DEVICE DRIVERS AND INTERRUPTS SERVICE MECHANISM

Lesson-7: Interrupt Vector mechanism

Interrupt Vector

- A memory address INT_VECTAddr to which processor vectors (transfers into program counter or IP and CS registers in case of 80x86) a new address on an interrupt for servicing that interrupt.
- The memory addresses for vectoring by the processor are processor or microcontroller specific.
- Vectoring is as per interrupt handling mechanism, which the processor provides.

Interrupt vector– an important part of interrupts service mechanism

- Interrupt service mechanism function as per provisioning in processor.
- Processor first saves program counter and/or other registers of CPU on interrupt
- Loads a vector address INT_VECTAddr into the program counter

Interrupt vector– an important part of interrupts service mechanism

- Either the Vector address INT_VECTAddr corresponds to the ISR address itself from where processor executes ISR
- or INT_VECTAddr corresponds to another ISR address from where processor then executes ISR

Interrupt vector– an important part of interrupts service mechanism

- or from INT_VECTAddr the processor gets a word which specifies the interrupt source or group of sources and processor calculates IR address from where the processor executes ISR
- or INT_VECTAddr gives Interrupt type (or level). and processor calculates IR address from where the processor executes ISR

Interrupt Vector

- System software designer puts the bytes at a ISR_VECTADDR address.

The bytes are for either

- the ISR short code or jump instruction to ISR instruction or
- ISR short code with call to the full code of the ISR at an ISR address or
- Bytes points to an ISR address

Processor Vectoring to an ISR_ VECTADDR

- On an interrupt, a processor vectors to a new address, ISR_VECTADDR.
- Vector means the program counter (PC), which was going to have the program or routine executing at instruction address of next instruction, now saves that address on stack (or in some CPU register, called link register) and processor loads the ISR_VECTADDR into the PC.
- When PC saves on the stack, the stack pointer register of CPU provides the address of memory stack.

Link Register in certain Processors

- A part of the CPU register set
- The PC saves at link register (in place of stack) before the processor vectors to an address by loading new value in PC

Return from ISR

- Because the PC is saved at stack or link register before vectoring, it enables return from the ISR later on an RETI (return from interrupt) instruction

ISR_VECTADDR based addressing mechanism

- A system has the internal devices like the on-chip timer and on-chip A/D converter.
- In a given microcontroller, each internal device interrupt source or source group has separate ISR_VECTADDR address.
- Each external interrupt pins have separate ISR_VECTADDR, example, 8051.

