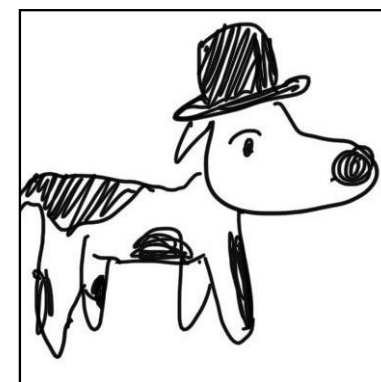


Brainfuckが動くCPUを作りたい！

チズチズ (Yuma Ochi) 6月2日

越智 優真 (Yuma Ochi)

- 木更津高専 情報工学科 3年
- 専門: 深層学習、自動運転モデル
- 趣味: クラスメイトと自作CPU、CTF
- 担当: 回路設計、なんでも



@chizu_potato

Tsunaneko

- 木更津高専 情報工学科 3年
- 趣味：自作CPU
- 担当：RAM、クロック

doit^6

- 木更津高専 情報工学科 3年
- 趣味：？
- 担当：デコーダ設計、回路設計

おことわり

- ハードウェアは完成しませんでした (部分までしか作れていない)

今作ろうとしてるCPU

- 命令が8個のみの難解プログラミング言語をインタプリタ・コンパイラを介さず直接実行する独自ISAのCPU

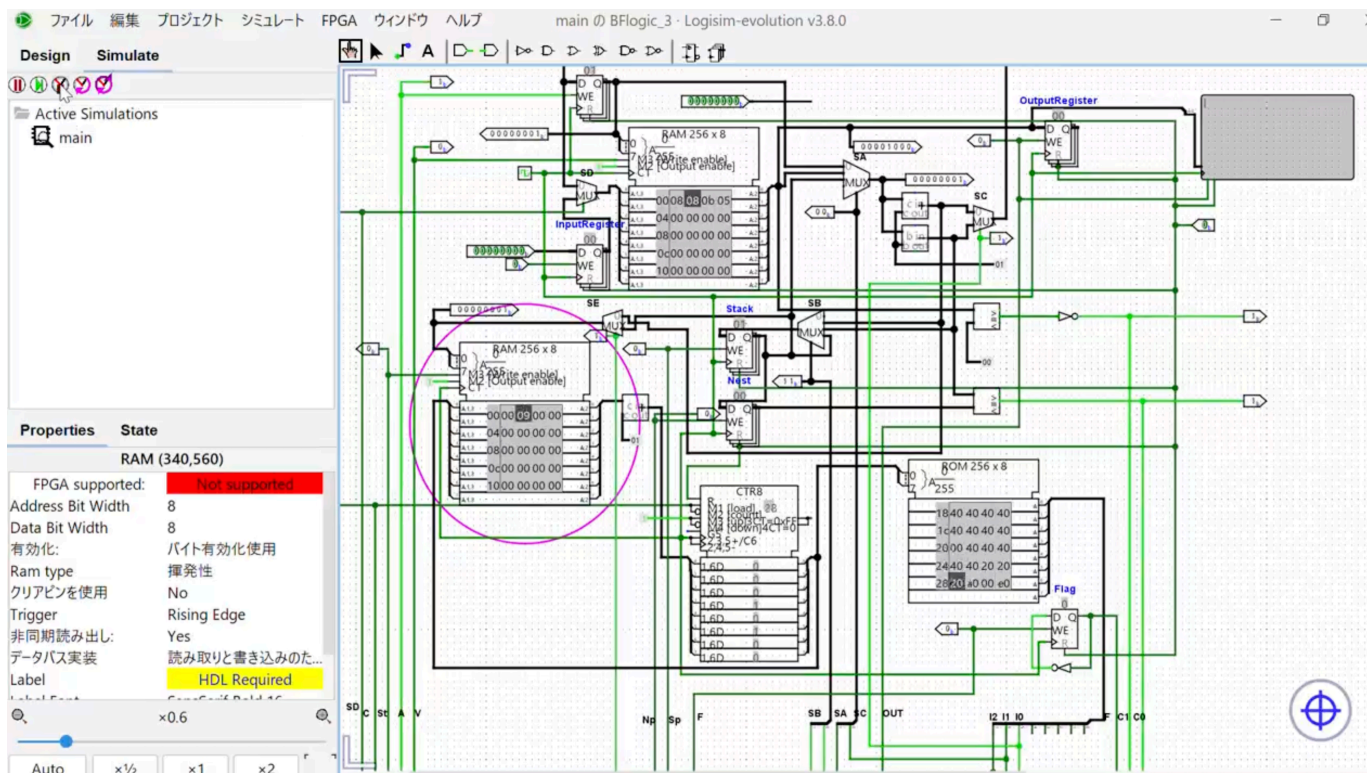
今作ろうとしてるCPU

- ~~命令が8個のみの難解プログラミング言語をインタプリタ・コンパイラを介さず
直接実行する独自ISAのCPU~~

面白い性質を持ったCPUを作ってます！ 世界で我々だけ！！！！！！

どんなことをしているか (1/3)

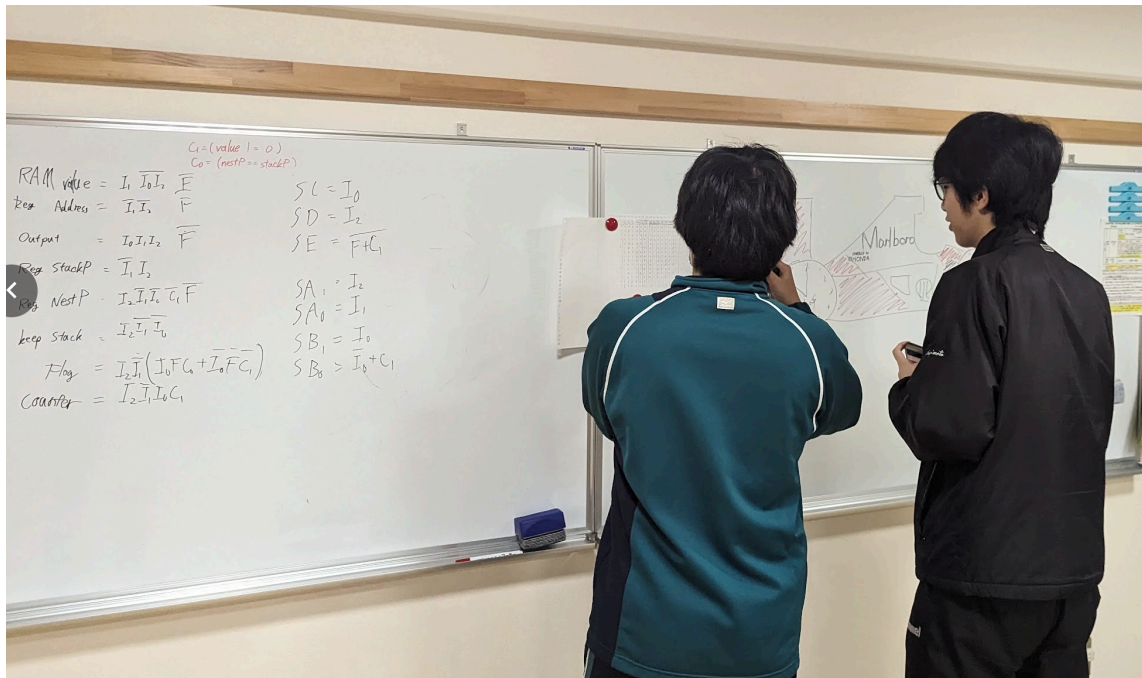
論理回路図を作る



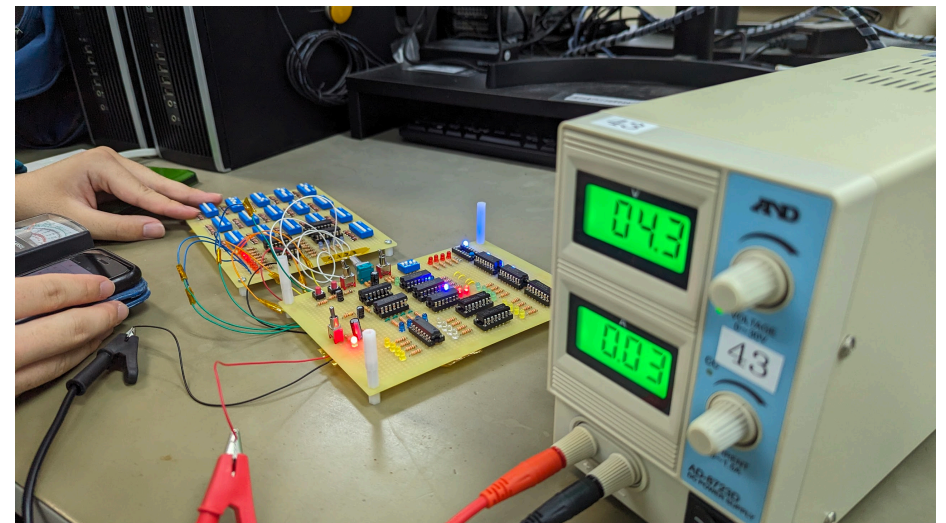
論理回路シミュレータで設計

どんなことをしているか (2/3)

議論したり実験したり



デコーダの仕様を決める会議



CPUのデバッグ

どんなことをしているか (3/3)

uchanさんのpodcastに出たり



電子の森ラジオ

By uchan

電子の森ラジオは電子工作やプログラミングに関する話題を扱うポッドキャストです。司会（パーソナリティ）はuchanでお送りします。

話題性のあるゲストを招いてお話を聞いたり、技術的な話題に関して議論したり、リスナー...

