

# CPUのしくみ

(2)記憶装置、クロック、  
プログラム内蔵方式

人類文明継続装置  
輪廻 ヒロ

監修: 矢口 裕明  
(博士(情報理工学))

# はじめに

前回は二進数と演算器の基本を説明しました。  
今回は記憶装置とクロック、  
プログラム内蔵方式についてです。

# 記憶とクロック

前回は $a+b$ はできたけど、  
 $a+b+c+d$ はどうやったらいい？

どこかに $x$ を作って、  
 $x=a+b$ ;  $x=x+c$ ;  $x=x+d$ ; とする。  
 $x$ は計算結果を記憶したもの

時間の概念=過去と現在を区別したい  
拍子をとって管理する=クロック

# フリップフロップ回路

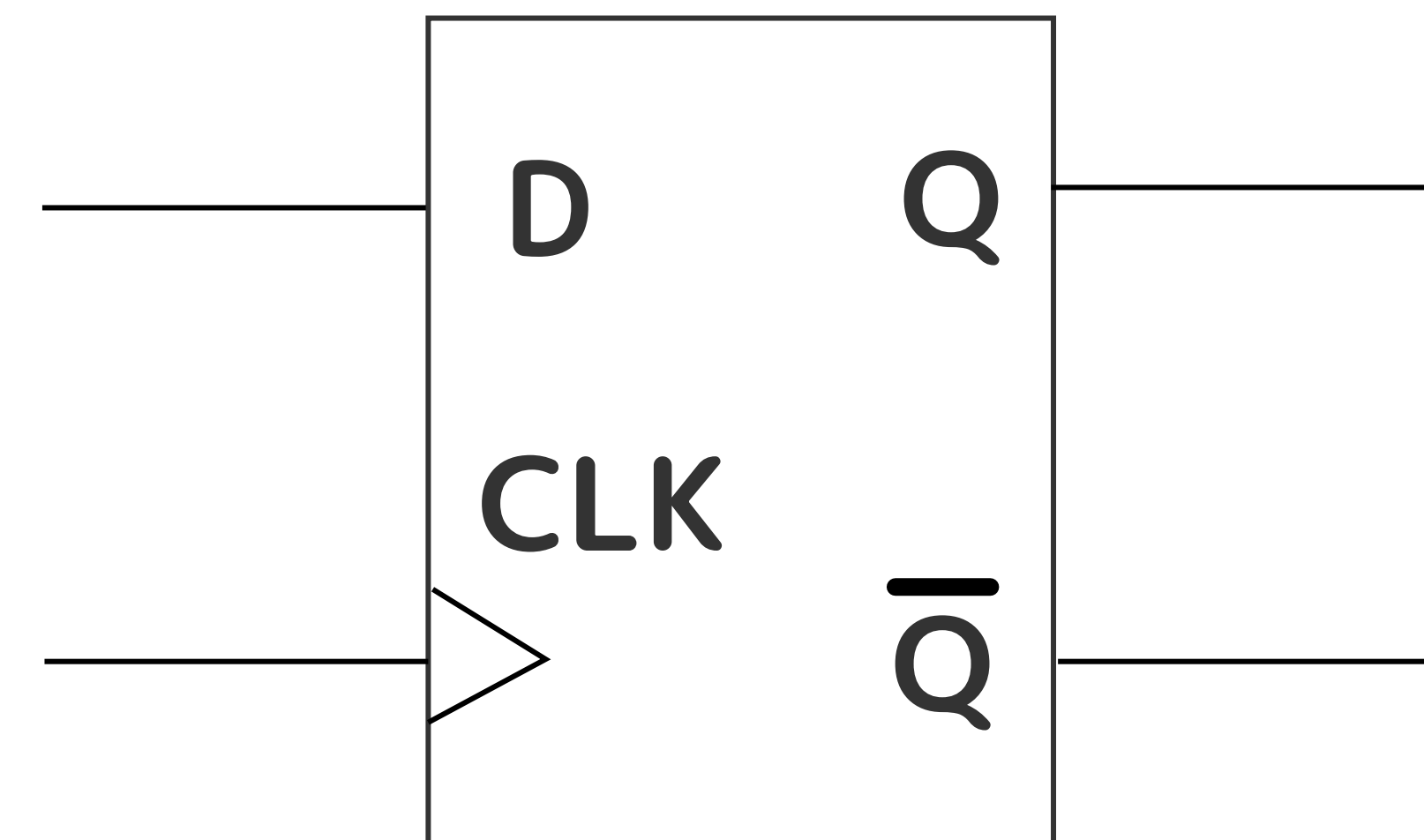
加算器は入力電圧だけで出力が決まる

=組み合わせ論理回路

入力電圧と過去に出力した結果、

つまり入力の順序で出力が決まる=順序回路

Dフリップフロップ回路



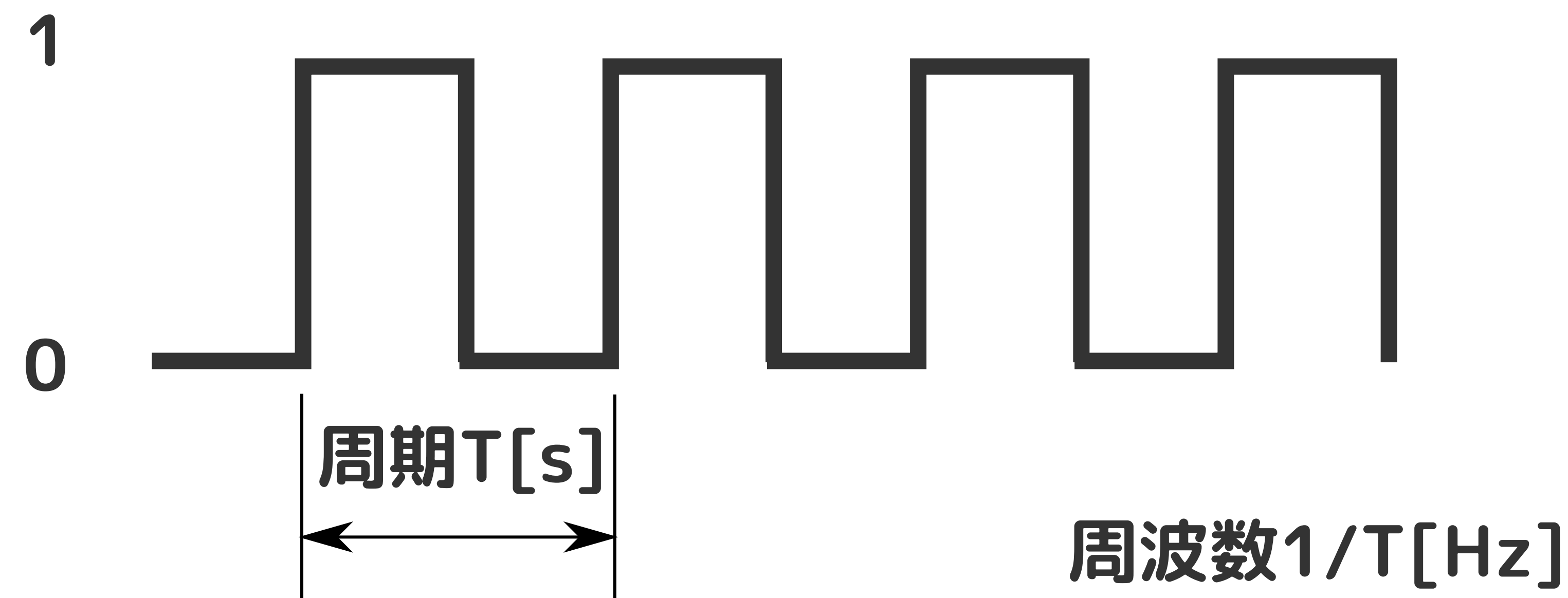
CLKにはクロックが入力される。  
QはCLKが1になった瞬間の  
Dの値を出力し続ける。

