

Design Process Journal

Rias Hilliard, Benjamin Hall, Luke Evard, Katarina Chan, Dominic Grazioli



Group 2C

# October 1, 2017

We decided on building a stack-based processor because of its simplicity in instruction and reduced data path. We discussed the instructions and registers we would use in our processor. No specific work was divvied out. It was agreed upon to finish working the next day to finalize everything.

# October 2, 2017

We came to a concrete conclusion on the method of using a stack as a processor. We would use a top-of-stack register as a sort of buffer for the top value and then a pointer to the top of the rest of the stack. This decision was made because it should speed up instructions instead of having to call two pops and a push for most instructions, we simply modify the register. We went through the MIPS green sheet and used instructions that seemed to be universally necessary for our stack as well. Rias was in charge of instructions, and he came up with some stack manipulation instructions, such as toR, fromR, burn, dup, over, and swap. Once we had our instructions decided, which we all relied upon for our own work, we then divvied up the workload so that we could get most of the work done at the same time. Once all of us had finished our assignments, we reviewed each other’s work as team.

## Work Distribution

* Rias will work on descriptions for assembly language programmer and each machine language instruction; estimate 1 day. [1 day]
* Luke will write an explanation of any procedure call conventions; estimate 1 day. [1 day]
* Dominic will write the rules for translating each assembly instruction into machine language and write the machine language translations for Euclid’s algorithm; estimate 1 day. [1 day]
* Ben will finish writing the sample assembly language program for Euclid’s algorithm; estimate 1 day. [1 day]
* Kat will write the descriptions of the instructions, example assembly fragments, and keep track of the Design Journal; estimate 1 day. [1 day]

# October 7, 2017

We were working on cleaning up our design document to fix our instructions to be 16-bit and so adding a few additional instructions. We also worked on our RTL design. Our main method of verification(“test”) for our RTL was just going through every single instruction, making sure it executes as intended (which values are what from the stack, where the result is stores, etc.).

## Work Distribution

* Rias will reorganize the instructions to be grouped more logically. [1 hour]
* Ben will change the Pseudo instructions (beq & bne) in the Machine Code

# October 9, 2017

Rias changed the size of the instructions due to them being too big to fit into the data bus. With a change in instruction length, Rias made new opcodes and changed the documents accordingly. We went through our RTL and saw where things changed and overlapped and created mux’s and other appropriate components to implement the RTL and create as simple a datapath as possible.

## Work Distribution

* Ben will change the machine code instructions (jeq & jne) in the Machine Code; estimated 1 hour [<1 hour]
* Luke write descriptions for each component; estimated 1 [1 hour]
* Rias will run through the RTL and try to have an unbiased viewpoint when checking validity; estimated 2 hours [3 hours]

# October 10, 2017

Rias and Luke went through the design documents and design journal to make sure everything was correct and ready for milestone 2 check with Micah tomorrow. Rias and a third party stepped through the operation to validate its logic. A third party was consulted because all of us had worked together on the RTL and could not provide an unbiased input on its logic.

## Work Distribution

* Rias performing checks and changes; estimated 1 hour [2 hours]
* Rias stepped through the RTL in order to verify proper operation; estimated 1 hour; [<1 hour]
* Luke performing checks and changes; estimated 1 hour [2 hours]

# October 17, 2017

The group worked together to determine new opcode patterns and updating documentation. This is so that our opcodes include more detail as to what the instruction does and ultimately simplifies control logic. Because the opcodes changed, the S-Type and I-Type instruction formats changed and a new J-Type format was introduced. The opcodes are now 4-bits long, thus the S- and I-Types need a function code to specify what to do, while the J-Type can now use a 12-bit immediate as an offset for PC-relative jumps.

# October 18, 2017

Luke and Kat worked on cleaning up the RTL and the rest of the document formatting. We also updated instruction bits now that some I-Type instructions have 12-bit immediates. This is so that we will be able to jump farther now. Rias and Ben worked on the datapath while Luke and Kat continued to modify the design document as the 2 gave us updated information on the implementation.

