

1 579 061

- (21) Application No. 6145/77 (22) Filed 14 Feb. 1977 (19)
 (31) Convention Application No. 658 113 (32) Filed 13 Feb. 1976 in
 (33) United States of America (US)
 (44) Complete Specification published 12 Nov. 1980
 (51) INT. CL.³ G06F 13/06
 (52) Index at acceptance
 G4A 12D 12N 12P 13E 2E MX



(54) IMPROVEMENTS IN OR RELATING TO RANDOM ACCESS MEMORY
 MODULES FOR DIGITAL DATA PROCESSING SYSTEMS

(71) We, DIGITAL EQUIPMENT CORPORATION, a corporation organised under the laws of the State of Massachusetts, United States of America, of 146 Main Street, Maynard, Massachusetts 01754, United States of America, do hereby declare the invention, for which we pray that a patent may be granted to us, and the method by which it is to be performed, to be particularly described in and by the following statement:—

This invention relates to the combination of a digital data processing system and a random access memory module. Our corresponding Patent Applications No. 4 5422/78 (Serial No. 1,579,062), No. 7930868 (Serial No. 1,579,063) and 7930869 (Serial No. 1,579,064) which are divisional applications from this case, all contain subject matter which is common to the present specification.

In general a digital data processing system has three basic elements: namely, a central processor, a memory arrangement for storing programs and data for manipulation by the central processor, and input/output units for communicating with the central processor and memory arrangement. One important aspect of a memory arrangement is the dependence of the overall speed of the system on the operation of the memory arrangement. For example, if a memory arrangement performs a memory cycle to retrieve data from a location therein in 1.2 microseconds (i.e., its characteristic retrieval interval), the minimum time required to process an instruction is 1.2 microseconds. As two or more memory cycles are often required in order to process an instruction, the average time for processing each of successive instructions will be greater than the 1.2 microsecond characteristic retrieval interval.

Thus, in order to enable the data processing system to operate at the greatest possible speed, it is necessary to minimize the characteristic retrieval interval for the memory arrangement. However, memory

cost generally increases as the characteristic retrieval interval is reduced. Therefore, the entire memory arrangement usually can not be constructed from fastest elements because its cost would be prohibitively high.

A memory arrangement for a data processing system thus may contain several types of memory units that have diverse characteristics. Each type has particular characteristics that makes it suitable for specific applications. Generally a digital data processing system includes a random access memory unit. The time required to obtain information from a random access memory unit is independent of the location of the information most recently obtained. Such memory units have included magnetic cores or solid state devices as the storage elements. Magnetic core memory units are very popular because they are reliable and retain data even in the absence of electrical power. Their characteristic retrieval intervals are usually measured in terms of one or two microseconds. Semiconductor random access memory units are considerably faster than magnetic core memory units and have characteristic retrieval intervals on the order of hundreds of nanoseconds. However, they require constant electrical power to retain data. If the electrical power is interrupted, the contents are lost. Both types of random access memory units have similar costs and in some data processing systems they constitute the entire memory arrangement.

Direct access memory units include, conventionally, disk and drum memory units. Their characteristic retrieval intervals are slower than random access memory units and are measured usually in terms of milliseconds. However, they are able to store relatively large amounts of data at costs that are significantly less than the costs of storage in random access memory units. Generally a direct access memory unit is used to supplement the random access memory unit. In many memory arrangements, the contents of the two memory units are constantly being interchanged in order to provide the most rapid data processing

operations with a minimum random access memory unit storage capability.

Sequential access memory units, such as magnetic tape memories, constitute another type of memory unit. These memory units often are used for archival storage or to provide a "copy" of the contents of the random access and direct access memory units. They have even longer characteristic retrieval intervals than direct access memory units have, but the storage costs again are significantly lower.

As previously indicated, a particular memory arrangement in a data processing system may include two or more of these memory units in combination. A typical memory arrangement might include a magnetic core random access memory unit and a direct access memory unit.

A more recent configuration includes both semiconductor random access memory units and magnetic core access memory units. One such system is described in U.S. Patent Specification No. 4,016,541. In this system the central processor normally communicates with various input/output devices and a magnetic core random access memory unit over an asynchronous bus. However, the central processor also transfers data to and from a semiconductor memory unit over a second bus to provide increased operating speeds. Typically the semiconductor memory unit has sufficient capacity to store a complete program or a significant portion of such a program.

All the foregoing memory units transfer data to or from locations identified by address signals. Another type of memory unit is an associative, or content-addressable, memory unit. In such a memory unit a location is selected on the basis of what it contains and not on the basis of its address. An associative memory unit is useful in several applications. For example, in a machine that utilizes virtual addresses, each location in the memory unit contains both a virtual address and the corresponding address for the actual location. Such a system is described in U.S. Patent Specification No. 3,893,084.

Other systems combine an associative memory unit as a "cache" memory unit for storing small portions of a program. Each time the central processor issues a memory address during a memory retrieval cycle, the associative memory unit searches to determine whether an address storage location contains that address. If it does, data in a corresponding data storage location in that memory unit is immediately transferred to the central processor. If it does not, the contents of that location in the random access memory unit identified by the address as well as data from a block of successive locations are transferred to the associative

memory unit. This memory arrangement can improve the overall speed of the memory whenever the data is contained in the associative memory unit, because the memory operates with the speed of a semiconductor memory.

Given a specific combination of memory units, shorter characteristic retrieval intervals also can be attained by other methods. For example, the memory arrangement may include memory units that are divided into separate elements or banks so that sequential memory cycles in different elements can be "overlapped". One such memory arrangement that is described in the foregoing U.S. Patent 3,810,110, contains memory units of different characteristic retrieval intervals, all connected to a common memory bus. However, separate memory retrieval control signals are required, so the operating programs for the central processor must determine which memory retrieval control signal is to be sent to initiate each memory cycle.

Therefore it is an object of this invention to provide a random access memory module in combination with a data processing system that enables overlapped transfers without the need for multiple control signals.

Accordingly the invention consists in the combination of a digital data processing system and a random access memory module connected via a bus to the digital data processing system that, for each of successive memory cycles including reading and writing memory cycles initiates a data transfer by transmitting binary address signals and asynchronous control signals that control the transfer of data between said random access memory module and the digital data processing system, said random access memory module comprising:

A. addressable storage means that store digital data in addressable storage locations therein,

B. address decoding means that produce an enabling signal when said address decoding means receives, from the digital data processing system, binary address signals that identify a said storage location in said addressable storage means, and

C. control means connected to said addressable storage means and said address decoding means, said control means including:

i. means that respond to the enabling signal and respective asynchronous control signals to initiate a reading or writing memory cycle, each such reading memory cycle including a reading operation during which digital data is destructively read from an identified one of said locations in said addressable storage means and a subsequent writing operation during which data is restored to said identified location in said addressable storage means, and

ii. means that transmit to the digital data processing system a BUS OCCUPIED signal during the reading operation of each reading memory cycle, the BUS OCCUPIED signal temporarily inhibiting the digital data processing system from transmitting binary address signals and asynchronous control signals to initiate successive memory cycles.

In order that the invention may be more readily understood, reference will now be made to the accompanying drawings, in which:—

Figure 1 is a block diagram of an embodiment according to the present invention comprising the combination of a data processing system with a random access module,

Figure 2 indicates the control signals which pass between the central processor and the associative memory shown in Figure 1;

Figure 3 illustrates the control signals which pass between a main bus mapping circuit and the associative memory shown in Figure 1;

Figure 4 illustrates the signals which pass between a high speed controller and the associative memory shown in Figure 1;

Figure 5 depicts the signals which pass between a random access memory module and the associative memory shown in Figure 1;

Figure 6 is a block diagram of a memory management portion of the central processor shown in Figure 1;

Figure 7 is a block diagram of the main bus mapping circuit shown in Figure 1;

Figure 8 is a block diagram of the associative memory shown in Figure 1;

Figure 9 is a detailed block diagram of portions of the associative memory shown in Figure 8;

Figure 10 is a detailed block diagram of a portion of the control circuitry for the associative memory in Figure 8 and includes Figures 10A, 10B and 10C;

Figure 11 illustrates a memory controller used in a memory module shown in Figure 1;

Figure 12 is a detailed circuit diagram of start memory circuit shown in Figure 11;

Figure 13 is a detailed circuit schematic of a sense decision circuit shown in Figure 11;

Figure 14 is a detailed circuit schematic of a margin decoding circuit in Figure 11;

Figure 15 is a detailed timing diagram that illustrates the timing signals that are transferred between a memory controller and a memory stack in Figure 11 during a transfer from the associative memory to the memory module;

Figure 16 is a timing diagram that illustrates the timing signals that are transferred between a memory controller and a memory stack in Figure 11 during a transfer from

the associative memory to the memory module;

Figure 17 is a timing diagram that illustrates the timing signals that are transferred between a memory controller and a memory stack during an exchange of the contents of a location in the associative memory and the contents of the location in memory module.

Figure 18 is a detailed circuit diagram of an address decoder shown in Figure 7 and including Figures 18A and 18B;

Figure 19 is a detailed circuit description of a limit comparator circuit shown in Figure 7; and

Figure 20 is a simplified diagram of the circuitry associated with one data bit position in a memory stack and the corresponding elements in the memory transceiver of Figure 1.

It will be helpful in the following discussion to define certain terms. A "bit" is a binary bit. A "byte" is a measurable portion of consecutive binary digits, (e.g., an 8-bit byte). A "word" is an ordered set of bytes which occupies one storage location and that is treated by the central processor as a unit. In a specific embodiment of the system shown in Figure 1, the central processor operates with a 16-bit word comprising two 8-bit bytes. In the detailed drawings, a signal is "asserted" when it is at a positive level and "not asserted" when it is at a ground potential. However, data processing systems conventionally used "ground assertion" standards on interconnecting buses, so we include inverters to provide "ground assertion" signals on a bus and, positive assertion signals for the other circuits. These inverters are shown but not identified in the drawings.

The phrase "memory cycle" includes a sequence of events occurring when information is transferred to or from a memory. The duration of a memory cycle for retrieving data from that memory is its "characteristic retrieval interval".

Now referring to Figure 1, there is shown one embodiment of the data processing system which incorporates this invention. Basic elements in this system include a processor system 20 that includes a central processor 21, a memory management unit 22, a main bus map unit 23 and an associative memory 24. Peripheral device(s) 25 and terminal(s) 26 are examples of input/output devices which communicate with the central processor 21 over a main bus 27. The construction and operation of the central processor 21 and communications with the peripheral devices 25 and terminals 26 over the main bus 27 are described in the U.S. Patent Specifications 3,614,740, U.S. Patent 3,614,741 and U.S. Patent 3,710,324.

The associative member 24 contains only

a few locations (e.g., 1024) for storing data.

The transfer of data between the central processor 21 and the associative memory 24 is analogous to the transfer described in the U.S. Patent Specification No. 4,016,541. Figure 2 illustrates the corresponding signals and their respective timing sequences. The central processor 21 asserts a BUST signal to initiate a memory cycle of the type indicated by C0 and C1 control bits. After the BUST signal is asserted, the control signals and address signals, and, if a writing memory cycle, the data signals, are transmitted to the associative memory 24. As successive memory cycles may be overlapped, it is possible to initiate a memory cycle and then later terminate it if the program sequence is altered. The central processor 21 transmits to the associative memory a CONTROL, OK signal when the memory cycle is to be continued. Then operations in the central processor terminate until the memory cycle is completed. After a delay during which the writing operation occurs or the data is retrieved, the associative memory 24 transmits back to the central processor 21 an MEM SYNC signal. This signal allows the processor to restart. During a reading memory cycle, the MEM SYNC signal causes the central processor 21 to store the data in a buffer register.

Other signals which pass between the central processor 21 and the associative memory 24 include data Parity signals and a BEND signal which the processor uses to abort a memory cycle initiated by a BUST signal.

Referring to Figure 1, memory modules 30 through 33 constitute the main random access memory and include a plurality of magnetic core storage locations. The memory management unit 22 receives "virtual" addresses from the central processor 21 and expands the number of bits in each address to define actual locations. For example, the unit 22 in one embodiment expands 18-bit addresses from the central processor 21 to 22-bit addresses.

Peripheral device(s) and terminal(s) 25 and 26 also may communicate directly with random access memory in this system. As described in the aforementioned U.S. Patent 3,170,324, transfers over the main bus 27 are performed when the central processor 21 or one of the peripheral devices 25 or terminals 26 gains control of the main bus 27 and becomes a "bus master". Any unit which becomes a bus master contains the necessary circuits for transmitting only the bus address signals to define a "slave unit" and various control signals to initiate a reading or writing operation with the slave unit. However, the addresses transmitted by a bus master to the main bus 27 do not

define a unique location in the memory modules. The main bus map 23 performs the necessary address conversion and it is described in more detail later.

Referring to Figures 1 and 3, a bus master device transmits onto the bus 27, address (A) signals, C0 and C1 direction control (C) signals and, if the transfer is to the memory, data (D) signals. Thereafter the bus master transmits a MSYN signal that causes the main bus map 23 to transmit an ENBUS signal and a ground assertion UB REQUEST signal that initiate the memory cycle identified by C0 and C1 signals from the main bus 27. Once the associative memory 24 initiates a memory cycle, it transmits a ground assertion ACKN signal which shifts the UB REQUEST signal to a non-asserted or Positive level. After the memory cycle is completed, the associative memory transmits a DONE signal to indicate that the memory cycle has been completed. In the case of a transfer to the bus master, the main bus map 23 accepts data in response to the DONE signal and completes the transfer to the bus master by transmitting an SSYN signal onto the main bus 27.

The system shown in Figure 1 also employs direct access memory facilities such as disk or drum units. Two such facilities are shown. In one, a high speed controller 34 connects to the main bus 27 and to the associative memory 24 and it also has connected to it one or more secondary storage devices 35. A similar high speed controller 36 and secondary storage device 37 are also shown in Figure 1. High speed controllers and storage devices such as described in U.S. Patent Specification No. 3,999,163 are particularly adapted for use in this system. As described in this latter U.S. Specification, the controller contains synchronous and asynchronous data paths. The asynchronous data path connects to the main bus 27 and the synchronous data path connects to the associative memory 24 in this system.

