Chapter 1

terms:

computer architecture: refers to those attributes of a system visible to a programmer or, those attributes that have a direct impact on the logical execution of a program.
computer organization: refers to the operational units and their interconnections that realize the architectural specifications.

Examples of each terms.
A family of compute models: all with the same architecture but with differences in organization.
In general terms, the **function of computer** are as following: Data processing, Data storage, Data movement, Control.
There are four main **structural components** within a computer: Central processing unit (CPU), Main memory, I/O, System interconnection.

Chapter 2

terms:

The **stored-program concept**: Binary numbers for both data and instructions, An arithmetic-logic unit performs the basic operations, The data and instructions are stored in the memory, selected by addresses, The proper sequencing of the operation is controlled by a control unit.
5 **functional units** of the von Neumann Machine: ALU, memory, control unit, input devices, output devices.
A **main memory**, which stores both data and instructions. An arithmetic and logic unit (**ALU**) capable of operating on binary data. A **control unit**, which interprets the instructions in memory and causes them to be executed. Input and output (**I/O**) equipment operated by the control unit.
Set of registers: **MBR** (Memory Buffer Register): Contains a word to be stored in memory, or is used to receive a word from memory. **MAR** (Memory Address Register): Specifies the address in memory of the word to be written from or read into the MBR. **IR** (Instruction Register): Contains the opcode instruction being executed. **IBR** (Instruction Buffer Register): Employed to hold temporarily the right-hand instruction from a word in memory. **PC** (Program Counter): Contains the address of the next instruction-pair to be fetched from memory. **AC** (Accumulator) and **MQ** (Multiplier Quotient): Employed to hold temporarily operands and results of ALU operations.
Instruction cycle: **Fetch cycle** and **Execute cycle**.
Designing for Performance：System clock speed, CPU constant frequency, constant cycle time，CPI, MIPS, MFLOPS, Amdahl's Law.

$$Speedup = \frac{\text{time to execute program on a single processor}}{\text{time to execute program on } N \text{ parallel processors}} = \frac{T(1-f)+Tf}{T(1-f)+\frac{Tf}{N}} = \frac{1}{(1-f)+\frac{f}{N}}$$

A brief history of computers: The First generation: Vacuum Tubes; The Second generation: Transistors; The Third generation: Integrated Circuits; Later generation: LSI, VLSI……

Chapter 3

terms:

The **basic function** performed by a computer is **execution of a program**. The processing required for a single instruction is called an **instruction cycle**.
The detail of **instruction fetch and execute**.
Interrupts are provided primarily as a way to improve processing efficiency.
When the external device becomes ready to be serviced, the I/O module sends an **interrupt request signal** to the processor.
The detail of **interrupt cycle**.
A **disabled interrupt** means that the processor can and will ignore that interrupt request.
A bus is a communication pathway connecting two or more devices. A bus that connects major computer components is called a **system bus**.
The **Data Bus** provides a path for moving data between system modules. Width of Data Bus is a key determinant of performance. **Addressing bus** specifies the source or destination of data. Bus width determines maximum memory capacity of system. The **control bus** controls the use of data and address lines.
Bus hierarchy: **System bus**: connects to CPU and the memory. **Expansion bus**: device bus, connects to I/O system. **Local bus**: for high speed devices, in between of the above.
Bus lines can be separated into two generic types: **dedicated and multiplexed**.
Bus Arbitration: The various methods can be roughly classified as being either **centralized or distributed**.
Centralised: Single hardware device controlling bus access - bus controller (or arbiter): **The chaining method**, **the counter polling method**, **independent method**.
Distributed: there is no central controller.
Timing: Refers to way in which events are coordinated on bus. **Synchronous timing**: Occurrence of events on bus is determined by a clock. **Asynchronous timing**: Occurrence of one event on a bus follows and depends on occurrence of a previous event.
The detail of Synchronous Read and Synchronous Write.
The detail of Asynchronous Read and Asynchronous Write Operations.
The **width of data bus** impacts system performance (How much information can be transferred at once). The width of address bus impacts system capacity (How much

information can be stored).
The bus speed reflects how many bits of information can be sent across each wire each second. Bandwidth, also called throughput, bus transfer rate, refers to the total amount of data that can be transferred on the bus in a second. **Bandwidth** = bus width x bus frequency. Bus speed: reflects how many bits of information can be sent across each wire each second. Bandwidth, also called throughput(吞吐量), bus transfer rate, refers to the total. Bandwidth = bus width x bus frequency

Chapter 4

terms:

