|  |  |
| --- | --- |
| Address | Value |
| R0 | 0x32 |
| R1 | 0x44 |
| R2 | 0xD1 |
| R3 | 0x87 |
| R4 | 0x0A |
| .. |  |
| R13 | 0x55 |
| R14 | 0x23 |
| R15 | 0xFA |
| .. |  |
| R30 | 0x66 |
| R31 | 0x11 |

Reg File

|  |  |
| --- | --- |
| Address | Value |
| 0x00 | 0x07 |
| 0x01 | 0x4C |
| 0x02 | 0x81 |
| 0x03 | 0x21 |
| 0x04 | 0xFF |
|  |  |
|  |  |
|  |  |
|  |  |
| 0xFD | 0x22 |
| 0xFE | 0x51 |
| 0xFF | 0x29 |

Scratch Pad

Complete the timing diagram

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

.CSEG

.ORG 0x03

LD R2,0x04

ADD R2,0x01

RET

PUSH R3

PUSH R4

CALL Func

OR R0, 0xFF

LSR R1

BRNE end

Func: SUBC R3, R1

POP R1

RET

end:

Table 1: Fetch/Execute Cycles

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Cycle | IR | PC\_SEL | SCR\_ADD\_SEL | PC | SP | C\_FLAG | Z\_FLAG |
| F1 |  |  |  |  |  |  |  |
| E1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

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:

1. What value do the shadow carry and shadow zero registers become?
2. What value does the stack pointer (SP) become and what value is stored on the stack?
3. What value does the PC become? Why?
4. What is missing from the code above to be able to service this interrupt?

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.). Write all values in decimal.

.CSEG

|  |  |  |  |
| --- | --- | --- | --- |
| Cycle | Instruction | PC | SP |
| F | X | 8 | 1 |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |
| F |  |  |  |
| E |  |  |  |

.ORG 0x08

ST R15, (R15)

PUSH R20

PUSH R21

POP R1

WSP R15

CALL My\_Func

POP R1

WSP R0

RET

My\_Func: POP R1

RET

POP R2

POP R3