Please enter your name and Athena login name in the spaces above. Enter your answers in the spaces provided after each question. You can use the extra white space and the backs of the pages for scratch work.

**Problem 1. Assembly Language (6 points)\(^\text{1}\)**

(A) (1 point) What assembly instruction could a compiler use to implement \(y = x \times 8\) on the Beta assuming that MUL and MULC are not available? Assume \(x\) is in R0 and \(y\) is in R1.

Equivalent assembly instruction: \(\text{SHLC}(R0, 3, R1)\)

(B) (4 points) Assume that the registers are initialized to: R0=8, R1=10, R2=12, R3=0x1234, R4=24 before execution of each of the following assembly instructions. For each instruction, provide the value of the specified register or memory location. **If your answers are in hexadecimal, make sure to prepend them with the prefix 0x.**

1. SHL(R3, R4, R5)  Value of R5: \(0x39000000\)  \(0x1234 \approx 24\)
2. ADD(R2, R1, R6)  Value of R6: \(22\)  \(12+10\)
3. ADD(R0, 2, R7)  Value of R7: \(20\)  \(8+12=20\)
4. ST(R1, 4, R3)  Value stored: \(10\) at address: \(0x1238\)

(C) (1 point) Suppose you have a circuit that is built completely out of NOR2 gates. Assume the \(t_{PD}\) of the NOR2 gate is 3ns. You are given non-ideal pipeline registers whose \(t_{PD}=2nS\) and \(t_{Setup}=1nS\). If you can only add registers and want to pipeline your circuit for maximum throughput, what throughput can you achieve?

\[
\begin{align*}
t_{PD, \text{stage}} &= 2 + 3 + 1 = t_{Adv} \\
t_{PD, \text{Adv}} &= t_{PD, \text{stage}} + t_{s, \text{REG}}
\end{align*}
\]

\[
\text{Maximum throughput (1/ns)}: \frac{1}{6}
\]
Problem 2. Finite State Machines (6 points)

(A) (1 point) For each of the following FSMs please indicate if they are or are not well formed. Note that the state names have been omitted for clarity; you may assume the state names are unique.

 FSM A (circle one): Well Formed / Not Well Formed
 FSM B (circle one): Well Formed / Not Well Formed
 FSM C (circle one): Well Formed / Not Well Formed

(B) (4 points) Given the partially completed truth table and FSM diagram below. Complete all the missing entries in the truth table and the FSM diagram. The FSM is a Moore machine, i.e., the Out signal is determined only by the current state. In each state circle, the top entry is $S_iS_o$ and the bottom entry is the value of Out. Make sure that you have labeled all missing states, inputs, and outputs, and that you have added and labeled any missing transitions in the FSM.

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_0$</th>
<th>$I_{in}$</th>
<th>$S'_1$</th>
<th>$S'_0$</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

(C) (1 point) If this FSM is implemented using a 2-bit state register and a ROM, what size ROM would be needed? Please specify the number of locations (entries) of the ROM, and the width of each entry.

Number of locations in ROM: $2^3 = 8$

Width of each ROM entry (bits): $2 + 1 = 3$
Problem 3. Pipelining (8 points)

(A) (1 point) You are provided with the circuit shown below. Each box represents some combinational logic. The number in each box is the $t_{PD}$ of that combinational logic. The circuit has two inputs, X and Y, and one output Out. Pay close attention to the direction of the arrows especially the arrows shown in bold. What is the latency and throughput of this combinational circuit?

![Circuit Diagram]

Latency (ns): 22
Throughput (1/ns): $\sqrt{22}$

(B) (4 points) Draw contours through the circuit above to produce a valid pipelined circuit whose $t_{CLK} = 9\text{ns}$ with minimum latency. Extra copies of the diagram are included below. Please use a large dot to indicate the location of each pipeline register. Assume that you have ideal pipeline registers ($t_{PD}=t_{CD}=t_{Setup}=t_{Hold}=0$ ns). Pay close attention to the direction of each arrow to ensure that you produce a valid pipeline. What is the latency and throughput of this pipelined circuit?

$\frac{X \cdot t_{at}}{X} = \text{Latency (ns): } \frac{x + 2}{9} = 2\frac{7}{9}$

$\frac{1}{t_{at}} = \text{Throughput (1/ns): } \frac{1}{9}$

![Pipelined Circuit Diagrams]
(C) (3 points) You are now asked to consider the performance of this circuit using different clock periods while achieving the minimum latency. For each suggested $t_{CLK}$, specify whether or not you can create a valid pipelined circuit using that clock period. If you can, then provide the latency and throughput of the resulting circuit and specify the number of registers at each input. If it results in an invalid pipeline, enter NA for the rest of the row.

Extra copies of the circuit diagram are provided below.
Problem 4. Stack Detective (10 points)

You are given an incomplete listing of a C program (shown below) and its translation to Beta assembly code (shown on the right):

```c
int f(int x, int y) {
    x = (x >> 1) + y;
    if (y == 0) return x;
    else return ???;
}
```

(A) (2 points) What is the missing C source code corresponding to ??? in the above program?

C source code: \( f(\_\_\_, y-1) \)

(B) (1 point) Suppose the instruction bearing the tag ‘zz:’ were eliminated from the assembly language program. Would the modified procedure work the same as the original procedure?

Work the same (circle one)? YES . NO

The procedure \( f \) is called from an external procedure and then execution is stopped just prior to one of the executions of the instruction labeled ‘rtn:’. The addresses and contents of a region of memory are shown in the table on the right; all addresses and data values in the table are in hex. When execution is stopped BP contains the value 0x14C and SP contains the value 0x150.

(C) (1 point) What are the arguments to the currently active call to \( f \)?

Most recent arguments (in hex): x = 0x\_\_\_, y = 0x\_\_\_

(D) (1 point) If you can tell from the information provided, specify the arguments to the original call to \( f \), otherwise select CAN’T TELL.

Original arguments (in hex): x = 0x\_\_\_, y = 0x\_\_\_, or CAN’T TELL

(E) (2 points) What is the missing value in location 0x12C?

\[ f(x, y) \rightarrow f(3, 2) \rightarrow f(0, 1) \rightarrow f(0,0) \rightarrow \text{return} \, z \]

Contents of location 0x12C (in hex): 0x\_\_\_

(F) (1 point) What is the hex address of the instruction labeled rtn?:

Address of instruction labeled rtin: (in hex): 0x\_\_\_

(G) (1 point) What is the hex address of the BR instruction that called \( f \) originally?

Address of original call (in hex): 0x\_\_\_, or CAN’T TELL

(H) (1 point) What value will be returned to the original caller?

Return value for original call (in hex): 0x\_\_\_

END OF QUIZ 2!