

# Microprocessor

BCA 3<sup>rd</sup> Semester 2020

## Lecture- 13

**Subhadip MUKHERJEE**

Department of computer science

Kharagpur college

# Assembly Language Programming

## What is a procedure?

- The procedure (or subroutine) is a set of codes that can be branched to and returned from. The branch to a procedure is known as CALL and the return from the procedure is known as RETURN.

## Discuss the technique of passing parameters to a procedure.

- MOV CX, T

CALL DELAY

where, T represents delay parameter.

- A second technique is to use a memory location like,

MOV TEMP, T

CALL DELAY

where, TEMP is representative of memory locations.

- A third technique is to pass the address of the memory variable like,

MOV SI, POINTER

CALL DELAY

while in the procedure, it extracts the delay parameter by using the instruction MOV CX, [SI].

# Assembly Language Programming

**A typical program format using assembler directives.**

```
Line 1.  MODEL SMALL  ; selects small model
Line 2.  DATA        ; indicates data segment
        ...
        ...
        ...
Line 15. CODE         ; indicates start of code segment
        ...
        ...         body of the program
        ...
Line 20. END          ; End of file
```

# Assembly Language Programming

## What is a macro?

- A macro, like a procedure, is a group of instructions that perform one task. The macro instructions are placed in the program by the macro assembler at the point it is invoked.

## The general format of a macro is

NAME MACRO Arg 1 Arg 2 Arg 3

Statements .....

.....

ENDM

# Assembly Language Programming

**For the conditional assembly process, show the forms used for the IF statement**

Statement	Function
IF	If the expression is true
IFB	If argument is blank
IFE	If the expression is not true
OFDEF	If the label has been defined
IFNB	If argument is not blank
IFNDEF	If the label has not been defined
IFIDN	If argument 1 equals argument 2
IFDIFWWW	If argument 1 does not equal argument 2

# Assembly Language Programming

**For the conditional assembly process, show the relational operators used with WHILE and REPEAT.**

Operator	Function
EQ	Equal
NE	Not Equal
LE	Less than or Equal
LT	Less than
GT	Greater than
GE	Greater than or Equal
NOT	Logical inversion
AND	Logical AND
OR	Logical OR
XOR	Logical XOR

# Assembly Language Programming

**Check whether the given number is even or odd.**

LDA 2050 –loads the content of memory location 2050 in accumulator A

ANI 01 –performs AND operation between accumulator A and 01 and store the result in A

JZ 200D –jump to memory location 200D if ZF = 1

MVI A 11 –assign 11 to accumulator

JMP 200F –jump to memory location 200F

MVI A 22 –assign 22 to accumulator

STA 3050 –stores value of A in 3050

HLT –stops executing the program and halts any further execution

# Assembly Language Programming

**A block of data consisting of 256 bytes is stored in memory starting at 3000H. This block is to be shifted (relocated) in memory from 3050H onwards. Do not shift the block or part of the block anywhere else in the memory.**

Two blocks (3000 – 30FF and 3050 – 314F) are overlapping. Therefore it is necessary to transfer last byte first and first byte last.

```
MVI C, FFH : "Initialize counter"
```

```
LX I H, 30FFH : "Initialize source memory pointer 314FH"
```

```
LXI D, 314FH : "Initialize destination memory pointer"
```

```
BACK: MOV A, M : "Get byte from source memory block"
```

```
STAX D : "Store byte in the destination memory block"
```

```
DCX H : "Decrement source memory pointer"
```

```
DCX : "Decrement destination memory pointer"
```

```
DCR C : "Decrement counter"
```

```
JNZ BACK : "If counter 0 repeat"
```

```
HLT : "Stop execution"
```



