# UNIT 3: DATA REPRESENTATION

**UNIT STRUCTURE**

3.1 Learning Objectives

3.2 Introduction

3.3 Data Representation

- 3.3.1 Decimal Number System
- 3.3.2 Binary Number System
- 3.3.3 Octal Number System
- 3.3.4 Hexadecimal Number System
- 3.3.5 Binary Coded Decimal (BCD)
- 3.3.6 American Standard Code for Information Interchange (ASCII)
- 3.3.7 Negative Number Representation in Binary System

3.4 Computer Arithmetic and their Implementation

3.5 Control and Data Path

3.6 Data Path Components

3.7 Design of ALU and Data Path

3.8 Control Unit Design

3.9 Let Us Sum Up

3.10 Further Readings

3.11 Answers to Check Your Progress

3.12 Model Questions

## 3.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- learn all the data representation techniques
- describe the negative number representation techniques
- describe computer arithmetic and their implementation
- describe control and data path including data path components
- describe design of ALU and Data Path

## 3.2 INTRODUCTION

In the previous unit, combinational circuits, sequential circuits along with counters and registers were covered in detail. In this unit, we will discuss all the data representation systems such as Decimal Number System, Binary Number System, BCD, Octal Number System, Hexadecimal Number System, ASCII etc. We will also discuss the negative number representation methods used in binary system. The concepts of design of ALU, Data Path as well as Control Unit Design are also covered in this unit. In the next unit, we will explore the different types of instructions and addressing modes.

## 3.3 DATA REPRESENTATION

We use decimal number system to represent data. Decimal number system is one where the base is ten. The range of base of decimal number is from 0 to 9.

In digital computer there is a special number system to represent the data. This number system is called as Binary Number system.

There are some other number systems, namely:

- Octal number (Base is 8)
- Hexadecimal number (Base is 16)

### 3.3.1 Decimal Number System

The symbols of Decimal number system are 0,1,2,3,4,5,6,7,8,9.There are ten symbols called digits. The order of the least-significant digit (right-most digit) is 10^0 (units or ones), the second right-most digit is of the order of 10^1 (tens), the third right-most digit is of the order of 10^2 (hundreds), and so on. For example,

$468=4x10^2 + 6x10^1+8x10^0$

## 3.3.2 Binary Number System

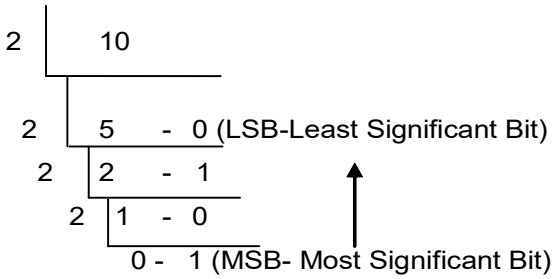
The base of binary number system is 2 as this number system follows two valued logic such as YES or NO, HIGH or LOW, ON or OFF, TRUE or FALSE. To represent two valued logic binary number system uses either 1 or i.e. the symbols of binary number system are 0,1. These 0 or 1 is called bit. Four numbers of bits are called nibble, and 8 numbers of bits together are called byte.

**Conversion Technique:**

**From Decimal number to Binary Number:**

**Example 3.1:** $(10)_{10}= (?)_2$

Answer:

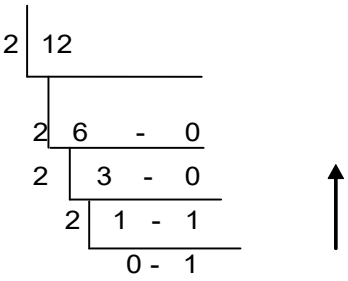



Therefore, $(10)_{10}= (1010)_2$

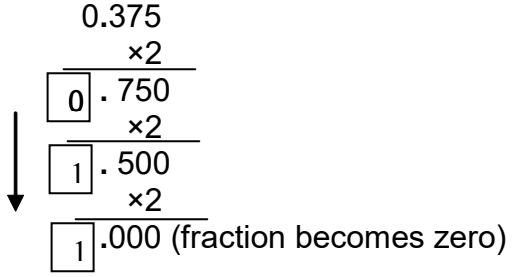
**Example 3.2:** $(12.375)_{10}= (?)_2$

Answer:

