TEST1 ... Show ALL work BY _____

1. In an 8-bit machine, $D7_H$ is added to 61_H. Give the contents of the flags C, S, V, Z and PE (parity even) after this operation.

2. The micro code to the right is as in Ch 8

```
     ORG 36
NOP  Z  JMP  FETCH
NOP  I  CALL INDRCT
BRTPC U  JMP  FETCH
```

A. Give a name to this instruction and state what it does

B. Write the op code for this instruction in binary assuming MAP as in Ch. 8

3. Suppose microinstructions have a width of 24 bits and has 3 fields: a micro-operation field of 13 bits, a next address field and a MUX select field. There are eight status bits in the inputs of the multiplexer

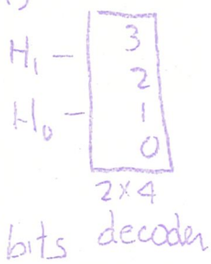
A. How many bits in the MUX select field? How many bits in the next address field?

B. What is the size of control memory? How many bits in the CAR?

C. Suppose input 4 of the multiplex unit is hooked to logical ONE. Formulate a procedure for an unconditional branch.

4. Show the j-th stage of a combinational shifter from S (source) to D (destination) which does

$H_1 H_0$	function
0 0	no shift
0 1	shift right
1 0	shift left
1 1	shift left by 8 bits



assume $0 \le j-8$ and $j+8 \le n$

5. Write a sequence of micro code as in Chapt 8 which does $M \leftarrow M + AC$. The value of AC does not change.

6. The Test 2 computer is a 16-bit machine with 4 general purpose registers and lots of addressing modes. It has 64K of 16 bit memory words. Opcodes are 16 bits wide and instructions which need an address or constant are two words long, the second word being the address or constant

The PC = 0 and the contents of the registers and some of memory are given to right. The opcode X is for an instruction which loads the register R0.

Registers

Name	Contents
R0	13
R1	6
R2	5
R3	1

Memory

Address	Contents
0	X
1	3
2	11
3	7
4	9
5	4
6	8
7	10
8	2
9	12

Give the contents of R0 after this instruction is executed if the source is in the addressing mode:

A. Immediate
B. Direct
C. Indirect
D. Register R1 indirect
E. Relative
F. Indexed by R2
G. Register R3

7. Compare and Contrast

A. Vertical vs Horizontal μ-code (in particular in regrads to nanomemory)

B. Microkids vs Hardy Boys

C. Internal vs External interupts

8 The code on the right is like in Ch 6 and is for the Ch. 5 computer. It attempts to add a stack with the memory location SP acting as a stack pointer.

The subroutines W, X, Y, Z are (in some order) a subset of the standard instructions PUSH, POP, 0-address ADD, CALL and RETURN.

AB Which routine is which?

C Does the stack grow to low or high memory?
Does SP point to "the top of the stack" or "above" or "below" it?

DEF WRITE SUBROUTINE 'T' which is the one missing above

GH. In your main program you want to CALL PRINT a subroutine which sents a character (in lower half of AC) to the printer (it waits til the printer is ready. Assume the char is in AC and the location of print is PRINT. Write the instruction(s) needed to make this call

```
SP:
OLDAC
SUBADD:
TEMP:          } storage

W:
        STA  SP I  (indirect)
        LDA  SP
        INC
        STA  SP
        BUN  W I   (indirect)
X:
        STA  OLDAC
        LDA  X
        BSA  W
        LDA  OLDAC
        BUN  SUBADD I  (")
Y:
        STA OLDAC
        BSA  Z
        STA  Y
        LDA OLDAC
        BUN Y I    (")
Z:
        CLA          (clear)
        CMA          (complement)
        ADD SP
        STA SP
        LDA SP I   (indirect)
        BUN Z I    (")
T:
```

The infamous 1 Dec CRG test by _____

SHOW ALL WORK 1-4 worth 10pts ea 5-8 worth 15 pts each

1. True or false (T or F)
   A. Software is logically completely equivalent to hardware.
   B. In sending a block of 1K bytes the overhead using asynchronous transmission is always larger than any synchronous protocol in the text
   C. In floating point arithmetic, underflow refers to getting a result that is too negative to express in the number of bits available
   D. A Cache operates at the assembly or machine language level rather than the μ-op or μ-code level
   E. All hardware priority interupt systems prevent a device with lower priority from interupting the service routine of a device with a higher priority

2. A. A rather average reader can read 300 words per minute. Figuring 6 Characters per word (including the blanks between) and that we are comparing this to standard ASCII + parity + 2 framing bits then give the baud rate of this reading speed.

   B. Is this slower or faster than the baud rate of the public terminals in the Education building?

   CD. What is the average access time of a Cache system where 80% of the memory references are reads, 60% of the reads are hits (reads check Cache before trying Memory). Writes are write through in parallel. The cycle times are Cache 100ns and Memory 200 ns.

3. ✍. Carefully "step by step" illustrate the restore method when $B = 11_2$ is divided into $AQ = 1010_2$. Label what the steps are doing and indicate the quotient and remainder. [Use E (carry) but no other registers]

4A. From fastest to slowest list a memory hierarchy with at least 6 levels

B. From largest to smallest list the divisions (logical or physical) of a disc unit with at least 6 ~~units~~ levels

5. A Master device wants data from a slave device. The two devices are connected by a data bus and control line(s)

AB. List the sequence of events for 1-wire strobe control

cD. List the sequence of events for 2-wire handshaking control

E. What is the requirement on the slave device that allows you to do the strobe rather than the handshaking?

6. Explain and give use (ie why/what for/where or the like)

A. UART

B. Associative Memory

C. Dirty Bit

D. Block check character

E. Valid Bit

7. Synchronous Communication

A. The sync character in ASCII is 16H, "a" is 55H why can't you trade the codes for these two characters that is what function of the sync character would you lose

B. The message below uses the character-oriented protocol with data transparency using the DLE character. what is the message (indicate logical divisions

DLE STX DLE DLE ETX DLE DLE ETX DLE ETX

C. The message below uses the bit oriented protocol and is all data. What is message? (ie delete inserted bits)

0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0 0 1 1 1 1 1 0

D. Put "I have (had) μ-class (when)" or "I did μ-lab with (who)" or put "I lose"

8. The DMA unit (like that of TP12) is connected to a very high speed output device. Explain why this DMA unit can transmit data at faster rate then the CPU could under program control. I.E. analyze the timing demands of each method.

ORG final orgy(?) done to excess by _____

SHOW ALL WORK 1-8 worth 10pts each 9-16 worth 15 pts each

1. True or False (T or F)
   A. Software is logically equivalent to hardware
   B. A micro-operation is an elementary operation that can be performed during one clock pulse
   C. The use of a stack to store the return address allows recursive subroutine calls.
   D. "Hardwired Control" is generally faster than control by "μ-code"
   E. For sending blocks of ASCII text asynchonous communication is faster than synchonous communication.

2. A. Each Track has 25 sectors, Each Sector 128 bytes (recorded serially bit by bit). The disc rotates at 3600 RPM compute the Baud Rate.

   B. Compute the average access time: Cache acess 50ns, Main Memory is 100ns. 70% are reads (40% hit the Cache, the misses access the main memory after trying the Cache. Writes are writethrough in parallel.

3. AB Formulate a mapping process from (macro) op-codes to (μ) control memory addresses given that μ-memory has 1024 words, op-codes are 5 bits wide and 8 control words are need for each macro-op.

