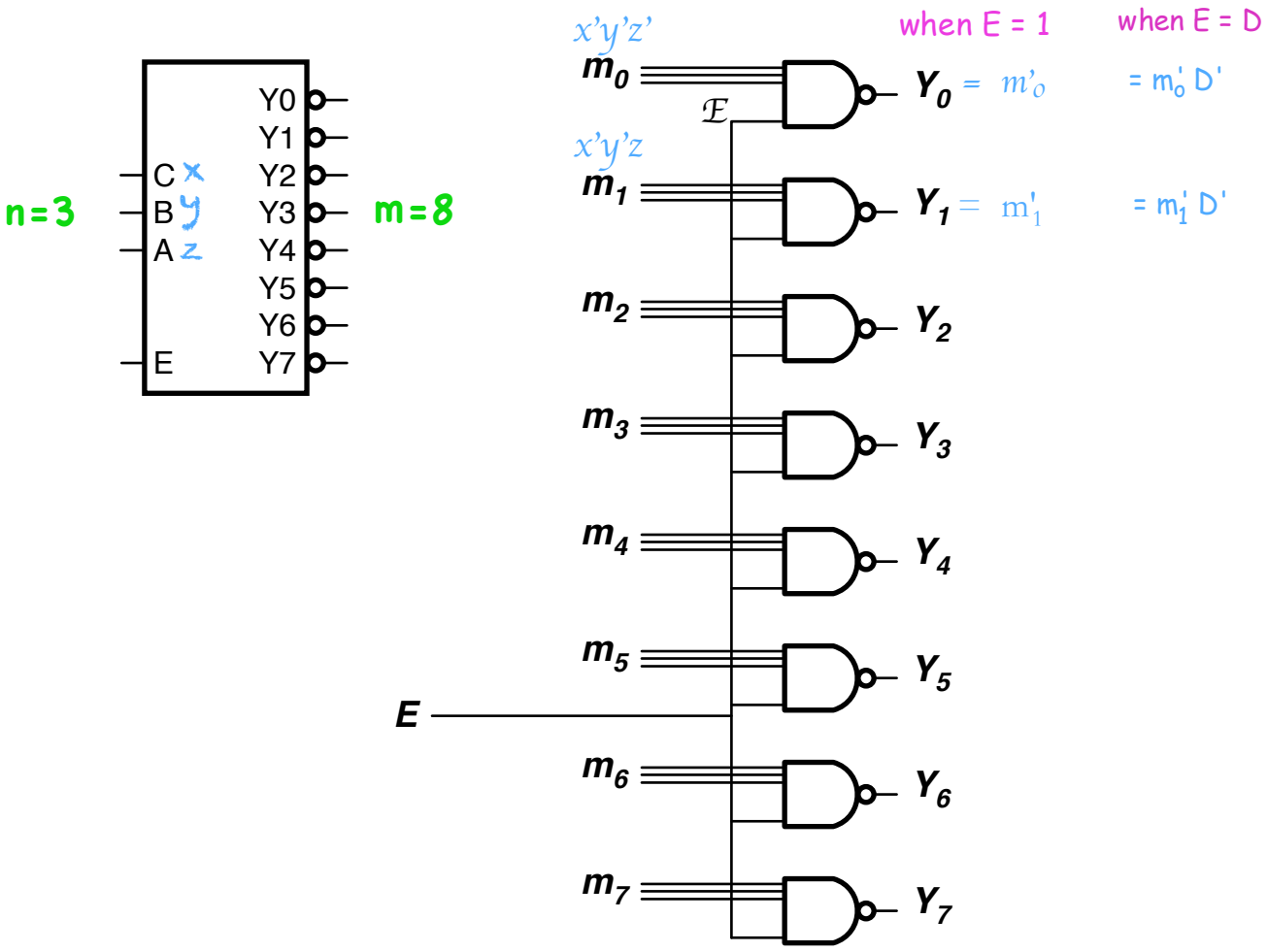
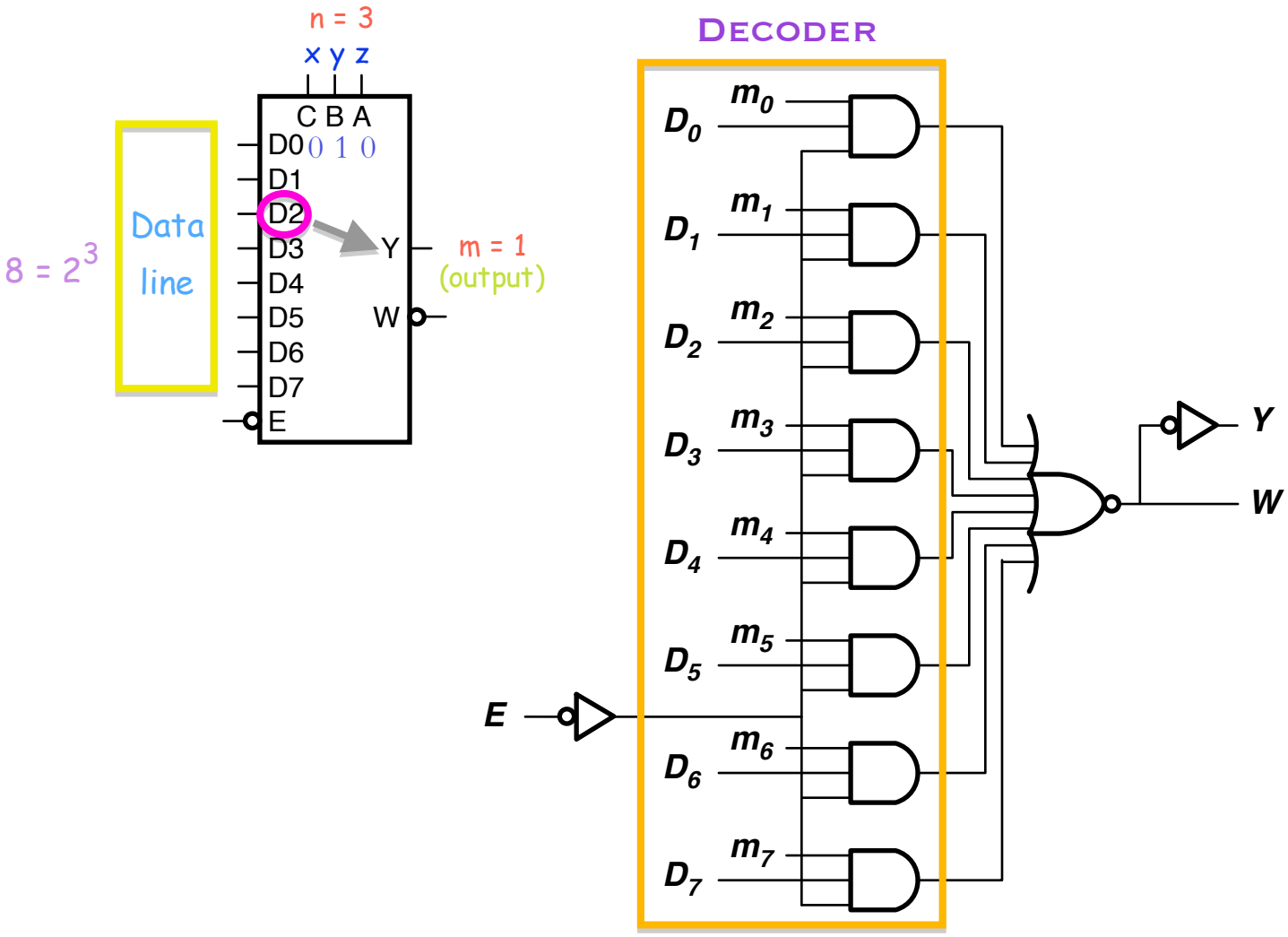
3-line to 8-line Decoder



