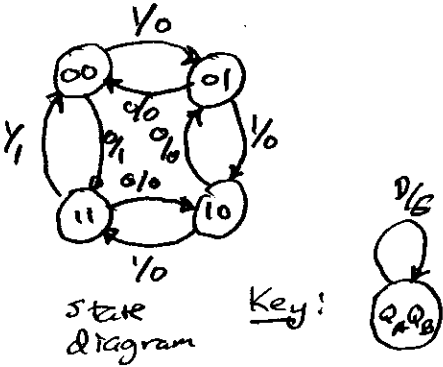


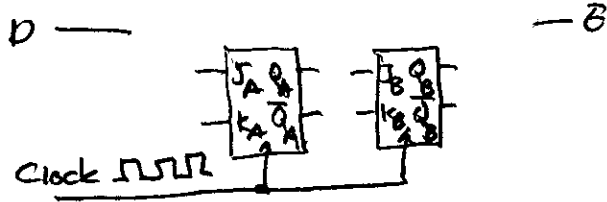
Physics 116B: Final Exam

3/19/2001

Closed book and notes except for three 8.5 × 11 in² sheets of paper, the MAS Manual and the M68000 Programmer's Reference Manual excerpts. Show reasoning for full credit. There are 7 problems (4 long and 3 short) and 200 points.



Skeleton of circuit:



1. Consider the state diagram above. There is one input (D) and one output (E). The circuit is to be implemented using two JK flip-flops, NAND gates and inverters. This is a two bit binary up/down counter.
 - (a) Design the combinatorial circuits for J_B and K_B . Be reasonably efficient. Note that you are *not* being asked to design the connections for the inputs of flip-flop A.
 - (b) Design the combinatorial circuit for the output, E.
 - (c) Calculate the maximum clock frequency for the circuit as developed so far (ignoring the input gating for flip-flop A) using the specifications in Table 1.
 - (d) Explain how the output, E, could be used to cascade two of these circuits to make a 4 bit counter.

