to the memory 32 by posting the data to a write buffer (not shown) in the data bus controller 22. The data bus controller 22 provides the data through the memory terminal 46 through the multiplexer 24 to a selected memory terminal 40 or 42 for writing to the memory 32.

The data bus controller 22 accesses the memory 32 for reads or writes to the local bus 50 through the memory terminal 46 through the multiplexer 24 to either memory terminals 40, 42 for writing to the memory 32. The CPU 20 accesses the local bus 50 directly through the data bus controller 22 via the host terminal 21.

By being arranged in this way, the multiplexer 24 accelerates the performance of the memory 32 by reducing the read cycle by one cycle. This allows the system either to use slower less expensive memory to achieve the same performance as faster, more expensive memory, or to use the faster expensive memory to achieve faster performance. The multiplexer 24 also allows interleaving between memory banks 34, 36. This again provides a performance boost for less expensive memory. This allows EDO memory to be used to achieve the same performance as faster, more expensive burst EDO memory. Similarly conventional DRAM may be used instead of EDO. Burst EDO may be used for asynchronous operation. This higher performance eliminates the need and cost of the cache 23. For example, if a CPU read may be performed in a 7:2:2:2 pattern for a CPU 20 having a processor operating at 66 MHz and a memory 32 that includes EDO DRAMs having 70 nanosecond access time, with the present invention the pattern may be 6:1:1:1. The form  $t_0:t_1:t_2:\dots:t_n$  indicates that reads or writes of data are done in  $t_0$  clock cycles for the first read or write,  $t_1$  clock cycles for the second read or write, through  $t_n$  clock cycles for the  $n+1$ -th read or write. The 7 cycles becomes 6 because reads are not delayed through the multiplexer 24 as they are through a conventional chip set. The 2:2:2 portion of the pattern becomes 1:1:1 because of the interleaving. A CPU read may be performed, for example, in a 5:1:1:1 pattern for a memory having 50 nanosecond EDO DRAMs. A read hit may be performed in a 3:1:1:1 pattern. A sequential read hit may be performed in a 2:1:1:1 pattern, because the data is prefetched and left on the bus with the bus hold.



Referring to FIG. 2, there is shown a block diagram illustrating a multi-processor multiplexer interleaved computer 210 in accordance with a second embodiment of the present invention. The computer 210 of FIG. 2 is similar to the computer 10 of FIG. 1 with a data transfer controller 212 including a data bus controller 214 and a first multiplexer 216, and with a second multiplexer 219. The second multiplexer 219 may be an optional external multiplexer. Like elements have like reference numbers. The computer system 210 includes a system bus 12 having a system data bus 14, a system address bus 16, and a system control bus 18. The system data bus 14 interconnects the CPU 20, the cache 23, and a host terminal 220 of the multiplexer 216 for communicating data. The memory bank 34, 36 are coupled to respective bi-directional memory terminals 221 and 223 of the multiplexer 219, which provides the interleave capability. A bi-directional terminal 224 of the multiplexer 219 is coupled to a bi-directional terminal 225 of the multiplexer 216. A negligible delay path 222 couples the host terminal 220 to the bi-directional terminal 225. The integration of the negligible delay path 222 into the data transfer controller 212 provides a reduction of the first read cycle by one cycle without any additional cost to the chip set from adding additional pins. The data bus controller 212 controls the transfer of data between the memory 32 and the local bus 50 and between the memory 32 and the system data bus 14. The data transfer controller 212 may be, for example, a model 82438FX manufactured by Intel Corporation (Santa Clara, California) modified to include the negligible delay path from the host terminal 220 to memory terminal 221. The multiplexer 215 may be, for example, a Model 35257 manufactured by Quality Semiconductor (Santa Clara, California).

The system address bus 16 interconnects the CPU 20, the cache 23, the data transfer controller 212, and a memory controller 26 for communicating address signals.

The system control bus 18 interconnects the CPU 20 and the memory controller 26 for communicating control signals. The memory controller 28 provides control signals to a row address driver 28 for supplying the row address to a memory 32 during an address cycle responsive to such signals. The memory controller 28 provides control signals, such as write enable, row enable, column enable, and output enable, to the memory 32. The memory controller 26 and the data transfer controller 212 are coupled for communicating control and status signals. The memory controller 26 also provides control and addressing signals to the cache. The memory 32 is organized into memory banks 38-1 through 38-4 as in FIG. 1.

The multiplexer 219 selectively couples either the memory bank 34 or the memory bank 36 to the multiplexer 216 responsive to control signals from the memory controller 26. The multiplexer 216 selectively couples the data bus 14 either to the multiplexer 219 for CPU reads or to a host terminal 228 of the data bus controller 214 for CPU writes to memory which are posted or for CPU reads or writes from or to the

input/output or selectively couples the multiplexer 219 to a memory terminal 230 of the data bus controller 214 for communicating data responsive to control signals from the memory controller 26 for memory writes or input/output read or writes. Each memory subbank 38 is coupled to a bi-directional memory terminal 222 of the multiplexer 216. The data transfer controller 212 couples the memory subbanks 38 to the system data bus 14.

Referring to FIG. 3, there is shown a block diagram illustrating a multi-processor multiplexer interleaved computer 310 in accordance with a third embodiment of the present invention. The computer 310 of FIG. 3 is similar to the computer 10 of FIG. 1 with a data transfer controller 312 integrating a data bus controller 314, a multiplexer 316, and a multiplexer controller 318 onto one chip. This integration may reduce overall cost by eliminating the external multiplexer. Like elements have like reference numbers. The computer system 310 includes a system bus 12 having a system data bus 14, a system address bus 16, and a system control bus 18. The system data bus 14 interconnects the CPU 20 and a bi-directional data terminal 320 of the multiplexer 316 for communicating data. Additional CPUs 20' may be coupled to the system bus 12. For clarity, only one additional CPU 20' is shown and is indicated by dashed lines. The system address bus 16 interconnects the CPU 20, the data transfer controller 312, and a memory controller 326 for communicating address signals. The computer system 310 may operate without cache memory. Cache memory may be added to the system 310 by connecting the cache to the system data bus 12 and to the system address bus 14 as shown in FIG. 1 and by adding control logic to the memory controller 326.

The system control bus 18 interconnects the CPU 20 and the memory controller 326 for communicating control signals. The memory 32 is organized into memory banks 38-1 through 38-4 as in FIG. 1. The memory controller 326 provides control signals to a row address driver 28 for supplying row addresses to memory banks 38 during an address cycle responsive to such signals. The memory controller 326 provides control signals, such as write enable, row enable, column enable, output enable, to the memory banks 38. The memory controller 326 and the data transfer controller 312 are coupled for communicating control and status signals.

The data transfer controller 312 couples the memory 32, the system data bus 12, and the local bus 50 for communicating data. The operation of the data transfer controller 312 is similar to that of the data transfer controller 22 of FIG. 1 with the multiplexer 24 internal to the controller 312. The multiplexer 316 provides switching between the signals communicated with the system data bus 12 and the signals communicated with the memory 32. The multiplexer 316 has a bus hold capability for holding the output signal of the multiplexer 316 at the value the signal reaches after the coupling has switched and the signal is no longer being driven. The data transfer

controller 312 selectively couples a first bi-directional memory terminal 332 of the controller 312, which is coupled to the first memory bank 34, either to the system data bus 14 for CPU reads or to a memory terminal 330 of the data bus controller 318 for CPU writes to memory which are posted or for CPU reads or writes from or to input/output responsive to control signals from the memory controller 326. Similarly, the data transfer controller 312 selectively couples a second bi-directional memory terminal 334 of the controller 312, which is coupled to the second memory bank 36, either to the system data bus 14 for CPU reads or to the memory terminal 330 of the data bus controller 318 for memory writes or input/output reads or writes responsive to control signals from the memory controller 326. The data transfer controller 312 also selectively couples the memory terminal 320 to a host terminal 336 of the data bus controller 314 for CPU writes to memory which are posted or for CPU reads or writes from or to input/output. The data bus controller 314 controls the transfer of data between the memory 32 and the local bus 50 and between the memory 32 and the system data bus 14. The multiplexer 316 has a negligible delay for data transfers between the terminals 320, 330, 332, 334, and 336. The multiplexer 316 preferably comprises pass transistors.

Because the multiplexer 316 is internal to it, the data transfer controller 312 has one more connection than the data bus controller 22 (FIG. 1). For a 72 bit bus, the number of pins on the packaging of the data transfer controller 312 is 72 pins more than the controller 22. Although this increases the size of the package and the cost, this package eliminates discrete multiplexers and reduces the area required on the motherboard.

Referring to FIG. 4, there is shown a block diagram illustrating a multiprocessor multiplexer interleaved computer 410 in accordance with a fourth embodiment of the present invention. The computer system 410 has a processor 411 that includes at least one CPU 420. For example, the processor 411 may be a processor module having a single 128-bit CPU 420, a single 64-bit CPU 420, or a pair of modules 411-1 and 411-2 each with its own 64-bit CPU 420-1 and 420-2, respectively. For simplicity, two CPUs 420-1 and 420-2 are shown. The CPU 420 may be, for example, a Pentium processor that are each commercially available from Intel Corporation of Santa Clara, California. A program memory 413 stores the object code for programming a memory controller 426 at system bootup. The object code may be, for example, a unique algorithm for the controller 426 and is specific for the type and grade of the CPU 420. The program memory 413 may be, for example, a conventional electrically erasable programmable read only memory (EEPROM). Upon power on of the computer system 410, the memory controller 426 is preferably unprogrammed. During the initialization of the system 410, the program memory 413 transfers the object code in the program memory 413 over a bus 417 via the connector 419-1 to the

memory controller 426, which then programs itself with the object code. The bus 417 may be, for example, a JTAG bus. Therefore, if the computer system 410 is upgraded with a new processor card 411, the new program memory 413 reconfigures the memory controller 426 for the specific type of CPU 420 on the CPU module 411.

5           The processor module 411 also includes voltage regulators 460-1 and 460-2 for each respective CPU 420-1 and 420-2 for providing operational power for each so that each CPU 420 may operate at a different voltage. For some CPUs, faster performance may be achieved by operating the CPU at a non-standard voltage. For example, a 100 MHz Pentium processor operates at 3.45 volts and a 90 MHz Pentium  
10 processor operates at a standard voltage of 3.3 volts.

          The computer system 410 includes a first system bus 412 having a first system data bus 414, a second system data bus 415, a system address bus 416, and a system control bus 418. The first system data bus 414 interconnects the CPU 420-1 via a connector 419-1 and a first bi-directional data terminal 421 of a multiplexer 422 (or a  
15 "crossbar switch") for communicating data. The second system data bus 415 interconnects the CPU 420-2 via a connector 419-2 and a second data bi-directional terminal 423 of the multiplexer 422 for communicating data. The system address bus 416 interconnects the CPUs 420-1, 420-2, via the connectors 419-1, 419-2, a PCI data bus controller (PCI DBC) 424, the memory controller 426, and a row address driver 428.

20 The PCI data bus controller 424 may be, for example, a model KS82C533 manufactured by Samsung Electronics (San Jose, California). The memory controller 426 may be, for example, a Field Programmable Gate Array, such as model iFX8160-10 manufactured by Altera Semiconductor (San Jose, California). The memory controller 426 provides control signals to the row address driver 428 for supplying the row address through a  
25 multiplexer 450-4 to respective memory banks 38-1 through 38-4 during an address cycle responsive to such signals. The system control bus 418 interconnects the first CPU 420-1, via the connector 419-1, the second CPU 420-2, via the connector 419-2, the PCI data bus controller 424, and the memory controller 426. A host terminal 444 of the data bus controller 424 is coupled to a host terminal 446 of the multiplexer 422 for  
30 communicating between the system data buses 414 or 415 and the local bus 50 and for posting writes to the memory 32 if required. For memories, such as an EDRAM by RAMTRON, which are capable of posting writes, writes are not posted in the data bus controller 424 and the memory 32 is coupled directly to the CPU 420 through the multiplexer 422. A memory terminal 448 of the data bus controller 424 is coupled to a  
35 memory terminal 450 of the multiplexer 422 for communicating data between the memory 32 and the local bus 50. The address buses of the CPUs 420 and the data bus controller 424 are coupled to allow each CPU 420 to monitor (or "snoop") the other to ensure cache coherency.

The connector 419-1 comprises connectors 435-1 and 435-2. Similarly, the connector 419-2 comprises connectors 435-3 and 435-4. A daughterboard module 429-1 includes the CPU 420-1, the program memory 413, the voltage regulator 460-1, and the connector 435-1. A daughterboard module 429-2 includes the CPU 420-2, the voltage  
5 regulator 460-2, and the connector 435-3. A motherboard 427 comprises the memory controller 426 and the remainder of the system 410. The motherboard 427 includes connectors 435-2 and 435-4.

The multiplexer 422 couples the memory 32 to the system data buses 414, 415 to provide switching between the signals communicated with the system data  
10 buses and the signals communicated with the memory banks. The multiplexer 422 selectively couples the first system data bus 414 to the first, second, third, or fourth memory banks 38-1 through 38-4 for CPU reads or the host terminal 444 of the PCI data bus controller 424 for CPU writes to memory which are posted or for CPU reads or  
15 writes from or to input/output responsive to control signals, respectively, from the memory controller 426. Similarly, the multiplexer 422 selectively couples the second system data bus 415 to the first, second, third, or fourth memory banks 38-1 through 38-4 for CPU reads or the host terminal 444 of the PCI data bus controller 424 for CPU  
20 writes to memory which are posted or for CPU reads or writes from or to input/output responsive to control signals, respectively, from the memory controller 426. Further, the multiplexer 422 selectively couples the memory terminal 448 of the PCI data bus controller 424 to the first, second, third, or fourth memory banks 38-1 through 38-4 for  
memory writes or input/output reads or writes responsive to control signals, respectively, from the memory controller 426.

The multiplexer 422 has a bus hold capability for holding the output  
25 signals of the multiplexer 422 at the value the signal reaches after the multiplexer 422 is switched and the signal is no longer being driven. The multiplexer 422 has a negligible delay for data transfers between the terminals 421, 423, and 450 and the terminals 430, 431, 432, 433, and 436. In this way, the memory 32 may be coupled directly to the CPUs 420 without intervening buffering to thereby reduce the delay and the number of clock  
30 cycles for completing a memory access. The multiplexer 422 preferably comprises pass transistors.

The multiplexer 422 provides an interface to allow the CPU 420 and the memory banks 38 to operate at different voltages. The multiplexer 422 provides the interconnections as described later herein in conjunction with FIG. 8. In this way, the  
35 memory may be coupled directly to the CPU without intervening buffering to thereby reduce the delay and the number of clock cycles for completing a memory access.

The data bus controller 424 controls the transfer of data between the memory 32 and a local bus 50 and between the memory and the system data buses 414, 415.

The main memory 32 is organized into memory banks 38-1 through 38-4 as in FIG. 1. The memory banks 38 are interleaved. The memory banks 38-1 through 38-4 are coupled to respective memory terminals 430, 431, 432, 433 of the multiplexer 422 for communicating data. The architecture of the computer system 410 allows simultaneous  
5 access from any CPU 420 or the local bus 50 to any memory bank 38 or from any CPU 420 to the local bus 50. The architecture also allows interleaving of slower memory to achieve higher system performance for the memory.

When either CPU 420 initiates a memory cycle, the memory controller 426 provides the necessary commands to the multiplexer 422 to couple the CPU 420 to the  
10 requested memory bank 38. The memory controller 426 also controls the function of the local bus 50. For this case, the memory controller 426 either couples the memory (M) port of the PCI data bus controller 424 to the appropriate memory bank or couples the CPU 420 to the host terminal 444 of the PCI data bus controller 420 if either CPU 420 is requesting a direct memory access.

