1. Rules

   A. On 8½ by 11 paper

   B. Written in ink

   C. Using only one side of each page

   D. Pages stapled or paper clipped together

   Failure to follow any rule costs a point each

2. Graded on 0-10 basis on your reasoning, your ability to express it, neatness and English.

3. Your TP average is computed using only your best n out of m, where $\frac{n}{m}$ is roughly $\frac{2}{3}$

4. Since TP's are assigned a week in advance, the solutions handed in are assumed to be carefully worked out. In any case, they will be graded as if they were.

5. They must be your <u>OWN</u> work.

TP2 due monday 12 sept ORG

Hardware Queue (FIFO LIST)

n registers R0, R1, ... Rn-1 together with n flags (flip flops) F0, F1, ... Fn-1 are to be wired to form a queue, with the following μ-op's.

ENTER (control variable = a)

if $F_{n-1} = 1$ nothing happens. otherwise the contents of register A is put into Rn-1 and F_{n-1} is set to 1.

EXIT (control variable z)

sends the contents of R0 to register Z and clears F0

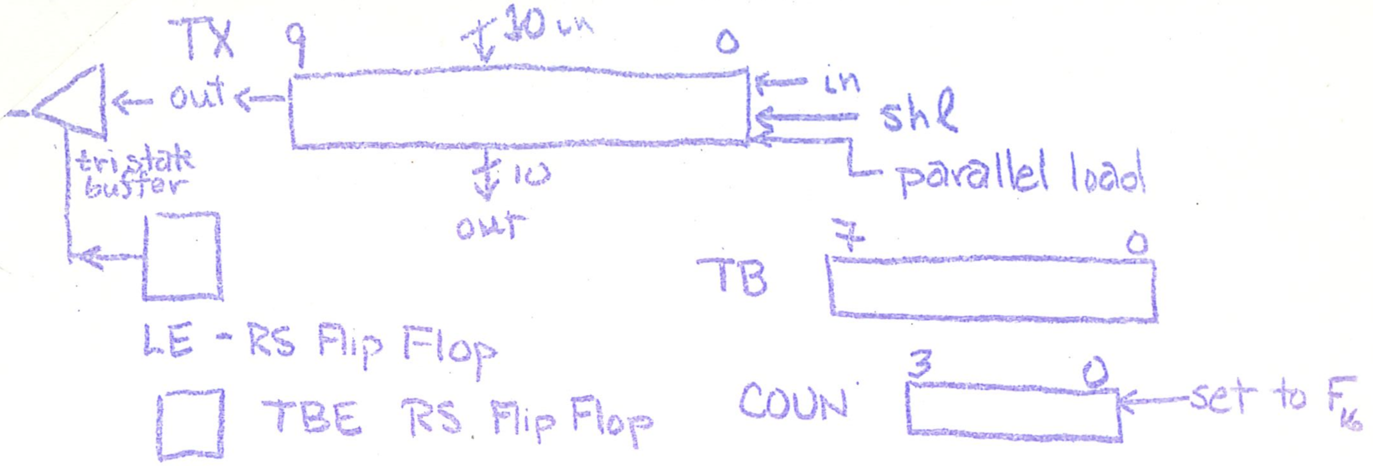
ADVANCE (always on)

for i = 1 to n-2 "moves" R_{i+1} to R_i, clears Fi+1, and sets Fi provided $F_i = 0$ and $F_{i+1} = 1$

1. WRITE these μ-ops in Register Transfer Lang
2. Draw a block diagram that excutes these instructions
3. Add " a: $A \leftarrow A+1$ " to this list. Show the contents of each register and flag after the first 10 clock pulses. (Assume $a \equiv 1$, $z \equiv 0$) Initially $A = 1$ $R0 = R1 = \ldots = R_{n-1} = F0 = F1 = \ldots = F_{n-1} = Z = 0$.

