Spring 2003 ENGR/ECE 212 Homework 5

Problem 1

In this problem you are going to implement a PLA that receives four bits, a_1, a_0, b_1, and b_0, as input and produces the four-bit result of multiplying a_1a_0 by b_1b_0. For example, if the four input bits are 1011, that's 10 and 11, and four output bits should be 0110.

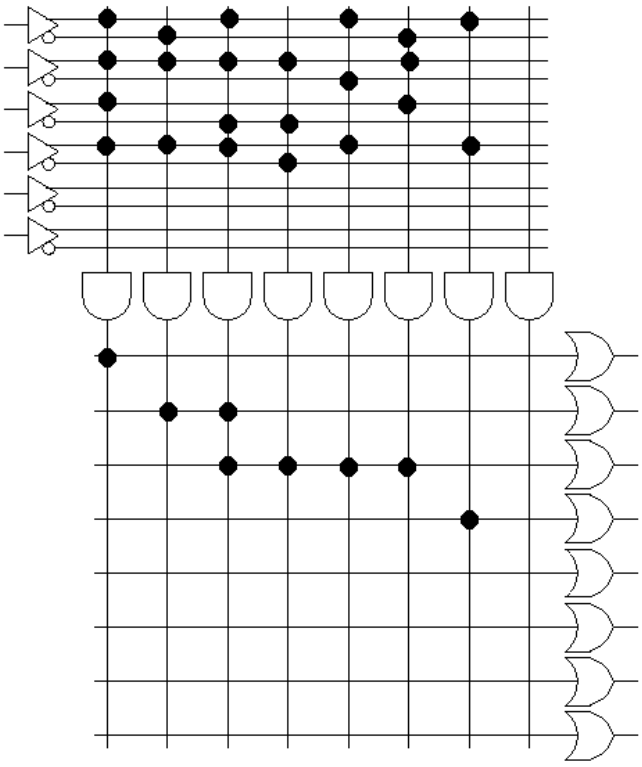
Write the Boolean expressions you will use to implement this function. You will probably need to use espresso to get your implementation to fit into the PLA.

Solution for problem 1

If you optimize the problem using espresso, you arrive at the following equations:

$p_3 = a_1\ a_0\ b_1\ b_0$
$p_2 = a_1\ b_1\ b_0' + a_1\ a_0'\ b_1\ b_0$
$p_1 = a_1'\ a_0\ b_1 + a_1\ b_1'\ b_0 + a_1\ a_0'\ b_1\ b_0 + a_0\ b_1\ b_0'$
$p_0 = a_0\ b_0$

Since there are seven unique product terms ($a_1\ a_0'\ b_1\ b_0$ is shared), the circuit can be implemented in a circuit with eight product terms.