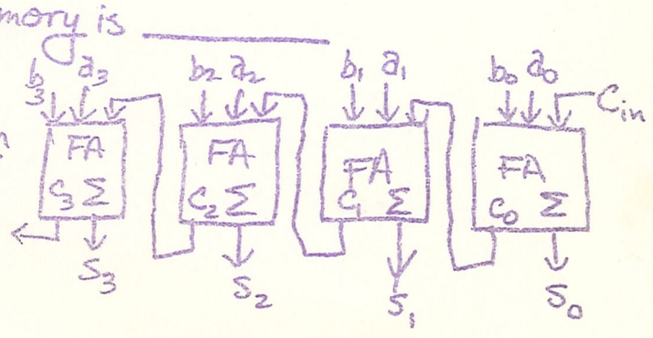
   C. Illustrate your map with the op-code 13_{10} give address in bindry

   D. What is the size of the CAR?

4. AB. It has been decided to replace a μ-memory of 1024 words each 100 bits wide with a μ-memory/nanomemory combination. If there are only 128 different bit patterns in the old μ-memory, how many bits would be saved with the nanomemory

   C. Fill blanks with H (Horizontal) V (Vertical) B (Both) or N (Neither)
   new μ-memory is ____ nano-memory is ____

5. For the 4-bit parallel adder write down the boolean expression which will yield the following flags



   A. C=
   B. S=
   C. Z=
   D. V=
   E (Parity Even) PE=

6. Carefully "Step by Step" illustrate the restore method of division ~~by~~ on $5 \div 3$. The registers A, Q, B, SC are 2 bits wide and D/F is a one bit flag (Show the shifts and indicate the quotient and the remainder)

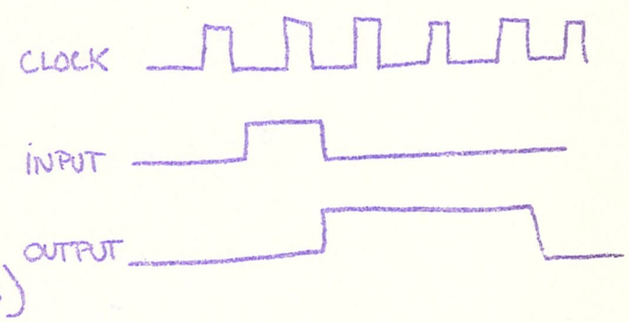
7. In a UART or the Receiver of one of the TP's (I forget which) Input is double buffered. There are two registers REC and INR and an ~~input~~ flag FGI. On some control (say H) INR ← REC and FGI ← 1. On some other control (say T) the CPU reads INR and clears FGI. Actually we use the table to right.

H T	current FGI	next FGI
00	Q	Q
01	Q	0
10	Q	1
11	Q	1

A. Explain the logic of the last line of this table.

B. Realize this table using an RS flip-flop for the flag FGI

8. Another circuit problem. This one generates a "STROBE" of 3 clock pulses. The "input" is hooked to the S input of an RS flip-flop and the "output" is the Q output of the same flip-flop. Realize this circuit (HINT: use some D flip-flops)



9. A Master device wants to SEND data to a slave device. The two devices are connected by a data bus and control line(s). [Return to initial state]

AB. List the sequence of events for 1-wire strobe control.

CD. List the sequence of events for 2-wire handshaking control

E. Why do you need to go to the expense of the extra wire. That is what is it about the slave devise which would make the handshaking more efficient than just making the strobe long enuff to cover the worse possible delay?

10. Another Stack Pointer problem. An SP register is added to the chapter 5 computer. SP points to the TOP of STACK (TOS) and grows toward low memory. Write a sequence of μ-operations which will do the following Macro operations

A. PUSH (i.e. AC becomes TOS) B. POP (i.e. TOS to AC)

11. Write μ-Code in the style of Chapter 8 which does XCHG: AC ← M, M ← AC. To start you off right the first line should worry about effective addresses.

12. This computer has a 19 ~~prime~~ bit word size (One is tempted to call it PRIME, but that has been used). Instructions are one word long and uses the last 12 bits for an address or constant if required. The word at location zero has the last 12 bits equal to 5 and the op-code (first 7 bits (left most)) which loads register R0.

PC = 0, show the contents of R0 after execution if the source is in the addressing mode:

A. Immediate
B. Direct
C. Indirect
D. Register R1 indirect
E. Relative
F. Indexed by R2
G. Register R3

	Contents
R0	10
R1	1
R2	3
R3	9

memory

address	contents
0	see above
1	8
2	11
3	5
4	6
5	7
6	12
7	2
8	13

13. Macro instruction addressing formats. List a sequence of Assembly like instructions which will do $M[X] = (M[A] + M[B]) * (M[C] + M[D])$ or register transfer like instructions

A. General register (8 of them) with 3-address format (Arithematic only between registers)

B. As in A. but with 2-address format

C. Accumulator with 1-address format

D. Stack based with 0-address format.

14.

14. Explain & give use : A. Mask register for priority interrupts (what it allows & prevents)

B. Ring Counter

C. Associative Memory

15. A sort of FIFO buffer designed by two people mad at each other or a schizoid. A, B, G, D, E are 128 bit registers while b, c, d, e are 1-bit flags. The table below gives the starting contents, fill in the rest of the table given the μ-operations below, a clock pulse per line.

1: A ← A+1
b'+c': B ← A, b ← 1
bc': G ← B, c ← 1
cd': D ← G, d ← 1, c ← 0
de': E ← D, e ← 1, d ← 0

	A	b	B	c	G	d	D	e	E
3	0	2	0	2	0	2	0	2
4	1	3						
5	1	4	1	3				
6	1	4	0	3	1	3		
7	1	6	1	4	0	3	1	3
8			0		1	4		
9	1	8	1	6				
10								

16. We have a parallel priority interrupt where each service routine takes exactly 100 μsec to complete. The CPU is working on a long program P at time t=0 μsec. Draw a "timegraph" indicating what the CPU is doing at all times (i.e. P for Program, 0 for device 0, 7 for device 7 etc) given the following sequence of interrupt requests. Devices continue to request service until answered. However if the device again requests service before its service routine is completed then data is lost (indicate what device loses data if and when it happens.) [low numbered devices have priority]
Here is the sequence of requests
t=20 μsec device 5; t=50 μsec device 3; t=150 μsec device 7;
t=220 μsec device 3; t=300 μsec device 6; t=500 μsec device 4;
t=560 μsec device 2; t=600 μsec device 4; t=700 μsec device 7;
t=710 μsec device 0; t=720 μsec device 1; t=800 μsec device 3;
t=850 μsec device 0; t=1000 μsec device 3; t=1200 μsec device 7;
t=1500 μsec device 5; Do not stop your time graph until you have finally completed all the interrupts. When and if data is lost the service routine still takes the same time (i.e. incorrect data is copied) and the new overriding request is honored too (i.e. data could be copied twice)

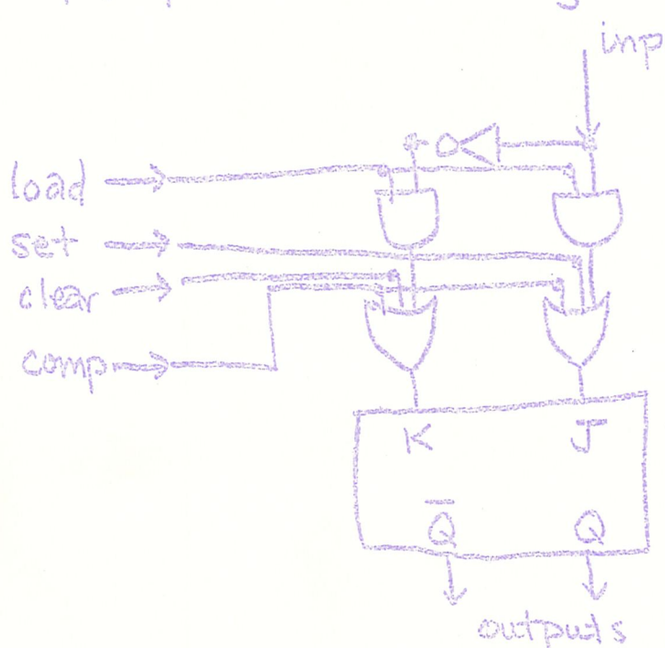
TP5 Due 5 OCT 83 OEG

1.- bit register R.

R has two outputs $(Q, \bar{Q})$, one input (for load) and four control lines (CLEAR, SET, LOAD, COMPLEMENT).
R satisfies the following table

	control inputs					
input	clear	set	load	compl.	current state	next state
y	0	0	0	0	Q	Q
y	1	0	0	0	Q	0
y	0	1	0	0	Q	1
y	0	0	1	0	Q	y
y	0	0	0	1	Q	$\bar{Q}$
y	← all others →				Q	? (doesn't matter)

For example R could be realized using a JK flip flop via the following circuit



R.

