

Computer Components and BUS Interconnection



ATANU
SHOME

CSE, KU

Confession

- Most of the materials have been collected from Internet.
- Images are taken from Internet.
- Various books are used to make these slides.
- Primary reference book:
 - ❖ Computer Organization and Design: the Hardware/Software Interface - Textbook by David A Patterson and John L. Hennessy.
 - ❖ Computer Organization and Architecture - Book by William Stallings



This slide is not enough to learn these topics. It's just for your guideline. (NAH! Just a reminder for me - which topics to cover) Reading from the slides only, will be a BIG mistake. 

YOU NEED TO READ THE BOOK

Programming

- Program: A set of Instructions
- For each step, an arithmetic or logical operation is done
- For each operation, a different set of control signals is needed

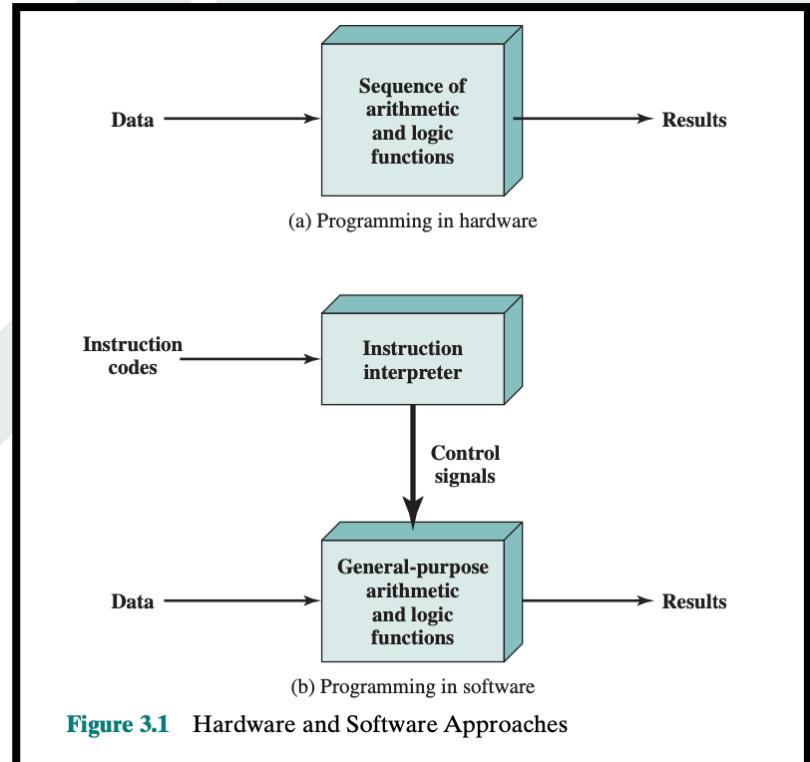
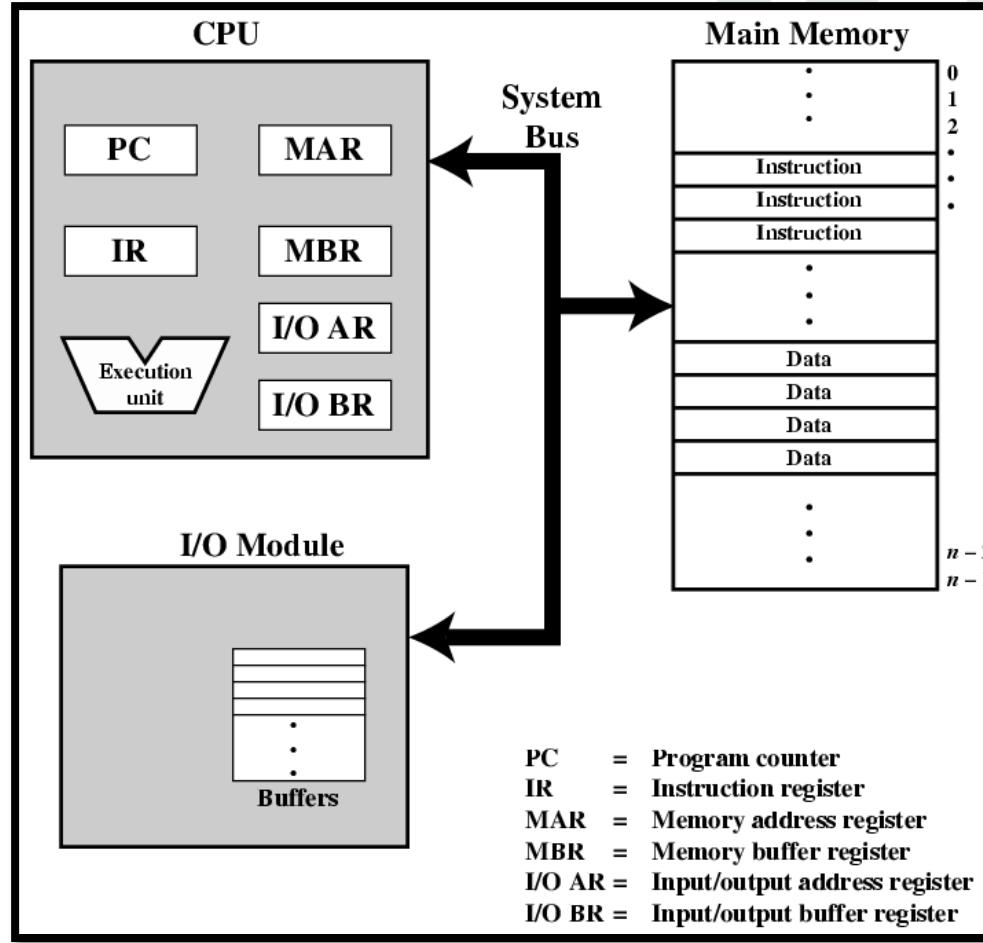
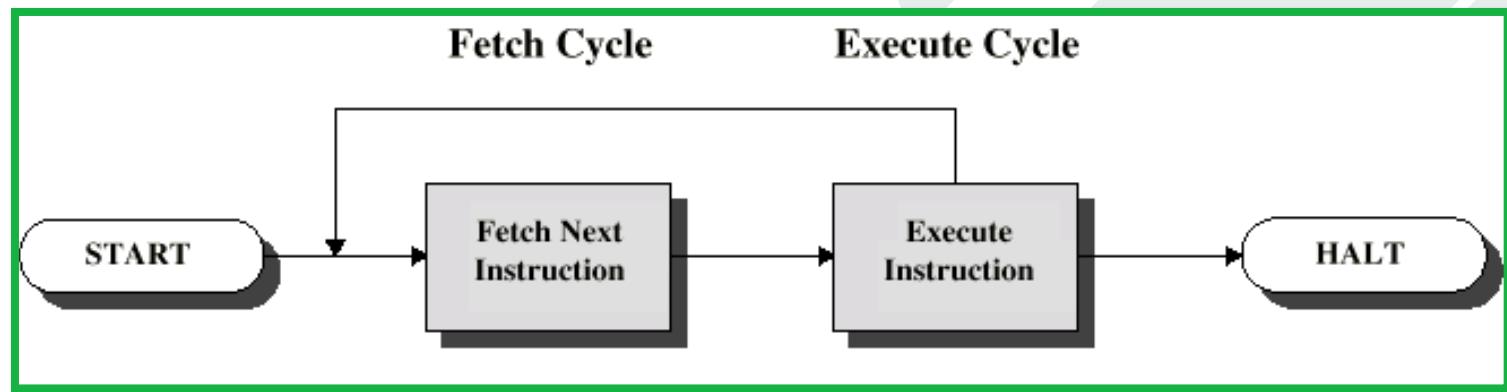
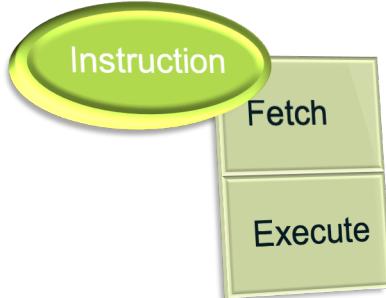


Figure 3.1 Hardware and Software Approaches

Top-Level View



Instruction Cycle



Fetch Cycle

- Program Counter (PC) holds address of next instruction to fetch
- Processor fetches instruction from memory location pointed to by PC
- Increment PC
 - Unless told otherwise
- Instruction loaded into Instruction Register (IR)
- Processor interprets instruction and performs required actions

Execute Cycle

- Processor-memory
 - data transfer between CPU and main memory
- Processor I/O
 - Data transfer between CPU and I/O module
- Data processing
 - Some arithmetic or logical operation on data
- Control
 - Alteration of sequence of operations
 - e.g. jump
- Combination of above

