Digital Logic and Circuit
Paper Code: CS-102

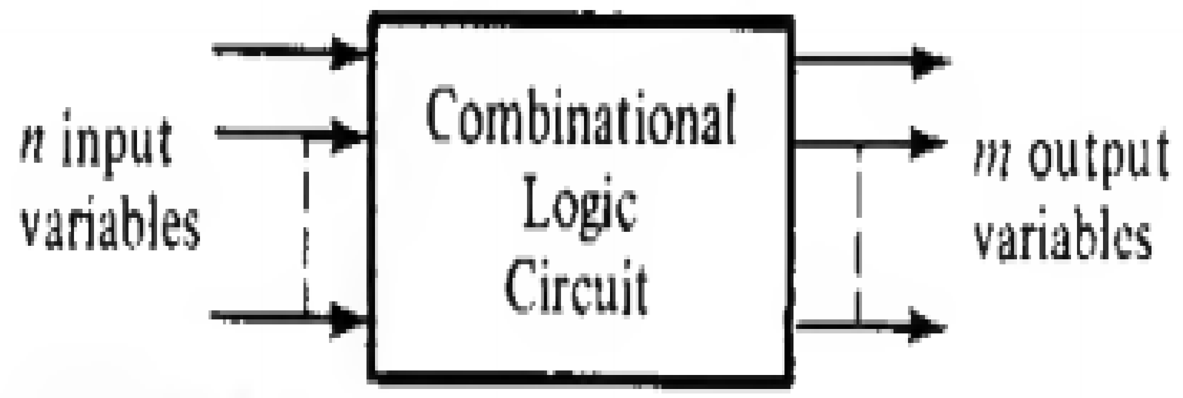
Outline

Combinational Circuit

- SUBTRACTORS
  - Half-Subtractor
  - Full-Subtractor
- Binary Parallel Adder
- Magnitude Comparator

Combinational Circuit

- For n input variables, there are 2^n possible combinations of binary input values.
- For each possible input combination, there is one and only one possible output combination.
- A combinational circuit can be described by m Boolean functions, one for each output variable. Each output function is expressed in terms of the n input variables.



Block diagram of a combinational circuit

Design Procedure of Combinational circuit

The design of combinational circuits starts from the verbal outline of the problem and ends in a logic circuit diagram or a set of Boolean functions from which the logic diagram can be easily obtained. The procedure involves the following steps:

1. The problem is stated.

2. The number of available input variables and required output variables is determined.

3 . The input and output variables are assigned letter symbols.

4 . The truth table that defines the required relationships between inputs and outputs is derived.

5 . The simplified Boolean function for each output is obtained.

6. The logic diagram is drawn.

Addition in any base

$(188)_{10}$
$+(295)_{10}$

483

$(10111)_2$
$+(10010)_2$

101001

$(524)_6$
$+(321)_6$

1245

$(5A7D)_{16}$
$+(F43B)_{16}$

14EB8

Add numbers
But if exceeds base subtract it,
Carry : if needs to subtract from base

Subtraction in any base

Borrow=base

$$\begin{array}{r} (278)_{10} \\ -\ (179)_{10} \\ \hline 099 \\ \hline \end{array}$$

$$\begin{array}{r} (100010)_{2} \\ -\ (010111)_{2} \\ \hline 001011 \\ \hline \end{array}$$

$$\begin{array}{r} (524)_{6} \\ -\ (355)_{6} \\ \hline 125 \\ \hline \end{array}$$

Subtractors

The subtraction of two binary numbers may be accomplished by taking the complement of the subtrahend and adding it to the minuend.

By this method, the subtraction operation becomes an addition operation requiring full- adders for its machine implementation.

Half-Subtractor

- A half-subtractor is a combinational circuit that subtracts two bits and produces their difference.
- It also has an output to specify if a 1 has been borrowed.
- The half-subtractor needs two outputs.
- One output generates the difference and will be designated by the symbol D. The second output, designated B for borrow, generates the binary signal that informs the next stage that a 1 has been borrowed.

Half-Subtractor (Cont..)

- Designate the minuend bit by x and the subtrahend bit by y.
- To perform x - y, we have to check the relative magnitudes of x and y.
- If x ≥y, we have three possibilities: 0-0 = 0, 1-0=1, and 1-1=0.

  The result is called the difference bit.
- If x < y, we have 0-1, and it is necessary to borrow a 1 from the next higher stage.
- The 1 borrowed from the next higher stage adds 2 to the minuend bit, just as in the decimal system a borrow adds 10 to a minuend digit.
- With the minuend equal to 2, the difference becomes 2-1 = 1.

The truth table for the input-output relationships of a half-subtractor can now be derived as follows:

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

The output borrow B is a 0 as long as $x \geq y$.

