Homework #1

Katerina Chinnappan – kachinna@ucsc.edu

Due: October 2, 2016 @ 23:99

Section 01-C

1) (3pts) Can a higher-level programming language instruct a computer to computer more than a lower-level programming language?

Answer: No, a computer cannot understand a higher or lower level programming language. A computer can understand only one language – the machine language. High and low level programming are translated into that machine language and only then the computer is able to understand.

2) (3 pts) What difficulty with analog computers encourage computer designers to use digital designs?

Answer: It is very difficult to build a very accurate analog computer/machine.

3) (5 pts) Name three characteristics of algorithms. Briefly explain each of these three characteristics

Answer: 1) Definiteness: This term is used to describe the notion of a procedure and steps. Each step is described in particular detail and nothing is missed. 2) Effective Computability: This term is used to describe that each step/procedure can be carried out by a computer in detail. 3)Finiteness: This term is used to describe that any kind of procedure can end/terminate/stop. Basically one problem can be solved in different ways. It can be solved with one algorithm that works efficiently and faster and with another algorithm that takes more steps, hence solves it slower. Usually, when a computer allows more than one thing to solve the same problem, it uses the faster algorithm and terminates the rest of them.

4) (5 pts) Identify one advantage of programming in a higher-level language compared to a lower level language. Identify one disadvantage.

Answer: One advantage of programming in a higher-level language is that it is closer to a human being. It is easier for us to understand and program in the language rather than a lower level language. It consists of English like structures and keywords so hence it is easier for us to understand it. It is more difficult to understand a low level language and it is difficult to modify it. One big disadvantage of a higher-level language is that it has to get translated into machine language and only then execute. Low-level language on the other hand directly executes on computer.

5) (5 pts) Name at least three things specified by the ISA.

Answer: 1) The ISA particularly specifies a set of instructions that a computer can carry out; what particular operations the computer can perform and what particular elements/data are needed by the operation. 2) The ISA also specifies the representation for operands. These are called data types. 3) The ISA also specifies the mechanisms that a computer can use to figure out where the particular operands are located. These are called addressing modes.

6) (5 pts) Briefly describe the difference between an ISA and a micro architecture.

Answer: ISA is used to describe the type/size of an operand, how exceptions are being handled, basically ISA describes the interface of a computer. Microarchitecture on the other hand is used to describe for example pipelining, and implementation of the ISA. So basically the ISA is what we see(interface) and microarchitecture are the components/details.

7) (12 pts) Convert the following numbers to unsigned binary.

a. 26 b. 49 c. 255 d. 129

a) **Answer:** 26 → 2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32

26 – 16 = 10. 10 – 8 = 2. 2 – 2 = 0. → Binary notation: **11010**

b) **Answer:** 49 →2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32

49-32 = 17. 17-16 = 1. 1-1 = 0. → Binary notation: **110001**

c) **Answer:** 255→255/2 = 127 R1, 127/2 = 63 R1, 63/2 = 31 R1, 31/2 = 15 R1, 15/2 = 7 R1, 7/2 = 3 R1, 3/2 = 1 R1, ½ = 0 R1. →Binary notation: **11111111**

d) **Answer:**129→129/2 = 64 R1, 64/2 = 32 R0, 32/2 = 16 R0, 16/2 = 8 R0, 8/2 = 4 R0, 4/2 = 2 R0, 2/2 = 1 R0, 1/2 = 0 R1. →Binary notation: **10000001**

8) (25 pts) Convert the following unsigned binary numbers to base 10 and hexadecimal.

a. 00101010 b. 00111111 c. 10000000 d. 11101001 e. 00001001

a) **Answer:** 00101010 →decimal: 0 * 2^7 + 0 * 2^6 + 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = 0 +0 + 32 + 0+ 8 + 0 + 2 +0 = **42**

