CSI404 Spring 2002
Exam Key
1. (15 points) Using a diagram and arrows like those from lecture, show any bubbles/stalls created by the following pair of MIPS instructions.

\[
\begin{align*}
\text{add} & \ $s1, $s2, $s3 \ # \ $s1 \leftarrow ($s2) + ($s3) \\
\text{sw} & \ $s1, 32 \ $s4 \ # \ M[(\$s4) + 32] \leftarrow ($s1)
\end{align*}
\]

Answer this question in two parts:

a) Show the bubble/stall without any attempts to lessen it.

b) Show the bubble after forwarding.

For a MIPS \textit{sw} instruction, assume that the source register (\$s1 here) is read in the Memory Access stage (and not the initial Register Reading stage).

Model Answer:

a) Without forwarding, there is a bubble/stall of one stage (two cycles).

b) With forwarding, there is no bubble/stall.

2.
2. (25 points) The contents of two registers can be swapped by the following sequence of operations:

\[
\begin{align*}
Y &\leftarrow X \oplus Y \\
X &\leftarrow X \oplus Y \\
Y &\leftarrow X \oplus Y
\end{align*}
\]

Design a register-transfer level circuit to implement this operation. Assume the following.

- There are two 16-bit registers, \(X\) and \(Y\). They have the usual \(Clock\) and \(Load\) inputs, and an output, \(X_i\) or \(Y_i\) for each bit.
- There is a two-bit counter register, \(C\), that increments with each upward clock transition. There is a \(Clock\) signal input and an output for each counter bit, \(C_i\).
- The three operations for the swap happen at counter times 00, 01 and 10. Nothing happens at counter time 11. You may assume the counter is always at 00 when the swap begins.
- There is an external control signal, \(S\). The swap operation only takes place if \(S = 1\).

In your circuit be sure to connect appropriate elements to the system clock.
3. (10 points) What is the principal advantage to having a large number of registers on a processor?

Model answer:

Since it is faster to access data in registers than in main memory, if there are a large number of registers, most data the program needs can be stored in registers and accessed there. This avoids having to load and save them from memory, and speeds up the execution of the program using the data.
4. (25 points) Design an optimal circuit corresponding to the following truth table. The inputs are $W$, $X$, $Y$ and $Z$. The circuit output is $F$. Show your work.

<table>
<thead>
<tr>
<th>$W$</th>
<th>$X$</th>
<th>$Y$</th>
<th>$Z$</th>
<th>$F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Model answer:
The standard way to make an optimal circuit is to create a Karnaugh map from the truth table.

From that, derive the optimal formula:

$$F = WX + XYZ + WYZ + WXYZ$$

A circuit can be created from the formula:
5. (25 points) Given the following as the values of certain registers and memory addresses, assume that the processor is about to fetch and execute the instruction at the location given by the \( PC \) register. Give the value of all register or memory locations changed by the execution of the instruction.

\[ AC: \quad 0111 \ 1111 \ 1111 \ 0101 \]
\[ DR: \quad 0000 \ 1111 \ 0110 \ 0101 \]
\[ IR: \quad 1101 \ 0111 \ 1101 \ 0000 \]
\[ TR: \quad 1001 \ 1101 \ 0011 \ 1101 \]
\[ AR: \quad 0001 \ 1111 \ 0100 \]
\[ PC: \quad 0001 \ 1111 \ 0110 \]

Memory location \( 500_{10} \): \[ 0110 \ 0111 \ 1101 \ 0000 \]
Memory location \( 1000_{10} \):: \[ 0000 \ 0000 \ 0000 \ 0001 \]
Memory location \( 2000_{10} \): \[ 1111 \ 1111 \ 1111 \ 1111 \]
Memory location \( 3000_{10} \): \[ 0000 \ 0000 \ 0000 \ 0000 \]

Model Answer:

\[ AR: \quad 0111 \ 1101 \ 0000 = 2000_{10} \]
\[ IR: \quad 0110 \ 0111 \ 1101 \ 0000 = \text{ISZ} \ 2000 \]
\[ PC: \quad 0001 \ 1111 \ 0110 = 502_{10} \]
\[ DR: \quad 0000 \ 0000 \ 0000 \ 0000 = 0 \]

Memory location \( 2000_{10} \): \[ 0000 \ 0000 \ 0000 \ 0000 = 0 \]

Explanation:
The value in the PC register is \( 500_{10} \). This means the instruction to be executed is the one in \( M[500] \). That instruction is \( 0110 \ 0111 \ 1101 \ 0000 \). That decodes as

Addressing Mode: Direct (bit 15 is 0)
The instruction is ISZ (opcode 110)
The address field is \( 0111 \ 1101 \ 0000 \) (\( = 2000_{10} \))

Given this instruction, addressing mode and address, the computer will do the following:

\[ T_0: \quad AR \rightarrow PC \]
\[ T_1: \quad IR \rightarrow M[AR] \]
\[ PC \rightarrow PC + 1 \]
\[ T_2: \quad \text{Decode the operation in the IR.} \]
\[ AR \rightarrow IR(0-11) \]
\[ T_3: \quad \text{(nothing)} \]
\[ T_4: \quad DR \rightarrow M[AR] \]
\[ T_5: \quad DR \rightarrow DR + 1 \]
\[ T_6: \quad M[AR] \rightarrow DR \]
\[ \text{since \( DR = 0 \), } PC \rightarrow PC + 1 \]
\[ SC \rightarrow 0 \]

With the values in the registers and memory locations given, the following will occur:

\[ T_0: \quad AR \rightarrow 500_{10} \]
T1: \( IR \) \( M[500_{10}] = 0110 0111 1101 0000 \)
\( PC \) \( 501_{10} \)
T2: Decode the operation in the \( IR \).
\( AR \) \( 0111 1101 0000 = 2000_{10} \)
T3: (nothing)
T4: \( DR \) \( 1111 1111 1111 1111 = -1_{10} \)
T5: \( DR \) \( 0000 0000 0000 0000 = 0_{10} \)
T6: \( M[2000_{10}] \) \( 0000 0000 0000 0000 = 0_{10} \)
\( PC \) \( 502_{10} \)
\( SC \) \( 0 \)