Lecture1: introduction

Outline:

- *History overview*
- *Central processing unite*
- *Register set*
- *Special purpose address registers*
- *Datapath*
- *Control unit*

1. History overview

Computer systems have conventionally been defined through their interfaces at a number of layered abstraction levels, each providing functional support to its predecessor. Included among the levels are the application programs, the high-level languages, and the set of machine instructions. Based on the interface between different levels of the system, a number of computer architectures can be defined. History overview of computer development is briefly described in table (1).

Table 1: history overview of computer development

GENERATION	FEATURES	MILESTONES	YEAR	NOTES
Early machines (3000BC-1945)	Mech., Electro-mech.	Asia Minor, Abacus	3000BC	Only replaced by paper and pencil
		Blaise Pascal, Pascaline	1642	Decimal addition (8 decimal figs)
		Charles Babbage Differential Engine	1823	Steam powered
		Herman Hollerith, Punch Card	1889	US census, origin of IBM
		Howard Aiken, Harvard Mark I	1937	Ballistic charts of US Navy
First (1945-1956)	Vacuum tubes	Alan Turing, Colossus	1943	Decode German ENIGMA codes
		Eckert, Mauchly, ENIAC	1946	1st general purpose electronic computer
		Von Neumann, EDVAC	1950	Von Neumann architecture
Second (1956-1963)	Transistor (1947)	MIT Lincoln Labs, TXO	1953	1st computer based on transistors
		High level programming languages	1956	FORTRAN (1956), COBOL (1959)
		IBM Stretch, Sperry-Rand LARC	1950s	1st supercomputers, scientific computation
Third (1964-1971)	IC (SSI, MSI)	Seymour Cray, CDC 6600	1964	1st to use parallelism (10 processors)
		IBM SYSTEM 360	1964	Makes other systems obsolete
		DEC PDP-8	1965	1st successful minicomputer
Fourth (1971-present)	Micro-processor LSI, VLSI	Intel 4004	1971	4-bit (1st microprocessor)
		Intel 8008	1972	8/8/14
		Motorola 6800	1974	8/8/16
		Intel 8086	1978	16/16/20
		Motorola 68000	**1979**	**32/16/24**
		Intel 80286	1982	16/16/24
		Motorola 68020	1984	32/32/32
		Intel 80386	1985	32/32/32, pipelining
		Motorola 68030	1987	32/32/32, MMU
		Intel 80486	1989	32/32/32, cache, FPP
		Motorola 68040	1991	32/32/32, FPP
		Motorola Power PC 601 (G1)	1993	32/64/32, RISC, super-scalar
		Intel Pentium	1993	32/64/32, super-scalar
		Motorola 68060	1994	32/32/32, super-scalar
		Motorola Power PC 603 (G2)	1994	32/64/32, portable computing
		Motorola Power PC 604 (G3)	1994	32/64/32, server, workstations
		Intel Pentium Pro	1995	32/64/32 (optimized for 32-bit OS)
		Motorola Power PC 620 (G4)	1996	64/64/32
		Intel Pentium II	1997	32/64/32, MMX

2.Central Processing Unit

Central Processing Unit(CPU) is the main component of any computer system and The primary function is to execute a set of instructions stored in the computer's memory. The CPU fetches instructions from memory, reads and writes data from and to memory, and transfers data from and to input/output devices

A simple CPU consists of a set of registers, an arithmetic logic

unit (ALU), and a control unit (CU). The main component of CPU is shown in figure (1).

- The register set can divide to :
  - ❖ General –purpose set registers which use for any purpose
  - ❖ Special –purpose set registers have specific function within CPU such as program counter(PC),instruction register(IR),segment register, stack pointer ….ect.
- ALU performs the arithmetic, logic and shift operations
- CU is responsible for fetch the instruction to be executed from main memory and decoding and executing it.

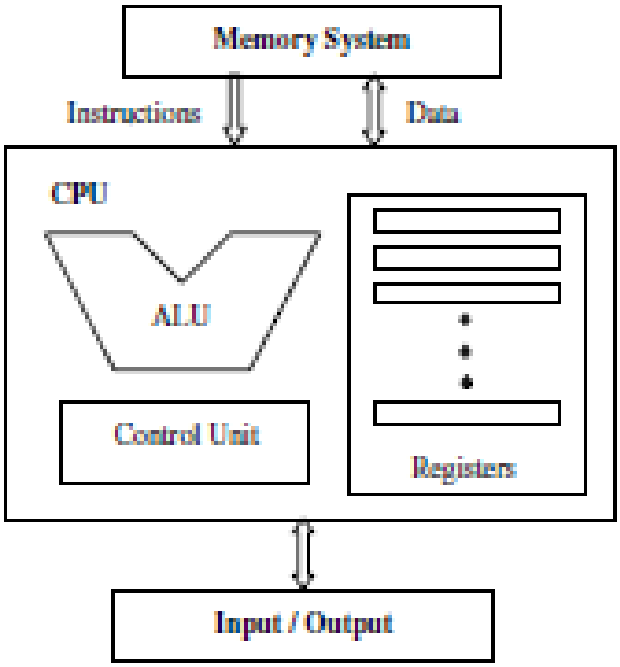

Figure 1: the main component of CPU

3. Register set

Registers are essentially extremely fast memory locations within the CPU that are used to create and store the results of CPU operations and other calculations. Different computers have different register sets. They differ in the number of registers, register types, and the length of each register. They also differ in the usage of each register. General-purpose registers can be used for multiple purposes and assigned to a variety of functions by the programmer. special- purpose registers are restricted to only specific functions. In some cases, some registers are used only to hold data and cannot be used in the calculations of operand addresses. Address registers may be dedicated to a particular addressing mode or may be used as address general purpose. Address registers must be long enough to hold the largest address.

