ECE/CS 5780/6780: Embedded System Design

Chris J. Myers

Review 1

Chapter 1 Topics

- Embedded microcomputer architecture
- I/O ports
- 6812 architecture
- Digital representations of numbers
- Addressing modes (INH, IMM, DIR, EXT, REL)
- Top-down and bottom-up design process

Chapter 2 Topics

- Assembly language basics
- Several types of indexed addressing modes
- 6812 assembly language and pseudo-ops
- Coding style, naming conventions, and comments
- FSM abstraction
- Modular software development
- Global and local variables
- Layered software systems
- Device drivers
- Debugging
- Power of 10

Chapter 3 Topics

- Blind cycle synchronization
- Gadfly synchronization

Section 8.1 Topics

- Interfacing a switch
- Hardware and software debouncing methods
- Interfacing to 4 by 4 scanned keypads

Chapter 4 Topics

- Basics of interrupts and ISRs
- Reentrant programming
- FIFOs
- 6812 interrupts
- Polled and vectored interrupts
- Priority
- Real-time interrupts and periodic polling

Question 1(a)

- What are pseudo instructions and what are they used for? Give two examples.

Question 1(a)

- What are pseudo instructions and what are they used for? Give two examples.
- ANSWER: They are not actual machine instructions rather they are instructions used by the assembler to assist in memory allocation. Examples include `org` to set the current memory address, `rmb` to reserve memory, `fcb` to assign a constant byte to memory, etc.

Question 1(b)

- When should you not use global variables? When must you use global variables? Be sure to explain your answer.

Question 1(b)

- When should you not use global variables? When must you use global variables? Be sure to explain your answer.
- ANSWER: You should not use global variables in subroutines as it makes them non-reentrant. You must use global variables to share data between a main thread and an ISR because they do not share any other state.

Question 2

- Consider the following assembly code. You may assume that the subroutine `OUTSTR` when called sends the string whose address is stored in the global variable `CSTR` to a serial output device. You may also assume that this subroutine is located at the address $E200. Note that PORTC bit 7 is an input.

Question 2(a)

```
       org   $0000                      org   $E000
PORTC  equ   $1003               MAIN   lds   #$00FF
CSTR   rmb   2                          ldx   IS
       org   $B600               LOOP   ldy   OUTP,X
OUTP   equ   0                          sty   CSTR
Next0  equ   2                          jsr   OUTSTR
Next1  equ   4                          ldaa  PORTC
ODD    fcc   ‘‘odd’’                    bita  #$80
EVEN   fcc   ‘‘even’’                   bpl   IS0
IS     fdb   SE                  IS1    ldx   Next1,X
SE     fdb   EVEN                       bra   LOOP
       fdb   SE                  IS0    ldx   Next0,X
       fdb   SO                         bra   LOOP
SO     fdb   ODD                        org   $FFFE
       fdb   SO                         fdb   MAIN
       fdb   SE
```

- What does this program do?

Question 2(a)

```
      org   $0000                     org   $E000
PORTC equ   $1003               MAIN  lds   #$00FF
CSTR  rmb   2                           ldx   IS
      org   $B600               LOOP  ldy   OUTP,X
OUTP  equ   0                           sty   CSTR
Next0 equ   2                           jsr   OUTSTR
Next1 equ   4                           ldaa  PORTC
ODD   fcc   ''odd''                     bita  #$80
EVEN  fcc   ''even''                    bpl   IS0
IS    fdb   SE                  IS1   ldx   Next1,X
SE    fdb   EVEN                        bra   LOOP
      fdb   SE                  IS0   ldx   Next0,X
      fdb   SO                          bra   LOOP
SO    fdb   ODD                         org   $FFFE
      fdb   SO                          fdb   MAIN
      fdb   SE
```

- What does this program do?
- ANSWER: It is a FSM which which samples bit 7 of Port C and reports whether it has seen a 1 an even or odd number of times on that pin.

