01/02/22

Tutorial - 04 - SS

U19CS009

Page No.:
Date: / /

Q-1 The various data structures used in assembler are :-

a.) **Location counter** :- it keeps track of each instruction location. LC is a variable that initialized to beginning address.
i.e, address specified in 'START' statement.
eg. START 501 ⇒ LC = 501 (LC contains value 501)

b.) **MOT (Machine Operation Table)** :- it is a static or fixed table. The contents do not change during lifetime of an assembler. It stores names of mnemonics, it's binary code and it's length.

	Mnemonic	Opcode	length
→	ADD	01	1
→	SUB	02	1

c. **Pseudo Operation Table (POT)** :- also known as OPTAB Operation code table
It is static or fixed table which contains :-
1. Mnemonic - opcode
2. class
3. Mnemonic info

	Mnemonic opcode	class	Mnemonic info
→	MOVER	IS	(04, 1)
→	OS	DL	R# 7

Class indicates whether opcodes corresponds to an imperative statement (IS), declarative (DS or Assembler directive (AD)

Page No.:
Date : / /

→ IS → mnemonic info contain pair of (machine opcode, instruction length).

→ DS & AD → ID of routine

d). Symbol table → used to store all symbol used in a program, which includes variables, constants, procedure, label etc. It is generated by analysis phase and is used by synthesis phase to generate machine code

Symbol	value	location	length
N	1	20	4

e) Literal table :- contains information about all the literal encountered in assembly along with their corresponding assigned location

Literal	Address
='2'	121
='1'	134

f). Pool table :- it contains starting literal number of each pool

literal number
0
2
4

Q.2

a → Pass 2
b → Pass 1
c → Pass 2
d → Pass 1

Page No.:
Date: / /

Q-3 → A single pass assembler scans the program only once and creates the equivalent binary program.

→ The assembler substitute all of the symbolic instruction with machine code in one pack.

→ These are used when.

a. It is necessary or desireable to avoid a second pass over the source program

b. The external storage for the immediate file between two passes is slow or is inconvenient to use

(Q9)

Source Program	Target code
START 100	01 - 100
MOVER AREG, A	(100) 04 01 107
PRINT B	101) 10 - 501
ADD BREG, ='a'	102) 01 02 105
SUB BREG, D	(03) 02 02 503
COMP CREG, ='2B'	104) 06 03 106
LTORG	105) - - 009
	106) - - 023
ADS 3	107) - - 003
LABEL : EQU A	NO code generation
ORIGIN 500	NO code generation
LI : MULT, CREG = '7'	500) 03 03 504
B PC 10	501) - - 010
MOVEM CREG, '7'	502) 05 03 504
D DC 8	503) - - 008
END	504) - - 007

Page No.:
Date : / /

SYMBOL TABLE

Symbol no.	Symbol	Address
01	A	107
02	B	501
03	D	
04	LABEL	107
05	L1	500

LITERAL TABLE

Lit no.	Literal	Address
01	='4'	105
02	='23'	106
03	='7'	504

Pool table

POOL TAB
01
03

Table of Incomplete Instruction (TII)

LC NO	Incomplete Instruction
100	A
101	B
102	='4'
103	D
104	='23'
500	='7'
502	='71

