# Sequential logic design

Vikas Dhiman for ECE275

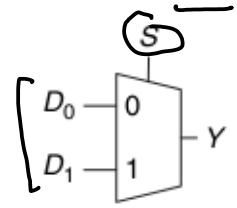
December 5, 2022

## 1 Objectives

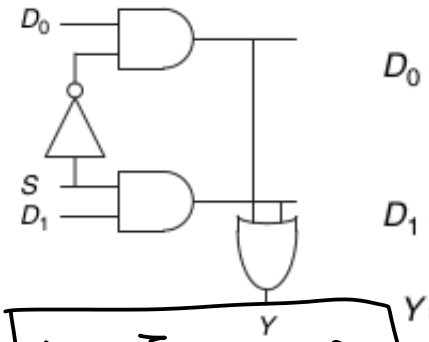
1. Design combinational circuits using multiplexers and decoders

## 2 Design combinational circuit using multiplexers [1, Section 2.8.1]

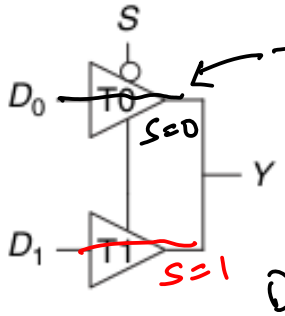
### 2.1 Review: 2to1 Multiplexer (MUX)




| S | $D_1$ | $D_0$ | Y |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |



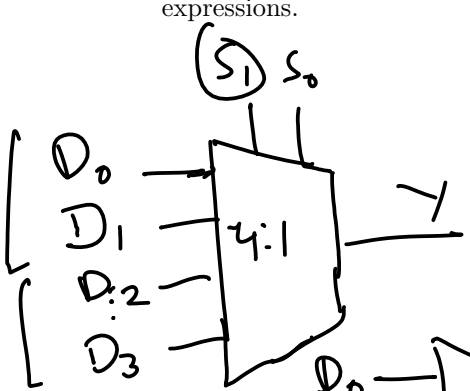

$Y = \bar{S} D_0 + S D_1$



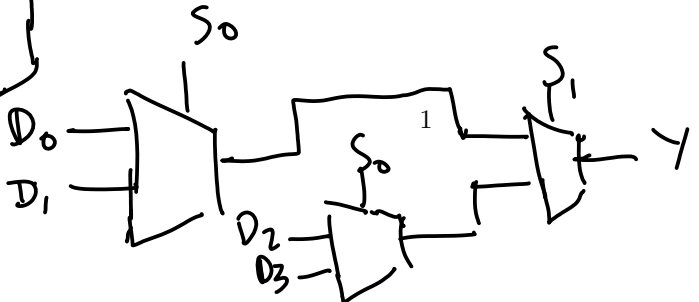

$Y = D_0\bar{S} + D_1 S$

### 2.2 Wider multiplexers

Draw the symbol for a 4:1 MUX, an 8:1 MUX and a $2^N : 1$ MUX and write corresponding Boolean expressions.




| $S_1$ | $S_0$ | Y |
|---|---|---|
| 0 | 0 | $D_0$ |
| 0 | 1 | $D_1$ |
| 1 | 0 | $D_2$ |
| 1 | 1 | $D_3$ |



$$Y = \bar{S}_1 \bar{S}_0 D_0 + \bar{S}_1 S_0 D_1 + S_1 \bar{S}_0 D_2 + S_1 S_0 D_3$$

Only 1 minterm can be true at a time. The minterm that is true, selects the data line.