The CPUs 420 are coupled in a glueless multiprocessor configuration. Built into each CPU 420 is the control logic to handle snooping the system address bus 416 when the other processor is executing memory accesses without the need for any external "glue" logic. This is a well understood capability as available in Intel Pentium processors for example. At any one time, one CPU is the master of the bus and the  
20 other CPU is the slave. The master CPU has control of the system bus 412 and may initiate data bus transfers. To initiate a data bus transfer, the slave CPU requests the bus from the master CPU and the master CPU must grant such request. The CPUs share the system address bus 416 so that whenever the master CPU provides an address to the memory, the slave CPU may snoop the address to determine if the  
25 requested data is present in its on-chip primary cache (not shown). If the data is present, the slave CPU notifies the master CPU via the system control bus 418 that there is a snoop hit. The master CPU terminates the memory cycle and allows the slave to write the data to the memory. While the slave CPU is writing the data to the memory, the master CPU may read the data at the same time. The multiplexer 422  
30 allows the master CPU to be coupled to the slave CPU while the slave CPU is coupled to the memory 32.

The CPUs 420 also may snoop the bus during writes to ensure that cache entries of a cache internal to the CPU 420 are invalidated if the corresponding data in the main memory is being overwritten. Once the master CPU initiates a cycle, it no  
35 longer needs to command the system address bus. In fact, the slave CPU or the local bus 50 may take control of the system address bus 416 and initiate its own memory cycle using its respective data bus even if the master CPU has not completed its cycle. The local bus 50 also presents its address on the system address bus 416 for the CPUs 420 to snoop.

In conventional systems, the slave CPU can conduct a memory cycle only after the master CPU completes its cycle and grants the bus to the slave CPU. Here, to take advantage of the dual CPU buses, the slave CPU may initiate a cycle even while the master CPU is completing a cycle. The CPU 420 has control logic that allows the  
5 slave CPU to take command of the system address bus 416 and the system control bus 418 to allow the slave CPU to initiate a cycle even before the master CPU has granted the slave CPU the bus. The cycle is not completed until the master CPU has become the slave and snoops the memory cycle to maintain cache coherency.

The memory controller 426 provides memory control signals to a  
10 multiplexer 450-1 for crossbar coupling to the memory bank 38. The memory controller 426 provides control signals to multiplexers 450-1, 450-2, 450-3, 450-4 for controlling the operation of the multiplexers 450. The multiplexers 450 have a bus hold capability for holding the output signal at the value the signal reaches after the switch has changed and the signal is no long being driven. This hold feature allows memory  
15 cycles to occur at the same time in multiple memory banks. Multiple cycles are not initiated at the same time because the address bus is shared.

The memory controller 426 arbitrates if there are contentions of the memory banks 38. In other words, both CPUs 420 or one CPU 420 and the local bus 50 may be requesting the same memory bank 38 at the same time. This may occur, for  
20 example, with multiple CPUs running a multi-threaded application. The memory 32 is organized to reduce locality and improve the likelihood that each CPU does not access the same memory bank 38. The memory 32 is organized on a cache line or multiple cache line basis with subsequent cache lines located in subsequent memory banks 38. By spreading the application across all memory banks, the likelihood that the CPUs 420  
25 are accessing the same bank is reduced. The interleave pattern is adjustable based on the type of operating system, the type of application, and the type of memory. For example, for burst EDO, synchronous EDO, or EDRAM, which can support 15 nanosecond cycles, the interleave group may be two banks. With conventional DRAM, the interleave group is four banks. An interleave group is defined so that a cache line  
30 access occurs all within that group. Because the SIMMs are designed to allow interleaving from one side to the other, using double sided SIMMs doubles the number of banks.

The memory controller 426 may prefetch the next location in the memory 32 in the next cycle after a read in anticipation of a subsequent request for this  
35 information. The data is left on the system data bus 414 or 415. The memory controller 426 detects a read to this next location and immediately responds to the CPU 420 that the data on the system data bus 414 is correct. The memory controller 426 also initiates the rest of the read cycle which continues in a 2:1:1:1 pattern.

A CPU read may be performed, for example, in a 5:1:1:1 pattern for a CPU 420 operating at 66 MHz and a memory 32 that includes EDRAMS having a 35 nanosecond access time. A read hit may be performed in a 3:1:1:1 pattern. A sequential read hit may be performed in a 2:1:1:1 pattern, because the data is prefetched and left on the bus with the bus hold. Writes are posted directly to the memory without buffering in a 2:1:1:1 pattern. For a memory 32 that includes EDO DRAMS having a 60 nanosecond access time, reads may be performed, for example, in a 7:1:1:1 pattern with interleaving of two banks 38.

The memory controller 426 provides control signals to a column address latch/write enable circuit (CAL/WE) 440, and a set of output enable latches (OE) 442. The column address latch/write enable circuit 440 latches the column address during an address cycle and provides the latched address through a multiplexer 450-2 to the memory 32 in response to the control signals from the memory controller 424. The column address latch/write enable circuit may be, for example, a model 16823 manufactured by Integrated Device Technology (Santa Clara, California). The output enable latches 442 provide output enable signals through a multiplexer 450-3 to the memory 32 in response to control signals from the memory controller 424. The output enable latches 442 may be, for example, a model 16R8-4 manufactured by Cypress Semiconductor (San Jose, California). The multiplexers 450 are described later herein in conjunction with FIG. 9.

The interleaved memory and the crossbar multiplexer 422 provides each CPU 420 with its own direct data path to each memory bank 38 and to the local bus interface. The local bus interface also has its own direct data path to the memory 32. Additionally, the architecture provides a programmable interface between the CPUs 420 and the memory controller 426. The system 410 allows the CPUs 420 to access (either a read or a write) one memory bank 38 while a direct memory access is occurring on another memory bank 38.

Referring to FIG. 5, there is shown a flowchart illustrating the initialization of the computer system 410. As is well known in the art, the CPU 420 is held in reset at power on. The memory controller 426 retrieves the programming code from the program memory 413 over the JTAG bus 417 to program the controller 426.

Such programming of the memory controller 426 may be used, for example, to reconfigure the computer system 410 for upgrading, such as performance improvement or repair defects discovered after the system is initially programmed. In addition, such programming allows the system to be upgraded for other types of CPUs, which may have different operating parameters, such as, significantly different timing/signal requirements.



Referring to FIG. 6, there is shown a block diagram illustrating a multi-processor multiplexer interleaved computer 610 in accordance with a fifth embodiment of the present invention. The computer system 610 is similar to the computer system 410 (FIG. 3) but has increased CPUs, memory, and local bus interfaces. The system 610 has a multiplexer 612 having eight data terminals 614-1 through 614-8 and eight memory terminals 616-1 through 616-8. The multiplexer 612 selectively couples the data terminals 614 to the memory terminals 616.

A processor 619 has four CPUs 620-1 through 620-4. Each CPU 620 has a program memory 621 for storing the object code for programming a memory controller 624 at system bootup. A system bus 626 has four system data buses 628-1 through 628-4 that each couple a respective CPU 620-1 through 620-4 to a respective data terminal 614-2 through 614-5 of the multiplexer 612 for communicating data. A shared memory 622 for a video graphics adapter (VGA) (not shown) is coupled to the data terminal 614-1 of the multiplexer 612 for communicating data. A system address bus 630 interconnect the CPUs 620, a row address driver 632, and the memory controller 624 for communicating addresses. A system control bus 634 couples the CPUs 620 and the memory controller 624 for communicating control signals.

The memory controller 624 provides control signals to column address latches 440, output enable latches 442, and a plurality of multiplexers 450-1 through 450-4 in a manner similar to that of the memory controller 426 of FIG. 4. The column address latches 440 and output enable latches 442 are coupled to respective multiplexers 450-2, 450-3, which in turn are coupled to memory banks 622-1 through 622-5 of the memory 621. The operation of the column address latches 450-2, the output enable latches 450-3, the row address driver 632, and the memory banks 622 is similar to that of FIG. 4 except that here five memory banks 632 are shown instead of four.

The memory controller 624 and a plurality of data bus controllers 634 are coupled for communicating control and status signals. For clarity only, three data bus controllers 634-1 through 634-3 are shown. The data bus controllers 634 couple to respective local buses. The data bus controllers 634-1 through 634-3 have a bi-directional memory terminals 636 coupled to a corresponding data terminal 614-5 through 614-7 of the multiplexer 612 and have a bi-directional host terminal 638 coupled to a corresponding bi-directional memory terminal 616-5 through 616-7 of the multiplexer 612 for communicating data.

Each memory bank 622-1 through 622-5 is coupled to a corresponding memory terminal 616-2 through 616-5 of the multiplexer 612 for communicating data. The multiplexer 612 selectively couples each memory bank 622 via a memory terminal 616-1 through 616-5 to a respective data terminal 614-1 through 614-5 for CPU reads or to a respective data terminal 614-6 through 614-8 for memory writes or input/output

reads or writes. The multiplexer 612 also selectively couples each host terminal 638 of the data bus controllers 634 via a memory terminal 616-6 through 616-8 to a respective data terminal 614 for CPU writes to memory which are posted or for CPU reads or writes from or to input/output. The multiplexer 612 has a bus hold capability for  
5 holding the output signals of the multiplexer at the value the signal reaches after the multiplexer is switched and the signal is no longer being driven. The multiplexer 612 has a negligible delay for data transfers between the data terminals 614 and the memory terminals 616. In this way, the memory 32 may be coupled directly to the CPUs 620 without intervening buffering to thereby reduce the delay and the number of  
10 clock cycles for completing a memory access. The multiplexer 612 preferably comprises pass transistors.

Referring to FIG. 7, there is shown a block diagram illustrating a multiprocessor multiplexer interleaved computer 710 in accordance with a sixth embodiment of the present invention. The computer system 710 is similar to the  
15 computer system 410 (FIG. 4) with the multiplexing functions, the local bus interface, and the memory controller integrated into a data transfer controller 712 and with the address drivers and multiplexers integrated into an address driver 728. A data transfer controller has a multiplexer 722, a local bus interface 724, and a memory controller 726.

The computer system 710 has a processor 411 that includes at least one  
20 CPU 420. For simplicity, two CPUs 420-1, 420-2 are shown. The CPU 420-1 includes a program memory 413 for storing the object code for transferring over a bus 417 to program the memory controller 726 at system bootup and includes a voltage regulator 460-1. The CPU 420-2 includes a voltage regulator 460-2. The voltage regulators 460-1 and 460-2 provide operational power to the respective CPU 420-1 and 420-2 so that  
25 each CPU 420 may operate at a different voltage for different operational performance. The CPUs 420 are on respective daughterboard modules 429-1 and 429-2, respectively. The daughterboard modules 429-1 and 429-2 include connectors 435-1 and 435-3, respectively, which are part of respective connectors 419-1 and 419-2. A motherboard 760 includes the remainder of the system 710 and in particular includes connectors 435-  
30 2 and 435-4 which are part of connectors 419-1 and 419-2, respectively.

The computer system 710 includes a first system bus 712 having a first system data bus 714, a second system data bus 415, a system address bus 716, and a system control bus 718. The first system data bus 714 interconnects via the connector 419-1 the CPU 420-1 and a first bi-directional data terminal 721 of the multiplexer 722  
35 for communicating data. The second system data bus 715 interconnects via the connector 415-2 the CPU 420-2 and a second data bi-directional terminal 723 of the multiplexer 722 for communicating data. The system address bus 716 interconnects via the connectors 419 the CPUs 420-1, 420-2, the local bus interface 724, the memory controller 726, and a CPU address driver 762 of the address driver 728 for

communicating address signals. The CPU address driver 762 receives only row addresses from the system address bus 716. The address buses of the CPUs 420 and the local bus controller 724 are coupled to allow each CPU 420 to monitor (or "snoop") the other to ensure cache coherency.

5           The memory controller 726 provides column and row addresses to a local bus address driver 760 of the address driver 728 for supplying the address through multiplexers 750-1 and 750-2 of the address driver 728 to respective memory banks 38-1 through 38-4 during an address cycle responsive to such signals. Similarly, the memory controller 726 column addresses to the CPU address driver 762 of the address  
10 driver 728 for supplying the address through a multiplexers 750-1 and 750-2 of the address driver 728 to respective memory banks 38-1 through 38-4 during an address cycle responsive to such signals. The memory controller 726 provides control signals to multiplexers 750-1, 750-2 for controlling the operation of the multiplexers 750. Each multiplexer 750 has the bus hold feature on its outputs which provide memory for  
15 those lines when they are not actively driven. The multiplexers have negligible delay for communicating signals. This configuration minimizes the number of active drivers and provides the voltage matching required for 3.3 volt and 5 volt memories.

          The system control bus 718 interconnects via the connectors 419 the first CPU 420-1, the second CPU 420-2, the local bus controller 724, and the memory  
20 controller 726. A host terminal 744 of the local bus controller 724 couples to the local bus 50 for communicating between the system data buses 714, 715 and the local bus 50.

          The multiplexer 722 couples the memory 32 to the system data buses 714, 715 to provide switching between the signals communicated with the system data buses and the signals communicated with the memory 32. The multiplexer 722  
25 selectively couples the first system data bus 714 to the first, second, third, or fourth memory banks 38-1 through 38-4 for CPU reads or writes to memory or the host terminal 744 of the PCI data bus controller 424 for reads or writes from or to input/output responsive to control signals, respectively, from the memory controller 426. Similarly, the multiplexer 722 selectively couples the second system data bus 715  
30 to the first, second, third, or fourth memory banks 38-1 through 38-4 for CPU reads or the host terminal 744 of the local bus controller 724 for reads or writes from or to input/output responsive to control signals, respectively, from the memory controller 726.

          The multiplexer 722 has a bus hold capability for holding the output signal  
35 of the multiplexer 722 at the value the signal reaches after the multiplexer 722 is switched and the signal is no longer being driven. The multiplexer 722 has a negligible delay for data transfers between the data terminals 721, 723 and either the memory terminals 730, 731, 732, 733 or the host terminal 744. In this way, the memory 32 may be coupled directly to the CPUs 420 without intervening buffering to thereby reduce

the delay and the number of clock cycles for completing a memory access. The multiplexer 722 preferably comprises pass transistors. The multiplexer 722 provides the interconnections as described later herein in conjunction with FIG. 8.

The local bus controller 724 controls the transfer of data between the  
 5 memory 32 and a local bus 50 and between the memory and the system data buses 714, 715.

The main memory 32 is organized into memory banks 38-1 through 38-4 as in FIG. 1. The memory banks 38 are interleaved. The memory banks 38-1 through 38-4 are coupled to respective memory terminals 730, 731, 732, 733 of the multiplexer 722 for  
 10 communicating data.

The memory controller 726 provides control signals to a column address latch (CAL) 740, and a set of output enable latches (OE) 742. The column address latch 740 latches the column address during an address cycle and provides the latched address to the memory 32 in response to the control signals from the memory  
 15 controller 724. The output enable latches 742 provide output enable signals to the memory 32 in response to control signals from the memory controller 724. The multiplexers 750 are described later herein in conjunction with FIG. 8.

Referring to FIG. 8, there is shown a schematic diagram illustrating the interconnections of a quad multiplexer 824. The multiplexer 824 includes four 4 x 4 crossbars 825-1 through 825-4. For simplicity, only crossbar 825-1 is described. Bi-  
 20 directional terminals 826-1 through 826-4 each may be selectively coupled to one or more of bi-directional terminals 828-1 through 828-4 or selectively decoupled from any of the terminals 828.

Referring to FIG. 9, there is shown a schematic diagram illustrating the interconnections of the multiplexer of FIG. 4. A multiplexer 924 includes four 3 x 5 crossbars 925-1 through 925-4. For simplicity only one crossbar 924-1 is described. A crossbar 925 may be used for the multiplexer 422 (FIG. 4). The crossbar 924-1 has 15 possible states of interconnection. Bi-directional terminals 926-1 through 926-3 each may be selectively coupled to one or more of bi-directional terminals 928-1 through  
 30 928-5 or selectively decoupled from any of the terminals 928.