YOUR PROBLEM: WITHOUT puting any gates on the clock line NOR using preclear or preset, realize the 1-bit register R

A. with a D flip flop

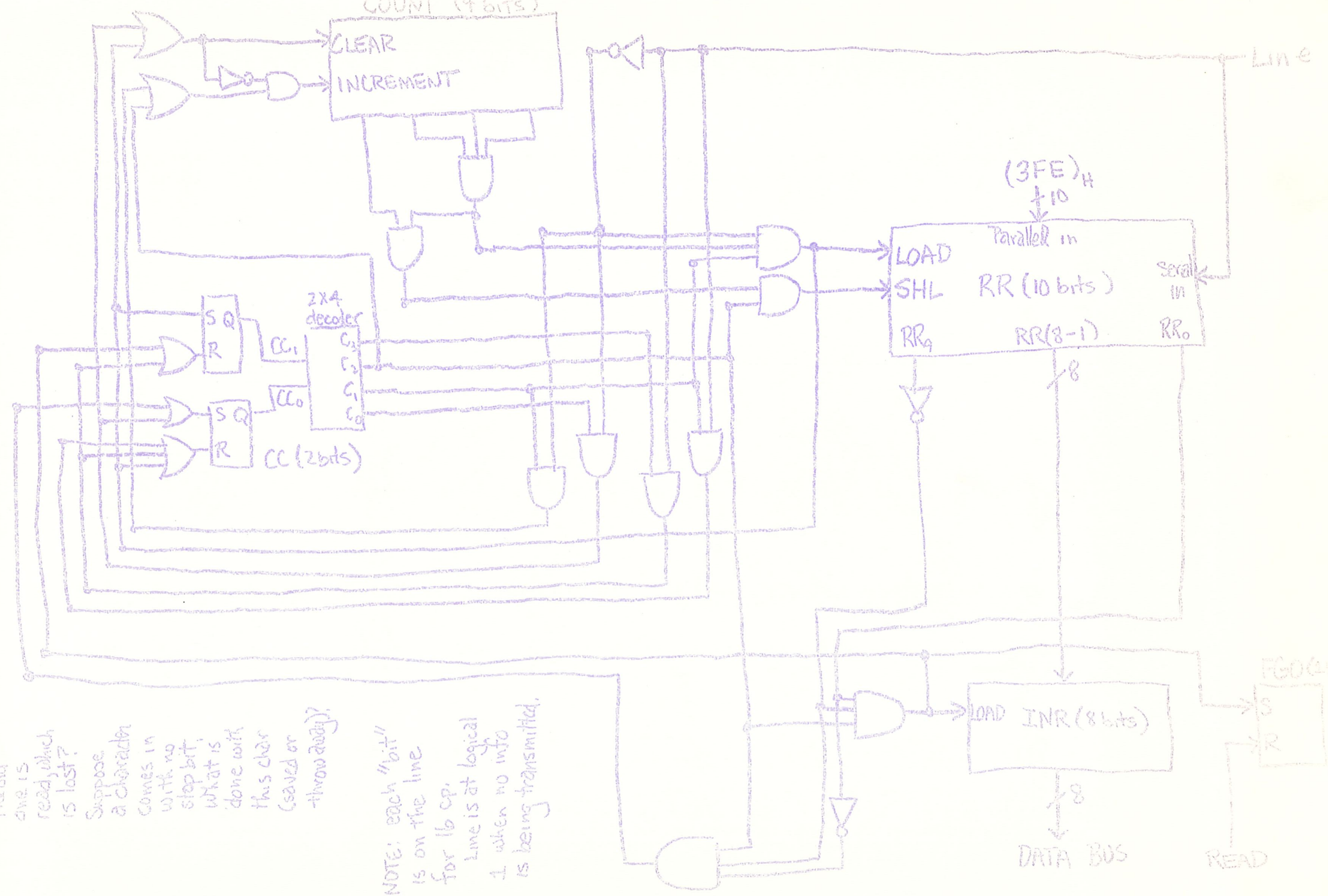
B. with an SR flip flop

P6 due Wed 12 Oct 83 ASYN RECEIVER ORG

A. For each of the cycles C_0, C_1, C_2, C_3 write down the μ-ops that block diagram performs (ASSUME READ=0)

B. Write in words what each cycle does with resp. to the start bit and incoming data. Suppose a new character comes before the old one is read, which is last? Suppose a character comes in with no stop bit. What is done with this char (saved or thrown away)?

NOTE: each "bit" is on the line for 16 cp. Line is at logical 1 when no info is being transmitted.



TT 7 due Wed 19 Oct 83 ORG

1. A. Change the micro code on Pg 4 so that STA has op code 9_1 and ADD has op code 9_3

   B. Add the instruction NOP (no operation) with op code $I_{9_7}B_{11}$ by changing the contents of one address in control memory (which address?)

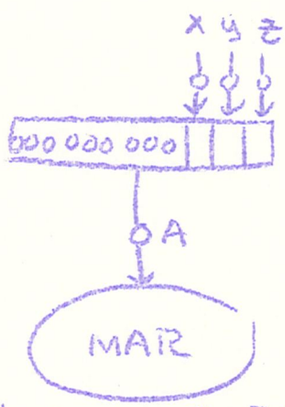
2. It is decided to replace ISZ with DSZ (decrement and skip if zero). Someone suggested changing value "5" from increment to decrement and not touching the micro-code.

   A. Will this in fact change ISZ to DSZ?

   B. Will this change have any unwanted side effects in terms of the micro control? why or why not?

3. By just changing the content of location 0101001 in control memory and with addition of FLAG 8 = MBR_1 (leftmost bit) change the computer to allow "infinite" indirection. i.e. if the left most bit is 1 then its address part is a pointer and not the effective address. Thus we stay in the indirect cycle until the leftmost bit is 0.

4. Add the figure to right to figure on page 1. If A=1, then MAR ← $(xyz)_2$ i.e. if x=1, y=0, z=1 and A=1 then MAR ← 5. Also add the values A, x, y & z to our control word. Using the micro-code format of page 1 (i.e. CW(11,21))

