Discussion on MCS012 - Block 3: The Central Processing Unit

Presented by: Mr Akshay Kumar

Associate Professor, SOCIS, IGNOU

For

MCA/BCA /PGDCA Students of IGNOU

Reference Material of IGNOU

Block 3 of MCS-012

Facebook Live Sessions on MCS012

1. The Basic Computer and Fixed Point Numbers
2. Floating Point Number representation and Error Detection Codes
3. Combinational Circuits
4. Sequential Circuits
5. Memory Organisation
6. Cache Mapping and I/O Organisation
7. Assembly Language Programming for 8086 Microprocessor
8. Discussion on 8086 Assembly Language Programs

Outline of Presentation

- The Instruction Set of a Computer
- Micro-operations
- ALU Organization

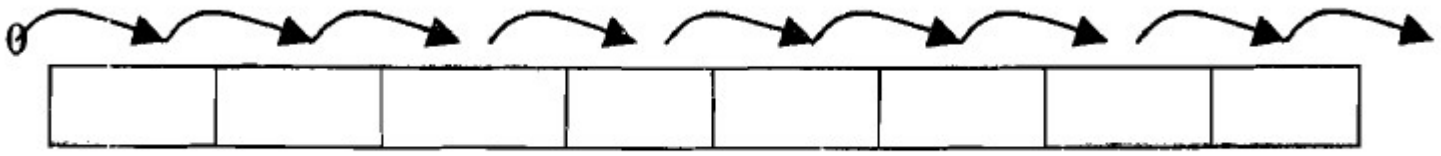
The Instruction Set

- Set of all the Instructions, which processor can execute
- Components of an Instruction:
  - Operation Code
  - Addressing modes
  - Operands
- Number of operands in an Instruction
  - Changes the program size, e.g.
    - ADD Z← X + Y or ADD Z, X, Y
    - LOAD X //Assume the first Operand is AX then AX← X
    - ADD Y // AX← X+Y
    - STORE Z // Z← AX

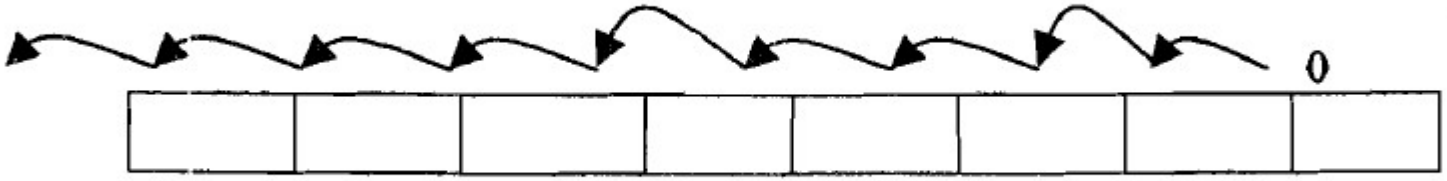
Issues relating to Instruction set

- Instruction formats and Instruction Length – bits allocated to each components
- How many number of operands
- Different data types supported by machine e.g. Numbers, Character Set, Logical bits
- Different types of instructions
  - Data Transfer: MOV, XCHG, XLAT, PUSH, POP etc.
  - Data Processing: Arithmetic - ADD, ADC, INC, MUL, AAA, DAA...Logical: NOT, AND, OR, XOR, SHL, SHR, TEST etc.
  - Program Control: CALL, RETURN, JMP, LOOP
  - Miscellaneous: String -REP, MOVS, Processing Control: STC (Set Carry bit)