See more

[Listen on Spotify](#)

[Send voice message](#)



Available on

Brainfuckとは

```
+++++[>+++++++>+++++++>+++++++>+++++++>++++
+<<<<-]>++++.>-----.>---.+++<<<+>[>>++++<-----<-]>>.>>+
+.<<<<++++[>>---<<-]>>+++++++>>.<<<.>.>+++++.----.-.-.<++++
+++.>>.<++++>>+>[<<+++++.>>-]<.<<<--.<+>[>>--->-----<<<-]>>>.<-.>+>
+++++.----.<---.+++.>+++++++.----.------.>>+>[<+++++++>-]<.
```

Brainfuckとは

```
+++++[>+++++++>+++++++>+++++++>+++++++>++++
+<<<<-]>++++.>-----.>---.+++<<<+>[>>++++<-----<-]>>.>>+
+.<<<<++++[>>----<<-]>>+++++++>>.<<<.>-.>+++++.-----.-.<++++
+++.>>.<++++>>+>[<<++++.>>-]<.<<<--.<+>[>>-->-----<<<-]>>>.<-.>+>
+++++.-.-.<--.+++.>+++++.-.-.-.-.-.>>+>[<+++++>-]<.
```

- 出力: “Today is Friday in California.”

Brainfuckとは

命令	意味
>	ポインタを+1
<	ポインタを-1
+	値を+1
-	値を-1
.	値を出力
,	入力を値に代入
[値が0なら]にjump
]	値が0でないなら[にjump

- たった”8つ”の命令
- 実用性は乏しいが、“チューリング完全”なので任意の計算が可能

Brainfuckの特徴

- メモリを直接扱う言語である
 - Pythonと違って勉強になる
- 比較、ループ、基本的演算はサポートされている
- ハードウェアの動作を頭で思い浮かべられる
 - 難解言語と言われるが…… あんまりそう思わない

C言語の授業中



今日から配列を扱います！
ポインタは、こうです！！！！

紙に書くとわかりやすいです！！！！

C言語の授業中



今日から配列を扱います！
ポインタは、こうです！！！！

紙に書くとわかりやすいです！！！！



Brainfuckのほうが
わかりやすいなあ…



(盛り上がる)

C言語の授業中



今日から配列を扱います！
ポインタは、こうです！！！！

紙に書くとわかりやすいです！！！！



BrainfuckをCPUに
できるだろうか？



(また盛り上がる)

C言語の授業中



今日から配列を扱います！
ポインタは、こうです！！！！

紙に書くとわかり



BrainfuckをCPUに
できるだろうか？



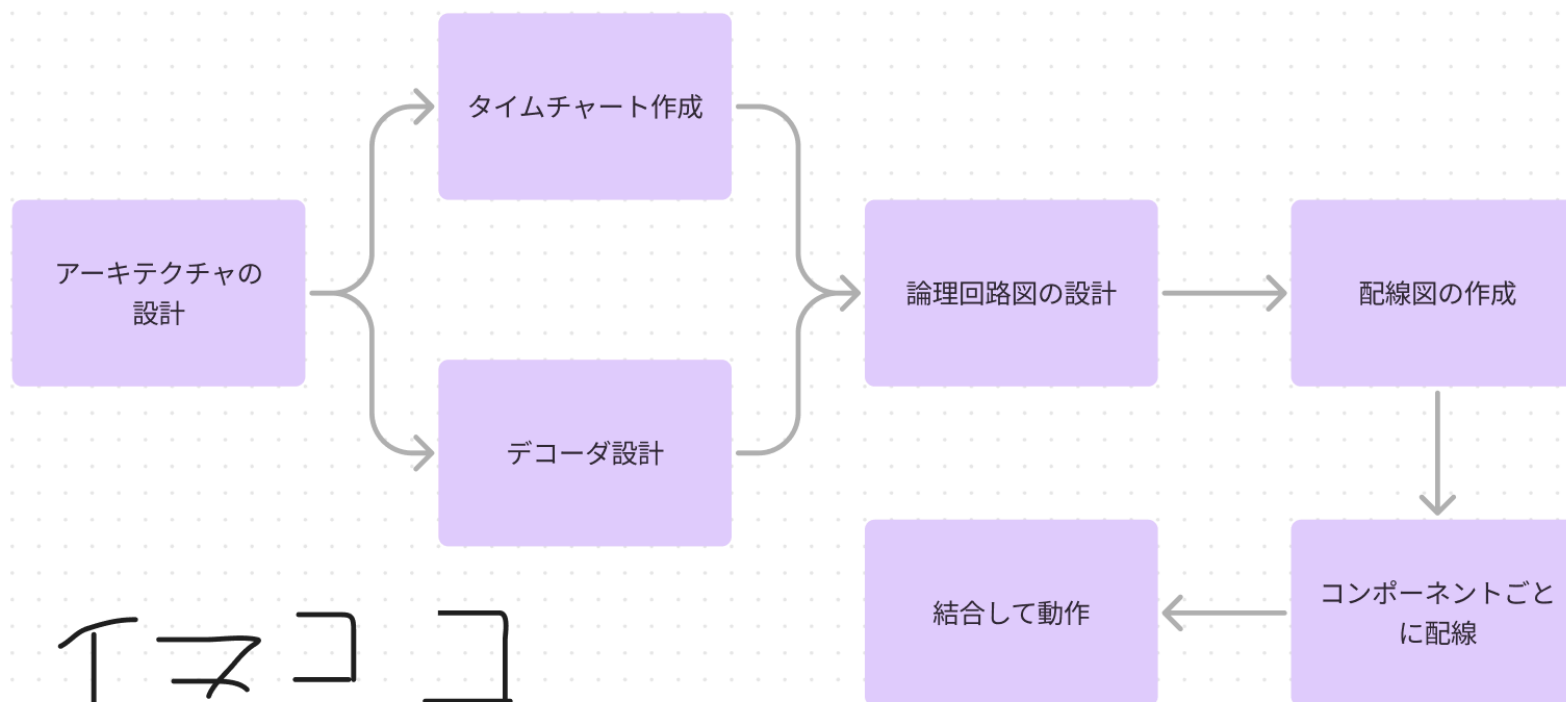
(また盛り上がる)

決定!

