2-1/2D core search memory

by PHILIP A. HARDING and MICHAEL. W. ROLUND

Bell Telephone Laboratories, Inc.
Naperville, Illinois

Usual memories allow addressing of individual word lines with each word line containing an assemblage of bits. Bit detectors which can sense all words are utilized to read the word contents. Associative memories allow bit lines to be addressed, as well as word lines, to determine which set of words match the input bit states. Such memories are useful because they eliminate time consuming word hunting in table look up operations, and because they allow easy access to specific words highlighted by activity or flag bits.

Unfortunately, most associative memories proposed are expensive. Possibly circuits based on large scale integration may allow low cost associative operations, but they are not economically feasible today. Core associative memories rarely have been proposed; those described have complex memory mat structures.1 In most cases, the cost penalties far outweigh the usefulness of the true associative array.

The semiasociative solution, one in which a segment of any single bit line in an array can be addressed to read out the corresponding bit locations for a number of words may be the compromise that finds a wide range of applications.2 This type of memory, defined as a single bit search memory, is symbolically illustrated in Figure 1. Such a memory may be operated in the ordinary sense; one of the “n” unique addresses can be selected to read, the “m” bit word contents. Similarly, a unique address may be chosen and the m bits can be independently written. The figure illustrates the selection of address “A,” either for reading or writing the “m” bit contents.

In the search mode, a single block of K address locations associated with any single word bit, Bk, may be selectively read or written. The figure highlights the Bkth bit of words 1 through K in block 1 and the Bkth bit words of SK through (S + 1)K in block S as possible search words. The entire word field of n can be arranged in n/K blocks of K words each to facilitate the search mode. It has been found that a modified 2 wire or 3 wire 2-1/2D Core Memory can economically achieve the single bit search memory function if only one m-bit normal address word or one K-bit search word is selected during one memory cycle.

2-1/2D, 2-wire memory

A single bit search memory can be thought of as an extension of the conventional, 2-1/2D, 2-wire core memory shown in Figure 2, which was reported on in an earlier paper.3 In a 2-wire, coincident current memory, the readout voltage from a particular core must obviously be sensed across one of the two drive wires intersecting the core. In the memory of Figure 2, the core array is composed of a front and rear plane, with the readout voltage being sensed differentially across a pair of selected bit wires, one in each plane. The center-tapped connection of the bit readout transformer also forces the bit drive current to divide equally between the front and rear planes. The group selection circuit provides a virtual ground to the selected pair of memory wires in each bit while simultaneously isolating the nonselected wires, which are multiplied to the readout transformers at the top of the array. Hence, any noise voltages induced on the nonselected wires are not coupled into the readout. The group selection circuit is formed of diode rails connected in a driven bridge configuration.
of the type described in Reference 4.

The word access consists of a diode matrix driving folded word loops each of which intersects two cores on a given bit line. Since the word and bit currents will add in one of the cores and cancel in the other, the direction of word current is used to select one core or the other, thus reducing the number of word access circuits required by a factor of two. The looping of the word wire has a number of additional advantages. The cores can be oriented in-line, rather than in a diamond pattern, increasing packing density. The shuttle voltages due to word current tend to cancel. And finally, no more than half of the cores on a bit line can be disturbed by word current into a worst case delta noise state. The word wiring scheme of Figure 3 lacks the shuttle

A word is read out of this memory in the following fashion. Bit read current is applied first causing a large noise spike in the readout. When this noise has expired, the word current is applied, reading out the state of the m bits of the word. The word is rewritten into memory by reversing the word current and applying reverse bit write current to those locations where a "one" is to be stored. This timing is indicated in Figure 4.

Single bit search memory

The word access of Figure 2 supplies current to only one selected word loop. At little additional cost, we can perform the same selection process with an access that is virtually identical to that used in the bit dimension, as shown in Figure 3. We can select a pair of word lines (rather than a single loop) by energizing the appropriate word driver and word group selection circuit. Note that the cores on the front and rear planes are oppositely phased with respect to the word current so that only one core is selected per bit. However, we now have the added flexibility of being able to readout in the word dimension. That is, rather than energizing m bit drivers and then a single word driver and finally sensing the readout on m pairs of bit wires, we energize all K word drivers first and then a single bit driver while sensing the readout on K pairs of word wires. Thus, we can readout the state of K different word locations of a given bit in a "search" mode, or all m bits of a word in the normal or "address" mode. The timing for the search mode is illustrated in Figure 5. The readout wave form is the same in either mode, but appears across the bit transformers in the address mode and across the word transformers in the search mode.

