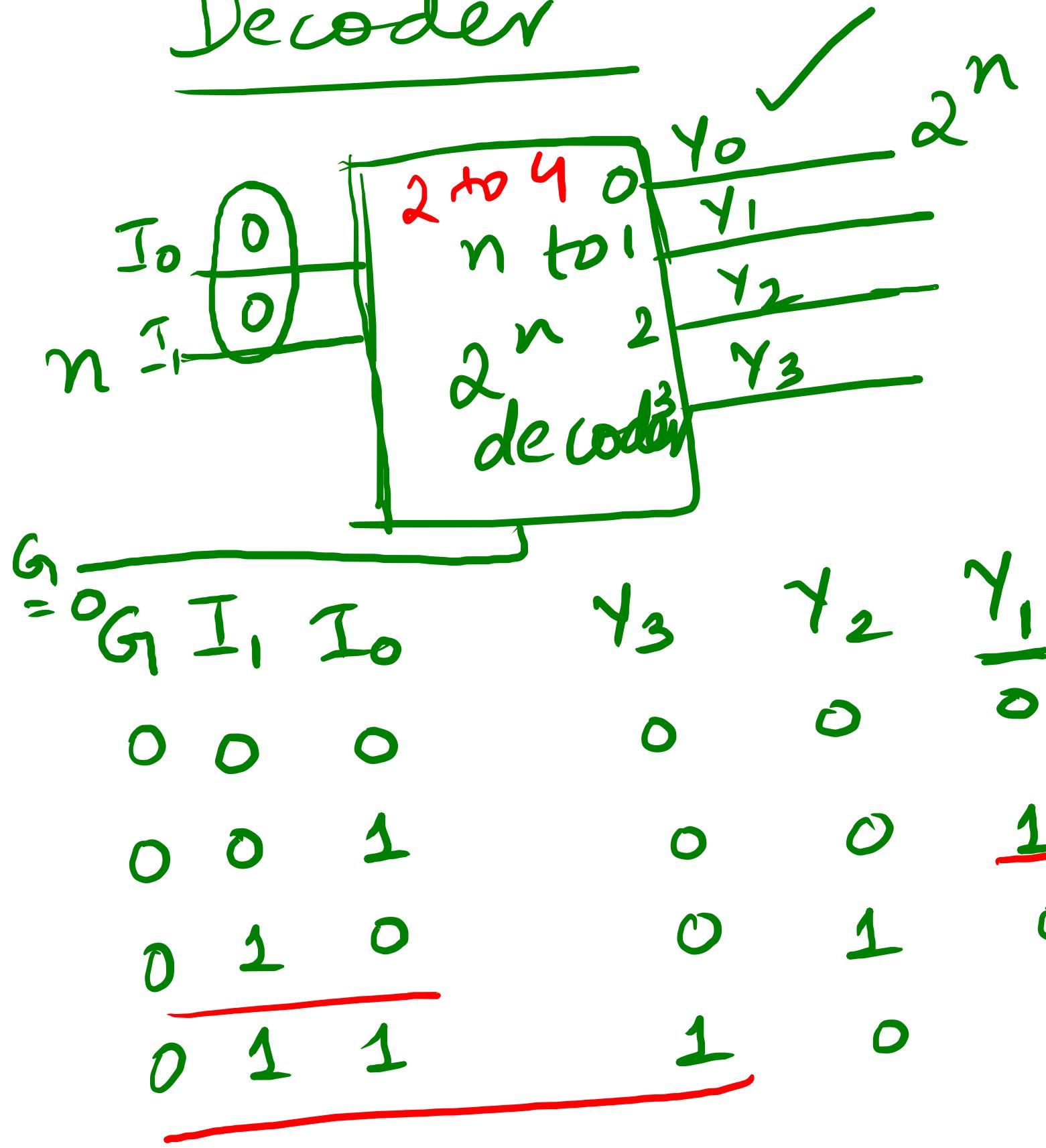


Decoder



G - enable signal

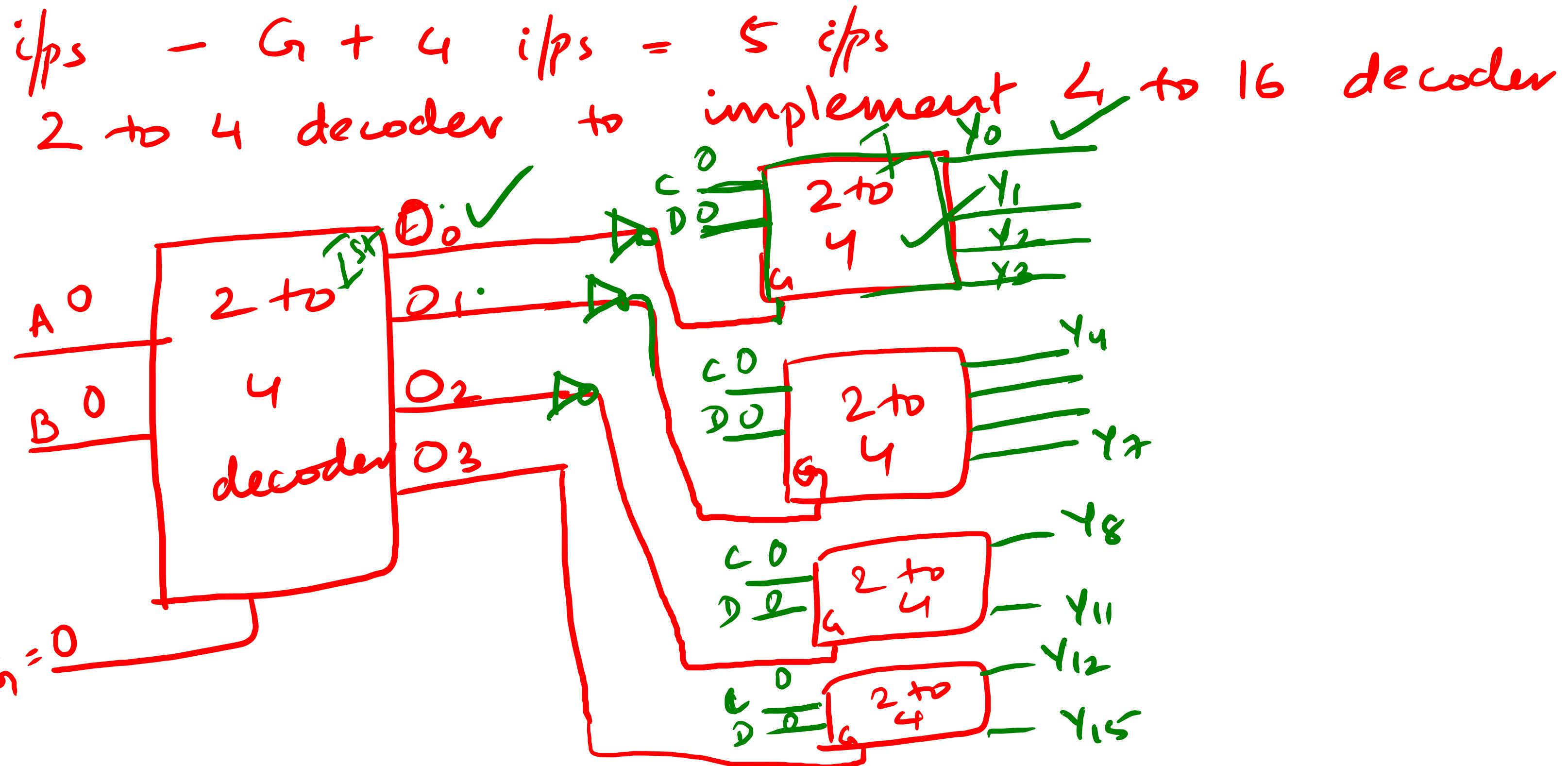
$G = 0$, the decoder
ckt is activated.

