

▷ その他の言語

・HTML... 文章タイプの言語 マークアップ言語
タグで Webサイトの見た目を構成している << CSSで装飾をする、JavaScriptで制御

・XML... HTMLと異なり自分でタグを定義できる
HTMLはWebページの作成 XMLはデータの交換に用いられる

・Ajax... Webページを再読み込みすることなくバックグラウンドでサーバでデータをやり取りできる技術。
JavaScriptとXMLを使って非同期通信をする。これがAjax
最近ではXMLの代わりにJSONを使ってもなる

・コンピュータの構成要素

▷ CPU (プロセッサ)

CPU: 中央処理装置 コンピュータ上の処理を行う機械 プロセッサともいう。

CPUの性能はクロック周波数、MIPS、コア数で決まる。

・クロック周波数: CPUの動作の速さを指標

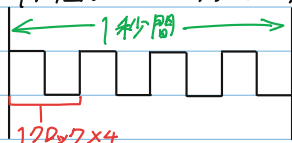
クロック数のテンポに合わせて処理を行う。

↑ 1秒間に何テンポもつのかを指す

単位はHzで、1秒間に1回の信号を発生させることを1Hzという。

コンピュータ内部では命令の処理にかかるCPUの時間をクロック数で表す。
1回分のテンポを1クロックといい、複雑な命令ほどクロック数も増える。

例: 4Hzのクロック周波数



命令をより速く実行するにはより速いテンポに合わせたCPUが必要

1秒間に処理できる命令数を計算する公式

$$1秒間に実行できる命令数 = \frac{\text{クロック数}}{1つの命令を実行するのに必要なクロック数}$$

例えばクロック周波数 2.5GHzのCPUが1秒間に処理できるクロック数は最大で
2.5ギガクロック 1ギガ=10⁹=10億なので、25億クロック、1つの命令に必要な
クロック数が2の場合、1秒間に処理できる命令数は

$$\text{命令数} = \frac{2500000000}{2} = 1250000000$$

12.5億個処理することが出来る //

CPUのクロック数が高いほど高性能といえる。

・ MIPS (Millions of Instructions Per Second)

MIPSのMはMillionsのM すなわちCPUが1秒間に何百万の命令を処理できるのかを表したモノ。

例: 2MIPS は 1秒間に200万の命令を処理できる。

MIPSの公式

$$\text{MIPS} = \frac{1\text{秒}}{\text{平均実行時間}} \times 10^{-6}$$

平均実行時間... 1つの命令を実行するのにかかる時間の平均値。
処理に時間がかかるもの、すぐに実行できるものもある。
なので平均値で求める。

10⁻⁶ ... 10⁻⁶を掛けるのはMIPSのMは100万を意味するから。
↑これを掛け算することで、MIPSの単位に合わせる事が出来る。

例題: 2MIPSの平均実行時間を求めよ。

公式にあてはめると

$$2\text{MIPS} = \frac{1\text{秒}}{\text{平均実行時間}} \times 10^{-6}$$

ここに10⁻⁶をかけるので平均実行時間の単位はマイクロ秒となる。

$$\begin{aligned} \text{平均実行時間} &= 1\text{秒} \div 2\text{MIPS} \times 10^{-6} \\ &= 1 \div 2000000 \times 10^{-6} \\ &= 1 \div 2 \\ &= 0.5 (\text{マイクロ秒}) \\ &= 500 (\text{ナノ秒}) \end{aligned}$$

秒	基本単位
ミリ秒	10 ⁻³ 秒
マイクロ秒	10 ⁻⁶ 秒
ナノ秒	10 ⁻⁹ 秒

・コア

コアはCPUにおける演算ユニット、コア数分、演算ユニットが増えるので処理に強くなる。

・シングルコア…演算ユニット1つのみ

・マルチコア…複数の演算ユニットを持つため、高速で処理を行うことができる。

— デュアルコア…2コアのCPU

— クアッドコア…4コアのCPU

— オクタコア…8コアのCPU

マルチコアCPUは、それぞれのコアが別々の処理を実行できるので、コア数の高いものがより優れている。

▷CPUの処理の流れ

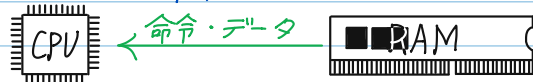
主記憶装置(RAM): 一時的にデータを保管する場所。主記憶装置の内部には、配列と同じように細かく区分され、それぞれにアドレスが割り振られている。

