National Tsing Hua University
Department of Computer Science
CS1356 -- Introduction to Information Engineering
Mid-term Examination, Fall 2009

1. (20 pt) Consider an 8-bit binary number system with 2's complement for negative numbers.
   (a) What is 37 (base 10) represented in this system?
   (b) What is -48 (base 10) represented in this system?
   (c) What is 5A (base 16) represented in this system?
   (d) What is the decimal number of 00101101 (base 2)?
   (e) Give an example that can cause overflow by using this number representation.

   Ans:
   (a) 0010 0101
   (b) 1101 0000
   (c) 0101 1010
   (d) 45
   (e) There are two cases that may incur overflow:
   Positive + Positive (e.g. 127 + 3) or Negative + Negative (e.g. -128 + (-1))

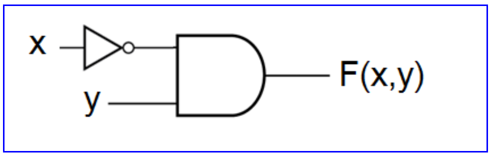
2. (12 pt) Consider an 8-bit floating point number whose bit pattern is 1 bit for sign, 3 bits for exponent, and 4 bits for mantissa in order. The exponent uses excess notation ranged from -4 to 3, and the mantissa is normalized so that it is larger than or equal to 1/2 and smaller than 1.
   (a) What is the binary representation of 21/16?
   (b) What is the floating point representation mentioned above for 21/16?
   (c) What is the truncation error of this floating point representation?

   Ans:
   (a) 1.0101
   (b) $1.0101 = 0.10101 \times 2^1$
   excess = 1 + 4 = 5 = 101 => 0 101 1010 (the least significant bit is truncated)
   (c) The truncation error is 0.0625_d ($=0.00001_b$)

3. (6pt) A Boolean function F(x,y) is defined as in the truth table. Design a circuit that uses AND, OR, and NOT gates to implement the function F(x,y) and justify your answer.

x	y	F(x,y)
0	0	0
0	1	1
1	0	0
1	1	0

Ans:



4. (12 pt) Suppose Alice wants to send an image file of size **4M bytes** to Bob. The network has a transfer rate of **2^{20} bps**. However, for every **1024 bits**, there will be one bit error.
   (a) Suppose an **odd parity bit** is added to every **byte** to detect the error. How large the file will become after the error detection bits are added?
   (b) How long does it take to transfer this encoded image? (You can assume the encoded image is of size X bytes and express your answer using X if you cannot make sure your answer in 4(a) is correct.)
   (c) After getting the entire encoded image, Bob knows which parts of file have errors. He sends a list showing errors to Alice, and asks Alice to resend those erroneous parts. How many bytes does Alice need to resend? Suppose this time, no error detection code is added and no error occurs.

   Ans:
   (a) 4 Mbytes + 4 Mbits = 4Mbytes + 0.5 Mbytes = 4.5 Mbytes (or 4.5 MB)
   (b) 4.5M byte / 1Mbps = 36M bits/ 1Mbps = 36 seconds (or 8X/M seconds)
   (c) For every 1024 bits, there will be one bit error. However, when there is one bit error, the entire byte need be resent, because Bob cannot know which bit is erroneous. (He can only know which "byte" has error.) Therefore, for every 1024 bits, there are 8 bits (1 byte) need be resent. Since the entire encoded image is 4.5MB, the number of data to be resent is 4.5M/(1024/8) = 36KB (or X/128)

5. (5 pt) Give one advantage for using the stored-program concept in designing computers and explain your reason.

6. (10 pt) Using the machine language described in the language description table given at the end, write a sequence of instructions that will find the two’s complement of the value stored at address A0. The result is stored in register 4. (Hint: You could use XOR to find the complements.)

   Ans: 10A0
   21FF
   2201
   9301
   5423

7. (15 pt) Using the machine language described in the language description table given at the end, write a sequence of instructions to implement the following C statement. Assume that the variable *A* is stored at address D0 with an initial value of F9. (Hint: Use shift and addition instructions to implement * and /. Note that the numbers have signs.)

   A = (A * 5) / 2;

   What is the final value in address D0 assuming a two’s complement representation?

Ans:

Idea: 1、Calculate A*5 by Branch operation Address A0 ~ B0
2、Calculate A*5/2 by Rotate Address BA
3、Sign extension Address B2~BC

Address	
A0	2005
A2	2101
A4	2200
A6	13D0
A8	2400
AA	5434
AC	5221
AE	B2B2
B0	B0AA
B2	21FE
B4	8414
B6	2280
B8	8224
BA	A401
BC	7424
BE	34D0

8. (10 pt) The following C program adds 3 numbers in an array A.

```
int main(){
   int A[3]={3,4,2}, B=0, i;
   for(i=0; i<3; i++) B=B+A[i];
}
```

```
2003
2101
2200
1303
14___
5334
5221
3239
B2___
B0___
3303
C000
```