(Careful there are two (or more) pitfalls)

RIAL TRANSMITER (due Mon 19 Sept 83) ORG



If LE = TBE = 0 then TX (Transmitter) is loaded by $TX_9 \leftarrow 0$ (Start bit) $TX_0 \leftarrow 1$ (stop bit) and the contents of TB (Transmit buffer) is put in the middle. AT the same time LE (line enable) and TBE (TB empty) are set and COUN (A 4-bit register which is always counting down) is set to F_{16}.

When LE = 1 the output of TX_9 is placed on the transmission line for 16 cp's (Clock pulses). Every 16 cp's TX shifts left (with zero filling). After all 10 bits have been sent, LE ← 0.

A. Use Reg. Trans. Lang to write the above as μ-op's (don't forget COUNT)

B. Do a block diagram showing control circuts. (I think 3 gates suffice)

C. Show the circuit need for the parallel input to TX_8 to modify this so that: TX_i, $i \neq 8$ is loaded as above but TX_8 is loaded according to the contents of P a 2 bit reg. If P = 00 then $TX_8 \leftarrow TB_7$; if P = 01 then $TX_8 \leftarrow 0$; If P = 10 (resp. 11) then TX_8 gets the value so that TX(8-1) has odd (resp. even) parity.

Due [redacted] ORG

An integer Multiplication Unit.

X & Y are 8-bit registers. T & PP are 16-bit registers. Z is a D flip flop. P is a 1-bit flag. (P = 1 means X is a positive 8-bit number, P = 0 means X is in 2-complement)

CC (cycle counter) is 3 RS flip-flops which when followed by a 3x8 decoder produces the controls $c_0, c_1, c_2, \ldots$ [reg. numbered from right starting w/zero]

```
C_0: idle
C_1: not used until part C

C_2:  { PP ← 0,
same  { shr Y Z  (i.e  $Y_7 ← 0$, $Z ← Y_0$),
clock { if (P = 1) then(T(7-0) ← X and T(15-8) ← 0),
pulse { if (P = 0) then (T(7-0) ← X and $T_{15} ← X_7$ and $T_{14} ← X_7$
      {    and $T_{13} ← X_7$ and ... $T_8 ← X_7$ ),
      { CC ← 3

C_3:     { if (Z = 1) then PP ← PP + T
same cp  { shr Y Z  ( $Y_7 ← 0$, $Z ← Y_0$ )
         { shl T     ( $T_0 ← 0$ )
         { if Y = 0  then CC ← 4

C_4: nothing
```

A. Suppose X contains −12 (2-comp), Y contains 5 and CC = 2 show the contents of each register and each flag in binary after each clock pulse until they stop changing. NOTE P = 0

B. Draw a block diagram show all control lines.

C. This will not work if Y is a negative number in 2's complement but if Y < 0 we can multiply both X & Y by −1. Write down this in Reg Trans Lang using c_1.

10001
0011100
32 16 8 21

64 1 — 00001100
1– 11110100

32 16 8 4 2
1 1 1 1 0
~~48~~
54

TP 8 due Wed. 26 Oct 83 ORG

ASYN RECEIVER FLAGS.

Let $q = c_2 RR_9'$ and change TP6 receiver so that

q: INR ← RR(8-1), set flags, go to c_3

(note INR is now loaded even without "stop bit" and that the control q is true for only 1 clock pulse per each char received)

Add 3 "command" flags ER, $E/\overline{O}$, $P/\overline{NP}$

Add 3 "status" flags FE, PE, VE (together with our old friend FGI)

Also there is our other friend READ (external control line (i.e. from CPU))

if READ = 1 then (DATABUS $\xleftarrow[\text{(via tristate)}]{}$ INR, FGI ← 0)

q: FGI ← 1

q: if $P/\overline{NP} = 1$ (Parity/NoPar) and parity in RR(8-1) doesn't match $E/\overline{O}$ (1 for Even, 0 for odd) then PE (parity error) ← 1

q: if $RR_0 = 0$ then FE ← 1 (framing error)

q: if FGI = 1 then VE ← 1 (overflow error)

Suppose FE = PE = VE = 1 and ER = 0 (Error Reset) then when q becomes one the next char is loaded into INR This new char may have no errors, but in any case the flags FE, PE, VE are not reset. (same true of any subset.)

