Chapter 5

Real Time Control: Interrupts

Lesson 1

Interrupt Identification, Interrupt Service Enabling and Allocating Interrupt priorities

Why are interrupts important?

- Phone ring Example
- Instance of a phone call event not known
- We continue our schedule and attend to phone on an interrupt-event (ringing)

Importance of Interrupt Mechanism

- Instance of occurrence of event at an interrupt source is not known when programming
- Interrupt handling when event occurs, saves wait period of the processor for that event

Importance of Interrupt Mechanism

- Processor executes foreground program till an interrupt-event starts a service routine
- Enables servicing of multiple sources of interrupt events

8051: Interrupt Source Event Identification by *flag bit*

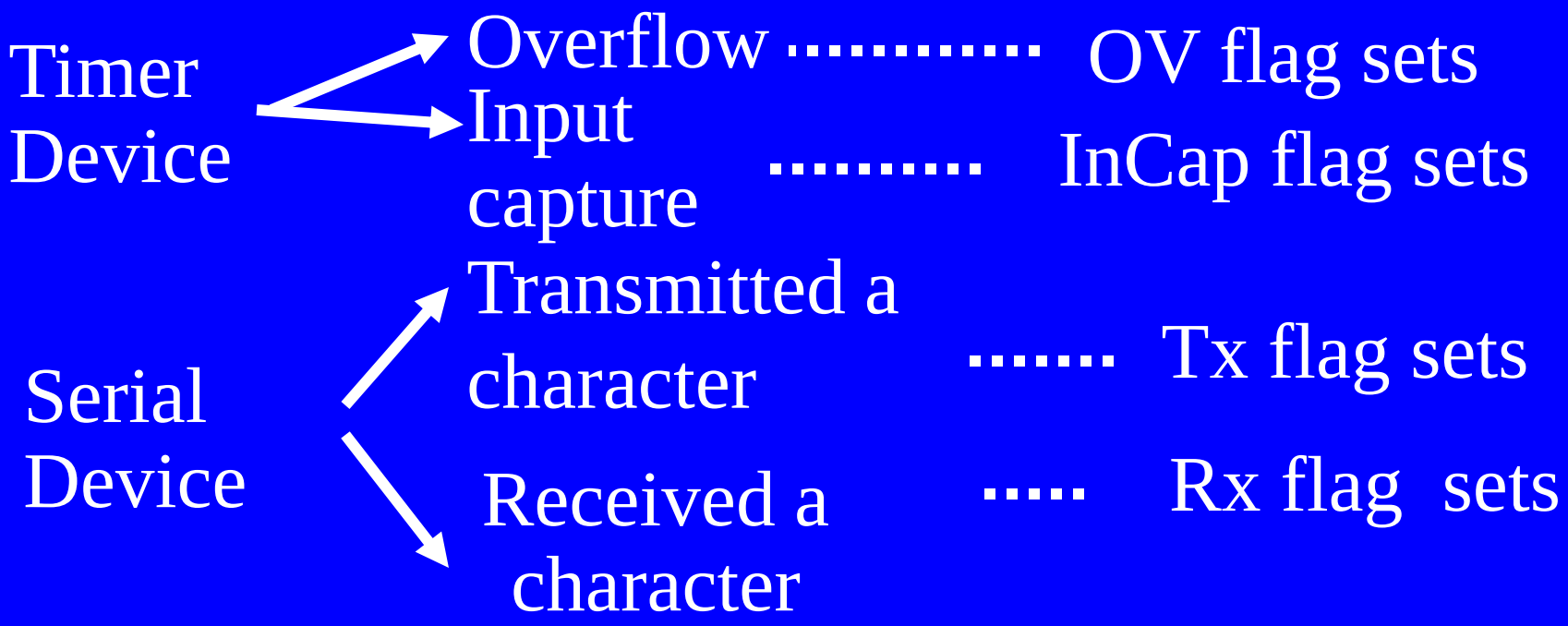
Each interrupt source of each device assigned a flag. Flag(s) shows the device status

A flag is reset at the start.

A device event sets the flag.

The flag, when sets, shows the need to the processor. The need is to service the interrupt (need of executing the service routine associated with that interrupt occurrence).