The Boolean functions for the two outputs of the half-subtractor are derived directly from the truth table:

$$D = x'y + xy'$$

$$B = x'y$$

Full-Subtractor

- A full-subtractor is a combinational circuit that performs a subtraction between two bits, taking into account that a 1 may have been borrowed by a lower significant stage.
- This circuit has three inputs and two outputs. The three inputs, x, y, and z, denote the minuend, subtrahend, and previous borrow, respectively.
- The two outputs, D and B, represent the difference and output borrow, respectively

The truth table for the circuit is

The simplified Boolean functions for the two outputs of the full-subtractor are derived in the maps of The simplified sum of products output functions are

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D = x'y'z + x'yz' + xy'z' + xyz$$

$$B = x'y + x'z + yz$$

Binary Parallel Adder

A binary parallel adder is a digital circuit that produces the arithmetic sum of two binary numbers in parallel.

It consists of full-adders connected in a chain, with the output carry from each full-adder connected to the input carry of the next full-adder in the chain.

The augend bits of A and the addend bits of B are designated by subscript numbers from right to left, with subscript 1 denoting the low-order bit.

The carries are connected in a chain through the full-adders.

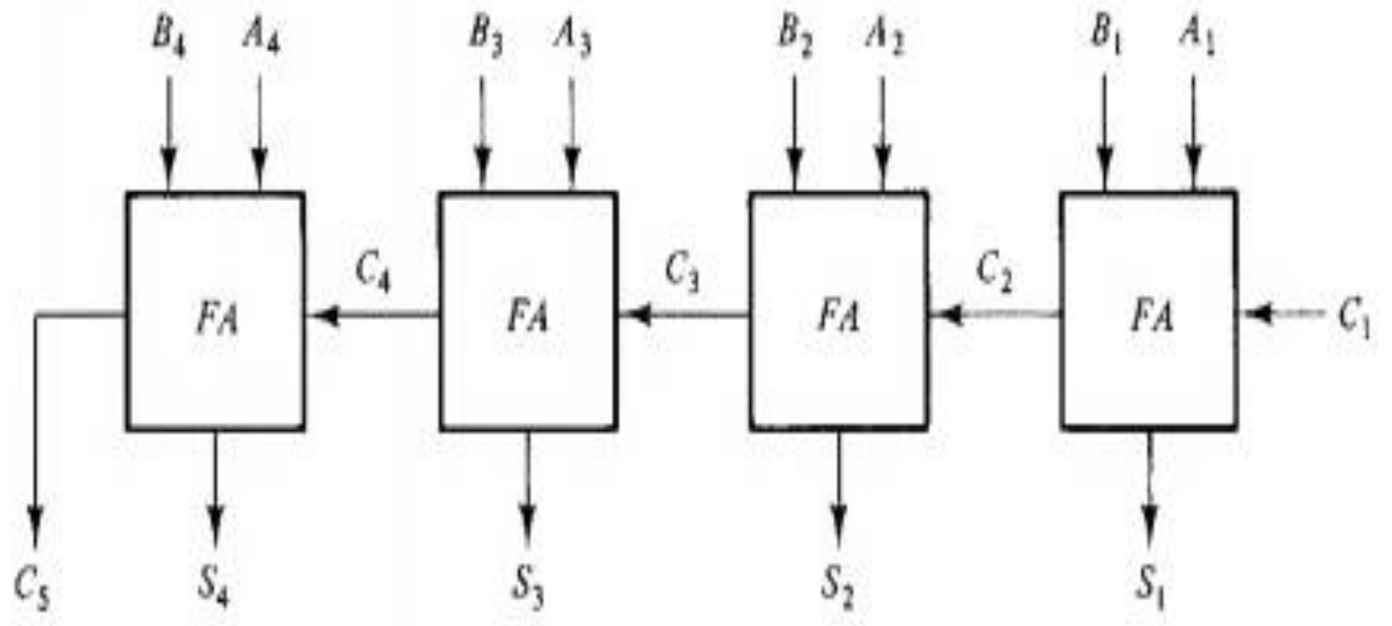
The input carry to the adder is Ci and the output carry is Cs. The 5 outputs ($C_sS_4S_3S_2S_1$) generate the required sum bits.

A=1001

\+ B=1100

Truth Table: Full Adder

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Binary Parallel Adder

Magnitude Comparator

The comparison of two numbers is an operation that determines if one number is greater than, less than, or equal to the other number.

A magnitude comparator is a combinational circuit that compares two numbers, A and B , and determines their relative magnitudes.

The outcome of the comparison is specified by three binary variables that indicate whether

$A > B$, $A = B$, or $A < B$.

The algorithm is a direct application of the procedure a person uses to compare the relative magnitudes of two numbers.

Consider two numbers, Ai and Bi, with four digits each. Write the coefficients of the numbers with descending significance as follows:

$$A = A_3 A_2 A_1 A_0,$$

$$B = B_3 B_2 B_1 B_0$$

The two numbers are equal if all pairs of significant digits are equal, i.e., if $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = B_1$ and $A_0 = B_0$.

When the numbers are binary, the digits are either 1 or 0 and the equality relation of each pair of bits can be expressed logically with an equivalence function:

$$X_i = A_iB_i + A'_i B_i' \qquad i = 0, 1, 2, 3$$

where $X_i = 1$ only if the pair of bits in position i are equal, i.e., if both are l's or both are 0's.