$\bar{Q}$  : NOT Q

これはつまり1bitの記憶回路である。

# クロック

一定周期で1,0が入れ替わる。  
数値が入れ替わるタイミングで  
記憶を更新することで、  
順番に計算を進めることができるように。



# 記憶装置の構造

0と1が示せればなんでもよい。

フリップフロップを並べる: SRAM

コンデンサの電荷: DRAM

磁気テープ、フラッシュメモリ、色の濃淡など。

記憶されている情報はたくさんある。

どこに記憶されているかを示す情報が必要。

アドレス(住所)と呼ばれる方式で、

先頭から何バイト目かを番地と読んでいる。



# ビット、バイト、ワード

ビット:1桁の二進数

バイト:8桁の二進数=2桁の16進数

ワード:その計算機が一度に扱える桁数(全部違う)

二進数がたくさん並ぶことになるので、

8桁をまとめて1バイトと定義する。

2桁の16進数で表現する。

複数のバイトにまたがるデータ(16桁など)を

1バイト区切りで下の桁から順番に詰める

=リトルエンディアン

上の桁から詰める=ビッグエンディアン

# 16進数

16で繰り上がる数の表現

0...9, A, B, C, D, E, Fで一桁を表記。

区別のために0xを先頭につけることが多い。

一桁の16進数は4桁の2進数を表すことができる。

よく使う16進数

$0xFF = 255$

$0xFFFF = 65535$



# プログラム内蔵方式

計算の仕方そのものを変更したい

->記憶装置の中に計算に必要な情報を書いておく。

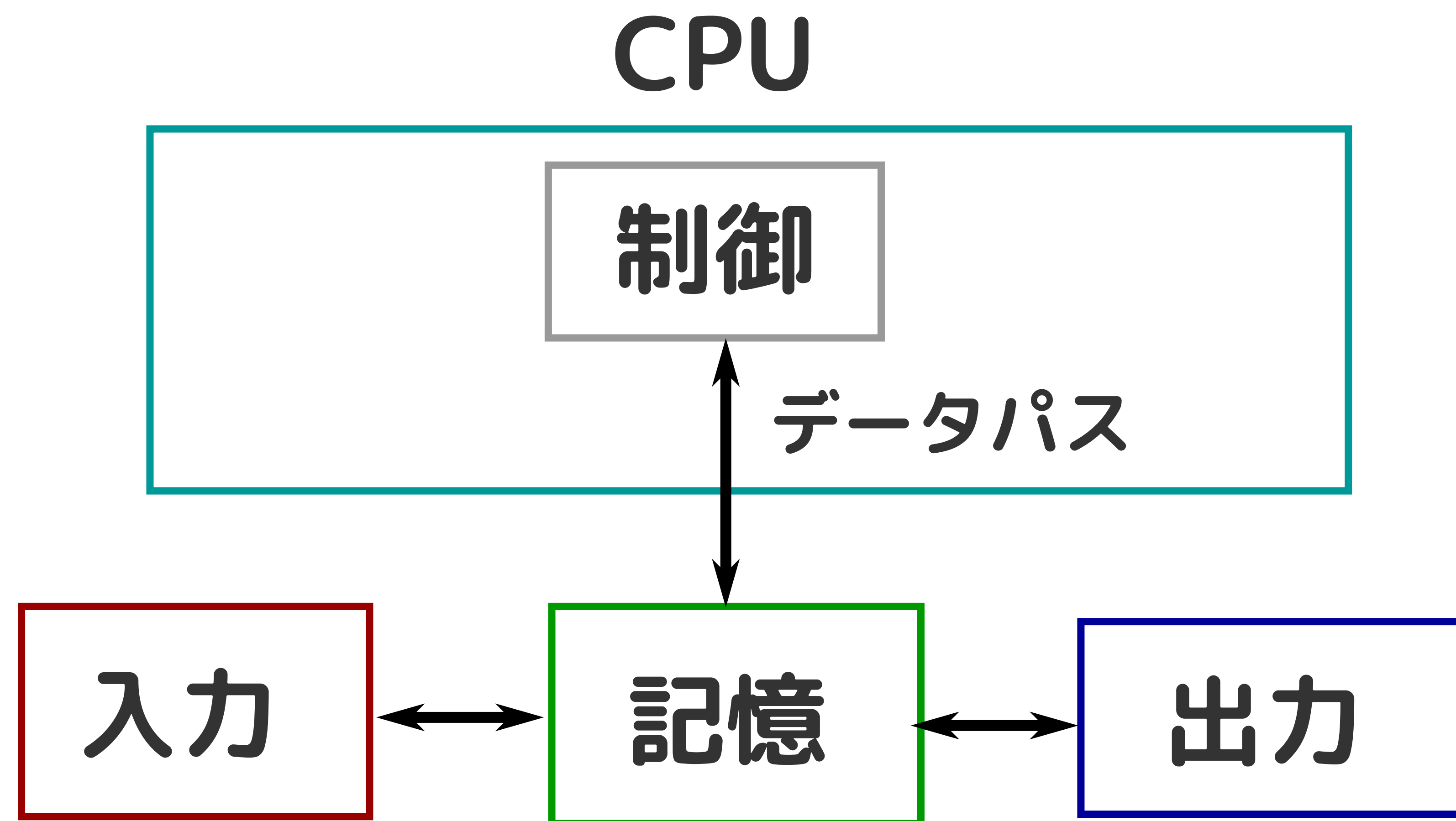
