■目次(ほげが今回の範囲)

第1部0と1だけでどうやって計算するの?

第2部 コンピュータの気持ち

第3部 情報をどのように表現するか 第4部 コンピュータシステム

★ 4 コンピュータの構成 (1)

★ 5 CPU の仕組み

★6 CPU と機械語

第5回の授業では、「第1部」の締めくくりとして、加減算の両方ができる論理回路を作った。また、「第2部」 の導入として、現代では様々な所で様々な種類のコンピュータが使われていること、それらは全て CPU とメモ リを中核とした共通の仕組みを持っていることを知った。今回と次回は、CPU とメモリの構成・仕組みについ て学ぶ、第1部で学んだ「0/1で数を表現してその計算をする論理回路」を発展させていったものが CPU で ある、ということを理解するのが目標である.

CPU の仕組み

★ 5.1 がんばれほげおくん ver. 2

次のような「ほげおくん」がいる(ある?)(☆1). 詳しくは授業時に説明します.

- ほげおくんには、「**実行制御装置**」が備わっている。「命令」に従って以下の ALU やレジスタを制御し、四則演算などの作業をする。ただし、命令の意 図を理解する能力はなく、与えられた命令を一つずつ順番に実行することし かできない.
- ほげおくんには、「**ALU**」が備わっている。ALU は、様々な種類の演算を 実行できる(ほげおくんの場合、2つの数の加算、減算、乗算、除算の4通 りの演算ができるものとする).
- ほげおくんには、「**レジスタ**」が備わっている。演算に使う数や複雑な演算 の途中の値など、一時的に覚えておく必要のあるものはそこに書きとめない といけない(ほげおくんの場合, A, B, C, D という名前の4つのレジスタ を持っているものとする).
- ほげおくんの近くには、記憶装置「メモリ」がある。メモリは、様々な情報 を1つずつ格納できる箱がたくさんならんだ棚のようなものである。それ ぞれの箱には、区別のために番号「番地 (アドレス)」がついている.

☆ 1) 「ほげおくん」は、CPUの 仕組みを説明するために, その 機能を簡略化して表したもの. 第1回資料の「ほげおくん」が もとになっているので, そちら も参照しよう.

Q1. 上記のほげおくんに以下の命令が与えられたとして、次の間に答えなさい。 ただし、命令実行直前のメモリは右のような状態だったとする.

命令1 メモリの番地 4649 の内容をレジスタ A にコピーする

命令 2 メモリの番地 4650 の内容をレジスタ B にコピーする

命令3 レジスタ A の内容とレジスタ B の内容を加算し、結果をレジスタ A に書き込む

命令 4 レジスタ A の内容をメモリの番地 5963 に書き込む

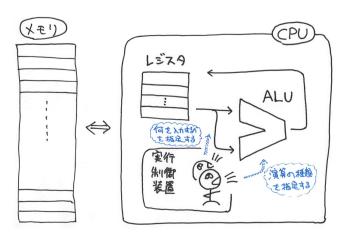
- (1) 命令 2 を実行した直後のレジスタ A.B の内容を答えなさい
- (2) 命令3を実行した直後のレジスタ A,B の内容を答えなさい
- (4) 命令 4 を実行した直後のメモリの番地 5963 の内容を答えなさい

メモリ

, = ,		
番地	内容	
:	:	
4649	200	
4650	-123	
:	:	
5963	2929	
:	:	
: 5963 :	: 2929 :	

CPU の構成 (1) – ALU, レジスタ, 実行制御装置 **★** 5.2

前節でいい加減に説明していた CPU の内部構成を、もう少しきちんと説明する $(\stackrel{\wedge}{\approx} 2)$.



☆ 2) 今回登場しているのは、 CPU の大事な構成要素のうち のまだ一部である. 残りはい ずれまた.

実行制御装置 「命令」を解読し、ALU やレジスタを制御して、指示された作業を 実行する.

ALU (☆ 3) 算術論理演算装置ともいう。2 つのデータの間で計算を行う。符号付 き/符号なしの整数の四則演算(☆4)の他,浮動小数点数(☆5)の四則演算 や, ビットパターンの論理演算(☆6)ができるようになっているものが多い.

レジスタ 演算に使う数や複雑な演算の途中の値などを一時的に保持しておく. 通 常, 1 つの CPU には数個から数百個のレジスタが備わっている。ほげおく ん ver.2 で登場しているような、一般的な計算等の用途に用いられる**汎用レ ジスタ**と呼ばれるものの他、特別の役割をもった専用のレジスタもある(次 回登場予定).

