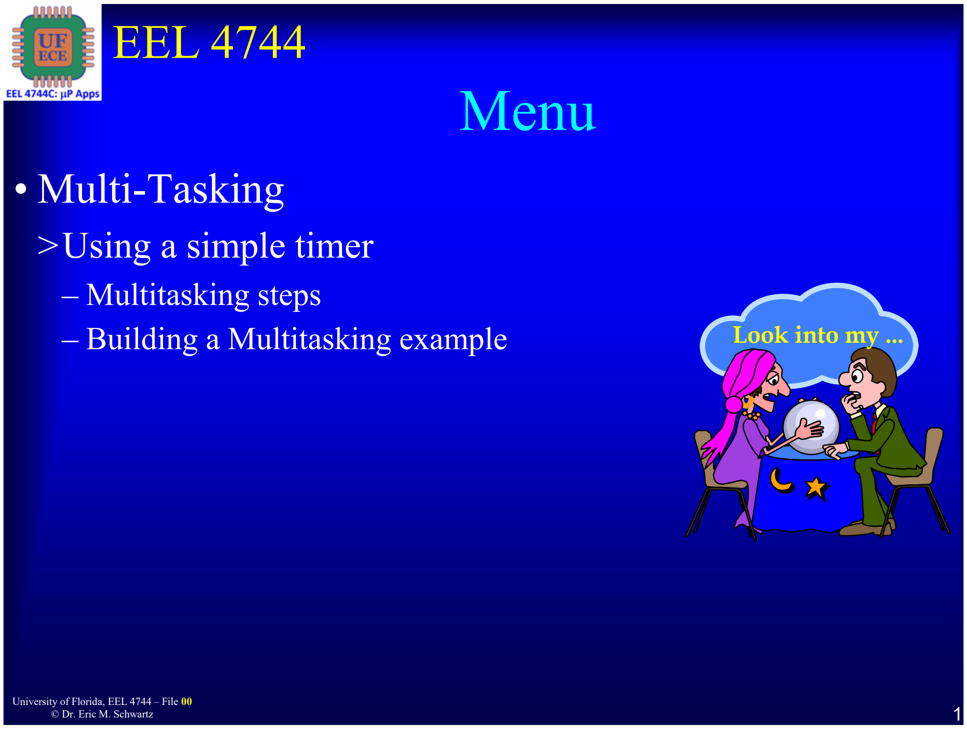
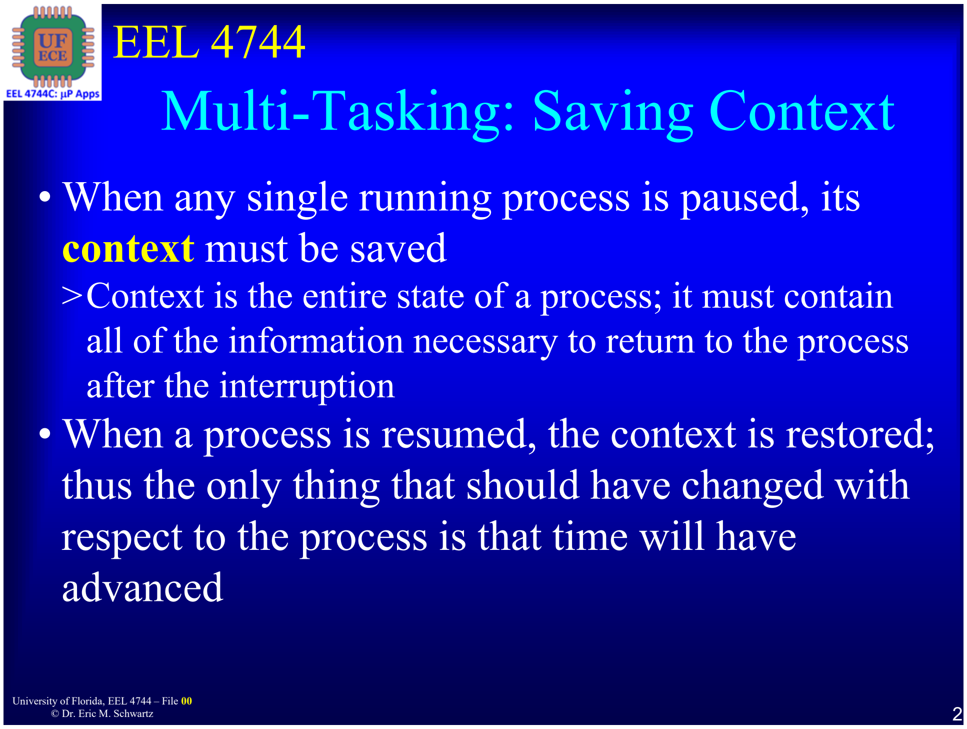