Table 1: Chip Timing Specifications for Problem 1

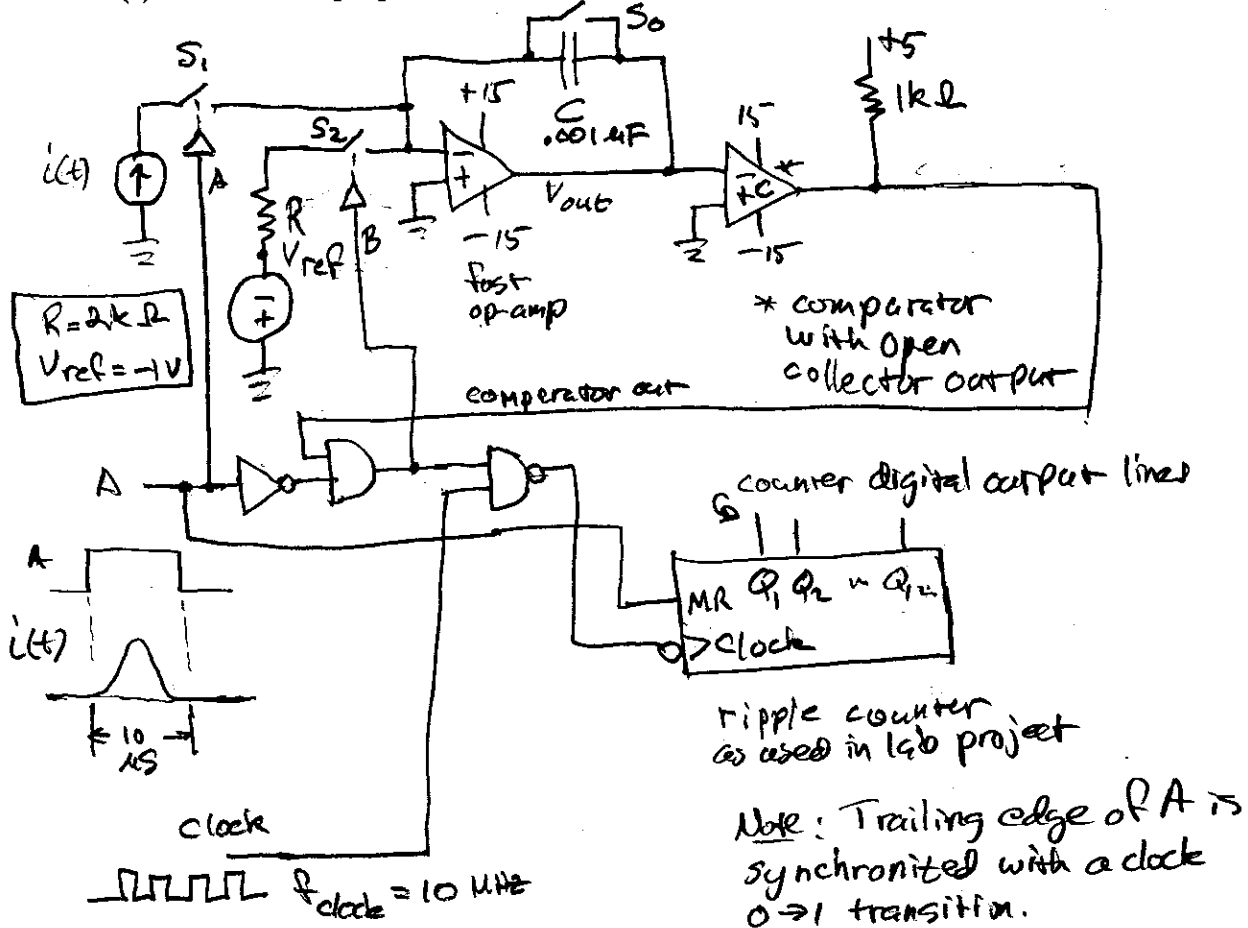
Parameter	Maximum	Typical	Minimum
NAND propagation delay	15 ns	10 ns	-
Inverter propagation delay	15 ns	10 ns	-
Flip-flop propagation delay	30 ns	20 ns	-
Flip-flop setup time	-	-	25 ns
Flip-flop hold time	-	-	5 ns
Flip-flop clock pulse width	-	-	20 ns
Flip-flop clock frequency	30 MHz	-	-

2. The circuit below shows an ADC to produce a binary number proportional to the charge, Q , contained in a finite-length current pulse, $i(t)$ ($i(t) \geq 0$). The capacitor is discharged initially by closing S_0 momentarily. These switches are actually FET circuits controlled by logic levels. S_1 is controlled by A and S_2 by B (closed when the corresponding inputs are "true"). (The gating for switch S_0 is not shown. It is closed momentarily and opened to discharge C before S_1 is closed, and remains open for the rest of the digitization cycle.)

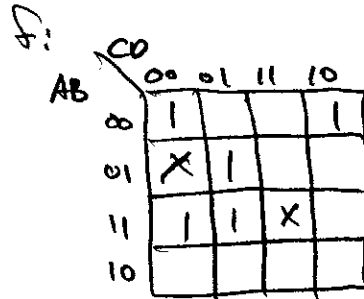
As you can see from the diagram, S_1 is closed for the $10 \mu\text{s}$ maximum duration of the current pulse, then opened again by the signal on A . The op-amp output, V_{out} , is fed into the "-" input of the comparator which has an open collector output.

With S_1 open, closing S_2 will cause V_{out} to increase.

- Find an expression for V_{out} in terms of Q and C at the instant just before A makes its downward transition.
- Find dV_{out}/dt when S_2 is closed and S_1 is open. Show that this quantity is constant. ($R = 2 \text{ k}\Omega$, $V_{ref} = -1 \text{ V}$)
- Explain in some detail how this circuit works to produce a number proportional to the charge. Include a timing diagram showing A , B , $i(t)$, V_{out} and the comparator output.
- Find the time required to digitize a pulse resulting in $V_{out} = -10 \text{ V}$ at the end of the input current pulse. (clock frequency = 10 MHz)
- What is the purpose of the $1 \text{ k}\Omega$ resistor connected to the comparator output?



