Computer Systems

- Computer Component (Part 1) -

Objectives

- Processor
  - Understand the types of computers and their configurations
  - Understand the architecture, structure, scheme, and operating principles of the processor
  - Understand the indexes for processor performance
  - Understand high-speed and high-reliability technologies for processors
- Memory
  - Understand the types and characteristics of memory
  - Understand the mechanism of main storage including its configuration, memory system configuration, and storage hierarchy
  - Understand the types of storage media and their characteristics

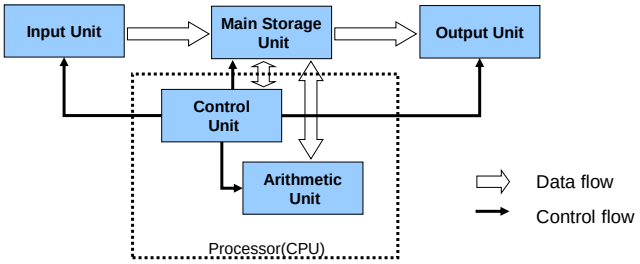
Objectives

- Bus
  - Understand the overview of the types of buses along with their characteristics and configurations.
- Input/output interface
  - Understand the types of typical input/output interfaces and their characteristics
  - Understand the basic roles and functions of device drivers
- Input/output device
  - Understand the types of typical I/O devices along with their characteristics
  - Understand the types of typical auxiliary storage devices along with their characteristics

Introduction

- Computer 5 Main Units

Functions	Unit	Definition
Input	Input	This unit inputs the data and programs for computer processing.
Storage	Storage	This unit stores the input data and programs.
Operation	Arithmetic	This unit conducts calculation and decision on the stored data according to the instructions of the program
Control	Control	This unit controls the input unit, storage unit, arithmetic unit and the output unit.
Output	Output	This unit outputs the results of computer processing in a format that can be understood by humans.



Computer Types

- Personal Computer
  - Computers that were developed for personal use, commonly called PCs for short. These personal computers can be classified as: Desktop type, Laptop type, and Notebook type.



Computer Types

- Workstation
  - Also called as **engineering workstations** (EWS). Compared to PCs, workstations are capable of performing high quality image processing, etc. with high speed. Main applications are:
    - **Research and development fields**: High-speed processing of complex scientific and engineering calculations
    - **Product design/manufacturing fields**: Used in CAD (Computer Aided Design), CAM (Computer Aided Manufacturing), etc. application
    - **Software development field**: Use of CASE tools (Computer Aided Software Engineering) tool, etc.
    - **Communication network field**: Used as client machines or server machines in distributed processing systems.

Computer Types

- General Purpose Computer
  - A computer that is capable of performing both office work as well as scientific and engineering calculations. It is also called mainframe since it is the mainframe of a great number of computers used in an enterprise.
  - Computers that conduct enterprise core business system processing, an automatic ticketing processing, bank services, etc. are all general-purpose computers.

Computer Types

- Supercomputer
  - Computers capable of performing enormous and complex calculations at extremely high speeds
  - Computers that compile computer high-speed technology using forefront semiconductor element technology as well as vector processors that perform floating point operations and vector operations, etc.
  - Among the main purposes are: weather forecast, simulation of nuclear power generation, orbit calculation of artificial satellites

Computer Types

- Microcomputer
  - Small-size computers into which a microprocessor is built. The computers that are embedded into machines, especially household appliances such as washing machines, air conditioners, and AV appliances, in order to control the machine operation. These microcomputers are electronic parts with bare integrated circuits.

Computer Types

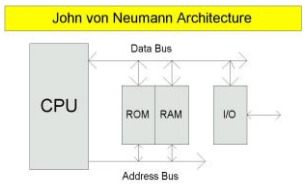
- Process control computer
  - Computers that control different types of machines in steel mills, automobile plants, petroleum refineries, chemical plants, etc. When the supervising computer detects an abnormality, it immediately controls each machine and adjusts the production process.
  - Centralized control and automation have been achieved through the use of process control computers in power system control, general building security systems, highway traffic control, etc.

