Milestone1

Meetings:

1. Sunday, January 8, 2017
2. Tuesday, January 10, 2017

We decided to build a stack-based processor that can successfully fit the basic requirement. Initially, we decided to have an 5-bit opcode, so we will have 3 unused bits. In terms of how to deal with the 3 bits, we may use them later on, but we have not decided yet. The MSB of our opcode determines if the operation deals with addresses. For example, if the MSB is 1, it means the operation is the D-type.
We also want to use the cache so that we can deal with the 16-bits data transfer.

Work for the week:
Instruction format documentation, estimated 1 day (Minzhe Luo, Peicheng Tang)
Translation to the Machine code, estimated 2 days (Wenkang Dang, Minzhe Luo)
Write Euclid's algo in our language, estimated 2 days (Weize Sun, Peicheng Tang)

Milestone2

Meetings:

1. Sunday, January 15, 2017
2. Tuesday, January 17, 2017

We continued to complete the RTL description of each instruction. We also discussed if we need to use 8-bit instructions instead of 16-bit instructions. It is believed that 8-bit instructions may cause lots of muxes in further design, thus it would cause longer delay time. Considering the difficulty and the low efficiency of the 8-bit instructions, we decided to be concentrated on the 16-bit instructions and improve the cache design in the following days. We decided to use dual channel datapath on stack so that we could access the top two stacks at the same time.

Work for the week:
RTL description of the instuctions, estimated 1 day (Minzhe Luo, Peicheng Tang, Weize Sun,Wenkang Dang)
Testing of the RTL, estimated 1 day (Wenkang Dang, Minzhe Luo, Peicheng Tang)
List of all the componet needed, estimated 1 day (Weize Sun)

Milestone3

Meeting:

1, Sunday, January 22, 2017
2. Tuesday, January 24, 2017

We change the operation of the jump instruction. When the data of top row of the stack is equal the data in the second row of the stack, PC will be the value of the third row of the stack. As the result, we do not need JPCP and JPCN, the two instructions any more, since we can access the 16-bit address.
We also list out all the control signals we need to implement each instruction. It also help a lot when we try to implement the iteration test and component unit test.

Work for the week:
graph of the datapath, estimated 1 day (Minzhe Luo, Peicheng Tang, Weize Sun)
Description of the component and how to build in Xilinx, estimated 1 day (Weize Sun, MInzhe Luo)
Control signal needed, estimated 1 day(Peicheng Tang, Weize Sun)
Unit test of each component, estimated 1 days (MInzhe Luo, Weize Sun)
integration test and plan, estimated 1 days(Minzhe Luo, Weize Sun)
implementation of components in hardware, estimated 1 day(Wenkang Dang)

Milestone 4
Meetings:

1. Sunday, January 22, 2017
2. Tuesday, January 24, 2017

We decided to make the instructions multicycle and we change the RTL of the instructions. Since we have finish all the control signals we need to operate the instructions, this week we spent more time on implement the components in the hardware Xilinx.
We draw the control signal state diagram and the transition table. In order to test if the control unit work functionally, we also prepared to build a ControlunitTester to test it.
Since our memory unit and cache can both be implemented by the example distributed memory available in the course resources page. We decided to simply use that implementation for our memory and cache

Work for the week:
Implementation of ALU & Implementation of PC and its testbench, estimated 1 day (Wenkang Dang)
ALU Testbench & Implementation of Stack and its testbench, estimated 1 days(Peicheng Tang)
Subcomponents for stack, estimated 1 day(Weize Sun 80%, Peicheng Tang 20%)
DataPath/ control signal/ RTL Revise,estimated 1 day (Minzhe Luo&Weize Sun)
Control signals state and transition table, estimated 1 day(MInzhe Luo)
Control signals for each cycle of the instrction, estimated 1 day(Minzhe Luo&Weize Sun)
PC, ALU unit, estimate 1 day (Wengkang Dang)
Note: No work for memory and cache, since we can use the example given.

Milestone 5

Meetings:

1. Sunday, February 5, 2017
2. Tuesday, February 7, 2017

We change the RTL and simplify the multicycle RTL. We also change the control signal and redraw the control state bubbles to make the control unit simpler. After the change, we have 16 control stages and 18 instructions. We also plan the system test in the design document and we modify the integration test in the M3.
We also change and simplified the datapath slightly. As for the hardware, we complete all the components in verilog language. We finish the logic of the whole processor and we also set the test of it.
Although the processor does not pass the system test, we will debug and try to fix it in the next week.

Work for the week:
implement the new PC component, estimated 1 day (Wenkang Dang)
improve and simplify the RTL, estimated 1 day(Weize Sun & Peicheng Tang)
assmble and arrange the processor, estimated 1 day(Weize Sun & Peicheng Tang
system test, implement integration test, improve the control diagram and transition table, estimated 2 days. (Minzhe Luo)

Milestone 6

Meetings:

1. Sunday, February 12, 2017
2. Tuesday, February 14, 2017
3. Wednesday, February 15, 2017

We debug and spend lot of time to test and make the processor to be functional.
We also change and improve the design document according to the improvement and change we made in the hardware design. We clean up the design document to make it consistent. Multiple changes were made to the algorithm. New control signal was added to PC,ALU during debugging. We found ALU operation SUB was inversed, the jmp/jpg/jpe address must be desired address -1 as the instruction will only be load at the beginning with the new PC

Work for the week:
proofread and improve the design document, estimated 1 day(Minzhe Luo)
change and improve the control unit, estimated 1 day(Minzhe Luo)
debug and test the processor, estimated 2 days(Weize Sun & Peicheng Tang)
Implementation in FPGA board, estimated 1 day(Wenkang Dang)