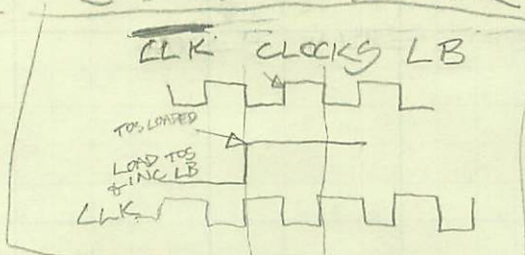


CLR (TOS)
LOADS IT
TO 11 (TOS)

MCLR

CLR
(TOS
BOS) CLR
LB

"LB" IS LOWER BITS, BOTTOM 2 BITS OF STACK ADDR.

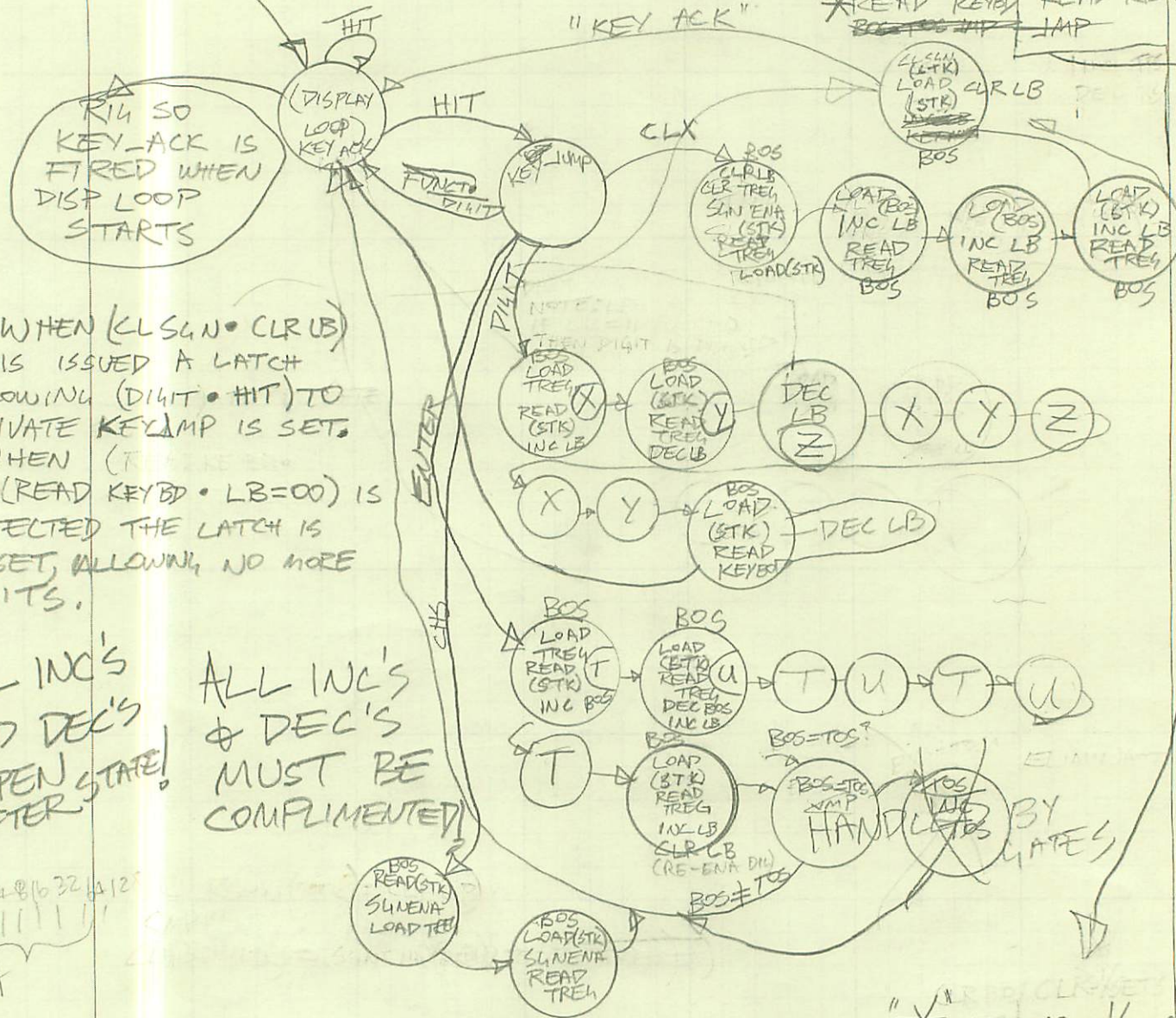


BOS SEL
CLR BTOS

SELECTS BOS AS STR
CLR BOT/TOS

* DEC STK
* INC STK
KEY JMP
~~KEY ACK~~
* 1/2 LB
DEC LB
CLR LB
LOAD (STK)
SYNENA (STK)
READ (STK)
CLR TREQ
LOAD TREQ
* READ TREQ
* READ KEY
~~READ KEY~~

DECREMENT STK PTR
INCREMENT STK PTR
GATES KEYPNT ^{ON BUS}
RESET KEYBD
INX LOWER BITS
(HAPPENS AFTER STATE!)
CLR LOWER BITS
LOADS MEM @ ⁷⁰⁵ BUS / LB
ENA SYN BIT, ASSUME
~~STK~~ LB = 0
READS ~~BC~~ ⁷⁰⁵ / LB
} ^{JUMP} TREG MANIPULATION
D READ ICB ONTD BUS
~~JUMP~~



WHEN (CLSN • CLRUB)
IS ISSUED A LATCH

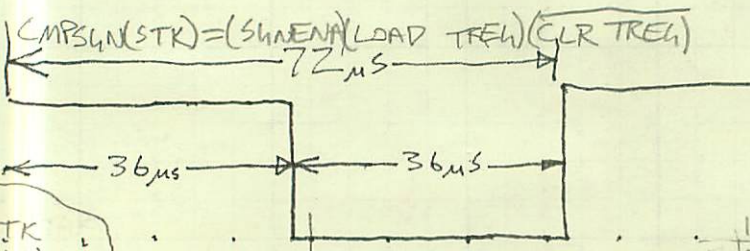
LOWING (DIGIT • HIT) TO
IVATE KEYAMP IS SET.

WHEN (T
(READ KEYBD • LB=00) IS
TECTED THE LATCH IS
SET, ALLOWING NO MORE
ITS.