Processor Architecture

- Processor Structure
  - Handles the control unit and arithmetic units

1) Control Unit

- **Controls all operations** of the computer
- **Retrieves** instruction stored in main storage unit
- **Decodes** retrieved instruction using the instruction decoder
- **Executes** and transmits instructions to each unit.
- *The control unit controls each unit and implements the function of each of the units as a computer system. The system by which instructions are executed in this way, sequentially, is called **sequential control system**, which is based on the concept of John Von Neumann.*



Processor Architecture

- Processor Structure

2) Arithmetic Unit (ALU)

- Performs **calculations, comparison, branch** and other processes.
- Depending on the representation method of data assigned subject to operations, ALU has functions performing fixed point operation, floating point operation and decimal
- Functions of ALU:

Basic operations	Basic instructions
Arithmetic operations	Addition, subtraction, multiplication and division
Logical operations	Logical sum (OR), logical product (AND), negation (NOT)
Comparison	Comparison instruction (size comparison)
Branch	Branch instruction (change of the sequence of instruction execution according to the conditions)

Processor Architecture

- Register is a device used for temporarily storing various data inside the CPU. The main registers are as follows:
  - **Instruction register.** It stores the instructions to be executed. It is composed of an instruction part and an address part.
  - **Instruction address register (program counter, program register).** It stores the address (storing position in the main memory) of the instruction to be executed next.
  - **General register**. It is used for various purposes such as storing the data to be processed.
  - **Accumulator**. It stores the data for performing computations. It is sometimes substituted with a general register.
  - **Base address register.** It stores the beginning address of a program
  - **Index register.** It stores the index for address modification.

Processor Architecture

- **Flag register.** It stores the information of a certain status (whether the results of computations are positive or negative). Depending on the status of this register, the next action, such as branch destination of the condition branch instruction, is decided.
- **PSW (Program Status Word).** It stores the running status of a program (value of program counter, value of flag register, etc.)

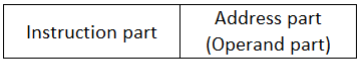
Processor Architecture

- Main types of machine language instructions
  - **Arithmetic operation instruction**. This is an instruction for performing arithmetic operations such as addition and subtraction.
  - **Logical operation instruction**. This is an instruction that performs logical operations such as logical product and logical sum operations.
  - **Transfer instruction**. This is an instruction that transfers data such as load and store.
  - **Comparison instruction**. This is an instruction that compares the magnitude relation between two values.
  - **Branch instruction**. This is an instruction that branches (jumps) the control on the basis of the value of a flag register.
  - **Shift instruction**. This is an instruction that performs shift operations such as arithmetic shift and logical shift.
  - **Input/output instruction**. This is an instruction that reads data from or write data to I/O devices.

Processor Architecture

- The length of instructions may vary, it may include instructions recorded in 1 word (corresponding to 1 address) of main memory (1-word instruction), and instructions recorded consecutively in multiple words (when recorded in 2 words, it is referred to as a 2-word instruction).

- A machine language instruction is composed of:
  - **Instruction part** – where the instruction code specifying the process to be executed is recorded
  - **Address part** (operand part) – for specifying the address to be processed

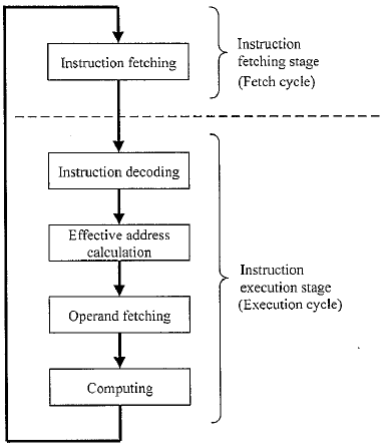


Configuration of machine language instructions

- Based on the number of address parts (operand parts), they are also separately referred to as 0-address instruction, 1-address instruction, 2-address instruction, and 3-address instruction.

