**ASSIGNMENT 1**

Name: Kulsoom Khurshid

Reg#: Sp20-BCS-044

Course: Microprocessor and Assembly Language

**Question 1)** Perform the following conversions. Show all your steps.

1. **(3D7E)­16 = ( )10 = ( )2**

D = 13, E = 14

= 163 x 3 + 162 x 13 + 161 x 7 + 160 x 14

= 12288 + 3328 + 112 + 14

= 15742

(3D7E)­16 = (15742)10

(15742)­10 = ( )2

|  |  |
| --- | --- |
| **2** | **15742** |
| **2** | **7871 - 0** |
| **2** | **3935 - 1** |
| **2** | **1967 - 1** |
| **2** | **983 - 1** |
| **2** | **491 - 1** |
| **2** | **245 - 1** |
| **2** | **122 - 1** |
| **2** | **61 - 0** |
| **2** | **30 - 1** |
| **2** | **15 - 0** |
| **2** | **7 - 1** |
| **2** | **3 - 1** |
| **2** | **1 - 1** |

(3D7E)­16 = (15742)10 = (11110101111110)2

1. **(764)­10 = ( )2 = ( )16**

|  |  |
| --- | --- |
| **2** | **764** |
| **2** | **382 - 0** |
| **2** | **191 - 1** |
| **2** | **95 - 1** |
| **2** | **47 - 1** |
| **2** | **23 - 1** |
| **2** | **11 - 1** |
| **2** | **5 - 1** |
| **2** | **2 - 1** |
| **2** | **1 - 0** |

(764)­10 = (1011111110)2 = ( )16

(1011111110)2 = ( )16

0010 = 2, 1111 = 15-F, 1110 = 14-E

**=** (764)­10 = (1011111110)2 = (2FE)16

**Question 2)** Find two’s complement binary representation of -39 using

1. **8-Bit**

Value = -39

Binary Representation:

|  |  |
| --- | --- |
| **2** | **39** |
| **2** | **19 - 1** |
| **2** | **9 - 1** |
| **2** | **4 - 1** |
| **2** | **2 - 0** |
| **2** | **1 - 0** |

(39)10 = (100111)2

8-bit representation of 39 = 00100111

1’s Complement of 00100111 = 11011000

2’s complement:

11011000

+ 1

11011001

(-39)10 in 2’s complement = 11011001

The left most 1 represent the negative sign.

1. **16-Bit**

Value = -39

Binary Representation:

|  |  |
| --- | --- |
| **2** | **39** |
| **2** | **19 - 1** |
| **2** | **9 - 1** |
| **2** | **4 - 1** |
| **2** | **2 - 0** |
| **2** | **1 - 0** |

(39)10 = (100111)2

16-bit representation of 39 = 0000000000100111

1’s Complement of 0000000000100111 = 1111111111011000

2’s complement:

1111111111011000

+ 1

1111111111011001

(-39)10 in 2’s complement = 1111111111011001

The left most 1 represent the negative sign.

**Question 3)** Convert decimal real numbers to 32-bit floating point representation using IEEE 754 format.

1. **0.15625**

0.15625 x 2 = 0.3125

0.3125 x 2 = 0.625

0.625 x 2 = 1.25

0.25 x 2 = 0.5

0.5 x 2 = 1

The binary representation of (0.15625)10 = (0.00101)2 = 1.01 x 2-3

Adding -3 to 127 = -3 + 127 = (124)10

Binary representation of 124:

|  |  |
| --- | --- |
| **2** | **124** |
| **2** | **62 - 0** |
| **2** | **31 - 0** |
| **2** | **15 - 1** |
| **2** | **7 - 1** |
| **2** | **3 - 1** |
| **2** | **1 - 1** |

(124)10 = (1111100)2

Since exponent is 8 bits so we will add 0 to the left of this binary number; (01111100)2

|  |  |  |
| --- | --- | --- |
| **Sign (1-bit)** | **Exponent (8-bits)** | **Mantissa (23-bits)** |
| 0 | 01111100 | 01000000000000000000000 |

1. **11.1**

|  |  |
| --- | --- |
| **2** | **11** |
| **2** | **5 - 1** |
| **2** | **2 - 1** |
| **2** | **1 - 0** |

(11)10 = (1011)2

0.1 x 2 = 0.2

0.2 x 2 = 0.4

0.4 x 2 = 0.8

0.8 x 2 = 1.6

(0.1)10 = (0001)2­

The binary representation of (11.1)10 = (1011.0001)2 = 1.0110001 x 23

Adding 3 to 127 = 3 + 127 = (130)10

Binary representation of 130:

|  |  |
| --- | --- |
| **2** | **130** |
| **2** | **65 - 0** |
| **2** | **32 - 1** |
| **2** | **16 - 0** |
| **2** | **8 - 0** |
| **2** | **4 - 0** |
| **2** | **2 - 0** |
| **2** | **1 - 0** |