CPUは、アドレスを指定して主記憶装置から命令やデータを取り出す。

主記憶装置に格納されている命令は、プログラムに書かれた複数を命令のうちの一つで、例えば、「～のデータを取り出せ」、「～を足し算しろ」といったCPUへの命令が書かれている。

また、主記憶装置に格納されているデータには、計算対象となるデータ、計算結果のデータがある。

1. CPUが命令とデータを取り出す。



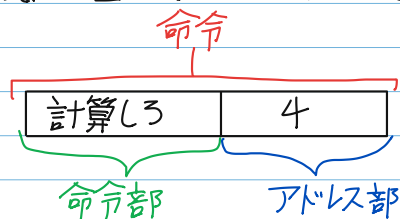
2. CPUが計算処理



3. CPUが計算結果をRAMに保存



主記憶装置からCPUに書き込まれる命令は、以下のよう構成される。



- ・命令部: CPUが実行する処理
- ・アドレス部: その命令に必要なデータを呼び出すためのアドレス。
ここでは「4」になっているので、RAMの4番のデータを読み込む。

RAM内には命令と命令によって呼び出されるデータ、命令を実行した結果が入っている。

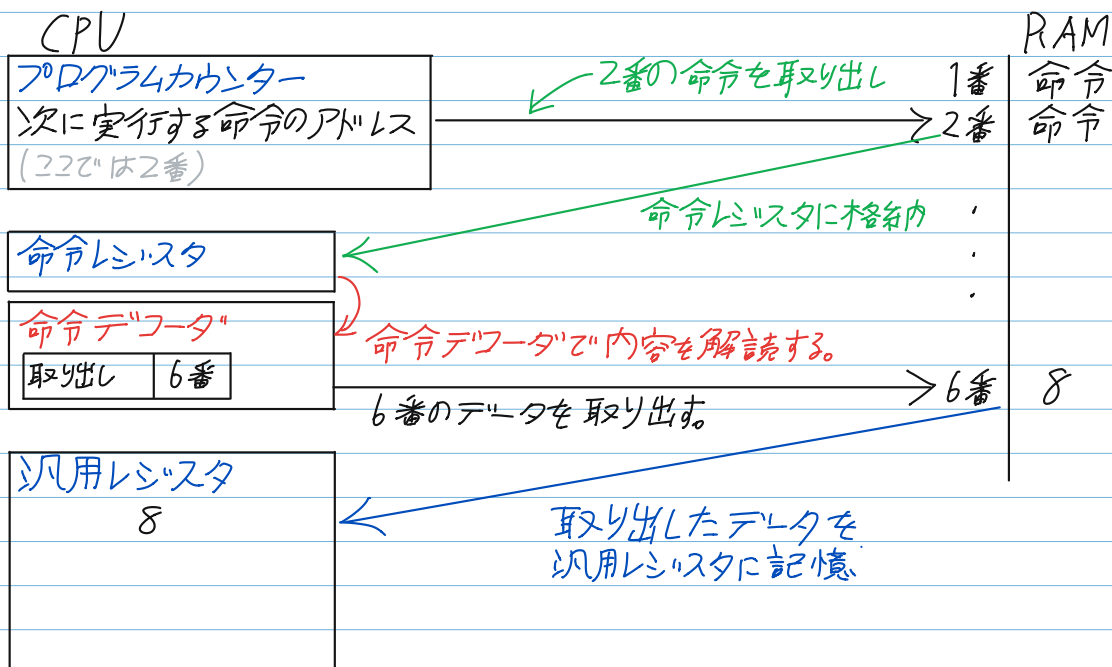
レジスタ: RAMから読み込んだデータはCPU内のレジスタと呼ばれる記憶装置に格納される。

(CPU内にある一時的に命令やデータを格納する装置)