Example of Program Execution



(a) Instruction format



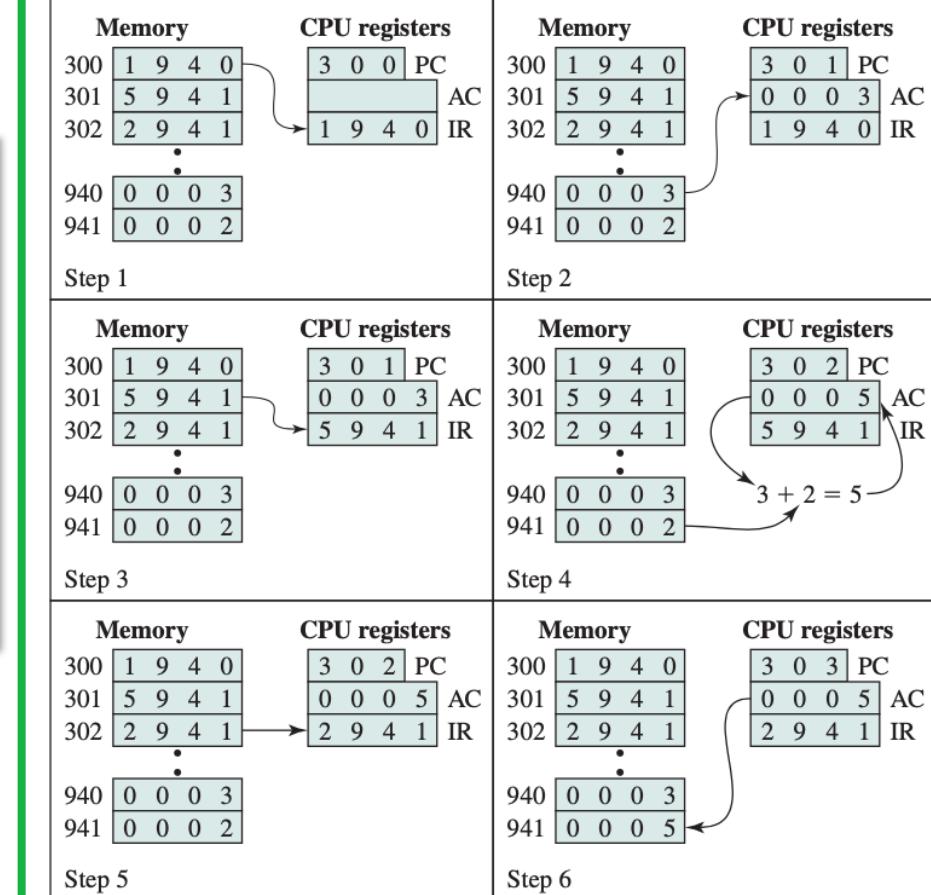
(b) Integer format

Program counter (PC) = Address of instruction
 Instruction register (IR) = Instruction being executed
 Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from memory
 0010 = Store AC to memory
 0101 = Add to AC from memory

(d) Partial list of opcodes





I FEEL SO UNDERSTOOD

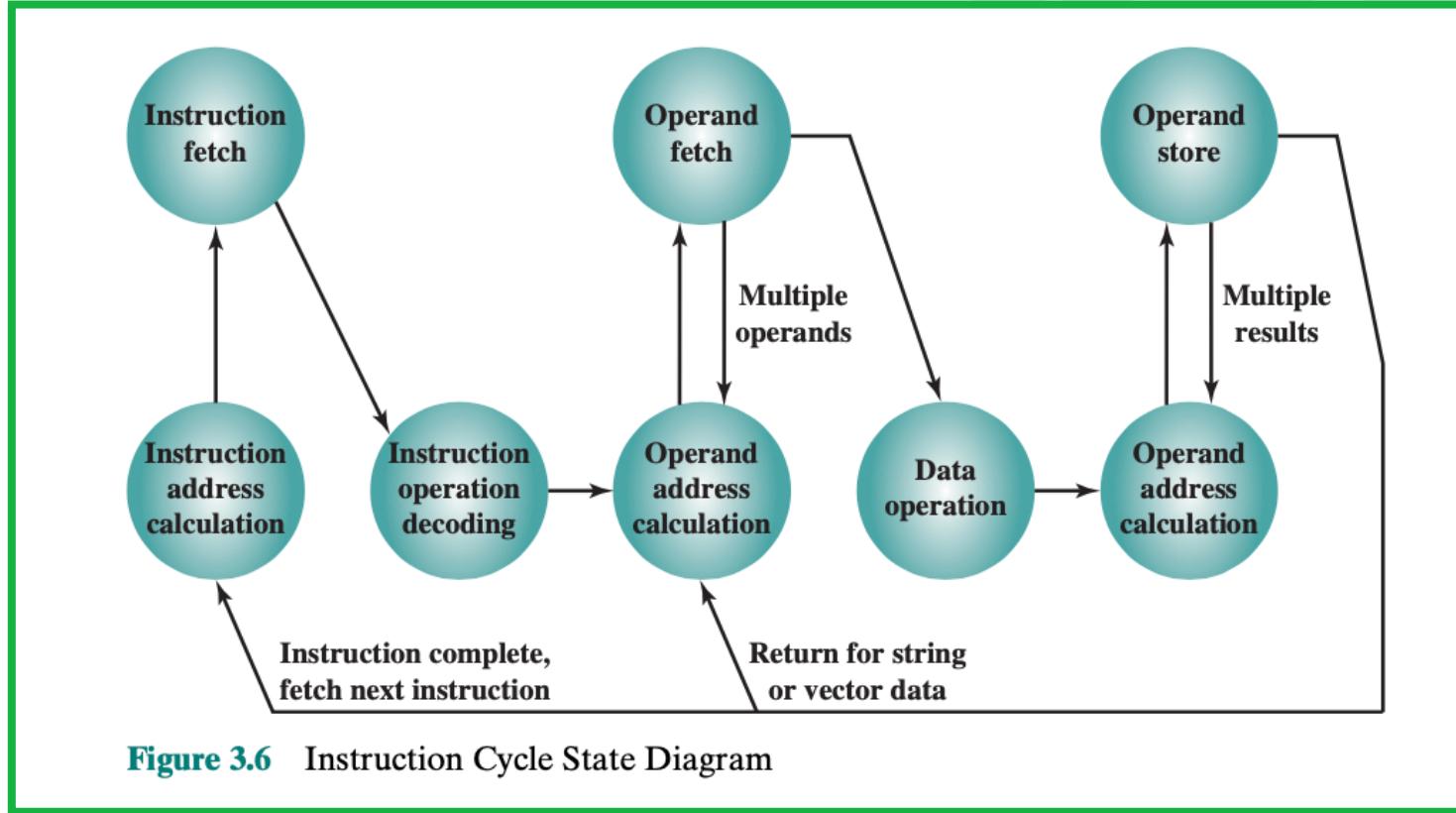
imgflip.com

Problem ???

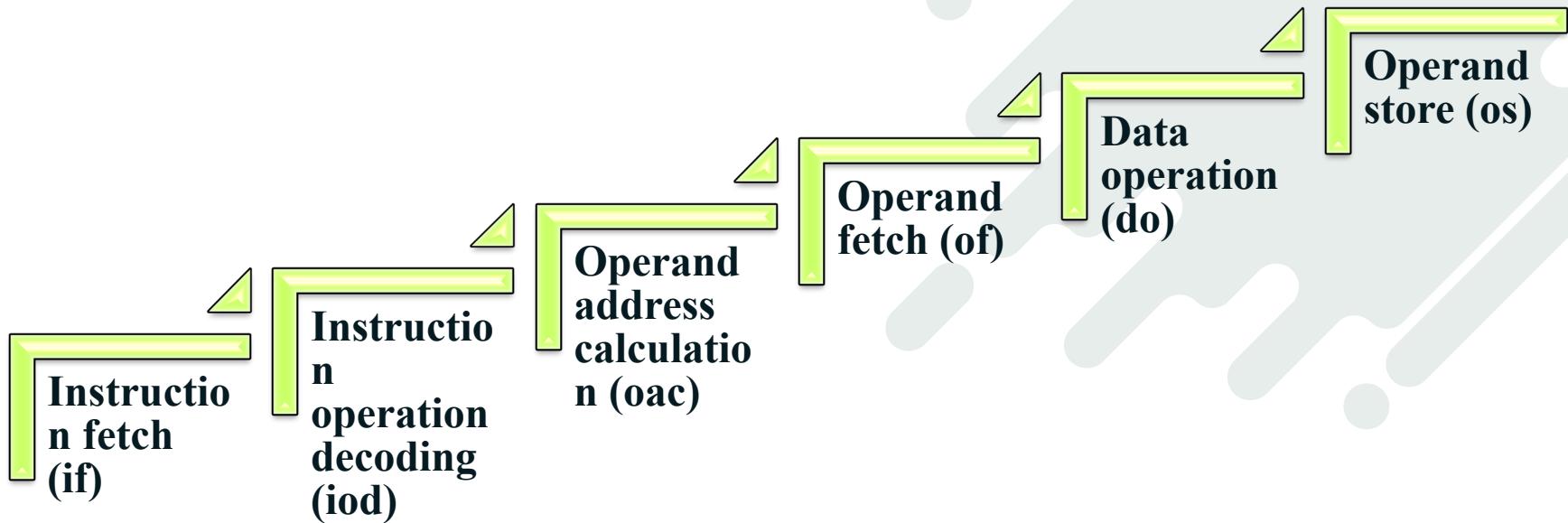
- The hypothetical machine of Figure 3.4 also has two I/O instructions:
 - 0011 = Load AC from I/O
 - 0111 = Store AC to I/O
- In these cases, the 12-bit address identifies a particular I/O device.
Show the program execution for the following program:
 - i. Load AC from device 5.
 - ii. Add contents of memory location 940.
 - iii. Store AC to device 6.
- Assume that the next value retrieved from device 5 is 3 and that location 940 contains a value of 2.

