



Combinational Logic

Rab Nawaz Khan Jadoon

Department of Computer Science

DCS

COMSATS Institute of
Information Technology

Lecturer

COMSATS Lahore

Pakistan

■ Design Procedures

- Starts from the verbal outline of the problem and ends in a logic circuit diagram.
- The procedure involves the following step,
 - The problem is stated.
 - Input and required output variables are determined.
 - Assigned the variables letter symbols.
 - Make the truth table.
 - The simplified Boolean functions for each output is obtained.
 - The logic diagram is drawn.

Carry Propagation

- Addition of two binary numbers in parallel implies that all the bits of augend and addend are available for computation at the same time.
 - Total propagation time is equal to the propagation delay of a typical gate times the number of gate levels in the circuit.
 - The longest propagation delay time in a parallel adder is the time it takes the carry to propagate through the full adders.
 - Each bit of the sum output depends on the value of the input carry.



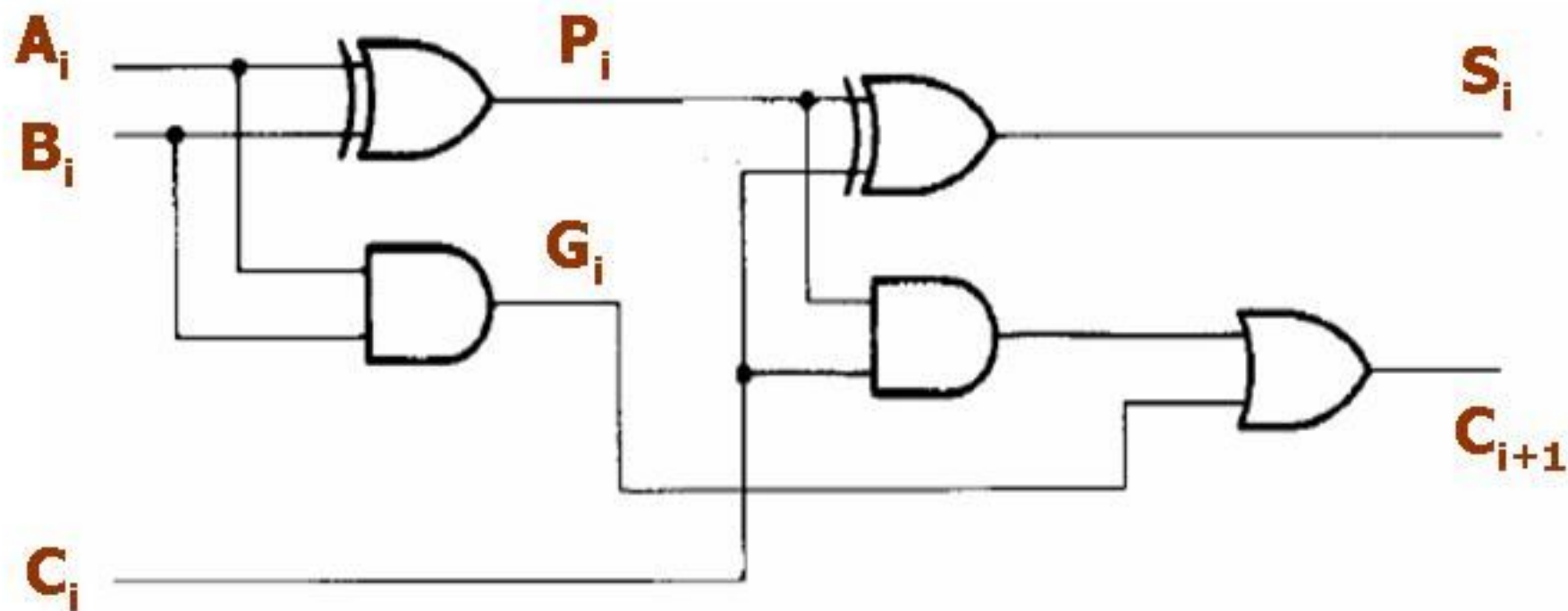
Carry propagation

- The value of S_i in any given stage in the adder will be in its steady state final value only after the input carry to that stage has been propagated.
 - Inputs A_4 and B_4 reach a steady value as soon as input signals are applied to the adder.
 - But input carry C_4 does not settle to its final steady state value until C_3 is available in its steady state value.
 - C_3 has to wait for C_2 and so on down to C_1 .