Shift Operation

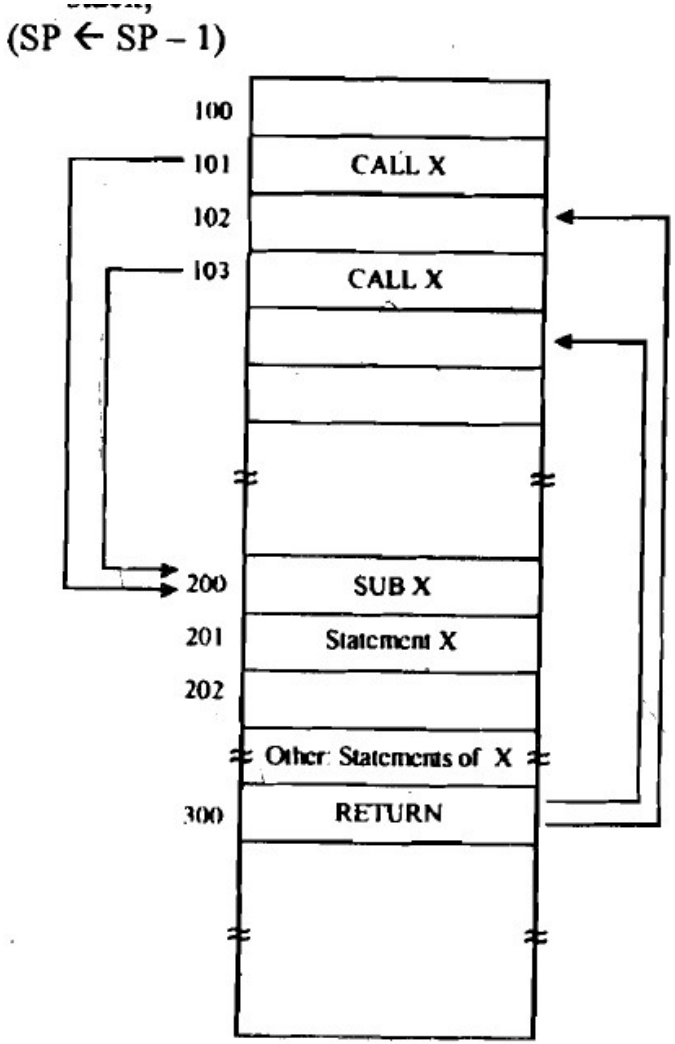


Logical Shift Right

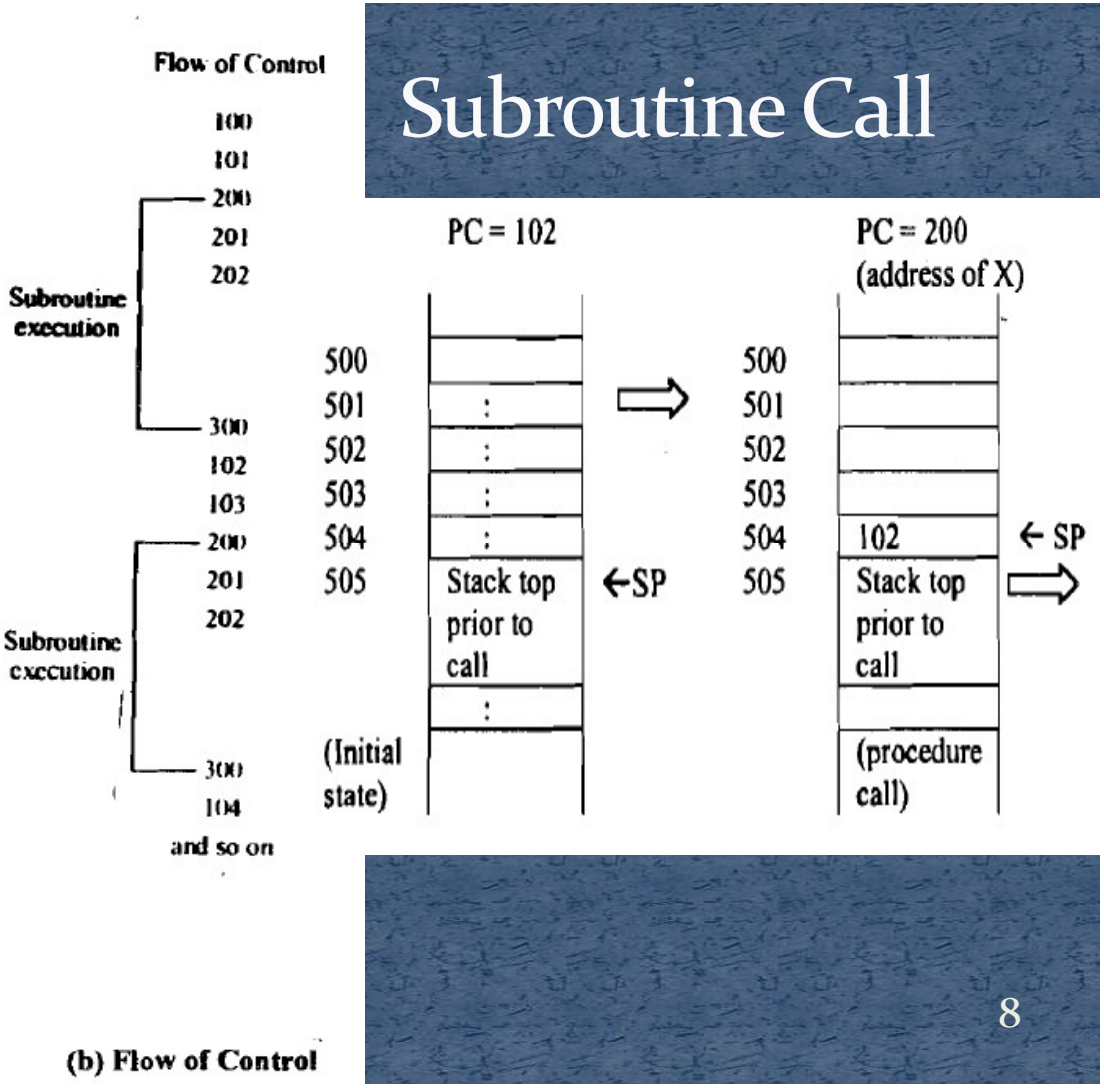


Logical Shift Left

Subroutine Call

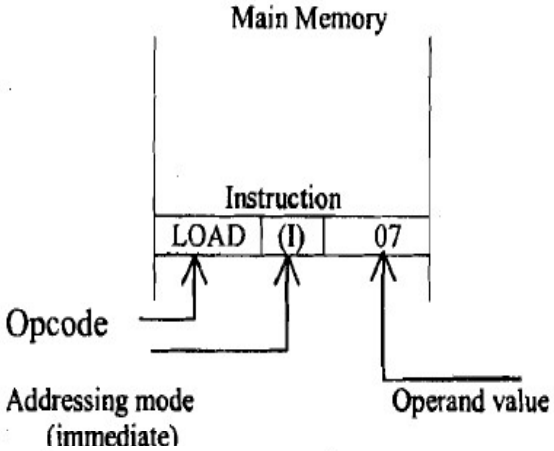


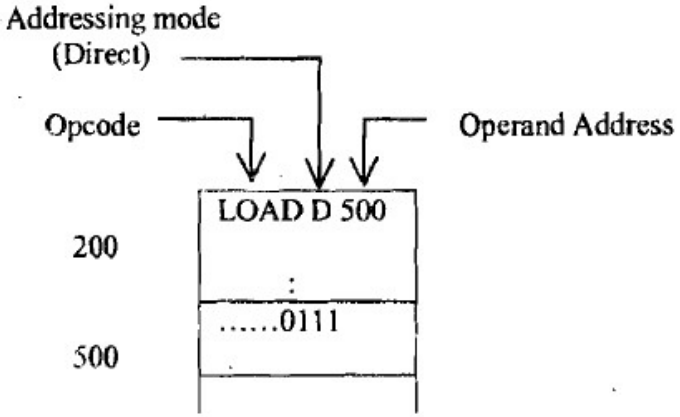
(a) Program in the Memory

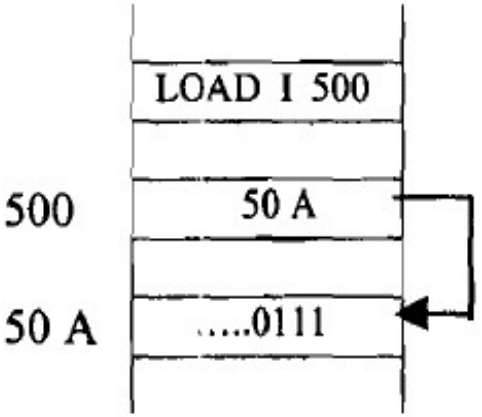


(b) Flow of Control

Addressing Scheme







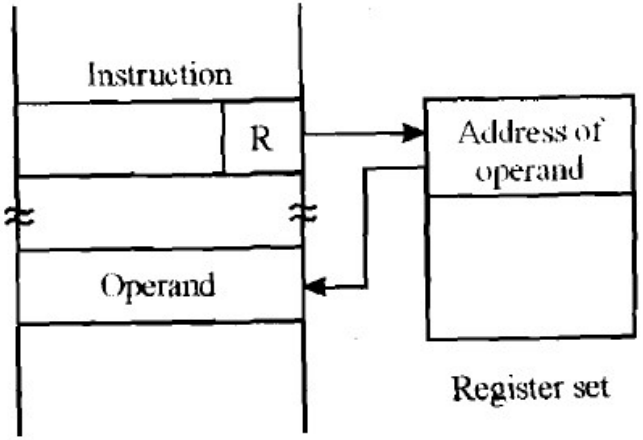