Memory Access Registers :Two registers are essential in memory write and read operations: the memory data register (MDR) and memory address register (MAR). The MDR and MAR are used exclusively by the CPU and are not directly accessible to programmers.

In order to perform a write operation into a specified memory location, the MDR and MAR are used as follows:

1. The word to be stored into the memory location is first loaded by the CPU into MDR.
2. . The address of the location into which the word is to be stored is loaded by the CPU into a MAR.
3. A write signal is issued by the CPU.

Similarly, to perform a memory read operation, the MDR and MAR are used as follows:

1. The address of the location from which the word is to be read is loaded into the MAR.
2. A read signal is issued by the CPU.
3. The required word will be loaded by the memory into the MDR ready for use by the CPU.

Instruction Fetching Registers: Two main registers are involved in fetching an instruction for execution: the program counter (PC) and the instruction register (IR). The PC is the register that contains the address

of the next instruction to be fetched. The fetched instruction is loaded in the IR for execution. After a successful instruction fetch, the PC is updated to point to the next instruction to be executed. In the case of a branch operation, the PC is updated to point to the branch target instruction after the branch is resolved, that is, the target address is known.

Condition Registers: Condition registers, or flags, are used to maintain status information. Some architectures contain a special program status word (PSW) register. The PSW contains bits that are set by the CPU to indicate the current status of an executing program. These indicators are typically for arithmetic operations, interrupts, memory protection information, or processor status.

4.Special-Purpose Address Registers:

Index Register , in index addressing, the address of the operand is obtained by adding a constant to the content of a register,. The index register holds an address displacement.

Segment Pointers , in order to support segmentation, the address issued by the processor should consist of a segment number (base) and a displacement (or an offset) within the segment. A segment register holds the address of the base of the segment.

Stack Pointer is used to indicate the stack location that can be addressed. In the stack push operation, the SP value is used to indicate the location (called the top of the stack). After storing (pushing) this value, the SP is incremented .

5.datapath

The CPU can be divided into a data section and a control section. The data section, which is also called the datapath, contains the registers and the ALU. The datapath is capable of performing certain operations on data items. The control section is basically the control unit, which issues control signals to the datapath. Internal to the CPU, data move from one register to another and between ALU and registers.

Internal data movements are performed via local buses, which may carry data, instructions, and addresses. Externally, data move from registers to

memory and I/O devices, often by means of a system bus. Internal data movement among registers and between the ALU and registers may be carried out using different organizations including one-bus, two-bus, or three-bus organizations. Dedicated datapaths may also be used between components that transfer data between themselves.

5.3.1. One-Bus Organization

Using one bus, the CPU registers and the ALU use a single bus to move outgoing and incoming data. Since a bus can handle only a single data movement within one clock cycle, two-operand operations will need two cycles to fetch the operands for the ALU. Additional registers may also be needed to buffer data for the ALU.

5.3.2. Two-Bus Organization

Using two buses is a faster solution than the one-bus organization. In this case, general- purpose registers are connected to both buses. Data can be transferred from two different registers to the input point of the ALU at the same time. Therefore, a two operand operation can fetch both operands in the same clock cycle. An additional buffer register may be needed to hold the output of the ALU when the two buses are busy carrying the two operands.

5.3.3. Three-Bus Organization

In a three-bus organization, two buses may be used as source buses while the third is used as destination. The source buses move data out of registers (out-bus), and the destination bus may move data into a register (in-bus). Each of the two out-buses is connected to an ALU input point. The output of the ALU is connected directly to the in-bus.

6.control unit

The control unit is the main component that directs the system operations by sending control signals to the datapath. These signals control the flow of data within the CPU and between the CPU and external units such as memory and I/O. Control buses generally carry signals between the control unit and other computer components in a clock-driven manner

There are mainly two different types of control units: microprogrammed and hardwired. In microprogrammed control, the control signals associated with operations are stored in special memory units inaccessible by the programmer as control words. A control word is a microinstruction that specifies one or more microoperations. A sequence of microinstructions is called a microprogram, which is stored in a ROM or RAM called a control memory CM.

In hardwired control, fixed logic circuits that correspond directly to the Boolean expressions are used to generate the control signals. Clearly hardwired control is faster than microprogrammed control. However, hardwired control could be very expensive and complicated for complex systems. Hardwired control is more economical for small control units. It should also be noted that microprogrammed control could adapt easily to changes in the system design. We can easily add new instructions

without changing hardware. Hardwired control will require a redesign of the entire systems in the case of any change.