INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

System Software
CSN-252
Assembler

2.2.2 (Program Relocation)
2.3.1 (Literals)
- L. L. Beck

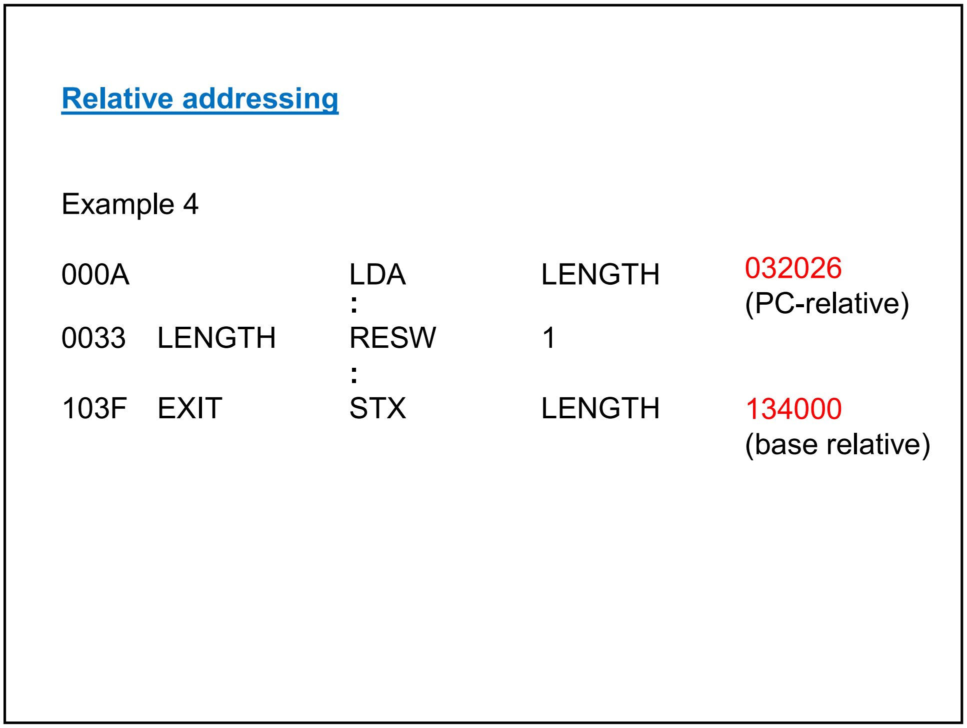
- Immediate mode of addressing
- Indirect addressing

Base relative addressing

Example 3:

```
    0003            LDB       #LENGTH
                    BASE      LENGTH
                    :
    0033  LENGTH    RESW      1
    0036  BUFFER    RESB      4096
                    :
    104E            STCH      BUFFER, X        57C003
```

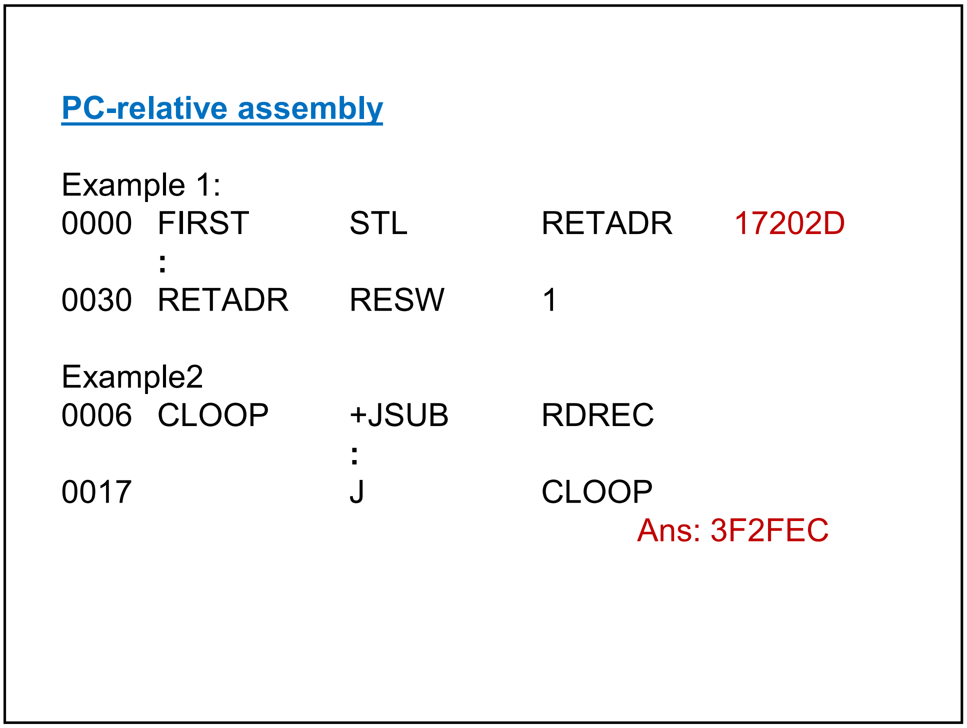
- Assembler knows the contents of the program counter
- Programmer must tell the contents of base register.
- Done by using BASE.
- **NOBASE** tells the assembler that the contents of base register can no longer be relied upon for addressing.