4. (a) Use NANDs and inverters to implement the function represented by the Karnaugh map below. Be efficient, making use of "don't care" states appropriately.

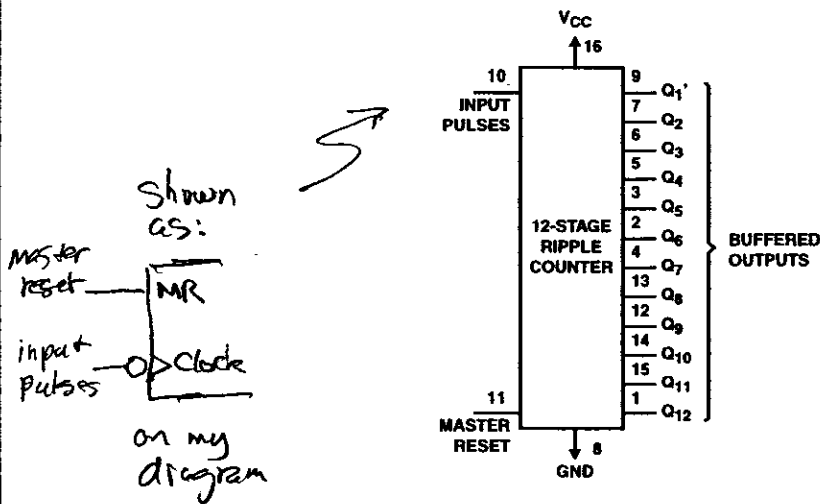


- (b) The logical variables A and B represent a two bit binary number, m , with B the less significant bit. C and D represent another such number, n . Both are normal weighted binary positive integers. Make a Karnaugh map for the logical function, $m > n$, and implement efficiently using NANDs and inverters.
5. The Macintosh system does not allow absolute addressing modes to be used for accessing code or data words via their labels in an assembly language program.
- What sort of indirect addressing is used to access data?
 - What important precaution is therefore necessary in writing instructions involving the A registers?
 - What 68000 register is used for indirect addressing of instruction code?
 - What is the advantage of this restriction on absolute addressing?
6. Explain the advantages of digitizing an audio waveform of a musical performance at a sampling rate of 44 kHz and 16 bit resolution rather than 22 kHz and 8 bit resolution. (Note: Human hearing typically extends up to 17 kHz or so.) What is a possible disadvantage?
7. What is meant by "saturation" in the context of BJT operation in a logic circuit? Why is it desirable to avoid this mode of BJT operation for high speed circuits? Which of the following logic families are subject to this effect: (a) TTL; (b) LSTTL; (c) CMOS? Explain briefly.

CD54/74HC4040, CD54/74HCT4040

Functional Diagram

Problem 2



TRUTH TABLE

CP COUNT	MR	OUTPUT STATE
↑	L	No Change
↓	L	Advance to Next State
X	H	All Outputs Are Low

NOTE: H = High Voltage Level, L = Low Voltage Level, X = Don't Care, ↑ = Transition from Low to High Level, ↓ = Transition from High to Low.

Problem 1

Information:

Input equations for JK flip-flop:

$$\left\{ \begin{array}{l} J = \alpha \quad (\text{transitions where } J \text{ must equal } 1) \\ DC_J = \beta, 1, X \quad (\text{transitions for which we don't care what } J \text{ is}) \\ J = 0 \text{ otherwise.} \end{array} \right.$$

$$\left\{ \begin{array}{l} K = \beta \quad (\text{similarly for } K) \\ DC_K = \alpha, 0, X \end{array} \right.$$

