Introduction to computers
Solved Exercises

Exercise 1. Convert the following 16-bit binary number to hexadecimal: 1101001011101010

Solution:

Every 4 binary digits form a hexadecimal digit: 1101 0010 1110 1010 = 0xD2EA

Exercise 2. Convert the following hexadecimal number to binary: 0xF73AB591

Solution:

0x7F3AB591 = 0111 1111 0011 1010 1011 0101 1001 0001

Exercise 3. Considering a hypothetical computer with the following characteristic:

- Size of a memory location: 16 bits
- Instruction size: 16 bits
- Operation code: 3 bits
- Number of general purpose registers: 4 (2 bits)R0 (00)
  - R1 (01)
  - R2 (10)
  - R3 (11)

Instruction	Description
000010010XXXXXXX	Adds the content of register 00 to 10 and stores the result in register 01
0010100000000101	Store in register 01 the value 00000000101
0100100000001001	Store in register 01 the value stored in the memory location 00000001001
0110100000001001	Store in the address 00000001001 the content of register 01
1000000000001001	Jump to execute the instruction store in address 0000000001001
1010100000001001	If register 01 is equal to register 00 jump to execute the instruction stored in 000001001

Write a program using the above instructions to calculate the sum of the first 10 natural numbers: 1 + 2 +3 + 4 ... + 10. The result will be stored in memory position 0010000.

Solution:

Assuming that the program is stored starting from memory location 0:

Address	Instruction
0000	0010000000000000
0001	0010100000000001
0010	0011000000000001
0011	0011100000001011
0100	1011011000001000
0101	0000000100000000
0110	0001010010000000
0111	1000000000000100
1000	0110000000010000

Exercise 4. Consider a hypothetical computer with a word width of 20 bits with 60 registers that addresses the memory by bytes. Answer the following questions:

a) How many bits are used for the memory addresses?
b) What is the size of the registers?
c) How many bits are stored in each memory location?
d) How many memory locations can be addressed? Express the result in KB.
e) How many bits are needed to identify the registers?

Solution:

a) 20 bits are used, which coincides with the word width of the computer.
b) The register size coincides with the word width, 20 bits.
c) In each memory position is stored a byte because the memory is addressed by bytes.
d) You can address 2^{20} memory locations. In each memory position is stored a byte, then the memory size is 2^{20} bytes = $2^{20} / 2^{10} = 2^{10}$ = 1024 KB.
e) As there are 60 registers, it is necessary to have $\lceil \log_2 60 \rceil = 6$ bits.

Exercise 5. Consider a hypothetical computer with 100 registers that addresses the memory per bytes. In this computer you can address at most 64 KB of memory. Assuming that the word size of this computer matches the number of bits used for memory addressing. Answer the following questions:

a) How many bits are used for the memory addresses?
b) What is the size of the registers?
c) How many bits are stored in each memory location?
d) If the computer was 32 bits, what would be the size of the maximum addressable memory? Express the result in MB.
e) How many bits are needed to identify the registers?

Solution:

a) The computer has 64 KB / 1 byte = 64 Kpositions of memory. Each position stores a byte, because the memory is addressed by bytes. To address 64 Kpositions = (64×1024) you need $\log_2 (64 \times 1024) = 16$ bits.
b) The register size coincides with the word width, which in this case coincides with the number of bits needed to represent a memory address, i.e. 16 bits.
c) In each memory position is stored a byte.
d) 2^{32} bytes = $2^{32} / 2^{20} = 2^{12}$ MB.
e) The computer has 100 registers, you need $\lceil \log^2 100 \rceil = 7$ bits