

UNIVERSITY OF TORONTO
Faculty of Arts and Science
DECEMBER 2011 EXAMINATIONS
CSC258H1F Computer Organization
Aid allowed: one letter-size aid sheet (both sides)
1 pages, 6 questions, 100 marks, 2 hours

The value of each question is shown in square brackets.

A blank answer is worth about one-third of the marks;
to that, marks will be added for readable, relevant, and correct information,
and marks will be subtracted for unreadable, irrelevant, or incorrect information.

1 A majority function has three inputs a , b , and c , and one output m . The output is 1 if the inputs include more 1s than 0s, and 0 otherwise.

(a)[9] Design a majority function circuit.

(b)[7] How can you build a 6-input majority function by using two 3-bit majority functions (from part (a)), feeding three inputs into one of them and the other three inputs into the other, and then combining the two results to obtain the final result?

2[21] Design a serial 2's complement negater. It has three inputs x , f , and c , and one output y . Input c is a clock input. At the rising edge of each clock cycle, x is one bit of a 2's complement integer, starting with the rightmost bit, and proceeding through the integer, one bit per cycle. Input f (which stands for first) tells whether x is the first bit of the integer; $f=1$ just for the first bit. Output y is one bit of the negation of the integer, also starting with the rightmost bit. The output integer has the same number of bits as the input integer; ignore overflow. For example, if we want to negate the sequence of integers 12, -3, 2, here's what we could feed in, and what we get out, each clock cycle:

f	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
x	0	0	1	1	0	1	0	1	1	1	0	1	0	0	0
y	0	0	1	0	1	1	1	0	0	0	0	1	1	1	1

In this example, I have used 5 bits per integer, but there's no need to use the same number of bits for each integer. You can use any basic gates and flip-flops.

3(a)[9] In binary quote notation, perform the division $1'0 / 1'011$. There is no need to convert these numbers, or the answer, to a familiar decimal notation.

(b)[3] State whether each of the numbers in part (a), and your answer to part (a), is negative, zero, or positive. How do you know?

4 How many parity bits must be added to 8 data bits to detect and correct

(a)[6] up to 1 bit of error?

(b)[6] up to 2 bits of error?

5[21] Here is part of a C or Java program. The declarations are

int A[100]; **int** n;

Array A is initialized (that is not your concern). Then the statements

for ($n=0$; $n<100$; $n++$) **if** ($A[n]<0$) $A[n]=-A[n]$;

make each element of A into its absolute (nonnegative) value. Translate these declarations and statements to CSC258 assembly language. (They are only part of a program, not the entire program).

6 Write a microprogram for the CSC258 computer that implements each of the following instructions. Describe any special details or changes to the basic computer that are needed to add these instructions to the machine. Do not make any changes to the machine that are not needed.

(a)[9] LAI m : (Load Accumulator Indirect) Load into the accumulator the contents of the memory location whose address is in memory location m .

(b)[9] SZA m : (Skip on Zero Accumulator) If the content of the accumulator is zero, then skip over (branch forward) m instructions; otherwise continue with the next instruction. If m is zero, the instruction executed next is the one located after this SZA instruction.

end of questions