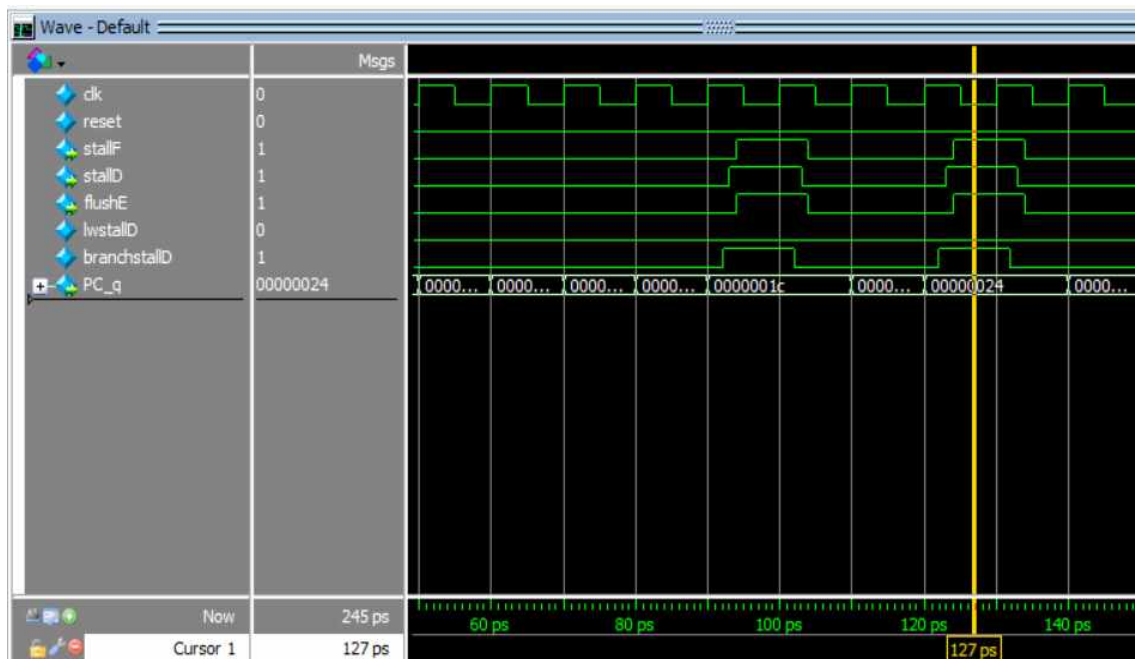


Computer Architecture

HOMEWORK: pipelined MIPS simulation and analysis

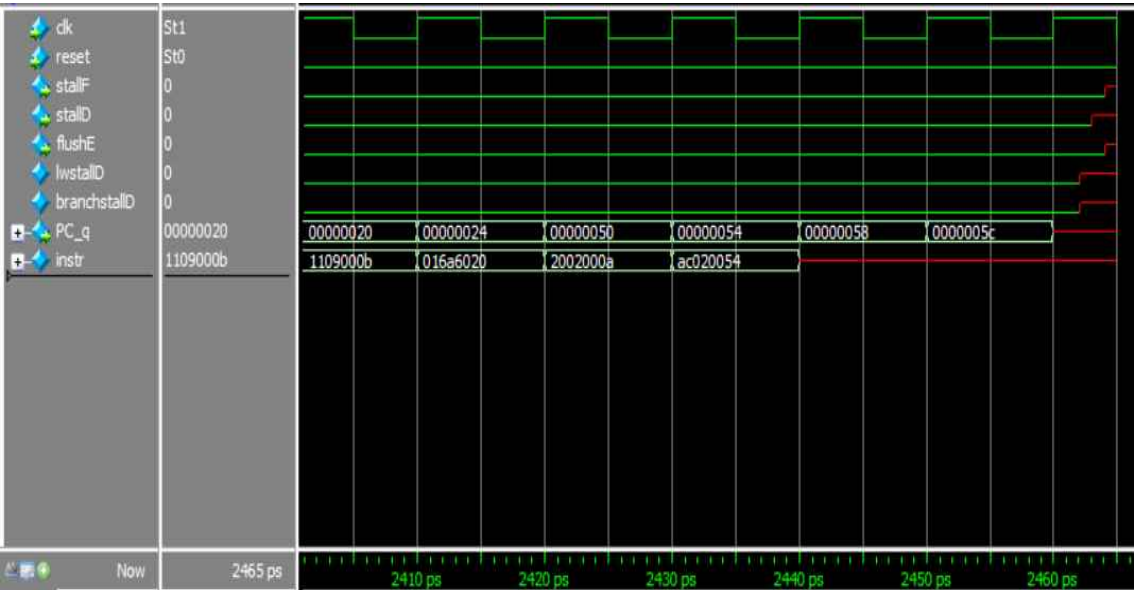
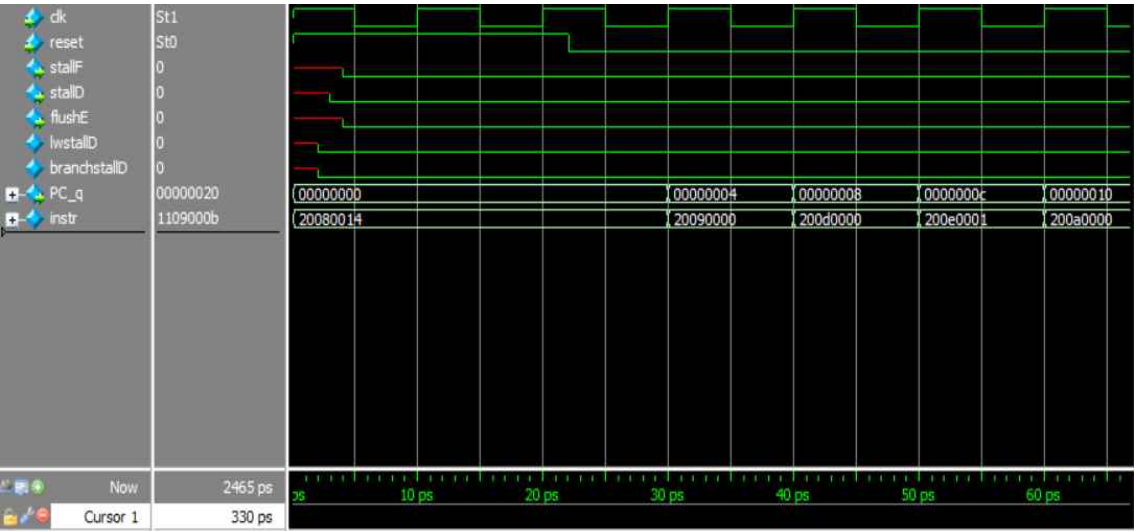
학번 : 2016707079 이름 : 하상천