1st step:




2nd step:




Therefore $(0.375)_{10}= (0.011)_2$ and $(12.375)_{10}= (1100.011)_2$

**From Decimal number to Binary Number:**

The order of the least-significant digit (right-most digit) is 2^0 (units or ones), the second right-most digit is of the order of 2^1 (tens), the third right-most digit is of the order of 2^2 (hundreds), and so on. For example,

$110 = 1x2^2 + 1x2^1 + 0x2^0$

**Example 3.3:** $(1010)_2 = (?)_{10}$

Answer: $(1010)_2 = 1x2^3+0x2^2+1x2^1+0x2^0$

$= 1X8+0x4+1x2+0x1$

$= 8+0+2+0$

$= (10)_2$

**Example 3.4:** $(1100.011)_2 = (?)_{10}$

Answer: $1100.011 = 1x2^3+1x2^2+0x2^1+0x2^0+0x2^{-1}+1x2^{-2+}1x2^{-3}$

$= 8+4+0+0+0+.25+.125$

$=12.375$

Therefore, $(1100.011)_2 = (12.375)_{10}$

## 3.3.3 Octal Number System

The symbols of Octal number system are 0,1,2,3,4,5,6,7.There are eight symbols. The order of the least-significant digit (right-most digit) is 8^0 (units or ones), the second right-most digit is of the order of 8^1 (eight), the third right-most digit is of the order of 8^2 (sixty four), and so on. For example,

$564 = 5x8^2 + 6x8^1 + 4x8^0$

Table 3.1 shows the equivalent octal numbers corresponding to their decimal numbers:

Table 3.1 : Decimal and equivalent Octal Number System

| DECIMAL NUMBER | OCTAL NUMBER | DECIMAL NUMBER | OCTAL NUMBER | DECIMAL NUMBER | OCTAL NUMBER |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 2 | 2 |
| 3 | 3 | 4 | 4 | 5 | 5 |
| 6 | 6 | 7 | 7 | 8 | 10 |
| 9 | 11 | 10 | 12 | 11 | 13 |
| 12 | 14 | 13 | 15 | 14 | 16 |
| 15 | 17 | 16 | 20 | 17 | 21 |
| 18 | 22 | 19 | 23 | 20 | 24 |
| 21 | 25 | 22 | 26 | 23 | 27 |
| 24 | 30 | 25 | 31 | 26 | 32 |
| 27 | 33 | 28 | 34 | 29 | 35 |
| 30 | 36 | 31 | 37 | 32 | 40 |

…… AND SO ON.

## 3.3.4 Hexadecimal Number System

The symbols of Hexadecimal number system are 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. There are sixteen symbols. The order of the least-significant digit (right-most digit) is 16^0 (units or ones), the second right-most digit is of the order of 16^1 (sixteen), the third right-most digit is of the order of 16^2 (two hundred fifty six), and so on. For example,

$564 = 5x16^2 + 6x16^1 + 4x16^0$

Table 3.2 shows the equivalent hexadecimal numbers corresponding to their decimal numbers:

Table 3.2 : Decimal and equivalent hexadecimal number system

| DECIMAL NUMBER | HEXADE XIMAL NUMBER | DECIMAL NUMBER | HEXADE XIMAL NUMBER | DECIMAL NUMBER | HEXADE XIMAL NUMBER |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 2 | 2 |
| 3 | 3 | 4 | 4 | 5 | 5 |
| 6 | 6 | 7 | 7 | 8 | 8 |
| 9 | 9 | 10 | A | 11 | B |
| 12 | C | 13 | D | 14 | E |
| 15 | F | 16 | 10 | 17 | 11 |
| 18 | 12 | 19 | 13 | 20 | 1A |
| 21 | 1B | 22 | 1C | 23 | 1D |
| 24 | 1E | 25 | 1F | 26 | 14 |
| 27 | 15 | 28 | 16 | 29 | 17 |
| 30 | 18 | 31 | 19 | 32 | 2A |
| 33 | 2B | 34 | 2C | 35 | 2D |
| 36 | 2E | 37 | 2F | 38 | 20 |
| 39 | 21 | 40 | 22 | 41 | 23 |

…AND SO ON.

