A Simple ITTAGE for Indirect branch Prediction

Palash Parmar
palparmar@tamu.edu
Texas A&M University

Abstract: *Indirect branch predictor gained important because indirect branches occurs more frequently in object-oriented programming. Even though indirect branches are less frequent then better predictive conditional branches, it creates significant misprediction overhead because of deep pipeline in modern processors. A better speculation of the indirect target could boost the performance of such processors. With this idea, this paper present a fairly simple Indirect Target Tagged GEometric Branch Predictor [1]. ITTAGE sums the idea of many predictors. It uses geometric history length for a different table as in O-GEHL [2], tables are partially tagged as in the case of PPM-like, tag-based branch predictor [3] and finally takes the shape of a hybrid predictor [4] for deciding final prediction.*

I. INTRODUCTION

There is already lots of research progressed in conditional branch predictor and we already have state of art conditional branch predictor available which are very accurate. One such conditional branch predictor is O-GEHL branch predictor [2] which take advantage of multiple history length tables cascaded into single predictor. O-GEHL predictor using its geometric history length based tables prove that global history is sufficient to provide good accuracy.

Another good predictor is PPM-like tag based predictor [3]. Tag-based predictor has good accuracy to because of tagging the tables to get the predictions. Since both the index and tag are calculated from global history and program counter but with different folding length, it additionally increases the accuracy from just using O-GEHL.

PPM-like tag based conditional branch predictor suffers from the poor performance is due to the update policy. TAGE conditional branch predictor introduces new update policy using a useful bit along with the control bits. Useful bits in TAGE suggest whether to use prediction with the longest history or second longest history if longest history is the new entry.

Derived from TAGE predictor is the ITTAGE branch predictor. It has a tagless table which provides default prediction, and multiple tagged table indexed using different history bits as seen in O-GEHL branch predictor. The history length of different tables can be given by $L(i) = (int)(\propto^{i-1} * L(1) + 0.5$. Each entry in tagged table has a useful bit for implementing update policy, 2 bit control bit for providing some hysteresis, a tag and a target column. The overall predictor and table entry can be seen in fig.2 and fig.1 respectively.

Target	Tag	ctr	u

Fig.1 Tagged Table Entry

II. TARGET PREDICTION

For the sake of understanding, let's assume that the prediction with longest history table is main prediction and prediction with second longest history be an alternate prediction. The tag and indexes are calculated prior to access the table for searching valid entries. For calculation of tags and index, we use same folding history method as used in PPM-like tag based predictor and then XOR with program counter. For searching of entries in the table, all tables are accessed simultaneously and main, alternate prediction is calculated. In case of no matching found in tagged tables, the tagless table provides default prediction.

It has been observed that when the entry provided by the main prediction is weak i.e. confidence counter of entry is less than 9 in simulation, it is better to use alternate prediction instead if the main prediction. In simulation, it is monitored by global variable useAlt.

Computation summary:

1. Calculate main and alternate prediction by accessing all tables simultaneously.
2. The alternate prediction will provide the final prediction if the entry is weak i.e. useAlt>=9 and ctr for entry is zero. The Else main prediction will provide a final prediction.