EEL 4744

Menu

- Multi-Tasking
  - >Using a simple timer
    - – Multitasking steps
    - – Building a Multitasking example

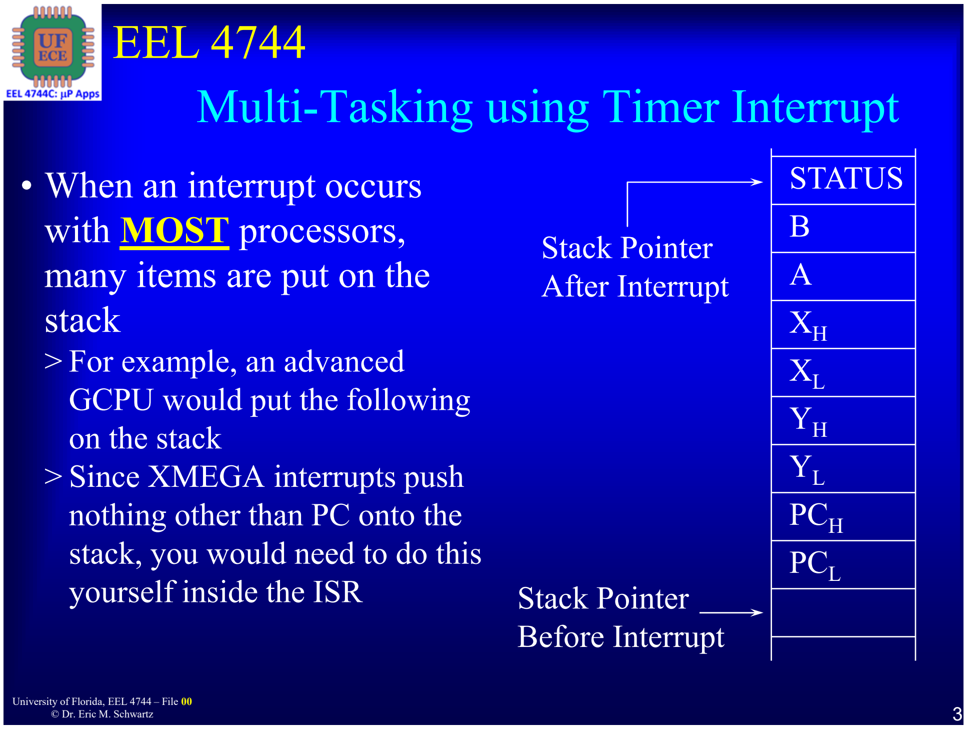
Look into my ...



EEL 4744

Multi-Tasking: Saving Context

- When any single running process is paused, its **context** must be saved
  - >Context is the entire state of a process; it must contain all of the information necessary to return to the process after the interruption
- When a process is resumed, the context is restored; thus the only thing that should have changed with respect to the process is that time will have advanced



EEL 4744

Multi-Tasking using Timer Interrupt

- When an interrupt occurs with **MOST** processors, many items are put on the stack
  - > For example, an advanced GCPU would put the following on the stack
  - > Since XMEGA interrupts push nothing other than PC onto the stack, you would need to do this yourself inside the ISR