1. Use Modelsim to simulate pipelined MIPS micro-architecture (source code and test files are provided)



`stallF` = `stallD` = `flushE` = `branchstallD` = 1 인 구간이 두 번 나왔다.

2. Use the new test program machine code (fibonacci.dat) for simulation



3. Discuss how many clock cycles are needed to run the new test program and discuss why.

첫 번째 방법으로는 시뮬레이션을 통해 fetch 된 instruction 수를 세고, 마지막에 fetch된 instruction sw가 마무리 되어야 하므로 +3을 하는 것이다. 더하기 3을 하는 이유는 sw는 memory에 저장하고 마무리 되기 때문이다. reset을 빼면 2440ps에 fetch가 마무리되고 20ps에 fetch가 시작된다고 볼 수 있다. 따라서 2440ps에서 20ps를 빼주면 2420ps가 되고 하나의 clock당 10ps 이기 때문에 2420ps를 10ps로 나누어 주면 242번 fetch 됨을 알 수 있다. 여기서 마지막에 fetch된 instruction이 마무리 되어야 하므로 +3을 해주면 245번의 clock cycle이 필요함을 알 수 있다. 이렇게 구할 수 있는 이유는 한번의 clock cycle에 하나의 instruction이 fetch 되기 때문인 것과 new test program에 stall이 없기 때문이다. 만약에 stall이 있었다면, fetch가 되지 않아도 하나의 clock cycle이 진행될 수 있기 때문이다.

두 번째 방법으로는 main문과 반복문, end문으로 나누어서 clock cycle을 계산하는 것이다. main문에서는 8개의 instruction이 fetch되기 때문에 8번의 clock cycle이 필요하다.

반복문에서는 두 가지의 경우로 나눌 수 있다. 11개의 instruction이 fetch 되는 경우와 12개의 instruction이 fetch 되는 경우이다.

먼저 11개의 instruction이 fetch 되는 경우를 보면, branch misprediction penalty 즉 Control Hazard로 fetch 된 instruction이 3번, store1로 이동하는 반복문에서의 instruction이 8번 일어남을 확인할 수 있다. 12개의 instruction이 fetch 되는 경우를 보면, branch misprediction penalty 즉 Control Hazard로 fetch 된 instruction이 3번, store2로 이동하는 반복문에서의 instruction이 9번 일어남을 확인할 수 있다. 11개의 instruction이 fetch 되는 반복문 10번, 12개의 instruction이 fetch 되는 반복문 10번이 일어난다. 그리고 반복문을 탈출하는 instruction beq \$t0, \$t1, end 와 branch misprediction penalty 즉 Control Hazard로 fetch 된 instruction 1개까지 해서 $(11 \times 10) + (12 \times 10) + 2 = 232$ 번의 clock cycle이 필요하다. 마지막 end문에서는 두 개의 instruction이 fetch 되고, 마지막에 fetch된 instruction sw가 마무리 되어야 하므로 더하기 3을 해주면 5번의 clock cycle이 필요함을 알 수 있다. 따라서 모든 clock cycle 수를 더하면 $8 + 232 + 5 = 245$ 가 되어서 245번의 clock cycle이 필요함을 알 수 있다.