

Interfacing and Interrupt Sequence for a SINGLE 8259

INTE flip-flop of the μP is **set** by the **EI** instruction and the **individual interrupts enabled** using **SIM** instruction.

8259 is **initialized** by giving **ICW1** and **ICW2** (compulsory) and **ICW4** (optional). Note that **ICW3** is **not given** as Single 8259 is used. OCWs are given if required.

Once 8259 is initialized, the **following sequence** of events takes place when one or more **interrupts occur** on the IR lines of the 8259.

- 1) The **corresponding bit** for an interrupt is **set in IRR**.
- 2) The **Priority Resolver checks** the 3 registers:
IRR (for highest interrupt request)
IMR (for the masking Status)
InSR (for the current level serviced)
and **determines** the **highest priority** interrupt.
It **sends** the **INT** signal to the μP .
- 3) The μP **finishes** the **current instruction** and **acknowledges** the interrupt by **sending** the **first INTA pulse**.
- 4) On receiving the INTA signal, the **corresponding bit** in the **InSR** is **set** (indicating that the service of this interrupt is started) and the **bit** in the **IRR** is **reset** (to indicate that the request is accepted).
8259 now **sends** the **Opcode of CALL** instruction to the μP on the data bus.
- 5) The μP **decodes** the **CALL instruction** and **sends 2 more INTA pulses** to the 8259.
- 6) In response to the two INTA pulses, the **8259 sends the address of the ISR to the μP** .
First the lower byte and then the higher byte.
- 7) Thus, the complete 3-byte CALL Instruction code is released by the 8259.
In the AEOI Mode the **InSR bit is reset** at this point, **otherwise it remains set until an appropriate EOI** command is given at the End of the ISR.
- 8) The μP **pushes** the contents of **PC** onto the **Stack** and **transfers program to the address** of the **ISR** sent by the 8259. **The ISR thus begins**. *#Please refer Bharat Sir's Lecture Notes for this ...*

