Pomona College

Computer Science

Fall 2024: CSCI 181RT

Real-Time Systems in the Real World

Lecture 15

Thursday, October 17, 2024
Edmunds Hall 105
2:45 PM - 4:00 PM

Professor Jennifer DesCombes

Agenda

- Go Backs
- Discussion on Reading
- Lab #7 Review
- Interrupts and OS Support
- Look Ahead
- Assignment
- Action Items

Pomona College

Go Backs

- General?
- Additional Reading: Prof Alfred Hero EECS206 F02 Lect 20
- 20 Minutes Independent Research on Nyquist and Signal Aliasing
- Action Item Status
  - AI240910-2: Find recommended book on computer architecture.
  - AI240924-1: At what point as a development team grows does it make sense to have dedicated software and integration testers?

Discussion on Reading

- The Mythical Man Month
  - Chapter 16: No Silver Bullet - Essence and Accident

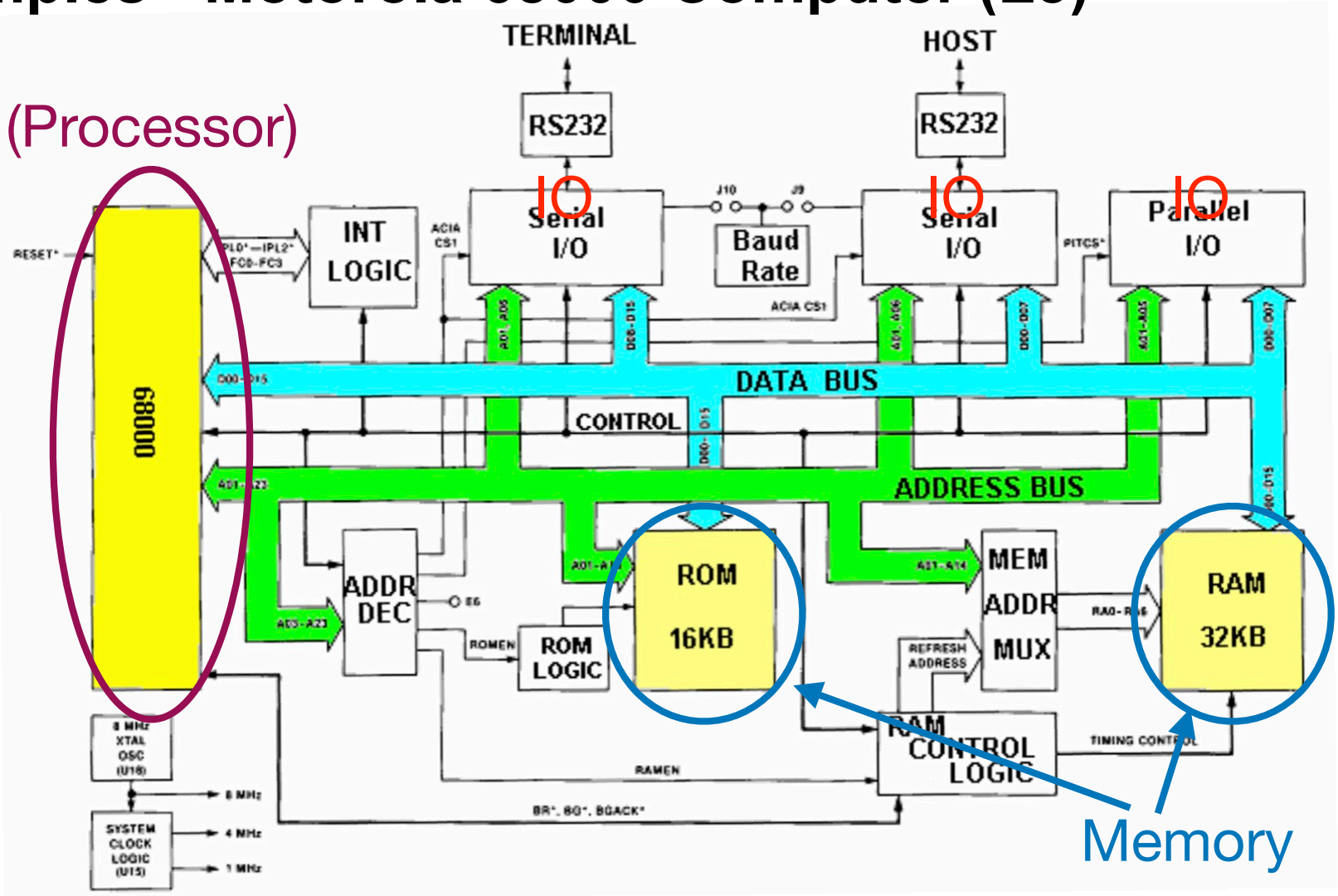
Lab #6 Review

- Goals for Lab
  - Read Digital Input (GPIO1, Connector 501-Pin 5, Processor RK4)
  - Drive LED to Match Digital Input
- Sampling Rate and Data Input Rate
  - Use Function Generator to Experiment

Interrupts and OS Support

- Disparate Concepts - Bits and Pieces
- Hardware (L5)
- Stacks and Frames (L6)
- Tasks, Stacks, and Timing (L11)
- The Assumed is Revealed - Onion Peeled (no tears)
  - OS Hardware Interrupts - Time and Timers

