Lecture 5

Kindly go through by yourself

Intel 8088 instructions

- Two types
  - Arithmetic instructions
  - Control transfer instructions

Arithmetic instructions – 1/7

- Operands can be in one of four 16 bit registers, or in memory location designated by one of the addressing modes

- Three instruction formats supported
  - Register/Memory to Register
  - Immediate to Register/Memory
  - Immediate to Accumulator

Arithmetic instructions – 2/7

- The ***mod*** and ***r/m*** fields specify **first operand**, which can be in a **register or memory**
- The ***reg*** field describes the **second operand** which is a **register**
- **Instruction opcode** indicates which **instructions format** is applicable
- ***Direction field (d)*** in instruction indicates **which operand is destination**
- If **d=0**, **register/memory operand is the destination**, else register operand indicated by *reg* is destination
- The ***width field (w)*** indicates whether **8 or 16 bit** arithmetic is to be used

Arithmetic instructions – 3/7

(a) Register/Memory to Register

(b) Immediate to Register/Memory

(c) Immediate to Accumulator

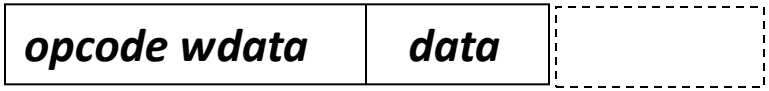

Fig: Instruction formats of Intel 8088

Arithmetic instructions – 4/7

r/m	mod=00	mod=01	mod=10	mod=11	
				w=0	w=1
000	(BX)+(SI)	(BX)+(SI)+d8	Note 2	AL	AX
001	(BX)+(DI)	(BX)+(DI)+d8	Note 2	CL	CX
010	(BP)+(SI)	(BP)+(SI)+d8	Note 2	DL	DX
011	(BP)+(DI)	(BP)+(DI)+d8	Note 2	BL	BX
100	(SI)	(SI)+d8	Note 2	AH	SP
101	(DI)	(DI)+d8	Note 2	CH	BP
110	Note 1	(BP)+d8	Note 2	DH	SI
111	(BX)	(BX)+d8	Note 2	BH	DI

Note 1 : (BP) + DISP for indirect addressing, d16 for direct
Note 2 : Same as previous column, except d16 instead of d8

Fig: Instruction formats of Intel 8088

Arithmetic instructions – 5/7

reg	Register	
	8-bit (w=0)	16-bit (w=1)
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

Fig: Instruction formats of Intel 8088

Arithmetic instructions – 6/7

Assembly statement	*Opcode d w*	*mod reg r/m*	*data/ displacement*
ADD AL, BL	000000 0 0	11 000 011	
ADD AL, 12H[SI]	000000 1 0	01 000 100	0001 0010
ADD AX, 3456H	100000 0 1	11 000 000	0101 0110 0011 0100

Fig: Sample instructions of 8088

Arithmetic instructions – 7/7

- First statement AL could be encoded into first or second operand of instruction
- In second statement, it has to be encoded into second operand since first operand has to be 12H [SI]
- Here, mod=01 since only one byte of displacement is adequate
- Third instruction contains 16 bits of immediate data
- Here, w=1 implies that the accumulator is AX register

Segment overrides – 1/2

- For arithmetic and MOV instructions, uses data segment by default
- To override this, instruction can be preceded by a 1-byte segment override prefix with the format

001	seg	110

Where seg represented in 2 bits, has the meaning below

seg	***segment register***
00	**ES**
01	**CS**
10	**SS**
11	**DS**

Fig: Segment codes

Segment overrides – 2/2

- Example : If the code segment is to be used instead of data segment in the second statement, it can be rewritten as

  ADD AL, CS:12H[SI]

- The assembler would encode this as

segment override *instruction*

001 **<u>01</u>** 110 000000 1 0 01 000 100

00010010

Control transfer instructions – 1/3

- Two groups
  - Calls, jumps and returns
  - Iteration control instructions

- Calls, jumps and returns can occur within the same segment, or can cross segment boundaries

Control transfer instructions – 2/3

- **Inter-segment indicate a new segment base and offset**
- Control transfers can be direct and indirect
- A call pushes the offset of next instruction's address from the segment base on the stack
- This address is used to return to the calling program
- For inter-segment call, CS register is pushed first, followed by offset
- Iteration control operations perform looping decisions in string operations

Control transfer instructions – 3/3

(a) Intrasegment

Opcode	Disp. Low	Disp. high

(b) Intersegment

Opcode	Offset	Offset	Segment	base

(c) Indirect

Opcode	mod 100 r/m	Disp. Low	Disp. high

Fig: Formats of control transfer instructions