



# **SNS COLLEGE OF ENGINEERING**

Kurumbapalayam (Po), Coimbatore – 641 107

**An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai



## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**COURSE NAME :19IT301 COMPUTER ORGANIZATION AND  
ARCHITECTURE**

**II YEAR /III SEMESTER**

**Unit 1- BASIC STRUCTURE OF COMPUTERS**

**Topic 8 :Assembly language**



# Assembly language



- ✓ An **assembly language** is a low-level programming **language** designed for a specific type of processor.
- ✓ The set of rules for using the mnemonics in the specification of complete instructions and programs is called the **syntax of the language**
- ✓ The user program in its original alphanumeric text format is called a **source program**
- ✓ The assembled machine language program is called an **object program**
- ✓ Programs written in an assembly language can be automatically translated into a sequence of machine instructions by a program called an **assembler**



# Assembly language – Assembler directives



✓ **Assembler directives** supply data to the program and control the assembly process

✓ It improves program readability.

✓ **Examples of common assembler directives** are

➤ ORG (origin),

➤ EQU (equate)

➤ DATAWORD

➤ RESERVE

➤ RETURN

✓ SUM EQU 200

It simply informs the assembler that the name SUM should be replaced by the value 200 wherever it appears in the program



# Assembly language – Assembler directives



- ✓ If the assembler is to produce an object program, it has to know
  - How to interpret the names
  - Where to place the instructions in the memory
  - Where to place the data operands in the memory
- ✓ To provide this information, the source program may be written as shown in Figure 2.18
- ✓ The program begins with assembler directives



# Assembly language – Assembler directives



	Memory address label	Operation	Addressing or data information
Assembler directives	SUM	EQU	200
		ORIGIN	204
	N	DATAWORD	100
	NUM1	RESERVE	400
		ORIGIN	100
Statements that generate machine instructions	START	MOVE	N,R1
		MOVE	#NUM1,R2
		CLR	R0
	LOOP	ADD	(R2),R0
		ADD	#4,R2
		DEC	R1
		BGTZ	LOOP
		MOVE	R0,SUM
Assembler directives		RETURN	
		END	START

**Figure 2.18** Assembly language representation for the program in Figure 2.17.

	100	Move	N,R1
	104	Move	#NUM1,R2
	108	Clear	R0
LOOP	112	Add	(R2),R0
	116	Add	#4,R2
	120	Decrement	R1
	124	Branch>0	LOOP
	128	Move	R0,SUM
	132		
		⋮	
		⋮	
SUM	200		
N	204	100	
NUM1	208		
NUM2	212		
		⋮	
		⋮	
NUM <sub>n</sub>	604		

