

A DIGITAL COMPARATOR

by

LEONARD FARBER

B. S., Kansas State University, 1964

A MASTER'S REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1966

Approved by:

Charles A. Halijak
Major Professor

LD
2668
R4
1966
F 219
C.2

TABLE OF CONTENTS

INTRODUCTION	1
BIT-BY-BIT PROCESS	2
THE LARGER OF TWO BINARY NUMBERS	5
The Basic Operation	5
Realization of the Basic Operation	6
The Maximum of N Binary Numbers	8
Ordering a Set of N Binary Numbers	14
The Minimum of N Binary Numbers	15
APPLICATIONS	16
The Median	16
Rank Difference Method of Correlation	17
The Mode	17
Numbers in a Specified Interval	18
SUMMARY	18
ACKNOWLEDGMENT	20
REFERENCES	21

INTRODUCTION

The simplest digital computer system is the one associated with bit-by-bit approximation. The bit-by-bit machine (BBBM) consists of two elementary operations--addition and decision. Having a set, S , and a binary relation, R , for any two elements of S , a , and b , the decision that is made is whether a is related to b by R . This report will concentrate on developing such a decision device, a digital comparator. The set, S , will be the set of positive binary integers; and three binary relations will be considered--greater than, less than, and equal to. The decision to be made will therefore concern the relative size of two positive binary numbers. Also considered is the nonnumerical problem of ordering a set of numbers ascendingly or descendingly according as the binary relation greater than or less than is used, respectively.

THE BIT-BY-BIT PROCESS

The bit-by-bit process consists of successive binary approximations for evaluating exactly or approximately, within specified limits of accuracy, the value of a function when a positive finite value is assigned to the independent variable. Basic to this method is the construction of a test function; and for each approximation the test function is compared with zero by means of a comparator to determine the next approximation. The bit-by-bit process will be illustrated by calculation of $\log_2 x$.

Given a value of x such that $1 < x < 2$ and x is of finite length, consider finding f such that $2^f = x$. The given interval for x implies that $0 < f < 1$. Construct the following test function, t_n ,

$$t_n = x - 2^{(f_n + 2^{-(n+1)})}$$

where f_n is the n^{th} bit approximation to f , and define an auxiliary function, a_n , as

$$a_n = 2^{2 \cdot 2^n (f - f_n)}$$

In terms of the auxiliary function, the test function becomes

$$t_n = a_n - 2.$$

As a first approximation, choose

$$f_n = f_0 = 0$$

and obtain $a_0 = x^2$. If $t_n < 0$, then the approximation is too large, and as the next approximation, f_{n+1} , choose $f_{n+1} = f_n$.

From the definition of the auxiliary function, $a_{n+1} = (a_n)^2$ in this case. If $t_n \geq 0$, then the n^{th} bit approximation is too small, and as the next approximation select $f_{n+1} = f_n + 2^{-(n+1)}$. For this case, $a_{n+1} = (a_n/2)^2$.

The stop function for the process is an arbitrary given number. For example, the process may be terminated when $n = 10$, or when the difference of f_{n+1} and f_n is smaller than some specified value. The flow chart illustrating the process is shown in Fig. 1.

Suppose a value $X > 2$ is initially given. Now, an $F > 1$ is sought such that $2^F = X$. Divide X by 2 repeatedly until X is transformed into a value x such that $1 < x < 2$; that is,

$$X2^{-P} = x, \text{ where } 1 < x < 2.$$

Now f may be found such that $2^f = x$. The inverse of the above transformation is now applied to x to obtain F ; that is,

$$F = f + P.$$

Similarly, if a value $X < 1$ is given, transform X by multiplying by 2^P until an x is obtained such that $1 < x < 2$. Then apply the inverse transformation, 2^{-P} , to x and get F .

In the above bit-by-bit process and in all digital computers, an exceptionally important and very elementary operation is the comparison of two numbers. The device which accomplishes this task shall be singled out, and the remainder of this paper will be concerned with the investigation, construction, and application of such a digital comparator.