Summary: Yi = M'i * E'

(if E = 1 ; Yi = M'i
if E = 0; Yi = 1)

8-line to 1-line MUX:



SUMMARY: Y = (D0M0 + D1M1 + D2M2 + ... +D7M7)E

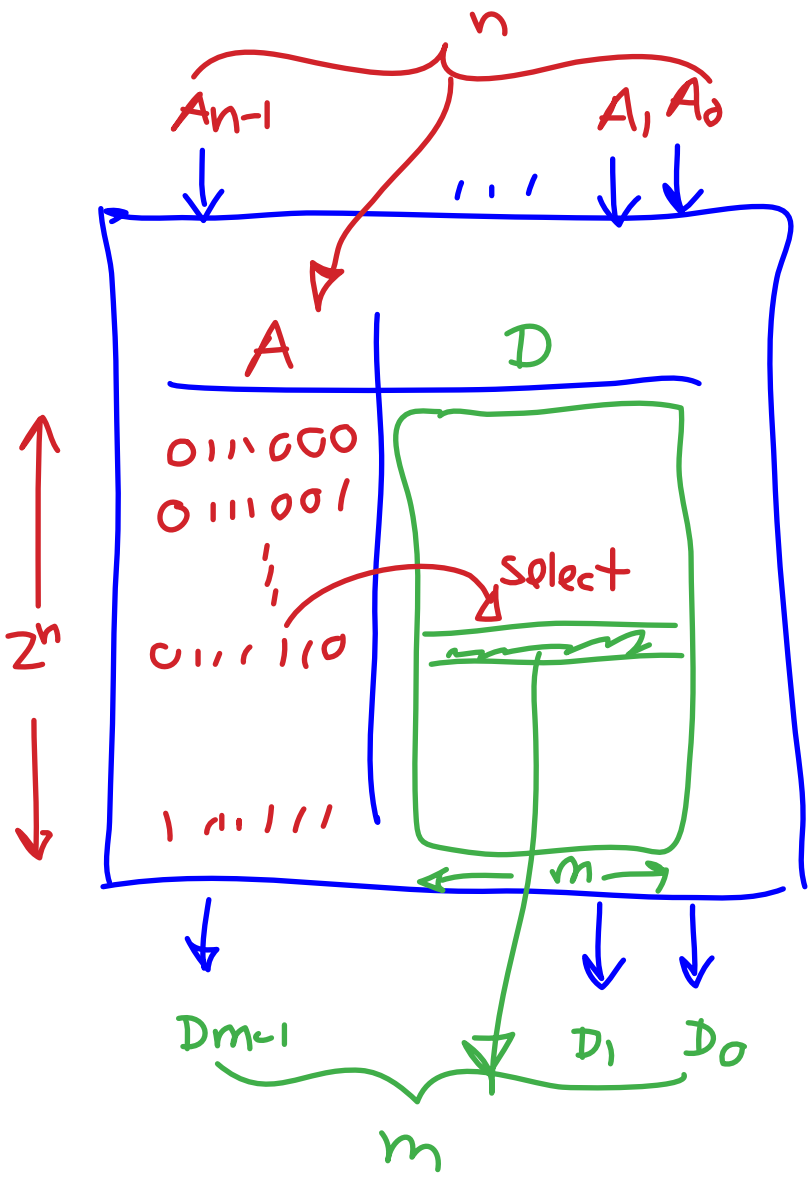
if E = 0 : Y = D0M0 + D1M1 + D2M2 + ... +D7M7

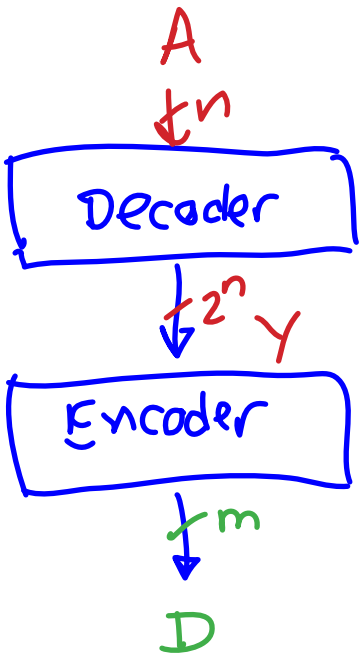
and xyz = 010 then Y = D2 ENABLE

if E = 1 : then Y = 0 DISABLE

Y=(D0*m0+D1*m1+D2*m2+...+D7*m7)*E)

