

CONTENT ADDRESSABLE MEMORYBACKGROUND OF THE INVENTION.

5 1. Field of the Invention.

The invention relates to a content addressable memory suitable for an address translation unit for memory management, particularly in a microprocessor system.

10 2. Prior Art.

There are many well-known mechanisms for memory management. In some systems, a larger address (virtual address) is translated to a smaller physical address. In others, a smaller address is used to access a larger
15 memory space, for instance, by using bank switching. The present invention relates to the former category, that is, where larger virtual address is used to access a limited physical memory.

20 In memory management systems, it is also known to provide various protection mechanisms. For example, a system may prevent a user from writing into an operating system or perhaps even from reading the operating system to external ports.

25 A content addressable memory is described in U.K. Patent No. 1457423, and it is the object of the invention to provide an improved content addressable memory suitable for use in memory management systems.

30 According to the present invention there is described a content addressable memory (CAM) comprising:

35 (a) a plurality of buffers, each for receiving first signals and providing said first signals and second signals, said second signals being complements of said first signals;

(b) a plurality of generally parallel pairs of lines, each pair being coupled to receive one of said first signals and one of said second signals;

5 (c) a plurality of memory cells coupled between each pair of lines, said cells being arranged in rows generally perpendicular to said pairs of lines;

10 (d) a plurality of row comparator lines, one associated with each of said rows of cells;

15 (e) a plurality of comparators, one for coupling between each of said memory cells, its respective pair of lines and one of said row comparators lines, said comparators for comparing a binary state stored on said memory cell with said first and second signals, each of said plurality of comparators being disabled when its respective pairs of lines are both maintained at a certain binary state, each of said plurality of comparators
20 comprising:

i) a first transistor coupled with said first line for receiving said first signals;

25 ii) a second transistor coupled with a first node of said memory cell for receiving the binary state of said first node;

iii) a third transistor coupled with said second line for receiving said second signal;

30 iv) a fourth transistor coupled with a second node of said memory cell for receiving the binary state of said second node;

(f) loading means for loading data from said pairs of line to said cells;

M

(g) preset means coupled with a first of said row comparator lines for bringing said first row comparator line to a first binary state.

BRIEF DESCRIPTION OF THE DRAWINGS



Figure 1 is a block diagram showing the overall architecture of the microprocessor in which the present invention is currently realized.

5 Figure 2 is a block diagram illustrating the segmentation mechanism embodied in the microprocessor of Figure 1.

Figure 3 is a block diagram illustrating the page field mapping for a hit or match in the page cache memory.

10 Figure 4 is a block diagram illustrating the page field mapping for no hit or match in the page cache memory of Figure 3. For this condition, the page directory and page table in main memory are used and, hence, are shown in Figure 4.

15 Figure 5 is a diagram used to illustrate the attributes stored in the page directory, page table page cache memory.

Figure 6 is a block diagram illustrating the organization of the content addressable memory of the present invention
20 and data storage contained within the page cache memory,

Figure 7 is an electrical schematic of a portion of the content addressable memory of Figure 6.

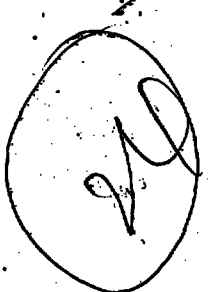
25 Figure 8 is an electrical schematic of the logic circuits associated with the detector of Figure 6.

5

10

20

25



The memory management makes use of both segmentation and paging. Segments are defined by a set of segment descriptor tables that are separate from the page tables used to describe the page translation. The two mechanisms are completely separate and independent. A virtual address is translated to a physical address in two distinct steps, using two distinct mapping mechanisms. A segmentation technique is used for the first translation step, and a paging technique is used for the second translation step. The paging translation can be turned off to produce a one-step translation with segmentation only, which is compatible with the 286.