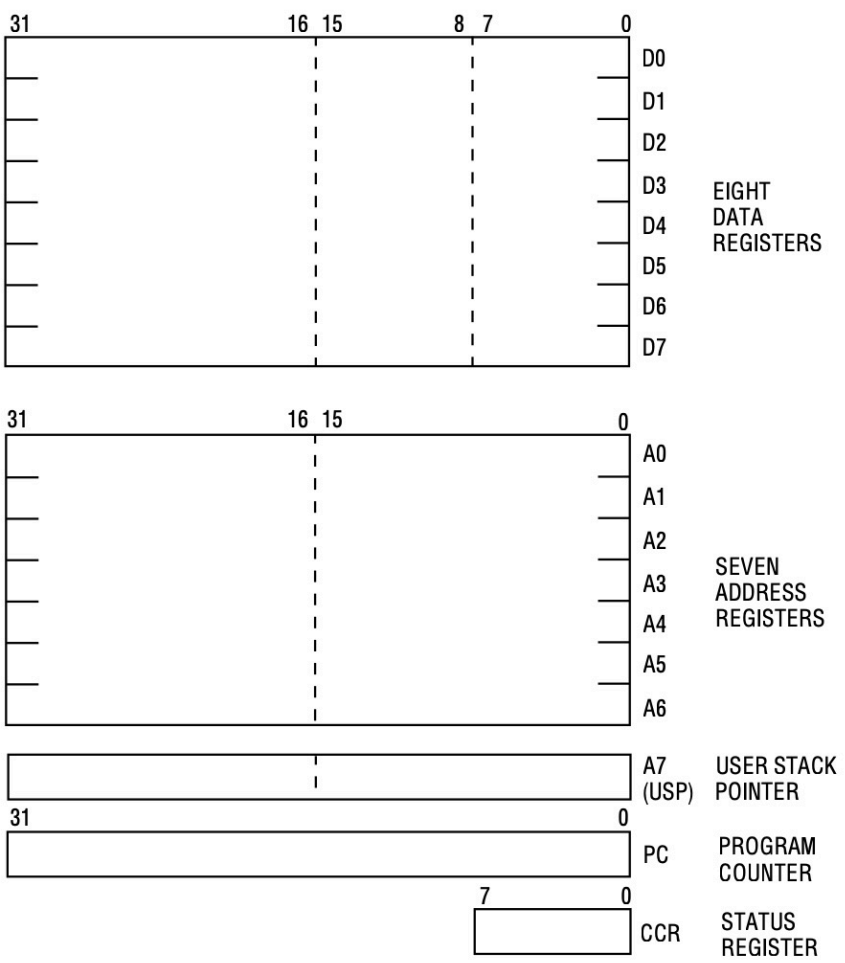
Examples - Motorola 68000 Computer (L5)



Pomona College

Examples - Motorola 68000 Processor - Registers (L5)

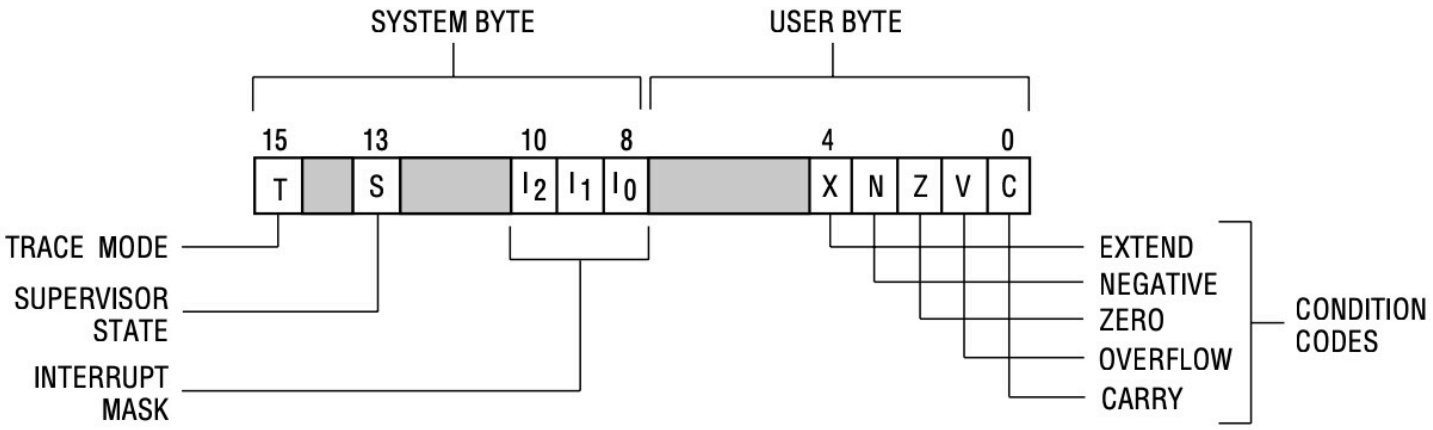
- 8 Data Registers
- 8 Address Registers
  - 7 General Purpose
  - 1 Dedicated to Stack Pointer
- Program Counter
- Status Register



Examples - Motorola 68000 Processor - Registers (L5)

2.1.3 Status Register

The status register (SR),contains the interrupt mask (eight levels available) and the following condition codes: overflow (V), zero (Z), negative (N), carry (C), and extend (X). Additional status bits indicate that the processor is in the trace (T) mode and/or in the supervisor (S) state (see Figure 2-4). Bits 5, 6, 7, 11, 12, and 14 are undefined and reserved for future expansion

