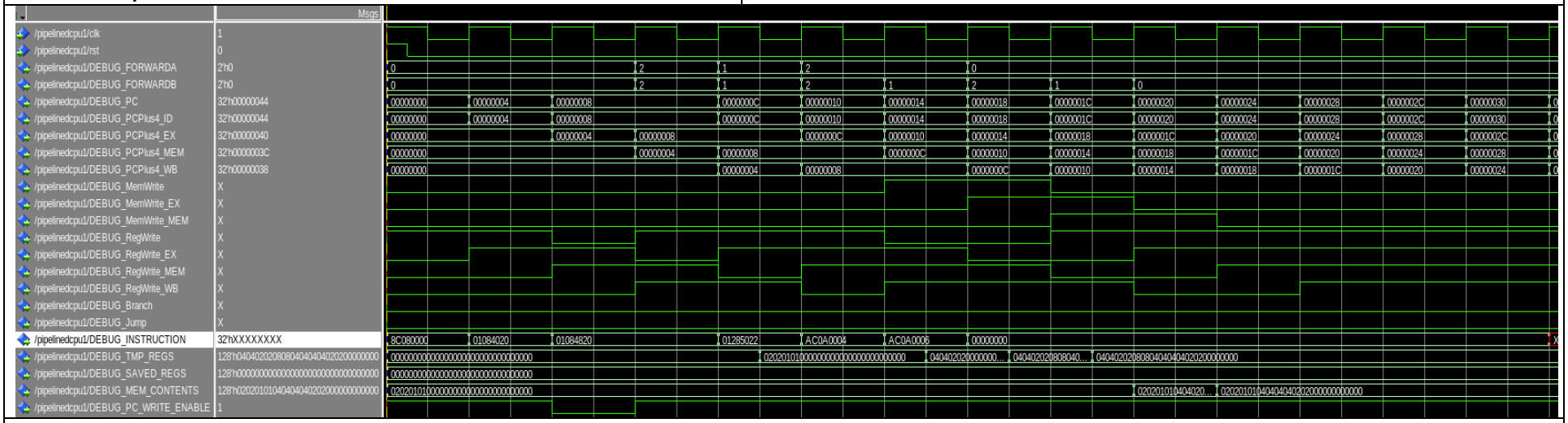
LAB5 Report

Qinlong Cui 1290554



Incomplete data in cycle7&8: 128'h04040202000000000000000000000000
Incomplete data in cycle8&9: 128'h04040202080804040000000000000000
Incomplete data in cycle10: 128'h02020101040402020000000000000000

1	Everything got reset. The 1st instruction**, lw $t0 0x0**, get fetched.
2	The 2nd instruction, **add $t0 $t0 $t0**, get fetched. The 1st instruction enters ID stage.
3	The 3rd instruction, **add $t1 $t0 $t0**, get fetched. The 1st instruction goes to the EXE stage. The 2nd instruction goes to ID stage. Because the data needed is still not prepared by the 1st instruction. The PC_WRITE_ENABLE goes 0.
4	It stalls. The 1st instruction goes to MEM stage. The 2nd instruction remain in the ID stage. The 3rd instruction remain in the IF stage.
5	The 4th instruction, **sub $t2 $t1 $t0**, get fetched. The 1st instruction goes to the WB stage. The value of $t0 changes to **0x02020101**. The data is forwarded. The 2nd instruction goes to EXE stage. The 3rd instruction goes ID stage.
6	The 5th instruction, **sw $t2 0x4**, get fetched. The 2nd instruction goes to MEM stage. The data is forwarded. The 3rd instruction goes to EXE stage.

	The 4th instruction goes to ID stage.
7	The 6th instruction**, sw $t2 0x6**, get fetched. The 2nd instruction goes to WB stage. The latest value of $t0, **0x04040202** is written back. The 3rd instruction goes to MEM stage. The 4th instruction goes to EXE stage. Because it needs the latest $t0 and $t1, but $t1 is still in MEM stage. So, the data is forwarded. The 5th instruction goes to ID stage.
8	Nop is fetched. The 3rd instruction goes to WB stage. Write $t1 to **0x08080404**. The 4th instruction goes to MEM stage. The 5th instruction goes to EXE stage. Needs the latest data of $t2 and it got data by forwarding. The forwarded data is useless because the target address is certain. The 6th instruction goes to ID stage.
9	Nop is fetched. The 4th instruction goes to WB stage. Writes $t2 to **0x04040202** The 5th instruction goes to MEM stage. Writes the DMEN(4) to **0x04040202**. The 6th instruction goes to EXE stage. Needs the latest data of $t2, so the data is forwarded. But the data is useless because the target address is certain.
10	Nop is fetched. The 5th instruction goes to WB stage. The 6th instruction goes to MEM stage. Writes the DMEM(4) to 0x04040404 and DMEM(8) to 0x02020000.
11	Nop is fetched. The 6th instruction goes to WB stage.

If we add a Nop instruction after the 1st instruction. It could be regard as we manually stall the pipeline in this condition. When the 2nd instruction goes to the EXE stage, the data has already WB to $t0. The number of cycles, value of PC, states of DMEM and Register at the end are the same as the waveform above.