具体的な仕様

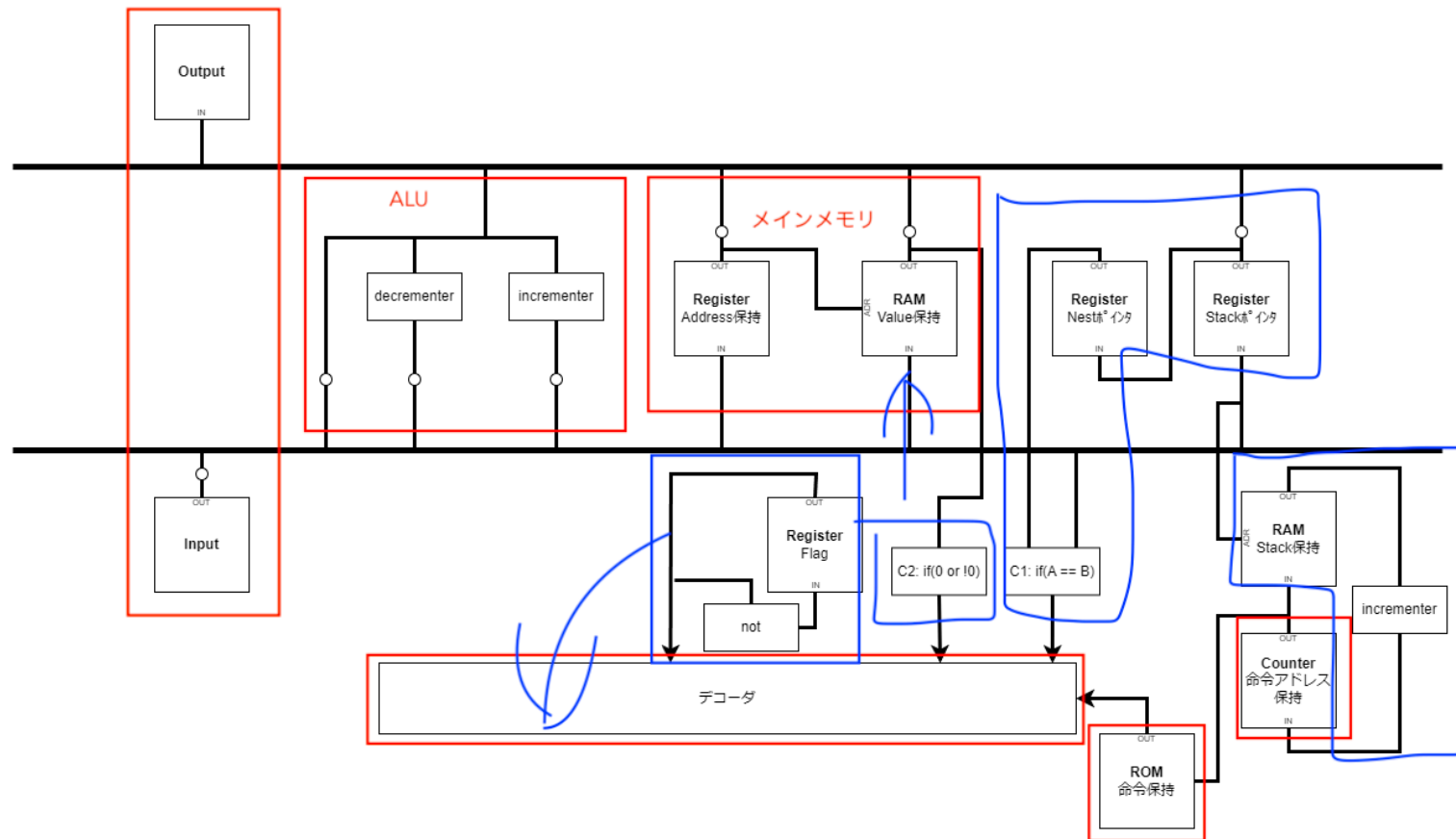
- Brainfuckのプログラムを入力し計算するハードウェアインタプリタ
w/o コンパイラ
- 論理ICのみを使って実装 (e.g. 74HCxx)
- 動作周波数は～2MHzを想定
- アドレス幅、データのサイズは8bit