(130)10 = (10000010)2

|  |  |  |
| --- | --- | --- |
| **Sign (1-bit)** | **Exponent (8-bits)** | **Mantissa (23-bits)** |
| 0 | 10000010 | 01100010000000000000000 |

**Question 4)** Describe segmented memory model and flat memory model?**Segment Memory Model**

Definition

The Segment memory model divides the system memory into independent segments that are referenced by pointers that are stored in the segment registers. Every segment is responsible to hold a specific type of data. One of the segment can be used for instruction codes, another segment can be used to store the data elements and the third segment maintain the program stack.

Types of Memory Segments

Following are the various memory segments.

1. **Data Segment**

This segment is represented by two sections i.e. .data and .bss section. The .data section is used for memory declaration region, which holds the data elements for the program. Once the data elements are declared, this section cannot be extended further which means it is static throughout the program.

The .bss section is also static but it contains buffer for data elements that will be declared later in the program. This memory buffer is zero-filled.

1. **Code Segment**

This segment is represented by .text section. It is used to create an area in the memory for instruction codes. This section is also fixed which means it cannot be expanded further.

1. **Stack**

This segment is used to hold the data values that are passed to functions and procedures available within the program.

**Flat Memory Model**

Definition

Flat memory model also known as linear memory model that states to the memory addressing paradigm in which memory is the single contiguous address space for the program. The CPU can directly address the available memory locations without resorting through the memory segmentation or any paging schemes. It is used to facilitate the operating system functionality, for protecting the resources, for increasing memory capacity or multitasking, but the purpose of flat memory is to have linear space memory, sequential and contiguous.

Features of Flat memory

Following are some features of flat memory

* It has clean design and simple interface for users.
* Due to uniform access speed the flexibility is greatest.
* For simple controller application, minimum CPU and hardware is required.
* The execution speed is maximum.
* This model is not suitable for multitasking or general computing unless the system is enhanced with additional memory management. But this is usually the case in modern CISC processors.

**Question 5)** Define pipelining and superscalar Processor in detail. Discuss its types of pipelining and merits and de-merits?

**Pipelining**

Instructions are considered the smallest execution packet of a program. Every instructions has one or more operations. Simple scalar processor executes one or more execution per cycle that contains only one instruction. These instructions execute in a sequence of phase to get the expected results. This sequence is as follows

IF (Instruction Fetch) > ID (Instruction Decode) > AG (Address Generate) > DF (Data Fetch) > EX (Execution) > WB (Write back).

It is not necessary that all instructions had to go through these phases but most of them do. These steps use different hardware functions. Through pipelining these different phases can be performed concurrently. These phases are considered independent between different operations and can be overlapped. Thus, multiple operations can be performed simultaneously with each operation being in its own independent phase.

**Superscalar processor**

Processors are equipped in such a way that multiple processing units can handle several instructions in parallel at every stage. Several instructions may start the execution in the same clock cycle and it is said that process may use multiple issue. Such processors can achieve an instruction execution throughput of more than one instruction per cycle. They are known as Superscalar Processors.

**Types of pipelining**

Following are the types of pipelining.

1. **Arithmetic pipelining**

It is used for performing high speed floating point addition, multiplication and division. ALUs are built for performing parallel arithmetic computations on different data formats.

1. **Instruction pipelining**

The instructions are pipelined and current instruction is overlapped by the execution of the subsequent instruction.

1. **Processor pipelining**

The processors are pipelined to process the same data stream. The data stream is processed by the first processor and the result is stored in the memory block. The result in the memory block is accessed by the second processor. The second processor reprocesses the result and passes the refined result to the third processor and so on.

1. **Unifunction Vs. Multifunction Pipelining**

The pipeline performing the precise function every time is unifunctional pipeline.

The pipeline performing multiple functions at a different time or multiple functions at the same time is multifunction pipeline.

1. **Static vs Dynamic Pipelining**

The static pipeline is unifunctional. The static pipeline executes the same type of instructions continuously.

Dynamic pipeline performs several functions simultaneously. It is a multifunction pipelining.

1. **Scalar vs Vector Pipelining**

Scalar pipelining processes the instructions with scalar operands.

The vector pipeline processes the instruction with vector operands.

**Merits**

Following are the advantages.

* The throughput of the system is increased.
* In every clock cycle, a new instruction execution is completed.
* Multiple instructions can be executed concurrently.

**Demerits**

Following are the disadvantages.

* Data dependency: when one instruction depends on another instruction for data.
* Memory delay: the number of cycle the pipeline has to stall
* Branch delay: the delay of cycle is branch delay
* Resource limitation: two instructions requesting the same resource.