DEPARTMENT OF ELECTRICAL ENGINEERING
IIT BOMBAY

Processor Design-EE739

A REPORT ON

5 Stage Pipelined RISC Processor

SUBMITTED BY

Gaurav Tilwankar (203070051)
Harsh Paryani (203070052)
Tanuj Kumar (203070059)
Deepak Chand (203070060)

UNDER THE GUIDANCE OF

Prof. Virendra Singh

Design Decisions

1. R7 and program counter are different register
2. there is one cycle delay between R7 and PC write. only PC can be written in R 7
3. any instruction that can write r7 can also write in PC. first written in PC then in R7
4. branch prediction done for BEQ and JAL. not done for JLR and JRI because the next branch target address value is dependent on a a particular register in next register

If in between value of Register changes then the branch predictor fails some mechanism has to be design for that which is not put in this design

1. For ADC, ADZ, NDC and NDZ instruction, the data forwarding has to be taken such that the carry or Zero Flag is one then by some Gate combination forwarding is being done in this design
2. for branch prediction the evolution of the entry from lookup table if lookup table is full then 0th entry is only removed every time. any type of LRU or NMRU is not used.
3. If the data is to be used is not yet generated which is in execution stage and other instruction is in decode stage then we have inserted an halt
4. The carry and Zero Flag the given to the controller has to be data forwarded same as RFD1 and RFd2

Instruction Memory:

Test1: Fibonacci Program. R2 has the result

B”0000_001_001_000001”--------ADI R1, R1,000001 (16’h0241)

B”0000_110_110_001111”----------ADI R6, R6 ,001111(16’h0D8F)

B”0000_101_101_000011”----------ADI R5 R5 #000011 (16’h0B43)

B”1000_011_110_000110”----------BEQ R3 R6 #000110 (16’h8786)

B”0001_001_000_010_000”---------ADD R2 R1 R0   (16’h1210)

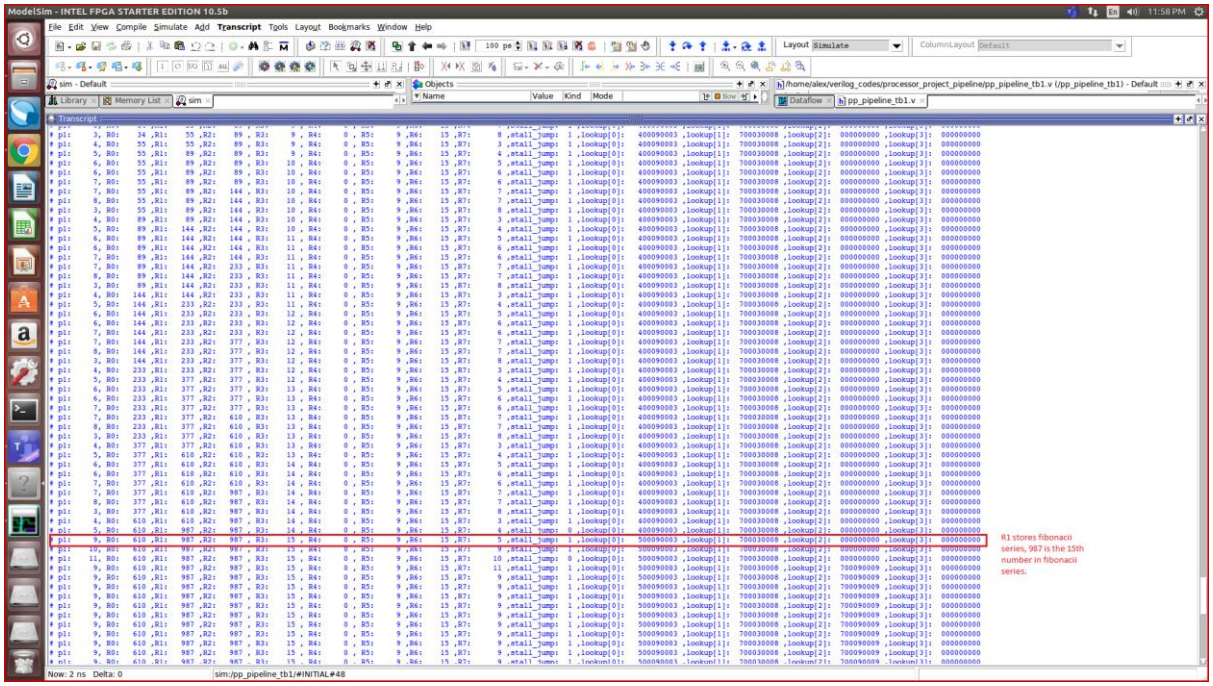
B”0000_100_001100000”----------SM R4 #00000110 (16’hD860)