4 plane single bit search

The word wiring scheme of Figure 3 lacks the shuttle
and delta noise reduction advantages of the word line looping of Figure 2. These advantages can be regained by using the four-plane configuration of Figure 6. The word and bit lines are wired so that only one plane receives a simultaneous word and bit current. This is indicated in Table 1.

**TABLE 1—4 memory plane wiring**

<table>
<thead>
<tr>
<th>CORE SELECTED</th>
<th>BIT LINES</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLANE 1</td>
<td>PLANES 1, 2</td>
</tr>
<tr>
<td>PLANE 2</td>
<td>PLANES 1, 2</td>
</tr>
<tr>
<td>PLANE 3</td>
<td>PLANES 3, 4</td>
</tr>
<tr>
<td>PLANE 4</td>
<td>PLANES 3, 4</td>
</tr>
</tbody>
</table>

**Search memory operation**

Let us consider the operation of a search memory of \( n \) words, having \( m \) bits per word, and with the word access divided into \( K \) segments (each segment having an independent driver, detector, and register cell). The block diagram of such a memory is shown in Figure 7.
In a normal or address mode, the address to be interrogated is supplied in two parts; \( \log_{2} \frac{n}{K} \) address bits are supplied to the address register which controls the bit group and word group selector circuits. As an example let us assume equal bit and word group selector circuits. Then the bit group selector circuit decodes \( \frac{1}{2} \log_{2} \frac{n}{K} \) inputs and selects one of the \( \sqrt{\frac{n}{K}} \) groups of \( m \) bits each. The word group selector decodes the remaining \( \frac{1}{2} \log_{2} \frac{n}{K} \) inputs and selects one of the \( \sqrt{\frac{n}{K}} \) groups of \( K \) bits each.

The remaining \( 1 \) out of \( K \) address selection is performed via the word data input, which energizes the appropriate \( 1 \) out of \( K \) word drivers. All \( m \) bit drivers are activated and readout is accomplished via the \( m \) bit detectors.

In a search read mode, \( \log_{2} \frac{n}{K} \) address bits are again supplied to the address register to control the group selector circuits. However, now the roles of the word data and bit data inputs are interchanged. All \( K \) word drivers are energized, corresponding to the \( K \) addresses over which the search is to be made, while only a single bit driver is energized, dependent on which bit is to be searched. Readout is accomplished via the \( K \) word detectors.

**Flag bit memory**

As mentioned earlier, one of the attractive applications of a search memory is in the reduction of hunting for an active or "flagged" word. If a single bit of the word is reserved to indicate activity in the remainder of the word, that particular bit can be called a "flag" bit. If one performs a search read on the flag bit, then those locations in the word register which are set will correspond to the words which are "active," and subsequent normal reads can be used, with the active word locations automatically stored in the word register, to determine the entire contents of the active words. If multiple words are flagged in a word block, some form of priority selection may be necessary. In the case of few active words, the flag bit search can reduce the hunting time by a factor of \( K \). In practice, the reduction can be as great as one or two orders of magnitude.

**Ripple search**

In the case where more than a single flag bit is required to locate a desired address in memory, a "ripple search" technique can be employed. This simply involves sequentially searching through the flag bits, eliminating all addresses which mismatch on any of the desired flag bits. The matching operation is readily accomplished by using a word register such as the one shown in Figure 8. The register is initially set to the "1's" state by a timing pulse. If the readout from a given word detector mismatches the match bit \( B_j \), the corresponding flip-flop in the word register is reset. On the subsequent search read, the match bit becomes the second desired flag bit \( B_{j+1} \), and those readouts which mismatch will again reset their corresponding word register flip-flops. Thus, any word flip-flop which remains set after all of the flag bits have been searched corresponds to an address whose flag bits match all of the desired flag bits.