Carry propagation

- The number of gate levels for the carry propagation can be found from the circuit of the full adder.



Carry propagation

- The inputs and output variables use the subscript i to denote a typical stage in the parallel adder.
 - The signals at P_i and G_i settle to their steady state values after the propagation through their respective gates.
 - These two signals are common to all full adders and depend only on the input augend and addend bits.
 - The signal from the input carry C_i to the output carry C_{i+1} , propagates through an AND gate and an OR Gate, which constitute two gate level.



Carry propagation

- If there are four full adders in the parallel adder, the output carry C_5 would have $2 \times 4 = 8$ gate levels from C_1 to C_5 .
 - The total propagation time in the adder would be the propagation time in one half adder plus eight gate level.
 - For an n bit parallel adder there are $2n$ gate level for the carry to propagate through.
 - The carry propagation is a limiting factor on the speed with which two numbers are added in parallel.



Carry propagation

- The solution is to employ faster gates with reduced delays. But this is a limitation.
 - There are several techniques for reducing the carry propagation time in a parallel adder.
 - The most widely used technique is the **principle of look-ahead carry**.



Carry propagation

- Consider the circuit of the full adder, if we define two new binary variables,

$$P_i = A_i \oplus B_i$$

$$G_i = A_i B_i$$

- The output sum and carry can be expressed as,

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

Carry propagation

- G_i is called a carry generate and it produces an output carry when both A_i and B_i are one regardless of the input carry.
 - P_i is called carry propagate because it is the term associated with the propagation of the carry from C_i to C_{i+1} .



Carry propagation

- The Boolean function for the carry output of each stage and substitute for each C_i , its value from the previous equation.

$$C_2 = G_1 + P_1 C_1$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2(G_1 + P_1 C_1) = G_2 + P_2 G_1 + P_2 P_1 C_1$$

