On Processors, Pipelines and Simulators

Suren | 2021CS10072
Pravar Kataria | 2021CS10075

April 15, 2023

§1 On pipelines

§1.1 5 stage pipeline

The first pipeline to be simulated for the assignment was the basic 5 stage MIPS pipeline. We made a structure defining 32 registers, and locks on the registers which specify whether the register would be written to by the instructions which are currently being processed. This helps us to introduce stalls in the pipeline whenever required, wherein, an instruction has to wait at the decode stage for all instructions ahead of it, which would modify registers it needs, to finish their execution.

§1.2 5 stage pipeline with bypassing

The overall idea is basically the same as above with one small change. We maintain a set of dummy registers, which represent all the data that is either in the registers or the latches in between different pipeline stages. This ensures that the data in all these dummy registers is the data which should be used in all computations, since it encaptures full information about the state of the registers as well as the next few writes (to registers) which would happen.

§1.3 7-9 stage pipeline

The main difference in the implementation of this pipeline and the 5 stage pipelines is that of the first-in-first-out constraint. To satisfy this constraint during the execution of the program, we have kept an instruction number along with each of the instructions, and we keep track of the number of instructions that have written their result back to the register file at any point of time. These two things let us decide which of the two write back stages of the processor would use the write port of the register file (since there is only one). One thing to be noted is that instructions like `beq, bne, j` and `sw` do not write anything back to the registers, so they would just 'annihilate' upon reaching the write back stage of the pipeline, irrespective of what the other write back stage is doing. This is more of a design decision than a sacrosanct rule.

§1.4 7-9 stage pipeline with bypassing

There are only two sources for forwarding in this case, namely the latches before the writeback stage. So, we keep track of all the data that is being written into the registers,

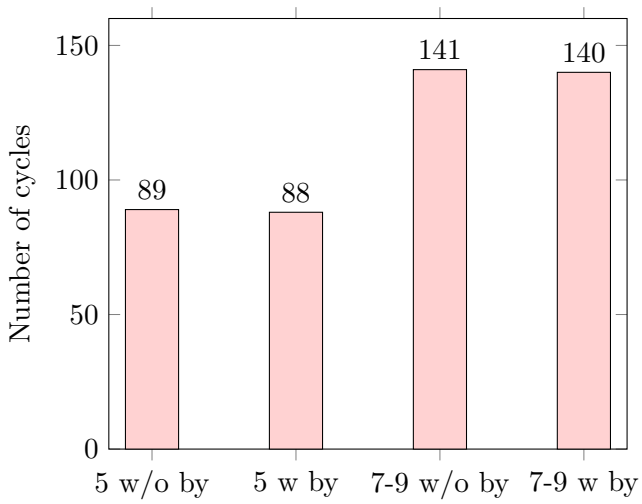
just before they are about to be written to registers. This allows us to forward some of the data.

§2 Cycle Comparison Plots

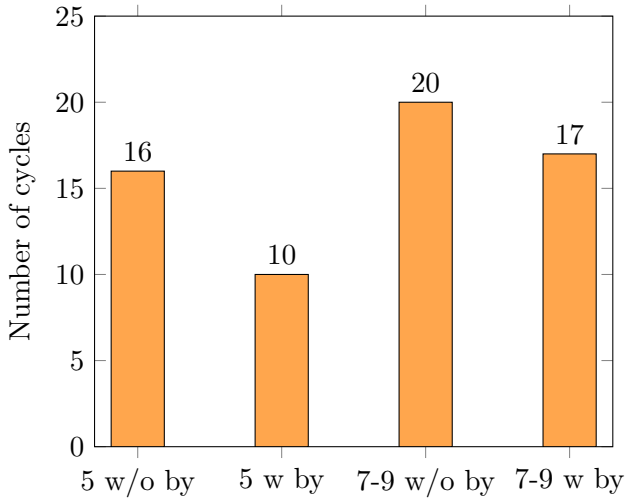
This section makes comparisons between the number of cycles required to run the given test cases in four different kinds of settings:

- 5 stage pipeline
- 5 stage pipeline with bypassing
- 7-9 stage pipeline
- 7-9 stage pipeline with bypassing