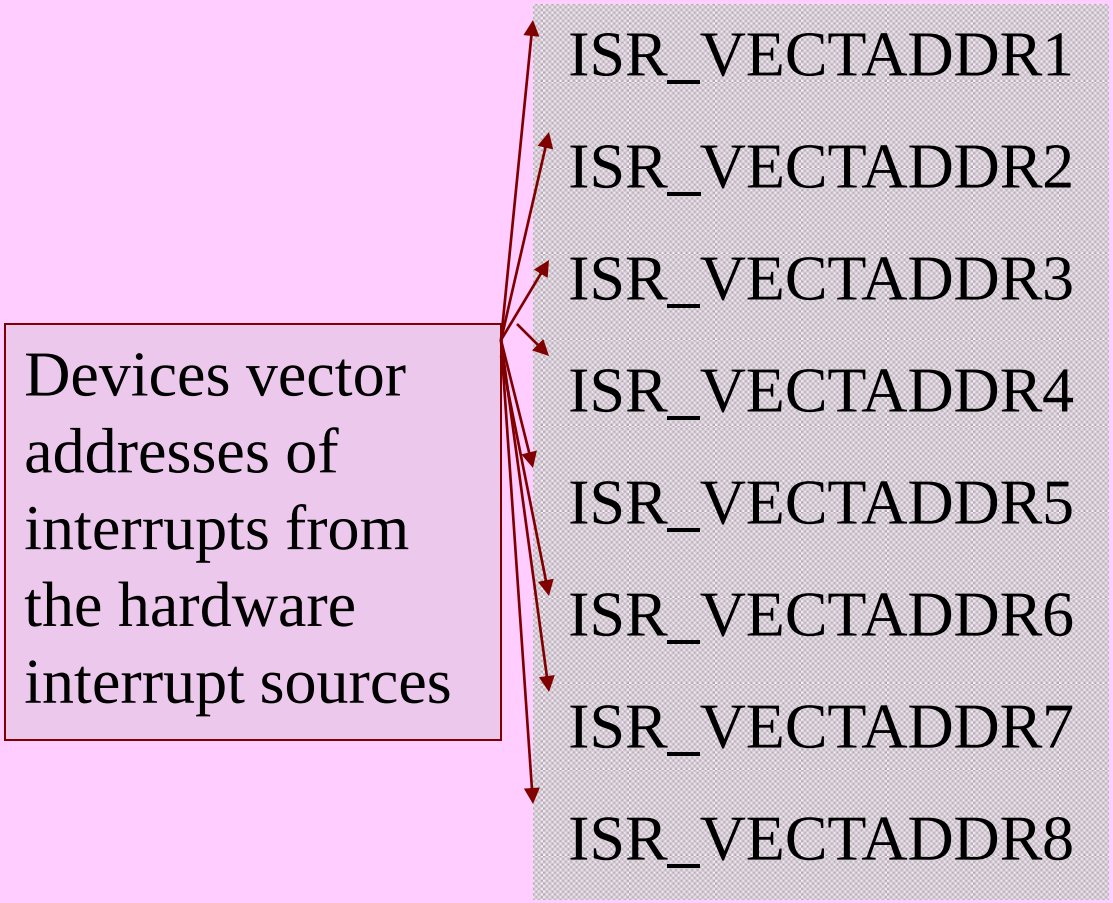
Commonly used method

- The internal device (interrupt source or interrupt source group) in microcontroller auto generates the corresponding interrupt-vector address, ISR_VECTADDR.
- These vector addresses specific for a specific microcontroller or processor with that internal device.
- An internal hardware signal from the device is sent for interrupt source in device interrupts source group

