Chapter 5

Real Time Control: Interrupts

Lesson 2

Interrupt Servicing Routine

Interrupt Service Routine Execution Start in an MCU

Source(s) Identify,

↓

check enable bit(s)

↓

Save present context and generate ISR_Vect_Addr as per the source(s)

↓

Get ISR_address

↓

PC = ISR_ address, Start Execute Interrupt Service routine

Saving Context Before Interrupt Servicing Start - 1

8051

1 SP ← SP + 1 →

2 At SP pointed memory, save LSB of PC

3 SP ← SP + 1 →

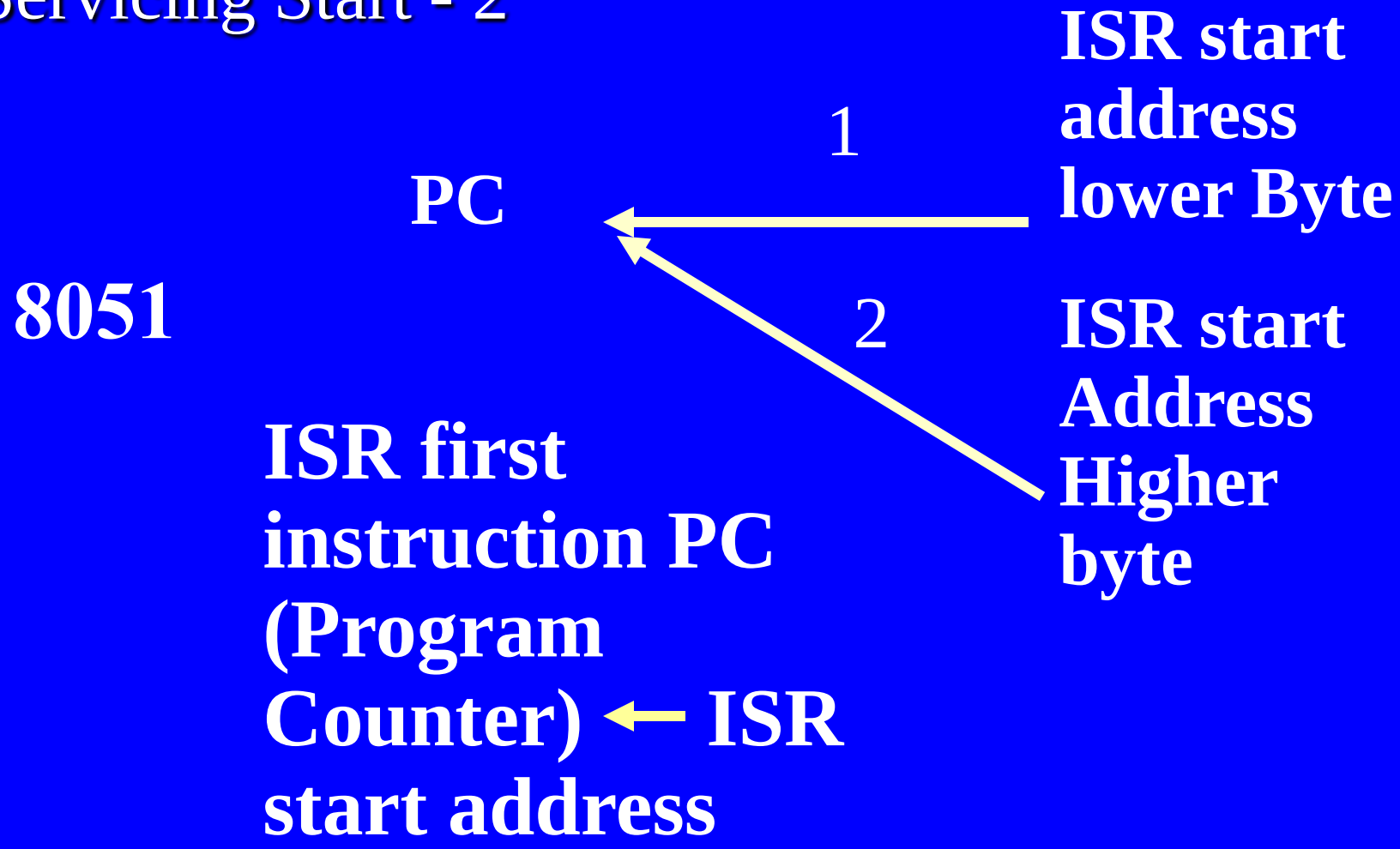
4 At SP pointed memory, save LSB of PC

5. Next instruction PC (Program Counter) saved on stack. SP is now SP+02H

Context

- Program Counter (PC) for next instruction—must save context of a running routine.
- Saves on interrupt if interrupt not masked
- Some MCUs – PC saves on an interrupt
- Some MCUs – PC and CPU registers also considered as context and save on an interrupt

