

### 1.0 Microbenchmark

The microbenchmark was used to test the validity of the 2level predictor. The compilation flags used were gcc mb.c -O0 -o mb. In the benchmark, we tested the history prediction and correlated prediction. This history prediction has repeating patterns due to the for loop and thus, it generates a set of repeating history bits for that conditional branch. Because all the conditional branches are close together, and we index using the least significant bits, all the conditional branches map to different entries in the table. We ran the outer for loop on Line 22 1M and 10M times to find out how many branches and mispredictions were occurring from this outer for loop and additional external library code. Then we ran each conditional branch, first on Line 31 and then on Line 46, 50 and 54. We figured out how many mispredictions occurred for the branch on Line 31, and traced the program to see if the 2level predictor supported this. It did. Therefore, it verified our 2level solution. A more detailed explanation is found in the microbenchmark code.

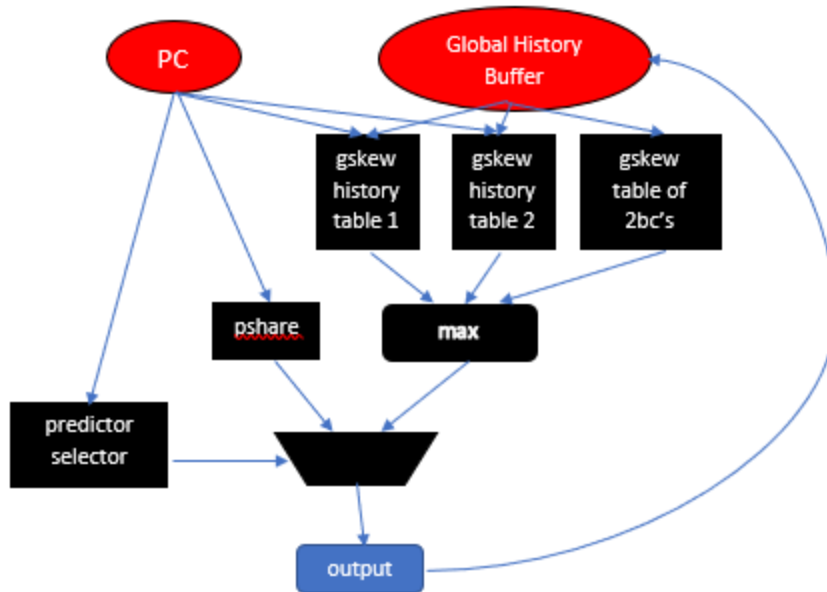
### 2.0 Benchmarks vs. Predictors

	2bitsat		2level		openend	
	Mispred. Branches	MPKI	Mispred. Branches	MPKI	Mispred. Branches	MPKI
astar	3695830	24.639	1785464	11.903	856221	5.708
bwaves	1182969	7.886	1071909	7.146	900103	6.001
bzip2	1224967	8.166	1297677	8.651	1214545	8.097
gcc	3161868	21.079	2223671	14.824	941696	6.278
gromacs	1363248	9.088	1122586	7.484	836279	5.575
hmmer	2035080	13.567	2230774	14.872	1970397	13.136
mcf	3657986	24.387	2024172	13.494	1669274	11.128
soplex	1065988	7.107	1022869	6.819	843411	5.623
microbenchm ark	1535444	32.916	1929	0.041	68646	1.472
avg MPKI(8)	14.49		10.65		7.69	

### 3.0 Open-ended Branch Predictor

The openended predictor was implemented using a tournament style predictor. There were two main components, the pshare and gskew sections. The pshare was a 2level predictor. The gskew consisted of 3 predictors. A very large gshare history table (GHT 1), a smaller gshare history table (GHT 2) and a table of 2bc's (GHT 2bc). Each table was hashed using a different algorithm. The max prediction of the three gskew components was taken to determine the final gskew prediction. Then, a predictor selector (PS) selected which of pshare or gskew to use. The gshare ideas were taken from the YouTube video, Advanced Branch Prediction [1]. The gskew idea was taken from what gskew is, but the 3 predictors were selected by our own accord. A more detailed visual is shown below.

[1] Juurlink, P. D. B. H. [Prof. Dr. Ben H. Juurlink]. (2018, October 22). 2 | 8 Advanced Branch Prediction [Video]. YouTube.  
[https://www.youtube.com/watch?v=avp3bDqCXYM&ab\\_channel=Prof.Dr.BenH.Juurlink](https://www.youtube.com/watch?v=avp3bDqCXYM&ab_channel=Prof.Dr.BenH.Juurlink)



The storage requirements are found below. The total number of bits used was 116,751 bits or 14,594 bytes. The max allowed was 131,072 bits( $128 \times 2^{10}$ ).

global history buffer (bits)	pshare (bits)	GHT 1 (bits)	GHT 2 (bits)	GHT 2bc (bits)	PS (bits)
15	$4096 \times 6 + 16 \times 64 \times 2$	$32768 \times 2$	$4096 \times 2$	$4096 \times 2$	$4096 \times 2$

#### 4.0 Parameters

	2level	openend (config file in open-ended-bpred-X.cfg)		
		pshare (bpred-1.cfg)	GHT 1 (bpred-2.cfg)	GHT 2, GHT 2bc, PS (each) (bpred-3.cfg)
Area (mm <sup>2</sup> )	0.0030028	0.0086828	0.0127178	0.0019355
access latency (ns)	0.265674	0.370057	0.279886	0.164342
leakage power (mW)	0.23978	1.43923	2.87418	0.366395
Modified Parameters	size: 512 block size: 16 bus width: 128 tag size: 6	size: 512 block size: 16 bus width: 128 tag size: 6	size: 8192 block size: 1	size: 1024 block size: 1

#### 5.0 Statement of Work

Bing - Implemented the 2bitsat, 2level predictors, mb, created the 2level CACTI config, report  
 Tapasvi - Implemented the openend predictor, mb, created the openend CACTI configs, report