INC'S
DEC'S
PEN STATE
STER

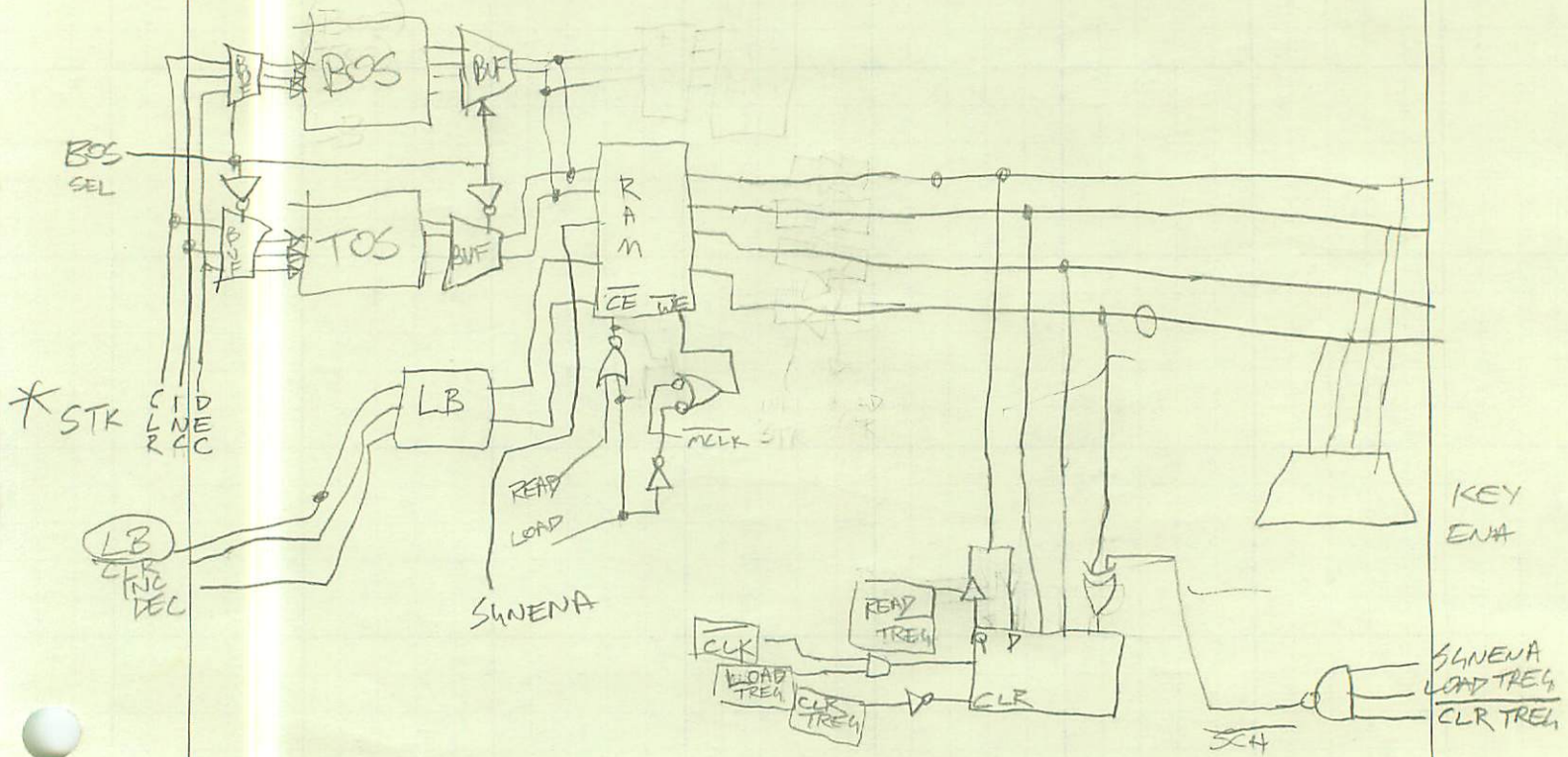
ALL INC'S
& DEC'S
MUST BE
COMPLIMENTED