Figure 4 illustrates the signals that pass between a high speed controller such as controller 34 and the associative memory 24 over the various address (A), control (C) and data (D) conductors. When the high speed controller 34 is prepared to transfer data to or from an appropriate one of the memory modules 30 through 33 in a backup memory system 29, it transmits a CTRL REQ signal. An arbitration circuit within the associative memory 24 selects one of the high speed controllers when two or more controllers transmit the CTRL REQ signals simultaneously. The associative memory 24 then transmits back to the selected controller a corresponding SEL ADRS signal. The controller 34 then trans-

mits the various address and control signals to the associative memory 24 that are necessary to effect the transfer. These include address signals and direction signals to indicate whether the transfer is to be to or from the memory unit. An SELDATA signal is then transmitted by the associative memory 24 to the corresponding selected high speed controller, thereby to enable the controller to transmit data and parity bits to associative memory 24. If the transfer is to the controller, the controller ignores the SELDATA signal. Next, the associative memory 24 transmits a REQ ACKN signal as it begins servicing the request. The selected high speed controller removes its CTRL REQ signal from the bus in response to the REQ ACKN signal. When the REQ ACKN signal, that is received by the high speed controller, terminates, the address, control and, during a writing operation, the data signals, cease to be valid. The associative memory, 24 also transmits to the high speed controller 34 an ADRS ACKN signal in response to an ACK signal (described later) from a memory module. This indicates to the selected high speed controller that one of the memory modules 30 through 33 is responding and, during a transfer to the memory module, it indicates that the current transaction has been terminated. When data is retrieved from the selected memory module, the associative memory 24 transmits a DATA RDY signal to initiate the reading operation in the high speed controller.

Thus, the associative memory 24 is a common element in communications with the memory. Moreover, any unit connected to the main bus can communicate with the memory modules 30 through 33 through the associative memory 24. Such a transfer occurs during any transfer to a memory module (hereinafter a memory writing operation) or during a memory reading operation initiated by a bus master or high speed controller or initiated by the associative memory when it does not contain the requested data. During a writing operation, the associative memory 24 in Figure 1 transmits various address and control signals (Figure 5) onto a memory bus 40 to which all the memory modules 30 through 33 connect in parallel. Each memory module includes a memory transceiver 41, a memory control and timing circuit 42 and a plurality of memory stacks, or elements, interconnected by internal buses 43. Each memory stack, in one specific embodiment, stores in each location a number of bits which correspond to a standard computer word. However, pairs of stacks operate in parallel. One such memory stack is a "low stack" and the other memory stack in the pair is the "high stack". Corresponding stacks have the same number. Thus, reference numeral

44 designates "low stack 0" and reference numeral 45 designates "high stack 0", the pair together constituting "stack 0". Thus, a given location in a stack stores two "words" or four "bytes". As described later, the two low order address bit positions identify a specific one of these bytes and the remaining address bit positions (i.e. 20 bit positions in a 22-bit address) define the particular stack and location within that stack. When data is retrieved from the memory all the "words" in the addressed location are retrieved. During a writing operation, the associative memory 24 transmits BYTE MASK signals (Figure 5) to the memory control and timing circuit 42 thereby to select one byte or some combination of bytes in the addressed location.

Still referring to Figures 1 and 5, the associative memory 24 transmits an ADDRESS PARITY signal which is based upon the value of the address and various control signals that initiate a memory cycle and also data signals if data is being transferred from the associative memory 24. Next the associative memory 24 transmits a START signal that enables the memory control and timing circuit 42 (Figure 1) to initiate a memory cycle. The circuit 42 transmits back to the associative memory 24 an ACKNOWLEDGE signal that terminates the address and control signals and the BYTE MASK, parity, data and START signals. During a reading memory cycle, the associative memory 24 can initiate another memory cycle with another memory unit after the ACKNOWLEDGE signal is terminated. No other signals are transmitted by a memory unit back to the associative memory 24 during a writing operation. During a reading operation, on the other hand, the associative memory 24 receives a BUS OCCUPIED signal and the ACKNOWLEDGE signal. After a predetermined interval, the ACKNOWLEDGE signal automatically terminates and the data is then transmitted onto the data conductors in the bus 40 by the memory unit. The presence of the data on these conductors is indicated when the memory control and timing circuit 42 transmits a DATA READY pulse. Thereafter the BUS OCCUPIED and data signals are terminated to complete the reading memory cycle and transfer the data to the associative memory 24.

MEMORY MANAGEMENT UNIT 22

Figure 6 illustrates one embodiment of the memory management unit 22 shown in Figure 1 and described in detail in U.S. Patent 3,854,126. The memory management unit contains page address (PAR) and page descriptor (PDR) registers 51. The three most significant bits of a 16-bit "virtual" address from a BAMX multiplexer are de-

coded to select one particular corresponding pair of page address and page descriptor registers. In a multiple-mode operating system which includes KERNEL, supervisor (SUPER) and USER, or other, modes, the register set has separate page address and page descriptor registers for each mode. Each page address register contains a 16-bit page address field which is coupled to an adder 52 on conductors having positional correspondence with bit positions 6 through 21 of the final address. An address buffer 53 stores the virtual address and provides therefrom signals on conductors having positional correspondence with bits 6 through 12 as the other input to the adder 52. Bits 6 through 21 of the sum from adder 52 are concatenated with bits 0 through 5 of the output of a physical address multiplexer (MUX) 54, thereby to produce, during a 22-bit mapping mode, represented by mapping MODE signals, the 22-bit address necessary to identify a particular memory unit location. Thus, the memory management unit 22 converts a 16-bit virtual address into a 22-bit physical address.

The memory management unit 22 also has other modes of operation which are controlled by the MAPPING MODE signals to the multiplexer 54. In one mode no mapping occurs, so the "virtual" address is not changed. In a second mode the "virtual" address is converted to an 18-bit address. These two modes correspond to the modes of operation of the unit described in U.S. Patent 3,854,126.

The main bus drivers 55 couple address signals onto the main bus 27 in Figure 1 in response to a main BUS ADRS signal from a valid address check circuit 57. A register address decoder 56 also receives the address signals from the multiplexer 54 and asserts a CPU REG ADDRESSES signal that corresponds to the address of a processor register or INTERNAL REG ADDRESSES signals in response to the MAIN BUS ADRS signal from the valid address check circuit 57. The valid address checking circuit 57 receives SYSTEM SIZE signals that define the size of the available memory, the page address field signals from the selected one of the registers in the set 51 and the intermediate or block number field from the virtual address. If the address designates a location that is not present in the associative memory, the valid-address checking circuit issues a NOT CACHE ADRS signal. As the operation of this memory management unit 22 is analogous to that of the unit described in the aforementioned U.S. Patent 3,854,126, there is no further discussion of the operation of the unit shown in Figure 6 or of other diagnostic portions that are described in that patent.

MAIN BUS MAP 23

As previously indicated, the main bus map 23 in Figure 1 is an interface between the main bus 27 and the associative memory 24. In this system, the main bus map 23 acts as a "slave" device in response to signals on the main bus 27 and converts an 18-bit address found on the main bus to a 22-bit address for the bus 40. Figure 7 is a detailed block diagram of the main bus mapping circuit 23. The circuit contains a plurality of mapping registers 60 which provide an address relation function. Each register 60 stores a base address and can be accessed individually over the main bus 27 in Figure 1.

If there are 31 mapping registers, the five high order address bits which are coupled from the main bus 27 by address receivers (ADDRS) 61, are applied to a multiplexer 62 thereby to select a particular mapping register 60. The base address from the selected register 60 is then coupled to an adder 63 to be combined with the remaining low order bits from the main bus address thereby to produce the twenty two bit associative-memory address on output conductors 64.

As previously indicated, the transfer through the main bus map 23 occurs in response to signals from a bus master. Initially the address signals are received by the ADDRDS receivers 61; the C1 and C0 direction control signals by receivers 65. These signals are applied to an address decoder 66 and the high order address signals are also applied to limit comparator 67.

Figure 18A illustrates the address decoder 66 and other circuit elements shown of Figure 7. Figure 18B illustrates two typical address ranges which cause a decoder 260 in Figure 18A to transmit MAP REG OP and CACHE REG signals. If the MAP REG OP signal is transmitted, a MSYN signal from the bus master energizes an AND gate 261, enabling a delay circuit 87 thereafter to energize an OR gate 86 and a couple of SSYN signal through a driver 73, represented by an AND gate and inverter, in response to an ENBUS signal. In addition, the ENBUS signal enables a driver 83 to transmit a PB signal onto the bus if a parity error exists. This can occur when the C1 signal is not asserted, indicating a reading operation, an AND gate 262 then causing the driver 83 to transmit the PB signal if a PAR ERR signal and a PAR ADRS OK signal are asserted.

The ENBUS signal is transmitted simultaneously with a UB REQUEST signal (Figure 7). In its nonasserted state, the ENBUS signal clears flip-flops 263 and 264. When the ENBUS signal is asserted, these overriding resetting signals are removed and the flip-flop 263 can be clocked to a set

condition in response to a TIMEOUT signal. However, so long as the flip-flop 263 remains reset, it enables a NAND gate 265 to clock the flip-flop 264 to a set condition on the termination of a DONE signal. When the flip-flop 264 sets, it enables the OR gate 86 and thereby generates the SSYN signal

Figure 19 depicts the limit comparator 67. High limit jumpers 270 identify the highest address available to the mapping circuit; low limit jumpers 271, the lowest available address. Comparators 272 and 273, respectively, transmit UP LIM OK and LO LIM OK signals when a requested address is less than the upper limit of the addresses to be mapped and greater than the lower limit. If the requested address is within the limits, an AND gate 274 energizes an OR gate 275 to transmit a CACHE BUS ADDR signal and enable the AND gate 70. The AND gate 274 also enables an AND gate 277. When the MSYN signal thereafter is received, the AND gate 70 clocks the flip-flop 72 to a set condition and transmits the UB REQUEST signal (Figure 7). The flip-flop 72 is cleared by an ACKN signal from the main memory unit 29. The MSYN signal also enables the AND gate 277 to transmit PAR ADRS OK signal that is applied to the AND gate 262 in Figure 18A.

Referring again to Figure 7, C1 and C0 signals from the receiver 65 indicate to the associative memory 24 (Figure 1) the direction of an ensuing data transfer. When the transfer is to the bus 27, the data appears on CDMX conductors and is coupled through a multiplexer 75 into a buffer register 76 in response to a DONE signal. Another multiplexer 77 couples the data to drivers 74. During a transfer to the memory, the data is transmitted by the bus master before the MYSN signal is transmitted and is passed directly through receivers 80 to the associative memory 24 (Figure 1).

Accordingly, the bus map circuit 23 in Figure 7 converts the address signals from the main bus 27 (Figure 1) to an address which uniquely identifies one location in the main memory 29. In addition, the bus mapping circuit contains the necessary buffers and circuitry to couple the data and interface the control signals necessary to communicate with the main bus 27 on one hand, and the associative memory 24 on the other hand.

ASSOCIATIVE MEMORY 24

Figure 8 depicts data paths in the associative memory 24 of Figure 1. An associative memory is highly useful in this type of system because programs do not generate memory addresses randomly. Rather, they tend to access locations in the neighborhoods of locations that have been recently

accessed. This phenomenon is generally known as the "principle of program locality". It can be understood by examining the small scale behavior of a typical program. In a program, code execution itself generally proceeds in straight lines or small loops. Thus, after each access, the next few memory accesses for instructions are most likely to be within a few words ahead or behind of that access. Moreover, a data stack tends to grow and shrink from one end, with the next few accesses of the stack being near the current top. Also character strings and vectors which are often scanned sequentially, usually occupy successive memory locations.

Although the principle of locality states how most programs tend to behave, it is not a law which all programs always obey. Branching, jump and other types of instructions oftentimes cause the operation to switch to a more remote area of the memory.

If an associative memory is to have a good probability of containing the words the program needs next, it should contain words whose memory locations are close to those recently accessed. In prior systems this has been accomplished by utilizing the previously described "block fetch" operation. When it is necessary to move a word of data from the random access memory units to the associative memory, an entire block of several adjacent words are moved at once from either preceding or succeeding locations. In such systems, the size of the block that is moved is critical. If it is too small, system performance suffers unless the programs contain many small loops. If the block is too large, there may not be sufficient room to enable adequate "look ahead" or "look behind" operations.

If an associative memory contains one contiguous block of locations, it will not operate optimally, because programs do reference other code segments, subroutines, stacks, lists and buffers that are located in scattered parts of the whole memory. Rather, it is preferred that the associative memory hold those words which are estimated as the most likely to be needed no matter how scattered those words are throughout the address space in main memory. Unfortunately, for this criterion to be fully satisfied, it is necessary to have an associative memory of substantial size. The time required to search the contents of the associative memory can then become unacceptably long.

At the opposite extreme, a direct-mapping associative memory requires only one address comparison. An address is separated into fields to provide a byte designation field an index field and a byte address field. Conventionally the direct-mapping associa-

tive memory uses the byte designation field to access a byte location. The index field specifies a block of locations. If the address field is for a corresponding block in the address memory, the same as the received address field, the number in the byte field is used to identify a particular byte.

The associative memory in Figures 8 and 9 uses certain aspects of both direct-mapping and fully associative memories. In response to an incoming requested address, the associative memory uses a multibit index field (e.g., address bits 2 through 9) as a basis for selecting one of several locations within a memory module 90. The organization of the associative memory module 90 is shown in more detail in Figure 9. It includes an address storage unit 93 and a data storage unit 94. For purposes of this explanation, each of the units 93 and 94 contains two separate groups. The unit 93 is divided into a GROUP 0 TAG MEMORY 93a and a GROUP TAG 1 MEMORY 93b; and the unit 94, into a GROUP 0 DATA MEMORY 94a and a GROUP 1 DATA MEMORY 94b. Each of the memories 93a, 93b, 94a and 94b contains 256 index positions that can be identified by the index field in an incoming address. Each of the two locations at each index position in the address storage unit 93 contains a VALID BIT position, an ADDRESS TAG FIELD and a PARITY BIT FIELD that verifies that the ADDRESS TAG FIELD has been properly stored.

