

計算機科学実験及演習3 ハードウェア 実験レポート
機能設計仕様書

情報学科 計算機科学コース 3回生 山田瑛平
学籍番号：1029282731

平成30年6月10日

機能設計仕様

ここでは、中間報告時点から新たに追加されたモジュール、あるいは中間報告時点から変更があったモジュールについて、機能設計仕様を述べる。

制御部

外部仕様 各モジュールを適切なフェーズごとに活性化させるためのクロック信号を送出する。

クロック信号、リセット信号、exec 信号を入力にとり、clp1、clp2、clp3、clp4、clp5 を出力する。具体的には、入力のクロックが 0 から 1 に変化する際、clp(クロックが 0 から 1 へと変化した累計回数%5) のみが 0 から 1 へと変化するような信号を出力している。

内部仕様 モジュール control.v は、以下のようなアルゴリズムで表現されている。

1. 3 ビット値 count を用意する。
2. クロックが 0 から 1 へ変化するたびに、count の値を、count+1 を 4 で割った余りの値に更新する。
3. 2 と同時に、clp[count] の値を 1、それ以外の clp を 0 に設定し、信号を出力する。ただし、clp5 は clp1 と同じ信号を出力する。

clp5 は clp1 と同じ信号を出力するため、clp5 の存在は冗長であり、clp1 に置き換えても良い。(この信号を clp1 に置き換えていないのは、拡張性を維持するためである。)

マルチプレクサ・ALU の制御

外部仕様 命令の 16 ビットの値および条件コードを入力にとり、マルチプレクサの制御に用いる 1 ビット値 7 つと、ls、regWrite、memWrite 4 ビット値 alu を出力する。ls はロードストア命令であるかどうかを、regWrite および memWrite はレジスタ・メモリに書き込むかどうかを 1 ビット値として出力する。4 ビット値 alu は alu のオペコードを出力する。

内部仕様 出力の値はいずれも命令に依存する。以下の表において各命令ごとの制御部の出力を示す。

命令	mux1	mux2	mux3	mux4	mux5	mux6	mux7	ls	regWrite	memWrite	alu
演算命令 ADD~MOV	0	0	0	0	0	0	0	0	1	0	ALU のオペコード
CMP	0	0	0	0	0	0	0	0	0	0	0101
即値加算	0	0	0	0	0	1	0	0	1	0	0000
シフトを用いる演算命令	1	0	0	1	0	0	0	0	1	0	0000
入力	1	0	0	0	0	0	1	0	1	0	0000
出力	1	0	0	0	0	0	0	0	1	0	0000
停止	1	0	0	0	0	0	0	0	1	0	0000
ロード	1	0	1	0	0	0	0	1	1	0	0000
ストア	1	0	1	0	0	0	0	1	0	1	0000
即値ロード	1	0	0	0	1	0	0	0	1	0	0101
無条件分岐	1	1	0	0	0	0	0	0	0	0	0000
条件分岐	1	分岐なら 1	0	0	0	0	0	0	0	0	0000

マルチプレクサ・ALU の制御を担当するモジュールにおいては、このように命令の種類によって出力をそれぞれ変化させている。

レジスタファイル周辺

汎用レジスタ

外部仕様 入力として与えられるのは、クロック、リセット、命令中のレジスタのアドレス ra および rb、レジスタの書き込みの際に入力が与えられる 16 ビット値 writeData、レジスタに書き込むかどうかを判定する regWrite、ロード・ストア命令であるかどうかを判定する ls である。また、レジスタの内容 dataA、dataB を出力する。

内部仕様 レジスタの内部には 8 つの 16 ビット値 r0,r1,...,r8 が常に格納されている。レジスタが活性化されると、dataA、dataB には ra、rb に対応するレジスタの値が出力される。これらの値の更新条件は、以下の通りである。

- regWrite= 1 かつ ls= 0 のとき、rb の指すレジスタに writeData が格納される。
- regWrite= 1 かつ ls = 1 のとき、ra の指すレジスタに writeData が格納される。
- regWrite= 0 のとき、レジスタの値は更新されない。

また、reset に 0 が入力された状態でクロックが 0 から 1 に変化した際には、レジスタの値がすべて 0 に初期化される。汎用レジスタの読み出しはフェーズ p2、書き込みはフェーズ p5(p1 と同時) において行われる。