## 3.3.5 Binary Coded Decimal (BCD)

Binary Coded Decimal (BCD) is a system for coding a decimal number in which each digit of a decimal number is represented individually by its binary equivalent. Since there are 10 digits in decimal number system, so we need 4 bits to represent each of these (0 to 9) decimal numbers in binary. BCD consists of four number of binary digits. For example, equivalent binary number of decimal number 2 is 10 and equivalent BCD is 0010. Some more examples of BCD are as follows:

| Decimal number | Binary Number | BCD |
|---|---|---|
| 5 | 101 | 0101 |
| 7 | 111 | 0111 |
| 8 | 1000 | 1000 |
| 9 | 1001 | 1001 |

From decimal number 0 to 9 there is nothing to observe for finding equivalent BCD other than four binary bits. From decimal number 10, there is a vital point to observe regarding the equivalent BCD. The binary equivalent of 10 is 1010 and though the binary equivalent of 10 consists of four numbers of binary bits, it does not represent the equivalent BCD. Because at decimal 10, there are two decimal digits (1 and 0) where each decimal digit may be converted to their equivalent BCD as follows:

$(10)_{10}$ = 1 0

= 0001 0000

It will violate the basic principle of BCD number system if the equivalent BCD of $(10)_{10}$ contains eight numbers of binary bits.

## 3.3.6 American Standard Code for Information Interchange (ASCII)

**ASCII** (/'æski/ ASS-kee), abbreviated from American Standard Code for Information Interchange, is a character encoding standard. **ASCII** codes represent text in computers, telecommunications equipment, and other devices.

## 3.3.7 Negative Number Representation in Binary System

There are three techniques to represent negative number in binary number system, namely:

1. Sign Magnitude Method
2. Signed 1s' Complement Method
3. Signed 2s' Complement Method

### ● 3.3.7.1 Sign Magnitude Method

In this method 0 is used as the sign of positive number and 1 is used as the sign of negative number. The procedure to find the equivalent binary number for a negative decimal number is as follows:

i. Find the equivalent binary number for the decimal number

ii. Put 0 as the sign bit (the MSB, bit number 8) if the number is positive, otherwise 1

iii. Separate the sign bit from the other bits by putting a comma (,)

For example,

$(10)_{10}$=0, 0001010

$(-10)_{10}$=1, 0001010

## ● 3.3.7.2 Signed 1s' Complement Method

The procedure to find the equivalent binary number for a negative decimal number is as follows:

i. Find the equivalent binary number for the decimal number

ii. Replace each 0 of the equivalent binary number for the said decimal number by 1 and each 1 by 0

iii. Separate the sign bit from the other bits by putting a comma (,)

For example,

$(10)_{10}$=0, 0001010

$(-10)_{10}$=1, 1110101

## ● 3.3.7.3 Signed 2s' Complement Method

The procedure to find the equivalent binary number for a negative decimal number is as follows:

i. Find the equivalent binary number for the decimal number by applying 1s' complement method.

ii. Add 1 with the result getting from 1s' complement method to get 2's complement.

iii. Separate the sign bit from the other bits by putting a comma (,)

For example,

$(10)_{10}$=0, 0001010

$(-10)_{10}$=1, 1110101

1

---

1, 1110110



**CHECK YOUR PROGRESS**

Q1. What dose BCD stand for ?

Q2. $(10)_{10} = (?)_{16}$

Q3. Base of Octal Number is.....

Q4. Can we find out the equivalent binary number for a negative decimal number?

## 3.4 COMPUTER ARITHMETIC AND THEIR IMPLEMENTATION

### A. Addition of Binary Number

1. Add $(111)_2$ with $(101)_2$

```
  111
  101        1+1 =0, Carry 1
 ----        1+0+Carry 1=0, Carry 1
 1100        1+1+ Carry 1=1, Carry 1
```

2. Add $(1011)_2$ with $(110)_2$

```
 1011
  110       1+0=1
 -----      1+1=0, Carry 1
 10001      0+1+Carry 1=0, Carry 1
            1+Carry 1 =0, Carry 1
```

### B. Subtraction of Binary Number

1. Subtract $(101)_2$ from $(111)_2$

```
  111
 -101       1-1 =0
 ----       1-0 =1
  010       1-1 = 0
```

2. Subtract $(110)_2$ from $(1011)_2$

1011
-110 1-0=1
101 1-1=0
0-1+Borrow 1=1 [after borrowing 1, it becomes 10, i.e. 2, so (2-1)=1.]

