Name: ______________________________

CS 33003 # Final Exam **CompOrg**

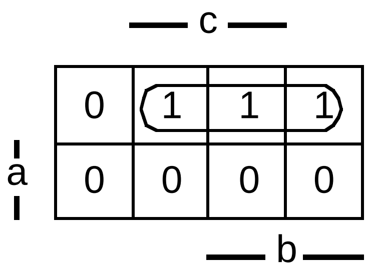
Wednesday 16 December 1998

1. Convert 495.453125_{10} to base 8, showing all your work. (10 points)

495/8 = 61, r 7 0.453125 * 8 = 3.625 $495.453125_{10} = 757.35_8$
61/8 = 7, r 5 0.625 * 8 = 5.0
7/8 = 0, r 7

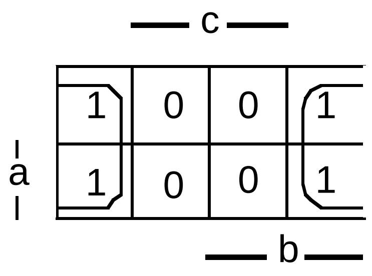
2. Given the Karnaugh maps below, identify any errors, either in the way the ovals are drawn, or in the way the ovals are interpreted. If there are no errors, write "no errors". (8 points)

a. output = a'bc



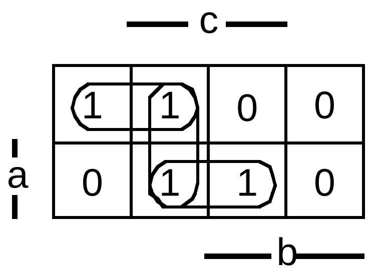
ovals have to be of size 1, 2, 4…

b. output = c'



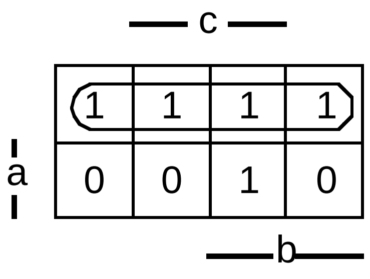
no errors

c. output = a'b' + b'c + ac



vertical oval is redundant
output = a'b' + ac

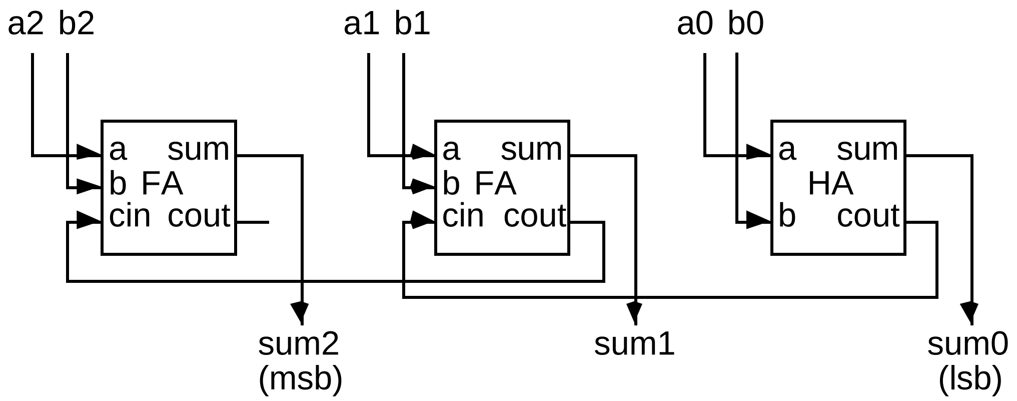
d. output = a'



need to cover lower "1"
output = a' + bc

3. Draw a diagram showing how a 3-bit adder can be built out of full adders and half adders (6 points)

Name: ______________________________



4. Suppose a CPU is referred to as a "32-bit" CPU. (6 points)

a. What are the implications of this with respect to the datapath?

All buses, ALUs, registers, etc. must be 32 bits wide. Instructions are also probably 32 bits wide, at least for RISC machines.

b. What are the implications of this with respect to the memory?

Each load or store operation can access 32 bits at a time (which also means there are probably 32 data lines between the CPU and memory). Note that this does not say anything about the number of memory locations or the number of address lines.

c. If this CPU is said to have "little endian byte order", what does that mean?

Byte on little end (LSB) is labeled zero: byte3 byte2 byte1 byte0

5. Suppose you need to implement a 32M (2^{25}) x 4 bit memory system. (8 points)

a. How many 16M (2^{24}) x 1 bit memory chips would be needed?

8 (two rows, four chips in each row)

b. How many 4M (2^{22}) x 4 bit memory chips would be needed?

8 (eight rows, one chip in each row)

c. In case (a.) above, how are the address lines for the memory system as a whole connected <u>to the CS inputs</u> of each memory chip? Explain your answer either textually, or draw a diagram. Note — you do <u>not</u> need to show how all the other lines are connected.

Use either address line 0 or 23 (MSB or LSB), and connect it directly to the CS line on one row of chips, and connect it through an inverter to the CS line on the other row of chips.

Name: ______________________________

6. Write code fragments to add the values at memory locations 100 and 101 and store the sum into memory location 200, using each of the following instruction formats: (12 points)

a. 0-operand (stack)

```
PUSH 100
PUSH 101
ADD
POP    200
```

b. 1-operand (accumulator)

```
LOAD  100
ADD    101
STORE 200
```

c. 2-operand

```
ADD    100,101          LOAD  200,100
STORE 200,100           ADD    200,101
(destroys value         (better solution)
 in 100, though)
```

d. LOAD/STORE

```
LOAD  R1,100
LOAD  R2,101
ADD    R3,R1,R2
STORE 200,R3
```

7. The assembler is responsible for translating source code into object code. (8 points)

a. In the source code, what items go in the text segment, data segment, and bss segment?