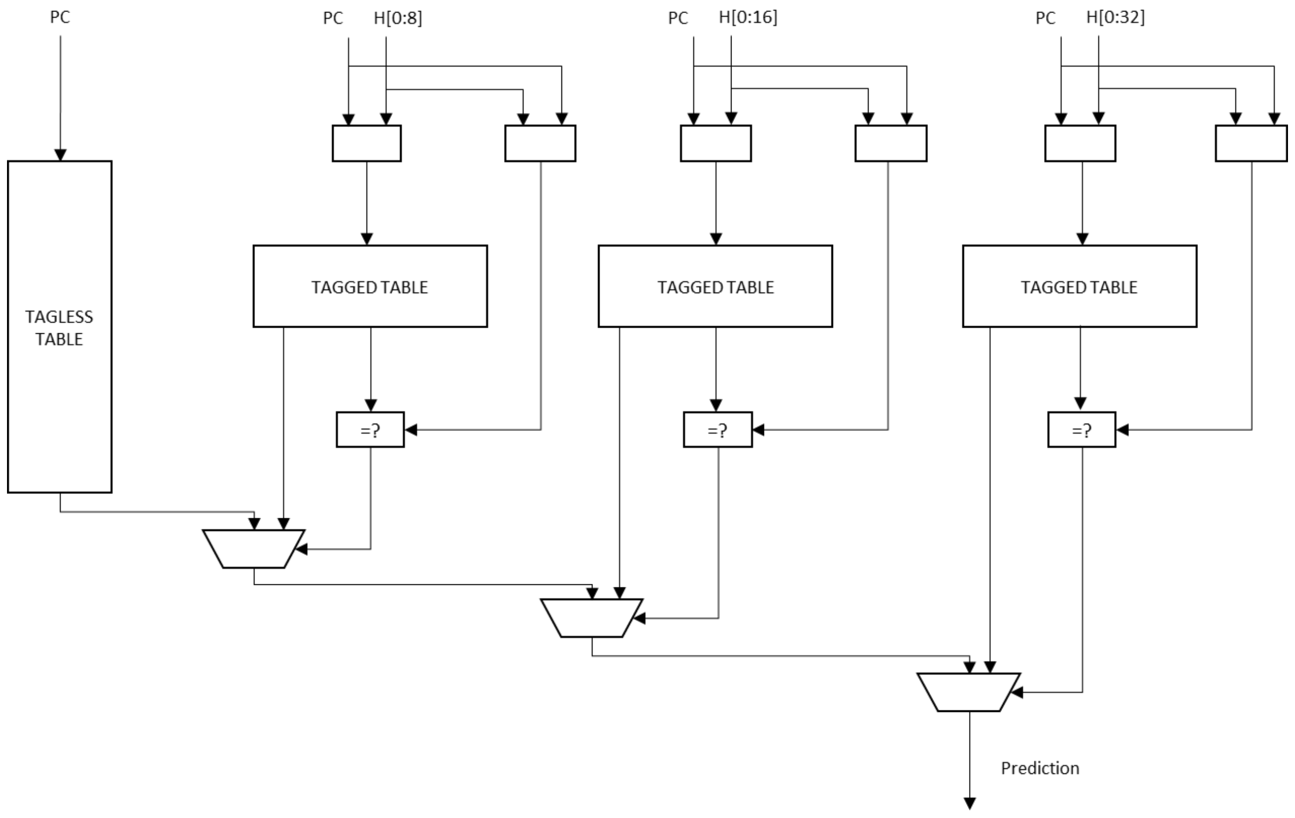


Fig.2 ITTAGE branch predictor model

III. UPDATING ITTAGE

1. In case of the main prediction provides the wrong target, provide new entries in at-most 3 tables in the tables with longer histories. Tables are not chosen continuously. If no table with tag match found, replace the entry in the table just next to the current table.
2. If the main predictor provides a correct target, increase the counter bit and decrease vice versa. If counter bits becomes zero for the wrong prediction, replace the entry with a new entry.
3. If main prediction and alternate prediction both are available, a counter bit of main prediction entry is zero and alternate prediction target is correct but main prediction target is wrong, increase salt. Similarly, decrease the salt if the main prediction is correct and alternate prediction is wrong.
4. If the main prediction is providing the correct target and alternate prediction provides a mispredicted target, make a useful bit of main prediction entry to 1.

IV. UPDATING GLOBAL HISTORY

A 512-bit global history is used for simulation. Global history is updated at the end of an update for both conditional and indirect branches. For conditional branches, the taken or not taken a bit is added to global history bits after shifting it to one bit. For indirect branches, a mixed program counter and target based history are calculated. 10 bits from that history is added to global history table after shifting global history to 10 bits.