Rias, Luke and Ben worked on fixing the RTL. We then created a datapath from the RTL. Datapath is affected a lot by the fact that two values from the stack are used almost everywhere, making the logic somewhat confusing. We changed the component specs and signals to match the datapath. We also have now documented our memory addressing and addressing modes. This took roughly 3 hours.

# October 21, 2017

Because Kat and Dominic were not able to attend tonight’s meeting, Rias, Luke, and Ben worked on fixing changes to previous Milestones. We started by having Luke and Ben make changes to the RTL to reflect the multicycle process. From there, they recreated a new and improved datapath. Meanwhile, Rias was making changes to the Design Document. By changes, we mean a complete overhaul of the entire document. We did this renovation for several reasons, such as a more organized document, better formatting, basic milestone changes, and the inclusion of the new RTL and Datapath.

## Work Distribution

* Luke and Ben – RTL changes; estimated 1 hour [1.5 hours]
* Luke and Ben – Datapath Recreation; estimated 3 hours [2 hours]
* Rias – Design Document complete overhaul; estimated 1 day [3 days]
* Luke – Rewriting Components list for M2 and M3; estimated 1 hour [1 hour]

# October 22, 2017

Luke was not able to attend today’s meeting because of an all-day fraternity event. Rias and Ben made some small changes to the RTL and datapath. Rias and Ben split the control unit into three separate units: the Memory Control Unit, the Register Control Unit, and the ALU Control Unit. We made this decision to simplify control logic and organize control signals. Kat began work on the FSM for each control unit’s control signals. Dominic began working on Lab 7.

## Work Distribution

* Rias and Ben – RTL, Datapath, Control Units; estimated 1 hour [1 hour]
* Kat – Multiple FSMs; estimate 1 day [Never finished]
* Dominic – Work on Lab 7; estimate 2 day [2 days]
* Rias – Control signals and Control Unit descriptions; estimated 1 day [1 day]
* Rias – More formatting work on the Design Document; estimated 2 days [2 days]

# October 23, 2017

In class today, we realized how far behind we were. Thus, Rias went back from M1 to the current Milestone noting all the things we need to change or include. Rias started catching us up by described the missing datapath components. Dominic started writing the unit test and Verilog test benches for our components. Luke helped us catch up by working on the integration and implementation plan. For work on M4, Ben is working on creating the Verilog components, Kat is still working on the FSM, and Rias is doing the control specs and updated signals. Ben made a Verilog unit test for the mux and checked Micah’s components to see what we could use for our architecture.

## Work Distribution

* Ben – Mux unit test and check; estimate 30 minutes [30 minutes]
* Ben and Luke – Making the ALU; estimate 1 day [<1 day]
* Luke – Integration plan; estimate 1 day [1 day]
* Kat – Still working on the FSMs; estimate 2 days [Never finished]
* Dominic – unit test descriptions for the control units and the parts; estimate 3 hours [3 hours]
* Dominic – Test-benches; estimate 2 day [Never finished]
* Dominic – Making the other Verilog Components; estimate 1 day [Never finished]
* Rias – Datapath Components; estimate 4 hours [5 hours]
* Rias – Control Unit descriptions; estimate 1 day [6 hours]
* Rias – Previous milestone fixes; estimate 2 days [To this day]

# October 24, 2017

Rias noticed a problem with the RTL and Datapath that required a moderate change in the Datapath and a slight change in the RTL. Rias re-drew the datapath, which introduced two new control signals, Jump and JumpCond, and fixed everything related accordingly. Dominic finished making our muxes. Rias made a shifter component and a test bench for it. Kat made a zero and sign extender and test benches for each respectively. Luke and Ben started working on the ALU for our datapath.

## Work Distribution

* Dominic – Muxes w/ Testbenches; estimated <1 hour [1 hour]
* Kat – Sign and Zero Extender; estimated <1 hour [2 hours]
* Ben and Luke – ALU; estimated 2 hours [<2 hours]
* Rias – Datapath re-draw and eight-bit-shifter; estimated 2 hours [2.5 hours]

# October 26, 2017

Kat and Dominic finished Lab 07 and demonstrated it to Micah. Rias noticed that the combinational components that Kat had built and tested were using a clock edge for some reason and thus fixed them. Rias also noticed that Dominic had made special muxes for three and five inputs for some reason and got rid of those, leaving only the two, four, and eight input muxes.