Segmentation (the first translation) translates a 48-bit virtual address to a 32-bit linear (intermediate) address. The 48-bit virtual address is composed of a 16-bit segment selector, and a 32-bit offset within this segment. The 16-bit segment selector identifies the segment, and is used to access an entry from the segment descriptor table. This segment descriptor entry contains a base address of the segment, the size (limit) of the segment, and various attributes of the segment. The translation step adds the segment base to the 32-bit offset in the virtual address to obtain a 32-bit linear address. At the same time, the 32-bit offset in the virtual address is compared against the segment limit, and the type of the access is checked against the segment attributes. A fault is generated and the addressing process is aborted, if the 32-bit offset is outside the segment limit, or if the type of the access is not allowed by the segment attributes.

Paging (the second translation) translates a 32-bit linear

address to a 32-bit physical address using a two-level paging table, in a process described in detail below.

The two steps are totally independent. This permits a (large) segment to be composed of several pages, or a page to be composed of several (small) segments.

A segment can start on any boundary, and be of arbitrary size, and is not limited to starting on a page boundary, or to have a length that is an exact multiple of pages. This allows segments to describe separately protected areas of memory that start at arbitrary addresses and to be of arbitrary size.

Segmentation can be used to cluster a number of small segments, each with its unique protection attributes and size, into a single page. In this case, segmentation provides the protection attributes, and paging provides a convenient method of physical memory mapping a group of related units that must be protected separately.

Paging can be used to break up very large segments into small units for physical memory management. This provides a single identifier (the segment selector), and a single descriptor (the segment descriptor) for a separately protected unit of memory, rather than requiring the use of a multitude of page descriptors. Within a segment, paging provides an additional level of mapping that allows large segments to be mapped into separate pages that need not be contiguous in physical memory. In fact, paging allows a large segment to be mapped so that only a few pages at a time are resident in physical memory, with the remaining parts of the segment mapped onto disk. Paging also supports the definition of sub-structure within a large segment, for example, to write protect some pages of a large segment, while other pages can be written into.

Segmentation provides a very comprehensive protection model which works on the "natural" units used by a programmer: arbitrary sized pieces of linearly addressed memory. Paging provides the most convenient method for managing physical memory, both system main memory and backing disk memory. The combination of the two methods provides a very flexible and powerful memory protection model.

OVERALL MICROPROCESSOR ARCHITECTURE

In Figure 1, the microprocessor includes a bus interface unit 14. The bus unit includes buffers for permitting transmission of the 32-bit address signals and for receiving and sending the 32 bits of data. Internal to the microprocessor, unit 14 communicates over the internal bus 19. The bus unit includes a pre-fetch unit for fetching instructions from the RAM 12 and a pre-fetch queue which communicates with the instruction unit of the instruction decode unit 16. The queued instructions are processed within the execution unit 18 (arithmetic logic unit) which includes a 32-bit register file. This unit, as well as the decode unit, communicate with the internal bus 19.

The address translation unit 20 provides two functions; one associated with the segment descriptor registers, and the other with the page descriptor cache memory. The segment registers are for the most part known in the prior art; even so, they are described in more detail in conjunction with Figure 2. The page cache memory and its interaction with the page directory and page table stored within the main memory 13 is discussed in conjunction with Figures 3-7.

SEGMENTATION MECHANISM

The segmentation unit of Figure 1 receives a virtual

9

22

address from the execution unit 18 and accesses the appropriate register segmentation information. The register contains the segment base address which along with the offset from the virtual address are coupled over lines 23 to the page unit.

5 Figure 2 illustrates the accessing of the tables in main memory when the segmentation registers are loaded with mapping information for a new segment. The segment field indexes the segment descriptor table in the main memory 13. The contents of the table provide a base address and additionally, provide
10 attributes associated with the data in the segment.

The offset is compared to the segment limit in comparator 27; the output of this comparator providing a fault signal. The adder 26 which is part of the microprocessor combines the base and offset to provide a "physical" address on lines 31.

