自作CPUを語る会

2023/06/11

@kanade_k_1228

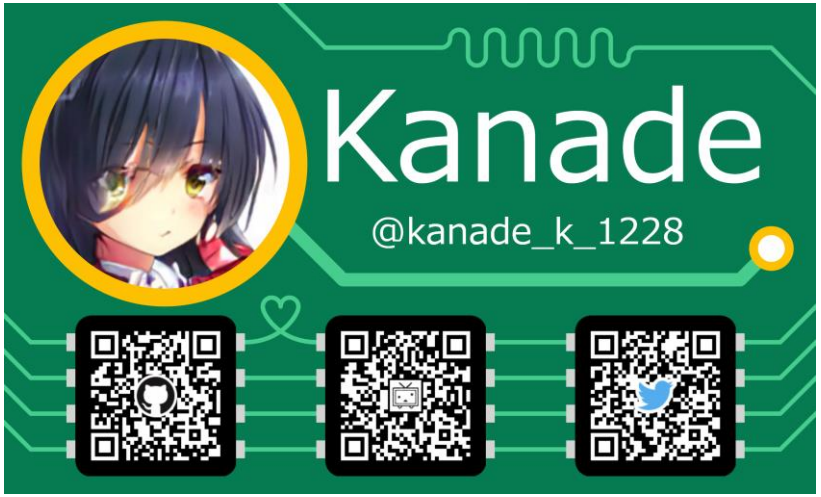
自作CPUを語る会：スライドはここにあります

https://making-cpu.github.io/archive/

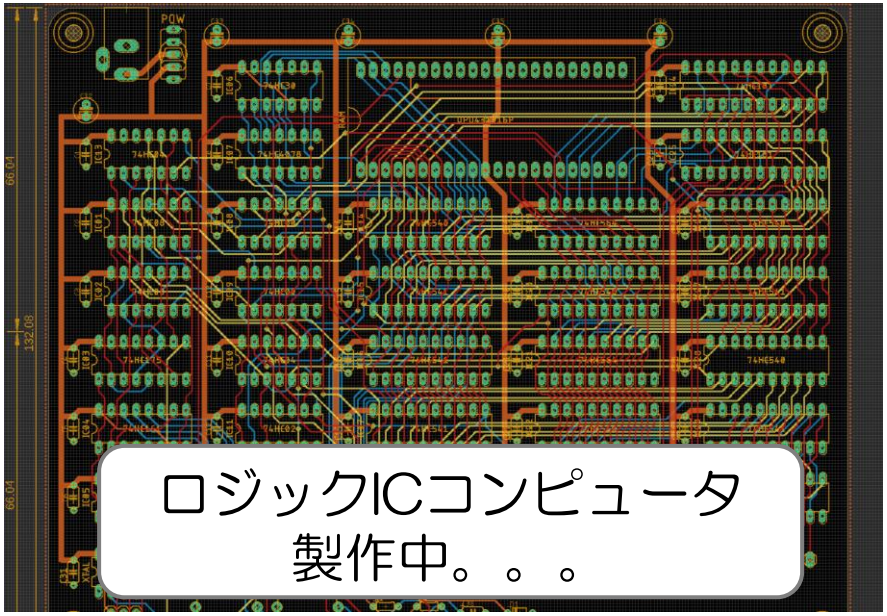
アンケート：CPUを

作ったことがある	作っている	作ったことがない

自己紹介



リレーコンピュータ
@ Maker Faire Tokyo 2022



ロジックICコンピュータ
製作中。。。

リレーコンピュータの開発史

沼の入り口 2017年

低レイヤの沼

混沌

忘却

再起

完成 2022年

本日の会

Канадэ
@kanade_k_1228

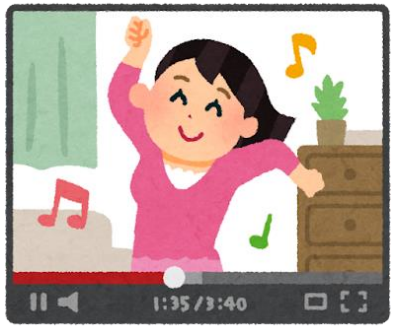
第1話　低レイヤにようこそ！！
第2話　初めてのCPUっ！
第3話　ドキドキ！7400っ！？
第4話　ワクワク！7474っ！？
第5話　ハザード
第6話　憤怒
第7話　怠惰
第8話　絶望
第9話　破壊
第10話　忘却
第11話　再起
最終話　FPGA