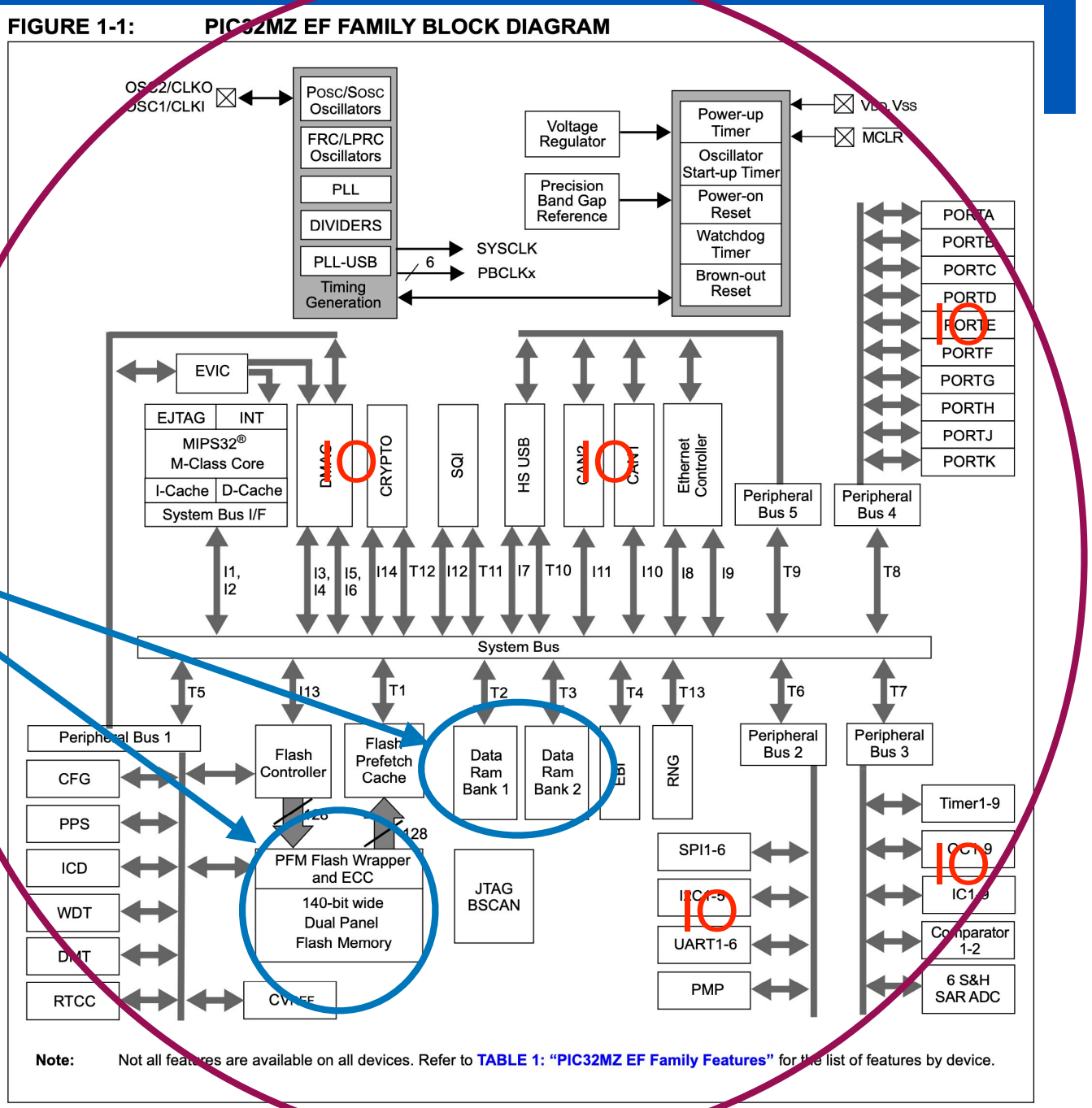


Pomona College

Examples - PIC32 Series - Chip (L5)

