Enloder (Address and Branch Control Logic)

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

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 ! Von 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 $I_n \rightarrow A_n$.

$y_7 \rightarrow 0$

By Inversion.

Address lines → Row terms.

y_0

A_2	A_1	A_0	
0	0	0	$A_2' A_1' A_0' = y_0$
0	0	1	$A_2' A_1 A_0 = y_1$
0	1	0	$A_2' A_1 A_0' = y_2$
0	1	1	$A_2' A_1 A_0 = y_3$
1	0	0	$A_2 A_1' A_0' = y_4$
1	0	1	$A_2 A_1' A_0 = y_5$
1	1	0	$A_2 A_1 A_0' = y_6$
1	1	1	$A_2 A_1 A_0 = y_7$

Exactly 1/8 LED's lit for any given Address.

Build "Matrix" (i.e, Mat/Hor: Z.) through Horizontal/Vertical Instruction arrangement.

Matrix is (8 × 4)
(Horiz × vert.)

$(Y_0 \rightarrow Y_7)$ Horizontally

$(Q_3 \rightarrow Q_0)$ Vertically

$Q_T = 1$ for $\forall n \in Y_n$.

Input

	Q_3 (T) BUS ENABLE	Q_2	Q_1	Q_0	$S_2 S_1 S_0$ output	
Y_0	0	0	0	0	000	CH_0
Y_1	0	0	0	1	001	CH_1
Y_2	0	0	1	0	010	CH_2
Y_3	0	0	1	1	011	CH_3
Y_4	0	1	0	0	100	CH_4
Y_5	0	1	0	1	101	CH_5
Y_6	0	1	1	0	110	CH_6
Y_7	0	1	1	1	111	CH_7

iff $\exists$ connection between Y_n and Q_k;
then, $Y_n = 1$, $Q_k = 0$.

iff $\nexists$ connection,
then, $Q_k = 1$.

By changing which intersections are connected, I can change the ROM's "program" [i.e, the control words it outputs for each Address (A)].

Thus, ~~Progra~~ Programmable in the sense of rewiring or rearranging the jumper wires to define <u>New</u> micro-instructions.

Bus Enable $= Y \cdot T$.