午後9:06 · 2023年2月16日 · **2,413** 件の表示

なぜ人はCPUを作りたくなるのか？

コンピュータという謎

- 沼の入り口
- 低レイヤの沼
- 混沌
- 忘却
- 再起
- 完成



電気を使って魔法を見せる
謎の物体「コンピュータ」

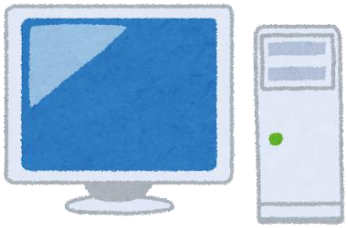
コンピュータってどうやって
動いてるの？

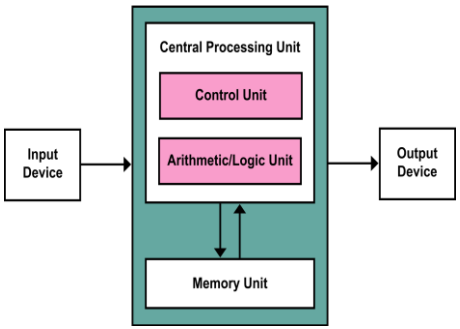
回路図レベルで理解したい

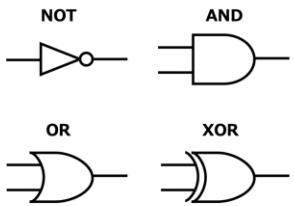
具体的に電気がどう流れてるか
物理的な部分まで知りたい

抽象化を剝がしていく

- 沼の入り口
- 低レイヤの沼
- 混沌
- 忘却
- 再起
- 完成







コンピュータは、

いろいろな装置から
できている

それぞれの装置は、

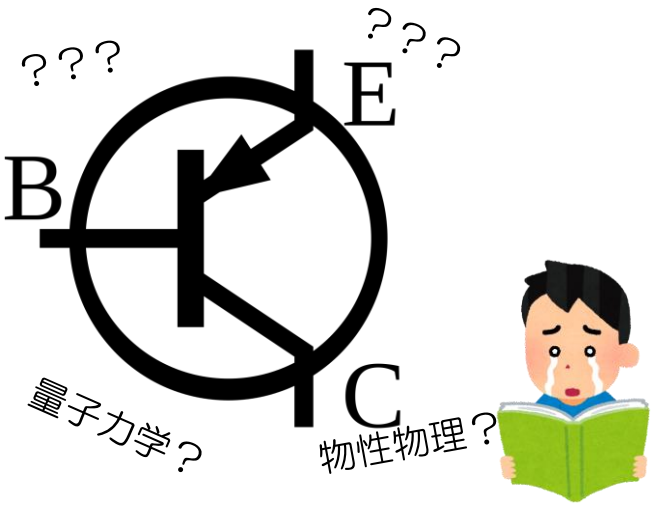
論理回路で
できている

論理回路は、

トランジスタという謎

- 沼の入り口
- 低レイヤの沼
- 混沌
- 忘却
- 再起
- 完成

トランジスタでできている



大学生でも難しい…

定性的な説明はわかっても、
定量的に理解するのは難しい

トランジスタという
ブラックボックスが残る

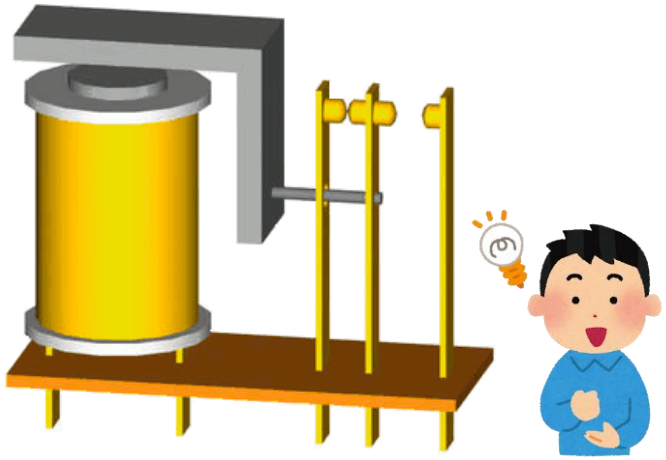
「当時の自分」に向けて
トランジスタを解説する
動画を作ってたりします

https://www.nicovideo.jp/user/73765122/series/329143

リレーコンピュータ

- 沼の入り口
- 低レイヤの沼
- 混沌
- 忘却
- 再起
- 完成

リレー



小学生でもわかる！

Harry Porter's Relay Computer

Last updated: 17 November 2009 (circuit diagrams added)

