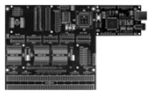
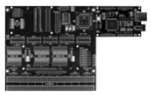


EEL 3744

Menu

- Interrupt motivation (polling example)
- Interrupt description
- Interrupt-driven I/O
  - > Interrupt Protocol
  - > Preemptive-Resume Priority Interrupts
  - > Vectored Interrupt
- Resets and Interrupts in XMEGA, 68HC11-12
  - > Resets and Interrupts Priority
  - > Interrupt and Reset Vector Assignments
    - – Pseudo-vector assignments
  - > Reset and Interrupt Processing
  - > Interrupt vector and pseudo-vector examples
- XMEGA interrupt details

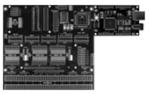
See readings & examples on web-site:
`Textbook (ch 10), doc8331 (sec 12-14), doc8385 (sec 14), External_Interrupt.asm`



EEL 3744

Polling Example

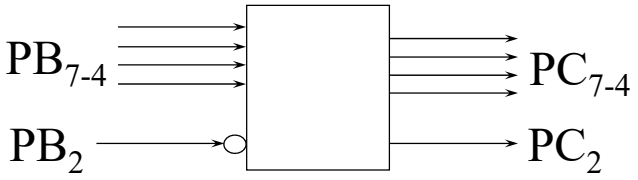
- You used polling in lab 2 when you were tasked to change some outputs based on the value of a specific input
- In the polling method, the μP "polls" an input or the status of a bit (or group of bits)
  - >If the bit(s) have the proper value(s), then an action should be taken
  - >If the bit(s) do not have the proper value(s), then the bit(s) will be polled again, and again, and again, …

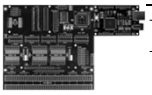

EEL 3744

Polling Example

Example:
Read in a 4-bit value from PB7-PB4, when PB2=0; CPU indicates ready status by setting PC2. CPU writes value of PB7-PB4 to PC7-PC4.

- Processor sets PC_2 to indicate ready for new data
- Wait for PB_2 to go true, then read the 4-bits at PB_7-PB_4
- Copy the read values at PB_7-PB_4 to at PC_7-PC_4
- Processor sets PC_2 to indicate ready for new data




EEL 3744

Interrupt Description

- In the interrupt-driven alternative to polling, a peripheral or an I/O device that is ready for service indicates so by "**interrupting**" the μP
- If the device is allowed to interrupt, then the μP will complete the execution of the current instruction, save the processor status (**the CPU registers, except the SP**) on the stack, and branch to a special location (**interrupt vector address**) to execute a special subroutine (ISR) that will service the interrupting device.

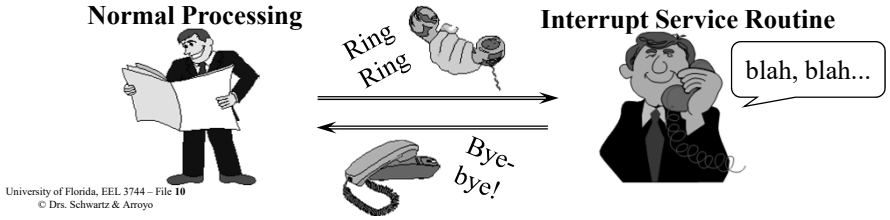
For XMEGA*: PCL, PCM, PCH

*Careful! XMEGA stores **NO** status info

For 68HC11/12: PCL, PCH, YL, YH, XL, XH, A, B, and CCR

EEL 3744 Interrupt Description

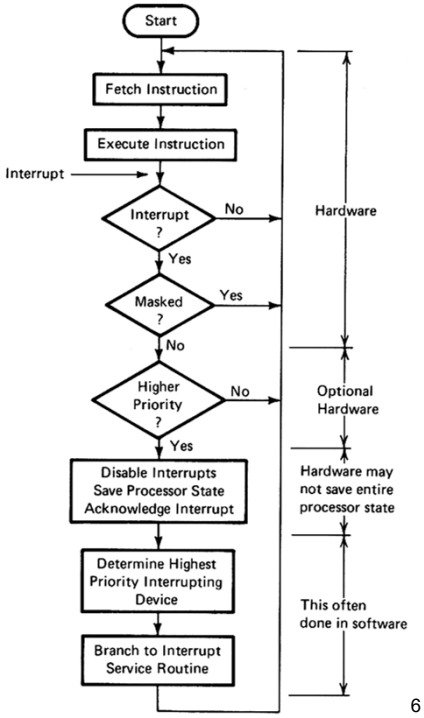
- At the completion of the interrupt-service routine, the processor status is restored from the stack by executing a return from interrupt (e.g., RETI [or RTI or IRET]) instruction
  - >This returns control to the program instruction that was originally scheduled to be executed before the interrupt was acknowledged
- The μP then resumes its normal processing

