1 Interrupts

An interrupt is a mechanism by which other modules may interrupt the normal sequencing of the processor Classes

- program
- timer
- I/O
- hardware failure

An interrupt improves processor utilization by allowing the processor to jump between threads when one of them is moving slowly

2 Handling Multiple Interrupts

Two approaches to handle multiple interrupts

- disable interrupts within an interrupt (sequential)
  - ignore interrupt request signal
  - interrupt remains pending until checked by processor
- define priorities for interrupts (nested)
  - allow interrupt handled to be interrupted by higher one
  - always execute higher priority it order

3 Chapter 1 Problems

3.1 Problem 1.2

Based on a hypothetical machine from slide 22 of lecture 2. Expand the description of the program execution in figure 1.4 to show the use of MAR and MBR. Fetch: MAR = 300 (PC) and MBR = 1940 (IR) Execute: MAR = 940, MBR = 003

Fetch: MAR = 301, MBR = 5941 Execute: MAR = 941, MBR = 002 the opcode is 5(add) so the MBR + AC =¿ AC == 002 + 003 = 005

Fetch: MAR = 302 =, MBR = 2941 Execute: MAR = 941 (where we want to store it), MBR = 005 (value we want to store)

Problem 1.3

In a hypothetical 32-bit processor having 32-bit instruction composed of two fields: First byte contains opcode, remainder for immediate operand or an operand address

- what's maximum directly accessible memory capacity
  - the data address is 3 bytes so 24 bits so 16 MB
- discuss impact on the system speed if microprocessor bus as
  - 32-bit local address bus and 16-bit local data buss
    - we will have two cycles to access the data since you can only move 16-buts to the local data bus at a time
  - 16-bit local address bus and 16-bit local data bus
    - this will also take 2 cycles since you still need to have a 32-bit address (since its a 32-bit system), this is just done in two separate loads (think matrix)
- how many bits needed for PC and IR, whats an opcode register
  - usually the PC will be 32-bit, but it must be at least 24 bits (1 byte for opcode)
  - the IR(opcode register) will be 32 bits

3.2 (

Problem 1.7) why DMA access to main memory is given higher priority than processor access to main memory

- DMA is stream data so it cannot stop (its an external stream) so if main memory has priority it might make the DMA stop and you would lose data which is baaaaad.

3.3 Problem 1.8

DMA module transferring characters to main memory from external device transmitting at 9300 bps. The processor can fetch instruction at rate of 1 million instructions per second. By how much will the processor be slowed down due to DMA activity (ignore read/write operations).

- character means an interrupt every 1 byte, so 9600 bps (1200 bytes per second)
- this means that a character/byte per 838 microsecond
- since the DMA has priority it will cause this interrupt every 838 microseconds

- assuming the processor as an access every microsecond the slow down would be $1\frac{}{838=0.12percent}$

4 Lab Tutorial

4.1 Memory management

We will be implementing static and heap allocations.

The simplest way to implement heap memory is through a linked list.

An atomic operation is one that cannot be divided (not fully sure what he means here). We want this when mucking about with memory to not lose anything.