For the equality condition to exist, all x_i variables must be equal to 1. This dictates an AND operation of all variables

$$(A = B) = X_3X_2X_1X_0$$

The binary variable (A = B) is equal to 1 only if all pairs of digits of the two numbers are equal.

To determine if A is greater than or less than B, we inspect the relative magnitudes of pairs of significant digits starting from the most significant position.

If the two digits are equal, we compare the next lower significant pair of digits.

This comparison continues until a pair of unequal digits is reached. If the corresponding digit of A is 1 and that of B is 0, we conclude that $A > B$.

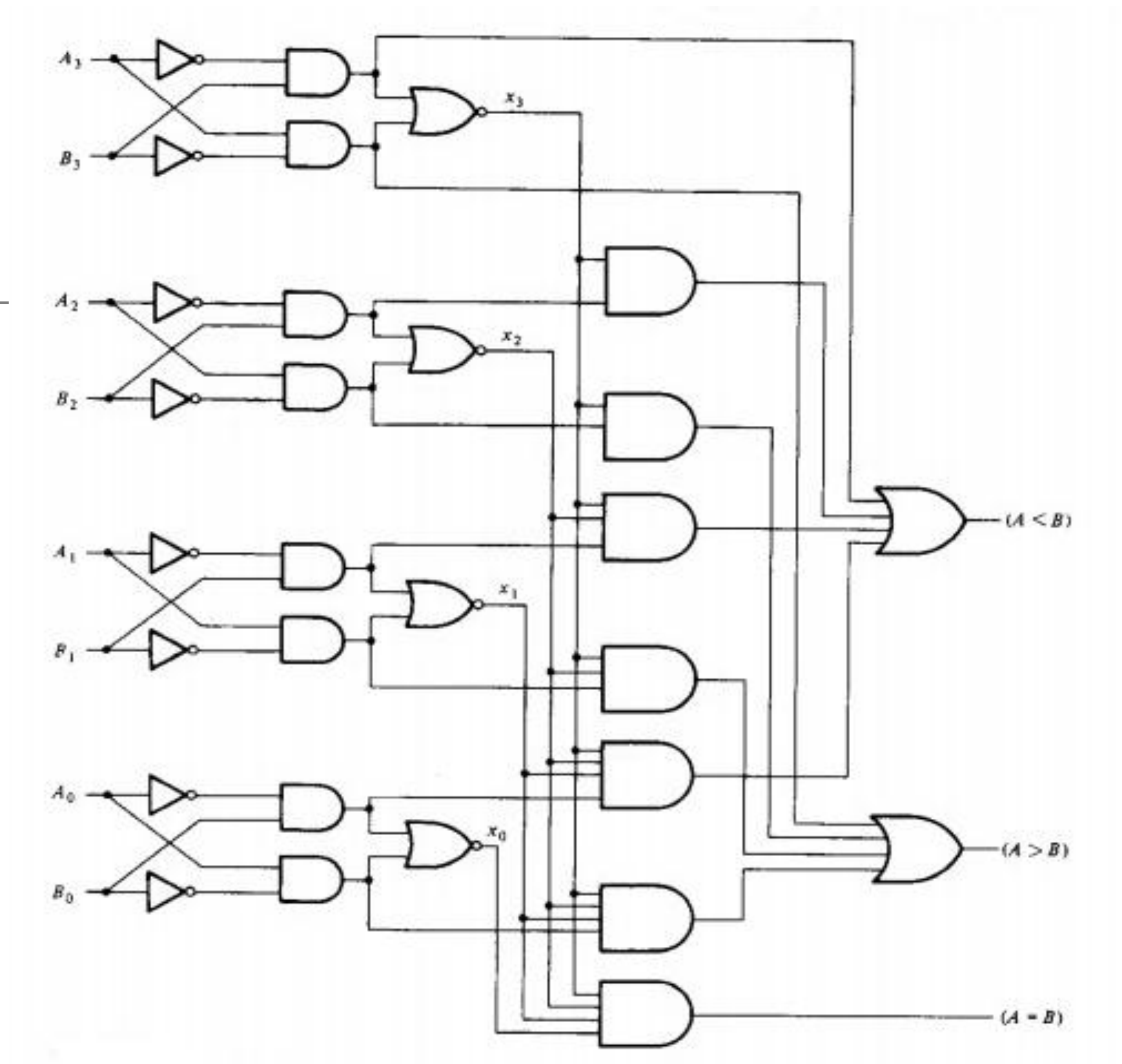
If the corresponding digit of A is 0 and that of B is 1, we have that $A < B$.

The sequential comparison can be expressed logically by the following two Boolean functions:

$$(A > B) = A_3 B_3' + X_3 A_2 B_2' + X_3 X_2 A_1 B_1' + X_3 X_2 X_1 A_0 B_0'$$

$$(A < B) = A_3' B_3 + X_3 A_2' B_2 + X_3 X_2 A_1' B_1 + X_3 X_2 X_1 A_0' B_0$$

The symbols $(A > B)$ and $(A < B)$ are binary output variables that are equal to 1 when $A > B$ or $A < B$, respectively.



4-bit magnitude comparator

Suggested Reading

- M. Morris Mano, Digital Logic and Computer Design, PHI.

Thank you