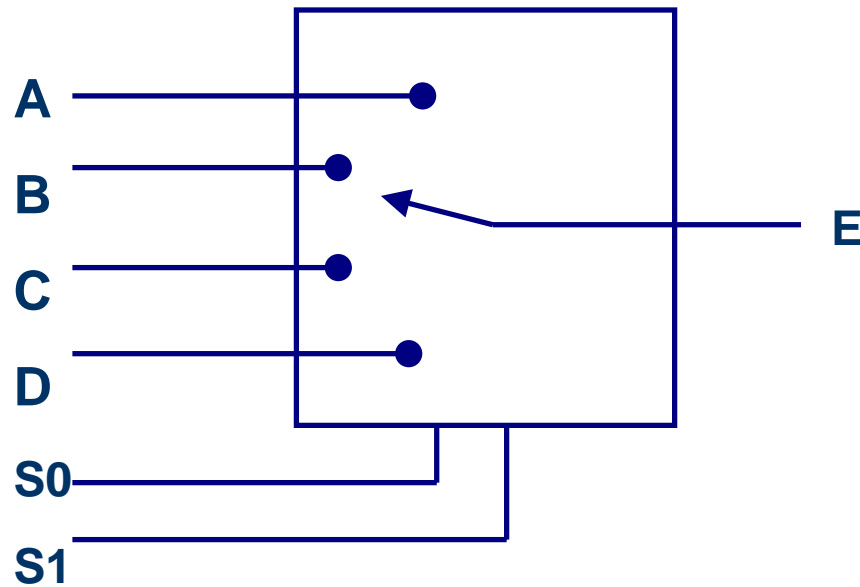


# Recall: Multiplexer



- Input variables connected to select lines
- Minterm values connected to input line

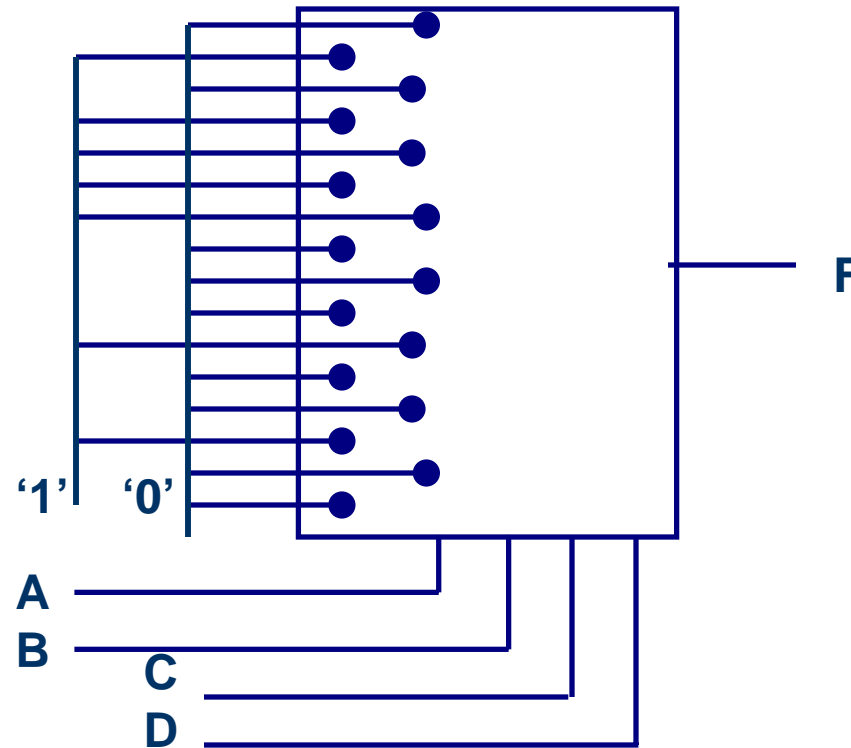
# Problem #1

Given  $F(A,B,C,D) = \Sigma m(1,3,4,5,6,10,13)$

- a.) Implement using a 16-to-1 MUX
- b.) Implement using a 8-to-1 MUX
- c.) Implement using 4-to-1 Multiplexers only

