ECE 571: Winter 2021

Group 8: MIPS Pipelined CPU

Vishal Parikh
Naga Varden
Hiranmaye
En-Ting Yang

We are planning to implement a 5-staged MIPS pipeline processor. The source material for the design part is from a book called Digital design and computer architecture (2nd edition). In the book, there is code for a single cycle but not for the pipeline CPU, we are going to convert the single-cycle processor to a 5 staged pipeline processor.

The book also provides block diagrams for a single cycle, multicycle, and pipelined MIPS processor. The processor consists of Control Unit, Datapath, Register file, Instruction memory, Data memory, ALU, and a few MUXs as shown in the diagram below.

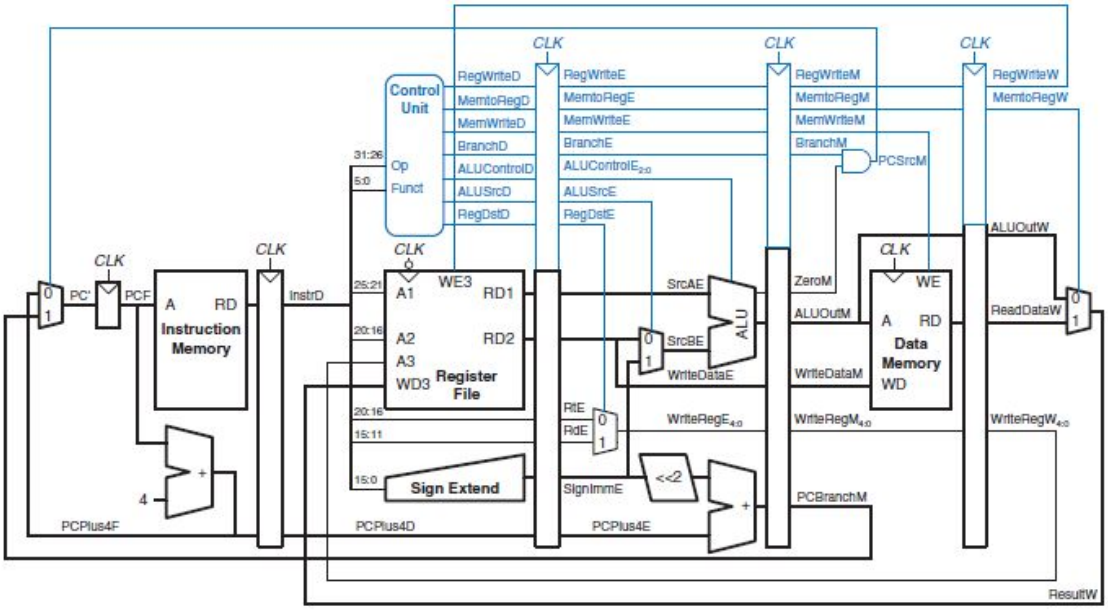

Figure 7.47 Pipelined processor with control

(tentative) Work Distribution and Timeline:

Due date	Vishal Parikh	Naga Varden	Hiranmaye	En-Ting Yang
Feb 14th	-Setup Github And enviornment	Study MIPS CPU from available resources	Study MIPS CPU from available resources	Study MIPS CPU from available resources
Feb 21st	Work on the design part of the CPU (will assign a module to each individual later)	Work on the design part of the CPU (will assign the module to each individual later)	Work on the design part of the CPU (will assign the module to each individual later)	Work on the design part of the CPU (will assign the module to each individual later)
Feb 28th	Work on the design/verification part of the CPU	Work on the design/verification part of the CPU	Work on the design/verification part of the CPU	Work on the design/verification part of the CPU
Mar 7th	Work on the Verification	Work on the Verification	Work on the Verification	Work on the Verification
Final week	The report, presentation slides	The report, presentation slides	The report, presentation slides	The report, presentation slides

Verification approach

- We are planning to approach the verification by using the testbench method.
- We will write a program in assembly or C to then get a memory image of the assembly using MARS simulator's (http://courses.missouristate.edu/KenVollmar/MARS/index.htm) mem dump function.
- We will then provide the memory image as input to the processor in the 'memfile.dat' file and will read that into the RAM (instruction cache) and will

check the behavior of the processor and compare the output to the MARS simulator's output.

- After we verify our design works properly we then will introduce some intentional errors or bugs in the design to check the behavior of the CPU.
- We will also try to apply some fake/buggy instruction in the mem file to then check the execution.