何をやっていったか

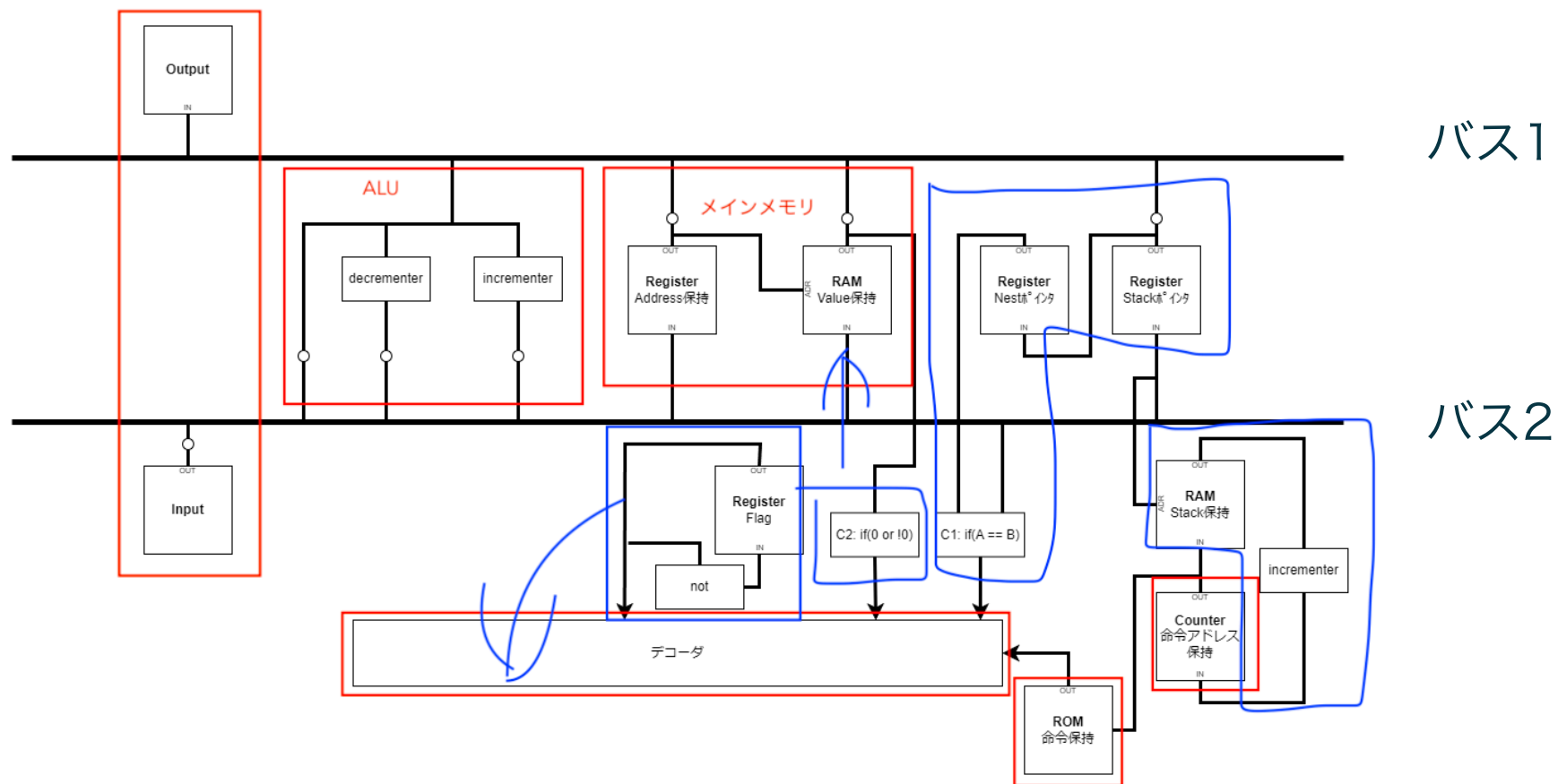


↑マココ

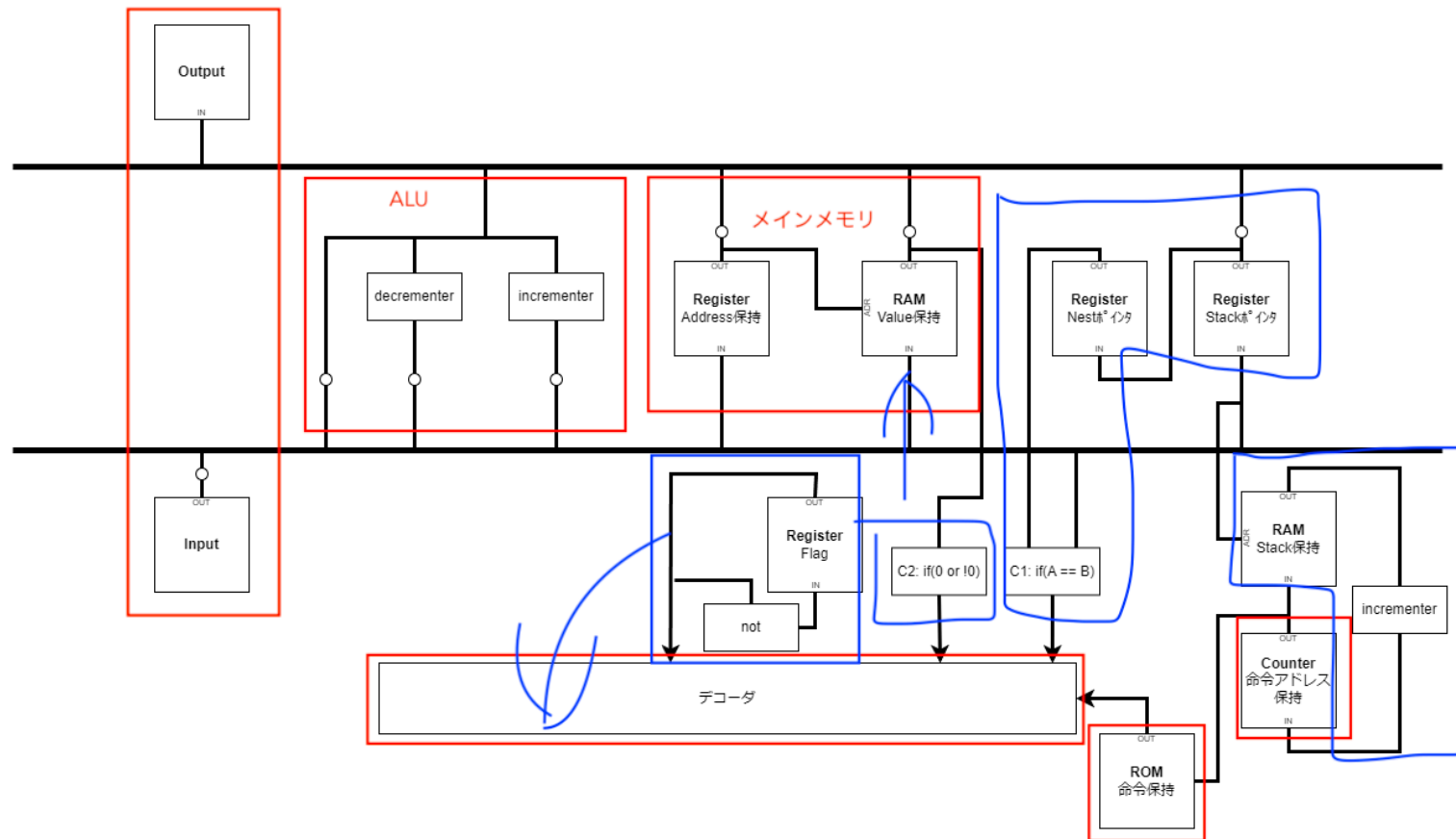
アーキテクチャ説明



アーキテクチャ説明

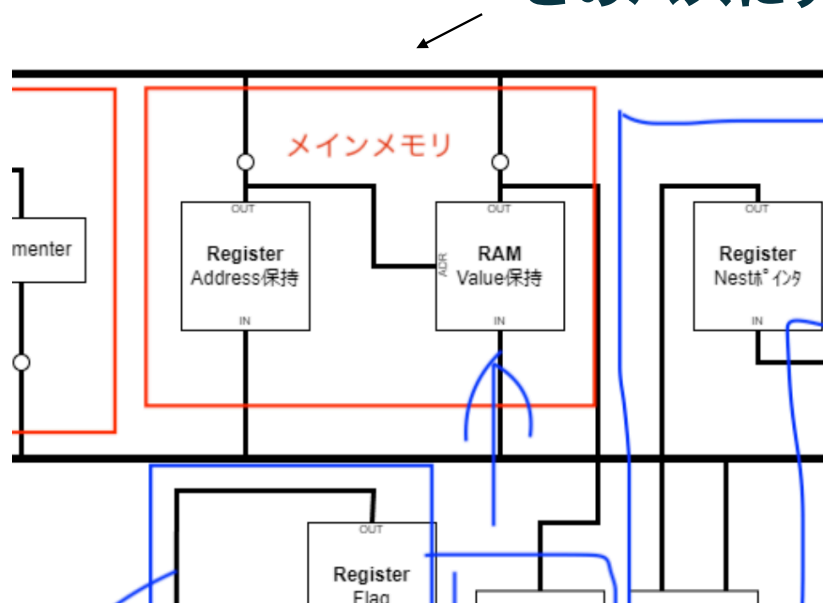


アーキテクチャ説明 (メモリ)



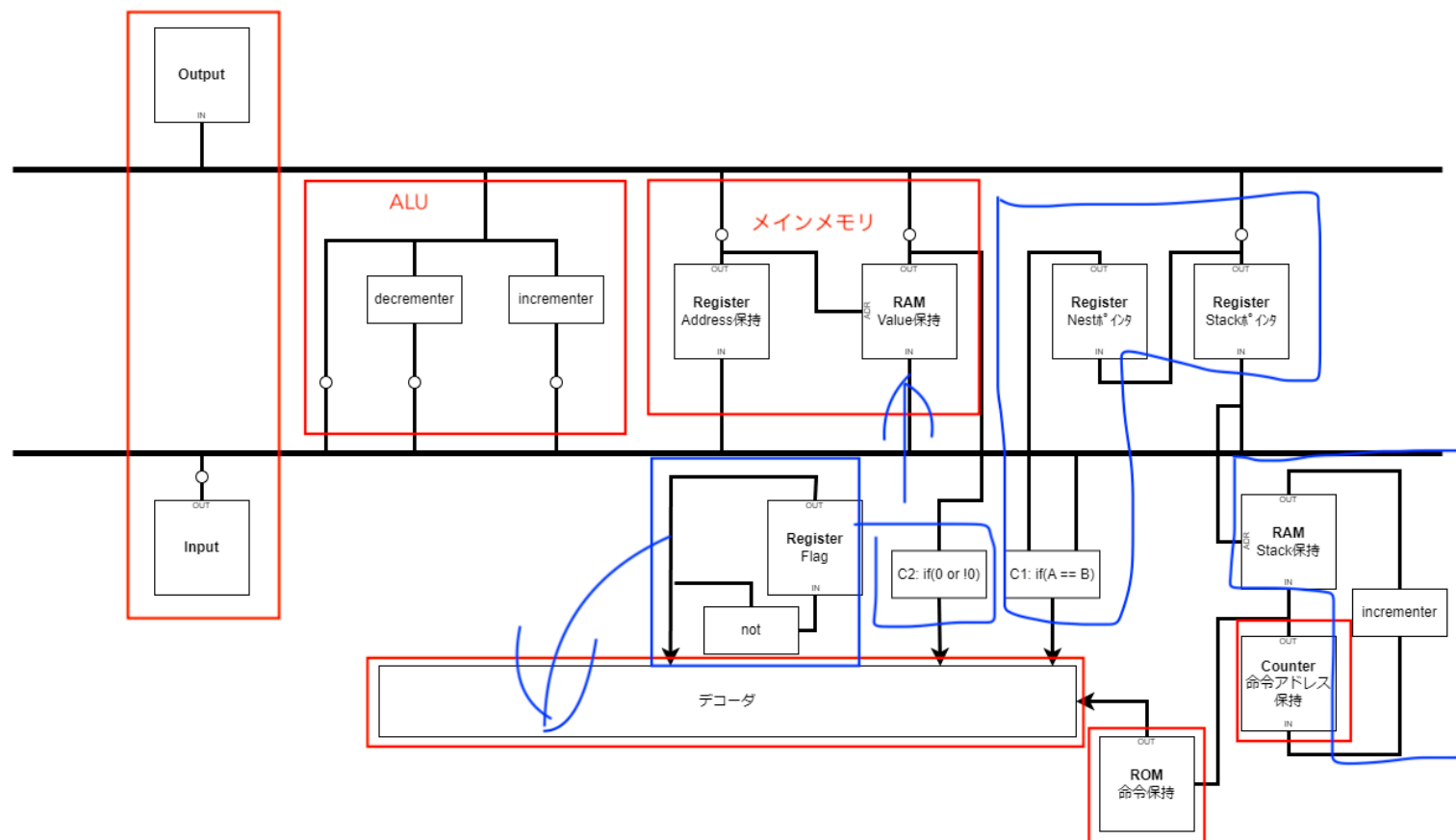
アーキテクチャ説明 (メモリ)

このバスにデータが流れる



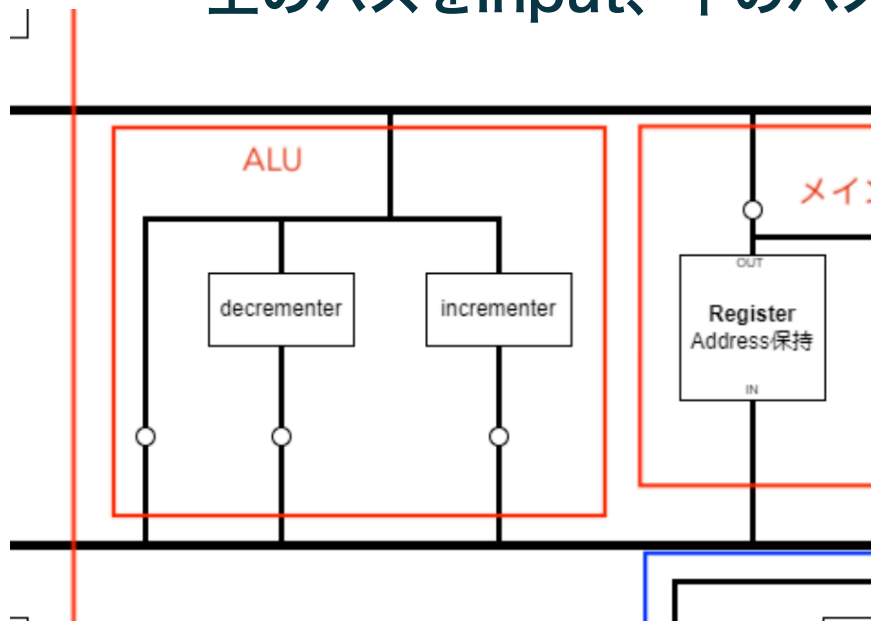
- アドレスを保持するレジスタ
- 値を保持するRAM
- どちらもクロックに良い感じに同期させる

