

## 表計算による教育用 CPU 設計・シミュレータのための分岐, ALU およびセクタの実現 Realization of a procedure for branch, ALU and selector of CPU design and simulation tool on spread sheet

福永 哲也<sup>†</sup>  
Tetsuya Fukunaga

### 1. はじめに

コンピュータを学習する上で CPU の設計は役に立つ。しかし、学生が自分で設計したオリジナルの CPU に対しては、その CPU を動作させる環境がないことから、プログラムの動作確認を行うことができない。そこで、これまでに Excel を利用して CPU を設計し、設計した CPU のシミュレーションを行えるツールを開発してきた[1]。この場合、設計とは各命令に対して制御信号出力を決め、自分の希望通りに動作する CPU を作り上げることを指す。従って、CPU の回路設計は行わず、回路としては与えられたものを用いる[2]。図 1 は設計する CPU の一例を示す。CPU は命令を命令レジスタに読み込んだ後デコーダで制御信号を出力する。その制御信号に従ってセクタ、ALU が動作することで、命令を実現する。また、全体の動作はプログラムカウンタで制御されて次に実行する命令が決まる。各命令と制御信号出力の一覧表を命令セット表と呼び、ここでは命令セット表を自分で考えることで CPU の動作を理解する。また、その後自分の設計した CPU でプログラムを実行することで理解が深まる。

Excel によるシミュレータは完成している。通常シミュレーションを行う場合、実際の CPU をプログラムで表現する。その場合、実際の CPU におけるレジスタなどは変数としてプログラム内に存在する。Excel によるシミュレータでも同様で、内部で VBA(Visual Basic for Applications) のプログラムを動作させてシミュレーションを行うものであった。しかし、学生によっては Windows でない環境であ

ったりするため、全員が Excel を動作させられない。そこで、Excel オンライン[3]を利用して、ブラウザがあれば誰でもシミュレーションを実行できるようにする。しかし、多くの Web ページではプログラムを動作させることができず、Excel オンラインでも VBA が動作しない[4]。そこで、プログラムを使わずに表計算のみでシミュレーションを実行するスプレッドシートを開発する。

### 2. シミュレータの構成

通常の回路やシミュレーションでは各レジスタや各変数が値を保持しており、その値によって動作を選択する。この動作を繰り返すことで時間が進み、シミュレーションが進行する。しかし、プログラムを使わない場合、この変数の値で動作を変えていく方法では、変数の値が変化した場合に以前の計算結果も変わってしまうため問題がある。これは、通常のシミュレーションでは時間変化がプログラムの進行と共に変化するが、スプレッドシートでは、時間変化もスプレッドシート的一方の軸にそって存在しているためである。従って、スプレッドシートでは 1 つの列が 1 つの変数となり、1 つの行が 1 つの実行ステップとなる。この場合 1 つのセルはある変数のある時刻 (ある実行ステップ) での値であり、ある変数の値は時刻の変化に従って複数存在し、時間と共に変化することになる。

このようにスプレッドシートだけでシミュレーションを進める場合は、通常のシミュレータと動作が異なる。そこでまず、CPU のシミュレータ (アセンブラのシミュレータ) を作成する上で重要な、分岐の実現方法と ALU やセクタの実現方法についてまとめる。

### 3. 分岐の実現方法

アセンブラプログラムのシミュレーションを行うには、分岐の実現が必要である。分岐を行うことでプログラムの複雑さを増すことができる。図 2 は分岐をもつアセンブラプログラムを示す。プログラムの意味は上から順にリセット、ロード (3)、デクリメント、ジャンプノットゼロである。JPNZ でラベル L へジャンプするため、上記のプログラムではループを 3 回繰り返す。このプログラムにおけるプログラムカウンタの値を図 3 に示す。プログラムカウンタの値は次に実行する命令を表しており、最初は 2 である。しかし、JPNZ (4 ステップ) では 3 となり、プログラムの実行は 3 へ戻り分岐を行っていることがわかる。また

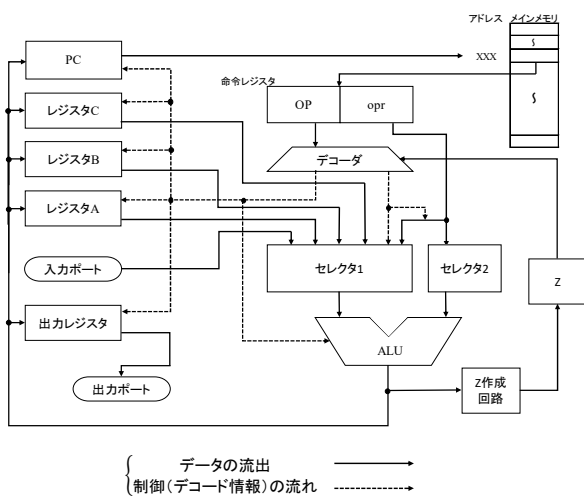


図 1 設計する CPU 例

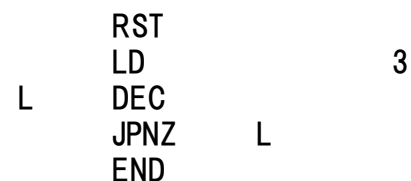


図 2 分岐をもつアセンブラプログラム

<sup>†</sup> 岐阜工業高等専門学校 National Institute of Technology, Gifu college

図 3 のプログラムカウンタの値をもとに次に実行する命令を決め、その命令を順に並べたもの（これが実際の CPU で動作する命令であり、シミュレータ実現に必要なものである）を図 4 に示す。プログラムはプログラムカウンタに従って読み込まれている。最初はプログラムカウンタが 1 ずつ増加して図 1 のプログラムを上から順に実行しているが、4 ステップ目の JPNZ では図 3 に示すプログラムカウンタの値は 3 となり、図 4 における次の命令は DEC となる。同じことを繰り返しながらループ処理を行うが、最後の JPNZ では Zero フラグがたっているため（DEC 命令の結果が Zero となっているため）ジャンプを行わず、図 3 の 8 ステップにおけるプログラムカウンタの値も 1 増えて 5 となる。従って図 4 の命令でも JPNZ の後に END となり、プログラムが終了する。図 4 では DEC 命令が 3 回実行され、このプログラムはループを 3 回繰り返していることがわかる。この様にして分岐が実現される。