Do it yourself

Instruction Cycle - State Diagram



Instruction Cycle

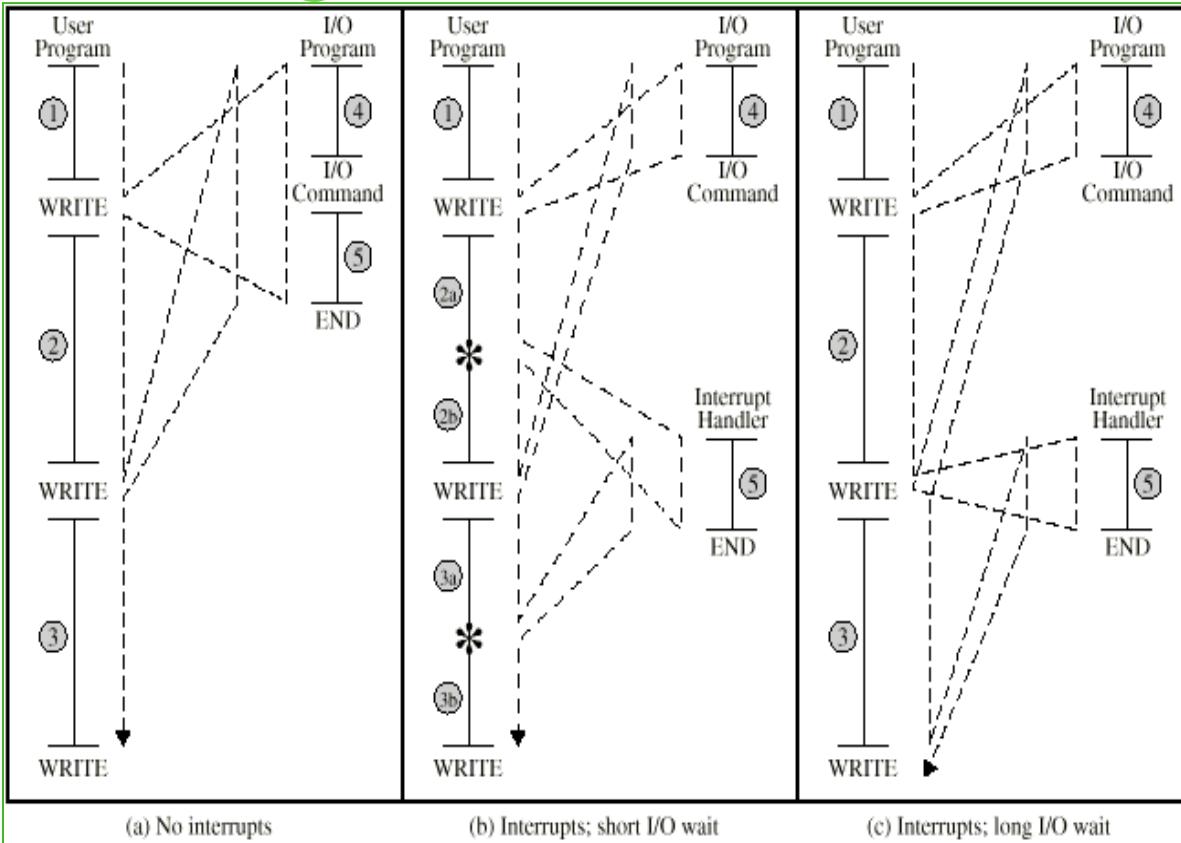


Interrupts

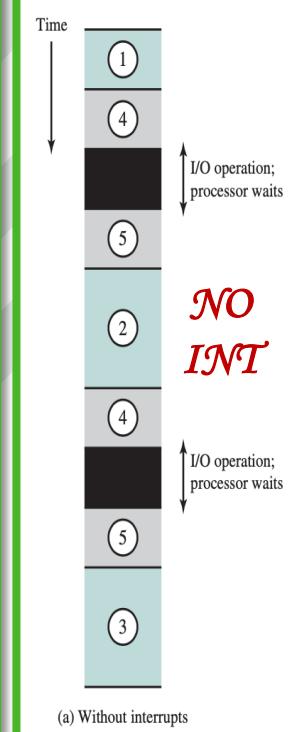
- Mechanism by which other modules (e.g. I/O) may interrupt normal sequence of processing

Program	Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, or reference outside a user's allowed memory space.
Timer	Generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis.
I/O	Generated by an I/O controller, to signal normal completion of an operation, request service from the processor, or to signal a variety of error conditions.
Hardware Failure	Generated by a failure such as power failure or memory parity error.

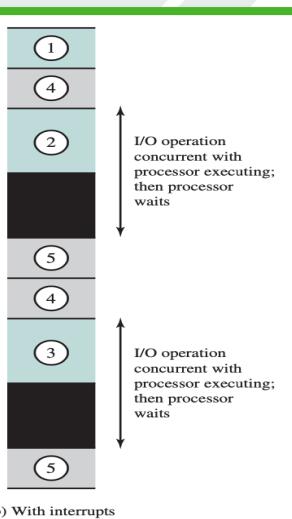
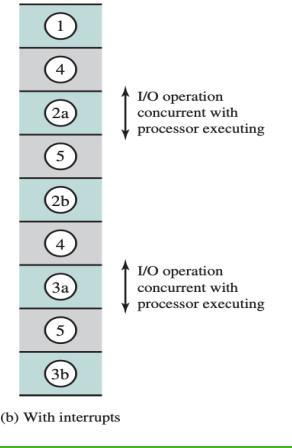
Program Flow Control



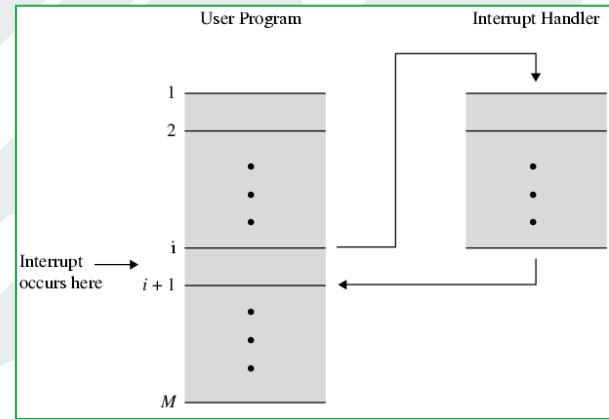
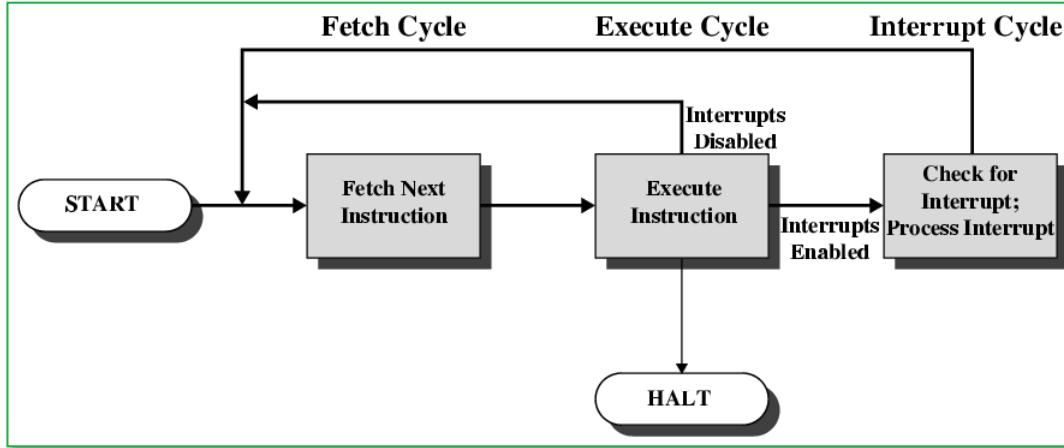
*SHORT IO
WAIT*



