INTERRUPTS

There are 2 methods for communicating between the microcontroller and the external system:

- POLLING
- INTERRUPTS

POLLING

- In this method, microcontroller accesses at the exact time interval the external device, and gets the required information.

- The time periods is determined by user. In fact, you can say that, when using the method - polling, the processor must access the device itself and request the desired information that is needed to be processed.

- In fact we see that in this method, there is no independence for the external systems themselves. They depend on the microcontroller. The processor may only access the external device and get from it the information needed.

- The main drawback writing program that uses this method is a waste of time. The micro needs to wait and review whether a new information arrived.

INTERRUPTS

- Interrupt is the signal sent to the micro to mark the event that requires immediate attention.
- Interrupt is “asking" the processor to stop to perform the current program and to “make time” to execute a special code.
- In fact, the method of interrupt defines the option to transfer the information generated by internal or external systems inside the micro by them self! Once the system has finished the task imposed on it, the processor will be notified that it can access and receive the information and use it.

Interrupt Sources

- External hardware devices are sending interrupts to microcontroller in order to receive “the treatment”.
- The micro can send to itself an interrupt as a result of executing the code to report the failure in the process.
- In the multi-processor system, processors can send interrupts to each other as communication between them, for example for the division of work between them.
- There are two types of interrupts: *software interrupts* and *hardware interrupts*.

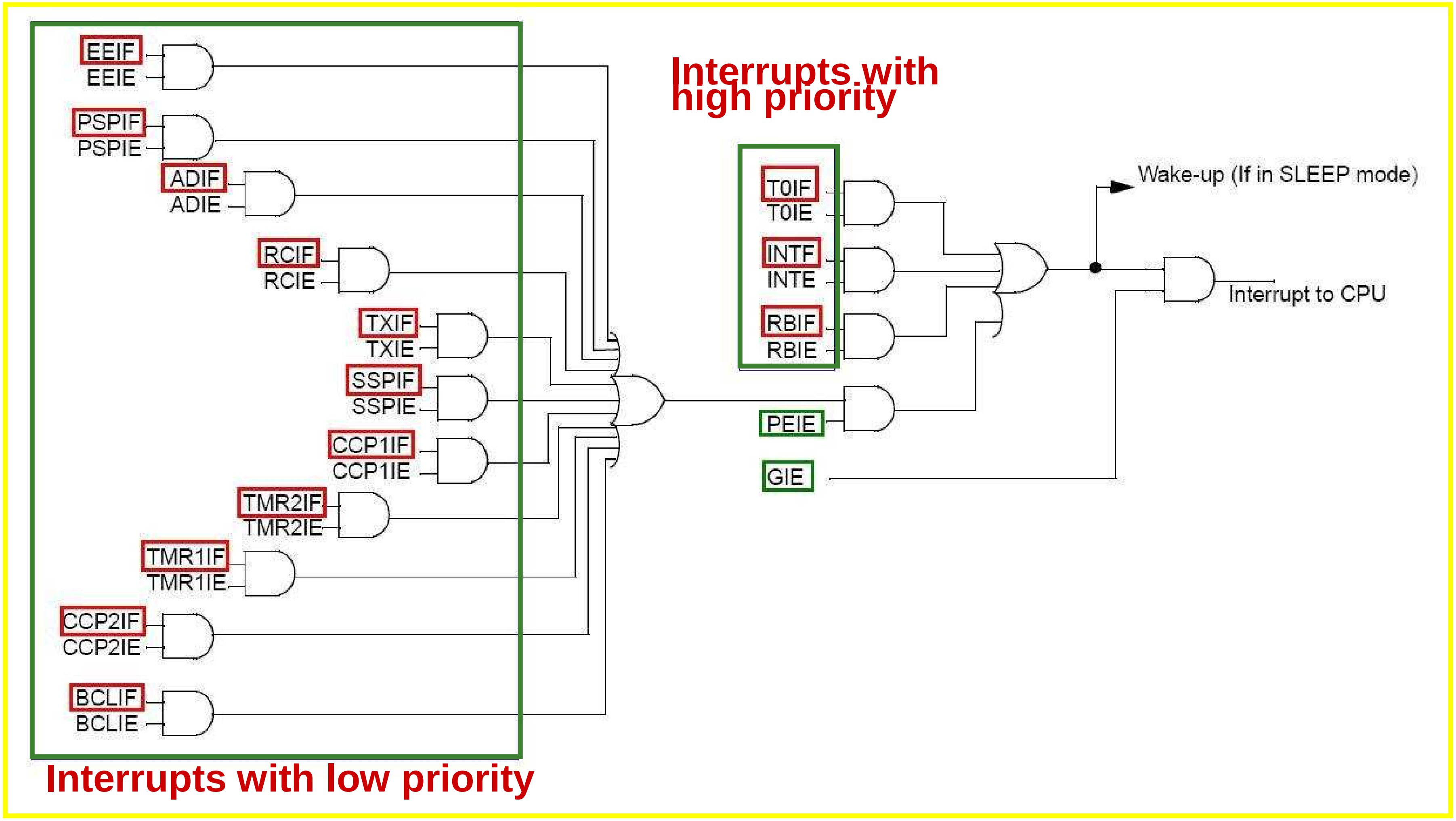
Software Interrupts

- Software interrupts come from a program that runs by the processor and “request” the processor to stop running the program, go to make a interrupt and then to return to continue to execute the program.
- Example: Procedure - when there is a procedure call, the processor stops the execution of the program, jumps to the place in memory that reserved for a procedure – executes the procedure and only then returns back to the program and continues to execute.

Hardware Interrupts

- The hardware interrupts are sent to the microcontroller by external hardware devices.
- Some of the interrupts can be “blocked” = (masking) by Interrupt Enable bit (IE). When the interrupt is blocked the micro “does not see“ the request for an interrupt, therefore won’t be available to execute it.
- The “blocked” interrupt won’t be executed till the “block” is removed.
- There are interrupts that can not be “blocked”. These are used to report on critical hardware issues, such as the drop of voltage. We want an immediate response from the microcontroller to these kind interrupts, without the ability to ignore them.

PIC16F877 Interrupts



PIC16F877 Interrupts cont

- The microcontroller has 14 interrupt sources
- XXIF is an interrupt flag that shows the result that we are getting from an interrupt.
- XXIE is an interrupt enable bit that is used to enable or “block” the interrupt.
- The interrupts on the left side of the figure (previous slide) are low priority and all of them together can be “blocked” by enabling bit interrupt PEIE = 0.
- We can determine whether or not to allow the system to address the interrupts. This is done by using Global Interrupt Enable bit GIE.

PIC16F877 Interrupts cont

- EEIF - Write Complete Flag Bit. This flag bit appears in the memory components such Data EEPROM and Flash Program Memory located inside the PIC. In order to begin writing again into memory the flag should be reset first - EEIF=0. EEIF must be cleared by software.
- PSPIF – This interrupt flag appears when we are utilizing PORTD. PORTD operates as an 8-bit wide Parallel Slave Port (PSP), or microprocessor port, when control bit PSPMODE (TRISE<4>) is set. Interrupt flag PSPIF is designed to inform that the operation of reading/writing from/to PORTD is ended.

To be continued