Characteristics of memory systems: Location, Capacity, Unit of transfer(Internal, External), **Access method** (Sequential access, Direct access, Random access, Associative), Performance, **Physical type** (Semiconductor, Magnetic, Optical, Others), **Physical characteristics** (Volatile, Nonvolatile), Organisation.
The **hierarchy** of main memory: Consists of distinct levels of memory components. Include: Registers, L1 Cache, L2 Cache, Main memory, Disk cache, Disk, Optical, Tape.
Locality of reference for the memory. A cache hit/miss, Hit Ratio, Hit time, Miss Ratio, Miss Penalty
The flowchart of cache read operation.
Mapping function of cache: **direct mapping**, **associative mapping**, **set associative mapping**.
Cache Hits and Misses, The access time of cache.
Write Policy of cache: **Write through**, **Write back**.

Chapter 5

terms:

characteristic of **ROM** (ROM, PROM, EPROM, EEPROM, Flash) and **RAM** (SRAM, DRAM).- Read/Write, Volatile/ Nonvolatile, Temporary storage
Organisation of memory (Generic pin configuration, Larger, Wider, address decoding, Module organization, two method of deconding strategies).
DRAM refresh and row access.
Main memory timing parameters: access time, cycle time.
Error Correction （Hamming Error-Correcting Code）

Chapter 6

terms:

Data organization and formatting of magnetic disk: **cylinde**r, **tracks**, **sectors**.

Disk performance parameters: **seek tim**e, **rotational delay**, **access time**, **transfer time**.
Concept of RAID, Solid state drives

Chapter 7

terms:

The major functions or requirements for an I/O module: control and timing, processor communication (command decoding, data, status reporting, address recognition), device communication, data buffering, error detection.
With **programmed I/O**, data are exchanged between the processor and the I/O module.
With **interrupt-driven I/O**, the processor issues and I/O command, continues to execute other instructions, and is interrupted by the I/O module when the latter has completed its work.
In **direct memory access (DMA)**, the I/O module and main memory exchange data directly, without processor involvement.
DMA 控制总线的方式
Two types of addressing mode are used: **Memory-mapped I/O**, there is a single address space for memory locations and I/O devices. **Isolated I/O**, the address space for I/O is isolated from that for memory.
The detail of interrupt processing.
For device identification, there are four general categories of techniques are in use: Multiple interrupt line – each device has a IRQ, Software poll, Hardware poll (Vectored interrupt), Bus arbitration (vectored).
The detail of DMA operation.

Chapter 8

terms:

Memory Management (What is Swapping? Fragmentation, Paging)
User mode /Kernel mode
The distinguish of logical address and physical address.
Paging, demand paging (page fault). Virtual memory

Chapter 9

terms:

Integer representation （unsigned, sign magnitude, one's complement, two's complement, biased）, Converting between different bit lengths. Sign extension
Overflow of integer representation.

Integer arithmetic (Negation, addition and subtraction, multiplication (unsigned integers, two complement multiplication), arithmetic shift, division)
Floating-point representation (Sign, significand, exponent, bias representation, floating point range, normalization)
Floating-point arithmetic (addition, subtraction). IEEE 754 Formats

Chapter 10

terms:

Elements of a machine instruction: operation code, source operand reference, result operand reference, next instruction reference. Instruction representation.
Instruction types: data processing, data storage, data movement, control.
Number of addresses: 3 addresses, 2 addresses, 1 addresses, 0 addresses. More addresses/Fewer addresses
The most important of the fundamental design issues include the following: **Operation repertoire**, **Data types**, **Instruction format**, **Registers**, **Addressing**.
Type of operations: data transfer, arithmetic, logical, conversion, I/O, system control, Transfer of control.

Chapter 11

terms:

The most common addressing techniques are: immediate, direct, indirect, register, register indirect, displacement (relative addressing, base-register addressing, indexing), stack. (**concepts, detail, combination of each other**)
Instruction format:includes opcode, (implicit or explicit) operand (instruction length, allocation of bits).
The width of opcodes/ the width of operands, addressing modes:determine the complexity and the length of the instruction