Stack Pointer After Interrupt

STATUS

B

A

X_H

X_L

Y_H

Y_L

PC_H

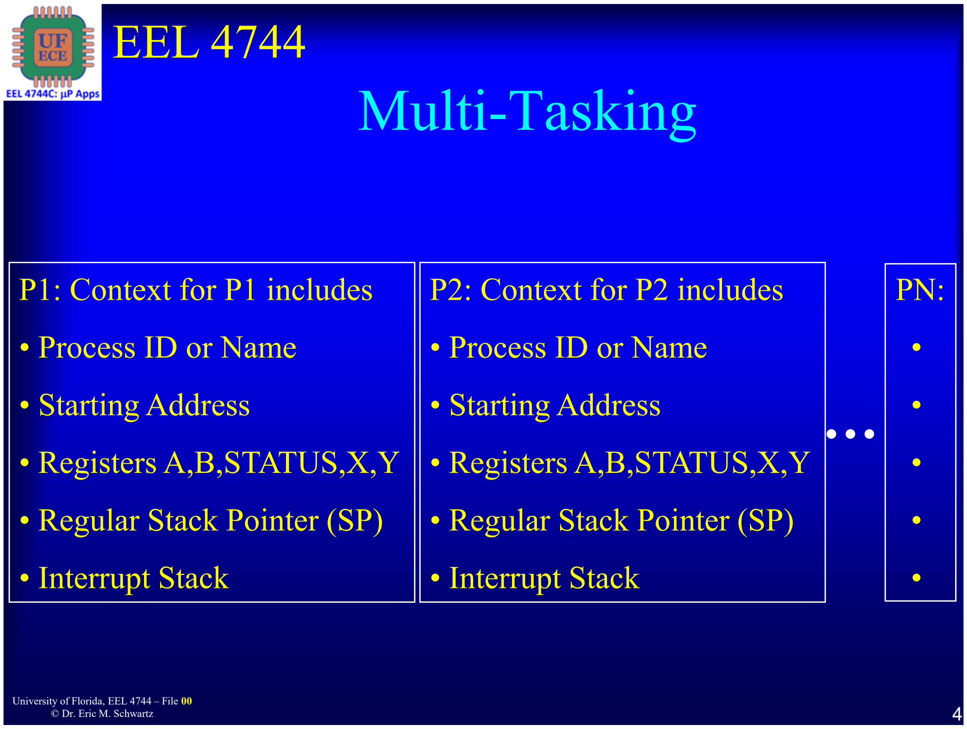
PC_L

Stack Pointer Before Interrupt

University of Florida, EEL 4744 – File 00
© Dr. Eric M. Schwartz

3

3



EEL 4744

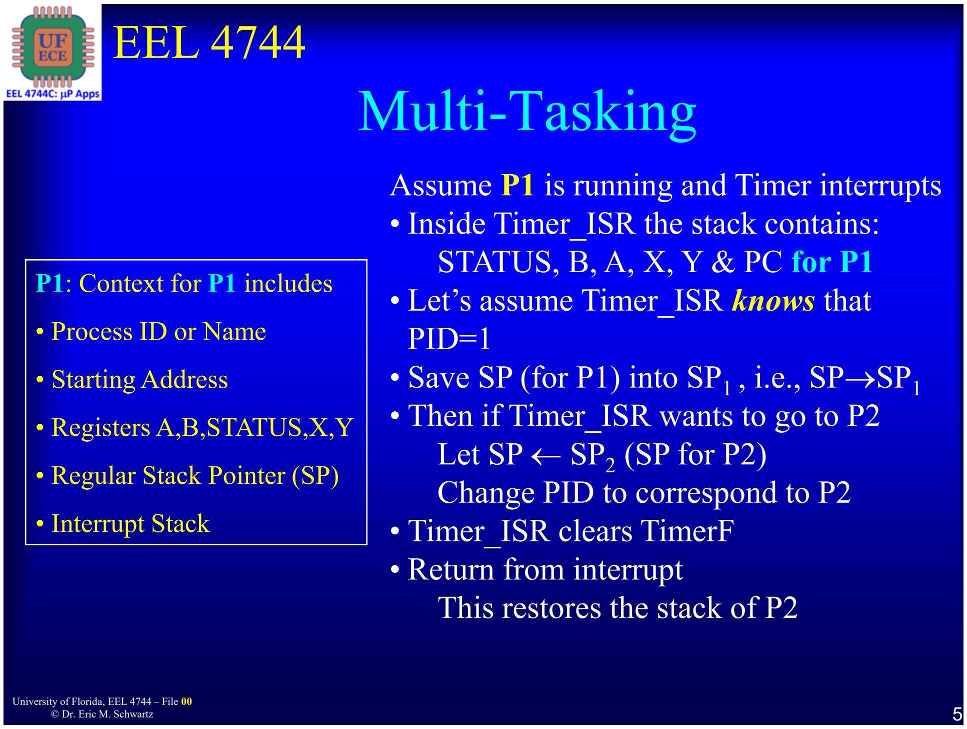
Multi-Tasking

P1: Context for P1 includes	P2: Context for P2 includes		PN:
• Process ID or Name	• Process ID or Name		•
• Starting Address	• Starting Address	• • •	•
• Registers A,B,STATUS,X,Y	• Registers A,B,STATUS,X,Y		•
• Regular Stack Pointer (SP)	• Regular Stack Pointer (SP)		•
• Interrupt Stack	• Interrupt Stack		•

University of Florida, EEL 4744 – File 00
© Dr. Eric M. Schwartz