*LONG IO
WAIT*

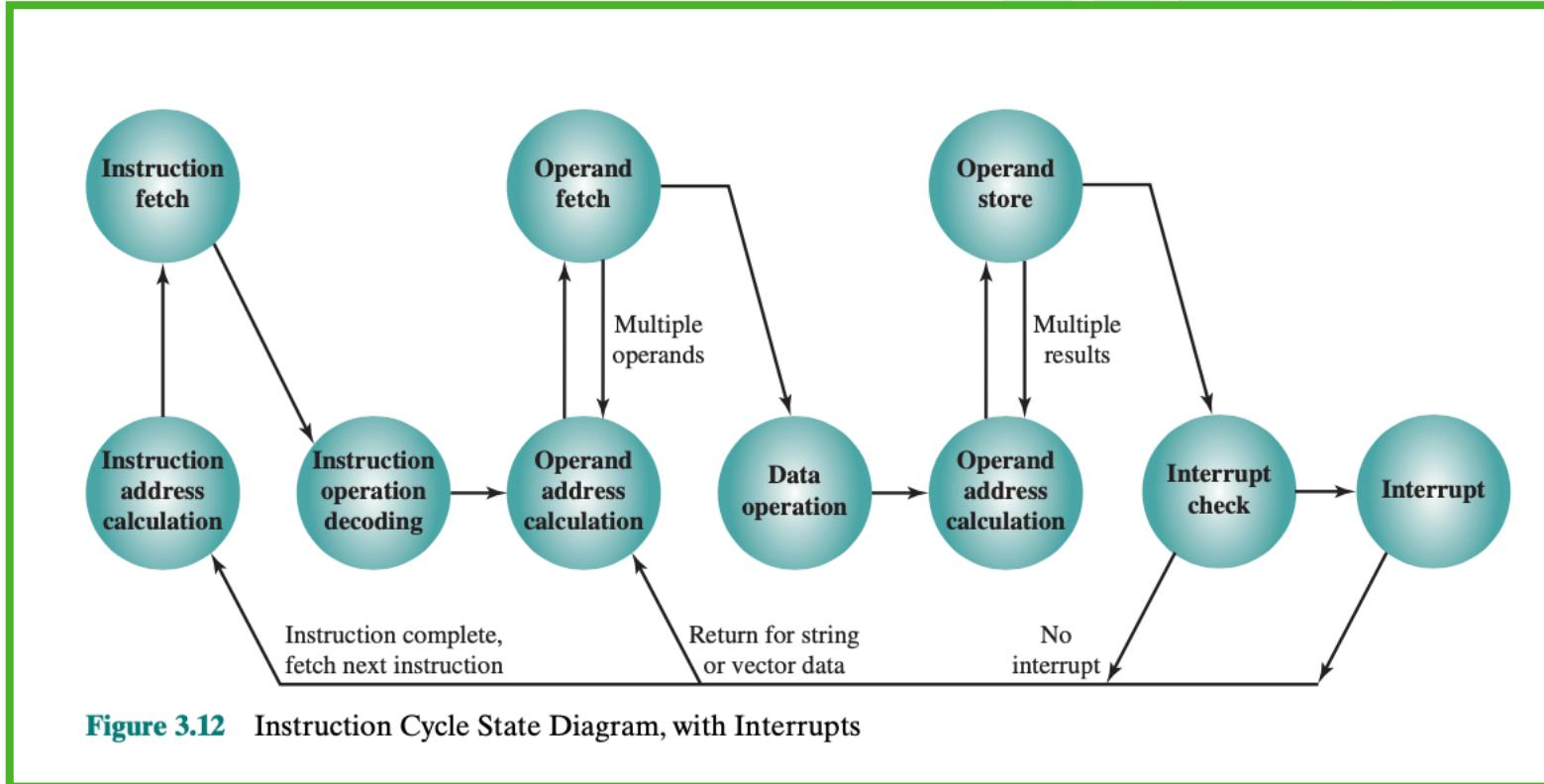


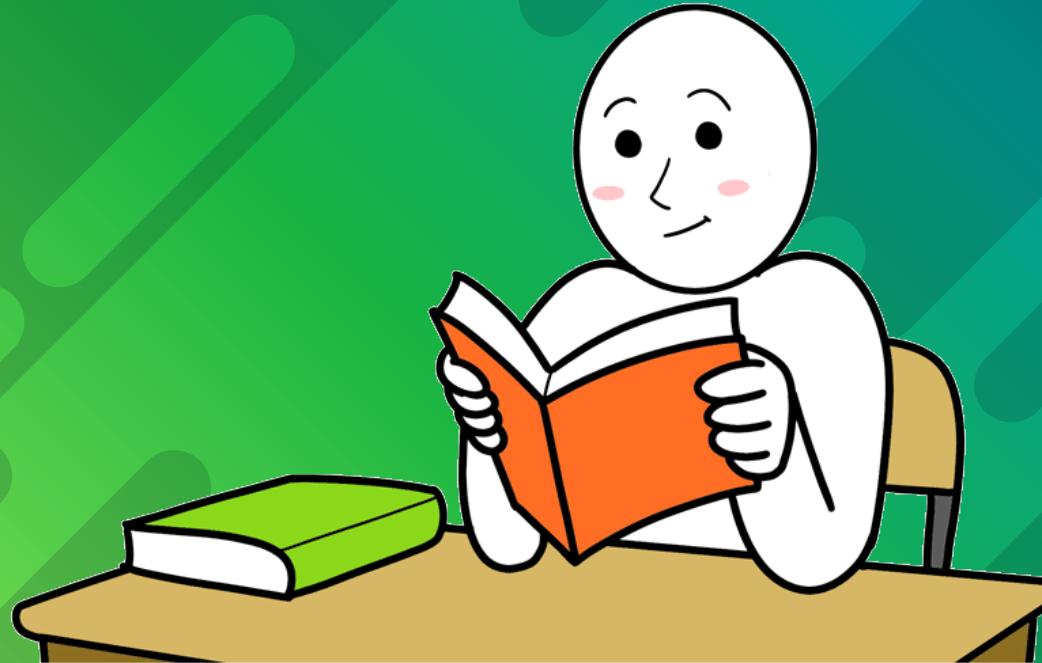
Transfer of Control via Interrupts



Instruction Cycle with Interrupts

Instruction Cycle with Interrupts



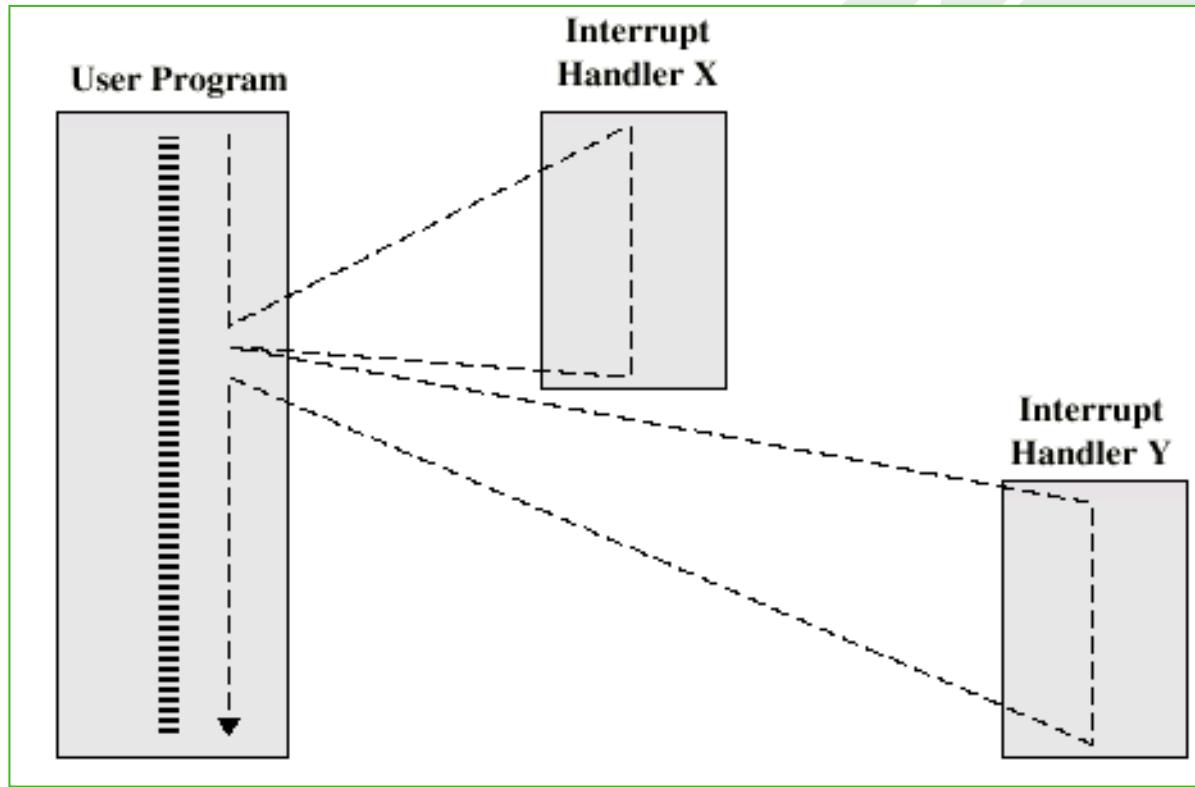


Read the Book

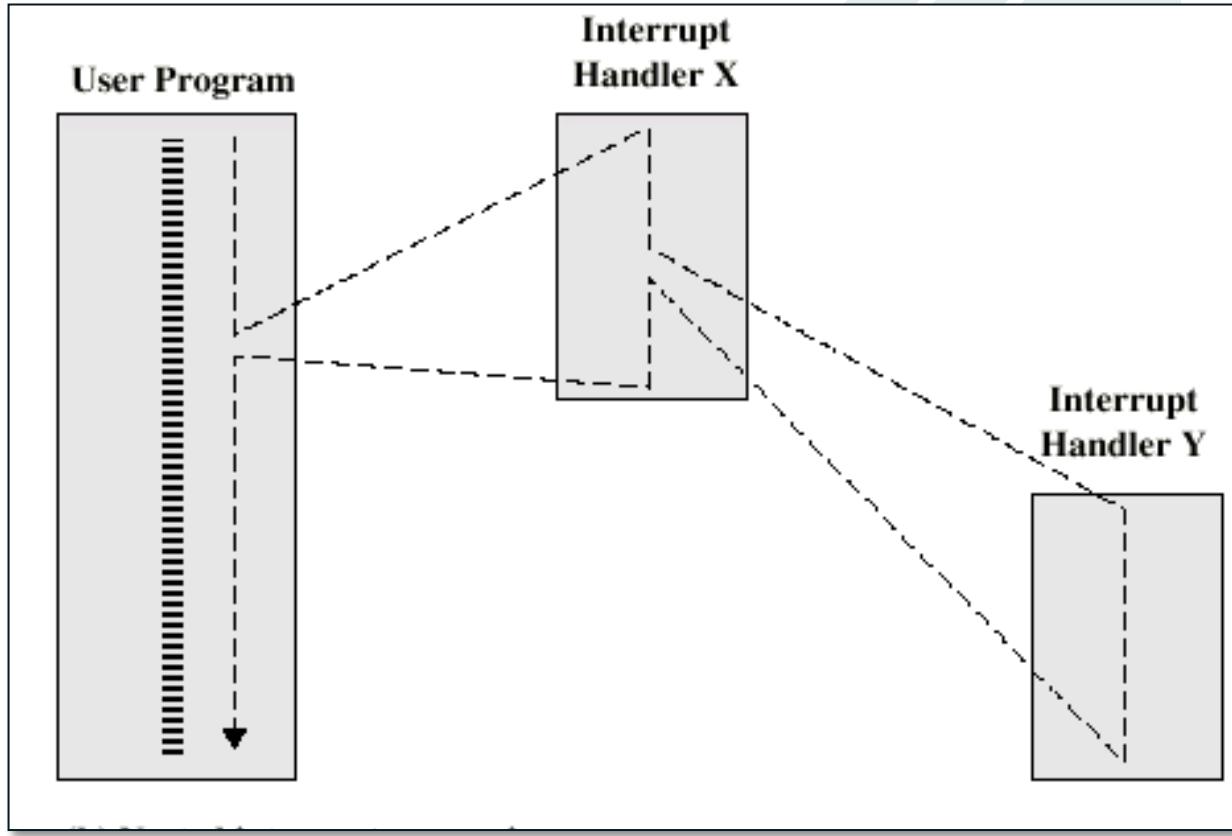
Multiple Interrupts

- Disable interrupts
 - Processor will ignore further interrupts whilst processing one interrupt
 - Interrupts remain pending and are checked after first interrupt has been processed
 - Interrupts handled in sequence as they occur
- Define priorities
 - Low priority interrupts can be interrupted by higher priority interrupts
 - When higher priority interrupt has been processed, processor returns to previous interrupt

