논 문
58-3-28

퍼셉트론을 이용하는 다중 분기 예측법

The Multiple Branch Predictor Using Perceptrons

이 종 복†
(Jong-Bok Lee)

Abstract - This paper presents a multiple branch predictor using perceptrons. The key idea is to apply neural networks to the multiple branch predictor. We describe our design and evaluate it with the SPEC 2000 integer benchmarks. Our predictor achieves increased accuracy than the Bi-Mode and the YAGS multiple branch predictor with the same hardware cost.

Key Words : Multiple branch predictor, Perceptron, neural network, Superscalar microprocessor, SPEC

1. 서 론

고성능 수퍼스칼라 마이크로 프로세서의 명령어 대역폭을 증가시키는 방편으로 다중 분기 예측법 또는 트레이스 캐쉬와 같은 방법이 널리 연구되고 있다. 다중 분기 예측법을 적용하면 한 싸이클에 2 개 및 3 개의 명령어 블록을 동시에 인출하여 파이프라인에서 수행시킬 수가 있으므로, 마이크로 프로세서의 성능이 크게 향상될 수 있다 [1-3]. 그러나 분기 예측이 옳지 않음이 판명된다면, 동시에 인출한 여러 개의 명령어 블록을 버려야 하기 때문에 이 방식을 뒷받침하기 위하여 분기 예측의 정확도가 매우 높아야 한다. 이와 관련하여, 최근에 이르러 디지털 시스템에 신경망 알고리즘을 도입하여 학습시키는 방안이 시도되고 있다 [4-6].

본 논문에서는 신경망 분야에서 활용되는 퍼셉트론 방식을 마이크로 프로세서의 다중 분기 예측법에 적용하였다. 이것을 위하여 SPEC 2000 벤치마크 정수형 프로그램을 대상으로 하여 모의실험을 수행하여, 기존의 Bi-Mode와 YAGS를 기반으로하는 다중 분기 예측법과 비교하여 그 정확도의 우수성을 보였다. 특히, 다양한 하드웨어 사양에 대하여 비교 연구를 수행하여 본 논문에서 제안하는 방법이 기존의 방식에 대하여 동일한 하드웨어 비용으로 더욱 높은 정확도를 기록하였다.

2. 배 경

기존의 신경망 분야에서 쓰이는 퍼셉트론 이론에 대하여 알아보고, 이것을 응용한 분기 예측법에 대하여 고찰한다. 또한, 2 단계 적응형 분기 예측기를 확장하여, 수퍼스칼라 마이크로 프로세서에서 명령어 대역폭의 확대를 위하여 널리 이용되는 다중 분기 예측법의 원리를 설명한다.

† 교신저자, 정회원 : 한성대 공대 정보통신공학과 부교수 · 공박
E-mail : jblee@hansung.ac..kr
접수일자 : 2009년 1월 28일
최종완료 : 2009년 2월 20일

2.1 퍼셉트론 분기 예측법

퍼셉트론이란 입력 값들을 가중치와 결합하여 출력을 산출하며, 학습 기능을 갖는 신경망이다. 그림 1은 퍼셉트론의 그래픽 모델이다. 퍼셉트론은 가중치를 요소로 갖는 벡터로 표현되며, 이 때 가중치는 양 또는 음의 정수로 나타낸다.

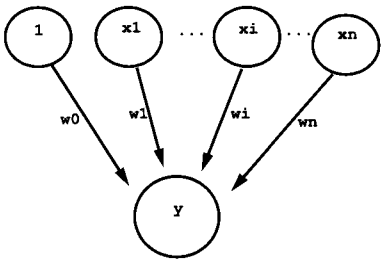

그림 1 퍼셉트론 모델
Fig. 1 The perceptron model

출력은 가중치 벡터 $w_{0..N}$과 입력벡터 $x_{0..N}$의 내적 (dot product)이며, x_0는 항상 1로 설정하여 바이어스 입력으로 공급한다. 따라서, 이전 분기 결과와의 상관 관계를 학습하기 이전에 바이어스 가중치 w_0는 히스토리와 상관없이 항상 그 사건의 바이어스를 학습하게 된다. 퍼셉트론의 출력 y는 수식 1과 같이 표현된다. 퍼셉트론에 대한 입력은 양극성이며 x_i가 -1이면 사건이 일어나지 않은 것이고, 1이면 사건이 일어난 것이다. 이러한 입력에 대한 위 출력 y의 계산결과가 음일 때는 사건이 일어나지 않는 것으로 예측하며, 양일 때는 사건이 일어나는 것으로 예측한다. 추후에 사건의 실