Figure 12: Register Indirect Addressing

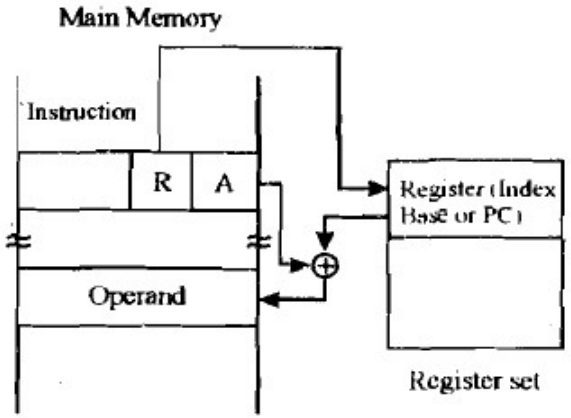


Figure 13: For Displacement Addressing

The ALU Structure and Registers

- The Units of ALU like Bus Interface Unit and Execution Units of 8086
- The Register Organisation
  - General Purpose Registers
    - General Purpose Registers: AX (AH+AL), BX, CX, DX
- Special Purpose Registers
  - Segment Registers : CS, DS, SS, ES
  - Pointer and Index Registers: BP, SI, DI
  - Instruction Pointer (IP), Stack Pointer(SP)
- Flag Register : Carry Flag, Parity Flag, Zero Flag, Overflow flag, Sign Flag

Micro-operations

- Instruction Fetch
- MAR ← PC
- DR ← [MAR],
  PC ← PC + 1
- IR← DR
- Instruction Decode by CU
- Direct Address:
  - IR (ADDRESS) and DR(Address)
- Indirect Address
  - MAR ← DR (Address)
  - DR ← [MAR]
  - IR (Address) ← DR(Address)