Relative addressing

Example 4

```
000A            LDA      LENGTH    032026
                 :                 (PC-relative)
0033   LENGTH   RESW     1
                 :
103F   EXIT     STX      LENGTH    134000
                                   (base relative)
```



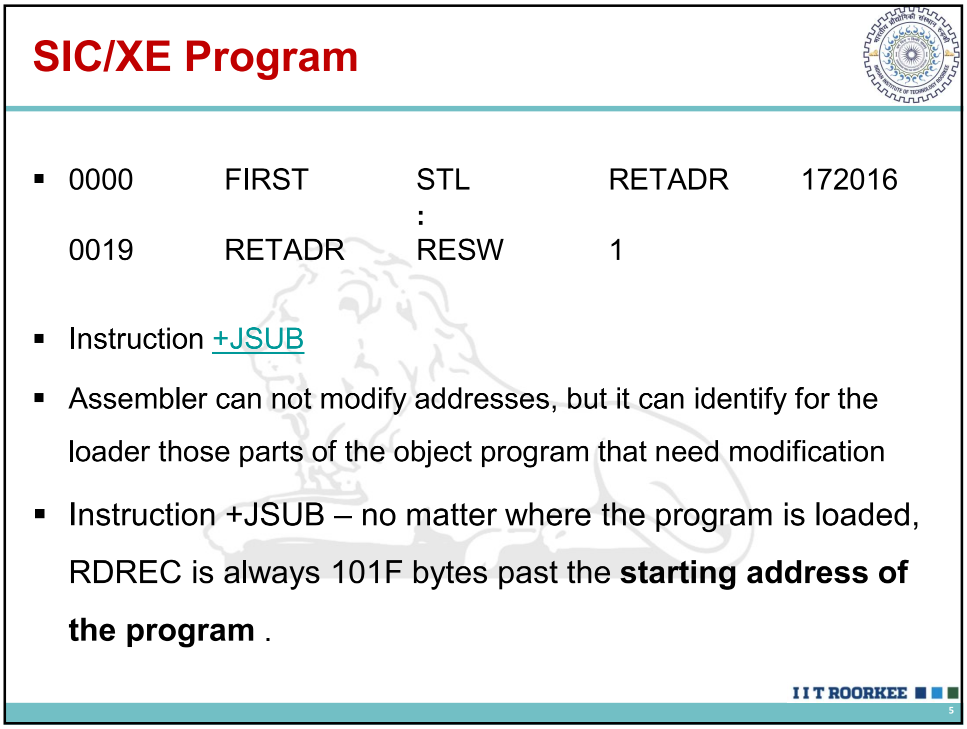
PC-relative assembly

Example 1:

```
0000   FIRST    STL      RETADR   17202D
         :
0030   RETADR   RESW     1
```

Example2

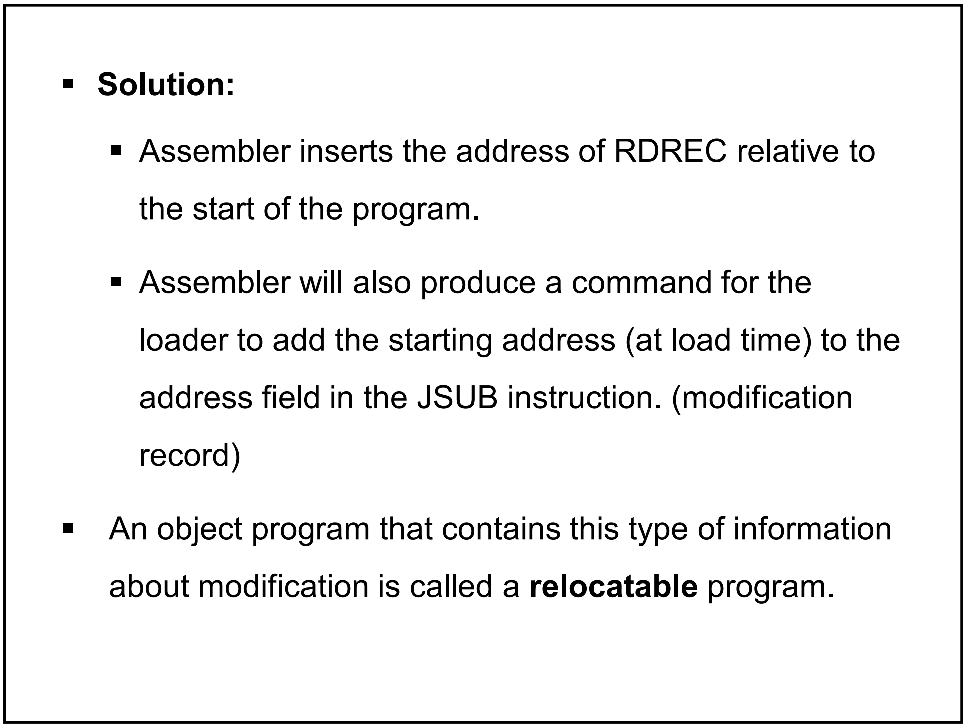
```
0006   CLOOP    +JSUB    RDREC
                 :
0017            J        CLOOP
                              Ans: 3F2FEC
```