For Prob. 3

; Program to read and store 20 ASCII characters and sum up any integers

```
xref    getchar, strout, decout, newline, stop

start:  lea    msg,a1    ①      ; put address of msg in a1
        move.w nchar,d1 ②      ; number of bytes to read
        clr   d2        ③      ; clear the register for the sum
        jmp  enter     ④      ; enter loop at end
loop:   jsr   getchar   ⑤      ; getchar puts the character in d0
        move.b d0,(a1)+ ⑥      ; move the character to memory
        jsr   addit     ⑦      ; subroutine to test and add integers
enter:  dbra  d1,loop   ⑧      ; subtract 1 from d1 and see if done
; now output the information
        lea    msg,a0    ⑨      ; set up for outputting character string
        move.w nchar,d0 ⑩      ;
        jsr   strout    ⑪      ; output the string
        jsr   newline   ⑫      ;
        move.w d2,d0    ⑬      ; output the sum
        jsr   decout    ⑭      ;
        jsr   newline   ⑮      ;
        jsr   stop      ⑯      ; end of program
; Subroutine addit tests ASCII characters to see if they represent numbers
; and if so, adds them to the sum in d2
addit:  and.b  #$7F,d0   ⑲      ; mask off parity bit of character
        cmp.b #$30,d0   ⑳      ; see if it is less than $30
        blt  skip      ㉑      ; if so, skip to return statement
        cmp.b #$39,d0   ㉒      ; see if it is greater than $39
        bgt  skip      ㉓      ; if so, skip to return statement
        and.w #$000F,d0 ㉔      ; get the number
        add.w d0,d2     ㉕      ; add to sum in d2
skip:   rts              ㉖      ; return from subroutine

data
msg:    ds.b  100       ㉗      ; set aside 100 bytes of storage
nchar:  dc.w  20        ㉘      ; number of characters to read
end
```

Branch Instructions and Signed Comparison Branches

Branch instructions (Bcc) check the condition codes, which take on meaning in the context of the instruction which set the codes. We will consider the case when the codes are set by execution of a CMP instruction, as in the program above. Some branch instructions and the condition codes they test are given in the next table. Note that the branch must be to a statement label, not an absolute address.

Instruction	Case for branch
BEQ <label>	Z=1
BNE <label>	Z=0
BMI <label>	N=1
BPL <label>	N=0
BVS <label>	V=1
BVC <label>	V=0
BCS <label>	C=1
BCC <label>	C=0

Some other variations of Bcc with more complex condition code tests are useful with CMP (or CMPI). The CMP instruction (as in `cmp.b <source>, <destination>`) compares two numbers by subtracting the destination from the source and setting the condition codes accordingly. For example, if the result is zero, the Z bit is set (see Sec. 1.1.4 of MPRM for details). The computer does this in such a way that the numbers themselves remain unchanged. Some other handy Bcc tests for use after a CMP instruction are:

Instruction	Case for branch	
BGE <label>	destination \geq source	(destination - source \geq 0)
BGT <label>	destination $>$ source	(destination - source $>$ 0)
BLE <label>	destination \leq source	(destination - source \leq 0)
BLT <label>	destination $<$ source	(destination - source $<$ 0)

These must be performed before another instruction changes the condition codes. The instruction descriptions tell which codes each affects. For example, MOVE affects N, Z, V and C. Conversely, MOVEA does not affect the condition codes at all, allowing addresses to be moved to address registers without affecting the results of a previous calculation.

Note the awkward inversion of the notation: BGE branches when *second* argument \geq *first* argument, for example.

Prob. 1(a)

116B Final 3/19/01
Solutions

FFB:

		$Q_A Q_B$			
D		00	01	11	10
0		2	β	β	2
1		2	β	β	2

Note that Q_B always "toggles!"
(trivial to implement)

J_B :

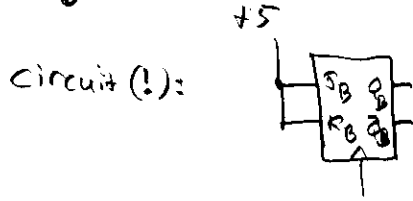
		$Q_A Q_B$			
D		00	01	11	10
0		1	X	X	1
1		1	X	X	1

$J_B = 1$

K_B :

