$2^{32} \div 2^{3}$ $2^{?}$

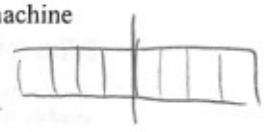
Introduction to Computer Science

First Midterm 3/21/2005

1. (5%)What would be the hexadecimal representation of the largest memory address in a memory consisting of 4GB if each cell had a one-byte capacity? (5%) How many cells can be in a computer's main memory if each cell's address can be presented by four .hexadecimal digits?

2. (10%) What are the base ten representations of 11011000, if it is in (a) floating point format? (b) excess format? (c) 2's complement format? (d) From the obtained three decimal numbers, convert the floating point one and excess one to 2's complement representations then perform the 2's complement add of these three numbers.

3. (10%) (a)Using the machine language described in Appendix C, write programs to perform the following task: Copy the least significant four bits from memory location A7 into the most significant four bits of A7. (b) Suppose the memory address AF through B1 in the machine described in Appendix C contain the following bit patterns:

A7 

Address	Contents
AF	B0
B0	B0
B1	AF

AF B0B0 → B0

B0B0B1 B0B0

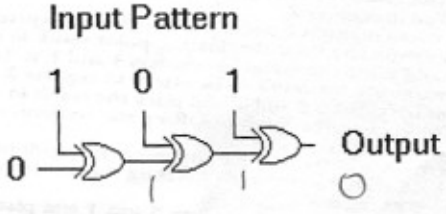
~~B1~~ B0AF

What would happen if we started the machine with its program counter equal to AF?

4. (12%) Give a brief description of each of the following:

(a)multitasking (b)kernel (c) booting process (d)shell

5. (10%) (a). What is the output of the circuit below?



(b) In general, how does the three-bit input pattern across the top of the diagram relate to the circuit's output?

6. (10%) Suppose a 32-bit code is generated by representing each symbol by four consecutive copies of its ASCII representations. What error-correcting properties does this new code have?

7. (5%) Describe the machine cycle. (5%)Explain the concept of throughput and techniques by which throughput is increased.

8. (4%) Which component of an operating system maintains the directory system?

(4%) Which component of an operating system handles the details associated with particular peripheral equipment?

9. (10 %) Suppose the contents of the memory cells starting at address 00 are (in hexadecimal) 1A, 02, 2B, 02, 9C, AB, 3C, 00, C0, 00, in the machine described in Appendix C. Suppose the machine starts with its program counter set to 00. (a) What will be in the memory cell at address 00 when the machine finally halts? (b) What bit pattern will be in the program counter when the machine halts?

10. (5 %) Explain the differences between RISC CPUs and CISC CPUs.

(5 %) Using gates, design a circuit to output the even parity bit from the input byte (8 bits)

Machine Language

Each machine instruction is two bytes long. The first four bits consist of the op-code; the last 12 bits make up the operand field. The table that follows lists the instructions in hexadecimal notation together with a short description of each. The letters R, S, and T are used in place of hexadecimal digits in those fields representing a register identifier that varies depending on the particular application of the instruction. The letters X and Y are used in lieu of hexadecimal digits in variable fields not representing a register.

Op-code	Operand	Description
1	RXY	LOAD the register R with the bit pattern found in the memory cell whose address is XY. *Example:* 14A3 would cause the contents of the memory cell located at address A3 to be placed in register 4.
2	RXY	LOAD the register R with the bit pattern XY. *Example:* 20A3 would cause the value A3 to be placed in register 0.
3	RXY	STORE the bit pattern found in register R in the memory cell whose address is XY. *Example:* 35B1 would cause the contents of register 5 to be placed in the memory cell whose address is B1.
4	0RS	MOVE the bit pattern found in register R to register S. *Example:* 40A4 would cause the contents of register A to be copied into register 4.
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. *Example:* 5726 would cause the binary values in registers 2 and 6 to be added and the sum placed in register 7.
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. *Example:* 634E would cause the values in registers 4 and E to be added as floating-point values and the result to be placed in register 3.
7	RST	OR the bit patterns in registers S and T and place the result in register R. *Example:* 7CB4 would cause the result of ORing the contents of registers B and 4 to be placed in register C.
8	RST	AND the bit patterns in register S and T and place the result in register R. *Example:* 8045 would cause the result of ANDing the contents of registers 4 and 5 to be placed in register 0.
9	RST	EXCLUSIVE OR the bit patterns in registers S and T and place the result in register R. *Example:* 95F3 would cause the result of EXCLUSIVE ORing the contents of registers F and 3 to be placed in register 5.
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. *Example:* A403 would cause the contents of register 4 to be rotated 3 bits to the right in a circular fashion.
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. *Example:* B48C would first compare the contents of register 4 with the contents of register 0. If the two were equal, the execution sequence would be altered so that the next instruction executed would be the one located at memory address 3C. Otherwise, program execution would continue in its normal sequence.
C	000	HALT execution. *Example:* C000 would cause program execution to stop.