アーキテクチャ説明 (ALU)



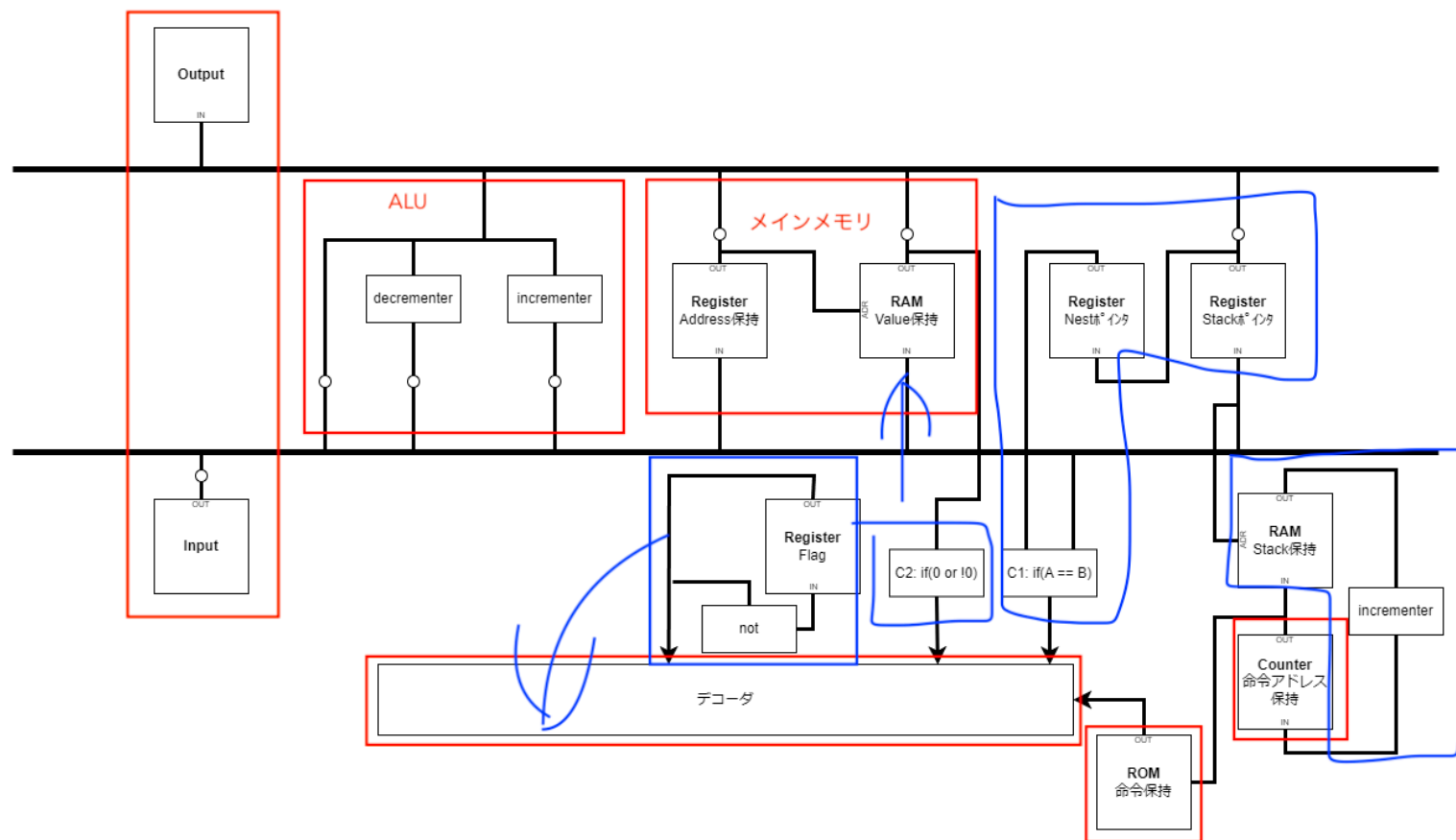
アーキテクチャ説明 (ALU)

上のバスをInput、下のバスにOutput

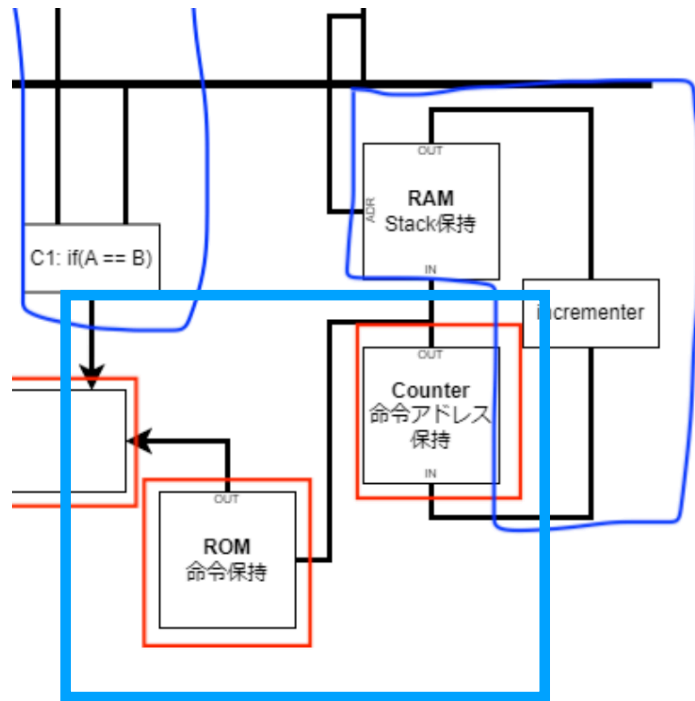


- ALUがすること
- 何もしない or +1 or -1
- デコーダによってトライステートバッファが制御される

アーキテクチャ説明 (ROM)

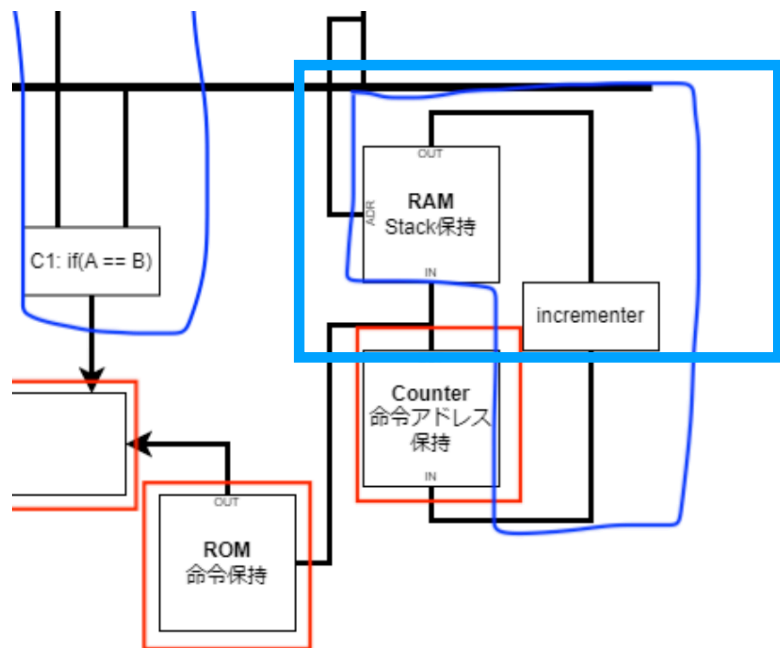


アーキテクチャ説明 (ROM)