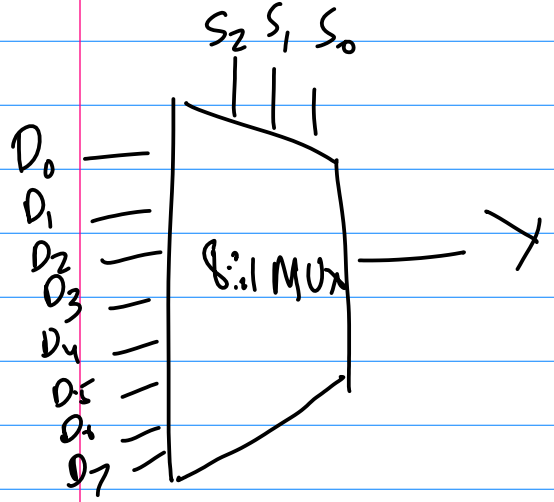
Minterms: $\bar{S}_1 \bar{S}_0$, $\bar{S}_1 S_0$, $S_1 \bar{S}_0$, $S_1 S_0$

Only 1 minterm can be true at a time.
The minterm that is true, selects the corresponding data line.

8:1 MUX

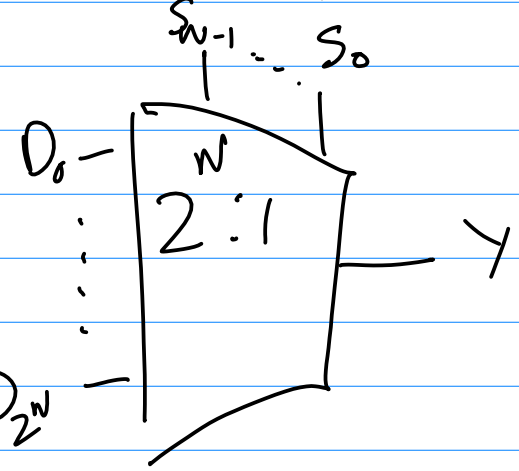
Draw a symbol, write an expression and TT

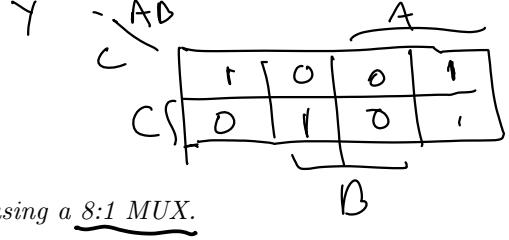
How many select lines?



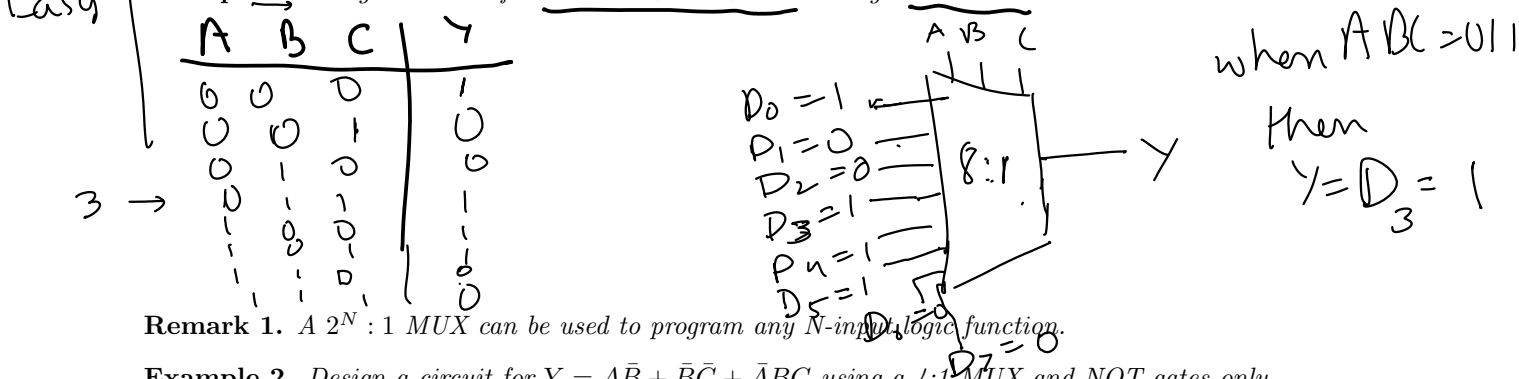

$$Y = \bar{S}_2 \bar{S}_1 \bar{S}_0 D_0 + \bar{S}_2 \bar{S}_1 S_0 D_1 + \bar{S}_2 S_1 \bar{S}_0 D_2 + \cdots + S_2 S_1 S_0 D_7$$

| $S_2$ | $S_1$ | $S_0$ | Y |
|---|---|---|---|
| 0 | 0 | 0 | $D_0$ |
| 0 | 0 | 1 | $D_1$ |
| 0 | 1 | 0 | $D_2$ |
| ⋮ | | | |
| 1 | 1 | 1 | $D_7$ |






**Example 1.** *Design a circuit for $Y = A\bar{B} + \bar{B}\bar{C} + \bar{A}BC$ using a 8:1 MUX.*




**Remark 1.** *A $2^N : 1$ MUX can be used to program any N-input logic function.*

**Example 2.** *Design a circuit for $Y = A\bar{B} + \bar{B}\bar{C} + \bar{A}BC$ using a 4:1 MUX and NOT gates only.*

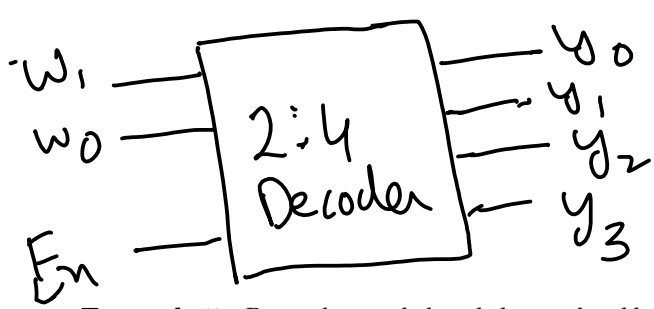
(1) Design which input should go to the SELECT inputs
and " " " " " " DATA inputs

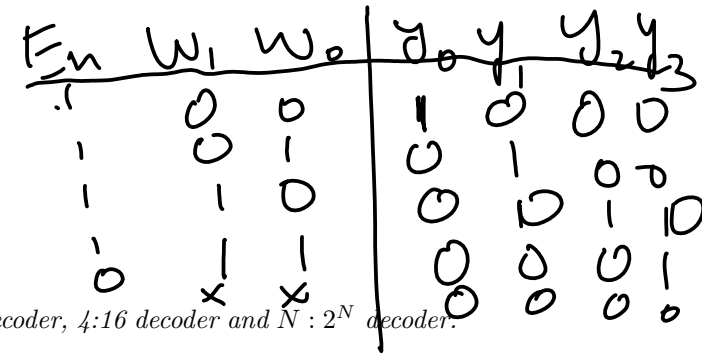
**Remark 2.** *A $2^{N-1} : 1$ MUX can be used to program any N-input logic function, if we use literals on the input side.*

**Example 3.** *Design a circuit for $Y = \bar{A}C + \bar{A}B + B\bar{D}$ using a 8:1 MUX and NOT gates only. Also design using 4:1 MUX and other gates. fewest gates.*

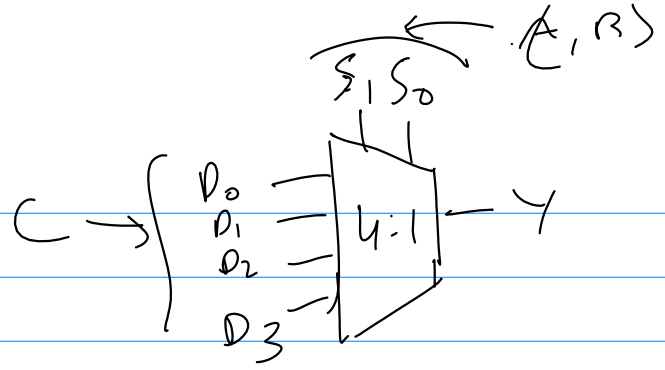
## 3 Encoders and Decoders

**Example 4.** *Draw the symbol and the truth table for 2:4 decoder. Also write the logic expressions.*




**Example 5.** *Draw the symbol and the truth table for 3:8 decoder, 4:16 decoder and $N : 2^N$ decoder. Also write the logic expressions.*



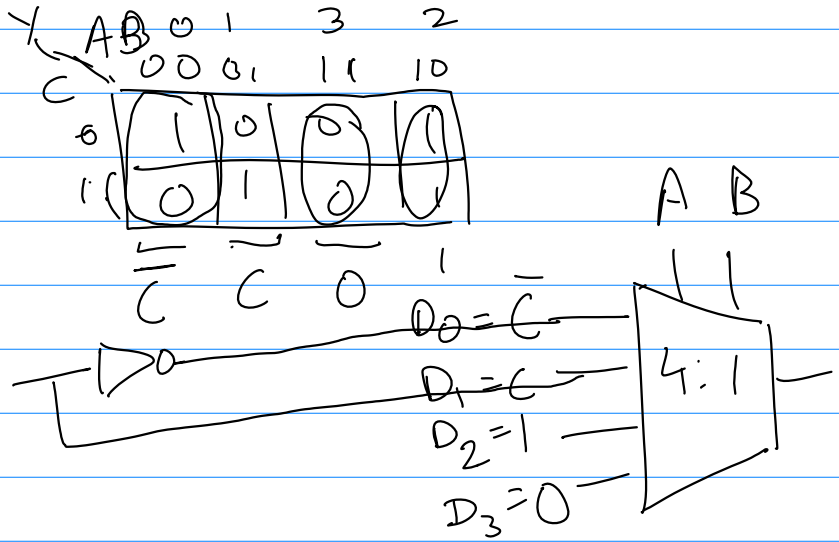

$Y = A\bar{B} + \bar{B}\bar{C} + \bar{A}BC$

The input signals that occur in most implicants go to the select bits.

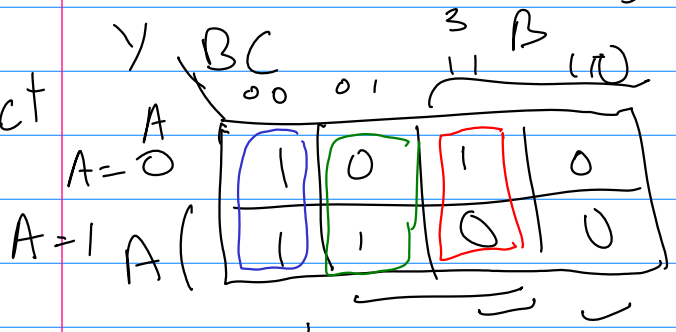
B must connet to the selectsign

because B occurs 3 times in the implicants while A and C occur 2 times.

AB goes to SELECT signals




BC select




when BC = 00

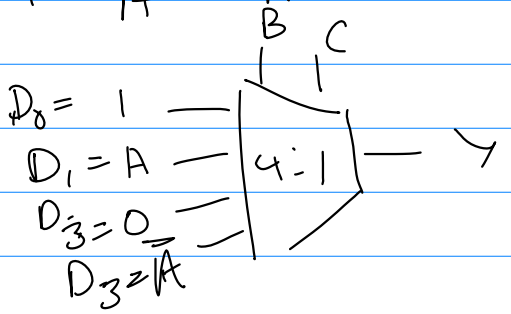
Y = 1

when BC = 01,

Y = A

| A | Y |
|---|---|
| 0 | 0 |
| 1 | 1 |

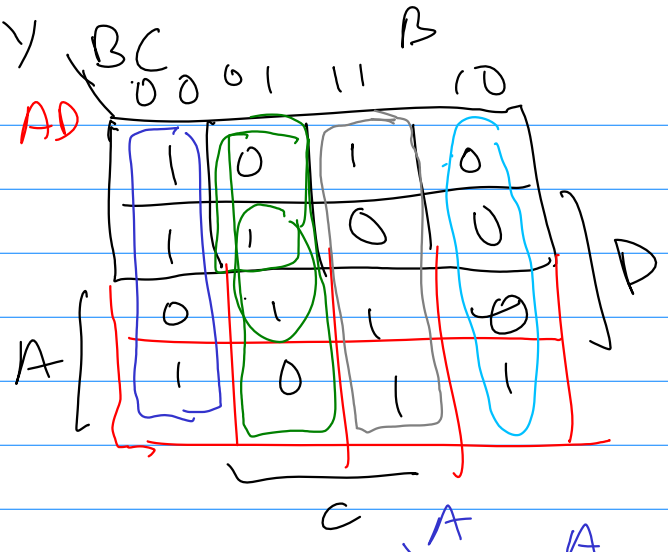
$Y = A\bar{B} + \bar{B}\bar{C} + \bar{A}BC$

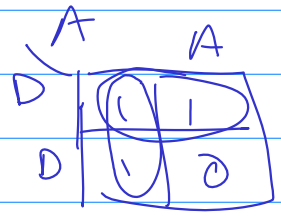


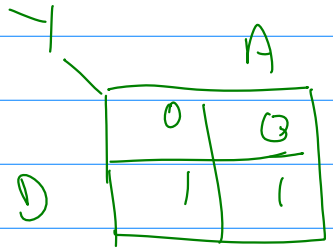

- when B = 1,

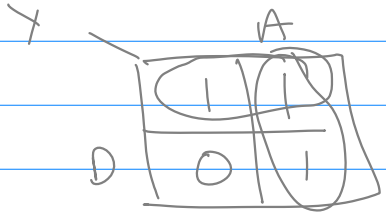
| A | Y |
|---|---|
| 0 | 1 |
| 1 | 0 |

$Y = \bar{A}$




when $BC = 00$

$Y = f(A, D) = ?$

$Y = \bar{A} + \bar{D}$

when $BC = 01$

$Y = D$

when $BC = 11$

$Y = A + \bar{D}$

when $BC = 10$

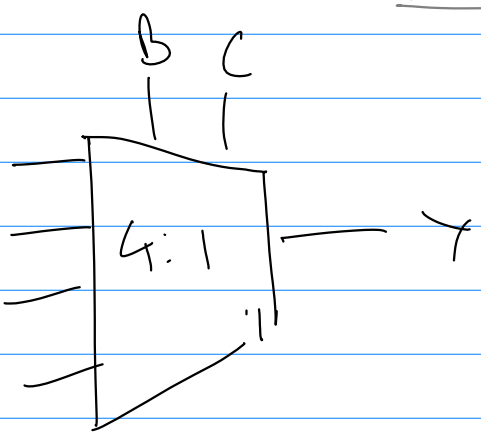
$Y = A\bar{D}$

$D_0 = \bar{A} + \bar{D}$

$D_1 = D$

$D_2 = A + \bar{D}$

$D_3 = A\bar{D}$



**Example 6.** *Design a circuit for a XOR gate using a 2:4 decoder and an OR gate.*

**Example 7.** *Design a circuit for* $Y = A\bar{B} + \bar{B}\bar{C} + \bar{A}BC$ *using a 3:8 decoder and an OR gate.*

### 3.1 Encoders

**Example 8.** *Draw symbol and truth table for a 4:2 priority encoder.*

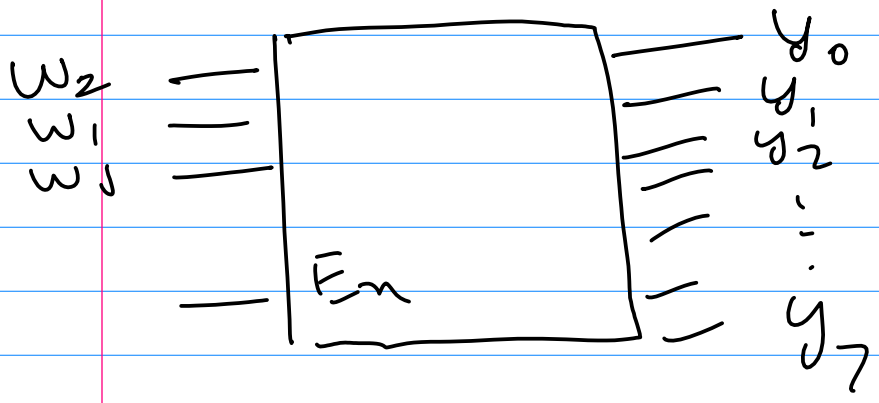
**Example 9.** *Draw symbol and truth table for a 8:3 priority encoder.*

## References

[1] Sarah L Harris and David Harris. *Digital design and computer architecture*. Morgan Kaufmann, 2022.

2:4 Decoder

3:8 Decoder




| En | $w_2$ | $w_1$ | $w_0$ | $y_0$ $y_1$ --- $y_2$ |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | |
| 1 | ⋮ | ⋮ | ⋮ | |
| 1 | 1 | 0 | 1 | 0 0 .. $y_5=1$ --0 |
| ⋮ | ⋮ | ⋮ | ⋮ | |

$y_0 = En \cdot \overline{w}_2 \overline{w}_1 \overline{w}_0$

$y_5 = En \cdot w_2 \overline{w}_1 w_0$ minterm

$5_{10} = 101_2$

$y_6 = En \cdot w_2 w_1 \overline{w}_0$

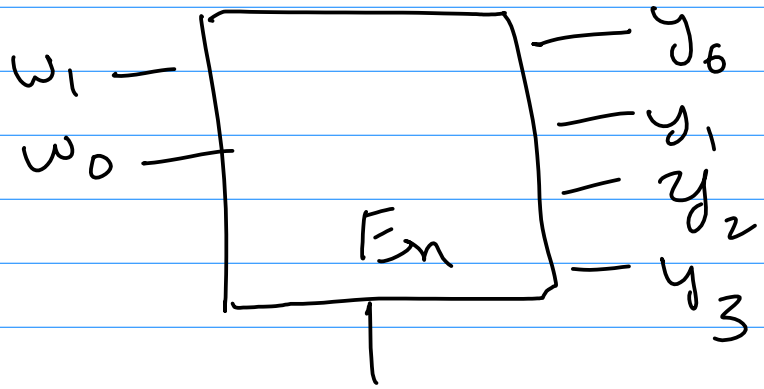
Ex 6

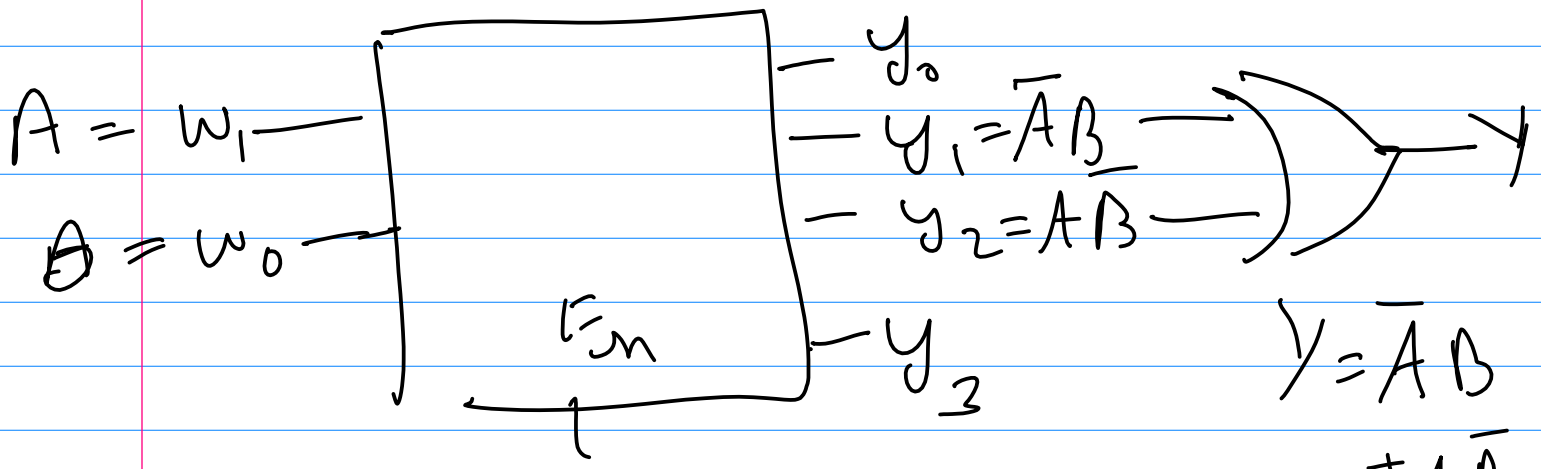
Use decoder to design circuit

$Y = \bar{A}B + A\bar{B}$ XOR gate

$\bar{A}B$ → 0 1

$A\bar{B}$ → 1 0




$Y = \bar{A}B + A\bar{B}$

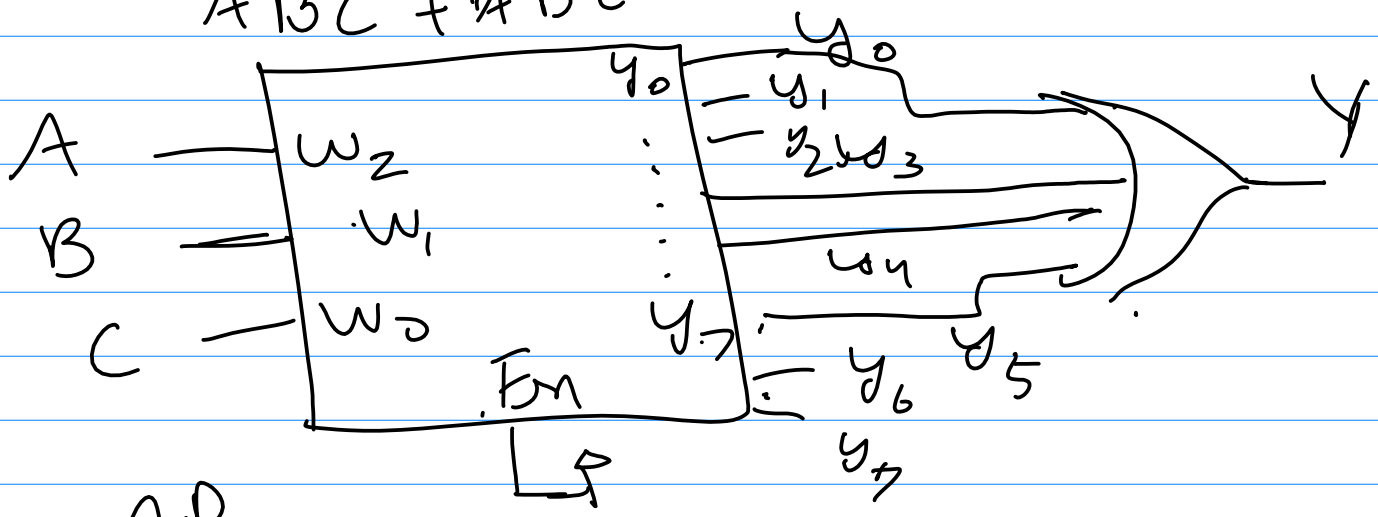
Ex 1

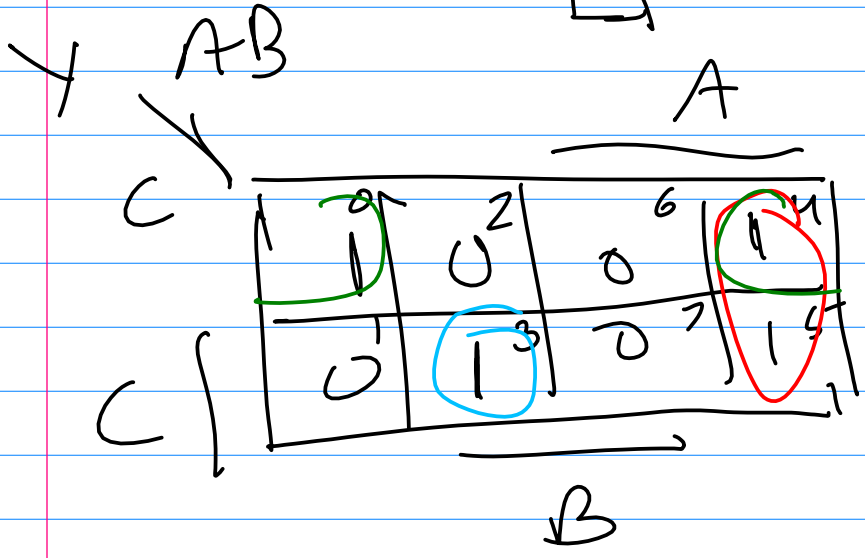
$Y = A\bar{B}(C+\bar{C})$ → $(A+\bar{A})BC$

$Y = A\bar{B} + \bar{B}\bar{C} + \bar{A}BC$

3:8 Decoder

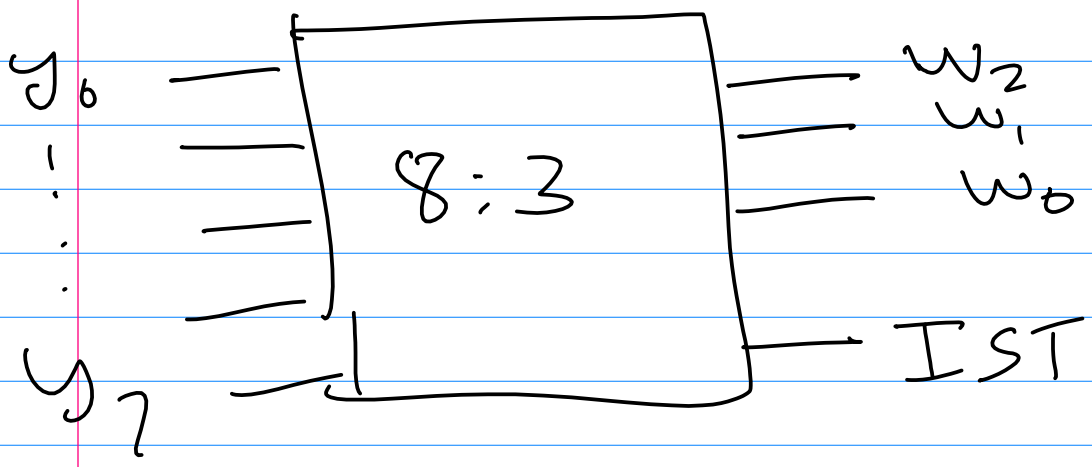
$A\bar{B}C + \bar{A}\bar{B}\bar{C}$




$Y = m_0 + m_3 + m_4 + m_5$

# Priority Encoder




| $y_0$ | $y_1$ | $y_2$ | ... | $y_7$ | $w_2$ | $w_1$ | $w_0$ | IST |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | ... | 0 | x | x | x | 0 |
| 1 | x | x | ... | x | 0 | 0 | 0 | 1 |
| 0 | 1 | x | ... | x | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | ... | x | 0 | 1 | 0 | 1 |

$w_2 =$