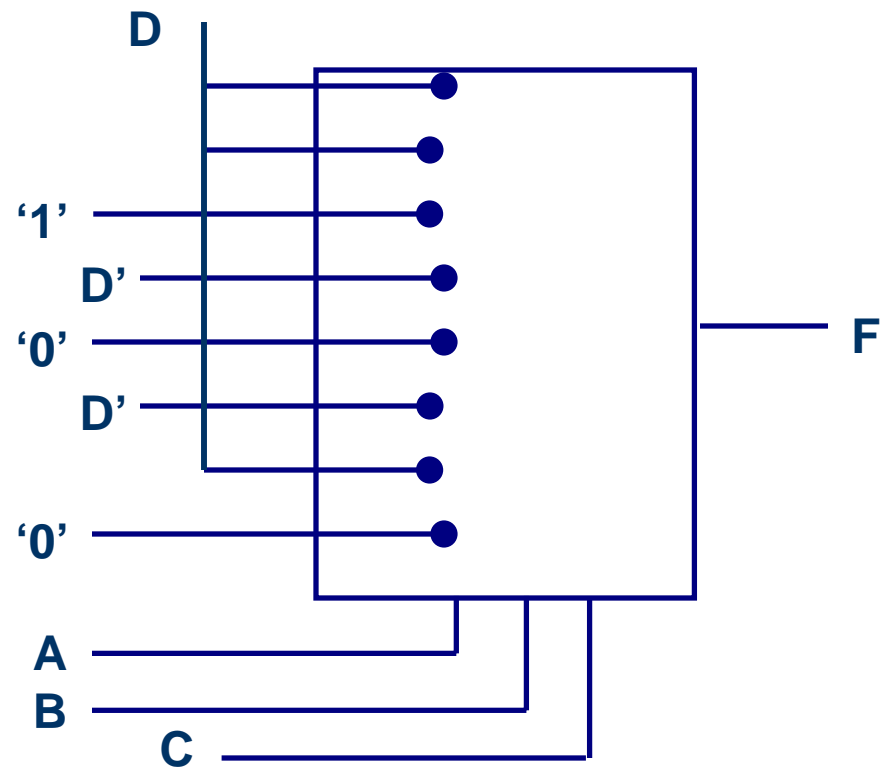
# Problem #1a Solution

Given  $F(A,B,C,D) = \sum m(1,3,4,5,6,10,13)$



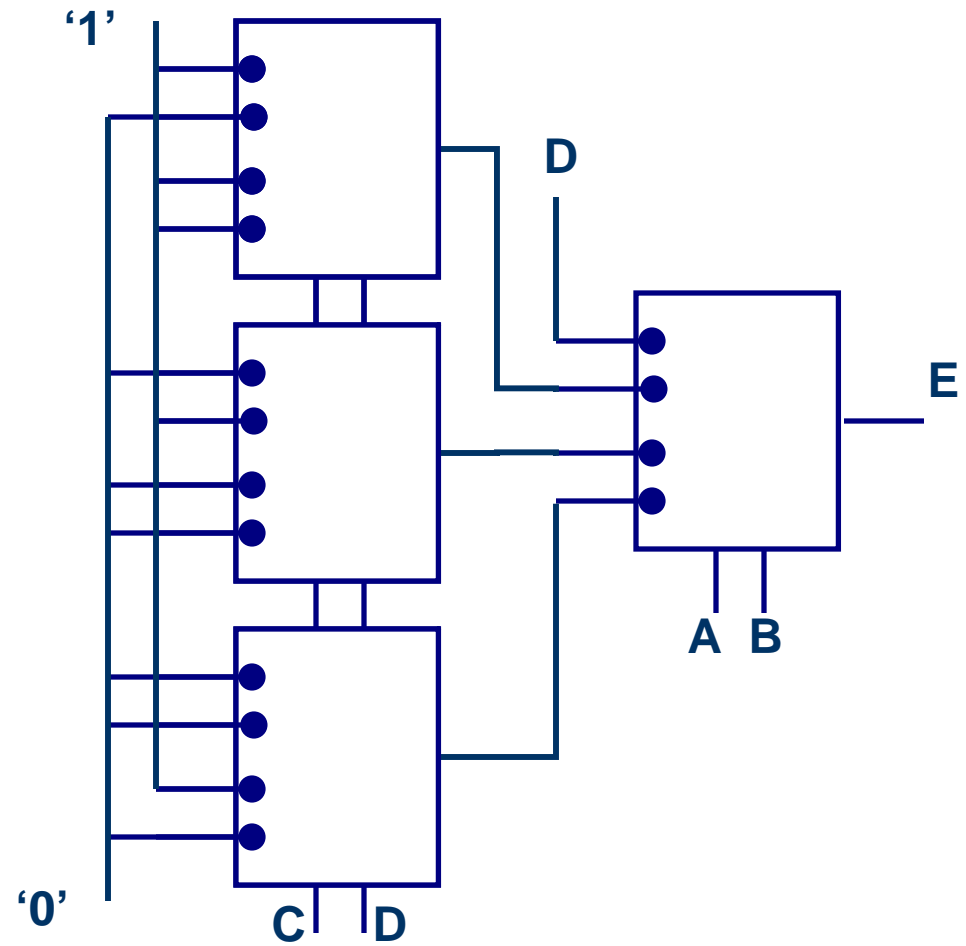
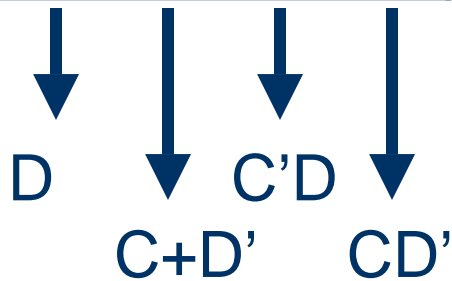
# Problem #1b Solution

AB	00	01	11	10
C				
0	D	1	D	
1	D	D'		D'



# Problem #1c Solution

AB	00	01	11	10
C				
0	D	1	D	
1	D	D'		D'



