**EENG 260 HW3:**

**Problem 1:** Overflow

Consider the addition of two signed 32-bit numbers.

1. If a positive number is added to a negative number under what conditions will a signed overflow occur?
   1. If the resulting number is within the 32-bit integer range but larger than the 31-bit or signed integer range, a signed overflow occurs because the sign bit is being messed with when it shouldn’t be
2. What is the name of the condition code register?
   1. CCR
3. What bit of this register will indicate that a signed overflow has occurred?
   1. Bit 1 or the V bit will be set for overflow

**Problem 2:** Port A Configuration

Please write software that initializes PortA so that pins 7, 5, 3, and 1 are output and the rest are input.

SysCtlPeripheralENable(SYSCTL\_PERIPH\_GPIOA);

GPIOPinTypeGPIOOutput(GPIO\_PORTA\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_3 | GPIO\_PIN\_5 | GPIO\_PIN\_7);

GPIOPinTypeGPIOInput(GPIO\_PORTA\_BASE, GPIO\_PIN\_0 | GPIO\_PIN\_2 | GPIO\_PIN\_4 | GPIO\_PIN\_6);

**Problem 3:** Interrupts

1. Please provide the definition of an interrupt.
   1. An interrupt is a hardware signal that tells the device to start executing a different piece of code designed to respond to said interrupt
2. Provide a description of Polled I/O
   1. Polled I/O is a periodic check for I/O requests made by the CPU. The CPU will continue executing code as normal unless an I/O request is made.
3. Provide a description of interrupt driven I/O
   1. Interrupt driven I/O means the CPU only stops to handle I/O requests when it is given an interrupt signal, and is then directed to the piece of code that handles the interrupt.

**Problem 4:** Logic Circuit for I/O interfacing

Design a logic circuit that is needed to drive a bidirectional I/O pin to either transfer a bit to the CPU or transfer a bit from the CPU to the I/O pin.

Basically this would be the logic circuit used to drive a data direction register to configure an I/O port to serve as an input or as an output. Employ your knowledge of ENGR 160/250 will come in handy.

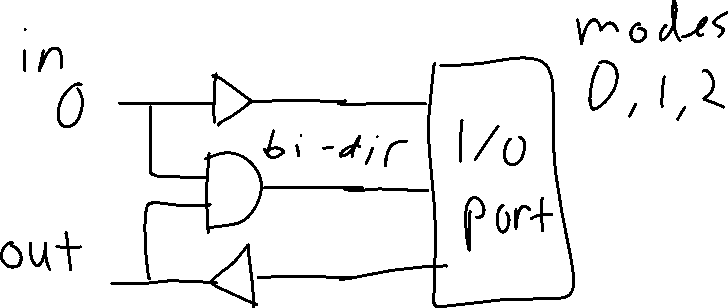
CPU



DDR for I/O PORT

Bi-Directional Interconnect

Design for just one bit



**Problem 5:** Interrupts [25 Points]

1. Provide steps needed to enable interrupts, both external and internal
   1. Enable and configure timers for the IRQs
   2. Enable and assign ports to be used as interrupts
   3. Set the priorities of the interrupts
   4. Write the code to handle the interrupts (ISRs)
2. Provide instructions you would use to disable interrupts, both internal and external.
   1. With NVIC hardware, interrupts and timers can be disabled by setting their control bits to 0.
   2. For example, NVIC\_ST\_CTRL\_R = 0 disables the SysTick control timer which means it will not start counting the clock until enabled again/assigned a clock
3. For TM4C123GH6PM, if interrupts are disabled does it mean that the CPU will not recognize them at all, or would they be queued as pending interrupts?
   1. “disabled” interrupts will cause the interrupts to be queued as pending interrupts that will go through when they are enabled again.
4. Do software interrupts have mask bits i.e. do they need to be disabled and enabled by setting registers of the NVIC appropriately?
   1. Yes
5. Nesting interrupt is normally not advisable because of the potential indeterminate nature of nesting interrupts. The TM4C123GH6PM is however designed to accommodate interrupt nesting. Please identify the design factors that mitigate the potential indeterminism caused by nesting,
   1. The tiva C series uses preemptive interrupts to allow proper interruption and resuming of interrupts that may need to be nested to prevent any issue with the timing of the interrupts (using NVIC hardware)
6. If you are using SysTick interrupts, how would you set the priority to level 6? (Problem 9.1 pp. 374 of textbook)
   1. In C you would use NVIC hardware using the NVIC\_SetPriority() function
   2. NVIC\_SetPriority(SysTick\_IRQn, 6)
7. If you are using Timer0A interrupts how would you set priority to level 5? (Problem 9.2 pp. 374 textbook)
   1. For Timer 0 interrupts, the INTCON register needs to be modified, specifically the TMR0IP bit. However, this only allows priority to be set at all and not levels of priority.