

# マイクロプロセッサ演習

2004 年度

第 10 回

## 1 単一サイクルのデータパスの欠点

前回は単一サイクルのデータパスについて学んだ。これは、図 1(a) のように、一つの命令を単一サイクルで実行するデータパスであった。今回はそ

(a) 単一サイクル



(b) マルチサイクル

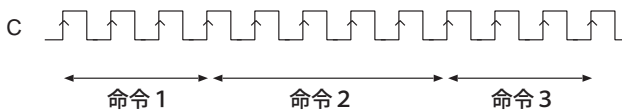


図 1: 一命令を (a) 単一サイクルで (b) マルチサイクルで処理する。

れとは異なり、一命令を複数クロックで実行するマルチサイクルのデータパスについて学ぶ。

まず、単一サイクルのデータパスでは、命令によって表 1 のようなパスを通して演算が行なわれた。表からわかるように、命令に応じて処理の内容が異なる。このように様々な命令を全て一サイクルで実行するためには、**クロック・サイクル時間を最も時間のかかる命令に合わせてねばならない**。表では、ロード命令 (lw) 命令が実行可能なように、クロック・サイクルを長く取る必要があるが、これでは、他の命令を実行する際に無駄が生ずる (表の空欄)。

さらに、この表には書かれていないような複雑な命令 (例えば浮動小数点命令など。教科書 346 ページ例題参照) の実行も考慮すると、無駄はさらに大きくなる。そのため、**現代のコンピュータでは単一サイクルによるデータパスはまず採用されない**。これを解決するのは図 1 のように、一命令の実行に

複数のサイクルをかけるようなデータ処理方式であり、それを行なうのが今回学ぶマルチサイクルのデータパスである。これは、ラフに言えば、表 1 において「R 形式には 4 サイクル」「ロード命令には 5 サイクル」かけるようなデータ処理方式である。

また、単一サイクルのデータパスにはもう一つ欠点がある。全ての命令を一サイクルで実行しなければならないため、**一つの機能ユニットを一命令に一回しか使えないこと**である。前回の単一サイクルのデータパスにおいて、メモリを「命令メモリ」と「データメモリ」に分けていたのはそのためであるし、ALU に関しても、「ALU ユニット 1 つ、加算ユニット 2 つ」が存在し、これではユニット数が多くなり無駄が多い。マルチサイクルのデータパスを用いると、一つのユニットを一命令で複数回用いることができるようになる。

## 2 マルチサイクルのデータパス

単一サイクルのデータパスの欠点を補うのが図 4 に示したマルチサイクルのデータパスである。前回も触れたように、このマルチサイクルのデータパスは図 3 のようにクロックエッジごとに状態論理要素に書き込みを行ないながら演算処理を行なう。その

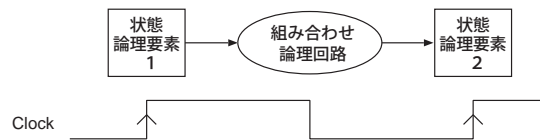


図 3: マルチサイクルのデータパスによるデータ処理の模式図。

ことが分かりやすいように、図 4 のデータパスのうち状態論理要素を緑色で囲った。新たな状態論理要素として

命令のタイプ	使用される機能ユニット				
R 形式	命令フェッチ	レジスタアクセス	ALU	レジスタアクセス	
ロード命令 (lw)	命令フェッチ	レジスタアクセス	ALU	メモリアクセス	レジスタアクセス
ストア命令 (sw)	命令フェッチ	レジスタアクセス	ALU	メモリアクセス	
分岐 (beq 等)	命令フェッチ	レジスタアクセス	ALU		
ジャンプ (j)	命令フェッチ				