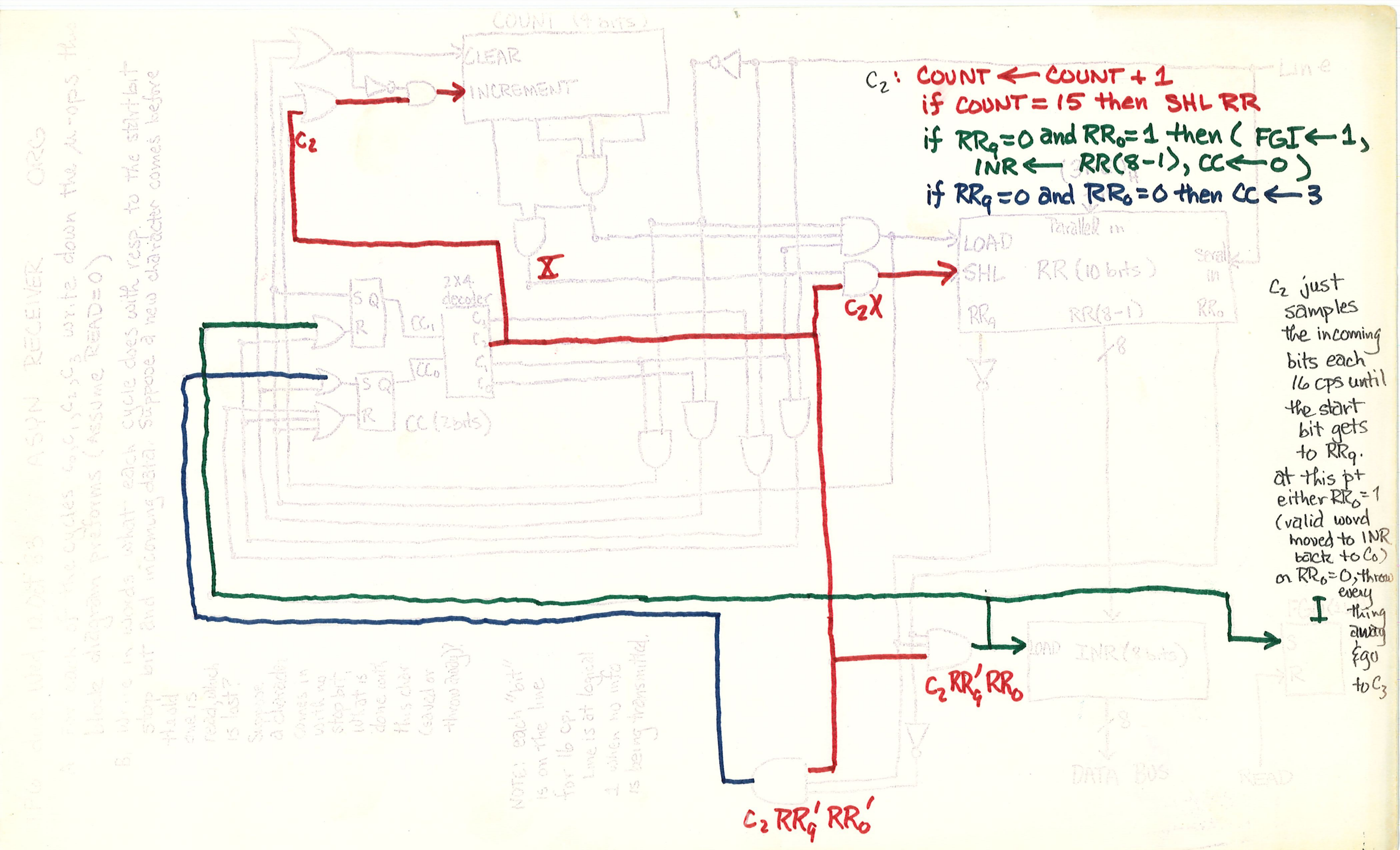


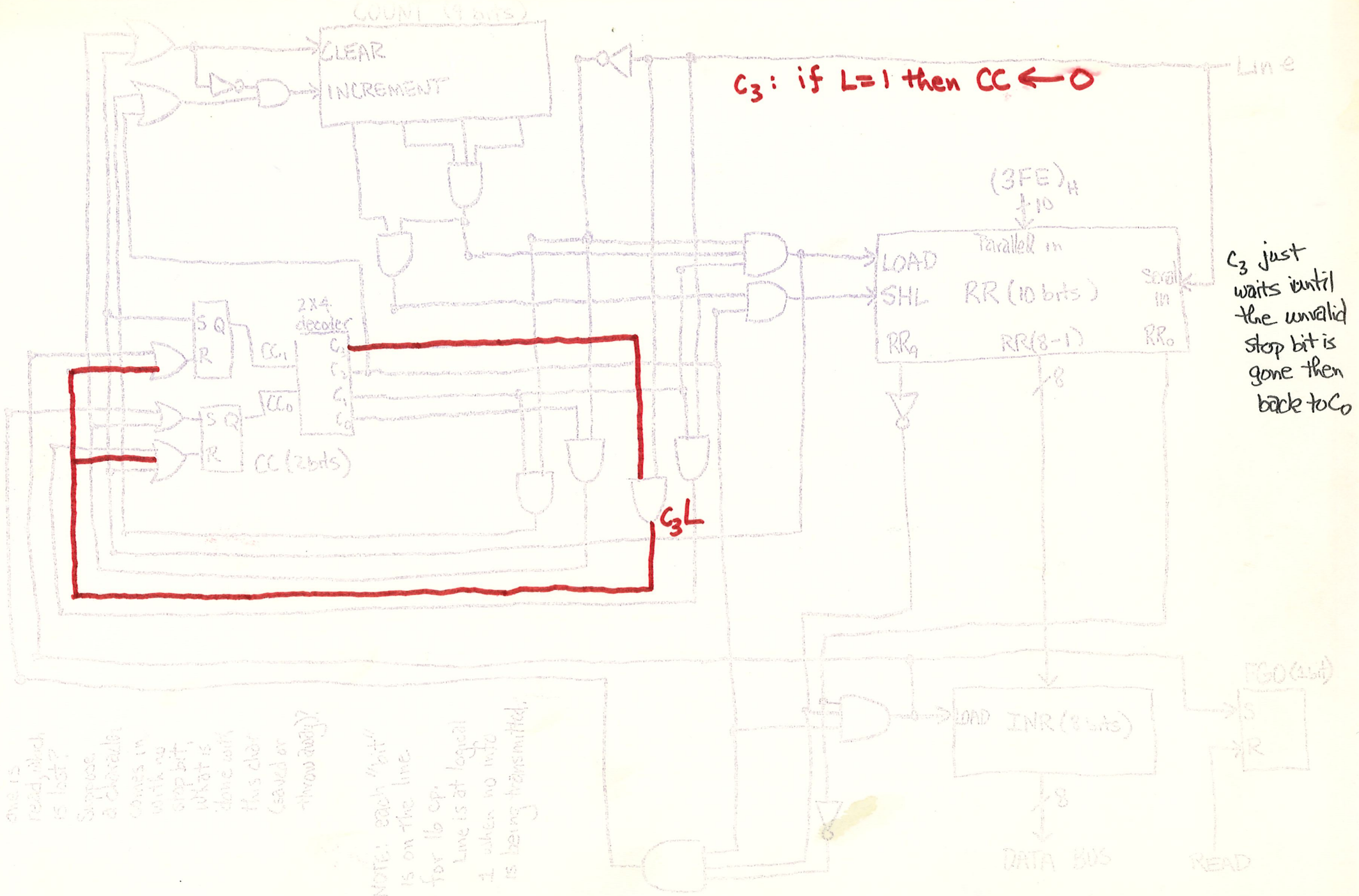
   A. Write a pair of control words which does "M[3] ← M[6]"

   B. Write a triple of control words which does "BUN 7" (no indirection)

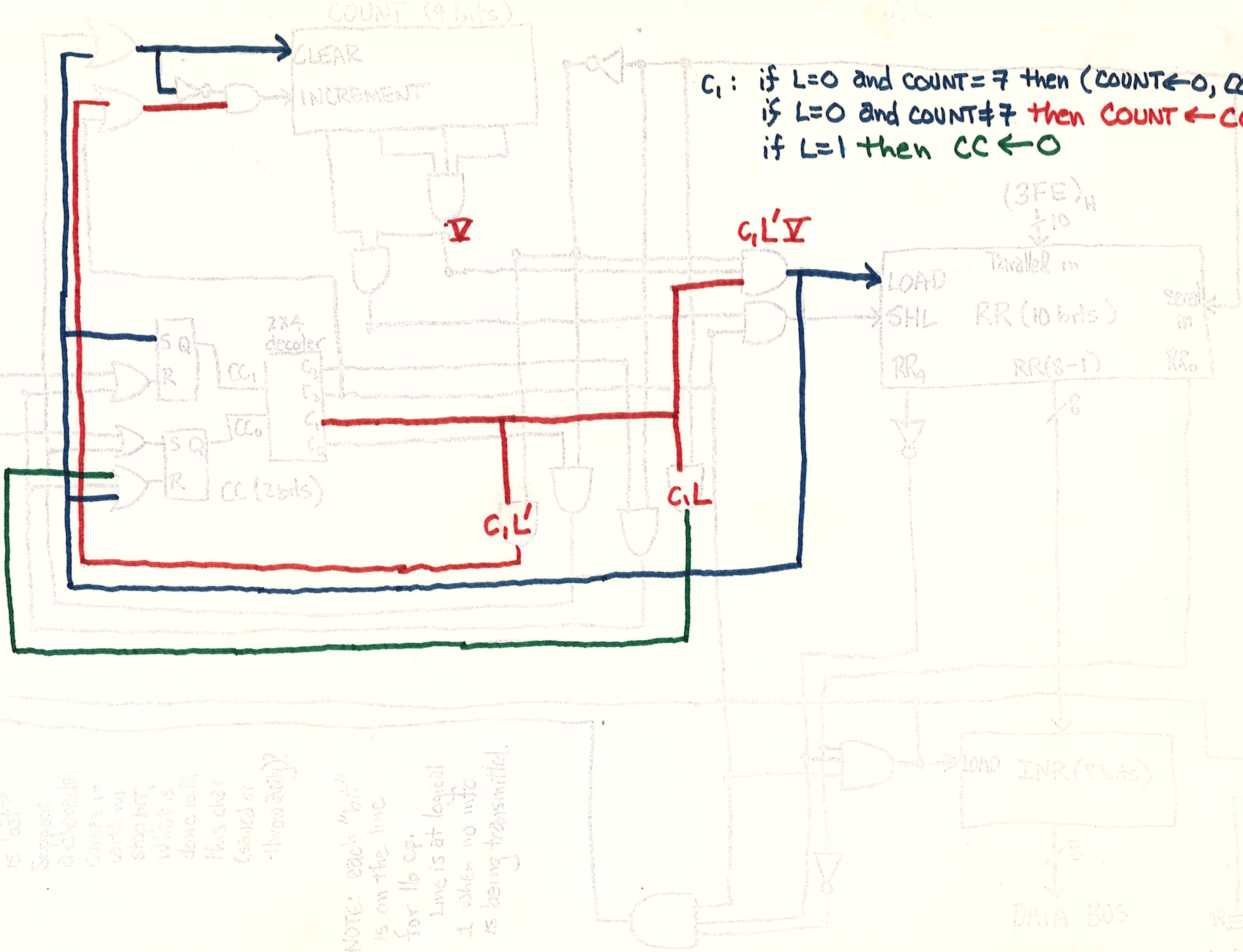
   [you may assume the pair, or triple is followed by JUMP Check.]