レジスタはカテゴリに分けてデータを記憶する。

種類	説明
プログラムカウンタ	次の命令のアドレスを格納するレジスタ
命令レジスタ	命令を格納するレジスタ
汎用レジスタ	データを格納するレジスタ

・CPUの処理の流れ

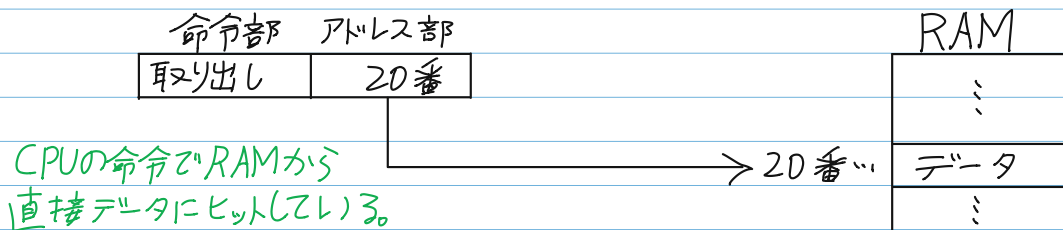


・アドレス指定方式

CPUがRAMからデータを取り出す上でアドレスを指定し命令やデータを取り出す。
このアドレス指定方式には主に2種類がある。

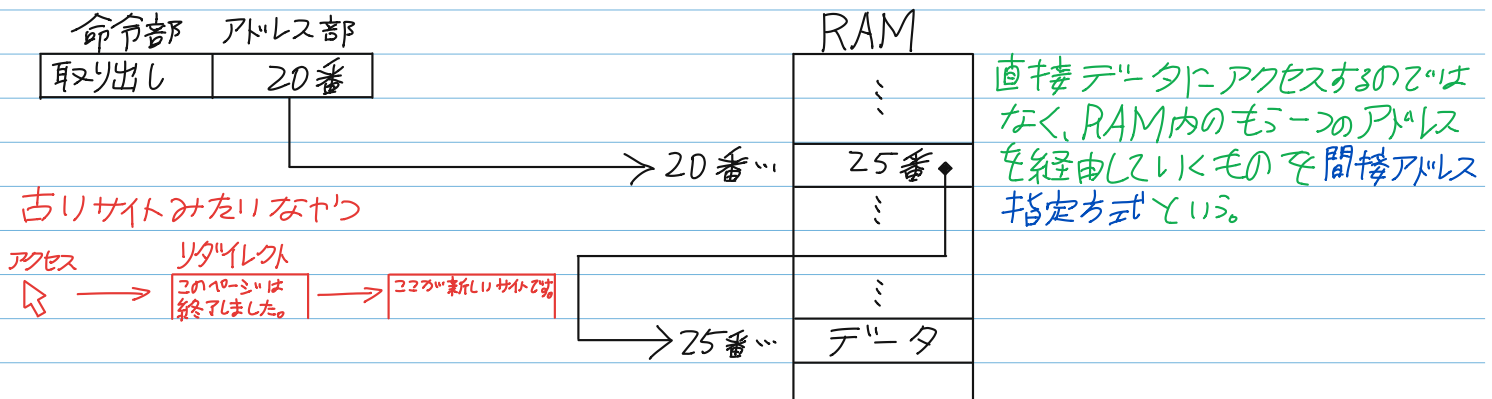
▷直接アドレス指定方式

RAMから呼び出した命令の中に 有効アドレス (CPUの処理対象となるデータが書かれているアドレス) を基に、そのデータへ直接アクセスして、データを取り出すことを 直接アドレス指定



▷間接アドレス指定方式

CPUがRAMから読み込んだ命令のアドレスの先にデータの 有効アドレス が書かれている。



← レジスタは非常に小さい

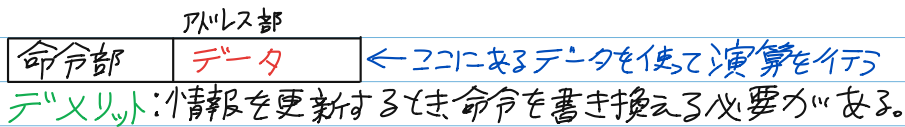
× リット: 命令のアドレス部には容量に限りがあるので、RAMを使って大規模なアドレスを保存して命令を呼び出せる。データを変更しやすい。

デ× リット: RAMへのアクセスが2回必要になり、処理に時間がかかる。

補足知識

▶ 即時値アドレス指定方式

命令のアドレス部に直接データを記入するタイプ



▶ 指標アドレス指定方式 (インデックスアドレス指定方式)