제 결과가 알려지면, 그 결과와 y 값을 가지고 학습 알고리즘을 이용하여 가중치 벡터 P의 각 필드값을 수정한다. 그

$$y = w_o + \sum_{i=1}^{n} x_i w_i$$

수식 1 퍼셉트론의 출력
Eq. 1 The perceptron output

리고 벡터 P를 테이블의 i 번째 항목에 기록하여 그 결과를 반영한다. 사건이 일어나지 않았을 때 t의 값을 -1, 사건이 일어났을 때 t의 값을 1로 설정하고, Θ는 학습이 충분히 이루어졌는가를 판단하는데 이용하는 임계값일 때, 학습 알고리즘을 다음과 같이 표현할 수 있다.

```
if sign(y_out) != t or |y_out| <= Θ then
  for I = 0 to n do
   w_I = w_I + tx_i
  end for
end if
```

t와 x_i의 값이 -1 또는 1이므로, 위 학습 알고리즘에 의하여 사건의 발생 경로가 x_i와 일치할 때 i 번째 가중치가 증가하고, 일치하지 않을 때 감소한다.

퍼셉트론 분기 예측법은 신경망 회로에서 이용하는 이러한 학습 방식을 마이크로 프로세서 구조의 분기 예측기법에 적용한 것이다 [4-6]. 이것을 위하여 길이 N의 분기 히스토리에 최근 N 개의 분기어가 각각 분기한 결과를 1로, 분기하지 않은 결과를 -1로 기록한다. 이렇게 하여 얻은 길이 N인 분기 히스토리 레지스터의 각 필드를, 1 또는 -1로 나타나는 현재 분기어의 결과와 모두 곱한 결과를 각 가중치 필드에 누적하여 더함으로써, 길이 N인 가중치 벡터를 생성한다. 따라서, 만일 임의의 분기 명령어가 계속 분기를 한다면 가중치 벡터의 필드값이 점점 커지고, 분기를 하지 않는다면 그 값이 점점 작아지게 된다.

이와 같이 분기 히스토리 레지스터와 가중치 벡터가 마련된 후에, 분기 명령어에 대한 예측을 수행하고자 할 때, 분기어의 어드레스를 이용하여 0 부터 N-1 사이의 인덱스 i를 생성하여 퍼셉트론 테이블을 접근한다. 이렇게 하여 테이블에서 i번째 퍼셉트론을 인출하여 가중치 벡터 $P_{0..N}$을 얻은 후에, 가중치 벡터와 전역 히스토리 레지스터와의 내적으로 출력 y를 계산하여 그 연산 결과가 양이면 분기할 것으로 예측하고, 음이면 분기하지 않는 것으로 예측한다.

추후에 분기 여부가 실제로 판명되었을 때, 본 가중치 벡터의 각 필드와 현재의 분기 히스토리 레지스터의 각 필드를 곱한 값들을 전부 더하여, 예측에 이용된 위 내적 결과값의 부호와 실제 분기결과의 부호를 비교한다. 두 값의 부호가 일치하지 않거나 내적값이 임계치보다 작을 때는 해당 가중치 벡터의 모든 필드에 현재의 분기 결과와 분기히스토리 레지스터의 각 필드를 곱하여 더함으로써 학습시킨다.

2.2 다중 분기 예측법

기존의 다중 분기 예측법은 2 단계 적응형 분기 예측법을 응용 및 확장한 것이다. 2 단계 적응형 분기 예측법은 분기 히스토리 레지스터에 최근의 분기 결과를 기록하고, 본 레지스터의 값으로 패턴 히스토리 테이블을 접근하여 2 비트 포화 카운터의 값에 따라 분기 여부를 예측하는 방법이다 [3].

다중 분기 예측을 수행하기 위하여, 최근 k 개 분기어의 분기 여부를 1 또는 0으로 기록한 히스토리 레지스터의 k 비트로 패턴 히스토리 테이블을 인덱스하여 첫 번째 분기어를 예측한다. 두 번째 분기어를 예측하기 위하여, 가장 우측의 k-1 분기 히스토리 비트로 패턴 히스토리 테이블을 인덱스한다. 이 때 인덱스를 위하여 k 비트 전부를 사용하지 않으므로, 패턴 히스토리 테이블 내의 두 개의 인접한 항목이 그 대상이 된다. 이 때 첫 번째 분기어를 이용하여 두 개의 항목 중에서 하나를 선택한다. 마찬가지로, 세 번째 분기어는 최우측의 k-2 개의 히스토리 레지스터를 사용하여 패턴 히스토리 테이블 내의 4 개의 인접한 항목을 접근한다. 그리고, 첫 번째와 두 번째 분기어의 예측을 바탕으로 세 번째 분기어에 대한 예측을 수행한다.