When an address is received, its index field is decoded to select one of the corresponding locations in each of the GROUP 0 and GROUP 1 TAG MEMORIES in the unit 93. A comparator 95 is enabled if the VALID bit in selected location in the GROUP 0 TAG MEMORY is asserted so as to enable comparator 95 to compare the value of the ADDRESS FIELD from the GROUP 0 TAG MEMORY 93a and the address field from the incoming address. If there is a match, the comparator 95 transmits a MATCH 0 signal that enables an AND gate 96 to transmit a HIT 0 signal if a parity check circuit 97 transmits a PAR OK signal based upon the signals from the VALID bit position, ADDRESS FIELD, and PARITY bit positions from the selected location.

In like fashion, a comparator circuit 100 enables an AND gate 101 to transmit a HIT 1 signal when the address field in the incoming address matches an ADDRESS FIELD from the GROUP 1 TAG MEMORY 93b and a parity check circuit 102 transmits a PAR OK signal.

Assuming that the associative memory unit 93 contains an ADDRESS FIELD which matches the address field in the incoming address, the corresponding HIT 0

or HIT 1 signal enables the multiplexer 92 in Figures 8 and 9 to pass the corresponding data from the associative memory unit 94. Specifically, the data is retrieved by addressing the unit 94 with the index field in the incoming address. Each pair of locations identified by the index field in the specific embodiment of the unit 94 shown in Figure 9 contains four words. A pair of words are shown as constituting a block with a pair of blocks being stored at each index position. In this specific embodiment, bit 1 of the incoming address selects the high or low order word of each block and therefore provides two data words at the input to the multiplexer. As the blocks are retrieved from locations having the same index positions correspond to the GROUP 0 and GROUP 1 TAG MEMORIES 93a and 94a and the HIT 0 and HIT 1 signal select the appropriate block from one of the data memories, at the multiplexer 93, to provide the 16 bits of data requested by means of the incoming address. The data is passed on to the main bus map 23 and the central processor 21 in Figure 1.

If the module 90 in Figure 9 transmits neither the HIT 0 nor the HIT 1 signal, a miss condition exists, i.e., the contents of the location identified by the incoming address are not contained in the associative memory 24. In this circumstance, control circuitry associated with the memory 24 initiates a memory cycle in the main memory 29 (Figure 1) to fetch an entire block (e.g. 2 data words) identified by certain high order bits of the incoming address. In this particular embodiment, the block is identified by address bits 21 through 10. While the block is being retrieved from the memory 29, the associative memory "decides" which of its own groups is to receive the incoming data.

When the data block arrives from the main memory 29, it is stored in the selected group of the associative memory unit 94 and, at the same time, the word in the memory location identified by the incoming address is passed along to the central processor 21 or to the main bus 27 through the main bus map 23 (Figure 1). Simultaneously, the address of the main memory location from which the data was retrieved is loaded into the corresponding location in the TAG MEMORY 93 along with a set "VALID" bit.

During a writing operation initiated either by the central processor 21 or another unit connected to a main bus 27, the initial sequence of the associative memory 24 is the same as a reading operation. The address unit 93 is accessed. If the address fields match and the corresponding VALID bit is set, i.e., a "hit" is indicated, the new data is written into the appropriate word

or byte of the associative memory unit 94, based upon the appropriate index position and byte fields of the address. The associative memory 24 also causes the new data to be written into the corresponding location in one of the main memory modules 30 through 33.

If no "hit" is indicated during such a writing operation, a write cycle is performed at the specified address in the main memory; but no changes are made in the associative memory 24.

DATA PATHS

Referring to both Figures 1 and 9, as previously indicated, it is also possible to transfer data from the main memory 29 through the associative memory 24 and one of the high speed controllers 34 or 36, to a secondary storage device. Conversely during a writing operation, data is transferred directly through the associative memory 24 to the appropriate one of the memory modules 30 through 33 identified by the address from the high speed controller. During such a writing operation, however, each address is also sent to the TAG MEMORY unit 93 in the memory 24. If a "hit" is detected during any such transfer, the corresponding VALID bit is cleared and the PARITY bit is altered. During a memory reading operation involving a secondary storage device, transfers are made directly from the designated locations in the main memory 29 to the appropriate high speed controller.

The foregoing operation will be more readily comprehended by referring again to Figure 8, which depicts the data paths through the associative memory 24. Latches 103 and 104 respond to an internal BD CLK pulse to load data from the bus 40 when data is being transferred from the main memory 29 and provide inputs to multiplexers 105 and 106 respectively. During such a reading operation, this information is coupled through the multiplexers 105 and 106 to appropriate locations in the module 90 defined by the ADDRESS FIELD bits. During a writing operation, data from the main bus 27 or the central processor 21 is coupled to a multiplexer 107. If it comes from the processor (identified when an MBC CYCLE signal is inactive) the processor data is passed by the multiplexer 107 to the main memory 29 and to the multiplexers 105 and 106, thereby to be loaded into the appropriate location. If the data appears on the main bus 27, then the MBC CYCLE is active; and the main bus data passes through the multiplexer 107 to be loaded into the module 90 along with a GENERATED PARITY signal produced by a circuit 108. During reading operations from the memory

29, main parity checking circuits 110 and 111 determine whether any parity errors exist. Likewise, a GROUP 0 parity checking circuit 112 and a GROUP 1 parity checking circuit 113 monitor data that is transmitted from the module 90 in order to detect any parity errors. These parity checking operations further insure the integrity of the data as it passes through the associative memory.

As previously stated, whenever it is necessary to transfer data between the associative memory 24 and main memory 29, a control portion of the associative memory and an addressed one of the memory modules 30 through 33 interchange signals to effect a data transfer. The circuits involved are shown in Figure 10, wherein Figure 10A illustrates the circuitry for transmitting the various control signals. More specifically, a decoder 110A has two separate sets of inputs which receive selection signals indicating which of the possible sources are providing the input information to the associative memory 24. These sources may include the central processor 21, the main bus map 23 or one of the high speed controllers 34 or 36. In response, the decoder 110A couples to a byte mask encoder 112A the selected C1 or C0 signal, the C1 signal also being active to produce a reading operation. The C1 and C0 signals from the decoder 110A also are coupled onto the bus 40.

The byte mask encoder 112A receives, in addition to the C0 and C1 signals, the least two significant bits of the address and, a CX signal that indicates that a double word writing cycle is to be performed by one of the high speed controllers. In response, the encoder 112A produces four byte mask signals which also are coupled onto the bus 40. Figure 10B tabulates the operations encoded by encoder 112A and the values of the corresponding byte mask signals for each set of input conditions. In addition, a parity generator 113A receives the address signals, the C0 and C1 signals, the byte mask signals and other signals thereby to produce a parity signal which also is coupled onto the bus 40.

The remaining signal transmitted by the associative memory 24 is the START signal. As shown in Figure 10C, decoders 115 and 116 establish the conditions under which a free-running clock 117 can clock, in succession, flip-flops 120 and 121 to their respective set states. A T150 signal controls the timing so it is only while the T150 signal is active that the flip-flop 120 can be set. The other input signals to the decoders 115 and 116 further define those conditions. For example, during a transfer to or from a secondary storage facility, the MBC CYCLE signal is active and the decoder 116 enables the flip-flop 120 to set. Another condition

is a transfer to the memory over the main bus, so long as an internal register in the associative memory is not involved. The START signal is also enabled during a transfer to the main memory from the central processor. Moreover, the START signal is generated if either the HIT0 or HIT 1 signal is inactive or if the PAR OK or PAR OK 1 signal is inactive. These signals indicate that data is invalid, as previously described, and control the START signal during a reading operation to the central processor or the main bus.

Any time the flip-flops 120 and 121 are set an OR gate 122 can energize a NAND gate 123. The NAND gate 123 generates a ground assertion signal that passes through a delay circuit 124 and energizes a NOT AND gate 125, thereby to transmit the START signal. The OR gate 122 is energized in response to any one of three conditions. First, during any reading cycle a READ signal energizes the OR gate 122. Secondly, during a writing cycle an AND gate 126 energizes the OR gate 122 so long as the BOCC signal and an MBL CYCLE signal are not asserted. The MBC CYCLE signal is asserted when the transfer involves one of the high speed controllers 34 or 36 in Figure 1. The OR gate 122 is also energized by an AND gate 127 during a writing cycle even if the internal memory bus 40 is occupied, provided the MBC CYCLE signal is asserted.

When the flip-flop 121 sets, it energizes an AND gate 130 that is enabled by an inverter 131 so long as the ACKNOWLEDGE signal from the internal bus 40 is not asserted. As a result, the AND gate 130 energizes a NOR gate 131, thereby to enable the NOT AND gate 125. The START signal is fed back to the second input to the NOR 131 to latch the START signal to an asserted level. When the ACKNOWLEDGE signal is asserted the inverter 131 disables the AND gate 130. When a transfer is completed from the associative memory 24, the DONE signal is asserted at the K input of the flip-flop 120. The next two successive clock pulses then reset the flip-flops 120 and 121 in succession. After an interval determined by the delay circuit 124, the NOT AND gate 125 is then de-energized and the START signal terminates.

Accordingly, Figures 10A and 10B show the circuits within the associative memory that generate the various control signals that the main memory unit 29 and to receive the control signals from that unit.

Figure 11 illustrates a main memory control and timing circuit 42 in one of the main memory units 30—33 of Figure 1. Memory bus receivers 130A receive signals from the internal memory bus 40 and any one of the memory modules 30 through 33 (Figure 1).

When address signals are received on the bus 40, the nine most significant bits are applied to an arithmetic unit 131A to be normalized by subtracting from the bits a value corresponding to the value of the starting address for the memory unit. This value is provided by a switch or jumper circuit identified as a starting address circuit 132. If there is a negative difference in the most significant bits or if the address is greater than any address in the memory unit, the address normalizing circuit 131A transmits an ADDRESS OUT OF RANGE signal and terminates any further operation within that particular memory unit. Otherwise the four least significant bits of the difference are coupled to a dividing circuit enabled, divides the resulting signal by two thereby to produce a 3 bit block address to specify one of eight potential blocks in the various stacks in the memory unit.

An odd parity checker 134 receives the address signals, the byte mask signals, the C0 and C1 signals and address parity signal from the memory bus receivers 130A. If the parity is incorrect, a parity error flip-flop 135 is set in response to the START signal thereby to couple a PARITY ERROR signal through memory bus drivers 136. Moreover, an ADDRESS PARITY ERROR signal from the parity checker 134 disables further operations within the memory unit.

Still referring to Figure 11, the block address signals from the divider 133 are coupled to address analysis circuitry thereby to determine whether further operations are possible based on the characteristics of the particular memory stack which has been selected. In a comparison circuit 140, the block address is compared with signals from a circuit 141 that indicate the number of potential blocks contained in the memory unit. The number is based upon the STACK SIZE ID signals that are received from the various modules. Circuit 141 also verifies that the units are in a correct orientation and transmits a CONFIG ERR signal if they are not. The circuit 141 is basically a read-only memory which transmits a digital number to the comparison circuit 140 but also transmits the CONFIG ERR signal if the STACK SIZE ID signals do not correspond to a recognized valid configuration value contained therein. If the incoming address identifies a block number which is greater than the available block numbers in the memory serviced by the circuitry in Figure 11, the comparison circuit 140 transmits an ABOVE TOP error signal. A stack pair type comparison logic circuit 142 monitors the STACK SIZE ID signals from each of the corresponding stacks in a pair to assure that stacks operating in parallel have the same characteristics. If they do not, the

circuit 142 transmits a MISMATCH ERR signal.

In this particular embodiment, the STACK SIZE ID signals can identify blocks of data in the memory unit stacks which have two different characteristics. A circuit 143 receives the STACK SIZE ID signals and calculates the highest 16K-word block contained in the system. These signals are coupled to a 16K/X decision logic circuit 144 which compares the block address in the number from the unit 143 and the incoming address (i.e. selected bits from the circuit 133) thereby to determine whether the particular address lies in a 16K-word module having one set of characteristics or a module with the other characteristics designated by the X signal. The output of this decision logic circuit 144 is coupled to a control signal generator 145. A stack selection logic and stack selection latch circuit 146 also receives the STACK SIZE ID signal and the BLOCK ADDRESS signals, thereby to identify the particular one of the stacks, such as low and high stacks 43 and 44 shown in Figure 1, that contain the addressed location.

The main START signal causes the start memory cycle logic 150 to transmit a START DELAY signal, assuming that no errors exist and that the control signal generator 145 does not assert a BUSY signal indicating that the memory unit already is executing a memory cycle. Also it is assumed that a power failure protection circuit 151 transmits neither a POWER FAILURE HOLD or RESET signal. The START DELAY signal energizes a read timing generator 152 which operates like a Johnston-type counter to provide a sequence of timing signals to the control signal generator 145.

One of the first signals thereupon transmitted by the control signal generator 145 is a LOCK MAR signal. This signal is coupled to a latching circuit 153 to latch the byte mask bits and C0 and C1 bits from the receivers 130. It also enables the stack selection logic and stack select latch 146 to latch the stack selection circuit. In addition, a memory address latch circuit 154 latches low order address bits including either the AO2 bit from a multiplexer 155, if interleaving operations are disabled, or the low order block address signal if interleaving operations are enabled.

As described in more detail later, the read timing generator 152 transmits RT1 through RTi signals in sequence. When the control signal generator 145 transmits a START WRITE signal, a write timing generator 156, which also is a Johnston-type counter, transmits WT0 through WTi signals in sequence to the control signal generator.

As a result, the control signal generator

145 transmits all the necessary signals to the individual stacks to enable a reading or writing operation. Also as described in more detail later, other signals control the transfer of data to or from the associative memory 24.

Figure 12 illustrates the start memory cycle logic 150 of Figure 11. A flip-flop 160 normally energizes an AND gate 161 to transmit the START DELAY signal which energizes the read timing generator 152 in Figure 11. If an OR gate 162, that receives all the potential error signal is energized, an inverter 163 prevents the flip-flop 160 from setting in response to the START signal and thereby inhibits the START DELAY signal. As can be seen in Figure 12, the OR gate 162 receives the error signals plus a RESET signal and a PF HOLD signal.

