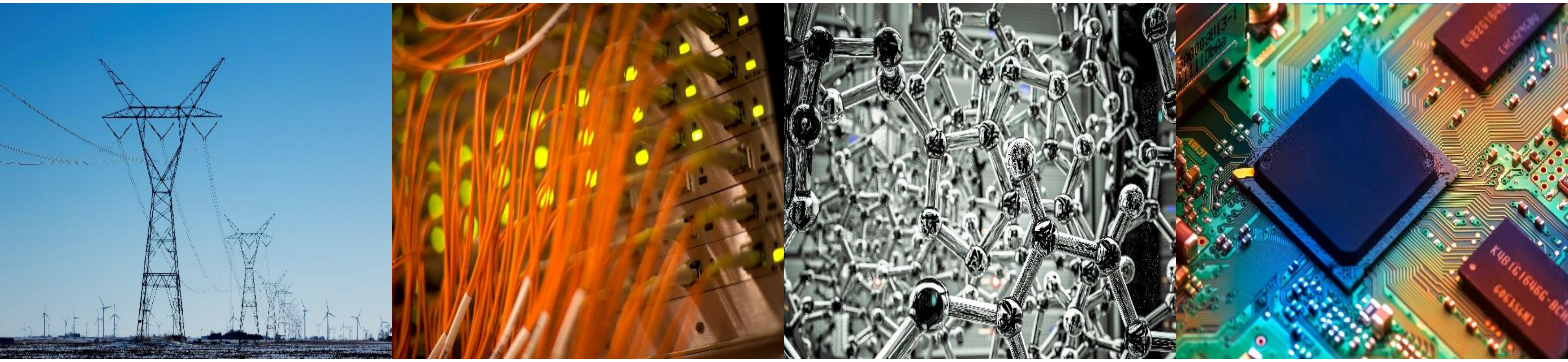
ECE 220 Computer Systems & Programming

Lecture 25 – Interrupts & Exceptions

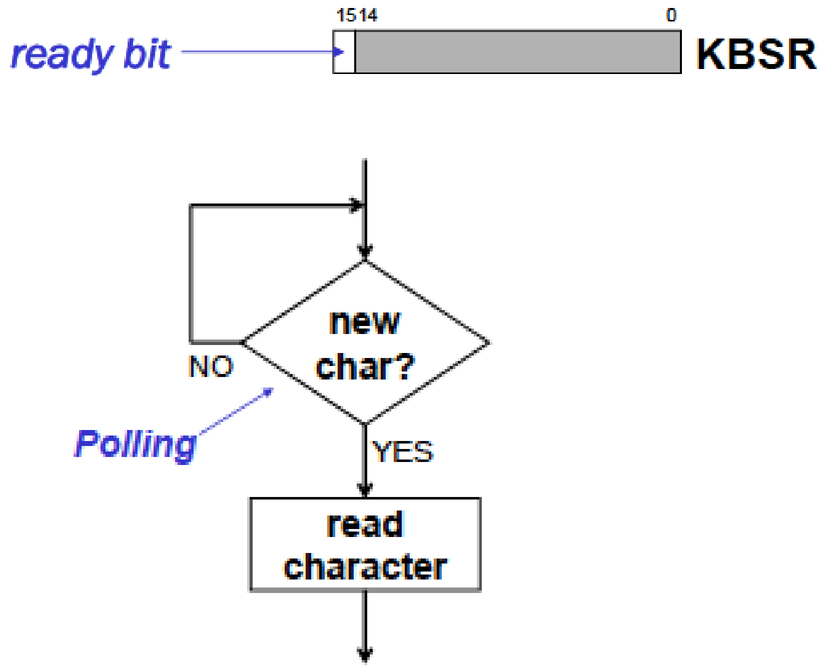
December 2, 2025



- Quiz6 should be taken at CBTF this week
- Extra-credit quiz is available now on PrairieLearn
- Final exam conflict request is due 12/9