$G = 1$ 2 to 4 decoder

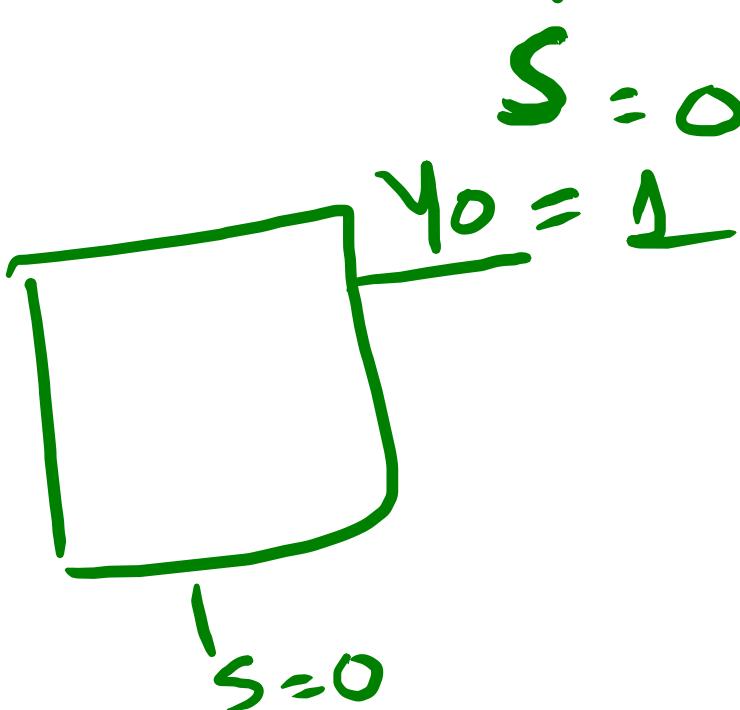
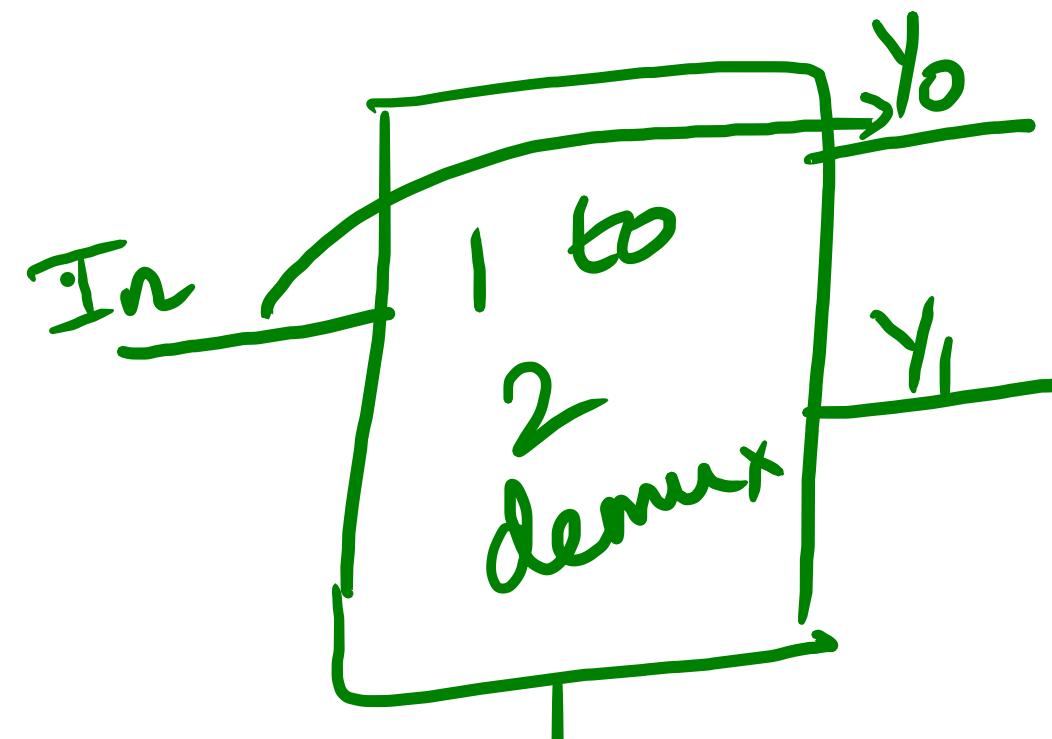
$$\begin{aligned}
 y_3 &= \overline{\bar{G} I_1 I_0} \\
 y_2 &= \bar{G} I_1 \bar{I}_0 \\
 y_1 &= \bar{G} \bar{I}_1 I_0 \\
 y_0 &= \bar{G} \bar{I}_1 \bar{I}_0
 \end{aligned}$$