$$C_4 = G_3 + P_3 C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_1$$

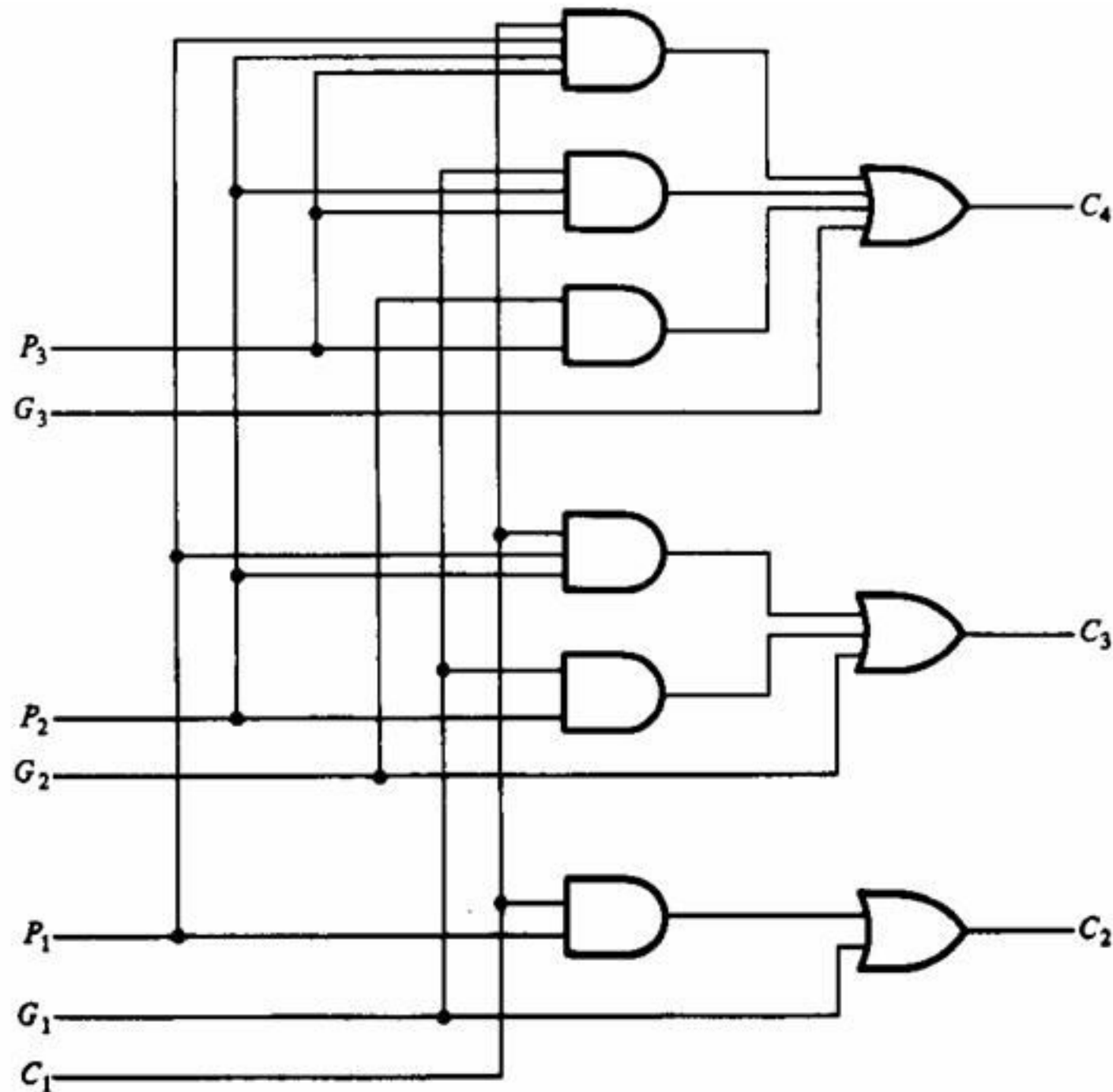


Carry propagation

- **Each output carry is expressed in SOP.**
 - So each function can be implemented with one level of AND Gates followed by an OR gate.
 - Three Boolean function for C_2 , C_3 and C_4 are implemented in the look ahead carry generator.
 - Note that C_4 does not have to wait for C_3 and C_2 to propagate; C_4 is propagated at the same time as C_2 and C_3 .



