

Slide 1

ECE/CE 3720: Embedded System Design

Chris J. Myers

Lecture 8: Interrupts in the 68HC11

Slide 2

### General Features of Interrupts

- All interrupting systems must have the:
  1. Ability for hardware to request action from computer.
  2. Ability for computer to determine the source.
  3. Ability for computer to acknowledge the interrupt.
- To *arm* (*disarm*) a device means to enable (shut off) the source of interrupts.
- To *enable* (*disable*) means to allow (postpone) interrupts at this time.

Slide 3

### Sequence of Events During Interrupt

1. Hardware needs service (*busy-to-done*) transition.
2. Flag is set in one of the I/O status registers.
  - (a) Interrupting event sets the flag (ex., *STAF*–1).
  - (b) The device is armed (ex., *STAI*–1).
  - (c) Microcomputer interrupts are enabled (ex., *I*–0).
3. Thread switch.
  - (a) Microcomputer finishes current instruction.
  - (b) All registers are pushed onto the stack.
  - (c) Vector address is obtained and put into the PC.
  - (d) Microcomputer sets *I*–1.
4. Execution of the ISR.
5. Return control back to the thread that was running.

Slide 4

### 6811 Stack Before and After an Interrupt

(See Figure 4.18)

Slide 5

### 6811 Interrupts

- 6811 has two external requests  $\overline{TRQ}$  and  $\overline{XTRQ}$ .
- Other interrupt sources include:
  - A STRA interrupt
  - Three input capture interrupts
  - Five output capture interrupts
  - Three timer interrupts (timer overflow, RTI, pulse accumulator)
  - Two serial port interrupts (SCI and SPI)

Slide 7

### 6811 Interrupt Vectors and Priority

| Vector | Interrupt Source     | Enable | Arm               |
|--------|----------------------|--------|-------------------|
| \$FFFE | Power on reset       | Always | Always highest    |
| \$FFFE | Hardware reset       | Always | Always            |
| \$FFFC | COP clk monitor fail | Always | OPTION.CME-1      |
| \$FFFA | COP failure          | Always | CONFIG.NOCOP-0    |
| \$FFF4 | Nonmaskable XIRQ     | X-0    | External hardware |
| \$FFF2 | External IRQ         | I-0    | External hardware |
| \$FFF2 | Parallel I/O, STAF   | I-0    | PIOC.STAI-1       |
| \$FFF0 | Real time int., RTIF | I-0    | TMSK2.RTI-1       |
| \$FEEE | Inp capture 1, IC1F  | I-0    | TMSK1.IC1I-1      |
| \$FEEC | Inp capture 2, IC2F  | I-0    | TMSK1.IC2I-1      |
| \$FEEA | Inp capture 3, IC3F  | I-0    | TMSK1.IC3I-1      |
| \$FEE8 | Outp capture 1, OC1F | I-0    | TMSK1.OC1I-1      |
| \$FEE6 | Outp capture 2, OC2F | I-0    | TMSK1.OC2I-1      |
| \$FEE4 | Outp capture 3, OC3F | I-0    | TMSK1.OC3I-1      |

Slide 6

### 6811 Interrupts

- Interrupts have a fixed priority, but can elevate one to highest priority using hardware priority interrupt (HPRIO) register.
- XIRQ is highest-priority device and has separate vector and enable bit (X).
- Once X bit is cleared, software cannot disable it.
- XIRQ handler sets X and I, and restores with `rti`.