일반적으로, M 개의 다중 분기 예측을 수행하기 위하여, 첫번째, 두번째, ..., M 번째 분기 명령어에 대하여, 각각 히스토리 레지스터의 k 비트, k-1 비트, ..., k-M+1 비트로 가중치 벡터 테이블을 각각 인덱스하여 M 개의 분기 명령어에 대한 예측을 수행한다.

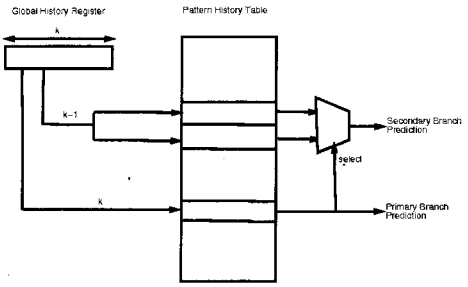

그림 2 다중 분기 예측법
Fig. 2 The multiple branch prediction

3. 퍼셉트론을 이용한 다중 분기 예측법

위에서 살펴본 다중 분기 예측법을 퍼셉트론 분기 예측법에 적용하여 퍼셉트론 방식의 다중 분기 예측법을 만들 수 있다. 그림 3과 같이, 제안하는 퍼셉트론의 다중 분기 예측기는 분기 히스토리 레지스터에 유한한 길이의 분기 행태를 기록하며, 가중치 벡터표를 접근하여 예측을 수행한다. 다중 분기 예측을 위하여, 분기 히스토리 레지스터와 더불어 임시 분기 히스토리 레지스터를 이용하여, 각 다중 분기 예측법의 초기 단계에 분기 히스토리 레지스터의 내용을 그대로 전달받는다.

다중 분기 예측법을 시행하기 위하여, 첫 번째 분기어의 주소를 가중치 벡터표의 항목 수를 이용하여 해쉉한 값으로 특정한 가중치 벡터를 색인하고, 이 값을 분기 히스토리 레지스터와 비트 단위로 곱하여 출력을 계산함으로써 첫 번째 분기어를 예측한다. 두 번째 분기어는 첫 번째 분기어에 대

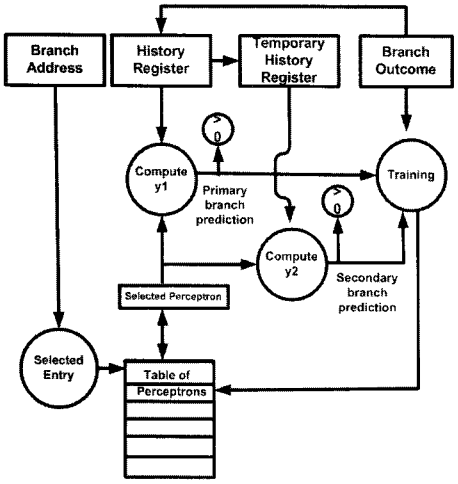

그림 3 퍼셉트론 기반의 다중 분기 예측기, 2차 분기
Fig. 3 The multiple branch predictor using perceptrons

한 예측이 옳다고 가정을 하여 시행하는데, 이것을 위하여 임시 분기 히스토리 레지스터의 최하위 1 비트를 퍼셉트론에 의한 예측값으로 갱신하고 색인된 가중치 벡터와 비트 단위 곱셈을 시행하여 구한다. 이러한 방식으로 한 싸이클에 2 개의 분기어를 동시에 예측할 수 있다. 파이프라인의 실행 단계가 완료하여 분기의 경로가 밝혀지면, 그 결과에 따라서 분기 히스토리 레지스터의 값을 갱신한다. 만일 모든 예측이 옳았다면, 임시 히스토리 레지스터의 내용은 히스토리 레지스터에 덮어쓸 수 있으나, 예측이 틀리면 버려야한다. 이 방식과 유사하게, 세 번째 분기어를 예측하기 위하여, 첫 번째와 두 번째 분기어를 기반으로 하여 임시 분기 히스토리 레지스터의 최하위 2 비트를 갱신하고, 가중치 벡터와 비트 단위 곱셈을 시행한다.

성공적인 다중 분기 예측법을 위하여, 분기 어드레스 캐쉬를 이용한다. 분기 어드레스 캐쉬의 각 항목에는 첫 번째 분기어의 어드레스로 색인되는 첫 번째 명령어 블록의 타겟 어드레스, 두 번째 명령어 블록의 타겟 어드레스, 세 번째 명령어 블록의 타겟 어드레스가 기록되어야 한다.