Chapter 12

terms:

processor organization: ALU, CU, registers, internal CPU bus.
The user-visible registers, control and status registers.
user-visible registers: general purpose registers, data registers, addressing registers (segment pointers, index registers, stack pointer).
Four registers are essential to instruction execution: **Program counter (PC)**, **Instruction register (IR)**, **Memory address register (MAR)**, **Memory buffer register (MBR)**.
The **program status word (PSW)** typically contains condition codes plus other status

information. Common fields or flags include the following: sign, zero, carry, equal, overflow, interrupt enable/disable, supervisor.
Data Flow (**Instruction Fetch**).
Data Flow (**Indirect Diagram**).
Data Flow (**Execute**)
Data Flow (**Interrupt Diagram**)
The concept of **instruction prefetch**. Instruction pilelining
Timing diagram for instruction pipeline operation. Branch in pileline
Serveral approaches have been taken for dealing with braches: multiple streams, pretetch branch target, loop buffer, branch prediction, delayed branching

Chapter 13

terms:

The key elements of **RISC** systems: Large number of general purpose registers or use of compiler technology to optimize register use. Limited and simple instruction set. Emphasis on optimising the instruction pipeline.
Characteristics of RISC: One instruction per cycle. Register to register operations. Few, simple addressing modes. Few, simple instruction formats.
Optimization of pilelining: delayed branch, delayed load, loop unrolling

Chapter 14

terms:

The essence of the **superscalar** approach is the ability to execute instructions independently in different pipelines.
Superpipelining exploits the fact that many pipeline stages need less than half a clock cycle.
Instruction level parallelism refers to the degree to which, on average, the instructions of a program can be executed in parallel. Governed by data and procedural dependency.
Machine parallelism: ability to take advantage of instruction level parallelism. Governed by number of parallel pilelines.
The **fundamental limitations** to parallelism with which the system must cope: true data dependency, procedural dependency, resource conflicts, output dependency, Antidependency.
Instruction issue policy: Order in which instructions are fetched, Order in which instructions are executed, Order in which instructions change registers and memory. In-order issue with in-order completion, in-order issue with out-of-order completion, out-of order issue with out-of-order completion.

Chapter 15

terms:

Micro-operations are the functional, or atomic, operations of a processor.
The main micro-operations of the **fetch cycle** and registers involved (MAR, MBR, PC, IR).
Sequence of events for fetch cycle.
The main micro-operations of the **Indirect cycle** and registers involved.
Sequence of events for Indirect cycle.
The main micro-operations of the **Interrupt cycle** and registers involved.
Sequence of events for Interrupt cycle.
The main micro-operations of the **Execute cycle** and registers involved.
Sequence of events for Execute cycle.
Flowchart for **Instruction Cycle**.
Functional Requirements for the control unit: Define basic elements of processor, Describe micro-operations processor performs, Determine functions control unit must perform.
Basic Elements of Processor: ALU, Registers, Internal data paths, External data paths, Control Unit.
Types of Micro-operation: Transfer data between registers, Transfer data from register to external, Transfer data from external to register, Perform arithmetic or logical ops.
Two tasks the control unit must perform: **Sequencing**, **Execution**.
It generates three types of output control signals: Data paths, ALU, System bus.
The control unit implementation can be done by two methods: **Hardwired implementation**, **Microprogrammed implementation**.
Hardwired implementation: the control unit is a combinational circuit.

Chapter 16

terms:

One line describes a set of micro-operations occurring at one time and is known as **microinstruction**.
A sequence of instructions to control complex operations is known as **microprogram**, or **firmware**.
We can construct a **control word** that represents the control signals that necessary for a specific micro-operation.
Horizontal micro-programming: Each micro-instruction specifies many different micro-operations to be performed in parallel.
Characteristic of **Horizontal micro-programming.**
vertical micro-programming: Each micro-instruction specifies single (or few) micro-operations to be performed.
Characteristic of **vertical micro-programming.**
The control words are put into a special memory block called **control memory**, with

each word having a unique address.
The microinstructions are organized as different routines: The microinstructions in each routine are executed sequentially. Each routine ends with a branch instruction points to the next routine.
The organization of microprogrammed control unit: Word specified in **control address register** is read into **control buffer register**, Control buffer register contents generates control signals and next address information, Sequence login loads new address into control buffer register based on next address information from control buffer register and ALU flags.
The decision of next address: Depending on ALU flags and control buffer register.
Advantages and disadvantages of microprogramming.

Chapter 17

terms:

Multiple processor organization: SISD, SIMD, MISD, MIMD. Different sets of MIMD: SMPs，clusters and NUMA systems. SMP: Share sigle memeory or pool, Share bus to access memory, Memory access time to given area of memory is approximately the same for each processor. Clusters: collection of independent uniprocessors or SMPs.

Chapter 18

terms:

Multicore Organization: Mumber of core processors on chip; Number of levels of cache on chip; Amount of shared cache. SMT: simultaneous multi-threading.