One state, the connection between terminal 926-1 (PCIM) and the terminal 928-1 (PCIH), is invalid and is indicated with dashed lines. This leaves 14 valid states of interconnection. Using four control terminals provides 16 possible states. With only 14 valid states, two states are unused. These states are used to either couple the first  
 35 CPU 420-1 to the second CPU 420-2 or vice versa. For example, if the terminal 926-2 (CPU0A) to the first CPU 420-1 is coupled (depicted in bold) to the terminal 928-2 (M0A) and a command to couple the first CPU 420-1 to the second CPU 420-2 is issued, the terminal 926-3 (CPU1A) for the second CPU 420-2 is also coupled to the terminal 928-2 (M0A). In the event of a snoop hit, the second CPU 420-2, in this example, may

flush its cache and at the same time that it is being written to the main memory 32, the first CPU 420-1 may read the data. Each terminal 926, 928 on the crossbar 925 has an active hold (T) device for holding the voltage of a signal on the terminal so that the bus is held at the last driven value if a connection has broken. The crossbar 925 has a negligible delay for data transfers between the bi-directional terminals 926-1 through 926-3 and the bi-directional terminals 928-1 through 928-5. The crossbar 925 preferably comprises pass transistors.

Referring to FIG. 10, there is shown a block diagram illustrating a conventional interconnection of a single in line modules (SIMMs). A memory SIMM 908 is organized into memory subbanks 904, 906, one on one side of the SIMM and the other on the opposite side. Each memory subbank 904 & 906 has 8 DRAM memory devices 902 organized in pairs 901 so that reads and writes can be performed in single byte increments. Each memory subbank 904 & 906 supplies a total of 32 bits of information. Optionally each subbank can contain 9 DRAM memory devices and supply 36 bits of information for parity or error correcting coding (ECC).

Each device 902 has an input for a row address strobe (/RAS), column address strobe (/CAS), a write enable (/WE), and an output enable (/OE) control signals. Each memory bank 904, 906 has its own row address strobes. The output enable (/OE) input of each device 902 is grounded.

A write enable (/WE) signal 901 is applied to each memory device 902. Address (A0-A11) signals 912 are applied to each memory device 902. A first row address strobe (/RAS0) 914, a second row address strobe (/RAS1) 916, a third respective row address strobe (/RAS2) 918, and a fourth row address strobe (/RAS3) 920 are applied to each memory device 902 of the memory banks, 903-1, 903-3, 903-2, and 903-4. A first column address strobe (/CAS0) 922 is applied to each memory device 902 of the DRAM pairs 901-1 and 901-5. A second column address strobe (/CAS1) 924 is applied to each memory device 902 of the DRAM pairs 901-2 and 901-6. A third column address strobe (/CAS2) 926 is applied to each memory device 902 of the DRAM pairs 901-3 and 901-7. A fourth column address strobe (/CAS3) 928 is applied to each memory device 902 of the DRAM pairs 901-4 and 901-8. In this configuration, column address strobes are common between the memory banks 904, 906.

Referring to FIG. 11, there is shown a block diagram illustrating the memory 1108. The SIMMs of FIG. 11 differ from the SIMMs of FIG. 10 by allowing interleaving between the two banks of memory that are on each SIMM. In conventional packages, one bank is one side of the package and the other bank is on the other side.

Each DRAM memory device 1102 has an input for a row address strobe (/RAS), column address strobe (/CAS), a write enable (/WE), and an output enable

(/OE) control signals. Each memory subbank 1104, 1106 has its own column address strobes, own row address strobes, output enable, and write enable. In particular, a first write enable (/WEA) signal 1110-1 and a second write enable signal (/WEB) signal 1110-2 are applied to the write enable input (/WE) of each memory device 1102 of the memory subbanks 1104, 1106, respectively. Similarly, a first output enable (/OEA) signal 1112-1 and a second output enable signal (/OEB) signal 1112-2 are applied to the output enable input (/OE) of each memory device 1102 of the memory subbanks 1104, 1106, respectively. A first row address strobe (/RASA0) 1014-1, a second row address strobe (/RASA1) 1014-2, a third row address strobe (/RASB0) 1014-3, and a fourth row address strobe (/RASB1) 1014-4 are applied to the row address strobe input (/RAS) of each memory device 1102 of the respective memory groups 1103-1, 1103-2, 1103-3, and 1103-4. A first column address strobe (/CASA0) 1116-1 is applied to the column address strobe input (/CAS) of each memory device 1102 of the DRAM pairs 1101-1 and 1101-3. A second column address strobe (/CASA1) 1116-2 is applied to the column address strobe input (/CAS) of each memory device 1102 of the DRAM pairs 1101-2 and 1101-4. A third column address strobe (/CASB0) 1116-3 is applied to the column address strobe input (/CAS) of each memory device 1102 of the DRAM pairs 1101-5 and 1101-7. A fourth column address strobe (/CASB1) 1116-4 is applied to the column address strobe input (/CAS) of each memory device 1102 of the DRAM pairs 1101-6 and 1101-8. In this configuration, column address strobes 1116 are not common between the memory subbanks 1104, 1106.

Address (A0 through A11) signals 1118-1 and address (A0 through A7, A11, A9, A10, A8) signals 1118-2 are applied to the address input of each memory device 1102 of the respective memory subbank 1104, 1106. Addresses 1118 are arranged so that one column address line is unique to each subbank pair 1104, 1106. Such an arrangement allows the address to be changed to each subbank at a different time to maximize separately the address setup time for each subbank as described in patent application 08/215,144 referenced earlier herein. The unique address line may also carry a common row address if the memory devices 1102 have more row addresses than column addresses. Hence, either balanced (e.g., 11 rows and 11 columns) or imbalanced (e.g., 12 rows and 10 column) memory chips may be used. The memory chips may be, for example, 4 Mb, 16 Mb, or 64 Mb balanced chips. Additional address lines are added to extend the memory for greater density memory chips. Table I shows the addressing for one implementation for balanced memory devices 1102 and for one implementation for imbalanced memory devices 1102. For the depicted design, the inputs of the memory devices 1102 have only rows and no columns for addresses A10 and A11. The uniqueness is used for address A11 only. The row addresses, column addresses H and column addresses L are described later herein in conjunction with the timing diagrams of FIGs. 12-15.

Referring to FIGs. 12a-12h, there are shown timing diagrams illustrating a read and a page hit with prefetch. In a burst read cycle, the row address driver 428 presents the row address 1201 (FIG. 12a), 1202 (FIG. 12b), 1203 (FIG. 12c) to both subbanks 1104, 1106. The row address latch (/RAS) signal 1204 (FIG. 12d) latches the row address in both subbanks 1104, 1106. The column address latches 440 provides the column address 1201, 1202, 1203 to both subbanks. The output enable latches 442 alternately provides the output enable (/OE) signals 1206 (FIG. 12f), 1207 (FIG. 12g) to each subbank 1104, 1106. The memory controller 426 increments the low order column address (column address L) 1201, 1202 after the read from its subbank. After the four word read burst completes, the memory controller 426 increments the high order column address (column address H) signal 1203 to prefetch the next word in the event the next access is sequential. The data 1205 (FIG. 12e) is left on the data bus. The memory controller detects a read to this next location and immediately responds to the CPU that the data on the data bus is correct. The memory controller also initiates the remainder of the read cycle which continues in a 2:1:1:1 pattern of clock cycles 1208 (FIG. 12h). With a pipelined CPU-like Pentium processor, back to back reads may occur in X:1:1:1:1:1:1 to thereby eliminate the extra cycle if an additional pair of address lines is used to create another column address L as depicted in Table II. A longer burst cycle may be accommodated by interleaving between banks.

Referring to FIGs. 13a-13h, there are shown timing diagrams illustrating a read with page hit column miss. In a burst read cycle, the row address driver 428 presents the row address 1301 (FIG. 13a), 1302 (FIG. 13b), 1303 (FIG. 13c) to both subbanks 1104, 1106. The row address latch (/RAS) signal 1304 (FIG. 13d) latches the row address in both subbanks 1104, 1106. The column address latches 440 provides the column address 1301, 1302, 1303 to both subbanks. The output enable latches 442 alternately provides the output enable (/OE) signals 1305 (FIG. 13e), 1306 (FIG. 13f) to each subbank 1104, 1106. The memory controller 426 increments the low order column address (column address L) 1301, 1302 after the read from its subbank. After the four word read burst completes, the memory controller 426 increments the high order column address (column address H) signal 1303 to prefetch the next word in the event the next access is sequential. The data 1307 (FIG. 13g) is left on the data bus. If the next access is a page hit but is not sequential, the new column address 1303 replaces the prefetch address to thereby add one clock cycle 1308 (FIG. 13h) to the access.

Referring to FIGs. 14a-14h, there are shown timing diagrams for a read with page miss. In a burst read cycle, the row address driver 428 presents the row address 1401 (FIG. 14a), 1402 (FIG. 14b), 1403 (FIG. 14c) to both subbanks 1104, 1106. The row address latch (/RAS) signal 1404 (FIG. 14d) latches the row address in both subbanks 1104, 1106. The column address latches 440 provides the column address 1401, 1402, 1403 to both subbanks. The output enable latches 442 alternately provides

the output enable (/OE) signals 1405 (FIG. 14e), 1406 (FIG. 14f) to each subbank 1104, 1106. The memory controller 426 increments the low order column address (column address L) 1401, 1402 after the read from its subbank. After the four word read burst completes, the memory controller 426 increments the high order column address

- 5 (column address H+1) signal 1403 to prefetch the next word in the event the next access is sequential. The data 1407 (FIG. 14g) is left on the data bus. If the next access is a page miss, a complete new read access begins by presenting new row address 1401, 1402, 1403. The clock 1408 (FIG. 14h) provides the timing.

- Referring to FIGs. 15a-15j, there are shown timing diagrams for an  
 10 interleaved posted write and an interleaved write. The writes may be interleaved as shown after time 1511 of the clock 1510 (FIG. 15j) or posted interleaved as shown after time 1511. In a posted write cycle, the row address driver 428 presents the row address 1501 (FIG. 15a), 1502 (FIG. 15b), 1503 (FIG. 15c) and the row address strobe (/RAS) 1504 (FIG. 15d) to both subbanks 1104, 1106. The data 1507 (FIG. 15g) is latched into the  
 15 memory 1108 with write enable (/WEA) signal 1508 (FIG. 15h) and write enable (/WEB) signal 1209 (FIG. 15i) even before the column address 1501, 1502 is available and latched by the column address strobes (/CASA, /CASB) 1505 (FIG. 15e), 1506 (FIG. 15f).

- Posted back to back writes can continue in a X:1:1:1:1:1:1:1 pattern like  
 20 reads if they are a page hit without the sequential constraint. A non-burst cycle may be to any 8-bit segment and is the same with the selected portion of the memory getting the necessary row address strobe (/RAS), column address strobe (/CAS), and write (/WE).

- Referring to FIG. 16a, 16b, and 16c, there are shown schematic diagrams  
 25 illustrating a conventional non-interleaved memory, a page interleaved architecture memory, and a cache line size architecture memory, respectively. Referring in particular to FIG. 16a, memory bank 38-1, memory bank 38-2, memory bank 38-3, and memory bank 38-4 are addressed by dividing the address space into equal consecutive blocks of addresses and assigning the blocks to memory banks. For example, memory  
 30 bank 38-1, memory bank 38-2, memory bank 38-3, and memory bank 38-4 are addressed as 0-16M, 16-32M, 32-48M, and 48-64M, respectively. Referring in particular to FIG. 16b, memory bank 38-1, memory bank 38-2, memory bank 38-3, and memory bank 38-4 are addressed by dividing the address space into pages and assigning the sequentially assigning the pages to a memory bank. For example, for 2 K pages,  
 35 memory bank 38-1 is assigned addresses 0-2 K, 8-10 K, 16-18K, through 63,992 K - 63,994 K; memory bank 38-2 is assigned addresses 2-4 K, 10-12 K, 18-20 K, through 63,994 K - 63,996 K; memory bank 38-3 is assigned addresses 4-6 K, 12-14 K, 20-22 K, through 63,996 K-63,998 K; and memory bank 38-4 is assigned addresses 6-8 K, 14-16 K, 22-24 K, through 63,998 K - 64M.



Referring in particular to FIG. 16c, the memory 32 is organized on a cache line basis. Memory bank 0, memory bank 1, memory bank 2, and memory bank 3 are addressed by dividing the address space on a cache line basis and assigning the subsequent cache lines to subsequent memory banks. For example, for 4 words per cache line, memory bank 0 is assigned addresses 0-3, 16-19, 32-35, through 67108849-67108852; memory bank 1 is assigned addresses 4-7, 20-23, 36-39 through 67108853-67108856; memory bank 2 is assigned addresses 8-11, 24-27, 40-43 through 67108857-67108860; and memory bank 3 is assigned addresses 12-15, 28-31, 44-47 through 67108861-67108864. However, the address space may be assigned in multiples of one cache line.

This organization of the memory 32 reduces memory bank contention in multiple CPU multi-threaded applications in which each CPU, because of locality, may commonly be accessing the same memory bank while running the same application. An interleaved architecture evenly spreads the addressing of the application across the memory banks to reduce the likelihood of both CPUs accessing the same memory bank. The interleave pattern is adjustable based on the type of operating system and the type of application.

The interleave may be done between banks of memory instead of just between subbanks residing in the same SIMM. Such interleaving allows the system to use other types of memory devices and still achieve the X:1:1:1 performance. For example, for burst EDO, synchronous EDO, or EDRAM, which can support 15 nanosecond cycles, the interleave group may be two banks. With conventional DRAM, the interleave group is four banks. An interleave group is defined so that a cache line access occurs all within that group. Because the SIMMs are designed to allow interleaving from one side to the other, using double sided SIMMs doubles the number of banks.

Referring to FIG. 17, there is shown a schematic diagram illustrating the interconnections of the multiplexer of FIG. 6. The multiplexer 1724 includes two 8 x 8 crossbars 1725-1 and 1725-2. For simplicity only one crossbar 1725-1 is described. A crossbar 1725 may be used for the multiplexer 611 (FIG. 6). The crossbar 1725-1 has 16 possible states of interconnection. Bi-directional terminals 1726-1 through 1726-8 each may be selectively coupled to one or more of bi-directional terminals 1728-1 through 1728-8. Each terminal 1726, 1728 on the crossbar 1725 has an active hold device for holding the voltage of a signal on the terminal so that the bus is held at the last driven value if a connection has broken. The crossbar 1725 has a negligible delay for data transfers between the bi-directional terminals 1726-1 through 1726-8 and the bi-directional terminals 1728-1 through 1728-8. The multiplexer 1724 may be, for example, a model 3B882 manufactured by Quality Semiconductor (Santa Clara, California). The crossbar 1725 preferably comprises pass transistors.

Referring to FIG. 18, there is shown a flowchart illustrating the operation of the interleaving of the memory 1108 (FIG. 11) for balanced row addresses. As shown in Table I, the column address L is address A8 for bank 1104 and is address A11 for bank 1106. In a read cycle, row addresses 1118 are applied 1802 via the address 1118 to the address inputs of both subbanks 1104, 1106. A row address strobe applied 1804 to the subbanks 1104, 1106 latches the row address in the subbanks. The column address is applied 1806 to both subbanks 1104, 1106 to enable reading from a memory device 1102 in an addressed DRAM pair 1101. A first output enable is applied 1808 to the first subbank 1104. The column address L to the first subbank 1104 is incremented 1810 and a first output enable is applied to the second subbank 1106. The column address L to the second subbank 1106 is incremented 1812 and a second output enable is applied to the first subbank 1104. The column address L to the first subbank is incremented 1814 and a second output enable is applied to the second subbank 1106. The column address H to both subbanks 1104, 1106 is incremented 1816 and the column address L to the second subbank 1106 is incremented.

If no further read is to be performed 1818, the read ends 1820. If a further read is to be performed 1818, the memory controller determines 1822 if the read is sequential. If the read is sequential 1822, a first output enable is applied 1808 as described earlier herein. If the read is not sequential 1822, the memory controller determines 1824 whether there is page hit. If there is, the column address is applied 1806 as described earlier herein. If there is not a page hit 1824, the row addresses are applied 1802 as described earlier herein.