4

4



EEL 4744

Multi-Tasking

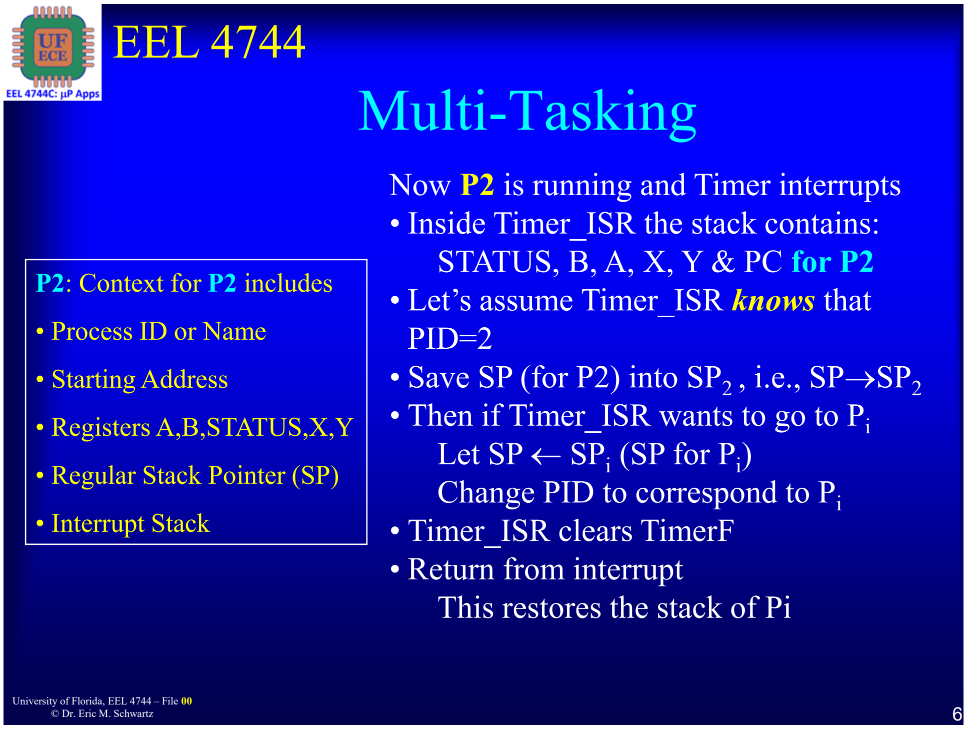
P1: Context for **P1** includes
- Process ID or Name
- Starting Address
- Registers A,B,STATUS,X,Y
- Regular Stack Pointer (SP)
- Interrupt Stack

Assume **P1** is running and Timer interrupts
- Inside Timer_ISR the stack contains:
  STATUS, B, A, X, Y & PC **for P1**
- Let's assume Timer_ISR ***knows*** that PID=1
- Save SP (for P1) into SP_1 , i.e., $SP \rightarrow SP_1$
- Then if Timer_ISR wants to go to P2
  Let $SP \leftarrow SP_2$ (SP for P2)
  Change PID to correspond to P2
- Timer_ISR clears TimerF
- Return from interrupt
  This restores the stack of P2

