CP2 Progress Report/Roadmap

Nicholas Logan, Himanshu Bhadoria, Ching-Chia Kuo

4/11/22

Ching-Chia worked on implementation of Hazard Detection, Forwarding, static branch predictors, overall integration of entire design, debugging and testing, and the design for the RISC-V M extension. Nicholas worked on the implementation of Forwarding, debugging and testing, and the designs for the Local Branch History Table, Global 2-level Branch History Table, and Tournament Branch Predictor. Himanshu worked on the implementation of Arbiter, L1 Cache and paper designs for the Eviction Write Buffer, L2+ Cache System, and 4-way Associative Cache.

Functionalities Implemented

The modules we added for this checkpoint include hazard_detection.sv, forwarding.sv, and arbiter.sv. The hazard_detection unit is in charge of executing load, stall, and flush of each intermediate stage register based on different conditions. It takes icache response, dcache read, dcache write, dcache_resp, branch enable, and opcode as deciding factors. In addition, the static-not-taken prediction function is also implemented in hazard_detection. The forwarding unit is in charge of checking if there is a write back to the register file in the execution or writeback stages and if the destination register matches any of the sources of the instruction in the execution stage. If the above condition is met, it will pass the output of the ALU or the data read from memory. The forwarding data will be passed to the cmpmux and alumux. The arbiter is in charge of handling concurrent access to physical memory from both instruction and data caches.

Updated Mux with forwarding data:

alumux1

alumux1_sel	= {alumux1_sel_forwarding, ID_EX_output.control_word.alumux1_sel}
0	alumux1_out = ID_EX_output.data_word.rs1_out;
1	alumux1_out = ID_EX_output.data_word.pc;
2, 3	alumux1_out = EX_MEM_alumux1_forwarding;
5, 6	alumux1_out = MEM_WB_alumux1_forwarding

alumux2

alumux2_sel	= {alumux2_sel_forwarding, alumux2_sel}
0	alumux2_out = ID_EX_output.data_word.imm;
1	alumux2_out = ID_EX_output.data_word.rs2_out;
3	alumux2_out = EX_MEM_alumux2_forwarding
6	alumux2_out = MEM_WB_alumux2_forwarding

cmpmux1 wire to rs1 for CP1

cmpmux1_sel	cmpmux1_sel_forwarding
0, 2	cmpmux2_out = ID_EX_output.data_word.rs1_out;
1	cmpmux2_out = EX_MEM_cmpmux1_forwarding
3	cmpmux2_out = MEM_WB_cmpmux1_forwarding

cmpmux2

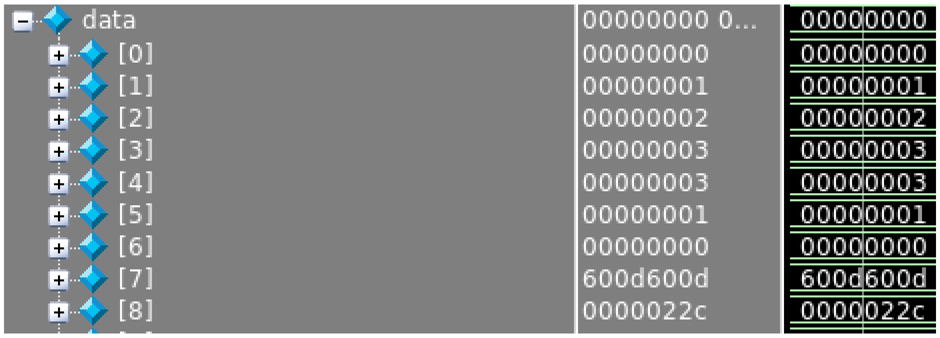
cmpmux2_sel	ID_EX_output.control_word.cmpmux_sel
0	cmpmux2_out = ID_EX_output.data_word.rs2_out;
1	cmpmux2_out = ID_EX_output.data_word.imm;
2, 3	cmpmux2_out = EX_MEM_cmpmux2_forwarding
5, 6	cmpmux2_out = MEM_WB_cmpmux2_forwarding

Testing Strategies

To test for this checkpoint, we first needed to make sure that the model was hooked up to the memory successfully after the integration of the L1 Cache and Arbiter. By using the provided test program, mp4-cp2.s, loading it into the memory, and looking at the waveforms in Modelsim, we first verified that the system was able to successfully pull data from the memory.

