Quadrant II – Transcript and Related Materials

Programme	**: Bachelor of Science (Third Year)**
Subject	**: Electronics**
Paper Code	**: ELC 126**
Paper Title	**: Embedded Systems**
Unit	**: II**
Module Name	**: Resets Interrupts and its uses,Interrupts v/s Polling**
Module No	**: 6**
Name of the Presenter	**: Mrs. Darshata Omkar Satarkar**

Notes:

- The code to handle an interrupt is called an interrupt handler or interrupt service routine (ISR)

Resets can be generated:

- When power is first applied.
- If the device detects a serious fault in hardware or software from which the user program cannot be expected to recover.

The important characteristics of the MSP430 are that, it has two levels of reset, depending on whether the reset is caused by Hardware or Software. Hardware reset is identified as:

- Power on Reset (POR) :

This is generated when the device is powered up. More generally, a POR is raised if the supply voltage drops to so low value that the device may not work correctly: a brownout.

- When reset line (RST/NMI) is pulled low:

This allows external signals on RST/NMI pin to reset the device if the pin is configured for the reset function rather than the non maskable interrupt. The Reset function RST is active by default.

And Software reset is identified as:

- Power up clear (PUC)

This always follows a power on reset. It is generated when software appears to be out of control in the following ways-

- The watchdog timer overflows in watchdog mode.
- An attempt is made to write to the watchdog control register WDTCTL, without the correct password 0x5A in the upper byte. A reset is triggered even if the watchdog is disabled.
- The registers for the flash memory controller FCTLn are protected by a password in the same way as WDTCTL. The value is 0xA5. This is to protect runaway software from corrupting the stored program.

Interrupt Priority

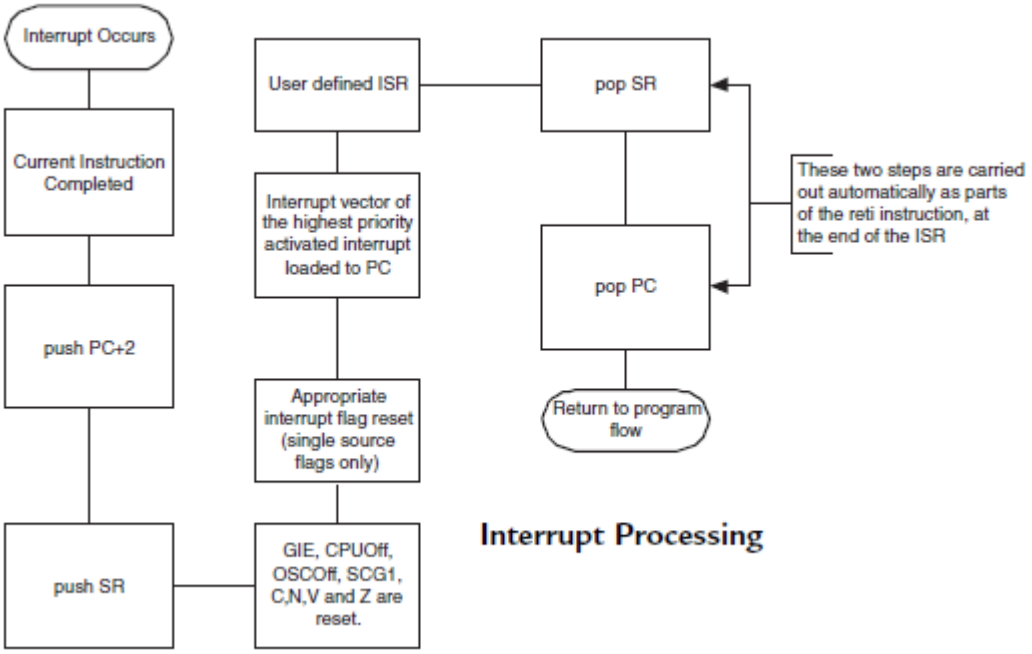
- Each interrupt vector has a distinct priority, which is used to select which vector is taken if more than one interrupt is active when the vector is fetched.
- The priorities are fixed in hardware and cannot be changed by the user.
- They are given simply by the address of the vector: A higher address means a higher priority.
- For example, the reset vector has address 0xFFFE, which gives it the top priority, followed by 0xFFFC for the single non-maskable interrupt vector.

Interrupts v/s Polling

- The main difference between interrupt and polling is that, in case of an interrupt, the device notifies the CPU that it requires attention while in the case of polling, the CPU continuously checks the status of the device to find whether it requires attention.
- When an interrupt occurs, the interrupt handler executes. In polling, the CPU provides the service.
- Interrupts can occur at any time. Polling occurs at regular intervals.
- Interrupt request line indicates that the device needs a service. Command ready bit indicates the device needs a service.

- Interrupts does not waste much CPU cycles whereas polling wastes a lot of CPU cycles.
- In case of interrupt, it is inefficient when the devices interrupt the CPU frequently. In contrast, polling is inefficient, when the CPU does not get much requests from the devices.

Interrupt Processing



Interrupt Processing

- When an interrupt occurs, the program counter of the next instruction and the status register are pushed to the stack.
- The SR is then cleared, along with the appropriate interrupt flags if the interrupt is single source.
- One of the important effects of the SR clearing is the disabling of interrupts, via the reset of the GIE flag.
- Commonly described as non-reentrant (or non-preemptive) interrupts, the effect of this is that interrupts service routines will not be called from other interrupt service routines unless the GIE bit has been toggled manually.
- Multiple (peripheral) source flags must be reset manually by the programmer. This functionality is charted in Figure.
- Return to program flow is accomplished by the reti instruction. Reti automatically pops the status register and program counter.
- If your code jumps to a common point, rather than using the reti instruction, you need to account for these extra items on the stack.