42→hex: 42/16 = 2 R 10, 10 = A →Hex notation: **2A**

b) **Answer:** 00111111→decimal: 0 * 2^7 + 0 * 2^6 + 1 * 2^5 + 1 * 2^4 + 1 * 2^3 + 1 * 2^2 + 1 * 2^1 + 1 * 2^0= 0 + 0 + 32 + 16 + 8 + 4 + 2 +1 = **63**

63→hex: 63/16= 3 R15, 15 = F →Hex notation: **3F**

c) **Answer:** 10000000→decimal: 1 * 2^7 + 0 * 2^6 + 0 * 2^5 + 0 * 2^4 + 0 * 2^3 + 0 * 2^2 + 0* 2^1 + 0 * 2^0 = **128**

128→hex: 128/16 = 8 R 0 →Hex notation: **80**

d) **Answer:** 11101001→decimal: 1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 = 128 + 64 + 32 + 8 + 1 = **233**

233→hex: 233/16 = 14 R 9, 14 = E →Hex notation: **E9**

e) **Answer:** 00001001→decimal: 0* 2^7 + 0 * 2^6 + 0 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 = 8 + 1 = **9**

9→hex: **9**

9) (12 pts) Convert 100 sub (10) to the following bases.

a. 3 b. 4 c. 5 d. 6

a) **Answer:** 100→base 3: 100/3 = 33 R1, 33/3 = 11 R0, 11/3 = 3 R2, 3/3 = 1 R0, 1/3 = 0 R1 →Base 3 notation: **10201**

b) **Answer:** 100→base 4: 100/4 = 25 R0, 25/4 = 6 R1, 6/4 = 2 R2, 2/4 = 0 R1 →Base4 notation: **1210**

c) **Answer:** 100→base 5: 100/5 = 20 R0, 20/5 = 4 R0, 4/5 = 0 R4 →Base 5 notation: **400**

d) **Answer:** 100→base 6: 100/6 = 16 R4, 16/6 = 2 R4, 2/6 = 0 R2→Base 6 notation: **244**

10) (12 pts) Convert the arbitrary base to base 10.

a. 210 sub (3) b. 321sub (4) c. 432sub (5)

a) **Answer:** 210 base 3→decimal: 2 * 3^2 + 1 * 3 ^1 + 0 * 3 ^0 = 18 + 3 + 0 = **21**

b) **Answer:** 321 base 4→decimal: 3 *4^2 + 2 * 4^1 + 1 * 4^0 = 48 + 8 + 1 = **57**

c) **Answer:** 432 base 5→decimal: 4 * 5^2 + 3 * 5^1 + 2 * 5^0 = 100 + 15 + 2 = **117**

11) (35 pts) Convert the following numbers to 8-bit 2's complement and perform the mathematical operation in binary.

a. -6+20 b. 67+30 c. 42-20 d. -44-23 e. 26-26

a) **Answer:** 1) convert positive 6 to binary→0000 0110
2)invert the bits→1111 1001
3)add 1→1111 1010
4)convert 20 to binary→0001 0100
5)add both numbers together→1111 1010
+ 0001 0100

10000 1110 →but it's 8 bit, so cross out the 1 and we get→**0000 1110 which is 14 in base 10.**

b) **Answer:** 1) convert 67 to binary→01000011
2)convert 30 to binary→00011110
3)add both numbers together→0100 0011
+ 0001 1110

0110 0001→**0110 0001 which is 97 in base 10.**

c) **Answer:** 1) convert 42 to binary→0010 1010
2)convert 20 to binary→0001 0100
3)subtract →0010 1010
- 0001 0100

0001 0110→**00010110 which is 22 in base 10.**

d) **Answer:** 1) convert (+)44 to binary→0010 1100
2)invert→1101 0011
3)add 1→1101 0100
4)convert 23 to binary→0001 0111
5)subtract→1101 0100
- 0001 0111

1000011→**1000011 which is -67 in base 10.**

e) **Answer:** 1) convert 26 to binary→00011010
2) subtract 26-26 in binary→0001 1010
- 0001 1010

0000 0000→**0000 0000 which is 0 in base 10.**