3 inputs AND

4 to 16 decoder



Decoder a special case of a demux



Mux - many to one

demux - one to many

Suppose
 $In = 1$

$$y_0 = \bar{s} In$$

$$y_1 = s In$$

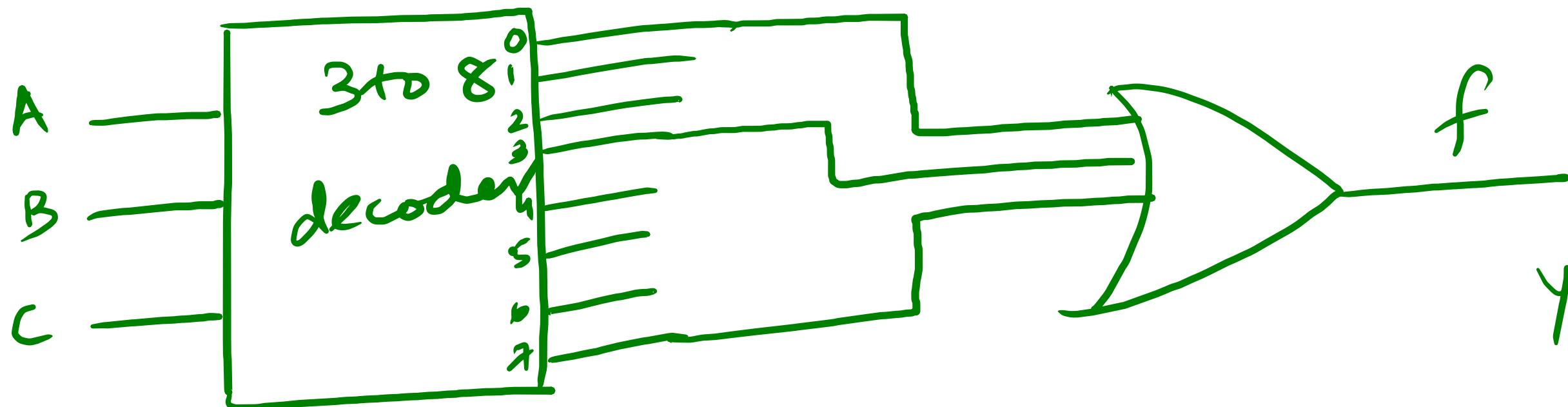
$$y_0 = \bar{s}, \quad y_1 = s$$

$$s = 0, \quad y_0 = 1$$

$$s = 1, \quad y_1 = 1$$

$$f(A, B, C) = \sum_m (0, 3, 7) = m_0 + m_3 + m_7$$

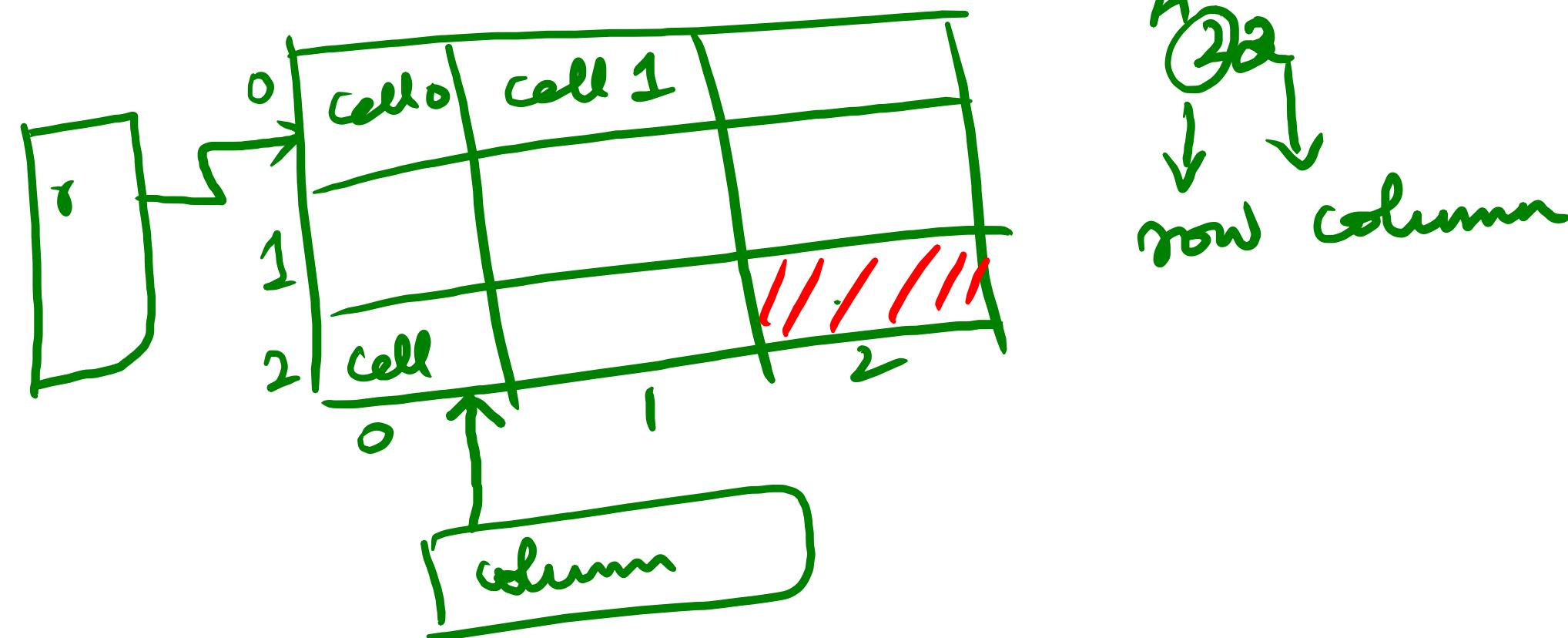
000	011	111
-----	-----	-----

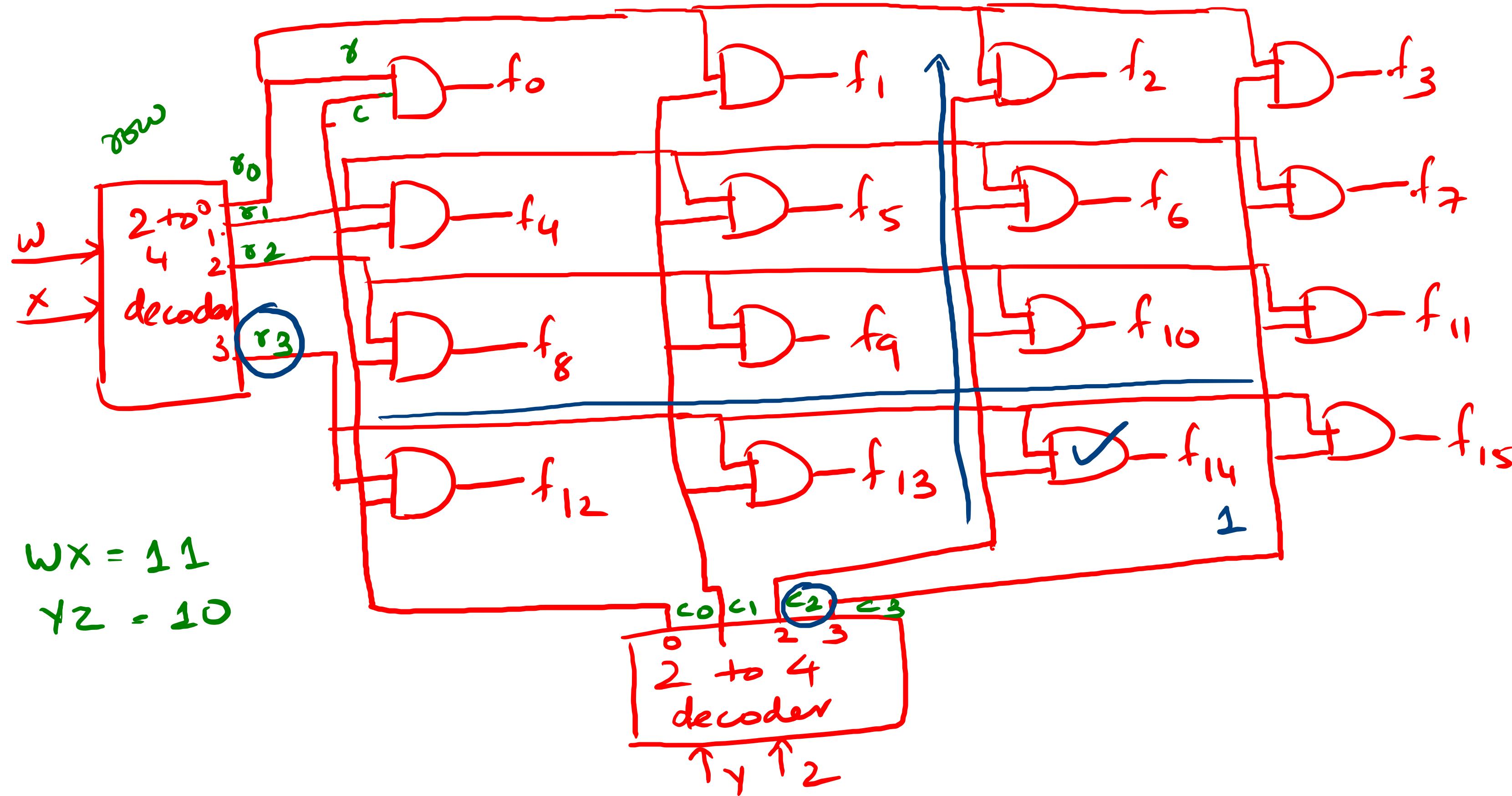


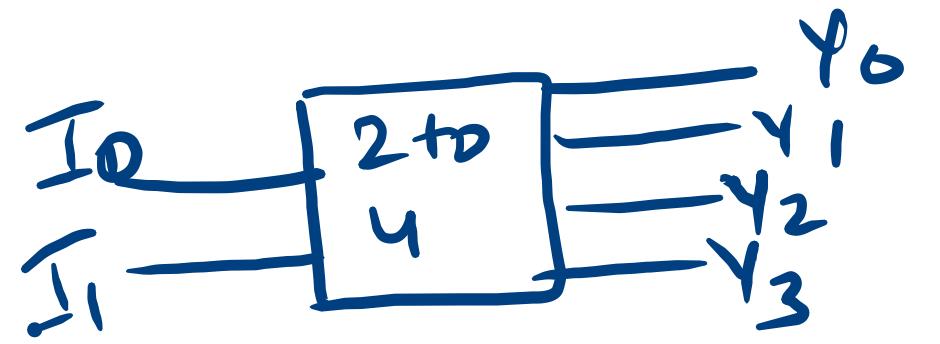
$$\gamma_{15} = \overline{I_3 I_2 I_1 I_0 G}$$