https://web.cecs.pdx.edu/~harry/Relay/

リレーでも、コンピュータが作れるじゃん！

リレーを使えば
全くブラックボックスがない
コンピュータが作れる！！！

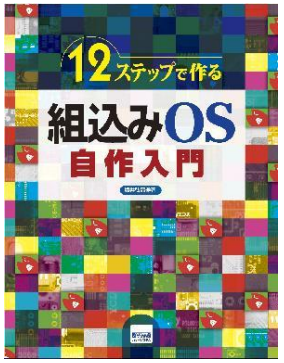
リレーコンピュータを設計するために

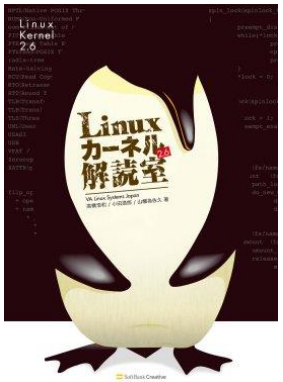
- 沼の入り口
- 低レイヤの沼
- 混沌
- 忘却
- 再起
- 完成

コンピュータを設計しよう！
→　どんなプログラムを走らせる？
→　OS走らせてみたい

プログラムを走らせるには
→　アセンブラは欲しいよね
→　Cコンパイラもあったらいいよね

- 低レイヤの知識が広がっていく
- 妄想がふくらむ
- やってみたいことが増えていく



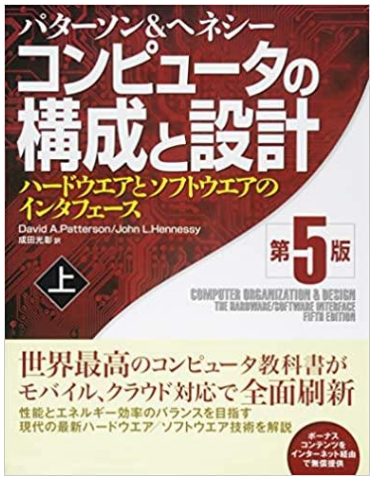


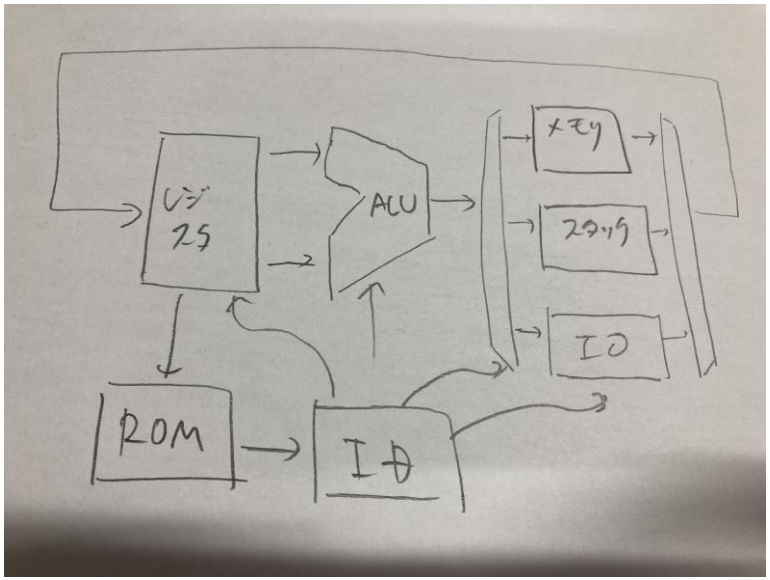
コンピュータを設計する

- 沼の入り口
- 低レイヤの沼
- 混沌
- 忘却
- 再起
- 完成

普通のコンピュータはどうなってる？

→ シングルサイクルの
単純なCPU





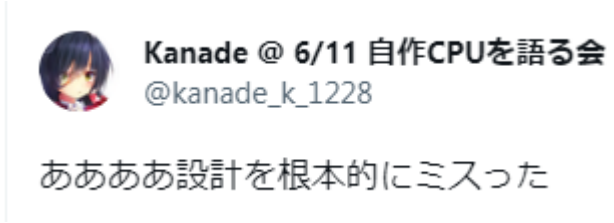
複雑になる設計　積みあがったゴミ

沼の入り口
↓
低レイヤの沼
↓
混沌
↓
忘却
↓
再起
↓
完成

脳内エミュレータの
キャパを超えた

実際に組んでみると、
動かない

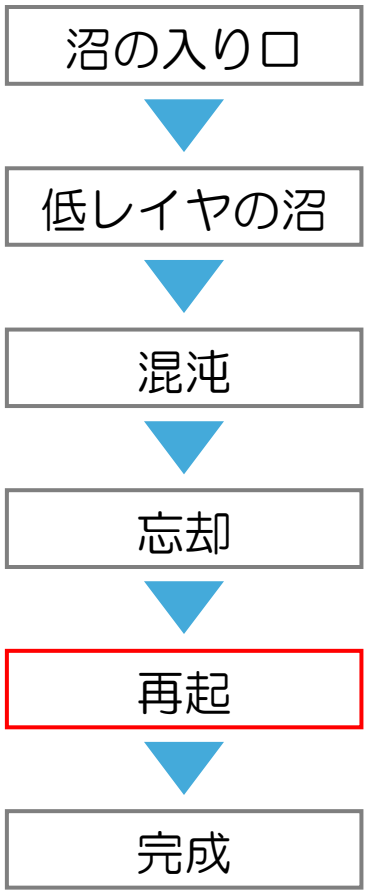
しばらく放置される



