CMP 1203

LECTURE 2

Computer Level Hierarchy

- Computers solve problems written in many languages.
- How do we go from physical components to program execution written in a high level language?
- Breakdown computer organization into a hierarchy of levels
- Each level is tasked with a specific function and exists distinctly as a hypothetical computer called a “virtual machine”
- Each virtual machine executes its specific set of instructions and can call on a lower level to perform tasks as required

Computer Level Hierarchy

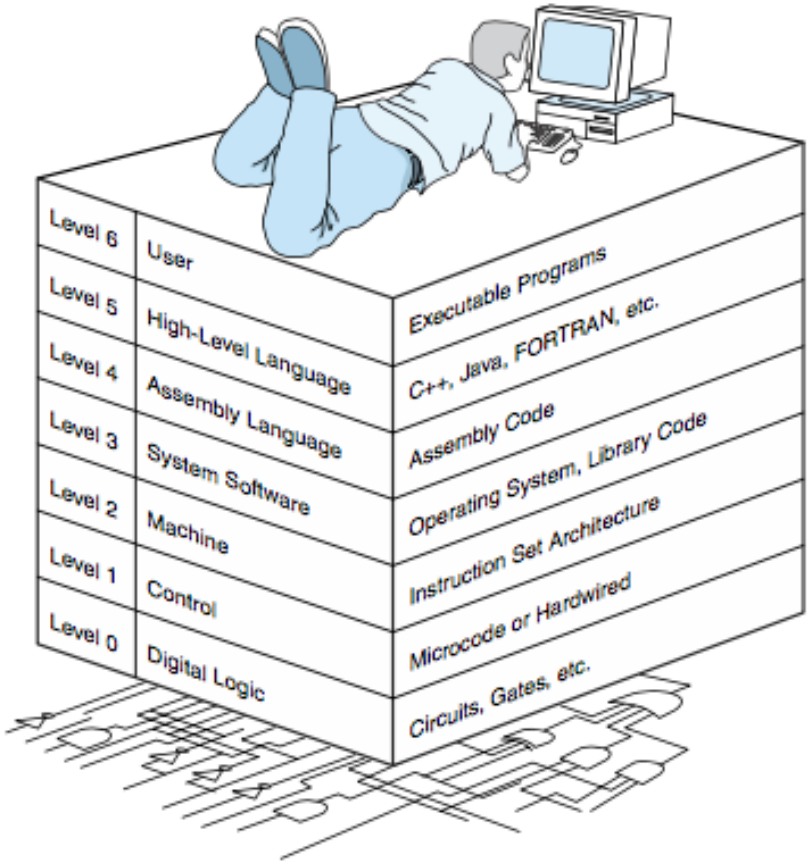


FIGURE 1.3 **The Abstract Levels of Modern Computing Systems**

Image source: Null and Lobur

- **Level 6:** applications run by users: word processors, graphics, games etc.
- **Level 5:** High level languages
- Translated to a language computer can understand by compiler/interpreter i.e. to assembly language and machine code

Computer Level Hierarchy

- **Level 4:** Assembly Language
- **Level 3:** operating system instructions.
- Tasks e.g. multiprogramming, process synchronization, etc.
- **Level 2:** ISA or machine level- programs written in this can be directly executed by computer without need for compiler
- **Level 1: Control**
- Instruction decoding and execution
- Movement of data at required time and to required places
- Control units can be hardwired or micro programmed

Computer Level Hierarchy

- **Hardwired control units**
- Control signals come from blocks of logic components
- Signals direct data and instructions to appropriate places
- Are fast because they are actually hardware components but difficult to modify once implemented
- **Micro programmed control units**
- A micro program is a program written in a low level language that can be directly implemented by the hardware
- Machine instructions produced by level 2 are passed to micro program which interprets them and activates the required hardware to perform required task
- Micro programs can be modified easily but are slower due to added level of translation
- **Level 0:** Digital Logic Level: physical components of a computer system