 MEANS $\frac{1}{2}$ CLK
OUT OF PHASE

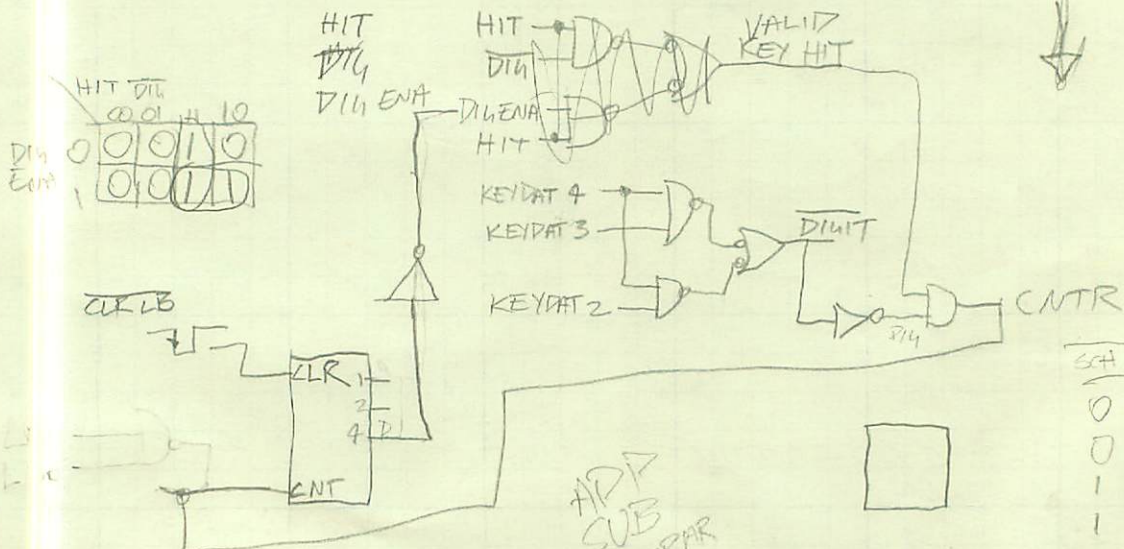
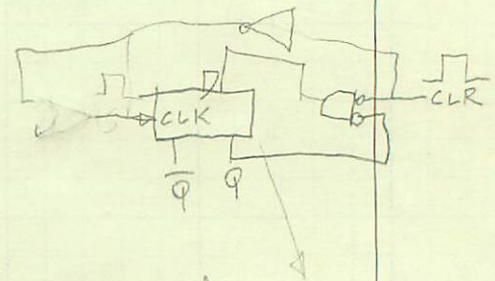


