

# **CSE 410 - Computer Systems**

## **Autumn 2001**

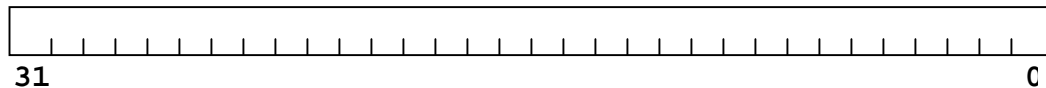
### **Midterm Practice Questions**

The attached questions will be helpful in understanding the type of questions that will be on the exam. The practice questions do not cover all the topics that may be on the exam.

There will be a question where you are given some code to read, and you are asked to comment it (see homework 1, question 3). You will be asked to draw a stack frame based on existing code (see project 1, question 3), and a call tree based on existing code (see project 2, question 1). You will be asked to review a short segment of code, and show the new contents of the registers affected by the instructions.

## CSE 410 Midterm Practice Questions

1. a. What is the 8-bit binary representation of the positive number 9?  
  
b. What is the 8-bit binary 2's complement representation of -9 (negative 9)?
  
2. a. How many different numbers can be represented in a 4-bit 2's complement representation?  
  
b. How many of them represent negative numbers?
  
3. a. Single precision floating point values are stored in one 32-bit word. The field width of the sign bit is 1, the field width of the exponent is 8 bits, and the field width of the mantissa is 23 bits. Show how the fields are placed in the word drawn below.



$$+ 1.1 \times 2^{12}$$

- b. Given the number above (written in "binary scientific notation") draw a circle around the parts of the number that are actually stored, and draw lines to show where they are placed in the word above when the value is stored in memory.

CSE 410 Midterm Practice Questions

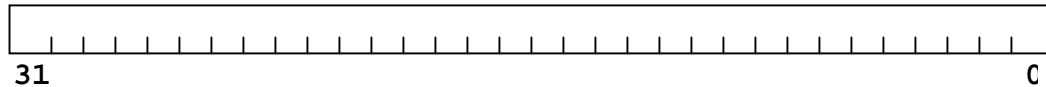
4. a. Fill in the "usage", "available?" and "restore required" columns in the following chart.

Usage? Brief description of the purpose of the register.  
 Available? Is the register available for use in your code, or is it reserved?  
 Restore Required? If the register is available for use, does a procedure have to restore the value of the register after using it? (If not available, leave blank or put n/a.)

<i>name</i>	<i>available?</i>	<i>Restore required?</i>	<i>usage?</i>
<b>zero</b>	<b>yes</b>	<b>no</b>	always returns zero
<b>at</b>	<b>no</b>	<b>n/a</b>	reserved for use as assembler temporary
<b>v0,v1</b>			
<b>a0-a3</b>			
<b>t0-t9</b>			
<b>s0-s7</b>			
<b>k0,k1</b>			
<b>gp</b>			
<b>sp</b>			
<b>fp (s8)</b>			
<b>ra</b>			

## CSE 410 Midterm Practice Questions

5. a. Consider a machine that has a 32-bit program address space, as we have been studying. The designers have implemented a Virtual Memory system that uses 4KB pages, so the page offset field is 12 bits wide. Using the drawing of a 32-bit word given below, indicate the Virtual Page Number Field and the offset field.



- b. How many virtual page numbers are there in this system? Give your answer in  $2^n$  notation, using the correct number for "n".
- c. If the physical addresses are 20 bits wide, how many Physical Page Numbers are there? Give your answer in  $2^n$  notation, using the correct number for "n".
6. How can the OS allow process P1 to share a region of memory with process P2 in a page-based Virtual Memory system?
7. Describe a pattern of accessing memory that is consistent with the assumption of temporal locality associated with the design of cache memory systems.
8. What is the primary advantage of a synchronous data bus?

## CSE 410 Midterm Practice Questions

9. On systems with large program address spaces (32-bit or 64-bit addresses), the page tables can be very , very large, since there is one entry in the table for every virtual page. Why is it that referencing the page tables using virtual addresses in the kernel reduces the amount of physical memory that is needed for these tables?
  
  
  
  
  
  
  
  
  
  
10. A cache holds key/value pairs for rapid access. What is the key and what is the value for an entry in the Translation Lookaside Buffer?
  
  
  
  
  
  
  
  
  
  
11. Draw a simple picture showing a program address space as we have studied it indicating where the program code is locate, the data (the heap), and the stack. Show the directions that the heap and the stack grow while the program is executing.