With further reference to Figure 12, an interleave disable switch 164 is open when interleaving operations are to occur. Thus, during interleaving operations, an inverter 165 produces a ground assertion output signal that disables an AND gate 166, so that the energization of the OR circuit 162 depends upon the other circuit elements in the system. When interleaving is enabled, a switch 170 in one memory unit will be closed, thereby to designate odd addresses and, as shown in Figure 12, the switch 170 will be open in the memory unit that contains even addresses. For the even memory which is indicated in Figure 12, the switch 170 enables an AND gate 171 and disables an AND gate 172. Whenever the ADDRESS (02) bit position in an address contains a ONE, it designates an even address and the AND gate 171 then energizes the OR gate 167. When this condition exists the inverter 168 does not provide an energizing input to the OR gate 162. Conversely, in the odd address memory unit, the AND gate 172 will energize the OR gate 167 if the ADDRESS (02) bit is a ONE.

When the START DELAY signal is transmitted, the read timing generator 152 (Figure 11) immediately transmits an RT0 signal which sets a flip-flop 173 through its preset input terminal. The flip-flop 173 thus transmits BUSY signal and disables the AND gate 161 thereby to terminate the START DELAY signal. Once this operation begins, it continues until the control signal generator 145 (Figure 11) transmits an END signal. An inverter 174 in Figure 12 receives the END signal and, when that signal is inactive, enables the AND gate 161. When, at the end of its operation, the control signal generator 145 transmits the END signal, the AND gate 161 is disabled; also the flip-flop 173 is reset, thereby terminating the BUSY signal. When the END signal terminates, the AND gate 161 is again enabled to generate a START DELAY signal. If a prior attempt

has been made to begin a memory operation with the memory unit, the flip-flop 173 will be in the set condition and thus will transmit the START DELAY signal immediately.

5 As previously indicated, the memory 29 (Figure 1) may comprise stacks having diverse characteristics. Figure 13 shows a circuit for sensing whether a particular stack being addressed contains a memory
10 with one characteristic or the other. The 16K BLK ADR (00-02) signals are applied to the A0-A2 inputs, respectively of an arithmetic and logic unit 180. The other inputs include HI STK OX through III STK 3X signals, which are provided by the individual high stack modules. These signals indicate whether the individual modules have the characteristics of 16K size modules. Each module contains a jumper or other means
15 for permanently transmitting that signal. If stack "3" has the 16K size characteristics, an inverter 181 produces a ONE at the B2 input of the logic unit 180. If stack; "3" has the alternative characteristics an AND gate 182 produces a ONE at the B1 input so long as high stack "1" has the 16K size characteristic, as indicated by a signal from an inverter 183. The B0 input to the arithmetic and logic unit 180 is provided by an OR gate 184. The B0 input signal receives B1 if stack "3" has the alternative characteristics and stack "2" has the 16K size characteristics, in which case an inverter 185 energizes an AND gate 186 and the OR gate 184. If both stack "1" and stack "0" have the 16K size characteristic, inverters 183 and 187 energize an AND gate 188 thereby to energize the OR gate 184.

Each time the memory transmits the
40 ACKNOWLEDGE signal, it clocks a flip-flop 190 to a set condition if the most significant bit of the remainder (i.e. the digital value at the B input) minus (the digital value at the A inputs) at the f3 output of the logic unit 180 contains a ONE. In this situation the flip-flop 190 transmits the 16K signal; otherwise it transmits the X signal. Thus the sense decision logic 144 provides either a 16K or an X output signal.

50 Figure 12 also shows the circuitry for generating the PF HOLD and reset signals which are applied to the OR gate 162. Whenever the DC supply voltage drops below a safe level, appropriate circuitry transmits a MAIN DC LOW signal which sets a latch 191. This transmits the PF HOLD signal. In addition, a network comprising transistors 192 and 193 and a diode 194 holds the input to an inverter 195 at a ground level generates the RESET signal when the MAIN DC LOW signal terminates.

Each of the stacks shown in Figure 1 comprises XY coordinate drivers, sense-inhibit circuits and the stack modules which
65 contain the cores and is a conventional

coincident-current magnetic core memory. The XY driver circuits decode the address signals to select a particular string of cores. Each sense-inhibit circuit connects to a winding which is threaded through cores in parallel with the Y current winding. As known, the X and Y currents are passed through the cores in the direction that will switch the cores from a ONE magnetic state to a ZERO magnetic state and the resulting change in magnetic field induces a voltage in the sense-inhibit winding. This voltage is detected and amplified by a sense amplifier to transmit a ONE if the core state changes. Otherwise, no voltage is induced in the sense-inhibit winding and the corresponding register bit is not set. Such a reading operation is known as a destructive reading operation because the state in the cores which have been sensed are in the ZERO state at the completion of the operation.

In order to write data in a core, the core is initially switched to a zero state by performing a destructive reading operation. The X and Y currents are then switched through the core in the direction that will switch the core from the ZERO magnetic state to a ONE magnetic state. If a ZERO is to be written into the core, an inhibit current is driven through the sense-inhibit winding and this induces a magnetic field having the opposite polarity to the field induced by the X and Y currents, so the core remains the ZERO state. The circuits which transmit these signals and the necessary timing signals are basically analog.

Referring again to Figure 11, a margin bit decode circuit 200 receives MARGIN bits from the bus 40 and produces various signals which can alter the operation of these analog circuits. For example, if the margin bits all have a zero value, the system operates at normal timing and current levels. If the MAIN MARGIN (1) bit is asserted, the value of a STROBE MARGIN signal is shifted to delay strobe pulses. More specifically, when the MARGIN 1 one bit is asserted, the STROBE MARGIN signal goes from a normal value of +2.5 volts to 0 volts and causes each memory strobe pulse to be generated earlier than normal.

In Figure 14, the margin bit decoder 200 is shown in detail. The margin bit decoder 200 is essentially a signal conversion means which in response to binary margin control signals generates an output signal having a plurality of signal levels in accordance with the margin control signals received. An array of inverters 201 and another array of inverters 202 provide positive and ground assertion levels respectively, for the margin signals. When the margin signals are all at a nonasserted level, the STROBE MARGIN, DRIVE HIGH MARGIN and DRIVE LOW MARGIN signals are at normal

levels. In addition, an AND gate 203 is de-energized. If only the MARGIN 0 bit is asserted, a special code exists and the AND gate 203 then produces the ADDRESS PARITY ERROR signal, which is coupled back to the parity checker 104 in Figure 11. If the strobe pulses that determine the time that data latches monitor the sense windings are late, then the MARGIN signals are sent in which the MARGIN 1 bit only is asserted. When this occurs, NAND gate 204, which was energized, is de-energized and an inverter 205 energizes a NAND gate 206, thereby shifting the STROBE MARGIN signal to ground potential and causing strobe pulses to occur earlier than normal by a fixed amount. If both the MARGIN 1 and MARGIN 2 bits are asserted, the outputs of the NAND gate 206 and an inverter 207 both shift to a positive potential, thereby to shift the STROBE MARGIN signal to a voltage of approximately +5 volts. This delays the generation of the sense amplifier strobe pulses until a fixed interval after the normal time.

The X and Y currents can be increased by energizing a NAND gate 211. This occurs when the MARGIN 0 and 2 bits are asserted and the MARGIN 1 bit is not asserted. With this condition an AND gate 212 enables the MARGIN 0 bit energize an AND gate 211 and thereby reduce the voltage level of the DRIVE HIGH MARGIN signal. Likewise, if only the MARGIN 2 bit is asserted, the AND gate 212 enables a NAND gate 213 to be energized by the MARGIN 0 bit, thereby to shift the DRIVE LOW MARGIN signal to a low value and reduce the X and Y current.

With the thus-described structure of a memory module in mind, we shall now discuss in detail the three typical transfer operations which occur between the associative memory 24 and a location in the memory module.

Now referring to Figures 11, 15 and 20, the reading operation begins when the address and control signals appear on the bus 40. During a reading operation, the byte masking bits are not used. The other control signals which are used are the C1 and C0 signals, which indicate the reading operation, and the parity signals. Assuming the requisite information is received at the controller in Figure 11 with no errors, the appearance of the START signal on a corresponding control conductor initiates a reading operation. When the read timing generator 152 transmits the RT0 signal to the start memory cycle logic shown in Figure 12 it sets the flip-flop 173 therein. Simultaneously, the control signal generator 145 transmits LDMBR, LOCK MAR, IACK, BOCC and READ EARLY signals.

The LDMBR signal corresponds to the

START DELAY signal shown in Figure 12. This signal clocks data at input data latches in the memory transceiver 41 in Figure 1; it corresponds to the DATA GATING control signal shown in Figure 1. However, that data is not thereafter used during a reading operation.

The LOCK MAR signal clocks the address signals into the memory address latch 154, the C0 and C1 control bits and byte mask byte into the latches 153 and the block address into the stack selection logic and latch 146. The IACK signal corresponds to the ACKNOWLEDGE signal shown in Figure 11; it is transmitted through bus drivers 136 in Figure 11 onto the bus 40.

Referring to Figure 13, the RT0 signal energizes an OR gate 220, thereby to transmit a BOCC ENABLE signal to enable an AND gate 221. During a reading operation, this signal remains asserted until the memory controller removes the signal from the bus 40. As also shown in Figure 13, during a reading cycle, the C0 and C1 bits energize an AND gate 222, thereby to transmit an ENABLE signal which energizes the AND gate 221, so the gate 221 transmits a (BOCC) signal back to the drivers 136 in Figure 11 and onto the bus 40.

The READ EARLY signal turns on the selected X and Y switches, the selected Y drivers and the X and Y current generators in the stack module and effectively turns on the current in the stacks.

As shown in Figure 15, the next transitions which occur are the assertion of the CLR MDR signal and the termination of the LDMBR signal. The CLR MDR signal shown in Figure 15 represents two signals. One signal is routed to a low stack module and the other to the high stack module (Figure 1). These signals clear memory data registers in the memory stacks, i.e., a register comprising an array of flip-flops 223 shown in Figure 20. This clearing operation is necessary to initialize the registers so that stages therein can be subsequently directly set during the destructive reading operation of the cores. Also flip-flops, corresponding to the flip-flop 223, in unselected stack modules must be cleared so that when read data is gated to the memory controller the internal bus lines are asserted only by data from the selected stack module.

During the CLR MDR pulse, the control signal generator 145 in Figure 11 asserts the READ LATE signal. This signal turns on the X drivers in the selected stack and forces current through the X wires. The cores being addressed are now destructively read. The selected Y drivers are not yet turned on, to allow transients to settle as long as possible before the cores being addressed are switched. When the latter occurs, a sense

amplifier, such as the sense amplifier 224 in Figure 20, connected to the appropriate sense-inhibit lines, produces an output voltage. A sense strobe generator 225 enables an AND gate 226 to directly set the flip-flop 223 if the selected core contains a ONE. Otherwise the flip-flop 223 remains reset. As shown in Figure 20, and previously discussed, the timing of the sense strobe generator can be controlled by the strobe margin signal, generated as shown in Figure 14.

Next the timing and control signal generator 145 (Figure 11) transmits the SAS EN signal. This signal, shown in Figure 15, represents one of four signals which are controlled by the byte mask signals from the latches 153. During a reading operation all four SAS EN signals are asserted so that the sense strobe generator 225 strobes all bytes into the internal data register comprising flip-flops 223.

Before the control signal generator 145 terminates the SAS EN signal, it also produces a ST WRITE pulse that initializes the write timing generator 156 and begins a timing sequence with a WTO signal. When the ST WRITE pulse terminates, the control signal generator 145 also terminates the SAS EN signal and asserts an MEM-OUT EN signal. As shown in Figure 20, this signal enables NAND gates, such as NAND gate 227 to couple, onto the data lines which connect to the transceiver 41 in Figure 1, a ground assertion signal representing the state of the flip-flops 223, i.e., the data which is to be retrieved from the memory 29. Thereafter, the control signal generator 145 transmits the TXR OUT EN signal which, conditioned by inverters 230 and 231, enables NAND gates 232 to couple the bus data from internal bus data lines through inverters 233 and onto the bus 40. The TXR OUT EN signal terminates with the MEM OUT EN signal.

When the MEM OUT EN signal terminates, the memory transceiver 41 is conditioned to couple data from the bus 40 to the internal bus. At the same time, the BOCC TIME signal terminates, so the AND gate 221 (Figure 13) terminates the BUS OCCUPIED signal on the bus 40, thereby to indicate to the associative memory that the bus is free for a subsequent operation.

While the TXR OUT EN signal is asserted, the control signal generator 145 also transmits the DATA READY signal, which passes through the memory transceiver 41 onto the bus 40 as a control signal. This informs the bus master that the data that has been requested is on the bus 40 and enables latches within the bus master to read the data.

As previously indicated, the ST WRITE signal shown in Figure 15 initiates a timing sequence by the write timing generator 156

in Figure 11. Simultaneously with the transmission of the TXR OUT EN signal by the control signal generator 145, signals from the write timing generator 156 cause the generator 145 to transmit ST CHG and WRITE EARLY signals. The ST CHG signal is routed to the selected stacks in response to the STACK SEL signals from the stack selection logic and switch 146. It activates a stack charge circuit used in subsequent writing operation. The WRITE EARLY signal turns on selected X and Y switches and the X and Y current generators in the stack module sets enabled by the STACK SEL signals. Next, the control signal generator 145 transmits the WRITE LATE and INHIBIT TIME signals. The inhibiting signal is routed to a NAND gate 240 in Figure 20. It provides the timing for turning on the inhibit driver 241, depending upon the condition of the flip-flop 223. If the flip-flop 223 is cleared, representing a ZERO value that was read, the corresponding inhibit driver 241 is not energized. If the flip-flop contains a ONE, however, the inhibit driver 241 is enabled to drive the inhibit winding through a sense terminator network 242. This operation thus rewrites into the stack the data that was destructively retrieved therefrom during the reading operation. When these four signals from the control signal generator 145 terminate, there is a delay and then the control signal generator transmits an END pulse which clears the timing generators and terminates the BUSY, LOCK and LOCK MAR signals, thereby enabling the stack to initiate a subsequent operation.

The second type of operation that can occur is a writing operation. The sequence of timing signals in the writing operation is shown in Figure 16. During a writing operation, the bus master produces on the bus 40 the ADDRESS, BYTE MASK signals, C0 and C1 signals indicating a write operation and MAIN ADDRESS PARITY signals. Shortly thereafter the bus master transmits onto the data lines the data signals and then transmits the START signal. As previously indicated, the START signal causes the start memory cycle logic 145 to transmit the START DELAY signal and initiate the timing sequence and, as also previously indicated, it causes the control signal generator 145 to transmit the BUSY, LD MBR, LOCK MAR, IACK and READ EARLY signals.