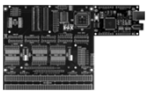


EEL 3744 Interrupt-driven I/O

- Processing interrupts

Doty: Fig 6.7-6

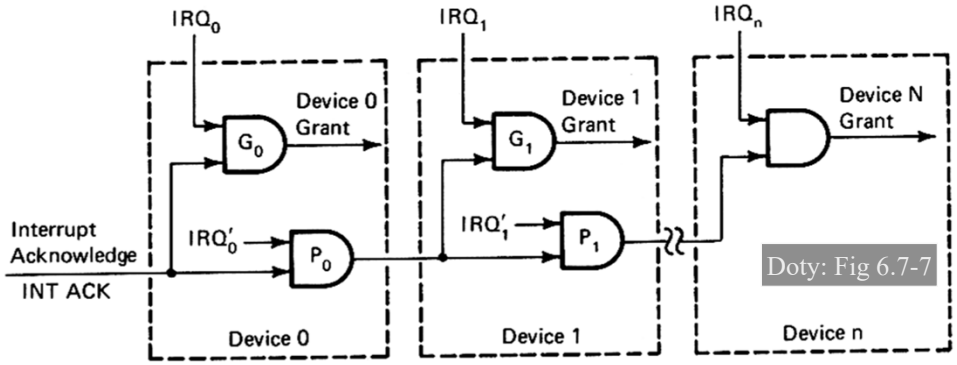


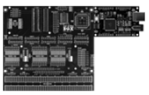


EEL 3744

Interrupt-driven I/O

- Daisy-chain priority and polling logic

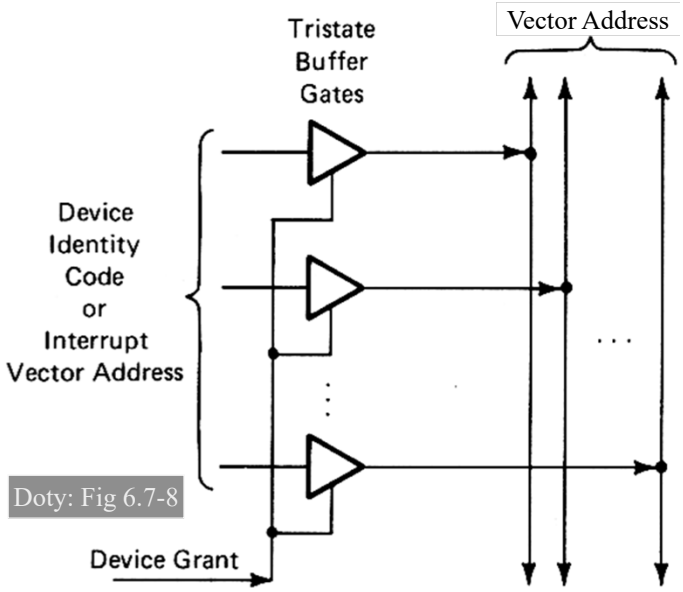


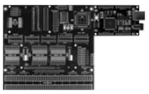


EEL 3744

Interrupt-driven I/O

- Generation of a device interrupt vector address

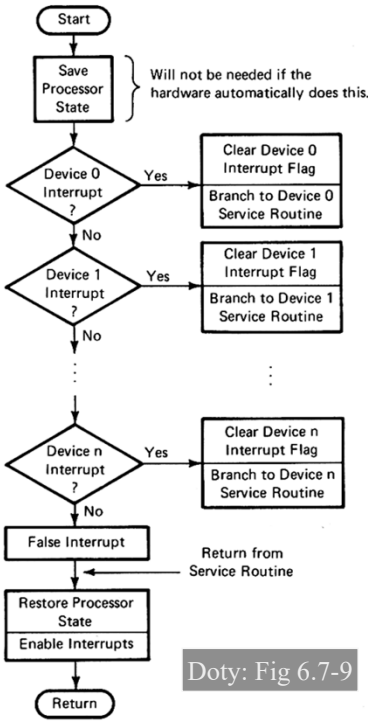




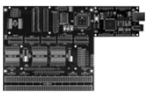
EEL 3744

Interrupt-driven I/O

- Determining interrupting device through software polling



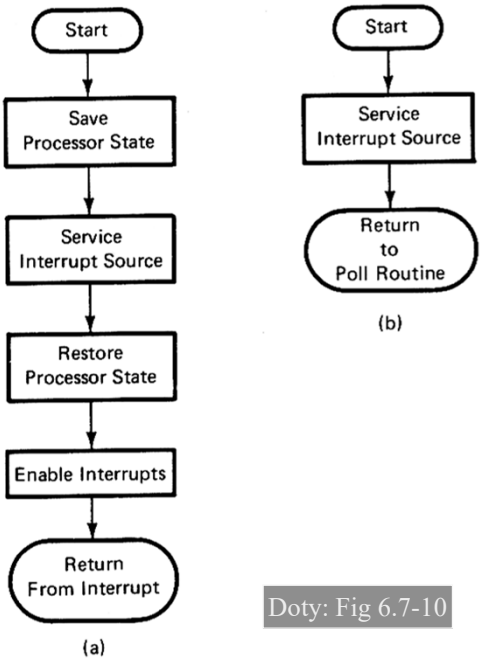
Doty: Fig 6.7-9



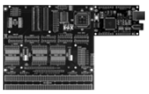
EEL 3744

Interrupt-driven I/O

- Interrupt service routine flowchart
  - >Vectored interrupts (a)
  - >Software polling (b)



