

T-Table

	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0
0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1	0
2	0	0	0	0	0	1	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	1	0	0	0	0
5	0	0	1	0	0	0	0	0
6	0	1	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Encoder

(Address and Branch)
(Control Logic)

A_2	A_1	A_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Address lines

ONE switch on at a time! In Neumann Architecture

$$A_2 = I_4 + I_5 + I_6 + I_7$$

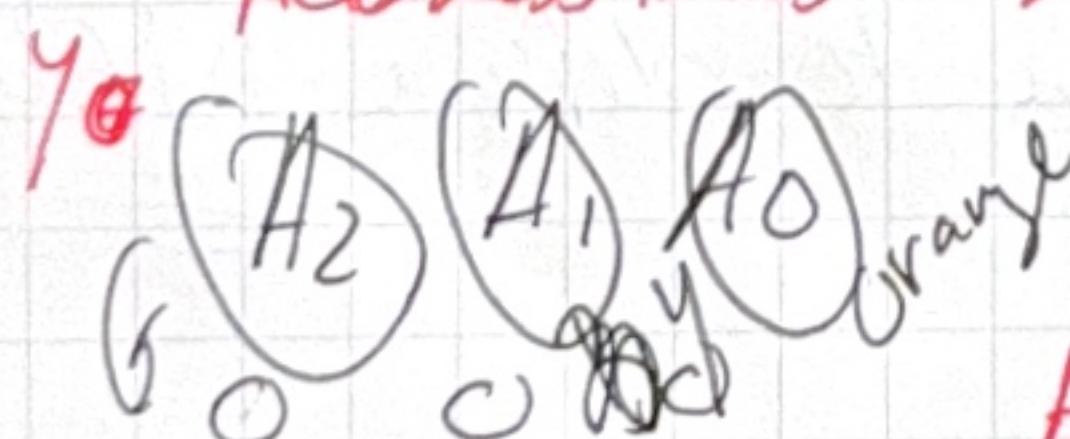
$$A_1 = I_2 + I_3 + I_6 + I_7$$

$$A_0 = I_1 + I_3 + I_5 + I_7$$

OR gate $I_n \rightarrow A_n$

$I_7 \rightarrow 0$

By inverter.



Address lines \rightarrow Row terms.

0 0 1

$$A_2' A_1' A_0 = y_0$$

0 1 0

$$A_2' A_1' A_0 = y_1$$

0 1 1

$$A_2' A_1' A_0 = y_2$$

1 0 0

$$A_2' A_1' A_0 = y_3$$

1 0 1

$$A_2' A_1' A_0 = y_4$$

1 1 0

$$A_2' A_1' A_0 = y_5$$

1 1 1

$$A_2' A_1' A_0 = y_6$$

1 1 1

$$A_2' A_1' A_0 = y_7$$

Exactly 7/8 LED's light
for any given
address.

Build "Matrix" (1.e. $2^n \times 2^n$)
through Horizontal/Vertical
instruction

arrangement.
arrangement.

Matrix is (8×4)
(Horiz. limit.)

$(Y_0 \rightarrow Y_7)$ Horizontally

$(Q_3 \rightarrow Q_0)$ Vertically.

$Q_T = 1$ for Y_n	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	S_2	S_1	S_0	Output
	0	0	0	0	0	0	0	0	0	0	0	CH_0
	0	0	0	1	0	0	0	0	0	0	1	CH_1
	0	0	1	0	0	0	1	0	0	1	0	CH_2
	0	0	1	1	0	0	1	1	0	1	1	CH_3
	0	1	0	0	0	0	0	0	1	0	0	CH_4
	0	1	0	0	1	0	1	0	1	0	1	CH_5
	0	1	1	0	0	1	1	0	1	1	0	CH_6
	0	1	1	1	1	1	1	1	1	1	1	CH_7

By changing which intersections are connected,

I can change the Row's "program" [i.e., the control words it outputs for each Address (A1)]

Thus, ~~program~~ programme in the sense of

Rewriting or rearranging the jumper wires to define NEW micro-instructions.

Bus Enable = $Y \cdot T$.