Microprocessors and Micro controllers
CSE3815

Fairuz Bilquis Khan

April 2024

Evolution of Microprocessors

There should be overview chart of difference between intel 4004 to pentium 4.

Microprocessors:

```
‘-->Is a semiconductor device consisting of electronics logic circuit.
‘--->Manufactured by using various fabrication schemes.
‘---->Capable of performing computing functions.
‘----->Capable of transporting data/information.

Can be divided into 3 segments:
    (-)Arithmetic and logic unit
    (-)Registers unit
    (-)Control unit
```

Von Neumann Machine:

```
Von Neumann machine may refer to: Von Neumann architecture, a conceptual model
of nearly all computer architecture

Three key concepts:
(-)Data and instructions are stored in a single set of read-write memory.
(-)Contents of memory are accessed by memory address without regard to the type
    of data.
(-)Execution occurs in a sequential fashion, unless explicitly stated otherwise.
```

Computer System Components:

```
(-) Memory:
        Stores data and instructions.
(-) Input/Output:
        Called peripherals.
        Used to input and output data and instructions.
(-) Arithmetic and Logic Unit:
        Performs arithmetic(+, -)  operations.
        Performs logical(AND, OR, XOR, Shift, Rotate) operations.
(-) Control Unit:
        Coordinates the execution of instructions by managing the flow of data
            between the CPU's components.
(-) System Interconnection and Interaction:
        (+)BUS: A group of lines used to transfer bits between µp and other components.
                Bus is used to communicate between parts of the computer. There is only
                one transmitter at a time and only the addressed device can respond.
                ^\types:
                    >> address
                    >> data
                    >> control
```

CPU Components:

(-) Registers:
    Small, fast storage in the CPU for data, instructions and other items.
    Includes the program counter (PC) and memory address registers (MAR).
    These registers need to match the width of the address bus for proper memory access.
    Data registers should match memory word size for efficient data transfer.

(-) Control Unit:
    Generates control signals which are necessary for execution of instructions.
    Connect registers to the bus.
    Controls the data flow between CPU and peripheral components.
    Provides status, control and timing signals required for the operations of memory and I/O devices.
    Acts as a brain of computer system.
    All actions of the control unit are associated with the decoding and execution of instructions(fetch and execute cycles).

(-) Arithmetic and Logic Unit:
    Execute arithmetic and logical operations.
    Accumulator is a special 8-bit register associated with ALU. Register 'A' in 8085 is an accumulator.
    Source of one of the operands of an arithmetic or logical operation serves as one input to ALU.
    Final result of an arithmetic or logical operation is placed in the accumulator.

Arithmetic and Logic Unit:

ALU performs the following arithmetic and logical operations:
- >> Addition
- >> Subtraction
- >> AND
- >> OR
- >> XOR
- >> NOT
- >> Increment
- >> Decrement
- >> Left Shift, Right Shift, Rotate .
- >> Clear etc.

```
Status Flag:
    Intel 8085 microprocessor contains five flip-flops to serve as status flags.
    These flip-flops are set or reset according to the conditions which arise due
        to an arithmetic & logical operation.

        Sign Flag (S): Indicates the sign of the result after an arithmetic operation.
        Zero Flag (Z): Indicates whether the result of an operation is zero.
        Auxiliary Carry Flag (AC): Indicates use for decimal arithmetic operations.
        Parity Flag (P): Indicates the parity of the result.
        Carry Flag (CY): Indicates whether there is a carry-out from the most
                            significant bit.
```

next is difference between 8085 and 8086 architecture diagram (will add later . not needed for Ct)

The Microprocessor and its Architecture

Internal Microprocessor Architecture:

Program or instructions are written based of microprocessor's internal configuration.
In multiple core processors, each core contains the same programming model.
Each core runs as a separate task or thread simultaneously.
A thread consists of
    >> a program counter
    >> a register set, and
    >> a stack space
A task shares with peer threads its code section, data section and operating system resources.

The programming Models:

8086 through Core2 registers are considered program-visible. While other registers are considered program-invisible.
    >> program-visible registers are used during programming while are directly specified by the instructions.
    >> program-invisible registers are not addressable directly during applications programming.
80286 and above contain program-invisible registers to control and operate protected memory and other features of the microprocessor.
80386 through Core2 microprocessors contain full 32-bit internal architecture.
8086 through 80286 are fully upward compatible to the 80386 through Core2.

Intel Microprocessor Evolution:

4004 → 8008 → 8080 → 8085 → 8086/8088 → 80286 → 80386 → 80486 → Pentium → Pentium Pro → Pentium II → Pentium III → Pentium 4 → Core → Core 2

Multipurpose Registers:

RAX is a versatile register with different sizes: 64-bit as RAX, 32-bit as REAX (accumulator), 16-bit as AX, or split into two 8-bit registers, AH and AL.
EAX, known as the accumulator, handles math operations like multiplication and division.
Intel aims to extend the address bus to 52 bits, allowing access to 4 petabytes (4P) of memory.