Question 2(a)

```
      org   $0000                     org   $E000
PORTC equ   $1003               MAIN  lds   #$00FF
CSTR  rmb   2                           ldx   IS
      org   $B600               LOOP  ldy   OUTP,X
OUTP  equ   0                           sty   CSTR
Next0 equ   2                           jsr   OUTSTR
Next1 equ   4                           ldaa  PORTC
ODD   fcc   ''odd''                     bita  #$80
EVEN  fcc   ''even''                    bpl   IS0
IS    fdb   SE                  IS1   ldx   Next1,X
SE    fdb   EVEN                        bra   LOOP
      fdb   SE                  IS0   ldx   Next0,X
      fdb   SO                          bra   LOOP
SO    fdb   ODD                         org   $FFFE
      fdb   SO                          fdb   MAIN
      fdb   SE
```

- Assemble this code into object code.

Question 2(b)

```
      org   $0000
PORTC equ   $1003
CSTR  rmb   2
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																

Question 2(b)

```
      org   $B600
OUTP  equ   0
Next0 equ   2
Next1 equ   4
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00																
B6 10																

Question 2(b)

```
ODD   fcc   ''odd''
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00																
B6 10																

Question 2(b)

```
ODD   fcc   ''odd''
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d													
B6 10																

Question 2(b)

```
EVEN    fcc    ''even''
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d													
B6 10																

Question 2(b)

```
EVEN    fcc    ''even''
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n									
B6 10																

Question 2(b)

```
IS      fdb    SE
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	IS								
B6 10																

Question 2(b)

```
IS      fdb    SE
SE      fdb    EVEN
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	IS		SE						
B6 10																

Question 2(b)

```
IS      fdb    SE
SE      fdb    EVEN
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	IS		B6 SE	03					
B6 10																

Question 2(b)

```
IS      fdb    SE
SE      fdb    EVEN
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03					
B6 10																

Question 2(b)

```
        fdb    SE
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03					
B6 10																

Question 2(b)

```
        fdb    SE
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09			
B6 10																

Question 2(b)

```
        fdb    SO
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09			
B6 10																

Question 2(b)

```
        fdb    SO
SO      fdb    ODD
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09			S0
B6 10																

Question 2(b)

```
        fdb    SO
SO      fdb    ODD
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09			B6 S0
B6 10	00															

Question 2(b)

```
        fdb    SO
SO      fdb    ODD
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00															

Question 2(b)

```
        fdb    S0
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00															

Question 2(b)

```
        fdb    S0
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF													

Question 2(b)

```
        fdb    SE
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF													

Question 2(b)

```
        fdb    SE
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											

Question 2(b)

```
        org    $E000
MAIN    lds    #$00FF
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00																
E0 10																
E0 20																
E0 30																
FF F0																

Question 2(b)

```
        org    $E000
MAIN    lds    #$00FF
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF													
E0 10																
E0 20																
E0 30																
FF F0																

Question 2(b)

```
      ldx   IS
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF													
E0 10																
E0 20																
E0 30																
FF F0																

Question 2(b)

```
      ldx   IS
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09										
E0 10																
E0 20																
E0 30																
FF F0																

Question 2(b)

```
LOOP  ldy   OUTP,X
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09										
E0 10																
E0 20																
E0 30																
FF F0																

Question 2(b)

```
OUTP  equ   0
LOOP  ldy   OUTP,X
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00								
E0 10																
E0 20																
E0 30																
FF F0																

Question 2(b)

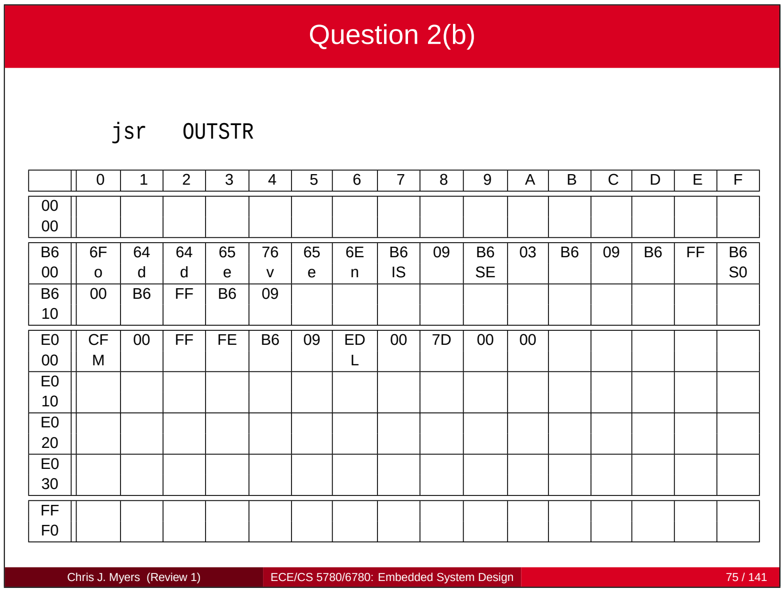
```
      sty   CSTR
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00								
E0 10																
E0 20																
E0 30																
FF F0																

Question 2(b)