V. CALCULATING FOLDING HISTORIES

Folding global history for calculation of tag and index is done by implementing a simple function which takes global history, geometric length of table and length to be folded into. It returns the folded history. Indexes are calculated by bit-wise XOR of folded history by pc[11:0] ^ pc[23:12]. Similarly, tags are calculated by XOR of folding history with LSB bits of the program counter.

VI. SIMULATION CHARACTERISTICS

The submitted simulation of ITTAGE uses one tagless table and 7 tagged table in total. The table characteristics are shown in Table 1.

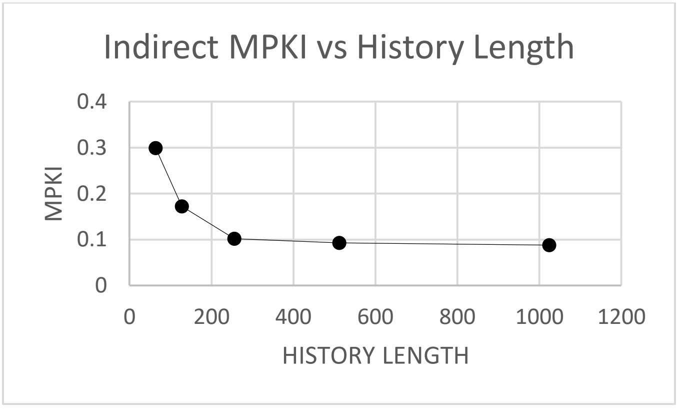
Table 1

Tables	T0	T1	T2	T3	T4	T5	T6	T7	Tot
Entries	4K	4K	4K	4K	4K	4K	4K	4K	32K
Tag	0	12	13	14	15	16	17	18	
U	0	1	1	1	1	1	1	1	
Bits/entry	32	47	48	49	50	51	52	53	
Kbits	128	188	192	196	200	204	208	212	1528

Global history used is of 512 bits. All the tables are indexed directly without the use of any special hash function. The history length of various tables is {0, 8, 16, 32, 64, 128, 256, 512}.

VII. RESULTS

The above indirect branch predictor is tested on the traces provided by the framework. The results are tabulated in Table2. With some traces, it takes little longer time but overall average time per trace comes out to be 23.5 seconds. The overall average indirect branch MPKI comes out to be 0.093. Some observations are stated with different history length and different tagged-table numbers.



The can be seen that a large gradient from history length 64 to 512 bits, but after 512 bit, increasing history length at the cost of hardware is now a good idea.