Special Decoder Design

4 to 16 decoder is implemented using 2 to 4 decoders
in a gate switching circuit

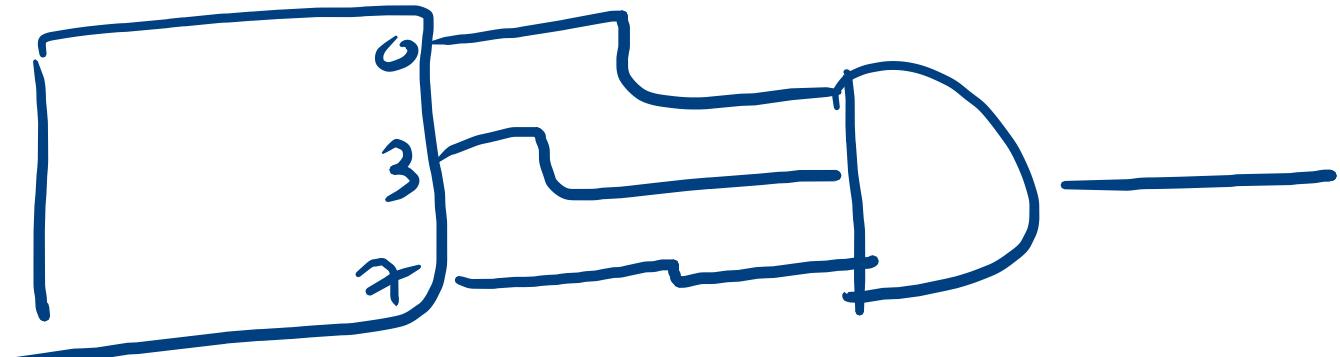




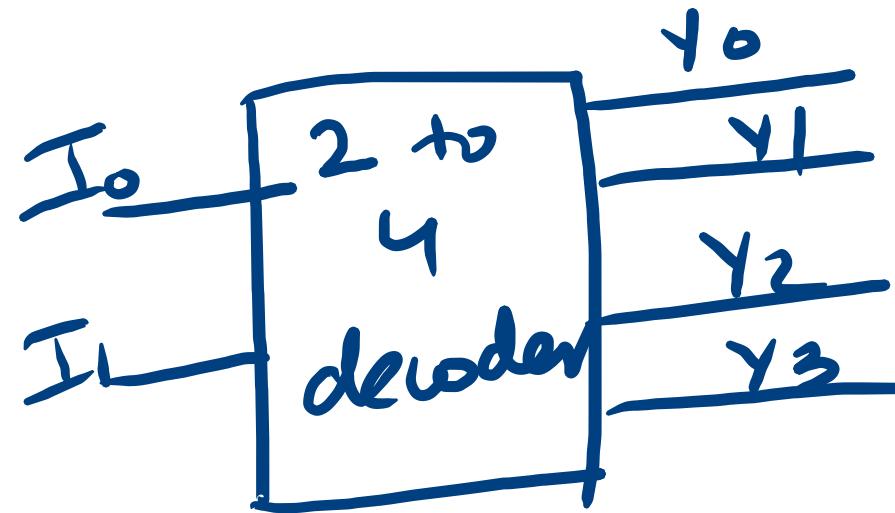


	0 0	y_3	y_2	y_1	y_0
0 1		1	1	1	0
1 0		1	0	0	1
1 1		0	1	1	1

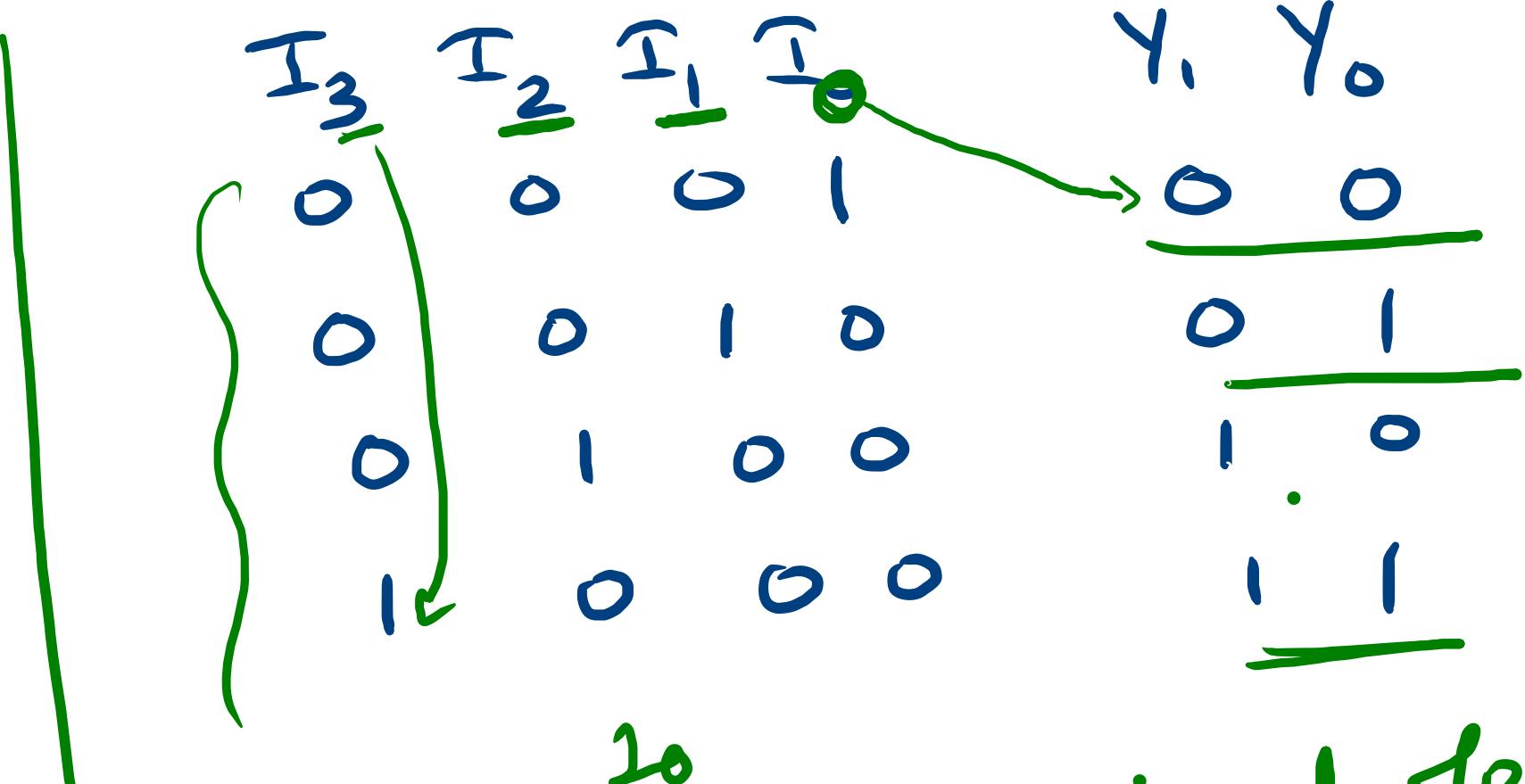