Page No.:
Date: / /

Q-4

PASS 1 :-

	LC
START 102	
READ X	102
READ Y	103
MOVER AREA, X	104
ADD AREA, Y	105
MOVEM AREA, RESULT	106
PRINT RESULT	107
STOP	108

X DS 1 109
Y DS 1 110
RESULT DS 1 111
END

Ans

Symbol table

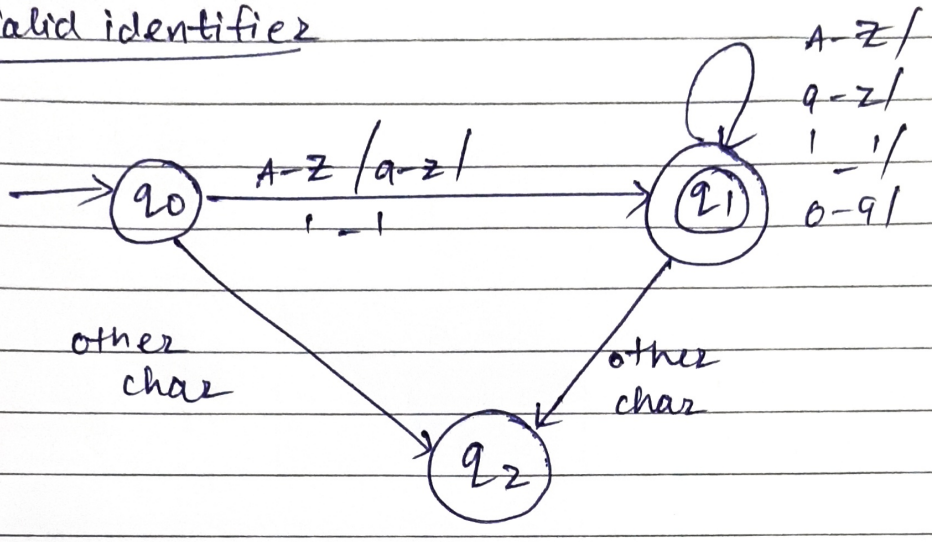
Symbol	Address	length
X	109	1
Y	110	1
RESULT	111	1

→

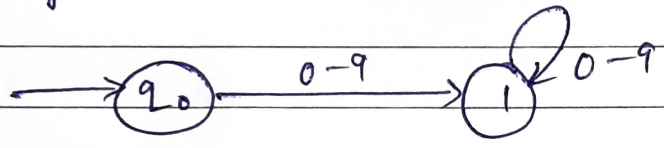
MOT LC	~~MNEMONIC~~ MNEMONIC	~~TYPE~~ TYPE	~~OPCODE~~ OPCODE	~~LENGTH~~ LENGTH
-	START/END	AD	01/02	-
102	READ	IS	09	01
103	READ	IS	09	01
104	MOVER	IS	04	01
105	ADD	IS	01	01
106	MOVEM	IS	05	01
107	PRINT	IS	10	01
108	STOP	IS	00	01
109	DS	DL	02	-
110	DS	DL	02	-
111	DS	DL	02	-

Q-5

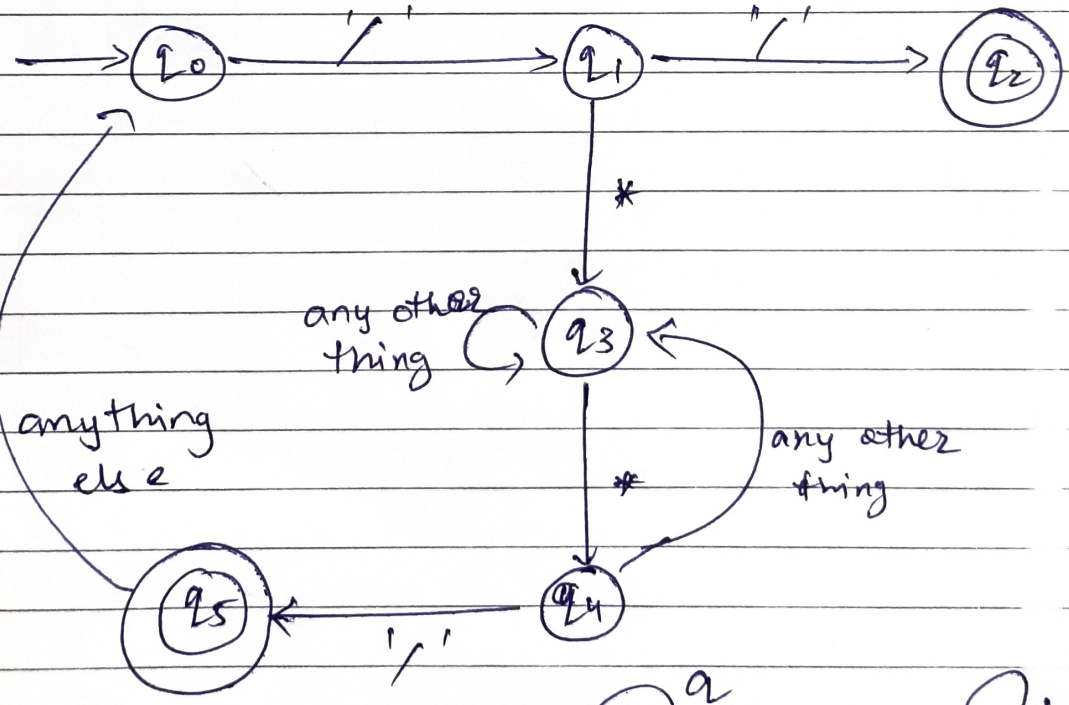
Valid identifier



Integer constant:



Comments :-



Q-6