Carry propagation



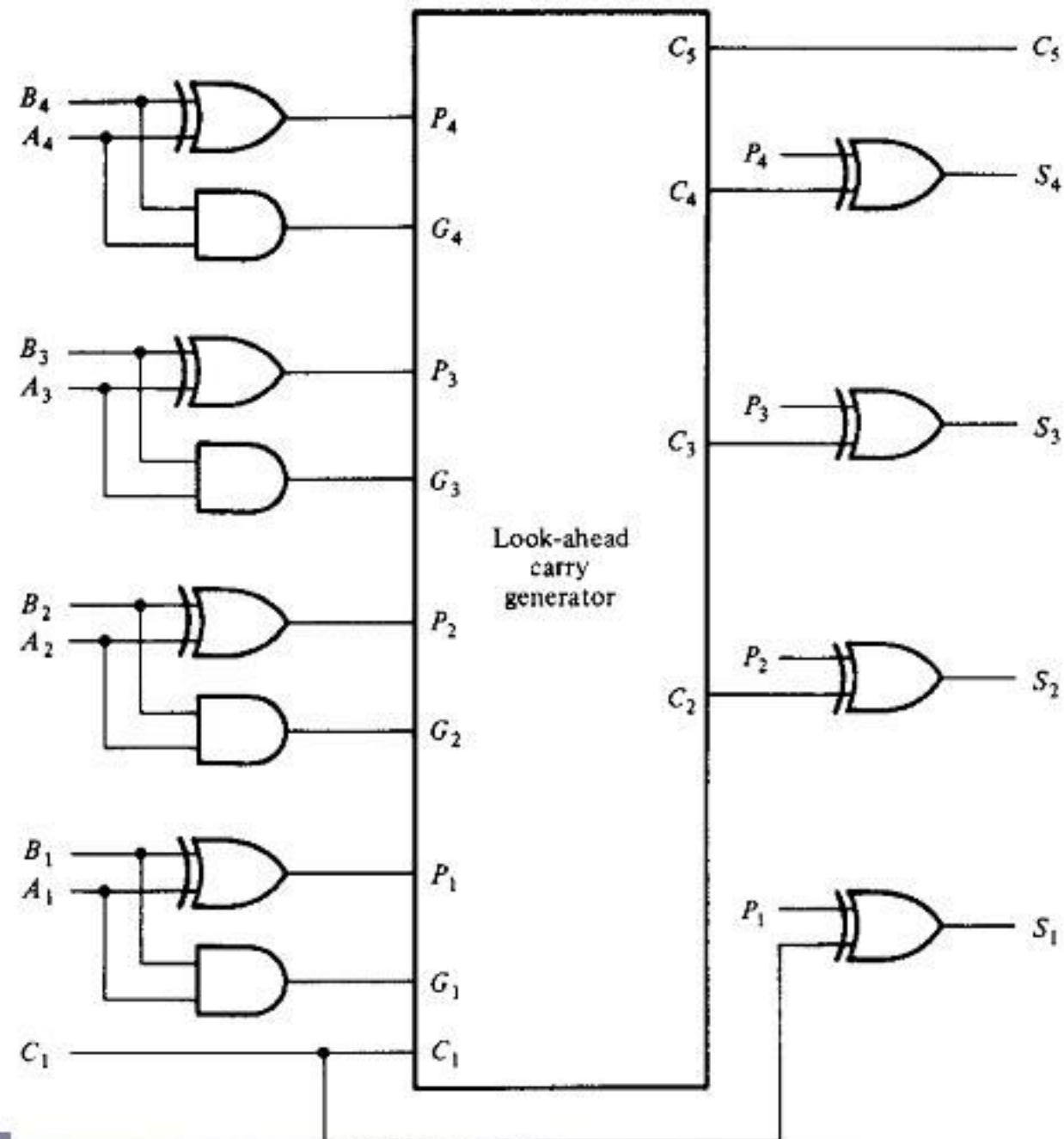
Carry propagation

- **The construction of a a 4 bit parallel adder with a look-ahead carry,**
 - Each sum output requires two exclusive OR Gate.
 - The output of the first gate generate the P_i variable.
 - And the AND gate generates the G_i .
 - All the P 's and G 's are generated in two gate levels.
 - The carries are propagated through a look-ahead carry generator and applied as inputs to the second exclusive OR gate.
 - After the P and G signals settle into their steady state value, all output carries are generated after a delay of the two level of gates.



Carry propagation

- Thus output S_2 through S_4 have equal propagation delay times.



Magnitude Comparator

- It is a combinational circuit that compares two numbers A and B and determines their relative magnitude.
 - The output is specified by three binary variables that indicate whether $A > B$, $A = B$, or $A < B$.
- Procedure
 - Consider the two number A and B with 4 digit each.
 - Write the coefficient of the numbers with descending significance as follows,

$$A = A_3 A_2 A_1 A_0$$

$$B = B_3 B_2 B_1 B_0$$



Magnitude Comparator

- Two numbers are equal if all pairs of significant digits are equal i.e.
 - $A_3=B_3$ $A_2=B_2$ $A_1=B_1$ $A_0=B_0$
- 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 equivalence function.

$$x_i = A_i B_i + A_i' B_i' \quad i = 0, 1, 2, 3$$

Magnitude Comparator

- For the equality condition, all X_i variables must be equal to 1.
 - this shows an AND operation of all variables.
 - $(A=B) = X_3 X_2 X_1 X_0$
 - To determine if A is greater or less than B, we inspect the relative magnitudes of pairs of significant digits starting from the MSB position.
 - If the two digits are equal we compare the next lower significant pair of digits.
 - The comparison continues until a pair of unequal digits is reached.
 - If corresponding digits of A is 1 and that of B is 0, then we conclude that $A > B$ else $A < B$.



Magnitude Comparator

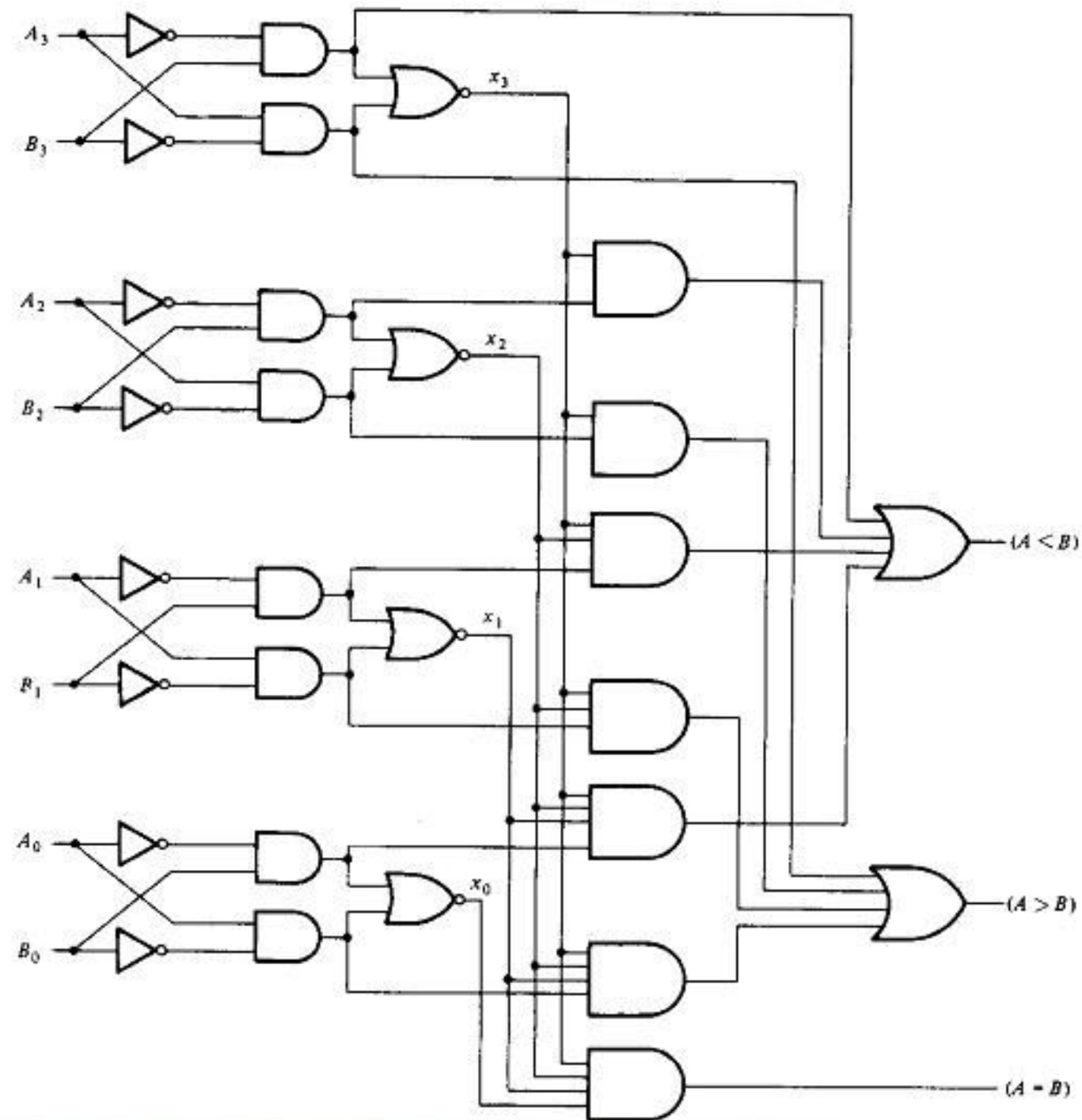
- The sequential comparison can be expressed logically by the following two Boolean functions.

$$(A > B) = A_3 B_3' + x_3 A_2 B_2' + x_3 x_2 A_1 B_1' + x_3 x_2 x_1 A_0 B_0'$$

$$(A < B) = A_3' B_3 + x_3 A_2' B_2 + x_3 x_2 A_1' B_1 + x_3 x_2 x_1 A_0' B_0$$



Magnitude Comparator



Magnitude Comparator

■ Problem

- Compare the following set of numbers through logic diagram,

- 1111 1001 (A ? B)
- 1000 1001 (A ? B)
- 1010 1100 (A ? B)
- 1100 1101 (A ? B)