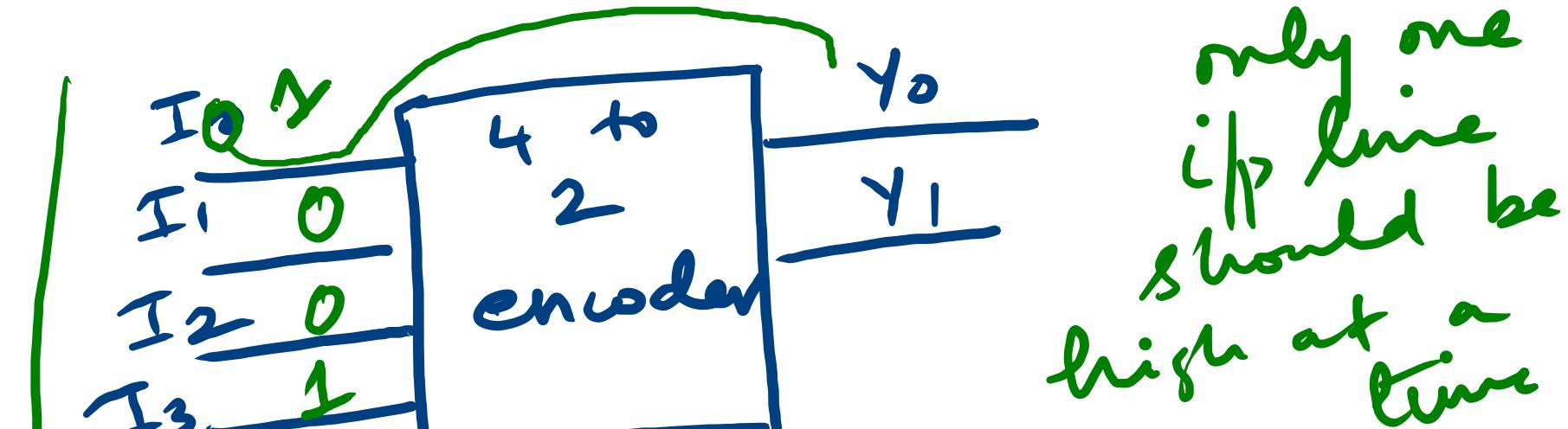
POS



Encoder



I_1 , I_0	Y_3	Y_2	Y_1	Y_0
0 0	0	0	0	1
0 1	0	0	1	0
1 0	0	1	0	0
1 1	1	0	0	0



Priority encoder

D_i, D_j , $i > j$

1

1

$\rightarrow q_P$ corresponding

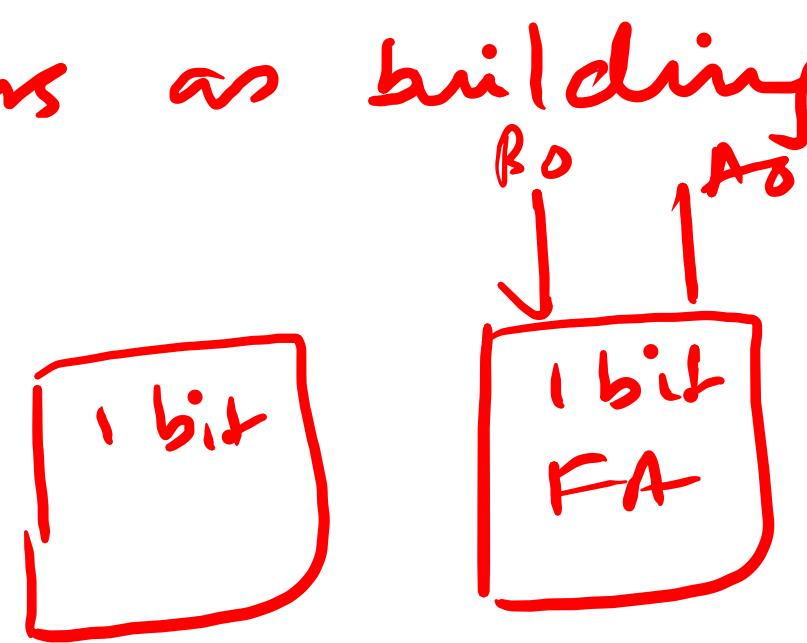
	D_3	D_2	D_1	D_0	y_1	y_0
1	1	x	x	x	1	1
0	0	1	x	x	1	0
0	0	0	1	x	0	1
0	0	0	0	1	0	0

