

# 情報科学実験C 期末レポート

氏名 山久保孝亮  
所属 大阪大学基礎工学部情報科学科ソフトウェア科学コース  
メールアドレス u327468b@ecs.osaka-u.ac.jp  
学籍番号 09B22084  
提出日 2025 年 2 月 5 日

## 1 課題 11 で作成したプログラムの動作仕様

課題 11 では、乗算を行うプログラムを実装した。このプログラムの仕様は以下の通りである。

- 掛けられる数と掛ける数は、FPGA ボードから入力を行う。どの番地に何が格納されるかは、以下の表 1 の通りである。

番地	格納される内容
8000	掛けられる数を格納する。乗算実行後も値は変化しない。
8001	掛ける数を格納する。乗算実行後は 0 となる。
8002	掛けられる数と掛ける数に、いくつ負の数が含まれるかを格納する。
8003	乗算の実行結果を格納する。結果が負の数なら負の値が格納される。

表 1: 各番地に格納される内容

- まず最初に掛けられる数の入力を行う。掛けられる数は KEY0 を押下することでその値が確定する。
- 次に掛ける数の入力を行う。掛ける数は KEY1 を押下することでその値が確定する。
- KEY1 を押下し、掛ける数を確定させた後すぐに掛け算の処理が実行される。

## 2 課題 12, 13, 14 で追加した命令の実装方法

課題 12,13,14 で追加した命令は、シフト命令である。以下にその仕様を記述する。

命令	オペランド	サイズ	コード	動作
SLL	num	1word	11000001(C1)	$IP \leftarrow IP+2, A \leftarrow A \ll \text{num}(C,Z)$
SRL	num	1word	11000010(C2)	$IP \leftarrow IP+2, A \leftarrow A \gg \text{num}(C,Z)$
SLA	num	1word	11000011(C3)	$IP \leftarrow IP+2, A \leftarrow A \ll \text{num}(C,Z)$
SRA	num	1word	11000100(C4)	$IP \leftarrow IP+2, A \leftarrow A \gg \text{num}(C,Z)$

表 2: 追加した命令の仕様

上記の命令は、以下の手順に従って実装を進めた。

1. ALU 内での処理の追加
2. ALU のデータパスの変更
3. 外部出力用ジョンソンカウンタを追加
4. 各信号の条件を変更する。

以下で、これらの詳細について述べる。

### 2.1 ALU 内での処理の追加

今回のシフト命令は ALU 内で演算を実行することにした。ALU 内では modeALU の値によって fout に格納される値が決定されるため、modeALU にシフト命令の場合を追加した。追加後の modeALU の信号割り当ては以下のようになる。



### 3 工夫点

#### 3.1 課題 11

課題 11 では、掛けられる数と掛ける数を確定させる際に使用する KEY の種類を変更するという工夫を行った。この工夫を行わず、掛けられる数と掛ける数をどちらも KEY0 で値を確定させてしまうと、掛けられる数のみを確定させるつもりが、クロック周波数が早く、掛ける数も同時に確定させてしまうためである。この問題に対する改善策として、私は掛けられる数を確定させるときは KEY0 を、掛ける数を確定させるときは KEY1 を使用することにより、クロック周波数が高いことの影響を受けない設計とすることができた。一方、これ以外の改善策として、JUMP 命令を用いて、何も処理しないループをクロック周波数に合わせて十分時間行うという方法が考えられた。しかし、この方法だと JUMP 命令を用いるため、プログラムに変更を加えた場合アドレスの変更を行う必要がある。一方、KEY の種類を変更する方法では、どの KEY を使用しても FFFE 番地の値が変更されるだけであるため、アドレスの変更を行う必要がない。また、SUBA 命令などを用いたゼロフラグの情報を活用すれば掛ける数と掛けられる数でほぼ同じプログラムで実装することができるため、可読性が高まる。

#### 3.2 課題 12,13,14

課題 12,13,14 の工夫点は 2.1 で述べた通り、レジスタ B を即値の入力の格納先として使用するのではなく、新たにデータバスを設定しレジスタ C を使用した点である。

### 4 拡張課題

#### 4.1 拡張課題 a-1：最大公約数を求めるプログラム

最大公約数を求めるプログラムの仕様は以下の通りである。

- レジスタ A に格納されている数値の方がレジスタ B に格納されている数値よりも大きい。
- 各番地の格納内容は以下の通りである。

番地	格納内容
8000	ユークリッドの互除法における、割る数 (計算用)
8001	ユークリッドの互除法における、余り (計算用)
8002	レジスタに格納された数値の最大公約数 (結果)

表 4: 各番地に格納される内容

最大公約数を求めるプログラムは、以下の手順にしたがって処理を実装した。

- 1.

#### 4.2 拡張課題 a-2：最小公倍数を求めるプログラム

#### 4.3 拡張課題 b：演算機の改善

### 5 感想

以前までシミュレーション上でしか動きを見られなかったものが実機で動いたときにとても嬉しかった。きちんと動作の仕組みを理解し、準備を十分に行ってから実践することの重要性を再認識できたと感じた。来年以降の研究でもこの講義を通して学んだことを活かしていきたい。