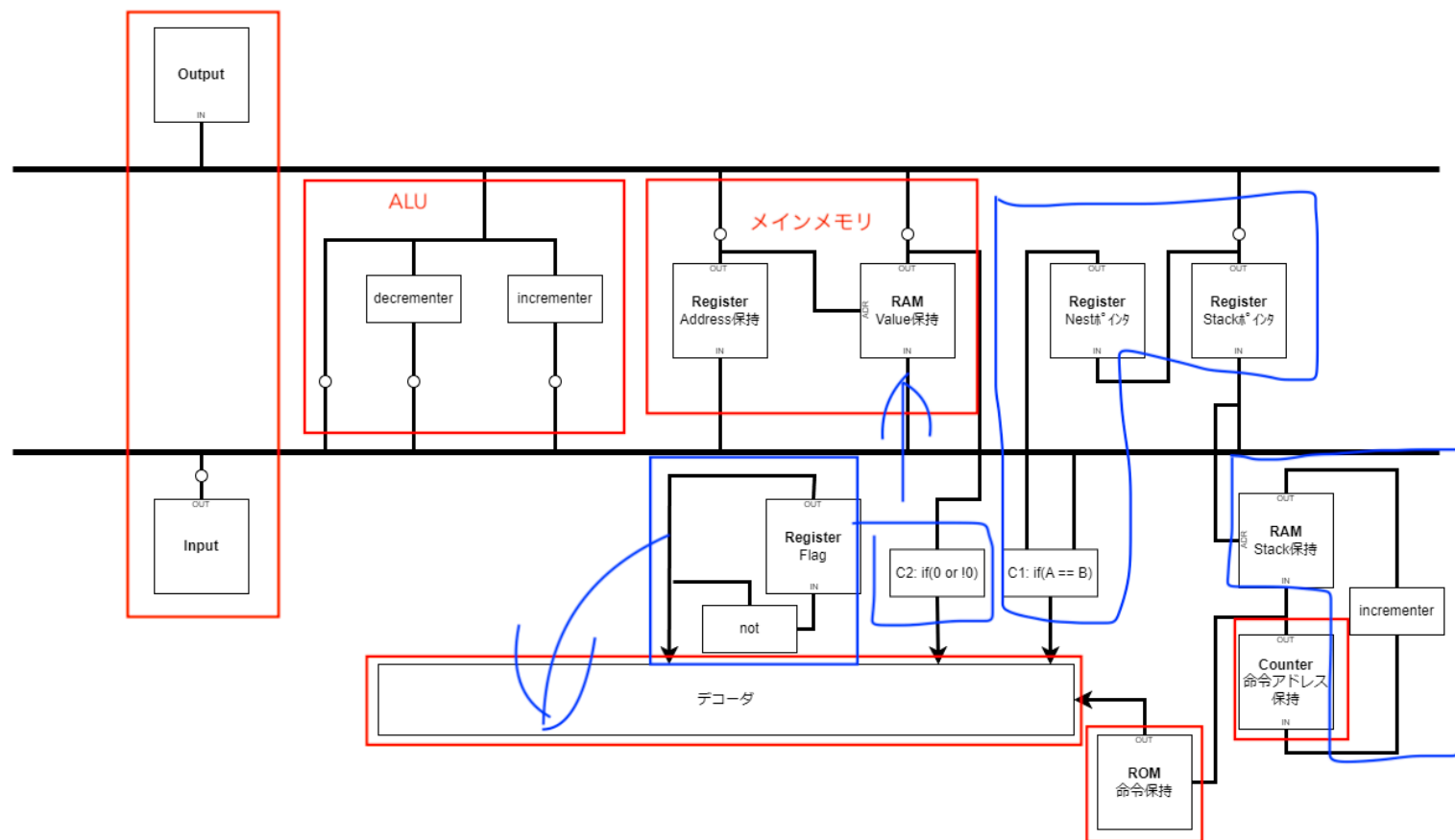
- プログラムカウンタをクロックごとに+1してROMに入力する

アーキテクチャ説明 (ROM)

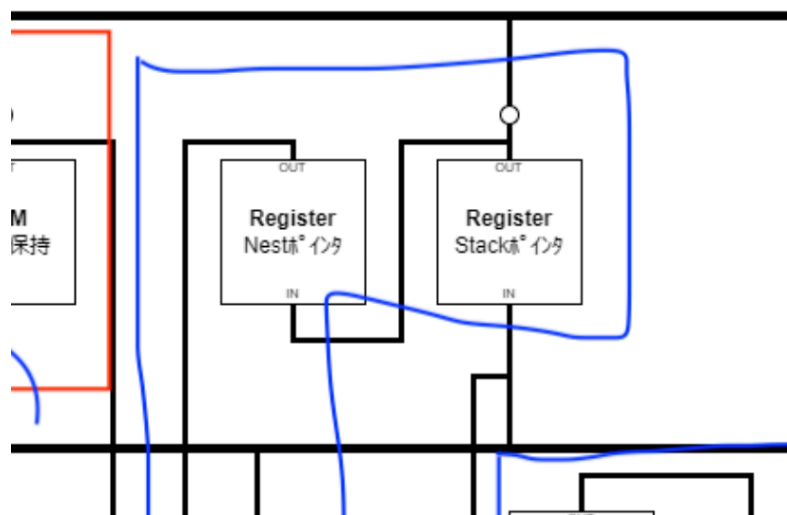


- ジャンプする場合があるので、スタック(後ほど)からpopしてきた値をロードする機能もついている

アーキテクチャ説明 (Nest, Stack Pointer)



アーキテクチャ説明 (Nest, Stack Pointer)



- Brainfuckのループの本質だが、説明が難しいので概念だけ説明
 - ループはジャンプによって実現される
 - C言語における関数呼び出しに近い

Brainfuckにおけるループ

- `[`: 値が0であれば対応する`]`の直後ジャンプ
- `]`: 値が0以外なら対応する `[`の直後にジャンプ

Brainfuckにおけるループ

- `[`: 値が0であれば対応する`]`の直後ジャンプ
- `]`: 値が0以外なら対応する `[`の直後にジャンプ

対応する`]`ってどこですか……

インタプリタなので、後ろのプログラムは見えない。

→`]`が見つかるまでNOP

Brainfuckにおけるループ

- `[`: 値が0であれば対応する`]`の直後ジャンプ
- `]`: 値が0以外なら対応する `[`の直後にジャンプ

対応する`]`ってどこですか……

既に見ているはず

→ `[`のアドレスをスタックにpushしていこう

satckの現在位置を**stack pointer**と呼ぶ

Brainfuckにおけるループ

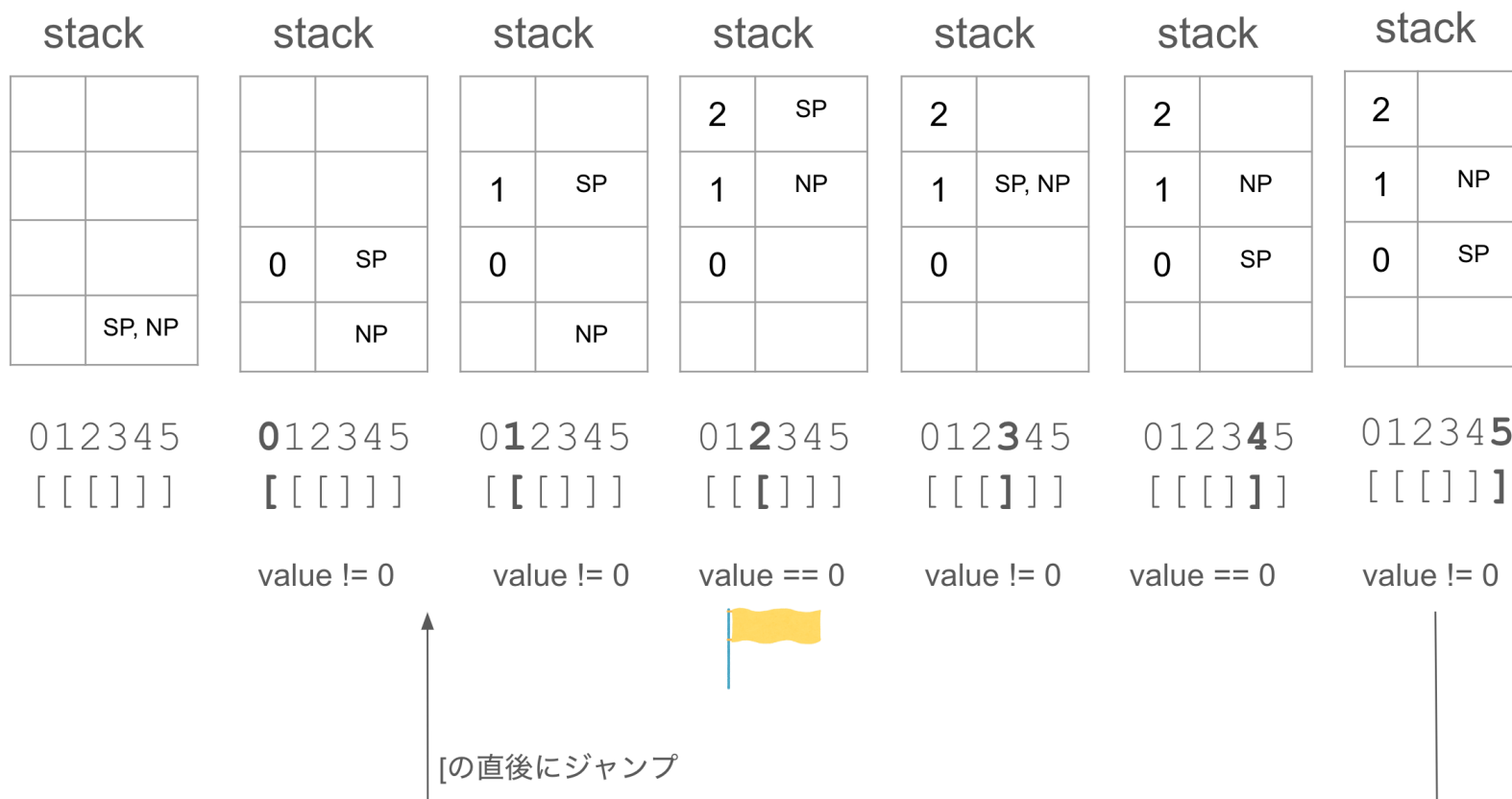
- `[`: 値が0であれば対応する`]`の直後ジャンプ
- `]`: 値が0以外なら対応する `[`の直後にジャンプ
- ネストしているときに `[`から`]`にジャンプする動作は非自明では？