B”1100_100_011000000”----------LM R4 #00000011 (16’hC8C0)

B”0000_011_011_000001”----------ADI R3 R3 #000001 (16’h06C1)

B”1001_101_111111011”----------JAL R5 11111011 (16’h9BFB)

B”1001_111000000000”----------NOP (16’h9E00)



TEST2: Sum of first n natural numbers

B”0000_000_001_001111”---------ADI R1 R0 #001111(16’h004F) Enter the value of n

B”0000_011_011_000010”---------ADI R3 R3 #000010(16’h06C2)

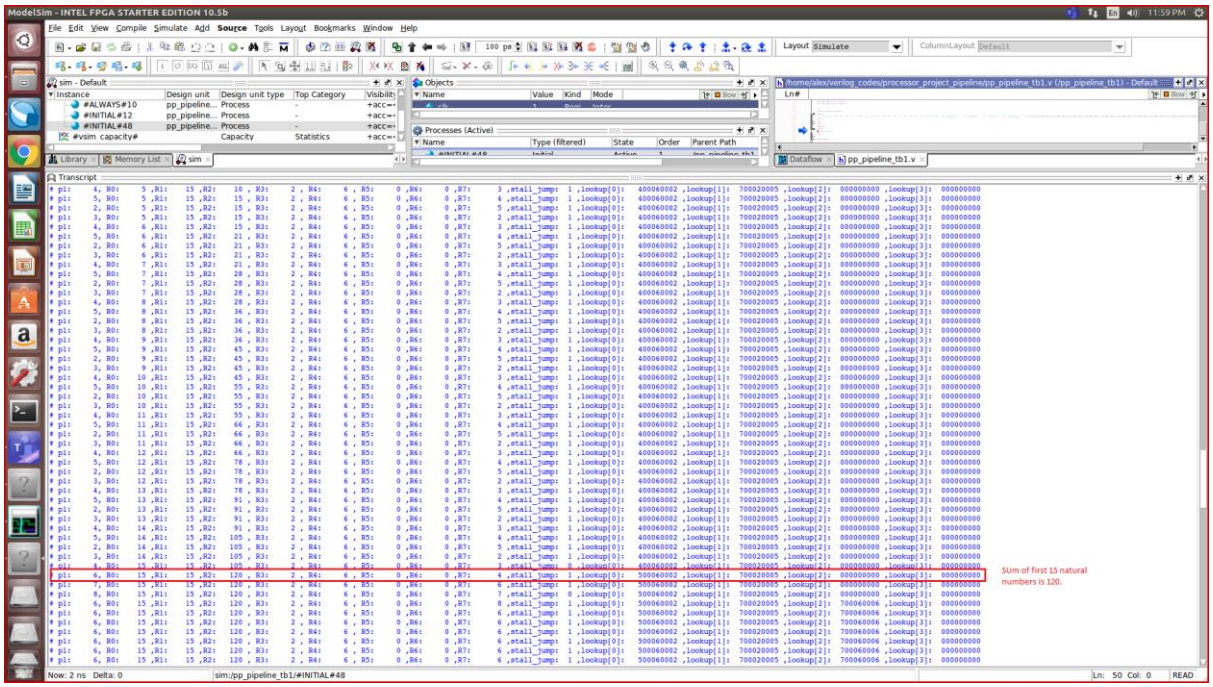
B”1000_000_001_000100”---------BEQ R0 R1 #000100(16’h8044)

B”0000_000_000_000001”---------ADI R0 R0 #000001(16’h0001)

B”0001_010_000_010_000”-------ADD R2 R2 R0(16’h1410)

B”1001_100_111111101”----------JAL R4 ,-3(16’h99FD)

B”1001_111_000000000”----------JAL R7,0(16’h9E00)



Test3: Cube of x number

B”0000_001_001_000101”---------ADI R1,R1,000101 (16’h0245)

B”0000_011_011_000010”---------ADI R3, R3, 000010(16’h06C2)

B”1000_001_000_000100”---------BEQ R1, R0,4(6’h8204)