Two types of handling mechanisms in processor hardware

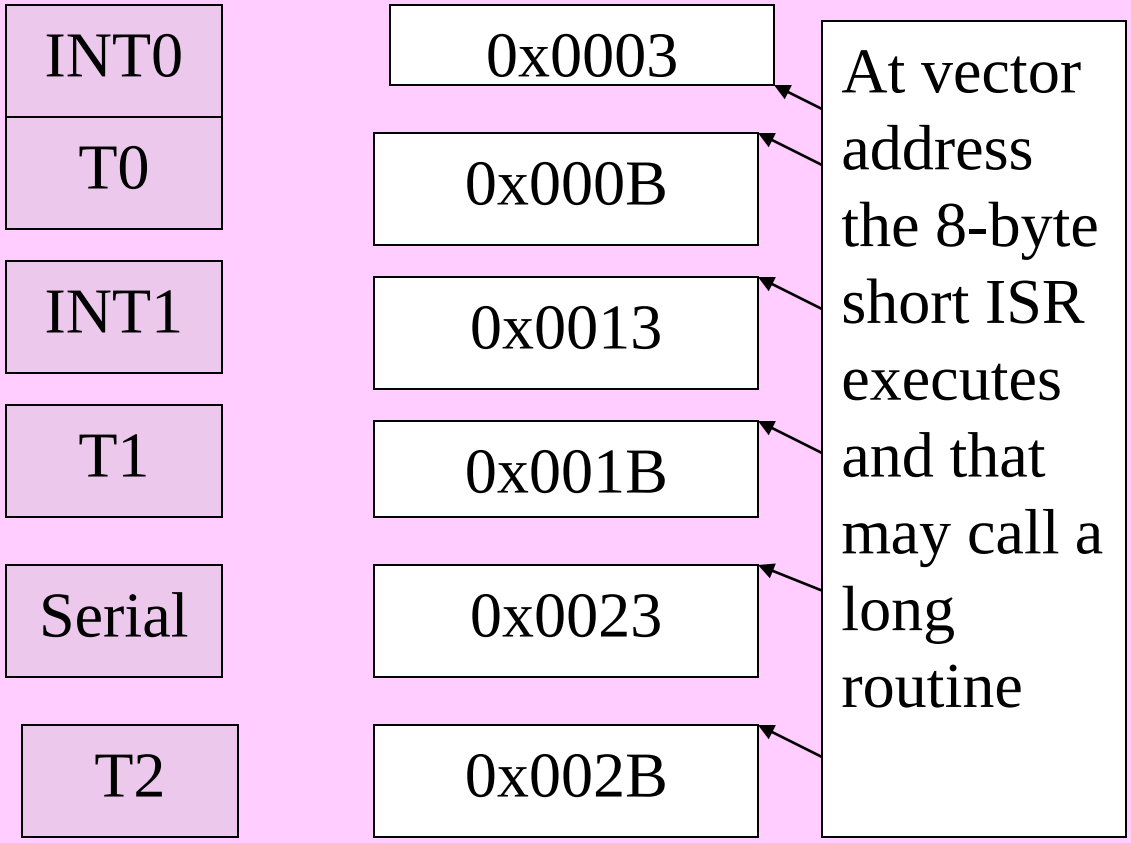
1. There are some processors, which use ISR_VECTADDR directly as ISR address and processor fetches from there the ISR instruction, for example, ARM or 8051
2. There are some processors, which use ISR_VECTADDR indirectly as ISR address and processor fetches the ISR address from the bytes saved at the ISR_VECTADDR, for example, 80x86

ISR_VECTADDRs for hardware interrupt sources or source-groups



From a vector address either the 4 or 8-byte short ISR executes or a Jump instruction executes for the long program to ISR codes starting address