5. Still using the setting of problem 4 write a sequence of six control words which does "M[0] ← M[7] + M[4]" (do NOT protect the value in AC).


C_2: COUNT ← COUNT + 1
if COUNT = 15 then SHL RR
if RR_9 = 0 and RR_0 = 1 then (FGI ← 1,
INR ← RR(8-1), CC ← 0)
if RR_9 = 0 and RR_0 = 0 then CC ← 3
COUNT (4 bits)
CLEAR
INCREMENT
C_2
X
C_2X
2x4 decoder
CC_1
CC_0
CC (2 bits)
Parallel in
LOAD
SHL
RR (10 bits)
Serial in
RR_9
RR(8-1)
RR_0
$C_2 RR_9' RR_0$
$C_2 RR_9' RR_0'$
C_2 just samples the incoming bits each 16 cps until the start bit gets to RR_9. at this pt either RR_0 = 1 (valid word moved to INR back to C_0) or RR_0 = 0, throw every thing away & go to C_3
DATA BUS
READ

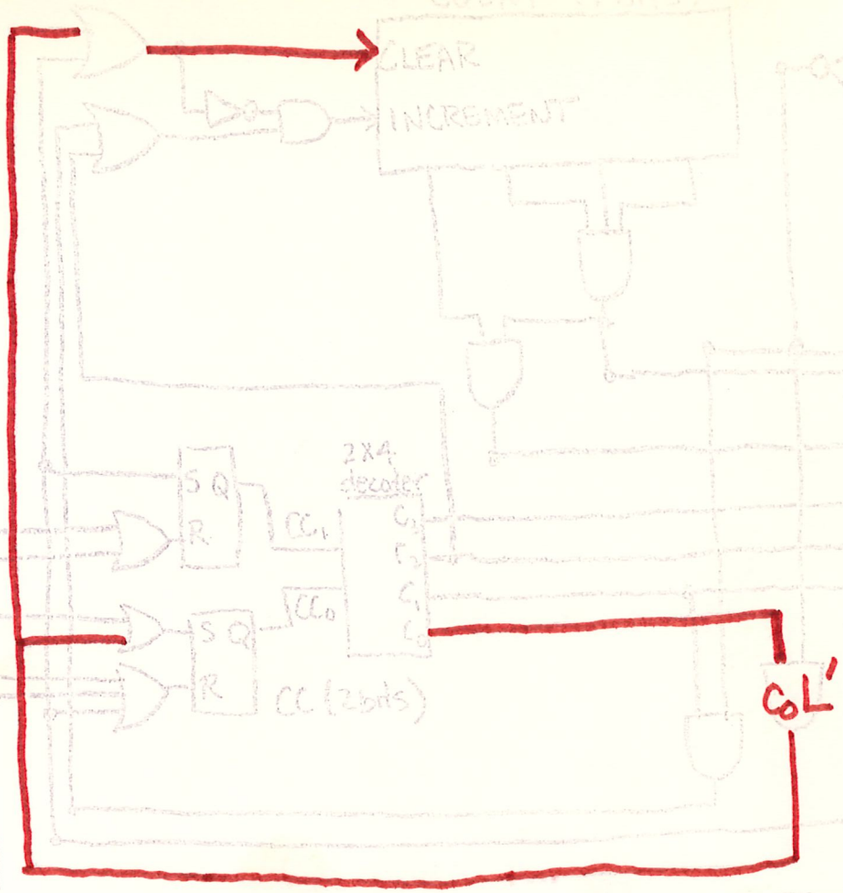


C_3 just waits until the unvalid stop bit is gone then back to C_0



C_1: if L=0 and COUNT=7 then (COUNT←0, CC←2)
if L=0 and COUNT≠7 then COUNT ← COUNT + 1
if L=1 then CC ← 0

C_1 checks carefully to see if it is really a start bit (if it is not then back to C_0) if it is a start bit it changes to C_2 in middle



C_0: if L=0 then (COUNT←0, CC←1)

= L

C_0 just hunts for the begining of a start bit then to C_1

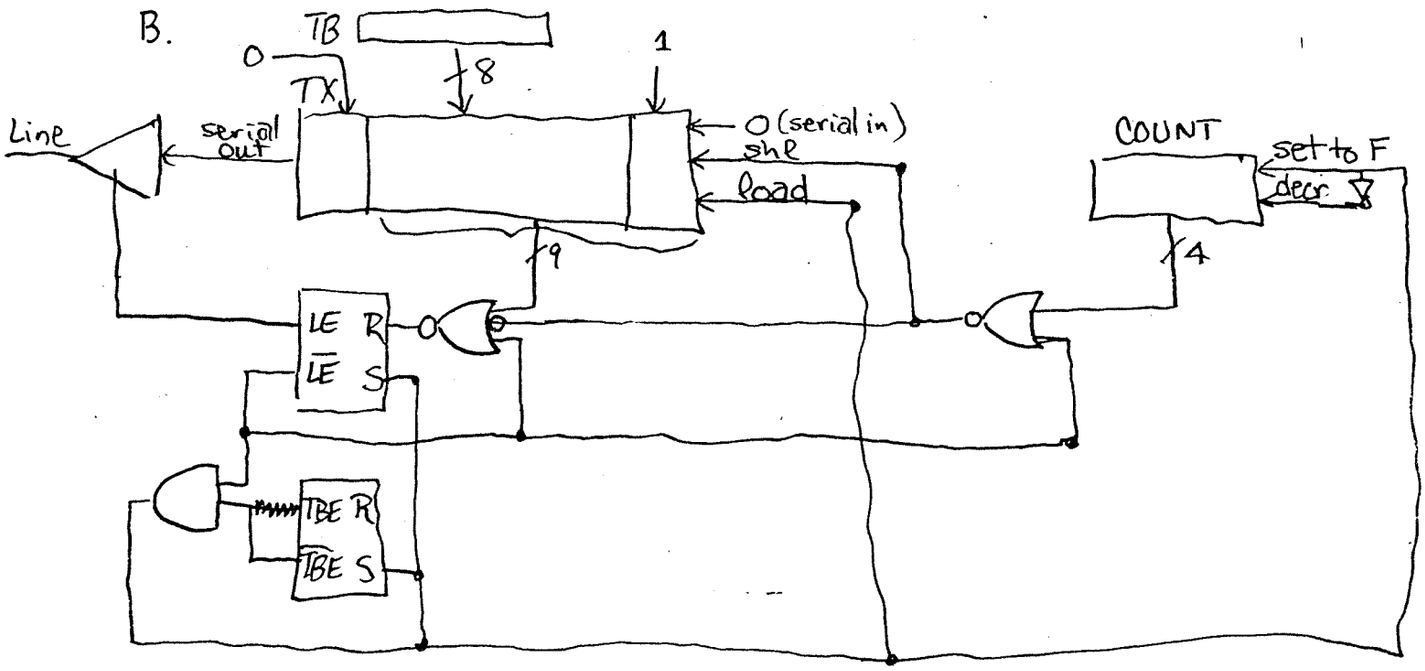
A. LE'TBE': $TX_9 \leftarrow 0$, $TX(8-1) \leftarrow TB$, $TX_0 \leftarrow 1$, $LE \leftarrow 1$, $TBE \leftarrow 1$, $COUN \leftarrow F_H$