On the other hand if ER = 1 then the flags FE, PE, VE are about the current char in INR.

Use RS flipflops for FGI, FE, PE, VE and <u>draw a circuit</u> that does this.

TP12 DMA Due Mon 28 Nov 83 ORG

Do a Blow by blow ~~rett~~ description of a DMA transfer of 80^{H} words (both input and output) for the following system

CPU = chapter 5 computer minus its I/O registers INR, OUTR, FGI, FGO but it still has IEN

It ~~a~~also has control lines busreq, busgranted and interrupt

The DMA is memory-mapped, with

location X : status word (busy, errors, done etc)

X+1 : control word (start, direction of transfer etc)

X+2 : buffer address (points to location for next enter and exit

X+3 : count (number left.)

it ~~$~~also has registers BR (like MBR) and I/O registers INR, OUTR, FGO, FGI like the chapt 5 computer.

The incoming or outgoing characters are not packed (i.e. only the low order bits contain real data).

You may need to add other control lines, but do so only if necessary.

DMA interrupts CPU when done.

Do NOT use a flowchart

ORG PT1

1. Side show Inc produces a 29 bit computer (it is prime) with 128K of memory. The instructions fit into one word with an address field, a register field and an indirect bit there are 32 registers.

   A. How many op codes? B What is the width of PC? C, MAR? D MBR?
   E. the most likely size of the registers. F. Draw a reasonable op code (instruction) format (include the number of bits in each part)

2, I, Write -12 in an 8-bit register using A sign-magnitude B, 1 complement C, two's complement

   II, A How many wires are needed to connect each pair of 32 registers each with 16 bits directly B, How many if we use a common bus

3, Estimate 80 printed characters per ~~page~~ line, 40 lines per page, 200 pages per volume, hundred volumes per row, 10 rows per bookcase 80 bookcases per floor and 10 floors per library.

   A. How many characters per library (estimate as a power 2)?
   B. How many pages is 64K characters ? ⇐
   C. The Cyber has 2^{17} words and packs 10 characters to a word. About how many volumes is this?

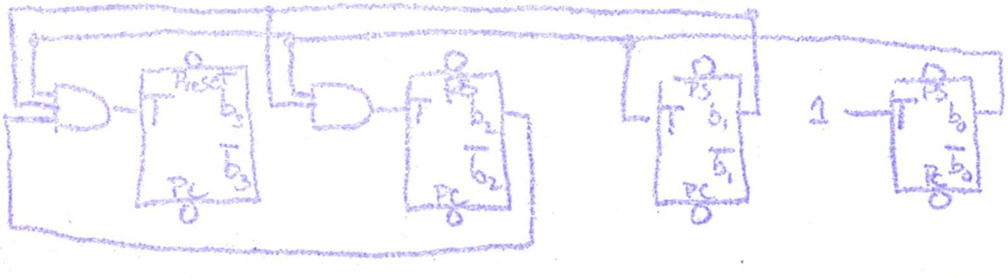
4, A. Register X is 4 bits wide, re write the μ-op $q'r$: if(X=0) then PC←PC+1 without the if-then.

   B. With 3- D Flip-Flops make a ring counter. Include a control signal swhich initializes the output with exactly one one,
   C. List the micro instructions which will transfer the contents at memory word $(0F96)_{16}$ to memory location $(83A7)_{16}$

5. Realize the state table to right using a RS flip flop (and some gates)

Current State	Next State / Output	
	inputs	
y	x=0	x=1
0	0/1	1/0
1	1/1	0/1

6, Here is BCD 2^4-counter modify it to make in a mod12 counter



7. ~~Use $I=1$, $q_{11}=1$ $B_{11}=1$ ($B_{12}=1$ resp)~~ Add a new 12-bit register SP (Stack pointer) and two new op codes (r, s control lines) to Ch5. Comp. when $r=1$ (push) the contents of AC is put into M[SP] and SP is then incremented. When $s=1$ (pop) SP is decremented then AC is loaded with M[SP]. Show the μ-op (with controls) for the execution cycle of each of these instructions.

