July 1971 Brief 71-10218

NASA TECH BRIEF
Marshall Space Flight Center

NASA Tech Briefs announce new technology derived from the U.S. space program. They are issued to encourage commercial application. Tech Briefs are available on a subscription basis from the National Technical Information Service, Springfield, Virginia 22151. Requests for individual copies or questions relating to the Tech Brief program may be directed to the Technology Utilization Office, NASA, Code KT, Washington, D.C. 20546.

Efficient Digital Comparison Technique for Logic Circuits

The problem:

Error detection and error masking schemes in highly reliable digital systems frequently involve comparing the outputs of simultaneously operating redundant units. Predeterminable variations can occur between output values due to asynchronous device operation and analog device variations. Because of the carry propagation characteristics of the binary numbering system, a bit-by-bit comparison of the binary output values may indicate significant discrepancies when the quantities they represent are within a close tolerance. The problem, then, is to devise a "tolerance compare" technique to indicate a "discompare" only when the numerical difference value exceeds a prescribed limit.

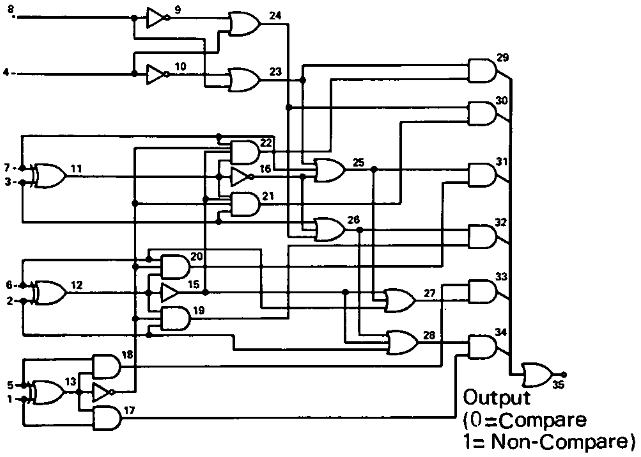

Figure 1. Logic for Four-Bit Word-Length Comparator

The solution:

An algorithm involving binary-number properties is defined, in lieu of an arithmetic operation which requires relatively complex circuitry. The binary-number property concerned is described as follows: Given a case where any binary number is to be incremented by one unit of its low-order bit position. If the low-order bit is zero, the number is incremented to one and there is no carry. If the low-order bit is one, the number becomes a zero, as do all successive higher-order bit positions containing ones. The carry propagates through all bit positions containing ones until a position containing zero is encountered. The zero becomes a one, and all higher-order bit positions remain unchanged. Using this definition property of the binary number system, a tolerance-compare algorithm of two binary numbers is implemented as follows:

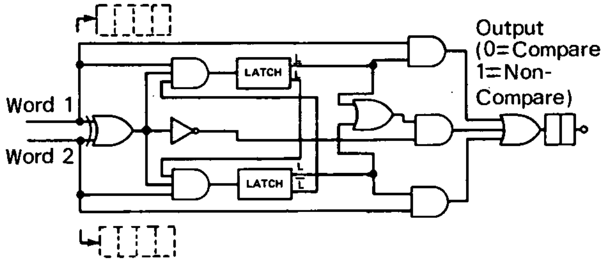

Figure 2. Two-Word Serial Compare

1. Determine the highest-order bit position of disagreement between the two numbers, i.e., the MARK position.
2. Define the following: If the two numbers agree within one lower-order unit value, then:
   A. The larger of the two numbers has a one in the MARK position and zeros in all lower-order positions.
   B. The smaller number has a zero in the MARK position and all ones in positions of lower order than the MARK position.

This algorithm may be extended to encompass tolerances other than one unit. In a given system of word length n, if only the n-1, n-2, n-3, . . . higher-order bits are compared, the tolerance-compare circuitry will

(continued overleaf)

This document was prepared under the sponsorship of the National Aeronautics and Space Administration. Neither the United States Government nor any person acting on behalf of the United States Government assumes any liability resulting from the use of the information contained in this document, or warrants that such use will be free from privately owned rights.

indicate a discompare only when the difference is greater than 2, 4, 8, . . . units, respectively.

How it's done:

Figure 1 shows the straightforward implementation, in a parallel organized system, using AND, OR, and INVERT logic for a four-bit word length comparator. Word 1 inputs, in high to low order, are on lines 1, 2, 3, and 4. Word 2 inputs, in high to low order, are on lines 5, 6, 7, and 8. Three exclusive OR's compare the higher-order bit positions. No compare is necessary for the low-order bit position. The AND gates immediately following the exclusive OR gates determine the MARK position and the larger word. This function is accomplished by the concurrence of: (1) a 1 from the exclusive OR (indicating a discompare of the bits in that position); (2) a 1 from the inverted output(s) from all higher-order exclusive OR's (indicating all higher-order bits compare); and (3) a 1 from the larger-word bit. Thus, only one of the AND gates will have an output, and that output will correspond to the MARK position of the larger-valued word.

When the MARK position (i.e., the larger-valued word) is identified, the OR gates 23 through 28 and the AND gates 29 through 34 logically determine that the lower-order bit values of the larger word are zeros and the lower-order bit values of the smaller words are complementary (i.e., ones). There is an OR gate and an AND gate for each bit position of each word being compared. Any AND gate which has a 1 output will cause a discompare; however, only that AND gate corresponding to the larger-word MARK position can have a logical 1 output. The remaining condition is obtained from the OR gate (e.g., gate 26 above AND gate 32, OR gate 24 above AND gate 30, etc.). These OR gates have a logical 1 output whenever: (1) a corresponding lower OR gate has a 1 output; (2) the input word bit they represent is a logical 1; or (3) there is a compare of the word bits in the position they represent.

Application to serial systems illustrated in Figure 2, is also straightforward. Assume that two words of any length are to be compared and are transmitted synchronously over two serial data lines in high-order to low-order bit sequence. Though not essential to this application, the words are frequently gathered in shift registers (dotted in figure). The serial implementation shown uses two latches because all required information on the words is not available simultaneously; therefore, a minimal memory capacity must be included. As the words are shifted into the first bit position of the register, a comparison is made of corresponding bits, using exclusive OR gates. The first discompare indicates the MARK position, and the respective AND gates set one of the two latches to correspond to the larger-valued word. Subsequently, any 1 bits in the larger word, or a failure to get a 1 output from the exclusive OR (indicating a 0 in one of the smaller-valued word bits), will cause a 1 output from the final OR gate to the modulo 2 counter. The modulo 2 counter prevents the first OR gate discompare from causing a discompare output.

Note:

No further documentation is available. Specific questions, however, may be directed to:

Technology Utilization Officer
Code A&TS-TU
Marshall Space Flight Center
Huntsville, Alabama 35812
Reference: B71-10218

Patent status:

No patent action is contemplated by NASA.

Source: C. E. McCarthy of
IBM Federal Systems Division
under contract to
Marshall Space Flight Center
(MFS-21080)