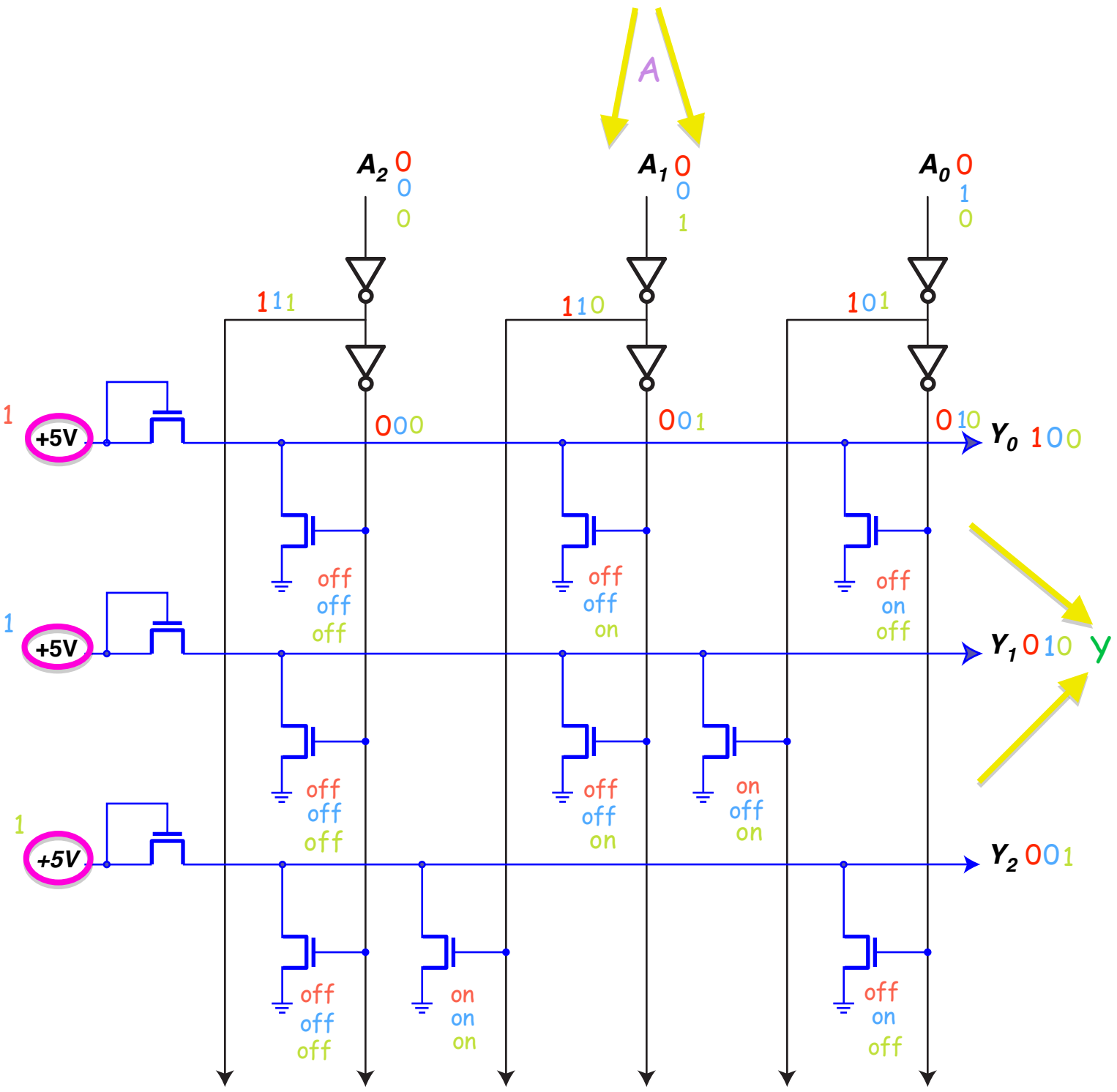
ROM: Stores a truth table



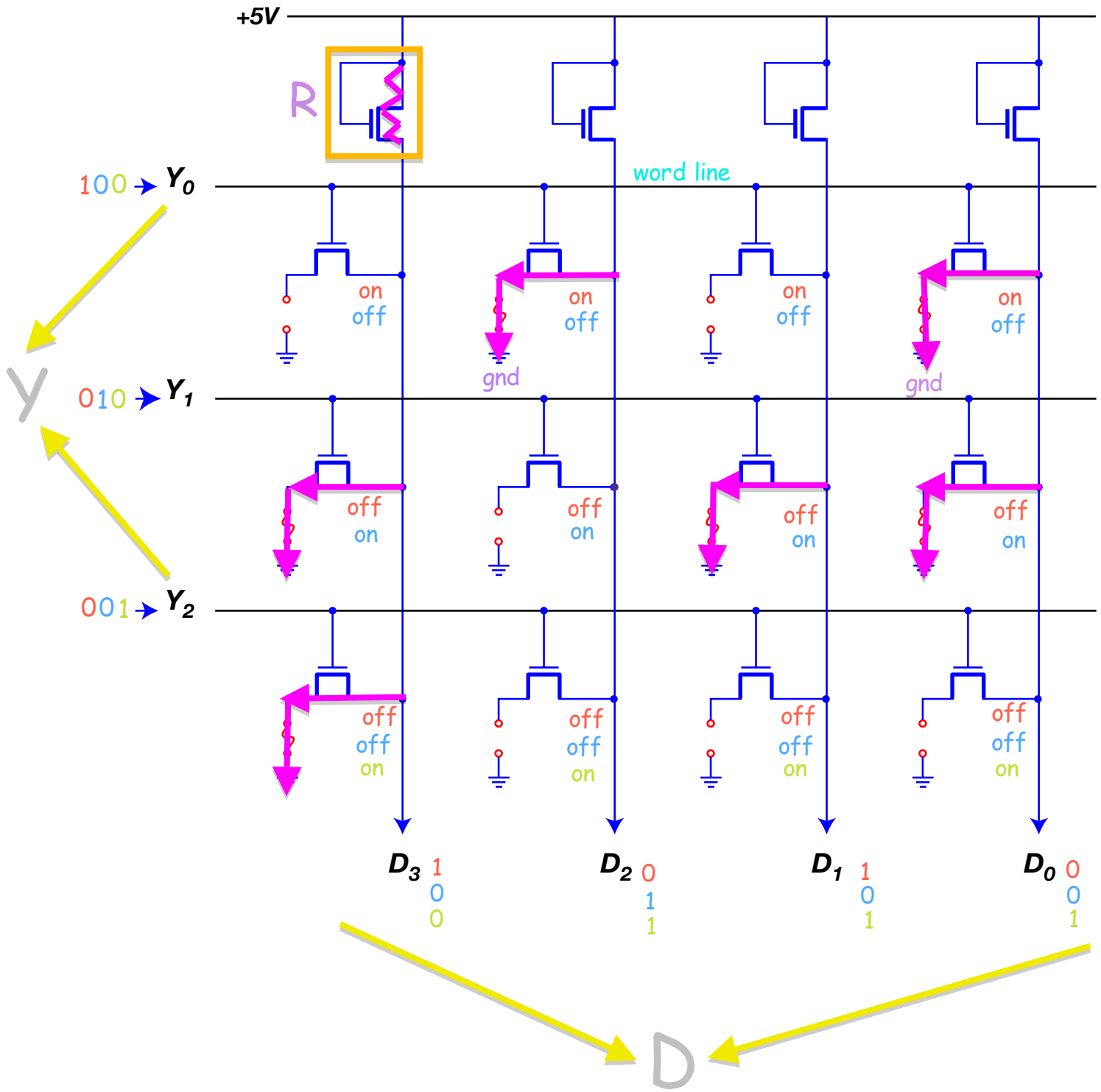


$C = 2^n \times m$

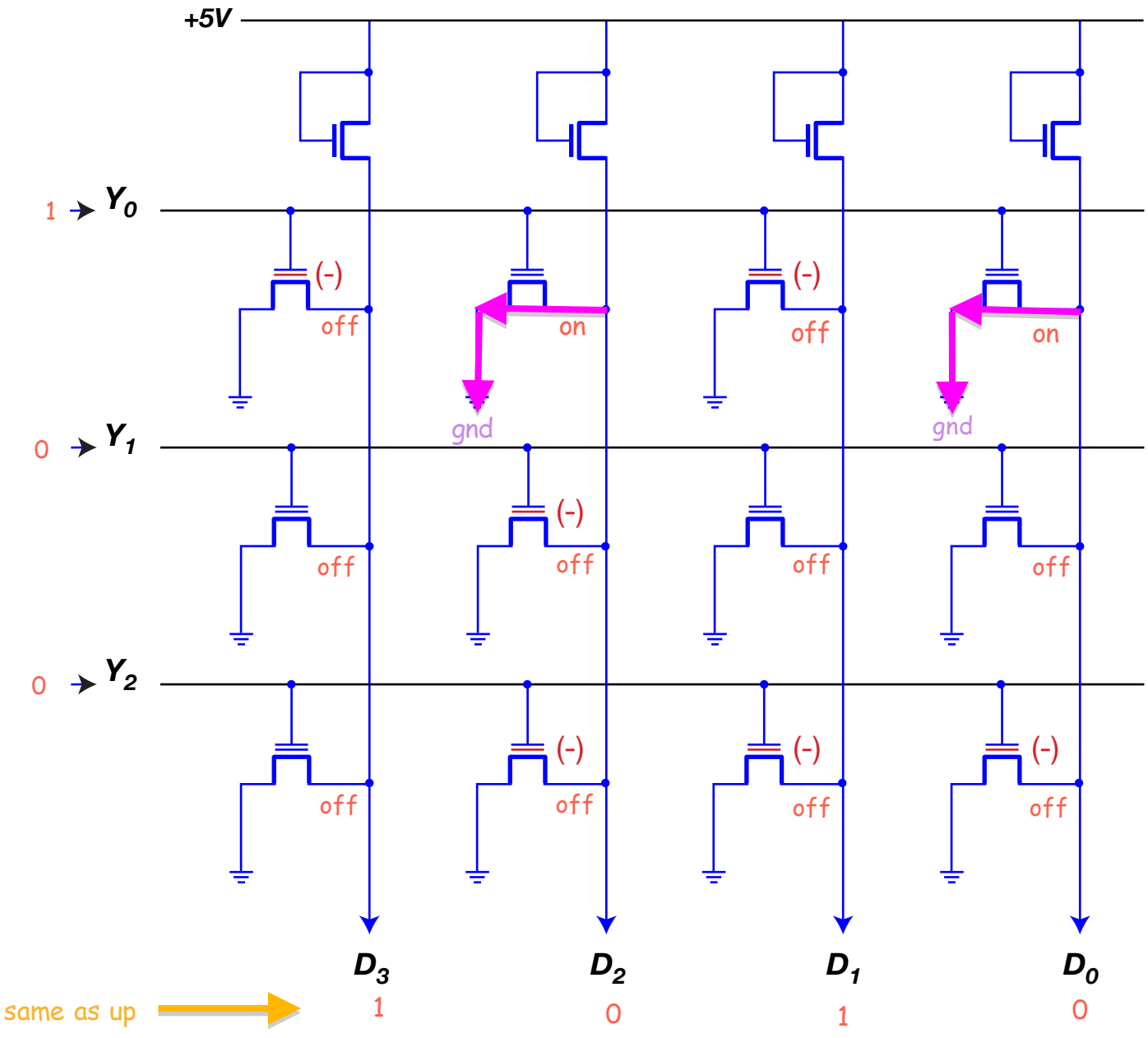
ROM NMOS Decoder