Von Neumann Model

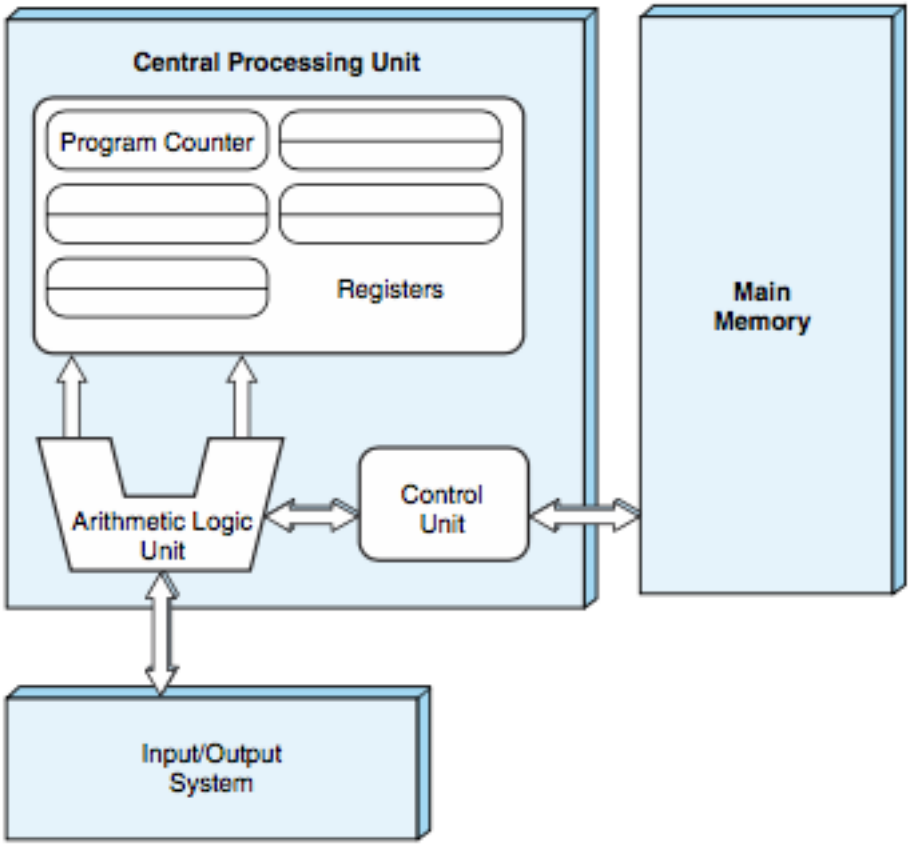


FIGURE 1.4 The von Neumann Architecture

1. Three hardware systems viz:
- CPU (ALU + registers + program counter + control unit)
- Main memory
- I/O system

2. able to carry out sequential instruction processing

3. Has one path (logical or physical) between main memory and CPU ➔ has to alternate between instruction and execution cycles ➔ ***von neumann bottleneck***

Von Neumann Execution cycle

- **Fetch-Decode-Execute cycl**e : programs run in this way in this model
- 1. Control unit gets next instruction to be executed from memory. Its located using a program counter
- 2. Instruction decoded into a language the ALU can understand
- 3. Obtain any data operands required for instruction execution from memory and placed into registers in CPU
- 4.ALU executes instruction and places results in registers or memory

Von Neumann model

- Its ideas have been extended over time to include things such as: buses, index registers, floating point data, virtual memory etc.
- The system bus model is shown below
- Data bus – moves data from memory to CPU and registers and vice versa
- Address bus - holds addresses of data the data bus is currently accessing
- Control bus – carries control signals that specify how information transfer should occur