Identification By a Flag bit for an interrupt source



Events at the Interrupt Sources

Interrupt Source Event Service Need Identification by *pend* Bit

Some MCUs show a service need for executing a service routine for an interrupt source of each device by a pending bit.

A device event sets the bit when service need arises.

Service routine execution resets the interrupt pend bit.

Interrupt Types

Non Maskable interrupts

Maskable interrupts

Examples- Maskable Interrupt Sources

- Timer overflow
- Serial Transmitter buffer empty
- Serial Receiver buffer full
- Serial Receiver FIFO half full
- ADC conversion over
- Real time clock ticks

Examples- Nonmaskable Interrupt Sources

- Illegal opcode
- Software interrupt
- RAM parity check error
- Clock failure

Masking the Interrupt Sources

- Software can mask servicing of the Maskable interrupt event so that the current routine continues when mask bit set or enable bit reset

Primary level mask

Software can enable all or disable all services to all the Maskable interrupt events by setting *enable all* bit (EA = 0) or resetting EA bit, called primary level mask bit.

Secondary level Mask

Software can enable or disable a service to a specific maskable interrupt event by setting a specific *enable* bit or resetting bit, called secondary level mask.

Primary and Secondary level Masks

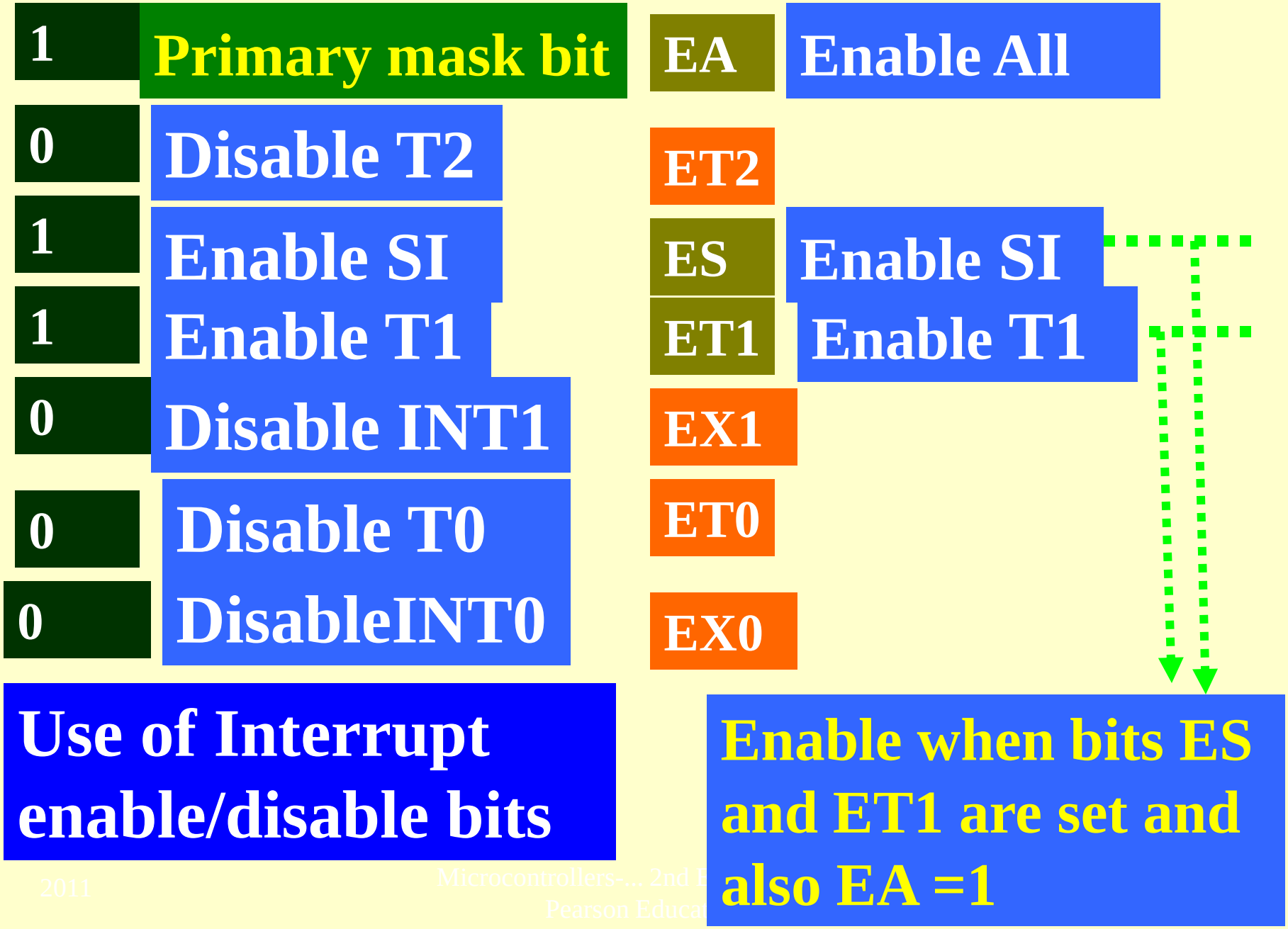
- Primary mask bit disables services of all maskable interrupts
- Secondary mask bit disables service for an individual maskable interrupt event