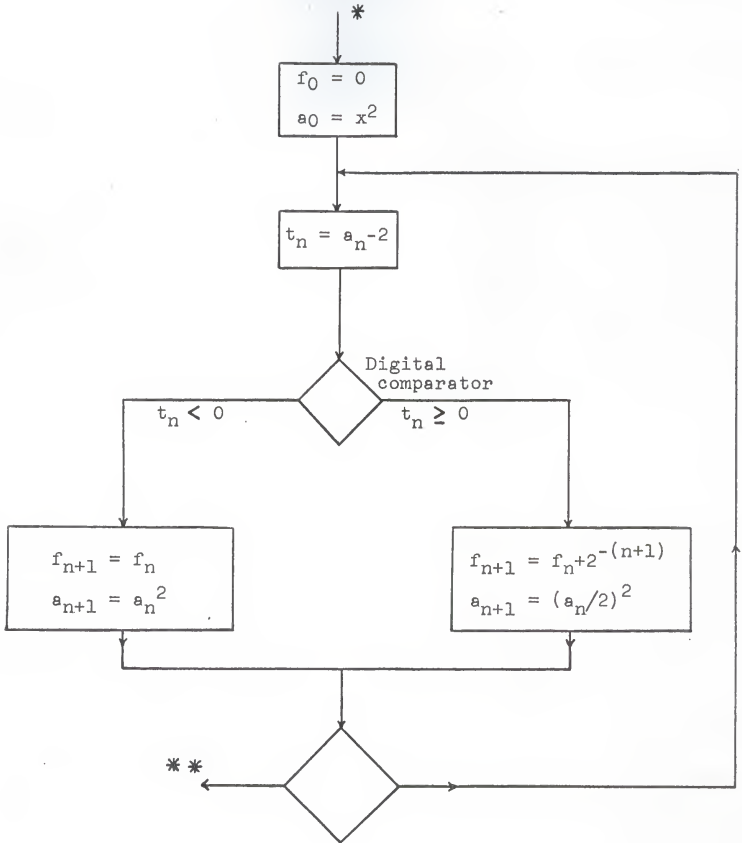


Fig. 1. A flow chart for the bit-by-bit approximation of $f = \log_2 x$.

THE LARGER OF TWO BINARY NUMBERS

The Basic Operation

Given two positive binary numbers, x and y , each b bits long, one may be subtracted from the other and the larger of the two may be found by checking the sign of the remainder. It will be shown that the subtraction operation may be replaced by the simpler addition operation.

Definition 1. The degree of a binary number having b bits is $b - 1$.

Definition 2. The ones complement of y is a number \bar{y} such that the ones of y are replaced by zeros and the zeros of y are replaced by ones.

Theorem. Given two positive binary numbers, x and y , each of degree n , and consider the basic operation $x + \bar{y}$. If $x > y$ then the $b + 1$ bit, the last carry or overflow bit, will be a 1. If $x \leq y$ then the last carry bit will be a 0, or equivalently there will be no overflow.

Proof. 1. Assume $x > y$ and consider $x - y$. For any positive binary number, y , of degree n ,

$$y + \bar{y} = 2^{n+1} - 1$$

or
$$-y = \bar{y} - 2^{n+1} + 1$$

Then, $x - y = x + \bar{y} - 2^{n+1} + 1 > 0$.

$$x + \bar{y} > 2^{n+1} - 1$$

$$x + \bar{y} \geq 2^{n+1}$$

Therefore there is always an overflow if $x > y$.

2. Assume $x < y$. Now,

$$\begin{aligned}x - y &= x + \bar{y} - 2^{n+1} + 1 < 0 \\x + \bar{y} &< 2^{n+1} - 1 \\x + \bar{y} &< 2^{n+1}\end{aligned}$$

3. Finally, if $x = y$,

$$\begin{aligned}x - y &= x + \bar{y} - 2^{n+1} + 1 = 0 \\x + \bar{y} &= 2^{n+1} - 1 \\x + \bar{y} &< 2^{n+1}\end{aligned}$$

Therefore there is no overflow if $x \leq y$. End of proof.

Therefore to find the larger of x and y , the sum of x and \bar{y} must be formed and the overflow bit determined.