In summary, the present invention provides a programmable memory controller that receives its operating program from a memory within a CPU module at power up. The memory controller controls the selection of a crossbar switch multiplexer that couples memory banks to CPUs or local buses or couples the CPUs to the local buses. The simultaneous processing of data between any CPU and memory or input/output or between the input/output and memory increases overall bandwidth. The CPU modules include a CPU without cache or cache control, a memory to program the motherboard programmable controller, and a voltage regulator to provide less expensive and more versatile upgradeability to convert the system from one processor type or class to another. For example, a Pentium based CPU module may be replaced by a MIPS based CPU module. Further, additional CPU modules may be added.

TABLE I

Imbalanced	SIMM Interface	Bank A 1104	Bank B 1106
Row Addresses	A0-A11	A0-A11	A0-A11
Column Addresses H	A0-A7, A9	A0-A7, A9	A0-A7, A9
Column Addresses L	A8/A11	A8	A11

Balanced	SIMM Interface	Bank A 1104	Bank B 1106
Row Addresses	A0-A9, A10/A11	A0-A10	A0-A9, A11
Column Addresses H	A0-A7, A9, A10	A0-A7, A9, A10	A0-A7, A9, A10
Column Addresses L	A8/A11	A8	A11

5 TABLE II

Imbalanced	SIMM Interface	Bank A 1104	Bank B 1106
Row Addresses	A0-A11	A0-A11	A0-A11
Column Addresses H	A0-A7	A0-A7	A0-A7
Column Addresses L0	A8/A11	A8	A11
Column Addresses L1	A9/A10	A9	A10

I claim:

1. A computer comprising:
  - 5 at least one central processing unit (CPU);
  - a bus coupled to the at least one CPU;
  - a plurality of memories
  - a data bus controller for communicating with a local bus;
  - a multiplexer having a first bi-directional terminal coupled to a first
  - 10 portion of the plurality of memories, having a second bi-directional terminal coupled to a second portion of the plurality of memories, having a third bi-directional terminal coupled to the bus, having a fourth bi-directional terminal coupled to the data bus controller, having a first input for receiving first, second, third, and fourth control signals, having a negligible delay for communicating signals between the first and third
  - 15 terminals, having a negligible delay for communicating signals between the first and fourth terminals, having a negligible delay for communicating signals between the second and third terminals, and having a negligible delay for communicating signals between the second and fourth terminals, the multiplexer communicating signals between the first and third terminals responsive to the first control signal,
  - 20 communicating signals between the first and fourth terminals responsive to the second control signal, communicating signals between the second and third terminals responsive to the third control signal, and communicating signals between the second and fourth terminals responsive to the fourth control signal;
  - a multiplexer controller having an output coupled to the first input
  - 25 of the multiplexer for providing the first, second, third, and fourth control signals; and
  - a memory controller coupled to the bus and the plurality of memories for providing addressing signals to the plurality of memories.
2. The computer of claim 1 wherein the multiplexer controller either
- 30 simultaneously provides the first and second control signals or simultaneously provides the third and fourth control signals and the multiplexer simultaneously communicates signals between the first and third terminals and between the first and fourth terminals responsive to the first and second control signals or simultaneously communicates signals between the second and third terminals and between the second
- 35 and fourth terminals responsive to the third and fourth control signals.
3. A computer system comprising:
  - a first bus having a data bus, having a control bus, and having an
  - address bus;

a plurality of memories coupled to the data bus  
a removable processor module coupled to the first bus, having a central processing unit (CPU) of a particular configuration, and having a program memory for storing an initialization program representative of the CPU of the particular configuration; and

a programmable memory controller coupled to the address bus, to the control bus, and to the plurality of memories for providing addressing signals to the plurality of memories and coupled to the first bus for receiving the initialization program.

4. The computer system of claim 3 further comprising:

a data bus controller for communicating with a second bus;

a multiplexer having a first bi-directional terminal coupled to a first portion of the plurality of memories, having a second bi-directional terminal coupled to a second portion of the plurality of memories, having a third bi-directional terminal coupled to the bus, having a fourth bi-directional terminal coupled to the data bus controller, and having a first input for receiving first, second, third, and fourth control signals, the multiplexer communicating signals between the first and third terminals responsive to the first control signal, communicating signals between the first and fourth terminals responsive to the second control signal, communicating signals between the second and third terminals responsive to the third control signal, and communicating signals between the second and fourth terminals responsive to the fourth control signal; and

a multiplexer controller coupled to the first input of the multiplexer for providing the first, second, third, and fourth control signals.

5. A method for operating a memory having a first memory circuit and a second memory circuit, each memory circuit having a corresponding plurality of address inputs, the method comprising the steps of:

coupling via a first connection a first portion of the plurality of address inputs of the first memory circuit to a corresponding first portion of the plurality of address inputs of the second memory circuit;

coupling a second connection to a second portion of the plurality of address inputs of the first memory circuit;

coupling a third connection to a second portion of the plurality of address inputs of the second memory circuit, the second portion of the plurality of address inputs of the second memory circuit being different from the second portion of the plurality of address inputs of the first memory circuit;

applying a first address signal to the first and second connections to access data in the first memory circuit; and

applying a second address signal to the first and third connections to access data in the second memory circuit.

5

6. A computer system for executing instructions arranged as cache lines comprising:

a plurality of memory banks, each memory bank having corresponding addressable locations; and

10

a bus coupled to the plurality of memory banks for providing addressing signals to the memory banks on a cache line basis and for providing subsequent cache lines to subsequent memory banks.

1/22

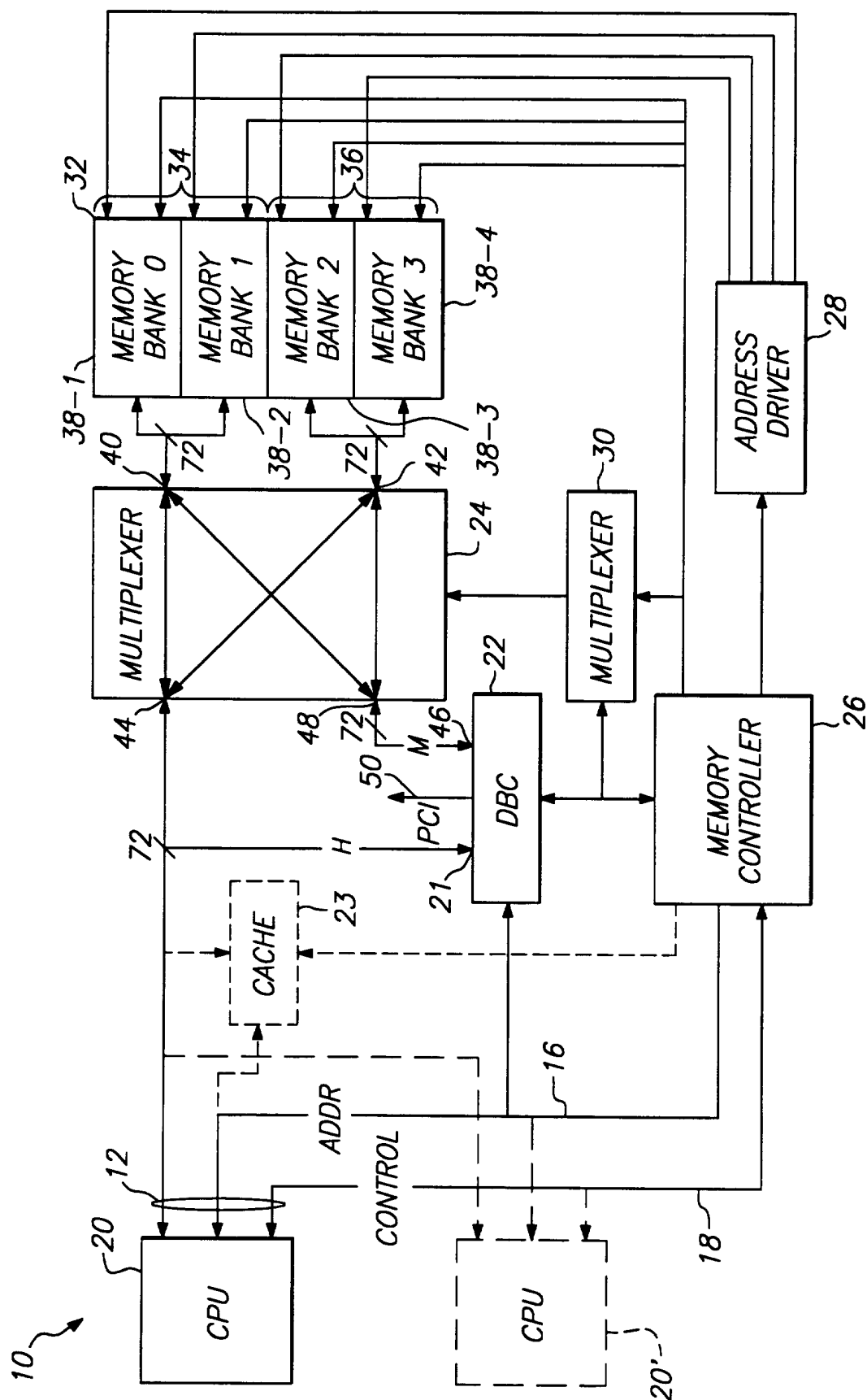
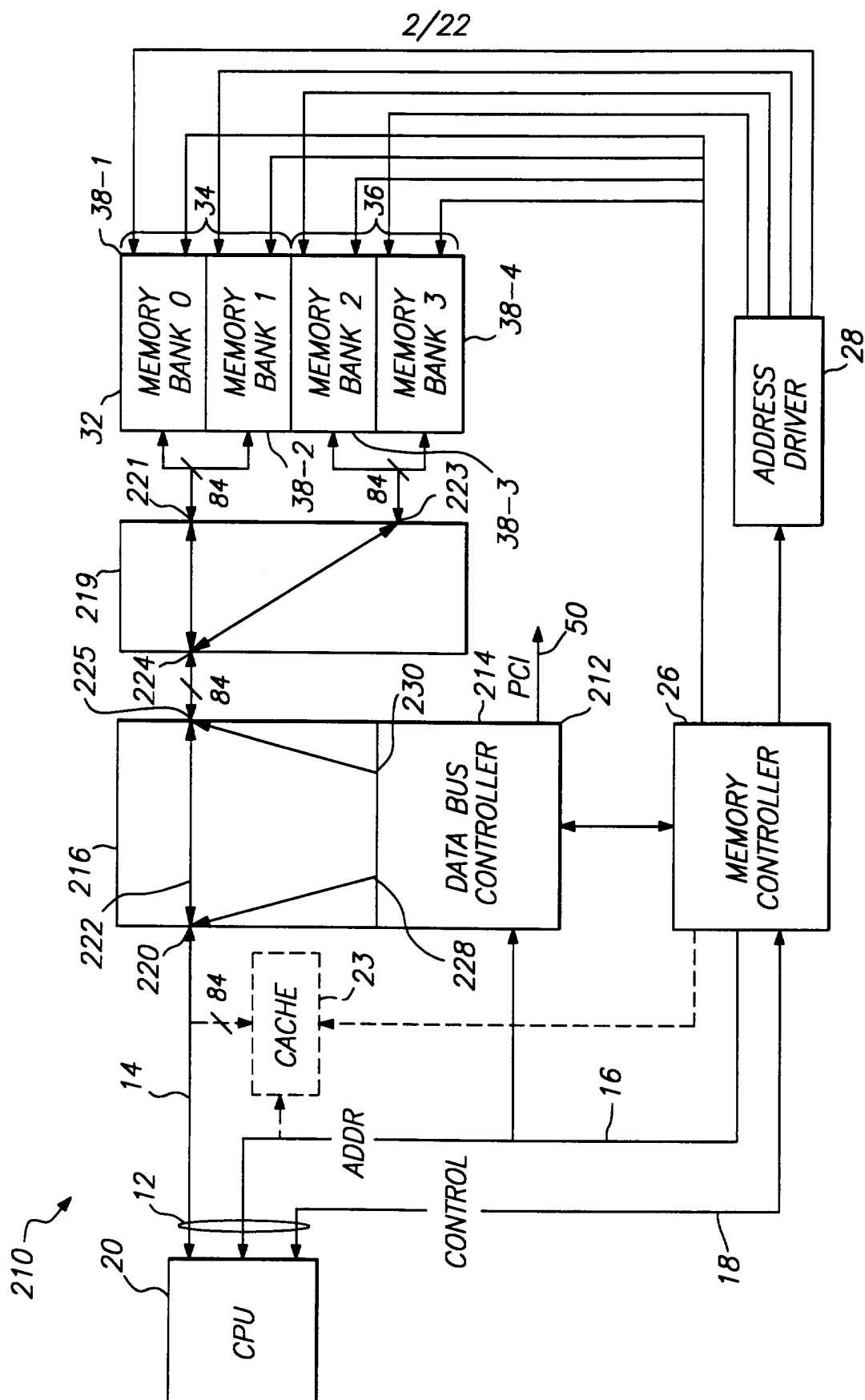
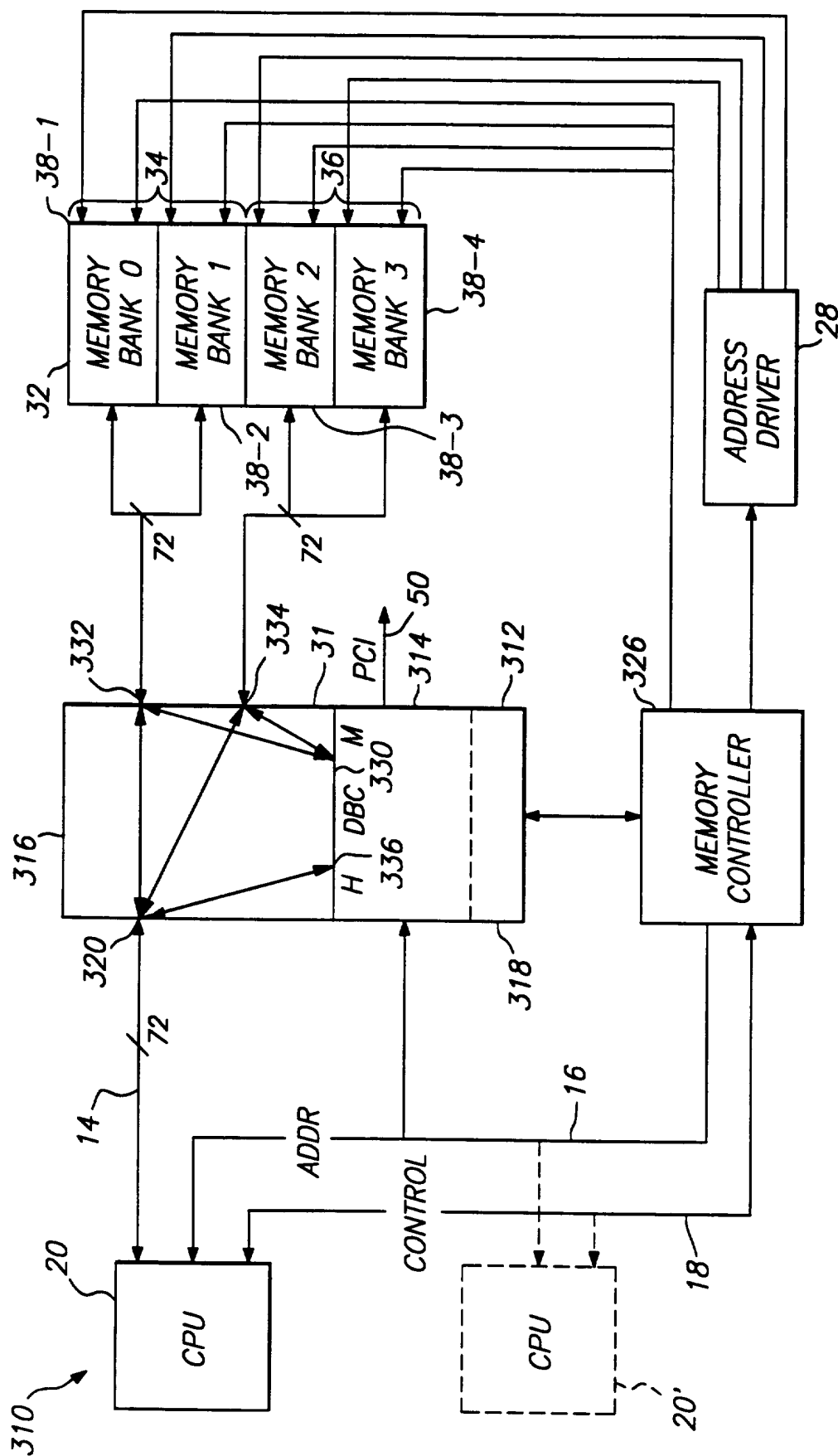