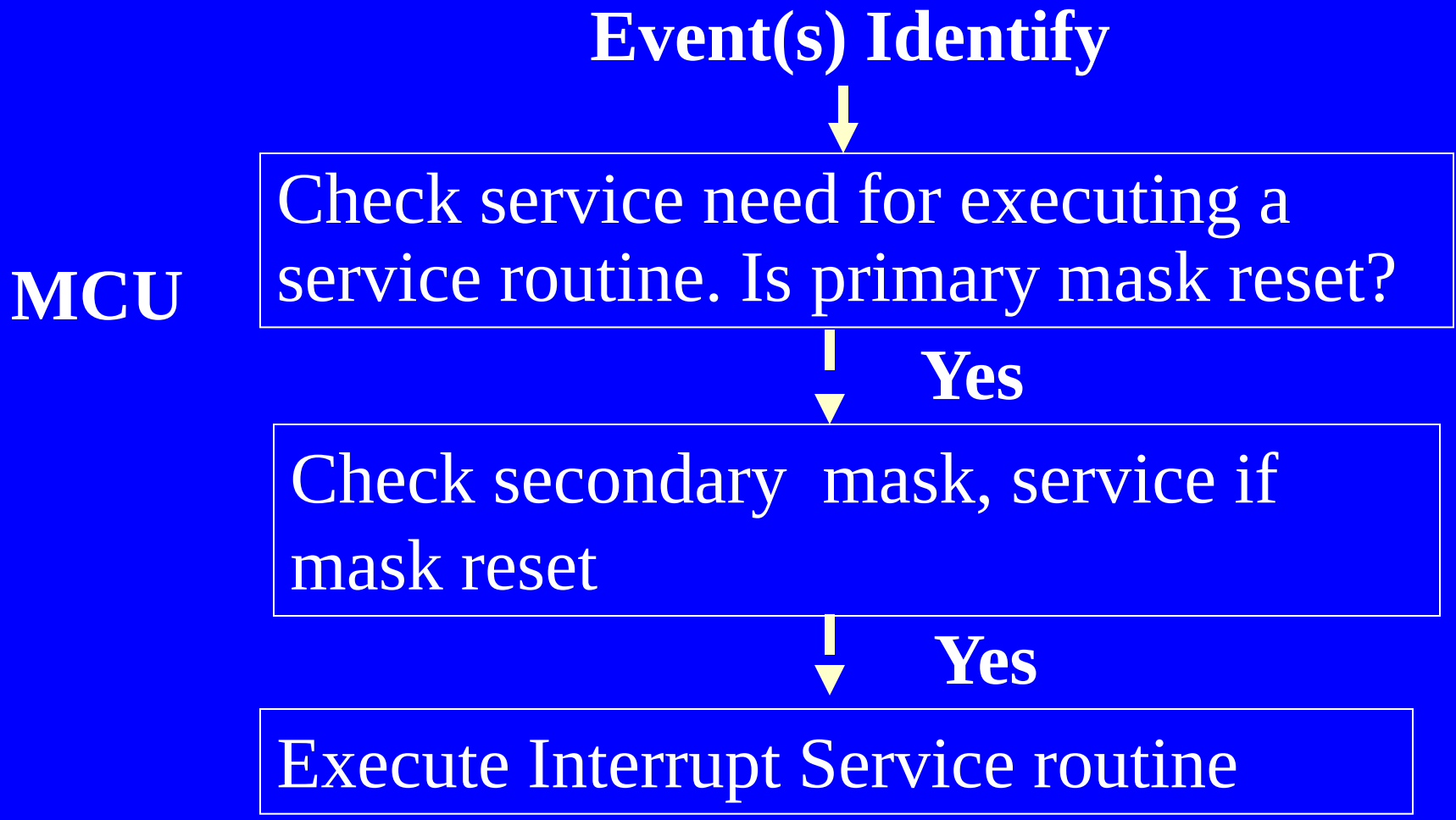
8051/52 Interrupt Enable Register (IE)

