**IRQ Interrupt Example**

* Counts up the number of IRQs that occur up to 65535.
* Uses real interrupt vector AND pseudo-vector for IRQ.
* For Simulator or when you can change interrupt vectors.
* Uses BSET and BCLR instead of blind loads and stores.
* (Could alternately use loads, ANDs/ORs and stores.)

---

**EQUATEs**

- `PORTB EQU $0001 ; Output Port B`
- `DDRB EQU $0003 ; Data Direction Port B`
- `INTCR EQU $001E ; Interrupt control register (for IRQ)`
- `RAM EQU $0900 ; Start of RAM`
- `STK EQU $0900 ; User Stack`
- `RAMPROG EQU $0970 ; Program space in RAM`
- `BIT0 EQU %00000001`
- `IPV_IRQ EQU $0836 ; Pseudo-vector location for IRQ`
- `IV_IRQ EQU $FFF2 ; Interrupt vector location for IRQ`

---

**Data Section**

- `ORG RAM`
- `CNTR16 DS.W 1 ; Could use DC.W, but BAD!!!`

---

**Interrupt Vector Initialization**

- `ORG IV_IRQ ; This is for interrupt vector`
- `DC.W IPV_IRQ ; not interrupt pseudo-vector`

---

**Interrupt Pseudo-Vector Initialization (to run on UF 68HC12 board)**

- `ORG IPV_IRQ ; This is for the interrupt pseudo-vector`
- `JMP IRQ_ISR ; that will run on the UF 68HC12 board`

---

**Main Program**

- `ORG RAMPROG`

---

**Initialization Section**

- `LDS #STK ; Initialize the stack pointer, SP`
- `LDD #0 ; Clear 16-bit D register`
- `STD CNTR16 ; Take the responsibility for clearing counter`

---

**Setup each hardware device**

- `BSET DDRB Bit0 ; Set PB for output port`
- `Initialize the IRQ FF to zero using the Pre-Clear input on PB0`
- `BCLR PORTB Bit0 ; Set PB0=0, i.e., pre-clear FF`
- `BSET PORTB Bit0 ; Turn off pre-clear input`

---

**Main program does its work here**

- `LDAA #%01000000 ; Enable the external IRQ by clearing`
- `STAA INTCR ; the (local) IRQ mask bit (bit 6)`
- `CLI ; Turn interrupt system on`
- `HERE BRA HERE ; Main program waits on interrupts`

---

**Interrupt Service Routine(s)**

**Part I : Service the interrupt**

- `ORG $09C0`
- `IRQ_ISR LDD CNTR16 ; Obtain current count`
irq_example.asm

ADDD #1 ; Increment the count
STD CNTR16 ; Store the current count

* Part II: Clear the interrupt flag

BCLR PORTB Bit0 ; Set PB0=0, i.e., pre-clear FF
BSET PORTB Bit0 ; Turn off pre-clear input
RTI ; Return from Interrupt Service