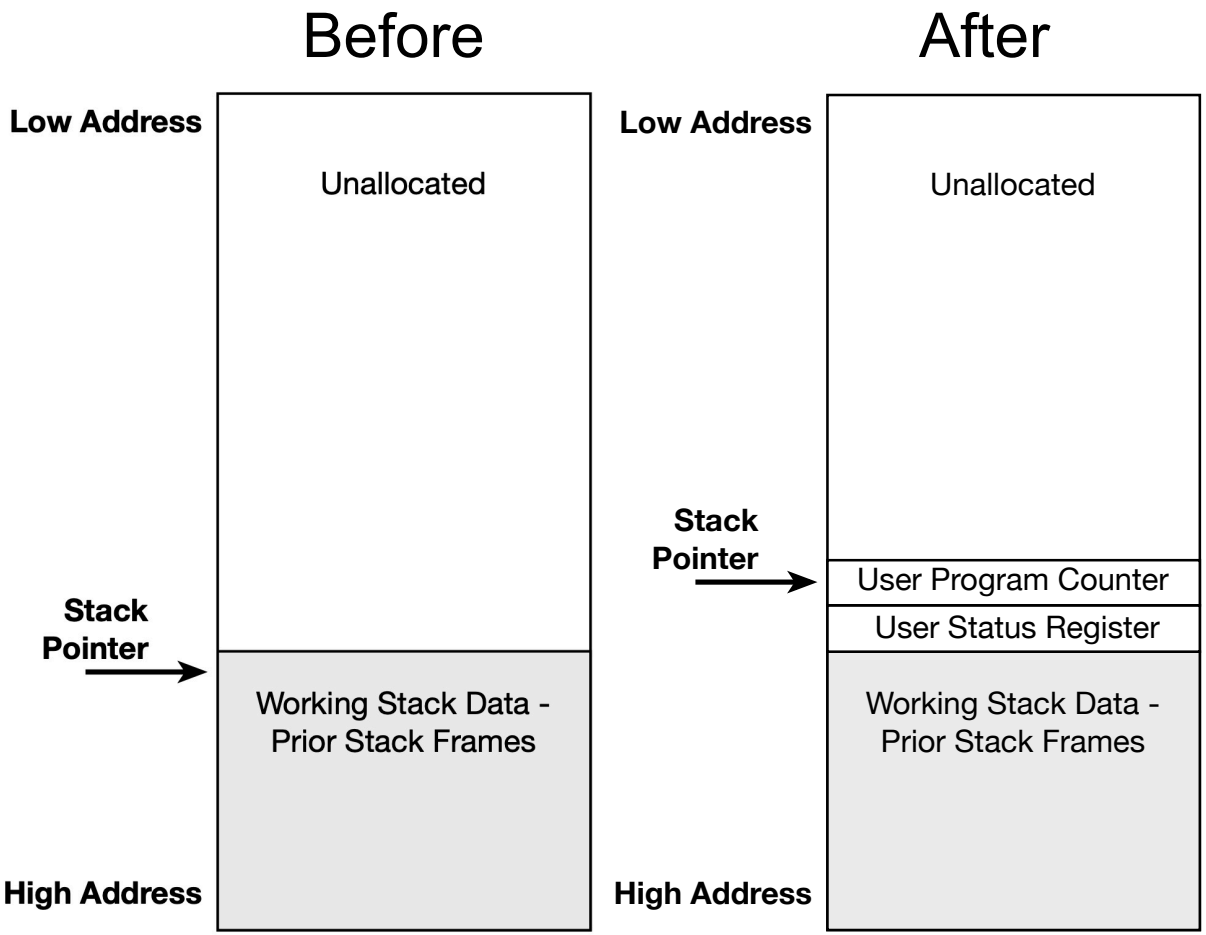
Memory

Chip (Micro-controller)



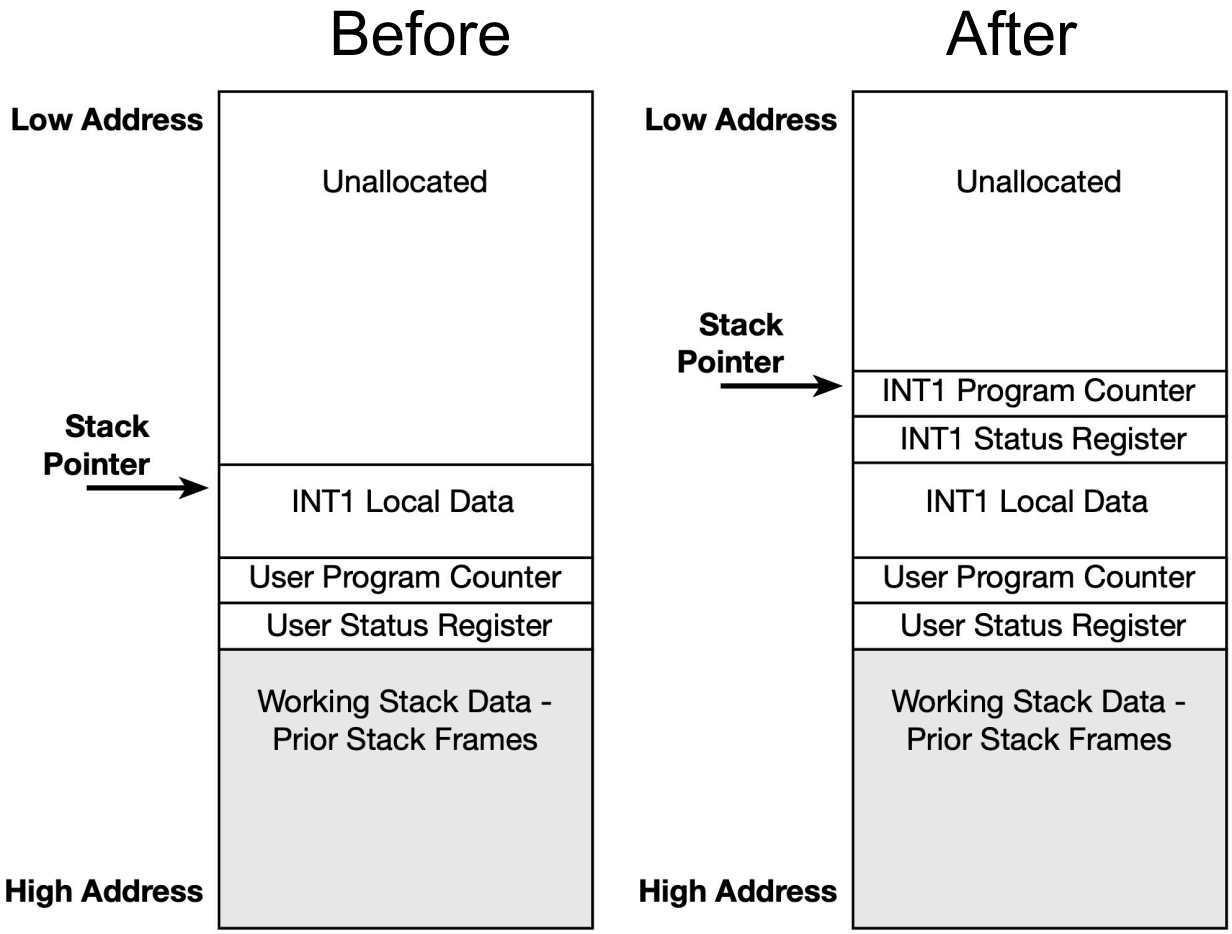
Interrupts - Single Interrupt Stack Frame (L6)