表 1: 単一サイクルのデータパスにおける機能ユニットの利用。

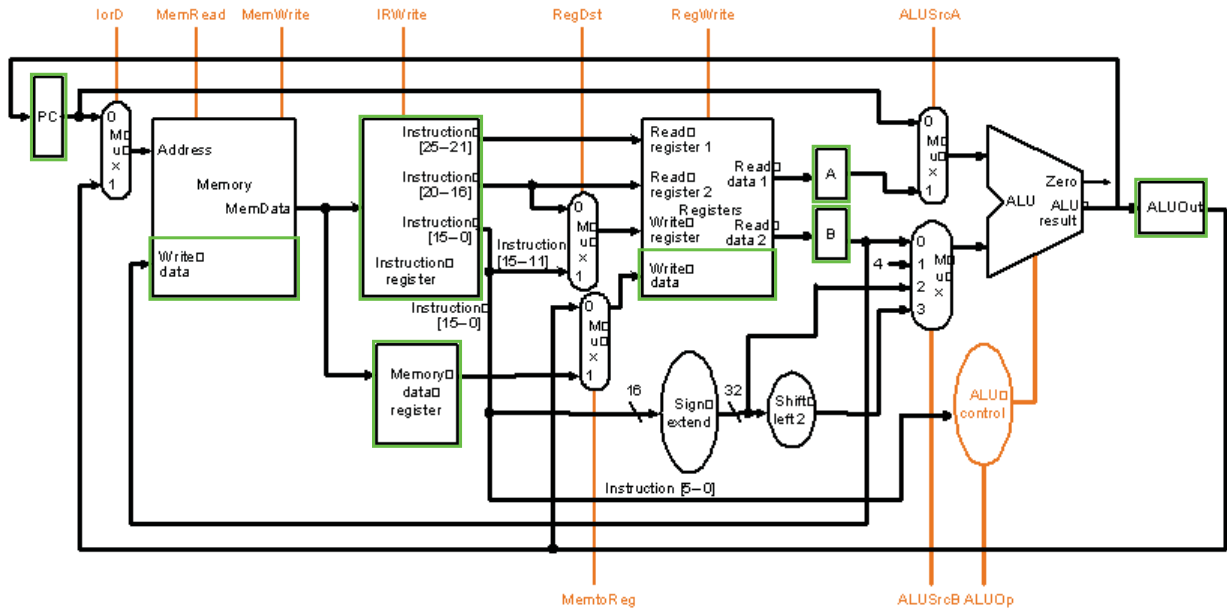


図 2: マルチサイクルのデータパス。状態論理要素として振舞う機能ユニットを緑色で囲った。教科書 351 ページ図 5.32 と同等である。

- 命令レジスタ (Instruction Register)
- メモリデータレジスタ (Memory Data Register)
- A および B
- ALUOut

行には 5 クロック、ストア命令の実行には 4 クロックかかる。

以上を元に、いくつかの命令について、データパスをたどって動作を理解せよ。この際、クロックごとに状態論理要素に書き込みを行なうので、状態論理要素に着目して動作を追って行くと理解が容易である。

が追加され、値が保持できるようになっていることに注意しよう。また、前回触れたようにメモリとレジスタファイルは「読み出し時は組合せ論理回路として、書き込み時は状態論理要素として」振舞うことに注意しよう。

このデータパスにおいて、代表的な命令の実行は表 2 に基づいて行なわれる。縦に時間が進み、一行あたり一クロックがかかることに注意。すなわち、R 形式命令の実行には 4 クロック、ロード命令の実

ステップ名	R 形式命令	メモリ参照命令 (lw/sw)	分岐命令	ジャンプ命令 (j)
命令フェッチ	IR = メモリ [PC] PC=PC+4			
命令デコードとレジスタのフェッチ	A = レジスタ [IR[25-21]] B = レジスタ [IR[20-16]] ALUOut = PC + (符号拡張 (IR[15-0]) << 2) (j 命令用。2 ビットシフトは 4 倍)			
実行、 メモリアドレス計算 分岐/ジャンプ完了	ALUOut = A 演算子 B	ALUOut = A + 符号拡張 (IR[15-0])	if(A==B) then PC = ALUOut	PC= PC[31-28]    (IR[25-0]<<2)
メモリアクセス または R 形式命令完了	レジスタ [IR[15-11]] =ALUOut	ロード:MDR =メモリ [ALUOut]、 ストア:メモリ [ALUOut] = B		
メモリ読み出し完了		レジスタ [IR[20-16]] = MDR (lw のみ)		

表 2: 命令を実行する 5 つのステップの命令タイプ別の要約。縦に時間が進む。略号は以下を表す。IR: 命令レジスタ (Instruction Register)、MDR: メモリデータレジスタ (Memory Data Register)。教科書 359 ページ 図 5.35 と同等である。

### 3 問題

#### [問 1]

単一サイクルマシンの性能について以下の値を求めなさい。ただし、主要なデータパス回路の実行時間は以下の値とし、これ以外のマルチプレクサなどの遅延は無視する。

命令メモリ	5ns
レジスタ読み出し	4ns
ALU 演算	3ns
データメモリ	5ns
レジスタ書き込み	4ns

1. R 形式命令、lw、sw、分岐、ジャンプの各命令の実行時間はいくらになるか。このマシンは単一サイクルマシンであると考えよ (すなわち、前回の復習)。
2. 命令に R 形式命令、lw、sw、分岐、ジャンプしかない場合、この単一サイクルマシンのクロック周期は最短でいくらにできるか。

#### [問 2]

マルチサイクルマシンの性能について以下の値を求めなさい。ただし、データパス回路の各ステップの実行時間は以下の値とする。なおこの値は、ALU

命令フェッチ	5ns
命令デコードとレジスタフェッチ	5ns
実行、メモリアドレスの計算、 または分岐の完了	4ns
メモリアクセス、または R 形式命令完了	4ns
メモリ読みだし完了	4ns

やメモリアクセスにかかる時間は問 1 と同等とし、PC、IR、MDR、A、B、ALUOut 等の読み出し/書き込みの遅延を無視して算出したものである。(すなわち、問 1 と問 2 は対応している)

1. このマルチサイクルマシンのクロックサイクル時間は最短いくらにできるか。
2. あるプログラムにおける命令の割合は、R 形式 50 %、ロード 20 %、ストア 10 %、分岐 18 %、ジャンプ 2 % であるという。この時、1 命令あたりの平均クロック数 (CPI) はいくらになるか。

