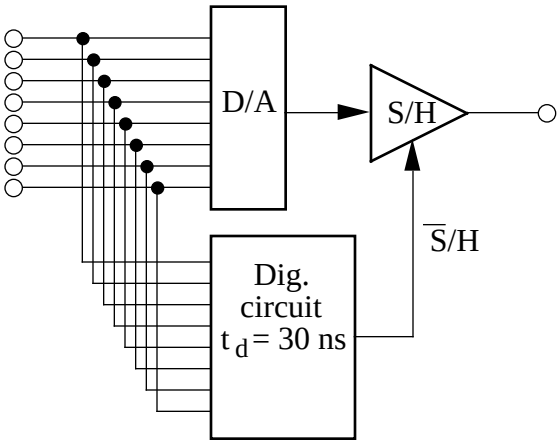
Solutions for Midterm - EECS 145M Spring 1996

PROBLEM 1

1a

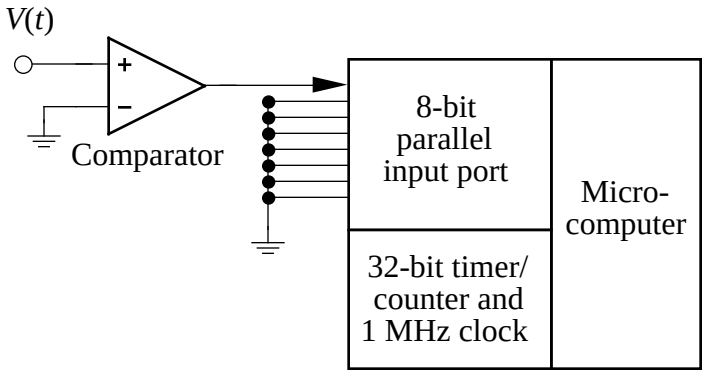


1b

When input changes, change detector circuit generates a pulse from 5 ns to 35 ns (t_d = 25 ns was also allowed). This puts S/H into hold mode during the glitch. Droop during 30 ns is only 3 µV.
[2 points off for 20 ns, which puts the S/H into its hold mode from 5 ns to 25 ns]
[5 points off if digital change is not detected (the purpose of circuit #1)]
[5 points off for omitting the D/A]

PROBLEM 2

2a



2b

```
long i1, i2, i3, i4;
double time, counter, freq;
while(inportb(2) == 1);          /* wait until comparator goes low */
while(inportb(2) == 0);          /*then high */
outportb(1,0);                   /*clear timer at first low-high edge */
time = 0;
counter = 0;
while (time < 1000000) {
    while(inportb(2) == 1);      /* wait until comparator goes low */
    while(inportb(2) == 0);      /*then high */
```

```
    outportb(1,1);              /* latch timer at low-high edge*/
    i1 = inportb(1);
    i2 = inportb(1);
    i3 = inportb(1);
    i4 = inportb(1);
    time = (i4 << 24 | i3 << 16 | i2 << 8 | i1);      /* time in µs */
    counter =+ 1.;
    freq = counter/time;        /* compute frequency*/
    time =+ 1/freq;             /* estimate time of next low-high edge */
    }
freq =* 1000000.                /* convert to Hz */
```

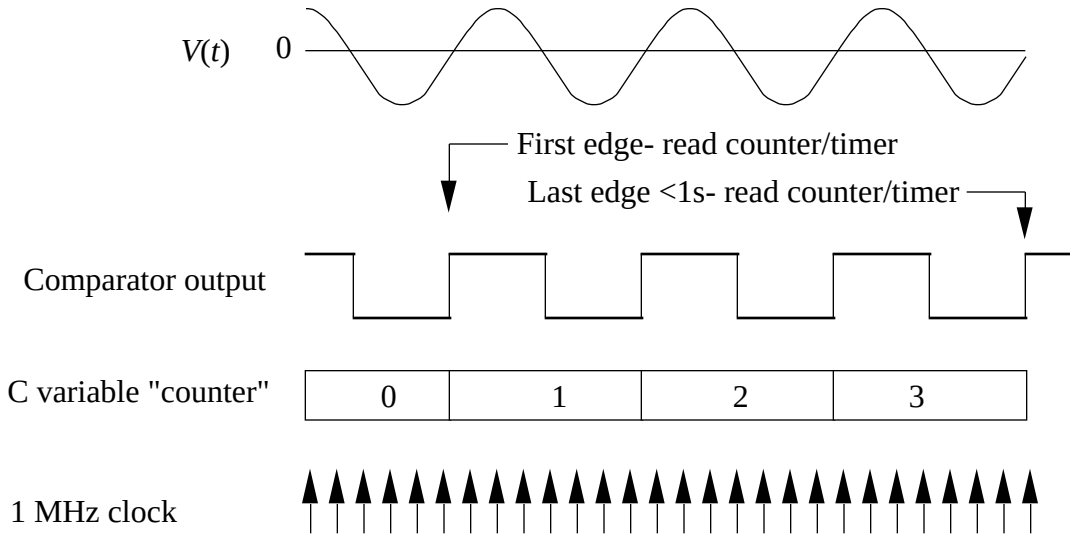
This code sets the timer to zero at the first low-high edge of the comparator output, and then latches the timer at each subsequent low-high transition. The frequency is computed as the number of low-high edges divided by the elapsed time. Note that the while loop quits when the estimated time of the next low-high edge exceeds 1 second. If the f > 1 Hz, the measurement will never take longer than 1s. If the f < 1 Hz, f will be measured, but the measurement will take a time = 1/f.