演算の種類、使用するデータ、など。

記憶装置の中ではこれらは区別されない。

つまり、計算の仕方そのものも

計算によって書き換えることができる。

# コンピュータの構成



※CPU=演算+制御とすることもあります。  
ここでは[1]に準じています。

# CPUの構成

演算装置:加算器など

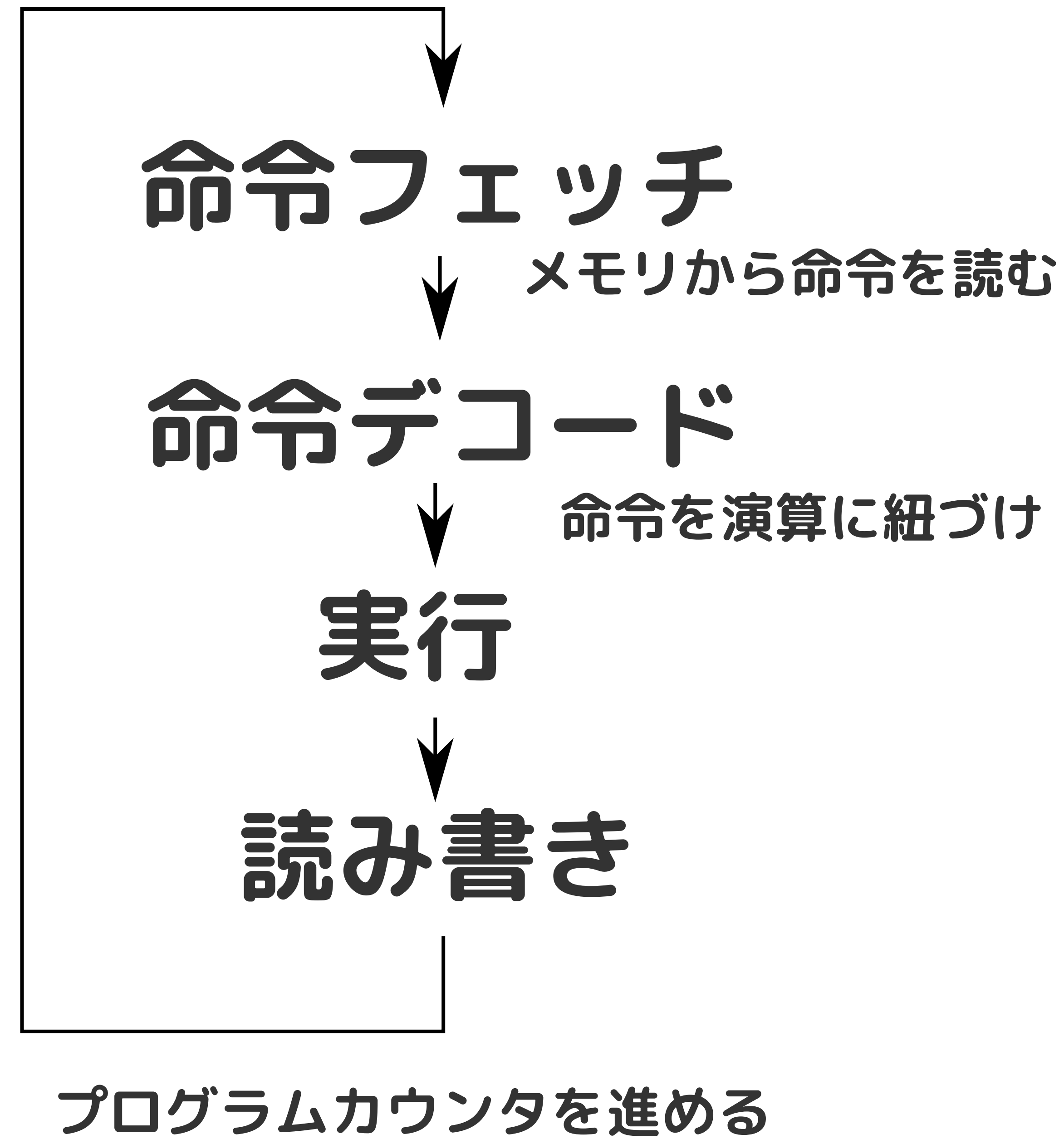
レジスタ:一時記憶

プログラムカウンタ:  
次に実行する命令が  
書き込まれている番地を記憶

デコーダ:  
命令と実際の演算を紐づける

※実際にはもっとたくさんあります

# 計算の進み方



# CPUの命令

CPUの計算を示すデータを命令と呼ぶ。  
全ての命令一覧を命令セットと呼ぶ。  
命令の構成は  
(演算の種類+演算に必要なデータ)

例えばこんな演算があるCPUなら

メモリ読み込み(LD) : 0b0000 + メモリの番地

メモリ書き込み(ST) : 0b0001 + メモリの番地

加算(ADD) : 0b0010 + レジスタ番号(a, b)

レジスタ移動(MOV) : 0b0011 + レジスタ番号(a, b)

という感じになる(適当に決めました)。

この時の命令全体の桁数=CPUのビット数=そのCPUの1ワード



# 機械語

CPUに対する命令を  
直に書き下したバイナリデータを  
機械語と呼ぶ。

LD, ST, MOV, ADD, などの  
人間にも解釈しやすい表現を  
アセンブリ言語と呼ぶ。

より複雑な処理を表現できるよう  
構文を工夫したものを  
高級プログラミング言語と呼ぶ。