# Assembly Language Programming

## Convert gray numbers to binary

2000 LDA 2050 is used to load the data from address 2050 in A

2003 MVI C, 07 is used to move the data 07 in C

2005 MOV B, A moves the data of A to B

2006 ANI 80 extracts the MSB (Most Significant Bit) of data available in A

2008 RRC rotates the bits of A to right without carry

2009 ANI 7F is used to Take AND between data in A and 7F

200B XRA B takes XOR between the data present in A and B

200C DCR C is used to decrement the contents of C

200D JNZ 2008 is used to jump to address 2008 if ZF = 0

2011 STA 3050 is used to store the result at memory address 3050

2014 HLT is used to end the program

# Assembly Language Programming

## **Distinguish between macro and procedure.**

- A procedure is invoked with a CALL instruction and terminated with a RET instruction.
- A macro is invoked, on the other hand, during program assembly and not when the program is run. Whenever in the program the macro is required, assembler substitutes the defined sequence of instructions corresponding to the macro.
- Macro does not require CALL–RET instructions and hence will be executed faster.

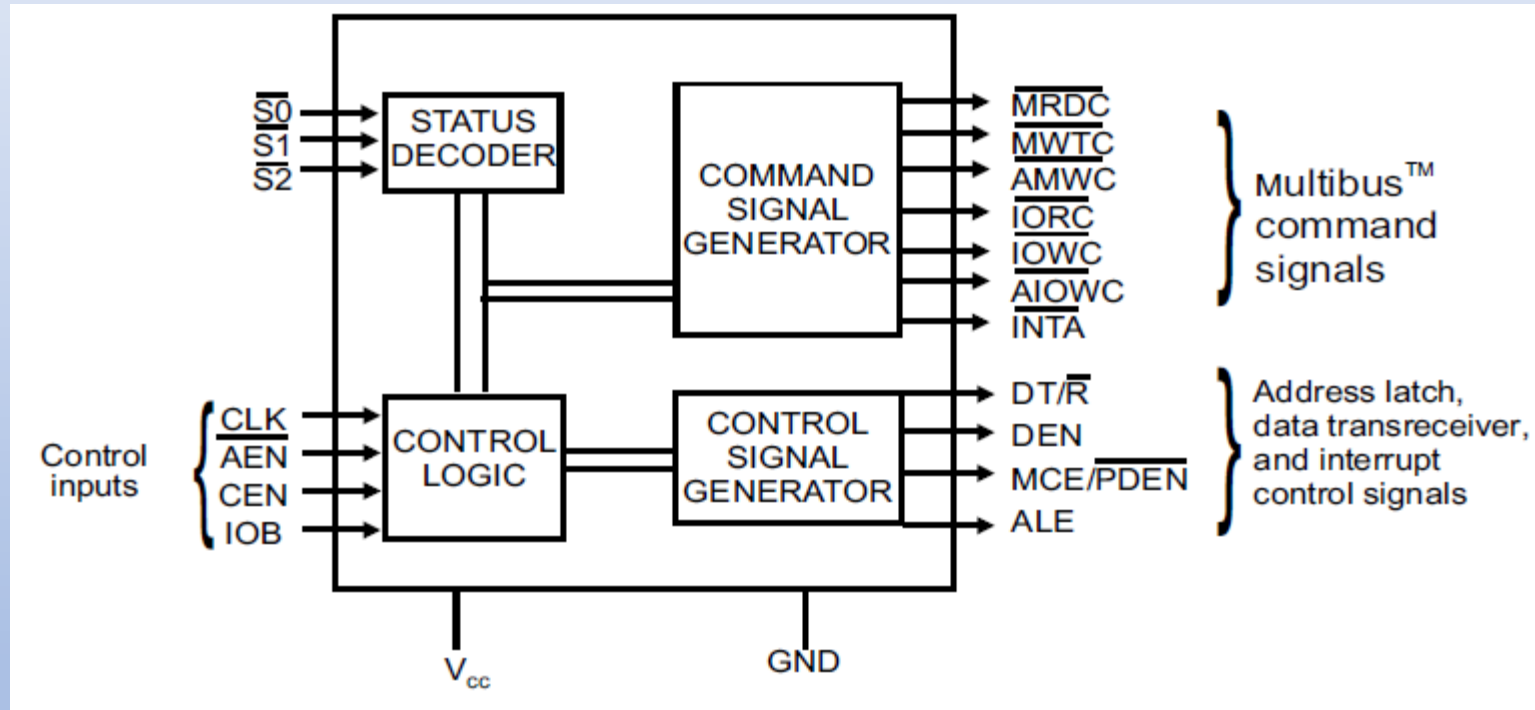
# Assembly Language Programming

**What kind of I/O is used for IN and OUT instructions?**

Mnemonic	Meaning	Format	Operation
IN	Input direct	IN Acc, Port	$(Acc) \leftarrow (Port)$ Acc = AL or AX
	Input indirect (variable)	IN Acc, DX	$(Acc) \leftarrow ((DX))$
OUT	Output direct	OUT Port, Acc	$(Port) \leftarrow (Acc)$
	Output indirect (variable)	OUT DX, Acc	$((DX)) \leftarrow (Acc)$

# Assembly Language Programming

## Block diagram of 8288.



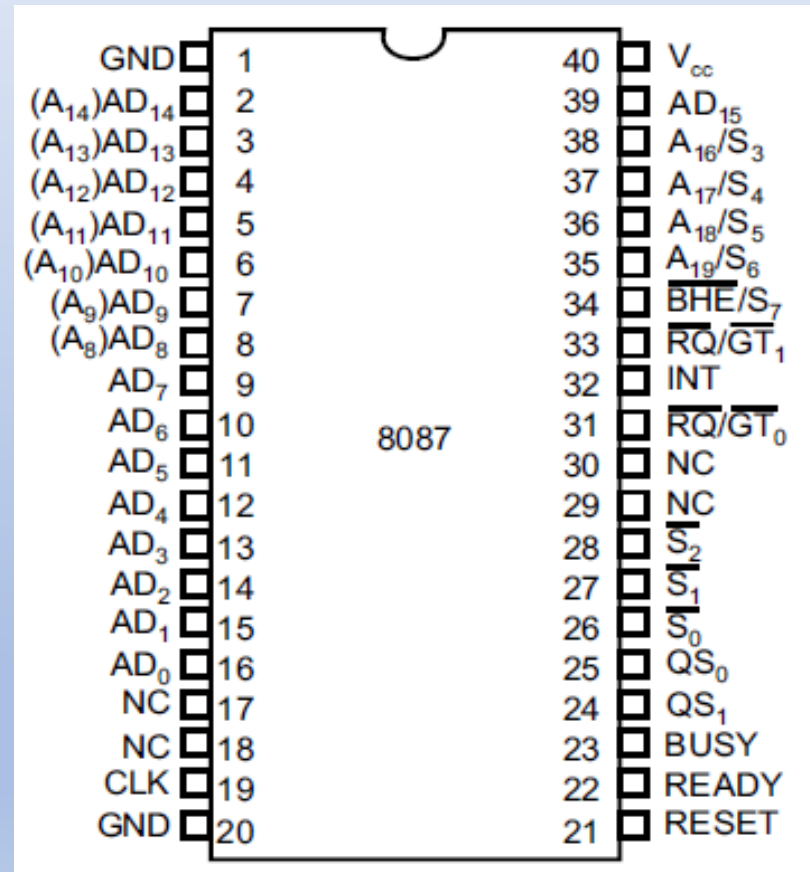
# Assembly Language Programming

The status pins S2 , S1 and S0 .

