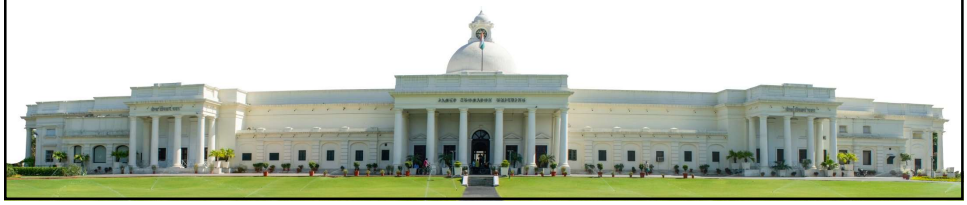
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

System Software
CSN-252
Assembler Design



- Object program contains three type of records

Header record:

Col.	Information
1	H
2-7 (6)	Program name
8-13 (6)	Starting address of the object program (hex)
14-19 (6)	Length of object program in bytes (hex)

Text record:

1	T
2-7 (6)	Starting address for object code in this record (hex)
8-9 (2)	Length of object code in this record in bytes (hex)
10-69 (60)	Object code in hex

End record:

1	E
2-7 (6)	Address of first executable instruction in object program (hex)

SIC Assembler V1.2

```
1000                test    start   1000
1000    001012      first   lda     five
1003    0C100F              sta     alpha
1006    501015              ldch    charz
1009    541016              stch    c1
100C    4C0000              rsub
100F                alpha   resw    1
1012    000005      five    word    5
1015    5A          charz   byte    c'Z'
1016                c1      resb    1
1017                        end     first
```

Header record:

Col.	Information
1	H
2-7 (6)	Program name
8-13 (6)	Starting address of the object program (hex)
14-19 (6)	Length of object program in bytes (hex)

SIC Assembler V1.2

```
1000                test    start   1000
1000    001012      first   lda     five
1003    0C100F              sta     alpha
1006    501015              ldch    charz
1009    541016              stch    c1
100C    4C0000              rsub
100F                alpha   resw    1
1012    000005      five    word    5
1015    5A          charz   byte    c'Z'
1016                c1      resb    1
1017                        end     first
```

Text record:

1	T
2-7 (6)	Starting address for object code in this record (hex)
8-9 (2)	Length of object code in this record in bytes (hex)
10-69 (60)	Object code in hex

SIC Assembler V1.2

```
1000                test    start   1000
1000    001012      first   lda     five
1003    0C100F              sta     alpha
1006    501015              ldch    charz
1009    541016              stch    c1
100C    4C0000              rsub
100F                alpha   resw    1
1012    000005      five    word    5
1015    5A          charz   byte    c'Z'
1016                c1      resb    1
1017                        end     first
```

End record:

1	E
2-7 (6)	Address of first executable instruction in object program (hex)

```
Pass 1:
begin
   read first input line
   if (OPCODE = 'START') then
            begin
               save #[operand] as starting address
               initialize LOCCTR to starting address
               write line to intermediate file;
                read next input line;
            end
    else    initialize LOCCTR to 0
    while (OPCODE != 'END') do
            begin
                if (this is not a comment line) then
                        begin
                           if (there is a symbol in the LABEL field) then
                                   begin
                                      search SYMTAB for LABEL
                                      if (found) then
                                              set error flag (duplicate)
                                      else    insert (LABEL, LOCCTR) into SYMTAB
                                    end
                           search OPTAB for OPCODE
                           if (found) then
                                   ...
```

```
test    start   1000
first   lda     five
        sta     alpha
        ldch    charz
        stch    c1
                :
alpha   resw    1
five    word    5
charz   byte    c'Z'
c1      resb    1
        end     first
```

```
                                                    test    start   1000
                                                    first   lda     five
                                                            sta     alpha
Pass 1: (contd.)                                            ldch    charz
                    search OPTAB for OPCODE                 stch    c1
                    if (found) then                                 :
                            add 3 to LOCCTR         alpha   resw    1
                    else if WORD                    five    word    5
                    else if RESW                    charz   byte    c'Z'
                    else if RESB                    c1      resb    1
                    else if BYTE                            end     first
                    else set error flag (invalid opcode)
                write line to intermediate file
                read next input line
        end (while)
        write last line to intermediate file
        save (LOCCTR – starting address) as program length
end
```

```
Pass 2:
begin                                               test    start   1000
    read first input line (from intermediate file)  first   lda     five
    if (OPCODE = 'START') then                              sta     alpha
            begin                                           ldch    charz
               write listing line                           stch    c1
               read next input line                                 :
            end                                     alpha   resw    1
    write header record to object program           five    word    5
    initialize first Text record                    charz   byte    c'Z'
    while (OPCODE != 'END') do                      c1      resb    1
            begin                                           end     first
              if (this is not a comment line) then
                    begin
                      search OPTAB for OPCODE
                      if (found) then
                            begin
                              if (there is a symbol in OPERAND field) then
                                    begin
                                      search SYMTAB for OPERAND
                                      if (found) then
                                            store symbol value as operand address
                                      else {store 0 as operand address; set error flag; }
                              else store 0 as operand address
                              assemble the instruction;
```

Assembler

- What else?
  - There may be WORD or BYTE in opcode field
  - Write / change text records
  - Write end record
- Why intermediate file?
- What is in intermediate file?
- Output of assembler?
  - An object file + A listing file

Assembler

- Do we need to search for OPCODE both in pass 1 and Pass 2?
- Enter symbol in SYMTAB if encountered in col. 1 or either in col. 1 or 3?
- Errors

  invalid opcodes / assembler directive

  Invalid symbols / duplicate symbols
- Retain pointers to OPCODE table / SYMTAB in intermediate file

```
1000                test    start   1000
1000    001012      first   lda     five
1003    0C100F              sta     alpha
1006    501015              ldch    charz
1009    541016              stch    c1
100C    4C0000              rsub
100F                alpha   resw    1
1012    000005      five    word    5
1015    5A          charz   byte    c'Z'
1016                c1      resb    1
1017                        end     first


Htest  001000000017
T0010000F0010120C100F501015541016 4C0000
T00100F040000005A
E001000
```