2J-01

マイクロプロセッサアーキテクチャ設計ツール MEIMAT
- キャッシュ機能の初期実装 -

川股　凌太†　　森本　智之‡　　荻田　忠弘†　　王　子銘†　　駱　家輝†　　陳　廷安§　　堤　利幸§

明治大学大学院理工学研究科情報科学専攻†　株式会社ヴィアックス‡　明治大学理工学部情報科学科§

1. 概要

我々の研究室ではマイクロプロセッサアーキテクチャ設計ツール MEIMAT の開発を行っている。マイクロプロセッサの性能評価に用いる MEIMAT シミュレータのキャッシュモジュールでは、今までミスヒットを想定していなかった。今回、キャッシュモジュールの再設計を行い、キャッシュのマッピング方法、ヒット時の処理及びミスヒット時のメインメモリからのデータ転送等について模擬できるように改良した。実際に用いられているキャッシュメカニズムを新たに実装したマイクロプロセッサシステムにおけるキャッシュメモリの動作評価を行い、実験結果からキャッシュメモリを持つマイクロプロセッサにおける性能評価のための枠組み構築ができたことを確認した。

2. 背景

我々の研究室では、任意の命令の実行に最適なマイクロプロセッサの設計ができるようになることを目標としてマイクロプロセッサアーキテクチャ設計ツール MEIMAT の開発を行っている[1][2]。設計したマイクロプロセッサの性能評価を行うシミュレータでは、MEIMAT で用いる汎用命令[1]で記述されたプログラムを実行し[2]、結果を GUI 上で確認することができる[2]。しかし、現在のシミュレータはキャッシュメモリのミスヒットを想定しておらず、キャッシュメモリを持つマイクロプロセッサの性能評価ができないという課題があった。そのため、シミュレータ上にミスヒットを想定したキャッシュを持つマイクロプロセッサの性能を評価するための枠組みを構築する必要があった。

3. 研究目的

ミスヒットを想定したマイクロプロセッサの性能を評価するためには、実際のマイクロプロセッサで使用されているキャッシュメカニズムをシミュレータに実装する必要がある。そこで、今回はキャッシュモジュールの再設計を行い、キャッシュのマッピングやヒット時の処理、ミスヒット時のメインメモリからのデータ転送を模擬できるように改良することを目的とする。

4. キャッシュモジュールの概要

MEIMAT に実装したキャッシュモジュールはダイレクトマップ方式で構成されており、ライトスルー方式で動作する。ダイレクトマップ方式ではキャッシュメモリをブロック分けしてデータを管理する。MEIMAT ではキャッシュサイズと 1 ブロックごとのメモリ容量についてパラメータを変更することができる。ダイレクトマップ方式ではデータを保存するメモリ部分にアクセスする際、一定のルールに基づいてキャッシュメモリにアクセスしているアドレスをビット単位で分割することでそのアドレスのタグと属するブロック番号の情報を得る。その後、対象アドレスのタグとキャッシュ内の該当ブロック番号の現在のタグを照らし合わせる。タグが一致していた場合はヒットとしてキャッシュ内からデータの読み書きを行うが、不一致だった場合はミスヒットとして対象アドレスが含まれるブロックをメインメモリからキャッシュに転送したうえで対象アドレスのデータの読み書きを行う。また、MEIMAT では以下の図 1 のようにデータキャッシュの情報をモジュールビューワを用いて表示することができる。

