**Personal computer**: Single end user computer built for personal use.

**Server**: Multi-user computer built to host and serve resources over a network.

**Super computer**: Purpose built computer made of an array of small computer to solve complex problems.

**Embedded computer**: Task built computer where software interacts directly with hardware.

**Personal mobile device**: Portable wireless computer capable of accessing the internet as well as manipulating many types of data by use of purpose-built applications.

**Cloud computing**: The consolidation of many servers that provide software, infrastructure, or computing resources as a service.

What are the three levels of program code? Describe each level. (15 Points)

**High level language**: Closest to human language which are more accessible to read and write.

**Low level language**: Language that the computer can understand by instruction set that are not readily obvious to humans.

**Machine language**: Programs that are written in binary which is what high level and low level languages are reduced to before a computer can execute a program.

Discuss five components of a computer? Give at least two examples for each component. (10 Points)

**Input**: Mouse and keyboard.

**Output**: Monitor and speaker.

**Memory**: Hard disk and ram.

**Control**: CPU command of the data path by instruction set.

**Datapath**: Bus and registers.

State Amdahl’s law. (5 Points)

Overall performance improvement gained by optimizing a single part of a system is limited by the fraction of time that the improved part is used.

a. 4510

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

**32 + 0 + 8 + 4 + 0 + 1**

**0010 = 2, 1101 = D**

Consider three different processors P1, P2, and P3 executing the same instruction set. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions for each processor? (20 Points)

|  |  |  |  |
| --- | --- | --- | --- |
|  | P1 | P2 | P3 |
| Clock Rate | 3 GHZ | 2.5 GHZ | 4 GHZ |
| CPI | 1.5 | 1 | 2.2 |

=

CPU clock cycles = instruction count \* clock cycles per instruction

CPU Time = instruction count \* clock cycles per instruction \* clock cycle time =



which is faster: P1 or P2?

P2 is faster with 2.0E6

What is the global (average) CPI for each implementation?

P1 CPI = 2.6

P2 CPI = 2

Find the clock cycles required in both cases.

P1 clock cycles = 2.6E6

P2 clock cycles = 2.0E6

Consider two different implementations of the same ISA. The instructions can be divided into classes as follows (Classes A, B, C, D): (35 Points)

P1 and P2 have clock rate of 2.5 GHZ and 3.0 GHZ, respectively. Given a program of Dynamic Instruction count of 1.0E6 instructions divided into classes as follows: 10% class A, 20% class B, 50%class C and 20% class D.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | A | B | C | D |
| CPI (P1) | 1 | 2 | 3 | 3 |
| CPI (P2) | 2 | 2 | 2 | 2 |



LSL – multiply register by 2i

LSR – divides register by 2i

What will be the value of X1 after running the following instruction: LSL X1, X2, #2. Assume that X2 = 4. (show the steps of calculation) (10 points)

X1 = 4 \* 22 = 16

X2 = 0X0000 0000 0000 0004

4 = 0000 0100

1ST SHIFT = 0000 1000

2ND SHIFT = 0001 0000

X1 = 0001 0000

Convert following assembly instruction to 32 bit machine code and then change it to Hexadecimal format. (25 Points)

* 1. LDUR X10, [X5, #16]
  2. SUB X12, X14, X15
  3. LSR X11, X19, #2

1. D-Format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OPCODE (11 bit)** | **ADDRESS (9 bit)** | **OP2 (2 bit)** | **Rn (5 bit)** | **Rt (5 bit)** |
| **198610** | **1610** | **010** | **510** | **1010** |
| **1111 1000 0102** | **0 0001 00002** | **002** | **00 1012** | **0 10102** |
| **F 8 4 1 0 0 A A16** | | | | |

1. R-Format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OPCODE (11 bit)** | **Rm (5 bit)** | **SHAMT (6 bit)** | **Rn (5 bit)** | **Rd (5 bit)** |
| **162410** | **1510** | **010** | **1410** | **1210** |
| **0110 0101 1002** | **0 11112** | **0000 002** | **01 1102** | **0 1100** |
| **6 5 8 F 0 1 C C16** | | | | |

1. I-Format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OPCODE (11 bit)** | **Rm (5 bit)** | **SHAMT (6 bit)** | **Rn (5 bit)** | **Rd (5 bit)** |
| **169010** | **010** | **210** | **1910** | **1110** |
| **1101 0011 0102** | **0 00002** | **0000 102** | **10 0112** | **0 10112** |
| **D3400A6B16** | | | | |

What will happen to X2 after running the following instruction: LDUR X2, [X5, #0]. Assume that X5 =

0X8000000000004000 and locations 0X8000000000004000 through 0X8000000000004007 contain

0X00, 0X00, 0X00, 0X00, 0X00, 0X00, 0X02, and 0X23, respectively. (10 points)

**Instruction is to LoaD Unscaled Register X2 (64 bit) with the contents of the memory pointed at by X5 + 0 (i \* 8 bit memory contents). The 8 memory slots from 4000 to 4007 contain 0, 0, 0, 0, 0, 0, 2, 35 in decimal which is being loaded to X5.**

**X2 = 0000 0000 0000 022316**

Convert C++ code snippet to LEGv8 assembly code. The following variables x, y, and z are associated with registers X19, X20, and X21 respectively, and base address of the array A is in X22. Comment the code. (15 Points)

x = x + y;

z = x + 4;

A[8] = A[3] + z;

ADD X19, X19, X20// x = x + y

ADDI X21, X19, #4// z = x + 4

LDUR X9, [X22, #24] // X9 = A[3]

ADD X9, X9, X21// X9 = A[3] + z

STUR X9, [X22, #64] // A[8] = X9

Convert C++ code snippet to LEGv8 assembly code. The following variables x, y, and z are associated with registers X19, X20, and X21, respectively, and base address of the array d is in X22. Comment the code. (30 Points)

if (x > y) z = y + 4;

else z = y - 16;

CMP X19, X20//compare x and y

B.LT L1// if x is less than y, branch to L1

ADDI X21,X20, #4// z = y + 4

L1: SUBI X21, X20, #16 // z = y – 16

for (i=0; i<x; i++)

{y = d[i] + z;}