8. Suppose in Ch.5 computer we change the op-code q_3 add registers T1, T2 (12-bits wide); NU (4-bits wide) and a Flag A (1 bit) [initially cleared]. Consider the execution cycle c_2

```
$A'q_3c_2t_0$ : $T1 \leftarrow MBR(AD)$, $MAR \leftarrow PC$
$A'q_3c_2t_1$ : $MBR \leftarrow M$, $PC \leftarrow PC+1$
$A'q_3c_2t_2$ : $T2 \leftarrow MBR(AD)$, $NU \leftarrow MBR(1-4)$   (I+OP fields)
$A'q_3c_2t_3$ : $A \leftarrow 1$, stay in $C_2$
$Aq_3c_2t_0$ : $MAR \leftarrow T1$, $T1 \leftarrow T1+1$
$Aq_3c_2t_1$ : $MBR \leftarrow M$, $MAR \leftarrow T2$, $T2 \leftarrow T2+1$, $NU \leftarrow NU-1$
$Aq_3c_2t_2$ : $M \leftarrow MBR$, ~~if (NU=0) then~~
$Aq_3c_2t_3$ : if (NU=0) then ($A \leftarrow 0$, change to fetch or whatever (but not execute)
        [else stay in execute]
```

Question what does this instruction do? if X+1: $\overset{1-4}{\boxed{}}\ \overset{5 - 16}{\boxed{}}$ is the memory word after the instruction above. What is the role X(AD), X+1(AD) and X+1(1-4). Why is PC incremented?

9. Use 5 3×8 decoders with enable to make a 5×32 decoder

10. Draw a block diagram and control lines to realize T1, T2, NU, A (a RS Flip Flop) in problem 8.

11. Explain how subroutine calls are made on the Ch.5 computer and how returns work.

12. Draw an analogy between interrupts and phone calls, what acts like ION and IOF? What alternative is there to interrupt driven phone calls. What feature does the chapt5 computer have that the absent minded professor doesn't in regards to this subject.

13. Change instruction $\overset{I}{\boxed{0}}\overset{OP}{\boxed{0}}\overset{AD}{\boxed{XX\cdots X}}$ to NOP (No operation) show the register which must change. Which do not change? which can change?

PT2 ORG

1 Either name the "game" or give one its features. It was the first program they tested their new computer on in "The soul."

2, Substract C6 from B7 in an 8-bit machine show how the flags C, V, S, Z at the end.

3, The μcode to right is in the style of Ch 8

A. What does it do?

```
        ORG 40
NOP            I  CALL  INDRCT
READ           U  JMP   NEXT
BRTAC, ACTBR   U  JMP   NEXT
WRITE          U  JMP   FETCH
```

B, Assumming the mapping opcodes $\Longrightarrow$ control memory addresses as in chapter 8 what is the op code of this instruction.

C, Write μ-code in this style which does AC ← AC ⊕ M with op code 110

4, Write programs to evaluate $X = (A+B)/(C-D)$ using

A. three address general register machine

B. two " " " "

C, one " accumulator "

D zero " stack "

5, Tell how the effective address is found in each of these addressing modes A. immediate B. direct C. indirect D register indirect E. register D. autoincrement E. indexed F relative G base register