Realization of the Basic Operation

Using parallel addition, if x and \bar{y} are b bits long, then b adders are required. The first adder is a half-adder since the inputs are x_0 and \bar{y}_0 with no previous carry bit. The truth table for the half-adder is given in Table 1. In Table 1, $s_0 = (x_0 + \bar{y}_0) \bmod 2$ and c_0 is the carry bit. Resulting Boolean expressions are:

$$\begin{aligned}s_0 &= \bar{x}_0\bar{y}_0 + x_0y_0 \\c_0 &= x_0\bar{y}_0\end{aligned}$$

The remaining $b - 1$ adders are full-adders since each will have the three inputs x_i , y_i , and c_{i-1} , where c_{i-1} is the previous

Table 1. Truth table for the half-adder.

x_0	\bar{y}_0	s_0	c_0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

carry bit. The full-adder truth table is given in Table 2.

Table 2. Truth table for the full-adder.

x_i	\bar{y}_i	c_{i-1}	s_i	c_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

The full-adder's Boolean expressions are:

$$s_i = \bar{x}_i y_i c_{i-1} + \bar{x}_i \bar{y}_i \bar{c}_{i-1} + x_i y_i \bar{c}_{i-1} + x_i \bar{y}_i c_{i-1}$$

$$c_i = \bar{y}_i c_{i-1} + x_i c_{i-1} + x_i \bar{y}_i$$

Physical realization of these Boolean expressions are shown in Fig. 2.

In the circuit of Fig. 2, it would be just as feasible to use direct subtraction in finding the larger of x and y . However, using the basic operation, the sum itself is of no importance. Only the overflow bit contains the information of interest. Therefore the previous circuit components may be reduced by constructing each adder modulo its summer, that is, constructing only the "carry bit generator" (CBG). The resulting circuit is shown in Fig. 3. With the overflow bit now available, it remains only to determine whether it is a 1 or a 0.

The above process has destroyed both x and y . The larger of the two is not available as an input if it should be desired to compare it with a third positive binary number, z . In the next section, reconstruction of the larger number will be undertaken.

The Maximum of N Binary Numbers

Given a set of positive binary numbers, $\{x_1, x_2, x_3, \dots, x_n\}$, the maximum of the set may be found by finding the larger of x_1 and x_2 , then the larger of this and x_3 , and continuing in this fashion until all the numbers in the set have been exhausted. It was seen earlier though that in finding the larger of x_1 and x_2 both were destroyed. Therefore the larger of the two must be reconstructed so that it is available as an input to be compared with x_3 , and likewise for each successive maximum.

