**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI - RAJASTHAN**

**Lab Sheet- Enhancing performance with pipelining**

**Objectives:**

At the end of this lab session you should be able to:

* Execute simple programs in
* in non-pipelined processor
* pipelined processor
* Compare the performance of pipelined processor versus non pipelined processor in terms of speed up, efficiency and throughput
* Understand data and control hazards

**Prerequisites: Module No. 7 - Scalar and Super Scalar Instruction Pipeline**

**Contents:**

1.0 Program execution using non-pipelined processor and pipelined processor

1.1 : Non-pipelined processor

1.2 : Pipelined processor

2.0 Performance analysis of pipelined processor with reference to data hazards.

2.1 Method 1: Rearranging the instructions

2.2 Method 2: Inserting bubbles

2.3 Method 3: Operand forwarding

3.0 Performance analysis of pipelined processor with reference to control hazards.

3.1 Pipelined processor without branch prediction

3.2 Pipelined processor with branch prediction

**1.0 Program execution using non-pipelined processor and pipelined processor**

**1.1 Program execution using non-pipelined processor**

**Step 1:** Enter the following program in the CPU-OS simulator

**MOV #10, R00**

**MOV #20, R01**

**MOV #30, R02**

**ADD R02, R01**

**SUB R02, R01**

**SUB R00, R02**

**HLT**

**Step 2:** Press Cache – Pipeline tab and select “SHOW PIPELINE” button as shown in Figure 1. Instruction Pipeline window appears on the screen which is as shown in Figure 2. Click on “No Instruction Pipeline” check box. Reset all the registers by pressing “RESET ALL” button.

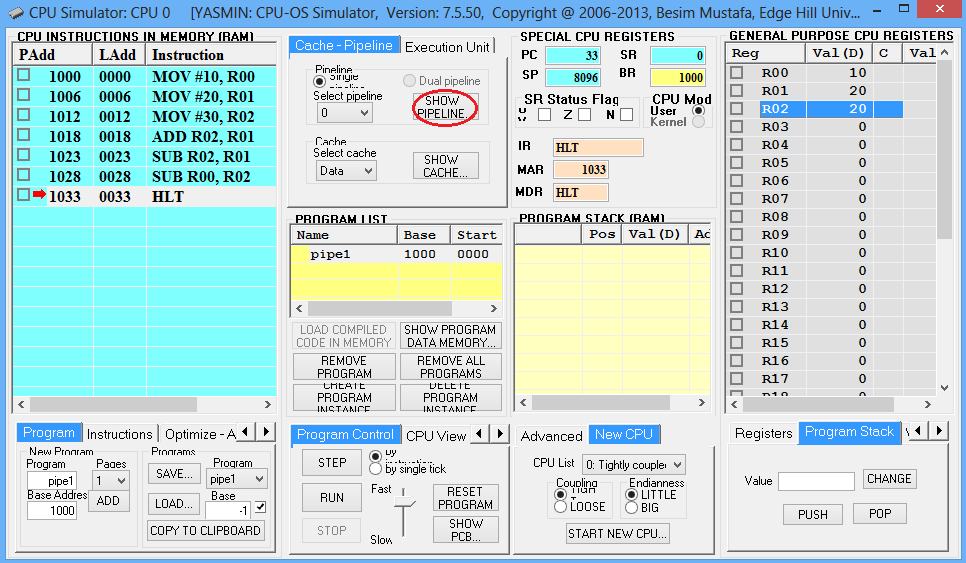
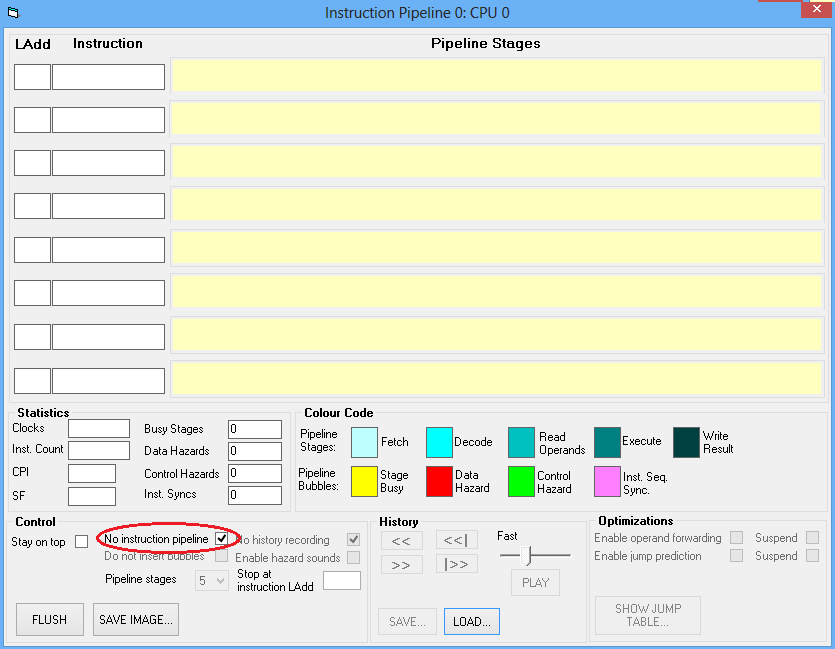