**Figure 2.17** Memory arrangement for the program in Figure 2.12.

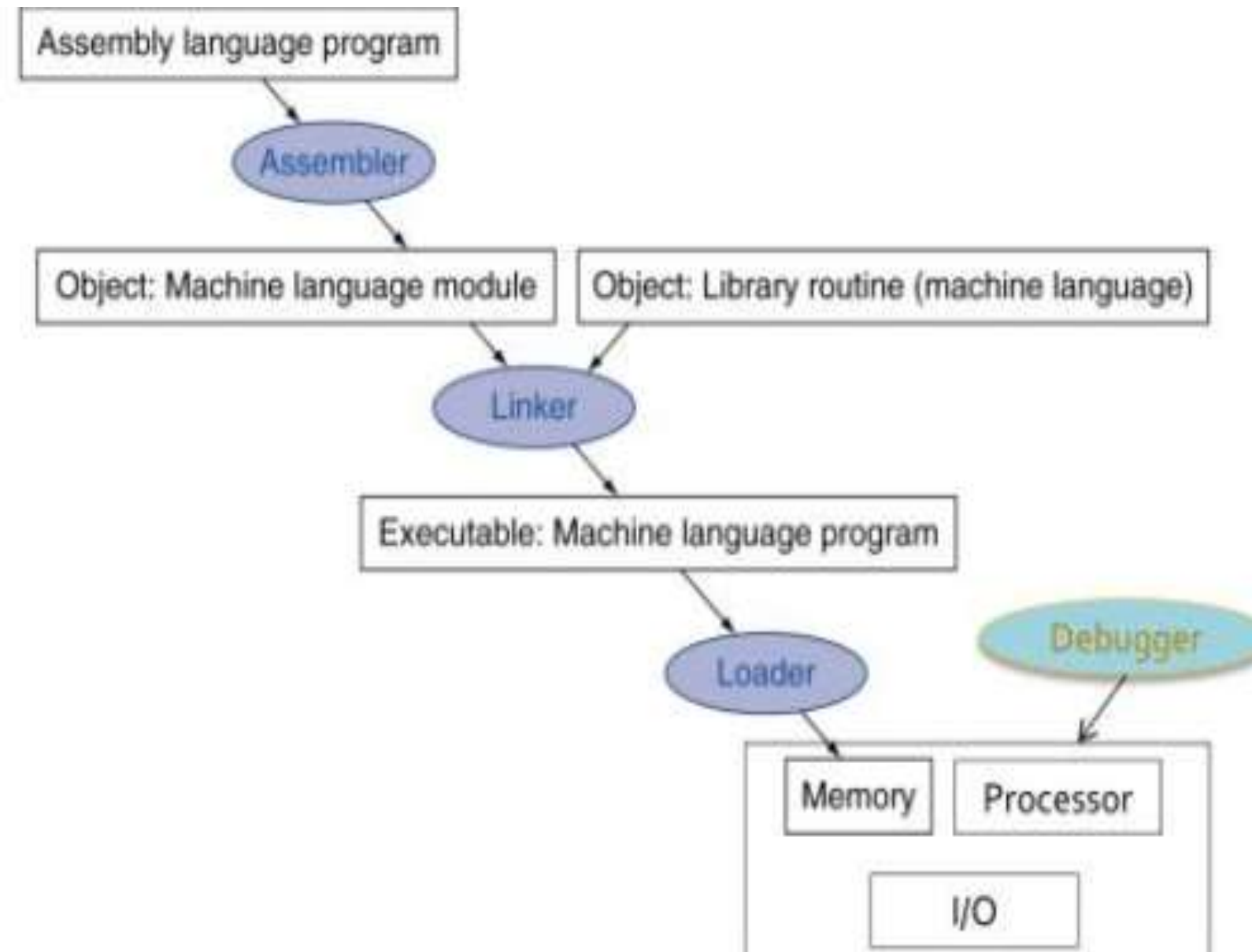


## Assembly language – Assembler directives

- ✓ The Equate directive, EQU, which informs the assembler about the value of SUM.
- ✓ The directive, ORIGIN, tells the assembler program where in the memory to place the data block that follows.
- ✓ DATAWORD: states that the data value 100 is to be placed in the memory word at address 204
- ✓ The RESERVE directive declares that a memory block of 400 bytes is to be reserved for data
- ✓ The second ORIGIN directive specifies that the instructions of the object program are to be loaded in the memory starting at address 100.
- ✓ RETURN: returns control to the operating system of the computer.
- ✓ Labels may also be associated with addresses of data items . In Figure 2.18 there are four labels: SUM, N, NUM1 and LOOP



# Assembly language – Assembly and execution of programs





# Assembly language – Number notation



- ADD #93,R1                      Decimal number system
- ADD #%01011101,R1      Binary number system
- ADD #\$5D,R1                  Hexa decimal number system
- MOVE #5,(R2) or MOVEI 5, (R2) depends on assembly language





# Assessment



a). What is Assembly language?

---

---

---



b) Give the purpose of the following assembler directives:

1. ORIGIN
2. RETURN
3. EQU
4. RESERVE



# Reference



1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “Computer Organization”, McGraw-Hill, 6<sup>th</sup> Edition 2012.