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Processor state	82C88 command
0	0	0	Interrupt Acknowledge	$\overline{INTA}$
0	0	1	Read I/O Port	$\overline{IORC}$
0	1	0	Write I/O Port	$\overline{IOWC}, \overline{AIOWC}$
0	1	1	Halt	None
1	0	0	Code Access	$\overline{MRDC}$
1	0	1	Read Memory	$\overline{MRDC}$
1	1	0	Write Memory	$\overline{MWTC}, \overline{AMWC}$
1	1	1	Passive	None

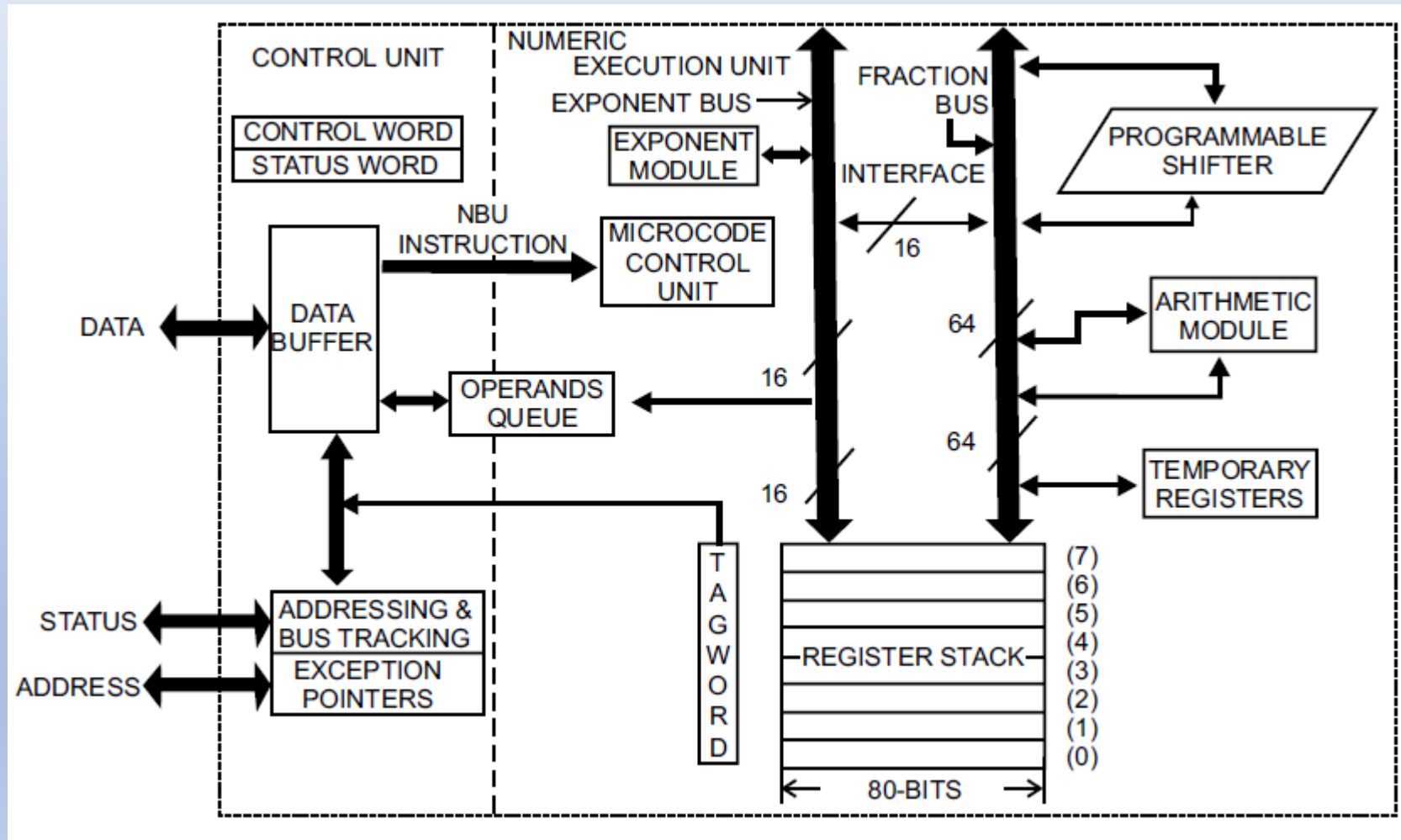
# Assembly Language Programming

## The pin diagram of 8087



# Assembly Language Programming

## The architecture of 8087



**Thank You**