



System Software CSN-252 Assembler

2.3.2 (ORG)
2.3.3 (Expressions)
- L. L. Beck



Assumptions about SIC/XE ISA



RSUB	uses format 3/4 (Appendix A)
DIV m	stores rounded off value in register A
COMP m	sets condition code = Less Than (LT) if $(A) < (m..m+2)$
COMPR r1, r2	sets condition code = Less Than (LT) if $(r1) < (r2)$
AND m	Bitwise and operation

Assembler directive EQU

```
MAXLEN      EQU      4096
```

```
•           +LDT      #MAXLEN
```

- When the assembler encounters the EQU statement, it enters MAXLEN into SYMTAB **with value 4096**.
- During assembly of LDT it searches SYMTAB for MAXLEN, using its value **(address or constant?)** as the operand in the instruction.
- The resulting object code is exactly same, however it is easier to understand the source statement.
- It is also much easier to find and change the value of MAXLEN.

EQU



Write object code

CASE 1:

```

ZERO      EQU      0
LDA        #ZERO      010000

```

CASE 2:

```

1003      LDA        #ZERO      012009
:
100F      ZERO      WORD      0

```

- Defining mnemonic names for registers.
- If the assembler only understands register numbers and not names:

RMO A, X

RMO 0,1

- A EQU 0
 X EQU 1 etc.
 :

- Normally used when machine has general purpose registers.

BASE EQU R1

ORG

- Assembler directive that can be used to **indirectly assign values to symbols**.

ORG	value
-----	-------

- Value is a constant or an expression.
- Resets the value of LOCCTR to the specified value.
- As the values of the symbols used as labels are taken from LOCCTR, this will affect the values of all labels defined until the next ORG.

COPY	START	0
FIRST	LDA	ALPHA
	ADD	BETA
	ORG	256
ALPHA	WORD	2
BETA	WORD	5
	END	FIRST

- Altering LOCCTR may result in an incorrect assembly
- Example – defining a symbol table

STAB (100 entries)	SYMBOL(6)	VALUE(3)	FLAGS(2)

Method 1:

STAB	RESB	1100
SYMBOL	EQU	STAB
VALUE	EQU	STAB+6
FLAGS	EQU	STAB+9
	LDA	VALUE, X

Method 2:

STAB	RESB	1100
	ORG	STAB
SYMBOL	RESB	6
VALUE	RESW	1
FLAGS	RESB	2
	ORG	STAB+1100
	LDA	VALUE, X

- In case of EQU / ORG all symbols used on the right-hand side of the statement must have been defined previously in the program.

ALPHA	RESW	1	
BETA	EQU	ALPHA	permitted

BETA	EQU	ALPHA	
ALPHA	RESW	1	not permitted

- Why?

- **Product of the forward reference problem:**

ALPHA	EQU	BETA
BETA	EQU	DELTA
DELTA	RESW	1

- Cannot be resolved by an ordinary two-pass assembler regardless of how the work is divided between the passes.

Similar restrictions apply to ORG

Example

	ORG	ALPHA	✗
BYTE1	RESB	1	
BYTE2	RESB	1	
BYTE3	RESB	1	
	ORG		
ALPHA	RESB	1	

Expressions



- Most assemblers allow expressions in place of single operand.
 - Expressions may use +, -, *, /.
- ```
MAXLEN EQU BUFFEND - BUFFER
```
- ```
LOC EQU BUFFEND + 100
```
- Division is usually defined to produce an integer result
 - Individual terms in the expression may be **constants**¹, **user-defined symbols**², or **special terms**³ (*).
 - Assembler must evaluate this value and produce a single operand address or value.

IIT ROORKEE

11

- Some values in the object program are relative and some are absolute.
 \Rightarrow expressions are also either relative or absolute depending on the value they produce.
- **absolute expression.**
 - contains only absolute terms or
 - relative terms occur in pairs with opposite signs.
- None of the relative terms may appear into a multiplication or division operation.
- **Relative expression**
 - all of the relative terms except one can be paired.
 - The remaining unpaired relative term must have a positive sign

```
MAXLEN EQU BUFFEND - BUFFER      absolute
```

```
ABC EQU BUFFEND + BUFFER        (?)
```

Expressions



ABC EQU BUFEND – BUFFER – RETADDR

CDE EQU BUFEND – BUFFER + RETADDR

LMN EQU BUFEND - 100

XYZ EQU BUFFER * 10 – BUFEND

