Jonathan W. Valvano, April 18, 1:00pm-1:50pm. Open book, open notes, calculator

(20) Question 1. Review the attached data sheets for the 9S12DP512.

(10) Part a) There are two possible answers, depending on how you get to the start of the interval:

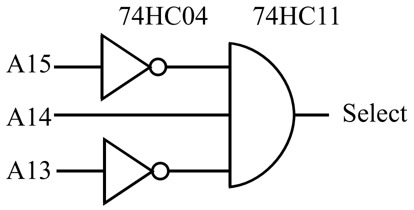
Read data available is (t_2-t_{10}, t_2+t_{11}) = (40-13, 40+0) = (27, 40) in ns
Read data available is (t_3+1ns+t_{16}, t_2+t_{11}) = (19+1+6, 40+0) = (26, 40) in ns

(10) Part b) There are two possible answers, depending on how you get to the start of the interval:

Write data required is (t_3+1ns+t_{12}, t_2+t_{13}) = (19+1+7, 40+2) = (27, 42) in ns
Write data required is (t_2-t_{14}, t_2+t_{13}) = (40-12, 40+2) = (28, 42) in ns

(10) Question 2. Fully-decoded means Select if and only if the address is $4000 to $5FFF.

1) write $4000-$5FFF as 010X,XXXX,XXXX,XXXX
2) give logic equation directly using all 0's and 1's,
Select = not(A15)*A14*not(A13)
3) Show digital logic (chip numbers are in front cover of the book)



(25) Question 3.

Part a)

Data available starts at **t_9** = 30ns (from SPI timing)
Data required starts at $\frac{1}{2}$**t_1**-**t_2** = $\frac{1}{2}$**t_1**- 200ns
To work, we need 30ns ≤ $\frac{1}{2}$**t_1**- 200ns or 460ns ≤ **t_1**

Part b) Clock out on rising edge CPHA=1, CPOL=0 or CPHA=0, CPOL=1

Part c) Write a C function that writes an 8-bit digital number to the shift register

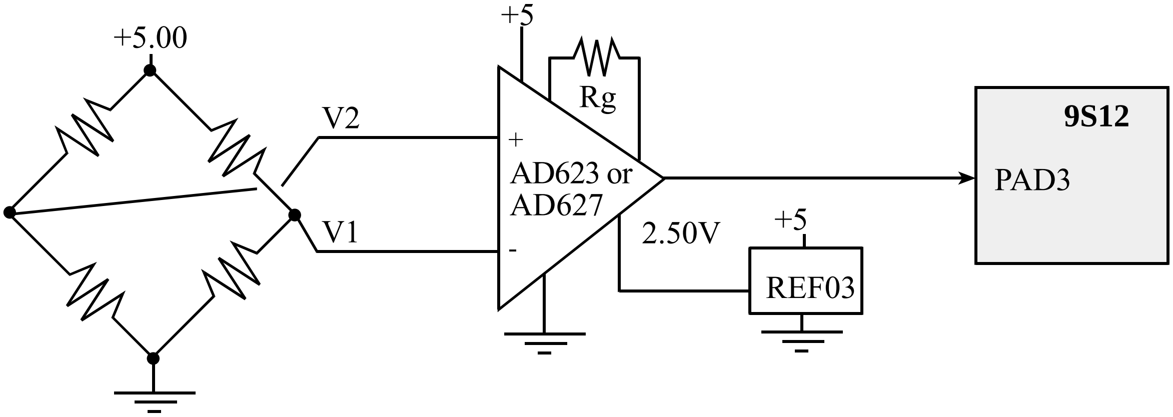
```
void SPI_Out(unsigned char data){  unsigned char dummy
  while(((SPISR&0x20)==0)){};// 1) wait for SPTEF=1,
  SPIDR = data;             // 2) output to start SPI
  while(((SPISR&0x80)==0)){};// 3) wait for SPIF=1,
  dummy = SPIDR;            // 4) read result, clear SPIF
}
```

(45) Question 4. You will design a data acquisition system to measure force.

Part a) The sampling rate must be greater than 198 Hz according to Nyquist Theorem. I choose 200 Hz.

Part b) To match the full scale force range into the full scale ADC range, I need a gain of 5V/0.5V = 10. I also need a 2.5 V offset so the output remains in the 0 to +5V range. I can use either the AD623 or the AD627, both are rail to rail and have the offset feature. The 2.5V offset is created by a REF03.

V_{out} = 10*(V_2-V_1) + 2.5
(AD623) Gain = 1+(100kΩ/R_g), so R_g is 100kΩ/9 = 11.1kΩ
(AD627) Gain = 5+(200kΩ/R_g), so R_g is 200kΩ/5 = 40kΩ



Part c) Resolution, range/precision, is 20N/1024, which is about 0.02N.

Part d) Write the entire program that implements this real time data acquisition system. I will choose a 16-bit signed decimal fixed-point system with resolution of 0.01N. The conversion is linear, mapping 0 to -1000, and 1023 to 1000. For example, if the force is 0.0 N, the bridge (V2-V1) voltage will be 0.0V, the PAD3 voltage will be 2.5V, the ADC conversion will be 512, and the output of the function will be 000. Overflow is handled by promoting to 32-bits. Any of these equations is ok

(1000*(data-512))/511
(1000*(data-512))/512
(1000*(data-512)+256)/511 (implements rounding)
(1000*(data-512)+256)/512 (implements rounding)
(125*(data-512))/64
(125*(data-512)+32)/64 (implements rounding)

```
short Convert(unsigned short data){ long force;
  force = (1000*((short)data-512) + 256)/512;
  return (short) force;
}
short force;             // fixed point, units 0.01N
void main(void){
  PLL_Init();            // initializes the PLL, changing the E clock to 24 MHz
  ADC_Init();            // initializes the ADC
  TIOS |= 0x20;          // channel 5 is output compare
  TSCR1 = 0x80;          // activate timer
  TSCR2 = 0x05;          // TCNT at 24MHz/32 = 750 kHz
  TIE |= 0x20;           // arm channel 5
  TC5 = TCNT+50;         // interrupt right away
  asm cli                // enable
  for(;;){
  }
}
interrupt 13 void TC5handler(void){ // executes at 200 Hz, 50ms
unsigned short data;     // ADC sample 0 to 1023
  TFLG1 = 0x20;          // acknowledge OC5
  TC5 = TC5+3750;        // 5 ms (24,000,000/32/3750 = 200 Hz)
  data = ADC_In(0x83);   // returns 10-bit sample from channel 4, 0 to 1023
  Force = Convert(data); // -1000 to +1000, meaning -10.00 to 10.00N
}
```