SIC/XE Program

```
0000    FIRST     STL      RETADR    172016
                  :
0019    RETADR    RESW     1
```

- Instruction +JSUB
- Assembler can not modify addresses, but it can identify for the loader those parts of the object program that need modification
- Instruction +JSUB – no matter where the program is loaded, RDREC is always 101F bytes past the **starting address of the program** .



- **Solution:**
  - Assembler inserts the address of RDREC relative to the start of the program.
  - Assembler will also produce a command for the loader to add the starting address (at load time) to the address field in the JSUB instruction. (modification record)
- An object program that contains this type of information about modification is called a **relocatable** program.

Modification record

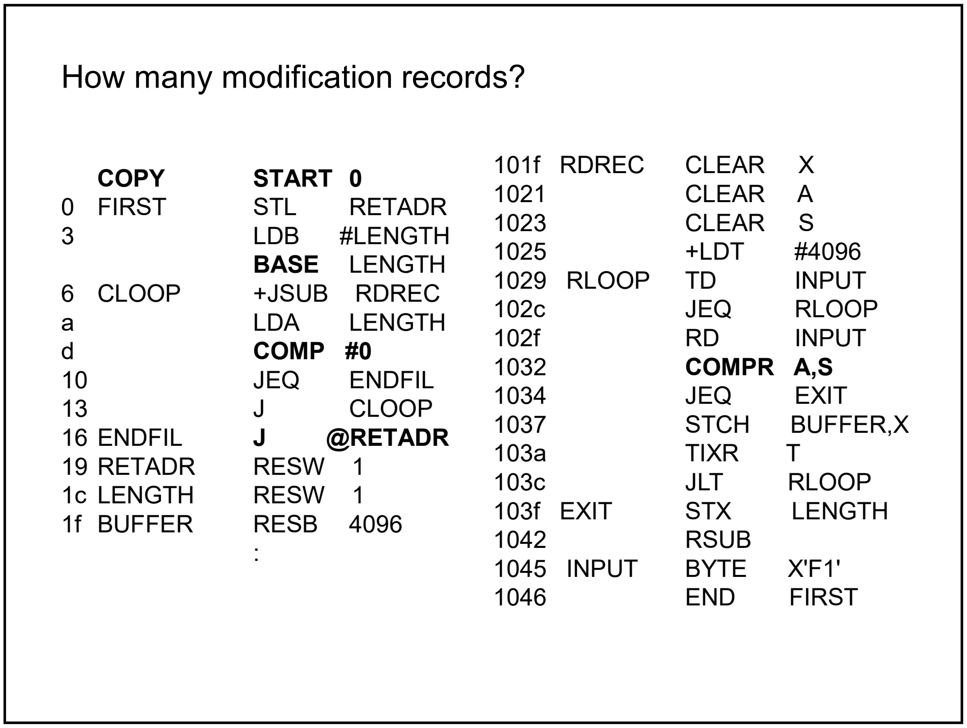
Col. 1	M
Col. 2-7	starting location of the **address field** to be modified, relative to the beginning of the program (hex)
Col. 8-9	Length of the address field to be modified, in half bytes (hex.)

