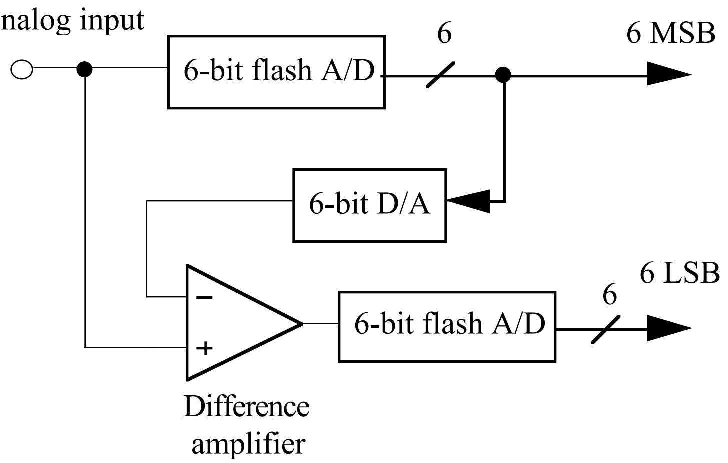
Solutions for Midterm #1 - EECS 145M Spring 2002

1a



[1 point off for difference amplifier input signs not correct or not given]

[5 points of for omitting the difference amplifier; connecting the D/A output to the second A/D just generates the same 6 most significant bits]

1b

1 Use one 6-bit flash A/D converter to determine the 6 most significant bits

2 Use the 6-bit D/A converter to convert the 6 most significant bits into an analog voltage

3 Use a difference amplifier to subtract the 6-bit D/A output from the analog input

4 Use the second 6-bit flash A/D converter to convert the analog difference from step 3 into the 6 least significant bits

1c An N-bit flash converter has 2^N resistors in its resistor ladder. So a 6-bit flash converter has $2^6 = 64$ resistors

[2 points off for 6 or 12]

1d An N-bit R-2R D/A converter has 2N resistors in its resistor ladder. So a 6-bit R-2R D/A converter has 2N = 12 resistors plus one or two additional.

[1 point off for 6; two points off for 3, 4, or 64]

1e The 6-bit A/D converter that determines the 6 most significant bits and the 6-bit D/A converter that converts these into an analog voltage must be so accurate that the output of the difference amplifier is accurate to 1/8 step size out of $2^{12} = 4096$ steps or one part in 32k. Since their accuracy is determined by the accuracy of their resistors, a resistor accuracy of 1 part in 32k would be safe.

Here is an example: Imagine that the first stage has 64 steps each 64 mV wide for a total of 4096 mV and the second stage has 64 steps each 1 mV wide. Since the entire 12-bit converter has 1 mV steps and an absolute accuracy of 1/8 LSB, the transition voltages of the first stage must be accurate to 1/8 mV. As a worst case, imagine that the first 32 resistors of the first stage have resistance R + e and the second 32 resistors have resistance R – e, where e is a resistance error. The voltage at the center point of the resistor string of the first stage would then be 4096 mV [32 (R + e) / 64R], which would be 2048 mV plus a error of 2048 mV (e/R) < 1/8 mV. This means that e/R < 1/16,000 and the resistors must be accurate to 1 part in 16k. Note that statistical analysis does not apply if the errors are systematic.

The 6-bit A/D converter that determines the 6 least significant bits only needs to have the accuracy of a 6-bit converter, so the resistors need an accuracy of one part in 2^9, or one part in 512.

[Full credit for resistor accuracy 1 part in 32k or 1 part in 16k]

[2 points off for "very accurate"; 3 points off for 1/8 LSB or 1/16 LSB; 4 points off for 1/2 LSB or 1/4 LSB; 4 points off for > 1%]

[asking for a resistor accuracy of "1/8 LSB" will not be meaningful to a resistor manufacturer]

1f We require that the input change by less than 1/2 LSB in 100 ns, so the formula gives:

$$f_{\max} = \frac{1}{\pi T 2^{N+1}} = \frac{1}{\pi(100\,ns)(8192)} = 389 \text{ Hz}$$

[3 points off for N = 6; 3 points off for N = 1 or 3]

[400 Hz was accepted for full credit]

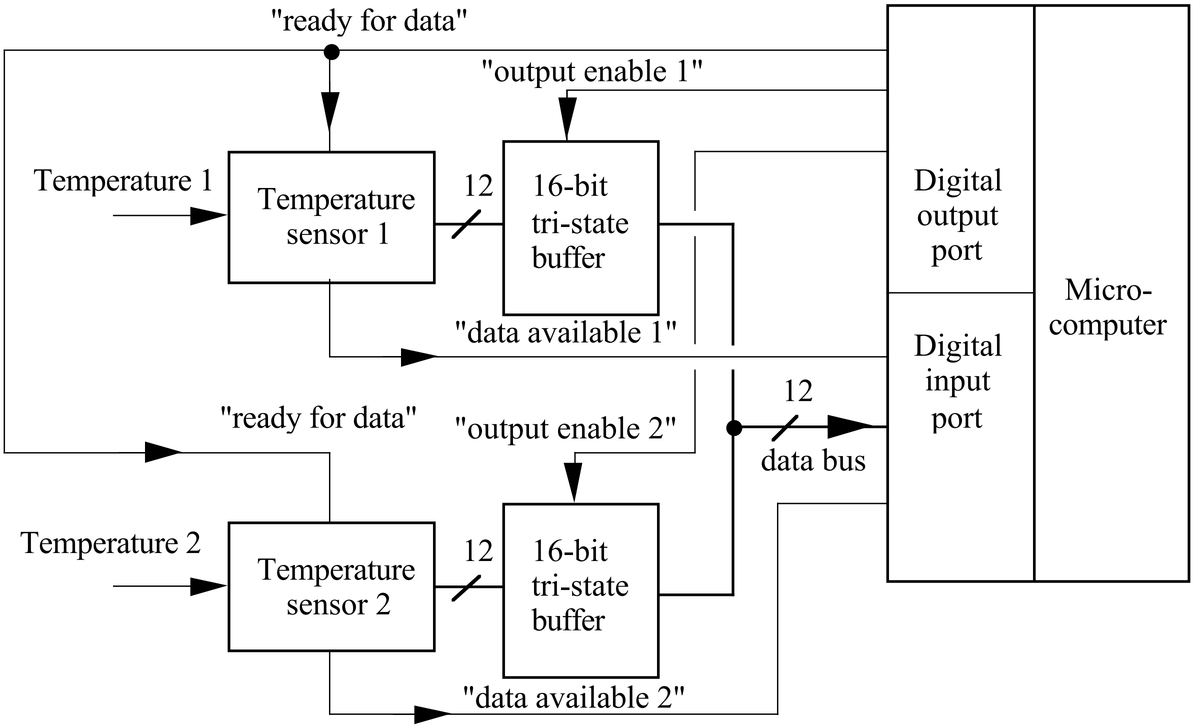
1g We require that the input change by less than 1/2 LSB in 100 ps, so the formula gives:

$$f_{\max} = \frac{1}{\pi T 2^{N+1}} = \frac{1}{\pi(100\,ps)(8192)} = 389 \text{ kHz}$$

[3 points off for T = 100 ns + 100 ps; the whole point of the S/H is to maintain a constant input for 100 ns]

[400 kHz was accepted for full credit]

2a



[5 points off for 24 data lines into input port]

2b

1 The program outputs a word to the digital output port that has a one on the "ready for data" bit, and ones on both output enable 1 and 2 to ignore the sensors.

2 The two temperature sensors simultaneously start producing a digital output

3 The program reads the digital input port in a loop waiting for "output data available n", n = 1 or 2 to go high (Note that these two signals do not need tri-state buffers and can be read directly to constantly monitor both temperature sensors)

4 If sensor n (n = 1 or 2) is ready first, it makes "output data available n" high

5 The program detects the "output data available n" signal and writes a word to the digital output port with one on the "ready for data" bit, a zero on the "output enable n" bit to select sensor n, and a one on the "output enable 3-n" bit to ignore sensor 3-n.

(Note: If n = 1, 3-n = 2. If n = 2, 3-n = 1.)

6 The program reads the input port for the sensor n data

7 The program masks the data to produce the 12-bit value from temperature sensor n

8 The program continues to read the digital input port in a loop waiting for the "output data available 3-n" bit to go high

9 When sensor 3-n is ready, it brings "output data available 3-n" high

10 The program detects the "output data available 3-n" and writes a word to the digital output port that has a one on the "ready for data" bit, a one on the "output enable n" bit to ignore sensor n, and a zero on the "output enable 3-n" bit to select sensor 3-n.

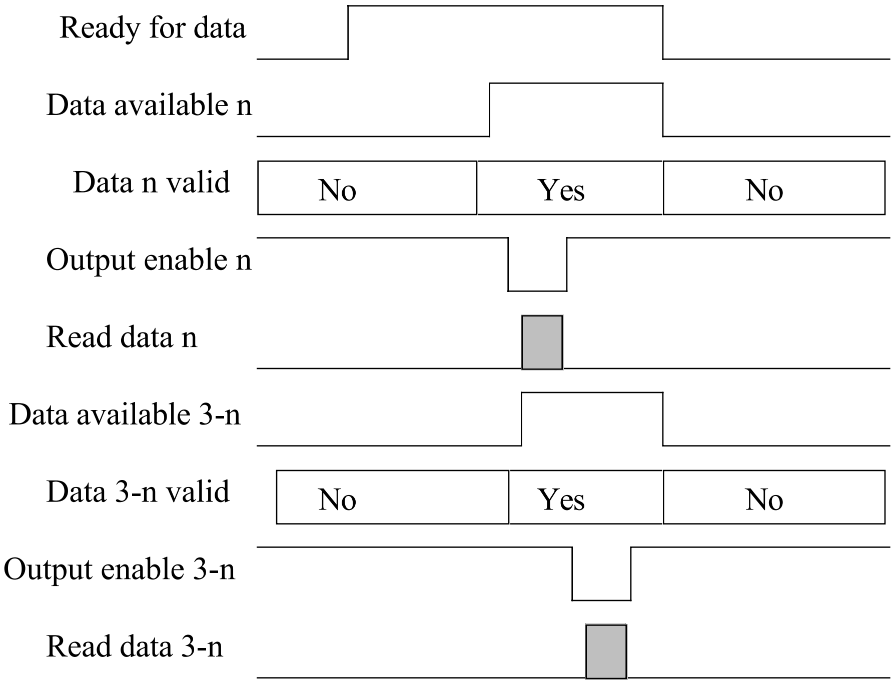
11 The program reads the input port for the sensor n data

12 The program masks the data to produce the 12-bit value from temperature sensor n

13 The program writes a word to the digital output port that brings "ready for data" low

[2 points off if second DA signal not sensed]

2c



Midterm #1 class statistics:

Problem	max	average	rms
1	50	35.1	8.6
2	50	47.4	6.3
total	100	82.5 (B+)	12.7

Grade distribution:

Range	number	*approximate* letter grade
51-55	1	D
56-60	0	D+
61-65	0	C–
66-70	2	C
71-75	0	C+
76-80	2	B
81-85	1	B+
86-90	1	A–
91-95	5	A
96-100	0	A+