Organization of the IBM Personal Computers

Outline

- The Intel 8086 family of microprocessors
- Organization of 8086 microprocessors
- Organization of the PC

Intel 8086 Family

- 8086 and 8088 Microprocessors
  - 8086 is the first 16-bit microprocessor
  - 8086 has 16-bit data bus where 8088 has 8-bit data bus
  - 8086 has faster clock rate than 8088
  - Less expensive to build a original computer around 8088
- 80286 Microprocessor
  - 16-bit microprocessor
  - Faster than 8086
  - Can operate in real address mode and protected virtual address mode
  - In protected mode, can address 16MB of physical memory
  - Can treat external storage as physical memory

Intel 8086 Family

- 80386 Microprocessors
  - 80386 is the first 32-bit microprocessor
  - 8086 has 32-bit data bus
  - High clock rate
  - Execute instructions in fewer clock cycles
  - Can operate in real and protected mode
  - In protected mode can address 4GB of physical memory and 64TB of virtual memory
- 80486 Microprocessor
  - 32-bit microprocessor
  - Fastest and more powerful processor of the family
  - Has numeric processor, cache memory and more advanced design
  - Three times faster than 80386 running at same clock speed

Organization of the 8086/8088 Microprocessors

- Registers
- Data Registers
- Segment Registers
- Pointer and Index Registers
- Instruction Pointer
- FLAGS Register

Registers

- Information inside the microprocessor is stored in registers
- Classified according to the functions they perform
- Data registers hold data for an operation. 8086 has four general data registers
- Address registers hold the address of an instruction or data. Address registers divided into segment, pointer and index registers in 8086
- Status register keeps the current status of the processor. In 8086 status register is called FLAGS register
- Fourteen 16-bit registers in 8086

Registers

AX	AH	AL	Data Registers
BX	BH	BL	
CX	CH	CL	
DX	DH	DL	
CS			Segment Registers
DS			
SS			
ES			
SI			Pointer and Index Registers
DI			
SP			
BP			
IP			
			FLAGS Registers

Data Registers

- Accumulator Register(AX)
  - High byte of AX is called AH and low byte of AX is called AL
  - Used in arithmetic , logic and data transfer instructions
  - Generates shortest machine codes
  - In multiplication and division, one of the numbers involved must be in AX or AL
  - Input and output operations also required AX and AL
- Base Register(BX)
  - High byte of BX is called BH and low byte of BX is called BL
  - Serves as an address register

Data Registers

- Count Register(CX)
  - High byte of CX is called CH and low byte of CX is called CL
  - CX serves as a loop counter
  - CL is used as a count in instructions that shift and rotate bits
- Data Register(DX)
  - High byte of DX is called DH and low byte of DX is called DL
  - Used in multiplication and division
  - Also used in I/O operation

Memory Segment

- 8086 is a 16-bit processor using 20-bit address
- Addresses are too big to fit in a 16 bit register or memory word
- 8086 gets around this problem by partitioning its memory into segments
- Segment Number
  - Memory segment is a block of $2^{16} = 64\text{K}$ consecutive memory bytes identified by a segment number starting with 0
  - Segment number is 16 bit so the highest segment number is FFFFh
- Offset
  - Within a segment a memory location is specified by giving an offset
  - This is the number of bytes from the beginning of segment
  - With 64-bit segment the offset can be given as a 16-bit number
  - The first byte in a segment is offset 0 and the last offset in a segment is FFFFh

Physical Address

- A memory location may be specified by providing a segment number and an offset
- Written in the form segment: offset. This form is known as logical address
- A4FB:4872h means offset 4872h within segment A4FBh
- For obtaining 20 bit physical address, 8086 first shifts the segment address four bits to the left and then adds the offset
- The physical address for A4FB:4872 is A9822h

Segment Registers

- Code Segment(CS)
  - Points at the segment containing the current program.
- Data Segment(DS)
  - Generally points at segment where variables are defined. By default BX, SI and DI registers work with DS segment register
- Stack Segment(SS)
  - Points at the segment containing the stack. BP and SP work with SS segment register
- Extra Segment(ES)
  - Extra segment register, it's up to a coder to define its usage.

Segment Registers

- It is possible to store any data in the segment registers but this is not a good idea.
- Segment registers have a very special purpose, pointing at accessible blocks of memory.
- Segment registers work together with general purpose register to access any memory value.
- For example if we would like to access memory at the physical address 12345h (hexadecimal), we should set the DS =1230h and SI = 0045h.
- The address formed with two registers is called an effective address.
- Although BX can form an effective address, BH and BL cannot.

Pointer and Index Registers

- Stack Pointer(SP)
  - Used with SS for accessing the stack segment
- Base Pointer(BP)
  - Used primarily to access data on the stack
  - Access data in other segments
- Source Index(SI)
  - Used to point to memory locations in the data segment addressed by DS
  - Incrementation of SI give easy access to consecutive memory locations
- Destination index(DI)
  - Same function as SI
  - String operations use DI to access memory locations addressed by ES

Instruction Pointer

- It is used to access instructions
- CS contains the segment number of the next instruction and IP contains the offset
- IP is updated each time an instruction is executed
- An instruction can not contain IP as an operand

FLAGS Register

- Indicates the status of microprocessor
- Status flags reflect the result of an instruction executed by the processor
  - For example if an arithmetic operation produce a zero value as a result then the zero flag(ZF) is set to 1
- Control flags enable or disable certain operations of the processor
  - For example if interrupt flag(IF) is set to zero, inputs from the keyboard are ignored by the processor

Organization of the PC

- Operating System
- Memory Organization
- I/O Port Addresses
- Start-up Operation