16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISP SQU	FN INCEL	CLR REG	CLR REG	CLR REG	REG SEL	KEY LMP	PER LB	INC LB	REG TRF	LD TRF	REG STZ	LD STK	SHW END	PER STR	INC STR	NEXT ADR

BUFFER KBD



HIT	DIGIT	(DIGIT ENA)
0	X	X
1	1	X
1	8	0
1	8	1

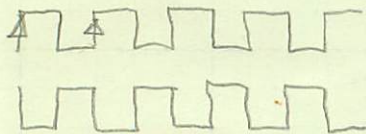


ADP
SUB
COMPAR
COMPL

0	0	9
0	1	0
1	0	0
1	1	0

Sci		
0	0	1
0	1	0
1	0	0
1	1	1

[illegible]



INC STK ✓
 DEC STK ✓
 SLN ENA ✓
 LOAD STK ✓
 READ STK ✓
LOAD TREG ✓
READ TREG ✓
 INC LB ✓
 DEC LB ✓
 BOS SEL ✓
 CLR LB ✓
~~READ KBD~~ ✓
 CLR TREG ✓
 CLR BTOS ✓
 KEY JMP* ✓
 FUNCT JMP ✓
 0 = NORM, 1 = ALT

F1 } ALU
 F0 }
~~READ KBD~~
 KEYBD ACK
 READ ALU
~~LOAD~~ ALU

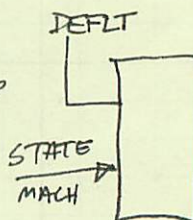
MISPLACED OUTPUT

~~CARRY~~