どの深さの `[` `]`までNOPさせるかを保存することで解決する

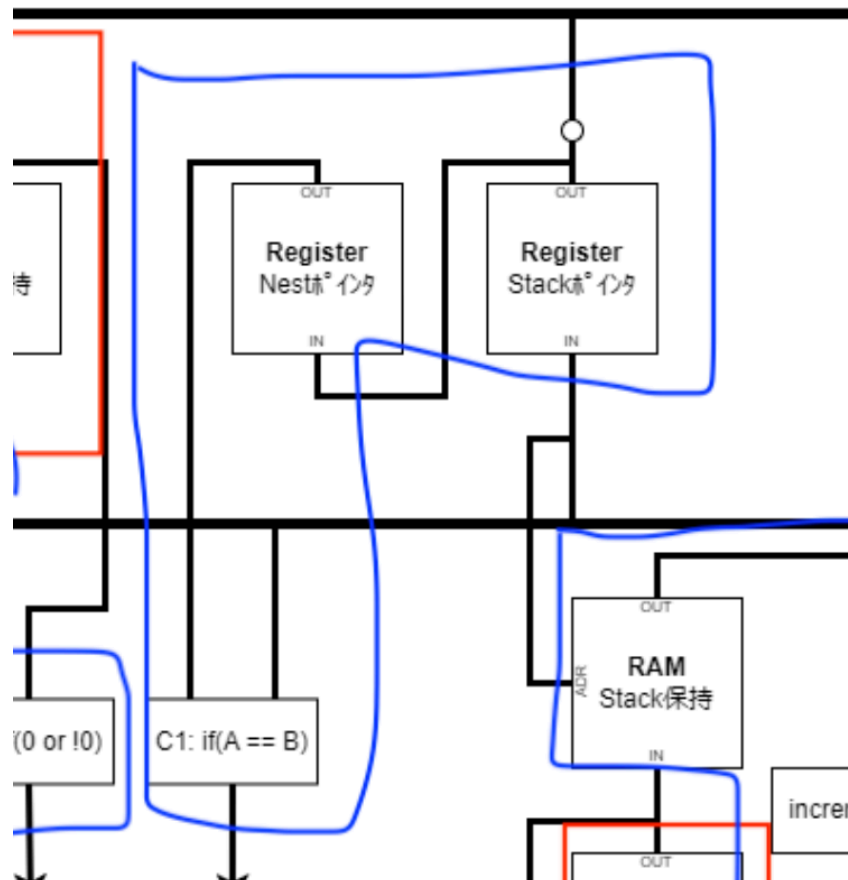
→ **Nest Pointer**と呼ぶ

Stack PointerとNest Pointerを比較することで、
どの深さまでNOPさせるか判断可能

SP, NPの動きの例

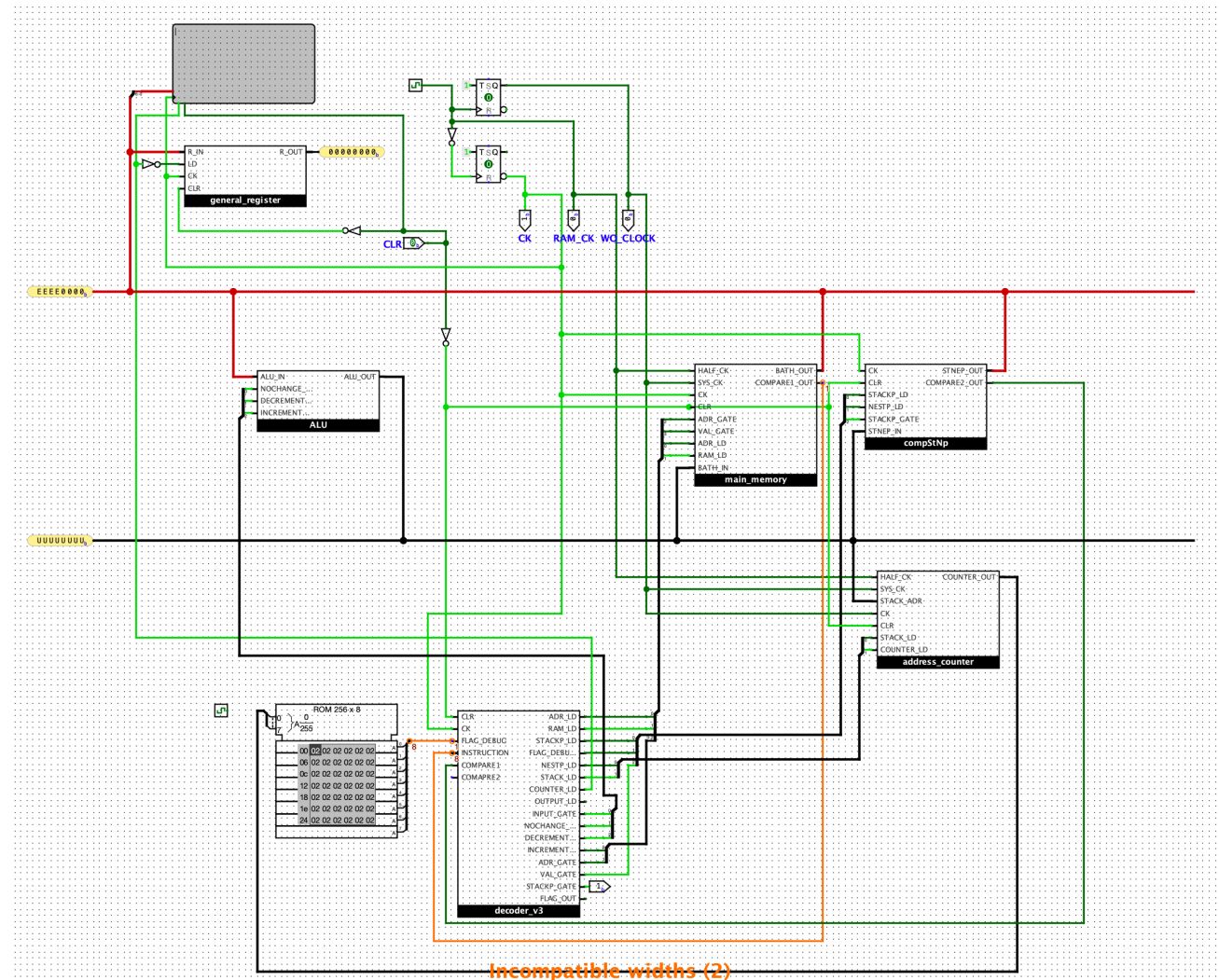


アーキテクチャ説明 (Nest, Stack Pointer)

