James Vrionis
1562232
1-17-2017 CMPE 12/L

Homework 1

1) No, a higher-level programming language cannot instruct a computer to compute more than a lower-level programming lang. Higer-level langs are converted to low-level langs before the Machine executes them. Instructions can be provided in either language.

2) Analog computers operate on a set of real numbers i.e $\mathbb{R}$ whereas digital computers quantize the signal into a finite number of values, which means they only work w/ the rational set i.e $\mathbb{Q}$. Analog computers were design-ed in theorey to solve problems that weren't solveable on digitial computeres, but this is not the case. Digital computers are more ideal for complex problems when given ∞ amount of time and menory, digital computers may solve real # problems.

3) Three characteristics of a step-by-step procedure that is guaranteed to terminate, s.t each step is precisely stated and carried out by the computer, A.K.A Algorithm.
   1) Definiteness; used to describe the notion that each step is precisely stated.
   2) Effective Computability: used to describe the notion that each step can be carried out by a computer
   3) Finiteness: used to describe the notion that the procedure terminates.

4) one advantage: High-level languages are at a distance from the underlying computer. At best, they are independent of the computer which the programs will execute on.

one disadvantage: High-level langs include slower speed, slower compiler time and more limited access to lower-level computer functionalities.

..... when compared to a lower-level lang.

5) Three things specified by the ISA are Instructions, Data types, and addressing Modes.

6) The microarchitecture specifies how circuits are put together to create the computer. The Instruction Set Architecture (ISA) provides an interface which specifies what sort of instructions a computer supporting this interface can perform. Aside from an improvement in performance/cost/power the actual user sees no difference when programming or running the computer with these two diff architectures.

7) convert to unsigned binary

a) 26 .. 26−16 = 10 ... 1, 10−8 = 2... 1, 2−4 = 0 .. 0, 2−2 = 0 ... 1, 0−1 = ... 0
=> $\boxed{11010}$

b) 49 49−32 = 17 ... 1, 17−16 = 1 ... 1, 0, 0, 0, 1−1 = 0 ... 1 => $\boxed{110001}$

c) 255 255−128−64−32−16−8−4−2−1 => $\boxed{11111111}$

d) 129 −128 = 1, 0, 0, 0, 0, 0, 0, 1−1 = 0 .. 1 => $\boxed{10000001}$

8) unsigned bin → base 10 → hex

a) 0010 1010 → 32+8+2 = $\boxed{42_{10}}$ → 0010 = 2, 1010 = A = $\boxed{2A_{16}}$

b) 00111111 → 1+2+4+8+16+32 = $\boxed{63_{10}}$ → 0011 = 3, 1111 = F = $\boxed{3F_{16}}$

c) 10000000 → $\boxed{128_{10}}$ → 1000 = 8, 0000 = 0 = $\boxed{80_{16}}$

d) 11101001 → 128+64+32+8+1 = $\boxed{233_{10}}$ → 1110 = E, 1001 = 9 = $\boxed{E9_{16}}$

e) 00001001 → 8+1 = $\boxed{9_{10}}$ → 0000 = 0, 1001 = 9 = $\boxed{9_{16}}$

9) $100_{10} \rightarrow$ Following bases

a) 3 $100/3 = 33 \text{ r } \underline{1}$, $33/3 = 11 \text{ r } 0$, $11/3 = 3 \text{ r } \underline{2}$, $3/3 = 1 \text{ r } 0$, $1/3 = 0 \text{ r } \underline{1} \Rightarrow \boxed{10201_3}$

b) 4 $100/4 = 25 \text{ r } \underline{0}$, $25/4 = 6 \text{ r } \underline{1}$, $6/4 = 1 \text{ r } \underline{2}$, $1/4 = 0 \text{ r } \underline{1}$ $\Rightarrow \boxed{1210_4}$

c) 5 $100/5 = 20 \text{ r } 0$, $20/5 = 4 \text{ r } 0$, $4/5 = 0 \text{ r } 4 \Rightarrow \boxed{400_5}$

d) 6 $100/6 = 16 \text{ r } \underline{4}$, $16/6 = 2 \text{ r } \underline{4}$, $2/6 = 0 \text{ r } \underline{2} \Rightarrow \boxed{244_6}$

10) Arbitrary BASE → BASE 10

a) $210_3 \rightarrow 2 \cdot 3^2 + 1 \cdot 3^1 + 0 \cdot 3^0 = 18 + 3 = \boxed{21_{10}}$

b) $321_4 \rightarrow 3 \cdot 4^2 + 2 \cdot 4^1 + 1 \cdot 4^0 = 3(16) + 2(4) + 1(1) = 48 + 8 + 1 = \boxed{57_{10}}$

c) $432_5 \rightarrow 4 \cdot 5^2 + 3 \cdot 5^1 + 2 \cdot 5^0 = 4(25) + 3(5) + 2(1) = 100 + 15 + 2 = \boxed{117_{10}}$

11) 8-bit 2's comp : Perform mathematical op's in binary

a) $-6+20$

```
00000110 (6)
11111001
      +1
11111010 (-6)
```

```
  11111010 (-6)
+ 00010100 (20)
X00001110 (14)
```

$\Rightarrow \boxed{00001110}$

b) $67+30$

```
  01000011 (67)
+ 00011110 (30)
  01100001 (97)
```

$\Rightarrow \boxed{0110\ 0001}$

c) $42-20$

```
 00101010 (42)
 11101100 (-20)
X00010110 (22)
```

$\Rightarrow \boxed{00010110}$

```
00010100 (20)
11101011
      +1
11101100 (-20)
```

d) −44 − 23

```
 1
 11010100 (-44)
 11101001 (-23)
 ----------------
X10111101 (-67)  =>  [10111101₂]
```

```
00101100 (44)
11010011
      +1
---------
11010100 (-44)
```

```
00010111 (23)
11101000
      +1
---------
11101001 (-23)
```

e) 26 − 26

```
  1 1 1 1 1  1
  0 0 0 1 1 0 1 0   (26)
+ 1 1 1 0 0 1 1 0   (-26)
---------------------
X 0 0 0 0 0 0 0 0

   =>  [00000000]
```

```
00011010 (26)
11100101
      +1
---------
11100110 (-26)
```