University of Florida, EEL 4744 – File **00**
© Dr. Eric M. Schwartz

5

5



EEL 4744

Multi-Tasking

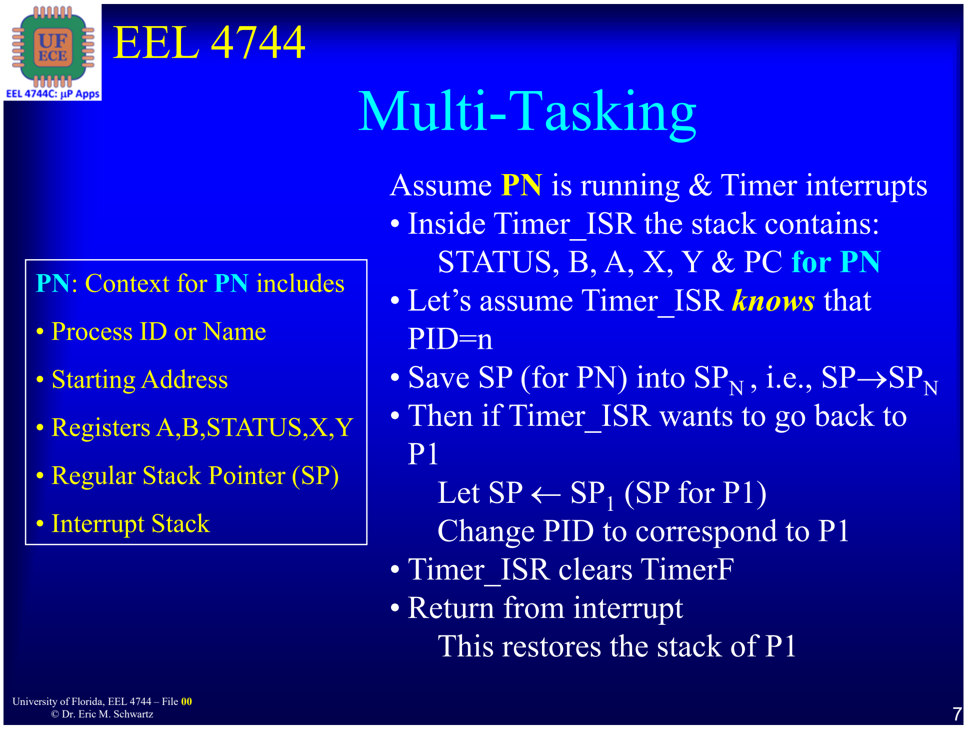
P2: Context for **P2** includes
- Process ID or Name
- Starting Address
- Registers A,B,STATUS,X,Y
- Regular Stack Pointer (SP)
- Interrupt Stack

Now **P2** is running and Timer interrupts
- Inside Timer_ISR the stack contains:
  STATUS, B, A, X, Y & PC **for P2**
- Let's assume Timer_ISR ***knows*** that PID=2
- Save SP (for P2) into SP_2 , i.e., $SP \rightarrow SP_2$
- Then if Timer_ISR wants to go to P_i
  Let $SP \leftarrow SP_i$ (SP for P_i)
  Change PID to correspond to P_i
- Timer_ISR clears TimerF
- Return from interrupt
  This restores the stack of Pi

