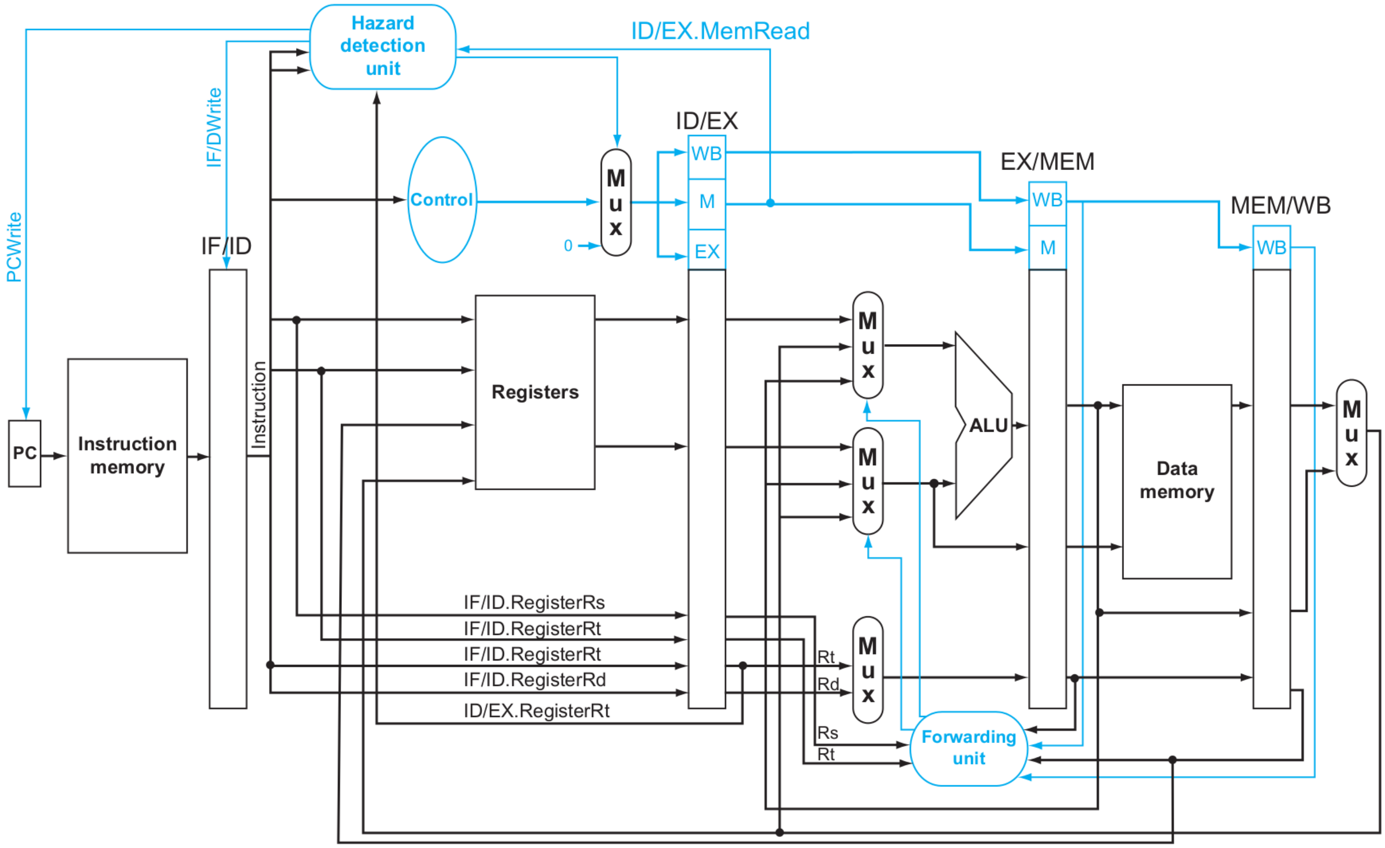
MIPS 5-stage Pipelined processor design



- The diagram given above shows the design of a MIPS 5-stage pipelined processor with all the necessary modules. This diagram was used as a reference for coding the pipelined processor.
- The various instructions that are implemented as a part of this design are R-type instructions (add,sub,and,or,nor,slt), lw,sw,andi,addi,ori,slti and beq.

