

コンピュータ科学実験レポート

坪井正太郎 (101830245)

2020 年 12 月 2 日

はじめに

この実験では、一部の命令が実装されていないプロセッサに、適切な命令を実装して、条件付きループ命令を含む動作を行えるようにする。

また、各実験では、シミュレータや論理合成のソフトウェアを使うために、以下の設定を行う。端末を終了した場合、再度 `source` コマンドを実行する。

ソースコード 1 設定の読み込み

HDL のコンパイルには Quartus Prime を、機能レベルシミュレーションには Model Sim を使用した。
バイナリファイルの内容は、`hexdump` コマンドによる。一番左のカラムは、`hexdump` の行数である。

各実験

1 実験 5-1

1.1 実験の目的、概要

この実験では、実験 4-2 で作成したプロセッサで画面上に文字を表示する C プログラムを実行するその中で関数呼び出しを行い、動作を予想し、結果を確認する。

これによって、現在足りていない機能を確認することを目的とする。

1.2 実験方法

以下のプログラムを配置した。

ソースコード 2 my_print.c

1.2.1 クロスコンパイル、メモリイメージファイルの作成

以下の操作でクロスコンパイルし、メモリイメージファイルを作成した。

ソースコード 3 クロスコンパイル、メモリイメージファイルの作成

1.2.2 命令列の確認、動作予想

生成された、rom8x1024.mif を確認して、以下の点について結果を予測した。

- 最初に PC=0x0074 を実行した直後の REG[31] の値
- 最初に PC=0x0074 を実行した直後の PC の値

1.2.3 論理合成、ダウンロード

以下の操作で、論理合成し、FPGA にダウンロードした。

ソースコード 4 論理合成、ダウンロード

クロックを手動モードで送り、70 個ほどの命令を実行、予想した点と、ディスプレイに表示されるはずの文字について確認した。

1.3 実験結果

1.3.1 命令列の確認、動作予想

以下のようなメモリイメージファイルが生成された。

ソースコード 5 rom1024.mif

命令列を確認して、このような予想をたてた。

- 最初に PC=0x0074 を実行した直後の REG[31] の値
 - REG[31]=0x0078
- 最初に PC=0x0074 を実行した直後の PC の値
 - PC=0x00a0

1.3.2 FPGA での実行結果

予想した点について結果は、以下のようになった。

- 最初に PC=0x0074 を実行した直後の REG[31] の値
 - REGWRITED=00000000, WEN=0 であり、レジスタへの書き込みは発生していない
- 最初に PC=0x0074 を実行した直後の PC の値
 - PC=0x0078 であり、ジャンプはしていない

ディスプレイに文字は表示されなかった。

1.4 考察

このプロセッサには、jal 命令が実装されていないので、プログラムが正しく動作しなかった。jal 命令が正しく実装されている場合、31 番目のレジスタに次の戻り先プログラムカウンタの位置が退避されて、プログラムカウンタの位置が命令で指定された場所に更新されるはずである。

2 実験 5-2

2.1 実験の目的、概要

実験 4-2 で作成したプロセッサに jal 命令が足りないことが、実験 5-1 で確認できた。本実験では、プロセッサに jal 命令を追加実装し、動作を確認する。その際、実験 5-1 での予想と実際を比較する。

これによって、関数呼び出し時のプロセッサの動作、レジスタに保存されるデータなどを確認することを目指すとする。

2.2 実験方法

2.3 追加設計

main_ctrl.v に以下の変更を加え、jal 命令のオペコードと、実行されるとき制御信号を定義した。

ソースコード 6 main_ctrl.v の追加設計

2.3.1 論理合成、ダウンロード

以下の操作で、論理合成し、FPGA にダウンロードした。

ソースコード 7 論理合成、ダウンロード

実験 5-1 で予想した点と、ディスプレイに表示されるはずの文字について確認した。

2.4 実験結果

予想した点について結果は、以下のようになった。

- 最初に PC=0x0074 を実行した直後の REG[31] の値
 - REGWRITED=00400078,index=0x1f(=31),write_enable=1 が読み取れた
 - 予想通り、次に実行する PC の値が、REG[31]
- 最初に PC=0x0074 を実行した直後の PC の値
 - 予想通り PC=0x=0x00a0 であり、ジャンプしている

画面上には、"HELLO!!" という文字列が表示された。

2.5 考察

jal 命令を実装したことで、\$ra に PC の値が退避され、即値で指定した値にジャンプできるようになった。そのため、予想した点について正しい動作を確認することができた。これは、my_print 関数の呼び出しに成功しているということであると考えられる。実際に、画面上に文字列が表示されることが分かった。

この実験で、関数呼び出し時のプロセッサの動作、レジスタに保存されるデータを確認する事ができた。

3 実験 6-1

3.1 実験の目的、概要

実験 5-2 で jal 命令を追加したプロセッサ上で、キーボードからの入力を受け取る C プログラムを実行する。その中で関数からの復帰を行い、それらの動作を予想し、結果を確認する。

これによって、現在足りていない機能を確認することを目的とする。

3.2 実験方法

3.3 実験結果

3.4 考察

4 実験 6-2

4.1 実験の目的、概要

実験 5-2 で作成したプロセッサに jr 命令が足りないことが、実験 6-1 で確認できた。本実験では、プロセッサに jr 命令を追加実装し、動作を確認する。その際、実験 6-1 での予想と実際を比較する。

これによって、関数からの復帰時のプロセッサの動作、レジスタから読み取れるデータなどを確認することを目的とする。

4.2 実験方法

4.3 追加設計

cpu.v に、以下の変更を加え、jpr_sel モジュールを追加した。

ソースコード 8 cpu.v の追加設計

main_ctrl.v に以下の変更を加え、jr 命令が実行されときの制御信号を定義した。

ソースコード 9 main_ctrl.v の追加設計

4.4 実験結果

4.5 考察

5 実験 7

5.1 実験の目的、概要

本実験では、実験 6-2 で作成したプロセッサ上で、キーボード入力に応答を行うプログラムを動作させる。このプログラムは、プロセッサに実装されていない命令を使用するため、正常に動作しない。

本実験では、このプログラムを、正常に動作させることを目的とする。

5.2 実験方法

5.3 実験結果

5.4 考察

6 実験 8

6.1 実験の目的、概要

本実験では、プロセッサからステッピングモータを扱う。最終的に、キーボードからモータを制御するプログラムを作成、実行する。

これによって、このプロセッサで提供されている機能のを使用することで実現できるプログラムを作成することを目的とする。

6.2 実験方法

6.3 実験結果

6.4 考察