University of Florida, EEL 4744 – File **00**
© Dr. Eric M. Schwartz

6

6



EEL 4744

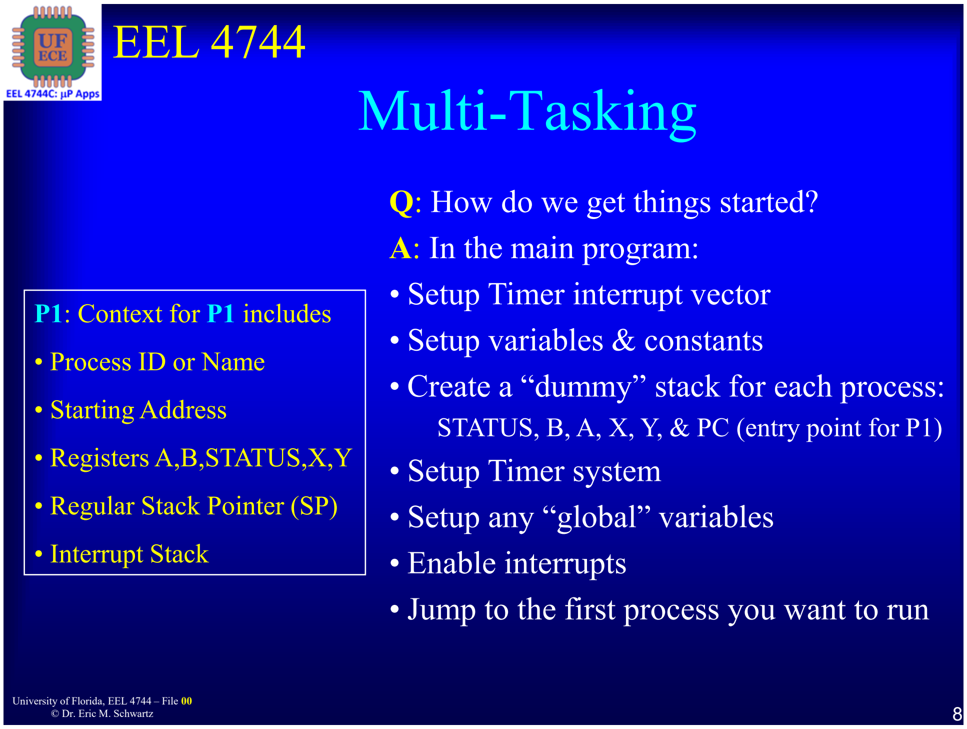
Multi-Tasking

PN: Context for **PN** includes

- Process ID or Name
- Starting Address
- Registers A,B,STATUS,X,Y
- Regular Stack Pointer (SP)
- Interrupt Stack

Assume **PN** is running & Timer interrupts

- Inside Timer_ISR the stack contains:
  STATUS, B, A, X, Y & PC **for PN**
- Let's assume Timer_ISR ***knows*** that PID=n
- Save SP (for PN) into SP_N , i.e., SP→SP_N
- Then if Timer_ISR wants to go back to P1
  Let SP ← SP_1 (SP for P1)
  Change PID to correspond to P1
- Timer_ISR clears TimerF
- Return from interrupt
  This restores the stack of P1



EEL 4744

Multi-Tasking

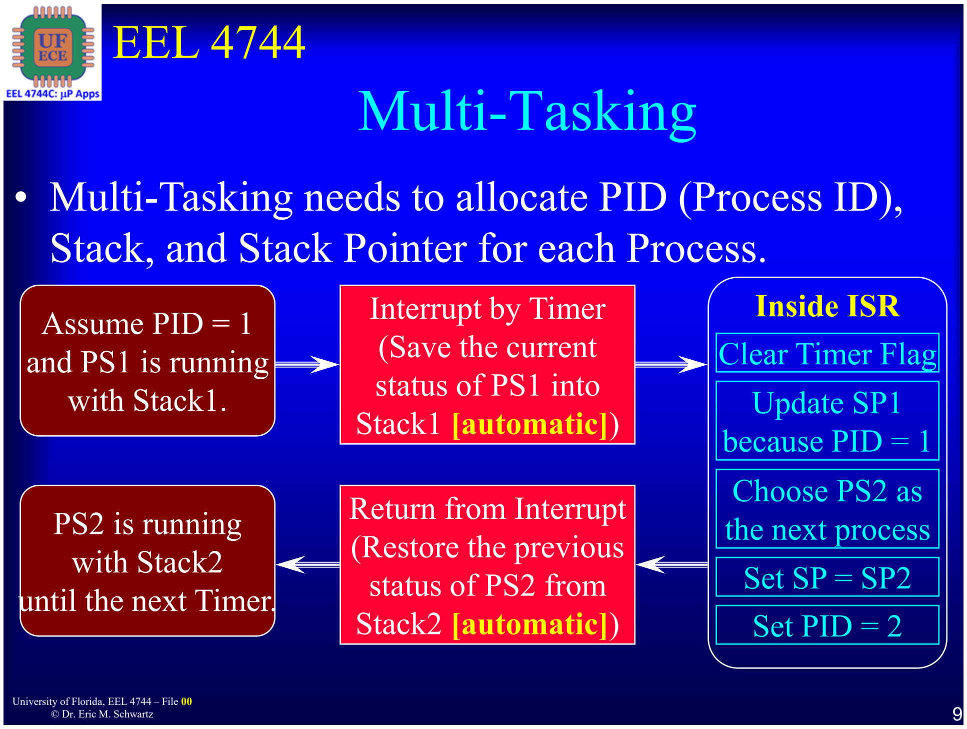
P1: Context for **P1** includes

- Process ID or Name
- Starting Address
- Registers A,B,STATUS,X,Y
- Regular Stack Pointer (SP)
- Interrupt Stack

Q: How do we get things started?