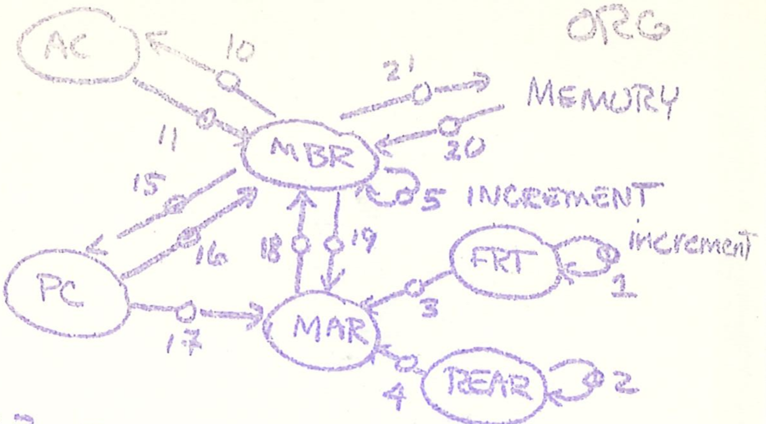
6, Design a circuit for ALU which yields →

S_1 S_0	Cin = 0
0 0	A
0 1	A + B
1 0	$A + \overline{B}$
1 1	B − 1

7, Convert (A+B)/C*(D+A) to reverse polish. Suppose A=1, B=2, C=3, D=4 Show the stack operations and the stack at each stage when evaluating it

8, Design a 1-bit register using an RS flip-flop which loads or sets i.e.

load	set	current state	next state
0	0	Q	Q
1	0	Q	y
0	1	Q	1
1	1	Q	? doesn't matter



9. The old queque problem

A. Write a sequence of control words a la purple press which will take a word where FRT points & place its content in AC (dequeque)

B. What should value #2 be labeled? Do part A for enqueque (AC → M[REAR])

C. Write yet another sequence which UMPS TO (FRT)

10. What does this sequence do CW(20) CW(10, 16) CW(11, 21) CW(5), CW(15)?

11. Using a stack pointer for routine calls show the status of stack, stack pointer, and AC as this programis running

```
X: If AC ≤ 1 then (AC ← 1 and ret)
     else ( push AC, AC ← AC − 1, Y: CALL X, AC ← AC + POP, RET )
```

intially AC is 3 and X is called from location Z

12. "Pascal Case Statement"

```
CASE I OF
  0 : CALL A
  1 : CALL B
  2 : CALL C
  3 : CALL D
END
```

A. Suppose the AC contains I write a ~~problem~~ program part (a la chapt 6) which will do this (Hint substract 3) using the SZA.

B. same problem but now we use a "jump table", Memory word @ location X holds the address of A. B, C & D are located at $A + 2^3$, $A + 2 \cdot 2^3$ and $A + 3 \cdot 2^3$. Without using any skips write the programpart (do not change the value of X.)

C. The hard one, which of these methods is better? when is A better than B and vice versa? Which does "Pascal" use?

D. Find two examples of each method in the text.

13. A. What is a COLD BOOT?

B. What is a TRAP?

C. What is the difference between PARALLEL ARRAYS and PIPELINING

D. Explain the difference between Horizontal & Vertical u code

Summary: Ch 9 & 10 (& review) [illegible]

Ch 9 1. Addition & Subtraction of <u>unsigned</u> numbers. What conditions indicate "out of range" (i.e. carry for addition, no carry for subtraction. Modifications necessary for <u>sign-magnitude</u> numbers

2. Multiplication and division (via restoring method) of <u>unsigned</u> and <u>sign-magnitude</u> numbers. "Out of range" conditions (which <u>vary</u> by the bit-size of the "result"). In particular the algorithms in 9-4 & 9-5.

3. The μ-programed calculator in 9-8

Ch 10 0. (Addition & subtraction of signed 2's complement numbers — covered in earlier chapts)

1. Array multiplers

2. Floating point formats, excess 2^{n-1} numbers. "out of range" conditions. (For <u>Normalized</u> numbers)

Add & Sub: 1. check for zeros
2. align mantissas
3. do it
4. normalize & check "out of range"

Mult & Div: 1. check for zeros
2. do it for exponents
3. do it for mantissas
4. Normalize & check "out of range"

3. Changes for mantissas "as integers" vs "as fractions"

PT3 ORG

1. Suppose we are using n-bit registers A & B and unsigned numbers. Explain why $\bar{B}+1=2^n-B$ what is the condition on $A+\bar{B}+1$ that is equivalent to $B > A$. Explan why.