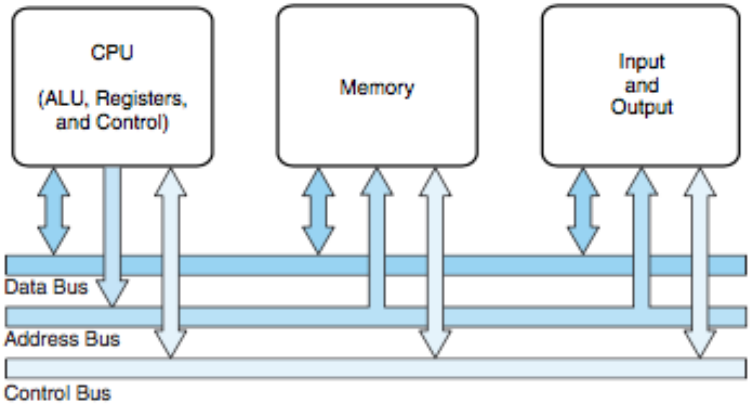


FIGURE 1.5 **The Modified von Neumann Architecture, Adding a System Bus**

Image source: null and lobur

CPU Basics

- **Role**: fetch instructions, decoding them and performing the required operations on the correct data
- Consists of a data path and a control unit
- Data path: registers (storage) and ALUs (manipulate data) connected by buses (move data from place to place); timing is controlled by clocks
- Control unit: sequences operations and ensures correct data is where it needs to be at the correct time
- Design of these two directly affects the performance of a computer

CPU basics: Registers

- Places for data storage – addresses, program counters, data needed for program execution
- Basically stores binary data
- Located on processor to enable quick information access
- Most computers have registers of set sizes (e.g. 16, 32 and 64 bits) since data processing is done on binary words of a fixed size
- Can be special purpose or general purpose i.e. holding only specific data or different data at various times
- Information is written to, read from and transferred to registers

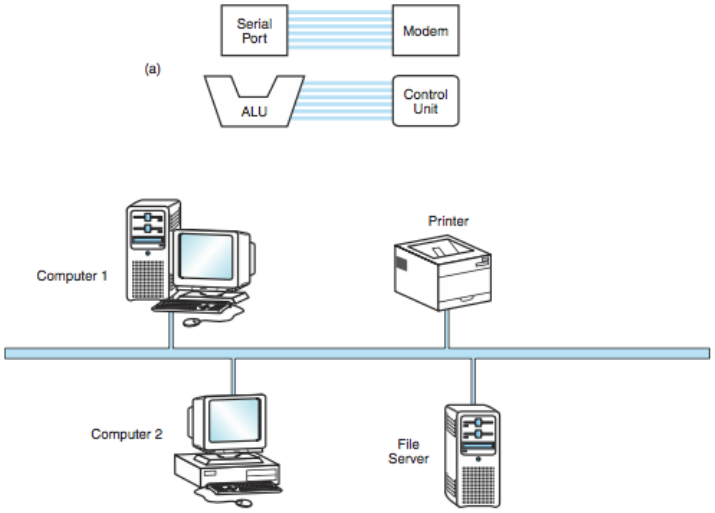
CPU basics

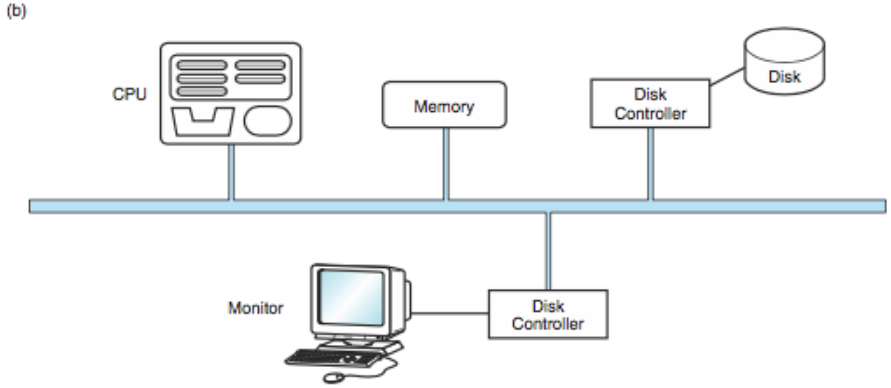
- **The ALU**
- Carries out logic operations e.g. logic or arithmetic
- Usually 2 inputs and 1 output
- Controlled by signals from control unit
- **The Control Unit**
- Manages CPU
- Monitors instruction execution, information transfer, extracts instructions from memory, decodes it, ensures data in the right place at right time, tells ALU which registers to use
- Uses a program counter register to identify the next instruction
- Uses a status register for tracking – overflows, carries, borrows etc.