		$Q_A Q_B$			
D		00	01	11	10
0		X	1	1	X
1		X	1	1	X

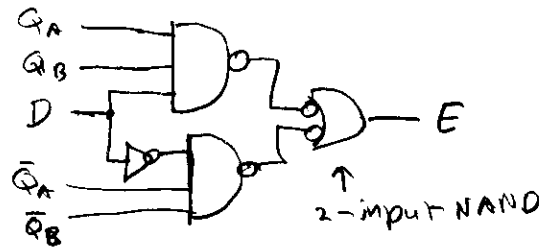
$K_B = 1$



(b) $E = Q_A Q_B D + \bar{Q}_A \bar{Q}_B \bar{D}$

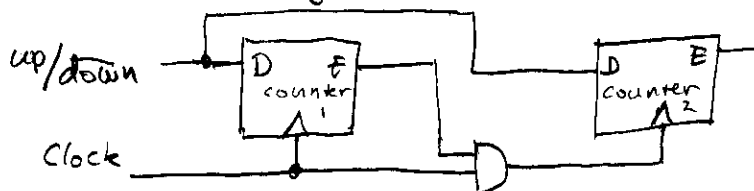
		$Q_A Q_B$			
D		00	01	11	10
0		1			
1				1	1

(no obvious simplification)



(c) Since there are no gate delays and the inputs don't change, FFB could run at the max. frequency, 30 MHz. Presumably, the requirements on FFA would limit the frequency at a lower value.

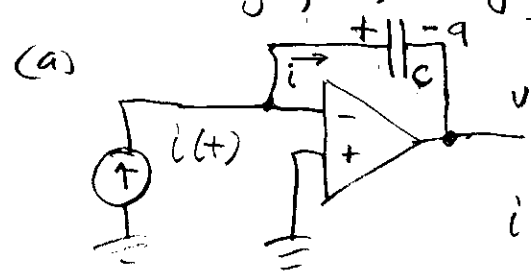
(d) Since there is no "enable" input, the E output will have to gate the clock for the next stage.



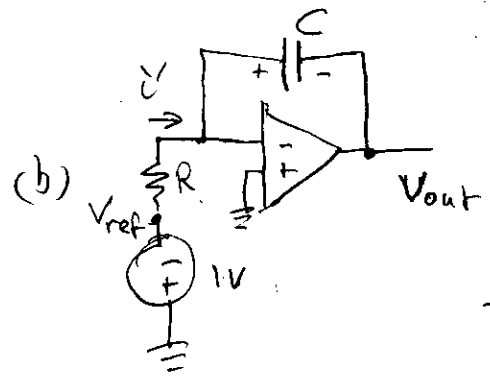
Note that the clock pulse width needs to be near the minimum to prevent spurious clocking of Counter 2.

2.

This circuit integrates the current pulse to produce a charge on the capacitor, C , then discharges the capacitor at a constant rate while running the counter. When the charge (and hence V_{out}) reaches zero, the counter is stopped. The number is thus proportional to the charge, Q . The gating of the clock pulses to the counter is shown below.