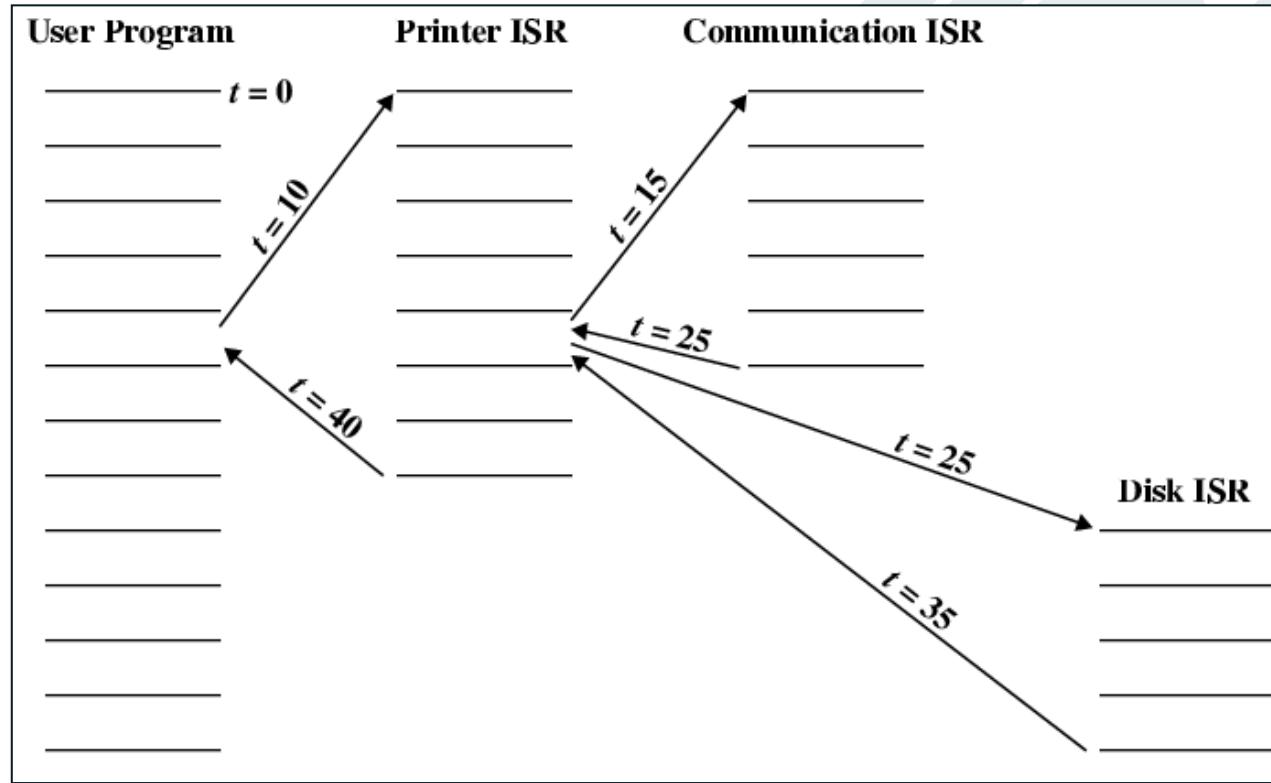
Multiple Interrupts - Sequential



Multiple Interrupts – Nested



Time Sequence of Multiple Interrupts

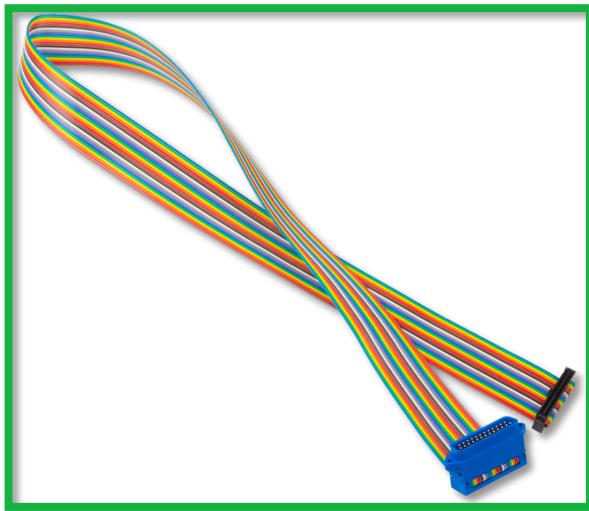


BUS Interconnections

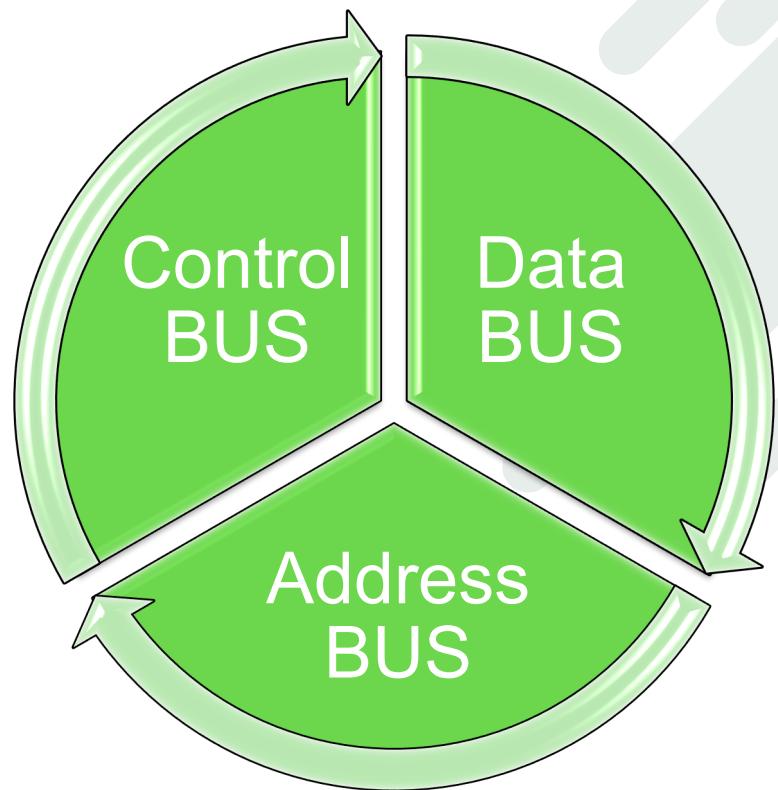
System BUS

- A BUS that connects major computer components (processor, memory, I/O) is called a **system bus**.
- The most common computer interconnection structures are based on the use of one or more system buses.

BUS



BUS



Data BUS

- Carries data
 - Remember that there is no difference between “data” and “instruction” at this level
- Width is a key determinant of ***performance***
 - 8, 16, 32, 64 bit
- If the data bus is 32 bits wide and each instruction is 64 bits long, then the processor must access the memory module **twice** during each instruction cycle.

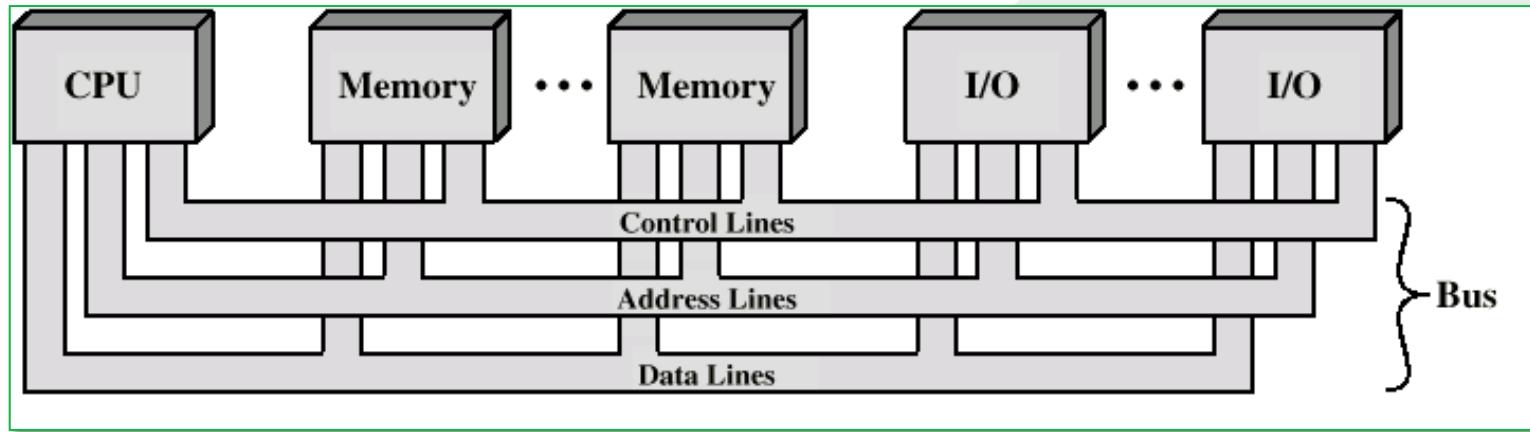
Address BUS

- Designate the source or destination of data.
- Determines the maximum possible memory capacity of the system.

Control BUS

- Both command and timing information among system modules.
- Types: Read/Write, Interrupt, Acknowledgement, Clock, Reset, etc.