Figure 1: CPU – OS main window

****

**Figure 2: Instruction Pipeline Window**

**Step 3:** Go to “program control” menu (Refer Figure 1) and execute the program in single step. Observe the changes in “Instruction Pipeline Window”

**Step 4:** Note down the contents of various registers during each step.

**Step 5**: Answer the following questions:

1. How many stages are there in non-pipelined processor? List them
2. How many clock cycles are required to execute the program ?
3. Fill in the following table.

|  |  |
| --- | --- |
| Register | Contents after execution |
| R00 |  |
| R01 |  |
| R02 |  |

**1.2 Program execution using pipelined processor**

**Step 1:** Enter the following program in the CPU-OS simulator

**MOV #10, R00**

**MOV #20, R01**

**MOV #30, R02**

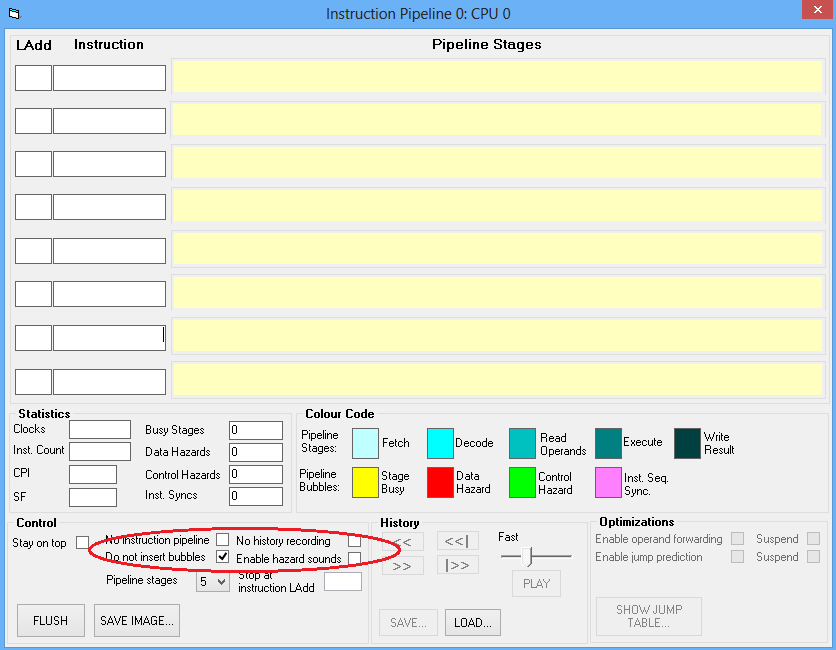
**ADD R02, R01**

**SUB R02, R01**

**SUB R00, R02**

**HLT**

**Step 2:** Press Cache – Pipeline tab and select “SHOW PIPELINE” button as shown in Figure 1. Instruction Pipeline window appears on the screen which is as shown in Figure 3. Clear “No Instruction Pipeline” and “No History recording” check box and select “Do not insert bubbles” check box. Reset all the registers by pressing “RESET ALL” button.