2. Illustrate the restore method of division on B = 11 ~~AQ~~ 1000 = AQ

3. A UART is operating at 9600 baud. What is the bit-time? Assuming we are sending ASCII characters with parity, framed with 1 stop bit and 1 start bit. What is the character rate and the character time? What is the shortest time it will take to send 1K characters? What amount of this time is spent in the framing overhead?

4. Design an array multipler (~~a "2 x 3" one~~ a "3 x 2" one)

5. Compare and contrast
   A. Isolated vs Memory-Mapped I/O
   B. Strobe control vs three-wire handshaking
   C. Synchronous vs Asynchronous Serial transfer
   D. Daisy chain vs parallel priority interrupt
   E. DMA vs IOP
   F. write through vs write back
   G. rollover vs lockout
   H. Polled vs Interrupt driven I/O.

6. True or False (T or F)
   A. The operating system "needs to know" if the main memory is paged.
   B. The operating system has no need to know of the existence of a CACHE
   C. Associative Memories are more expensive than regular old memory/bit.
   D. A CACHE has an ~~speed~~ access of time of 50ns and main memory's is 250ns. A hit ratio ~~yields~~ of 90% yields an average access time of 75ns.
   E. A mask register is used with parallel priority interrupt.
   F. The logical address space is always bigger than the phyiscal address space.
   G. The "bus arbitrater" gives priority to DMA over CPU.
   H. "Cycle stealing" is what most DMA units do
   I. Divide overflow condition in the restore method will catch dividing by 0.
   J. The "dirty bit" saves time by allowing clean pages to be thrown away.
   K. Rollover and lockout are terms applied to keyboards
   L. IBM "channels" have no memory of their own but CDC PPU do.
   M. Memory interleaving determines the number of memory words brought into the CACHE in one memory cycle.
   N. Xon, Xoff is protocol for FIFO buffer management
   O. Track is to sector as segment is to page (when all four are present)
   P. Main memory is roughly 10^4 times as fast as either discs or drums.

PT3 P2 ORG

7. the code, to right is in 8080 code what does in do?

```
LOOP:   IN   CONSTATUS
        ANI  TX BUFFEMTY
        JZ   LOOP
        MOV  A,M
        OUT  CONDATA
        INX  H
        DCR  C
        JNZ  LOOP
        RET
```

8. It has been decide to wire up a piano as an output device. (88 keys, 3 pedals, 16 time lengths) How many bits are needed to provide this info/note. Assume the shortest length of time is 1/8 of a second. What is the baud rate of the fastest this output device can use.

9. Give a typical message format for synchonous character-oriented protocol. Show how zero insertion works in the bit-oriented protocol when a zero followed by the 10 bits that represent the binary equiv. of 1023 are sent.

10. Illustrate the match logic in one word of an associative memory (i.e. circuits and gates)

11. List the sequence of events in a DMA transfer of one word of memory to some I/O device (assume DMA is initized) List the sequence of events needed to access one sector on a disc.

12. Suppose the MARS Lander sends back digital pictures. There are 4K by 4K pixtels per picture and 256 shades from Black to white for each pixtel. At what bit rate will we receive a picture/minute?

13. List the sequence of events ~~when~~ in a paged memory system of a memory reference when the logical address "in main memory" and when it isn't

14. Descibe the process of multipling two floating point numbers and give the possible "out of range" errors

15. Describe the registers in the 8080 family that are programable (acessable to the programer) and their sizes.

16. Why is there a need of a test and set instruction?

17. A disc rotates at 3600 rpm each track has 26 sectors and each sector has 128 ~~cha~~ bytes recorded serially. What is the bit rate, and time per character.