Tutorial Questions for CSC 311

1) A full adder has three inputs A, B, and a carry in C, such that multiple adders can be used to add larger numbers. The truth table for representing a full adder is given below:

Input			Output	
A	B	C_i	C_0	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

a) Find the corresponding expressions for sum and carry out using the above truth table
b) Design a circuit diagram to implement the expressions in (a) using NAND gates only
c) Differentiate between binary adder and binary subtractor

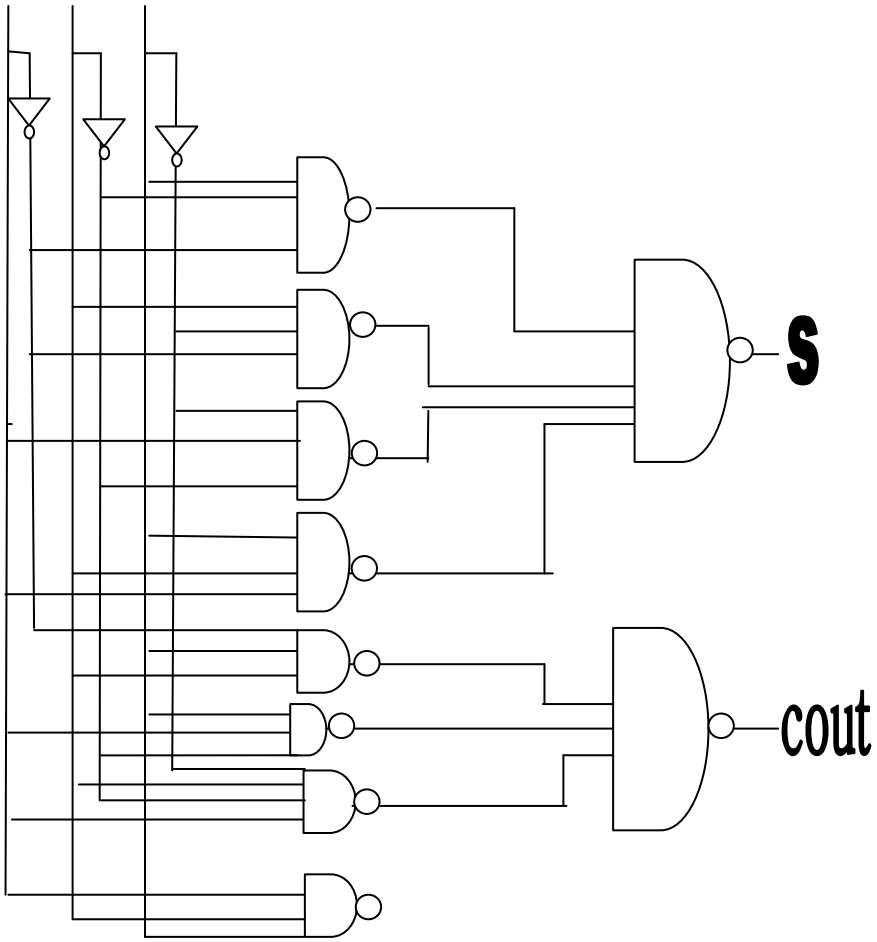
Solution to Q1

(a) To find corresponding expressions for sum & carry out

Carry out = Co = $A'BC_i + AB'C_i + ABC_i' + ABC_i$

Sum = S = $A'B'C_i + A'BC_i' + AB'C_i' + ABC_i$

b.



C.

Half subtraction is an operation carried out by the half subtractor, which is a combinational circuit that is used to perform subtraction of two bits. It has two inputs, X (minuend) and Y (subtrahend) and two outputs D (difference) and B (borrow). Such a circuit is called a half – subtractor because it enables a borrow out of the current arithmetic operations but no borrow in from a previous arithmetic operation.

2) Suppose we have 8 memory banks with a bank busy time of 6 clocks and a total memory latency of 12 cycles. How long will it take to complete a 64-element vector load with a stride of 1? With a stride of 32?

Solution to Q2

Since the number of banks is larger than the bank busy time, for a stride of 1 the load will take 12 + 64 = 76 clock cycles, or 1.2 clock cycles per element. The worst possible stride is a value that is a multiple of the number of memory banks, as in this case with a stride

of 32 and 8 memory banks. Every access to memory (after the first one) will collide with the previous access and will have to wait for the 6-clock-cycle bank busy time. The total time will be 12 + 1 + (6 * 63) = 391 clock cycles, or 6.1 clock cycles per element.

3) There is an application running on a 32-processor multiprocessor, which has a 200 ns time to handle reference to a remote memory. For this application, assume that all the references except those involving communication hit in the local memory hierarchy, which is slightly optimistic. Processors are stalled on a remote request, and the processor clock rate is 3.3 GHz. If the base CPI (assuming that all references hit in the cache) is 0.5, how much faster is the multiprocessor if there is no communication versus if 0.2% of the instructions involve a remote communication reference?

Solution to Q3

It is simpler to first calculate the clock cycles per instruction. The effective CPI for the multiprocessor with 0.2% remote references is:

CPI = Base CPI + Remote request rate ×Remote request cost

= 0.5 + 0.2% × Remote request cost

The remote request cost is

$$\frac{Remote\ access\ cost}{Cycle\ time} = \frac{200ns}{0.3ns} = 666cycles$$

Hence, we can compute the CPI:

CPI = 0.5 + 1.2 =1.7

The multiprocessor with all local references is 1.7/0.5 = 3.4 times faster. In practice, the performance analysis is much more complex, since some fraction of non-communication references will miss in the local hierarchy and the remote access time does not have a single constant value. For example, the cost of a remote reference could be quite a bit worse, since contention caused by many references trying to use the global interconnect can lead to increased delays.

4) With the aid of diagrams, explain generic computer organization
5) Differentiate between Read only Memory and Random Access Memory
6) Differentiate between memory read operation and memory write operation

7) Write short notes on the following:
    a. Data transfer instructions
    b. Data operational instructions
    c. Program control instructions
    d. 16 Bit function and truth table of Microoperations
    e. Binary Adder
    f. Binary Subtractor
    g. Binary Increment

8) A typical digital computer has many registers and paths that must be provided to transfer information from one register to another. Discuss a bus system for four multiplexer registers in a computer system.
9) Discuss how an 8x4 memory subsystem can be constructed from two 8x2 ROM chips
10) What do you understand by instruction set architecture design?
11) What is Arithmetic Microoperations
12) How do you implement an arithmetic microoperation with a combinational circuit
13) Draw a 4-bit combinational circuit for a micro operation incrementer
14) Prepare truth tables and logic micro-operations for 16 functions of two variables
15) Describe the hardware implementation for the tables prepared in (14)
16) Prepare a function table for Arithmetic Logic Shift Unit
17) What is control memory?
18) What do you understand by half adder? Sum is the two-bit XOR of A and B, and Carry is the AND of A and B. Give the truth table to represent a half adder. Implement the sum and carry adder with a circuit diagram.
19) How do you implement an arithmetic micro-operation with a combinational circuit