Table 2

Trace	Indirect MPKI
252.eon.trace.xz	0.031
400.perlbench-41B.trace.xz	0.140
400.perlbench-50B.trace.xz	0.086
403.gcc-16B.trace.xz	0.017
403.gcc-17B.trace.xz	0.005
453.povray-252B.trace.xz	0.014
453.povray-576B.trace.xz	0.008
453.povray-800B.trace.xz	0.007
453.povray-887B.trace.xz	0.002
458.sjeng-1088B.trace.xz	0.061
458.sjeng-283B.trace.xz	0.088
458.sjeng-31B.trace.xz	0.088
458.sjeng-767B.trace.xz	0.084
600.perlbench_s-1273B.trace.xz	0.069
600.perlbench_s-210B.trace.xz	0.070
600.perlbench_s-570B.trace.xz	0.086
602.gcc_s-2375B.trace.xz	0.103
623.xalancbmk_s-325B.trace.xz	0.002
623.xalancbmk_s-592B.trace.xz	0.001
623.xalancbmk_s-700B.trace.xz	0.001
LONG_MOBILE-10.trace.xz	0.214
LONG_MOBILE-14.trace.xz	0.001
LONG_MOBILE-8.trace.xz	0.042
LONG_SERVER-1.trace.xz	0.031
LONG_SERVER-2.trace.xz	0.131
LONG_SERVER-4.trace.xz	0.032
SHORT_MOBILE-20.trace.xz	0.121
SHORT_MOBILE-21.trace.xz	0.108
SHORT_MOBILE-22.trace.xz	0.029
SHORT_MOBILE-30.trace.xz	0.000
SHORT_MOBILE-42.trace.xz	0.002
SHORT_MOBILE-53.trace.xz	0.054
SHORT_MOBILE-61.trace.xz	0.064
SHORT_SERVER-100.trace.xz	0.191
SHORT_SERVER-106.trace.xz	0.027
SHORT_SERVER-108.trace.xz	0.086
SHORT_SERVER-109.trace.xz	0.062
SHORT_SERVER-110.trace.xz	0.112
SHORT_SERVER-111.trace.xz	0.166
SHORT_SERVER-112.trace.xz	0.141
SHORT_SERVER-113.trace.xz	0.206
SHORT_SERVER-114.trace.xz	0.046
SHORT_SERVER-133.trace.xz	0.165
SHORT_SERVER-134.trace.xz	0.003
SHORT_SERVER-135.trace.xz	0.022
SHORT_SERVER-136.trace.xz	0.022
SHORT_SERVER-137.trace.xz	0.105
SHORT_SERVER-1.trace.xz	0.022
SHORT_SERVER-28.trace.xz	0.632
SHORT_SERVER-29.trace.xz	0.003
SHORT_SERVER-2.trace.xz	0.008
SHORT_SERVER-32.trace.xz	0.184
SHORT_SERVER-33.trace.xz	0.168
SHORT_SERVER-34.trace.xz	0.055
SHORT_SERVER-3.trace.xz	0.131
SHORT_SERVER-5.trace.xz	0.076
SHORT_SERVER-86.trace.xz	0.171
SHORT_SERVER-87.trace.xz	0.102
SHORT_SERVER-88.trace.xz	0.044
SHORT_SERVER-89.trace.xz	0.124
SHORT_SERVER-90.trace.xz	0.195
SHORT_SERVER-91.trace.xz	0.189
SHORT_SERVER-92.trace.xz	0.092
SHORT_SERVER-93.trace.xz	0.170
SHORT_SERVER-94.trace.xz	0.103
SHORT_SERVER-96.trace.xz	0.125
SHORT_SERVER-97.trace.xz	0.126
SHORT_SERVER-98.trace.xz	0.263

VIII. CONCLUSION

The main contribution of ITTAGE over other indirect branch predictor is it exploits the best of state of art branch predictors. Simulation results shows that ITTAGE performance is better that O-GHEL and PPM-like tag based predictor. With implementing various optimizations, the accuracy of ITTAGE can be further improved.

IX. ACKNOWLEDGE

I would like to thank Professor **Daniel A. Jiménez** who gave the opportunity to make this project through which I gained valuable knowledge on topic Indirect Branch Predictor. His constant guidance and motivation helped in completing the project successfully.

X. REFERENCE

1. A. Seznec. A 64-Kbytes ITTAGE indirect branch predictor. *JWAC-2: Championship Branch Prediction*, Jun 2011, San Jose, United States. 2011.
2. A. Seznec. Genesis of the O-Gehl branch predictor. *Journal of Instruction Level-Parallelism (http://www.jilp.org/vol7)*, April 2005.
3. P. Michaud. A ppm-like, tag-based predictor. *Journal of Instruction Level-Parallelism (http://www.jilp.org/vol7)*, April 2005.
4. K. Driesen and U. Holzle. The cascaded predictor: Economical and adaptive branch target prediction. In *Proceeding of the 30th Symposium on Microarchitecture*, December 1998.
5. A. Seznec and Pierre Michaud. A case for (partially)-tagged geometric history length predictors. Journal of Instruction Level Parallelism (http://www.jilp.org/vol8), April 2006