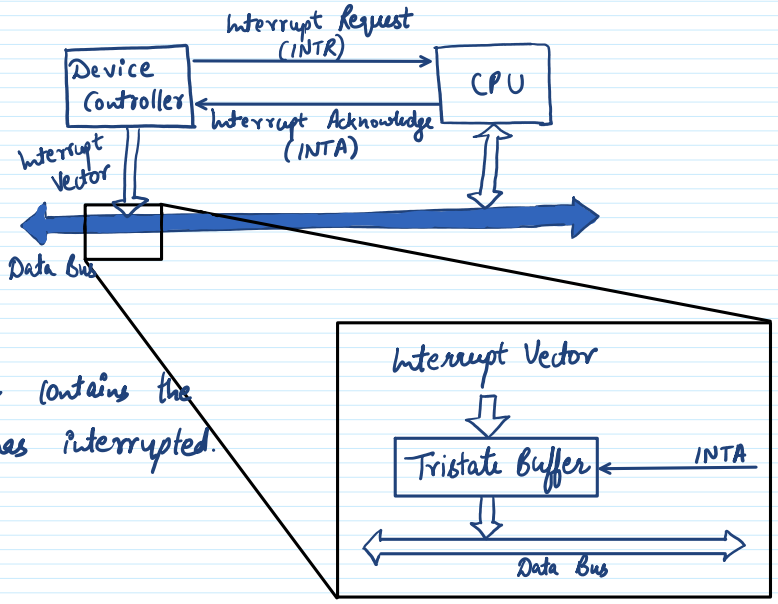
Lecture-23 : Interrupts Handling

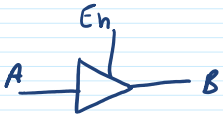
11 March 2024 14:30

Interrupt Processing :-

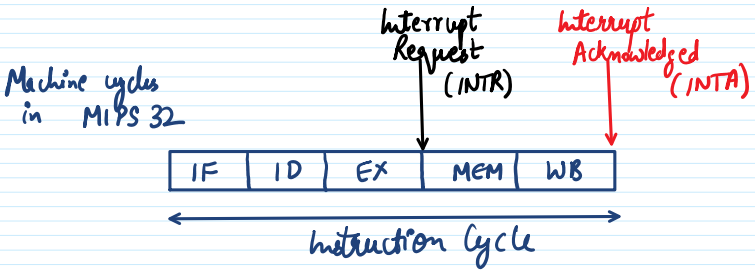


Interrupt Vector contains the device which has interrupted.

Tristate buffer is enabled when INTA is active.



B=A if En=1
=2 if En=0



Steps :-

1. Device controller sends INTR to the CPU.
2. CPU finishes the current instruction and sends back INTA.
3. Device controller sends interrupt vector (or number) over data bus.
4. CPU reads the interrupt vector, and identifies the device.

The occurence of an event triggers a number of events both in processor hadware and software.