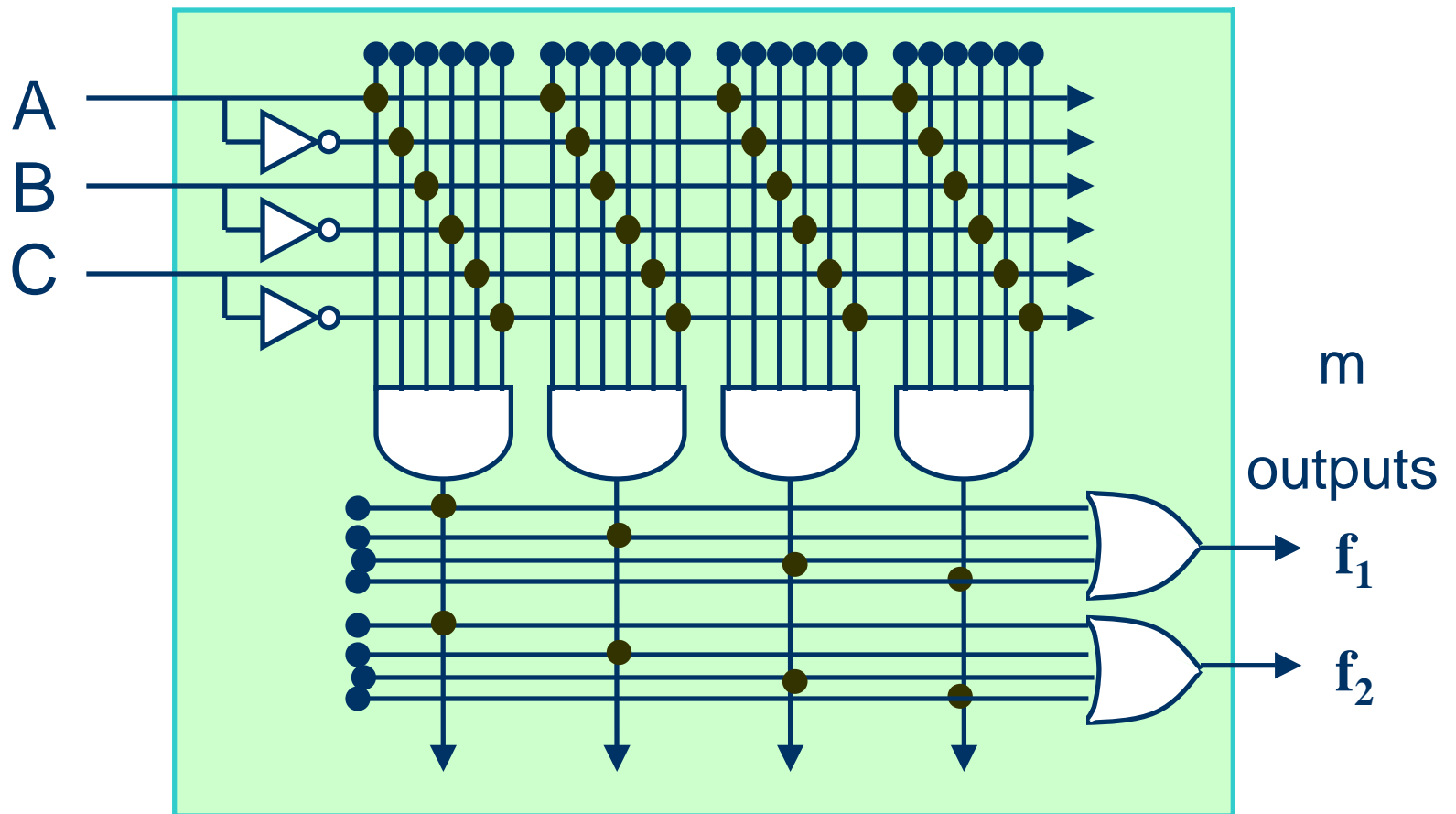


## DC 3-3

Array Logic  
Mixed Rails



# Recall: Array Logic

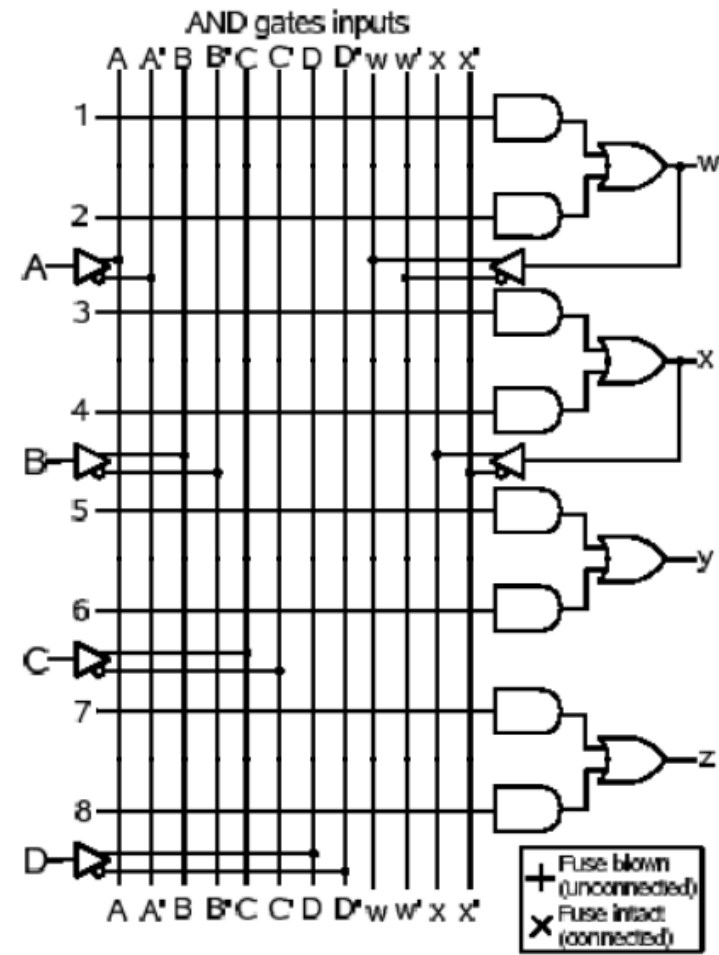


# Recall: Array Logic

	<b>AND Array</b>	<b>OR Array</b>
<b>ROM</b>	fixed	programmable
<b>PLA</b>	programmable	programmable
<b>PAL</b>	programmable	fixed

# Example 1

- The fuse map of a PAL with 4 inputs, 2 product terms per function, 2 output to input connections and 4 outputs is shown on the right.



## Example 1 (cont'd)

- Use the PAL to implement the Boolean functions given below:

$$F_1 = \Sigma m(6, 10, 11, 12, 13)$$

$$F_2 = \Sigma m(0, 2, 6, 8, 10, 12, 13)$$

$$F_3 = \Sigma m(2, 3, 6, 12, 13)$$

# Example 1 Solution

- Step 1: Use K-map to check product terms of the different functions

	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

	00	01	11	10
00	1		1	1
01			1	
11				
10	1	1		1

	00	01	11	10
00			1	
01			1	
11	1			
10	1	1		

Note: all functions need more than 2 product terms to implement. Therefore, need to use multiple output technique

# Example 1 Solution (cont'd)

- Step 2: Check minterms common to all 3 functions

	00	01	11	10
00	0	4	12 <b>1</b>	8
01	1	5	13 <b>1</b>	9
11	3	7	15	11
10	2	6 <b>1</b>	14	10

Note:  $A'B'C' + AB'CD'$  common to all three and can be implemented using one of the output-to-input connections (w)

