1. (10) State the five axioms and their duals that define the Boolean algebra ($2^1$).

1 a. $x = 0$, $\frac{1}{x} \neq 1$

1 b. $x = 1$, $\frac{1}{x} \neq 0$

2 a. $1 \div x = 0$, then $x = 1$

2 b. $1 \div x = 1$, then $x = 0$

3 a. $0 \cdot 0 = 0$

3 b. $1 + 1 = 1$

4 a. $1 \cdot 1 = 1$

4 b. $0 + 0 = 0$

5 a. $0 \cdot 1 = 1 \cdot 0 = 0$

5 b. $1 + 0 = 0 + 1 = 1$

2. Given the following logic circuit with the following timing specifications:
D flip-flops: setup time = 2 ns, hold time = 1 ns and output propagation delay = 8 ns
Gates: propagation delay = 5 ns

a) (10) determine the maximum propagation delay for the circuit; you must show all work and state all assumptions to get full credit.

Critical path either from $x$ or $y$

$\tau_{pd} = (\tau_{pd})_{\text{mu}} + (\tau_{pd})_{\text{mu}} + (\tau_{\text{setup}}) + (\tau_{pd})_{\text{D}}$

$\tau_{pd} = 5 \text{ ns} + 5 \text{ ns} + 2 \text{ ns} + 8 \text{ ns}$

$\tau_{pd} = 20 \text{ ns}$

b) (5) determine the maximum frequency that the circuit can operate; you must present a number value (not an arithmetic expression) to get credit.

$f_{\text{max}} = \frac{1}{\tau_{pd}} = \frac{1}{20 \text{ ns}} = \frac{1}{20 \times 10^{-9}} = \frac{1000}{20} \text{ MHz}$

$f_{\text{max}} = 50 \text{ MHz}$
3. (20 @ 2) Provide a very brief answer to each of the following questions; the objective is to assess that you know each of the concepts presented.

a) What are the names of the five components that constitute a computer?
   
   input, output, memory, control (state machine), data path (alu)

b) What is the difference between a Harvard and a Princeton computer architecture?
   
   Harvard has separate memories, one for data and one for instructions;
   Princeton has one memory for both.

c) What are the four basic operations that constitute a computer machine cycle?
   
   fetch, decode, execute, write-back

d) What does the program counter do?
   
   contains the address of the next instruction to fetch

e) What is the basic architectural difference between CISC and RISC computers?
   
   RISC computer has much fewer machine instructions

f) Given that the two registers R0 and R1 are two of the seven elements that constitute the programmer's register model for the VBC1 as used in EASY1, what are the names of the other five elements/registers?
   
   input
   output
   IE (instruction register)
   instruction memory
   PC (program counter)

g) What is the function of an assembler macro instruction?
   
   allows a sequence of machine code to be referenced

h) What is metastability in a logic circuit?
   
   oscillations caused by asynchronous input to synchronous circuit

i) What function does a debouncing circuit perform?
   
   allows single response for an input switch or button

j) If a system clock frequency is 50 MHz and a 3-bit counter is used to divide the frequency, what is the lowest frequency that can be generated for the 3-bit register?
   
   $$f_{\text{low}} = \frac{50 \text{ MHz}}{2^3} = \frac{50}{8} \text{ MHz} = 6.25 \text{ MHz}$$
4. The compressed truth table for the data path control signals for the two VBC1 instructions SR0 and ADD are given below with - being a don’t care.

<table>
<thead>
<tr>
<th>IR</th>
<th>M1 M2 M3 M4 M5 M6 LOAD_R0 LOAD_R1 LOAD_OP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR0 R0,R0</td>
<td>10000000 0 - 1 0 0 0 1 0 0</td>
</tr>
<tr>
<td>SR0 R0,R1</td>
<td>10001000 0 - 1 1 0 0 1 0 0</td>
</tr>
<tr>
<td>SR0 R1,R0</td>
<td>10010000 0 - 1 0 1 0 0 0 1 0</td>
</tr>
<tr>
<td>SR0 R1,R1</td>
<td>10011000 0 - 1 1 0 0 0 1 0 0</td>
</tr>
<tr>
<td>ADD R0,R0</td>
<td>01000000 0 0 1 0 0 0 1 0 0</td>
</tr>
<tr>
<td>ADD R0,R1</td>
<td>01001000 0 0 1 1 0 0 0 1 0 0</td>
</tr>
<tr>
<td>ADD R1,R0</td>
<td>01010000 0 1 1 0 0 0 0 1 0</td>
</tr>
<tr>
<td>ADD R1,R1</td>
<td>01011000 0 1 1 1 0 0 0 1 0</td>
</tr>
</tbody>
</table>

a)(5) determine a Boolean function for the control signal M4.
\[ M_4 = \overline{IR(7)} \cdot \overline{IR(6)} \cdot \overline{IR(5)} \cdot IR(3) + \overline{IR(7)} \cdot \overline{IR(6)} \cdot \overline{IR(5)} \cdot \overline{IR(3)} \]

or \[ M_4 = \overline{IR(3)} \]

b)(5) determine a Boolean function for the control signal LOAD_R0.
\[ LOAD_R0 = \overline{IR(7)} \cdot \overline{IR(6)} \cdot \overline{IR(5)} \cdot \overline{IR(4)} + \overline{IR(7)} \cdot \overline{IR(6)} \cdot \overline{IR(5)} \cdot \overline{IR(4)} \]

or \[ LOAD_R0 = \overline{IR(4)} \]

c)(5) write a VHDL selected signal assignment for the control signal LOAD_R0.

