Lab Assignment 03

Consider the following data path and the set of 9 instructions of the single-cycle processor:

- The memory-reference instructions \texttt{lw} (load word) and \texttt{sw} (store word)
- The arithmetic-logical instructions \texttt{add}, \texttt{sub}, \texttt{and}, \texttt{or}, and \texttt{slt}
- The instructions \texttt{beq} (branch equal) and \texttt{j} (jump)

In the following pages you will be asked to add new instructions to the processor and make appropriate changes on the data path in order to support the existing 9 instructions and the new ones. You need to clearly specify new functionality on the existing units (such as ALU) and/or new components you may need (MUXes, control lines, gates, etc), and draw them on the data path figure.

Completely specify the logic values of the existing control signals (RegDst, Branch, MemRead, etc) and of the new ones you may need to introduce for the new instruction. Consider each new instruction independently from the other new instructions. Make copies of this figure as needed. See the 1-page large figure at the end.
1. Add the “And Immediate” instruction to the single cycle processor.
The opcode: \( c_{\text{hex}} \)
\[
\text{andi } rt, rs, \text{ imm}
\]
\[
\begin{array}{cccc}
6 & 5 & 5 & 16 \\
\text{opcode} & rs & rt & \text{imm}
\end{array}
\]
\[
R[rt] \leftarrow R[rs] + \text{ZeroExtImm}
\]
\[
\text{PC} \leftarrow \text{PC} + 4
\]

2. Add the “Nor” instruction to the single cycle processor.
The opcode: \( 0_{\text{hex}} \); The function: \( 27_{\text{hex}} \)
\[
nor \ rd, rs, rt
\]
\[
\begin{array}{ccccccc}
6 & 5 & 5 & 5 & 5 & 6 \\
\text{opcode} & rs & rt & rd & \text{shamt} & \text{func}
\end{array}
\]
\[
R[rd] \leftarrow (R[rs] \mid R[rt])'
\]
\[
\text{PC} \leftarrow \text{PC} + 4
\]

3. Add the “Set Less Than Unsigned” instruction to the single cycle processor.
The opcode: \( 0_{\text{hex}} \); The function: \( 2b_{\text{hex}} \)
\[
\text{sltu } rd, rs, rt
\]
\[
\begin{array}{ccccccc}
6 & 5 & 5 & 5 & 5 & 6 \\
\text{opcode} & rs & rt & rd & \text{shamt} & \text{func}
\end{array}
\]
\[
\text{if } R[rs] \lt R[rt]
\]
\[
R[rd] \leftarrow 1
\]
\[
\text{else}
\]
\[
R[rd] \leftarrow 0
\]
\[
\text{PC} \leftarrow \text{PC} + 4
\]

4. Add the “Branch On Not Equal” instruction to the single cycle processor.
The opcode: \( 5_{\text{hex}} \)
\[
bne \ rt, rs, \text{ label}
\]
\[
\begin{array}{cccc}
6 & 5 & 5 & 16 \\
\text{opcode} & rs & rt & \text{offset}
\end{array}
\]
\[
\text{if } R[rs] \neq R[rt]
\]
\[
\text{PC} \leftarrow \text{PC} + 4 + \text{SignExtOffset}
\]
\[
\text{else}
\]
\[
\text{PC} \leftarrow \text{PC} + 4
\]

5. Add the “Store Byte” instruction to the single cycle processor.
The opcode: \( 2b_{\text{hex}} \)
\[
sb \ rt, rs, \text{ offset}
\]
\[
\begin{array}{cccc}
6 & 5 & 5 & 16 \\
\text{opcode} & rs & rt & \text{offset}
\end{array}
\]
\[
M[R[rs]+\text{SignExtOffset}](7:0) \leftarrow R[rt](7:0)
\]
\[
\text{PC} \leftarrow \text{PC} + 4
\]