FIG. 1







**FIG. 3**

4/22

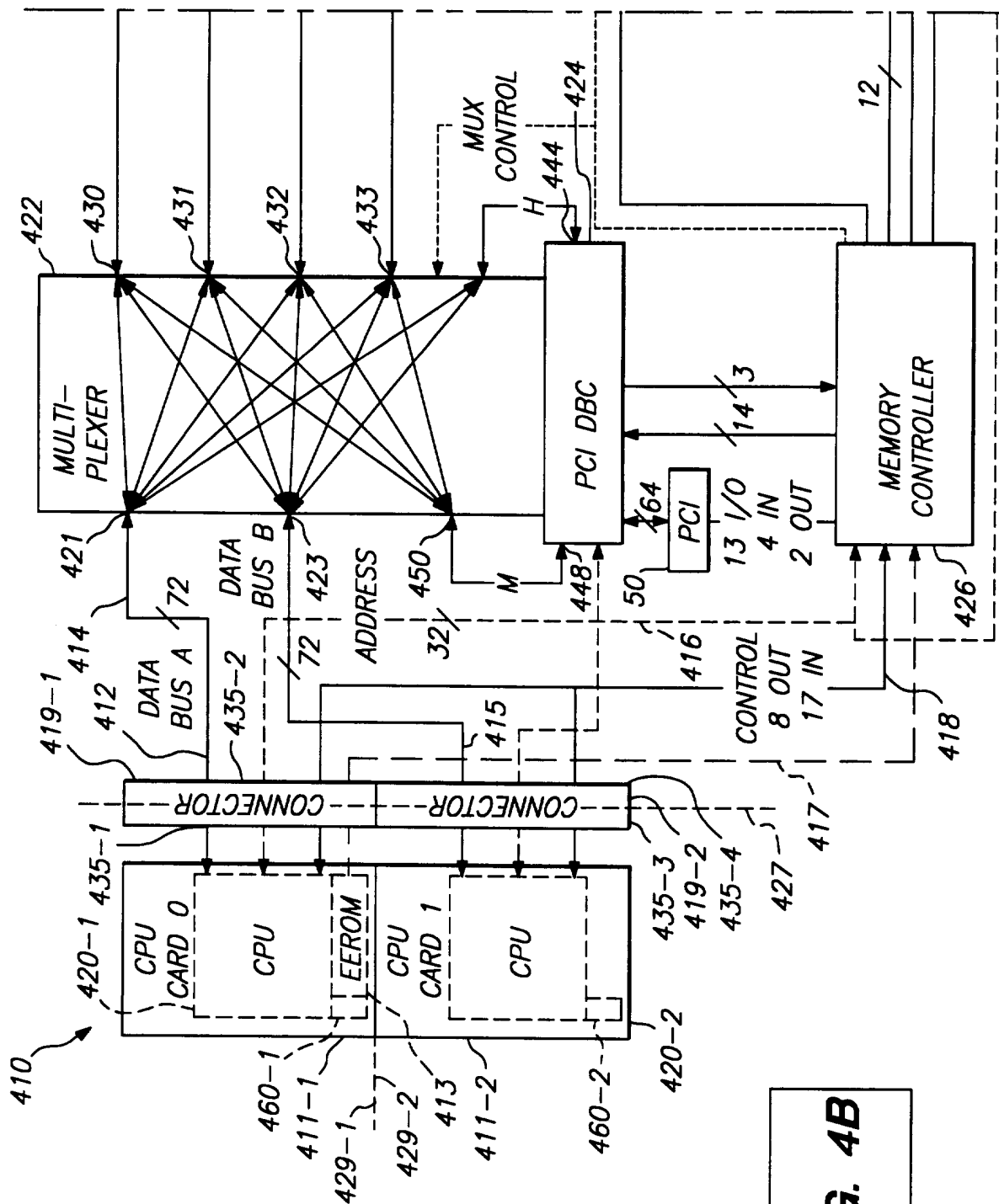


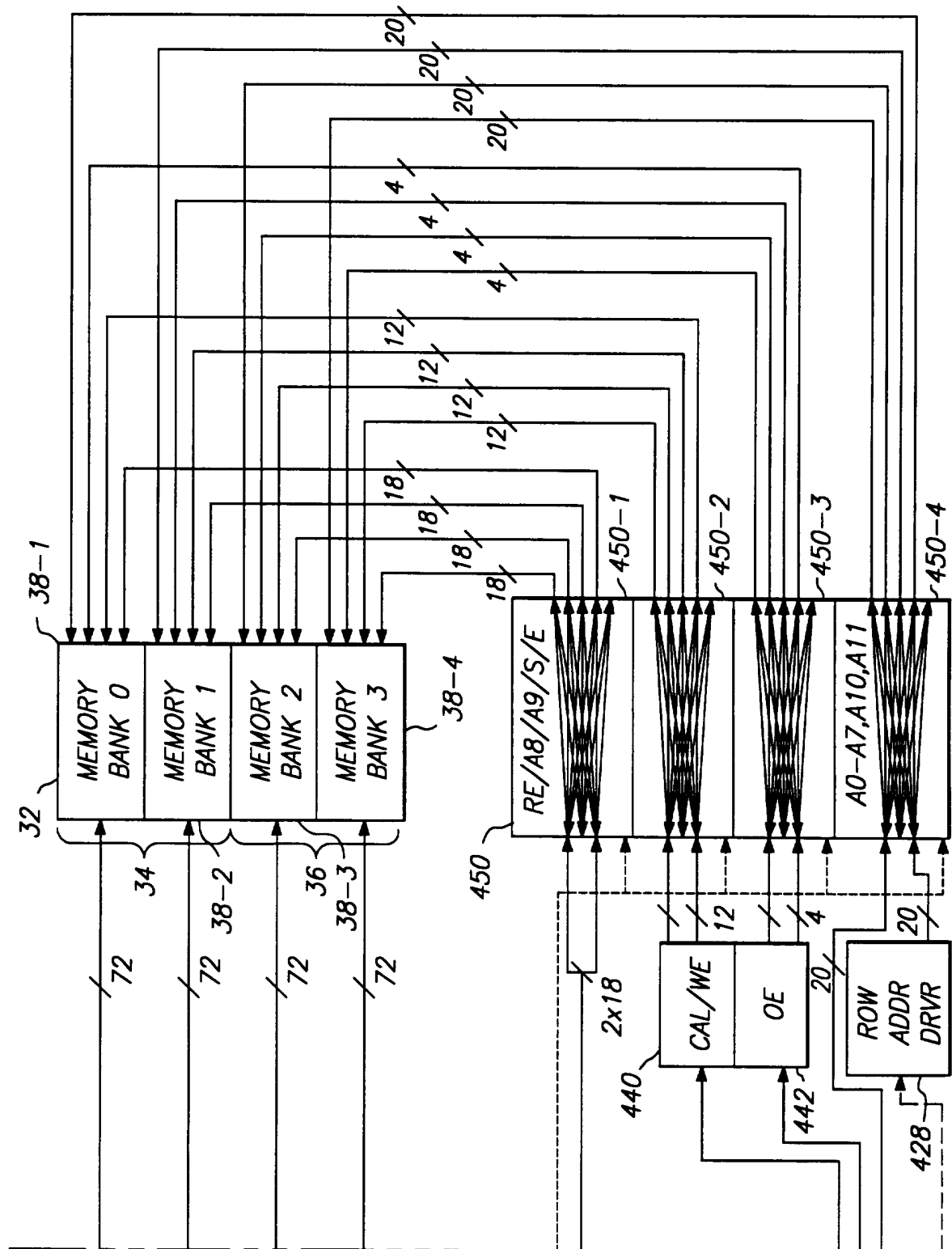
FIG. 4A

FIG. 4A FIG. 4B

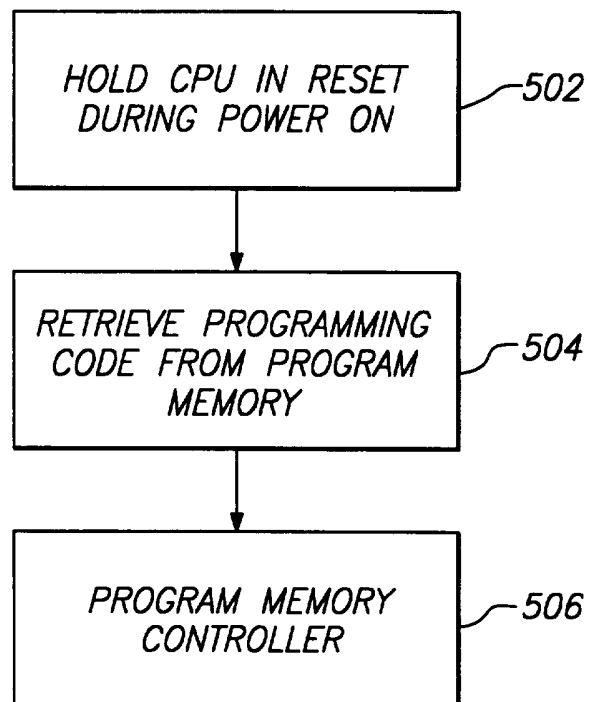
FIG. 4

5/22

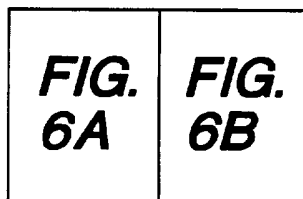
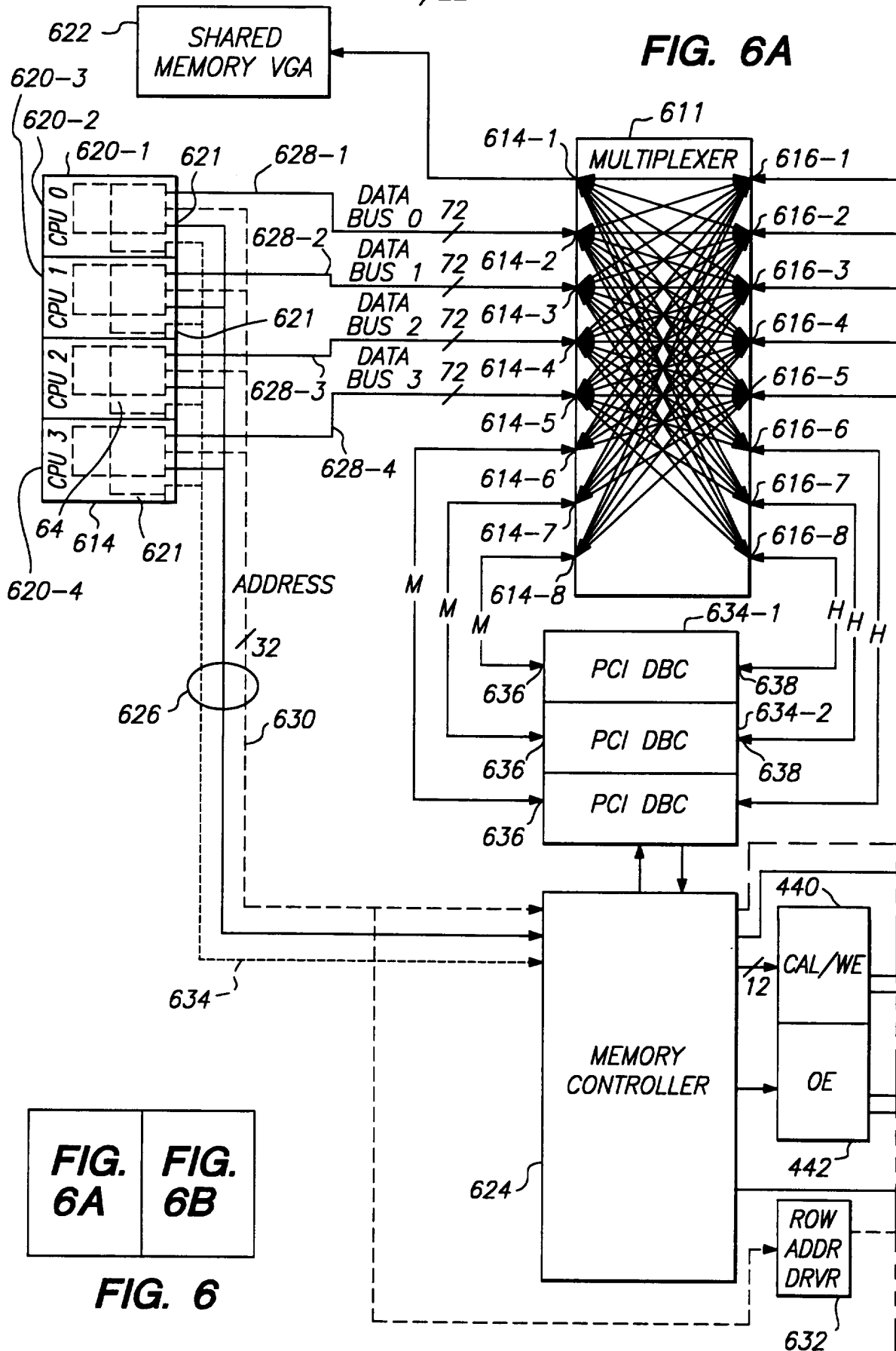
**FIG. 4B**

