CS-311 Assignment-5 Lab Report

K.S.N Manikanta (210010050)
M. Suresh (210010030)

Results:

Without Pipeline(Assignment3)

TEST CASE	Hash of state	Number of instructions	Number of cycles
descending.out	255541867	277	1135
evenorodd.out	-224294686	6	30
fibonacci.out	-1518357572	78	390
palindrome.out	155317940	49	245
prime.out	-1414219998	29	145

With Pipeline(Assignment4)

TEST CASE	Hash of state	Number of Instructions	Number of cycles	No of OF stalls	No of Wrong branch instructions	Throughput (IPC)
descending.out	255541867	277	658	126	220	**0.4209726**
evenorodd.out	-224294686	6	19	10	4	**0.3157894**
fibonacci.out	-1518357572	78	157	44	36	**0.4968152**
palindrome.out	155317940	49	124	51	18	**0.3951612**
prime.out	-1414219998	29	79	19	28	**0.3670886**

For a Discrete Event Simulator(Assignment5)

Test Case	*Hash of State*	*No. of instructions*	*No. of cycles*	*Throughput(IPC)*
descending.out	255541867	277	11779	0.023516428
evenorodd.out	-224294686	6	259	0.023166023
fibonacci.out	-1518357572	78	3581	0.021781625
palindrome.out	155317940	49	2062	0.023763336
prime.out	-1414219998	29	1219	0.023789993

Observations:

The hash of the state remains the same, showing that our program is functionally correct. We can see that the number of cycles for each benchmark program has increased compared to the number of cycles we got for the pipelined processor in assignment 4. The throughput of an ideal pipelined processor is 1. Our pipelined processor with data interlocks and branch interlocks has throughputs(IPC) between 0.35 and 0.5 whereas the throughputs(IPC) we obtained for the benchmark programs are between 0.021 and 0.024. The increase in the number of cycles is due to the latency of the main memory access, i.e., more cycles are taken to access the memory for operations like memory read in the instruction fetch stage, memory write in the memory access stage and so on.