Primary mask bit
Enable all

Secondary level Interrupt masks

IE Register


1
Primary mask bit
EA
Enable All
0
Disable T2
ET2
1
Enable SI
ES
Enable SI
1
Enable T1
ET1
Enable T1
0
Disable INT1
EX1
0
Disable T0
ET0
0
DisableINT0
EX0
Use of Interrupt enable/disable bits
Enable when bits ES and ET1 are set and also EA =1


Event(s) Identify
MCU
Check service need for executing a service routine. Is primary mask reset?
Yes
Check secondary mask, service if mask reset
Yes
Execute Interrupt Service routine

Priority of interrupts

An MCU Interrupt events

- MCU Assigned Internal Priority
- User Assigned new priority

MCU Internally Assigned Priorities

8051 Interrupts Defaults Priorities

High

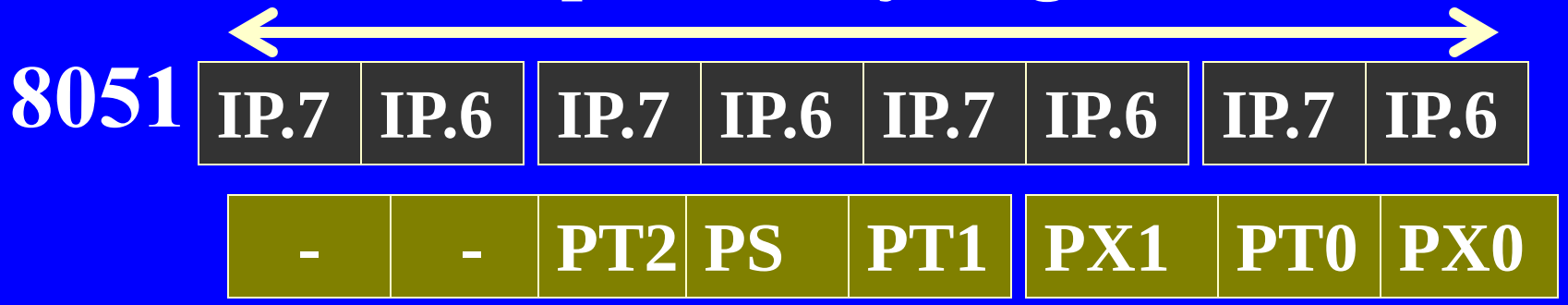
INT0	INT0 pin interrupt
T0	T0OVF interrupt
INT1	INT0 pin interrupt
T1	T1OVF interrupt
RI or TI	RI or Tx complete at SI
T2	T2EX pin-ve edge capture/reload interrupt
SI Synch mode	Synchronous Serial Device mode interrupt