```
      org   $0000
CSTR  rmb   2
      sty   CSTR
```

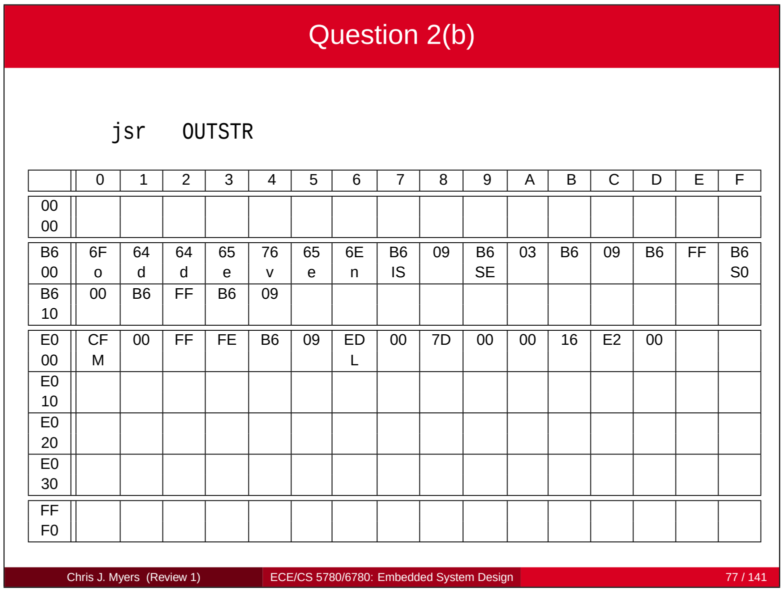
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00					
E0 10																
E0 20																
E0 30																
FF F0																



Question 2(b)

```
jsr   OUTSTR
```

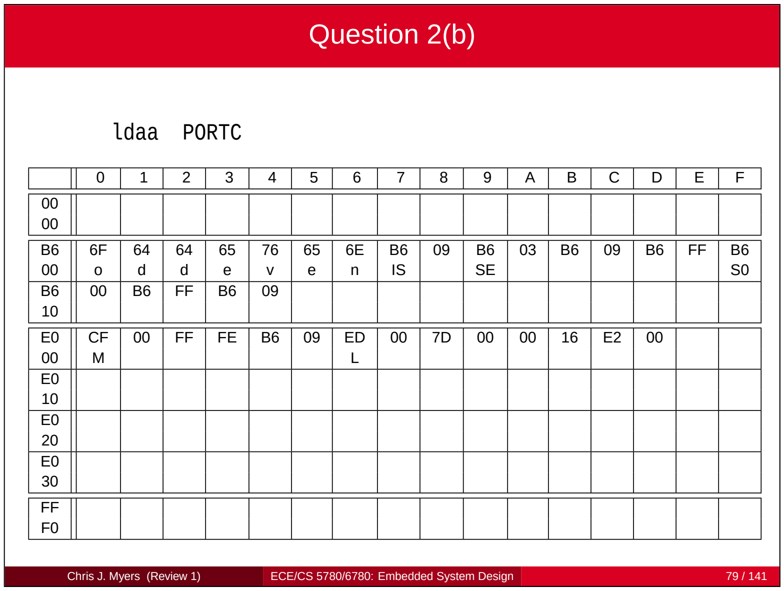
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00					
E0 10																
E0 20																
E0 30																
FF F0																



Question 2(b)