↑発掘された当時の発狂

↑発掘されたゴミになった基板

新たなアーキテクチャ

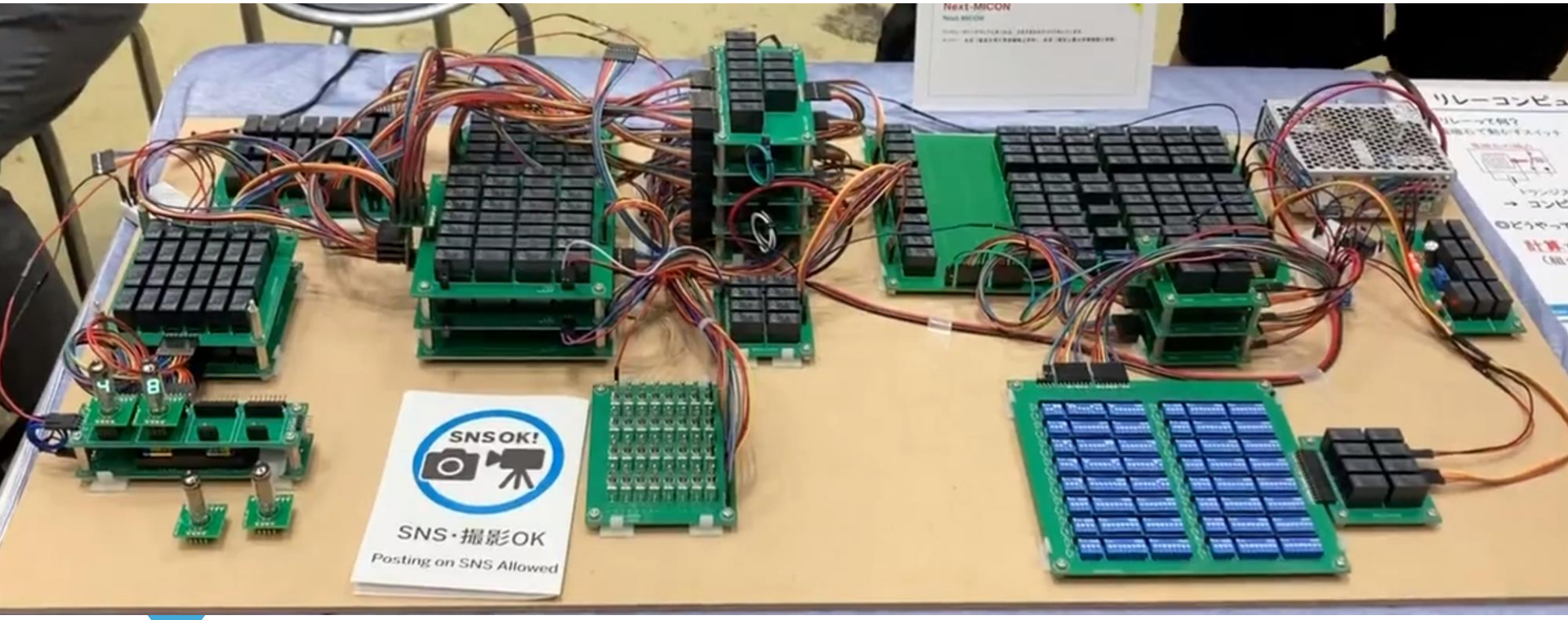


特殊なハードウェアには特殊な設計思想が必要

簡潔なアーキテクチャに（後述）

次こそはうまくいきそう

リレーコンピュータ：RK8R



完成

@ Maker Farie Tokyo 2022

動画：https://www.youtube.com/shorts/h44TmS2Qn1c

「自作CPUを語る会」開催の経緯

- 沼の入り口
- 低レイヤの沼
- 混沌
- 忘却
- 再起
- 完成
- 本日の会

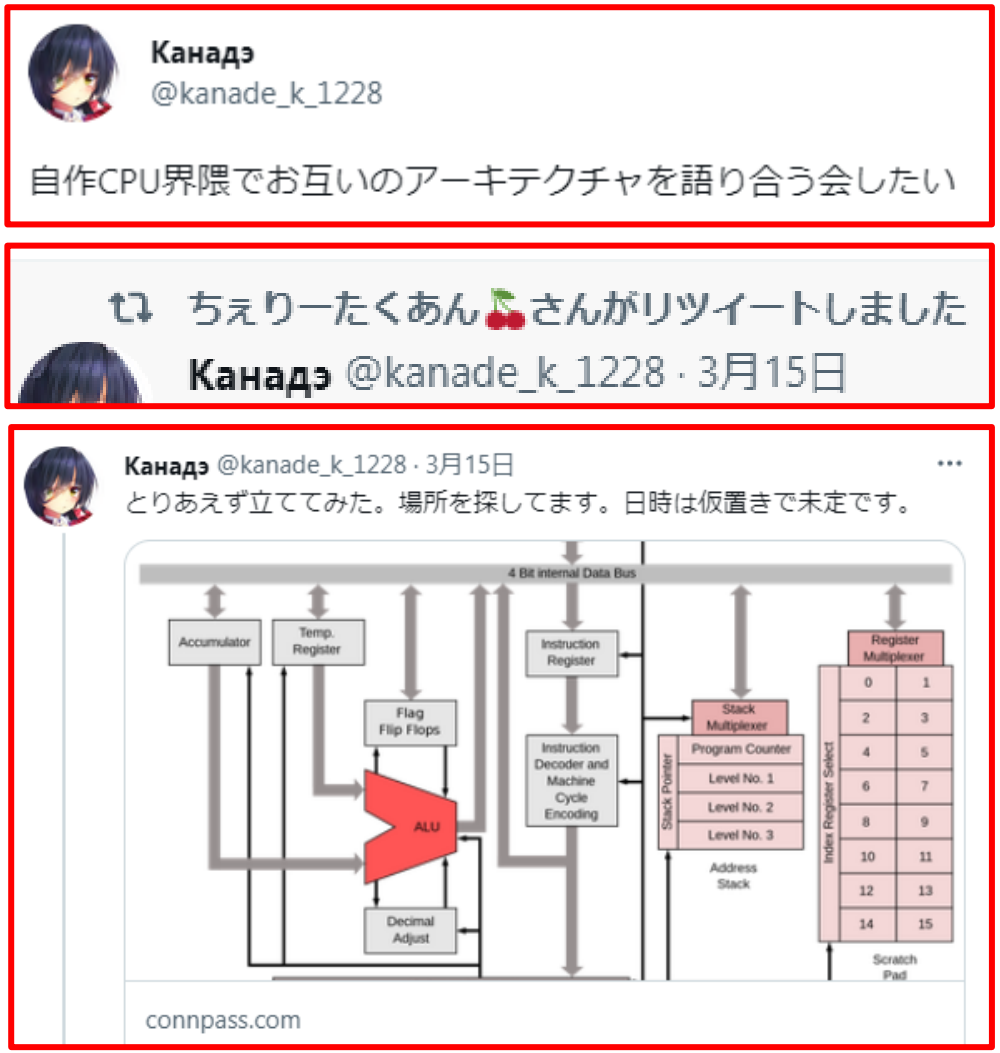
それぞれのCPUに、それぞれの設計思想がある

→ 機械語が書けるレベルまで細かく
アーキテクチャを理解したい

展示会では、そこまで語り合うのは難しかった

→

「自作CPUを語る会」開催の経緯



↑当初想定してた会場

「自作CPUを語る会」開催の経緯

発表枠（発表15分+質問5分）
無料 3/3人
先着順

現地参加
無料 41/50人
先着順

オンライン聴講（試験的）
無料 131人
参加者数

現地参加（展示ブースあり）
無料 3/6人
先着順

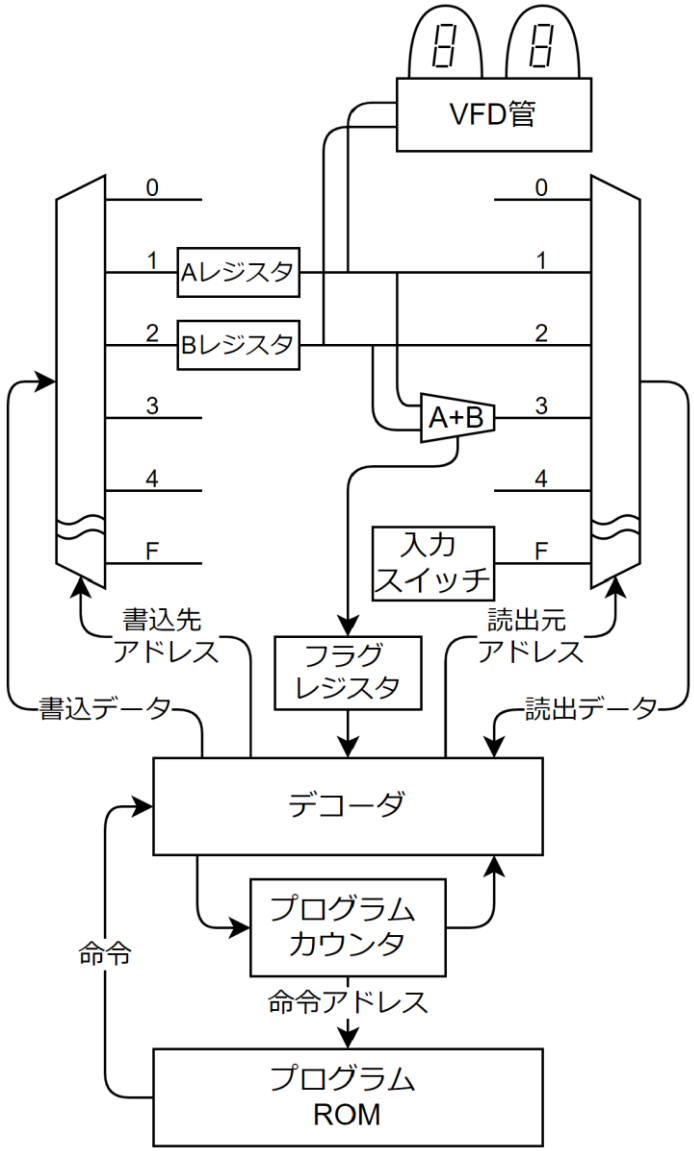
cybozu様ありがとうございます！！！

リレーコンピュータ RK8R のアーキテクチャ

2023/06/11

@kanade_k_1228

RK8Rのアーキテクチャ



R(RISC) K(Kanade) 8(8 bit) R(Relay)

