**Note:**  (Late: -15% Penalty).

1. (15 pts) Write a brief essay paragraph explaining one breakthrough in the history of computing and the significance of the breakthrough to advancing Computer Science.

Acceptable answers include explanations of vacuum tubes, transistors, integrated circuits, VLSI, binary arithmetic, quantum computing, and parallel computing.

Ans.

1. (15 pts) Using the CRC polynomial 1101, compute the CRC code for the information word, 1 0110 1101

Ans:

1. (10 pts) Convert the following decimal fractions to binary with a maximum of six places to the right of the binary point:
   1. 26.78125
   2. 194.03125
2. (15 pts) Define the following
   1. Combination Logic:
   2. Sequential Logic:
   3. How are sequential circuits different than combinational circuits?
3. (10 pts) Add the following 8-bit two’s complement numbers (i.e. one sign bit and seven data bits) AND indicate “Overflow” if it occurs.

a.  1111 0101

+ 1101 0101

Answer:

b.  0110 1011

+ 0101 0101

Answer:

1. (15 pts) Given the boolean equation: x’yz + x(yz)' + x'(y+z) + (xyz)'
   1. Construct the truth table for the Boolean equation:

|  |  |  |  |
| --- | --- | --- | --- |
| x | y | z | F(xyz) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* 1. Construct the K-map for this circuit.

xy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| z | 00 | 01 | 11 | 10 |
| 0 |  |  |  |  |
| 1 |  |  |  |  |

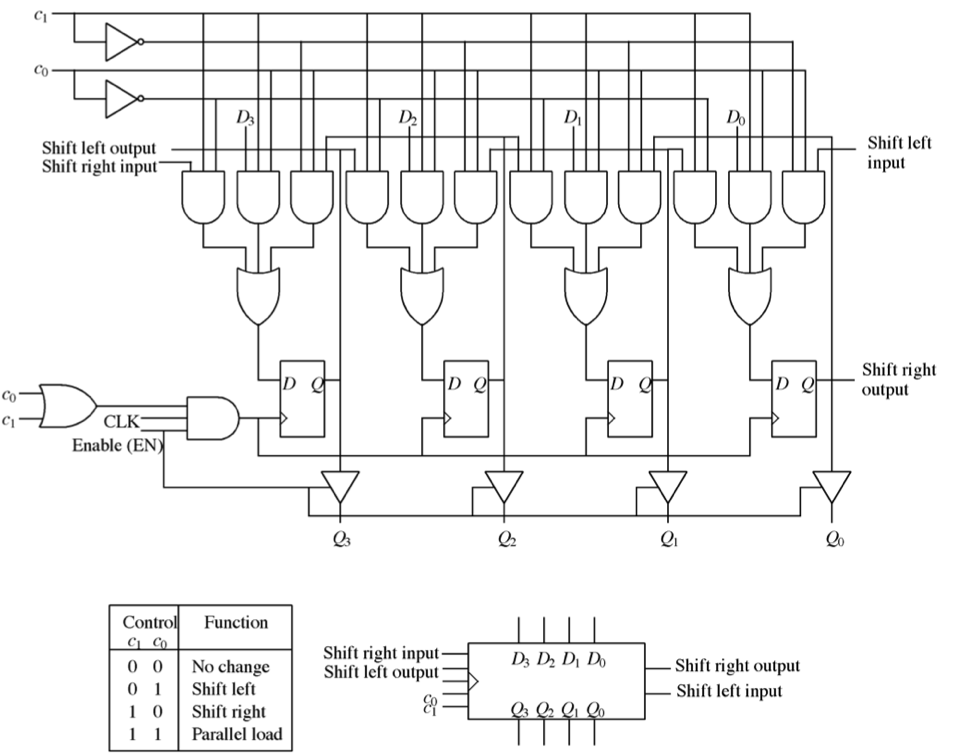
* 1. Write a reduced but equivalent Boolean equation. Use either the K-map or apply the Boolean postulates/theorem’s to do your reduction.

1. (15 pts) Assume a 220 byte memory:
   1. What are the lowest and highest addresses if memory is byte-addressable?
   2. What are the lowest and highest addresses if memory is word-addressable, assuming a 16-bit word?

* 1. What are the lowest and highest addresses if memory is word-addressable, assuming a 32-bit word?

1. (5 pts) Given the left-right shift register shown below answer the following questions. Place an ‘x’ for don’t care conditions on the non-relevant outputs.

Note: Each Control input ‘c0 and c1’ as listed in the function table is shown twice in the diagram.



Enable

1. Fill in the state output of Q0-to-Q3 on clock Clk4 given the input ‘Enable=1’ AND the following Input serial load of the shift-register in clocks Clk0-to-Clk3

Clk0-to-Clk3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Clk0 | Clk1 | Clk2 | Clk3 | C1 | C0 |
| Shift Right Input | 0 | 1 | 0 | 0 | 0 | 1 |
| Shift Left Input | 1 | 0 | 1 | 1 |

Clk4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Q3 | Q2 | Q1 | Q0 | C1 | C0 | Enable |
|  |  |  |  | 0 | 0 | 1 |

1. (15 pts) Using the MarieSim emulator - Write working assembly code that executes the logical programming construct:

Turing machines and Finite State Machines will continue until told to ‘halt’. Please halt your code upon completion of the assigned task. Include comments to your code!

Initialize A=5; B=9;

UNTIL (A >=B) ! Until A is greater than or equal to B

DO

C = A + B ! Add A to B

STORE C ! store in C

A=A-1 ! Subtract 1 from A

B=B-2 ! Subtract 2 from B

DONE

Answer:

1. (15 pts) Using the MarieSim emulator - Write a working assembly code that executes the logical programming construct:

Turing machines and Finite State Machines will continue until told to ‘halt’. Please halt your code upon completion of the assigned task. Include comments to your code!

IF (A = B)

THEN

C = A + B

ELSE

C = A – B

Answer:

1. (15 pts) Using the MarieSim emulator - Write working assembly code that executes the logical programming construct:

Include comments to your code!

Write a working assembly code that uses **stack parameter passing linkage** to call a subroutine labeled “**sub\_1**” to subtract two numbers and return the results to the main routine. t = (r – s)

Your main routine must:

* 1. load your two parameters (r,s) from memory and push them onto your stack.
  2. call sub\_1
  3. pop the result from your stack and store it into memory label ‘t: ‘.

Your sub-routine must:

1. pop your two parameters from your stack
2. subtract the parameters as shown in the problems formula
3. push the result onto the stack
4. return to your ‘main’ routine

Please initialize ‘t: 0’ to start your code. t = (r – s)

Use care not to let your memory stack step on your code.

Answer:

1. (15 pts) Using the MarieSim emulator - Write working assembly code that executes the array initialization C code shown below:

Include comments to your code!

*#include <stdio.h>*

*int main () {*

*int n[ 10 ]; /\* n is an array of 10 integers \*/*

*int i;*

*/\* initialize elements of array n to 0 \*/*

*for ( i = 0; i < 10; i++ ) {*

*n[ i ] = i + 100; /\* set element at location i to i + 100 \*/*

*}*

*return 0;*

*}*