# Example 1 Solution (cont'd)

- Step 3: Use w in the functions

	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

$$F_1 = w + A'BC$$

	00	01	11	10
00	1		12	8
01			13	9
11			15	11
10	1	6	14	10

$$F_2 = w + BD'$$

	00	01	11	10
00			12	8
01			13	9
11	3	7	15	11
10	1	6	14	10

$$F_3 = w + ABC$$

# Example 1 Solution (cont'd)

- Step 4: create the fuse table

$$1 = A'B'C'$$

$$2 = AB'CD' \quad w = 1 + 2$$

$$3 = w$$

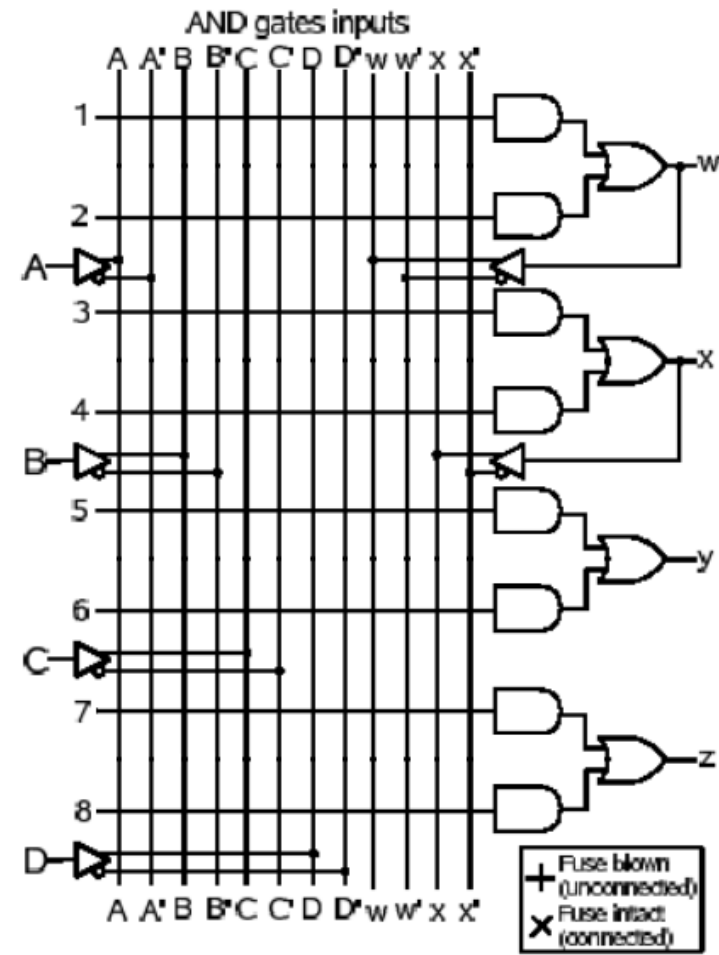
$$4 = A'BC \quad x = F_1 = 3 + 4$$

$$5 = w$$

$$6 = BD' \quad y = F_2 = 5 + 6$$

$$7 = w$$

$$8 = ABC \quad z = F_3 = 7 + 8$$



## Example 2

- You are to implement a combinational multiplier. It has two inputs, each 2-bit wide, and a 4-bit output. The first 2-bit input is represented by variables A,B while the second 2-bit input is represented by C,D. The outputs are W,X,Y,Z from the most significant bit to the least.
- Show how to implement the design using a ROM and a PLA

# Example 2 Solution

- Step 1: Determine the truth table

ABCD	WXYZ		ABCD	WXYZ
0000	0000		1000	0000
0001	0000		1001	0010
0010	0000		1010	0100
0011	0000		1011	0110
0100	0000		1100	0000
0101	0001		1101	0011
0110	0010		1110	0110
0111	0011		1111	1001

## Example 2 Solution

- ROM: We will need a  $2^4 \times 4$  ROM
- ABCD correspond to the address
- Values of WXYZ correspond to the data of the ROM, each output pin corresponding to W, X, Y and Z

# Example 2 Solution

- PLA (assuming a large PLA is available)

Product Terms	Inputs: A B C D	Outputs: W X Y Z
$A' B C' D$	0 1 0 1	0 0 0 1
$A' B C D'$	0 1 1 0	0 0 1 0
$A' B C D$	0 1 1 1	0 0 1 1
$A B' C' D$	1 0 0 1	0 0 1 0
$A B' C D'$	1 0 1 0	0 1 0 0
$A B' C D$	1 0 1 1	0 1 1 0
$A B C' D'$	1 1 0 0	0 0 0 0
$A B C' D$	1 1 0 1	0 0 1 1
$A B C D'$	1 1 1 0	0 1 1 0
$A B C D$	1 1 1 1	1 0 0 1

# Example 2 Solution

- If the size of the PLA is limited, we need to consider product terms

	00	01	11	10
00				
01				
11			1	
10				

	00	01	11	10
00				
01				
11				1
10			1	1

	00	01	11	10
00				
01			1	1
11		1		1
10		1	1	

	00	01	11	10
00				
01		1	1	
11		1	1	
10				

DC 3-3

EEE 21

# Example 2 Solution

- Based on previous maps

Product Terms	Inputs: A B C D	Outputs: W X Y Z
A B C D	1 1 1 1	1 0 0 0
A C D'	1 - 1 0	0 1 0 0
A B' C	1 0 1 -	0 1 0 0
A' B C	0 1 1 -	0 0 1 0
B C D'	- 1 1 0	0 0 1 0
A B' D	1 0 - 1	0 0 1 0
A C' D	1 - 0 1	0 0 1 0
B D	- 1 - 1	0 0 0 1