4. 모의실험 환경

입력 벤치마크 프로그램으로는 표 1에 나타낸 바와 같이 10 개의 SPEC 2000 정수형 프로그램이 사용되었다. 이 프로그램을 SunOS 운영체제하의 Sun Sparc 머신에서 C 컴파일러를 이용하여 실행 가능 화일을 얻었으며 [7], 이 실행 가능 화일과 Shade를 이용하여 Sparc V9 명령어 트레이스를 생성하였다 [8]. 트레이스 구동 모의실험에서 각 프로그램마다 초기 5 천만 개의 명령어를 발생시켜 모의실험에 이용하였다. 표 2는 본 모의실험에 대한 하드웨어 사양을 나타낸 것이다. 본 논문에서의 정확도 비교를 위하여, 기존의 Bi-Mode [9]와 YAGS [10]를 기반으로 하는 다중 분기 예측기를 구현하여 기준으로 삼았다. Bi-Mode 분기 예측기는 선택자 패턴 히스토리 테이블(Choice Pattern Hitstory Table), 미분기 방향 패턴 히스토리 테이블(Direction Pattern History Table Not-Taken), 분기 방향 패턴 히스토리 테이블(Direction Pattern History Table Taken)로 구성되는데, 각 방향 패턴 히스토리 테이블의 항목 수는 선택자 패턴 히스토리 테이블의 절반에 해당된다. 따라서, Bi-Mode 다중 분기 예측기의 하드웨어 비용은 k가 분기 히스토리 레지스터 길이일 때, 2^{k+2}비트로 나타낼 수 있다. 본 모의실험에서, .Bi-Mode 다중 분기 예측기는 분기 히스토리 레지스터의 길이를 12 비트에서 15 비트의 범위로 설정하였다.

표 1 SPEC 2000 정수형 벤치마크 프로그램
Table 1 The SPEC 2000 integer benchmark programs

벤치마크	설명
bzip2	압축
crafty	체스 경기 놀이
gap	그룹 이론 해석기
gcc	C 프로그래밍 언어 컴파일러
mcf	조합 최적화
parser	워드 프로세서
perlbmk	PERL 프로그래밍 언어
twolf	배선 및 배치 모의실험기
vortex	객체지향형 데이터베이스
vpr	FPGA 회로 배치 및 배선

YAGS 분기 예측기는 선택자 패턴 히스토리 테이블, 분기용 캐쉬(Taken Cache), 미분기용 캐쉬(Not-Taken Cache)로 구성된다. 각 캐쉬의 항목 수는 선택자 패턴 히스토리표의 절반에 해당되며, 각 항목은 6 비트의 태그와 2 비트 포화카운터로 구성된다. 분기 히스토리 레지스터의 길이를 k라할 때, 선택자 패턴 히스토리표의 항목 수는 2^k이므로, 하드웨어 비용은 $5\times2^{k+1}$ 비트이다. 따라서, YAGS의 하드웨어 비용은 표 2에 기록된 값보다 다소 높다. 예를 들어, 분기 히스토리 레지스터의 길이가 11 비트일 때, YAGS의 실제 하드웨어 비용은 16 K 비트가 아닌, 20 K 비트이다. 이와 같이 하드웨어 비용이 더 높음에도 불구하고, 본 논문에서 제안하는 퍼셉트론 방식의 다중 분기 예측기의 정확도가 YAGS보다 더 높음을 보일 것이다.

표 2에서 나타낸 것과 같이, 퍼셉트론 방식의 다중 분기 예측기에 대하여, 분기 히스토리 레지스터의 길이는 8 비트에서 64 비트 범위의 다양한 값을 대상으로 하여 모의실험을 수행하였다. 퍼셉트론을 기반으로 하는 예측기의 하드웨어 비용은 k가 분기 히스토리 레지스터의 길이, p가 가중치 벡터표의 항목 수일 때, $2^{k+1}\times p$ 비트에 해당한다. 이 때, 가중치 벡터표의 항목 수는 Bi-Mode 및 YAGS 다중 분기 예측법과 하드웨어 비용이 동일하거나 보다 적도록 설정하였다.

5. 모의실험 결과

그림 5에서부터 그림 12는 Bi-Mode, YAGS 및 본 논문에서 제안하는 퍼셉트론 방식의 다중 분기 예측기에 대하여 동일한 하드웨어 비용을 기준으로 광범위한 모의실험 결과를 비교하여 나타낸 것이다. 각 벤치마크 프로그램에 대하여, 2 차와 3 차의 다중 분기 예측 정확도를 Bi-Mode, YAGS, 퍼셉트론 방식 순으로 막대 그래프를 이용하여 도시하였다. 각 벤치마크마다 세 번째 막대 그래프가 더 높은

것은 본 논문에서 제안하는 퍼셉트론 방식의 예측 정확도가 보다 높다는 것을 의미한다. 그림 5와 6은 하드웨어 비용이 16 K 비트일 때의 결과로서, 퍼셉트론 방식의 경우 분기 히스토리 길이 16 비트이고 가중치 벡터의 항목 수는 1024이다. Bi-Mode와 YAGS 방식의 분기 히스토리 레지스터의 길이는 각각 12 비트와 11 비트이다.