8051 Interrupt Vector Addresses

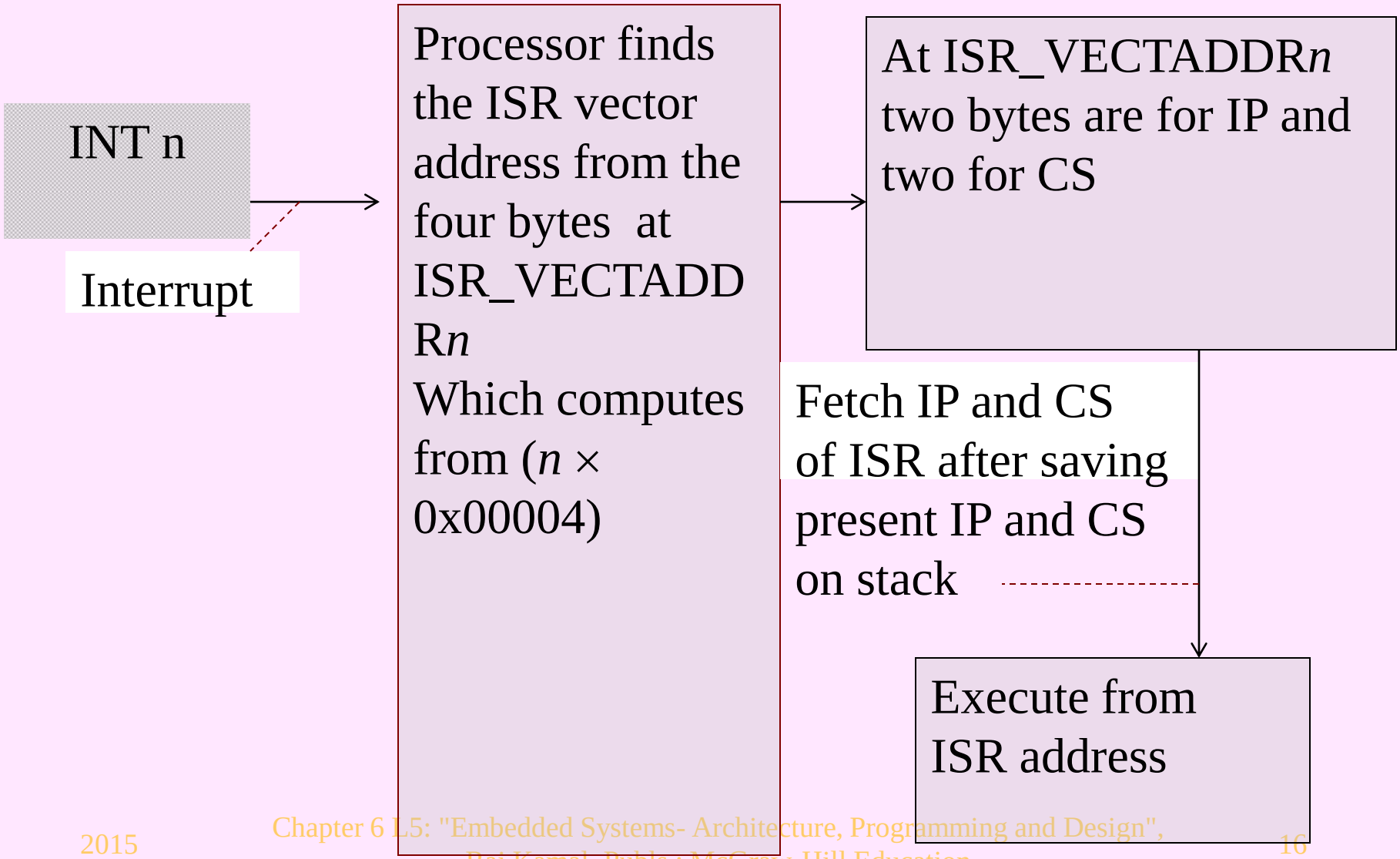


8051 vector addresses of interrupts from the hardware interrupt sources.

80x86 Processor Mechanism

- A software interrupt instruction, for example, *Int n* explicitly also defines *type* of interrupt and the *type* defines the ISR_VECTADDR
- Type value multiplied by 0x00004 gives the vectoring address from where the processor fetches the four bytes to compute the ISR address ~~for~~ executing the ISR