Then, our next hurdle was getting the RVFI monitor hooked up to be able to get the autograder to run successfully. Although we never got this to work completely in the end, we're able to get close enough for potential use in future checkpoints. Once we got the commit and halt signals, as well as the other signals that we had in top.sv connected correctly, our final issue was an error that caused a shadow_pc mismatch which we we're unable to fix. After realizing that we don't need autograder to run for full points, we transitioned to test using the provided test code.

Tracing signals that included the PC, rs1, rs2, rd, alumux1_sel, cmpmux_sel, for the IF_ID, ID_EX, and EX_MEM registers while going through mp4-cp2.s helped us verify the correctness of our design. The final register values are as follows.



```
data    00000000 0...   00000000
  [0]   00000000        00000000
  [1]   00000001        00000001
  [2]   00000002        00000002
  [3]   00000003        00000003
  [4]   00000003        00000003
  [5]   00000001        00000001
  [6]   00000000        00000000
  [7]   600d600d        600d600d
  [8]   0000022c        0000022c
```

Furthmore, we also passed the comp1.s, comp2.s, and comp3.s test codes with our design. c

Timing and Energy Analysis

Although we had some timing issues regarding the combinational path being to long for the PC from the instruction cache all the way down our registers, once we simplified our code we were able to get timing to work. The results are as follows. The fast model had a slack of 2.519.

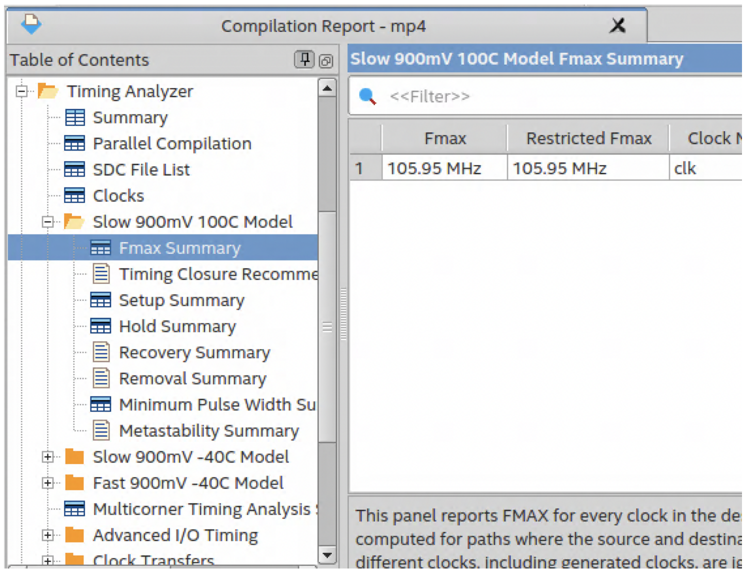

Compilation Report - mp4

Table of Contents

- Timing Analyzer
  - Summary
  - Parallel Compilation
  - SDC File List
  - Clocks
  - Slow 900mV 100C Model
    - Fmax Summary
    - Timing Closure Recomme
    - Setup Summary
    - Hold Summary
    - Recovery Summary
    - Removal Summary
    - Minimum Pulse Width Su
    - Metastability Summary
  - Slow 900mV -40C Model
  - Fast 900mV -40C Model
  - Multicorner Timing Analysis
  - Advanced I/O Timing
  - Clock Transfers

