Chapter 5

Real Time Control: Interrupts

Lesson 3

Multiple Interrupt Sources and Polling

Hardware Interrupts

1. Interrupt of the processing unit

2. External pin interrupts

Hardware Interrupts- 8051/52 MCU

INT0 **INT1** **INT2**

Hardware Interrupts-68HC11

XIRQ

Clock failure or slowing

NMI pin

Hardware Interrupts- ARM-MCU

- Reset
- Data Abort
- FIQ
- IRQ
- Pre fetch Abort

Device Interrupts—ARM MCU

Interrupt of the devices like timers, serial UART, serial devices, watchdog timer

Device Interrupts—8051/52

TI

RI

Timer1 overflow

Timer 0 Overflow

Watchdog timer

In-capture

Outline

- Hardware Interrupts
- Device Interrupts
- **Software Interrupts**
- Polling for Interrupt service in case of Multiple sources

Software Interrupts

Interrupt due an instruction like SWI (Software interrupt) or due to illegal opcode

Software Interrupts- 8051/52 MCU

None

Software Interrupts-68HC11

SWI

Illegal opcode

Software Interrupts- ARM- MCU

SWI

Illegal opcode

Polling for Interrupt service in case of Multiple sources (Finding the interrupts pending service)

Interrupt Service Start

MCU

Event(s) Identify, check enable bit(s)

↓

Poll for the service pending and select the event to be serviced first

↓

Save context and generate ISR_Vect_Addr as per the event(s)

↓

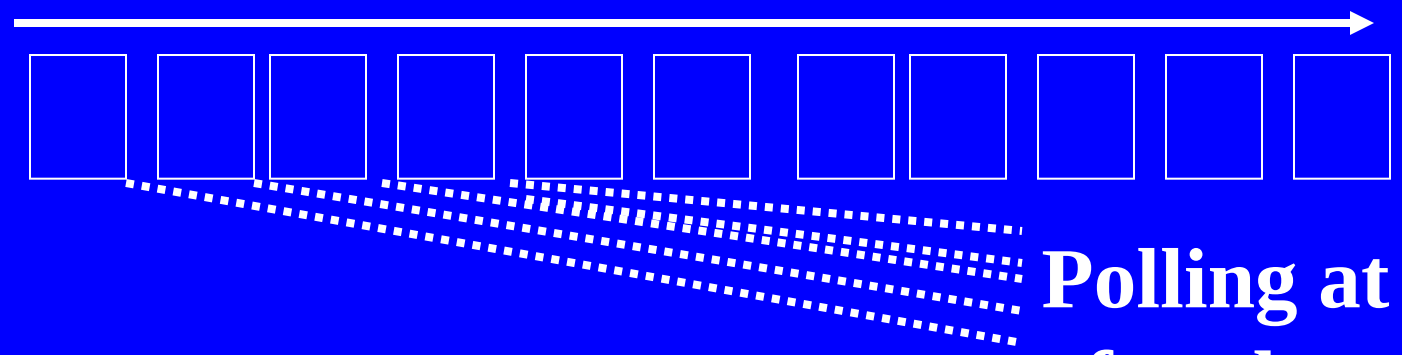
Get ISR_address

↓

PC = ISR_ address, Start Execute Interrupt Service routine

Poll in the foreground program at the end of each instruction

Polling within Main program under execution



ISR Instructions

Polling at end of each instruction if any of interrupts not-masked

Poll at the end of each ISR instruction

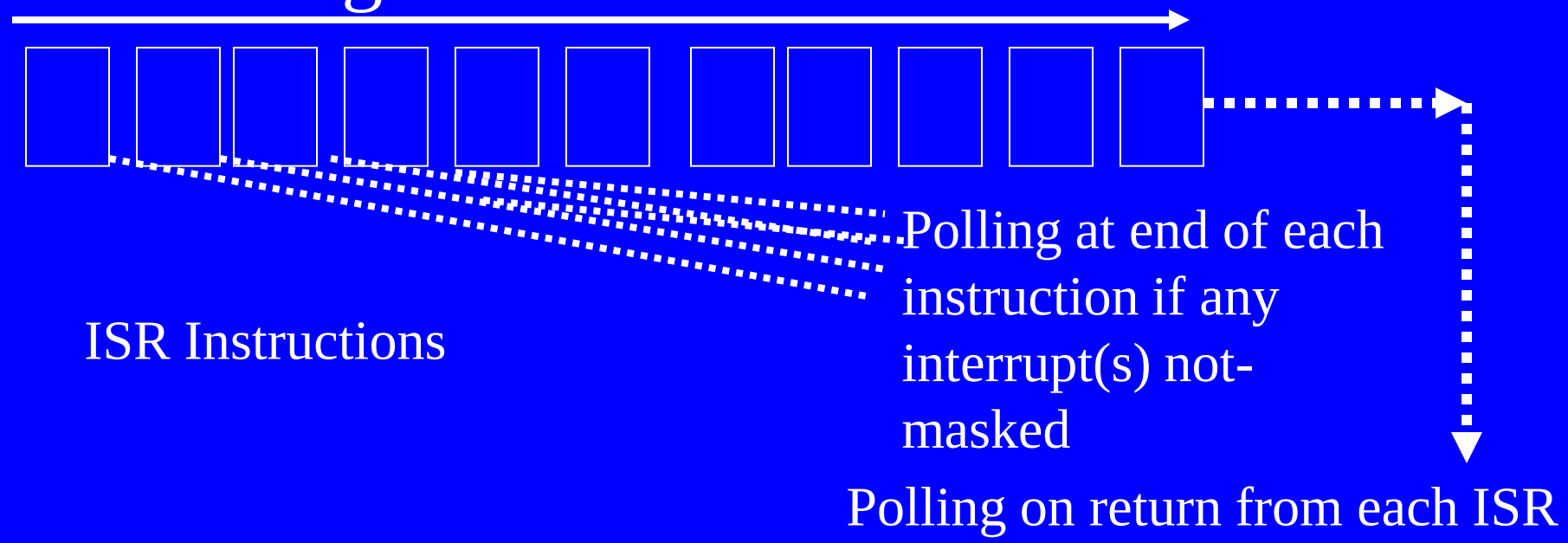
Option 1

Advantage:

Priorities when defined with the rule that earliest deadline first, lets each ISR finish within the deadline

[Deadline means maximum time interval permitted between a need for finishing an interrupt service and actual finishing of service routine.]

Polling within an ISR under execution

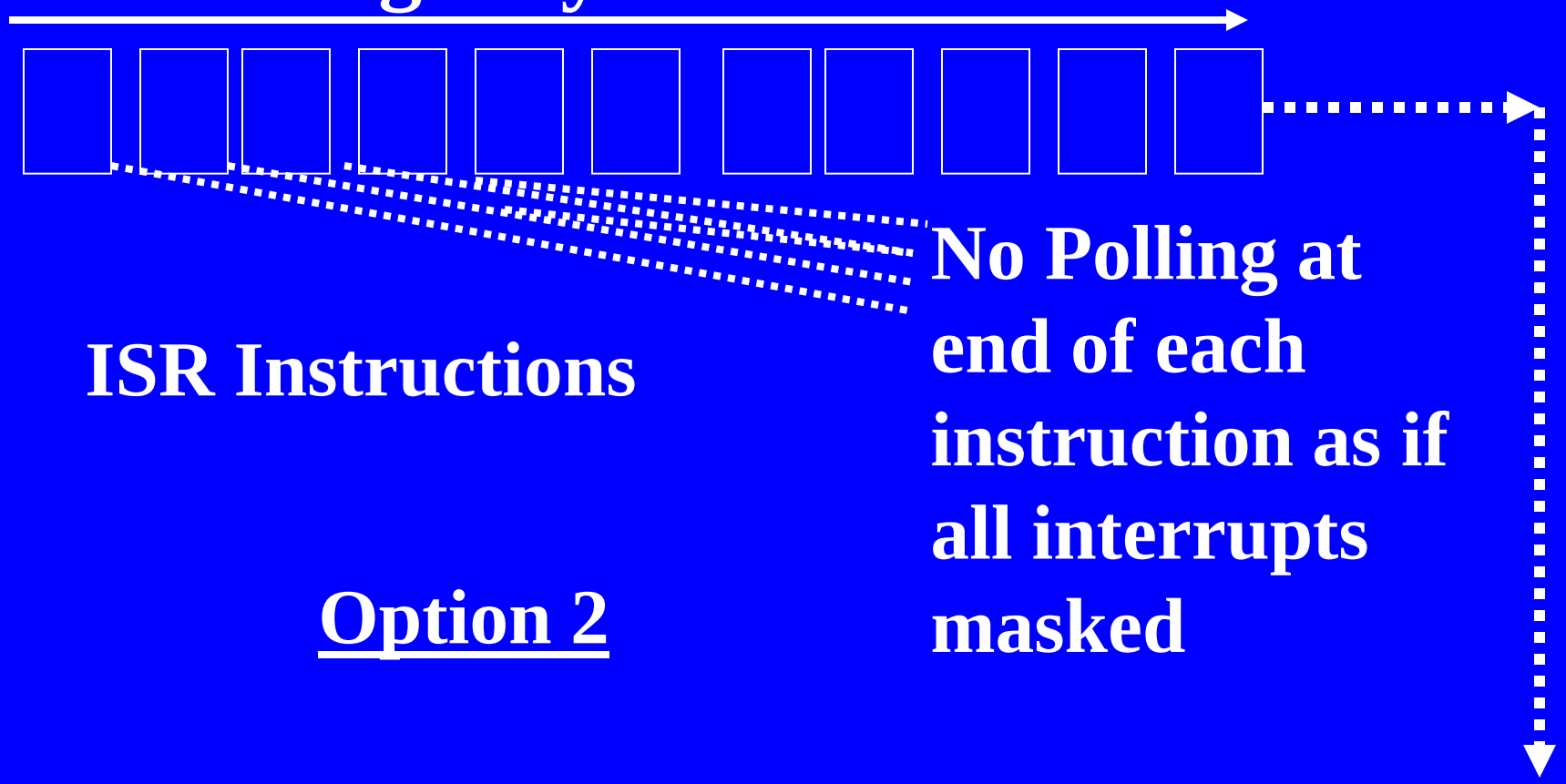


Option 1

Poll only at the end of each ISR on return

Advantage:
Any highest priority ISR will delay at best by the time left for finishing the current routine. But, each ISR once starts finishes till end.

Polling only at return from the ISR



Polling for start, finding which Interrupt to Service

8051 Option 1 - **Polling at end of each instruction if any of interrupts not-masked**

Polling order among seven possible interrupt service need as per priority when software priorities of all set equal

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

Polling for start, finding which Interrupt to Service

68HC11 Option 2 - **Polling only at return from an ISR**

Summary

We learnt

- Software interrupt
- Hardware interrupt
- Device Interrupt
- Reset of flag after the ISR start
- Polling to select among multiple sources