15 This address may be used by the microprocessor as a physical address or used by the paging unit. This is done to provide compatibility with certain programs written for a prior microprocessor (Intel 286). For the Intel 286, the physical address space is 24 bits.

20 The segment attributes including details on the descriptors employed such as the various privilege levels are set forth in U.S. Patent 4,442,484.

The fact that the segmentation mechanism is known in the prior art is represented in Figure 2 by the dotted line 28 which
25 indicates the prior art structures to the left of the dotted line.

The page field mapping block 30 which includes the page unit of Figure 1 as well as its interaction with the page directory and page table stored in the main memory is shown in Figures 3 through 7.

30 While the

segmentation mechanism uses shadow registers, it also could be implemented with a cache memory as is done with the paging mechanism.

PAGE DESCRIPTOR CACHE MEMORY

5 In Figure 3 the page descriptor cache memory of the page unit 22 of Figure 1 is shown within dotted line 22a. This memory comprises two arrays, a content addressable memory (CAM) 34 and a page data (base) memory 35. Both memories are implemented with static memory cells. The organization of memories 34 and 35 is
10 described in conjunction with Figure 6. The specific circuitry used for CAM 34 with its unique masking feature is described in conjunction with Figures 7 and 8.

The linear address from the segment unit 21 are coupled to the page unit 22 of Figure 1. As shown in Figure 3, this linear
15 address comprises two fields, the page information field (20 bits) and a displacement field (12 bits). Additionally, there is a four bit page attribute field provided by the microcode. The 20-bit page information field is compared with the contents of the CAM 34. Also, the four attribute bits ("dirty", "valid", "U/S", and
20 "W/R") must also match those in the CAM before a hit occurs. (There is an exception to this when "masking" is used as will be discussed.)

For a hit condition, the memory 35 provides a 20-bit base word which is combined with the 12-bit displacement field of the
25 linear address as represented by summer 36 of Figure 3 and the resultant physical address selects from a 4k byte page frame in main memory 13.

PAGE ADDRESSING FOR THE NO-HIT CONDITION

A page directory 13a and a page table 13b are stored in the main memory 13 (see Figure 4). The base address for the page directory is provided from the microprocessor and is shown in Figure 4 as the page directory base 38. Ten bits of the page information field are used as an index (after being scaled by a factor of 4) into the page directory as indicated by the summer 40 in Figure 4. The page directory provides a 32-bit word. Twenty bits of this word are used as a base for the page table. The other 10 bits of the page information field are similarly used as an index (again being scaled by a factor of 4) into the page table as indicated by the summer 41. The page table also provides a 32-bit word, 20 bits of which are the page base of the physical address. This page base address is combined as indicated by summer 42 with the 12-bit displacement field to provide a 32-bit physical address.

Five bits from the 12-bit fields of the page directory and table are used for attributes particularly "dirty", "access", "U/S", "R/W" and "present". These will be discussed in more detail in conjunction with Figure 5. Remaining bits of this field are unassigned.

The stored attributes from the page directory and table are coupled to control logic circuit 75 along with the 4 bits of attribute information associated with the linear address. Parts of this logic circuit are shown in subsequent figures are discussed in conjunction with these figures.

PAGE DIRECTORY ATTRIBUTES

In Figure 5 the page directory word, page table word and CAM word are again shown. The protective/control attributes assigned to the four bits of the page directory word are listed

within bracket 43. The same four attributes with one additional attribute are used for the page table word and are set forth within bracket 44. The four attributes used for the CAM word are set forth within bracket 45.

5 The attributes are used for the following purpose:

1. DIRTY. This bit indicates whether a page has been written into. The bit is changed once a page has been written into. This bit is used, for instance, to inform the operating system that an entire page is not "clean". This bit is stored in
10 the page table and in the CAM (not in the page directory). The processor sets this bit in the page table when a page is written into.