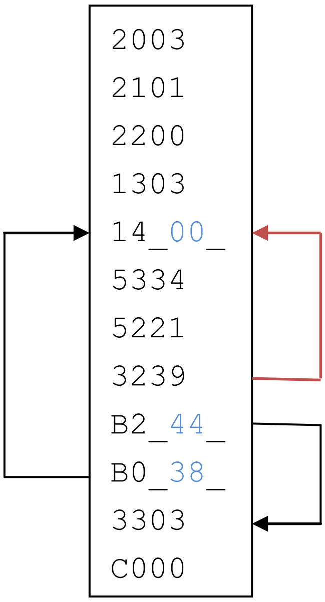
Suppose the program is translated to the machine program, written in the machine language described in the language description table given at the end, and is stored in main memory beginning at address **30** (hexadecimal). The data type "int" is 8 bits in the machine. Array A is stored in main memory **00** to **02**. Variable **B** is stored at **03**. Their initial values are set before the program is executed.

(a) The translated program is partially given in the left. To make the program execute correctly, what should be in the blank places?

(b) What happen to the program if the statement "3239" is erroneously entered as "3238"?

Ans:

(a)



30	2003
32	2101
34	2200
36	1303
38	14_00_
3A	5334
3C	5221
3E	3239
40	B2_44_
42	B0_38_
44	3303
46	C000

(b)

If 3239 is replaced with 3238, it will change the OP code which is in the “38” address of memory. Then the program logic will be wrong.

9. (10 pt) Suppose in a fast food restaurant, a hamburger menu consists of a hamburger, a drink, and French fries. Their regular prices are \$60, \$20, and \$20, respectively. Since this is your birthday, you are able to negotiate a special deal for the hamburger at \$15. Therefore, you get a saving of

$$saving = 1 - \frac{price\ of\ a\ discount\ hamburger\ menu}{price\ of\ a\ regular\ hamburger\ menu} = 1 - \frac{15+20+20}{60+20+20} = 45\%$$

This saving can be explained using Amdahl’s law, which states that the speedup using *n* processors to execute a program with an *f* portion ($0 < f \le 1$) that cannot be parallelized is

$$speedup = \frac{execution\ time\ uses\ one\ processor}{execution\ time\ uses\ n\ processors} = \frac{t_s}{t_p} = \frac{t_s}{f \cdot t_s + (1-f) \cdot t_s / n} = \frac{n}{1+(n-1)f}$$

(a) What are *f* and *n* in the hamburger case?

N=60/15=4(In Amdahl’s law, the n represents the number of processors. In this case, the cost of Hamburger is reduced with 15.)

F=20+20/60+20+20=0.4=40%

(b) If you are able to use your charm to negotiate even lower prices for the hamburger, what is the maximum saving that you can expect?

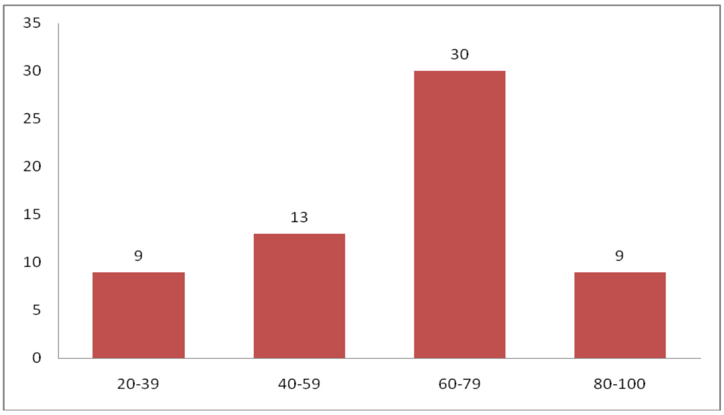
1-(20+20)/(60+20+20)=60%

Language Description Table

Opcode	Operand	Description
1	RXY	LOAD the register R with the bit pattern found in the memory cell whose address is XY.
2	RXY	LOAD the register R with the bit pattern XY.
3	RST	STORE the bit pattern found in register R in the memory cell whose address is XY.
4	ORS	MOVE the bit pattern found in register R to register S.
5	RST	ADD the bit patterns in registers S and T as though they were two's complement representations and leave the result in register R.
6	RST	ADD the bit patterns in registers S and T as though they represented values in floating point notation and leave the floating-point result in register R.
7	RST	OR the bit patterns in registers S and T and place the result in register R.
8	RST	AND the bit patterns in registers S and T and place the result in register R.
9	RST	EXCLUSIVE OR the bit patterns in registers S and T and place the result in register R.
A	R0X	ROTATE the bit pattern in register R one bit to the right X times. Each time place the bit that started at the low-order end at the high-order end.
B	RXY	JUMP to the instruction located in the memory cell at address XY if the bit pattern in register R is equal to the bit pattern in register number 0. Otherwise, continue with the normal sequence of execution.
C	000	HALT execution.

Statistics of midterm: Average: 61.93

Histogram


35
30
25
20
15
10
5
0
9
13
30
9
20-39
40-59
60-79
80-100