命令のアドレス部にインデックスレジスタの値を加算して、その値を有効アドレスとして使う方式

インデックスレジスタとは: RAMから連続したデータを取り出るとき、ベースのアドレスに決められた数値を加えて、要素にアクセスする仕組み。

インデックスレジスタの値はCPUに、今アクセスしているRAMの要素の位置を記憶するためのもの。

ベースアドレス (配列の先頭として扱うアドレス) にインデックスレジスタの値を足して、データサイズを掛けることで、アドレスを指定することが出来る。

$$\text{ベースアドレス} + \text{インデックスレジスタ} \times \text{データサイズ}$$

・JavaScriptでの例

```
let array = [10, 20, 30, 40, 50];
```

この配列で1つの整数が4バイトだとすると (ベースのアドレスは1000)

$$\text{インデックス} 0 = 1000 + 0 \times 4 = 1000$$

$$\text{インデックス} 1 = 1000 + 1 \times 4 = 1004$$

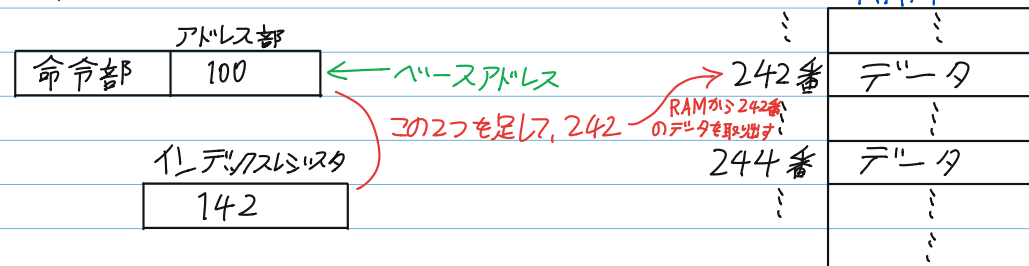
$$\text{インデックス} 2 = 1000 + 2 \times 4 = 1008$$

インデックスレジスタが0 → 1 → 2 ... と1ずつずれると、データのサイズに合わせて増えていく。

この特性を活用して、アドレスを指定

CPU

RAM

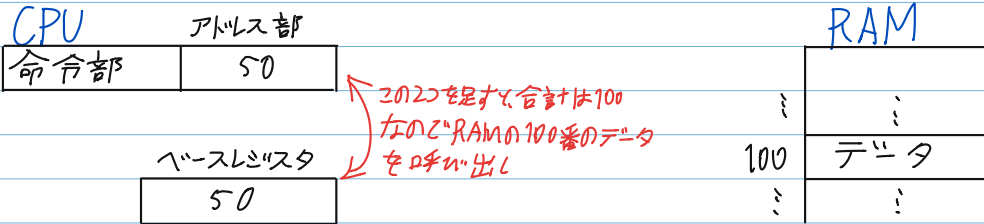


インデックスレジスタが142なら、足して244のデータをRAMから参照

連続したアドレスを扱うときに用いる。

▷ベースアドレス指定方式

インデックスレジスタと異なり、RAM上の配列の先頭や基準値を用いてアドレスを指定する方法。

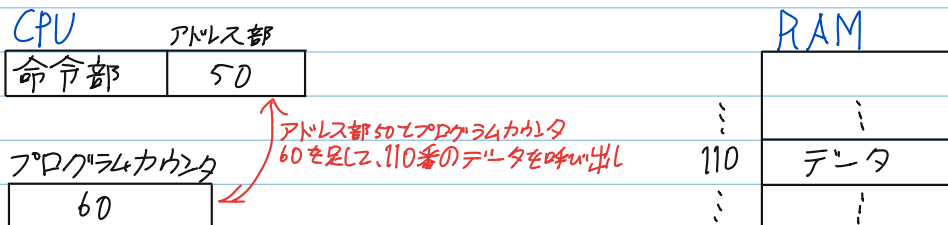


★指標アドレスとはちょっと違う

ベースアドレス指定方式では他のメモリで実行お際に命令を変えなくて済む
↑ベースレジスタはRAMの配列の先頭を記憶しているから。

▷相対アドレス指定方式

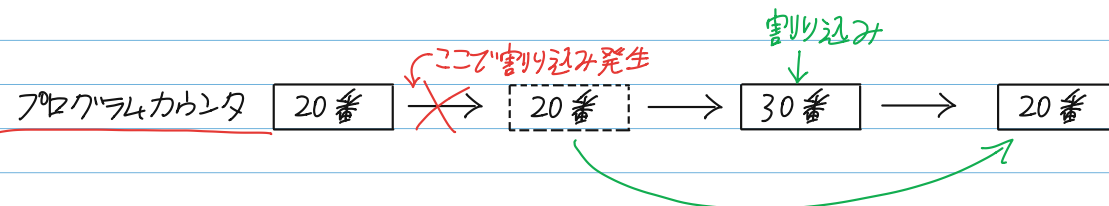
アドレス部の値にプログラムカウンタの値を加算、そうしてできた値を有効アドレスとする方式