A: In the main program:

- Setup Timer interrupt vector
- Setup variables & constants
- Create a "dummy" stack for each process:
  STATUS, B, A, X, Y, & PC (entry point for P1)
- Setup Timer system
- Setup any "global" variables
- Enable interrupts
- Jump to the first process you want to run



EEL 4744

Multi-Tasking

- Multi-Tasking needs to allocate PID (Process ID), Stack, and Stack Pointer for each Process.

Assume PID = 1 and PS1 is running with Stack1.

Interrupt by Timer (Save the current status of PS1 into Stack1 **[automatic]**)

Inside ISR

Clear Timer Flag

Update SP1 because PID = 1

Choose PS2 as the next process

Set SP = SP2

Set PID = 2

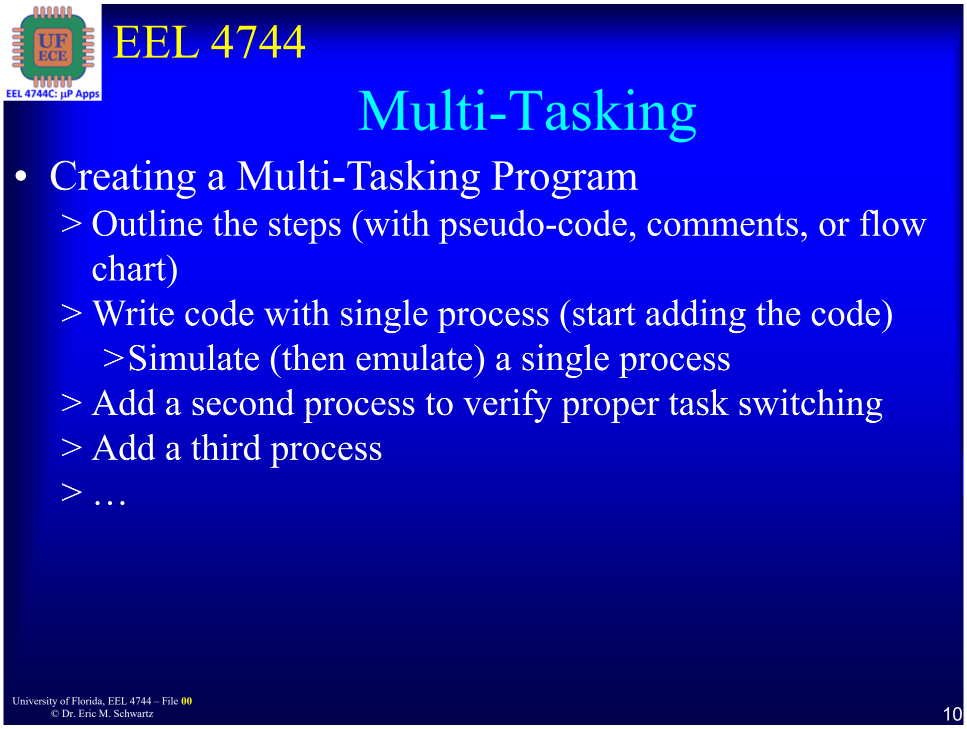
Return from Interrupt (Restore the previous status of PS2 from Stack2 **[automatic]**)

PS2 is running with Stack2 until the next Timer.

University of Florida, EEL 4744 – File 00
© Dr. Eric M. Schwartz

9

9



EEL 4744

Multi-Tasking

- Creating a Multi-Tasking Program
  - > Outline the steps (with pseudo-code, comments, or flow chart)
  - > Write code with single process (start adding the code)
    - >Simulate (then emulate) a single process
  - > Add a second process to verify proper task switching
  - > Add a third process
  - > …

University of Florida, EEL 4744 – File 00
© Dr. Eric M. Schwartz

10

10

EEL 4744

The End!

University of Florida, EEL 4744 – File **00**
© Dr. Eric M. Schwartz

11

11