CSEE 3827: Problem Set 5 Due: 4/8/09

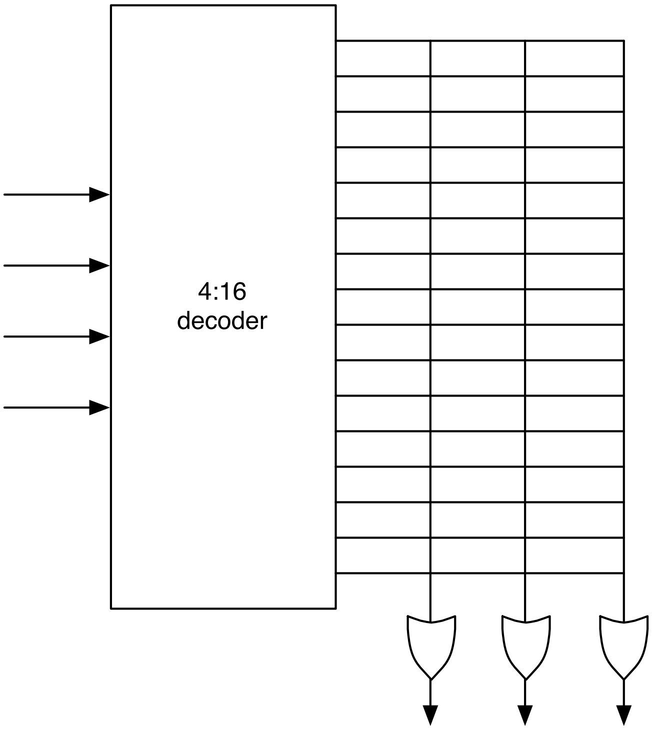
Complete the following problems. Be sure to show your work for partial credit.

1. Shown below is a drawing of a 16x3 ROM (16 words, each 3 bits wide). In class we examined the design of ROM in the context of programmable logic devices. By labeling the inputs and outputs and adding 'X's to the appropriate connections in the figure below, show how this ROM could be configured to implement the following three logic functions:

   (a) $X = A \cdot B + B \cdot \overline{C} \cdot D + \overline{A} \cdot \overline{B}$

   (b) $Y = A \cdot B + B \cdot D$

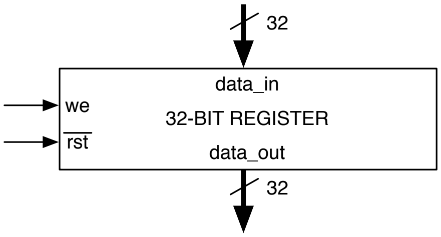
   (c) $Z = A + B + C + D$



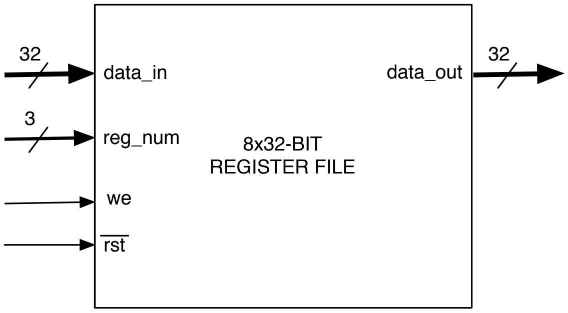
2. For this exercize you will design a register file. A register file is an array of registers which are read and written using a register number. A register's number is typically the index of the register in the file (i.e., the first register is register 0, the second is register 1, and so on.) The register number addresses the register array in much the same manner as a memory address accesses a memory array. This design will combine your understanding of registers and of memory design. The register file specification is as follows:

   - The register file is to contain 8 registers.
   - Each register is a 32-bit wide parallel load register with the interface shown in (a) below.
   - The top-level interface to the register file is shown in (b) below.
   - Only one register in the file can be read or written at a time.
   - The *we* control input to the register and to the register file itself stands for "write enable". $we = 1$ for a write operation and $we = 0$ for a read operation.

- The $\overline{rst}$ control input is an active low reset signal. When $\overline{rst} = 0$ for an individual register, the contents of the register will be reset. When $\overline{rst} = 0$ for the register file, all registers in the file will be reset.



(a)



(b)

Draw a block diagram of a register file with the behavior specified above and the external interface shown in (b). You needn't implement the register itself. You may simply use the registers shown in (a) as black boxes.