The following example further illustrates the ripple search technique. Suppose that the desired flag bits \( B_j \) through \( B_{j+3} \) are 1001 and that the memory contents
being searched at address block S is as shown in Figure 9. The word register is initially set. Flag bit position \( B_j \) is then searched, looking for those addresses in which a "1" is stored. Since only addresses \( SK + 2 \) and \( SK + 3 \) match, the remaining word register positions 1, 4 and \( K \) are reset. The next search is made on position \( B_j + 1 \), looking for locations containing a "0." The only mismatch occurs at the \( SK + 4 \) address or 4th bit of the register, but since this position was reset on the previous read, no change occurs. Note that 1st bit remains reset although it matched on \( B_j + 1 \). Subsequent search reads on \( B_j + 2 \) and \( B_j + 3 \) indicate that only address \( SK + 3 \) matches on all four desired flag bits.

Multiple bit searching

A completely associative array has the flexibility of addressing a number of bit line segments, \( r \), matching the memory contents with the input bit data, and selecting the words whose contents agree with the \( r \) data input bits. Such a memory which is fully associative over \( K \) words at a time is symbolically defined in Figure 10. Here the memory is segregated into \( r \) regions; a search word can be simultaneously written into or read out of each region in the search mode, but the memory can select a normal word, \( A_i \), in the standard address mode.

A block diagram representation of this scheme is shown in Figure 11 where each of the search memory blocks are identical to the single bit search memory described in Figures 7 and 8; the only deviation is the reduction of the \( m \) data bits to \( m/r \) and the reduction of the core array by a factor of \( r \). Since there are \( r \) memory blocks, the total number of cores and bit circuits do not increase over the single bit search, but the \( K \) word detectors and word register circuits increase by a factor of \( r \). Thus, it is possible to simultaneously read or write \( rK \) bits in the search mode. Although this approach appears at first to be expensive, there are a number of characteristics which make it economically attractive. First, the \( r \) search memory blocks can be recombined to form a common memory array and access with a more optimum bit to word line aspect ratio. Secondly, if the basic memory size is large, it is necessary to segment the array and replicate the circuitry in any event in order to maintain high speeds and good noise margin.

The relative costs of an 800,000 bit and a 6,400,000 bit memory as a function of the number of simultaneous search bits are plotted in Figure 12. For example, the single bit search mode for a 800,000 bit memory costs 10% more than a standard no searching 2-1/2D core memory. A simultaneous search capability of up to 8 bits will double the cost of the store. However, a 6,400,000 bit store cost is hardly affected by the search mode feature.

Applications

A single bit search memory has been proposed which can locate and read out a "flagged" word in 2 memory cycles. By means of the appropriate word logic, the memory can be used to make a ripple search in a few more memory cycles. In addition to the searching operations listed, this type of memory can be used in "pattern recognition" computers; it can be used as a serial to parallel or parallel to serial converter for massive streams of data; it can be used for matrix transposition and a number of possible multiple word operations such as simultaneous write into \( K \) words.

Extension of this memory into an \( r \) bit search allows true associative operation for word block sizes in the range of 10 to 100 words without a serious cost penalty.
for large sized memories.

The memory described is basically a 2-1/2D-2-wire core store; it therefore has the speed, size and cost limitations inherent to such systems. Microsecond operation for a few million bits is entirely feasible. However, it is also possible to extend the scheme to 3-wire systems for increased speed capability.

REFERENCES
1 W L McDermid  H E Peterson
A magnetic associative memory system
The IBM Journal of Research and Development Vol 5 No 1
January 1961 pp 59-62
2 H S Stone
Associative processing for general purpose computers through the use of modified memories
1968 Fall Joint Computer Conference
3 P A Harding  M W Rolund
Novel low cost design for 2-1/2D storage systems
1967 Solid State Circuits Conference Digest of Technical Papers Vol X IEEE Catalog No 4C-49 pp 82-83
4 P A Harding  M W Rolund
Bit access problems in 2-1/2D 2-wire memories