Name: Christian Miranda and Zachary Huang

Date: 06/02/2025

Hitachi SH-2 CPU Design

The Hitachi SH-2 is implemented in the `sh2_cpu.vhd` file, and tested in `sh2_cpu_tb.vhd`.

Control Unit and Instruction Register

The control unit and instruction register are implemented in `sh2_control.vhd`. They are tested in `sh2_cpu_tb.vhd`.

General-Purpose Registers

The register array is implemented in `sh2_reg.vhd` and tested in `sh2_reg_tb.vhd`.

ALU

The ALU is implemented in `sh2_alu.vhd` and tested in `sh2_alu_tb.vhd`.

Status Register

The status register is implemented in `sh2_cpu.vhd`, and tested along with the entire CPU in `sh2_cpu_tb.vhd`.

Data Memory Access Unit

The data memory access unit (DMAU) is implemented in `sh2_dmau.vhd`. Its test bench is located in the `dmau_test/` directory. It can be tested by invoking `make run` from within the `dmau_test/` directory.

Program Memory Access Unit

The data memory access unit (PMAU) is implemented in `sh2_pmau.vhd`. Its test bench is located in the `pmau_test/` directory. It can be tested by invoking `make run` from within the `pmau_test/` directory.

Design Testing

Each instruction is tested by assembling an assembly file that uses the instruction under test and writes expected results to memory. The actual memory contents are compared with the expected memory contents to check for correctness.

The assembly files are located in the `asm` and are postfixed with `.asm`. Each assembly file has an associated `.expect` file which describes the expect memory contents, as well as a `.dump` file that shows the memory contents after the tests are run.

The SH-2 CPU can be tested by invoking `make test` from the top level directory. This compiles any `.vhd` files that have changed and then enters the `asm/` directory and compiles and `.asm` files that have changed. Assembling the assembly files generates binary `.bin` files which are read by the test bench and then decoded and run by the CPU.

Testing requires `ghdl` with either the GCC or LLVM backend. The `AS` macro-assembler, linked below, is required for assembling the assembly files, although it is possible to test the SH-2 CPU without this by invoking `make test-bin`, which uses the supplied binaries for each assembly file.

http://john.ccac.rwth-aachen.de:8000/as/download.html

Extra Credit - Multi-bit shift Instructions (20 points)

All shift instructions are implemented.

Extra Credit - Implementation (60 points)

The SH-2 CPU design was implemented for the Xilinx Spartan 3E XC3S1200EFGG3204C.

The resource usage summary is provided below. It is in `SH2CPU.syr` in the `ise_work/` directory.

```
Device utilization summary:
---------------------------

Selected Device : 3s1200efg320-4

 Number of Slices:                     2612  out of   8672    30%
 Number of Slice Flip Flops:           1005  out of  17344     5%
 Number of 4 input LUTs:               4967  out of  17344    28%
 Number of IOs:                          77
 Number of bonded IOBs:                  75  out of    250    30%
 Number of GCLKs:                         3  out of     24    12%
```

The timing report is in `SH2CPU.par` in the `ise_work/` directory.

```
----------------------------------------------------------------------------------------------------------
  Constraint                                |    Check    | Worst Case |  Best Case | Timing |   Timing
                                            |             |    Slack   | Achievable | Errors |    Score
----------------------------------------------------------------------------------------------------------
  Autotimespec constraint for clock net clo | SETUP       |         N/A|    32.470ns|     N/A|           0
  ck_IBUF                                   | HOLD        |     1.064ns|            |       0|           0
----------------------------------------------------------------------------------------------------------
  Autotimespec constraint for clock net con | SETUP       |         N/A|     5.782ns|     N/A|           0
  trol_unit/state_FSM_FFd2                  | HOLD        |     1.390ns|            |       0|           0
----------------------------------------------------------------------------------------------------------
```

Thus our maximum clock speed is around 30.8 MHz.