Azim Ahmed Bijapur

DATE:

312011 SPCC - Asgn-3

Q1. Dissuss the design of single pass assembler for 8088 microprocessur.

→ (i) Registers

8088 consists of 8 16-bit registers :-

(a) Data registers used for arithmetic, logical and ip/op operations

- Accumulator (Ax)
- Base register (Bx)
- Count register (cx)
- Data register (DX)

(b) Pointer registers for memory access

- source index
- destination index

(c) Segment registers to point to different segments in the memory.

- code segment - Stack segment
- Data segment - Extra segment

(ii) Memory

The various units of memory available are as follows :-

	Unit	Byte	Bit
B	Byte	1	8
H	half word	2	16
F	Full word	4	32
D	Double word	8	64

(iii) Instruction format

(a) Register to register
The first and second operand is present in the register

(b) Register to memory
Here the first operand is in the register and the other one is in the memory.

(c) Immediate instruction
Data operands are specified in the opcode of the instruction itself rather than in a memory or register.

(d) Call or jump instructions
For these instructions, the operand is specified as a memory address which is used to transfer program control to a different part of the program.

(iv) Assembly language of 8088

1. statement format
[Label:] opcode operand (s) ; comment string

2. Assembler directives

(a) Declaration of constants

Declaration, and reservation of storage is attained in the same way.

eg:-

```
A  DB  25     ; Reserve byte and initialize
B  DW  ?      ; Reserve word, no initialization
```

(b) EQU

It is used to make program more readable. Wherever a symbol is defined using EQU statement no memory would be allocated, only entry is made ~~to~~ in the symbol table.

Q. 2. Explain in detail flow chart of single pass assembler.

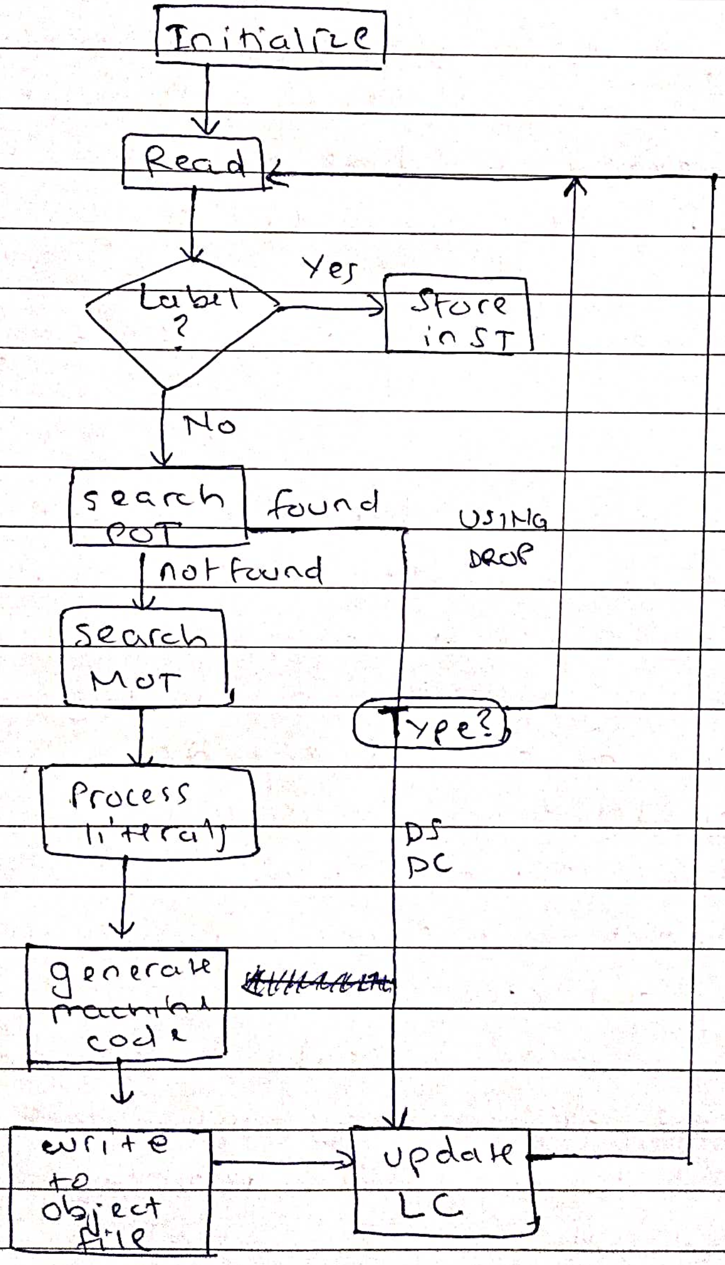
→ • First we initalize the literal and symbol table. Symbol table stores the labels and symbols used in the program. Literal table stores the value of literals used in the program.

• Then each line of source code is parsed to identity opcode, operands and labels or symbols

• If ~~we~~ the line contains any labels or symbls it is added to the symbol table with correspon-ding memory address.

• If a line contains literal, it is added to the liteeral table and a machine code instruction is generated to initialize memory location with literal's value.

- The assembler then generates ~~operates~~ machine code for each instruction using the opcode and operands. If the instruction references a symbol or label, the assembler replaces the symbol with corresponding memory address.
- The generated machine cod is output to an object file that can be loaded and executed.


Initialize
Read
Label ?
Yes
Store in ST
No
search POT
found
USING DROP
not found
search MOT
Type?
Process literals
DS
DC
generate machine code
write to object file
update LC