For full credit, it was essential to do the following:

1. **Detect and record (or zero the counter/timer) the first time that the comparator changed**
2. **Detect and record the last time that the comparator changed in the entire 1 s measurement period**
3. **Record the number of times that the comparator changed between steps 1 and 2**
4. **Compute the frequency as #3/(#2-#1)**

[8 points off for $f = (1/2\pi) \sin^{-1}$(comparator output)]
[8 points off for reading the comparator output once a second and not counting the number of cycles]
[3 points off for measuring the number of zero crossings in 1 s- this method is very inaccurate at low frequency]
[5 points off for only measuring the time for one cycle- this method is very inaccurate at high frequency]

2c



2d

Lowest frequency that can be measured is 1 s is 1 Hz. (While the code will measure lower frequencies, it will take longer than the 1 s required measurement period).
[2 points off for a minimum frequency of 0 Hz]

Since the edge detection loop requires 7 µs per cycle, the highest frequency is 143 kHz.

2e

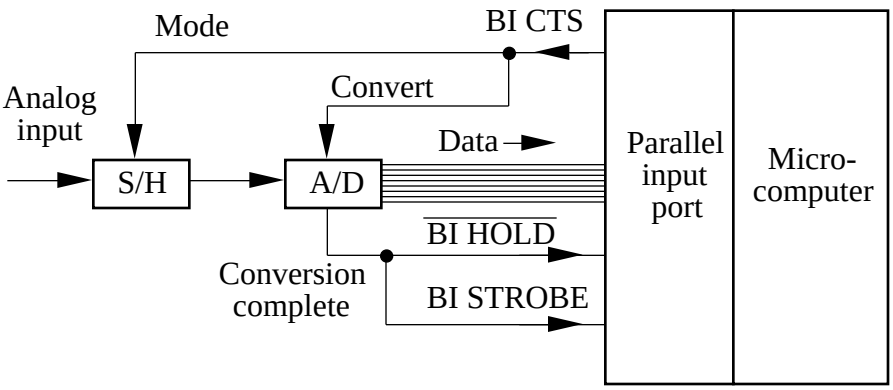
The computer "while" loop detects the first low-high edge and zeros the timer with ±1 µs time jitter because the computer clock and the high-low edge are running independently (not synchronized). The last high-low edge similarly has ±1 µs jitter. The time difference between the two events has ±2 µs jitter.

Minimum 1 Hz measurement can range from (1 count) / (0.999998 s) to (1 count) / (1.000002 s) or 1 Hz ± 2×10^{-6} Hz
Maximum 143 kHz can range from (143,000 counts) / (0.999998 s) to (143,000 counts) / (1.000002 s) or 143 kHz ± 0.28 Hz.

Note: a code that individually detected and counted the low-high and high-low edges was also acceptable, and the corresponding minimum and maximum frequencies were 0.5 Hz ± 10^{-6} Hz and 71 kHz ± 0.14 Hz.

PROBLEM 3

3a



3b

BI CTS, BI STROBE, and $\overline{\text{BI HOLD}}$ are normally low
1 Program makes BI CTS high
2 BI CTS high puts S/H into hold mode
3 BI CTS low-high edge begins A/D conversion (and promptly sets BI STROBE and $\overline{\text{BI HOLD}}$ high).
4 When A/D ends conversion, it makes $\overline{\text{BI HOLD}}$ and BI STROBE low
5 $\overline{\text{BI HOLD}}$ low puts parallel input port registers into hold mode
6 When program detects BI STROBE low, it reads the input port registers
7 Program makes BI CTS low to prepare for next conversion

Letting $\overline{\text{BI HOLD}}$ float high (transparent mode) was also OK.

The use of $\overline{\text{BI HOLD}}$ would be important in the case where is not certain whether the A/D holds the converted data on its output lines long enough for the computer to read them.

[4 points off for keeping the S/H amplifier in hold mode between new data and in sample mode during conversion- firstly, the A/D will be converting old, droopy data and secondly, if the analog data did not need to be held constant during A/D conversion, the S/H would not be needed at all]

Midterm #1 class statistics:

Problem	max	average	rms
1	25	19.0	6.8
2	50	25.7	10.4
3	25	16.7	5.4
total	100	61.3	18.5

Grade distribution:

Range	number	*approximate* letter grade
1-10	0	
11-20	1	F
21-30	0	
31-40	1	F
41-50	2	D
51-60	6	C
61-70	3	B
71-80	4	A-B
81-90	2	A
91-100	1	A+