```
jsr   OUTSTR
```

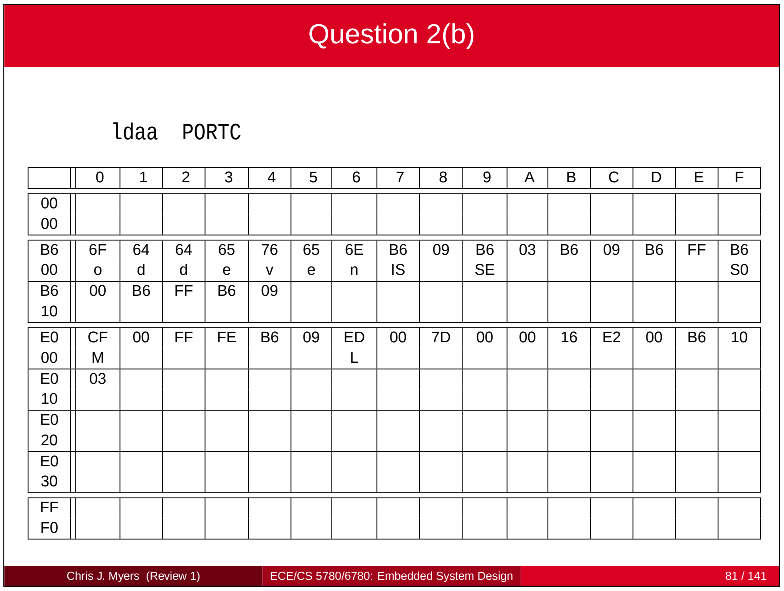
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00		
E0 10																
E0 20																
E0 30																
FF F0																



Question 2(b)

```
ldaa  PORTC
```

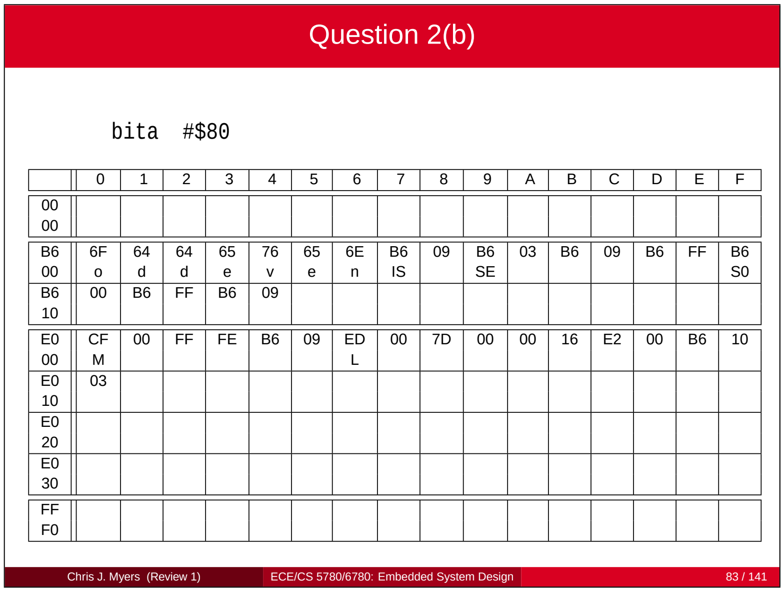
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00		
E0 10																
E0 20																
E0 30																
FF F0																



Question 2(b)