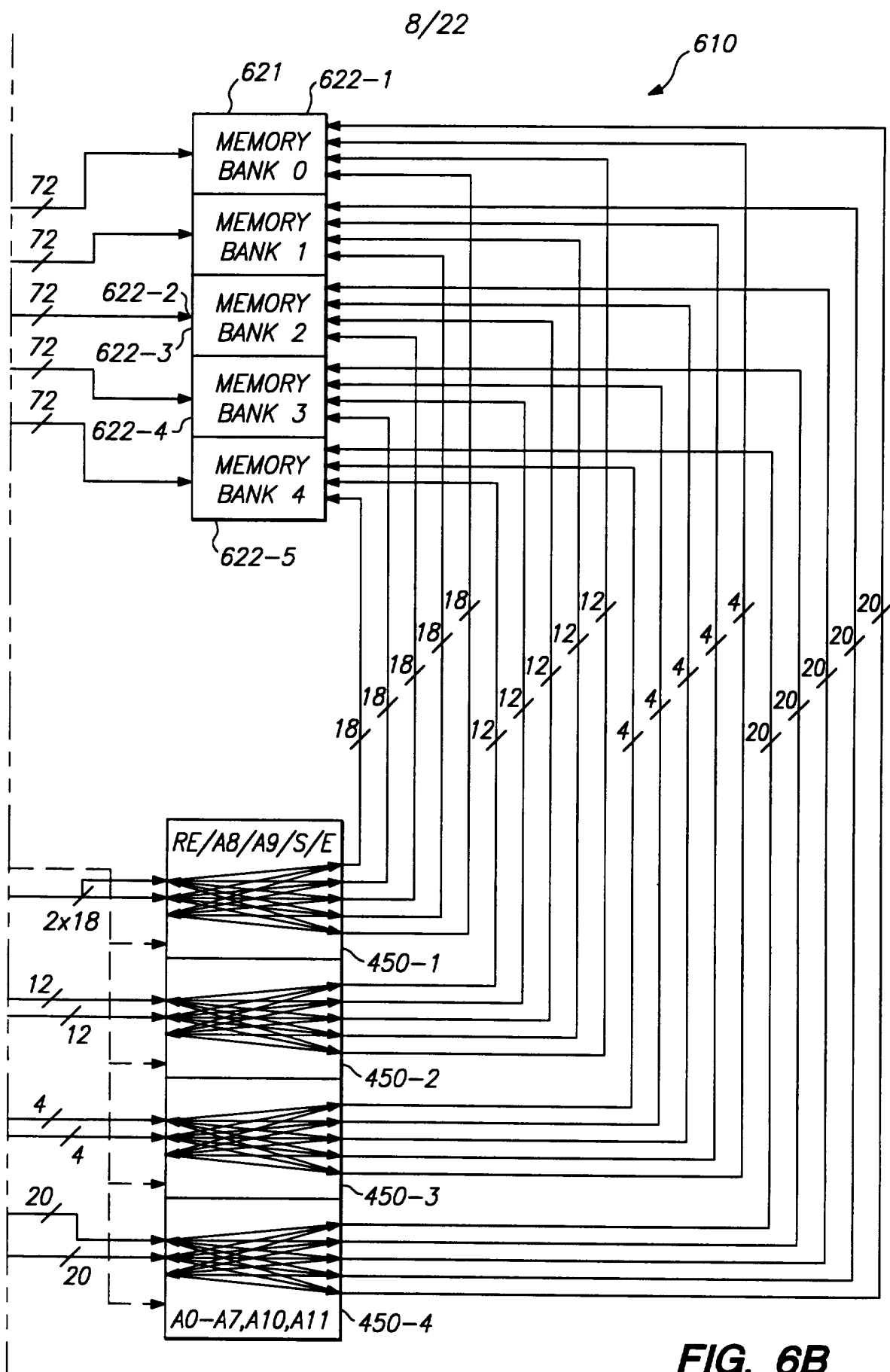


6/22

**FIG. 5**

7/22

**FIG. 6A****FIG. 6**

**FIG. 6B**

9/22

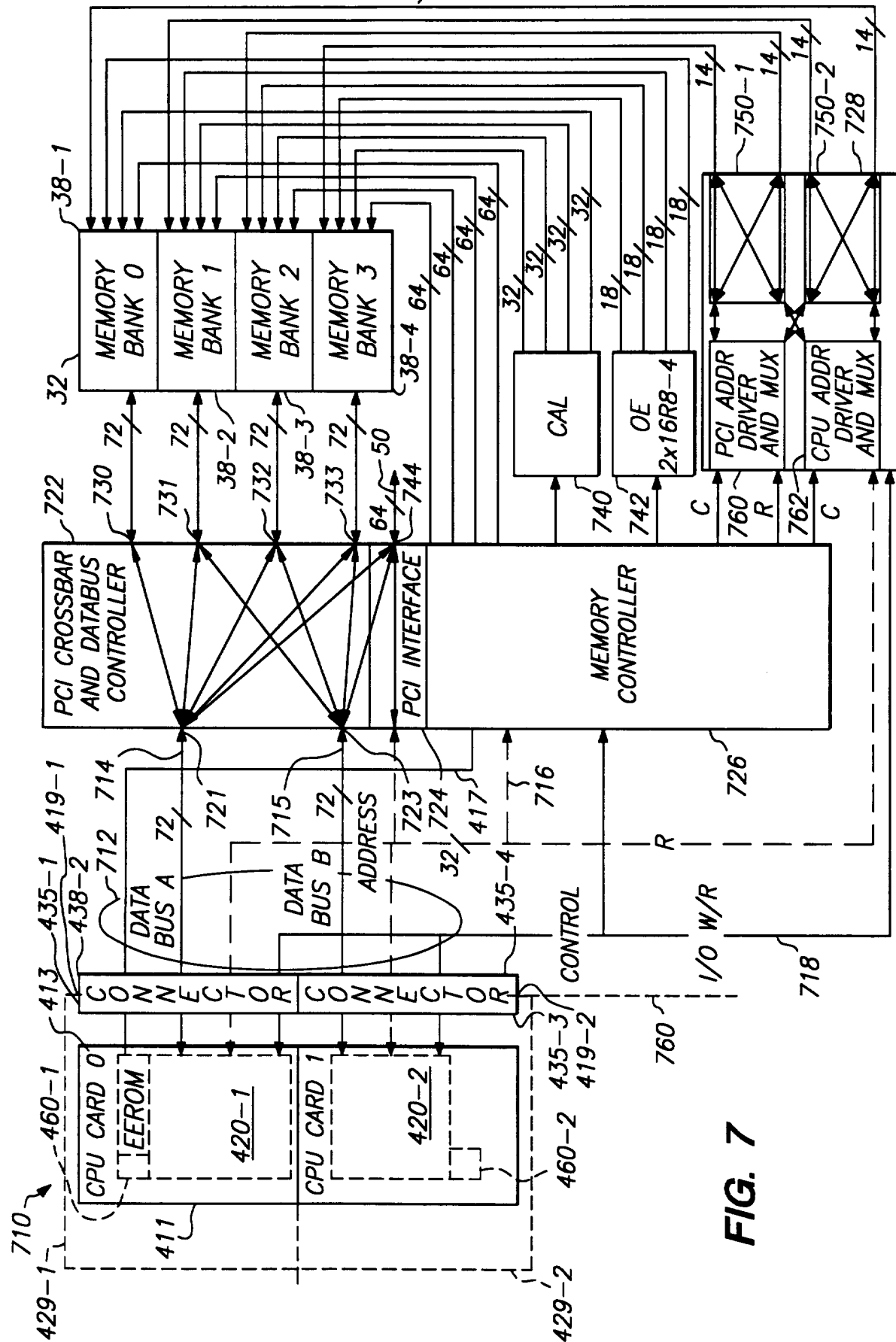
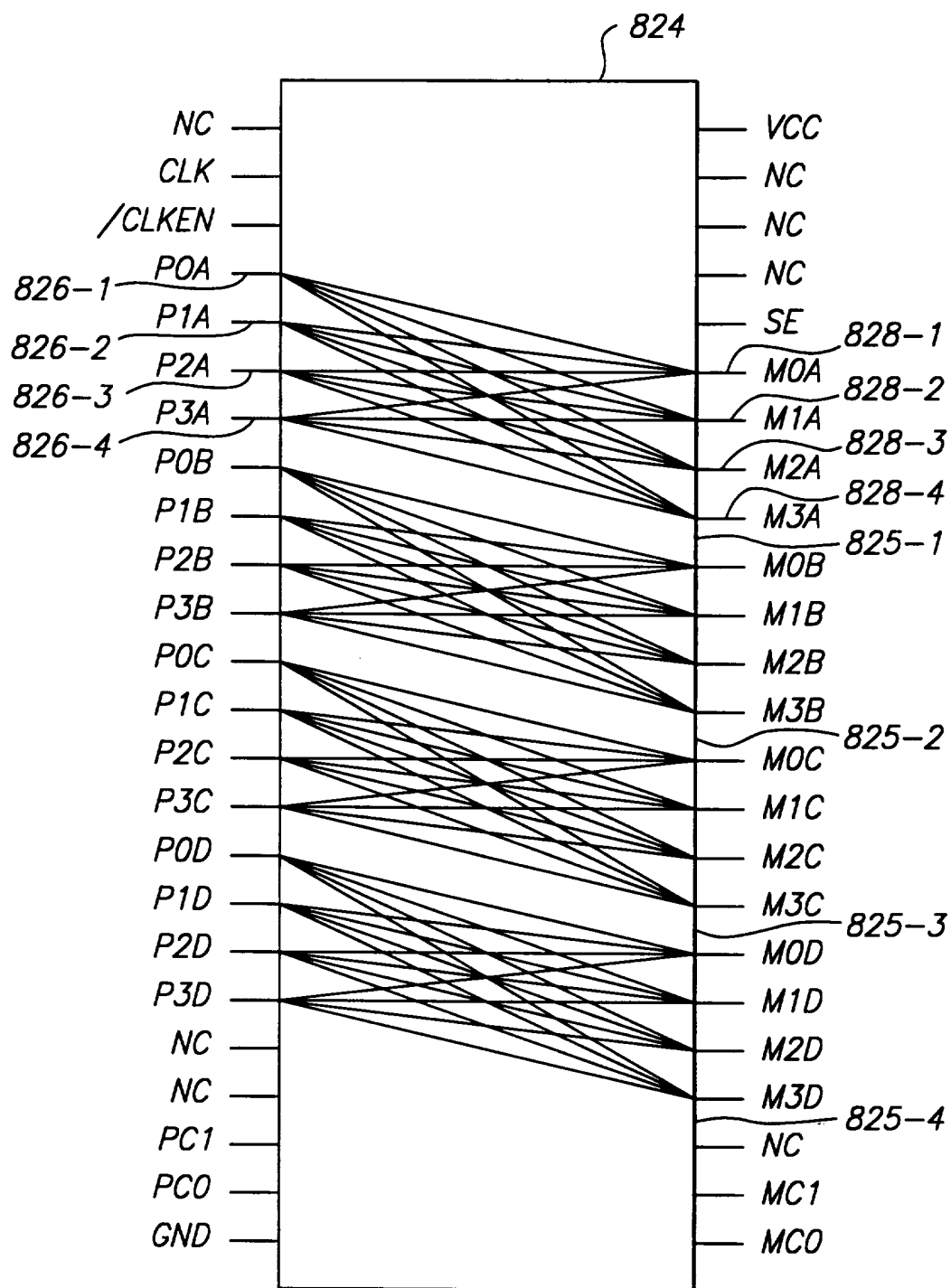