2. ACCESSED. This bit is stored in only the page directory and table (not in the CAM) and is used to indicate that
15 a page has been accessed. Once a page is accessed, this bit is changed in the memory by the processor. Unlike the dirty bit, this bit indicates whether a page has been accessed either for writing or reading.

3. U/S. The state of this bit indicates whether the
20 contents of the page is user and supervisory accessible (binary 1) or supervisor only (binary zero).

4. R/W. This read/write protection bit must be a binary 1 to allow the page to be written into by a user level program.

5. PRESENT. This bit in the page table indicates if the
25 associated page is present in the physical memory. This bit in the page directory indicates if the associated page table is present in physical memory.

6. VALID. This bit which is stored only in the CAM is used to indicate if the contents of the CAM is valid. This bit is
30 set to a first state on initialization then changed when a valid

CAM word is loaded.

The five bits from the page directory and table are coupled to control logic circuit 75 to provide appropriate fault signals within the microprocessor.

- 5 The user/supervisor bits from the page directory and table are logically ANDed as indicated by gate 46 to provide the R/W bit stored in the CAM 34 of Figure 3. Similarly, the read/write bits from the page directory and table are logically ANDed through gate 47 to provide the W/R bit stored in the CAM. The dirty bit from
10 the page table is stored in the CAM. These gates are part of the control logic 75 of Figure 4.

The attributes stored in the CAM are "automatically" tested since they are treated as part of the address and matched against the four bits from the microcode. A fault condition
15 results even if a valid page base is stored in the CAM, if, for instance, the linear address indicates that a "user" write cycle is to occur into a page with R/W=0.

- The ANDing of the U/S bits from the page directory and table ensures that the "worst case" is stored in the cache memory.
20 Similarly, the ANDing of the R/W bit provides the worst case for the cache memory.

ORGANIZATION OF THE PAGE DESCRIPTOR CACHE MEMORY

2

The CAM 34 as shown in Figure 6 is organized in 8 sets with 4 words in each set. Twenty-one bits (17 address and 4 attributes) are used to find a match in this array. The four comparator lines from the four stored words in each set are connected to a detector. For instance, the comparator lines for the four words of set 1 are connected to detector 53. Similarly, the comparator lines for the four words in sets 2 through 8 are connected to detectors. The comparator lines are sensed by the detectors to determine which word in the set matches the input (21 bits) to the CAM array. Each of the detectors contains "hard wired" logic which permits selection of one of the detectors depending upon the state of the 3 bits from the 20-bit page information field coupled to the detectors. (Note the other 17 bits of this bit page information field is coupled to the CAM array.)

For purposes of explanation, eight detectors are implied from Figure 6. In the current embodiment only one detector is used with the three bits selecting one set of four lines for coupling to the detector. The detector itself is shown in Figure 8.

The data storage portion of the cache memory is organized into four arrays shown as arrays 35a-d. The data words corresponding to each set of the CAM are distributed with one word being stored in each of the four arrays. For instance, the data word (base address) selected by a hit with word 1 of set 1 is in array 35a, the data word selected by a hit with word 2 of set 1 is in array 35b, etc. The three bits used to select a detector are also used to select a word in each of the arrays. Thus, simultaneously, words are selected from each of the four arrays.

The final selection of a word from the arrays is done through the multiplexer 55. This multiplexer is controlled by the four comparator lines in the detector. (33)

When the memory cache is accessed, the matching process which is a relatively slow process begins through use of the 21 bits. The other three bits are able to immediately select a set of four lines and the detector is prepared for sensing a drop in potential on the comparator lines. (As will be discussed, all the comparator (rows) lines are precharged with the selected (hit) line remaining charged while the non-selected lines discharge.) Simultaneously, four words from the selected set are accessed in arrays 35a-35d. If and when a match occurs, the detector is able to identify the word within the set and this information is transmitted to the multiplexer 55 allowing the selection of the data word. This organization improves access time in the cache memory.