Slide 8

### 6811 Interrupt Vectors and Priority (cont)

| Vector | Interrupt Source       | Enable | Arm           |
|--------|------------------------|--------|---------------|
| \$FEE2 | Outp capture 4, OC4F   | I-0    | TMSK1.OC4I-1  |
| \$FEE0 | Outp capture 5, OC5F   | I-0    | TMSK1.OC5I-1  |
| \$FDEE | Timer overflow, TOF    | I-0    | TMSK2.TOI-1   |
| \$FDC  | Pulse accum overflow   | I-0    | TMSK2.PAOVI-1 |
| \$FDA  | Pulse accum inp edge   | I-0    | TMSK2.PAI-1   |
| \$FD8  | SPI complete, SPIF     | I-0    | SPCR.SPIE-1   |
| \$FD6  | Rx data reg full, RDRF | I-0    | SCCR2.RIE-1   |
| \$FD6  | Rx overrun, OVRN       | I-0    | SCCR2.RIE-1   |
| \$FD6  | Tx data reg full, TDRE | I-0    | SCCR2.TIE-1   |
| \$FD6  | Tx complete, TC        | I-0    | SCCR2.TCIE-1  |
| \$FD6  | Idle line detect, IDLE | I-0    | SCCR2.IIE-1   |
| \$FFFS | Illegal opcode trap    | Always | Always        |
| \$FFFE | Software interrupt SWI | Always | Always lowest |

Slide 9

### Setting Interrupt Vectors

```
org $FFFF
fdb RTIHAN   Ptr to real time interrupt handler
org $FFF2
fdb IRQHAN   Ptr to external IRQ and STRA handler
org $FFF4
fdb XIRQHAN  Ptr to external XIRQ handler
org $FFFE
fdb RESETHAN Ptr to reset handler
```

Slide 11

### 6811 Pseudo-Vectors (cont)

| 6811 Vector | Vector Name            | Address       |
|-------------|------------------------|---------------|
| \$FFEA      | Inp capture 3, IC3F    | \$00E2-\$00E4 |
| \$FFEC      | Inp capture 2, IC2F    | \$00E5-\$00E7 |
| \$FFEE      | Inp capture 1, IC1F    | \$00E8-\$00EA |
| \$FFF0      | Real time int., RTIF   | \$00EB-\$00ED |
| \$FFF2      | External IRQ, STAF     | \$00EE-\$00F0 |
| \$FFF4      | Nonmaskable XIRQ       | \$00F1-\$00F3 |
| \$FFF6      | Software interrupt SWI | \$00F4-\$00F6 |
| \$FFF8      | Illegal opcode trap    | \$00F7-\$00F9 |
| \$FFFA      | COP failure            | \$00FA-\$00FC |
| \$FFFC      | COP clk monitor fail   | \$00FD-\$00FF |

Slide 10

### 6811 Pseudo-Vectors

| 6811 Vector | Vector Name          | Address       |
|-------------|----------------------|---------------|
| \$FFD6      | SCI                  | \$00C4-\$00C6 |
| \$FFD8      | SPI                  | \$00C7-\$00C9 |
| \$FFDA      | Pulse accum inp edge | \$00CA-\$00CC |
| \$FFDC      | Pulse accum overflow | \$00CD-\$00CF |
| \$FFDE      | Timer overflow, TOF  | \$00D0-\$00D2 |
| \$FFE0      | Outp capture 5, OC5F | \$00D3-\$00D5 |
| \$FFE2      | Outp capture 4, OC4F | \$00D6-\$00D8 |
| \$FFE4      | Outp capture 3, OC3F | \$00D9-\$00DB |
| \$FFE6      | Outp capture 2, OC2F | \$00DC-\$00DE |
| \$FFE8      | Outp capture 1, OC1F | \$00DF-\$00E1 |

Slide 12

### Setting Interrupt Pseudo-Vectors

```
ldaa #$7E   Opcode for JMP
staa $00EB
ldx #RTIHAN   Ptr to real time interrupt handler
stx $00EC    JMP RTIHAN
ldaa #$7E   Opcode for JMP
staa $00EE
ldx #IRQHAN   Ptr to external IRQ and STRA handler
stx $00EF    JMP IRQHAN
ldaa #$7E   Opcode for JMP
staa $00F1
ldx #XIRQHAN  Ptr to external IRQ and STRA handler
stx $00F2    JMP XIRQHAN
```