Next the control signal generator 145 terminates the LD MBR signal and loads the data into the latch 250 in the memory transceiver 41 (Figure 20). Simultaneously it transmits the CLR MDR signal, thereby to clear the flip-flops 223. Next the control signal generator 145 transmits the READ LATE signal, thereby to clear the cores in

the designated address, which is necessary before a writing operation can occur. The CL MDR signal then terminates and the LD MDR signal is asserted. The TXR OUT EN signal is active during this interval, so the inverter 230 enables a NAND gate 251 to couple a signal onto the internal bus which corresponds to the data to be transferred. This signal is coupled through a receiver 252, so the LD MDR signal loads the data into the latches corresponding to the flip-flop 223. The control signal generator 145 also transmits the READ LATE signal, thereby destroying the data in the cores.

The SAS EN signals complement the byte mask signals. Thus, if byte 0 is to be written, EN SAS BYTE signals 1—3 are asserted, so that the data in the cores is stored in the corresponding ones of the flip-flops 223. If the byte is to be written, then the sense strobe generator associated with that particular byte and the AND gates 226 are not enabled to couple the data to the flip-flop 223. Thus, at the termination of the READ LATE signal, the latches corresponding to the flip-flop 223 store data corresponding to the data to be written into the new bytes and, with respect to any other bytes, the data originally stored in those bytes.

As in a prior reading operation, the control signal generator thereafter transmits the ST CHG, WRITE EARLY, WRITE LATE and INHIBIT TIME signals to restore the data to the cores and then transmits the END signal to terminate the operation.

The third type of an operation which can be performed in this system is an exchange operation during which data in latches in the bus master and data in a designated memory location are exchanged. During an exchange operation, the bus master effectively performs a writing operation, thereby placing the address of the memory location, the control signals indicating an exchange operation, and the data, onto the bus 40. When the memory controller shown in Figure 11 initiates the operation, it transmits the BUSY, LD MBR signals, LOCKMAR and IACK signals. It also transmits the BOCC TIME and READ EARLY signals. These all perform the same functions as previously indicated and, when the LD-MBR signal terminates, the data to be loaded into the memory location is stored in the flip-flops 250. Next the system utilizes the CLR MDR signals to clear the flip-flops 223 and initiates the READ LATE signals. The SAS EN signals control which bytes will be thereafter read into the flip-flops 223 in order to provide an exchange operation on a byte-by-byte basis. When the SAS EN signals terminate, the control signal generator 145 transmits an MEM OUT EN

signal, thereby to enable the NAND gates 227 to couple data onto the internal bus data lines. Then the control signal generator 145 transmits the TXR OUT EN signal, to enable the selected bytes to pass onto the bus 40, and produces the DATA READY signal. In addition, the control signal generator 145 transmits over this entire interval the HOLD signal, which prevents the write timing generator from transmitting any signals.

Once the reading operation is terminated, however, the write timing generator 156 is enabled to initiate its timing sequence. The first signal which is then transmitted from the generator 145 is the LD MDR signal which loads into the transceiver flip-flops 250 (Figure 20) the data to be stored in memory in the exchange. Writing is done on a byte-by-byte basis and the LD MDR signal for a particular byte is asserted only if the corresponding byte mask bit is asserted. Thus, only the bytes to be written are loaded into the flip-flops 223 (Figure 20). This loading operation occurs because the TXR OUT EN signal terminates at the end of the reading operation and thereby enables the NAND gate 251 to couple the data in the flip-flops 251 onto the internal data bus. The remainder of the writing operation then occurs as previously discussed and the memory cycle then terminates.

In summary, there has been described in this application a highly efficient associative memory arrangement for a data processing system. In accordance with the principle of locality, the instances of "hits" in the associative memory material outnumber the number of "misses" so that the effective speed of data transfer from memory to processor is greatly increased. Moreover as previously indicated, this arrangement is capable of providing a large memory (e.g., 4 million bytes of random access memory) in a cost-effective manner from which data can be retrieved at an extremely rapid rate. Also, as shown especially in Figure 11, each memory controller contains circuitry that enables diverse types of memory stacks to be intermixed within a given memory unit. Interleaving operations also are provided. In combination, all these elements provide a memory which can furnish data words to the processor at a rate which greatly exceeds the rate normally imposed by the characteristic retrieval intervals of the memory units.

The foregoing detailed description is of a particular embodiment of a data processing system and memory circuits for implementing this intervention. It will be apparent, however, that many modifications and variations exist and might be used to adapt this invention for other types of data processing

systems. Therefore, it is the object of the appended claims to cover all such variations and modifications as come within the scope of this invention as defined by the following claims.

WHAT WE CLAIM IS:—

1. The combination of a digital data processing system and a random access memory module connected via a bus to the digital data processing system that, for each of successive memory cycles including reading and writing memory cycles, initiates a data transfer by transmitting binary address signals and asynchronous control signals that control the transfer of data between said random access memory module and the digital data processing system, said random access memory module comprising:

A. addressable storage means that store digital data in addressable storage locations therein.

B. address decoding means that produce an enabling signal when said address decoding means receives, from the digital data processing system, binary address signals that identify a said storage location in said addressable storage means, and

C. control means connected to said addressable storage means and said address decoding means, said control means including:

i. means that respond to the enabling signal and respective asynchronous control signals to initiate a reading or writing memory cycle, each such reading memory cycle including a reading operation during which digital data is destructively read from an identified one of said locations in said addressable storage means and a subsequent writing operation during which data is restored to said identified location in said addressable storage means, and

ii. means that transmit to the digital data processing system a BUS OCCUPIED signal during the reading operation of each reading memory cycle, the BUS OCCUPIED signal temporarily inhibiting the digital data processing system from transmitting binary address signals and asynchronous control signals to initiate successive memory cycles.

2. A combination as claimed in claim 1, wherein said control means additionally includes:

iii. means operative to generate a START DELAY signal that initiates the memory cycle in said random access memory module, said START DELAY signal generating means being responsive to the asynchronous control signals from the digital data processing system, and

iv. means connected to said START DELAY signal generating means and said

BUS OCCUPIED signal transmitting means that transmit an ACKNOWLEDGEMENT signal to said digital data processing system in response to the START DELAY signal.

3. A combination as claimed in either claim 1 or 2 wherein:

A. said addressable storage means comprises a plurality of random access memory stacks, each said stack containing a number of locations that are identified by consecutive memory addresses, certain of said stacks having first operating characteristics and the others of said stacks having second operating characteristics, each said memory stack signal,

B. said control means additionally comprises:

v means that receive the stack signals from said memory stacks, and

vi means that generate a stack characteristic signal in response to the stack signals and the ACKNOWLEDGEMENT signal, the stack characteristic signal corresponding to the stack signal of the stack identified by the binary address signals, the timing of said BUS OCCUPIED signal being selectively altered in response to the stack characteristic signal.

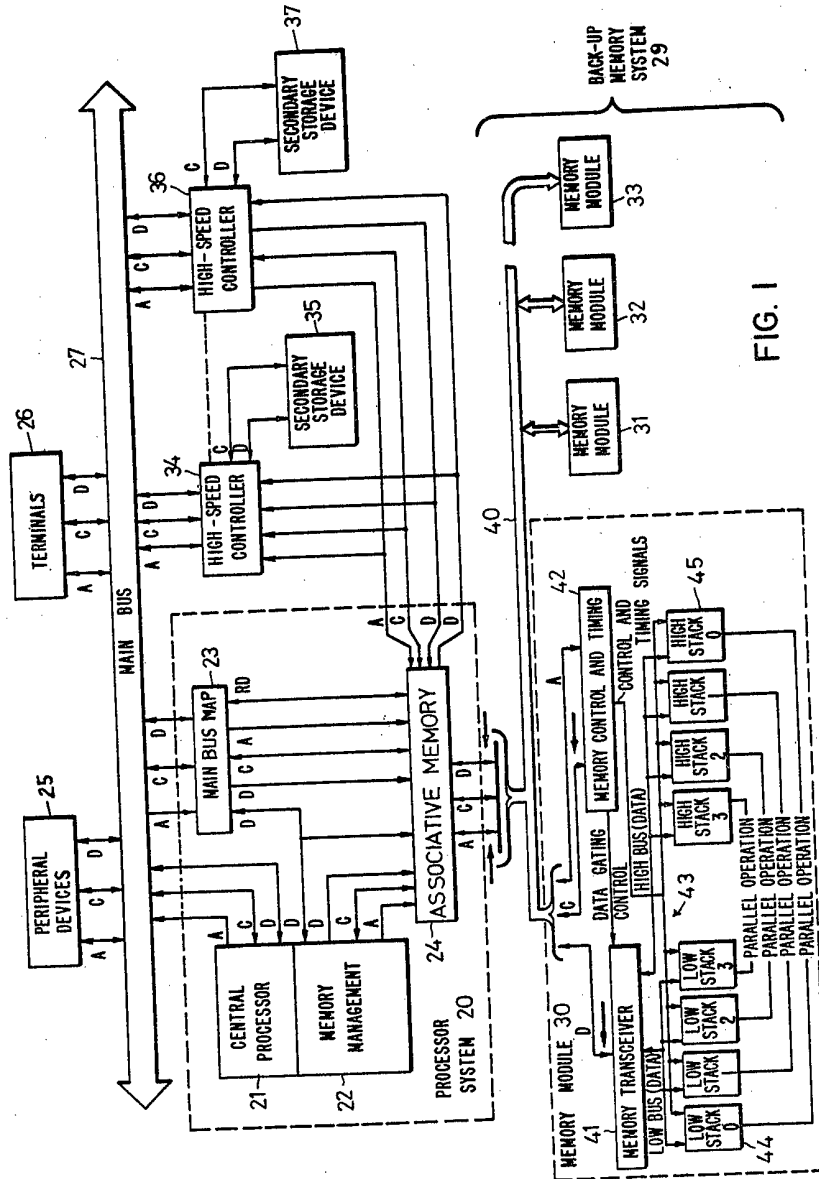
4. A combination as claimed in claim 3 as appended to claim 2, wherein said memory stacks are grouped in pairs of stacks, said control means additionally including:

vii means that generate an error signal in response to the stack signals from said memory stacks, the error signal disabling said START DELAY signal generating means when the stack signals indicate that the stacks in a said pair have different operating characteristics.

5. A combination as claimed in any of claims 1 to 4 wherein said control means additionally includes control circuits that control the operation of said addressable storage means in response to control signals and the digital data processing system transmits binary margin signals that establish the values of the control signals, said random access memory module additionally comprising means that convert the binary margin signals received from the digital data processing system to control signals having a plurality of levels thereby to control the operation of said control circuits.

6. The combination of a digital data processing system and a random access memory module as claimed in Claim 1 and substantially as hereinbefore described with reference to the accompanying drawings.

BARON & WARREN,
16, Kensington Square,
London, W8 5HL.
Chartered Patent Agents.