CONTENT ADDRESSABLE MEMORY (CAM)

In Figure 7, the 21 bits which are coupled to the CAM array are again shown with 17 of the bits being coupled to the complement generator and override circuit 56 and with the 4 attribute bits coupled to the VUDW logic circuit 57. The 3 bits associated with the selection of the detectors described in conjunction with Figure 6 are not shown in Figure 7.

The circuit 56 generates the true and complement signal for each of the address signals and couples them to parallel lines in the CAM array, such as lines 59 and 60. Similarly, the VUDW logic 57 generates both the true and complement signals for the attribute bits and couples them to parallel lines in the array. The lines 59 and 60 are duplicated for each of the true and complement bit lines (i.e., 21 pairs of bit and bit/ lines).

34

Each of the 32 rows in the CAM array has a pair of parallel row lines such as lines 68 and 70. An ordinary static memory cell such as cell 67 is coupled between each of the bit and bit/ lines (columns) and is associated with the pair of row lines.

5 In the presently preferred embodiment, the memory cells comprise ordinary flip-flop static cells using p-channel transistors. One line of each pair of row lines (line 70) permits the memory cell to be coupled to the bit and bit/ line when data is written into the array. Otherwise, the content of the memory cell is compared
10 to the data on the column lines and the results of the comparison is coupled to the hit line 68. The comparison is done by comparators, one associated with each cell. The comparator comprises the n-channel transistors 61-64. Each pair of the comparator transistors, for example, transistors 61 and 62, are
15 coupled between one side of the memory cell and the opposite bit line.

Assume that data is stored in the memory cell 67 and that the node of the cell closest to bit line 59 is high. When the contents of the CAM are examined, first the hit line 68 is
20 precharged through transistor 69. Then the signals coupled to the CAM are placed on the column lines. Assume first that line 59 is high. Transistor 62 does not conduct since line 60 is low. Transistor 63 does not conduct since the side of the cell to which it is connected is low. For these conditions, line 68 is not
25 discharged, indicating that a match has occurred in the cell. The hit line 68 provides ANDing of the comparisons occurring along the row. If a match does not occur, one or more of the comparators will cause the hit line to discharge.

During precharging the circuits 56 and 57 generate an
30 override signal causing all column lines (both bit and bit/) to be

17

low. This prevents the comparators from draining the charge from the hit lines before the comparison begins.

It should be noted that the comparators examine the "binary one" condition and, in effect, ignore the "binary zero" condition. That is, for instance, if the gate of transistor 64 is high (line 59 high) then transistors 63 and 64 control the comparison. Similarly, if the bit/ line 60 is high, then transistors 61 and 62 control the comparison. This feature of the comparator permits cells to be ignored. Thus, when a word is coupled to the CAM, certain bits can be masked from the matching process by making both the bit and bit/ line low. This makes it appear that the contents of the cell match the condition on the column lines. This feature is used by the VUDW logic circuit 57.

Microcode signals coupled to logic circuit 57 causes the bit and bit/ line for selected ones of the attribute bits to be low as a function of the microcode bits. This results in the attribute associated with that bit to be ignored. This feature is used, for instance, to ignore the U/S bit in the supervisory mode. That is, the supervisory mode can access user data. Similarly, the read/write bit can be ignored when reading or when the supervisory mode is active. The dirty bit is also ignored when reading. (The feature is not used for the valid bit.)

When the attribute bits are stored in main memory, they can be accessed and examined and logic circuits used to control accessing, for instance, based on the one or zero state of the U/S bit. However, with the cache memory no separate logic is used. The forcing of both the bit and bit/ lines low, in effect, provides the extra logic by allowing a match (or preventing a fault) even though the bit patterns of the attribute bits are not matched.