80x86 Steps on INT *n* Innstruction or on interrupt of type *n*



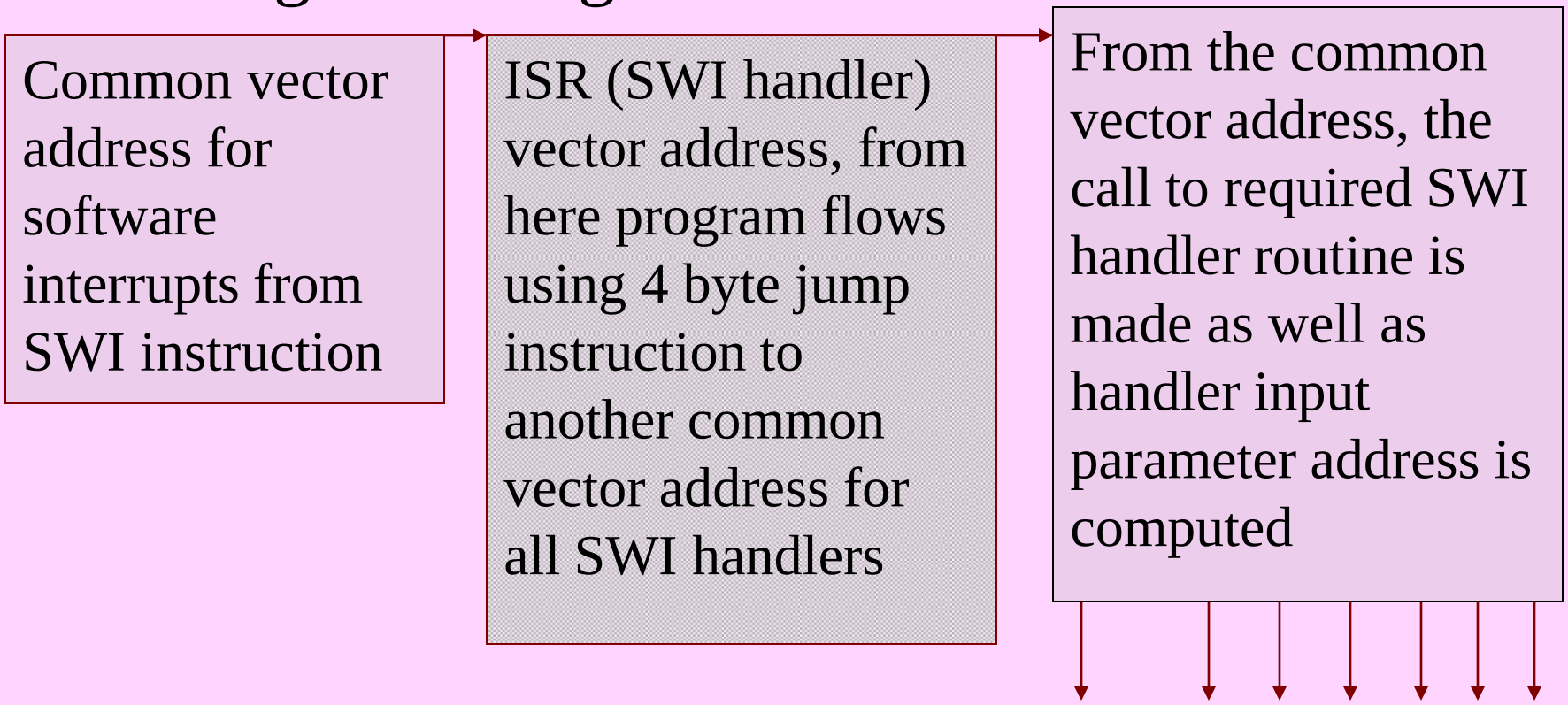
ARM processor Mechanism

1. In a certain processor architecture, for example, ARM, the software instruction SWI does not explicitly defines the type of interrupt for generating different vector address and instead there is a common ISR_VECTADDR for each *exception* or *signal* or *trap* generated using SWI instruction.