3. Subtract $(1011)_2$ from $(1101)_2$

1101
- 1011 1-1=0
0010 0-1+Borrow 1=1
0-0=0
1-1=0

**C. Addition of Octal Number**

1. Add 473 with 645

473
645 3+5 =8
1340 =10
=0+ Carry 1
7+4+ Carry 1=12
=14
=4+Carry 1
4+6+ Carry 1=11
=13
=3+Carry 1

2. Add 123 with 567

123
567 3+7=10=12=2+Carry 1
712 2+6+Carry 1=9=11=1+Carry 1
1+5+Carry 1=7

## D. Subtraction of Octal number

1. Subtract 473 from 645

```
  645
  473        5-3=2
 ----
  152        4-7+ Borrow 1=12-7=5
             [In Octal Number system Borrow 1 value is 8]
             5-4=1
```

2. Subtract 123 from 511

```
  511
  123        1-3+Borrow 1=9-3=6
  366        0-2+Borrow 1=8-2=6
 ----
             4-1=3
```

## E. Addition of Hexadecimal number

1. Add 473 with ADC

```
  4 7 3
  ADC        3+C              =3+12=15
 -----                        =5+ Carry 1
 1 5 B 5    7+D+ Carry 1     =7+13+Carry 1
                              =21
                              =1B
                              = B + Carry 1
            4+A+ Carry 1     =4+10+Carry1
                              =15
                              =5+Carry 1
```

2. Add 123 with 567

| | |
|---|---|
| 123 | |
| 567 | 3+7=10=A |
| 68A | 2+6=8 |
| | 1+5=6 |

**F. Subtraction of Hexadecimal number**

1. Subtract 473 from 645

| | |
|---|---|
| 645 | |
| 473 | 5-3=2 |
| 1D2 | 4-7+ Borrow 1=20-7=13=D |
| | [In Hexadecimal Number system Borrow 1 value is 16] |
| | 5-4=1 |

2. Subtract 123 with 511

| | |
|---|---|
| 511 | |
| 123 | 1-3+Borrow 1=17-3=14=E |
| 3EE | 0-2+Borrow 1=16-2=14=E |
| | 4-1=3 |

## 3.5 CONTROL AND DATA PATH

To perform the functional characteristics in a CPU, there are basically two sections, one is *control section* and another one is *data section*. Control unit, which is the only element in the control section, is responsible for providing different types of control signals to the data section where data section is nothing but the data path. A CPU basically consists of different types of registers (such as Accumulator, Temporary Register, Instruction Register, Flag Register, General Purpose Register, Program Counter,

Stack Pointer, Interrupt Block etc.), ALU, Memory unit, Control unit. There are some buses for interconnecting the different components of CPU. The registers, the ALU, and the interconnecting buses are collectively referred to as the **data path**. Each bit in data path is functionally identical. The data path is capable of performing certain operations on data items.

## 3.6 DATA PATH COMPONENTS

We already know that there are some buses for interconnecting the different components of CPU. Now the question is what is Bus? Collection of wires or distinct lines is called Bus. There are three types of bus, namely, Address Bus, Data Bus and Control Bus.

Data are stored in main memory against a memory location. Address bus is responsible for carrying the address of the main memory location from where the data can be accessed.

Data buses are used for transmission of data. Control Bus is used for providing different types of control signals, to indicate the direction of data transfer and to coordinate the timing of events during the transfer.

The other important components of data paths are as follows:

**Accumulator:** It is a very special register. Irrespective of the location, the immediate input data will be available at Accumulator. And after an arithmetic operation, the result will be available at the Accumulator.