~~BIT~~
 CARRY
 DIGIT
 FUNCT

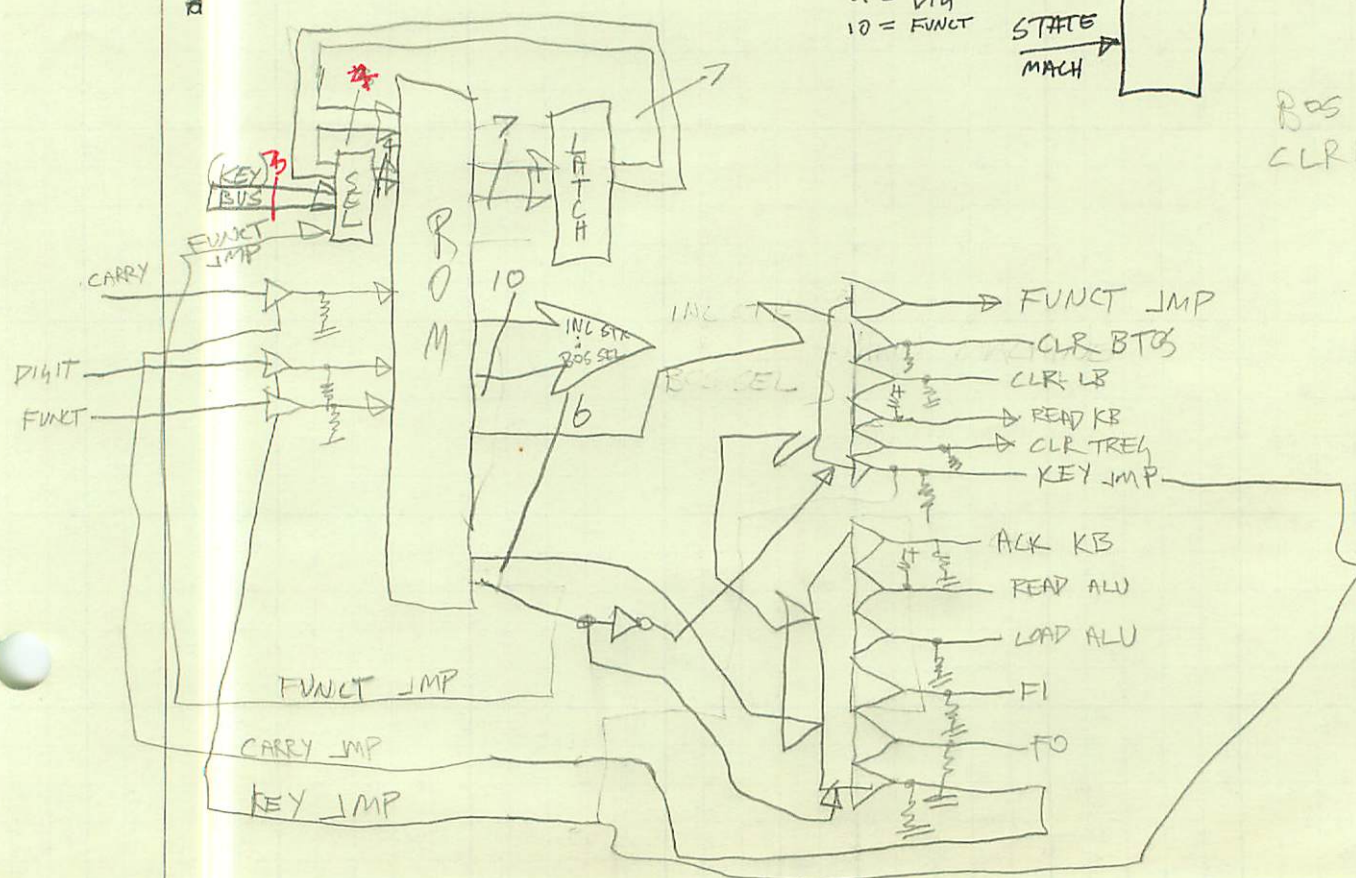
INPUTS

CARRY JMP
 (ACK KEYBD?)
~~ACK KL~~
~~READ ALU~~
 LOAD ALU

* 00 = NO JUMP
 01 = DIG
 10 = FUNCT



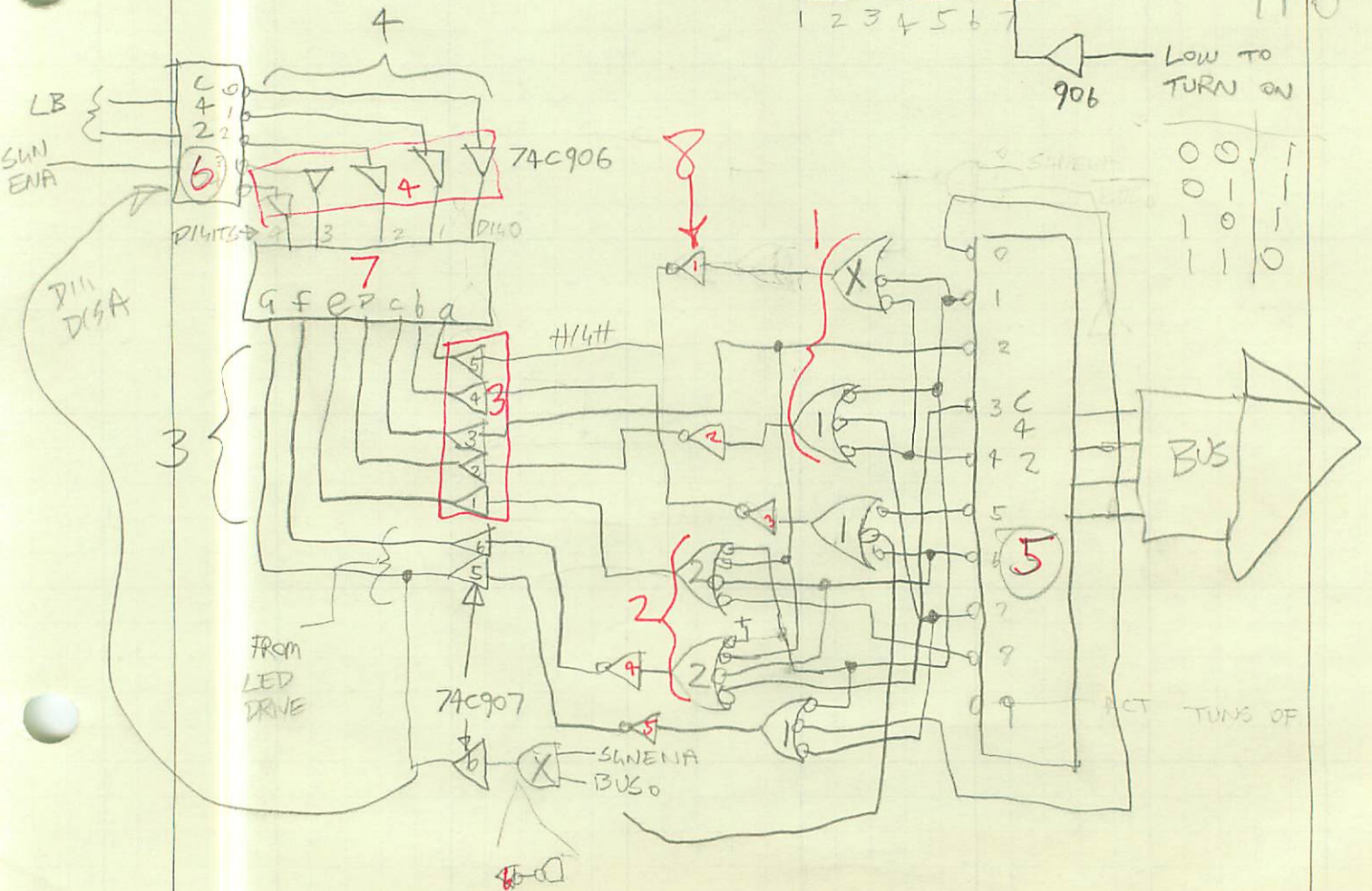
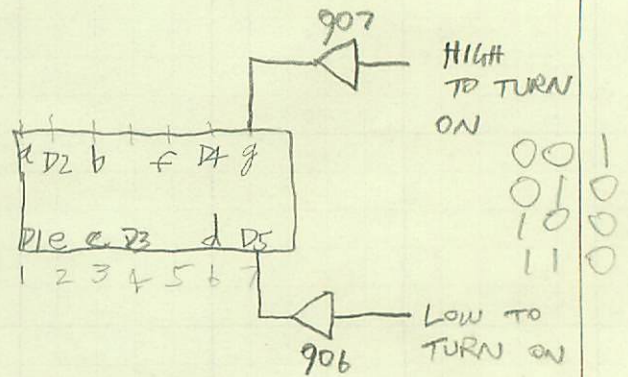
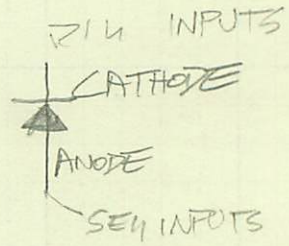
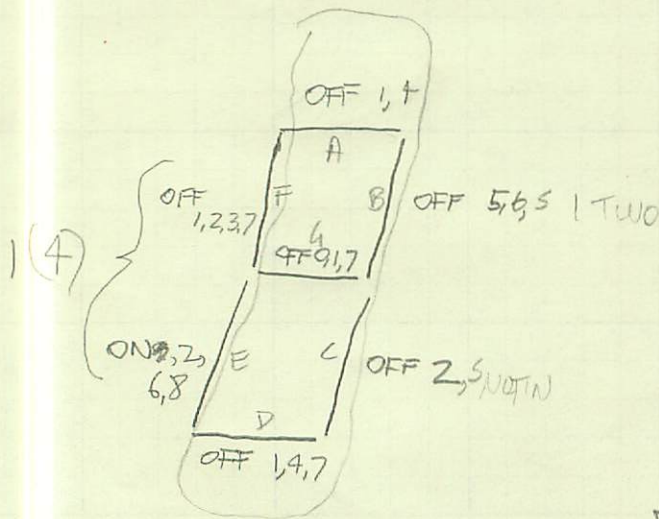
BOS SEL
CLR LB