The Bus

- CPU communicates with other components through a bus
- Set of wires ➔shared but common data path connecting multiple subsystems in a computer
- Multiple lines to allow parallel movement of bits
- Only one device can use the bus at a particular time : sharing leads to bottleneck
- Speed of bus determined by length and by number of devices that share it
- Can be point to point (connects two components) or multipoint (connecting multiple devices)

Buses





Point to point

multi point

image source: lobur and null

Buses

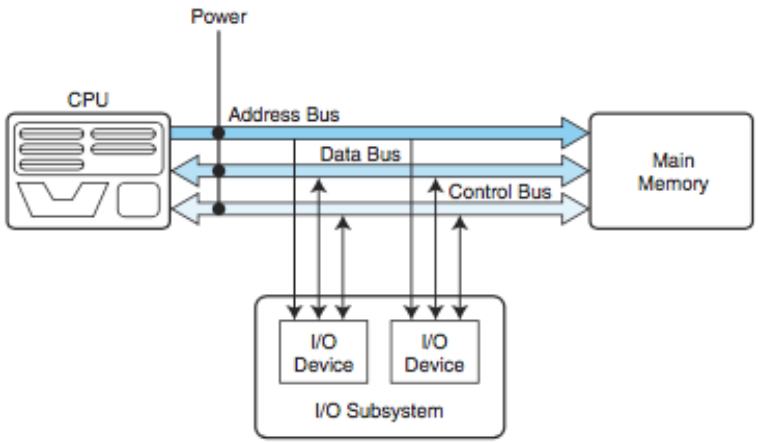


FIGURE 4.2 **The Components of a Typical Bus**

- **Bus protocol**: rules that govern sharing of bus
- **Data bus**: contain information to be moved from place to place
- **Control lines**
  - which device has permission to use bus at particular time and for what (read/ write to memory or I/O device)
  - transfer ACKs for bus requests, interrupts, clock synchronization signals
- **Address lines**: location that data should be read/ written from/to
- **Power lin**es: provide electrical power

Buses

- **Operations:**
- Sending addresses for read/write
- Memory read
- Memory write
- I/O reads and writes from peripherals

Types:

Processor –memory bus: short, high speed and matched to memory to maximize data transfer

I/O buses: longer, allow different types of devices

Back plane bus: built into machine chassis and connects processor, I/O devices and memory

Other terminologies: system bus, external/expansion buses, local buses : read on your own!

Buses

- Synchronous buses: clocked, events happen only at the tick of a clock
- Clock rate: rate at which clock ticks
- Bus cycle time – reciprocal of the bus clock rate
- Bus must be as short as possible
- Bus cycle time must be longer than the length of time it takes data to move across bus

Buses

- Asynchronous buses: control lines coordinate operation
- Require complex ***hand shaking*** protocol; but this improves scalability
- Bus master: can initiate transfer of information (control bus) so able to reserve the bus
- Bus slaves: only respond to to read/write requests from master

Clock

- Regulates how quickly instructions can be executed
- Synchronizes all system components
- CPU requires fixed number of clock ticks to execute each instruction thus instruction performance is measured in clock cycles
- A clock cycle is the time between clock ticks
- Clock frequency is measured in Hz
- Registers must wait for the clock to tick in order to load new data.
- **Question:**
- Can we make machine run faster by speeding up the clock?

Clock

- There is a limit to how short we can make the clock cycles!
- With each tick, register data changes ➔ register outputs change
- New data must propagate through all circuits in the system
- Clock cycle must be long enough to to allow the changes to reach next set of registers else there would be data inconsistencies
- Minimum clock cycle must be at least as big as maximum propagation delay of the circuit.

- Relating seconds to cycles

$$\text{CPU time} = \frac{\text{seconds}}{\text{program}} = \frac{\text{instructions}}{\text{program}} \times \frac{\text{average cycles}}{\text{instruction}} \times \frac{\text{seconds}}{\text{cycle}}$$

I/O Subsystem

- Users communicate with computers through I/O devices.
- Not connected directly to CPU but through interfaces which convert system bus signals to formats acceptable by the peripherals

Further reading: memory-mapped I/O vs. instruction-based I/O

Lecture Handout

- Chapter 1 and 4 of Null and Lobur
- Chapter 3 of Stallings