Processor Architecture

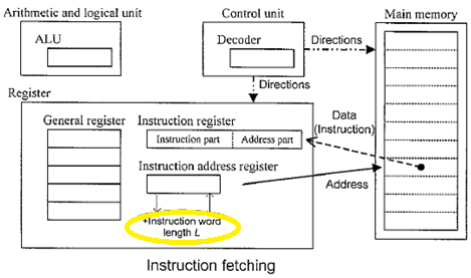
- Execution Sequence of Instructions



Execution sequence of instructions

Processor Architecture

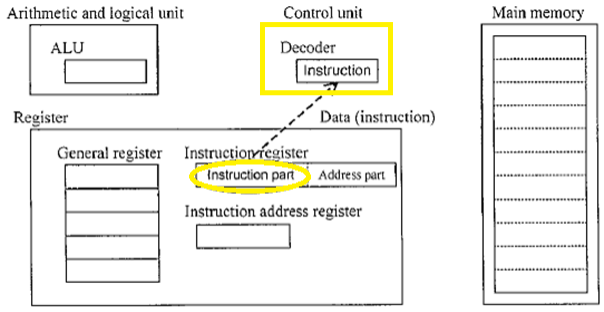
- Processor Operation
  1. **Instruction fetching**,
     - On the basis of the directions given by the control unit, an instruction stored in the address shown by the instruction address register is fetched from main memory, and it is stored in the instruction register.
     - After the instruction is fetched, instruction word length L (number of words where 1 instruction fetched is stored) is added to the instruction address register for fetching the next instruction.



Instruction fetching

Processor Architecture

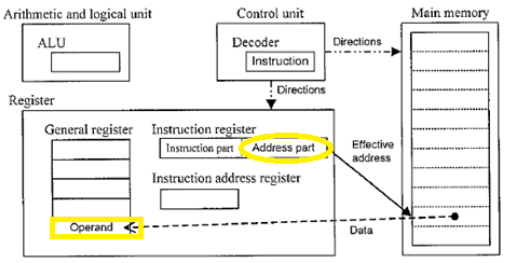
2. **Instruction decoding**
   - The instruction part of the instruction fetched in the instruction register is decoded by the decoder (instruction decoder) of the control unit.



Instruction decoding

Processor Architecture

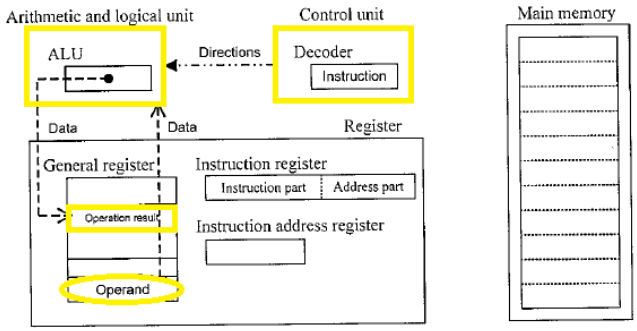
3. **Effective address calculation / operand fetching**
   - In effective address calculation, the storage position (effective address) of data stored in main memory is determined from the address part of the instruction. This is referred to as ***address modification***.
   - In operand fetching, the effective address calculated is sent to main memory, and the value or variable (operand) to be computed such as arithmetic operation instructions is read into the general register. There may not be any operand in some cases, depending on the type of instruction, and this process may be omitted in such cases.



Effective address calculation and operand fetching

Processor Architecture

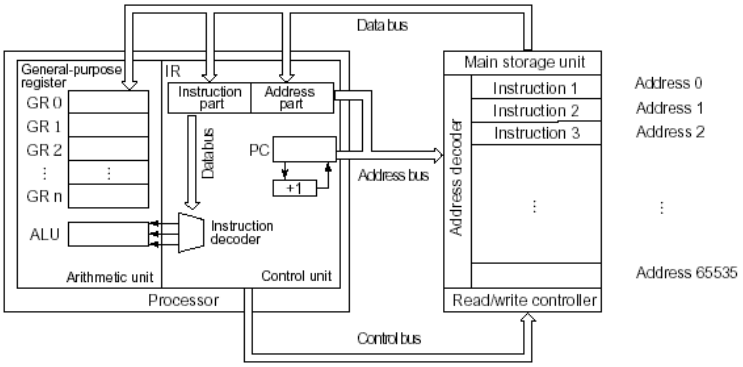
4. **Computation (instruction execution)**
   - The arithmetic and logical unit executes the computation based on the decoded instruction. The operation result is recorded in the general register and written in main memory.