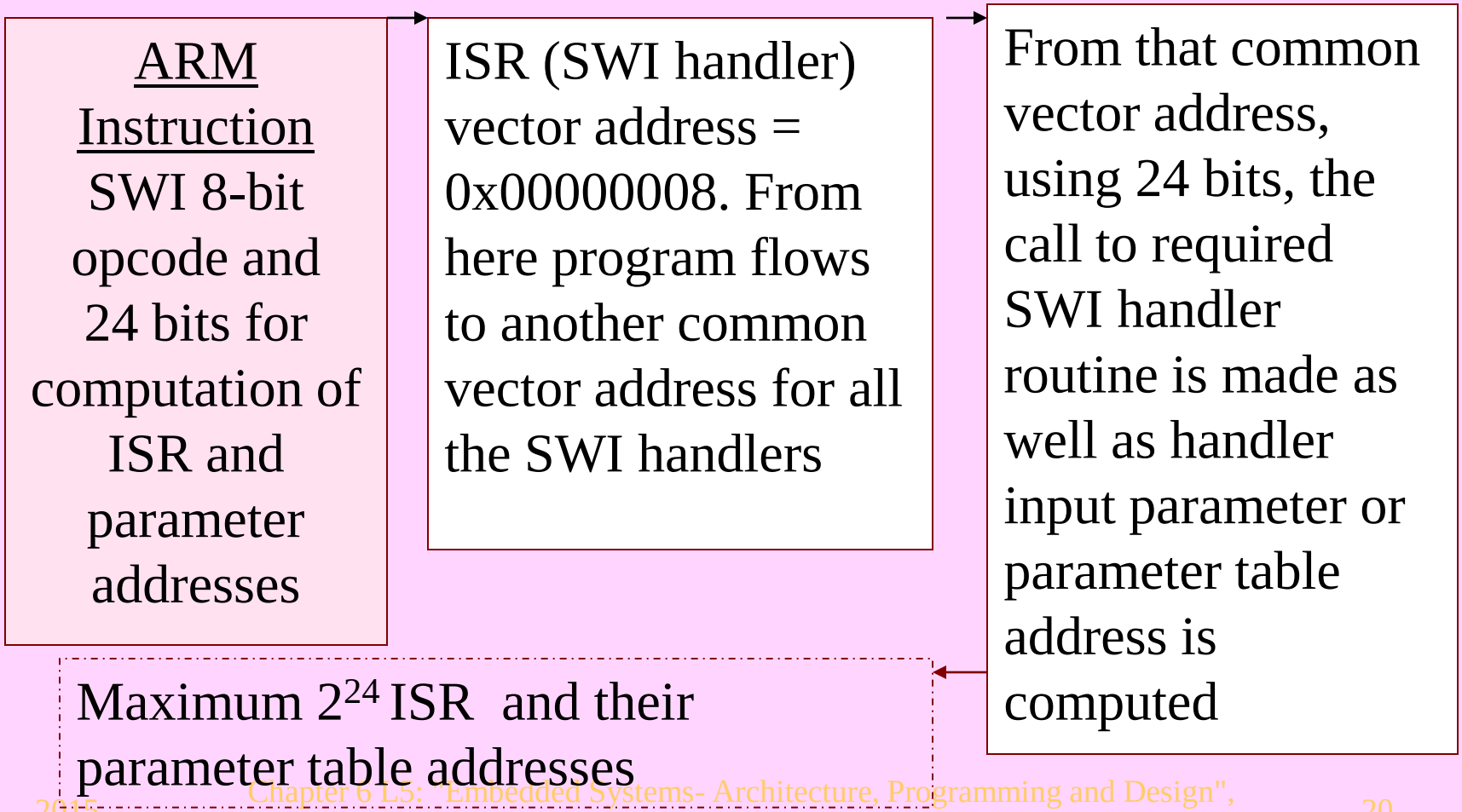
ARM processor Mechanism

2. ISR that executes after vectoring has to find out which exception caused the processor to interrupt and program diversion. Such a mechanism in processor architecture results in provisioning for the unlimited umber of exception handling routines in the system with common an interrupt vector address. ARM processor provisions for such a mechanism

ISR_VECTADDR with common vector addresses for different exceptions, traps and signals using SWI instruction in ARM



ARM SWI *n* instruction [*n* is 24 bits]



Interrupt Vector Table

- Facilitates the service of the multiple interrupting sources for each internal device.
- Each row of table has an ISR_VECTADDR and the bytes to be saved at the ISR_VECTADDR.
- Vector table location in memory depends on the processor.