표 2 퍼셉트론 다중 분기 예측기에 대한 하드웨어 사양
Table 2 The hardware configurations for the experiment.

하드웨어 비용[비트]	Bi-Mode [h 비트-p 항목]	YAGS [h 비트-p 항목]	퍼셉트론 방식 [h 비트-p 항목]		
16K	12-4096	11-2048	8-2048	16-1024	32-512
32K	13-8192	12-4096	8-4096	16-2048	32-1024
64K	14-16384	13-8192	8-8192	16-4096	32-2048
128K	15-32768	14-16384	16-8192	32-4096	64-2048

2 차 분기를 시행하였을 때 퍼셉트론 분기 예측법을 Bi-Mode 분기 예측법과 비교하면, vortex에서 최고 4.2 %, mcf에서 최저 0.5 % 더 높은 정확도를 나타내었다. 한편, 퍼셉트론 방식을 YAGS 분기 예측법과 비교하였을 때, 역시 퍼셉트론 방식이 vortex에서 최고 3.0 % 더 높았으나, bzip2에서는 0.1 %에 그쳤다. 3 차 분기를 시행하였을 때는 퍼셉트론 방식이 Bi-Mode 방식보다 vortex에서 최고 4.8 %, mcf에서 최저 0.6 % 더 높은 정확도를 나타냈으며, YAGS 방식보다 vortex에서 3.5 % 우수하였다. 퍼셉트론 방식의 정확도는 2 차와 3 차에서 평균 93.8 %와 93.7 %를 각각 기록하였으며, 이것은 Bi-Mode에 비하여 2 차와 3 차 분기 예측에 대하여 각각 2.0 %와 2.4 % 높으며, YAGS에 비하여 1.5 %와 1.9 % 더 높다.

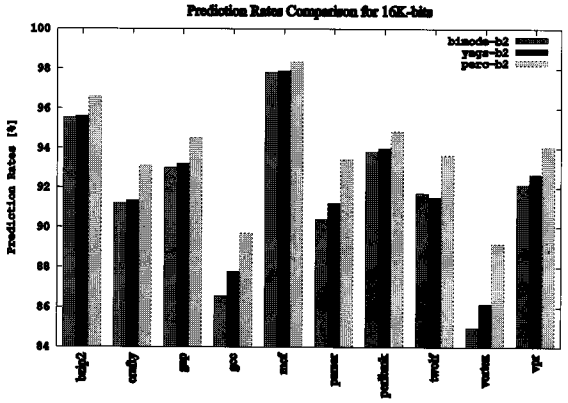

그림 5 하드웨어 16 K 비트, 2 차 분기
Fig. 5 16 K bit hardware budget, 2-branch

그림 7과 8은 하드웨어 비용이 32K 비트일 때의 결과를 동일한 방식으로 나타낸 것이다. 이 때, 퍼셉트론 분기 예측법에서 분기 히스토리 레지스터의 길이는 32 비트, 가중치 벡터표의 항목의 개수는 1024 개이다. Crafty 3 차 분기 예측에서 에서 퍼셉트론 방식이 13 비트의 분기 히스토리 레지스터를 사용한 Bi-Mode 방식에 비하여 최고 4.7 % 높은 정확도를 가져왔다. YAGS 방식은 12 비트의 분기 히스토리 레지스터를 이용하였으며, 이 경우 5.6 % 높은 결과를 기록하였다. 퍼셉트론 방식의 정확도는 2 차와 3 차에서 모두 평균 94.4 %를 기록하였으며, 이것은 Bi-Mode에 비하여 2 차와 3 차 분기 예측에 대하여 각각 2.0 %와 2.4 %, YAGS에 비하여 1.8 %와 2.0 % 더 높다.