Arithmetic operation (instruction execution)

Processor Architecture

- Hardware structure



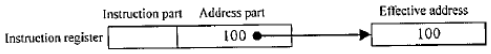
Processor Architecture

- **Addressing mode**
  - Method of determining the effective address from the value recorded in the address part of instruction, and then fetching the operand.
  - Classified into two methods:
    1. Address Modification, the method of determining the effective address from the value of the address part of instruction
    2. Method of determining the operand from the effective address
  - Effective address – the actual address obtained

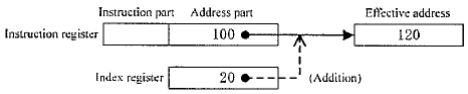
Processor Architecture

Method of determining effective address from the value of the address part (address modification):

1. **Absolute addressing**
   - The value of the address part is used as the effective address as it is

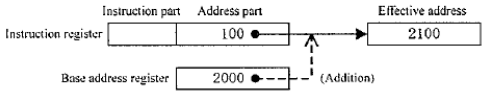


2. **Index addressing**
   - The effective address is determined by adding the value of the index register to the value of the address part. A general register is also used as the index register, and it is necessary to specify which general register to use in such cases.

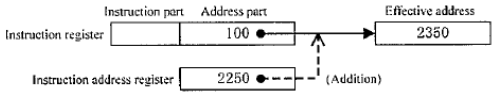


Processor Architecture

3. **Base addressing**
   - The effective address is determined by adding the value of the base address register to the value of the address part. This means the relative position from the beginning of the program (stored in base address register).



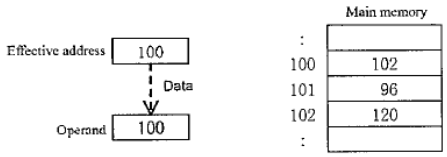
4. **Relative addressing**
   - The effective address is determined by adding the value of the instruction address register (program counter) to the value of the address part. This means the relative position from the instruction being executed (stored in program counter or PC).



Processor Architecture

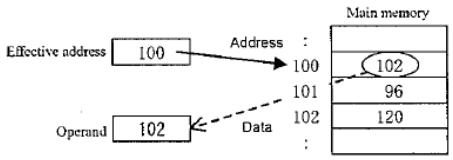
Method of determining operand from the effective address:

1. **Immediate addressing**
   - The effective address is used as it is as operand. Since it does not reference main memory, execution speed is somewhat faster than other addressing.



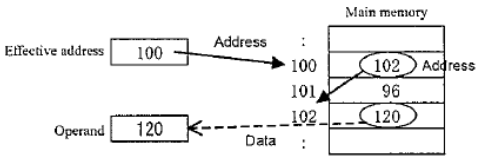
Processor Architecture

2. **Direct addressing**
   - The content (value) of main memory referenced by the effective address is used as operand. Generally, direct addressing where there is no description concerning address modification can be considered as absolute addressing, and it can be considered as direct addressing when only address modification is specified.



Processor Architecture

3. **Indirect addressing**
   - The content (value) of main memory referenced by the effective address is used as the address of operand. Double (in the example figure, the data stored in address 120 is fetched as operand), and triple indirect addressing is also possible.



Interrupt

- Interrupt means executing a separate process (instruction) while a series of processes (instructions) are being executed.
- User mode – mode where there are restrictions on the use of CPU
- Privilege mode – mode where there are no restrictions on the use of CPU
- Processing sequence of processor when interrupt has occurred:
  1. Switch from user mode to privilege mode
  2. Save the values of various registers (program counter, etc.)
  3. Decide the starting address of the interrupt process routine (interrupt program).
  4. Execute the interrupt process routine.
  5. After the execution of the interrupt process routine is complete, restore the values of various registers that are saved in step 2.
  6. Switch from privilege mode to user mode, and restart the interrupted process.

Interrupt

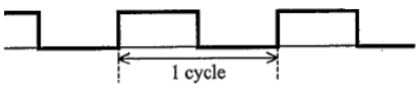
- **Classification**:
  1. **External interrupt**: This is an interrupt that occurs because of a reason not related to the process being executed.
     - **Time interrupt**. This is an interrupt that occurs when the time measured in the timer, such as interval timer and watchdog timer, has exceeded the specified time (i.e. the timer has been timed-out)
     - **Input / Output interrupt** (I/O completion interrupt). This is an interrupt that occurs when an input/output operation has been completed.
     - **Machine check interrupt**. This is an interrupt that occurs because of a hardware malfunction, a power failure, or such other factor.
     - **Restart interrupt**. This is an interrupt that occurs when the user has pressed the external restart switch.