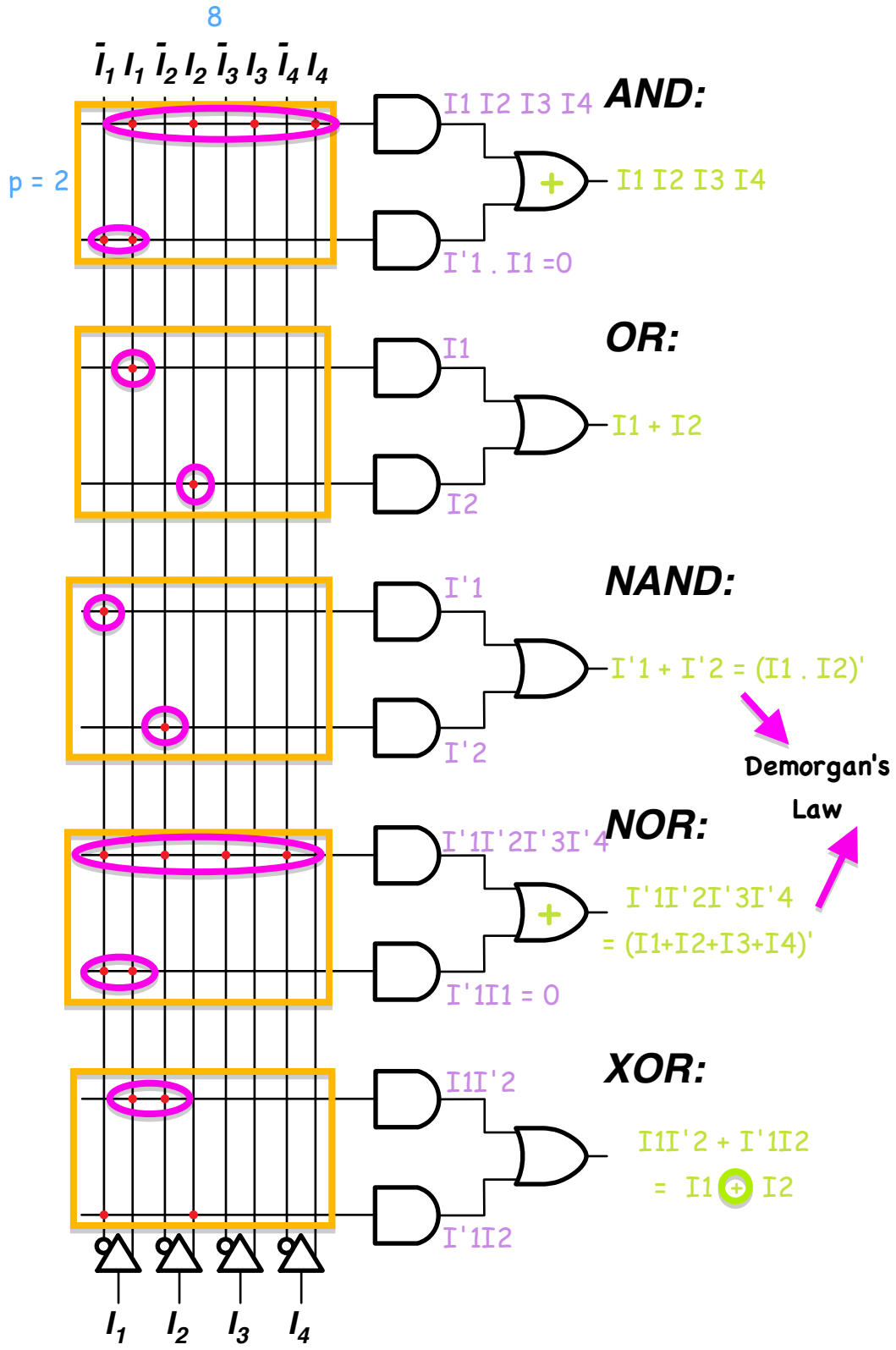
ROM NMOS Encoder



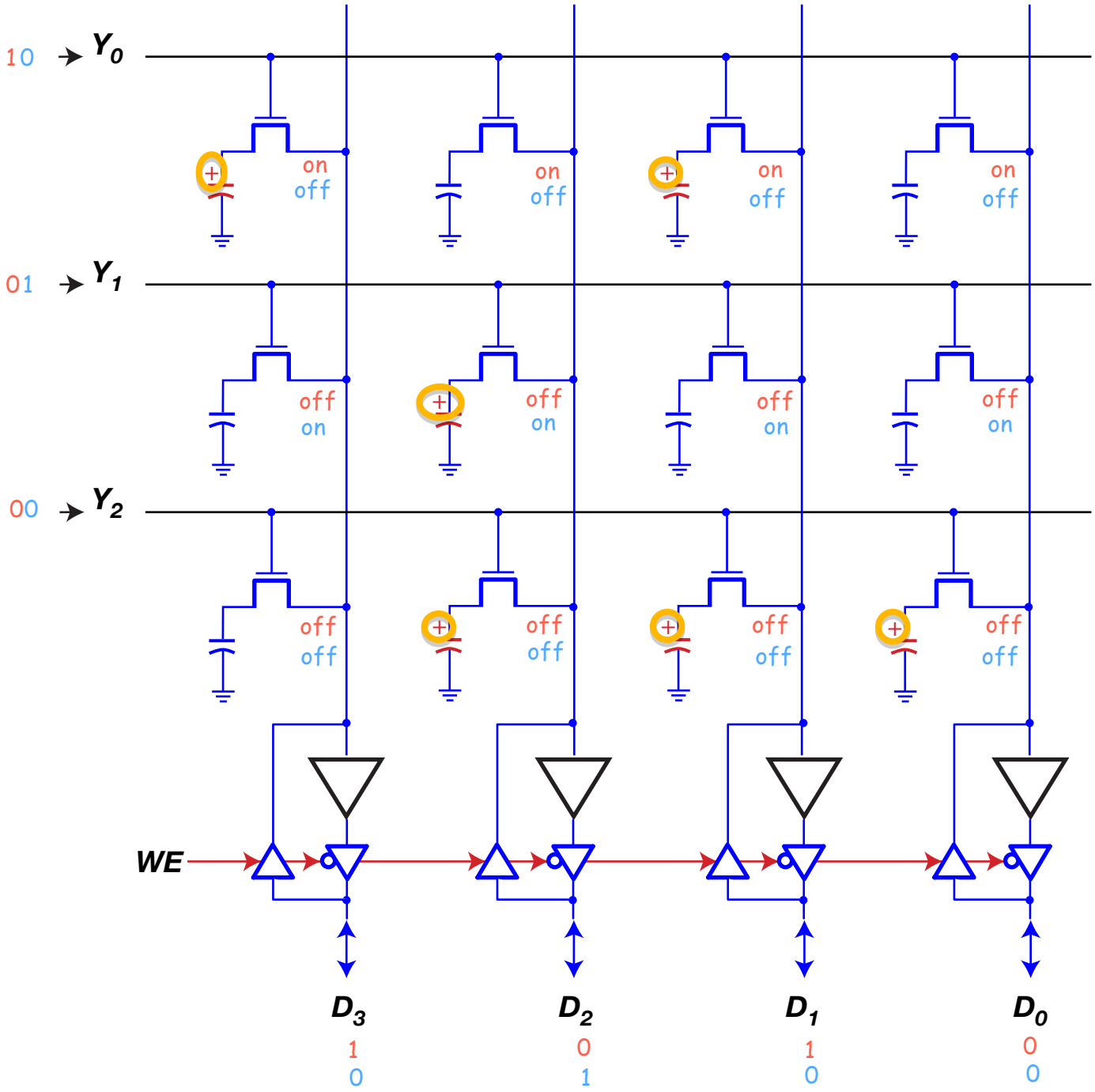
EPROM Encoder w/ Floating Gates



2 x 8 PLD AND Array Examples

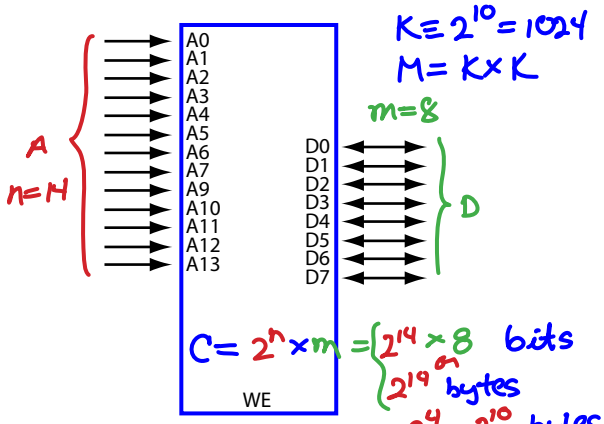