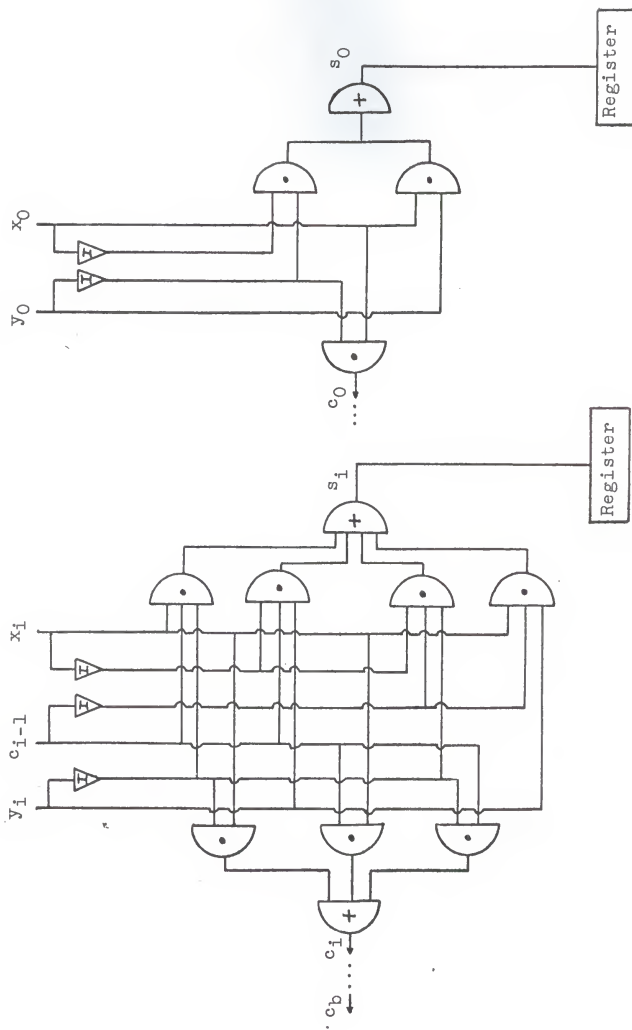


Fig. 2. A circuit to realize the basic operation.

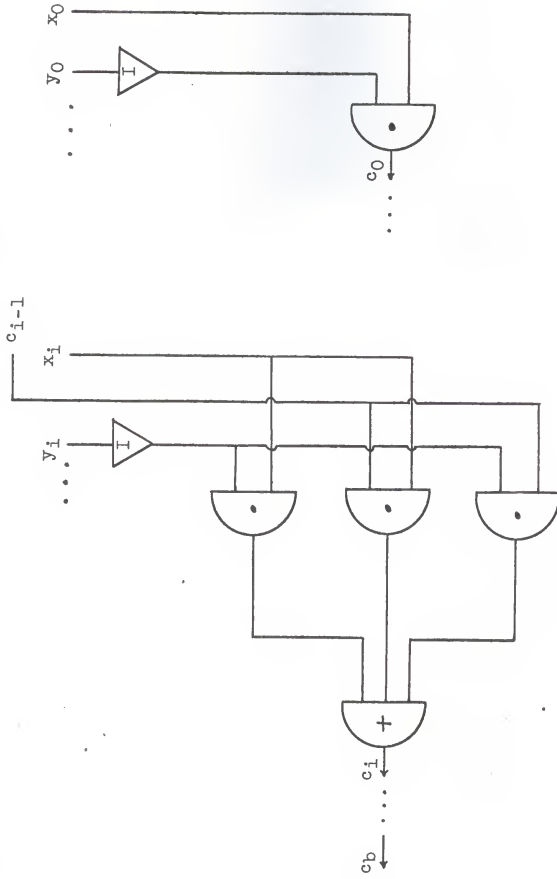


Fig. 3. The carry bit generator.

A circuit to accomplish this task is shown in Fig. 4. Represented is a single bit section.

Here x_{ij} and x_{ik} are the i^{th} bits of the j^{th} and k^{th} numbers in the set. As x_{ij} and x_{ik} are put into the i^{th} carry bit generator, each is also put into an RS flipflop containing a trigger input. A positive pulse will be used for a 1 signal, and a negative pulse for a 0 signal. The operation table of an RS flipflop is shown in Table 3. The present state is denoted

Table 3. Operation table of an RS flipflop.

	: r	: s	: q	: Q
1	0	0	0	0
2	0	0	1	1
3	0	1	0	1
4	0	1	1	1
5	1	0	0	0
6	1	0	1	0

by q and the next state by Q . Operation of the trigger terminal is such that if a negative pulse, a zero signal, is applied to it, the present state is complemented.

It is desired that the output state of the flipflop correspond to the set input s . Looking at conditions 3 and 5 of the operation table, it is seen that if the present state is 0 and if the reset input is the complement of the set input, then the next state or output will be the same as the set input.