Address generation for the ISR on Interrupt Servicing Start - 2

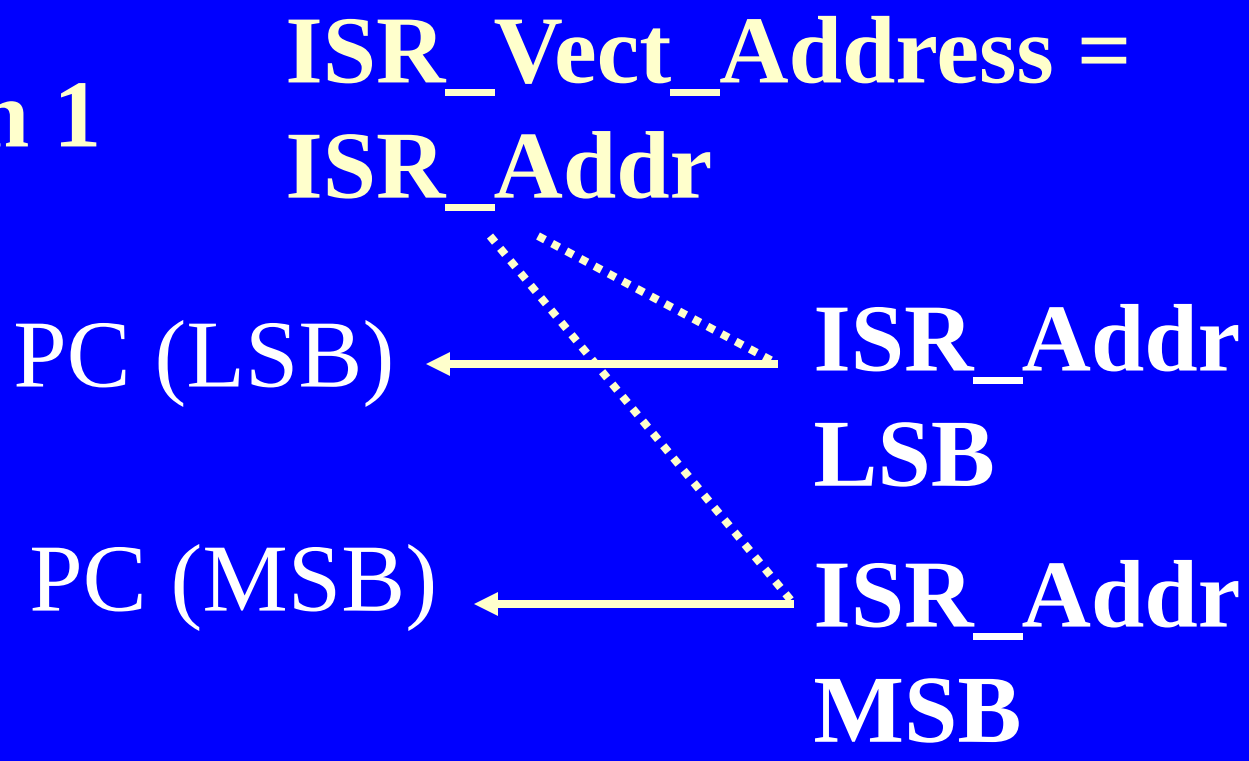


Interrupt Service Routine Address (ISR_Address)

For a specific interrupt-source or for specific source groups, the processor vectors to an address, ISR_Vect_Address

Vectoring to an ISR_Vect_Address

Option 1



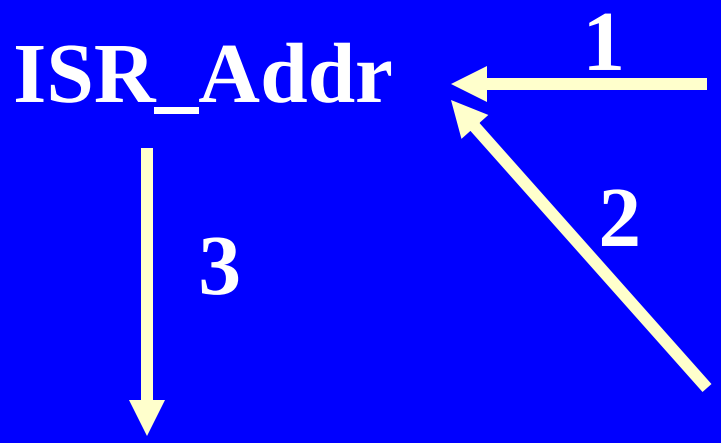
8051 ISR_Vect_Addr = ISR_addr

High priority	Assigned ISR start addresses
INT0	0003-000AH
T0	000B-0012H
INT1	0013-001AH
T1	001B-0022H
RI and TI	0023-002AH
T2	002B- onwards, before 0053H
SI Synch mode	0053H onwards

