assignment-4 report

Choudari Harshitha Reddy(220010015)
Mubarakpur Keerthi(220010032)

1 observation

stat file of all the test cases in the folder

1.1 descending

Number of static instructions executed in the assembly file = 21
Number of dynamic instructions executed in the assembly file= 368
Number of cycles taken to complete = 129
IPC = 2.852713
Frequency = 0.19907407 GHz
Number of stalls in the assembly file= 192
Number of incorrect branches in the assembly file= 88

1.2 evenorodd

Number of static instructions executed in the assembly file = 9
Number of dynamic instructions executed in the assembly file= 7
Number of cycles taken to complete = 4
IPC = 1.75
Frequency = 0.1904762 GHz
Number of stalls in the assembly file= 14
Number of incorrect branches in the assembly file= 0

1.3 fibonacci

Number of static instructions executed in the assembly file = 21
Number of dynamic instructions executed in the assembly file= 96
Number of cycles taken to complete = 36
IPC = 2.6666667
Frequency = 0.2 GHz
Number of stalls in the assembly file= 68
Number of incorrect branches in the assembly file= 16

1.4 palindrome

Number of static instructions executed in the assembly file = 16
Number of dynamic instructions executed in the assembly file= 57
Number of cycles taken to complete = 28
IPC = 2.0357144
Frequency = 0.1958042 GHz
Number of stalls in the assembly file= 79
Number of incorrect branches in the assembly file= 7

1.5 prime

Number of static instructions executed in the assembly file = 16
Number of dynamic instructions executed in the assembly file= 35
Number of cycles taken to complete = 13
IPC = 2.6923077
Frequency = 0.19117647 GHz
Number of stalls in the assembly file= 28
Number of incorrect branches in the assembly file= 5

test case	**dynamic instructions executed**	**cycles**	**incorrect branches**	**stall**
descending	368	129	88	192
even or odd	7	4	14	0
fibonacci	96	36	68	16
palindrome	57	28	79	7
prime	35	13	28	5

Table 1: Caption

2 conclusion

- The number of cycles taken with using pipeline is less than the number of cycles taken without using pipeline.
- The more the dynamic instructions in the program , the more is the chances that instruction enters a wrong path in pipeline and also, the more is the frequency of Data Hazard occurring due to Read After Write Hazard.
- While the Fibonacci and Palindrome algorithms exhibit relatively higher instruction execution and cycle consumption, they also demonstrate more instances of wrong branch path entries and data hazard stalls.

- The frequency of instructions entering wrong path also depends upon the input data values stored in register and memory and whether the branch conditions satisfy to true or not, satisfying to true leads to instruction being fetched on wrong data path
- the more number of times branch condition satisfies to true, the more is number of times instruction entered wrong path in pipeline and we need to set that instruction to Nop and hence it will lead to wastage of that cycle and hence increasing the number of cycles
- The more is the Read After Write operations from the same position, the more will be the Number of times OF stage needed to be stalled and it will therefore also increase the number of cycles because of stalling.
- But the total number of cycles taken will be still less than the case when we are not using pipelining.