**Fall 2024 ECE 552 Phase 3 Descriptions**

**Group: Jingyi Yao, Yukun Qin, Weichen Zhang**

**Overview**

This project focuses on the design and implementation of a processor system based on the WISC-F24 instruction set architecture. Throughout the three phases of the project, the processor was progressively enhanced, culminating in Phase 3, where a cache hierarchy was implemented to improve performance and efficiency.

In Phase 3, the work involved designing and implementing separate instruction (I-cache) and data (D-cache) caches, as well as cache controllers to manage interactions between the processor pipeline, the caches, and the main memory. The caches were configured as 2-way set associative with a total capacity of 2048 bytes and 16-byte blocks. Write-through and write-allocate policies were used for handling data read and write operations. The system also included enhancements such as interfaces between the caches and specific pipeline stages, with the I-cache interacting with the instruction fetch stage and the D-cache interfacing with the memory access stage.

A state machine was developed to handle cache misses, ensuring that data blocks could be efficiently retrieved from memory while minimizing pipeline stalls. The successful integration of the cache hierarchy not only optimized memory access latency but also ensured smooth operation of the processor under various scenarios.

**Submission ZIP Structure**

**Phase3**

|--- project-phase3

| |---cup.v

| |\_\_\_|---pc\_reg.v

| |\_\_\_|---pipelined\_cache\_control.v

| | | |---IcacheIndex

| | | |---DcacheIndex

| | | |---IcacheOffset

| | | |---DcacheOffset

| | | |---Instr\_cache

| | | |---Data\_cache

| | | |---Icache\_fsm

| | | |---Dcache\_fsm

| | | |---memory

| |\_\_\_|---if\_id

| |\_\_\_|---id\_ex

| |\_\_\_|---ex\_mem

| |\_\_\_|---mem\_wb

| |\_\_\_|---ALU

| | | |---add

| | | |---sub

| | | |---psa

| | | |---shift

| |\_\_\_|---Flag\_Register

| |\_\_\_|---pc\_control

| |\_\_\_|---Forwarding\_unit

| |\_\_\_|---Hazard\_Detection\_Unit

**Task Breakdown**

Phase1: Jingyi Yao 33.3%, Yukun Qin 33.3%, Weichen Zhang 33.3%

Phase2: Jingyi Yao 60%, Yukun Qin 20%, Weichen Zhang 20%

Phase3: Jingyi Yao 60%, Yukun Qin 20%, Weichen Zhang 20%

**Special Features**

While implement the design we found that when cache miss, instruction updating will faster than PC value updating 1 clock period. At first we think that it is because of logic problem in the fsm. But every times we modify the logic of fsm, the design will crash. So we turn to the other part of the module, finally we found that the problem is the enable signal of counter in the fsm.

**Completeness**

The design nearlly meets all the requirements specified for Phase 1, Phase 2, and Phase 3 of the project. In Phase 1, the single-cycle processor was fully implemented to support the WISC-F24 instruction set architecture, including accurate handling of all 16 instructions, proper register file operations, and flag register updates. All test programs provided for Phase 1 were executed correctly, with results matching hand-calculated expectations. We didn’t implement 3-1mux shifter. But we achieve that in phase2 and 3.

In Phase 2, the processor was extended to a 5-stage pipelined design with complete hazard detection and mitigation mechanisms. Data forwarding paths (EX-to-EX, MEM-to-EX, and MEM-to-MEM) were implemented to handle data hazards, while control hazards were addressed using a predict-not-taken strategy. Pipeline stalls and flushes were accurately managed to ensure correctness. Test programs for Phase 2 validated these functionalities, and all outputs aligned with expected results.

In Phase 3, the design incorporated a hierarchical cache system with 2-way set-associative instruction and data caches. The cache controllers were implemented with write-through and write-allocate policies. The design handles cache misses using an FSM that coordinates block transfers from main memory, ensuring seamless operation without pipeline disruptions. Testing included both individual component validation and full system integration. The processor trace outputs for all Phase 3 test programs matched the expected results, and cycle counts were consistent with design specifications.

**Testing**

We pass test1-4 in the top level of project of phase3, and we also test adder with addr\_16bit\_tb. v . It includes overflow testing in the extreme situation which is the maximum value. Every time we finish the parts of ALU such as psa, sub and shift, we write a testbench to make sure it works well.

**Results**

We successfully pass all the test of phase3

|  |  |  |
| --- | --- | --- |
| **name** | **outcome** | **cycle** |
| **Test1** | **R1:0x0051**  **R2:0xa0b0**  **R3:0xf201**  **R4:0x7fff**  **R6:0x8dfe**  **R7:0x0000**  **R8:0xe37f**  **R9:0xdff8** | **41** |
| **Test2** | **R1:0x5151**  **R2:0x00b0**  **R3:0x0004**  **R4:0x5151**  **R50x00a4**  **memory[0x00b4]=0x5151** | **43** |
| **Test3** | **R1:0x0000**  **R2:0x0001**  **R3:040005**  **R50x0010** | **51** |
| **Test4** | **R1:0xaaaa**  **R2:0x0010**  **R3:0x0002**  **R4:0x0002**  **R5:0x0090**  **R6:0x0080**  **memory[0x0070] = 0x0010**  **memory[0x0072] = 0x000e**  **memory[0x0074] = 0x000c**  **memory[0x0076] = 0x000a**  **memory[0x0078] = 0x0008**  **memory[0x007a] = 0x0006**  **memory[0x007c] = 0x0004**  **memory[0x007e] = 0x0002**  **memory[0x0080] = 0x0010**  **memory[0x0082] = 0x000e**  **memory[0x0084] = 0x000c**  **memory[0x0086] = 0x000a**  **memory[0x0088] = 0x0008**  **memory[0x008a] = 0x0006**  **memory[0x008c] = 0x0004**  **memory[0x8e] = 0x0002** | **279** |

**Waveforms:**

**Test\_1**

图形用户界面, 文本

描述已自动生成

**Test\_2**

图形用户界面

中度可信度描述已自动生成

**Test\_3**

图形用户界面

中度可信度描述已自动生成

**Test\_4**

电脑萤幕画面

描述已自动生成

**The testing thoroughly covered all Phase 3 requirements, including cache operations (hit/miss handling, write-through, and write-allocate policies), integration with the pipeline, and memory arbitration for simultaneous cache requests. Cache miss handling was validated with the FSM, ensuring correct data retrieval and stall management. All test cases, including edge scenarios, demonstrated functionality and met performance expectations, confirming the design's correctness and reliability.**