with IR( 7 downto 5 ) select
| LOAD_R0 <= not IR(4) when "100", |
| not IR(4) when "010", |
| '0' when others ; |

5. A 1-bit data path computer has two registers R0 and R1, an input port DIN, an output port DOUT, and three instructions: IN DR, OUT DR, and NAND DR, SR, where DR is the destination register and SR is the source register. The ports DIN and DOUT contain Boolean values, i.e., either a 0 or 1. The transfer function forms for the instructions are:

- **IN DR**: DR <- DIN
- **OUT DR**: DOUT <- DR
- **NAND DR, SR**: DR <- not(DR and SR)

Write an assembly language program to perform the following Boolean functions:

a) (5) NOT

- \[ \text{IN } R\Phi \]
- \[ \text{NAND } R\Phi, R\Phi \]
- \[ \text{OUT } R\Phi \]

b) (5) AND

- \[ \text{IN } R\Phi \]
- \[ \text{IN } R1 \]
- \[ \text{NAND } R\Phi, R1 \]
- \[ \text{NAND } R\Phi, R\Phi \]
- \[ \text{OUT } R\Phi \]

Note: \[ x + y = \overline{x \cdot y} \]

c) (10) OR

- \[ \text{IN } R\Phi \]
- \[ \text{NAND } R\Phi, R\Phi \]
- \[ \text{NAND } R\Phi, R1 \]
- \[ \text{NAND } R\Phi, R1 \]
- \[ \text{OUT } R\Phi \]
6. Using good comments with your code, write an assembly language program for the VBC1 using its eight instructions to perform the following algorithms. The first and last line of your program is given below; the other instructions and comments are to be determined by you. To be unambiguous in your written answer, use O = letter O and 0 = number zero. Remember that your program is limited to 16 machine code entries.

a)(10) Perform the arithmetic calculation: \( y = (8/2 + 1) - 2 \) where 8 is entered through the input port switches and 1 is entered via an instruction, i.e., immediate data, and the answer \( y \) is provided through the output port to the LEDs.

\[
\begin{align*}
\text{IN R0} & \quad ; \text{input the number 8 via the nexys switches sw3:sw0 = (1000)} \\
\text{SR}O & \quad \text{R0}; R0 \text{ divide 8 by 2} \\
\text{ADDI} & \quad \text{R0}, 1 \quad ; \text{add 1} \\
\text{ADDI} & \quad \text{R0}, 14 \quad ; \text{subtract 2} \\
\text{MOV} & \quad \text{R1}, \text{R0} \quad ; \text{move answer to R1}
\end{align*}
\]

\[
\begin{align*}
\text{note:} & \quad 2_{10} = 0010_{2} \\
\text{complement:} \quad 1101 & \quad (1's \text{ complement}) \\
+1 & \quad 0001 \\
2's \text{ complement:} \quad 1110 & \quad (-2_{10}) \\
equals & \quad 14_{10}
\end{align*}
\]

\[
\begin{align*}
\text{OUT R1} & \quad ; \text{output answer y to ld3:ld0 = (0011)}
\end{align*}
\]

b)(10) With a loop implemented with the JNZ instruction, calculate the binary sequence 3, 2, 1, 0 and output each value through the output port to the LEDs. You must use the JNZ instruction for the calculation to receive any credit.

\[
\begin{align*}
\text{one solution without last statement - announced during test didn't have to use} \\
\text{LOADI R0, 3} & \quad ; \text{initial output value loaded in R0} \\
\text{LOAD} & \quad \text{R1}, 4 \quad ; \text{initialize loop counter to 4} \\
\text{LOOP:} & \quad \text{OUT R0} \quad ; \text{output value in sequence to LEDs} \\
\text{ADDI} & \quad \text{R0}, 15 \quad ; \text{decrement value in sequence} \\
\text{ADDC} & \quad \text{R1}, 15 \quad ; \text{decrement loop counter} \\
\text{JNZ} & \quad \text{R1}, \text{LOOP} \quad ; \text{jump to loop if R1 \neq 0}
\end{align*}
\]

\[
\begin{align*}
\text{one solution with last statement} \\
\text{LOAD} & \quad \text{R0, 3} \quad ; \text{initial output value loaded in R0} \\
\text{LOOP:} & \quad \text{OUT R0} \quad ; \text{output value in sequence to LEDs} \\
\text{ADDI} & \quad \text{R0}, 15 \quad ; \text{decrement value in sequence} \\
\text{JNZ} & \quad \text{R0, LOOP} \quad ; \text{jump to loop if R0 \neq 0}
\end{align*}
\]

\[
\begin{align*}
\text{OUT R0} & \quad ; \text{output last value 0 to ld3:ld0 = (0000)}
\end{align*}
\]