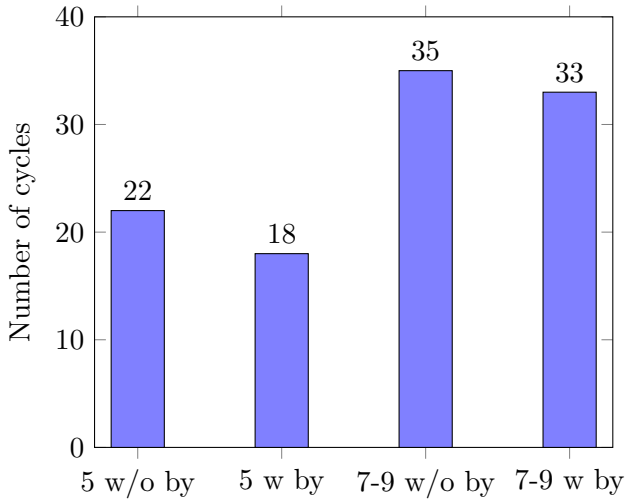
§2.1 Test case 0



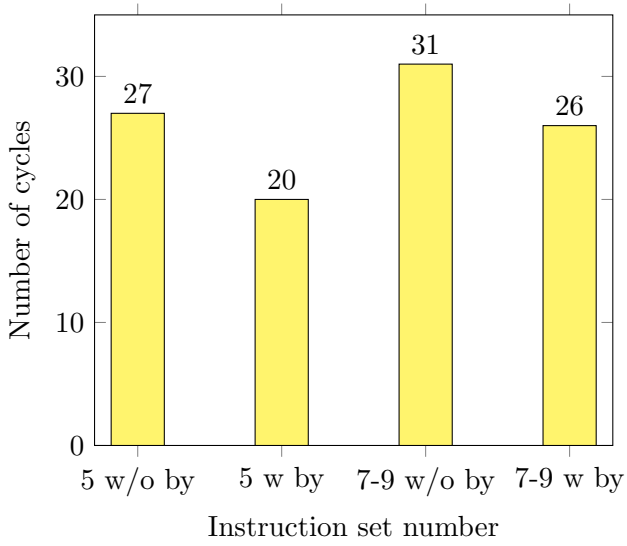
§2.2 Test case 1



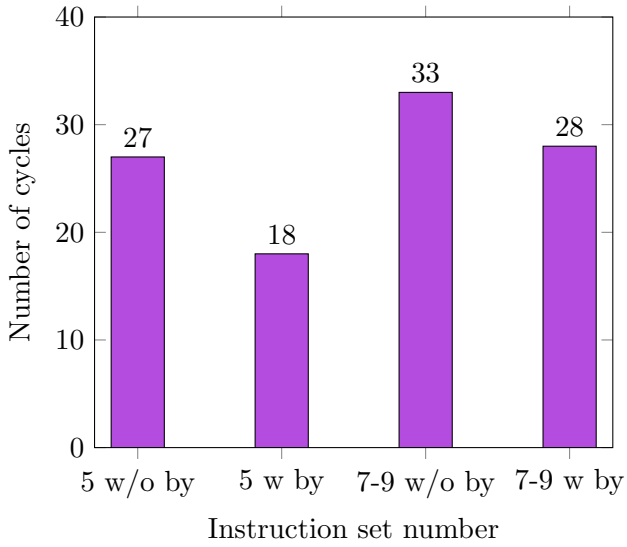
§2.3 Test case 2



§2.4 Test case 3

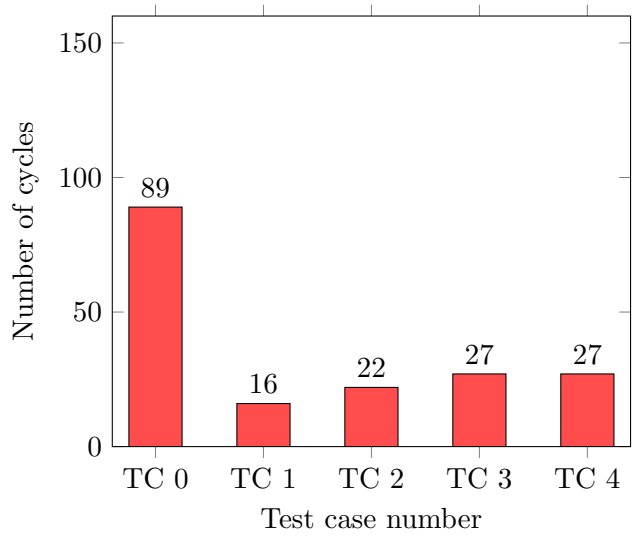


§2.5 Test case 4

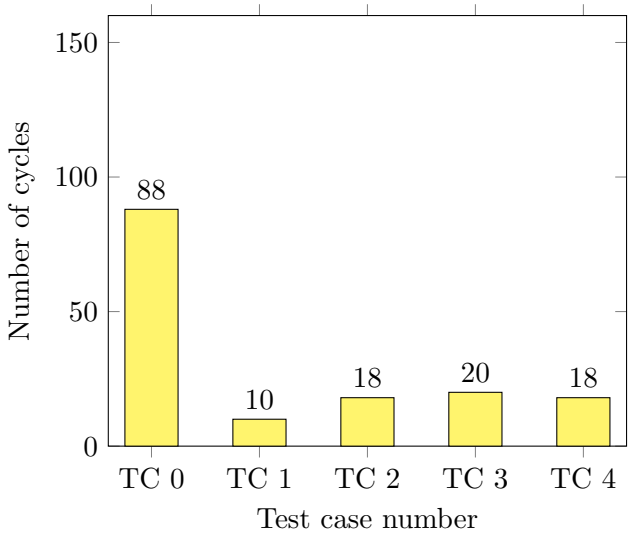


§3 Cycles vs test case plot for all 4 different strategies

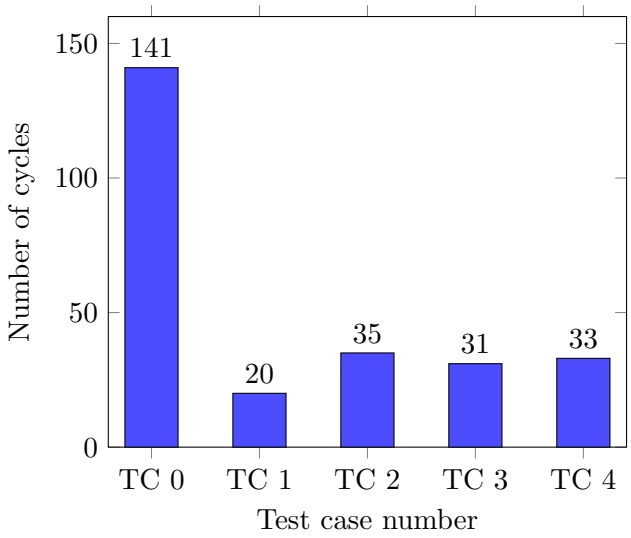
§3.1 5 stage pipeline



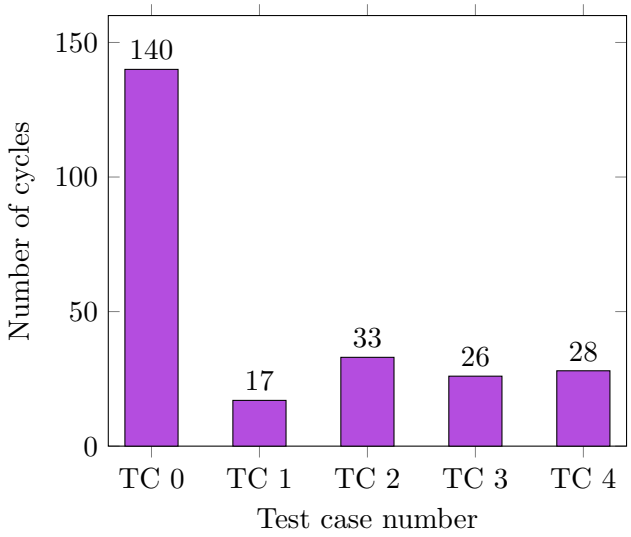
§3.2 5 stage pipeline with bypass



§3.3 7-9 stage pipeline



§3.4 7-9 stage pipeline with bypass



§4 Analysing the graphs

These graphs show the clear pattern that the nubmer of cycles needed for the 7-9 stage pipelines is slightly higher in number. This is mainly because of two reasons:

- Since our test cases are pretty small, one reason is that the warming-up time for the two pipelines is different (because of different number of stages in the two pipelines), and this makes a small albeit significant contribution in such small test cases.
- Another reason is that the dependencies in case of the memory access and write instructions are resolved later, because of more stages in the pipeline for memory type instructions.