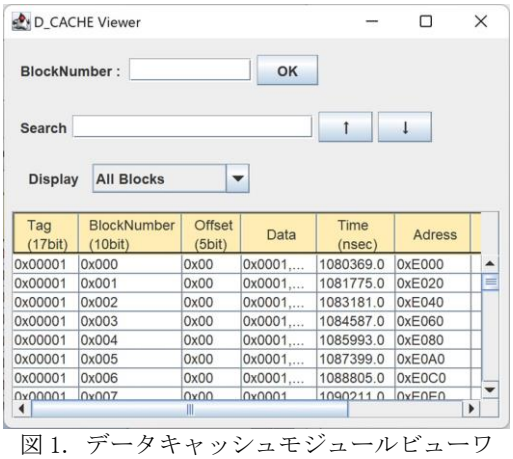

図 1. データキャッシュモジュールビューワ

Microprocessor Architecture Design Tools, MEIMAT - Initial Implementation of Cache Mechanism -
† R. Kawamata, T. Ogita, Z. Wang, L. Jiahui: Meiji University Guraduate School
‡ T. Morimoto: VIAX Co.,Ltd.
§ C. Tingan, T. Tsutsumi: Meiji.University

```
.program     PROG
.text        0x0000
0x0000 ! PROG ! (R0)<=0x0000 @ALU (Rz),@ALU=ADDS,SRset//BL32;          # (R0)=0x0000
0x0004 ! ____ ! (R1)<=0x0000 @ALU (Rz),@ALU=ADDS,SRset//BL32;          # (R1)=0x0000
0x0008 ! ____ ! (R2)<=0x4000 @ALU (Rz),@ALU=ADDS,SRset//BL32;          # (R2)=0x4000
0x000C ! LOOP ! (R0) <= (R0) @ALU Mem((R1)+0x8000),@ALU=ADDS,SRSet//BL32; # (R0)=(R0)+Mem((R1)+0x8000)
0x0010 ! ____ ! (R1) <= SEQUENTIAL(0x0002,0x7FFE)//BL32;               # (R1)=Sequnetial/Random[0x0002,0x7FFE]
0x0014 ! ____ ! (R2)<=(R2) @ALU 0x0001,@ALU=SUBS,SRset//BL16;          # (R2)=(R2)-0x0001
0x0016 ! ____ ! SRgt,(PC)<=0x000C,SRunset//BL32;                       # If(R2)>=0 then (PC)=0x000C
0x001A ! END  ! SRal,(PC)<=0x001A,SRunset//BL32;                       # Program end
.data        0x8000, little
0x8000 ! ____ !  0x0001
0x8002 ! ____ !  0x0001
0x8004 ! ____ !  0x0001
....
0xFFFA ! ____ !  0x0001
0xFFFC ! ____ !  0x0001
0xFFFE ! ____ !  0x0001
```

図 2. MEIMAT アセンブリ言語プログラム(ランダムアクセス)

5. 実験

我々はデータキャッシュの動作を検証するために、データキャッシュ全体の容量が 8 キロバイト、ブロック内の容量が 32 バイト、ブロック数が 256 個のキャッシュモジュールを用意した。キャッシュのアクセス時間は 1.5ns、メモリからの 2 バイトのアクセス時間は 60ns、キャッシュがミスヒットした場合のブロック転送時間は 510ns とした。検証のために、32 キロバイトのデータメモリを昇順に 2 バイトずつシーケンシャルにアクセスするプログラムと、2 バイトずつランダムに同数回アクセスするプログラムを作成した。実験は(1)再設計したキャッシュモジュールでシーケンシャルアクセスした場合、(2)同キャッシュモジュールでランダムアクセスした場合、(3)従来のミスヒットしないキャッシュの場合、(4)メインメモリのみでキャッシュを用いない場合のそれぞれで行う。図 2 はランダムアクセスの検証に用いた MEIMAT アセンブリ言語プログラムが，他の場合もデータアクセス以外は同様のプログラムである。

上記のデータアクセスの実装方法(1)～(4)のそれぞれで、プログラム中のループ回数ごとに実行時刻を計測し，その実験結果を図 3 に示す。(4)はキャッシュが存在しない場合の実行時間である。(3)はキャッシュが全くミスヒットしないと仮定した場合の理想的な実行時間である。再設計したキャッシュでランダムアクセスする(2)の場合はミスヒットが多発し、キャッシュメモリを用いているにもかかわらずメインメモリのみの(4)の場合より実行時間が長くなってしまう。再設計したキャッシュでシーケンシャルアクセスする(1)の場合、キャッシュを用いたランダムアクセスの(2)の場合よりヒット率が向上し、実行時間が大幅に短くなる。また、メインメモリのみを用いる(4)の場合と比べても実行時間が短くなっていることが確認できる。以上より、プログラムの実行時間はキャッシュの実装方法によって大きく変動することがわかる。よって、

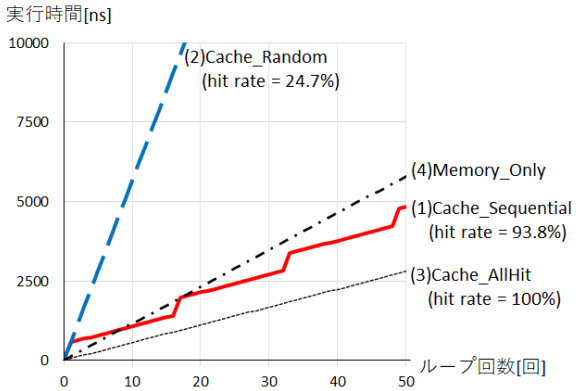

図 3. データアクセスの実装方法別のマイクロプロセッサのプログラム実行時間

マイクロプロセッサの性能評価には，実験で示したようにキャッシュメモリを正確にシミュレーションできることがとても重要となる。

6. 結論

今回、我々は MEIMAT におけるキャッシュモジュールの再設計を行い、実際に用いられているキャッシュメカニズムを実装した。キャッシュメモリの動作評価を行った結果からキャッシュメモリを持つマイクロプロセッサにおける性能評価のための枠組み構築ができたことを確認した。今後はマイクロプロセッサアーキテクチャの性能評価を行うため、セットアソシアティブ方式やライトバック方式の実装が必要になる。

参考文献

[1] 榎本崇，花井北斗，鈴木祐一郎，佐野友輝，森本智之，堤利幸，”マイクロプロセッサアーキテクチャツール MEIMAT の開発 - 汎用命令の提案 -“，FIT2018,C-016(2018).

[2] T. Ogita, T. Morimoto, S. Onogi, R. Kawamata, Z. Wang and T. Tsutsumi, ″Initial Implementation and Utilization of Instruction Dynamic Simulation in MEIMAT,″ *2022 21st International Symposium on Communications and Information Technologies (ISCIT)*, 2022, pp. 154-159.