Brief description of the modules coded in Verilog

- Pipelined_mainprocessor.v is the main file which consists of a compilation of all the modules present in different files and models the pipelined processor. Care has been taken that the module name matches with the file name for easy readability and relatability of functions of different modules.
- The various mux files are used to select from various inputs and direct it to the output according to the control signal present.

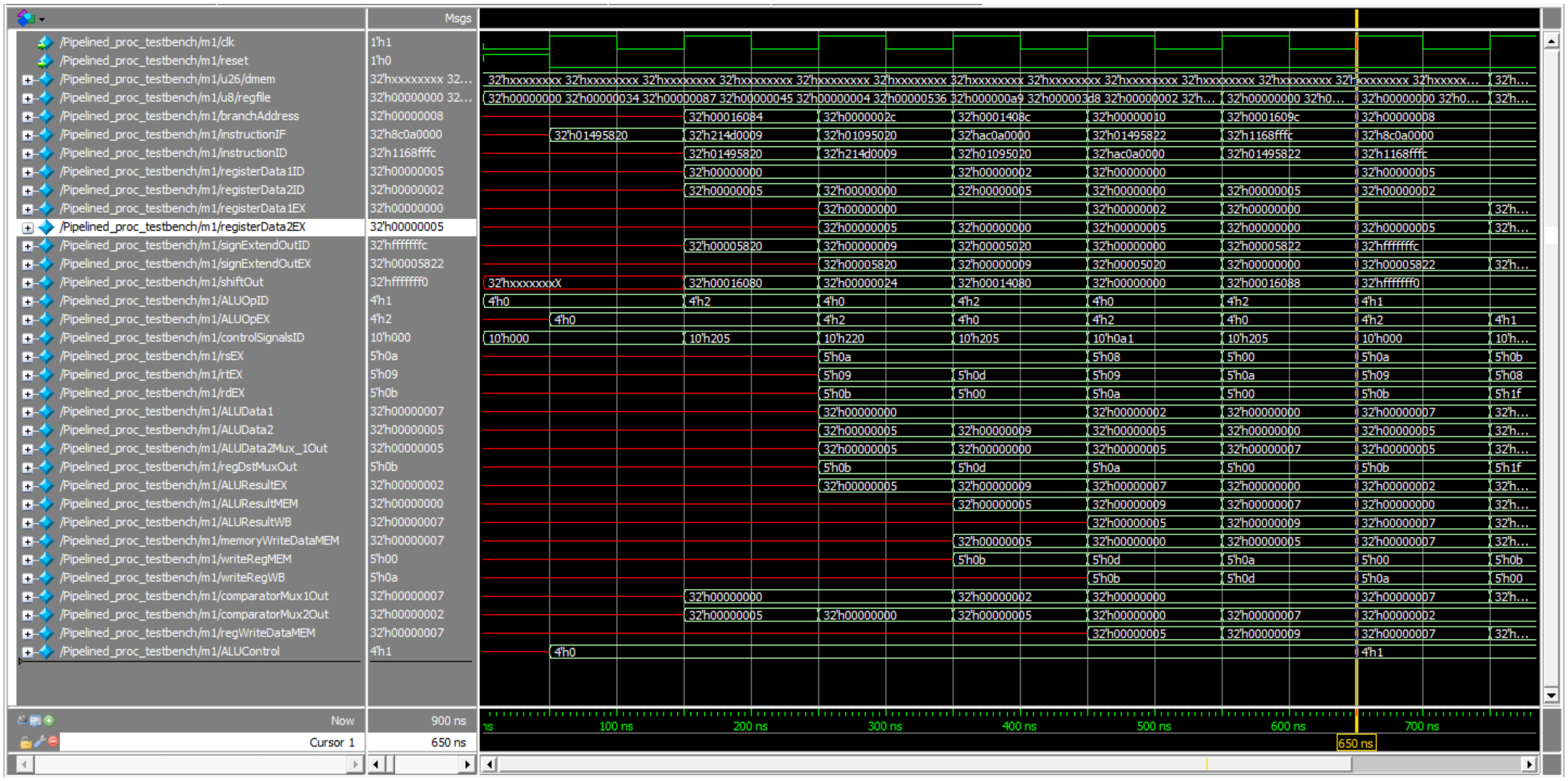
- The control signals required for the working of various modules and instructions are generated in MainControl.v and ALUControl.v files.
- The various memories present in the processor (Instruction Memory, Data Memory and Register File) are modelled in files Instruction_Memory.v, DataMemory.v and Register_file.v.
- Data hazards are resolved in the processor by using the Forwarding_Unit.v and HazardDetection_Unit.v files. Control hazards are resolved by introducing stalls.
- The various registers present between different stages in the pipeline are modelled as such in Verilog.
- The Programcounter.v is used for updating the next value of PC.
- The signextend module is used for sign extending 16-bit input to a 32-bit input.
- The shifter is used for shifting input by 2 bits for calculation of branch address later.