Low

User Assigned Priorities

IP

interrupt Priority Register



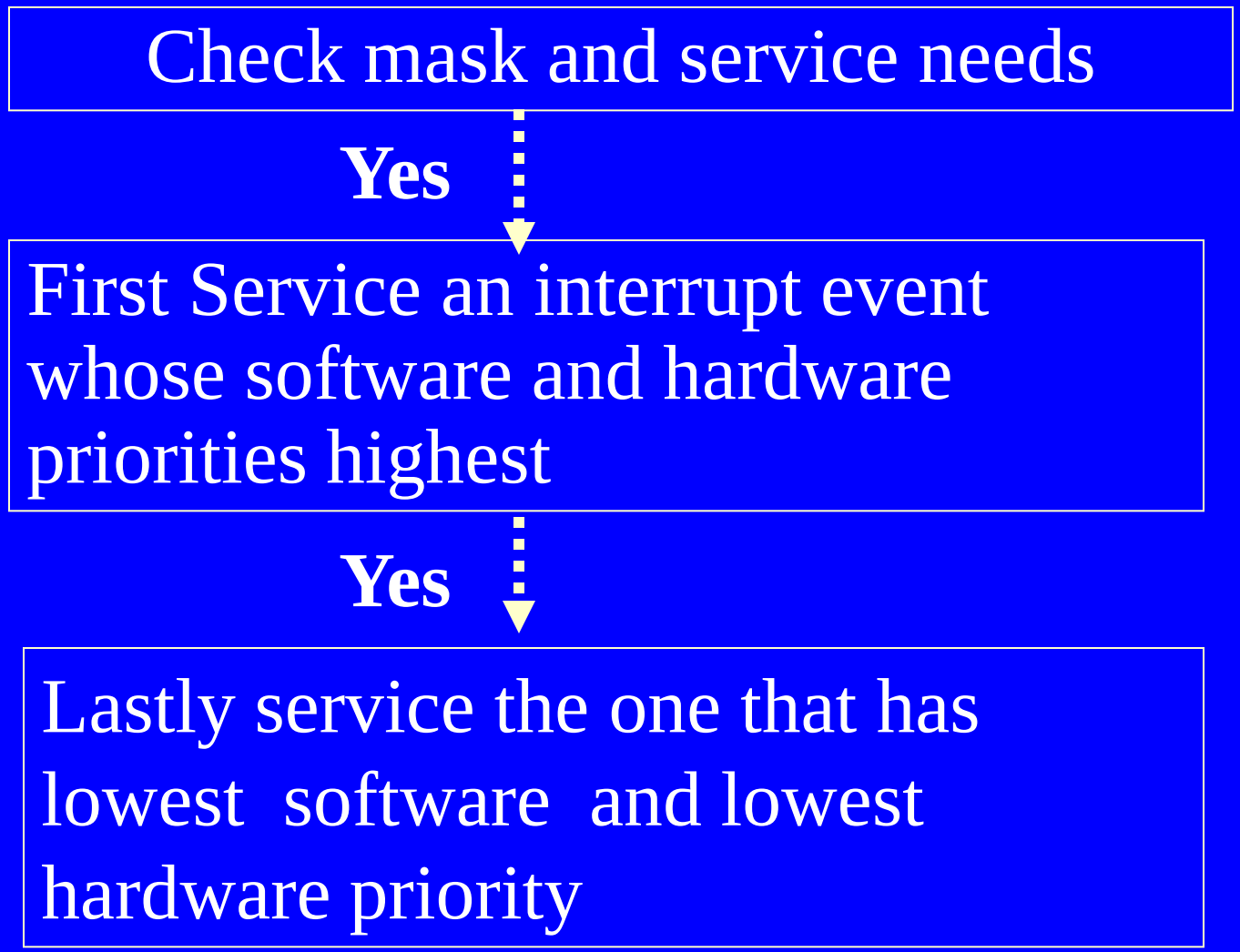
Assigning Serial Device and INT1 interrupts Higher priorities for service

= 0	Low		PT2	Priority T2
1	High	.....	PS	Priority SI
0	Low		PT1	Priority T1
1	High	.....	PX1	Priority INT1
0	Low		PT0	Priority T0
0	Low		PX0	Priority INT0

User assigned IP register bits

Priority Set High for SI and INT1

Software assigned priority overriding the hardware priority



Summary

We learnt

- Interrupts from Devices and events are important
- A Flag identifies an interrupt source
- Primary mask bits disable services of all maskable interrupts

We learnt

- Secondary mask bit disables service for an individual maskable interrupt event
- Hardware assigns default priorities
- Software priority override a hardware priority