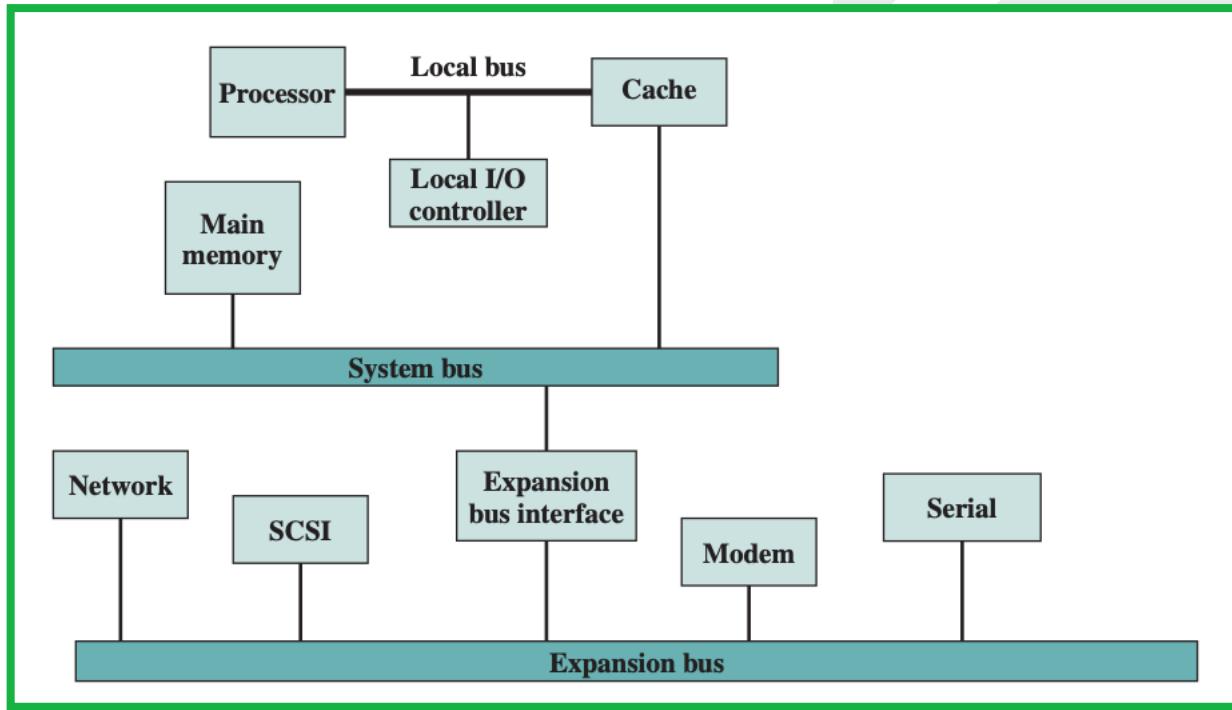
BUS Interconnection Scheme



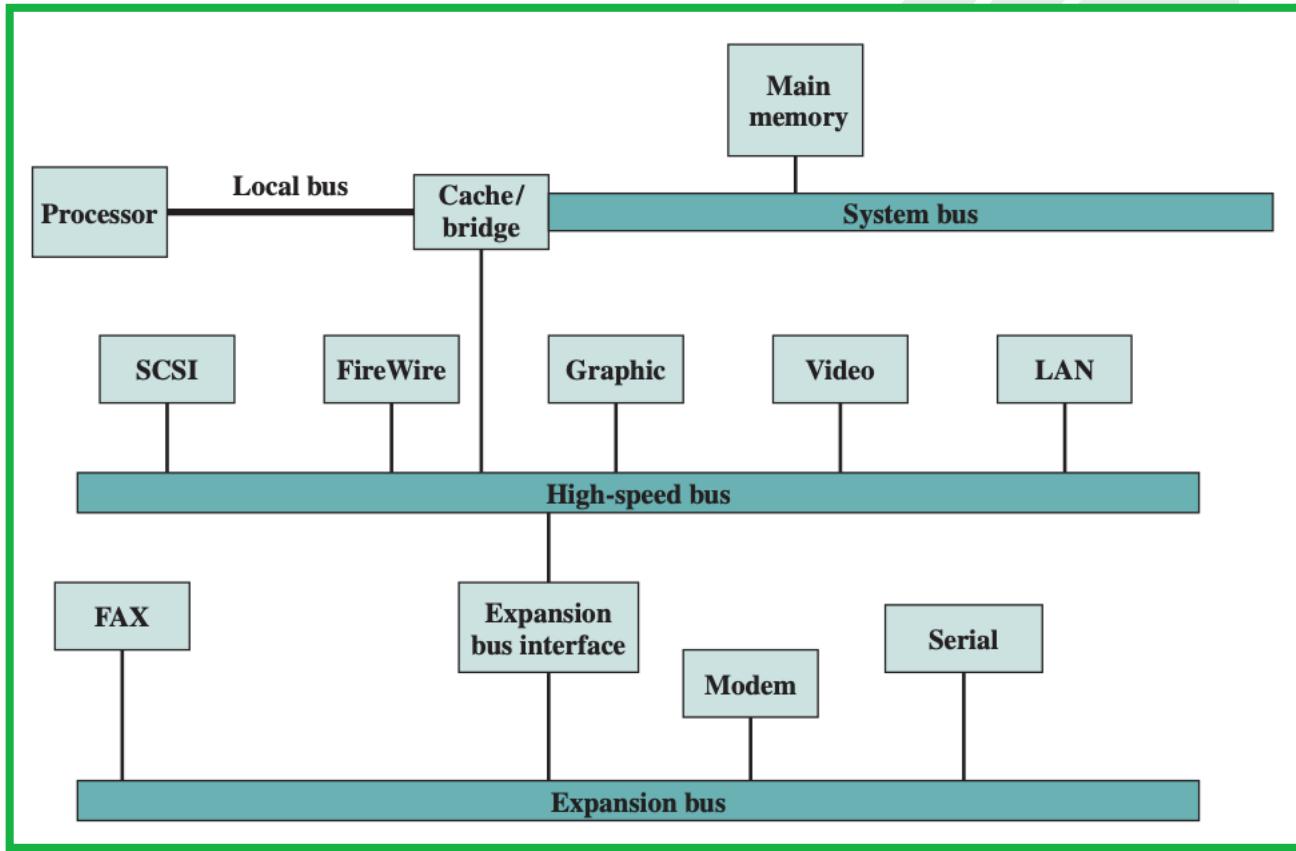
Single BUS Problem

- Lots of devices on one bus leads to:
 - Propagation delays
 - Long data paths mean that co-ordination of bus use can adversely affect performance
 - If aggregate data transfer approaches bus capacity
- Most systems use multiple buses to overcome these problems

Traditional (ISA) (with cache)



High Performance BUS

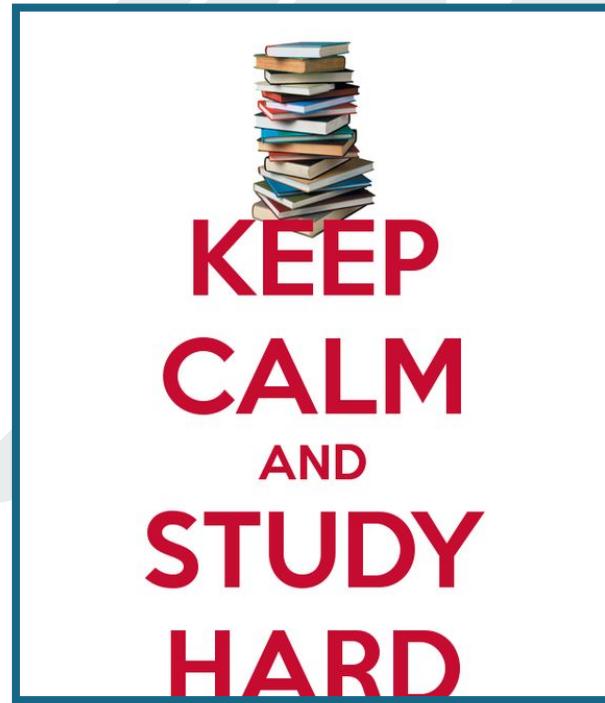


BUS Types

- Dedicated
 - Separate data & address lines
- Multiplexed
 - Shared lines
 - Address valid or data valid control line
 - Advantage - fewer lines
 - Disadvantages
 - More complex control
 - Ultimate performance

