

B5 CPUを作る

目的・概要

「論理回路基礎」「コンピュータハードウェア」で得た知識をじっさいの CPU の設計に活かそう。まず、B5-1 では、命令セットが定義されている CPU の設計を要素設計からボトムアップに行っていく。ここでは、CAD ツールを各自の PC に実装し、この使い方を学びながら、レジスタファイル、ALU、シーケンサ、メモリなどの設計・シミュレーションを行う。B5-2 では、設計した要素を統合することで CPU を全体設計し、動作確認を行い、これを FPGA 評価ボード上に回路をダウンロードして CPU を動作させる。最後に、CPU の改良設計を行い、FPGA 上で動作確認する。

B5-1 基本設計篇

1 解説

1.1 概要

PC 上で動く CAD ツールをダウンロードし、ライセンス認証を行う。ツールの使い方と設計フローを理解する。レジスタ、レジスタファイル、ALU、シーケンサなどの設計を行い、シミュレータにテストパターンを入力することで個々の動作を検証する。

1.2 CAD ツールのダウンロード

[1]のサイトから、各人の PC に Quartus II をダウンロードする。Quartus II は、デザイン入力、論理合成、シミュレーション、タイミング解析などの機能をもつ CAD ツールであり、基本部は Altera 社から無償で提供される。

Quartus II は、普通の Windows PC で動作する。ただし、諸君のマシン環境に特有の現象に対してアドバイスすることには限界があることに了解願いたい。

1.3 CAD ツールが何であるか知る

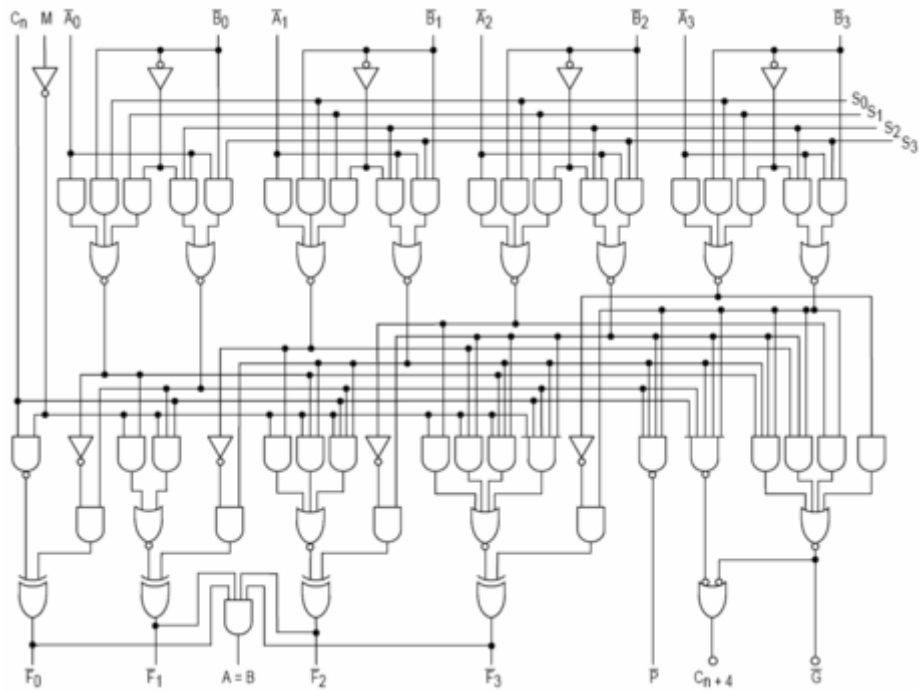
まず、Quartus II 付属のドキュメントである Introduction to Quartus II を読む、Tutorial を実践するなどして、CAD ツールの概要を知る。

1.4 HDL(Hardware Description Language)が何であるか知る

ハードウェア記述言語 HDL を理解する。ここでは、Verilog-HDL (以下、単に HDL) を用いることにする。

1.5 簡単な論理回路の設計

- Quartus II を使って、74181 型 ALU (4 ビット) (図 1) を設計し、シミュレータによって動作を検証する。スキマティックエディタを用いてもよいし、プログラマ的な記述をしてもよいが、ゲートレベルの記述を行うこと。
- HDL (ライブラリ) を使って 32 ビットレジスタを設計し、シミュレータによって動作を確認する。