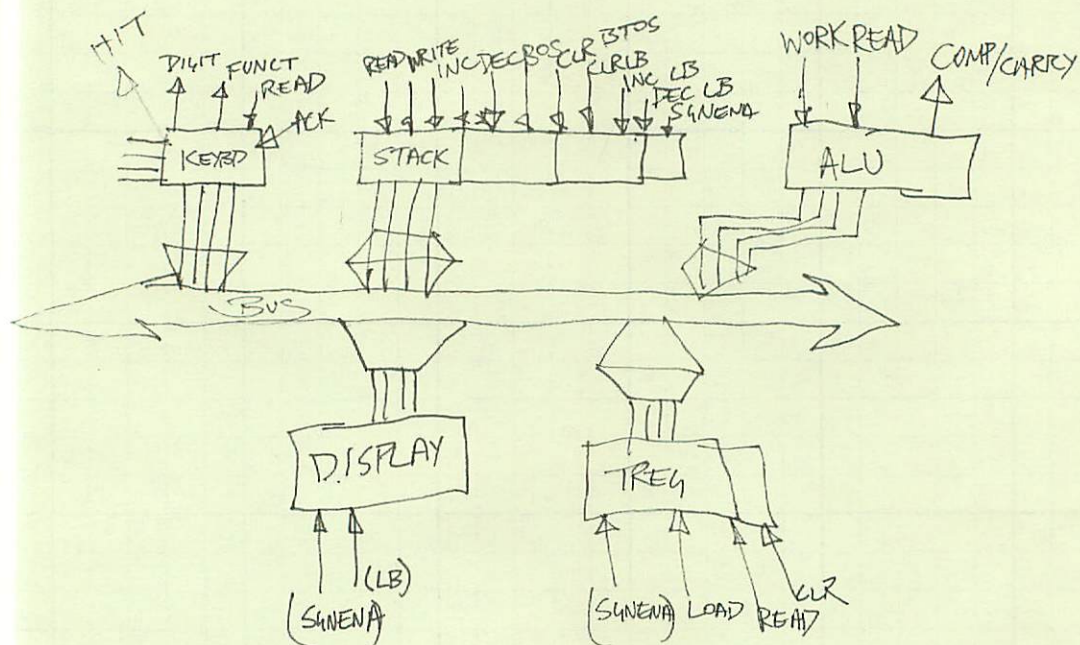


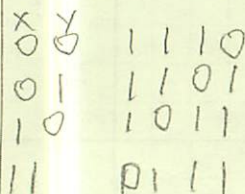
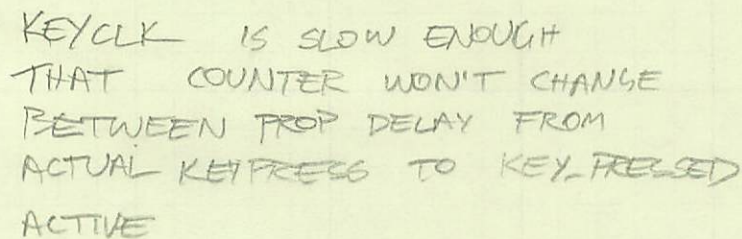
SWENNA
 0 0 0
 0 1 0