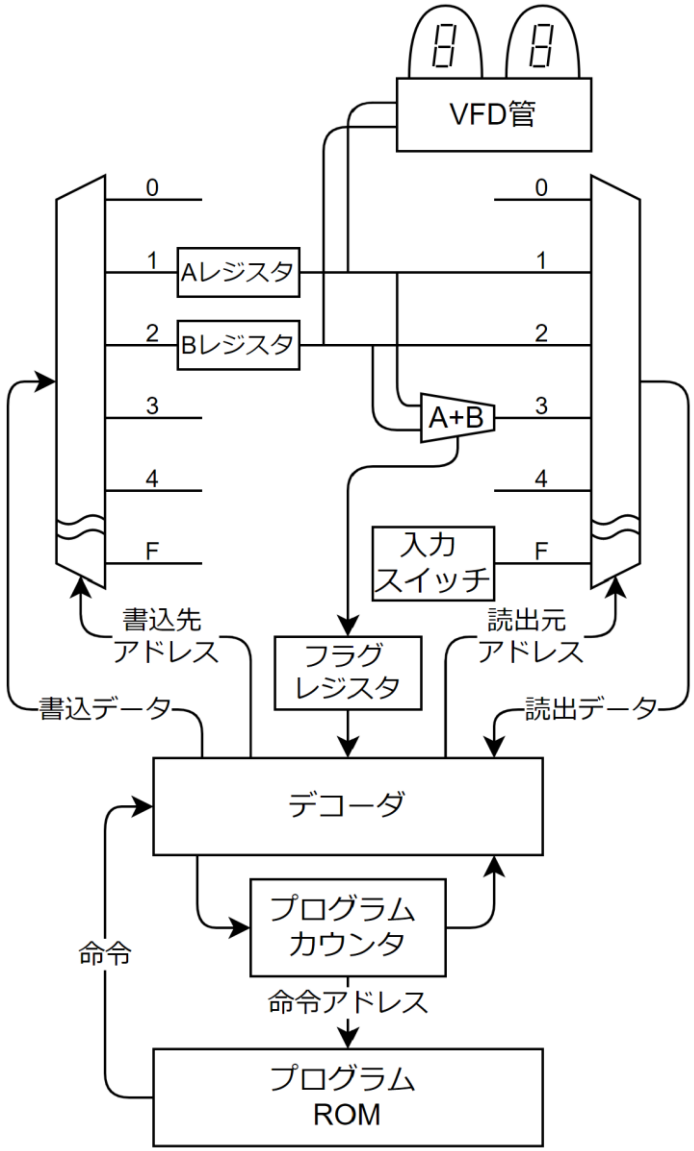
特徴

- 命令が4種類だけ
- クロック周波数が高い

諸元

- クロック周波数：40 Hz
- レジスタのビット数：8 bit
- メモリ空間：256 × 8 bit
- 命令ビット長：18 bit
- ROM 空間：256 × 18 bit
- 入力装置：トグルスイッチ × 32
- 出力装置：7セグVFD管 × 4

命令セット

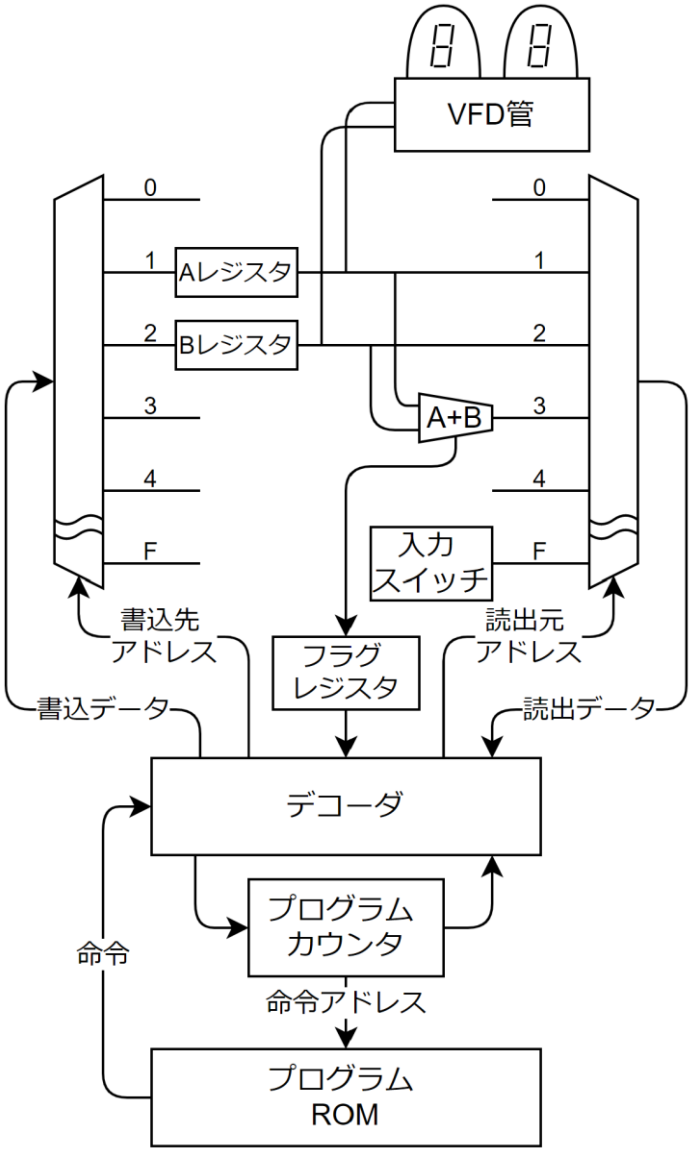


命令	[0:1]	[2:7]	[11:17]
MOV	10	読み出し先	書き込み先
LOAD	11	即値	書き込み先
JMPIF	01	分岐条件	分岐先アドレス
NOP	00		