プログラムカウンタは次に実行する命令のアドレスを記憶するので、ベースアドレスと同様にRAMのどこデータが格納されても相対的な位置関係から、データを取り出すことができる。

▷データの割り込み

実行中のプログラムを中断し、別のプログラムに切り替えることをデータの割り込みと言う。



例えば「20番の命令実行中に30番の命令が割り込みをしたら、先に30番の命令を処理し、その後に20番の命令を再び実行する。」

↑だからCPUのプログラムカウンタに一度保存する。

プログラムカウンタが割り込まれた場所から処理を再開する。

・割り込みを行う主な目的

優先度の高い命令があった場合に、CPUが他の命令を実行中であっても優先度の高い命令を先に処理すること。

これをエラーが発生したときにイテコで、CPUは、そのエラー解決の命令にセカリ替え、後の処理が進むようになる。

割り込みの種類

・内部割り込み：実行中のプログラムなどソフトウェアが原因で起こる

・外部割り込み：ハードウェアが原因で起こる

▷内部割り込みの種類

・プログラム割り込み：プログラムの実行中に発生したエラーによって起こる割り込み。

例：ゼロ除算、演算結果のオーバーフロー、未定義の命令
メモリアドレスが存在しない領域にアクセスしたとき

・スーパーバイザーコール割り込み：OSの機能を実行するとき生じる割り込み

例：プログラムの実行中にユーザーにデータ入力并要求

・ハードウェア割り込み：主記憶装置に存在しないハードウェアへのアクセスで生じる割り込み

▷外部割り込みの種類

・タイマ割り込み：CPUのタイマによって発生する割り込み

インターバルタイマやウォッチドッグタイマの割り込み

・インターバルタイマ：一定時間ごとに割り込みを行ってマルチタスクを実現する。

・ウォッチドッグタイマ：一定時間カウントしてプログラムに異常があった際、割り込みをする。

・入出力割り込み：入出力が完了したことによる割り込み、処理中の装置の異常でも割り込みを行う。

・機械チェック割り込み：装置内の検査で異常が見つかったときに生じる割り込み

例：電源異常、メモリにデータを正しく転送できなかったとき。

・コンソール割り込み：プログラム実行中にコンソール（入力装置など）を通じて処理を要求したときに生じる。

例：強制終了、再起動など...

過去問

1. 平均命令実行時間が20ナノ秒のコンピュータがある。

このコンピュータの性能は何MIPSか。

$$20 \text{ ナノ秒} = 0.02 \text{ マイクロ秒} = 0.00002 \text{ ミリ秒} = 0.0000002$$

$$\text{MIPS} = \frac{1 \text{ 秒}}{20 \times 10^{-9}} \times 10^{-6}$$
$$= 50$$

$$\begin{array}{r} 50 \\ 0.02 \overline{) 100} \end{array}$$

答え: 50MIPS //

2. 50MIPSのプロセッサの平均実行時間は幾らか。

$$50 \text{ MIPS} = \frac{1 \text{ 秒}}{\text{平均実行時間}} \times 10^{-6}$$

$$\text{平均実行時間} = 1 \text{ 秒} \div 50 \times 10^{-6}$$

$$= 1 \div 50$$

$$50 \overline{) 1.00}$$

$$= 0.02 \text{ マイクロ秒}$$

$$= 20 \text{ ナノ秒}$$

答え: 20ナノ秒 //

3. 1GHzのクロックで動作するCPUがある。このCPUは機械語の1命令を平均0.8クロックで実行できることがわかっている。このCPUは1秒間に平均何万命令を実行できるか。

$$1 \text{ 秒間に実行できる命令数} = \frac{10 \text{ 億クロック}}{0.8 \text{ クロック}}$$

$$= 125000 \text{ 万}$$

$$\begin{array}{r} 1250000000 \\ 8 \overline{) 10000000000} \\ 8 \\ \hline 20 \\ 16 \\ \hline 40 \end{array}$$

4. 割り込み処理の終了後に、割り込みによって中断された処理を割り込まれた場所から再開するために、割り込み発生時にプロセッサが保存するものはどれか。

A. インテックスタ

I. テンタリスト

ウ. プログラムカウンタ

E. 命令リスト

割り込み時、CPUは実行中のタスクを中断して、

プログラムカウンタに、割り込みの処理終了後の命令として、

保存する。

答え: ウ //