D_i
 $D_3 >>> D_2, D_1, D_0$

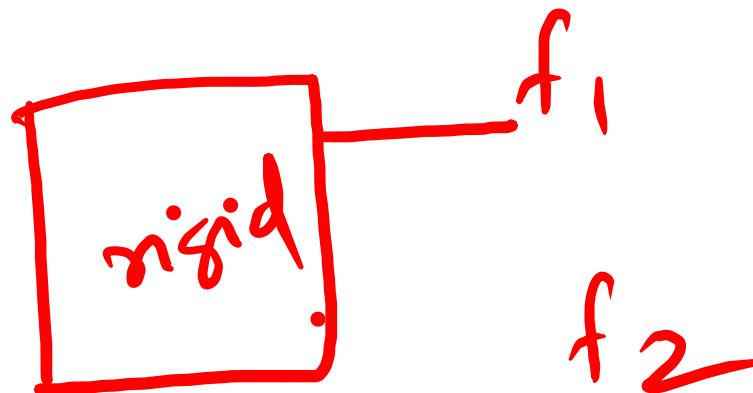
$$y_1 = D_3 + \bar{D}_3 D_2$$

$$y_0 = D_3 + \bar{D}_3 \bar{D}_2 D_1$$

Design a 4 bit adder / subtractor using full adders and multiplexers as building blocks.



Programmable Logic Devices (PLD)