命令は４種類だけ！

命令長は 18 bit
- 2 bit のオペコード
- 8 bit の読み取り情報
- 8 bit の書き込み情報

MOV命令

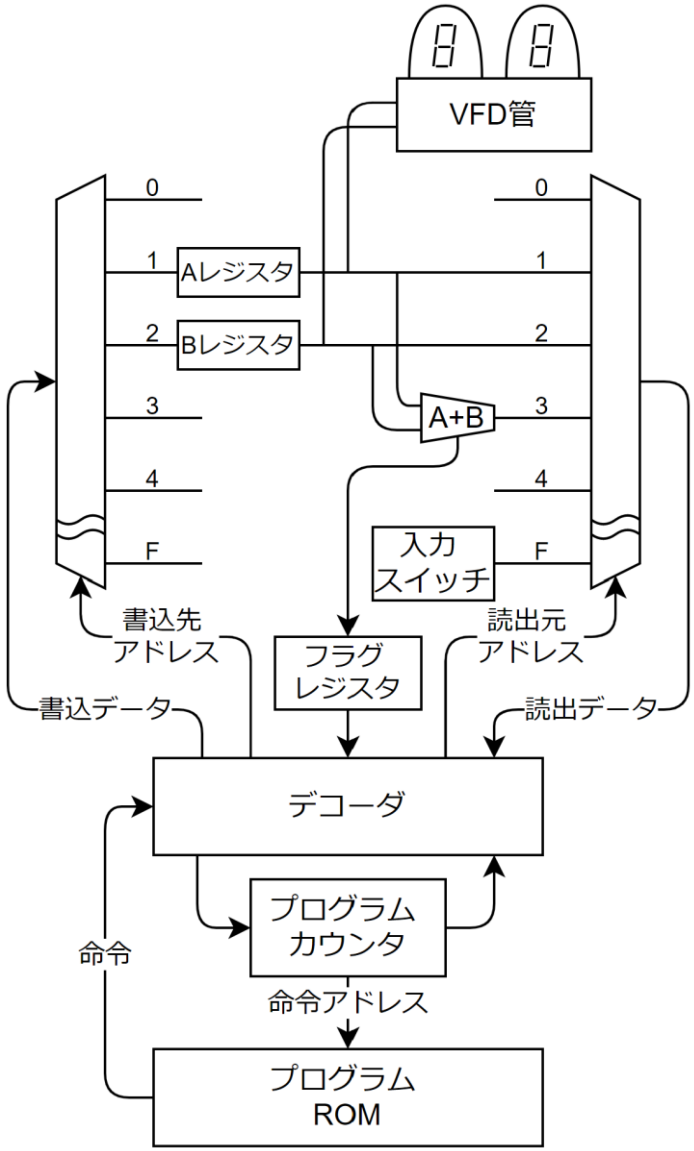


mov	Source	Destination
10	xxxx_xxxx	xxxx_xxxx

RS（Source Register）から、
RD（Destination Register）に値をコピー

演算命令は存在しない！
→　特定のレジスタが演算結果

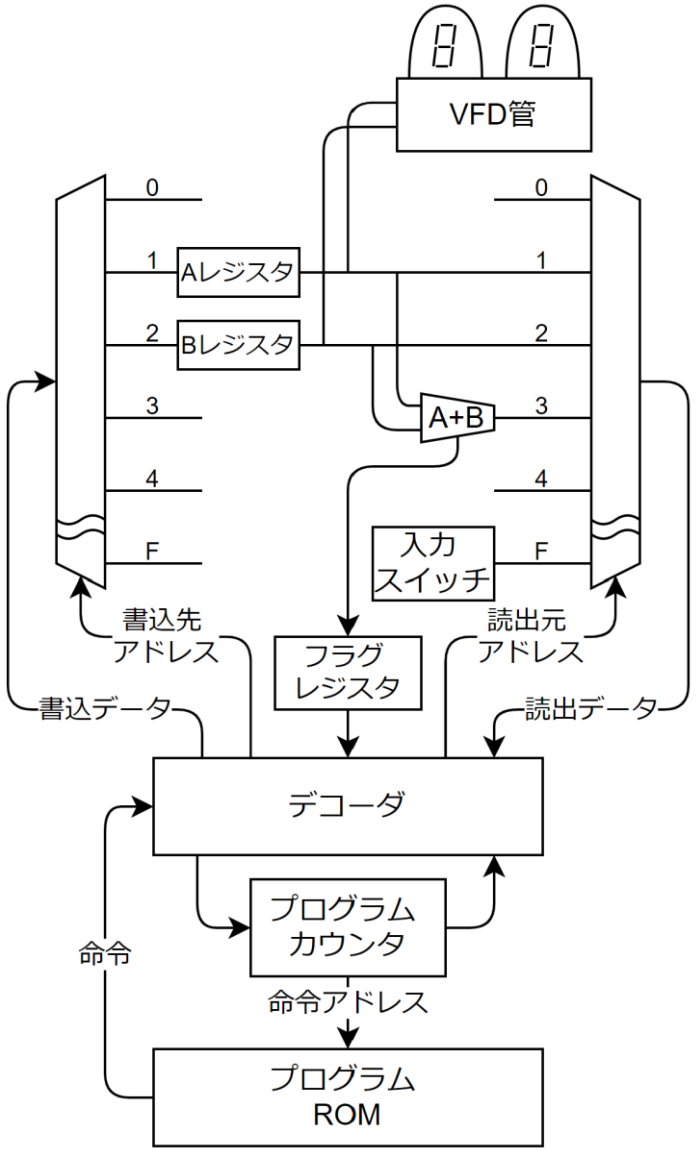
LOAD命令



load	Immidiate	Destination
11	xxxx_xxxx	xxxx_xxxx

Immidiate の値を
RD (Destination Register) に書き込む

JMPIF命令

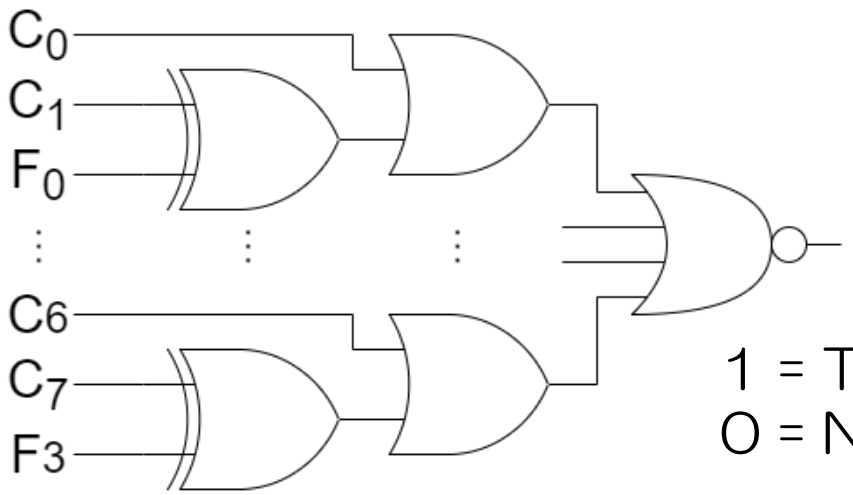


jmpif	Condition	JmpTo
01	xx_xx_xx_xx	xxxx_xxxx

