Assignment No. 2

Questions about the DLX.

1. Architecture. Why define $R0$ to be the constant 0^{32} (i.e., string of 32 zeros)? Where and how does it help?

2. The PC register. When and where should the PC be incremented (by 4) to compute the address of the next instruction? Note that the suggested data-path always increments the PC (even with a jump and a branch). Is it a good idea not to increment when there is a jump? How would it effect the compiler and the data-path?

3. Format of R-type instructions. Consider swapping the order of the fields in an R-type instruction:

   opcode, RS1, RD, RS2, SA, function

   Which parts of the data-path does it simplify and which does it complicate? Is it a good idea?

4. Semantics of store instruction. Compare the effect:

   $$M[RD + Sext(Immediate)] \leftarrow RS1$$

   with the suggested effect. Which parts of the data-path does it simplify and which does it complicate? Is it a good idea?

5. Data-path design. Why is the 0^{32} output of the IR-environment needed?

6. Data-path design. Why can B write to both buses (S1 and S2), whereas A can only write to bus $S1$?

7. Data-path design. Why does the PC-environment have an output of 4?