Text segment contains program code, data segment contains initialized data, and bss segment contains uninitialized data

b. How does the Location Counter differ from the Program Counter?

The Location Counter is a variable used by the assembler to keep track of the current instruction's location as the program is being translated to object code. The Program Counter is a register in hardware that holds the address of the next instruction as the program is being executed.

8. Show the 5-bit representation of each of the decimal values below in each of the specified formats. If it is not possible to represent a particular value in a particular format, write "not possible" in that location. (9 points)

Value	**Signed Magnitude**	**Excess 16**	**2's Complement**
15	01111	11111	01111
2	00010	10010	00010
–16	not possible	00000	10000

Name: ______________________________

9. The SPARC CPU has some "features" that may seem rather unusual at first. (9 points)

a. Register %g0 is permanently hardwired to zero. Give one example of when this is useful.

The clr synthetic instruction uses it to clear a register — or %g0,%g0,%l2

The mov synthetic instruction uses it to copy from register to register — or %g0,%l2,%l3

The ld instruction can use it to access memory when a displacement is not needed —
ld [%l1+%g0],%l5

b. "inc" is a synthetic instruction that adds the constant "1" to a specified register. What is meant by "synthetic instruction"?

One recognized by the assembler, but not part of the CPU's instruction set. Synthetic instructions are macros "implemented" by the assembler using regular instrucdtions.

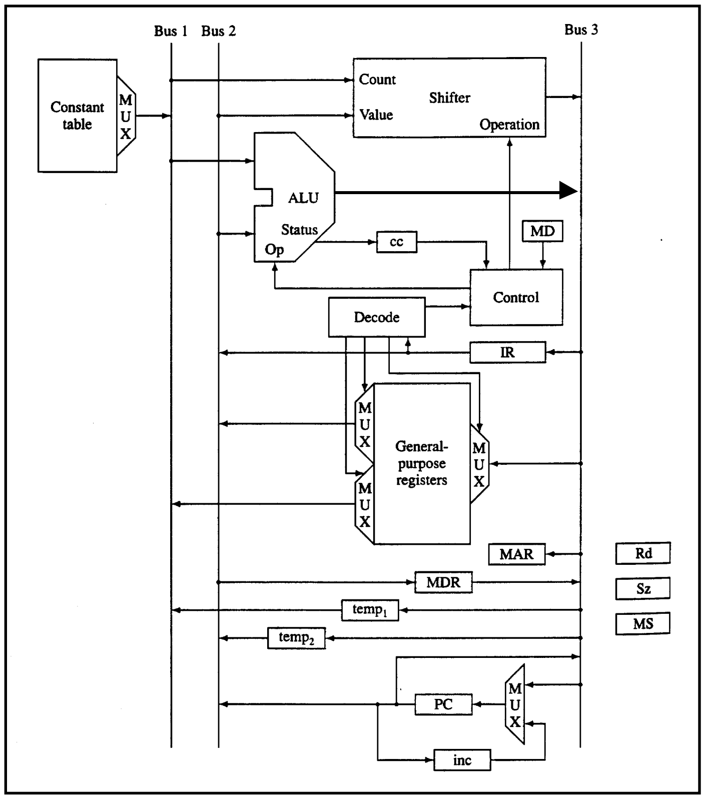
c. There is no multiply instruction in the instruction set. What do you do in your program if you need to multiply?

Call the multiply subroutine provided by the assembler.

d. What is the "branch delay slot", and why is it important to be careful what instructions are placed there?

The instruction following a branch or call instruction. Since it is being fetched while the branch or call is executed, it will be executed before the branch or call instruction, and this must be considered by the programmer when writing the code.

10. Consider the Chapter 9 Simple machine, shown at the right. (11 points)



a. One of the functions that can be performed by the ALU is addition, and it might be required to do this for at least three different reasons. List <u>two</u> of these reasons, or list <u>all three</u> for extra credit.

Add two register values in the add instruction, add offsets to the PC in branch instructions, add two register values to produce a memory address in data migration instructions

b. Why are bus 1 and bus 2 both required? Why not

connect a single bus to both ALU inputs?

Because the ALU needs both inputs at the same time, and it is not possible to put two different values on one bus at the same time.

c. What does the "controller" (labeled "control" in the figure) control? Be specific.

ALU and shifter — operation to perform
multiplexers — select inputs
registers — load
bus tri-state drivers

11. Almost every modern machine has at least one or more caches. (6 points)

a. Why is it important to cache instructions?

If the program executes the same instruction repeatedly, as might happen in a loop, caching that instruction will improve program performance (since getting data from the cache is much faster than getting data from memory).

b. Why is it important to cache data?

Because the same data is often used repeatedly — global variables, or data in a loop

12. For years, CPUs have supported interrupts. (7 points)

a. In what way are interrupts better than using programmed I/O?

With programmed I/O, the CPU must constantly check to see if the I/O operation is finished, which wastes CPU time. With interrupts, the CPU can perform other tasks, and when the I/O operation is finished, that device will interrupt the CPU, which can then process the data.

b. What items must the interrupt handler save and restore?

General purpose registers (note that technically the CPU is responsible for saving the PC and PSW, not the interrupt handler).