```
ldaa  PORTC
```

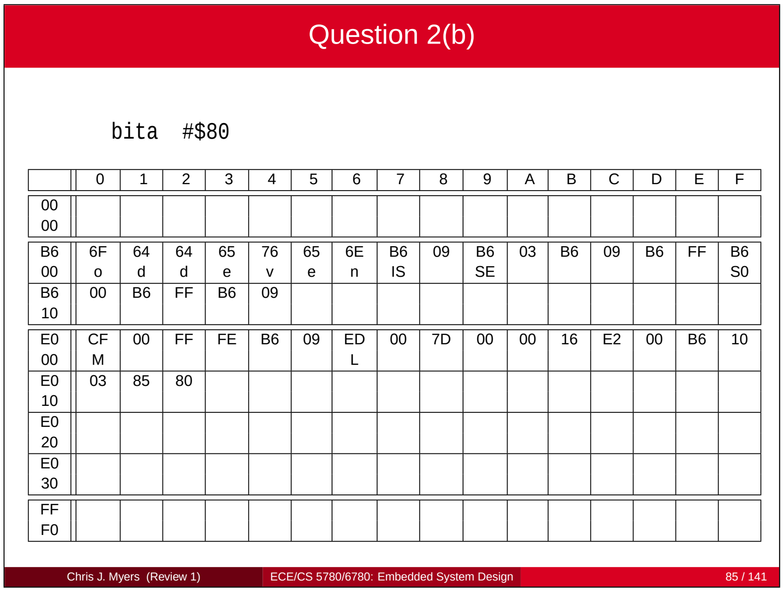
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03															
E0 20																
E0 30																
FF F0																



Question 2(b)

```
bita  #$80
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03															
E0 20																
E0 30																
FF F0																



Question 2(b)

```
bita  #$80
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80													
E0 20																
E0 30																
FF F0																

Question 2(b)

```
        bpl    IS0
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80													
E0 20																
E0 30																
FF F0																

Question 2(b)

```
        bpl    IS0
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80	2A	??											
E0 20																
E0 30																
FF F0																

Question 2(b)

```
IS1     ldx    Next1,X
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80	2A	??											
E0 20																
E0 30																
FF F0																

Question 2(b)

```
Next1   equ    4
IS1     ldx    Next1,X
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80	2A	??	EE I1	04									
E0 20																
E0 30																
FF F0																

Question 2(b)

```
        bra    LOOP
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80	2A	??	EE I1	04									
E0 20																
E0 30																
FF F0																

Question 2(b)

```
        bra    LOOP
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80	2A	??	EE I1	04	20	ED							
E0 20																
E0 30																
FF F0																

Question 2(b)

```
IS0     ldx    Next0,X
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80	2A	??	EE I1	04	20	ED							
E0 20																
E0 30																
FF F0																

Question 2(b)

```
Next0  equ    2
IS0    ldx    Next0,X
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80	2A	??	EE I1	04	20	ED	EE I0	02					
E0 20																
E0 30																
FF F0																

Question 2(b)

```
Next0  equ    2
IS0    ldx    Next0,X
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80	2A	04	EE I1	04	20	ED	EE I0	02					
E0 20																
E0 30																
FF F0																

Question 2(b)

```
       bra    LOOP
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80	2A	04	EE I1	04	20	ED	EE I0	02					
E0 20																
E0 30																
FF F0																

Question 2(b)

```
       bra    LOOP
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80	2A	04	EE I1	04	20	ED	EE I0	02	20	E9			
E0 20																
E0 30																
FF F0																

Question 2(b)

```
       org    $FFFE
       fdb    MAIN
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80	2A	04	EE I1	04	20	ED	EE I0	02	20	E9			
E0 20																
E0 30																
FF F0																

Question 2(b)

```
org   $FFFE
fdb   MAIN
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 00																
B6 00	6F o	64 d	64 d	65 e	76 v	65 e	6E n	B6 IS	09	B6 SE	03	B6	09	B6	FF	B6 S0
B6 10	00	B6	FF	B6	09											
E0 00	CF M	00	FF	FE	B6	09	ED L	00	7D	00	00	16	E2	00	B6	10
E0 10	03	85	80	2A	04	EE I1	04	20	ED	EE I0	02	20	E9			
E0 20																
E0 30																
FF F0															E0	00

Question 3

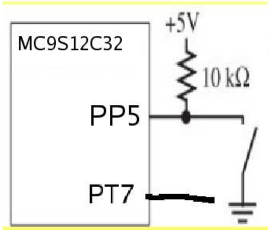
- In this problem, you are to design a simple interface to a single switch. You should get an interrupt each time the switch changes from being open to closed or from closed to open. We also want there to be a high pulse on Port T bit 7 each time the switch changes state. You may assume that the switch starts in the open state which produces 5V. Your solution should use interrupts.