Doty: Fig 6.7-10

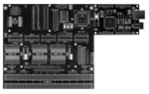


EEL 3744

68HC11
Interrupt
Promotion Table

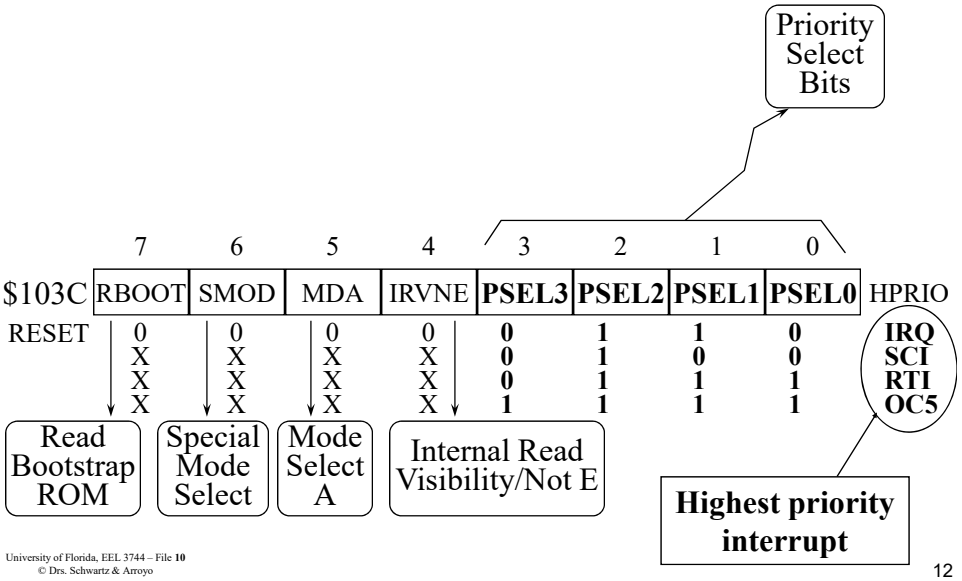
- Reset & Interrupt Priority
  - >HPRIO - Highest priority I-Bit interrupt using (PSEL**i**)

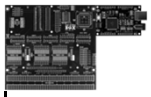
PSEL[3:0]	Interrupt Source Promoted
0000	Timer Overflow
0001	Pulse Accumulator Overflow
0010	Pulse Accumulator Input Edge
0011	SPI Serial Transfer Complete
0100	SCI Serial System
0101	Reserved (Default to $\overline{\text{IRQ}}$)
0110	$\overline{\text{IRQ}}$ (External Pin or Parallel I/O)
0111	Real-Time Interrupt
1000	Timer Input Capture 1
1001	Timer Input Capture 2
1010	Timer Input Capture 3
1011	Timer Output Compare 1
1100	Timer Output Compare 2
1101	Timer Output Compare 3
1110	Timer Output Compare 4
1111	Timer Input Capture 4/Output Compare 5