4個の Flag と Condition の値に応じて、
プログラムカウンタの値を JmpTo にする

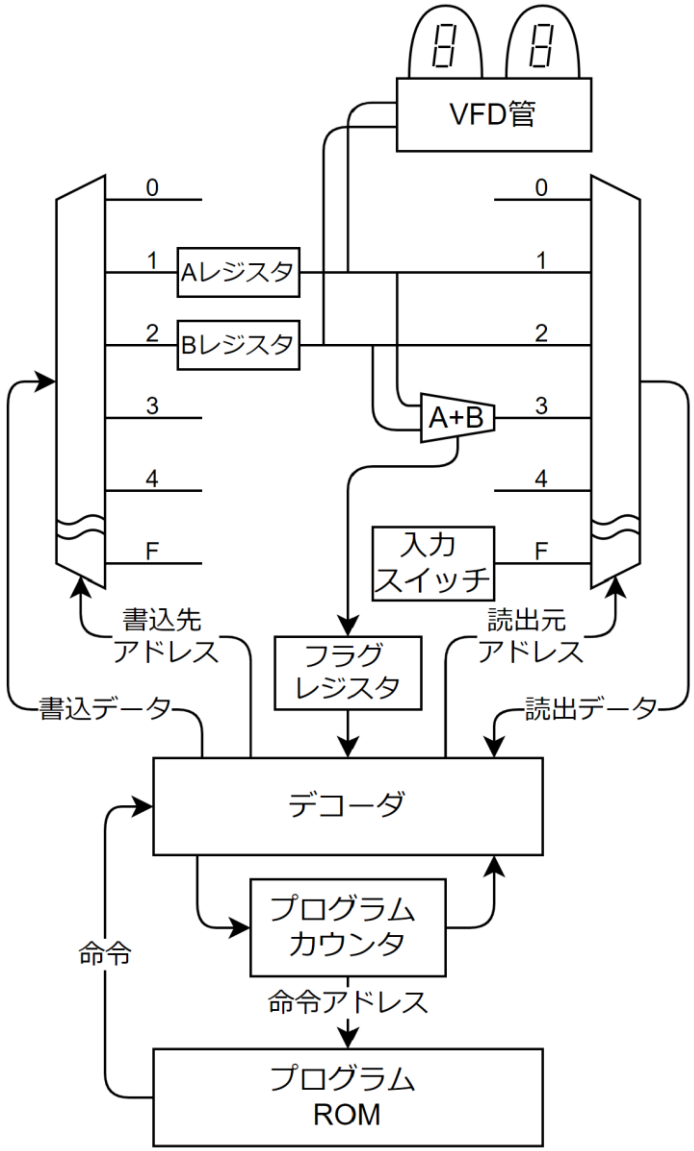
条件の読み方

- C_{2i}：フラグ F_iを使うか？
- C_{2i+1}：フラグ F_i がどちらと一致するか？



1 = Taken
0 = Not Taken

NOP命令

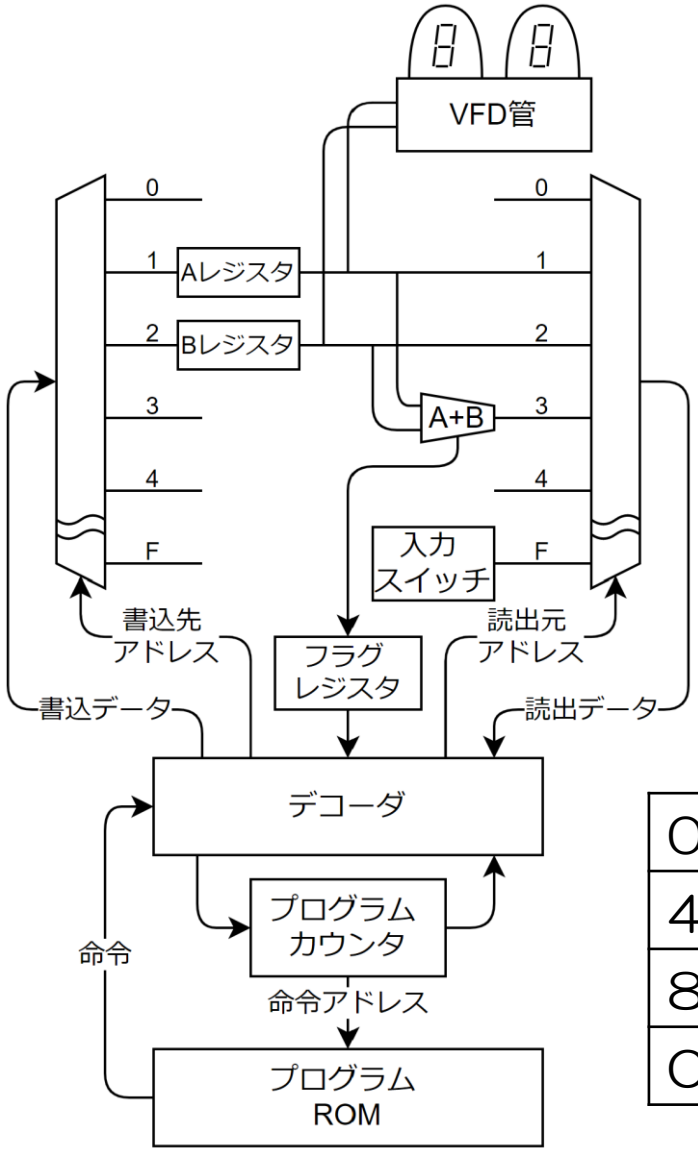


nop 00	xxxx_xxxx	xxxx_xxxx

１クロックの間、何もしません。

未使用の ROM エントリはゼロにしておく。

レジスタ空間



レジスタにいろいろな演算器が接続

MOV命令がいろいろな演算命令に化ける

例：3番のレジスタを読みだすと、
加算 A+B の結果が得られる

0: 0	1: Aレジスタ	2: Bレジスタ	3: A + B
4:	5: ~A	6: A << 1	7: A >> 1
8:	9: A ^ B	A: A & B	B: A | B
C: SW0	D: SW1	E: SW2	F: SW3