- PUSH SR
- PUSH Program Counter
- User Mode Code is Now Interrupted by INT1, Which is Now Running
- INT1 Local Data Routine Data Can Be Placed On Stack



Interrupts - Multiple Nested Stack Frames (L6)

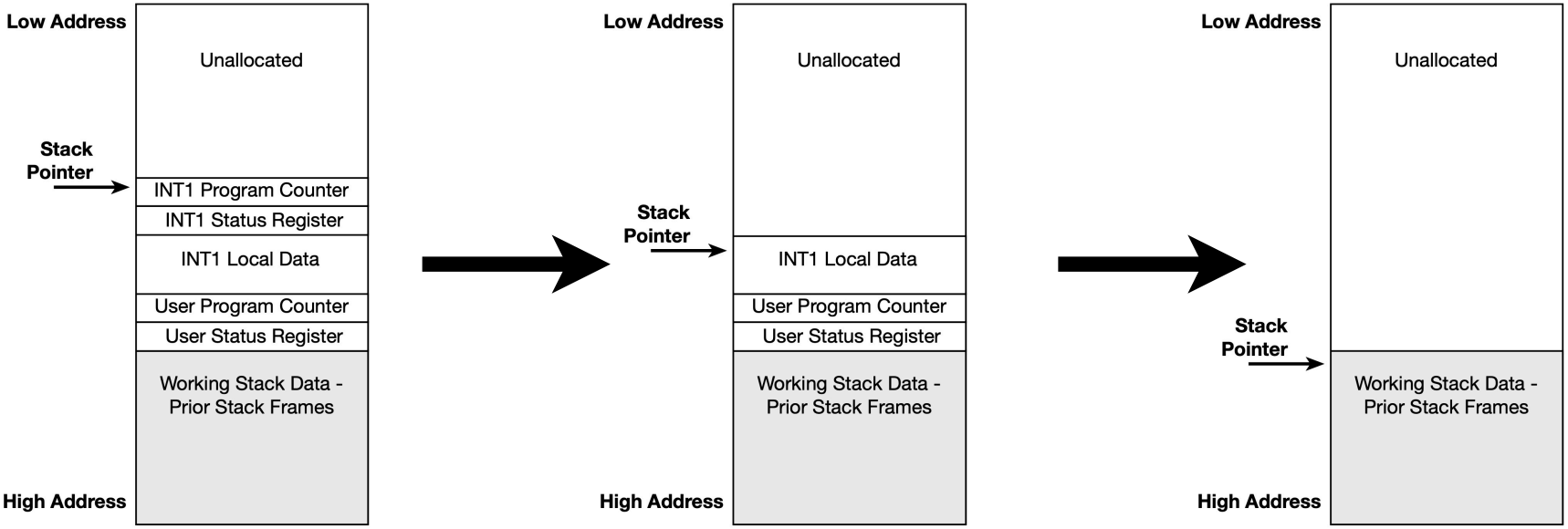
- PUSH INT1 SR
- PUSH INT1 Program Counter
- Note INT1 Local Data Storage Space on Stack
- INT1 is Now Interrupted by INT2, Which is Now Running



Pomona College

Interrupts - Unwinding and Returning (RINT) (L6)

- Use POP to Restore PC and SR
- Compiler Knows How Much Local Data to Purge
- Determine User Mode, from SR, to End Interrupt Processing



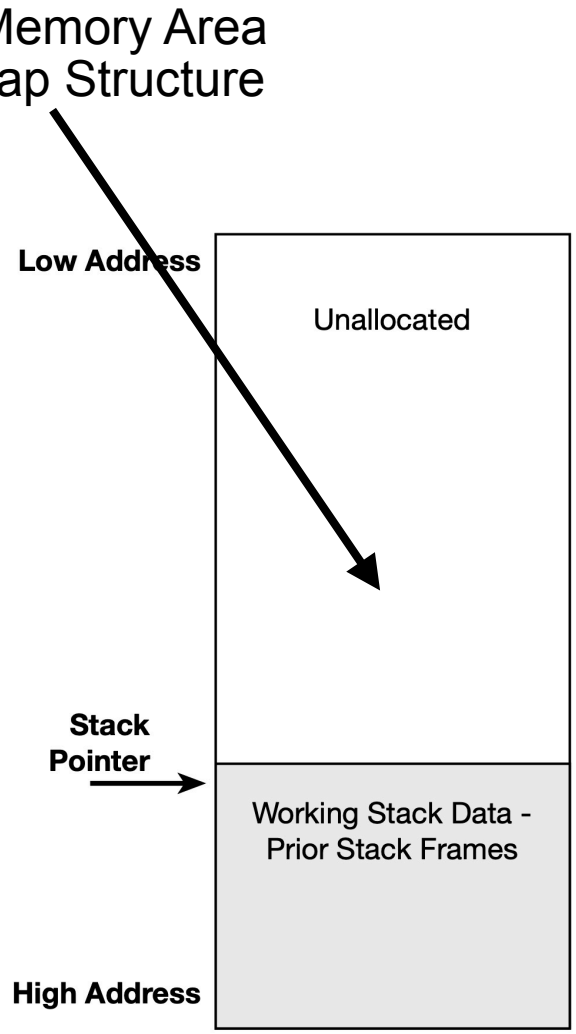
What Defines a Task (L11)

