1. **Polling:**

Polling is synchronous. Polling, also known as busy waiting, is when an I/O device is repeatedly checked if it is ready. While polling, the processor waits for the device and does nothing else.

**Interrupts:**

Interrupts are asynchronous. An interrupt can be received at any time. In the ATMega 128, there are 35 interrupt vectors. When an interrupt is received, the processor will handle it accordingly. Using interrupts can be more effective because we can decide when to deal with an interrupt and what interrupt should be dealt with first. Interrupts cause the processor to save the current state of everything before they deal with the interrupt. This potentially uses up a large amount of memory.

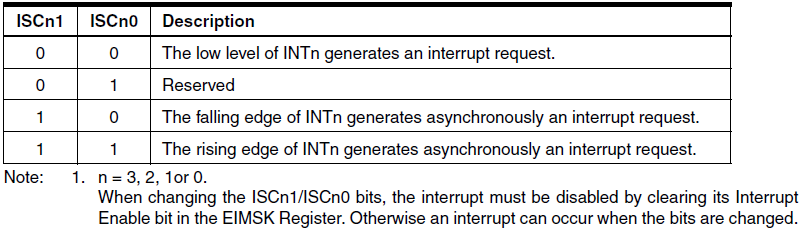
**Polling vs Interrupts:**

If you want your program to be more dynamic and have the ability to be notified by multiple devices at a time, use interrupts. You can control when you want to deal with an interrupt, which interrupt has the highest priority, etc. If your program will get to a point where it can’t continue until a certain device is ready, you can just use polling.

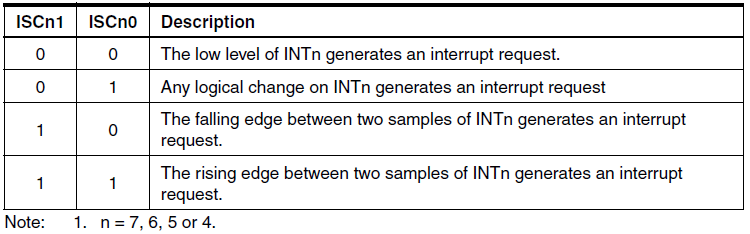
2. What is the function for each bit in the following registers in the ATMega128? (EICRA, EICRB, and EIMSK)

EICRA and EICRB define whether the interrupt is set on the falling edge, rising edge, or otherwise.

**EICRA:**



**EICRB:**



**EIMSK: (External Interrupt Mask)**

An interrupt may be requested by either the EICRA or EICRB, but the corresponding register in the EIMSK (and I-Bit in the status register) must also be set for the interrupt to make it all the way to the external interrupt flag register (EIFR). It is important to note that any activity on the EIMSK pins will trigger an interrupt request even if the pins are enabled for output.

|  |  |  |
| --- | --- | --- |
| EICRA | EICRB | EIMSK |
| ISC00: | ISC40: | INT0: |
| ISC01: | ISC41: | INT1: |
| ISC10: | ISC50: | INT2: |
| ISC11: | ISC51: | INT3: |
| ISC20: | ISC60: | INT4: |
| ISC21: | ISC61: | INT5: |
| ISC30: | ISC70: | INT6: |
| ISC31: | ISC71: | INT7: |

3. What is an interrupt vector? List the memory locations for the following vectors in the AVR microcontroller: Timer/Counter2 Comparison Match, External Interrupt 2, and USART1-Rx Complete.

An interrupt vector is the memory address to the source of an interrupt. Alternatively, an interrupt vector can be an index to a table (known as an interrupt vector table). The memory address of the interrupt source is contained in the interrupt vector table.

|  |  |  |  |
| --- | --- | --- | --- |
| Interrupt Definition | **USART1, Rx Complete** | **Timer/Counter2** | **External Interrupt 2** |
| Vector Number | 31 | 10 | 4 |
| Program Address | $003C | $0012 | $0006 |

4. In the AVR microcontroller, like many others, there are several different ways of triggering interrupts. Below is a sample signal being input onto one of the external interrupt pins.

List where the interrupt would trigger on this waveform if the interrupt was set up as:

a.) rising edge

b.) Falling Edge

c.) Level High

d.) Level Low

