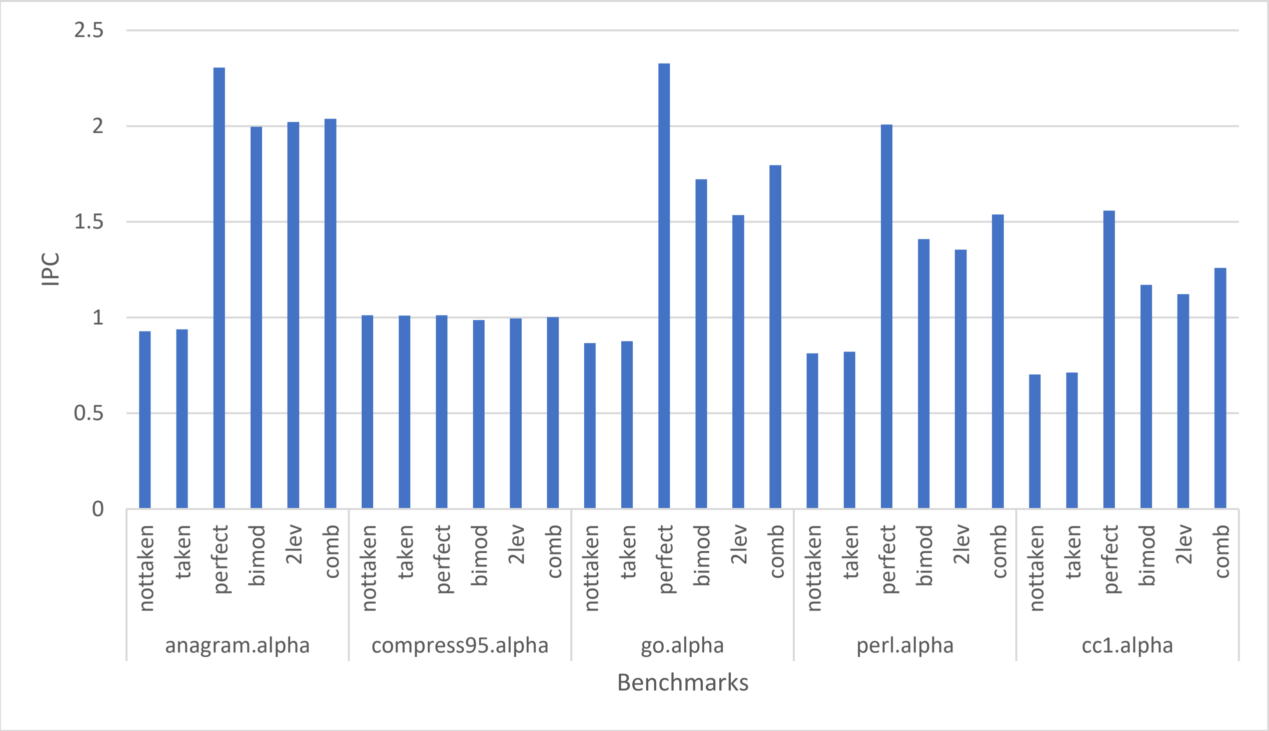
Name: Maahir Kalban (ert547)

To evaluate the performance impact of different branch predictors.

