ABHIJIT TRIPATHY
atripath@eng.ucsd.edu
PID - A53095669
Branch Predictor Project

Name

Piecewise Linear Branch Predictor (both GHR & past histories ending in this branch)

Description

A Piecewise linear branch predictor is used. We assumed a weight array with number of indices = (Lower bits of PC)×(Past n Branches leading here i.e. Past Global addresses)×GHR length. The weight vectors were assumed to be 7 bits, saturating at 63 and -64 in both directions.

Results

Serial No.	Trace	Branches	Incorrect	1000*wrong_cc_predicts/ total instructions
1	DIST-MM-1	2229289	215970	7.321
2	DIST-MM-2	3809780	264950	8.982
3	DIST-INT-1	4184792	178433	6.049
4	DIST-INT-2	2866495	233513	7.916
5	DIST-SERV-1	3660616	193947	6.575
6	DIST-SERV-2	3537562	204575	6.935
7	DIST-FP-1	2213673	88933	3.015
8	DIST-FP-2	1792835	31763	1.077
Sum				47.87
Arithmetic Mean				5.98375

Complete Description

Linear Piecewise Branch Predictor

The algorithm has been borrowed from ***Daniel A. Jimenez's*** paper titled "***Piecewise linear branch predictor***". The branch history has been considered for all past **n** program paths of length ***h*** (Length of GHR) ending in a branch B (indexed by lower bits of PC). The identities, position and outcomes of all the paths usually correlate highly with the past outcome of the branch. For all these paths **weights** are calculated - which are assumed to be **7bit** values. We have assumed a threshold $\theta = 2.14 * (h + 1) + 20.58$, $where\ h = no\ of\ bits\ in\ GHR$; . A 1 is added to account for the bias entry. The product of the i^{th} weight and corresponding history bits are added and summed into a variable sum which contains the bias value for that pc address. The outcome is then determined by the rule:

$sum \geq \theta$: $Maps\ to\ Hyperplane\ 1\ or\ Branch\ taken\ else\ Branch\ not\ taken.$

Variables Taken :

- **w** – A 3D array of characters with maximum bit usage. If I,j, k are the indices corresponding to each dimension, then :
    - I = which branch is taken. This is pc%n , where n is 16 in my case.
    - J = Last n Past Global addresses who end in this branch. Ranges from 0-17 in my case (n = 17)
    - K = No of Bits in GHR.

      0<I<B where B = 16;
      0<J<n where n = 17
      0<K<n where h = 17

  Total bits = **size of element in weight × n × h × B.**

- **h -** Global history length . This is 17 in my case.
- **GHR** - The global history register. This vector of bits accumulates the outcomes of branches as they are executed. Branch outcomes are shifted into the first position of the array.
- **GA -** An array of past **n** global addresses. As branches are executed, their addresses are shifted into the first position of this array. Taken together, GHR and GA give the path history for the current branch to be predicted. ***The contents of this array are past addresses of 4bits each (0-15).***
- $\boldsymbol{\theta}$ **-** The threshold. It is defined as: - $2.14 \times (h + 1) + 20.58$. If Sum of weights is more or equal to this, branch is predicted taken else not taken.

Cost Analysis :

Serial No.	Table	No. of elements	Size of each element	Total Bits
1	GHR	1	17	17
2	GA	17	4	68
3	W	4624	7	32368
4	Sum	1	32	32
5	Bias	16	7	112
6	Theta	1	7	7
Total Bits				32604

Thus we see that **total bits =32604** which is **less than 32Kbits.**

Reference : http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1431572