- Simple Serial Port Monitoring Task

```
// Serial Port task

#include myOSCalls.h
#include mySerialPort.h

#define true 1

while(true) {

   // Process serial port
   uartChar = readUART( );
   If (uartChar != 0)
   {
      processChar( uartChar );
   }
   myOSSleepms(5);

}
```

Endless Loop

Timing and Start / Stop Controlled by OS Calls and Events

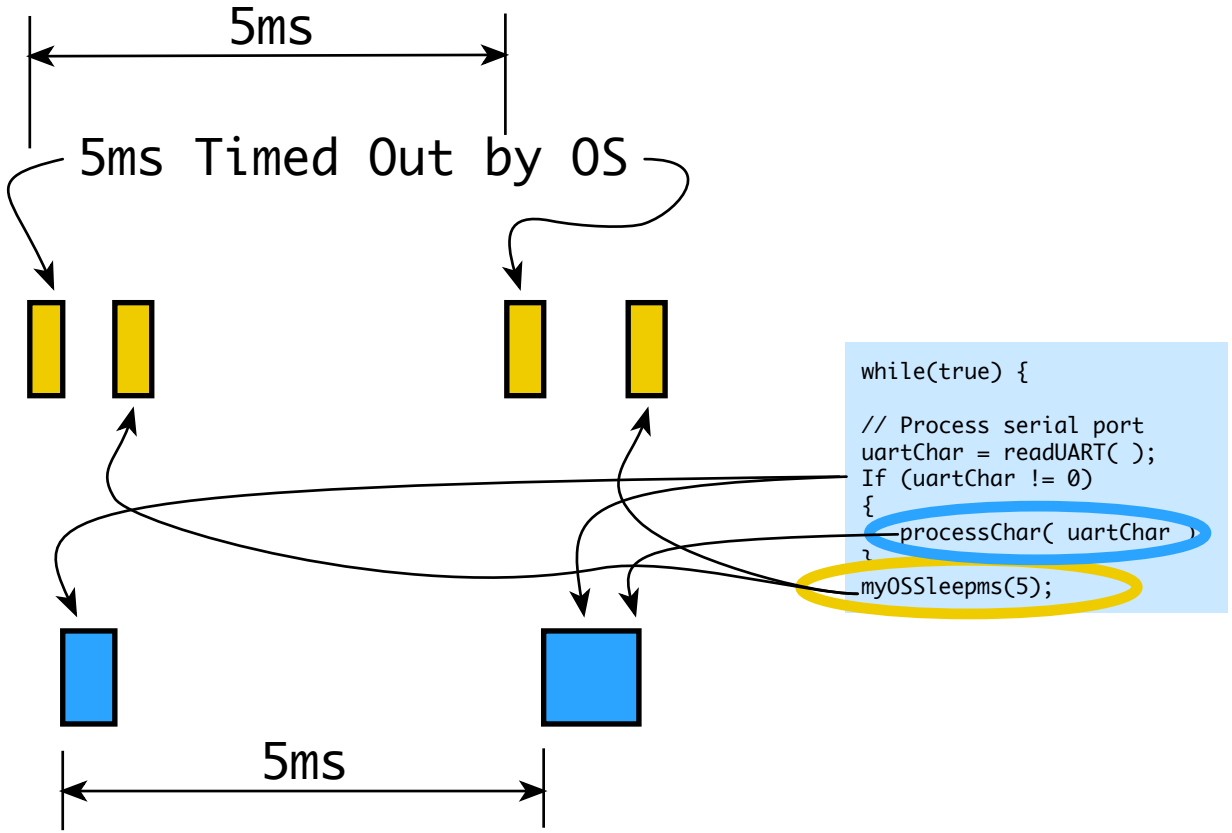


What Defines a Task (L11)

