(5) Question 1. \$A5 = 10100101 = -128+32+4+1 = -128+37 = -91

(5) Question 2. 0-9 is one decimal digital, and 0-3 is ¾ of a decimal digit.
Part a) This system implements 4¾ decimal digits.
Part b) 2^{16}=65536 and 2^{15}=32768. So, 40,000 alternatives would require at least 16 ADC bits.

(5) Question 3. For the overflow (V) bit, convert both to signed and add -1+1=0. Since it fits, V=0.
For the carry (C) bit, convert both to unsigned and add 255+1=256. Since it doesn't fit, C=1.
Since the result is zero, Z=1.

(5) Question 4. This interface works because the LED current of 1 mA is less than the maximum output current of the 9S12, which is 1.6 mA. The current in the LED is controlled by the resistor. When the LED is on, there will be 1.9 V across the LED, so there will be 4.9-1.9= 3V across the resistor. To get an LED current of 1mA, set R = (4.9-1.9)/1mA = 3 kΩ.

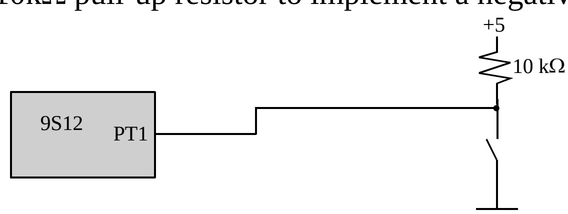
(10) Question 5. The return address of \$403C is pushed on the stack big endian, then the value of Register A is pushed \$0A. SP points to the top

```
SP ->  $3FFD  $0A
       $3FFE  $40
       $3FFF  $3C
```

(10) Question 6. The first three cycles fetch the entire instruction, the next two cycles fetch the 16-bi data at memory \$0812.

R/W	Addr	Data	Changes to D,X,Y,S,PC,IR,EAR
R	\$5000	\$B3	IR=\$B3, PC=\$5001
R	\$5001	\$08	PC=\$5002
R	\$5002	\$12	EAR=\$0812, PC=\$5003
R	\$0812	\$00	
R	\$0813	\$00	(RegD is not changed)

(10) Question 7. We use a 10kΩ pull-up resistor to implement a negative logic switch interface.




```
;Solution to Question 8
     ldd  Sg        ;needs 16-bit reg
     cpd  #100      ;needs signed branch
     blt  skip      ;skip if Sg<100
     bclr PTT,#$03
skip
```

```
ReadPorts  ;solution to Question 9
  ldab PTT ; ? , ? , ? , ? ,PT3,PT2,PT1,PT0
  lslb
  lslb
  lslb
  lslb      ;PT3,PT2,PT1,PT0, 0 , 0 , 0 , 0
  ldaa PTP  ; ? , ? , ? , ? ,PP3,PP2,PP1,PP0
  anda #$0F ; 0 , 0 , 0 , 0 ,PP3,PP2,PP1,PP0
  aba       ;PT3,PT2,PT1,PT0,PP3,PP2,PP1,PP0
  rts
```