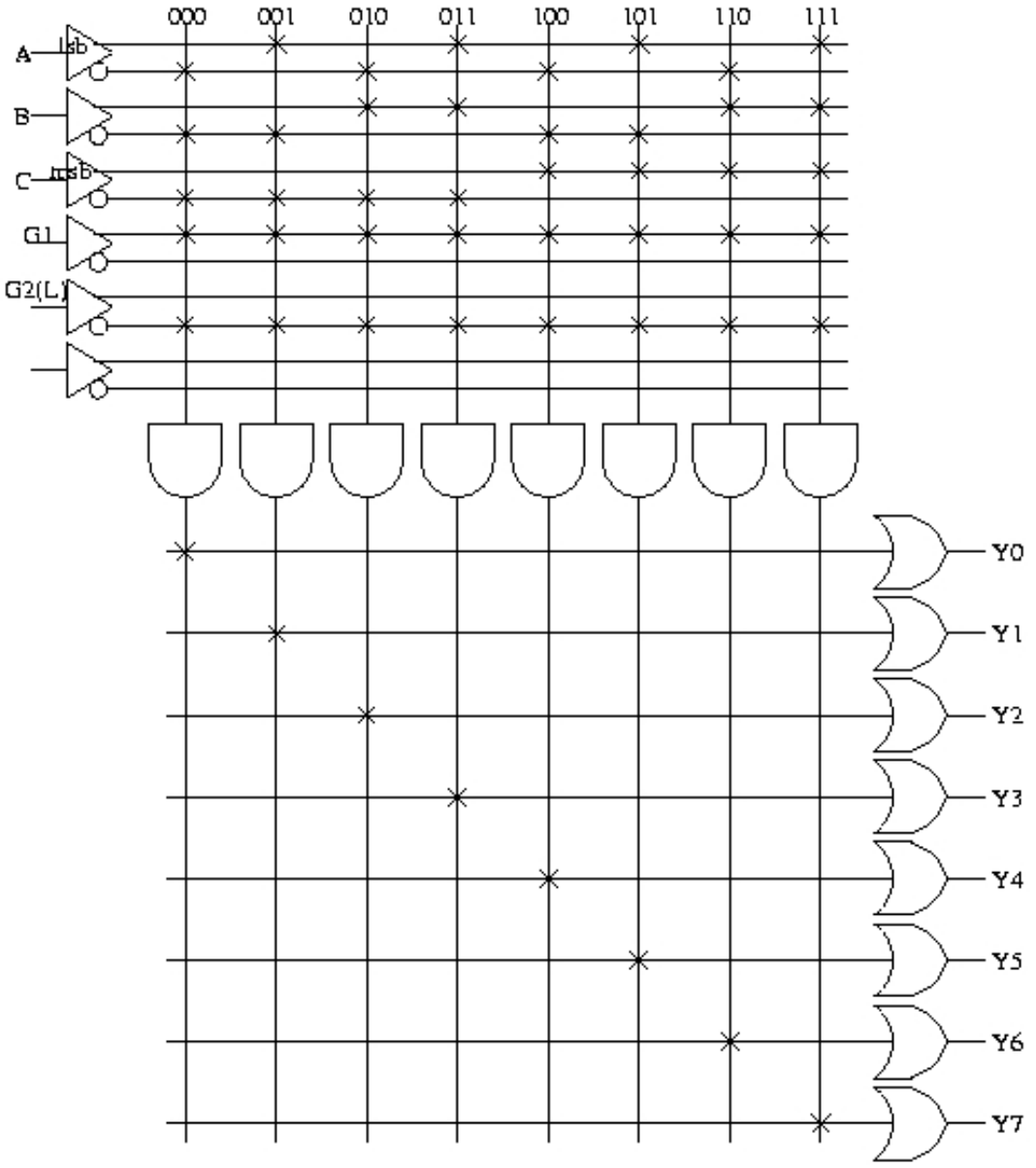


6 x 8 PLA with 8 product terms

Problem 2

Program the PLA below to construct a 3-to-8 Decoder. Include one active-low enable signal and one active-high enable signal.