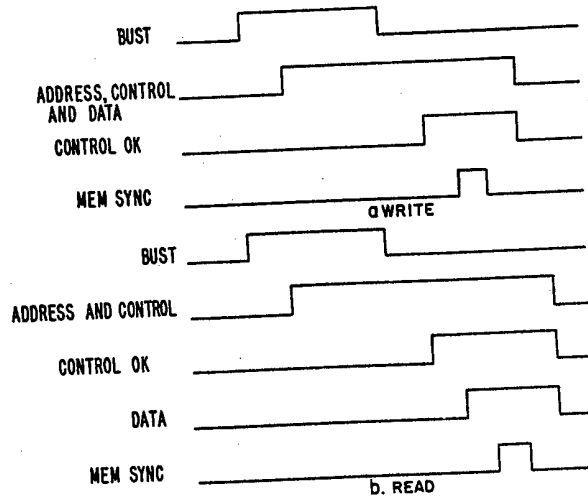


FIG. 2

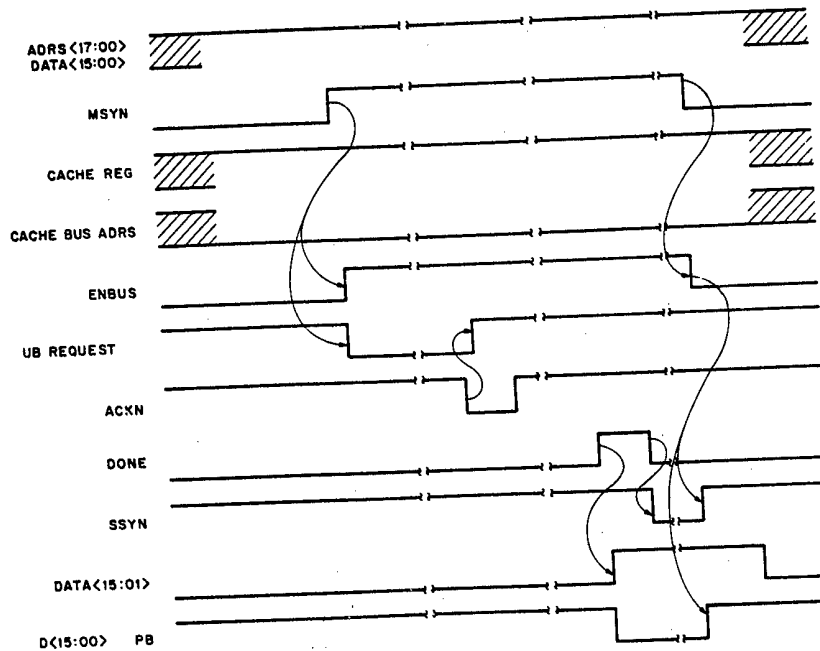


FIG. 3

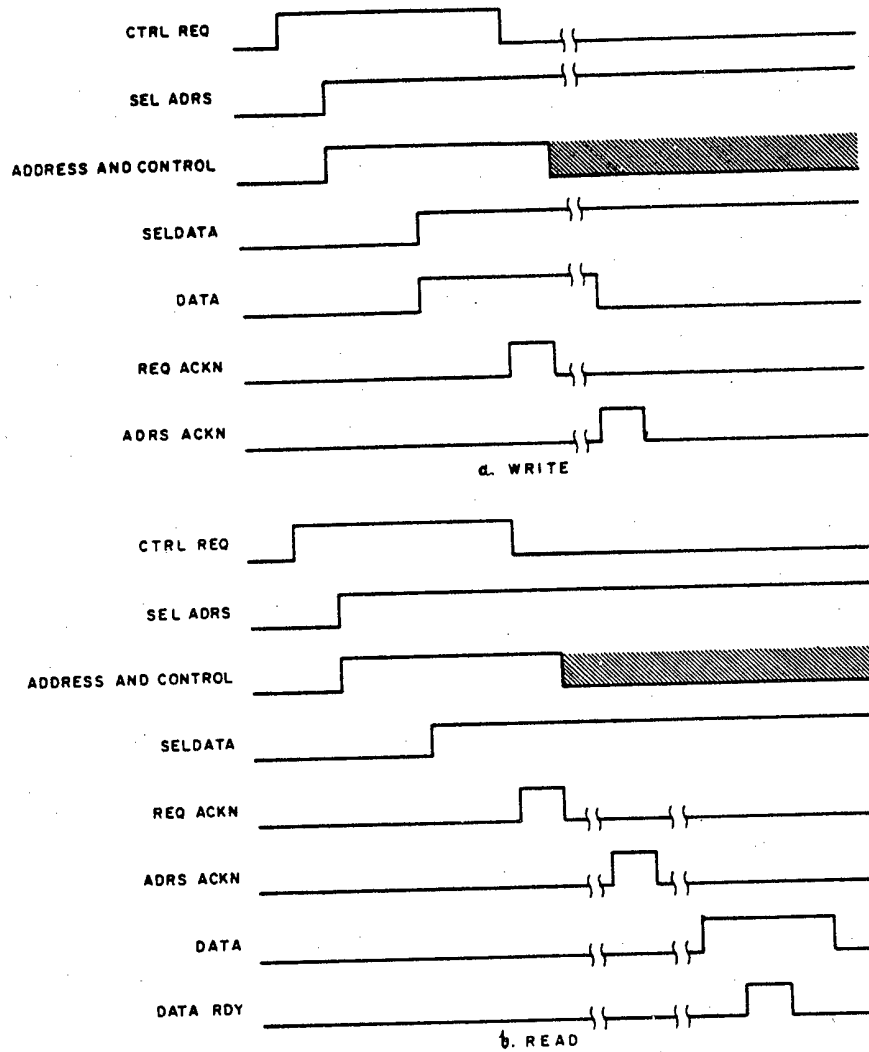


FIG. 4

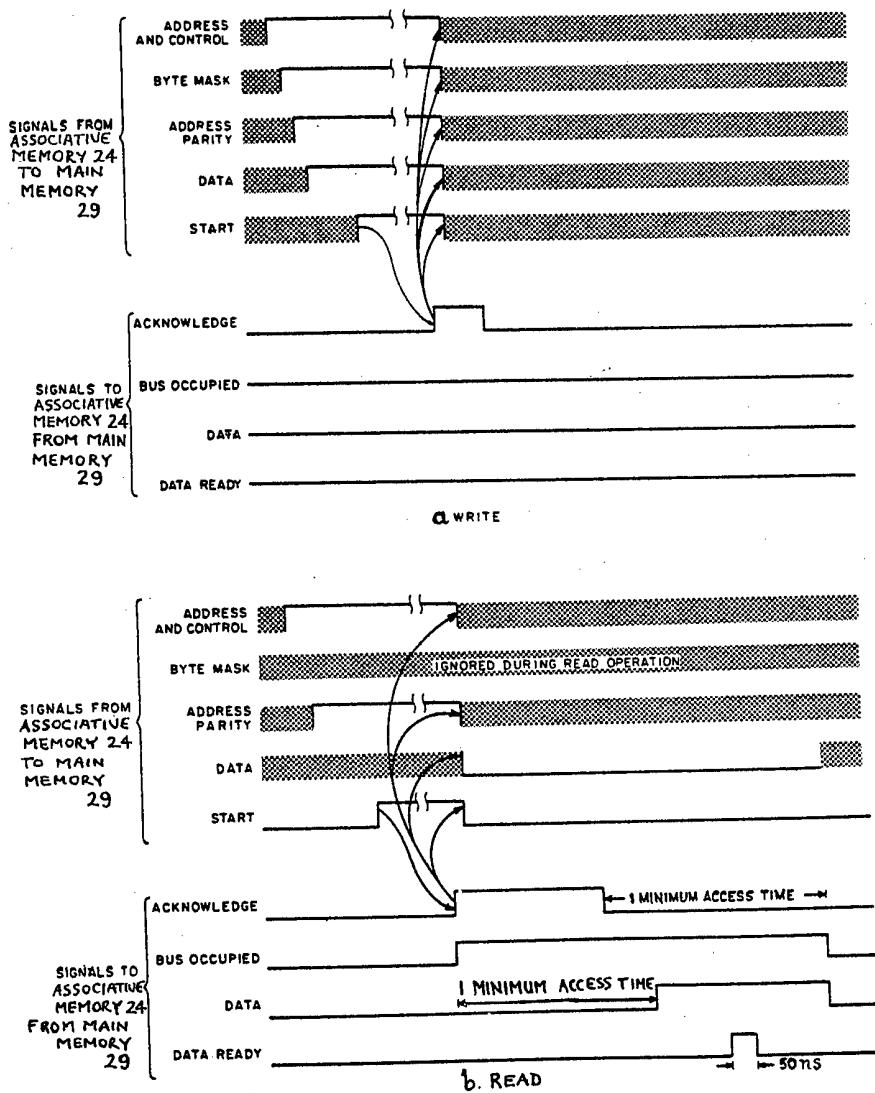


FIG. 5

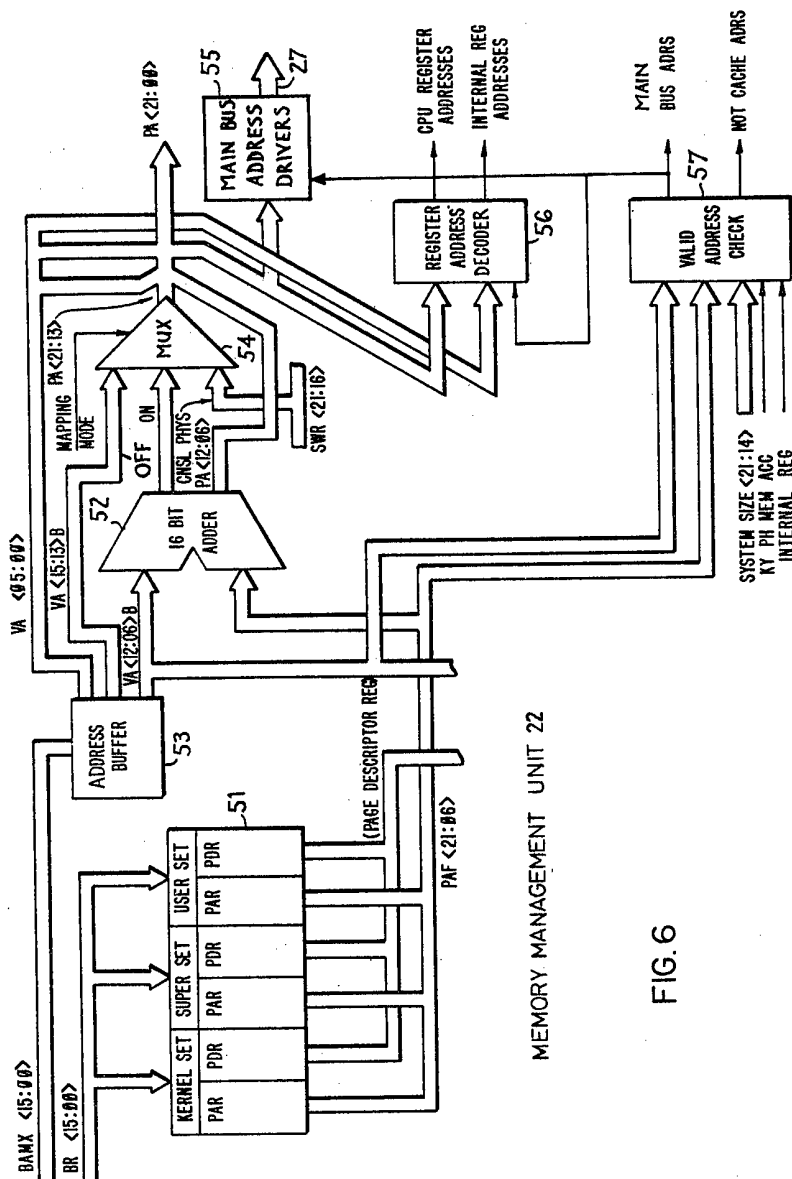
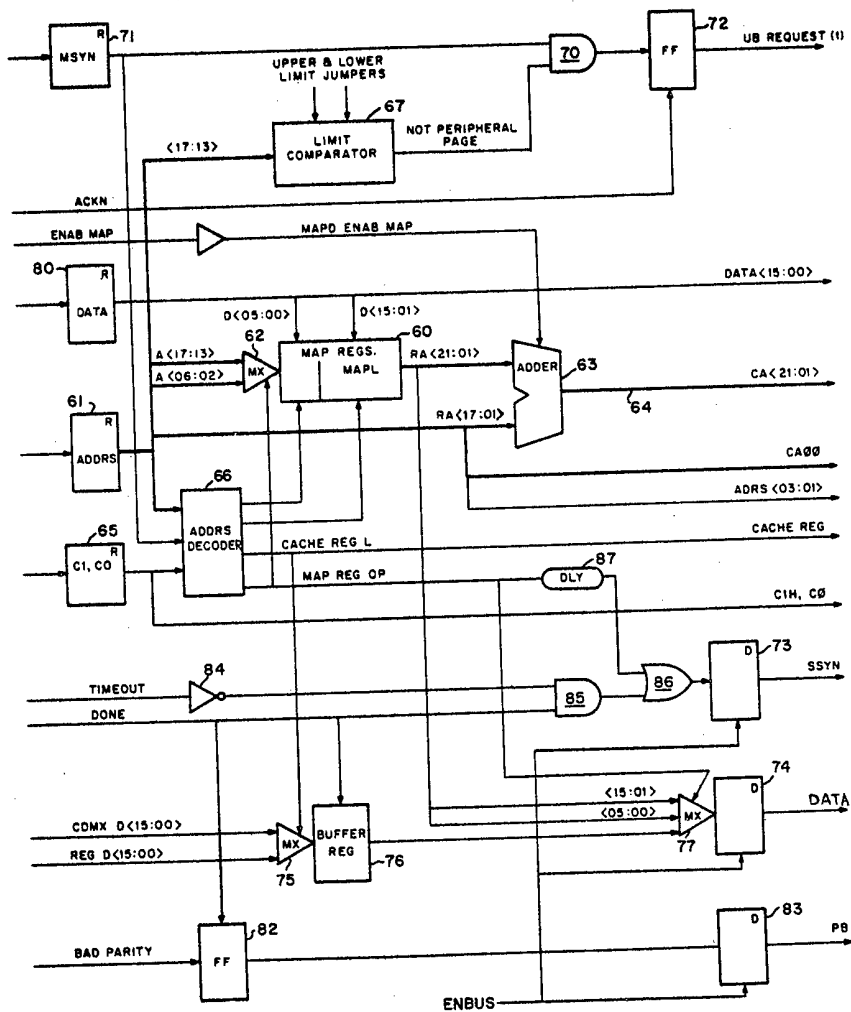