その他のレジスタ

外部仕様 中間報告以降、新たに 4 ビットレジスタを 2 つ、1 ビットレジスタを 1 つ追加した。いずれのレジスタも役割はデータの保持のみである。4 ビットレジスタは ALU の出力した条件コードを制御部に送るのに 1 つ、オペコードを制御部から ALU に送るのに 1 つ使われている。1 ビットレジスタはマルチプレクサのセレクトを必要なタイミングで送出するのに用いられている。

内部仕様 クロックが 0 から 1 に変化したタイミングで、入力した値をそのまま出力する。

その他のモジュール

7SEG 表示モジュール

外部仕様 3 ビット値を受け取り、8 ビット値を返す。8 ビット値は入力の 3 ビット値を 16 進数に変換した際の 7SEG LED の出力である。出力命令の際に用いられるモジュールであり、2 進法 16 ビット値を 4 桁の 16 進数に変換するのに使用される。

内部仕様 入力に対する出力の対応を以下の表に示す。

入力	出力	入力	出力
0	11111100	8	11111110
1	01100000	9	11110110
2	11011010	10	11101110
3	11110010	11	00111110
4	01100110	12	00011010
5	10110110	13	01111010
6	10111110	14	10011110
7	11100000	15	10001110

出力モジュール

外部仕様 出力モジュール outregister.v は 16 ビット値 rs、クロック信号 clock および 2 ビット値 cal、4 ビット値 op を入力とし、4 ビット値 o1、o2、o3、o4 およびセクタ信号 select を出力する。命令が出力命令であるときにセクタに 1 を、o1 から o4 には 16 ビット値を 4 ビットずつ分割した値を出力する。

内部仕様 クロックが 0 から 1 へ変化した際に外部仕様通りの出力をするモジュールである。すなわち、cal が 11 かつ op が 1101 であるとき、セクタを 1 にセットし、o1 に rs の 15 から 12 ビットを、o2 に 11 から 8 ビットを、o3 に 7 から 4 ビットを、o4 に 3 から 0 ビットを割り当てる。

停止モジュール

外部仕様 停止モジュール haltjudge はクロック信号と 2 ビット値 calc、4 ビット値 op を入力とし、1 ビット値 cl を出力する。停止命令が呼び出された際に cl を常に 0 にセットすることで、停止命令以後プロセッサの動作を停止させるモジュールである。

内部仕様 calc が 11 かつ op が 1111 である場合に 0 を、そうでない場合にクロック信号をそのまま出力する。

モジュールごとの性能評価

今回の実験で作成したプロセッサの全体での論理素子数は 1528 である。最も多くの論理素子を使用しているモジュールは汎用レジスタであり、570 の論理素子を使用している。汎用レジスタ内でのクリティカルパスは r1[15] から dataB[15] へのパスであり、遅延時間は最大で 5.940ns であった。汎用レジスタでの遅延時間が 1 周期の大部分を占めていることが窺える。

考察

論理素子数に着目すると、やはり汎用レジスタに用いられる素子数の多さが目立つ。今回の実験で設計したプロセッサの汎用レジスタで保持できる語数は 8 語であったが、保持できる語数が少ないと感じられる場面も多々あった。保持できるデータの数が増えれば増えるほど素子

数や遅延が大きくなるため、そのバランスをとり、適切な語数の汎用レジスタを設計することが肝要だと考えられる。

制御部の設計において特に難しく感じられたのがクロックの分周であった。今回の実装では、制御モジュールによってクロックを分周し、それぞれの信号を各モジュールに送出した。手法としては他に、分周を各モジュールで管理する方法が考えられる。この場合、フェーズカウンタをすべてのモジュールが格納しなければならず、ハードのコストの増加が見込まれる。しかし、分周モジュールによる分周パルスを用いる際、単純に否定素子を接続するだけでは逆エッジを利用することができないため、negcontrol モジュールを製作する必要がある。negcontrol の素子数は5と非常に少なく、各モジュールにカウンタを用意するよりもハード量は抑えられていると考えられる。

感想

本実験の開始前に学習していたプロセッサに関する知識は全て理論の上に存在していたため、理解こそできて実際にプロセッサがどう動くのかについて本質的な理解をしていなかったように思う。そういう意味では今回の実験はプロセッサに真摯に向き合う初めての機会だと言える。実装中には、理論を学習している際には気付かなかったこと、例えばパイプライン化に立ちはだかる幾多もの問題の解決が如何に難しいかであったり、ハードウェアの挙動を制御することの難しさなどを経験や知識として得ることが出来た。ハードウェアの実験中は苦痛な時間も長かったが、この実験を通過点として精進していきたいと思う。