0 0
 0 1
 0 0
 1 1
 0

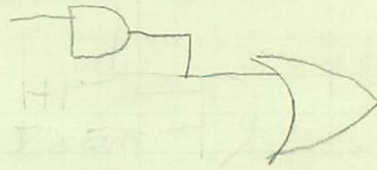
13



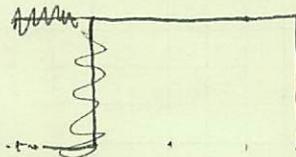




IF FUNCT = CLX THEN
~~FUNC~~
 (BOS) $\leftarrow$ 0

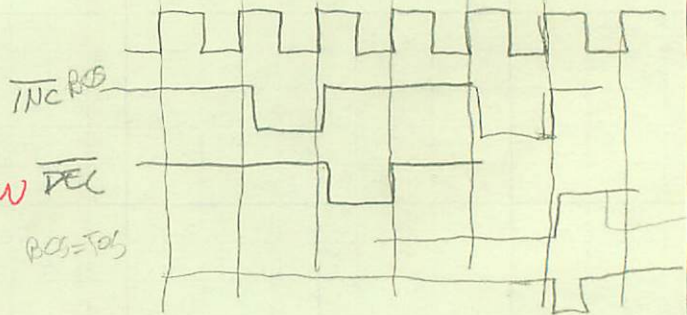


 IF FUNCT = KEY
 (BOS) $\leftarrow$ KEY



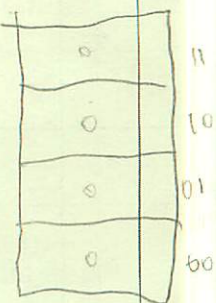
IF FUNCT = CHS
~~SRE4 (BOS) $\leftarrow$ SRE4 (BOS)~~

IF FUNCT = ENT
 (BOS + 1) $\leftarrow$ (BOS)
 INC BOS
 IF BOS = TOS THEN
 INC TOS



IF FUNCT = ADD | SUB THEN
 DEC BOS
 AR4 1 $\leftarrow$ (BOS + 1)
~~DEC BOS~~
 AR4 2 $\leftarrow$ (BOS)
 (BOS) $\leftarrow$ RESULT

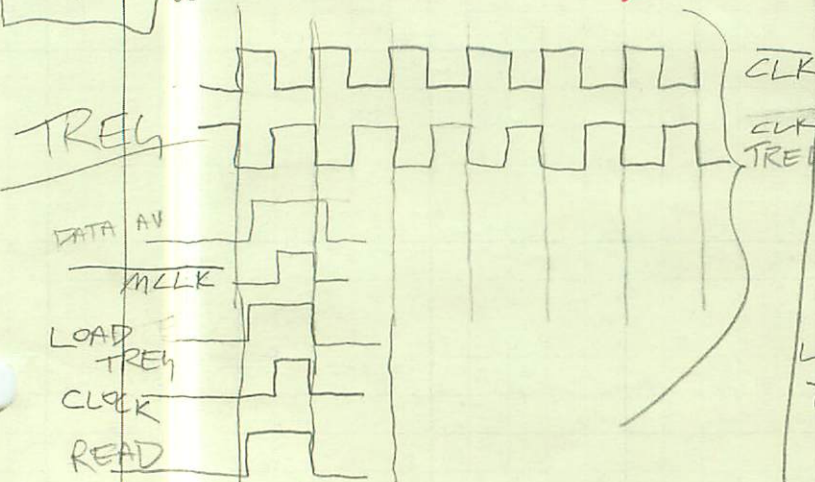
(TOS - 1) $\leftarrow$ TOS
 DEC TOS
 0 0 1



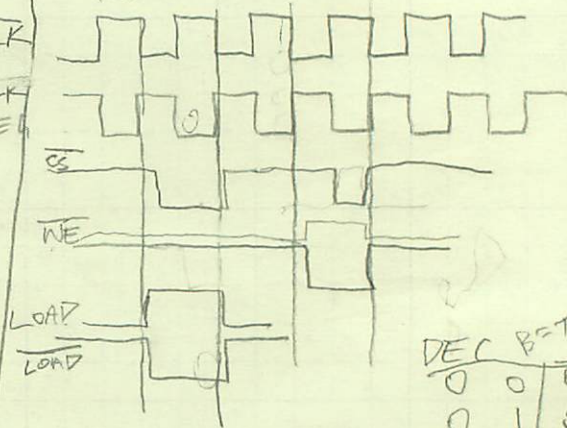
DEC TOS
~~(TOS - 1) $\leftarrow$ (TOS)~~
 (TOS) $\leftarrow$ (TOS + 1)

221

NAND			NOR		
0	0	1	0	0	1
0	0	1	0	1	0
1	0	1	1	0	0
1	1	0	1	1	0



RAM



DEC	BET	OUT
0	0	0
0	1	0
1	0	0
1	1	1