## Work Distribution

* Rias – Changes to the extenders and deletion of unnecessary muxes; estimated <1 hour

# October 27, 2017

We met with Micah today. Main detraction was that our FSMs were missing. He also mentioned that our RTL could have a few optimizations. With the exception of the Adder, Rias started and finished the entire lab 06 by 4:30. Rias then went and made RTL optimizations, now represented in the RTL\_Update.xlsx file.

## Work Distributions

* Rias – Optimizations and changes to the RTL estimated 1 hour [<2 hours]

# October 28, 2017

Rias made some edits to the existing Verilog parts. Rias made a register file with 256 registers and a new register component. Files changed to reflect naming convention of part[inputs]\_[input size]b\_[output size]b, where parts with the same input and output size will have the convention of part[inputs]\_[size]b. Rias started creating the FSM for the control unit. Rias also began implementing the control unit.

## Work Distribution

* Rias – Verilog edits for naming conventions and small changes; estimated <1 hour [<1 hour]
* Rias – Register file with test bench; estimated <1 hour [30 minutes]
* Rias – Finite State Machine for our single control unit; estimated 1 day [3 days]

# October 30, 2017

Kat went through the previous milestones and checked for inconsistencies and missing things in our design document. She made the list and checked it twice. Luke began Verilog integration and testing.

## Work Distribution

* Kat – design document maintenance and milestone checks; estimated 2 hours [<2 hours]
* Luke – Integration and testing as specified in the design document; estimated 1 day [1 day]

# October 31, 2017

Rias and Luke started burning the midnight oil at 8:17 pm. Rias finally finished the control FSM. Rias made the return stack implemented in memory start at the opposite side of memory from the program, meaning all increments to RP in the RTL became decrements and vice versa. Luke began making control unit tests. At 4:30 am, Rias finished implementing the control unit in Verilog. Luke finished the Control Unit tests at 5 in the morning. Rias added a memory unit at around 6. At 6:45, Rias and Luke integrated the RP, DP, and PC subsystems. Rias finished the rest of the Gen1 Subsystems and Luke has gotten considerably far in the integration tests. It is now 9:20 am. All-nighter success!

## Work Distribution

* Rias – Finish control FSM (written down on a white board); estimate 3 hours [3 hours]
* Luke – Control Unit Tests; estimate 2 hours [~1.5 hours]
* Rias – Control Unit Implementation; estimate 3 hours [~3.5 hours]
* Rias and Luke – Component Integration for complete datapath; estimate 1 hour [1 day]
* Luke – Integration tests; estimate 1 day [~1 day]

# November 1, 2017

Kat marked down the FSM in a portable form using Umlet. Rias and Kat went through and tested the state transition diagram with our datapath to make sure it works. At 4:04 pm, the complete datapath was fully implemented in Verilog. Unfortunately, Rias and Luke were unable to incorporate testing for the entire datapath by the deadline of 4:30 today.

## Work Distribution

* Rias and Luke – complete integration; estimate 1 day [1 day]

# November 2, 2017

Rias called a meeting after our team meeting in order to catch up Ben, Dominic, and Kat. Dominic was unable to make it, replying “Can’t make it tonight for other class but I’ll do whatever as far as work goes.” Kat was feeling out of it and had to leave early. Ben, Rias, and Luke worked on the integration tests. The FinalDatapath synthesizes with a severe amount of warnings that we do not yet know how to fix.

## Work Distribution

* Ben – Integration tests for TRSystem, ALUSystem, Memory; estimated 1 day [<1 day]
* Luke – Integration tests for RP and DP loops; estimated 1 hour [1 hour]
* Kat – Check our control unit signal outputs with the datapath; estimated 1 day [1 day]
* Rias – Tests for register file, 8-bit shifter, and DataStack; estimated 2 hours [<2 hours]
* Rias – Control Unit fixes; estimated 1 day [1 day]

# November 5, 2017

Rias, Ben, and Luke met at 1pm Sunday. Kat and Dominic were both unable to make it due to obligations in other classes. Rias and Ben are still investigating the synthesis warnings. Rias came up with a system test plan. Luke began performing system tests by first manually entering control signals into the datapath to check specific states of our datapath during each cycle of an instruction. Ben made a complete ALU test bench.