$$i = \frac{dq}{dt} = -C \frac{dV_{out}}{dt}$$

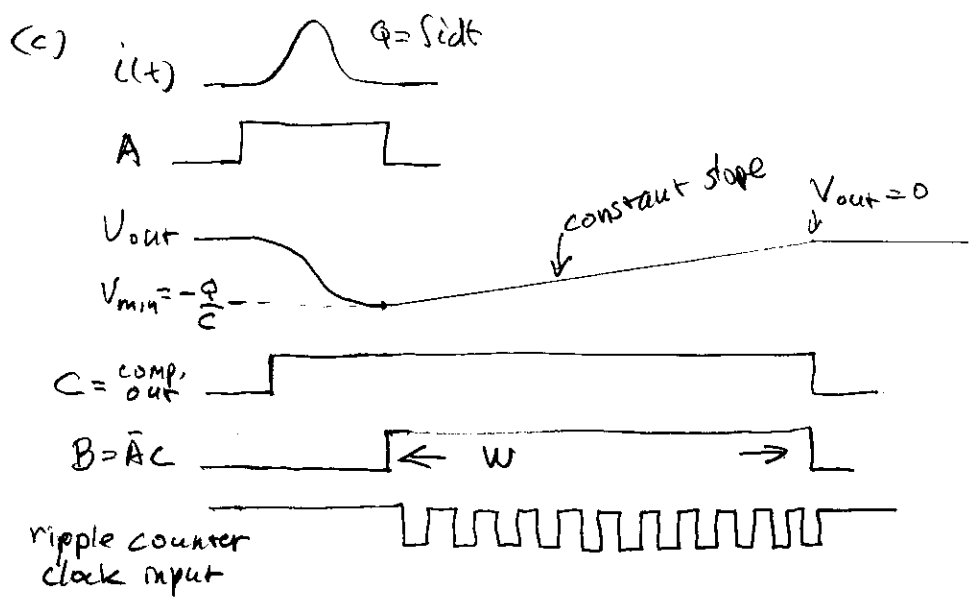


$$V_{out} = -\frac{1}{C} \int_0^T \frac{dq}{dt} dt = -\frac{Q}{C}$$

Use the result above. The - op amp input is virtual ground,

so
$$i' = V_{ref} / R = -C \frac{dV_{out}}{dt}$$

$$\frac{dV_{out}}{dt} = \frac{-V_{ref}}{RC} = \frac{-(-1V)}{2k\Omega \times 0.001\mu F} = 5 \times 10^5 \text{ V/s. (a const.)}$$



(c) (Continued)

When A goes high, the counter is reset to 0.
The current is integrated while A is high.

When A goes low, the charge on C, $Q = \int i dt$.

$V_{out} = -\frac{Q}{C}$ and the comparator output is high.

S_2 closes (\bar{A} comparator) and the capacitor discharges at a constant rate. The clock is also gated on by (\bar{A} comparator). Eventually

V_{out} reaches zero and the comparator output goes low. This causes S_2 to open and the counter to stop. The counter was gated on

for a time, $W = |V_{min}| / \frac{dV_{out}}{dt} = \frac{Q/C}{V_{ref}/RC} = \frac{QR}{V_{ref}}$.

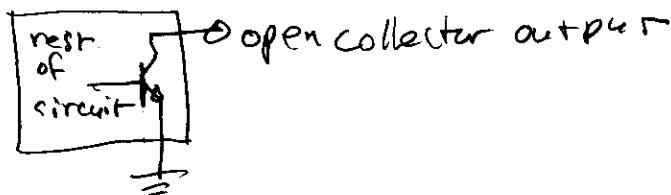
The number of counts is thus

$$N = f_{clock} W = \frac{f_{clock} R}{V_{ref}} Q = \text{const.} \times Q.$$

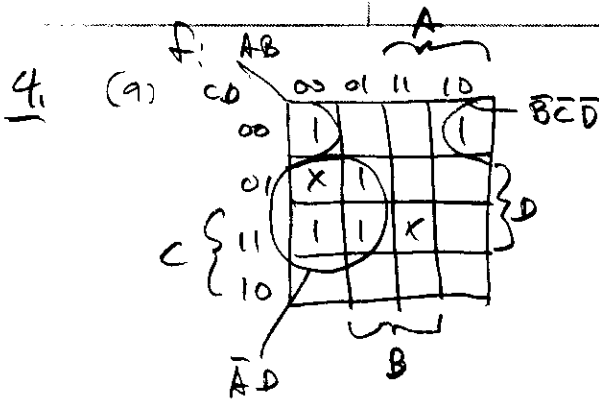
(d)

$$W = \frac{|V_{min}|}{\frac{dV_{out}}{dt}} = \frac{10V}{0.5V/\mu s} = \underline{\underline{20 \mu s}}$$