FIG. 6



MAIN BUS MAP 23

FIG. 7

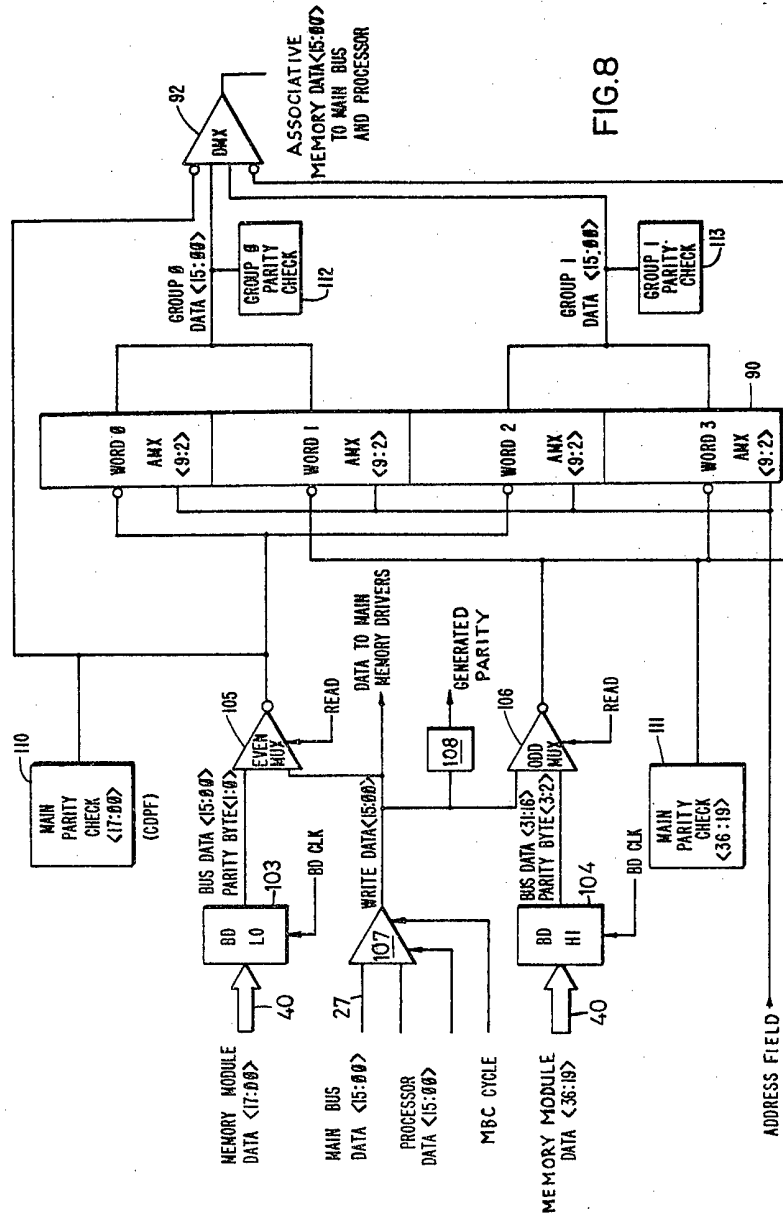


FIG. 8

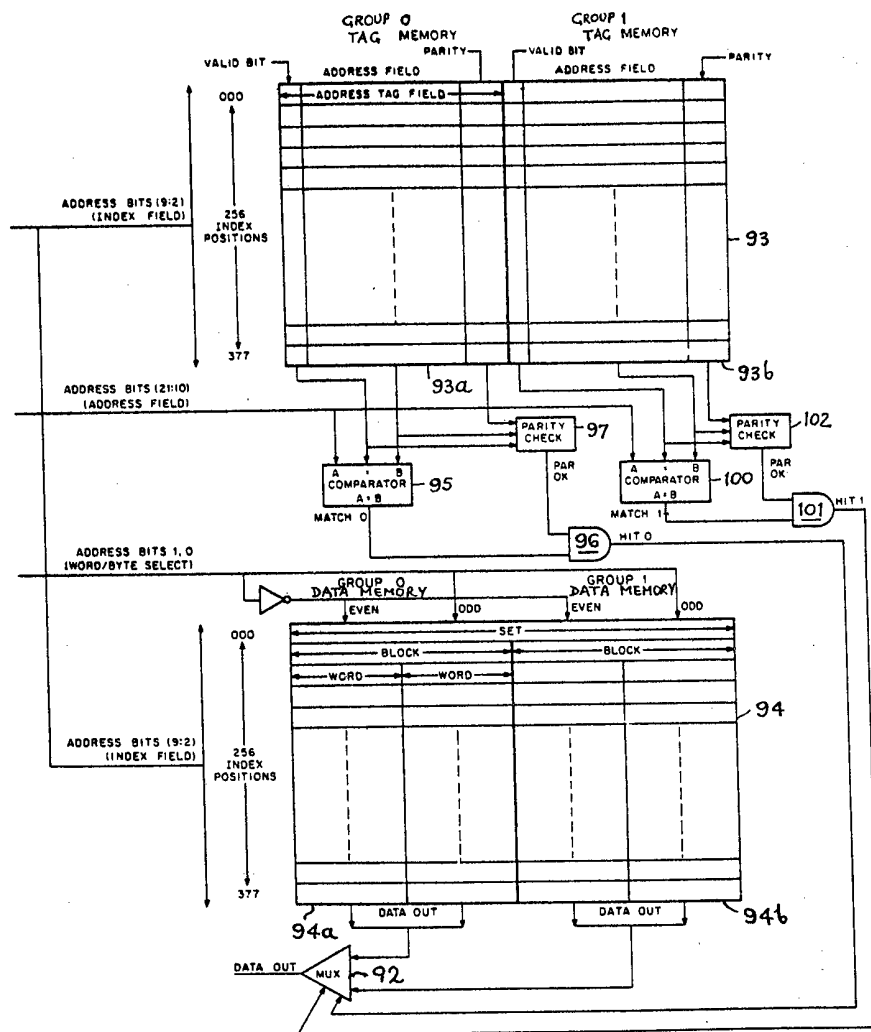


FIG. 9

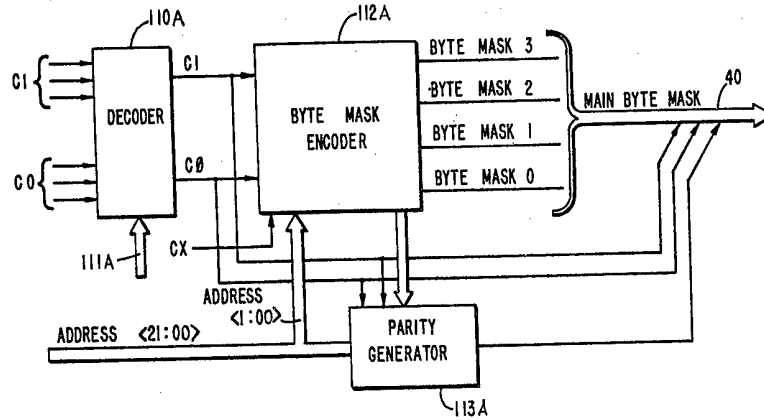


FIG. 10A

C1	C0	CX	A1	A0	OPERATION	BYTE MASK			
						3	2	1	0
0	0	0	X	X	READ DOUBLE	X	X	X	X
0	1	0	X	X	READ DOUBLE	X	X	X	X
1	0	0	0	X	WRITE WORD	0	0	1	1
1	0	0	1	X	WRITE WORD	1	1	0	0
1	1	0	0	0	WRITE BYTE	0	0	0	1
1	1	0	0	1	WRITE BYTE	0	0	1	0
1	1	0	1	0	WRITE BYTE	0	1	0	0
1	1	0	1	1	WRITE BYTE	1	0	0	0
1	0	1	X	X	WRITE DOUBLE	1	1	1	1

FIG. 10B

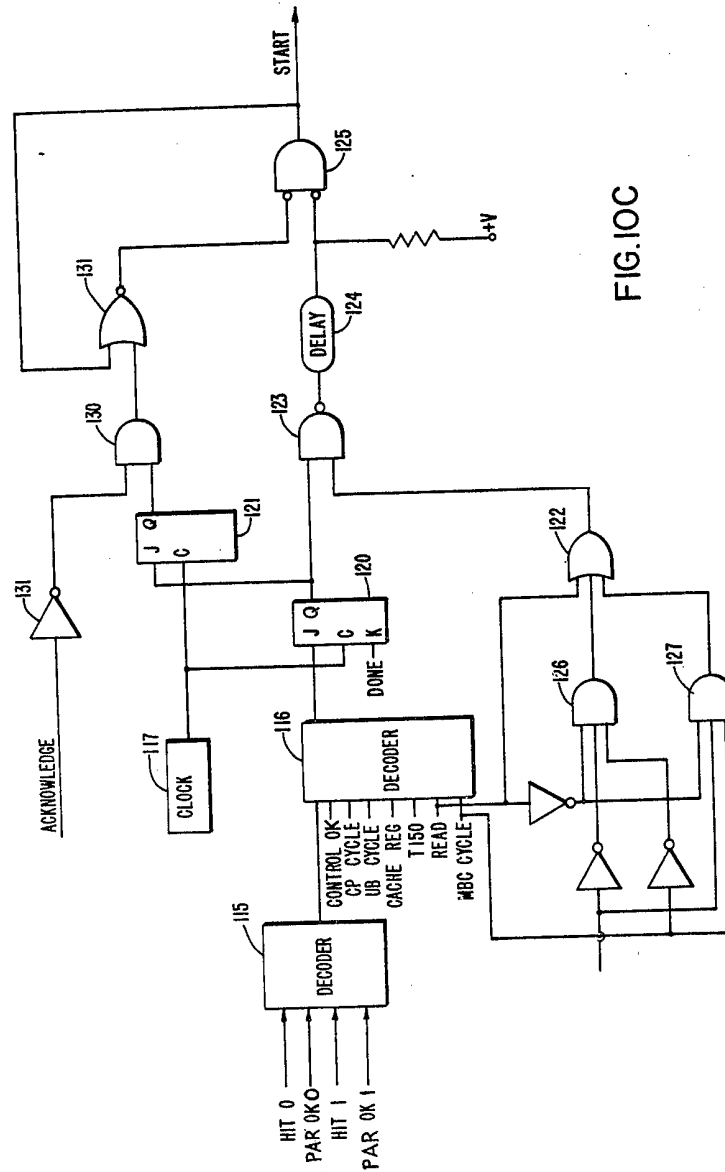
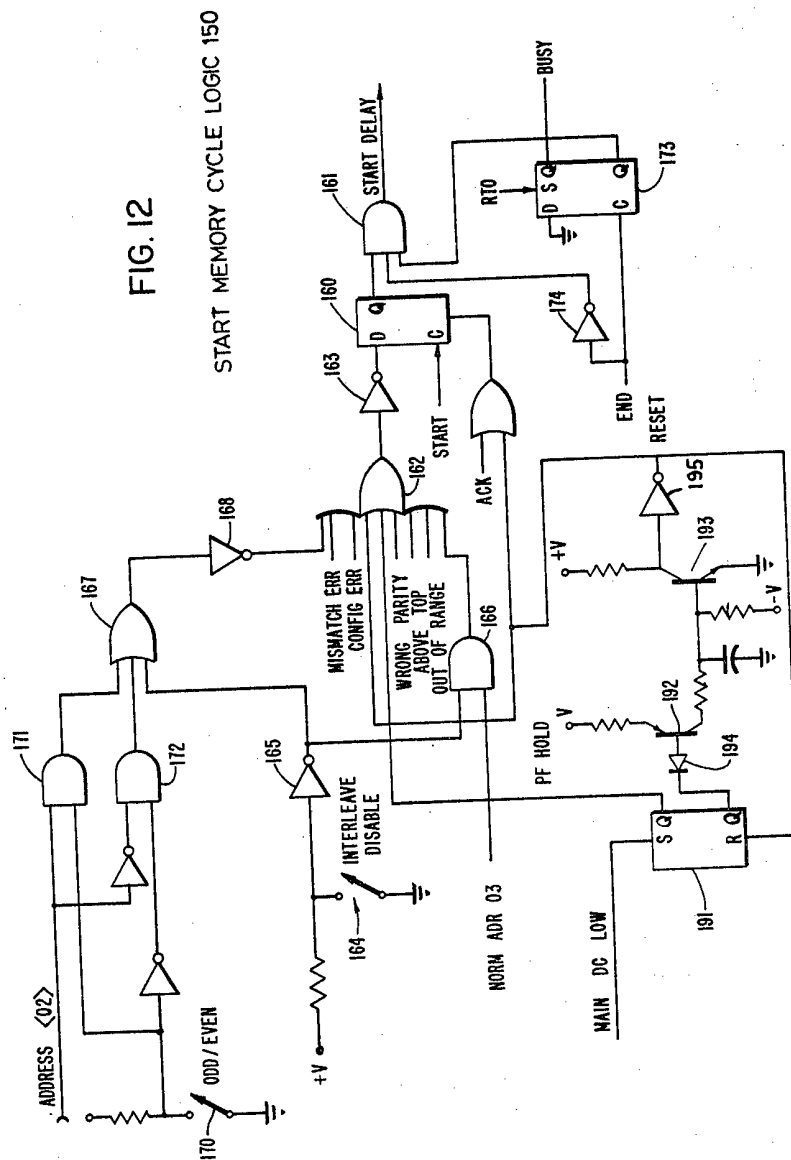
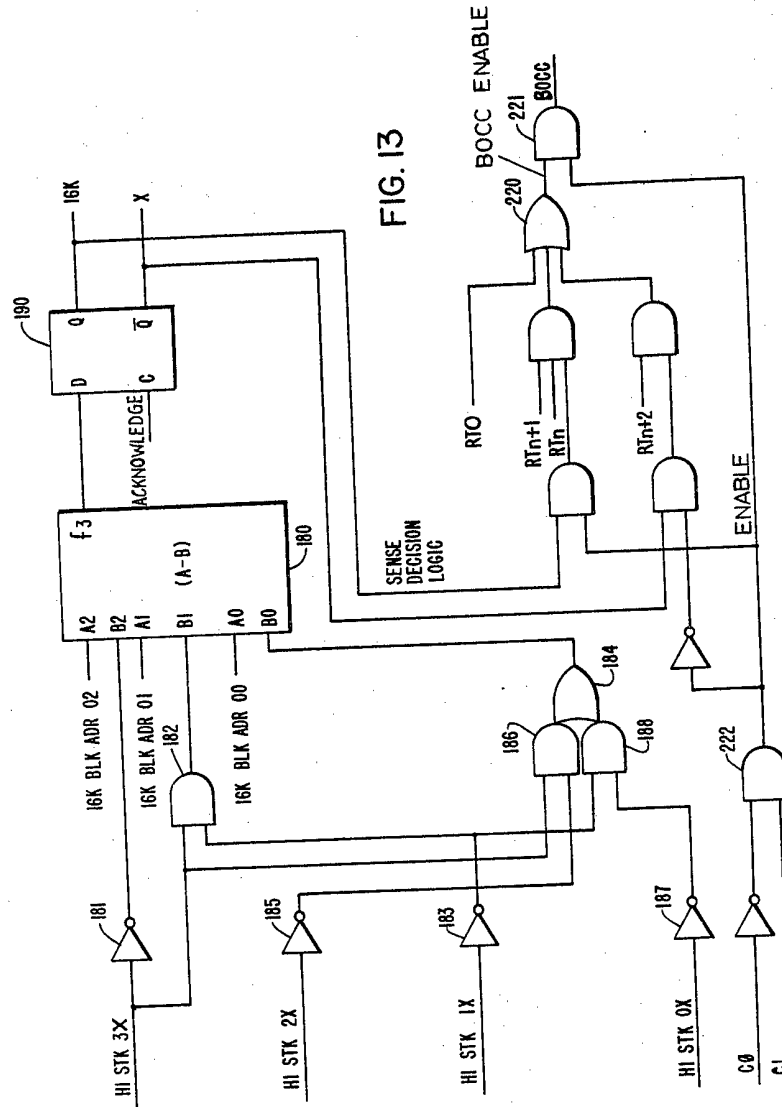