**PC (Program Counter):** It is responsible for holding the address of the instruction to be executed next.

**IR (Instruction Register)**: It stores or holds the instruction which is current or executing.

**Instruction Cache**: 'Fast' memory where the next instruction comes from Reg [index].

**Arithmetic Logic Unit (ALU)**: It performs all the arithmetic and logical operations.

**Data Cache**: Data read from or written to 'fast' memory.

**Multiplexer**: Multiplexer has many Inputs and single out put line. So, we can say that multiplexer is responsible for selecting one output from multiple inputs based upon control signal(s).

**Single-Cycle Data Path**: Each instruction executes in one clock cycle.

**Multi-Cycle Data Path**: Each instruction takes multiple clock cycles.

Types of elements in the Data path are:

1. State element:
   - i. A memory element, i.e., it contains a state e.g., program counter, instruction memory
2. Combinational element:
   - i. Elements that operate on values e.g., adder, ALU
3. Now, we will look at data path elements required by the different classes of instructions such as
   - i. Arithmetic and logical instructions
   - ii. Data transfer instructions
   - iii. Branch instructions



**CHECK YOUR PROGRESS**

**Q5.** To perform the functional characteristics of a CPU, basically how many types of section are there?

**Q6.** How many types of buses are there in a computer?

**Q7.** What is Program Counter?

**Q8.** By whom Data read's from or written to 'fast' memory?

**Q9.** Multiplexer has ..... inputs and ..... output line

**Q10.** In Hexadecimal Number system Borrow 1 value is .....

**Q11.** The registers, the ALU, and the interconnecting buses are collectively referred to as the.....

## 3.7 DESIGN OF ALU AND DATA PATH

The ALU (Arithmetic-Logic Unit, or Patterson and Hennessy's data path) performs all the arithmetic and logical operations. There is a communication path provided by data buses between ALU and storage elements. The ALU processor is normally composed of single element (one-bit) processors (primarily adders). These single element processors are assembled together to form a desired bit number processor. Some other processing elements such as a shifter are also included to it. An example of Bit Slicing is shown in the figure below for MIPS computer:

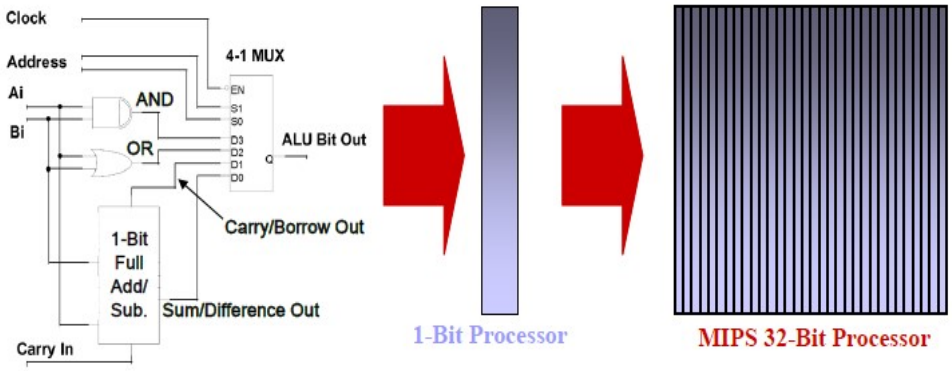



**Figure 3.1: Bit Slicing for MIPS computer**

ALU needs some components such as:

1. Registers to store arguments and results
2. Buses to carry data from registers to the ALU and results back to the register unit
3. Two memory access units, with associated buses
4. An instruction fetch unit to get instructions from computer memory as needed. This includes a program counter, which always points to the address of the next instruction to be accessed.
5. A second path to memory to obtain data to be used in the program and to store data back into memory as required.
6. A control unit that tells the ALU what to do.

The typical ALU is associated with the other block of a typical CPU is shown below in figure 3.2.