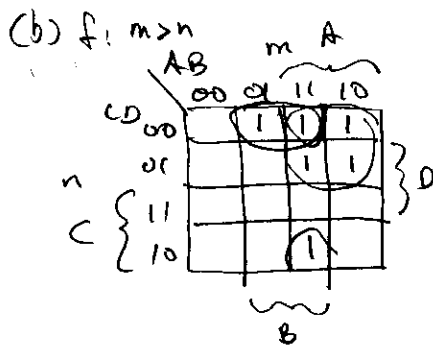
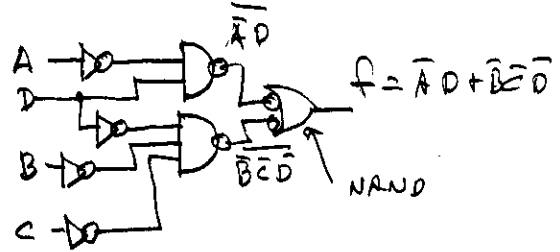
(e) The comparator has an open collector output. This means an external pull-up resistor connected to +5V is needed for it to produce a valid "high" output level.



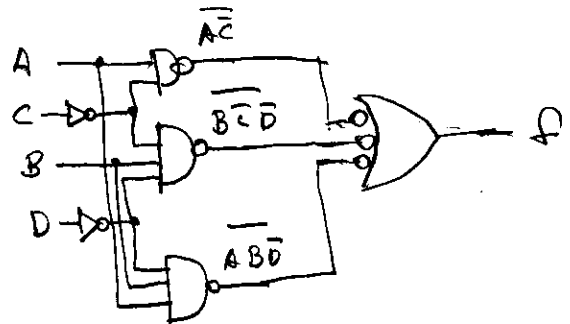

```
(i) addit: and.b    #$7F, d0  
          cmp.b    #$45, d0  
          beg      ltre  
          cmp.b    #$65, d0  
          beg      ltre  
          rts  
ltre:    add.w    #1, d2  
          rts
```



$$f = \bar{A}D + \bar{B}C\bar{D}$$



$$f = A\bar{C} + \bar{B}C\bar{D} + A\bar{B}\bar{D}$$



5. (a) The Macintosh system uses indirect addressing relative to register A5 for data.
- (b) A5 must not be changed by the user program.
- (c) Instructions are addressed indirectly relative to the contents of the program counter (PC).
- (d) This use of indirect addressing allows the resulting code and data to be placed anywhere in memory ("relocatable code") without changing the code. (see MMS manual, p. 97).
6. A sampling rate of 44 kHz gives a Nyquist critical frequency of 22 kHz ($f_c = f_s/2$). The signal to be digitized must be limited in bandwidth to frequencies less than this, but that is OK for audio signals. A 22 kHz sampling rate would require cutting off frequencies above 11 kHz, which would remove useful parts of the spectrum.
- 16 bit data allow a dynamic range of

$$R(\text{dB}) = 20 \log_{10} \frac{2^{16} - 1}{1} \approx 20 \log_{10} 2^{16} \approx 20 \times 16 \times 0.30 \text{ dB} = \underline{96 \text{ dB}}$$

\leftarrow number corresponding to largest voltage
 \uparrow number corresponding to smallest non-zero voltage

With 8 bits, $R(\text{dB}) \approx 20 \log_{10} 2^8 = \underline{48 \text{ dB}}$.

The useful dynamic range of an orchestra in a concert hall is approx. 60 dB. Thus for realistic reproduction of audio signals, 22 kHz and 8 bits are not enough. They may be adequate in less critical applications. The advantage is that the 8 bit 22 kHz digitization requires less storage space and can be done with simpler equipment.

7. Saturation is the region of operation of a BJT where the BE and BC junctions are both forward biased. A transistor in saturation requires a long time to turn off since the base region is flooded with minority carriers. Schottky transistors are used in LSTTL circuits to prevent the transistors from entering the saturation region. CMOS circuits use MOSFETs rather than BJTs so they are not subject to this phenomenon. Switching transistors in ordinary TTL circuits are subject to this effect. So the answer to the last part is (a) TTL.