# UNIVERSITY OF CALIFORNIA, RIVERSIDE

**Department of Computer Science and Engineering**

**Department of Electrical Engineering**

**CS/EE120A – Logic Design**

**Final**

**March 22, 2001**

**35**

**Name:** Solution Key **Student ID#:**

***Please print legibly***

**Lab Section: 21** (MW 3-6):_____ **22** (TR 2-5):______ **23** (MF 8-11):_____

---

(Numbers in parenthesis denote total possible points for question.)

1. Use one $4 \times 8 \times 4$ PLA (4 input columns, 8 rows of AND gates and 4 columns of OR gates) to implement the following two functions: (5)

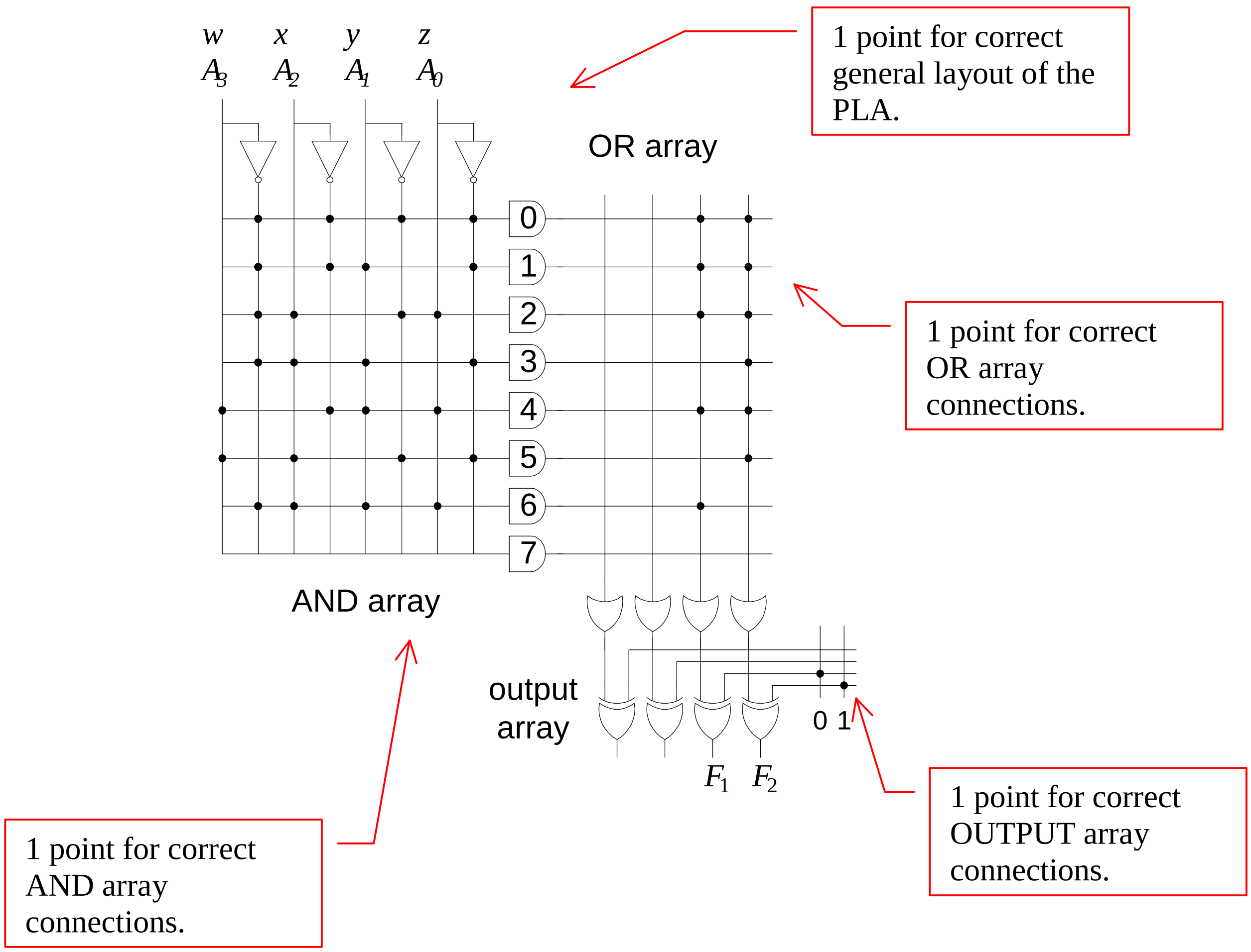
$$F_1(w,x,y,z) = \Sigma(0,2,5,7,11)$$

$$F_2(w,x,y,z) = \Sigma(1,3,4,7,8,9,10,13,14,15)$$

### *Answer*

There are not enough AND gates to implement both functions directly. We will implement $F_2' = \Sigma(0,2,5,6,11,12)$ using 6 minterms of which 4 minterms are the same as those for $F_1$. $F_1$ requires one more minterm $m_7$ that is not yet implemented from $F_2'$. So there are only a total of 7 minterms for the 2 equations. Thus, the following connection:

1 point for realizing that we need to implement $F_1$ and $F_2'$ and knowing the correct minterms to implement.


w x y z
$A_3$ $A_2$ $A_1$ $A_0$
OR array
0
1
2
3
4
5
6
7
AND array
output array
0 1
$F_1$ $F_2$
1 point for correct general layout of the PLA.
1 point for correct OR array connections.
1 point for correct OUTPUT array connections.
1 point for correct AND array connections.