- Simple Serial Port Monitoring Task

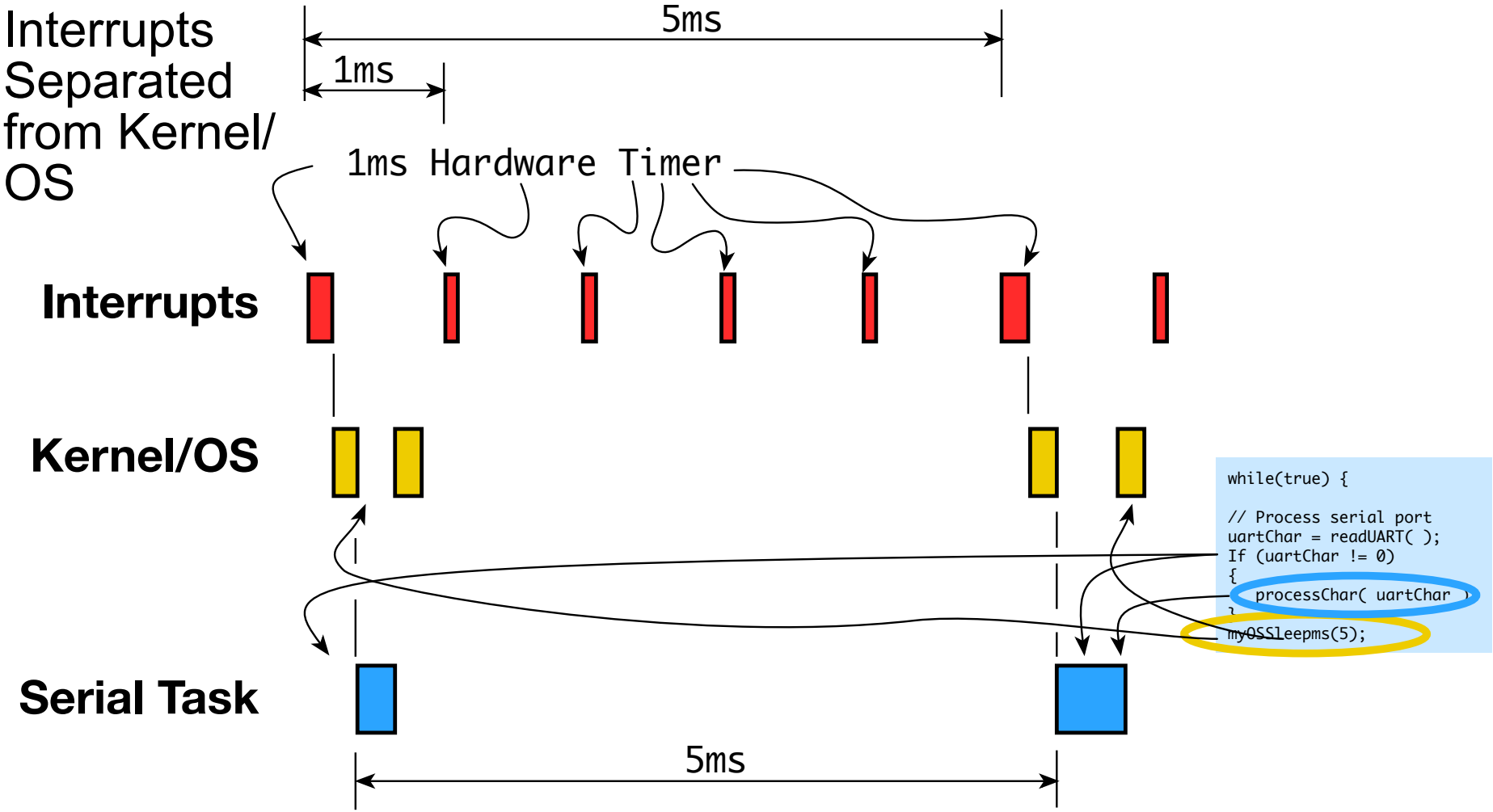
Kernel/OS/ Interrupts

Serial Task



Interrupts and OS Support

- Interrupts Separated from Kernel/ OS

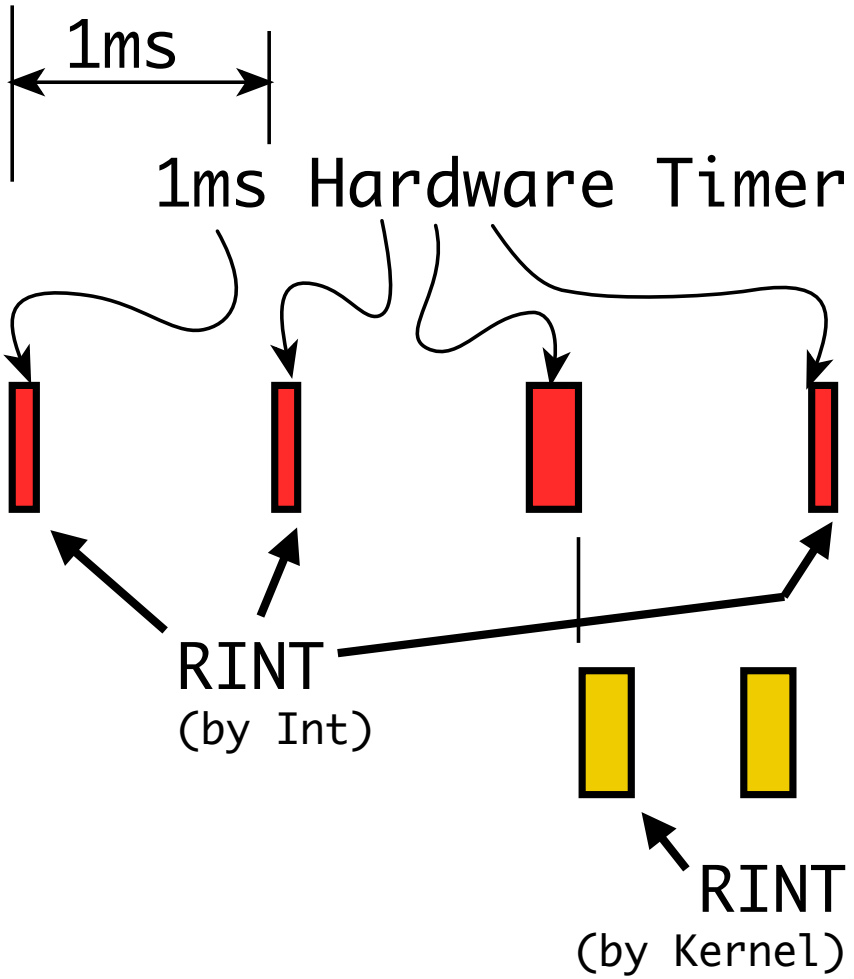


Interrupts and OS Support - Terms

- **Running -** the task is currently being executed by the processor
- **Runnable or Ready -** the task could be running, but a higher priority task is currently being executed by the processor
- **Waiting or Blocked -** the task can not be run because it is waiting for an event, data, or hardware to become available
- **Suspended -** the task has been removed from the list of tasks that can be run
- **Reschedule -** the process of determining which task should be run next based on the priority and status of the task
- **Highest Priority Runnable -** this is the task that should be currently or will be executed by the processor upon completion of a Reschedule
- **Critical Transition -** a Change in the Value of a Semaphore that Changes the Runnable or Waiting Status of a Task

Interrupts and OS Support

- Interrupt Code Determines if Reschedule is Necessary
- If Not Necessary, Interrupt Performs RINT
- Reschedule Necessary , Reschedule Performs RINT