Solution for problem 2

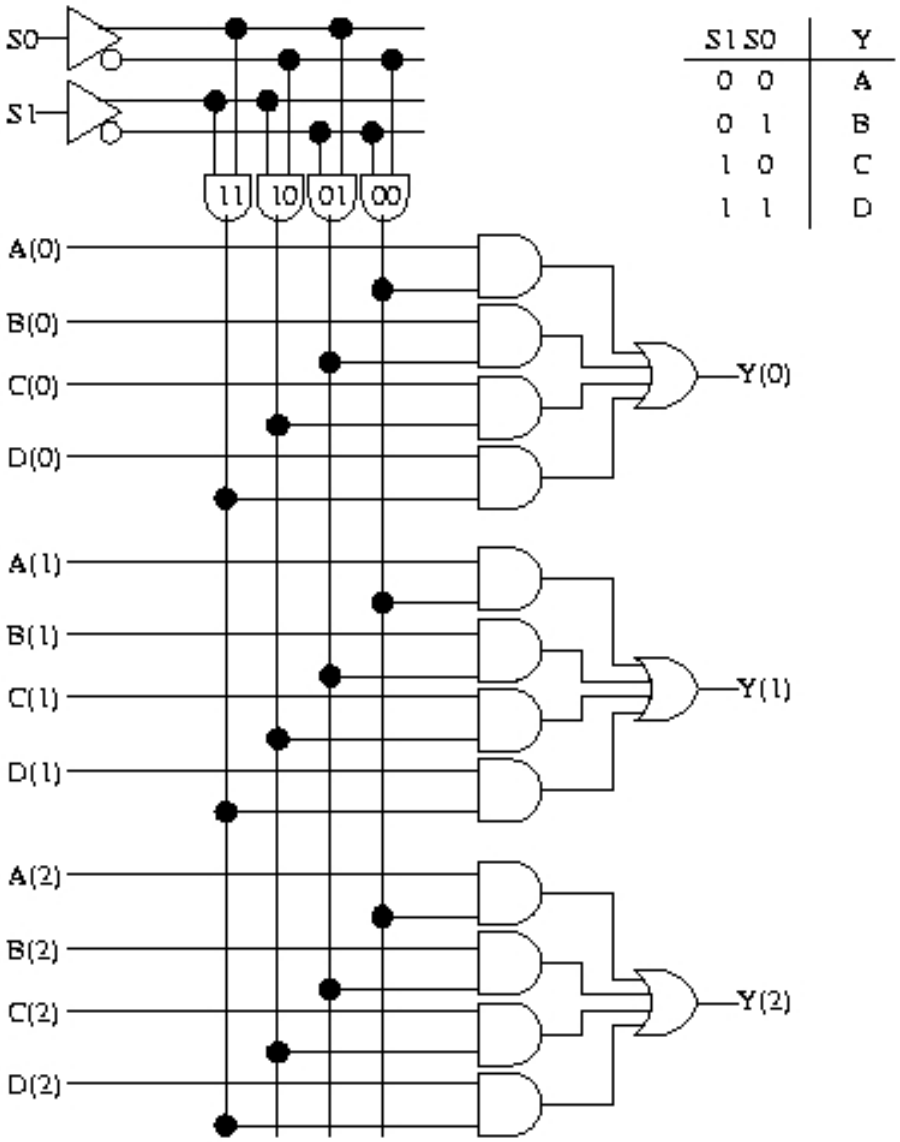


6 x 8 PLA with 8 product terms

Problem 3

Design a 2-bit 4:1 MUX using discrete logic gates. The 2-bit data inputs are A, B, C, and D. The 2-bit data output is Y. Individual bits of A are A_0 and A_1, etc. Assume the select control signals must be decoded.