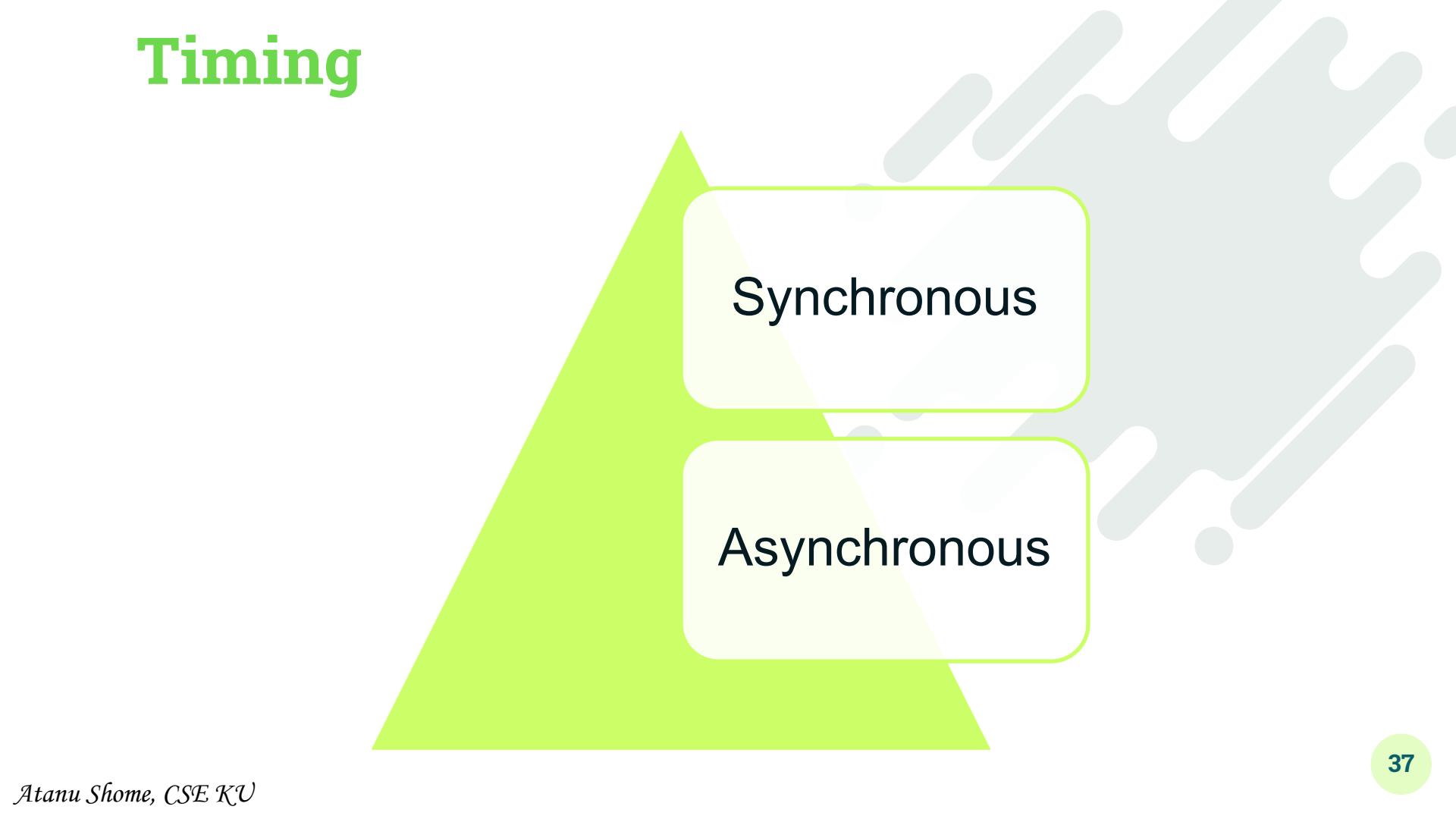
BUS Arbitration

- Centralized or Decentralized
- DMA Controller





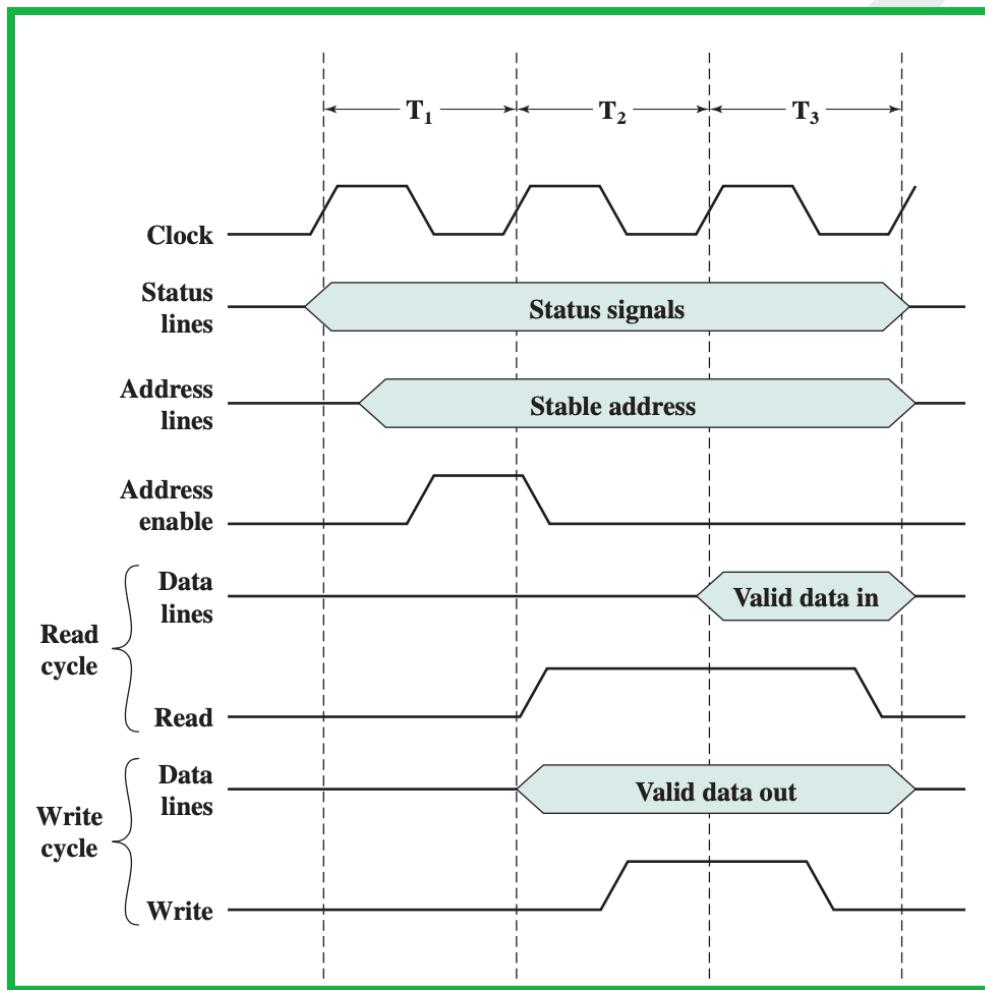
Timing



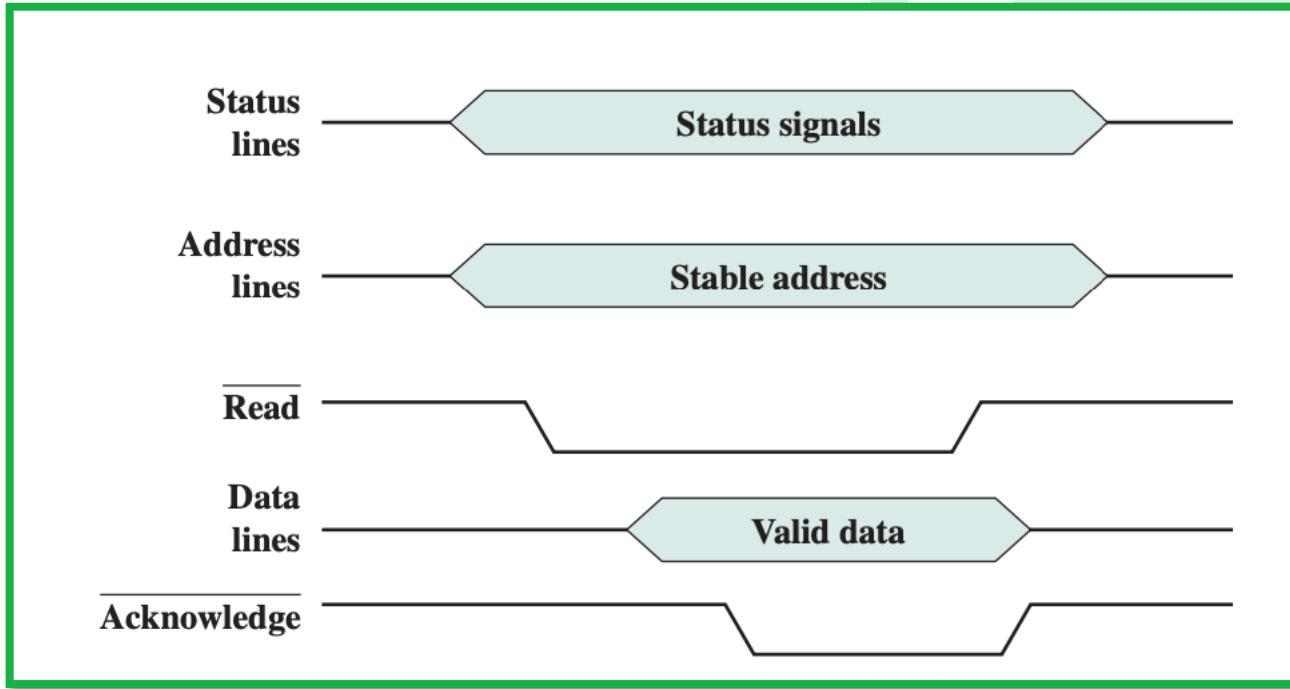