FIG. 10C







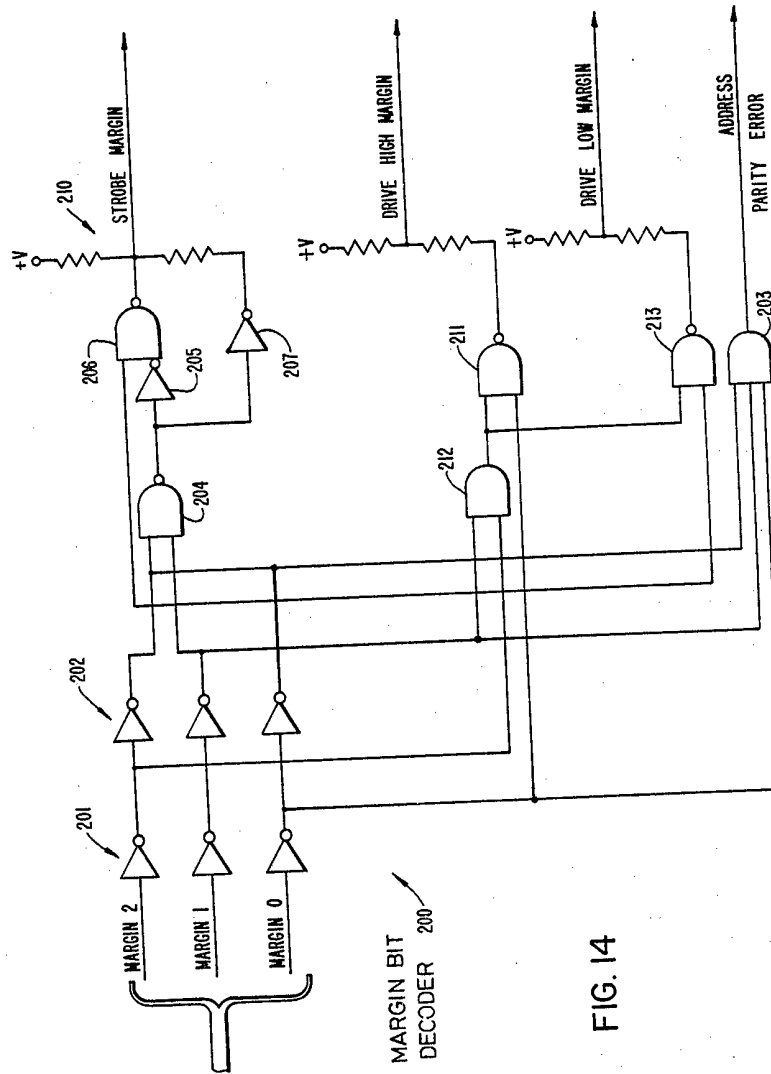


FIG. 14

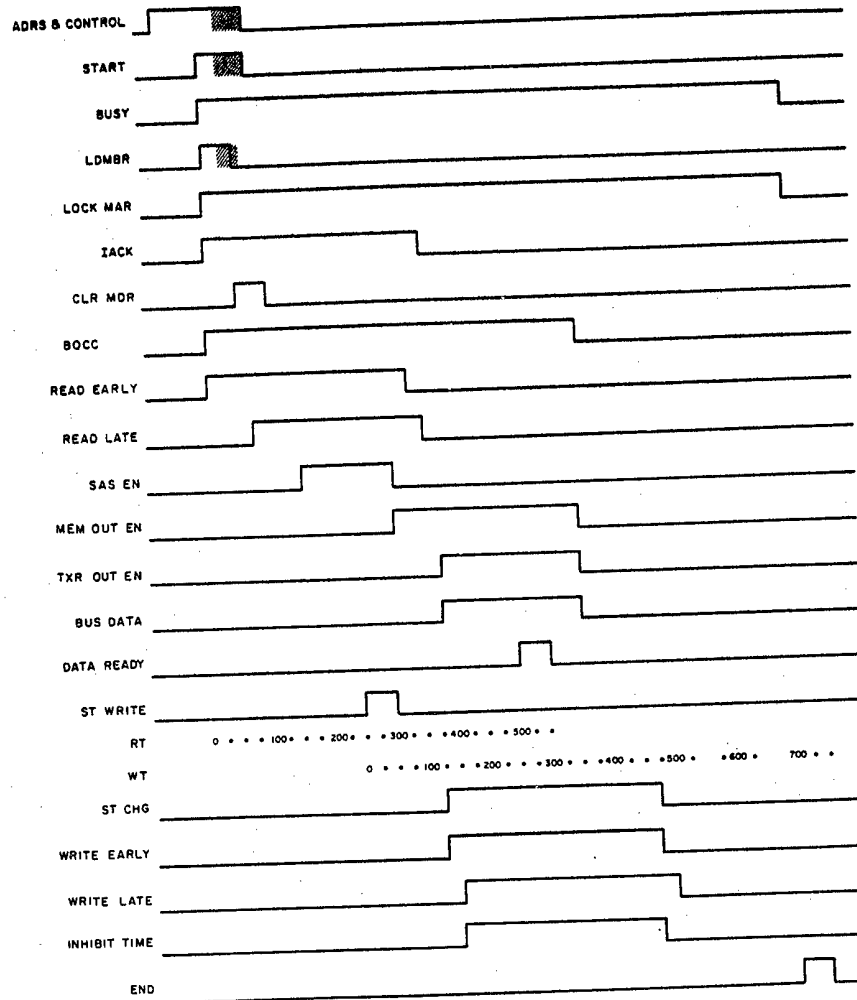


FIG. 15

1579061

COMPLETE SPECIFICATION

20 SHEETS

This drawing is a reproduction of
the Original on a reduced scale
Sheet 16

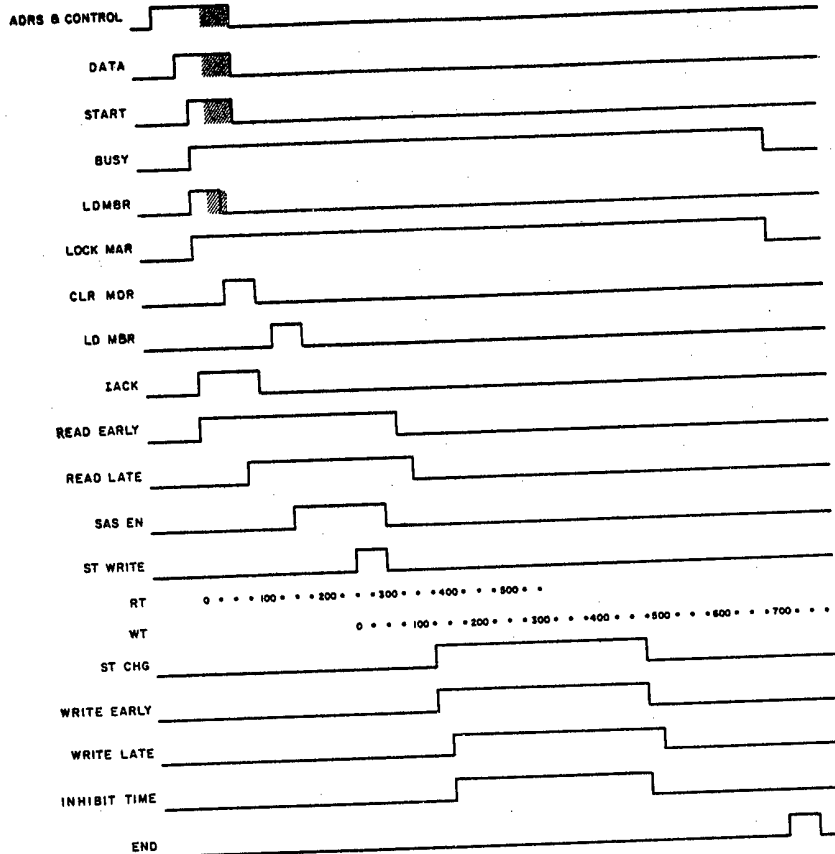


FIG. 16

1579061

COMPLETE SPECIFICATION

20 SHEETS

This drawing is a reproduction of
the Original on a reduced scale
Sheet 17

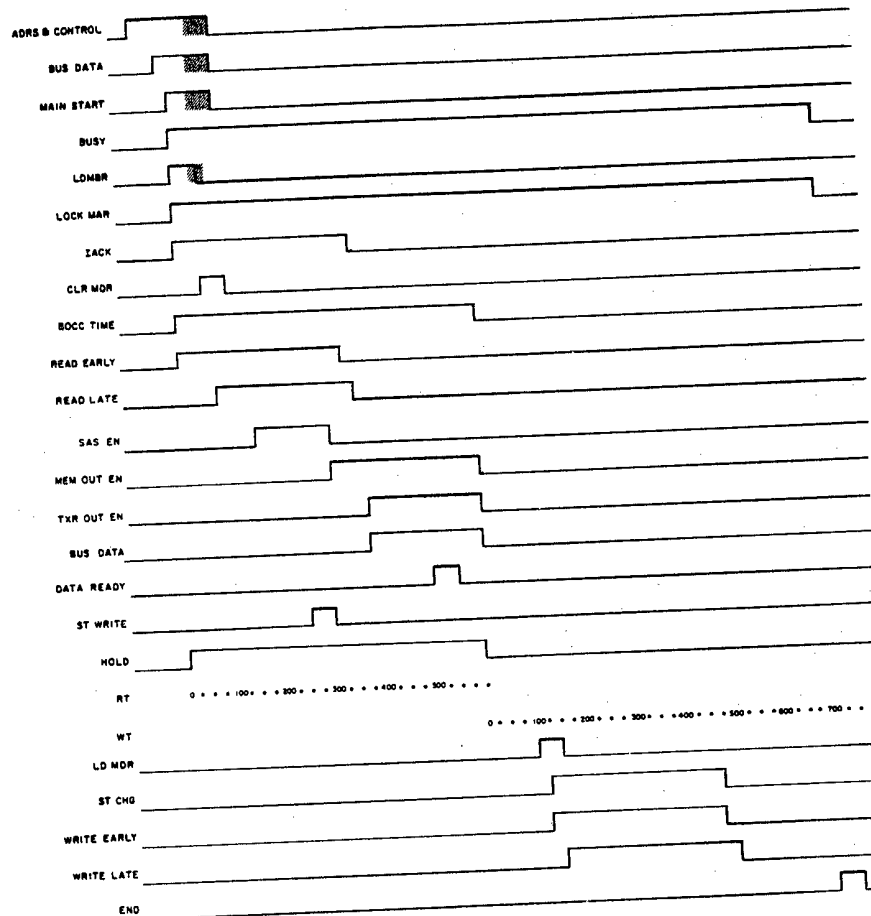


FIG. 17

FIG. 18A

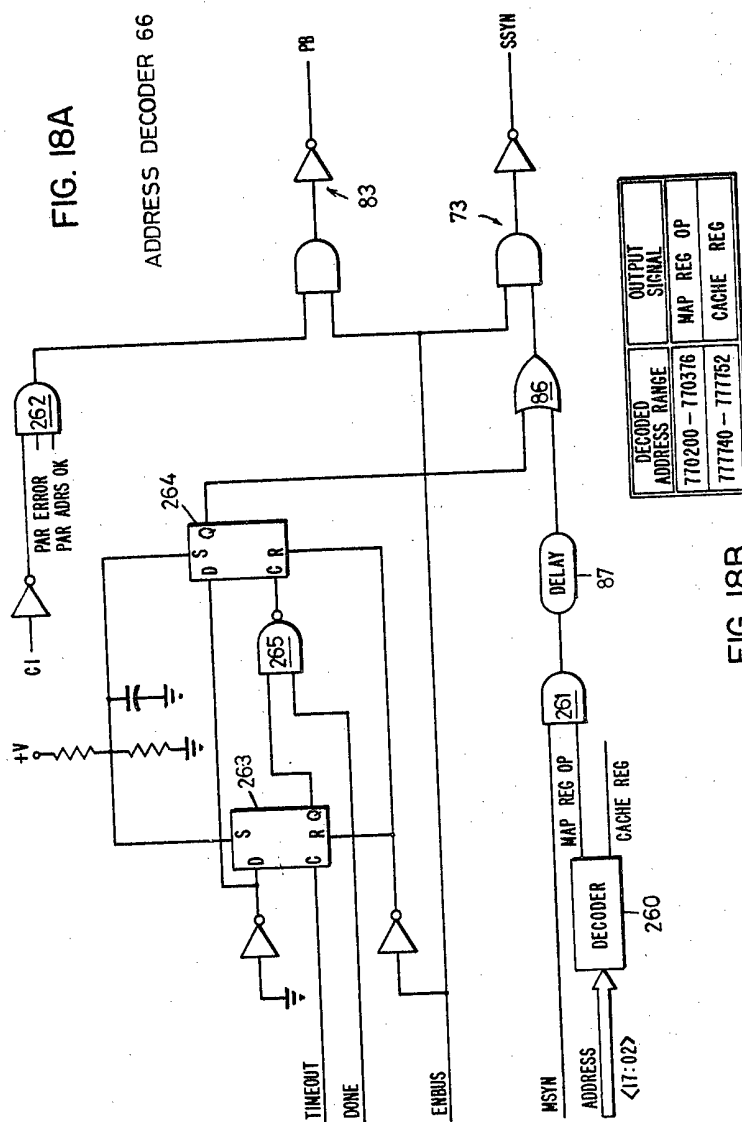


FIG. 18B

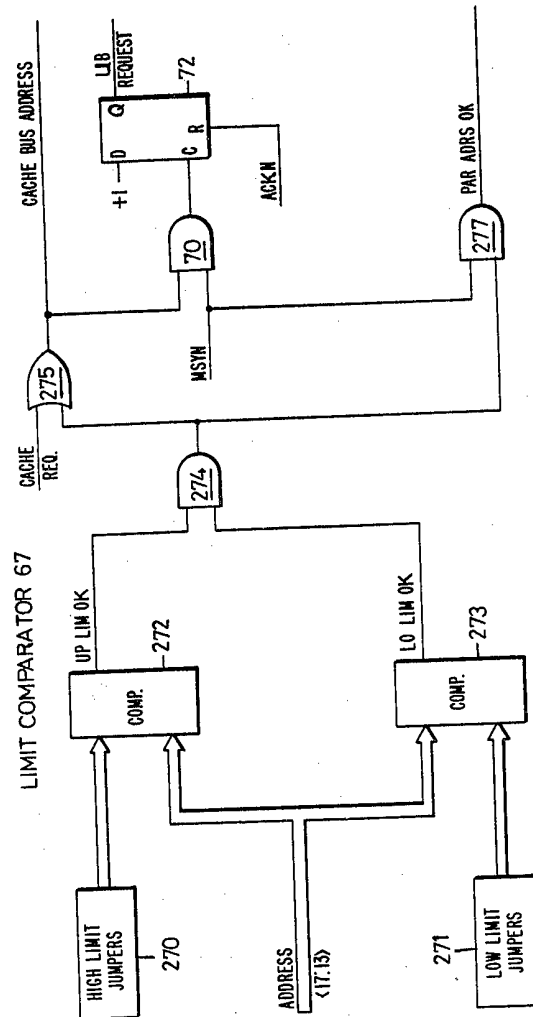


FIG. 19