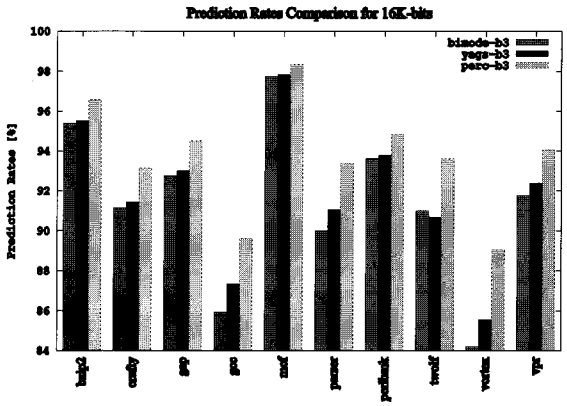

그림 6 하드웨어 16 K 비트, 3 차 분기
Fig. 6 16 K bit hardware budget, 3-branch

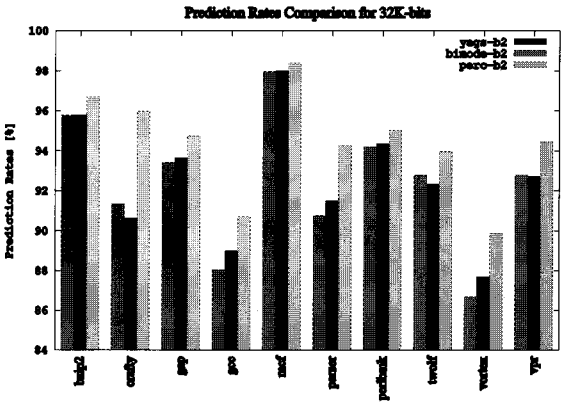

그림 7 하드웨어 32 K 비트, 2 차 분기
Fig. 7 32 K bit hardware budget, 2-branch

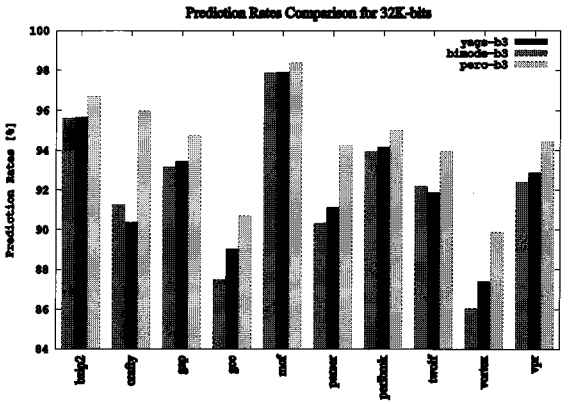

그림 8 하드웨어 32 K 비트, 3 차 분기
Fig. 8 32 K bit hardware budget, 3-branch

그림 9와 10은 하드웨어 비용이 64K 비트일 때의 결과를 나타낸 것이다. 퍼셉트론 방식에서 분기 히스토리 레지스터의 길이는 16 비트이고, 가중치 벡터표의 항목 수는 4096 개이다. Bi-Mode와 YAGS의 분기 히스토리 레지스터의 길이는 각각 14 비트와 13 비트로 증가하였다. 퍼셉트론 방식의 정확도는 2 차와 3 차에서 모두 평균 94.9 %를 기록하였으며, 이것은 Bi-Mode에 비하여 2 차와 3 차 분기 예측에 대하여 각각 2.0 %와 2.3 % 높으며, YAGS에 비하여 1.8 %와 2.0 % 더 높다. Bi-Mode와 YAGS 방식의 증가된 하드웨어 자원으로 인하여 정확도가 상승하여, 퍼셉트론 분기 예측법의 정확도의 증가가 16K 비트에서 32K 비트로 이행하는 경우에 비하여 64 K 비트에서 다소 둔화되었음을 알 수 있다. 마지막으로 그림 8은 128K 비트일 때의 분기 예측 정확도를 도시한 것이다. 퍼셉트론 방식의 경우 분기 히스토리 레지스터의 길이는 32 비트로 확장되었고, 가중치 벡터표의 개수 또한 4096 개로 증가하였다. 이 때, Bi-Mode와 YAGS 방식의 분기 히스토리 레지스터의 길이는 각각 15 비트와 14 비트이다. 퍼셉트론 분기 예측법의 2 차 분기의 평균 예측 정확도는 95.3 %, 3 차의 경우 95.2 %로 최고치를 기록하였으며, 퍼셉트론 방식의 예측기가 Bi-Mode 예측기보다 2 차의 분기에서 평균 2.1 %, 3 차의 분기에서 2.2 % 각각 우수하고, YAGS에 비하여 각각 1.8 %, 1.9 % 우수하다.