#### 4. ALU とセレクタの実現

CPU には ALU とセレクタも必要である。ALU は制御信号によって計算機能を変更する。制御信号のビット数によ

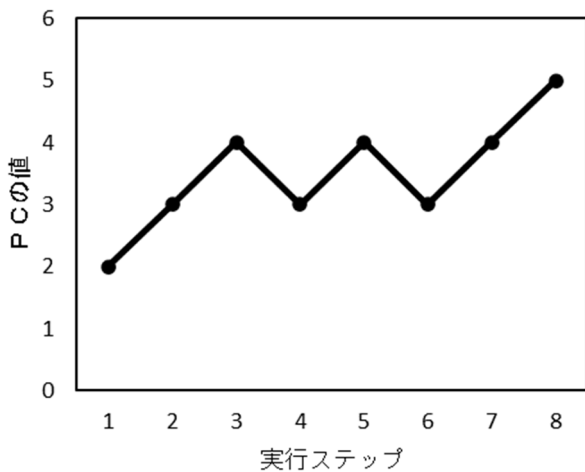


図 3 プログラムカウンタの値の変化

```
RST
LD
DEC
JPNZ
DEC
JPNZ
DEC
JPNZ
END
```

図 4 展開されたプログラム

って計算の種類も増えていくことになる。セレクタは制御信号によって複数の入力の中から一つを選択する。やはり制御信号のビット数によって選択できる入力信号の種類は増えていく。どちらも制御信号入力をもっており、制御信号の値によって動作を変える。

通常のプログラムによるシミュレータでも if 文や switch 文などによって動作を変える。その場合、条件となる制御信号の値は制御信号を表す変数の値となる。スプレッドシートでもセルで実行できる関数として、if 関数や switch 関数があり、それらの関数の条件に制御信号の値を使う。ただしスプレッドシートの場合には制御信号も複数（行の数だけ）存在することから、どの行の制御信号を使うかで動作が異なってくる。例えばセレクタにおいて、ある行では A レジスタを選択する制御信号がでており、別の行では他のレジスタを選択する制御信号が出ている場合がある。また、選択する値についても、ある行の A レジスタの値が決まっていると、別の行では A レジスタの値は別の値になる。これらの制御信号やレジスタの値は、ある一つの時刻での制御信号やレジスタの値を示す。そこで、スプレッドシートでの ALU やセレクタの実現では、同じ行の制御信号で制御し、同じ行のレジスタ等の値を用いる。

#### 5. おわりに

今回、Web ブラウザがあればだれでも利用できるようにプログラムを使わない CPU シミュレータの開発に必要な分岐、ALU およびセレクタの開発を行った。通常のプログラムによるシミュレータでは一つの変数として表現されるものがスプレッドシートによるシミュレータでは 1 つのセルとならず、1 つの列となる。これは、スプレッドシートによるシミュレータでは、X 軸と Y 軸で表されるセルにおいて一方の軸は時間を表すために使用しており、1 つのセルとは 1 つの変数の 1 つの時刻における値を表すからである。スプレッドシートの中にプログラムカウンタを表す列を入れて実行すると、ループ動作が実現できた。これは、今後作成する CPU シミュレータの基礎となるものである。また、CPU の構成要素として重要な ALU やセレクタも実現した。ALU やセレクタはプログラムの場合と同様に、セル関数としての if 関数や switch 関数で実現できる。これらの要素を組み合わせることで、プログラムを利用しない CPU シミュレータを構成することができる。今後はこれらの要素技術をまとめてシミュレータを完成する必要がある。

#### 参考文献

- [1] 福永哲也, “電子計算機のための Excel を用いた CPU シミュレータの開発と課題の実施”, 工学教育, Vol.67, No.6 (2019).
- [2] 堀圭太郎, “図解コンピュータアーキテクチャ入門第 2 版”, 森北出版, pp.1-63 (2011).
- [3] <https://blog.hubspot.jp/excel-online> (2020.6.17 参照).
- [4] [https://support.microsoft.com/ja-jp/office/%e3%83%96%e3%83%a9%e3%82%a6%e3%82%b6%e3%83%bc%e3%81%a8-excel-%e3%81%a7%e3%81%ae%e3%83%96%e3%83%83%e3%82%af%e3%81%ae%e4%bd%bf%e7%94%a8%e3%81%ae%e7%9b%b8%e9%81%95%e7%82%b9-f0dc28ed-b85d-4e1d-be6d-5878005db3b6?ui=ja-jp&rs=ja-jp&ad=jp#\\_functions](https://support.microsoft.com/ja-jp/office/%e3%83%96%e3%83%a9%e3%82%a6%e3%82%b6%e3%83%bc%e3%81%a8-excel-%e3%81%a7%e3%81%ae%e3%83%96%e3%83%83%e3%82%af%e3%81%ae%e4%bd%bf%e7%94%a8%e3%81%ae%e7%9b%b8%e9%81%95%e7%82%b9-f0dc28ed-b85d-4e1d-be6d-5878005db3b6?ui=ja-jp&rs=ja-jp&ad=jp#_functions) (2020.6.17 参照)