EEL 3744

Resets and Interrupts in **68HC11**





EEL 3744

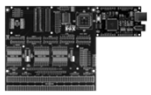
68HC11 Interrupt and Reset Vectors

Bible: Page 3

TD: Table 5-4

Vec Addr	Interrupt Source
FFD6, D7	Serial Comm. Interface (SCI)
FFD8, D9	Serial Peripheral Interface (SPI)
FFDA, DB	Pulse Accumulator Input Edge
FFDC, DD	Pulse Accumulator Overflow
FFDE, DF	Timer Overflow
FFE0, E1	Timer Output Compare 5
FFE2, E3	Timer Output Compare 4
FFE4, E5	Timer Output Compare 3
FFE6, E7	Timer Output Compare 2
FFE8, E9	Timer Output Compare 1
FFEA, EA	Timer Input Capture 3
FFEC, ED	Timer Input Capture 2
FFEE, EF	Timer Input Capture 1
FFF0, F1	Real Time Interrupt
FFF2, F3	IRQ
FFF4, F5	XIRQ
FFF6, F7	Software Interrupt (SWI)
FFF8, F9	Illegal Opcode
FFFA, FB	Computer Operating Properly (COP)
FFFC, FD	Clock Monitor
FFFE, FF	RESET

13



EEL 3744

68HC11 EVBU Interrupt Pseudo-Vectors (with BUFFALO)

BUFFALO memory dump:

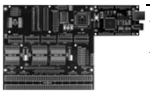
```
FFD6: 00 C4
FFD8: 00 C7 00 CA 00 CD 00 D0
FFE0: 00 D3 00 D6 00 D9 00 DC
FFE8: 00 DF 00 E2 00 E5 00 E8
FFF0: 00 EB 00 EE 00 F1 00 F4
FFF8: 00 F7 00 FA 00 FD B6 00
```

Reset Pseudo Vector

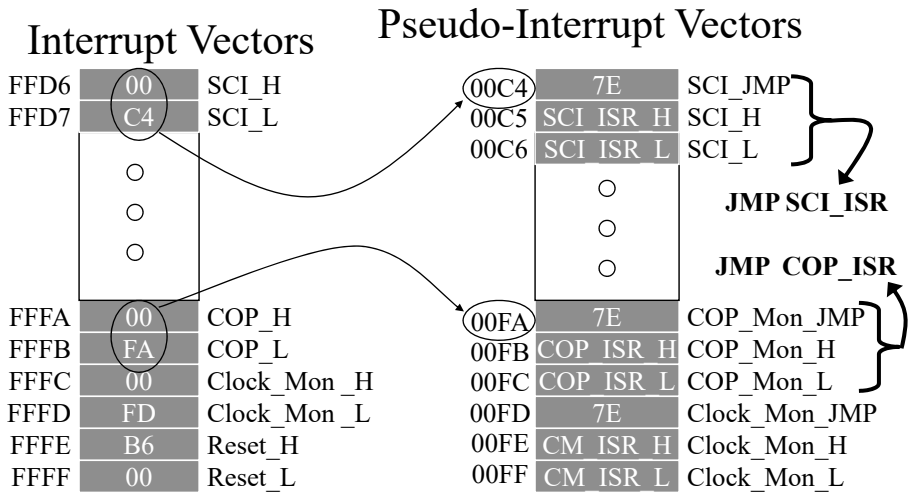
Pseudo Vector	Interrupt Source
$00C4-$00C6	Serial Comm. Interface (SCI)
$00C7-$00C9	Serial Peripheral Interface (SPI)
$00CA-$00CC	Pulse Accumulator Input Edge
$00CD-$00CF	Pulse Accumulator Overflow
$00DO-$00D2	Timer Overflow
$00D3-$00D5	Timer Output Compare 5
$00D6-$00D8	Timer Output Compare 4
$00D9-$00DB	Timer Output Compare 3
$00DC-$00DE	Timer Output Compare 2
$00DF-$00E1	Timer Output Compare 1
$00E2-$00E4	Timer Input Capture 3
$00E5-$00E7	Timer Input Capture 2
$00E8-$00EA	Timer Input Capture 1
$00EB-$00ED	Real Time Interrupt
$00EE-$00F0	IRQ
$00F1-$00F3	XIRQ
$00F4-$00F6	Software Interrupt (SWI)
$00F7-$00F9	Illegal Opcode
$00FA-$00FC	Computer Operating Properly (COP)
$00FD-$00FF	Clock Monitor