Interrupt

2. **Internal interrupt**: This is an interrupt that occurs because of the process being executed. This is also referred to as a *trap*.
   - **SVC (Supervisor Call) interrupt**. This is an interrupt that occurs when *supervisor* (a program that offers basic functions) is requested to invoke a process, such as when input/output instruction is used for storage protection exception happens.
   - **Program interrupt**. This is an interrupt that occurs by the error of a running program (e.g. divide-by-zero or overflow)

Processor Architecture

- **Clock generator**
  - A device for generating clock signals in order to synchronize and control the timing of operations between various devices inside computers. The speed of generating clock signals is shown with clock frequency, and **MHz** (equivalent to 1 Million cycles in 1 second) is used as the unit.
  - Usually, either the rising or falling edge of a signal is used as synchronization timing, and there is a technique called DDR (Double Data Rate) that uses both.

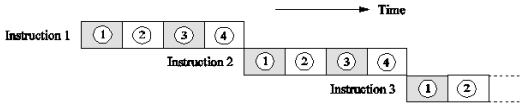


1 cycle of clock frequency

Processor Architecture

- Speed Performance Enhancement in Processor

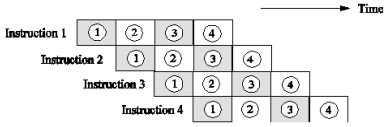
1. **Sequential architecture**
   - The process of reading, decoding and execution of the instructions till it is finished before another instruction is fetched.



   - Program execution steps are:
     - Instruction cycle (1): The instruction is read from the main storage unit
     - Instruction cycle (2): The instruction is decoded and the address calculated
     - Execution cycle (3): The reading and writing of the main storage unit is done
     - Execution cycle (4): The actions specified by the instruction is executed
   - Instruction cycle (1) and execution cycle (3) accesses the main storage unit. Instruction cycle (2) and execution (4) is executed within the processor.

Processor Architecture

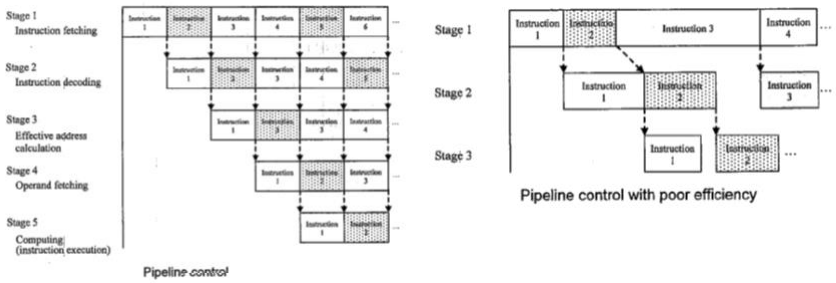
2. **Speculative architecture**
   - Used by pipeline processing
   - Pipeline control / processing divides the execution cycle of each instruction into multiple stages, and by shifting each stage little by little and executing multiple stages separately, multiple instructions are executed simultaneously and in parallel.



   - Figure shows the execution of instructions 1 to 4:
     1. Instruction one's instruction cycle (1) is executed
     2. When instruction one's instruction cycle (2) is executing, instruction two's instruction cycle (1) is executed
     3. When instruction one's execution cycle (3) is executing, instruction two's instruction cycle (2) is executed and instruction three's instruction cycle (1) is executed
     4. When instruction one's execution cycle (4) is executing, instruction two's execution cycle (3) is executed and instruction three's instruction cycle (2) is executed and instruction four's instruction cycle (1) is executed

Processor Architecture

- Efficiency will decline if there are instructions (jump instructions) that change the execution sequence of instruction. Moreover, if the execution time of each instruction is different, the waiting time may be required before starting the execution of a stage, and the efficiency will decline.



- For efficient pipeline control, it is necessary that the execution time of all instructions be almost the same. Among the different architectures of processors, RISC architecture is suitable for this concept, while CISC architecture is not very suitable.