Address Space(Main Memory: RAM):

- With a 16-bit address bus, it can access 64 kilobytes of memory.
- Expanding to a 20-bit address bus allows access to 1 megabyte.
- A 32-bit address bus extends access to 4 gigabytes.
- Progressing to a 34-bit address bus increases the capacity to 16 gigabytes.
- A 36-bit address bus reaches 64 gigabytes.
- Expanding further to a 38-bit address bus accesses 256 gigabytes.
- Intel's plan for a 52-bit address bus would enable access to a massive 1015 bytes of memory.
- **RBX:** Addressable as RBX, EBX, BX, BH, BL.
  - RBX register (base index) sometimes holds the offset address of a location in the memory system in all versions of the microprocessor.
- **RCX:** Addressable as RCX, ECX, CX, CH, or CL.
  - RCX, a (count) general-purpose register, also holds the count for various instructions.
- **RDX:** Addressable as RDX, EDX, DX, DH, or DL.
  - RDX, a (data) general-purpose register, holds a part of the result from a multiplication or part of the dividend before a division.
- **RBP:** Addressable as RBP, EBP, or BP.
  - RBP points to a memory (base pointer) location for memory data transfers.
- **RDI:** Addressable as RDI, EDI, or DI.
  - RDI often addresses the destination index string destination data for the string instructions.
- **RSI:** Addressable as RSI, ESI, or SI.
  - RSI, the source index register, addresses source string data for the string instructions. Like RDI, RSI also functions as a general-purpose register.
- **R8 - R15:** Found in the Pentium 4 and Core2 if 64-bit extensions are enabled.
  - Data are addressed as 64-, 32-, 16-, or 8-bit sizes and are of general-purpose.

Special-Purpose Registers:

- **RIP:** Addressable as RIP.
  - RIP addresses the next instruction in a section of memory, defined as the instruction pointer in a code segment.
- **RSP:** Addressable as RSP.
  - RSP addresses an area of memory called the stack. The stack pointer stores data through this pointer.
- **RFLAGS:**
  - RFLAGS indicate the condition of the microprocessor and control its operation. Flags are upward-compatible from the 8086/8088 through Core2. The rightmost five and the overflow flag are changed by most arithmetic and logic operations, although data transfers do not affect them. Flags never change for any data transfer or program control operation. Some of the flags are also used to control features found in the microprocessor.
- **Segment Registers:** Include CS, DS, ES, SS, FS, and GS.

List of Each Flag bit, with a brief description of function:

- **C (Carry):** Holds the carry after addition or borrow after subtraction. Also indicates error conditions.
- **P (Parity):** Is the count of ones in a number expressed as even or odd. Logic 0 for odd parity; logic 1 for even parity. If a number contains three binary one bits, it has odd parity; If a number contains no one bits, it has even parity.
- **A (Auxiliary Carry):** Holds the carry (half-carry) after addition or the borrow after subtraction between bit positions 3 and 4 of the result.
- **Z (Zero):** Shows that the result of an arithmetic or logic operation is zero.
- **S (Sign):** Flag holds the arithmetic sign of the result after an arithmetic or logic instruction executes.
- **T (Trap):** The trap flag enables trapping through an on-chip debugging feature.
- **I (Interrupt):** Controls operation of the INTR (interrupt request) input pin.
- **D (Direction):** Selects increment or decrement mode for the DI and/or SI registers.
- **O (Overflow):** Occurs when signed numbers are added or subtracted. An overflow indicates the result has exceeded the capacity of the machine.
- **IOPL (Input/Output Privilege Level):** Used in protected mode operation to select the privilege level for I/O devices.
- **NT (Nested Task):** Flag indicates the current task is nested within another task in protected mode operation.

- **RF (Resume):** Used with debugging to control resumption of execution after the next instruction.
- **VM (Virtual Mode):** Flag bit selects virtual mode operation in a protected mode system.
- **AC (Alignment Check):** Flag bit activates if a word or doubleword is addressed on a non-word or non-doubleword boundary.
- **VIF (Virtual Interrupt):** Is a copy of the interrupt flag bit available to the Pentium 4–(virtual interrupt).
- **VIP (Virtual Interrupt Pending):** Provides information about a virtual mode interrupt for Pentium. Used in multitasking environments to provide virtual interrupt flags.
- **ID (Identification):** Flag indicates that the Pentium microprocessors support the CPUID instruction. CPUID instruction provides the system with information about the Pentium microprocessor such as its version number and manufacturer.

Memory Addresses and Segment Registers

Generate memory addresses when combined with other registers in the microprocessor.

Segment Registers:

Four or six segment registers in various versions of the microprocessor. A segment register functions differently in real mode than in protected mode. Following is a list of each segment register, along with its function in the system.

- **CS (Code):** Holds code (programs and procedures) used by the microprocessor.
- **DS (Data):** Contains most data used by a program. Data are accessed by an offset address or contents of other registers that hold the offset address.
- **ES (Extra):** An additional data segment used by some instructions to hold destination data.
- **SS (Stack):** Defines the area of memory used for the stack. The stack entry point is determined by the stack segment and stack pointer registers. The BP register also addresses data within the stack segment.
- **FS and GS Segments:** Supplemental segment registers available in 80386–Core2 microprocessors. They allow two additional memory segments for access by programs. Windows uses these segments for internal operations, but no definition of their usage is available.

Real Mode Memory Addressing

80286 and above operate in either the real or protected mode. Real mode operation allows addressing of only the first 1M byte of memory space—even in Pentium 4 or Core2 microprocessor. The first 1M byte of memory is called the real memory, conventional memory, or DOS memory system.

Segments and Offsets:

- All real mode memory addresses must consist of a segment address plus an offset address.
- The segment address defines the beginning address of any 64K-byte memory segment.
- The offset address selects any location within the 64K byte memory segment.

Memory address = (Segment Address * 10) + Offset

```
Once the beginning address is known, the ending address is found by adding FFFFH because
    a real mode segment of memory is 64K in length. The offset address is always added
    to the segment starting address to locate the data. Segment and offset addresses
    are sometimes written as 1000:2000.

Suppose a segment address of 1000H and an offset of 246FH.
Now find the memory location:
```

$$\begin{aligned}\text{Memory location} &= (\text{Segment Address} \times 10) + \text{Offset} \\ &= (1000 \times 10)_{\text{H}} + 246F_{\text{H}} \\ &= 10000_{\text{H}} + 246F_{\text{H}} \\ &= 1246F_{\text{H}}\end{aligned}$$