Slow 900mV 100C Model Fmax Summary

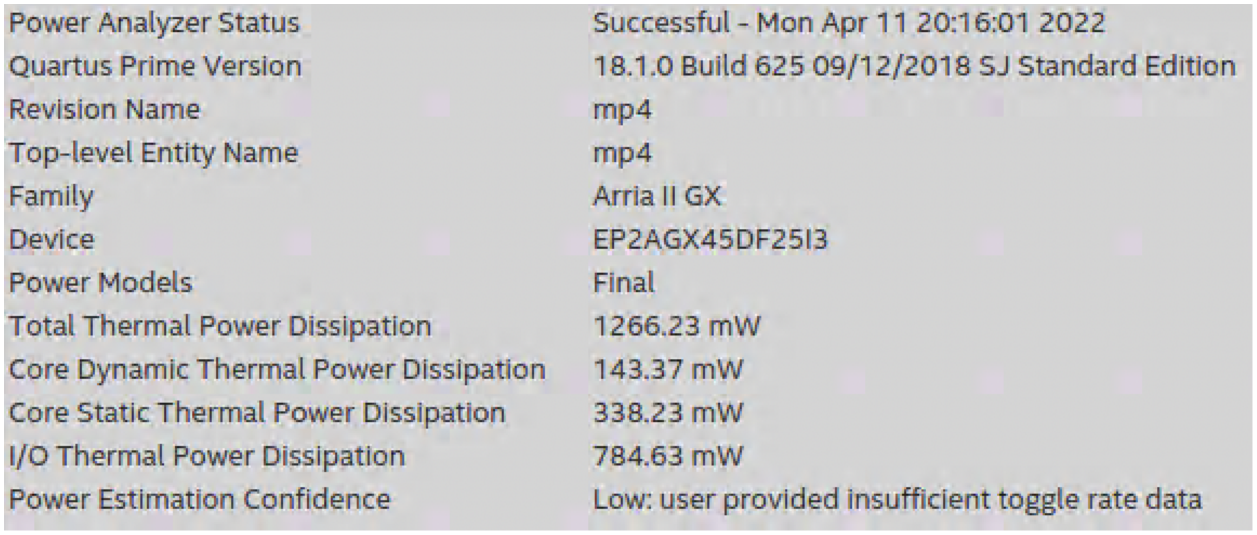
<<Filter>>

	Fmax	Restricted Fmax	Clock N
1	105.95 MHz	105.95 MHz	clk

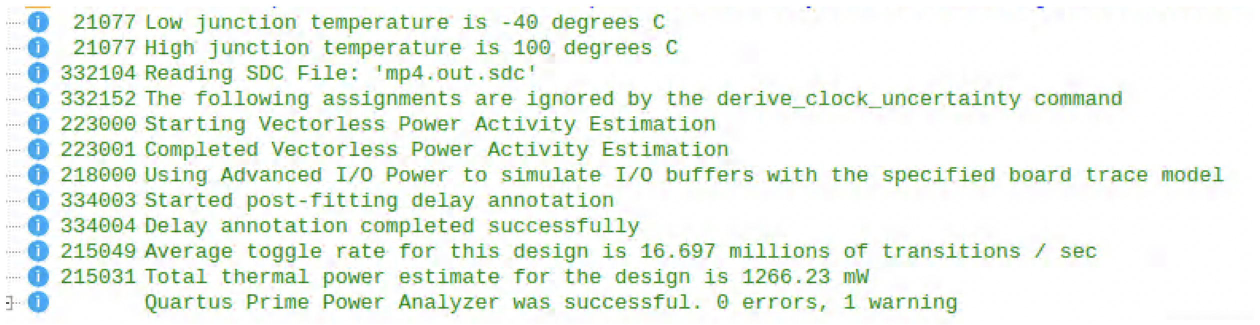
This panel reports FMAX for every clock in the de
computed for paths where the source and destina
different clocks, including generated clocks, are ig

	Fmax	Resticted Fmax
Slow 900mV 100C Model	105.95 MHz	105.95 MHz
Slow 900mV -40C Model	111.67 MHz	111.67 MHz
Fast 900mV -40C Model	n/a	n/a

The following is the result of the energy analysis.



Power Analyzer Status	Successful - Mon Apr 11 20:16:01 2022
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Standard Edition
Revision Name	mp4
Top-level Entity Name	mp4
Family	Arria II GX
Device	EP2AGX45DF25I3
Power Models	Final
Total Thermal Power Dissipation	1266.23 mW
Core Dynamic Thermal Power Dissipation	143.37 mW
Core Static Thermal Power Dissipation	338.23 mW
I/O Thermal Power Dissipation	784.63 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data



```
 21077 Low junction temperature is -40 degrees C
 21077 High junction temperature is 100 degrees C
332104 Reading SDC File: 'mp4.out.sdc'
332152 The following assignments are ignored by the derive_clock_uncertainty command
223000 Starting Vectorless Power Activity Estimation
223001 Completed Vectorless Power Activity Estimation
218000 Using Advanced I/O Power to simulate I/O buffers with the specified board trace model
334003 Started post-fitting delay annotation
334004 Delay annotation completed successfully
215049 Average toggle rate for this design is 16.697 millions of transitions / sec
215031 Total thermal power estimate for the design is 1266.23 mW
       Quartus Prime Power Analyzer was successful. 0 errors, 1 warning
```

Roadmap

We will work on completing the advanced features. Himanshu will work on the implementation of the L2+ cache system[2] and 4-way associative cache[2]. Nicholas will work on the local branch history table[2], global 2-level branch history table[3], and tournament branch predictor[5]. Ching-Chia will implement the RISC-V M extension[3], and the Eviction write buffer[4]. We will all work together to verify the code and help out where necessary.