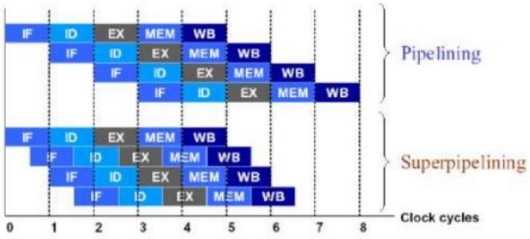
Processor Architecture

Complex Instruction Set Computer (CISC)	Reduced Instruction Set Computer (RISC)
▪ **Variation in the instruction size and length of execution**	▪ **About the same in the instruction size and length of execution**
▪ **Complex, high level type instructions**	▪ **Basic instructions**
▪ **Instructions are executed by the microprogram.** Micro instruction refers to giving control directions to hardware such as various logic circuits and registers.	▪ **Instructions are executed by the hardware (wired logic).** Wired logic control: Instruction is executed by passing signals through a logic circuit having a certain function.

Processor Architecture

- Parallel Method
  - Use multiple processors simultaneously to execute a program
  - Speeds execution
  - Requires special system software

1. **Super pipeline architecture**
   Increases the number of stages (depth of pipeline) and thereby reduces the processing time (pipeline pitch) of one stage in order to aim for performance enhancement.

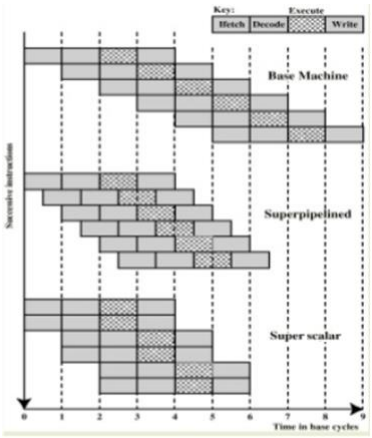


Processor Architecture

2. **Super scalar architecture**
   Extracts multiple instructions that can be executed simultaneously, and executes them in parallel while the arithmetic unit to be used is dynamically decided.

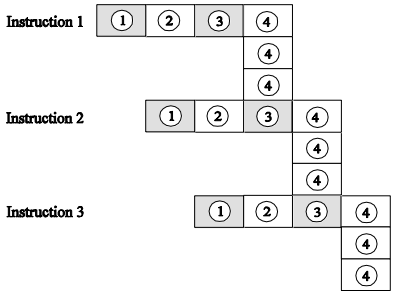
- Superscalar vs. Super pipeline

- Simple pipeline system performs only one pipeline stage per clock cycle
- Super pipeline system is capable of performing two pipeline stages per clock cycle
- Superscalar performs only one pipeline stage per clock cycle in each parallel pipeline



Processor Architecture

3. **VLIW (Very Long Instruction Word) Method**
   - Consolidates multiple instructions that can be executed simultaneously into one instruction at the stage of generating the object program (program translated into machine language) by compiling (translating into machine language) the program, and statically allocates the arithmetic and logical units to be used.
   - Means that multiple instructions (functions) are consolidated into one instruction, and thus the length of instruction increases.



Processor Architecture

- The architecture of the parallel computers is classified into 4 types from the relation between flow of instructions and flow of data:
  - **SISD (Single Instruction stream - Single Data stream)**

    This computer processes one unit of data with one instruction. General computers that do not have parallelized processors correspond to this architecture.
  - **SIMD (Single Instruction stream – Multiple Data stream)**

    This computer processes multiple units of data with one instruction. Vector computers equipped with a vector processor (array processor) that simultaneously computes multiple data in array with one instruction correspond to this architecture.
  - **MISD (Multiple Instruction stream – Single Data stream)**

    This computer processes a single unit of data with multiple instructions. Computers that use this architecture are not used in practice.
  - **MIMD (Multiple Instruction stream – Multiple Data stream)**

    This computer processes multiple units of data with multiple instructions. Multiprocessors where multiple processors are multiplexed correspond to this architecture.

Processor Architecture

- Multi-Processor
  - Designed to improve performance and reliability of the system
  - Multiple processors in parallel with each processor has a dedicated function
  - Fault-tolerance
  - Resource-sharing

  - Symmetric Multi-processor
    Memory is shared among all the processors executing the same OS

  - Array processor
    Used in high-speed scientific computing using pipeline processing

END OF 1ST PART