Solution to problem 3

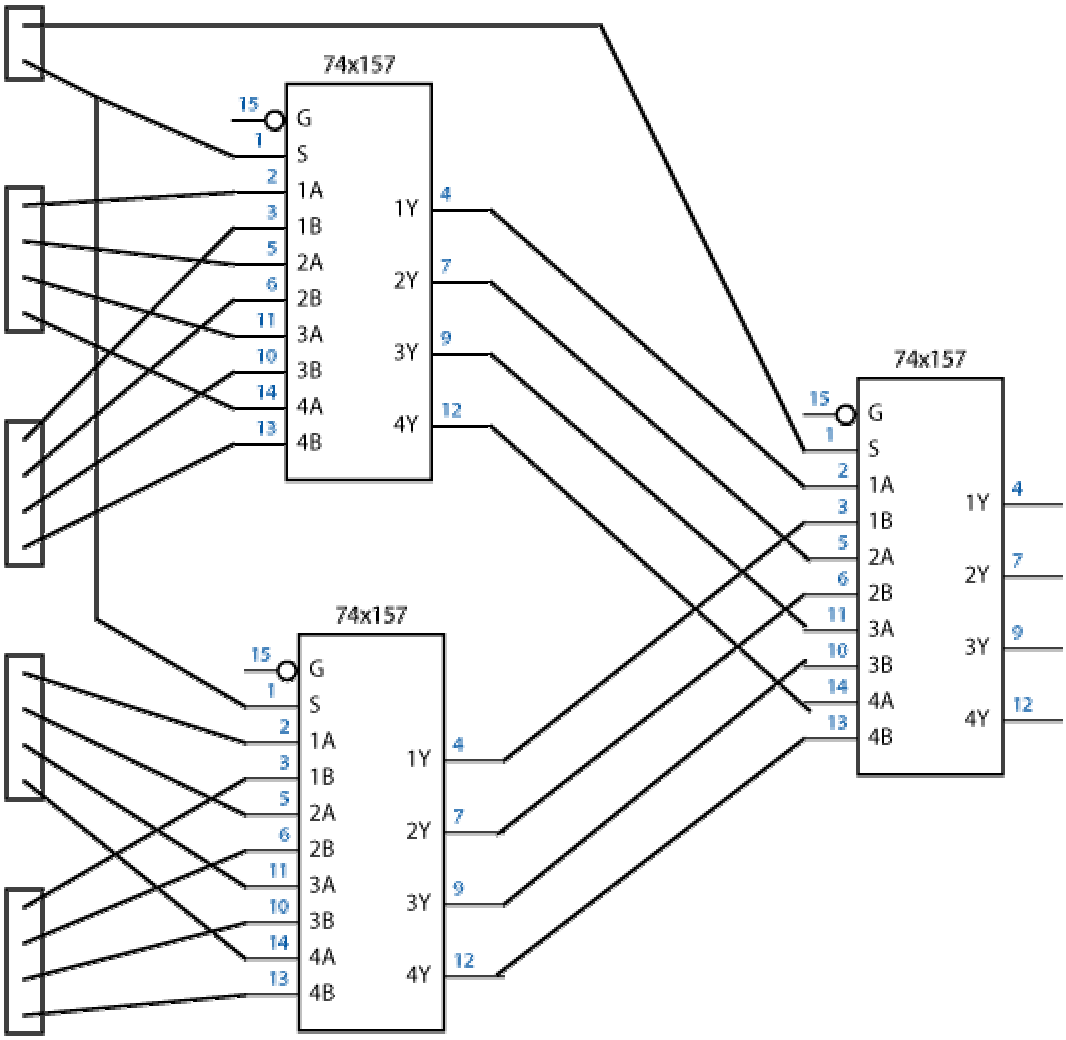


S1 S0	Y
0 0	A
0 1	B
1 0	C
1 1	D

Problem 4

Show how to build a quadruple 4-line to 1-line multiplexers using three quadruple 2-line to 1-line data selectors, in particular the 74157. Use the datasheet for TI's SN74F157A in generating your implementation.

Solution to Problem 4



Problem 5

Design a 2-bit comparator for two-bit values A and B. The comparator has a single output, but two control inputs C_1 and C_0. The output is set using the control inputs as follows:

C_1	C_0	output
0	0	0
0	1	$A \leq B$
1	0	$A \geq B$
1	1	$A \equiv B$

Solution to problem 5

Few restrictions were placed on the nature of the design; consequently there are many ways to solve this problem.

One would be to express the design into a 64-row truth table with the six Boolean variable A_1, A_0, B_1, B_0, C_1, and C_0 are inputs and there is a single output. This can be sent through espresso and optimized into a rather large expression with ten product terms.

Another would be to MSI components. You could use a "conventional" comparator, such as a 7485, to do the comparison and then use a multiplexer to select the right result. Since the 7485 produces $A < B$, $A \equiv B$, and $A > B$ outputs rather than $A \leq B$, $A \equiv B$, and $A \geq B$ outputs; you'll need to use a couple of OR-gates.

Finally, by using an ALU and subtracting A and B, you can generate a one-chip solution.