Low priority

Interrupt Servicing Start - 2

Option 2
80x96



ISR_Vect_Addr address lower Byte

ISR_Vect_ Addr Address higher byte

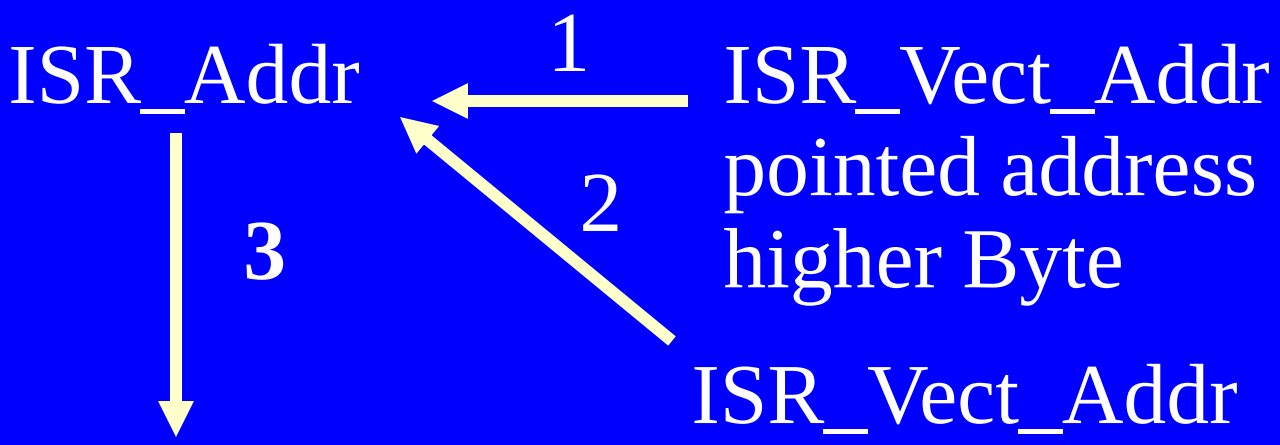
ISR first instruction PC (Program Counter) ←

As pointed address at ISR_Vect_Addr for the ISR start address

Interrupt Servicing Start - 2

Option 2

68HC11



ISR first instruction PC (Program Counter) ← As pointed address at ISR_Vect_Addr for the ISR start address

Option 1 case Interrupt Vector Address = ISR_Addr Advantage

- Processor uses the the ISR_Vect_Addr addresses for the ISRs and therefore internally defined ISR addresses used (for example, 8051)

Interrupt Vector Table - option 2 case

A table (set of memory addresses) used for the pointers ISR_Vect_Addr for the ISRs of various interrupt sources in an MCU

80x96 lower and upper vector tables are at 2000H-2013H and 2030H-20F0H

68HC11 vector table is at FEC0-FFFEH

Option 2 case Interrupt Vector Table Advantage

- Processor uses the pointers at the ISR_Vect_Addr addresses for the ISRs
- ISR executes from a pointed address only and that is a programmer defined address for a specific interrupt source.
- Option 2 Examples , 80x96, 68HC11

Interrupt flag user reset/auto reset Option 1

- Interrupt Pending Bit (flag)
- *Always reset* when ISR starts.

Interrupt Flag

- *May or may not reset* when ISR starts.

8051/52 Interrupts Two Cases

- *Auto reset* on start of ISR if interrupt vector is not common to a group of interrupt sources, for example, TF0 in 8051

- *Does not reset* when Interrupt vector ISR_Vect_Addr is common. For example, serial receiver and transmitter interrupts, hence TI transmitter flag does not auto reset on serial source ISR start.

8051/52 Interrupts Two Cases

Timer0 Overflow Interrupt

On ISR start, flag TF0, auto reset

On ISR start, flag TI, does not auto-reset

Serial Transmitter complete Interrupt

8051/52 Interrupts Four Cases

RI flag → Clear by CLR RI instruction

TI Flag → Clear by CLR TI instruction

TF0 Flag → Clear by itself on ISR start

TF1 Flag → Clear by itself on ISR start

Interrupt Servicing End - Last Step RETI instruction

8051

1

From SP pointed memory, save byte in PC (LSB)

2

SP ← SP–1

3

From SP pointed memory, save byte in PC (MSB)

4

SP ← SP + 1

Interrupt Servicing End - Last Step RETI instruction in 8051/52

- Next instruction PC (Program Counter) retrieved from stack.
- SP now SP − 02H, same as before ISR start.
- Now check if any other interrupt flag set, if yes, check if not masked, if yes then processor takes steps for highest priority pending interrupt service

Summary

We learnt

- How does Interrupt Service Start?
- How does Context Save?
- Vectoring to ISR_Vect_Addr
- ISR_Addr = ISR_Vect_Addr in 8051
- ISR_Vect_Addr is pointer to = ISR_Addr in 80x96, 68HC11
- Flag must reset to enable next identification of next event from same interrupt source

We learnt

- Retrieve context
- On Return from ISR to interrupted Routine Service a next pending ISR or continue with the previously running routine