プログラムの例：フィボナッチ数列

$$f_0 = 0, \qquad f_1 = 1, \qquad f_{n+2} = f_{n+1} + f_n$$

0	11_00000001_00000001	load 1 A	A = 1
1	11_00000000_00000010	load 0 B	B = 0
2	10_00000011_00000010	mov A+B B	B = A+B
3	10_00001001_00000010	mov A^B B	B = A^B
4	10_00001001_00000001	mov A^B A	A = A^B
5	10_00001001_00000010	mov A^B B	B = A^B
6	01_00000000_00000010	jmpif true 2	goto 2

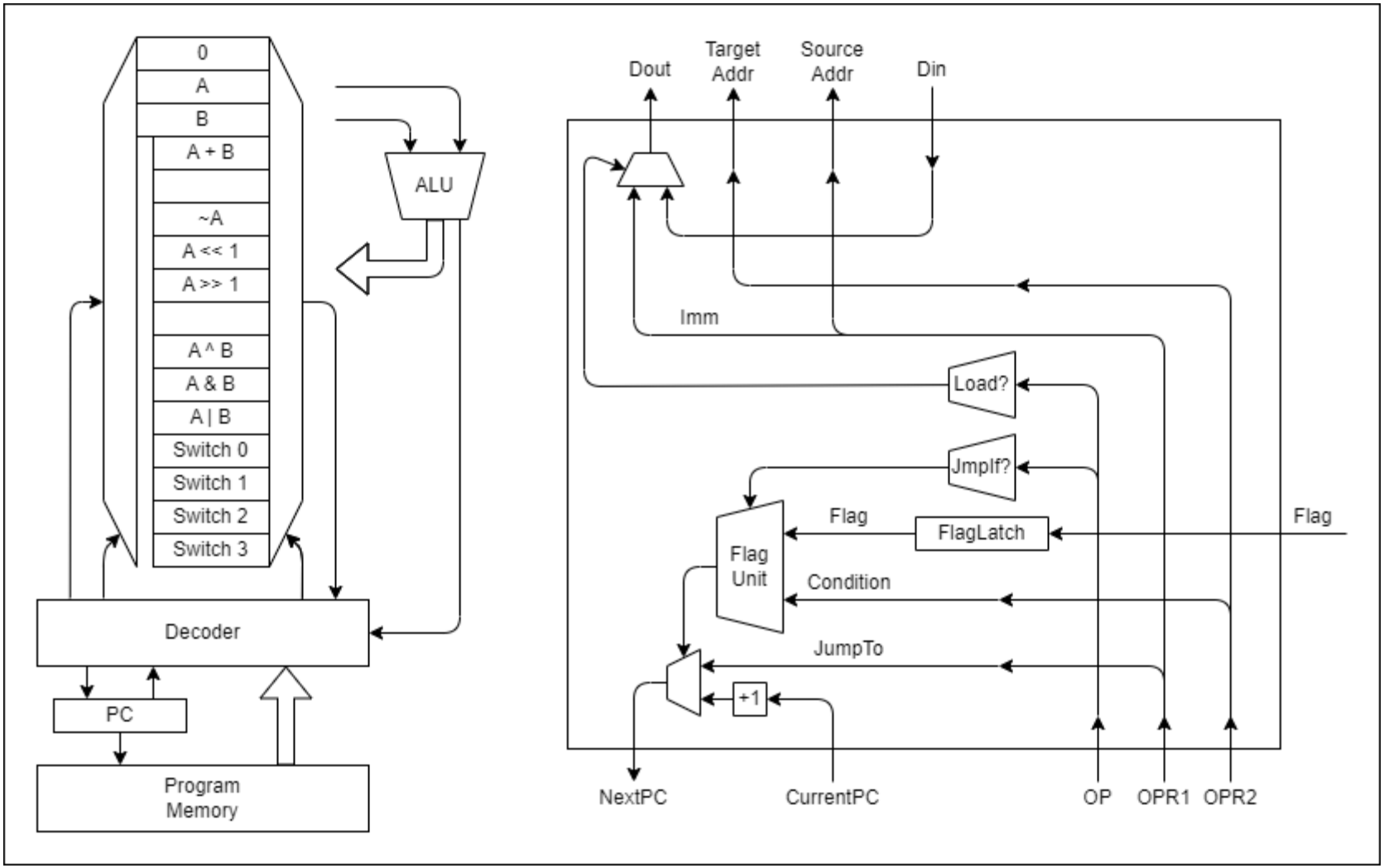
XOR Swap

	$f_1 = 1$ $f_0 = 0$	$f_2 = 1$ $f_1 = 1$		$f_3 = 2$ $f_2 = 1$		$f_4 = 3$ $f_3 = 2$		$f_5 = 5$ $f_4 = 3$		$f_6 = 8$ $f_5 = 5$		$f_7 = D$ $f_6 = 8$	
A	1	1	1	1	2	2	3	3	5	5	8	8	D
B	0	1	1	2	1	3	2	5	3	8	5	D	8

ありがとうございました！

補足資料

デコーダ回路


0
A
B
A + B
~A
A << 1
A >> 1
A ^ B
A & B
A | B
Switch 0
Switch 1
Switch 2
Switch 3
ALU
Decoder
PC
Program Memory
Dout
Target Addr
Source Addr
Din
Imm
Load?
JmpIf?
Flag
FlagLatch
Flag
Flag Unit
Condition
JumpTo
+1
NextPC
CurrentPC
OP
OPR1
OPR2

次のリレーコンピュータ

- 大きなのＭＤＦ板に基板を貼り付けている
  → 持ち運びが絶望的
  → 次回作は人が電車で運べるようにする

- 萌えポイントを増やす
  - VFD管に注目されがち。。。
  - 動作を表す麦球をつける
  → 音＋光で映える