## Work Distribution

* Ben – Integration test of ALUSystem; estimated 2 hours [<2 hours]
* Luke and Rias – System tests for simple instructions; estimated 2 hours [2 hours]
* Ben and Rias – System tests follow up; estimated 1 hour [2 hours]

# November 6, 2017

With some changes from the past meeting made to the control unit, the control unit now successfully synthesizes. We intend to get the processor working by the end of today’s meeting. Rias moved some wires and Verilog components, which somehow allows our entire processor to synthesize correctly. Luke and Ben began working on the design and implementation of I/O, which include control changes, additional components, and additional instructions.

For I/O we decided to have an input wire that allows the user to push a 16-bit value to the stack, and a 16-bit output register OUT that continuously outputs the value it holds. To accommodate this, we added two instructions, *in* and *out*. *In* pushes the input value to the stack, and *out* pops the top value from the stack and pushes it to the OUT register.

Ben also worked on updating our Euclid’s Algorithm program for input and output. Dominic made the changes to the machine to reflect the updated algorithm. Rias began implementing the system tests and getting the processor to work, excluding basic input and output. Kat made necessary changes to the control unit FSM and to our datapath. We will need to update these once we fully figure out I/O. Kat also added some logic in control for counting cycles for milestone 6.

## Work Distribution

* Kat – Control unit FSM updates; estimated 1 hour [1 hour]
* Kat – New, redrawn datapath; estimated 2 hours [Not completed at meeting]
* Kat – M6 counters in the control unit; estimated <1 hour [<1 hour]
* Ben – Program changes for allowing I/O; estimated 1 hour [30 minutes]
* Luke and Ben – I/O design; estimated 3 hours [2 hours]
* Rias – System test implementation; estimated 2 hours [4 hours]
* Rias – Proper fixes to the datapath
* Ben – assisting with system tests and fixing
* Dominic – Checking the updated Euclid algorithm [2 hours]
* Ben –I/O Design; estimated 3 hours [3 hours]
* Luke – Begin implementing I/O; estimated 1 day [1 day]

# November 7, 2017

We met today at 8 pm to get everything working that needs to be working. Rias wrote the assembler, compiler, and linker for our processor. Luke implemented I/O into hardware in Verilog with some assistance from Ben. Ben also updated the design document for our I/O design. Kat updated the control unit FSM to reflect our I/O design and changes made previously. Ben fixed our Euclid’s algorithm program to work properly. Dominic performed some logic for counting instructions. Dominic did all the performance data collection/evaluation necessary for M6. Around 1 o’clock our processor and algorithm were fully operational, just like the fully armed and operation battle station! Go team, go!

## Work Distribution

* Rias – Assembler, compiler, linker; estimated 5 hours [3 hours]
* Ben and Luke – I/O design and documentation; estimated 3 hours [2 hours]
* Luke and Ben – I/O hardware implementation; estimated 2 hours [~1.5 hours]
* Kat – FSM and rebuilding of datapath; estimated 2 hours [>4 hours]
* Dominic – Performance evaluation and collection; estimated 1 hour [1 hour]
* Team – Used newly made assembler to test Euclid’s with our processor

# November 13, 2017

The group met at 10:30 pm to finalize our presentation and final report document. Kat insisted on working on the final report, Rias finalized the assembler, and Luke, Ben, and Dominic put the slideshow together.

After hours of work, Kat left to go sleep, and some progress on the final report had been made.

## Work Distribution

* Rias – Assembler, compiler, linker – 3 hours
* Ben, Luke, Dominic – Presentation – estimated 1 hour, actual 3 hours
* Kat – Final report – estimated 2 hours, actual 4 hours
* Rias, Ben, Luke – Final Report – estimated 90 minutes