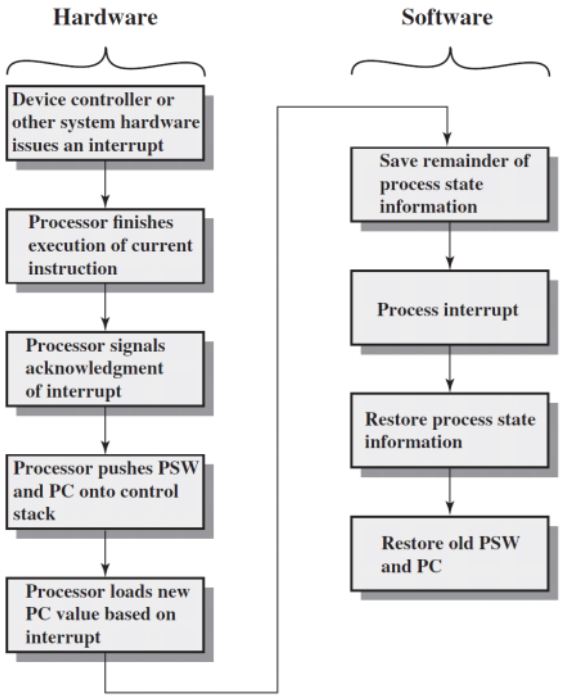


Figure 7.6 Simple Interrupt Processing

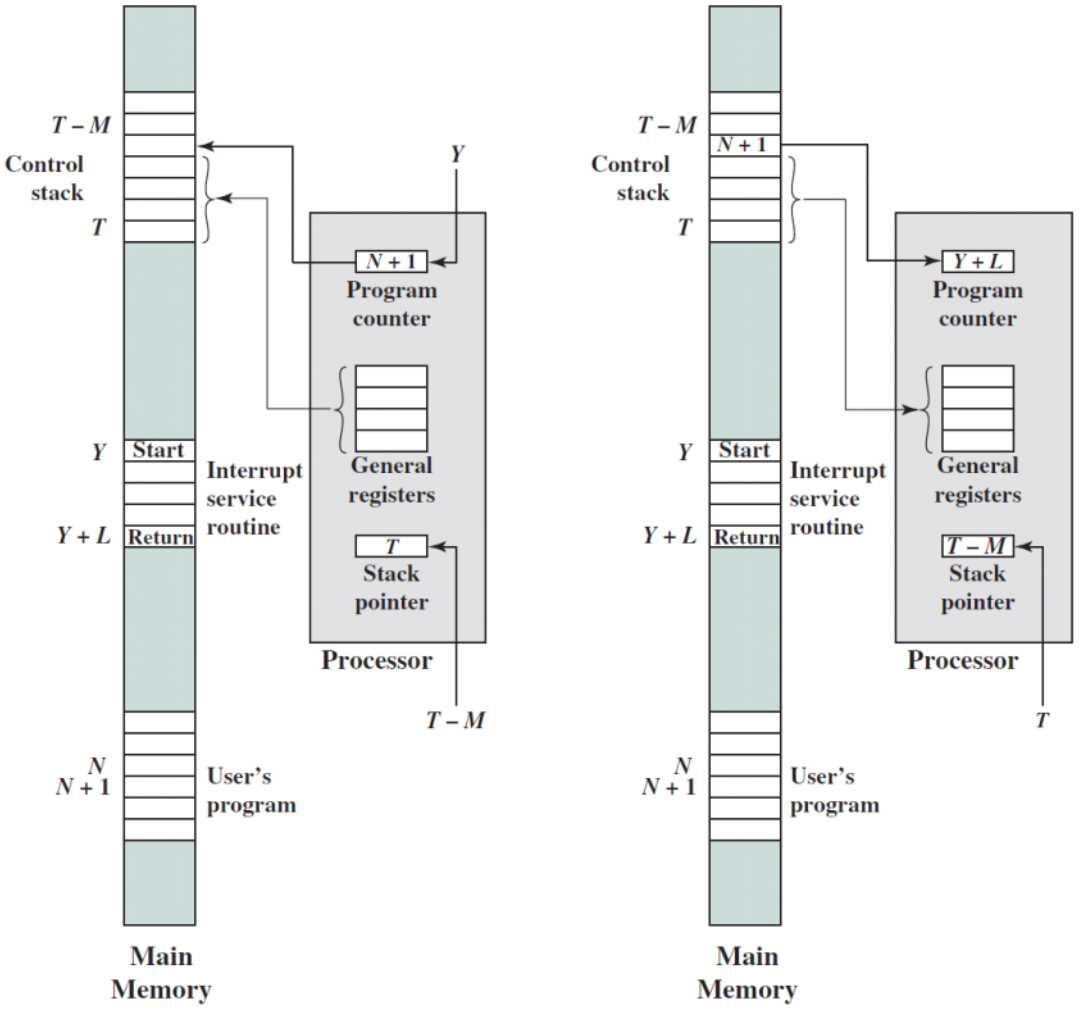
PSW (Program Status Word)

PC (Program Counter)

PSW :- Register containing the current status of the processor.

PC :- Location of the next instruction to be executed.

Changes in memory and register :-



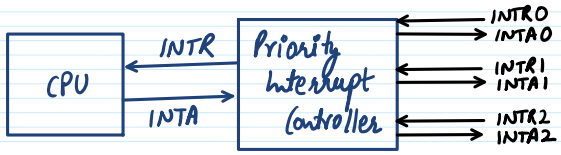
(a) Interrupt occurs after instruction at location N

(b) Return from interrupt

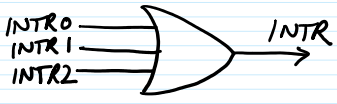
Figure 7.7 Changes in Memory and Registers for an Interrupt

Multiple devices interrupting the CPU :-

→ using a Priority Interrupt Controller.



INTR = INTR0 + INTR1 + INTR2



Simultaneous Requests :-

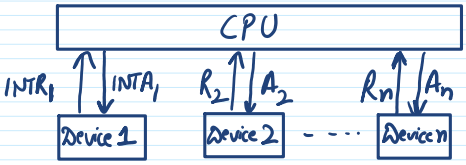
→ arrival of interrupt requests for two or more devices.

Priority Scheme	Daisy Chaining
→ CPU has multiple request	⇒ Priority is automatically

Priority Scheme

→ CPU has multiple request lines

→ accepts the request with highest priority.



$INTR_i$ has highest priority than $INTR_j$ where $i<j$.

Daisy Chaining

⇒ Priority is automatically assigned based on the order of checking devices.

⇒ So the priority is set based on the order in which is electrically connected.

⇒ INTA is passed to the next device only if the current device has not interrupted.