

SECTION

REG #NO

ASSIGNMENT

MBSD (02)

(1)

## QUESTION 01 ::

Answer ::

MCQ430FR23FF: Handling Overflow and underflow:-

### 1) Role of V and C ::

**V (overflow)** is used for signed operation. It will set to one when the result exceeds the range of a signed number.

**C (carry)** used for unsigned operations - set to 1 when there is a carry-out from most significant bit. Also used to detect borrow in subtraction - if  $C=0$  borrow occurred -

### 2) Overflow and underflow:-

**overflows** happens in signed arithmetic e.g.  $127 + 1 = -128$  (overflow in 8-bit signed)

**underflow** happens in unsigned subtraction e.g.  $0 - 1 = 0xFF$  (underflow no carry -

### 3) How to detect and handle overflow in Code:-

The V (bit 8) is in the status register (SR). For we can test it using the BIT instruction.

ADD R12, R13

(2)

BIT #0x0100, SR

JZ no\_overflow

; overflow occurred

MOV #0x7FFF, R13

no\_overflow:

### How to check Carry (C Flag)

SUB R12, R13 ; unsigned subtraction

BIT #0x0001, SR ; Test C flag (bit 0)

JNZ no\_borrow ; if C = 1 → no borrow

; underflow occurred

MOV #0, R13 ; Handle underflow

no\_borrow:

### QUESTION 02:-

#### Answer:-

#### Code:-

MOV.W #10, R14

MOV.W #20, R15

SUB R15, R14

JM negative\_result

JMP positive\_result

negative\_result:

MOV.W #DEAD, R12

JMP done

(3)

Positive result:

MOV #BEEF, R12

JMP done

done:

NOP



QUESTION #03:-

Answer:-

KEY I/O Registers And Their functions:-

PxDIR:-

Directions set as input (0) or output (1)

PxOUT:-

Use for output - send HIGH (1) or LOW (0) when pin is output -

PxIN:-

Input: Read HIGH (1) or LOW (0) when pin is input -

PxREN:-

Registers enable: Turn on internal pull-up/down resistors -

PxOUT:-

Also selects pull-up (1) or pull down (0) if used with REN -

Configure pin as <sup>input</sup> output:-

bic.b #BIT1, SP1DIR ; P1DIR &= ~BIT1

bis.b #BIT1, SP1REN ; P1REN |= BIT1

bis.b #BIT1, SP1OUT ; P1OUT |= BIT1



(4)

### Configure as output:-

bits.b #BIT0, &P1DIR ; P1DIR |= BIT0

bits.b #BIT0, &P1OUT ; P1OUT |= BIT0

### Enabling pull-up or pull down:-

P1DIR &= ~BIT1; // bits.b #BIT1, &P1DIR

P1REN |= BIT1; // bits.b #BIT1, &P1REN

P1OUT |= BIT1; // bits.b #BIT1, &P1OUT

### For pull down:-

P1OUT &= ~BIT1 // bits.b #BIT1, &P1OUT

### Reading Input pin:-

```
if (P1IN1 & BIT1) {  
    // pin is high
```

```
}
```

```
else {
```

```
    // pin is low
```

```
}
```

### Importance of pull-up and down:-

Pull-up input: input held high if unconnected-

pull-down input: input held low if unconnected-

(5)

## QUESTION 04:-

### Answer:-

#### Configure Edge-Triggered Interrupts on I/O pins:-

To trigger an interrupt when a pin changes state follows these steps

- 1) P1DIR set pins as input P1DIR &= ~BIT1
- 2) P1DIR enable resistor P1REN
- 3) P1OUT select pull-up/down P1OUT
- 4) P1IE enable interrupt for pin P1IE
- 5) P1IES select edge type P1IES
- 6) P1IFG Flag set when interrupt occurs Automatically set when triggered -

#### How interrupt works:-

When P1.1 changes as per the edge setting P1IFG bit for P1.1 is set

The MCU jumps to the interrupt vector (PORT1-VECTOR)

You must clear P1IFG in the ISR.

#### Clearing Interrupt Flag:-

In the ISR clear the flag

bic.b #BIT1, SP1IFG ; clear flag for P1.1