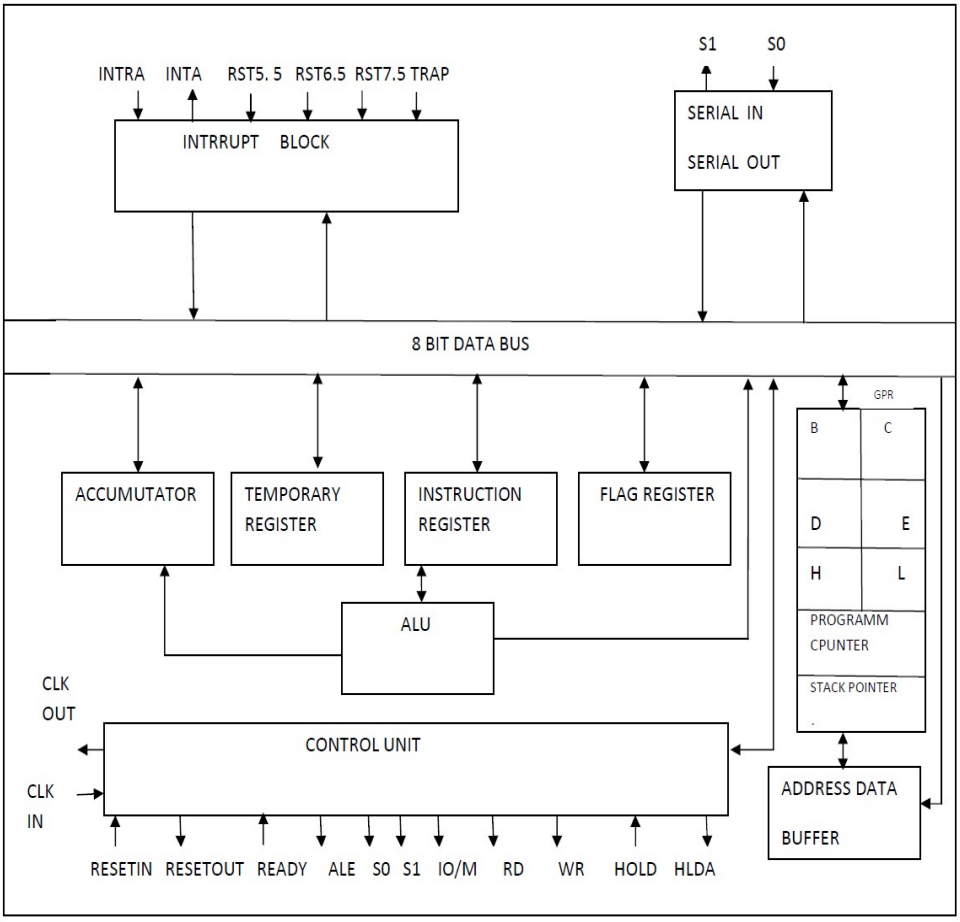



**Figure 3.2: A typical ALU is associated with the other block of a typical CPU**

The design of data path with ALU is distinctly realized from the above figure. ALU is connected with:

1. Accumulator. Data can transmit from ALU to Accumulator.
2. Instruction Register
3. Flag Register
4. And with the internal data bus.

Design of ALU is based on the points given below:

1. Instruction for a particular operation will come from instruction register directly.
2. Data on which operation is to be performed i.e. the operand will come to ALU from General Purpose Register (GPR) (or from memory location).

3. Either arithmetic or logical operation, which one is required, will be done by the different types of circuit (eg. Adder, Subtractor, Multiplier, Level Shifter, All logic gates etc.) of which ALU consists.
4. After an operation result the result will be available at Accumulator.
5. After an arithmetic operation done by ALU, different types of information about the result such as whether the result is positive number or negative number, whether the result has some value other than zero, whether there is some carry that has occurred during the operation or not etc. will be given by the Flag Register.

## 3.8 CONTROL UNIT DESIGN

Control unit is responsible for providing different types of control signals to the data section. The figure 3.2 shows that the control unit is directly connected to the Internal Data Bus. The design of a control unit demands a connection with crystal oscillator to produce suitable clock frequency for the processor. The other signals to be supplied by the control unit are as follows:

**RESET IN:** RESET signal is one by the application of which a digital circuit gets its proper output according to the status of its Input. The CPU is held at reset condition as long as RESET is supplied.

**RESET OUT:** To indicate the reset status of the CPU, it is used.

**READY:** To indicate whether the peripheral device is ready to give response, READY is used by the processor.

**ALE:** ALE stands for Address Latch Enable. It is used to demultiplex the address and data buses which are in multiplexed mode.