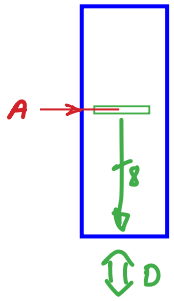
RAM NMOS Encoder



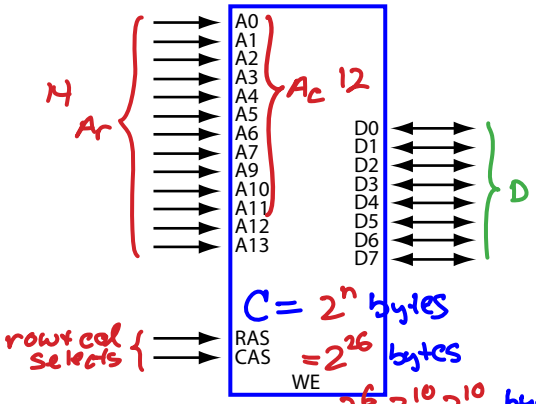
RAM Chip Addressing

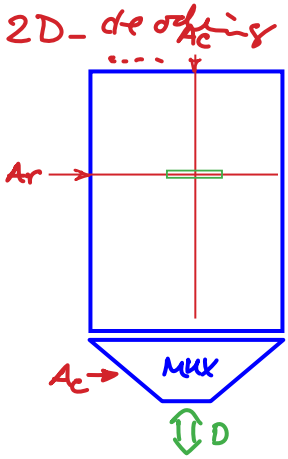
Straight Decoding:



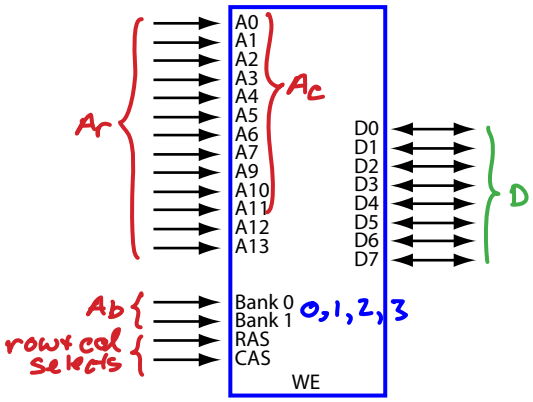


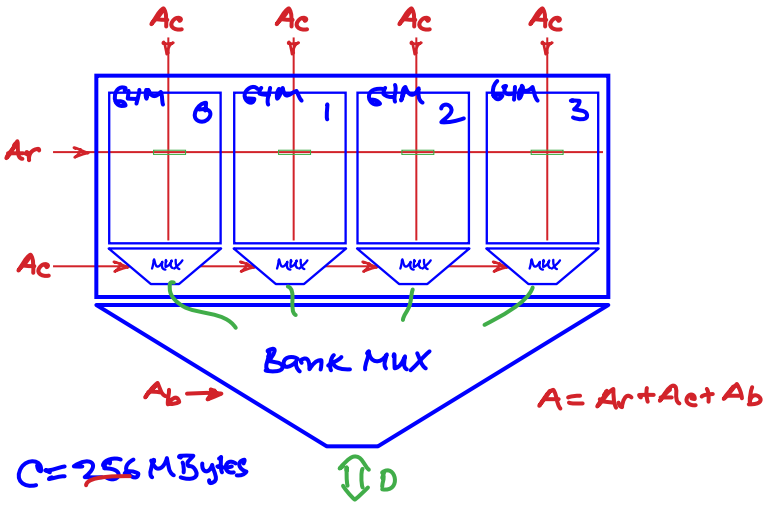
2D Decoding:



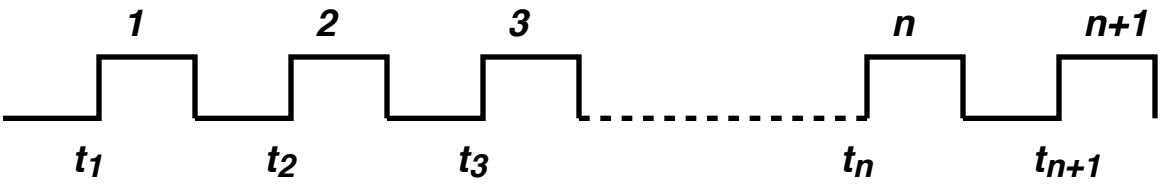


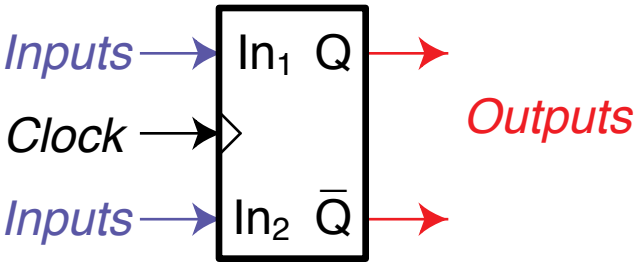
2D Decoding with Banks:

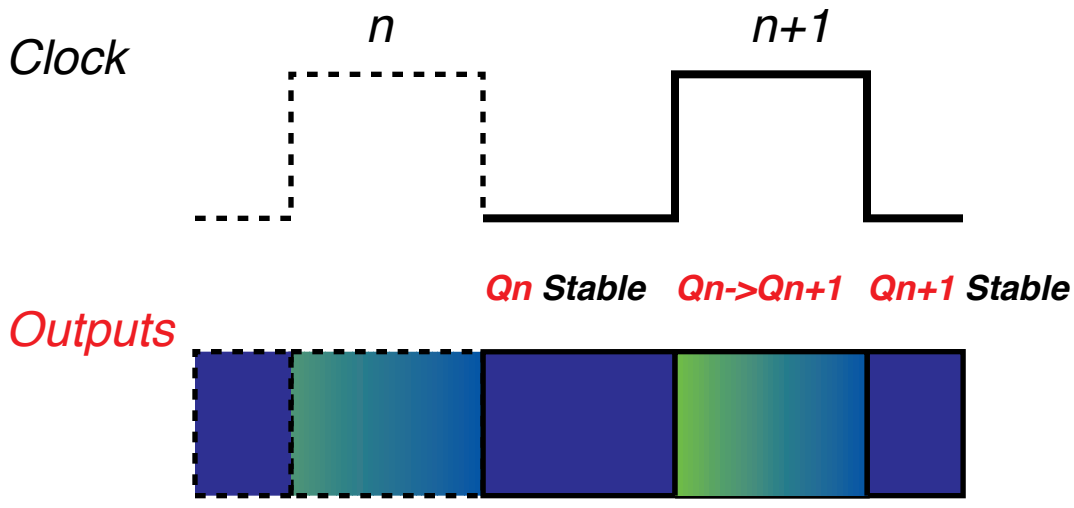




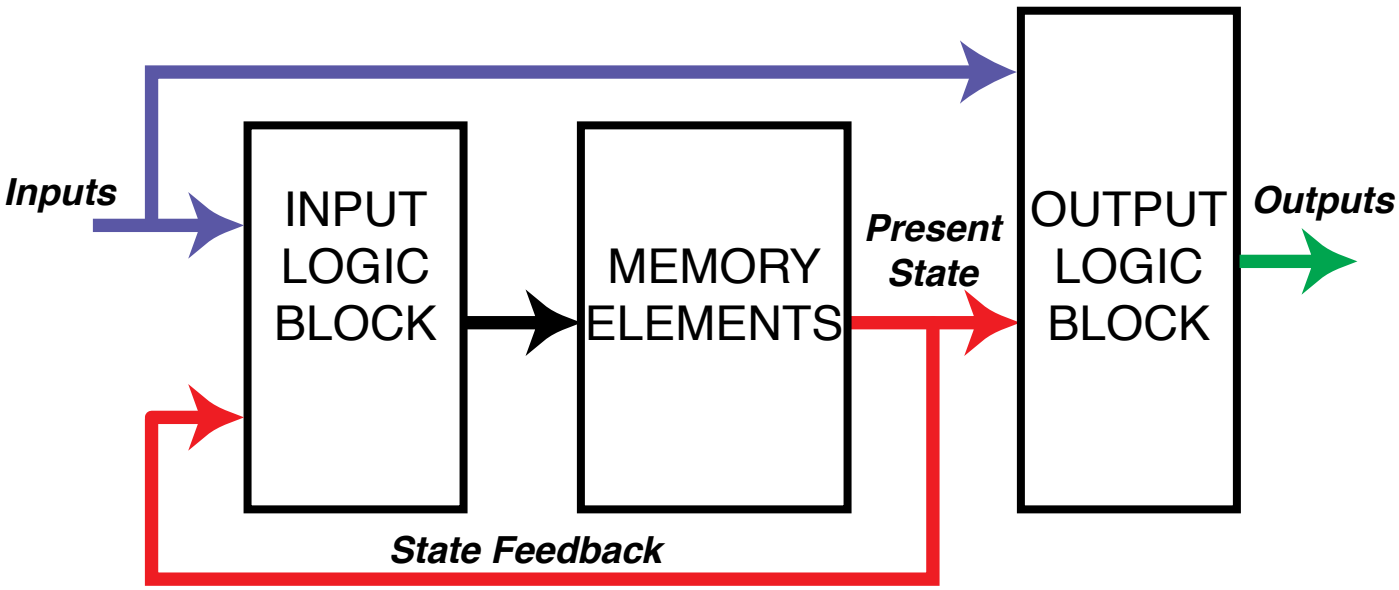
Clock WaveForm Illustrations





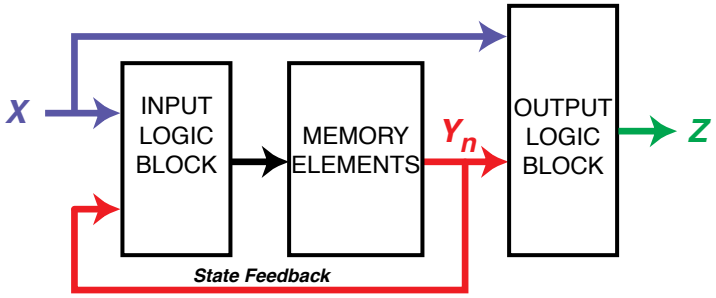


Sequential Logic Circuit Block Diagram

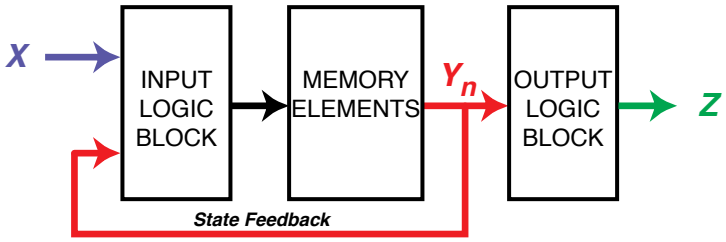


Sequential Logic Circuit Classes

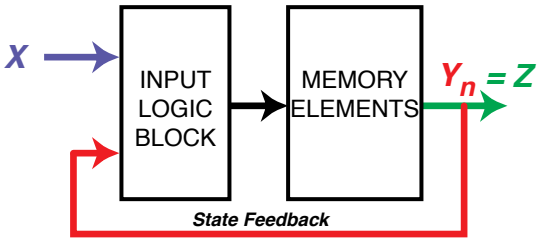
Class A SLC: Z = f(X, Y)



