1.2 Can a higher-level programming language instruct a computer to

compute more than a lower-level programming language?

*Answer*:

No. Anything that can be computed, can be computed by a computer provided it has enough time and enough memory. So we can only buy a faster computer, a monitor with higher resolution, or a nice sound system, but not a computer that can compute more.

1.4 Name one characteristic of natural languages that prevents them from being used as programming languages.

*Answer*:

Natural language is filled with ambiguity. Computers need specific instructions to follow, but such ambiguity would be unacceptable in instructions provided to a computer.

1.10 Name three characteristics of algorithms. Briefly explain each of

these three characteristics.

*Answer*:

Firstly, definiteness. Every step of the algorithm is precisely stated without ambiguity.

Secondly, effective computability. Every step of the algorithm can be carried out by a computer.

Thirdly, finiteness. The procedure of the algorithm can terminate.

1.16 Name at least three things specified by an ISA

*Answer*:

Firstly, the ISA specifies the interface between the computer program directing the computer hardware and the hardware carrying out those directions. In other words, the opcodes.

Secondly, the ISA specifies the acceptable representations for operands which are called data types. And the computer can perform operations on that representation.

Thirdly, the ISA specifies the mechanisms that the computer can use to figure out where the operands are located. These mechanisms are called addressing modes.

1.18 How many ISAs are normally implemented by a single microarchitecture? Conversely, how many microarchitectures could exist

for a single ISA?

*Answer*:

A microstructure can only implement one ISA. But an ISA can be implemented on a variety of microstructures.

2.4 Given n bits, how many unsigned integers can be represented with the n bits? What is the range of these integers?

*Answer*:

2^n unsigned integers can be represented and range from 0 to 2^n-1

2.8 a. What is the largest positive number one can represent in an 8-bit 2’s complement code? Write your result in binary and decimal.

b. What’s the greatest magnitude negative number one can represent in

an 8-bit 2’s complement code? Write your result in binary and decimal.

c. What is the largest positive number one can represent in n-bit 2’s

complement code?

d. What is the greatest magnitude negative number one can represent in

n-bit 2’s complement code?

*Answer*:

a.the largest positive number is 127(D) 01111111(B) (based on 2^7-1)

b.the greatest magnitude negative number is -128(D) 10000000(B) (-2^7)

c.the largest positive number is 2^(n-1)-1

d.the greatest magnitude negative number is -2^(n-1)

2.17 Add the following 2’s complement binary numbers. Also express the

*Answer* in decimal.

a. 01 + 1011

b. 11 + 01010101

c. 0101 + 110

d. 01 + 10

*Answer*:

1. 0001 + 1011 = 1100(B) -4(D)
2. 1111 1111 + 01010101 = 0101 0100(B) 84(D)
3. 0101 + 1110 = 0011(B) 3(D)
4. 01 + 10 = 11(B) -1(D)

2.20 The following binary numbers are four-bit 2’s complement binary

numbers. Which of the following operations generate overflflow? Justify

your *Answer* by translating the operands and results into decimal.

a. 1100 + 0011

b. 1100 + 0100

c. 0111 + 0001

d. 1000 - 0001

e. 0111 + 1001

*Answer*:

1. No overflow -4 + 3 = -1 (1100 + 0011 = 1111B)
2. No overflow -4 + 4 = 0 (1100 + 0100 = 0000B)
3. Overflow 7 + 1 = 8 (0111 + 0001 = 1000B = -8D)
4. Overflow -8 - 1 = -9 (1000 - 0001 = 0111B = 7D)
5. No overflow 7 + (-7) = 0 (0111 + 1001 = 0000B)

2.34 Compute the following:

a. NOT(1011) OR NOT(1100)

b. NOT(1000 AND (1100 OR 0101))

c. NOT(NOT(1101))

d. (0110 OR 0000) AND 1111

*Answer*:

1. 0100 OR 0011 = 0111
2. NOT(1000 AND 1101) = NOT(1000) = 0111
3. 1101
4. 0110 AND 1111 = 0110