**S0, S1:** These are used to indicate different types of operation as shown below:

| S0 | S1 | operation |
|---|---|---|
| 0 | 0 | HALT |
| 0 | 1 | WRITE |
| 1 | 0 | READ |
| 1 | 1 | FETCH |

**IO/M:** IO stands for Input/output and M stands for memory. IO is a high enable signal whereas M is a low enable signal. If IO/M=1 then Input/output will be selected for data transmission, otherwise memory device.

**RD:** RD means READ operation.

**WR:** WR means WRITE operation.

**HOLD**: Say, another device is requesting for the address and data buses.

To indicate this condition HOLD signal is used.

**HLDA:** This signal is for acknowledging the HOLD signal.



**CHECK YOUR PROGRESS**

**Q12.** To demultiplex the address and data buses which are in multiplexed mode, which control signal is used?

**Q13.** To indicate the reset status of the CPU, which control signal, is used?



## 3.9 LET US SUM UP

- In digital computer there is a special number system to represent the data. This number system is called the Binary Number system
- There are some other number systems, namely:
    - Decimal (Base is 10)
    - Octal number (Base is 8)
    - Hexadecimal number (Base is 16)
- There are three techniques to represent negative number in binary number system, namely:
    - Sign Magnitude Method
    - Signed 1s' Complement Method
    - Signed 2s' Complement Method
- The registers, the ALU, and the Interconnecting Buses are collectively referred to as the data Path.
- There are three types of bus, namely: Address bus, Data bus and control bus.

- Address Bus: Data are stored in the main memory against a memory location. Address bus is responsible for carrying the address of the main memory location from where the data can be accessed.
- Data Bus: Data buses are used for transmission of data.
- Control Bus: Control Bus is used for providing different types of control signals, to indicate the direction of data transfer and to coordinate the timing of events during the transfer.
- The ALU (Arithmetic-Logic Unit, or Patterson and Hennessy's data path) performs all the arithmetic and logical operations.
- ALU is connected with :
    - Accumulator. Data can transmit from ALU to Accumulator.
    - Instruction Register
    - Flag Register
    - And with the internal data bus.
- Control unit is responsible for providing different types of control signals to the data section such as RESET IN, RESET OUT, HOLD, HLDA, READY, ALE, IO/M, R/W etc.



## 3.10 FURTHER READINGS


1) Mano, M. M. (2006). Computer systems architecture.
2) Hamacher, V. C., Vranesic, Z. G., Zaky, S. G., Vransic, Z., & Zakay, S. (1984). Computer organization (Vol. 3). New York et al.: McGraw-Hill.




## 3.11 ANSWERS TO CHECK YOUR PROGRESS

**Ans to Q No 1:** Binary Coded Decimal

**Ans to Q No 2:** A

**Ans to Q No 3:** 8

**Ans to Q No 4:** Yes

**Ans to Q No 5:** Two

**Ans to Q No 6:** Three

**Ans to Q No 7:** Program Counter holds the address of the instruction to be executed next.

**Ans to Q No 8:** Data Cache

**Ans to Q No 9:** Many, Single

**Ans to Q No 10:**16

**Ans to Q No 11:** Data Path

**Ans to Q No 12:** ALE

**Ans to Q No 13:** RESET OUT



## 3.12 MODEL QUESTIONS

Q1. Find the following:

i. $(134.56)_{10} = (?)_2$

ii. $(101101.11)_2 = (?)_{10}$

iii. $(543)_{10} = (?)_{16}$

Q2. Find the following:

i. Add $(101)_2$ with $(11)_2$

ii. Add $(467)_8$ with $(1231)_8$

iii. Add $(BCF)_{16}$ with $(101)_{16}$

Q3. Find the following:

i. Subtract $(11)_2$ from $(101)_2$

ii. Subtract $(467)_8$ from $(1231)_8$

iii. Subtract $(101)_{16}$ from $(BCF)_{16}$

Q4. Find out the binary equivalent number of (-11) by following all three negative number representation technique of binary number

Q5. Define Data Path. Write all the important components of Data Path.

Q6. What is Bus? Explain in detail.

Q7. Explain the design of an ALU with its data path

Q8. Write about the entire control signal associated with the control unit in detail.