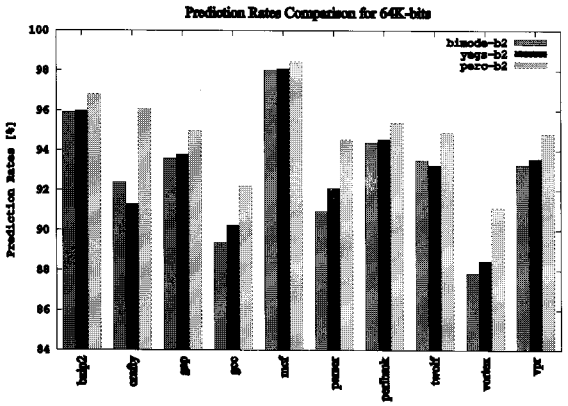

그림 9 하드웨어 64 K 비트, 2 차 분기
Fig. 9 64 K bit hardware budger, 2-branch

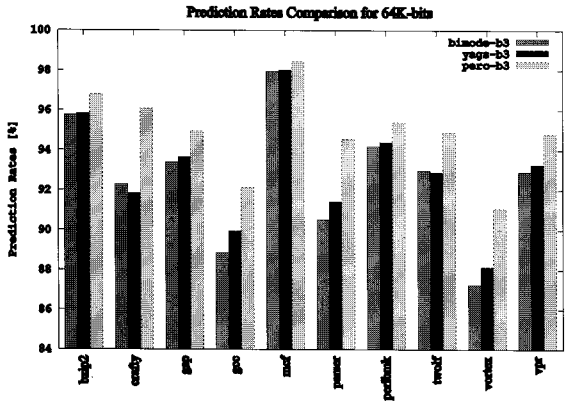

그림 10 하드웨어 64 K 비트, 3 차 분기
Fig. 10 64 K bit hardware budget, 3-branch

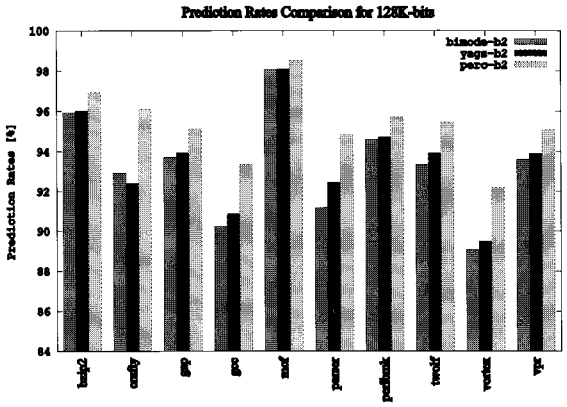

그림 11 하드웨어 128 K 비트, 2 차 분기
Fig. 11 128 K bit hardware budget, 2-branch

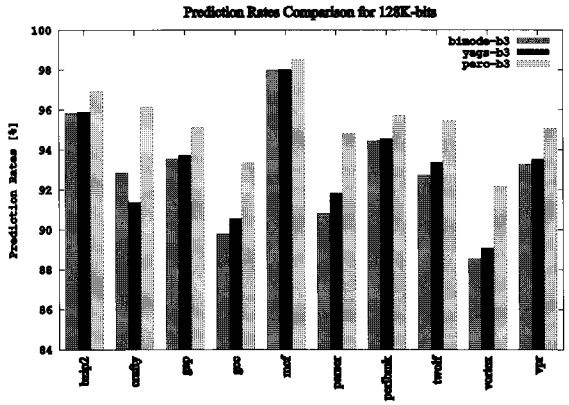

그림 12 하드웨어 128 K 비트, 3 차분기
Fig. 12 128 K bit hardware budget, 3-branch

이상 4 가지 하드웨어 비용에 대한 평균 예측 정확도를 분석하면, 하드웨어 비용이 증가할 수록 분기 예측 정확도가 증가하며, 같은 하드웨어 비용에서도 본 논문에서 제안하는 퍼셉트론 방식의 다중 분기 예측법이 기존의 Bi-Mode 및 YAGS 다중 분기 예측법보다 항상 우수함을 확인할 수 있다. 또한, 한 개의 명령어에 대한 예측을 기반으로 하는 2 차의 분기 예측 정확도가, 두 개의 명령어에 대한 예측을 기반으로 하는 3 차의 분기 예측 정확도보다 공통적으로 보다 높음을 알 수 있다.

그림 13과 그림 14는, 각각 2차와 3차의 분기 예측에 대하여, 하드웨어 비용이 16K, 32K, 64K, 128K 비트이고 분기 히스토리 레지스터의 길이가 16, 32, 64로 점차 증가할 때의 평균 예측 정확도를 함께 나타낸 것이다. 동일한 하드웨어 비용에 대하여 분기 히스토리 레지스터의 길이가 증가할 수록 예측 정확도 역시 향상됨을 알 수 있다. 한편, 2 차의 분기 예측에서 3 차의 분기 예측으로 이행할 때, 본 논문에서 제안하는 퍼셉트론 방식은 높은 정확도를 여전히 유지하지만, 기존의 Bi-Mode 및 YAGS 방식에서는 정확도의 하락폭이 커지기 때문에, 본 논문에서 제안하는 방식이 기존의 방식보다 2 차에 비하여 3 차에서 상대적인 분기 예측 정확도의 값이 더욱 증가함을 알 수 있다.