# 参考文献

- [1] David Patterson, John L. Hennessy, 成田光彰(訳).  
「コンピュータの構成と設計 MIPS Edition 第6版 [上]  
～ハードウェアとソフトウェアのインタフェース～」, 日経BP, 2021.
- [2] David Patterson, John L. Hennessy, 成田光彰(訳).  
「コンピュータの構成と設計 MIPS Edition 第6版 [下]  
～ハードウェアとソフトウェアのインタフェース～」, 日経BP, 2021.
- [3] 渡波 郁. 「CPUの創り方」, 毎日コミュニケーションズ, 2003.
- [4] 西山 悠太郎, 井田 健太.  
「RISC-VとChiselで学ぶ はじめてのCPU自作  
オープンソース命令セットによるカスタムCPU実装への第一歩」, 技術評論社, 2021.
- [5] Takenobu Tani. 「プログラマーのためのCPU入門  
CPUは如何にしてソフトウェアを高速に実行するか」,  
ラムダノート株式会社, 2023.
- [6] 矢沢 久雄. 「コンピュータはなぜ動くのか 第2版  
知っておきたいハードウェア&ソフトウェアの基礎知識」, 日経BP, 2022.
- [7] 馬場 敬信. 「算数で読み解くコンピュータの仕組み」, 技術評論社, 2022.

# ライセンスについて

本文書のライセンスはクリエイティブ・コモンズ表示4.0 CC BY 4.0です。

© 2023 クシナダ機巧株式会社

フォントライセンス

- Rounded M+: M+ FONTS LICENSE