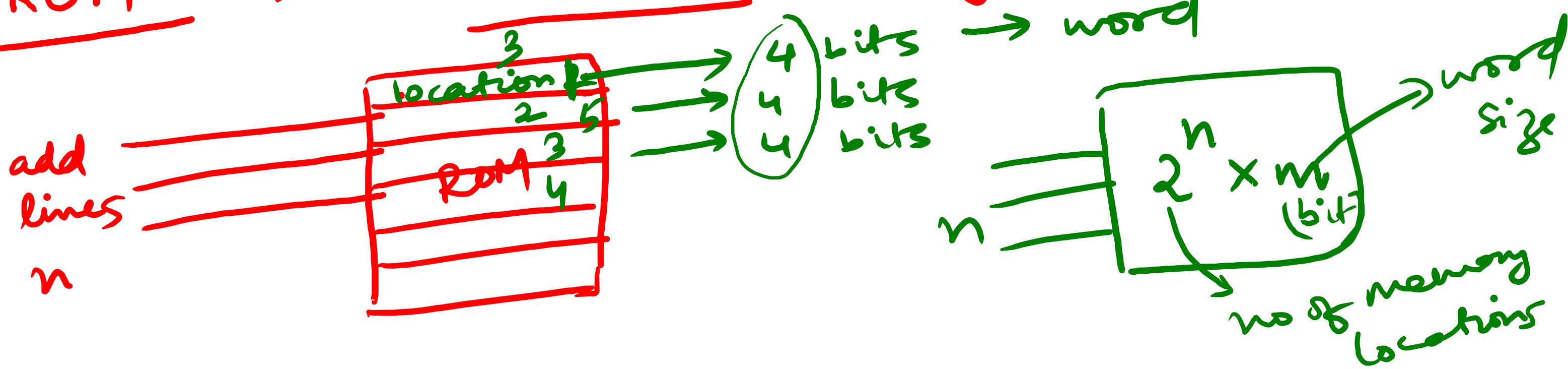


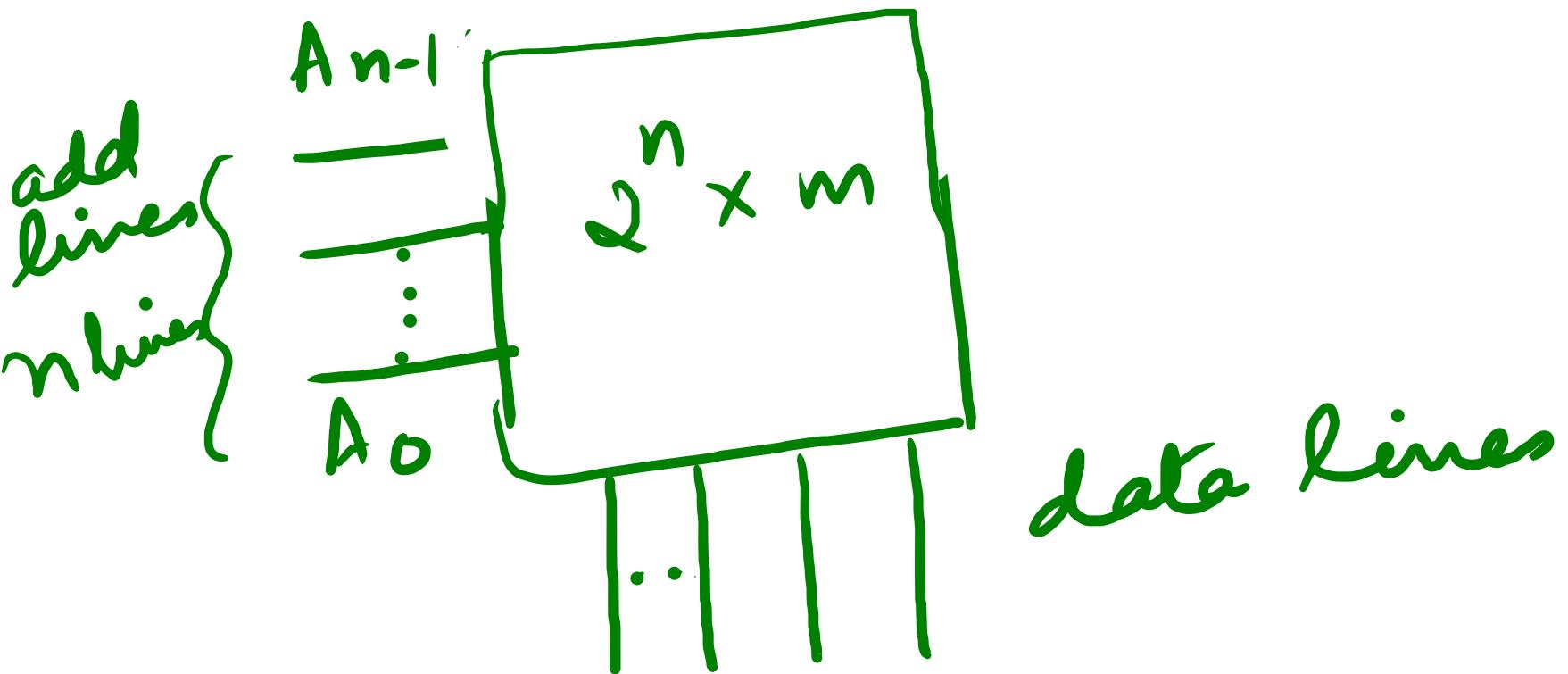
Cost
area

reconfigurable hardware

flexibility — PLD

ROM → non volatile memory





$D_m \dots D_1, D_0$

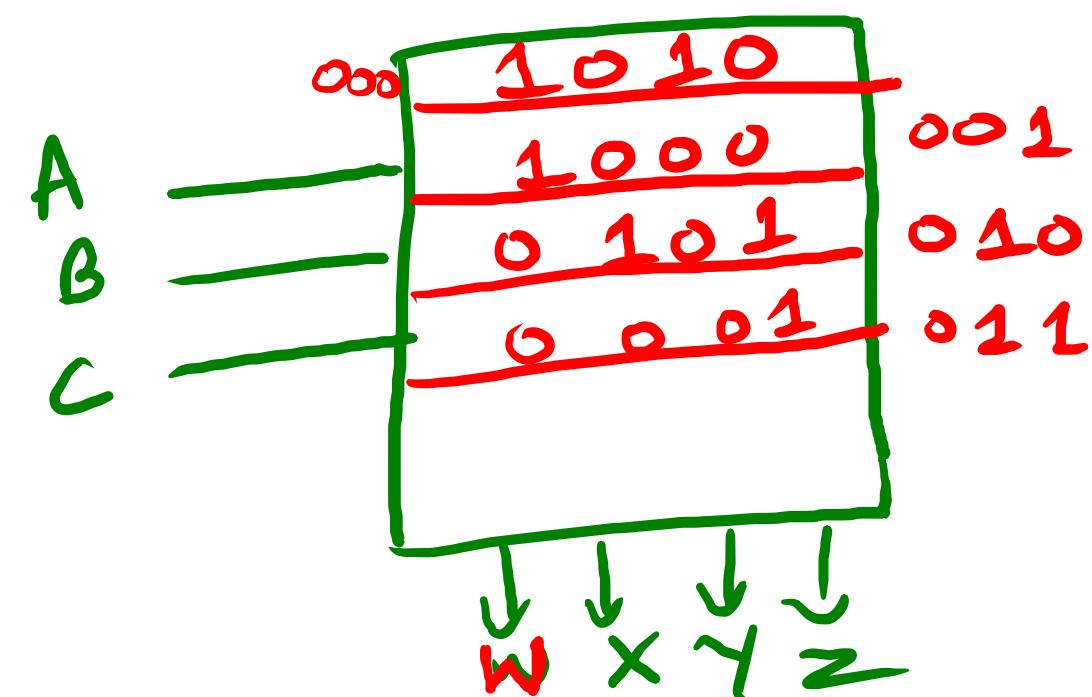


Table showing the addition of four lines of data. The input lines are labeled A, B, C and the output lines are labeled w, x, y, z . The data is represented as follows:

A	B	C	w x y z
0	0	0	1 0 1 0
0	0	1	1 0 0 0
0	1	0	0 1 0 1
0	1	1	0 0 0 1
1	0	0	1 1 0 0

Annotations in red:

- "add lines" and "n lines" are written near the top left.
- "data lines" is written below the input lines.
- "add lines" is written next to the first column of the table.
- "data" and "W = OR function" are written at the bottom right.

