BSUCA12 Register File

Must be submitted electronically by Midnight on Friday, March 2

The register file is the collection of registers that the processor uses in its computations. The BSUCA12 has eight 16-bit general-purpose registers, `$r0-$r7`, where `$r0` is *always* equal to zero and `$r3` is defaulted to the base address of the static data segment equal to `0xC000`. At any given time, we can be reading two registers (identified by `readRegA_sel[2:0]` and `readRegB_sel[2:0]`). Reading is not clocked. On any rising edge of the clock we can write one register (identified by `writeReg_sel[2:0]`), if `writeReg_en` is true, with the value on the `writeReg_dat` port. The signal `reset` resets all registers (sets their values to zero's with the exception of `$r3`). You'll want to be able to reset the registers when you boot up your BSUCA12. Your registers should have a setup of 3ns and a hold of 2ns.

The figure below is a block diagram of the register file, and it shows the signal names that you all MUST use in your design (to facilitate testing and grading). After implementing your register file, you should test it thoroughly to demonstrate that it works correctly. This part of the project will be graded by running tests that I wrote, so don't assume you can ignore bugs that may manifest themselves by running only the tests provided in the test bench. The high level module must be named `regFile` and file name must be `regFile.v` Names of internal wires/regs are unrestricted.