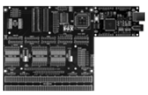
Vector Addr	Interrupt Source
FFD6, D7	Serial Comm. Interface (SCI)
FFE0, E1	Timer Output Compare 5
FFF0, F1	Real Time Interrupt
FFF2, F3	IRQ
FFFE, FF	RESET

14



EEL 3744 Interrupt Vectors & Interrupt Pseudo-vectors for **68HC11** with BUFFALO





EEL 3744 **68HC11** SCI Interrupts

- Make a pseudo-vector (for BUFFALO) for the SCI interrupt service routine:

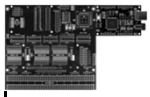
```
ORG     $????
JMP     SCI_ISR
```

Pseudo-vector (with BUFFALO): $00C4

- Make vector (for no BUFFALO) for the SCI interrupt service routine:

```
ORG   $????
DC.W  IRQ_ISR
```

Vector without BUFFALO: $FFD6

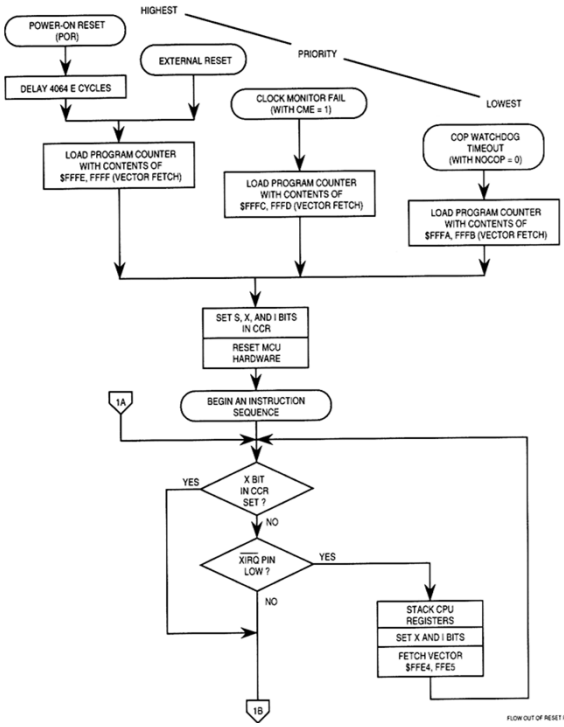


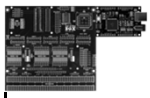
EEL 3744

68HC11 Resets and Interrupts Flow Chart (1/2)

- Processing flow out of reset (part a)

TD: Fig 5-1a
(Ref: Fig 5-2)



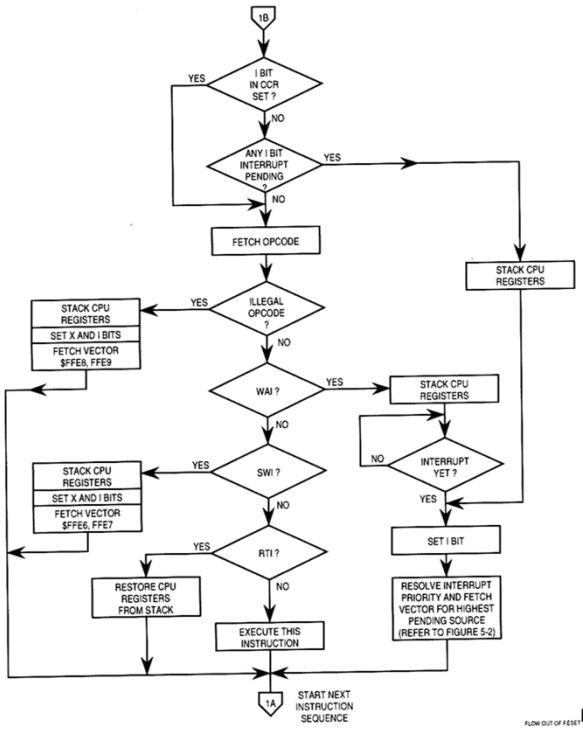


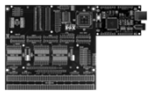
EEL 3744

68HC11 Resets and Interrupts Flow Chart (2/2)

- Processing flow out of reset (part b)

TD: Fig 5-1b





EEL 3744

68HC11 Interrupt Priority Resolution (1/2)

- Interrupt priority resolution

TD: Fig 5-2a
(Ref: Fig 5-3)