§5 Branch Predictions

§5.1 Saturating 2 Bit counter

This is the prediction model which uses address branching history which has a 2-bit saturating counter for each address.
The structure consists a table of size 2^{14} with 2 bit saturating counters for each address. 'taken' is predicted when the 2-bit counter is either 2 or 3, 'not taken' otherwise.

Initial value of 2-bit SC	Correct Predictions	Wrong predictions	Hit-rate
Initial value = 0	433	115	79.02 %
Initial value = 1	460	88	83.94 %
Initial value = 2	482	66	87.95 %
Initial value = 3	475	73	86.68 %

Table for 2 bit SC predictor

Initial value	Total count	True +ve	False +ve	True -ve	False -ve
0	548	279	15	154	100
1	548	311	20	149	68
2	548	365	52	117	14
2	548	374	68	101	5

Summarization of data (in terms of false positives and false negatives)

§5.2 BHR (Branch History register)

This is the prediction model which uses global branching history which is unrelated to any address.
The structure consists a table of size 4 with 2 bit saturating counters for each state of BHR. 'taken' is predicted when BHR counter is either 2 or 3, 'not taken' otherwise.

Initial value of 2-bit SC	Correct Predictions	Wrong predictions	Hit-rate
Initial value = 0	392	156	71.53 %
Initial value = 1	395	153	72.26 %
Initial value = 2	398	150	72.63 %
Initial value = 3	399	149	72.81 %

Table for BHR predictor

Initial value	Total count	True +ve	False +ve	True -ve	False -ve
0	548	318	95	74	61
1	548	323	96	73	56
2	548	327	98	71	52
2	548	328	98	71	51

Summarization of data (in terms of false positives and false negatives)

§5.3 Combination of BHR and address 2 bit SC

This is the prediction model which uses a combination of both global branching history which is unrelated to any address and 2-bit saturating counters which are associated with addresses.
The structure consists a table of size 2^{14} with 2 bit saturating counters for each state. 'taken' is predicted when counter is either 2 or 3, 'not taken' otherwise.

The basic logic used to derive combination is :

1. If address associated 2Bit saturating counters show strongly towards one side. Agree to it
2. Otherwise if BHR agrees strongly towards one side, agree to it
3. Otherwise listen to 2Bit saturating counters of addresses.

Initial value of 2-bit SC	Correct Predictions	Wrong predictions	Hit-rate
Initial value = 0	440	108	80.29 %
Initial value = 1	465	83	85.03 %
Initial value = 2	473	75	86.31 %
Initial value = 3	469	79	85.58 %

Table for combination predictor

Initial value	Total count	True +ve	False +ve	True -ve	False -ve
0	548	290	19	150	89
1	548	328	31	138	51
2	548	358	54	115	21
2	548	372	72	97	7

Summarization of data (in terms of false positives and false negatives)

§5.4 Comparison of different prediction methods

Initial value of counters	Hit Rate: 2Bit SC	Hit Rate: BHR	Hit-rate: Combination
Initial value = 0	79.02 %	71.53 %	80.29 %
Initial value = 1	83.94 %	72.26 %	85.03 %
Initial value = 2	87.95 %	72.63 %	86.31 %
Initial value = 3	86.68 %	72.81 %	85.58 %

Comparison Table for different predictors

§6 Division of work among team mates

Regarding the distribution of work for the assignment, both of us have worked on this assignment equally. Hence, we have decided to split the tokens equally among us.

Name	**Tokens**
Suren	50
Pravar Kataria	50

Token Distribution

§7 Acknowledgements

We have used the style file from here[1] to produce this document.

[1] `https://github.com/vEnhance/dotfiles/blob/main/texmf/tex/latex/evan/evan.sty`