制御信号 S ₃ S ₂ S ₁ S ₀	M=1: 論理演算	M=0: 算術演算	
		$\overline{C_{in}}=0$	$\overline{C_{in}}=1$
0 0 0 0	$F=\overline{A}$	$F=A$	F=A PLUS 1
0 0 0 1	$F=\overline{A+B}$	$F=A+B$	$F=(A+B) PLUS 1$
0 0 1 0	$F=\overline{A \cdot B}$	$F=A + \overline{B}$	$F=(A + \overline{B}) PLUS 1$
0 0 1 1	$F=0$	$F=1111$	F=ZERO
0 1 0 0	$F=\overline{A \cdot B}$	$F=A PLUS A \cdot \overline{B}$	$F=A PLUS A \cdot \overline{B} PLUS 1$
0 1 0 1	$F=\overline{B}$	$F=(A+B) PLUS A \cdot \overline{B}$	$F=(A+B) PLUS A \cdot \overline{B} PLUS 1$
0 1 1 0	$F=A \oplus B$	F=A MINUS B MINUS 1	F=A MINUS B
0 1 1 1	$F=A \cdot \overline{B}$	$F=A \cdot \overline{B} MINUS 1$	$F=A \cdot \overline{B}$
1 0 0 0	$F=\overline{A+B}$	$F=A PLUS A \cdot B$	$F=A PLUS A \cdot B PLUS 1$
1 0 0 1	$F=A \oplus B$	F=A PLUS B	F= A PLUS B PLUS 1
1 0 1 0	$F=B$	$F=(A + \overline{B}) PLUS AB$	$F=(A + \overline{B}) PLUS A \cdot B PLUS 1$
1 0 1 1	$F=A \cdot B$	$F=A \cdot B MINUS 1$	$F=A \cdot B$
1 1 0 0	$F=1$	$F=A PLUS A$	$F=A PLUS A PLUS 1$
1 1 0 1	$F=A + \overline{B}$	$F=(A+B) PLUS A$	$F=(A+B) PLUS A PLUS 1$
1 1 1 0	$F=A + B$	$F=(A + \overline{B}) PLUS A$	$F=(A + \overline{B}) PLUS A PLUS 1$
1 1 1 1	$F=A$	F=A MINUS 1	$F=A$

図 1. 74181 型 4 ビット ALU

1.6 RISC アーキテクチャの基本

命令セットを実現するためのアーキテクチャの基本は次の図2のようなものである。

1.7 RISC 命令セットの理解

参考文献[2]などを読んで、RISC 命令セットが何であるか理解する。算術論理演算命令、データ移動命令、分岐命令の書式と動作がわかればよい。

1.8 アセンブラの作成

図3で示す命令形式、表1で示す命令セットをもつ RISC 型コンピュータについて、アセンブラを製作する(「コンピュータハードウェア」のレポート課題をやった人は流用可)。

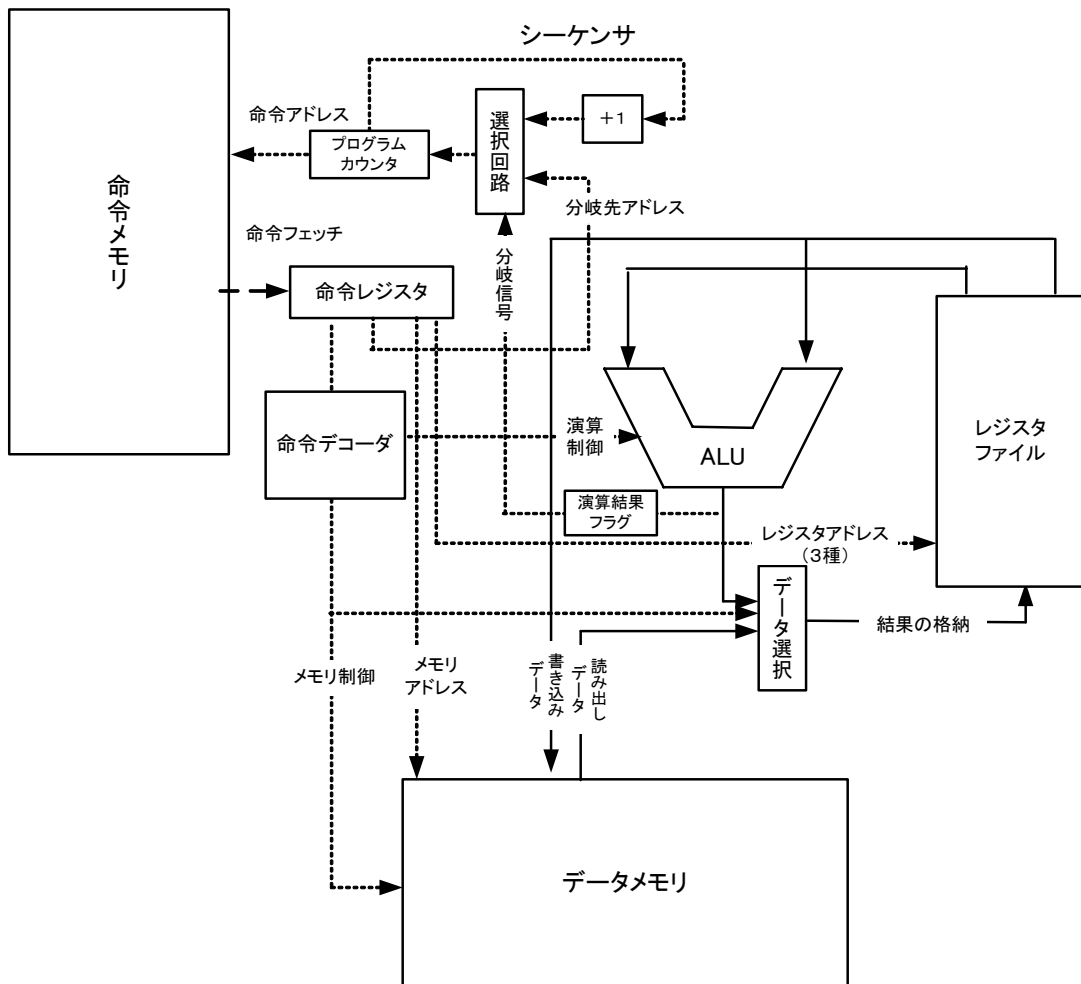
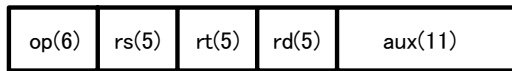
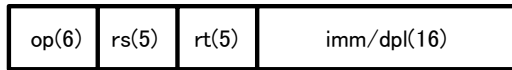