LE: if ($COUN = 0$) then Shl TX {$TX_0 \leftarrow 0$}

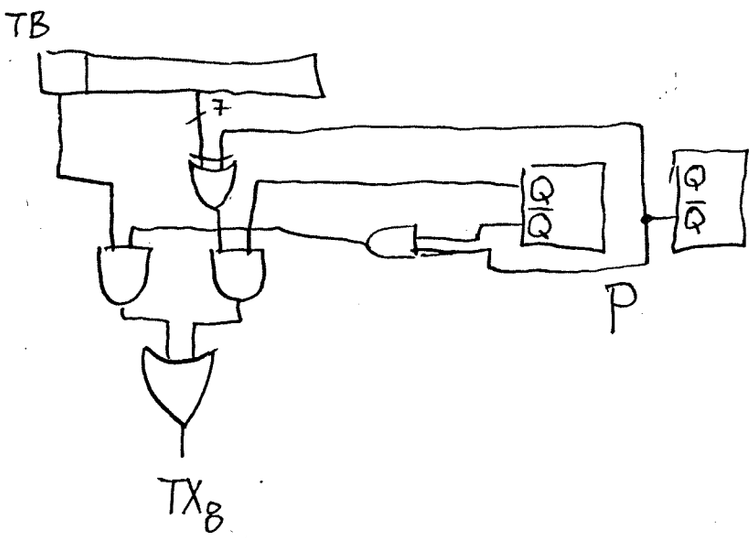
LE: if ($COUN \neq 0$) and ($TX(8-0) = 0$) then $LE \leftarrow 0$

LE+TBE: $COUN \leftarrow COUN - 1$

B.



C.



00 $TX_8 \leftarrow TB_7$

01 $TX_8 \leftarrow 0$

10 odd

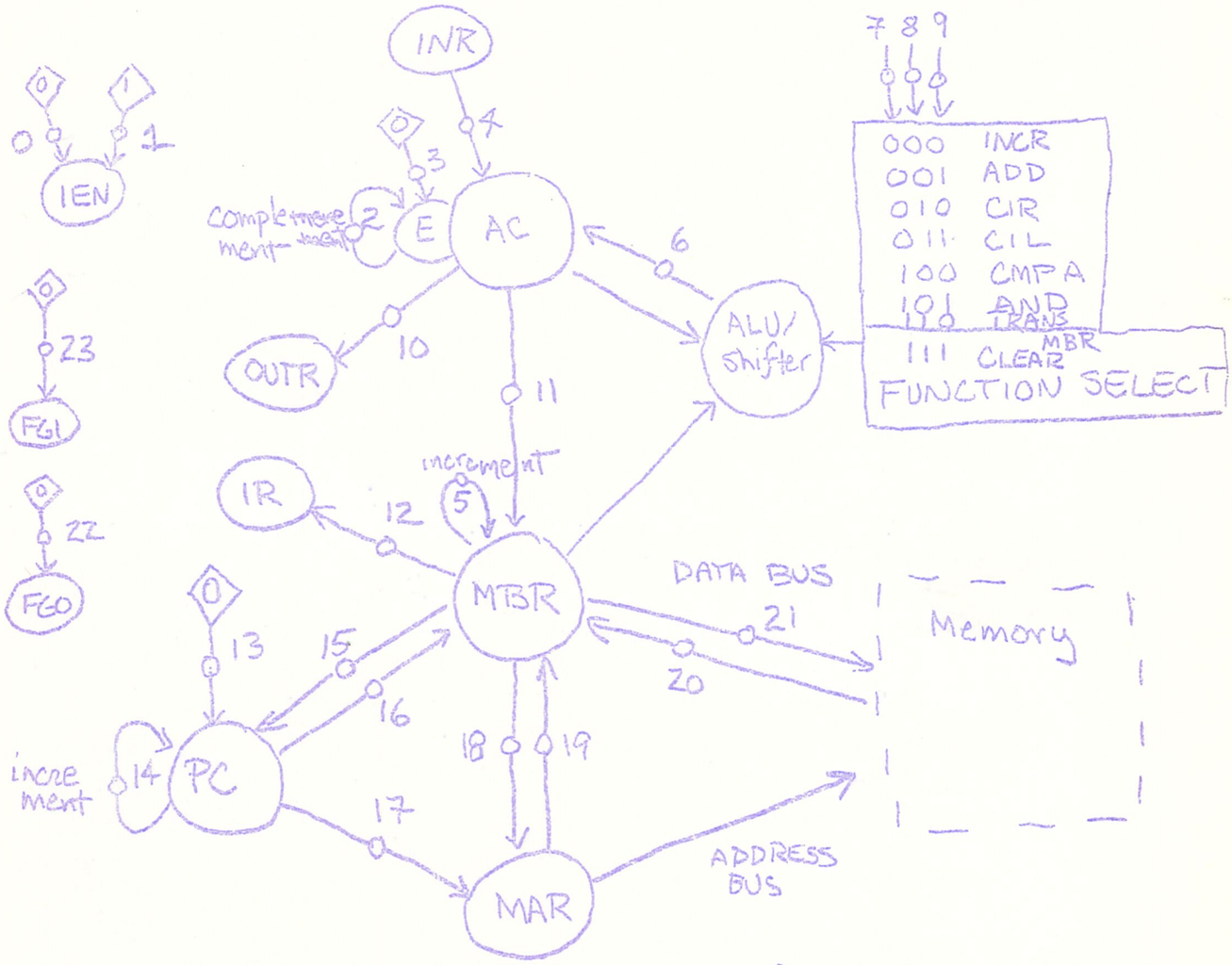
11 even

DATA FLOW / MICRO CODE
(or Hardcore Software)

ORG

Our object is to design a computer that will generate the control for the Chapt. 5 computer.

1. An Alternate view of the Ch. 5. Computer



We need a total of 24 "control* valves" (which are numbered 0 to 23). Once again we can turn this into a control word which we will write in Hex. Examples

02 0000 just turns on valve 17 i.e. MAR ← PC
104000 values 20 & 14 : MBR ← M, PC ← PC+1
001000 value 12 IR ← MBR (OP&I)

We have just done a fetch

To make valve 17 work like this is easy. It needs only to select line 17 over line 18 at the input of MAR and put a "1" on the LOAD control for the MAR.

The width of each line differs: Line 10 is 8 bits wide, line 11 is 16 bits, Line 6 is 16 bits when "7" = 1 and 17 bits when "7" = 0.

The diamonds contain values: line 0 when selected does IEN ← 0, while line 1 does IEN ← 1.

