COL216 Assignment 2

Stage 3

Rishabh Dhiman
2020CS10837

25 February 2022

1 Objective

Construct an ALU, a register file, a program memory unit and a data memory unit for a rudimentary ARM processor in VHDL.

2 Technical Details

- The VHDL code was analyzed, and simulated using GHDL 1.0.0.
- The VHDL code was synthesized using Quartus 21.1.
- The waveform viewer used is GTKWave Analyzer v3.3.104.

3 Documentation

The submission contains modified

- `processor.vhdl` which implements the processor as a multicycle design
- `program_counter.vhdl` without a dedicated adder
- `memory.vhdl` a combined memory module, the program information is still hardcode into the memory
- `decoder.vhdl`, `types.vhdl` which contain minor changes.

There's a single testbench `processor_tb.vhdl`.

Along with the code, the waveforms on simulating the testbench, in the form of `.ghw` and `.pdf` files are stored in the `simulation` folder. Two netlists of the processor have been added in the `synthesis` directory, one corresponding to a one-hot encoding of the enumerated type and other corresponding to a sequential encoding.

4 Testing Procdure

The code was analyzed, and simulated using GHDL. It was synthesized using Quartus 21.1.

You can simulate it yourself using the `makefile` provided in the `code` directory.

1. Ensure that Make and GHDL are installed.
2. Create a folder called `simulation` inside the directory (if it doesn't exist already).
3. In the commandline, run `make` or `make all` to analyze, and then simulate the testbenches.
4. A `processor.ghw` waveform file will be created in the simulation directory. Note that the earlier file will be overwritten.
5. You can finally run `make clean` to delete any temporary files created in the process.

`processor_tb`

This simply executes the program that has been hardcoded into the program memory.

The two sample programs provided in stage 2, after minor modification, were used for testing.

The modifcation being that `#10` in the first program was changed to `#64`.

5 Results

All the testbenches gave the expected results.

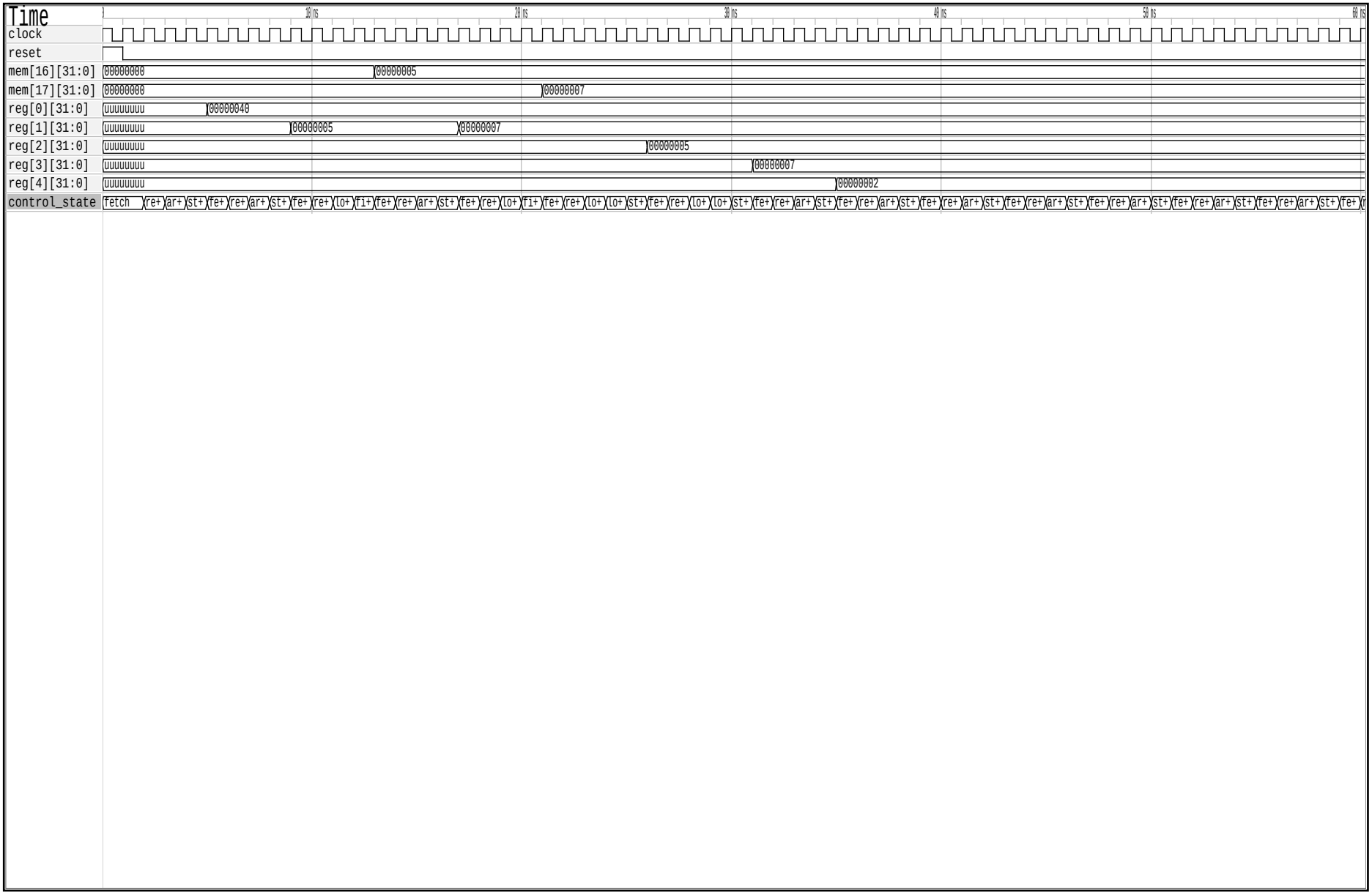
- In the first test, at the end of program, word location 16 and 17 in the memory contain 5 and 7 respectively. `r0`, `r1`, `r2`, `r3`, `r4` contain `0x40`, 7, 5, 7 and 2, respectively.
- In the second test, the program loops 5 times before terminating, ending with `r1` having value 5, and `r0` containing sum of the preceeding values of `r0` that is $0 + 1 + 2 + 3 + 4 = 10$.

The output waveform of the testbench results can be viewed in .ghw and .pdf files in the output folder.

The .pdf files don't contain the entire waveform in some cases as the total number of pages required would be prohibitively large. However, the .ghw files can be opened in any waveform analyzer like GTKWave to view the output waveform.

The pdf files of the simulation are also attached at the end of this report. The netlist haven't been appended due to their larger size.

processor_tb_program_1.pdf



processor_tb_program_2.pdf