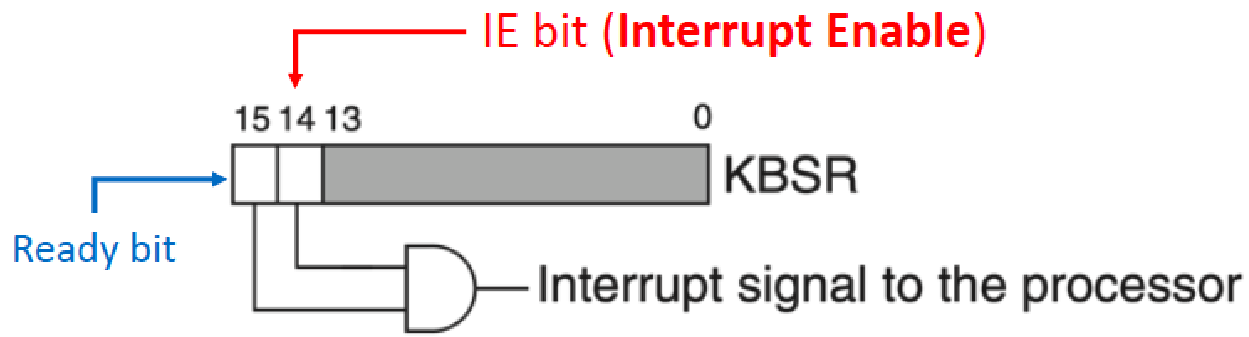
I ILLINOIS
Electrical & Computer Engineering
GRAINGER COLLEGE OF ENGINEERING

Polling vs. Interrupt-Driven I/O



```
KB_POLL     LDI R0, KBSR
            BRzp KB_POLL
            LDI R0, KBDR
            …
KBSR .FILL xFE00
KBDR .FILL xFE02
```

Ready bit: set by keyboard
IE bit: set by software



IE = 0

- **I/O device will NOT be able to interrupt**
- **Have to use polling**

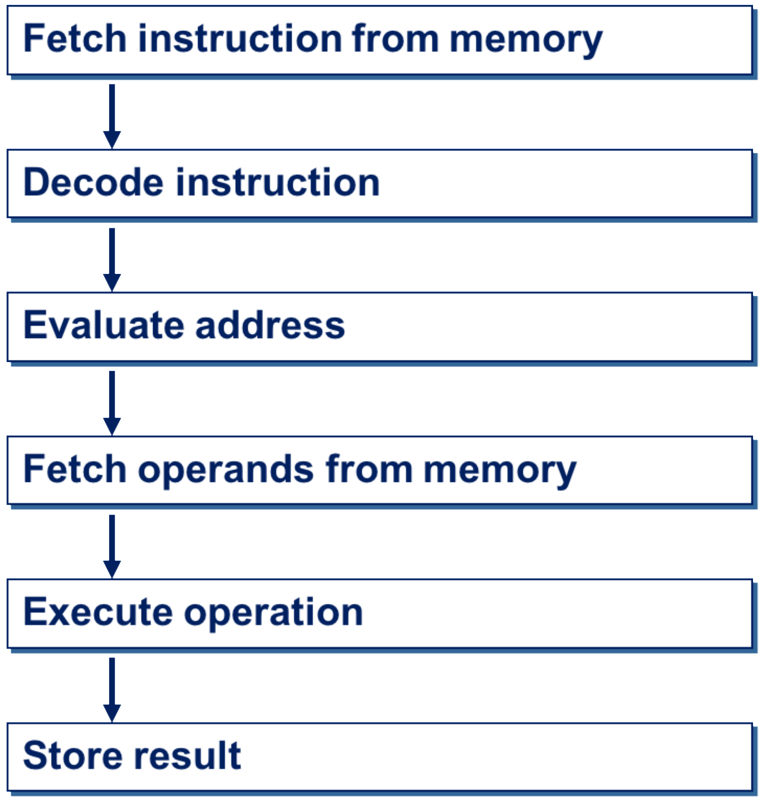
IE = 1

- **Interrupt-driven I/O enabled**
- **Interrupt request generated as soon as the ready bit is set (a key is typed)**

Interrupt-Driven I/O

An I/O device can:

1. Force currently executing program to stop
2. Have the processor carry out the need of the I/O device
3. Resume the stopped program as if nothing had happened



- If INT is not asserted
- If INT is asserted

Two Parts of Interrupt-Driven I/O

1. **The mechanism to interrupt the processor**
   - A way for the I/O device to ___________ the CPU that an interesting event has occurred
   - A way for the CPU to ___________ whether the interrupt signal is set and whether its priority is higher than the current program.