Simulation

- ‘code.txt’ is a file used for initializing the instruction memory and this file can be modified to add more instructions (those mentioned in first page) to simulate the working of the processor.
- The instructions implemented for simulation purposes are:

```
add $t4, $t2, $t1

addi $t5, $t2, 0x09

add $t2, $t0, $t1

sw $t2, 0($zero)

sub $t3, $t2, $t1

beq $t3, $t0, -4

lw $t2, 0($zero)
```

- Some of the instructions are chosen to demonstrate the working of the processor in situations of hazard. There is control hazard in instruction 6 (beq) and data hazard between instructions 4 and 5.
- While simulating, it is to be made sure that the ‘code.txt’ file is present in the simulation directory or project.



	Msgs
/Pipelined_proc_testbench/m1/clk	1'h1
/Pipelined_proc_testbench/m1/reset	1'h0
/Pipelined_proc_testbench/m1/u26/dmem	32'hxxxxxxxx 32...
/Pipelined_proc_testbench/m1/u8/regfile	32'h00000000 32...
/Pipelined_proc_testbench/m1/branchAddress	32'h00000008
/Pipelined_proc_testbench/m1/instructionIF	32'h8c0a0000
/Pipelined_proc_testbench/m1/instructionID	32'h1168fffc
/Pipelined_proc_testbench/m1/registerData1ID	32'h00000005
/Pipelined_proc_testbench/m1/registerData2ID	32'h00000002
/Pipelined_proc_testbench/m1/registerData1EX	32'h00000000
/Pipelined_proc_testbench/m1/registerData2EX	32'h00000005
/Pipelined_proc_testbench/m1/signExtendOutID	32'hfffffffc
/Pipelined_proc_testbench/m1/signExtendOutEX	32'h00005822
/Pipelined_proc_testbench/m1/shiftOut	32'hfffffff0
/Pipelined_proc_testbench/m1/ALUOpID	4'h1
/Pipelined_proc_testbench/m1/ALUOpEX	4'h2
/Pipelined_proc_testbench/m1/controlSignalsID	10'h000
/Pipelined_proc_testbench/m1/rsEX	5'h0a
/Pipelined_proc_testbench/m1/rtEX	5'h09
/Pipelined_proc_testbench/m1/rdEX	5'h0b
/Pipelined_proc_testbench/m1/ALUData1	32'h00000007
/Pipelined_proc_testbench/m1/ALUData2	32'h00000005
/Pipelined_proc_testbench/m1/ALUData2Mux_1Out	32'h00000005
/Pipelined_proc_testbench/m1/regDstMuxOut	5'h0b
/Pipelined_proc_testbench/m1/ALUResultEX	32'h00000002
/Pipelined_proc_testbench/m1/ALUResultMEM	32'h00000000
/Pipelined_proc_testbench/m1/ALUResultWB	32'h00000007
/Pipelined_proc_testbench/m1/memoryWriteDataMEM	32'h00000007
/Pipelined_proc_testbench/m1/writeRegMEM	5'h00
/Pipelined_proc_testbench/m1/writeRegWB	5'h0a
/Pipelined_proc_testbench/m1/comparatorMux1Out	32'h00000007
/Pipelined_proc_testbench/m1/comparatorMux2Out	32'h00000002
/Pipelined_proc_testbench/m1/regWriteDataMEM	32'h00000007
/Pipelined_proc_testbench/m1/ALUControl	4'h1
Now	900 ns
Cursor 1	650 ns

- The above diagram shows the simulation of the processor carried out in Modelsim. dmem and regfile can be monitored for verifying the correctness of various instructions. The signals shown in the picture are the main signals that can be used for checking the working of the processor.