Question 3(a)

- Draw a schematic for your design.

Question 3(a)

- Draw a schematic for your design.



Note that internal pullup cannot be used since we want interrupts on both rising and falling edges.

Question 3(b)

- Show bit by bit your choice for the parallel I/O control register's initial value.
- NOT APPLICABLE

Question 3(c)

- Show the ritual that is called by the main program.

Question 3(c)

- Show the ritual that is called by the main program.

```
void SW_Init(void){
...
}
```

Question 3(c)

- Show the ritual that is called by the main program.

```
void SW_Init(void){
unsigned char SaveCCR;
asm tpa             // save state of CCR
asm staa SaveCCR
asm sei             // disable interrupts, make atomic
...
asm ldaa SaveCCR    // restore state of CCR
asm tap
}
```

Question 3(c)

- Show the ritual that is called by the main program.

```
void SW_Init(void){
unsigned char SaveCCR;
asm tpa             // save state of CCR
asm staa SaveCCR
asm sei             // disable interrupts, make atomic
  DDRT |= 0x80;     // make PT7 an output
  PTT &= 0x80;      // clear PT7 bit
  ...
asm ldaa SaveCCR    // restore state of CCR
asm tap
}
```

Question 3(c)

- Show the ritual that is called by the main program.

```
void SW_Init(void){
unsigned char SaveCCR;
asm tpa             // save state of CCR
asm staa SaveCCR
asm sei             // disable interrupts, make atomic
  DDRT |= 0x80;     // make PT7 an output
  PTT &= 0x80;      // clear PT7 bit
  DDRP &= ~0x20;    // make PP5 an input
  PPSP &= ~0x20;    // wait for falling edge first
  PIEP |= 0x20;     // arm interrupts on PP5
  PIFP = 0x20;      // clear flag for PP5
asm ldaa SaveCCR    // restore state of CCR
asm tap
}
```

Question 3(d)

- Show the interrupt handler that is called when the switch changes state. You do not need to write the main program as the interrupt handler should do everything specified anyway.

Question 3(d)

- Show the interrupt handler that is called when the switch changes state. You do not need to write the main program as the interrupt handler should do everything specified anyway.

```
void interrupt 56 SW_Han(void){
...
}
```

Question 3(d)

- Show the interrupt handler that is called when the switch changes state. You do not need to write the main program as the interrupt handler should do everything specified anyway.

```
void interrupt 56 SW_Han(void){
  if((PIFP&0x20)==0) {   // check PP5 flag is set
    asm swi
  }
  ...
}
```

Question 3(d)

- Show the interrupt handler that is called when the switch changes state. You do not need to write the main program as the interrupt handler should do everything specified anyway.

```
void interrupt 56 SW_Han(void){
  if((PIFP&0x20)==0) {   // check PP5 flag is set
    asm swi
  }
  PIFP = 0x20;           // clear PP5 flag
  ...
}
```

Question 3(d)

- Show the interrupt handler that is called when the switch changes state. You do not need to write the main program as the interrupt handler should do everything specified anyway.

```
void interrupt 56 SW_Han(void){
  if((PIFP&0x20)==0) {   // check PP5 flag is set
    asm swi
  }
  PIFP = 0x20;           // clear PP5 flag
  PTT |= 0x80;           // start PT7 pulse
  ...
  PTT &= ~0x80;}         // end PT7 pulse
```

Question 3(d)

- Show the interrupt handler that is called when the switch changes state. You do not need to write the main program as the interrupt handler should do everything specified anyway.

```
void interrupt 56 SW_Han(void){
  if((PIFP&0x20)==0) {   // check PP5 flag is set
    asm swi
  }
  PIFP = 0x20;           // clear PP5 flag
  PTT |= 0x80;           // start PT7 pulse
  if((PTP&0x20)==0){
    PPSP |= 0x20;        // wait for rising edge next
  }else{
    PPSP &= ~0x20;       // wait for falling edge next
  }
  PTT &= ~0x80;}         // end PT7 pulse
```