2. **The mechanism to handle the interrupt request**
   - Initiate the interrupt (***saving*** the state of the interrupted program & ***loading*** the state of the Interrupt Service Routine)
   - Service the interrupt
   - Return from interrupt

Processor State

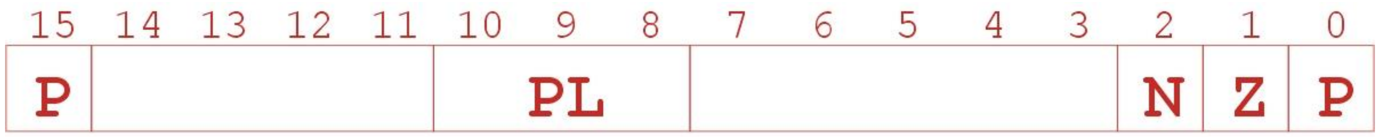
- Enough state info saved for interrupted program to resume later
- Enough state info loaded for the interrupt service routine to begin service

State of a Program (snapshot of the system):

1. PSR (processor status register)
2.
3.

PSR[15] – **privileged mode** (supervisor - 0) or **unprivileged mode** (user - 1)

PSR[10:8] – **priority level** (higher level = more urgent), PSR[2:0] – **condition code**



➢ Where should state information be saved?

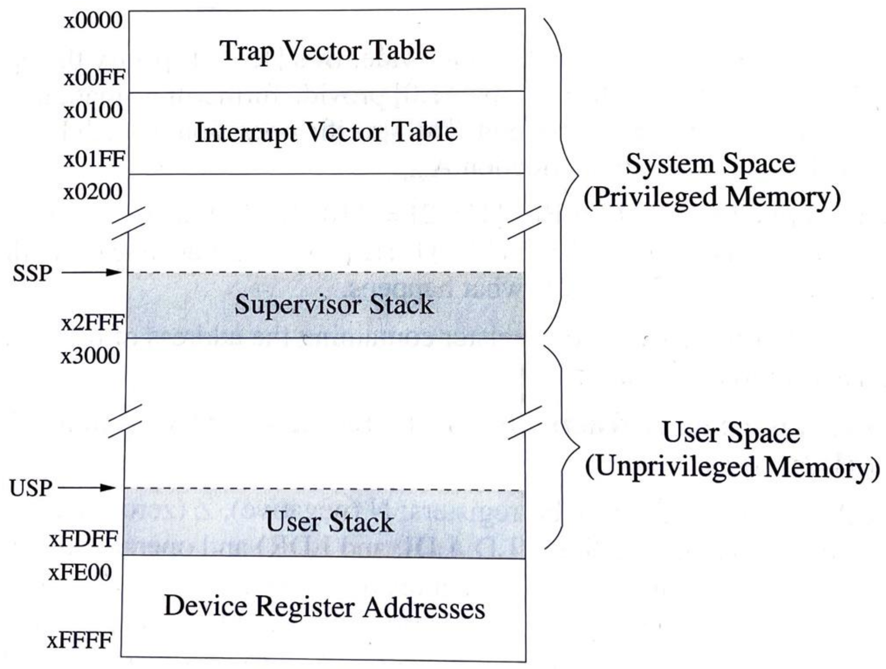
Supervisor Stack

A special region of memory used as the *stack* for interrupt service routines.

To keep track of stack top, two internal registers are used: Saved.SSP and Saved.USP

We want to use R6 as _____________, so that our PUSH/POP routines still work.

- When switching from user mode to supervisor mode (as result of interrupt), which register should R6 be saved to?



Invoking the Service Routine / Returning from Interrupt

Invoking:

I/O device transmits **Interrupt Vector** (INTV, 8-bit) along with interrupt signal and priority level.

Returning:

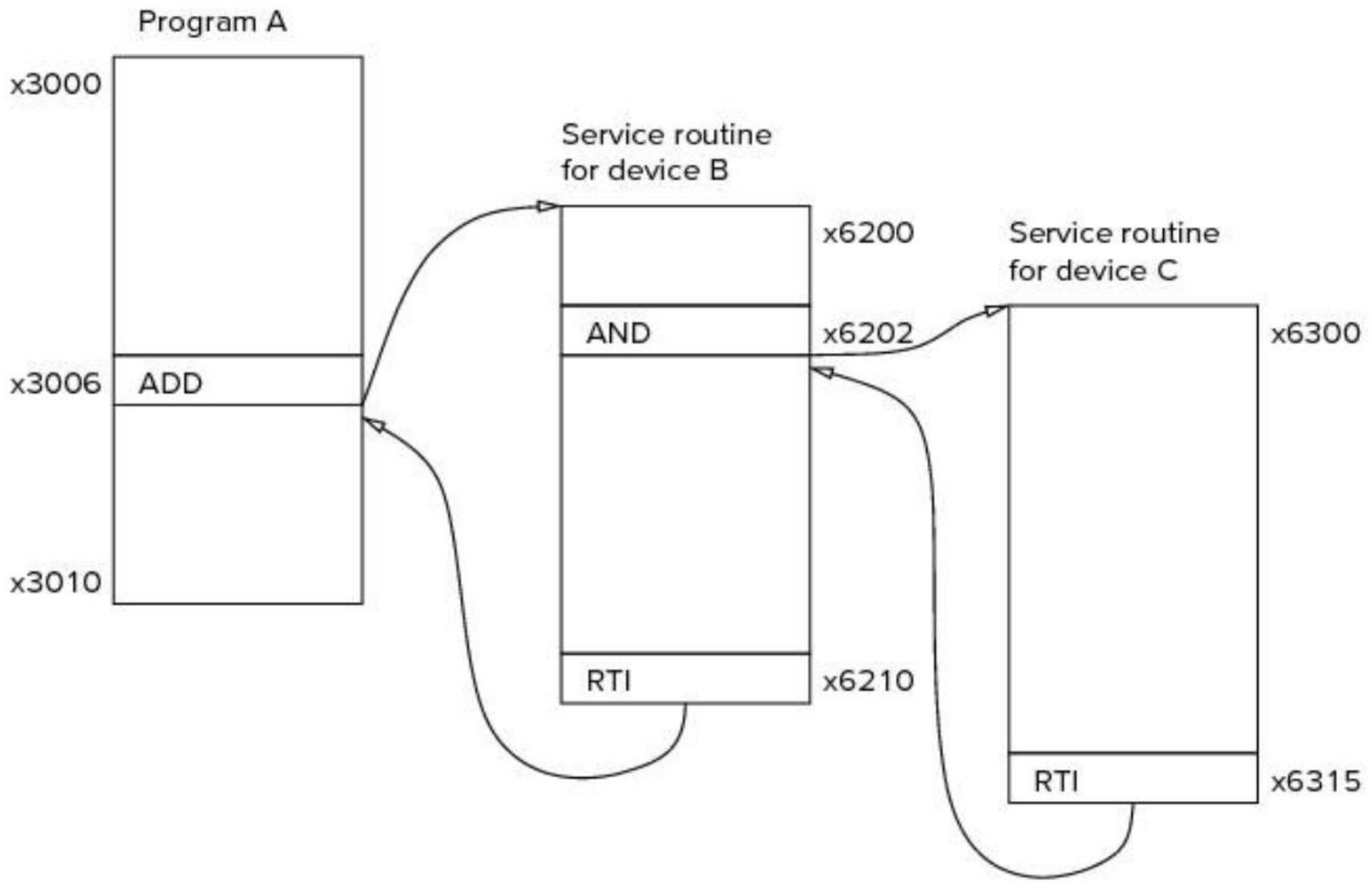
Special instruction (RTI) – restores state

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTI	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