Execution of Increment A and skip the next instruction of result is zero

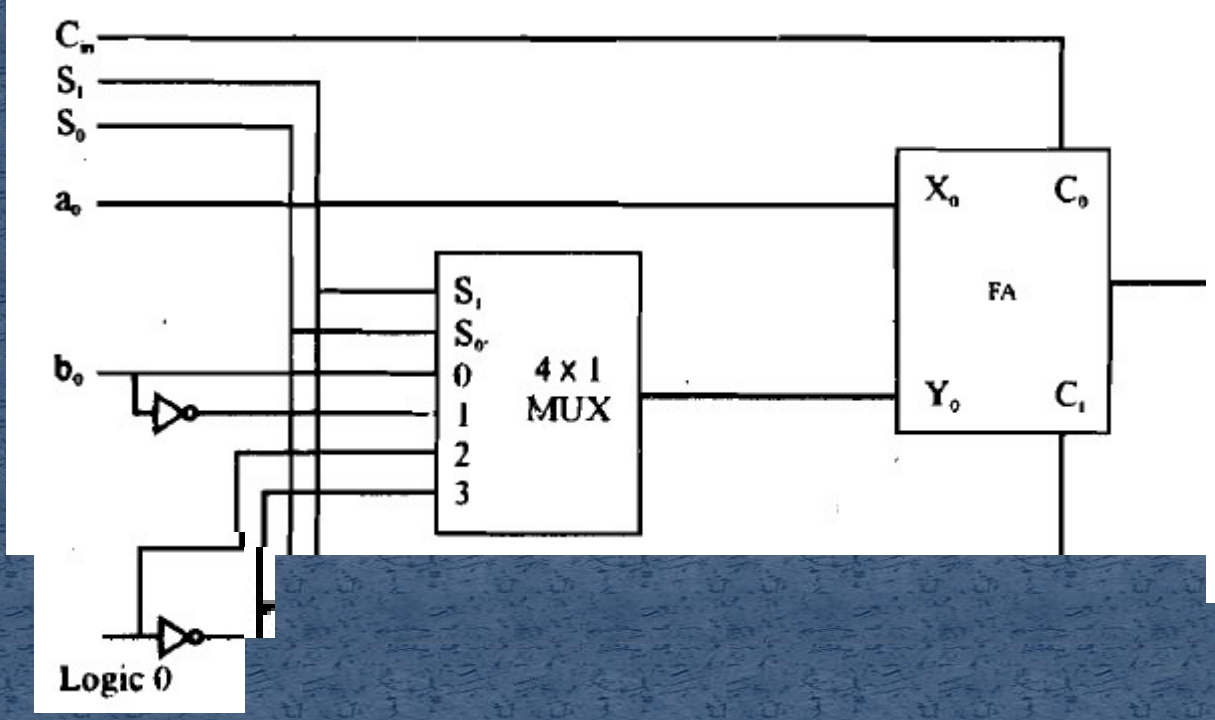
- MAR←IR(Address)
- DR ← [MAR]
- R1← DR
- R1← R1+1
- DR←R1
- MAR ← PC
- [MAR] ← DR
- If R1==0; then PC ← PC + 1

Instruction Pipeline

Time Slot ->	1	2	3	4	5	6	7	8	9	10	11
Instruction 1	IF	ID	OF	EX	SR						
Instruction 2		IF	ID	OF	EX	SR					
Instruction 3			IF	ID	OF	EX	SR				
Instruction 4				IF	ID	OF	EX	SR			
Instruction 5					IF	ID	OF	EX	SR		
Instruction 6						IF	ID	OF	EX	SR	
Instruction 7							IF	ID	OF	EX	SR

ALU Organisation

Arithmetic Operation



S_1	S_0	C_{in}	Y val
0	0	0	B
0	0	1	B
0	1	0	$\overline{B}$
0	1	1	$\overline{B}$
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

8 out of Possible 16 Logic Operations for 2 bits

x'=1	x'=1	x'=0	x'=0
y'=1	y'=0	y'=1	y'=0
x=0	x=0	x=1	x=1
y=0	y=1	y=0	y=1

0	0	0	0	$F_0 = 0$
0	0	0	1	$F_1 = x.y$
0	0	1	0	$F_2 = x.\bar{y}$
0	0	1	1	$F_3 = x$
0	1	0	0	$F_4 = \bar{x}.y$
0	1	0	1	$F_5 = y$
0	1	1	0	$F_6 = x \oplus y$
0	1	1	1	$F_7 = x + y$

Shit operation

- Use shift registers to do so

Activites to be Pefromed

- Study the Block 3, Units 1 to 3
- Solve questions of CYPs in the Block
- Solve questions given in assignments and previous year question papers
- Discuss with us, if there is any problem.