36

The detector from Figure 6, as shown in Figure 8, includes a plurality of NOR gates such as gates 81, 82, 83 and 84. Three of the hit lines from the selected set of CAM lines are coupled to gate 81; these are shown as lines A, B, and C. A different combination of the lines are connected to each of the other NOR gates. For instance, NOR gate 84 receives the hit lines D, A, and B. The output of each of the NOR gates is an input to a NAND gate such as NAND gate 86. A hit line provides one input to each NAND gate. This line is the one (of the four A,B,C,D) that is not an input to the NOR gate. This is also the bit line from the set entry to be selected. For example, gate 86 should select the set that is associated with hit line D. For instance, in the case of NOR gate 81, hit line D is coupled to the NAND gate 86. Similarly, for the NAND gate 90, the hit line C in addition to the output of gate 84, are inputs to this gate. An enable read signal is also coupled to the NAND gates to prevent the outputs of this logic from being enabled for a write. The output of the NAND gates, such as line 87, are used to control the multiplexer 55 of Figure 6. In practice, the signal from the NAND gate, such as the signal on line 87, controls the multiplexer through p-channel transistors. For purposes of explanation, an additional inverter 88 is shown with an output line 89.

The advantage to this detector is that it enables precharge lines to be used in the multiplexer 55. Alternately, a static arrangement could be used, but this would require considerably more power. With the arrangement as shown in Figure 8, the output from the inverters will remain in the same state until one of the hit lines drops in potential. When that occurs, only a single output line will drop in potential, permitting the multiplexer to select the correct word.

19

The present application has been divided out of our
copending U.K. Patent Application No. 8519991 in which
there is described and claimed address translation units for
memory management, particularly in a microprocessor system.

38

20

CLAIMS.

1. A content addressable memory (CAM) comprising:

(a) a plurality of buffers, each for receiving first signals and providing said first signals and second signals, said second signals being complements of said first signals;

(b) a plurality of generally parallel pairs of lines, each pair being coupled to receive one of said first signals and one of said second signals;

(c) a plurality of memory cells coupled between each pair of lines, said cells being arranged in rows generally perpendicular to said pairs of lines;

(d) a plurality of row comparator lines, one associated with each of said rows of cells;

(e) a plurality of comparators, one for coupling between each of said memory cells, its respective pair of lines and one of said row comparators lines, said comparators for comparing a binary state stored on said memory cell with said first and second signals, each of said plurality of comparators being disabled when its respective pairs of lines are both maintained at a certain binary state, each of said plurality of comparators comprising:

i) a first transistor coupled with said first line for receiving said first signal;

ii) a second transistor coupled with a first node of said memory cell for receiving the binary state of said first node;

iii) a third transistor coupled with said second line for receiving said second signal;



iv) a fourth transistor coupled with a second node of said memory cell for receiving the binary state of said second node;

5 (f) loading means for loading data from said pairs of lines to said cells;

(g) preset means coupled with a first of said row comparator lines for bringing said first row comparator
10 line to a first binary state.

2. The CAM defined by Claim 1 wherein said row comparator lines are precharged lines.

15 3. The CAM defined by Claim 2 including a storage memory which comprises a plurality of sections and wherein data is accessed simultaneously in all of said sections and an output from one of said sections being selected through said row lines.

20 4. The CAM defined by Claim 3 including detectors coupled to a predetermined number of said row lines, said detectors for sensing which one of said predetermined number of lines remains charged.

25 5. The CAM defined by Claim 4 wherein said selection of said output from one of said sections is made by said detectors.

30 6. A content addressable memory (CAM) as claimed in Claim 1, including detector means for detecting the binary state on said plurality of row comparator lines, said detector means being coupled to receive signals on a selected set of said row comparator lines, said detector
35 means having a NOR gate coupled with a first of said row



22

comparator lines, said detector means further having
a NAND gate coupled with a second of said row comparator
lines and an output of said NOR gate.



* C L H S *