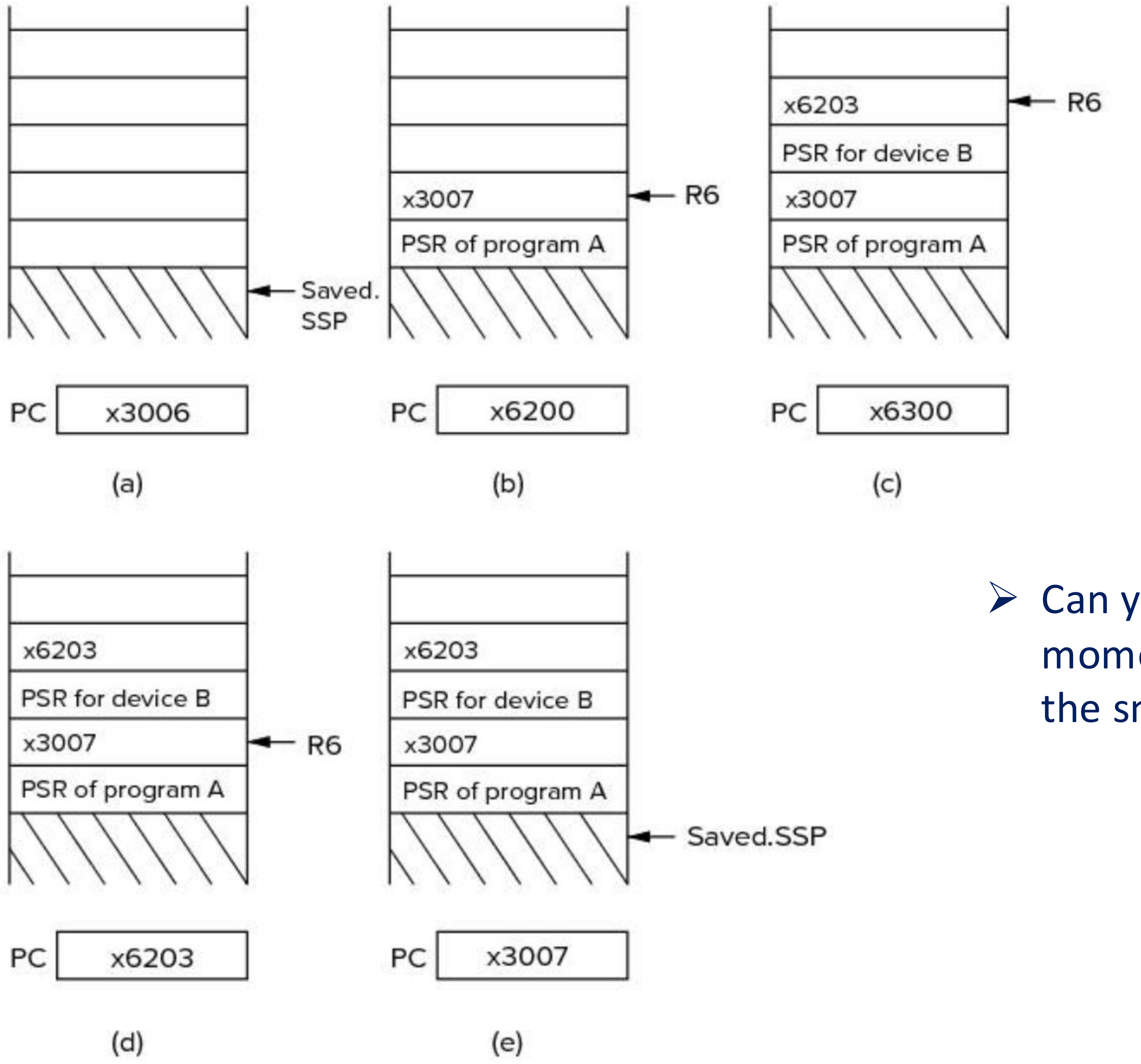
RTI is a privileged instruction.

- It can only be executed in **supervisor mode**.
- If executed in user mode, it will cause an **exception**.

Nested Interrupts & Contents on the Supervisor Stack



- What do we know about the priority levels of A, B, and C? And which mode are we in when executing A, B, and C?



- Can you identify the moments that correspond to the snapshots in (a) to (e)?

Interrupt Example

```
.ORIG   x3000
        LEA     R0,ISR_KB
        STI     R0,KBINTV           ;load ISR address to INTV
        LD      R3,EN_IE
        STI     R3,KBSR             ;set IE bit of KBSR
AGAIN   LD      R0,NUM2
        OUT
        BRnzp   AGAIN
ISR_KB  ST      R0,SaveR0           ;callee-saved R0
        LDI     R0,KBDR             ;read a char from KB and clear ready bit
        OUT
        LD      R0,SaveR0           ;callee-restored R0
        HALT
EN_IE   .FILL   x4000               ;enable IE 0100_0000_0000_0000
NUM2    .FILL   x0032               ;ASCII for '2'
KBSR    .FILL   xFE00
KBDR    .FILL   xFE02
KBINTV  .FILL   x0180               ;INT vector table address for keyboard
SaveR0  .BLKW   #1
.END
```

Exceptions: Internal Interrupt

When something *unexpected* happens *inside* the processor, it may cause an exception.

Examples of Exception in LC-3:

- Privileged operation (e.g., RTI in user mode)
- Executing an illegal opcode (Bits[15:12] = 1101)

Handled just like an interrupt:

- Vector is determined internally by type of exception
- Priority is the same as running program

Interrupt Vector Table

Exception Service Routines – x0100 to x017F

Interrupt Service Routines – x0180 to 01FF