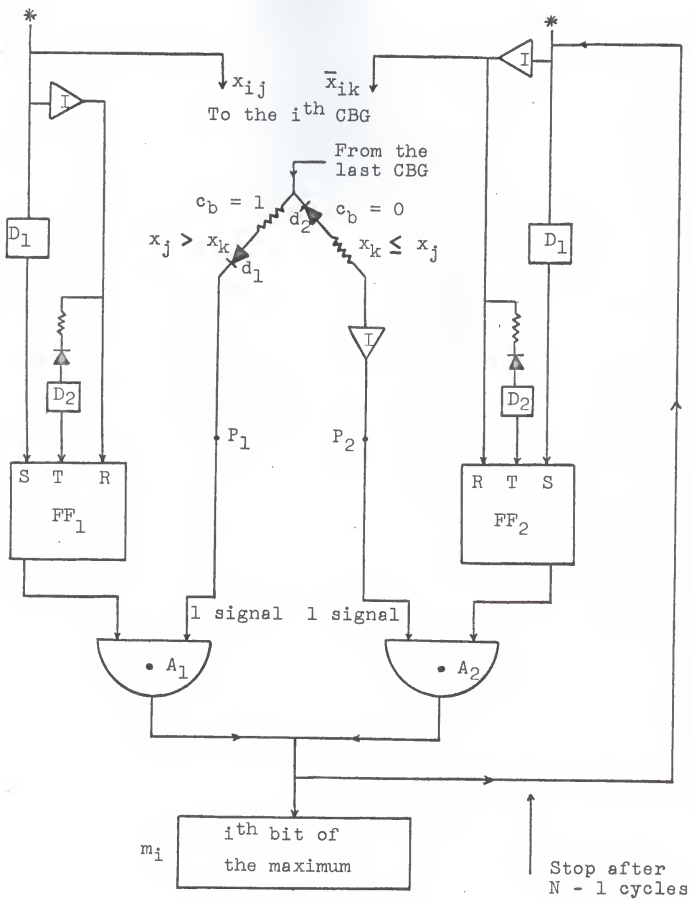


Fig. 4. A circuit to reconstruct the i th bit of the larger of x_j and x_k .

The input bit, x_{i_n} , will be applied to the set terminal and its complement to the reset terminal as shown in the circuit diagram. The delay, D_1 , is introduced to allow time for the inversion of x_{i_n} so that x_{i_n} and \bar{x}_{i_n} arrive at their respective terminals simultaneously.

The operation table of the flipflop shows that in condition 5 when x_{i_n} is a 0, the next state is a 0. However, this is the desired state at the time the input x_{i_n} is applied. Therefore no difficulty is encountered concerning the present state upon applying the input which follows x_{i_n} . In condition 3 where x_{i_n} is a 1, the next state is a 1. Thus in this case the state must be complemented before the next input is applied. If x_{i_n} is a 1, then the signal at the reset terminal is a 0 and if this is applied to the trigger terminal, the flipflop is put in the 0 state. If x_{i_n} is a 0, then the reset signal is not desired at the trigger input. Therefore the reset signal is first applied to a diode connected to admit a negative pulse and then applied to the trigger terminal. The delay, D_2 , is introduced so that the state is not changed before it is known whether x_{i_n} is to be reconstructed; that is, not before the signal from the last carry bit generator arrives.

To determine whether c_b is a 1 or a 0, c_b is brought to a junction of two diodes as shown in Fig. 4. The left diode, d_1 , admits a positive pulse and the right diode, d_2 , admits a negative pulse. If c_b is a 1, then d_2 does not conduct, and the 1 signal at d_1 along with the output of FF_1 , x_{ij} , are put into an AND gate, A_1 . The output of the AND gate, A_1 , x_{ij} is desired

since $c_b = 1$ implies $x_j > x_k$. Now x_{ij} is fed back and replaces x_{ik} in FF_2 for the next comparison. In FF_1 , x_{ij} is replaced by the i^{th} bit of the next number in the set, $x_{i\ell}$. In case c_b is a 0, diode d_1 does not conduct and the 0 signal at d_2 is inverted. The resulting 1 signal along with the output of FF_2 , x_{ik} , are put into an AND gate, A_2 . The output of this AND gate is x_{ik} which is now desired since $c_b = 0$ implies $x_k \geq x_j$. Here x_{ik} is fed back and used again in FF_2 for the next comparison. As before, x_{ij} in FF_1 is replaced by $x_{i\ell}$.

The STOP function in the feedback path may be set at $N - 1$ cycles for N numbers in the set. Then the i^{th} bit of the maximum is brought to a register cell, m_i . If the circuit of Fig. 4 is paralleled, one for each bit, the maximum of the set of numbers appears in register M .