Synchronous

Asynchronous

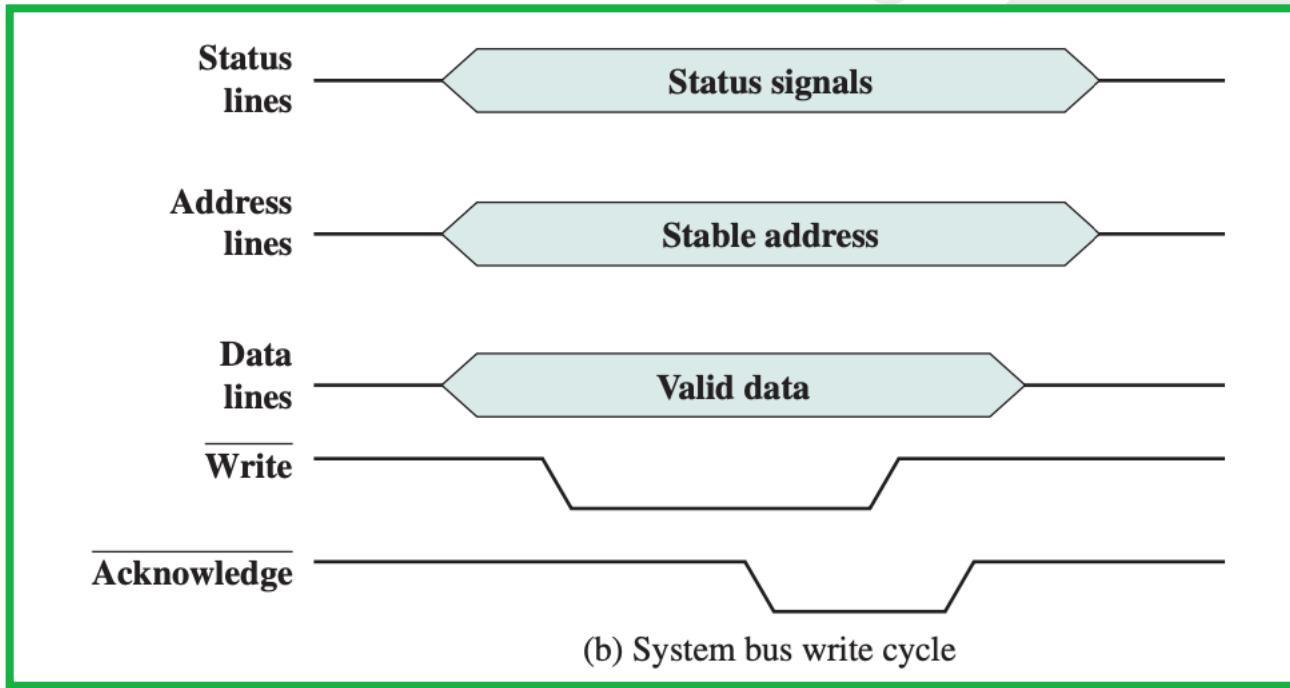
Synchronous Timing Diagram



Asynchronous Timing – Read Diagram



Asynchronous Timing – Write Diagram



Thanks!

