Complete the timing diagram

Assume the SP starts at 0x00 and the PC starts at 0x03

```
.CSEG
.ORG 0x03
     3 LD    R2,0x04
     4 ADD   R2,0x01
     5 RET
     6 PUSH  R3
     7 PUSH  R4
     8 CALL  Func
     9 OR    R0, 0xFF
     A LSR   R1
     B BRNE  end
C Func: SUBC  R3, R1
     D  POP   R1
     E  RET
  end: F
```

87 - 44 = 42

Address	Value
R0	0x32
R1 0x04	~~0x44~~ 0x09
R2 0x00	0xD1 ~~0xFF~~ 0
R3	~~0x87~~ 0x42
R4	0x0A
..	
R13	0x55
R14	0x23
R15	0xFA
..	
R30	0x66
R31	0x11

Reg File

Address	Value
0x00	~~0x07~~ 0x0A
0x01	0x4C
0x02	0x81
0x03	0x21
0x04	0xFF
0xFD	0x22
0xFE	0x51
0xFF	~~0x29~~ 0x09

Scratch Pad

Table 1: Fetch/Execute Cycles

Cycle	IR	PC_SEL	SCR_ADD_SEL	PC	SP	C_FLAG	Z_FLAG
F1	X	0	0	3	0	0	0
E1	LD R2, 0x04	0	1	4	0	0	0
F	ADD R2, 0x01	0	0	4	0	0	0
E	ADD R2, 0x01	0	0	5	0	0	0
F	RET	0	0	5	0	1	1
E	RET	1	2	6	0	1	1
F	PUSH R3	0	0	7	1	1	1
E	PUSH R4	0	3	8	1	1	1
F	CALL Func	0	0	8	0	1	1
E	CALL Func	0	3	9	0	1	1
F	LSR R1	0	0	C	FF	1	1
E	SUBC R3, R1	0	0	D	FF	1	1
F	POP R1	0	0	D	FF	0	0
E	POP R1	0	2	E	FF	0	0
F	RET	0	0	E	0	0	0
E	RET	1	2	F	0	0	0
F	AND R0 R0	0	0	A	1	0	0
E	LSR R1	0	0	B	1	0	0
F	BRNE end	0	0	B	1	1	0
E	BRNE end	0	0	C	1	1	0
F	SUBC R3, R1	0	0	F	1	1	0
E	AND R0, R0	0	0	10	1	1	0
F	AND R0, R0	0	0	10	1	0	0
F	AND, R0, R0	0	0	11	1	0	0

Assume you add the enable interrupt instruction (SEI) to the start of the code as shown below and change the origin to 0x02 so all the other instructions remain at the same memory locations. Suppose an interrupt occurs during the execute cycle of the PUSH R4 instruction.

```
.CSEG
.ORG 0x02

      SEI
      LD    R2,0x04
      ADD   R2,0x01
      RET
      PUSH  R3
      PUSH  R4
      CALL  Func
      OR     R0, 0xFF
      LSR   R2
      BRNE  end
Func: SUBC  R3, R1
      POP   R4
      RET
end:
```

After the interrupt cycle completes:

a) What value do the shadow carry and shadow zero registers become?

They both become 1

b) What value does the stack pointer (SP) become and what value is stored on the stack?

SP becomes FF, the value 8 is stored there

c) What value does the PC become? Why?

0x3FF - the interrupt vector

d) What is missing from the code above to be able to service this interrupt?

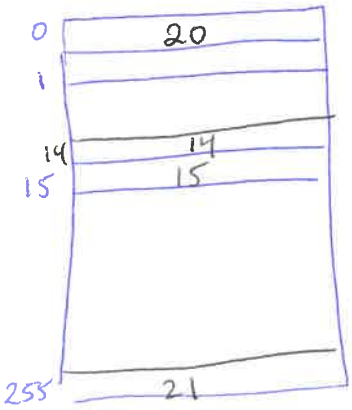
The ISR and interrupt vector

```
ISR :   ; do something
        RETIE

.ORG 0x3FF
     BRN  ISR
```

Complete the timing diagram for the following code. Assume the SP starts at 0x01 and the PC starts at 0x08. Assume all registers contain the value of their register number (i.e. R0 = 0x00, R21 = 21, etc.) All values in decimal

```
.CSEG
.ORG 0x08
          8 ST    R15, (R15)
          9 PUSH  R20
         10 PUSH  R21
         11 POP   R1
         12 WSP   R15
         13 CALL  My_Func
         14 POP   R1
         15 WSP   R0
         16 RET
My_Func: 17 POP   R1
         18 RET
         19 POP   R2
         20 POP   R3
```



Scratch Ram

Cycle	Instruction	PC	SP
F	X	8	1
E	ST R15, R15	9	1
F	PUSH R20	9	1
E	PUSH R20	10	1
F	PUSH R21	10	0
E	PUSH R21	11	0
F	POP R1	11	FF
E	POP R1	12	FF
F	WSP R15	12	0
E	WSP R15	13	0
F	CALL My-Func	13	15
E	CALL My-Func	14	15
F	POP R1	17	14
E	POP R1 (2)	18	14
F	RET	18	15
E	RET	19	15
F	POP R2	15	16
E	WSP R0	16	16
F	RET	16	0
E	RET	17	0
F	POP R1	20	1
E	POP R3	21	1
F	AND	21	2
E	AND	22	2
F			
E			
F			
E			
F			
E			
F			
E			
F			
E			