Not all control words are valid. At most one of "0" or "1" can be selected. Similarly for 16, 19, 20 & 11

A very simple computer A CONTROLER

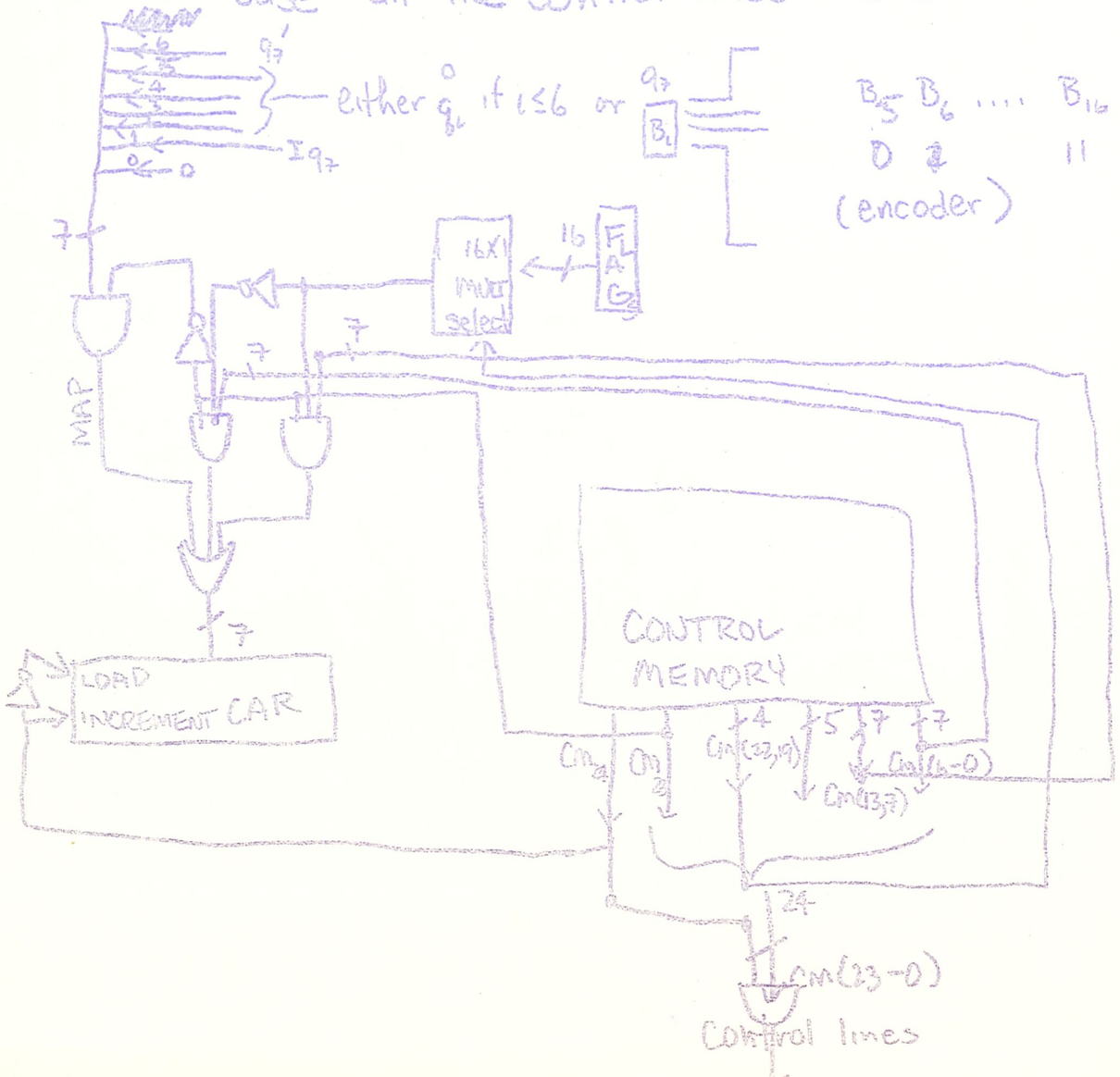
Our Control Computer has a 7-bit CAR (Control Memory Address register) hooked to a very fast ROM of 2^7 = 25 bit words. It has 1 Output ~~Register~~ namely the 24 control lines. It has 2 inputs the FLAGS which give the states of the chapter 5 computer and MAP which translates OPCODES of the chapter 5 computer into addresses for the CAR.

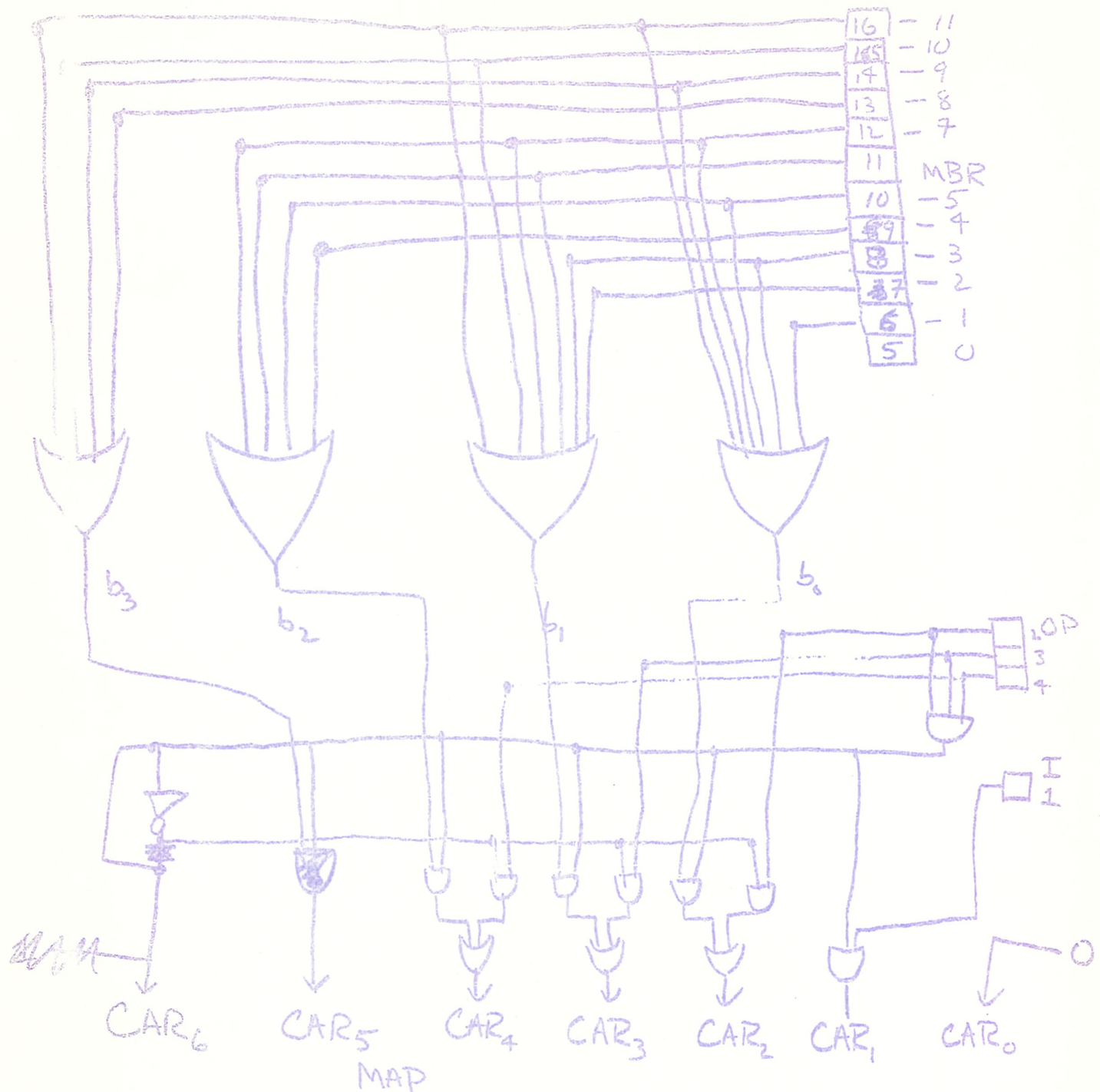
Basically it has two instructions; OUT with $CM_{24} = 1$ (CM = Control Memory [CAR]) which just outputs CM(23,0) for control, and a conditional jump $CM_{24} = 0$. This jump takes ~~two~~ forms:

$CM_{23} = 1$ Field CM(22-19) chooses a flag if it is 1
CAR ← CM(13-7) otherwise CAR ← CM(6-0)

$CM_{23} = 0$ CAR ← MAP

in either case all the control lines are zero.