Table of Results From Benchmak Simulations

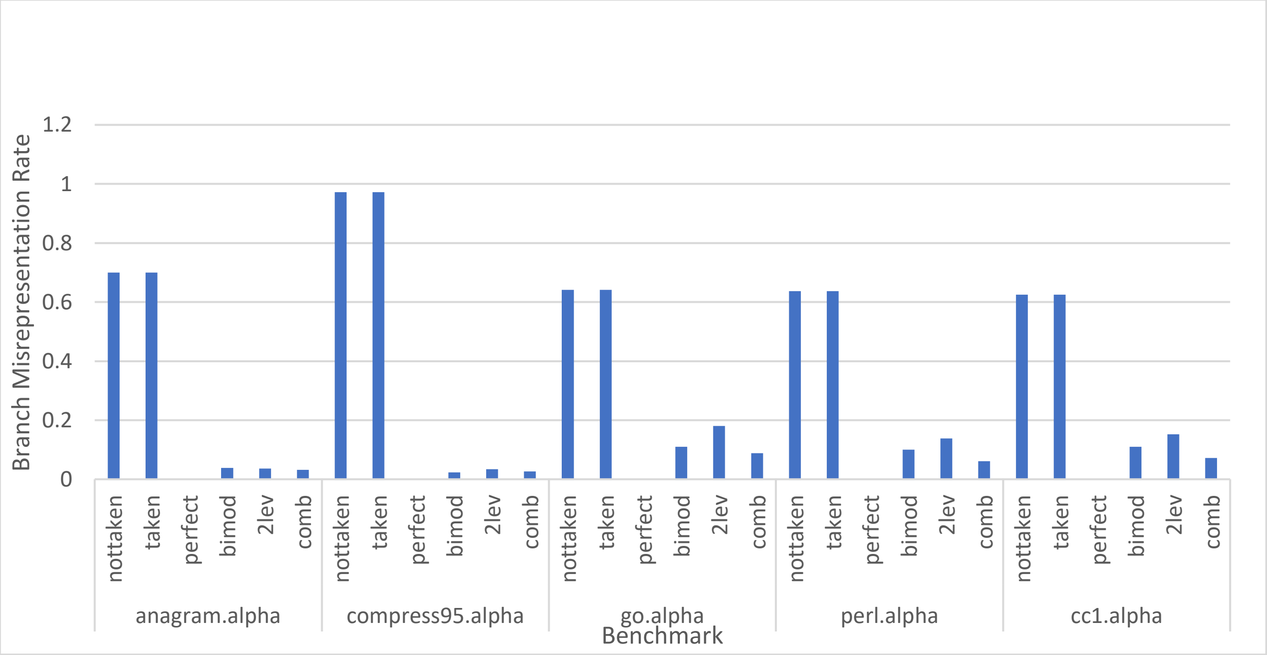
Benchmarks	Branch Predictor	Performance			
		Misses	NumOfBranches	Misprediction Rate	Sim_IPC
anagram.alpha	nottaken	2005971	2867987	0.699435179	0.929
	taken	2005971	2867987	0.699435179	0.9376
	perfect	0	2867987	0	2.3062
	bimod	110531	2867987	0.038539575	1.9961
	2lev	103027	2867987	0.035923106	2.0213
	comb	93047	2867987	0.032443313	2.0379
compress95.alpha	nottaken	5049	5195	0.971896054	1.0112
	taken	5049	5195	0.971896054	1.0105
	perfect	0	5195	0	1.0127
	bimod	122	5195	0.023484119	0.9865
	2lev	175	5195	0.033686237	0.9959
	comb	136	5195	0.026179018	1.0015
go.alpha	nottaken	1835629	2861944	0.641392354	0.8666
	taken	1835629	2861944	0.641392354	0.877
	perfect	0	2861944	0	2.3274
	bimod	314864	2861944	0.110017527	1.723
	2lev	515014	2861944	0.179952508	1.5345
	comb	253869	2861944	0.08870509	1.7961
perl.alpha	nottaken	2062056	3235173	0.637386625	0.8124
	taken	2062056	3235173	0.637386625	0.8208
	perfect	0	3235173	0	2.008
	bimod	323630	3235173	0.100034836	1.409
	2lev	448566	3235173	0.138652863	1.3551
	comb	196820	3235173	0.06083755	1.5386
cc1.alpha	nottaken	2052452	3284742	0.624844204	0.7032
	taken	2052452	3284742	0.624844204	0.7127
	perfect	0	3284742	0	1.5591
	bimod	362765	3284742	0.11043942	1.1699
	2lev	501120	3284742	0.152559927	1.1227
	comb	235321	3284742	0.071640634	1.2588

Graph to show Simulation Benchmarks vs IPC (Instructions per Cycle)



Findings: For all Benchmark Simulations highest (i.e Best) IPC is Perfect Branch Predictor and lowest (i.e Worst) are takken and nottaken Branch Predictors

Graph to show Simulation Benchmarks vs Branch misprediction Rate



Findings: For all Benchmark Simulations lowest (i.e Best) Branch Misprediction Rate is Perfect Branch Predictor and highest (i.e Worst) are taken and nottaken Branch Predictors

To evaluate the performance impact of out-of-order execution

Benchmarks	executions	Performance	
		NumOfCycles	Sim_IPC
anagram.alpha	inorder	25323163	0.7898
	outOfOrder	9881948	2.0239
compress95.alpha	inOrder	204965	0.4302
	outOfOrder	88677	0.9943
go.alpha	inOrder	25410489	0.7871
	outOfOrder	11982993	1.669
perl.alpha	inOrder	26628101	0.7511
	outOfOrder	13699908	1.4599
cc1.alpha	inOrder	31725461	0.6304
	outOfOrder	17139843	1.1669

Findings: For all the benchmarks the out-of-order executions performs better that than in-order

To evaluate the performance impact of different instruction decoding width and issue width on out-of-order execution.

Benchmarks	decode:width	issue:width	Performance	
			ExecutionCycles	Sim_IPC
anagram.alpha	4	4	9881948	2.0239
	8	8	8313977	2.4056
	16	16	8299076	2.4099
	32	32	8299076	2.4099
compress95.alpha	4	4	88677	0.9943
	8	8	83600	1.0547
	16	16	83594	1.0547
	32	32	83594	1.0547
go.alpha	4	4	593145	1.4543
	8	8	512949	1.6817
	16	16	508839	1.6953
	32	32	508839	1.6953
perl.alpha	4	4	13699908	1.4599
	8	8	12307012	1.6251
	16	16	12248955	1.6328
	32	32	12249973	1.6327
cc1.alpha	4	4	17139843	1.1669
	8	8	15664210	1.2768
	16	16	15544230	1.2867
	32	32	15536402	1.2873
	64	64	15536402	1.2873

Findings: For Best Performance Decode and Issue width are 16 for (anagram.alpha, compress95.aplha, go.alpha and perl.alpha) and 32 for cc1.alpha