Ordering a Set of N Binary Numbers

The maximum of the set of numbers has been found previously and has been placed in a register, M_1 . The process may now be repeated using the remaining subset of $N - 1$ numbers. The maximum of this subset is placed in a register, M_2 . Continuing in this fashion, and placing the last number in register M_N , the numbers in the set appear in registers M_1, M_2, \dots, M_N in descending order. In obtaining the first maximum, $N - 1$ comparisons were necessary. To find the second maximum, $N - 2$ comparisons were needed. Therefore the total number of comparisons, T , necessary to order the set of N numbers is given by

$$T = (N - 1) + (N - 2) + (N - 3) + \dots + 1.$$

Since this is just the sum of the first $N - 1$ positive integers,

$$T = \frac{N(N - 1)}{2}.$$

If the set is thus ordered and register M_N is considered, then the minimum of the set is available. The case in which the minimum is desired rather than the maximum will be discussed next.

The Minimum of N Binary Numbers

Given two positive binary numbers, x_j and x_k , and considering the basic operation, $x_j + \bar{x}_k$, it was shown that if the last carry bit, c_b , is a 1, then $x_j > x_k$; and if c_b is a 0, then $x_j \leq x_k$. In finding the larger of x_j and x_k , the circuit of Fig. 4 was used to reconstruct the i^{th} bit of x_j if c_b is a 1 or the i^{th} bit x_k if c_b is a 0. It is desired to use the circuit of Fig. 4 to reconstruct the i^{th} bit of the smaller of x_j and x_k . If c_b is a 1 or a 0, the i^{th} bit of x_k or x_j , respectively, is to be reconstructed.

In order to accomplish the above task, only a single addition to the circuit is necessary; that is, a reversing switch need be placed in the lines at points P_1, P_2 . With the switch in one direction, the circuit is unchanged from that shown in Fig. 4. If the switch is in the other direction, then diode, d_1 , is connected to AND gate, A_2 ; and diode, d_2 , is connected to AND gate, A_1 .

In this case, if c_b is a 1, the 1 signal at d_1 is brought to AND gate, A_2 , along with the output of FF_2 , x_{ik} . The output of A_2 , x_{ik} , is desired since $c_b = 1$ implies $x_k \leq x_j$. As before, x_{ik} is fed back and used again in FF_2 for the next comparison; and x_{ij} is replaced by the next input, $x_{i,l}$. If $c_b = 0$, the 0 signal at d_2 is inverted, but now the resulting 1 signal is brought to AND gate, A_1 , along with the output of FF_1 , x_{ij} . The output of A_1 is x_{ij} as desired since $c_b = 0$ implies $x_j \leq x_k$. Again, x_{ij} replaces x_{ik} in FF_2 . The minimum of the set of numbers will now appear in register M. The set may be ordered in the same manner discussed earlier, but now the numbers appear in registers M_1, M_2, \dots, M_N in ascending order. The same number of comparisons is needed in both cases.

APPLICATIONS

The Median

Given a set $S_a = \{x_1, x_2, \dots, x_N\}$ where the x_i are positive binary numbers and N is odd, the median of the set may be found by repeated use of the basic operation. The maximum of the set is found by the method described earlier. Similarly, the minimum element of S_a is found. Now, both the maximum and the minimum are removed from the set. The remaining subset, S_b , is $S_b = S_a - \{x_{a \max}, x_{a \min}\}$. The maximum and minimum of S_b are likewise found and removed. This leaves the subset $S_c = S_b - \{x_{b \max}, x_{b \min}\}$. By successively removing the

maximum and minimum from the remaining subsets, a subset S_M is ultimately obtained for which $x_M \max = x_M \min$. This value of x is the median of the original set, S . The subset S_M may contain more than one element, but in that case the elements will all be identical.