$$W = f \Sigma (m_0, m_1, m_4)$$

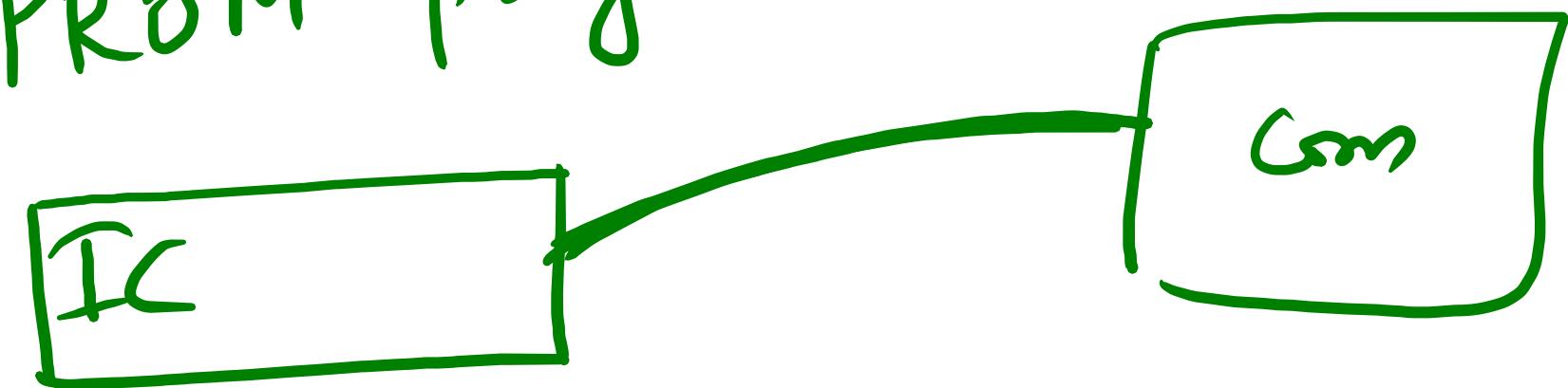
$$x = f \Sigma (m_2, m_4)$$

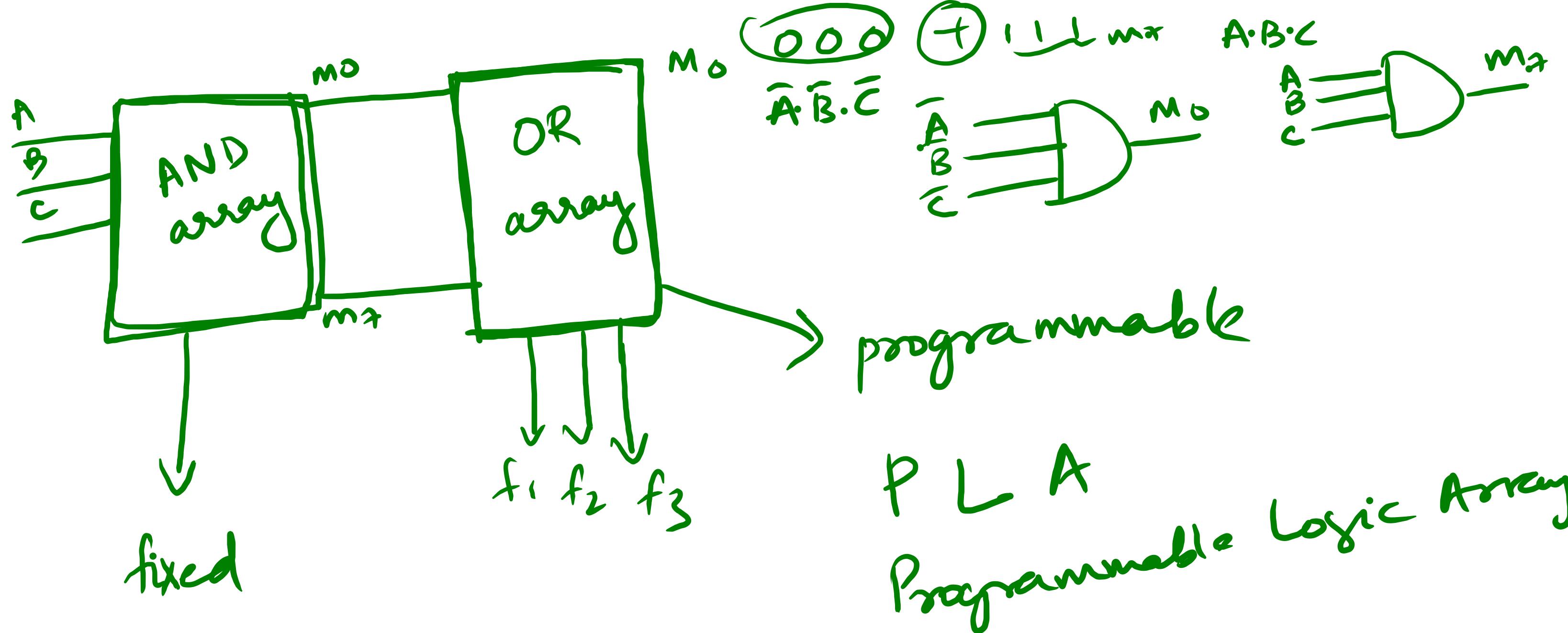
PROM

ROM whose contents can be changed according to the truthtable

A	X	X	X	X
B	X	X	X	X
C	X	X	X	X
	X	X	X	X
	X	X	X	X
	X	X	X	X
	X	X	X	X
	X	X	X	X

PROM programmer





PLA

Step1 - minimization of given functions

- AND array :

OR array :

$$f_1, f_2, f_3$$

using a, b, c

$$f_1(a, b, c) = \sum m(0, 2)$$

$$f_2 = \sum m(0, 3, 4)$$

$$f_3 = \sum m(3, 4, 5, 7)$$

	bc	00	01	11	10
a	0	0	1	1	0
	1	1	0	0	0

$$f_1 = \bar{a} \bar{c} \quad P_0$$

product

	bc	00	01	11	10
a	0	0	1	1	0
	1	0	0	0	0

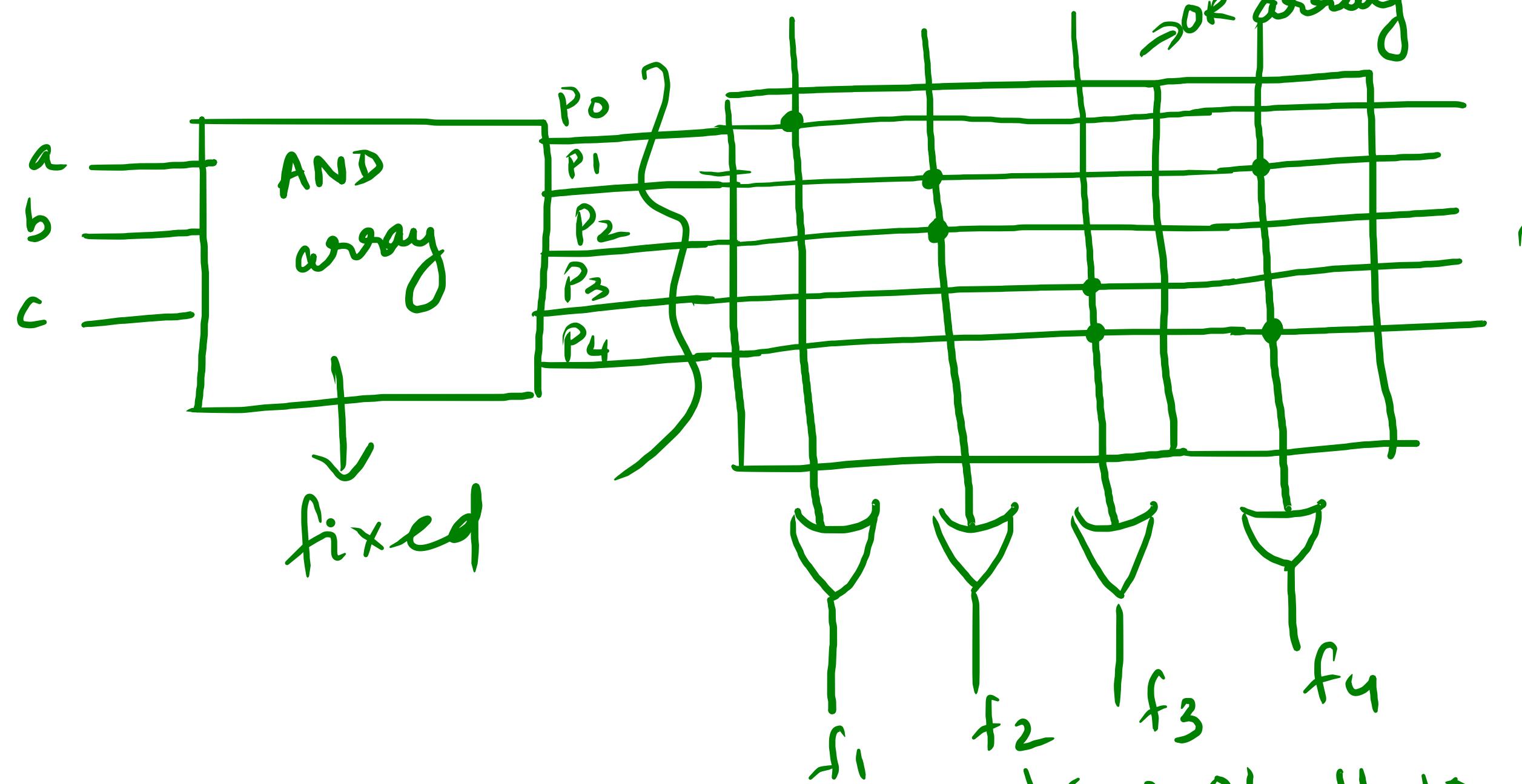
$$f_2 = \bar{b} \bar{c} + \bar{a} bc$$

$P_1 \quad P_2$

	bc	00	01	11	10
a	0	0	1	1	0
	1	0	0	0	0

$$f_3 = a \bar{b} + b c$$

$P_3 \quad P_4$



$$f_4(a, b, c) = \sum(0, 3, 4, 7)$$

$$f_4 = \overline{bc} + bc$$

$$P_1 + P_4$$

a	b	c	00	01	11	10
0	0	0	1	1	1	1

Product term sharing is allowed