Class B SLC: Z = f(Y)

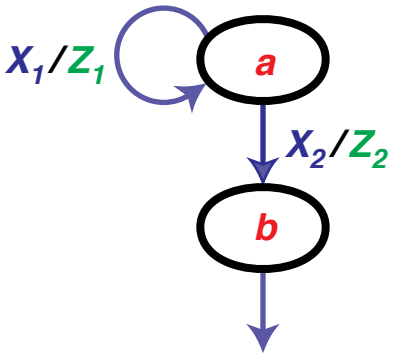


Class C SLC: Z = Y



State Diagrams and Tables

State Diagram:



Present State $Y_n = a$

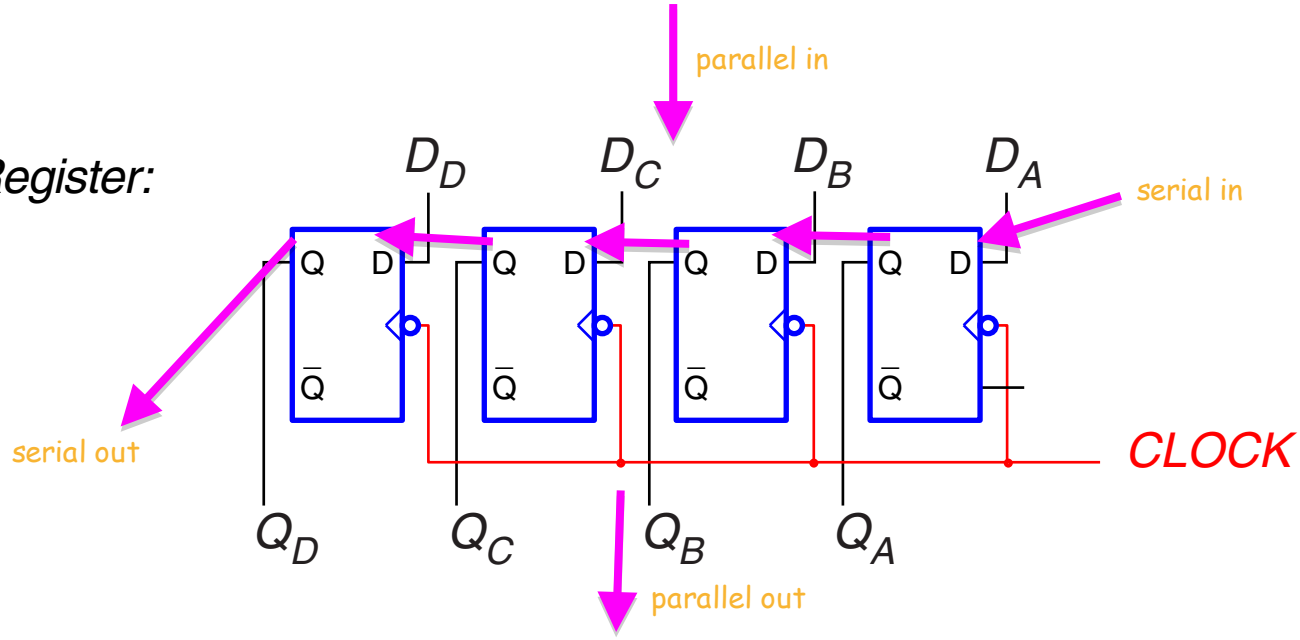
Next State $Y_{n+1} = b$

State Table:

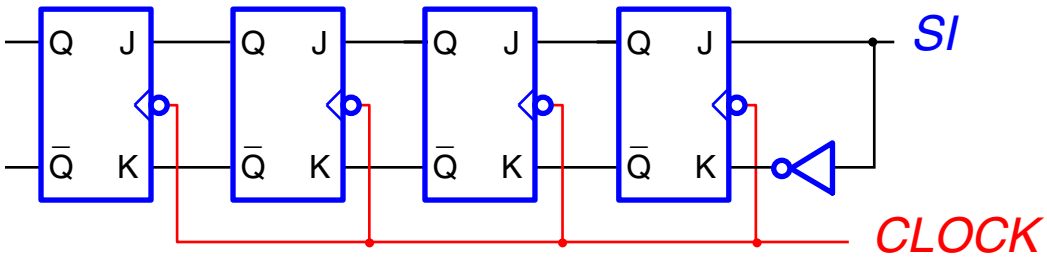
PS Y_n	In X	NS Y_{n+1}	Out Z	
a	X_1	a	Z_1	No Change
a	X_2	b	Z_2	Advance to b

Important Types of SLCs

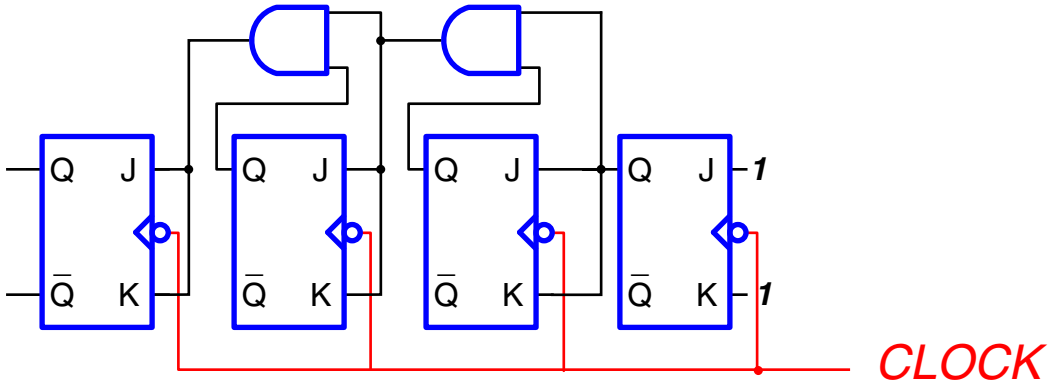
Data Register:



Shift Register:



Synchronous Counter:



Ripple Counter: