Digital Logic Design :

Lecture 11

The Look-Ahead Carry Adder

The propagation delay of ripple carry adder can be eliminated by look-ahead carry addition. The look-ahead carry adder produces output carry by either carry generation or carry propagation.

A carry is generated when both input bits or 1's. The generated carry C_g, is expressed as,

$$C_g = AB$$

An input carry may be propagated when any one of the input bits is 1. The propagated carry C_p, is expressed as

$$C_p = A \oplus B$$

$$\therefore C_{out} = C_g + C_p C_{in} \quad \left[C_{out} = AB + (A \oplus B) C_{in} \right]$$

Expression of the output carry, C_{out}, of each full-adder stage for the 4-bit example.

full adder 1

$$C_{out_1} = C_{g_1} + C_{p_1} C_{in_1}$$

Full-adder 2 :

$$C_{in2} = C_{out1}$$

$$\begin{aligned} C_{out2} &= C_{g2} + C_{P2} C_{in2} \\ &= C_{g2} + C_{P2} C_{out1} \\ &= C_{g2} + C_{P2} (C_{g1} + C_{P1} C_{in1}) \\ &= C_{g2} + C_{P2} C_{g1} + C_{P2} C_{P1} C_{in1} \end{aligned}$$

Full-adder 3 :

$$C_{in3} = C_{out2}$$

$$\begin{aligned} C_{out3} &= C_{g3} + C_{P3} C_{in3} \\ &= C_{g3} + C_{P3} C_{out2} \\ &= C_{g3} + C_{P3} (C_{g2} + C_{P2} C_{g1} + C_{P2} C_{P1} C_{in1}) \\ &= C_{g3} + C_{P3} C_{g2} + C_{P3} C_{P2} C_{g1} + C_{P3} C_{P2} C_{P1} C_{in1} \end{aligned}$$

Full-adder 4 :

$$C_{in4} = C_{out3}$$

$$\begin{aligned} C_{out4} &= C_{g4} + C_{P4} C_{in4} \\ &= C_{g4} + C_{P4} C_{out3} \\ &= C_{g4} + C_{P4} (C_{g3} + C_{P3} C_{g2} + C_{P3} C_{P2} C_{g1} + C_{P3} C_{P2} C_{P1} C_{in1}) \\ &= C_{g4} + C_{P4} C_{g3} + C_{P4} C_{P3} C_{g2} + C_{P4} C_{P3} C_{P2} C_{g1} + C_{P4} C_{P3} C_{P2} C_{P1} C_{in1} \end{aligned}$$

As the output carry for each full-adder stage is dependent only on the initial input carry C_{in1}, the C_g and C_P functions of that stage and the C_g and C_P functions of the preceding

stages, so a carry need not to propagate through all the stages to achieve a final result. Thus, the look-ahead carry adder speeds up the addition process.

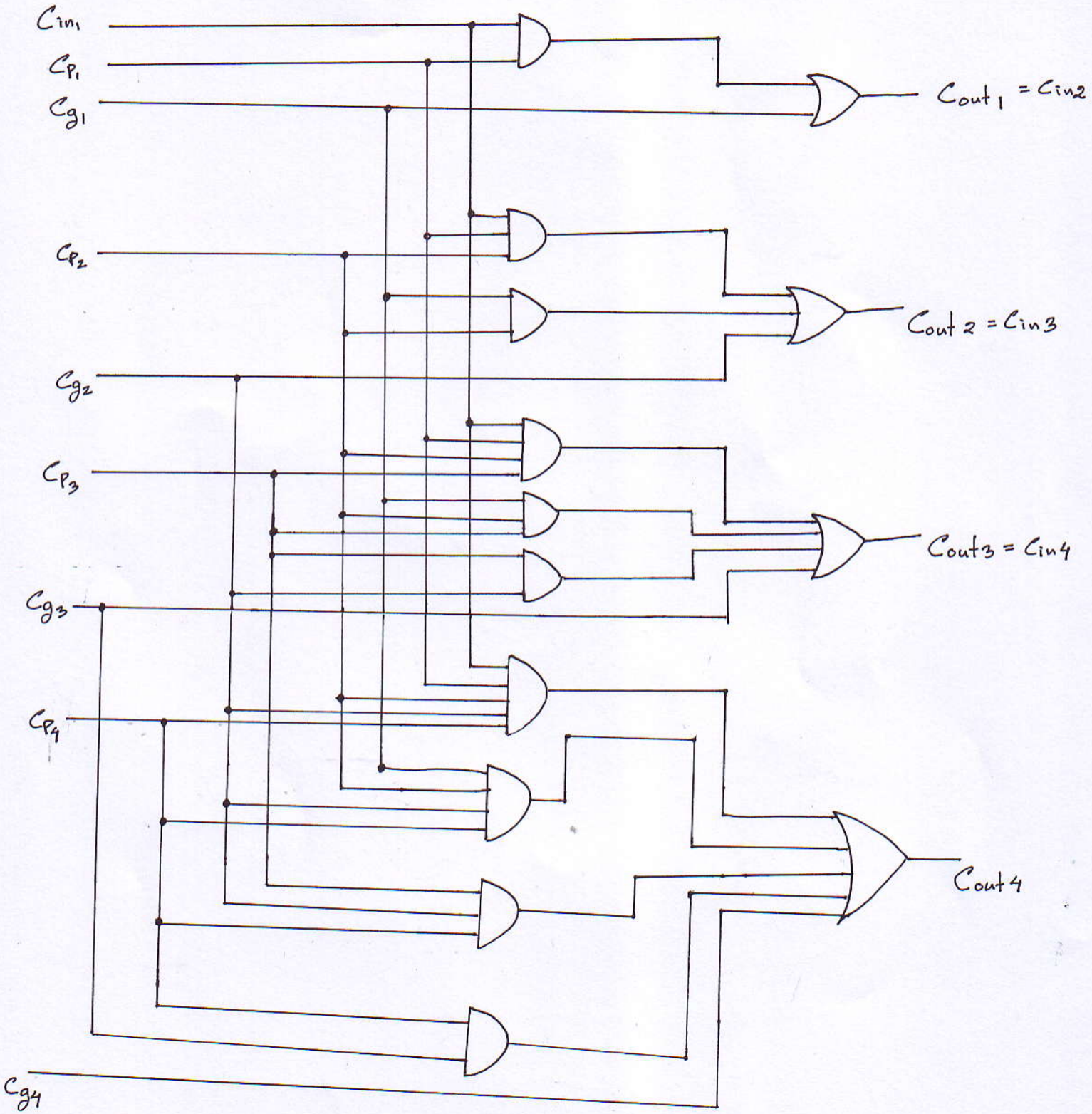


Fig : Logic diagram of a look-ahead carry generator

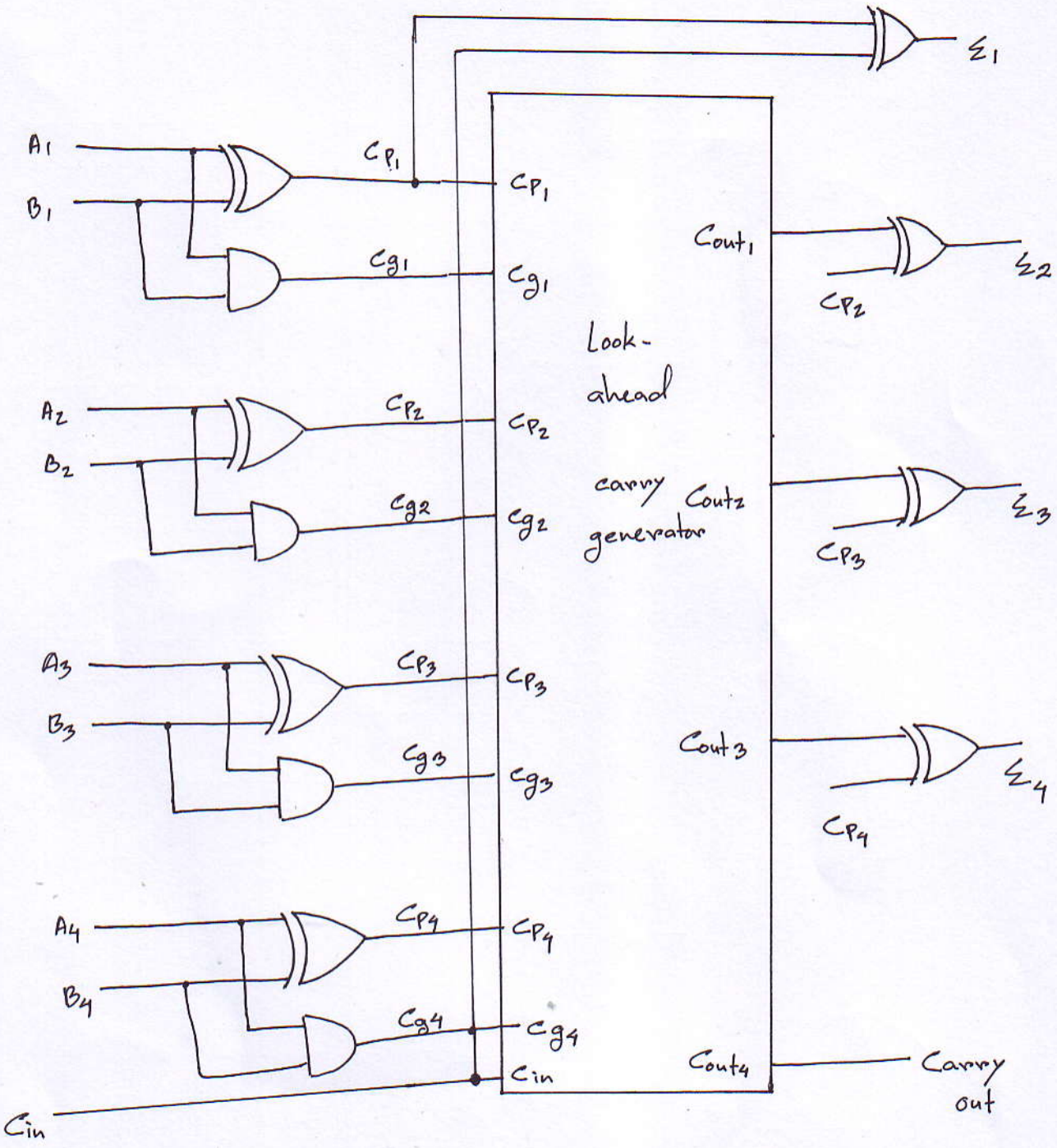


Fig : 4-bit full-adder with look-ahead carry

Comparators :

A magnitude comparator is a combinational circuit that compares two numbers, A and B, determines their relative magnitude whether $A>B$, $A=B$ or $A<B$.

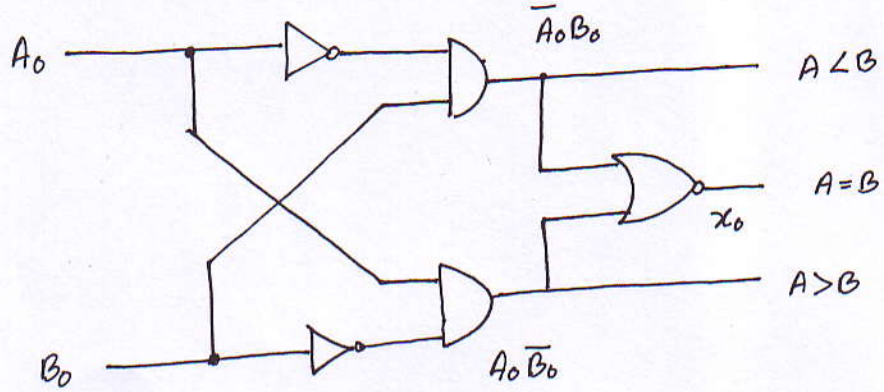


Fig : 1-bit magnitude comparator

$$x_0 = \overline{(\bar{A}_0B_0 + A_0\bar{B}_0)} = A_0B_0 + \bar{A}_0\bar{B}_0$$

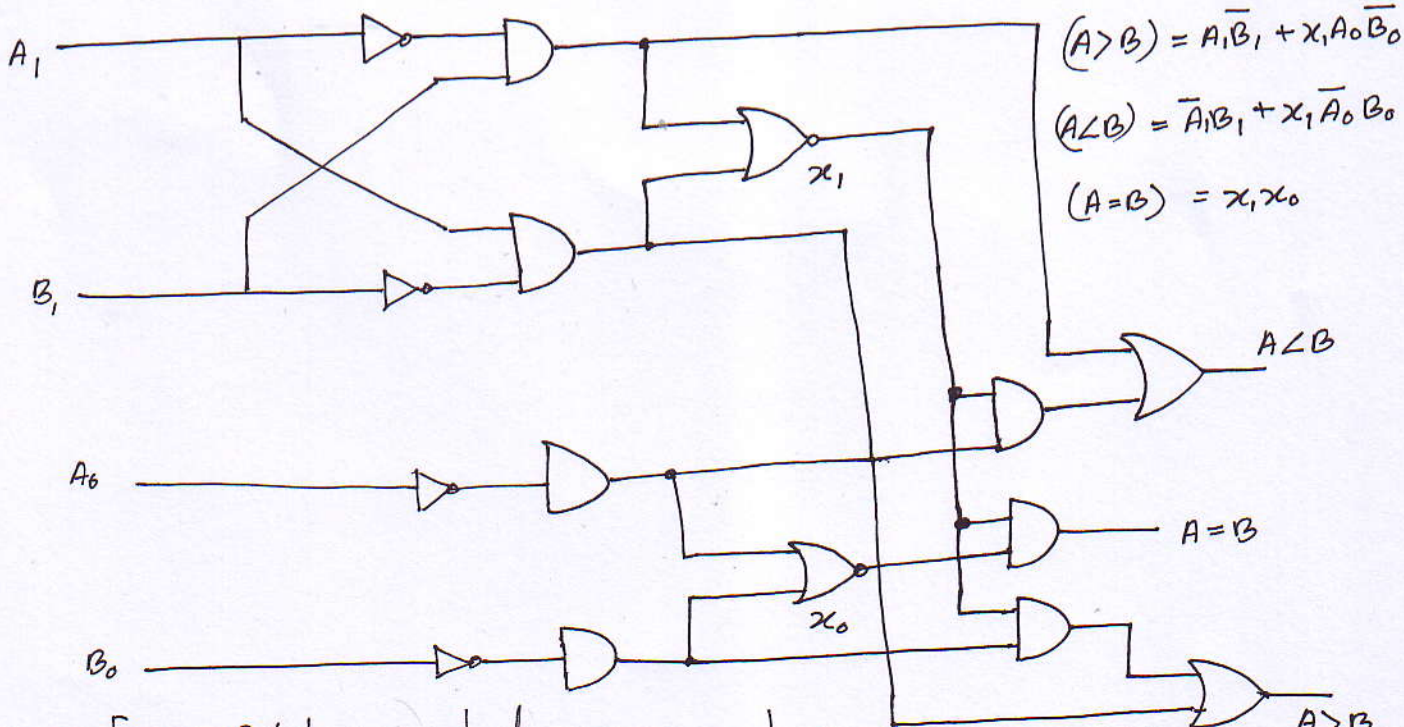
$$(A<B) = \bar{A}_0B_0 \;,\; (A>B) = A_0\bar{B}_0 \;,\; (A=B) = x_0$$

$$x_1 = A_1B_1 + \bar{A}_1\bar{B}_1$$

$$(A>B) = A_1\bar{B}_1 + x_1A_0\bar{B}_0$$

$$(A<B) = \bar{A}_1B_1 + x_1\bar{A}_0B_0$$

$$(A=B) = x_1x_0$$



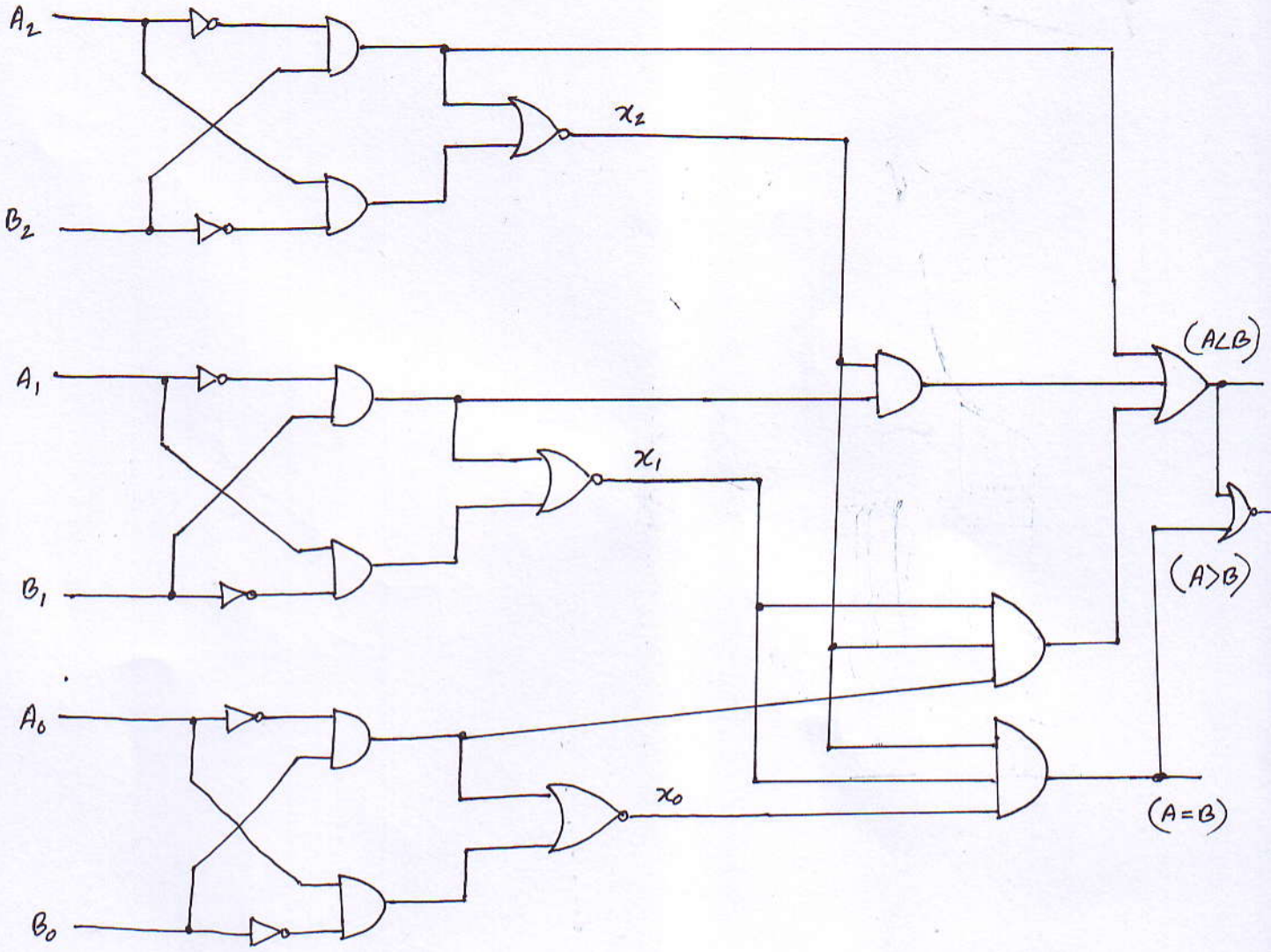


Fig : 3-bit magnitude comparator

$(A=B) = x_2 x_1 x_0$

$(A<B) = \overline{A_2} B_2 + x_2 \overline{A_1} B_1 + x_2 x_1 \overline{A_0} B_0$

$(A>B) = A_2 \overline{B_2} + x_2 A_1 \overline{B_1} + x_2 x_1 A_0 \overline{B_0}$