たとえば、前節の「命令3」の実行過程を考えると、CPU内では次のような処理 が行われる.

- 1. 実行制御装置が命令を受け取る.
- 2. 実行制御装置が命令を解読する、「レジスタ A の内容と B の内容を加算し、 結果をレジスタ A に書き込む」という命令なので、次のことを行う:
 - ALU への入力の一方にレジスタ A からの信号が、もう一方に B から の信号が入るように設定する
 - ALU の出力の信号がレジスタ A に入るように設定する
 - ALU が 4 種類の演算のうち加算を実行するように設定する
- 3. ALU が演算を実行し、結果をレジスタに書き込む(今の場合、加算を行っ てレジスタ A に書き込む).
- 4. 次の命令のための準備をする.

この節では、「命令はいったいどんなものなのか」、「命令はどこにあるのか」等の ことをごまかして説明している。その辺りの詳細については、次回解説予定。

- ☆ 3) Arithmetic Logic Unit
- ☆ 4) 加算, 減算, 乗算, 除算

☆ 5) 小数点の付いた数をコ ンピュータで扱う場合の形式 の一つ. C言語における float 型, double 型の値、詳しくは, 「数値計算法及び実習」で学ぶ かも.

☆ 6) 2 つのビットパターンの 間でビット毎に AND などの 論理演算を行うこと.

クロックとクロック周波数 **★** 5.3

論理回路では,論理ゲートたちがタイミングをあわせて 0/1 を切り替えないと, 動作がおかしなことになってしまう。そのため、CPUを含む論理回路の多くは、 **クロック**と呼ばれる,図に示すような規則的な信号に同期して(タイミングをあわ せて)動作するように作られている.

周期 全く同じ仕組みの CPU が2つあったとする と, **クロック周波数** (☆ 7) の大きい方が一定時間 」 内により多くの命令を実行できることになる.

Q2. クロック 1 周期の間に整数加減算を 1 回実行できる CPU(☆ 8) があ り、そのクロック周波数が $3.0 \text{GHz} = 3.0 \times 10^9 \text{Hz} \ (\text{\predext{g}}\ 9)$ だったとする。こ の CPU は、整数加減算を 1 秒間に何回実行できるか、

★ 5.4 メモリ

メモリ(主記憶装置)は、CPU の外にある装置だが、CPU の仕組みを語る上で 切り離せない大事な要素なので、ここで説明しておく、

メモリは、CPU が扱う様々な情報を格納し記憶しておく装置である。演算に用 いる数値などの様々なデータ (☆ 10) を記憶しておくだけでなく, CPU への命令を 集めた機械語プログラムを記憶しておくのにも用いられる(機械語プログラムにつ いては次回解説予定).

メモリは、一定ビット長の区画 (☆ 11) に区切られ、区画ごとに**番地 (アドレス**) と呼ばれる番号が割り当てられている。CPU は、メモリ上のどこの情報を読みだ すか/どこに書き込むかを、番地によって指定する.

コンピュータに接続される記憶装置には、メモリの他に、補助記憶装置と呼ばれ るものもある. これら2つは、それぞれ次のような特徴をもっている(☆12).

- メモリ CPU が情報を書き込んだり格納された情報を読み出したりするのにかか る時間が、補助記憶に比べて短い. その半面、記憶容量(格納できる情報の 量, すなわち格納できる 0/1 の数) は小さい. PC で一般的なメモリの場 合,コンピュータの電源を切ると内容が消えてしまう.
- 補助記憶装置 ハードディスクドライブ (HDD), SSD(Solid State Drive), フラッ シュメモリ等がこれにあたる。書き込み読み出しに要する時間がメモリに比 べて長いが、記憶容量は大きい、多くの場合、電源を切っても内容が保持さ れる.

Q3. あるコンピュータでは、メモリの番地を8ビットの符号なし整数で表 すという. このコンピュータのメモリは、最大でいくつのものを格納できる か(つまり、最大でいくつの区画に分けられるか). 番地を16ビット符号な し整数で表す場合はどうか.

☆ 7) 何かの量(電圧 etc.)が 時間とともに周期的に (同じパ ターンを繰り返すように)変動 する場合、繰り返し1回分の時 間を周期といい、1秒間の繰り 返し回数を周波数という. 周 期の単位は s (秒), 周波数の 単位は Hz (ヘルツ) である. 周期と周波数は逆数の関係に ある.

