Verilog Assignment -2 Q1 Report

Design Overview:

Register Bank Module

The register bank module is written in Verilog and consists of:

- Data Registers: eight 16-bit data registers labeled R0 to R7, implemented as an array.
- Data Input (data_in): A 16-bit input port for loading data into the selected register.
- Register Number (reg_no): A 3-bit input to specify the source or destination register.
- Clock (Clk): Controls synchronous data loading.
- Load (load): A control signal that, when asserted, enables data to be loaded into the selected register.
- Enable (enable): A control signal that, when asserted, enables the output of the selected register.
- Data Output (data_out): A 16-bit output port providing selected register data (high-impedance if not enabled).
- Reset (reset): A control signal to reset all registers to zero.

Functionality

- **Parallel Load:**

  Data can be loaded into any register (R0 to R7) using "data_in" and "reg_no" inputs when "load" is set to 1.

  This is done as, if(load) , registers[reg_no] <= data_in , which means that take the 16-bit value provided at data_in and store it into the register specified by the 3-bit value reg_no within the registers array.
- **Synchronous Loading**:

  Data loading is synchronized with the "clk" signal.
- **Output Enable:**

  "enable" controls whether data from the selected register is available at "data_out" (high-impedance if not enabled).

  This is done as, assign data_out = (enable) ? registers[reg_no] : 16'bz, which means that, assign the value of registers[reg_no] to data_out if the enable signal is 1. If enable is 0, assign a high-impedance state (16'bz) to data_out.
- **Reset:**

"reset" signal resets all registers to zero.