**Figure 3: Instruction Pipeline Window**

**Step 3:** Go to “program control” menu (Refer Figure 1) and execute the program in single step. Observe the changes in “Instruction Pipeline Window”

**Step 4:** Note down the contents of various registers during each step.

**Step 5**: Answer the following questions:

1. How many stages are there in non-pipelined processor? List them
2. How many clock cycles are required to execute the program ?
3. Fill in the following table.

|  |  |
| --- | --- |
| Register | Contents after execution |
| R00 |  |
| R01 |  |
| R02 |  |

1. Compare the results obtained under non-pipelined execution and pipelined execution.

**2.0 Performance analysis of pipelined processor with reference to data hazards.**

In the last experiment, it is observed that there is data hazard while executing the program. In this experiment, various techniques to overcome the data hazard are implemented.

**2.1 Method 1: Rearranging the instructions**

**Step 1:** Modify the above program by rearranging the instruction as given below

MOV #20, R01

MOV #30, R02

MOV #10, R00

ADD R02, R01

SUB R02, R01

SUB R00, R02

HLT

**Step 2:** Press Cache – Pipeline tab and select “SHOW PIPELINE” button as shown in Figure 1. Instruction Pipeline window appears on the screen which is as shown in Figure 3. Clear “No Instruction Pipeline” and “No History recording” check box and select “Do not insert bubbles” check box. Reset all the registers by pressing “RESET ALL” button.

**Step 3:** Go to “program control” menu (Refer Figure 1) and execute the program in single step. Observe the changes in “Instruction Pipeline Window”

**Step 4:** Note down the contents of various registers during each step.

**Step 5**: Fill in the following Table 1.

**2.2 Method 2: Inserting bubbles**

**Step 1:** Enter the following program (Similar to Experiment 1.1 and 1.2)

MOV #10, R00

MOV #20, R01

MOV #30, R02

ADD R02, R01

SUB R02, R01

SUB R00, R02

HLT

**Step 2:** Press Cache – Pipeline tab and select “SHOW PIPELINE” button as shown in Figure 1. Instruction Pipeline window appears on the screen which is as shown in Figure 3. Clear “No Instruction Pipeline” “No History recording” and “Do not insert bubbles” check box. Reset all the registers by pressing “RESET ALL” button.

**Step 3:** Go to “program control” menu (Refer Figure 1) and execute the program in single step. Observe the changes in “Instruction Pipeline Window”

**Step 4:** Note down the contents of various registers during each step.

**Step 5**: Fill in the following table 1.

**2.3 Method 3: Operand forwarding**

**Step 1:** Enter the following program (Similar to Experiment 1.1 and 1.2)

MOV #10, R00

MOV #20, R01

MOV #30, R02

ADD R02, R01

SUB R02, R01

SUB R00, R02

HLT

**Step 2:** Press Cache – Pipeline tab and select “SHOW PIPELINE” button as shown in Figure 1. Instruction Pipeline window appears on the screen which is as shown in Figure 3. Clear “No Instruction Pipeline” and “No History recording”. Select “Do not insert bubbles” and “Enable Operand Forwarding” check box. Reset all the registers by pressing “RESET ALL” button.