FLAG	
0	MBR ≠ 0 (set if MBR isn't zero)
1	FG1
2	FG0
3	AC(1)
4	AC = 0
5	E
6	Interupt
7	I (indirect bit) and not q_7

MICRO CODE

	00	01	10	11
0000	AND: CW(18)	SKIP: CW(14)	CLA: CW(6,7,8,9)	SNA: CJMP3 SKIP,CHECK
0001	CW(20)	CHECK: CJMP(2) RUP, FETCH	UJMP CHECK	NU
0010	CW(6,7,9)	FETCH: CW(17)	INP: CW(4,23)	NU
0011	UJMP CHECK	CW(20,14)	UJMP CHECK	NU
0100	ADD: CW(18)	CW(12)	CLE: CW(3)	SZA: CJMP4 SKIP CHECK
0101	CW(20)	CJMP(7) IND, EX	UJMP CHECK	NU
0110	CW(6,9)	EX: JMP MAP	OUT: CW(1,3,22)	NU
0111	UJMP CHECK	IND: CW(18)	UJMP CHECK	NU
1000	LDA: CW(18,6,7,8,9)	CW(20)	CMA: CW(6,7)	SZE: CJMP5 CHECK SKIP
1001	CW(20)	JMP MAP	UJMP CHECK	NU
1010	CW(6,9)	RUP: CW(16,13)	SKI: CJMP(1) SKIP,CHECK	NU
1011	UJMP CHECK	CW(17,14)	NU	NU
1100	STA: CW(18)	CW(21,0)	CME: CW(-)	HLT: Blow Fuse
1101	CW(11)	UJMP FETCH	UJMP CHECK	
1110	CW(21)		SKO: CJMP(2) SKIP,CHECK	
1111	UJMP CHECK		NU	
10000	BUN: CM(15)		CIR: CW(6,-)	
10001	UJMP CHECK		UJMP CHECK	
10010	NU		ION: CW(6)	
10011	NU		UJMP CHECK	
10100	BSA: CW(18,15,16)		CIL: CW(6,8,9)	
10101	CW(21)		UJMP CHECK	
10110	CW(14)		IOF: CW(1)	
10111	UJMP CHECK		UJMP CHECK	
11000	ISZ: CW(18)		INC: CW(6)	
11001	CW(20)		UJMP CHECK	
11010	CW(5)		NU	
11011	CW(21)		NU	
11100	CJUMP(0) CHECK,SKIP		SPA: CJMP3 CHECK,SKIP	
11101			NU	
11110			NU	
11111			NU	

ORG Test 1 by _____

Show All Work for Credit. 1 - 4 10 pts each 5 - 8 15 pts ea.

1. X is a register which can load (parallel), increment and clear. Show the gates and control lines needed to realize the following μ-op's (i.e. block diagram)

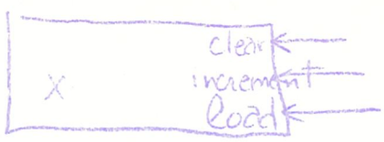
$C_2 t_0$: X ← X+1

$C_3 t_1$: X ← PC

$q_7 C_2 t_3$: X ← 0

$q_5 C_2 t_1$: X ← MBR(AD)

$C_0 t_0$: X ← MAR



2. This problem is about I/O in the ch.5 computer. Ignore interrupts for this problem

A. FGI - the input flag just ~~one~~ went from zero to one. what happened?

B. FGI - just went from one to zero. what happened?

C. FGO - the output flag went from zero to one. what happened?

D. FGO - went from one to zero. what happened?

E. Transfers to and from the input/output registers are to or from where?

3. Write down a sequence of micro-ops which will move the memory word pointed to by the address part of the memory word in $(0A7)_H$ to the memory word in $(38B)_H$

4. This problem is about the ch.5 computer

A. An interrupt has just happened, sometime soon we will jump to a certain location. What is it?

(Instead jump to the arrow on the next page)

5. Using $2^{10} \approx 10^3$, 80 char/line, 40 line/page and 400 page/book estimate the following

A. A mainframe hard disc can hold 512 Mega Char. How many books?

B. A tape 256 000 inches long holds 1600 char per inch. How many books?

C. A floppy disk (single sided & density & 8 in) has 77 tracks (call it 80) of 26 sectors (call it 32) of 128 char. How many pages?

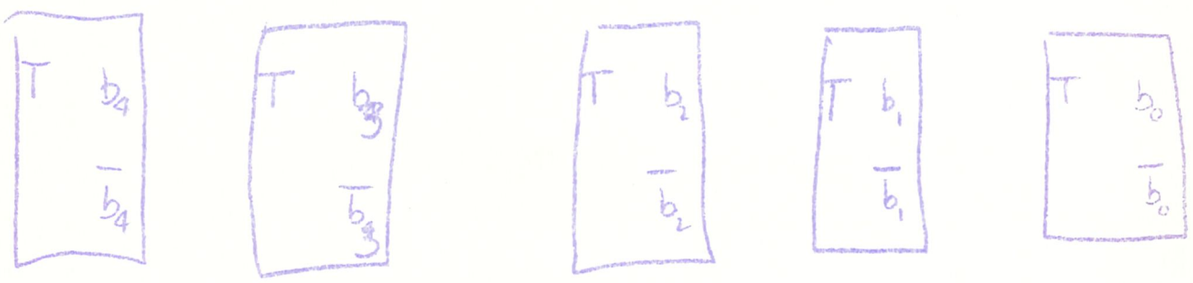
6 Add two registers (12 bits) Front & Rear to be pointers for a memory Queue. Add two new instructions ENQUEUE And DEQUEUE. ENQUEUE (opcode r) puts "the word" where Rear points and decreases Rear DEQUEUE takes "the word" where front points and decreases Front. "The word" either starts or stops in AC. Write the execution cycles in μ-ops for

A. ENQUEUE

B. DEQUEUE

7. Like increment, taking the two's complement of a register can be done in one clock pulse without using an adder. Indeed if $b_4 b_3 b_2 b_1 b_0$ is the content of the register, then b_i is complemented if there is a one to the right of b_i else it is left unchanged. Show how to do this for the T flip flops below. That is if TSC = 1, b ← 2's comp b

TSC



GO TO NEXT PAGE.

I this is the interrupt service routine. Something belongs where the arrow is pointing. Put the analogous "number" for this test there.

II. Can this interrupt be interrupted? (yes/No) So what was done to what flag?

III what is the next to last instruction of service routine (or what is its effect)?

IV what is the last instruction of service routine (or what is its effect)?

Well don't just sit there do them!

8 Consider the μ-ops

$MAR \leftarrow 123_H$

$MBR \leftarrow M$

$AC \leftarrow MBR$

$AC(1-8) \leftarrow A(9-16)$, $A(9-16) \leftarrow A(1-8)$ (*)

$MBR \leftarrow AC$

$M \leftarrow MBR$

A. If the contents of memory ~~bank~~ location 123_H contains $7F0A_H$ before these operations what is its content after?

B. what does this sequence of micro-ops do?

C. If we added the μ-ops (*) with op code $I' \cdot q_7 \cdot B_{17}$ to our instruction set what would be its execution cycle in the style of Ch 5

D E Instead we want the effect of this sequence to happen to the memory location m with op code t (i.e instruction format 

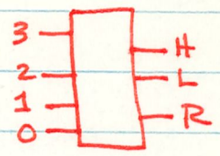
we can find μ-ops which will do this in our 4 timing sequence for execution. Show how to do this. [i.e write out execution cycle] (Assume a chapter 5 FETCH)

TP 5 due 14 Feb 85 Priority Encoders

A 4x2 priority encoder has 4 inputs labeled 0, 1, 2, 3 and three outputs H, L, R

0	1	2	3	H	L	R
0	0	0	0	x	x	0
1	x	x	x	0	0	1
0	1	x	x	0	1	1
0	0	1	x	1	0	1
0	0	0	1	1	1	1

logic table



block figure

note HL is the binary address of the lowest numbered input at logic 1. (That is the lower the input number the higher the priority)

part A. Construct a 4x2 priority encoder using 1 4x16 decoder and 3 'or' gates

part B. Show the block diagram of how to connect 5 4x2 priority encodes to obtain a 16x4 priority encoder. Carefully label your inputs $0_{..}15$ and your outputs $A_3 A_2 A_1 A_0$ and R. You have two 4x1 multiplexers and NO other gates.

(Note B is worth more than A)