FIG. 7

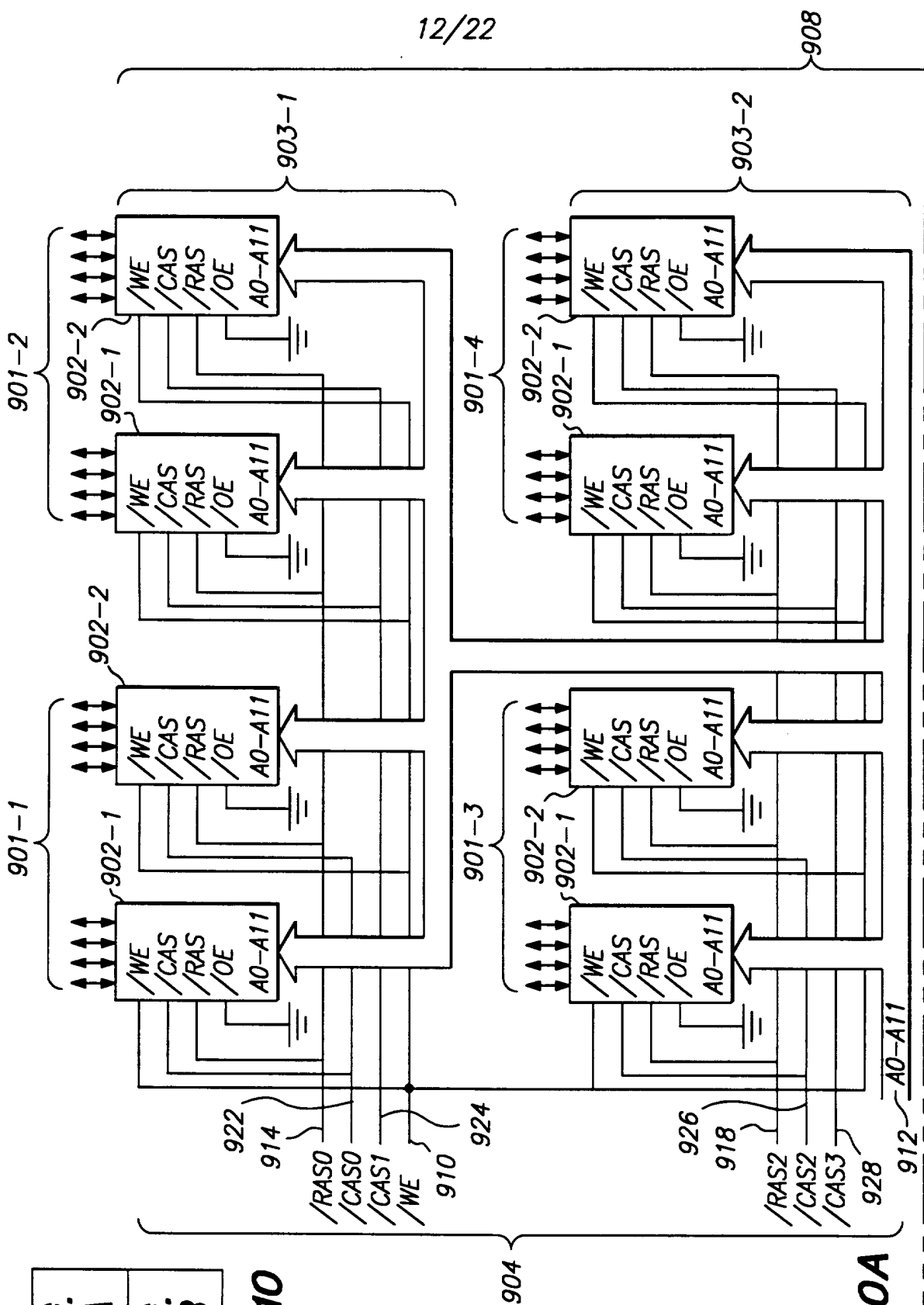
10/22

**FIG. 8**





12/22



**FIG. 10A**

**FIG. 10B**

**FIG. 10**

**FIG. 10A**

13/22

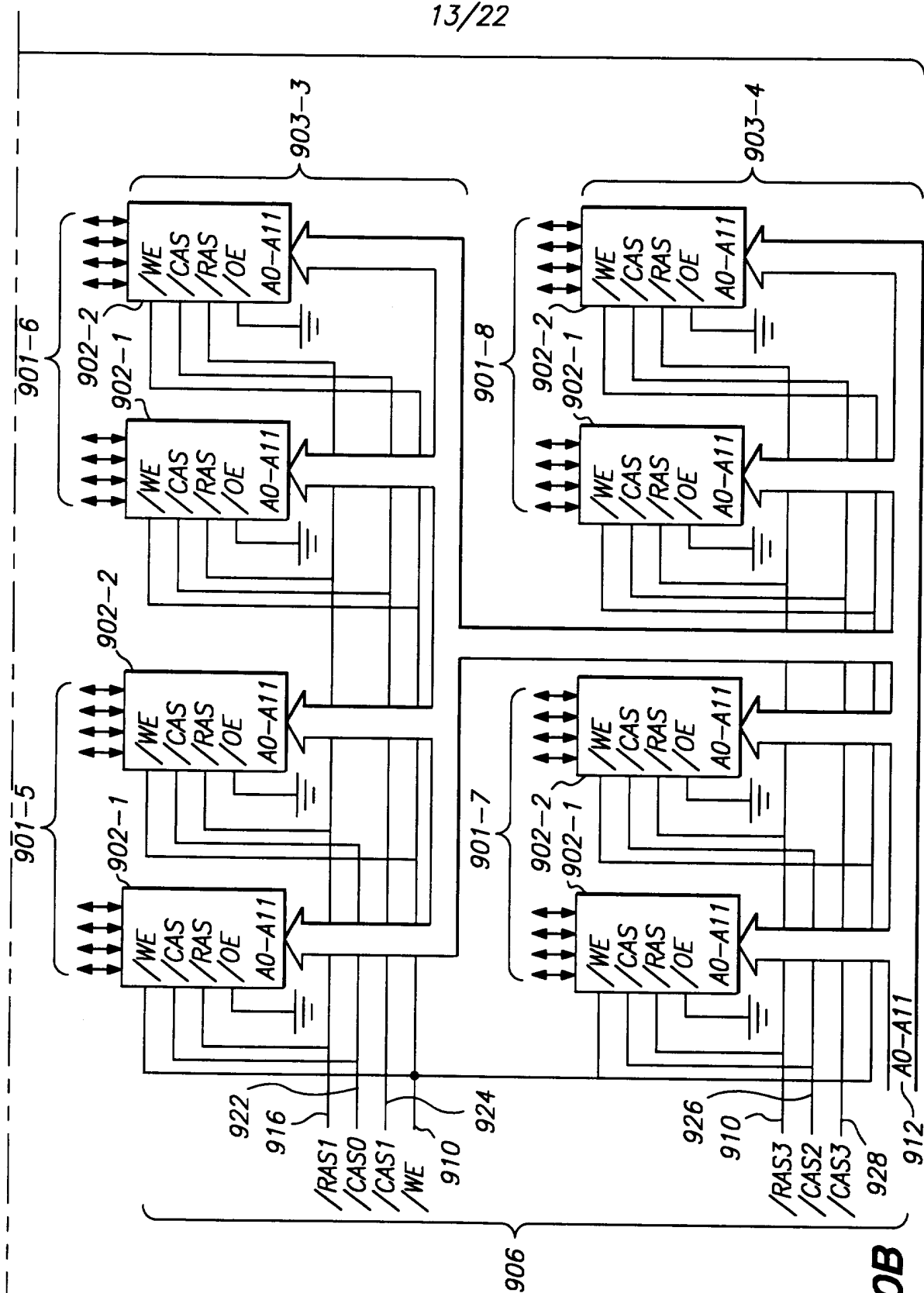


FIG. 10B



15/22

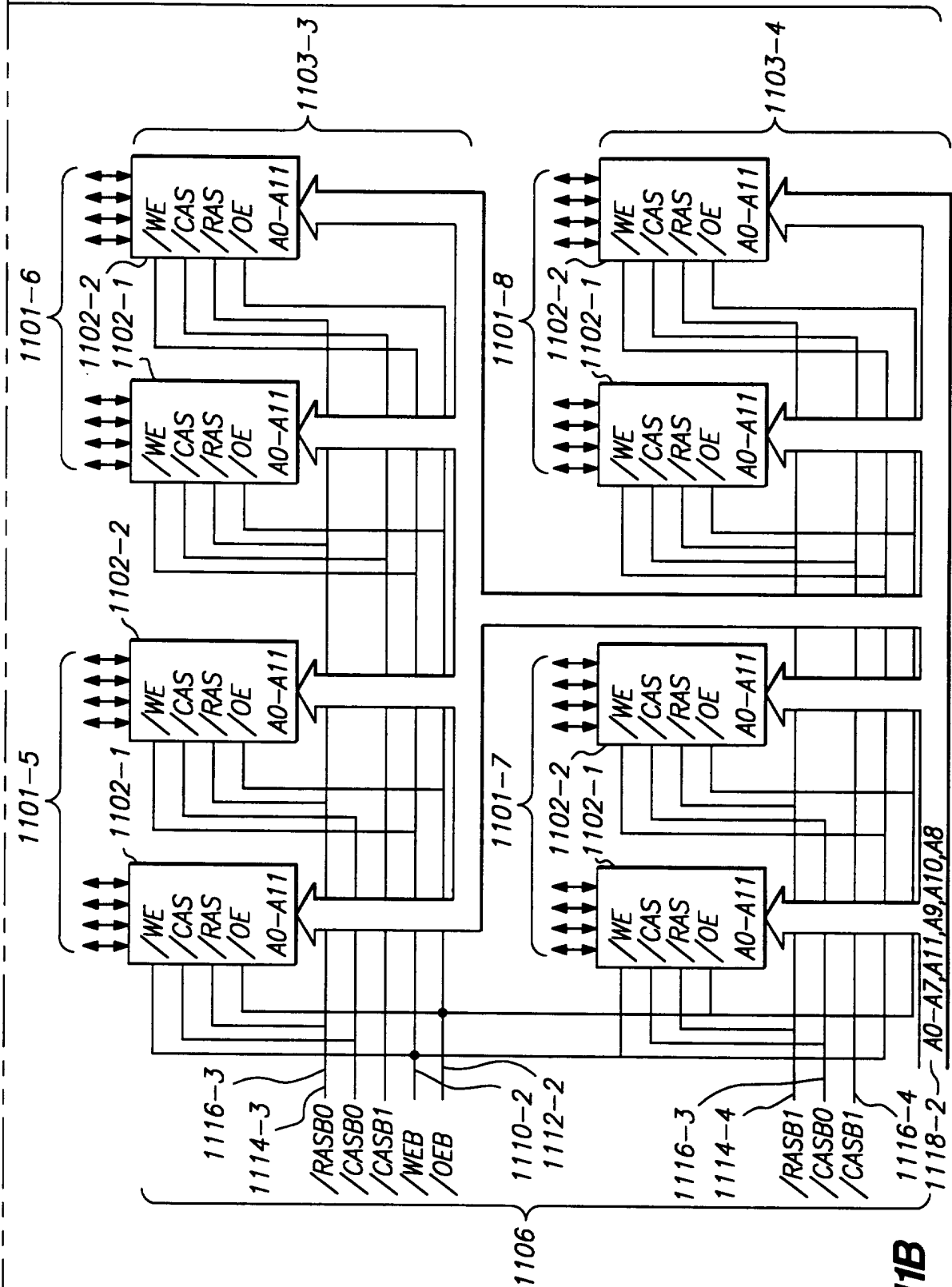


FIG. 11B



FIG. 12A

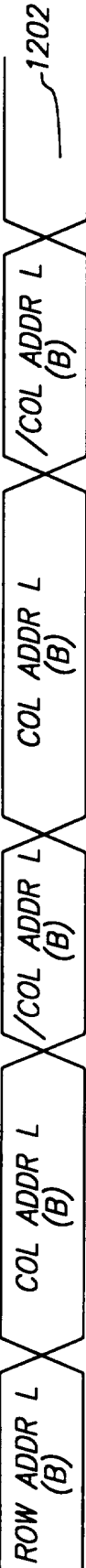


FIG. 12B

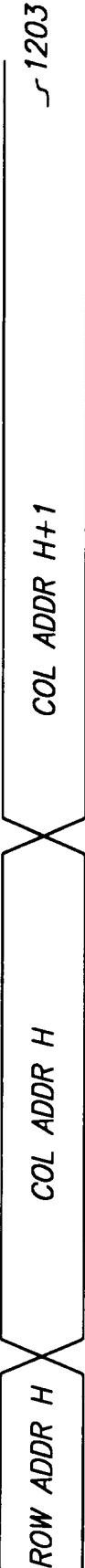


FIG. 12C

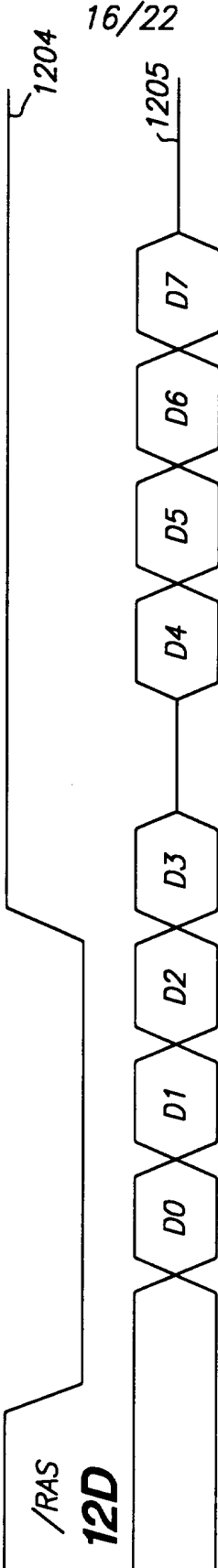


FIG. 12D

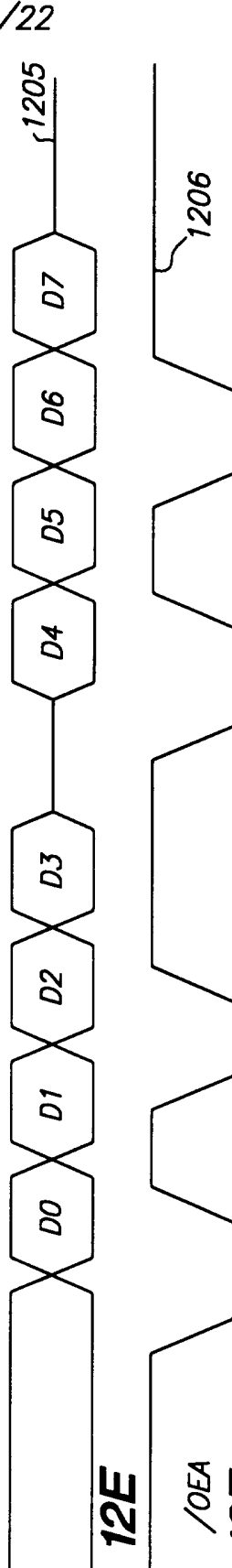


FIG. 12E

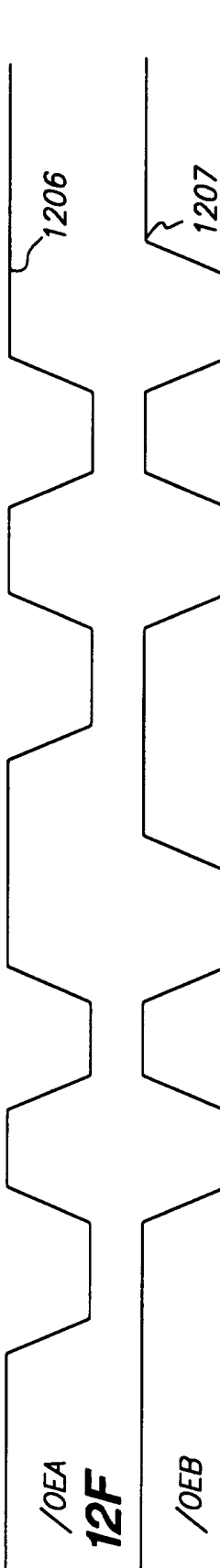


FIG. 12F

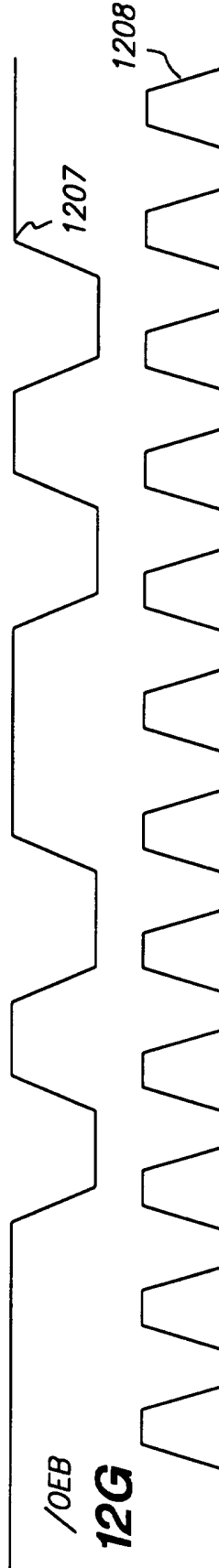


FIG. 12G

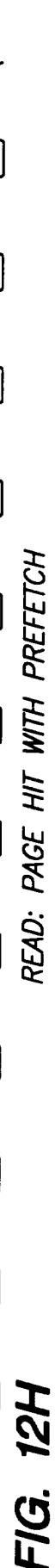


FIG. 12H

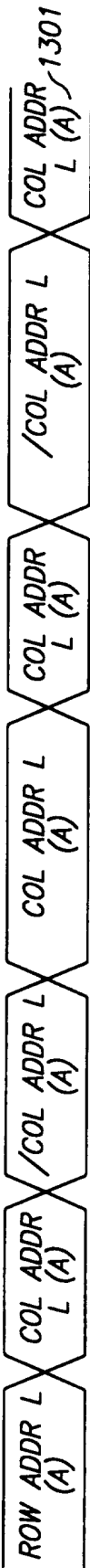


FIG. 13A



FIG. 13B



FIG. 13C

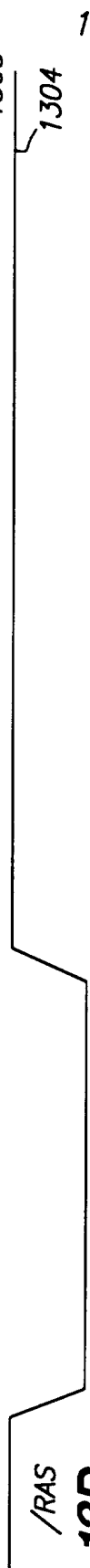


FIG. 13D



FIG. 13E



FIG. 13F

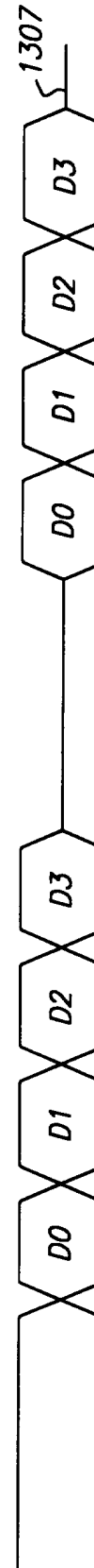


FIG. 13G

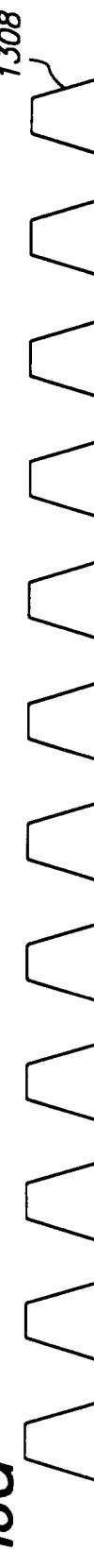
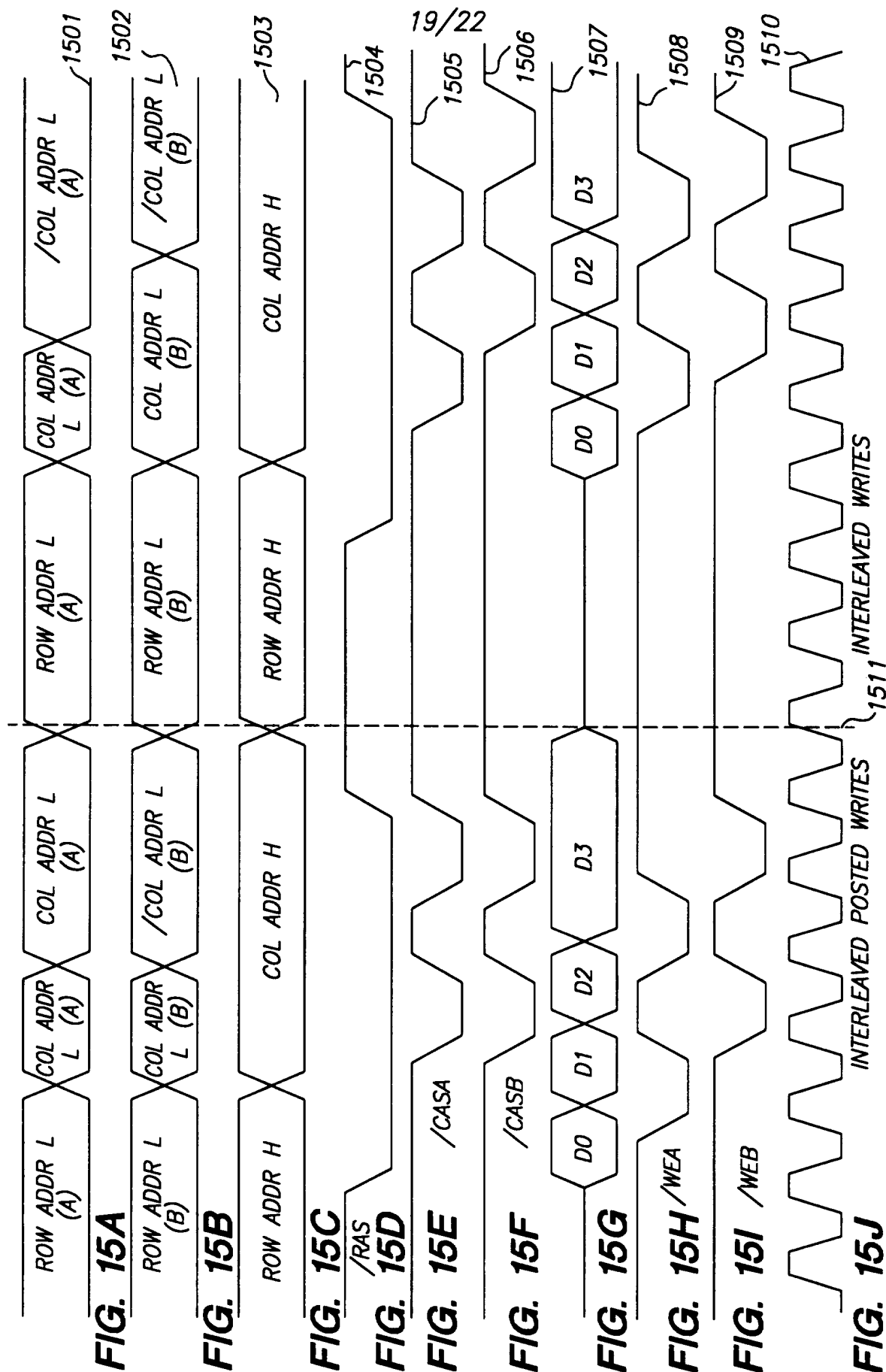