closely relative to the architecture of SIC/XE

- Modification record for JSUB instruction
  M00000705

Modification record

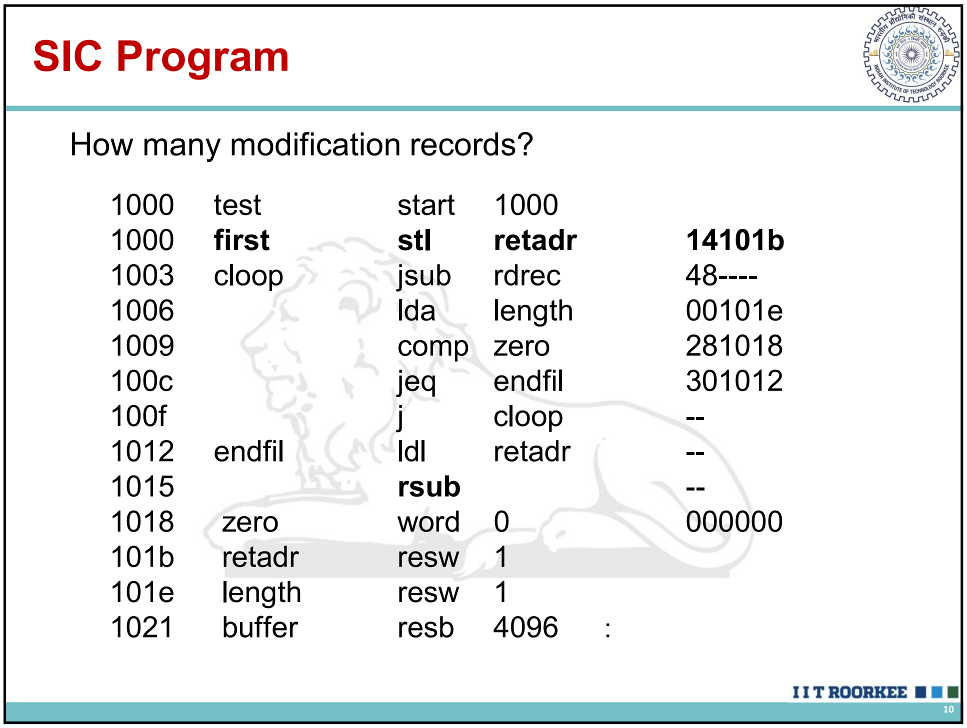
- Instructions: **COMP #0**; **+LDT #4096**
- 1037 STCH BUFFER,X 57C003
- Instructions using immediate, PC-relative or Base register relative addressing need not to be modified.
- Only parts of the program that require modification at load time are those that specify direct addresses
- **Text records** are exactly the same as those that would be produced by an absolute assembler however the load addresses are relative.
- Header and End records?
- An object program that contains information about modification is called a **relocatable** program



How many modification records?

```
    COPY          START  0
0   FIRST         STL     RETADR
3                 LDB    #LENGTH
                  BASE    LENGTH
6   CLOOP         +JSUB   RDREC
a                 LDA     LENGTH
d                 COMP   #0
10                JEQ     ENDFIL
13                J       CLOOP
16 ENDFIL         J      @RETADR
19 RETADR         RESW    1
1c LENGTH         RESW    1
1f  BUFFER        RESB    4096
                  :
```

```
101f  RDREC    CLEAR    X
1021           CLEAR    A
1023           CLEAR    S
1025           +LDT     #4096
1029  RLOOP    TD       INPUT
102c           JEQ      RLOOP
102f           RD       INPUT
1032           COMPR    A,S
1034           JEQ      EXIT
1037           STCH     BUFFER,X
103a           TIXR     T
103c           JLT      RLOOP
103f  EXIT     STX      LENGTH
1042           RSUB
1045  INPUT    BYTE     X'F1'
1046           END      FIRST
```



SIC Program

How many modification records?

```
1000    test        start   1000
1000    first       stl     retadr      14101b
1003    cloop       jsub    rdrec       48----
1006                lda     length      00101e
1009                comp    zero        281018
100c                jeq     endfil      301012
100f                j       cloop       --
1012    endfil      ldl     retadr      --
1015                rsub                --
1018     zero       word    0           000000
101b     retadr     resw    1
101e     length     resw    1
1021     buffer     resb    4096     :
```

IIT ROORKEE

10

SIC Program

- Instruction from our SIC program

```
1006        LDA        LENGTH     00101E
```

- What if this program is loaded from 2000?
- Do we need to change all instructions / words?

```
1016        ZERO       WORD       4126
```

- Looking at the object code alone, it is not possible to tell which values represent addresses and which represent constant data items.

Machine-independent Assembler features

- Presence or absence of these features depend on issues such as programmer convenience

Literals

- **Write the value of a constant operand as a part of the instruction that uses it**? – more convenient for the programmer

Example

```
001A  ENDFIL LDA       EOF        032010
:
002D  EOF    BYTE      C'EOF'
```

- 001A ENDFIL LDA =C'EOF' 032010? (3-byte literal)

```
       *      =C'EOF'             454F46
```