Interrupts and OS Support - Timer Interrupt Code

- Increment Time (typically lab = timer period)
- Determine if Any Task is Waiting for This Time
  - Possible Implementations
    - Linked List
    - Que
  - Check for a Time Match
- If a Match Exists
  - Increment the Semaphore for that Task
  - Transfer Control to Reschedules (goto)
- If no Match - Nothing to Change - RINT

Interrupts and OS Support - Kernel Reschedule Code

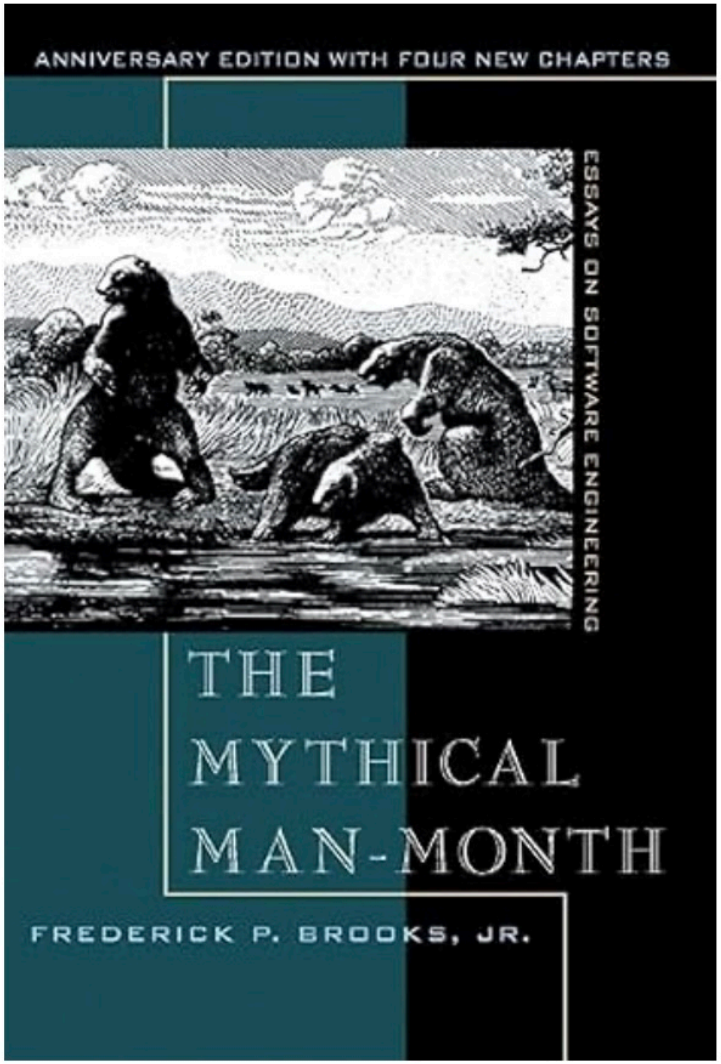
- Check for any Critical Transitions (might be done in interrupt)
- Update Runnable/Waiting Status
- Determine Highest Priority Runnable Task
- Perform Task Swap if Necessary
  - Save Registers on Prior Task
  - Stack Information for Prior Task
  - Restore Resisters on New Task
  - Update/Modify Stack Information for New Task
- RINT

Look Ahead

- Review of Reading
- Interrupts and OS Support - User and IO Devices
- Preview of Lab 8

Assignment - Readings

- The Mythical Man Month
  - Chapter 17 & 18: “No Silver Bullet” Refired, Propositions of *The Mythical Man Month*: True or False?
  - Send Me Discussion Topics by 10:00 AM on Tuesday, Oct. 22, 2024.



Pomona College

Action Items and Discussion

AI#:	Owner	Slide #	Document	Action