☆ 8) 全ての CPU が全ての命 令を 1 クロック周期の間に実 行できるわけではない. 1つ の命令の実行に何クロックも かかることも多い.

と読む. G (ギガ) や M (メ ガ) 等についてはいずれまた.

☆ 10) コンピュータは,数値 だけでなく文字や音声, 画像と いった情報も 0/1 のならびで 表す(そういうものをどうやっ て表すかについては,この授業 の後半に解説予定)ので、その ようなデータもメモリに格納 される.

☆ 11) PC 等の場合, 8 ビット 単位で区切られていることが 多い

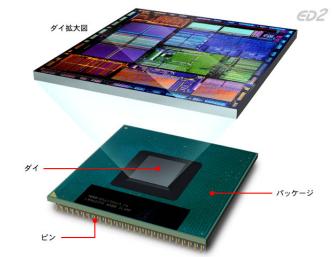
☆ 12) これらの装置の仕組み 等について, この授業の後半に 再度解説予定.

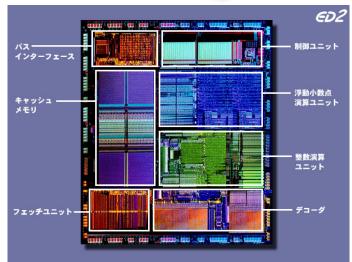
★ 5.5 CPU obanca

■CPU のつくり 現在一般的な CPU は, ダイと 呼ばれる小さく(指先にのる位のサイズ)薄いシ リコン基盤の表面に、トランジスタを用いた回路 を形成したものとなっている. 回路は印刷技術を 応用した方法で形成されている。その最小描画寸 法 (プロセスルール) は、2015 年現在、数十 nm $(1\text{nm} = 10^{-9} \text{ m} = 10 億分の 1 \text{ m})$ ほどである. 一つのプロセッサには、トランジスタなどの素 子が数百万個から数十億個集積されている.

参考 (wikipedia 「数量の比較」より):

- 髪の太さ: 0.1mm 弱 = 1×10^{-4} m 弱
- ヒト赤血球の直径: $6-8\mu m = (6-8) \times 10^{-6} m$
- HIV ウイルスの大きさ: $90\text{nm} = 90 \times 10^{-9}\text{m}$
- ■n ビット CPU CPU を分類する一つの目安 (厳 密な区分ではない)として、汎用レジスタや ALU 等のビット長がnであるとき、そのCPUはnビットの CPU である」ということがある。近年 の PC 向け CPU の場合, n は 32 または 64 で ある. 1980 年前後の PC には 8 ビットまたは 16 ビットの CPU が搭載されていた。ちなみに、任 天堂ファミリーコンピュータ (1983) は8ビット, スーパーファミコン (1990) は 16 ビット.
- ■CPU **の進歩** 例えば、インテル社のいくつかの CPU (多くは電卓や PC に搭載されてきた) は、 次のような進歩をしてきている (☆13).





年	名称	トランジスタ数	プロセスルール	クロック周波数
1971-	4004(\$\times 14)	2,300		数百 kHz
1978-	8086	29,000		数 MHz
1985-	i386	275,000	$1.5 ext{}1\mu\mathrm{m}$	数十 MHz
1993-	Pentium	3,100,000	$0.8 – 0.25 \mu \mathrm{m}$	60MHz 以上
2000-	Pentium4	42,000,000	130-65 nm	1GHz 以上
2006-	Core 2 Duo	291,000,000	$65-45\mathrm{nm}$	1–3GHz 程度
2008-	Core i7	約 1,000,000,000	45–14nm	3–4GHz 程度

CPU を含めた半導体集積回路の進歩に関しては、「ムーアの法則」というものがよく知ら れている。これは、インテル社の G.E.Moore が 1965 年に発表した、「半導体集積回路に集 積されるトランジスタ数は 18 から 24 ヶ月で倍増する」という経験則である (☆ 15). 何十 年も成り立ってきた(ねずみ算式にものすごい勢いで集積度が上がり続けてきた)が、そろ そろ転換点を迎えつつあるかもしれない.

☆ 13) CPU には, 同じ名称 (ブランド名) でも細かい構成 が異なる型がいくつもある. この表には、それらのうちの 代表的なものについての値を 示している.

☆ 14) 世界初のマイクロプロ セッサのひとつ.

☆ 15) 例えば 2 年で倍なら, 10年で $2^5 = 32$ 倍, 20年で $2^{10} = 1024$ 倍、30 年で $2^{15} =$ 32768 倍