Assignment No. 1

1. Mapping of bytes to memory. Consider the bytes B_3, B_2, B_1, B_0 as a word. How is this word stored in memory (say, in address $x[31:0]$)? Note that x is a multiple of 4, namely, $x[1:0] = 00$. There are two conventions:

   big endian: B_0 is stored in address x, B_1 in address $x+1$, B_2 in address $x+2$, B_3 in address $x+3$.

   little endian: B_0 is stored in address $x+3$, B_1 in address $x+2$, B_2 in address $x+1$, B_3 in address x.

   Try to justify each convention. How should a word be loaded into a register and then sent to the ALU?

2. Memory organization. We divide the memory into 4 banks; the width of each bank is one byte. We index the banks using two bits: $b_1 b_0$. During load instructions (read from memory), access to address $x[31:0]$ is implemented by accessing in parallel all 4 banks. The accessed address in all banks is identical, namely, $x[31:2]$ (the last two bits are ignored). The processor receives 4 bytes from memory.

   Explain how the following instructions are implemented (i.e., the effect of the instruction the address on the required shift): load byte (lb), load half-word (lh), load word (lw).

   During store instructions (write to memory), the two least significant bits of the address and the width of the data to be stored determine the banks in which the write is performed.

   Explain how the following instructions are implemented (i.e., the effect of the instruction the address and the required shift): store byte (sb), store half-word (sh), store word (sw).

3. Explain the implementation of the following instructions in the data-path (depicted in Figure 2) of the DLX: lh,addi,sgri,beqz,sll,add,sgr,j,jal.