Rank Difference Method of Correlation

Given N corresponding pairs of measured items (x_i, y_i) , $i = 1, 2, \dots, N$. Let the corresponding rank numbers be (p_i, q_i) where $p_i = 1$ for the largest x_i , $p_i = 2$ for the next largest x_i , and so on. Similarly, $q_i = 1$ for the largest y_i , $q_i = 2$ for the next largest y_i , etc. The rank difference coefficient of correlation is given by

$$r = 1 - \frac{6 \sum (p_i - q_i)^2}{N(N^2 - 1)}$$

If the pairs (x_i, y_i) are considered as two sets of numbers, $X = \{x_i\}$ and $Y = \{y_i\}$, each set may be ordered descendingly by using a digital comparator. Having the sets ordered, it is now very easy to determine the corresponding rank numbers to be used in the expression for the coefficient of correlation.

The Mode

Given a set of N binary numbers, $\{x_1, x_2, \dots, x_N\}$, a weighting factor, $f_i \geq 0$, is associated with each x_i . The

weighting factor may be the frequency, reliability, or some other indicator. The mode of the set is defined as that value, x_i , with which the maximum weighting factor, f_i , is associated. Therefore the comparator may be used to find the maximum of the set, f_i , and then the corresponding x_i is the mode.

Numbers in a Specified Interval

Given a set of positive binary numbers, $\{x_1, x_2, \dots, x_N\}$, the x_i which are sought are those which are in the open interval between α and β . First, each number in the set is compared with α , and those which are greater than α are stored in a register, M. After the elements of the set have been exhausted, those $x_i > \alpha$ are successively removed from the register and compared with β . Now those x_i for which $\beta > x_i$ are returned to register, M. When these comparisons are completed, the x_i in the register are such that $\alpha < x_i < \beta$.

SUMMARY

This report primarily contains a development of a digital comparator which in conjunction with the often discussed adder results in the integrated bit-by-bit machine.

A basic operation was defined, and it was shown that the operation yields a signal which always indicates which of two positive binary numbers is the larger; a realization of the

operation is given. A physical realization is then developed for finding the maximum or minimum of a set of binary numbers and is subsequently used in ordering the set. Several applications for the digital comparator are briefly discussed.

ACKNOWLEDGMENT

The writer wishes to express his appreciation to his major professor, Dr. Charles A. Halijak, for his guidance and helpful assistance throughout the preparation of this report.

REFERENCES

1. Combet, M., H. Van Zonneveld, and L. Verbeek.
Computation of the base two logarithm of binary numbers.
IEEE Transactions on Electronic Computation, Vol. EC-14,
No. 6, December, 1965.
2. Morrison, D. R.
A method for computing certain inverse functions.
Mathematical Tables and Aids to Computation, Vol. 10,
No. 56, p. 204, 1956.
3. Mott, H., and C. C. Carroll.
A versatile comparator for encoding devices. IEEE Trans-
actions on Electronic Computers, Vol. EC-14, No. 3,
June, 1965.
4. Phister, Montgomery, Jr.
Logical design of digital computers. New York: John
Wiley and Sons, Inc., 1958.
5. Wensley, J. H.
A class of non-analytical iterative processes. British
Computer Journal, Vol. 1, No. 4, 1959.
6. Wilkes, N. V., D. J. Wheeler, and S. Gill.
The preparation of programs for an electronic digital
computer. Cambridge, Mass.: Addison-Wesley Press,
1951.

A DIGITAL COMPARATOR

by

LEONARD FARBER

B. S., Kansas State University, 1964

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1966

The simplest digital computer system is the one associated with bit-by-bit approximation. The bit-by-bit machine consists of the two elementary operations of addition and decision. Having a set, S , and a binary relation, R , for two elements of S , a and b , the decision that is made is whether a is related to b by R .

This report primarily contains a development of a decision device, a digital comparator, which in conjunction with the often discussed adder results in the integrated bit-by-bit machine. The set, S , is the set of positive binary integers; and three binary relations--greater than, less than, and equal to are considered. The decision that is made therefore concerns the relative size of two positive binary numbers.

A basic operation, $x + \bar{y}$, is defined; and it is shown that if the last carry bit, c_b , is a 1, then $x > y$, and if c_b is a 0 then $x \leq y$. A realization of the operation is given. A physical realization is developed for finding the maximum or minimum of a set of binary numbers and is subsequently used in ordering the set ascendingly or descendingly according as the binary relation greater than or less than is used, respectively. Several applications for the digital comparator are briefly discussed.