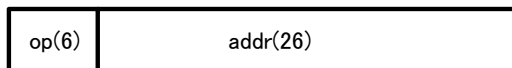
図2 RISC アーキテクチャの基本形



(1) R型



(2) I型



(3) A型

()内がフィールド長(ビット数)

図 3. 命令書式

表 1. 命令セット

ニューモニック	命令形式	OP	AUX等	動作	補注
add	R	0	0	rd ← rs + rt	
addi	I	1	imm	rt ← rs + imm	
sub	R	0	2	rd ← rs - rt	
lui	I	3	imm	rt ← imm << 16	上位16ビットをセット
and	R	0	8	rd ← rs AND rt	ビットごとのAND
andi	I	4	imm	rt ← rs AND imm	即値とのAND
or	R	0	9	rd ← rs OR rt	
ori	I	5	imm	rt ← rs OR imm	
xor	R	0	10	rd ← rs XOR rt	
xori	I	6	imm	rt ← rs XOR imm	
nor	R	0	11	rd ← rs NOR rt	
sll	R	0	16	rd ← rs lshift aux[10,6]	論理左シフト
srl	R	0	17	rd ← rs rshift aux[10,6]	論理右シフト
sra	R	0	18	rd ← rs arshift aux[10,6]	算術右シフト
lw	I	16	dpl	rt ← mem(rs+dpl)	一語読み出し
lh	I	18	dpl	rt ← mem(rs+dpl)	半語読み出し、符号拡張
lb	I	20	dpl	rt ← mem(rs+dpl)	バイト読み出し、符号拡張
sw	I	24	dpl	mem(rs+dpl) ← rt	一語書き込み
sh	I	26	dpl	mem(rs+dpl) ← rt	半語書き込み、符号拡張
sb	I	28	dpl	mem(rs+dpl) ← rt	バイト書き込み、符号拡張
beq	I	32	dpl	if rs == rt then PC = PC+dpl	等しければジャンプ
bne	I	33	dpl	if rs != rt then PC = PC+dpl	等しくなければジャンプ
blt	I	34	dpl	if rs < rt then PC = PC+dpl	未満であればジャンプ
ble	I	35	dpl	if rs <= rt then PC = PC+dpl	以下であればジャンプ
j	A	40	addr	PC ← addr	絶対番地のジャンプ
jal	A	41	addr	R31 ← PC+4; PC ← addr	ジャンプアンドリンク
jr	R	42		PC ← rs	レジスタ間接ジャンプ

