**Bonus:-**

1. 2-Bit branch predictor.

2. Extra instructions implemented :-

1. orr

2.and

3.mvn

4. beq

5. ble

6. Bgt.

7. If latency not found then default latency assignment is done.

8. All error show in one compilation

1. Algorithm is give in pdf file documentation

2. For normal code where there is no hazard it will be five times

   fast. (or data hazard of alu type instruction.)

3. For code with hazards it depends on program that how many

   hazards it contain (like data hazard for ldr we need stall which

   makes program slow).

TESTCASES:

I have provided 6 different test cases.

Two the testcase you said to put fibonacci.s  arraytraversal.s

Rest four are made by me

1.sum.s - performs sum of first n elements (overall testing program tests stall,data hazard,control hazard,branch prediction).

Specific testing program.

2.datahazard.s - testing program for data hazard and stalling.

3. controlhazard.s - testing program for control hazard.

4.testforwarding.s - test program for forwarding.s