2. Write the VHDL code at the dataflow level that solves the following problem. You need to have both the entity and architecture sections. (5)

   Input (given) a 4-bit data. Output a '1' if there are odd number of 1 bits in the data, otherwise, output a '0'.

### *Answer*

There can be several correct answers to this problem. Below are two possible solutions:

Must have the following entity defined for all solutions:

```
library IEEE;
use ieee.std_logic_1164.all;

ENTITY EvenParity IS
    PORT ( D : IN std_logic_vector(3 downto 0);
           F : OUT std_logic );
END EvenParity;
```

2 points for getting the correct entity.

Solution 1:

```
ARCHITECTURE Dataflow OF EvenParity IS
BEGIN
    with D select
    F <= '1' when "0001" | "0010" | "0100" | "0111" |
                  "1000" | "1011" | "1101" | "1110",
         '0' when others;
END Dataflow;
```

1 point for the overall architecture structure.

Solution 2:

```
ARCHITECTURE Dataflow OF EvenParity IS
BEGIN
    F <= '1' when D = "0001" or D = "0010"
               or D = "0100" or D = "0111"
               or D = "1000" or D = "1011"
               or D = "1101" or D = "1110"
         else '0';
END Dataflow;
```

2 points for the correct **dataflow** code.

Deduct 2 points if code is written in **behavioral** level, i.e. using a **process** statement and / or **case** statement.

3. Design a simplified combinational circuit that solves the problem in question 2 above. You need to come up with the truth table, the simplified function, and finally the circuit. (5)

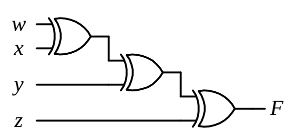
***Answer***

| $d_3$ | $d_2$ | $d_1$ | $d_0$ | $F$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

2 points for getting the truth table.

$$\begin{aligned} F &= w'x'y'z + w'x'yz' + w'xy'z' + w'xyz + wx'y'z' + wx'yz + wxy'z + wxyz' \\ &= w'x'(y \oplus z) + w'x(y \oplus z)' + wx'(y \oplus z)' + wx(y \oplus z) \\ &= [(y \oplus z)\,(w \oplus x)'] + [(y \oplus z)'\,(w \oplus x)] \\ &= w \oplus x \oplus y \oplus z \end{aligned}$$

1 point for getting the correct but not simplified equation.

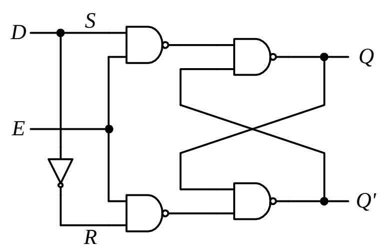



2 points for getting the correct simplified circuit.

The circuit can also be a 4-input XOR gate.

4. Design a D latch with active high enable using only NAND gates and inverters. The primary external signals are *D*, *E*, *Q*, and *Q'*. Clearly label these signals in your circuit. Also, write the truth table for the circuit. (5)

***Answer***




| $E$ | $D$ | $Q$ | $Q_{next}$ | $Q_{next}'$ |
|---|---|---|---|---|
| 0 | × | 0 | 0 | 1 |
| 0 | × | 1 | 1 | 0 |
| 1 | 0 | × | 0 | 1 |
| 1 | 1 | × | 1 | 0 |

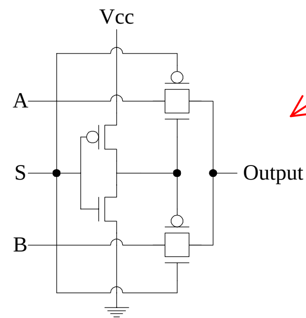
3 points for getting the circuit correct.

2 points for getting the truth table correct.

5. Design a 2-to-1 mux using as few CMOS transistors as possible. There should be only three inputs labeled *A*, *B*, and *S*. (5)

***Answer***

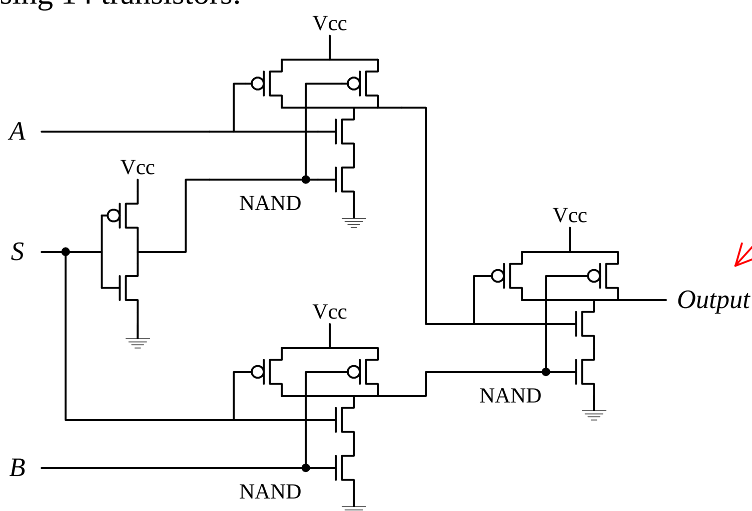
Using only 6 transistors:



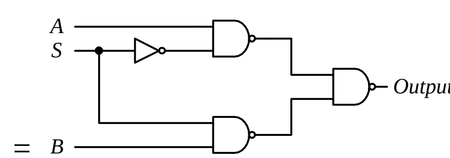

5 points for getting this circuit using only 6 transistors.

4 points only if it has both *S* and *S'*, i.e. without the inverter.

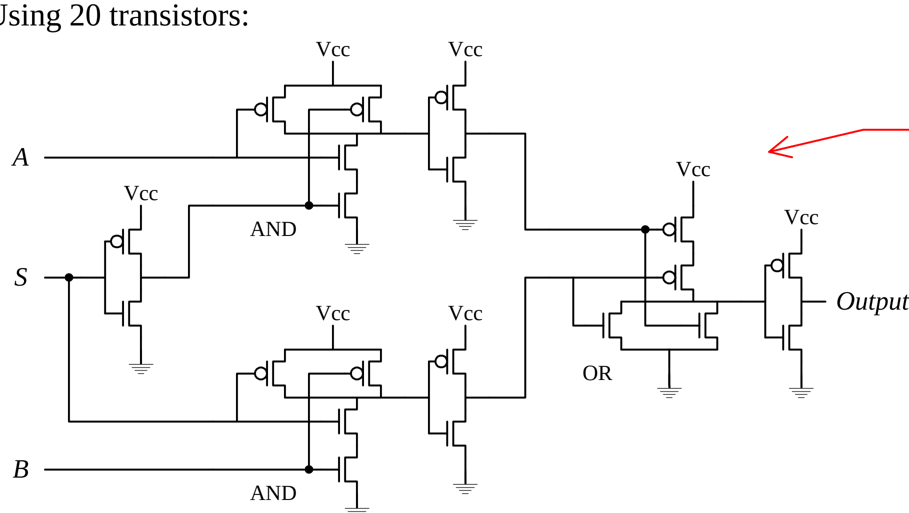
Using 14 transistors:



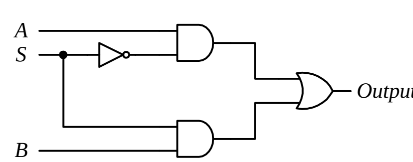

3 points only for getting this circuit using 14 transistors.




Using 20 transistors:




2 points only for getting this circuit using 20 transistors.



6. Design a 2-to-4 decoder using only 3-input NOR gates. To get full credit, you need to use as few gates as possible. (5)

### *Answer*

The truth table for the 2-to-4 decoder is as follows:

| $E$ | $A_1$ | $A_0$ | $C_3$ | $C_2$ | $C_1$ | $C_0$ |
|---|---|---|---|---|---|---|
| 0 | × | × | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |

Deduct 2 points if it does not have an enable $E$ input.

The equations are:

$C_3 = EA_1A_0$
$C_2 = EA_1A_0'$
$C_1 = EA_1'A_0$
$C_0 = EA_1'A_0'$

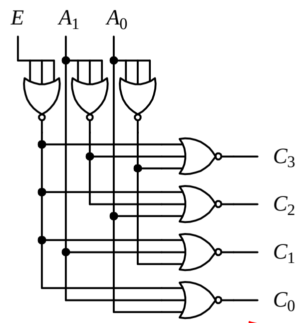
2 points for getting the correct equations.

To change the AND gates to NOR gates, we use the equality:

$$xyz = ((xyz)')' = (x' + y' + z')'$$

To change inverters to NOR gates, we simply connect the inputs together.

Thus, we get the following circuit:

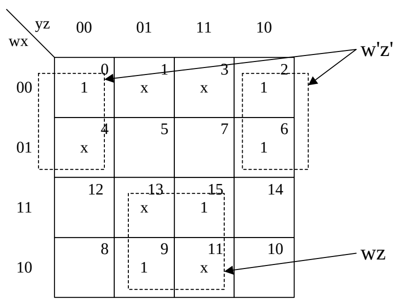



3 points for getting this circuit with only 7 3-input NOR gates.

7. Simplify a four-literal (*w*, *x*, *y*, *z*) Boolean expression with the following 1-minterms and "don't-care"-minterms to use as few gates as possible:

   1-minterms: 0, 2, 6, 9, 15
   d-minterms: 1, 3, 4, 11, 13 (5)

***Answer:***




2 points for the correct K-map.

$F = wz + w'z'$

$= w \odot z$

2 points for this equation as obtained from the K-map.

1 point for the simplified equation.