2 実験課題 基本要素の設計

- 2.1 Quartus II を使って、74181 型 ALU (4 ビット) (図 1) を設計し、シミュレータによって動作を確認せよ。スキマティックエディタを用いてもよいし、プログラマ的な記述をしてもよいが、ゲートレベルの記述を行うこと。なお、入力信号はすべてのパターンを尽くす必要はなく、代表的なものいくつかだけでよい (以下同様)。
- 2.2 Quartus、HDL を使って 32 ビットレジスタを設計し、シミュレータによって動作を確認せよ。なお、レジスタに用いるライブラリは別途与えることとする。
- 2.3 HDL を使って 32 ビット ALU を設計し、シミュレータによって動作を確認せよ。ALU は、表 1 の命令 (加算、減算、AND、OR、XOR、NOR) を実行できるように設計せよ。
- 2.4 表 1 の命令セットを実現するためのプログラムカウンタとその周辺回路 (シーケンサ) を設計し、シミュレータによって動作を確認せよ。特に分岐命令をどう実現するか、考えること。
- 2.5 図 3、表 1 の命令セットとアセンブリ言語の書式が与えられたときに、アセンブラ (アセンブリ言語を機械語に翻訳するプログラム) を作成せよ。これは「コンピュータハードウェア」の授業で出したレポート課題なので、レポートを転用してよい (他の人のレポートを参考にしてよいが、変換規則をきちんと理解しておくこと)。
- 2.6 命令メモリ、データメモリを設計せよ。ライブラリは別途与えることとする。命令メモリの読み出し、データメモリの読み書きについて、シミュレータによって動作を確認せよ。

3 レポート提出

2 の各課題について、HDL のリスト、シミュレータ動作を証拠づけるもの (波形の図など) を添付し、工夫したこと、結果について検討したことなどを記せ。

B5-2 CPU 設計・製作篇

1 解説

1.1 概要

B5-1 で基本設計した要素を組み合わせて CPU を設計し、シミュレーションによって動作確認した後、ピンアサインなどを決めて、FPGA 評価ボード上に実装する。ボード上での動作を確認した後、次に改良に取り組む。改良点としては、パイプライン化、命令セットの拡張、分岐予測の導入などがあり、各自の力量に応じて、設計・シミュレーション・実装・動作確認を行う。

1.2 CPU の全体設計・シミュレーション

B5-1 で設計した要素を組み合わせて、RISC 型 CPU を設計する。シミュレーションによって命令基本動作を検証する。次に、例題プログラムをシミュレータ上で稼働させる。
(ここにかかなりの時間を使うことが予想される)

1.3 FPGA 評価ボードの理解

三菱マイコン社製 FPGA ボードに触れて、これを理解する。特に PC との接続、FPGA のピン配置、LED の利用法について等。

2 実験課題 CPU の実装

- 2.1 B5-1 で設計した要素を組み合わせて、RISC 型 CPU を設計せよ。シミュレーションによって命令動作を確認せよ。動作検証はすべてを網羅する必要はないが、加算、メモリの読み書き、分岐など、基本的な命令について、それぞれ 1 つずつは行うこと。
- 2.2 下記の 2 種類の階和プログラム($S_n = \sum i$)をそれぞれアセンブラプログラムで表し、これを機械語に翻訳した後に、2.1 の CPU をシミュレートすることで動作させ、動作を確認せよ。

プログラム 1. ループ型

```
sum1(n)
int n;
{
    int i, s;
    for (i=0, s=0; i <= n; i++) s = s + i;
    return(s);
}
```

プログラム 2. 再帰型

```
sum2(n)
int n;
{
    if (n == 0) then return(0);
    else return(n+sum2(n-1));
}
```

- 2.3 B5-1 で設計した CPU について、ピンアサインなどを決めて、これを FPGA ボード上に実装せよ。
- 2.4 2.1 で実装した CPU で、2.2 の 2 種類の階和プログラムをそれぞれ動作させて、動作を確認せよ。
- 2.5 以上で設計した CPU を改良せよ。改良内容は自由とするが、以下のものを指針としてあげる（最低一つ、時間に余裕があればすべてやってもよい）。評価ボード上で実装し、動作確認を行って結果を記せ。
- (1) 新しい命令の追加
 - (2) パイプライン化。パイプラインは、F, D, E, W の 4 段など。
 - (3) キャッシュの導入（ダイレクトマップなど）

3 レポート提出

2.1 については、HDL のソースリストとシミュレーション結果を添付せよ。2.2, 2.4, 2.5 については、動作確認した証拠となるもの（状態と結果の出力など）を示せ。2.2 については（2.5 も必要に応じて）、アセンブラのプログラムリストとアSEMBル結果（バイナリコード）も添付すること。

参考文献

- [1] <http://www.altera.co.jp/products/software/products/quartus2web/sof-quarwebmain.html>
- [2] 坂井修一、コンピュータアーキテクチャ、コロナ社。
- [3] 小林 優、入門 Verilog HDL 記述、CQ 出版 など、Verilog HDL の教科書やチュートリアル