Introduction to Computer Science Midterm 4/18/2012

Name: ________________________ ID: ________________________

1. Answer the following questions using the data representations described in the textbook:
   (a) Express the fractional number 22/7 in 8-bit fixed-point binary notation. The radix point is set at the middle of the 8-bit notion. That is, the most significant 4-bit is the integral part and the least significant 4-bit is the truncated fractional part [**4 points**].
   (b) Express the floating point number 1.63 in binary notation using 1-bit sign, 3-bit exponent in excess-4 notation (100 stands for 0), and 4-bit mantissa [**6 points**].

[Solution]

(a) 0011 0010

(b) 0 101 1101

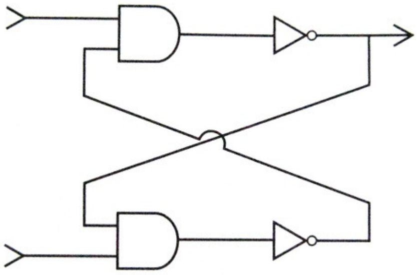
2. Answer the following questions about the stored-program concept.
   (a) What is the stored-program concept? [**4 points**]
   (b) In a computer designed with the stored-program concept, which special-purpose register is used to distinguish programs from data? How? [**6 points**]

[Solution]

(a) A program is just a special type of data and can be stored in main memory.

(b) The program counter (PC) is used to distinguish programs from data. Before the computer start executing the program, the program counter will be set to the memory address of the memory cell that contains the first instruction of the program. The program counter will be updated to point to the next instruction during the execution of each instruction.

3. Assume that both of the inputs in the circuit are 1.
   (a) What would happen if the upper input were temporarily changed to 0? What would happen if the lower input were temporarily changed to 0? [**5 points**]
   (b) Explain why this circuit can be used as a 1-bit storage in a computer. [**5 points**].



[Solution]

(a) If the upper input is set to 0, the circuit will store an 1 and the output will become 1. If the lower input is set to 0, the circuit will store a 0 and the output will become 0.

(b) If both inputs are set to 1, the circuit is in read-mode and the output of the circuit will be the previously stored value. If one of the input is set to 0 for one cycle, the circuit is in write-mode and either 0 or 1 will be stored into the circuit as described in (a).

4. When a CPU executes a program, it keeps performing a three-step process known as the machine cycle. The three steps are Fetch, Decode, and Execute. Please describe the operations of these steps in a machine cycle [**10 points**].

[Solution]

(a) *Fetch* – copies the instruction in the memory cells addressed by the Program Counter to the Instruction Register and increment the Program Counter to point to the next instruction in main memory.

(b) *Decode* – decodes the bit pattern in the instruction register to determine the operation required and generates control signals to setup the computation circuits.

(c) *Execute* – performs the operation specified by the instruction in the instruction register (and write back the result to the destination register or memory cell).

5. Please answer the following two questions related to the network protocols:

(a) What is the hidden terminal problem for wireless networks [**5 points**]?

(b) What is the main difference between CSMA/CD and the CSMA/CA [**5 points**]?

[Solution]

(a) A wireless terminal connected to an Access Point (AP) may not be able to detect the wireless signals generated by other wireless terminals connected to the same AP.

(b) In CSMA/CD, all terminals can detect all the signals from other terminals in the same subnet. Therefore, all terminals can detect whether there are communication collisions among terminals and resolve the collision by themselves. On the other hand, for CSMA/CA, not all connected terminals may detect signal collisions from other terminals. As a result, a central arbiter (i.e., an AP) is necessary to prevent communication collisions.

(c)

6. The number of different bits between two binary patterns is called the Hamming distance. The following assembly program (written for the simple CPU described in the textbook) computes the Hamming distance between two bit patterns (all numbers are in hexadecimal notation):

```
          xor   R1,R1,R2
next_bit: and    R2,R1,R3
          addi   R4,R4,R2
          xor    R1,R2,R1
          ror    R1, 1
          jmpEQ  R1=R0, done
          jmpEQ  R0=R0, next_bit
done:     halt
```

The computer memory used to store the program is as follows:

Memory address	Instruction
00	9112
02	8213

04	5442
06	9121
08	A101
0A	B10E
0C	B002
0E	C000

Please answer the following questions.

(a) What numbers would you use to initialize the registers R0~R4 so that you can compute the distance between the bit patterns 00100010 and 01000110? [**3 points**]

(b) When the execution is done, what are the values in registers R0~R4? [**3 points**]

(c) Without modifying the program in memory, how would you initialize the registers (including PC and R0~R4) to compute the even parity bit of a bit pattern? [**4 points**]

[Solution]

(a) R0 ~ R4 = 00h, 22h, 46h, 01h, 00h.

(b) R0 ~ R4 = 00h, 00h, 01h, 01h, 03h.

(c) R0 ~ R4 = 00h, input, 00h, 01h, 00h. PC = 00h or 02h.
When the CPU halts, the least significant bit of R4 is the even parity bit of the input bit pattern in R1.

7. For a single-CPU multiprogramming system, we can divide CPU time into time slices. Assume that it takes 500 microseconds for the system to perform a context switch. If all the processes are doing computation operations only (i.e. no I/O operations), and they have the same priority sharing the CPU. If we want to keep the context switching overhead less than or equal to 5% and support as many simultaneous processes per second as possible, what is your choice of the time slice (in milliseconds)? **[10 points]**

[Solution]

The context switching overhead is less than or equal to 5%. That means within 1 seconds,
we can perform at most (1,000,000 × 0.05) / 500 = 100 context switches. Divide the remaining time to each process evenly, the time slice per process should be equal to (1000 – 50) / 100 = 9.5 milliseconds.

8. There are three necessary conditions to cause a deadlock in a system. Suppose we have a single-lane bridge going from east to west. If two cars enter the bridge at the same time from the opposite sides, they will meet somewhere in the middle of the bridge and cause a deadlock situation. For each of the following three methods to avoid or resolve such deadlock situation, please describe which necessary condition for the deadlock has been removed.

(a) If two cars meet in the middle of the bridge, the west-side car back up [**4 points**].

(b) Do not let a car drive onto the bridge until the bridge is empty [**3 points**].

(c) Add a second lane to the bridge so that we have one lane for each direction [**3 points**].

[Solution]

(a) An occupied non-sharable resource is forcibly revoked and given to the other process.

(b) Partial allocation of non-sharable resources to different processes is not allowed.

(c) Remove competition for non-sharable resources.

9. Some of the following features are the characteristics of the UDP and some are the features of the TCP. Please specify which features belong to the UDP and which ones belong to the TCP. Note that some of the features may be common to both UDP and TCP [**10 points**].
(a) Establish a connection before communication, (b) Transport-layer protocol, (c) Best effort transmission, (d) Used in DNS lookup mechanism, (e) Having flow control and congestion control, (f) Less protocol overhead.

[Solution]

TCP – (a), (b), (e).
UDP – (b), (c), (d), (f).

10. If the output of the following C program is "Bad," how many bits does the computer use to store the mantissa of a floating-point variable? [**10 points**].

```
float a = 2.75;
if (a == (2.5 + 0.125 + 0.125)) {
    printf("Good.");
} else {
    printf("Bad");
}
```

[Solution]

Since the output is "Bad," that means that the truncation error of the computer makes 2.5 + 0.125 + 0.125 different from 2.75. This can only happens if the computer computes ((2.5 + 0.125) + 0.125) and rounds the intermediate results of the mantissa to 4 bits.
In this case, the mantissa of 2.75 is 1011 and the result of ((2.5 + 0.125) + 0.125) is 1010.