B”0001_010_001_010_000”-------ADD R2, R2,R1(16’h1450)

B”0000_000_000_000001”---------ADI R0, R0, 000001(16’h0001)

B”1001_100_111111101”-----------JAL R4, -3(16’h99FD)

B”0000_011_011_000101”---------ADI R3, R3, 000101(16’h06C5)

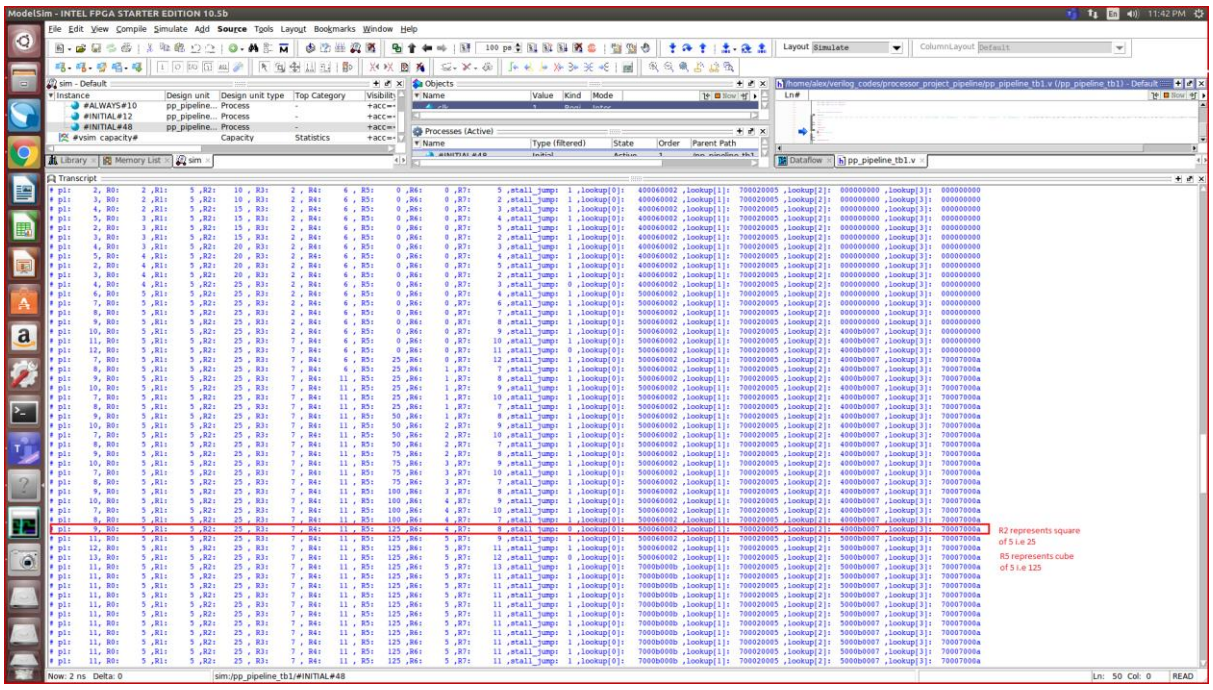
B”1000_001_110_000100”----------BEQ R1, R6, 000100(16’h8384)

B”0001_010_101_101000”----------ADD R5, R2, R5(16’h1568)

B”0000_110_110_000001”----------ADI R6, R6, 000001(16’h0D81)

B”1001_100_111111101”-----------JAL R4, -3(16’h99FD)

B”1001_111_000000000”-----------JAL R7, 0(16’h9E00)



Test4: Multiplication c=a*b. R1 stores a. R5 stores b. Result c is in R2

B”0000_000_000_000000”----------ADI R0 R0 #000000(16’h0000)

B”0000_001_001_000101”----------ADI R1 R1 #000101;(16’h0245) Enter binary equivalent of a

B”0000_101_101_000110”----------ADI R5 R5 #000110;(16’h0B46) Enter binary equivalent of b

/////B””ADI R3 R3 #000100(16)

B”1000_001_000_000100”----------BEQ R1 R0 #000100(16’h8204)

B”0001_010_101_010_000”---------ADD R2 R2 R5 ;(16’h1550) Result

B”0000_000_000_000001”----------ADI R0 R0 #000001(16’h0001)

B”1001_100_111111101”----------JAL R4, -3(16’h99FD)

B”1001_111_000000000”----------JAL R7, 0(16’h9E00)