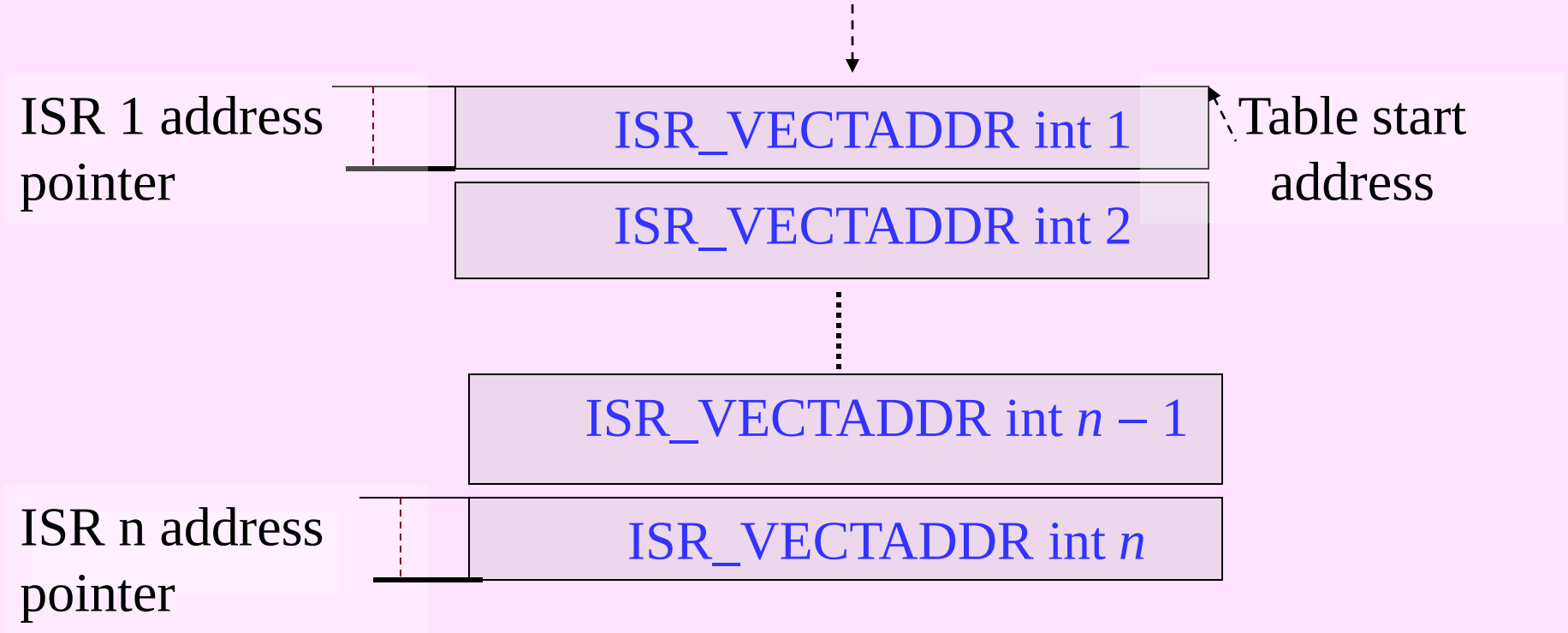
Interrupt Vector Table

- System software designer must provide for putting the bytes at each ISR_VECTADDR address.

The bytes are for either

- the ISR short code or jump instruction to ISR instruction or
- ISR short code with call to the full code of the ISR at an ISR address or
- Bytes points to an ISR address

Hardware and software interrupt *n* sources and *n* entries in vector table



Lookup table for n addresses of handlers for exceptions, traps, and device interrupts

Interrupt Vector Table

- At higher memory addresses, 0xFFC0 to 0xFFFB in 68HC11
- At lowest memory addresses 0x0000 to 0x03FF in 80x86 processors.
- Starts from lowest memory addresses 0x00000000 in ARM7.

Summary

We learnt

- An interrupt vector is an important part of interrupts service mechanism, which associates a processor.
- Processor first saves program counter and/or other registers of CPU on interrupt and then loads a vector address into the program counter.
- Vector address provides the ISR or ISR address to the processor for an interrupt source or a group of sources or for the given interrupt type.

We learnt

- The interrupt vector table is an important part of interrupts service mechanism, which associates the system provisioning for the multiple interrupt sources and source groups
- The table has ISR_VECTOR_ADDRESSes of the multiple interrupt-source groups

End of Lesson 5 of Chapter 6
on
Interrupt Vector mechanism