**Step 3:** Go to “program control” menu (Refer Figure 1) and execute the program in single step. Observe the changes in “Instruction Pipeline Window”

**Step 4:** Note down the contents of various registers during each step.

**Step 5**: Fill in the following table 1.

**Table 1: Performance analysis of Pipeline processor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Register | Contents after execution (Experiment 1.1) | Contents after execution (Experiment 1.1) | Contents after execution (Experiment 2.1) | Contents after execution (Experiment 2.2) | Contents after execution (Experiment 2.3) |
| R00 |  |  |  |  |  |
| R01 |  |  |  |  |  |
| R02 |  |  |  |  |  |
| No. of clock cycles |  |  |  |  |  |
| Speed Up Factor |  |  |  |  |  |

**3.0 Performance analysis of pipelined processor with reference to control hazards.**

**3.1 Pipeline execution without jump prediction**

Step 1: Enter the following program

1000 MOV #05, R00

1006 SUB #01, R00

1012 JEQ 06

1026 MOV #10, R00

1022 HLT

Step 2: Press Cache – Pipeline tab and select “SHOW PIPELINE” button as shown in Figure 1. Instruction Pipeline window appears on the screen which is as shown in Figure 2. Clear “No Instruction Pipeline”, “Do not insert bubbles”, “No History recording” and “Enable Jump Prediction” check boxes. Select “Enable Operand Forwarding” check box. Reset all the registers by pressing “RESET ALL” button.

Step 3: Go to “program control” menu (Refer Figure 1) and execute the program in single step. Observe the changes in “Instruction Pipeline Window”. Note down number of clock cycles, CPI, Speedup Factor (SF) and Control Hazards in Table 2.

Step 4: In instruction pipeline ( Figure 2) press “SHOW JUMP TABLE…” tab. A window similar to Figure 4 will appear. Note down total conditional jumps and correct prediction in Table 2.

**3.1 Pipeline execution with jump prediction**

Step 1: Enter the following program

1000 MOV #05, R00

1006 SUB #01, R00

1012 JEQ 06

1026 MOV #10, R00

1022 HLT

Step 2: Press Cache – Pipeline tab and select “SHOW PIPELINE” button as shown in Figure 1. Instruction Pipeline window appears on the screen which is as shown in Figure 2. Clear “No Instruction Pipeline”, “Do not insert bubbles”, “No History recording” and “Enable Jump Prediction” check boxes. Select “Enable Operand Forwarding” check box. Reset all the registers by pressing “RESET ALL” button.

Step 3: Go to “program control” menu (Refer Figure 1) and execute the program in single step. Observe the changes in “Instruction Pipeline Window”. Note down number of clock cycles, CPI, Speedup Factor (SF) and Control Hazards in Table 2.

Step 4: In instruction pipeline ( Figure 2) press “SHOW JUMP TABLE…” tab. A window similar to Figure 4 will appear. Note down total conditional jumps and correct prediction in Table 2.

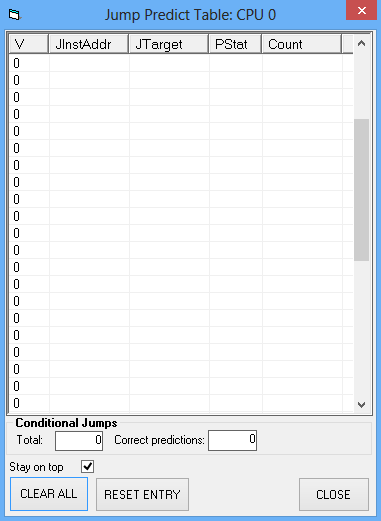


Figure: Jump Predict Table

Table 2: Performance analysis of pipeline without and with jump prediction

|  |  |  |
| --- | --- | --- |
|  | Without Jump Prediction | With Jump Prediction |
| Clocks |  |  |
| CPI |  |  |
| Speedup Factor |  |  |
| Control Hazards |  |  |
| Total Conditional Jumps |  |  |
| Correct prediction |  |  |