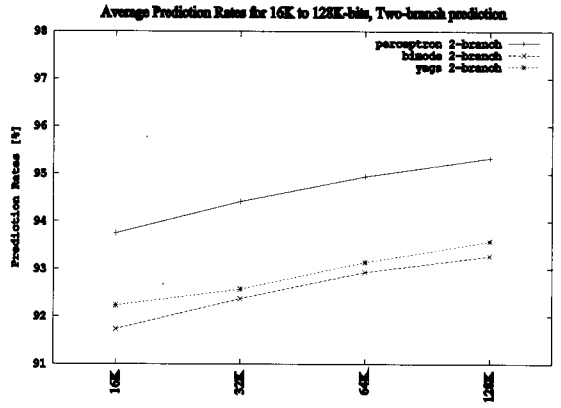

그림 13 여러 비용에 대한 2 차 분기 예측 정확도
Fig. 13 Different hardware budget, 2-branch

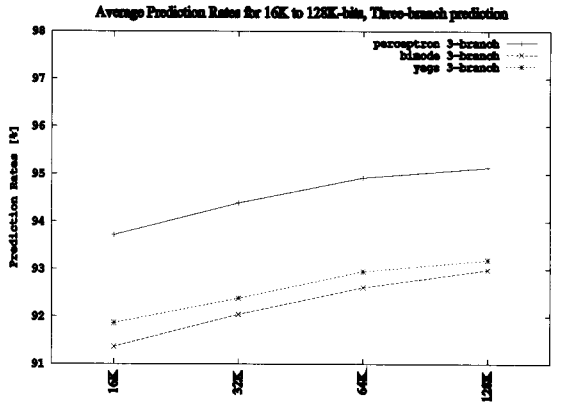

그림 14 여러 비용에 대한 3 차 분기 예측 정확도
Fig. 14 Different hardware budget, 3-branch

6. 결 론

높은 명령어 대역폭이 요구되는 고성능 수퍼스칼라 마이크로 프로세서에서는 보다 정밀한 다중 분기 예측기가 필요하다. 본 논문에서는 퍼셉트론을 적용하여, 기존의 방식보다 우수한 성능의 다중 분기 예측기를 제안하였다. 이것을 위하여 동일한 하드웨어 비용을 기준으로 하여 기존의 Bi-Mode및 YAGS 다중 분기 예측기와 비교하였으며, 하드웨어 비용이 16 K 비트에서 128 K 비트의 범위에 있을 때 2 차와 3 차에서 평균 1.9 %에서 2.4 % 범위의 높은 정확도를 나타내어 그 우수성을 확인 할 수 있었다. 향 후의 연구과제로는, 퍼셉트론 방식의 다중 분기 예측법에 경로를 기반으로 하는 기법을 도입하거나 하이브리드 형태로 변형을 가하여 그 정확도를 좀 더 높이는데 있다.

감사의 글

본 연구는 2007년도 한성대학교 교내연구비 지원과제임.

참 고 문 헌

[1] D. M. Koppelman, "The Benefit of Multiple Branch Prediction on Dynamically Scheduled Systems," Annual International Symposium on Computer Architecture", pp. 42-51, May. 2002,

[2] Ryan Rakvic and Bryan Black and John Paul Shen, "Completion time multiple branch prediction for enhancing trace cache performance", Annual International Symposium on Computer Architecture", pp. 47-58. 2000.

[3] T.-Y. Yeh, D. Marr, Y. Patt, "Increasing the Instruction Fetch Rate via Multiple Branch Prediction and a Branch Address Cache," The 7th International Conference on Supercomputing, pp. 67-76. 1993.

[4] D. A. Jimenez and C. Lin : "Fast Path-Based Neural Branch Prediction" Proceedings of the 36th International Symposium on Microarchitecture,, 2003.

[5] D. A. Jimenez and C. Lin : "Neural Methods for Dynamic Branch Prediction" ACM Transactions on Computer Systems, Vol.20, No.4, pp.369-397, 2002.

[6] D. A. Jimenez and C. Lin, "Dynamic Branch Prediction with Perceptrons," International Symposium on High Performance Computer Architecture, pp. 197-206. 2001.

[7] The SuperSPARC Microprocessor, Technical Paper, Sun Microsystems Computer Corporation, 1992.

[8] Introduction to Shade, Sun Microsystems. Inc, Jun. 1997.

[9] C. Lee, I. K. Chen, and T. Mudge, "The Bi-Mode Branch Predictor," 30th Annual ACM/IEEE International Symposium on Microarchitecture, pp. 4-13, Jun. 1997.

[10] A.N. Eden and T. Mudge, "The YAGS Branch Prediction Scheme," 31st Annual ACM/IEEE International Symposium on Microarchitecture, Jun. 1998.

저 자 소 개

이 종 복 (李 鍾 馥)

1964년 8월 20일생. 1988년 서울대 컴퓨터공학과 졸업. 1998년 동 대학 전기공학부 졸업(공박). 2000년~현재 한성대 정보통신공학과 부교수

Tel : 02-760-4497
Fax : 02-760-4435
E-mail : jblee@hansung.ac.kr