FIG. 13H

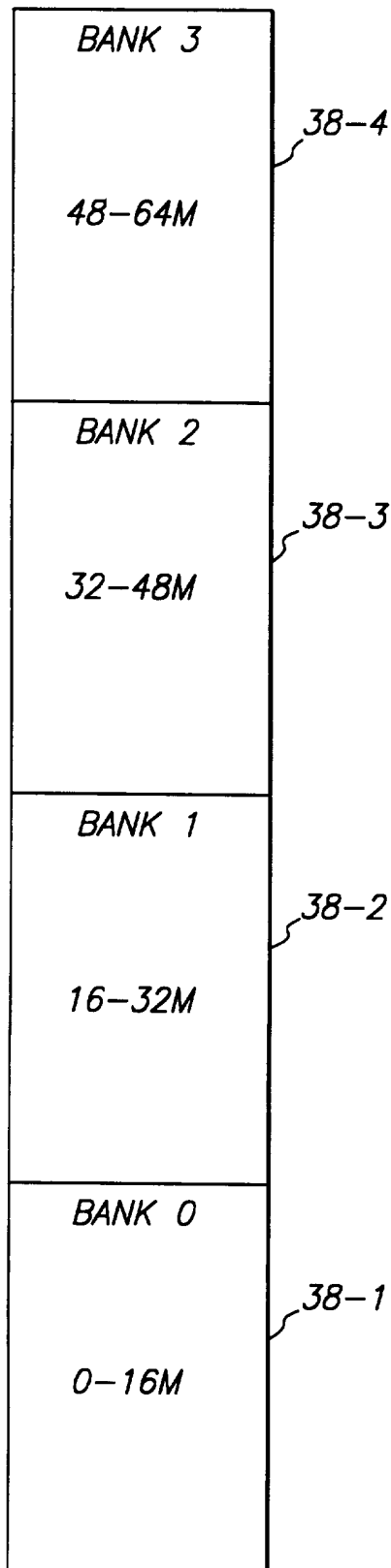
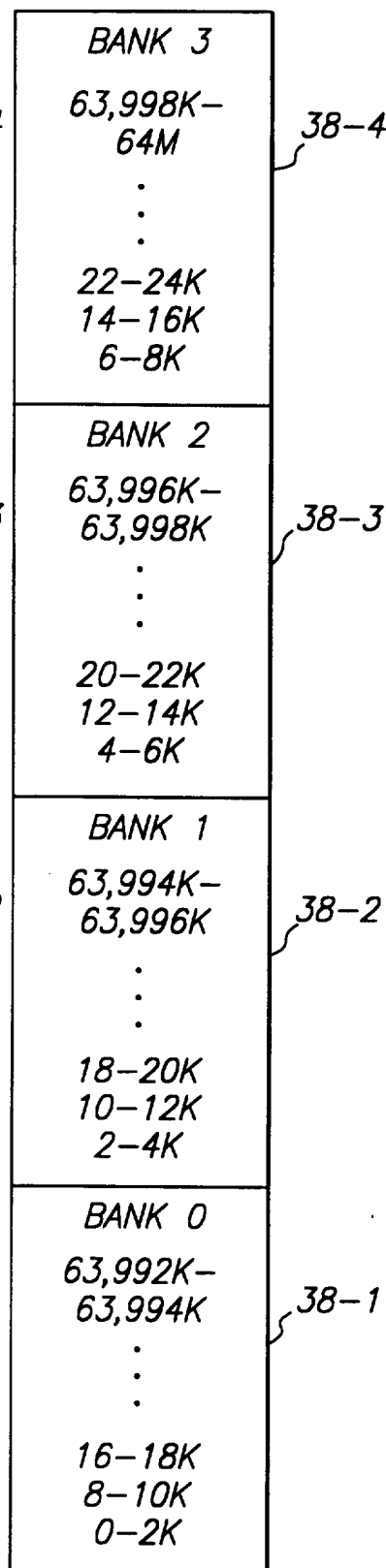
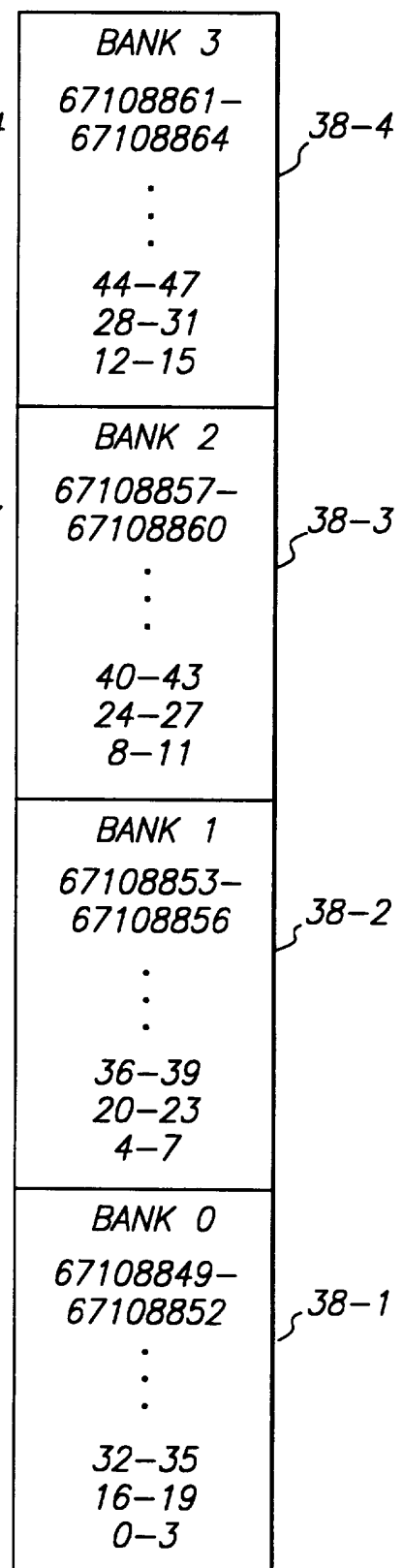
READ: PAGE HIT COLUMN MISS



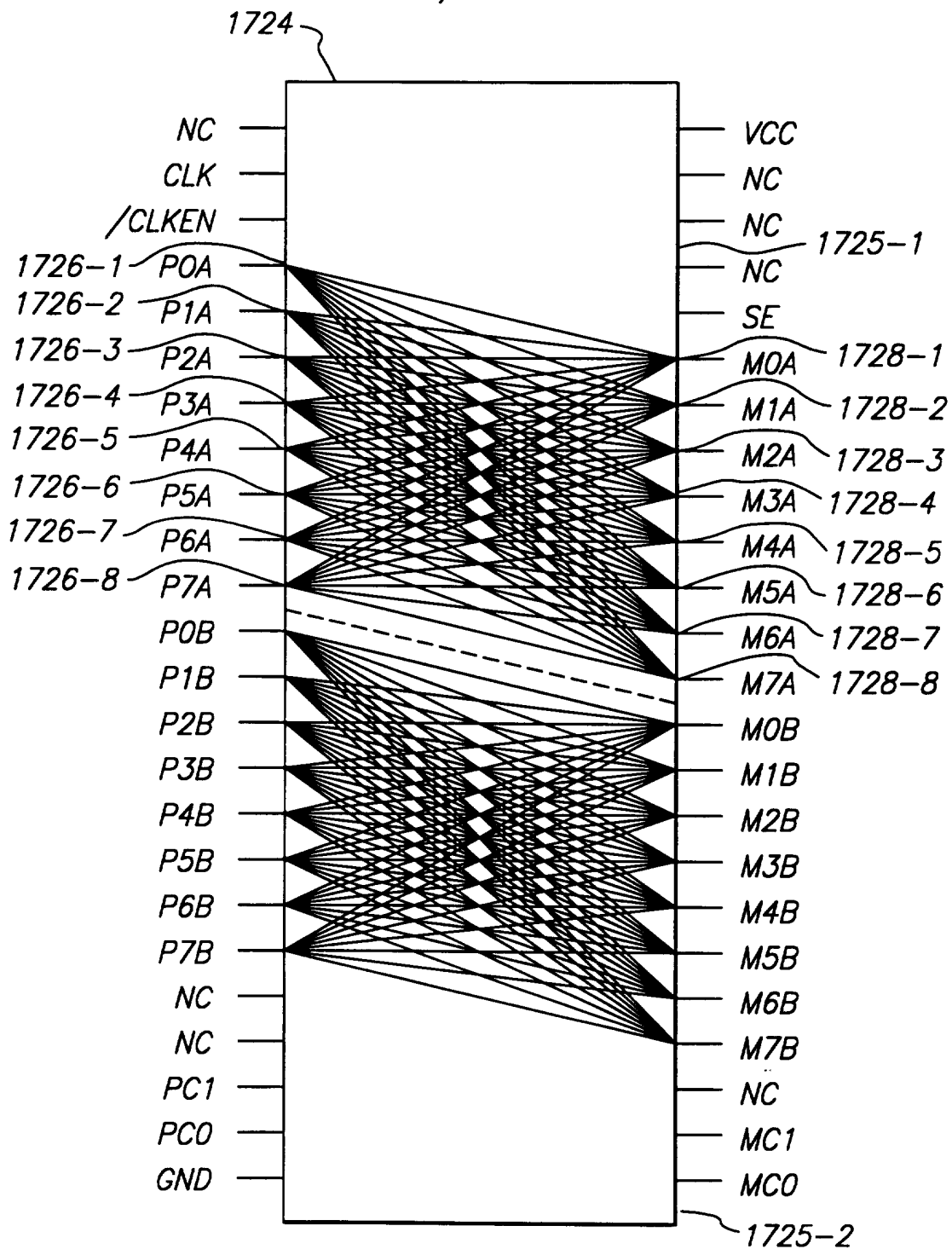




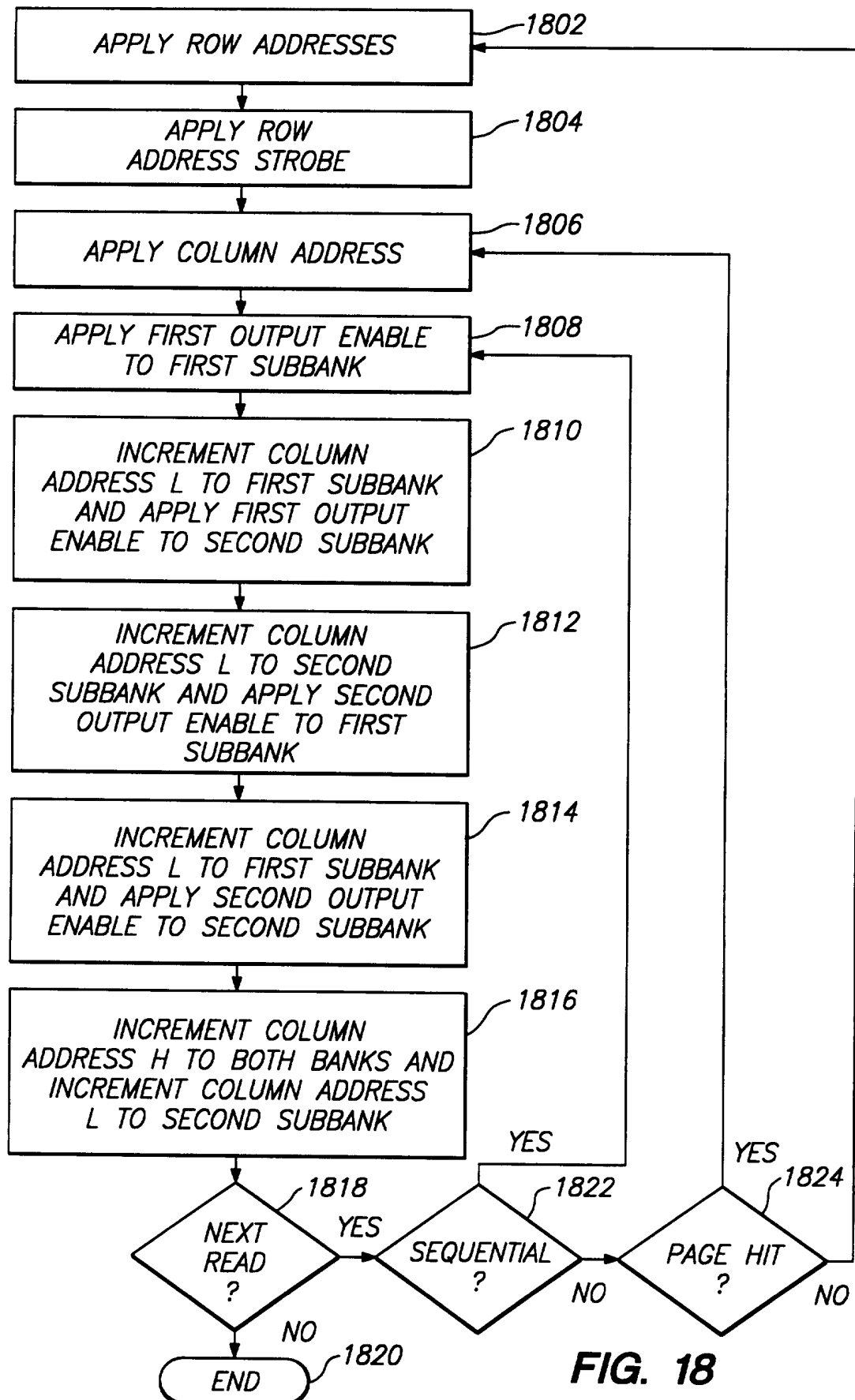
20/22

**FIG. 16A****FIG. 16B****FIG. 16C**

21/22

**FIG. 17**

22/22



## INTERNATIONAL SEARCH REPORT

 International application No.  
PCT/US96/04352

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 12/00

US CL : 395/403, 405, 421.01; 365/230.02, 230.03

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/403, 405, 421.01; 365/230.02, 230.03

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US, A, 5,386,511 (MURATA ET AL.) 31 January 1995, see figures 2 and 3.	1-4 and 6
A	US, A, 4,652,993 (SCHEUNEMAN ET AL.) 24 March 1987, entire document.	5



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
*A* document defining the general state of the art which is not considered to be part of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
*E* earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
*L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*Z* document member of the same patent family
*O* document referring to an oral disclosure, use, exhibition or other means	
*P* document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

22 JULY 1996

Date of mailing of the international search report

14 AUG 1996

 Name and mailing address of the ISA/US  
Commissioner of Patents and Trademarks  
Box PCT  
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

JACK LANE

Telephone No. (703) 305-0000