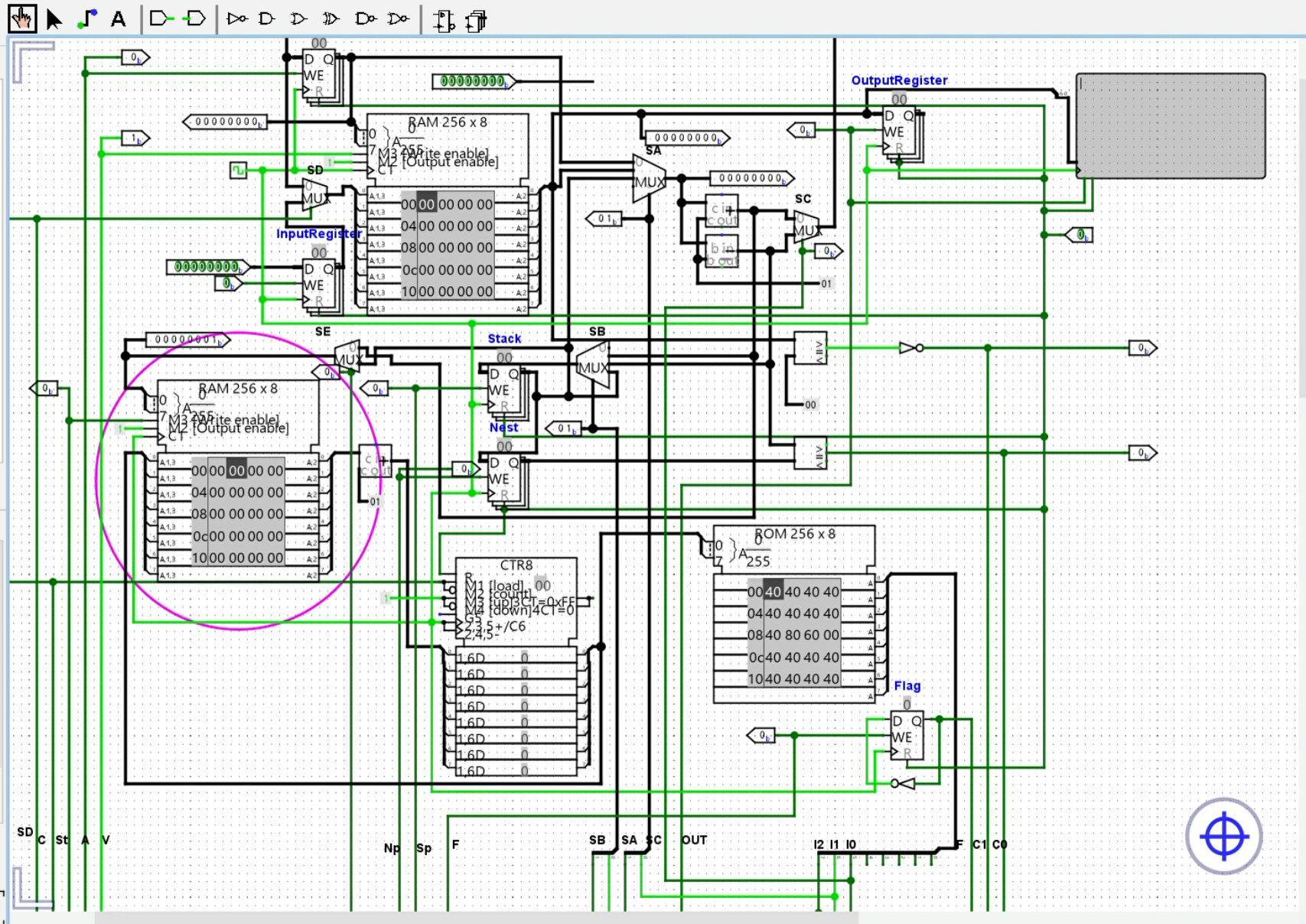


- Stackを保持するRAMと、Stack PointerとNest Pointerを保持するレジスタ

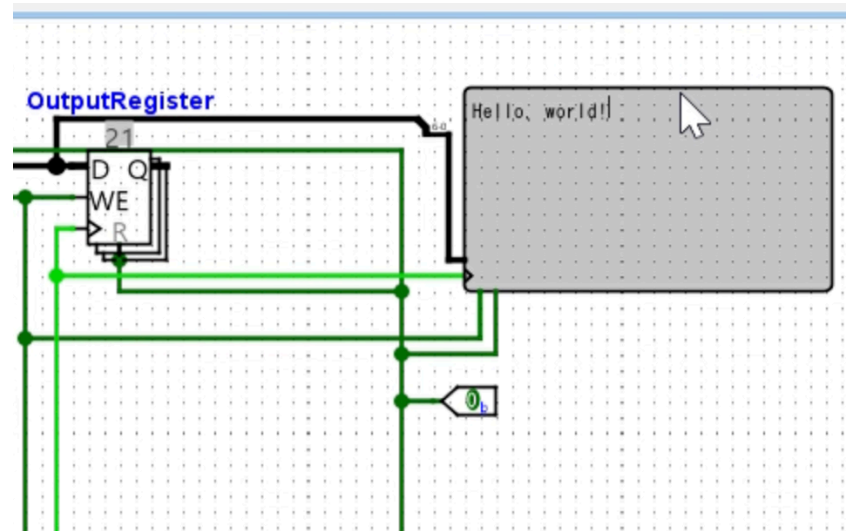
こんな感じ



Properties	State
RAM (340,560)	
FPGA supported:	Not supported
Address Bit Width	8
Data Bit Width	8
有効化:	バイト有効化使用
Ram type	揮発性
クリアピンを使用	No
Trigger	Rising Edge
非同期読み出し:	Yes
データバス実装	読み取りと書き込みのた...
Label	HDL Required



論理回路では動いた！



ハードウェアに実装するうえで困ったこと

- マルチプレクサに頼った設計
- 動作の掌握に2週間かかったRAM
- ICを焼きまくった日々
- 入手性の悪いIC、意味のわからないIC

ハードウェアに実装するうえで困ったこと

- マルチプレクサに頼った設計
- 動作の掌握に2週間かかったRAM
- ICを焼きまくった日々
- 入手性の悪いIC、意味のわからないIC

マルチプレクサ (157) に頼った設計

- 結論：トライステートバッファ (541) を使おう
 - 拡張性に欠ける

ハードウェアに実装するうえで困ったこと

- マルチプレクサに頼った設計
- 動作の掌握に2週間かかったRAM
- ICを焼きまくった日々
- 入手性の悪いIC、意味のわからないIC

SELECT * FROM * WHERE package = 'DIP' AND type = 'Parallel';

- まともに売ってるのがこれくらいしかない

ホーム > 各種 半導体 > SRAM > MB84256C-70L

MB84256C-70L



カテゴリ	SRAM 富士通
販売価格	500円(税込550円)
在庫数	44袋
購入数	<input type="text" value="1"/> 袋(1個入)

 [カートに入れる](#)

» [特定商取引法に基づく表記 \(返品など\)](#)

- 富士通
- 256K(32K×8) 70ns
- Vcc:5V CMOS

[資料PDF\(参考資料\)](#)

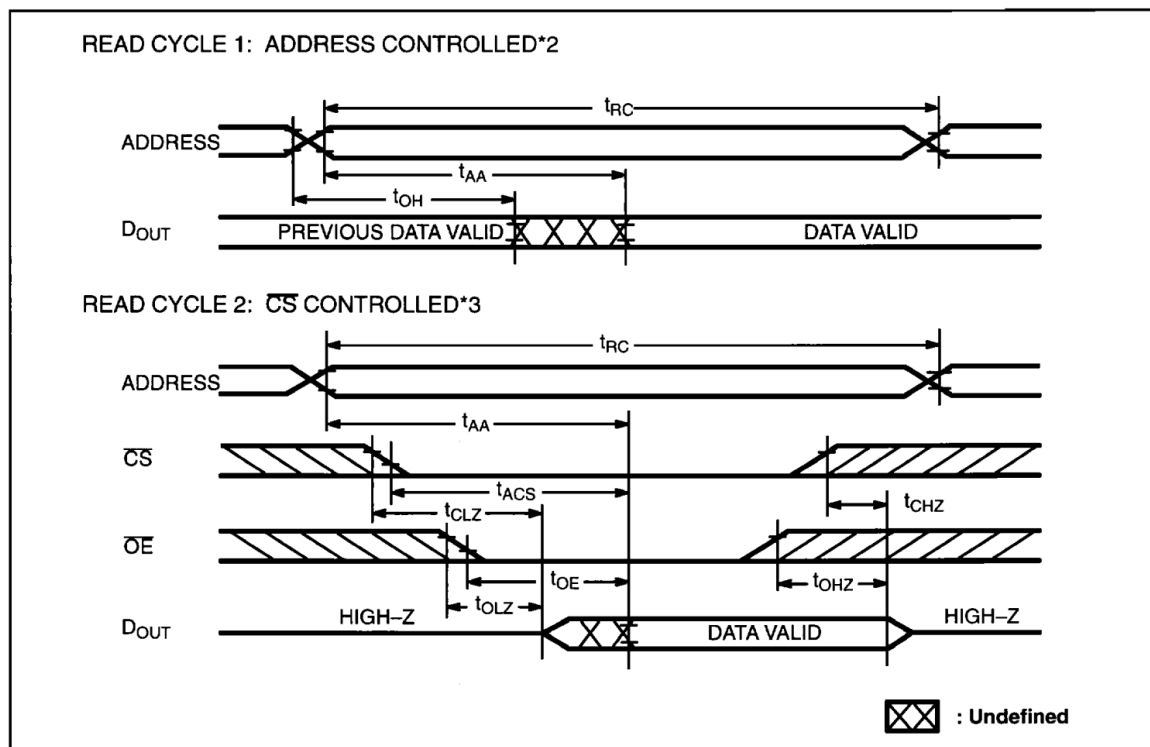
<https://suzushoweb.shop-pro.jp/?pid=101168919>


```
SELECT * FROM * WHERE package = 'DIP'  
AND type = 'Parallel';
```

RAMってクロックないんですね^^

Output Enable to Output High-Z

READ CYCLE TIMING DIAGRAM *1



Note: *1 WE is high for Read cycle.

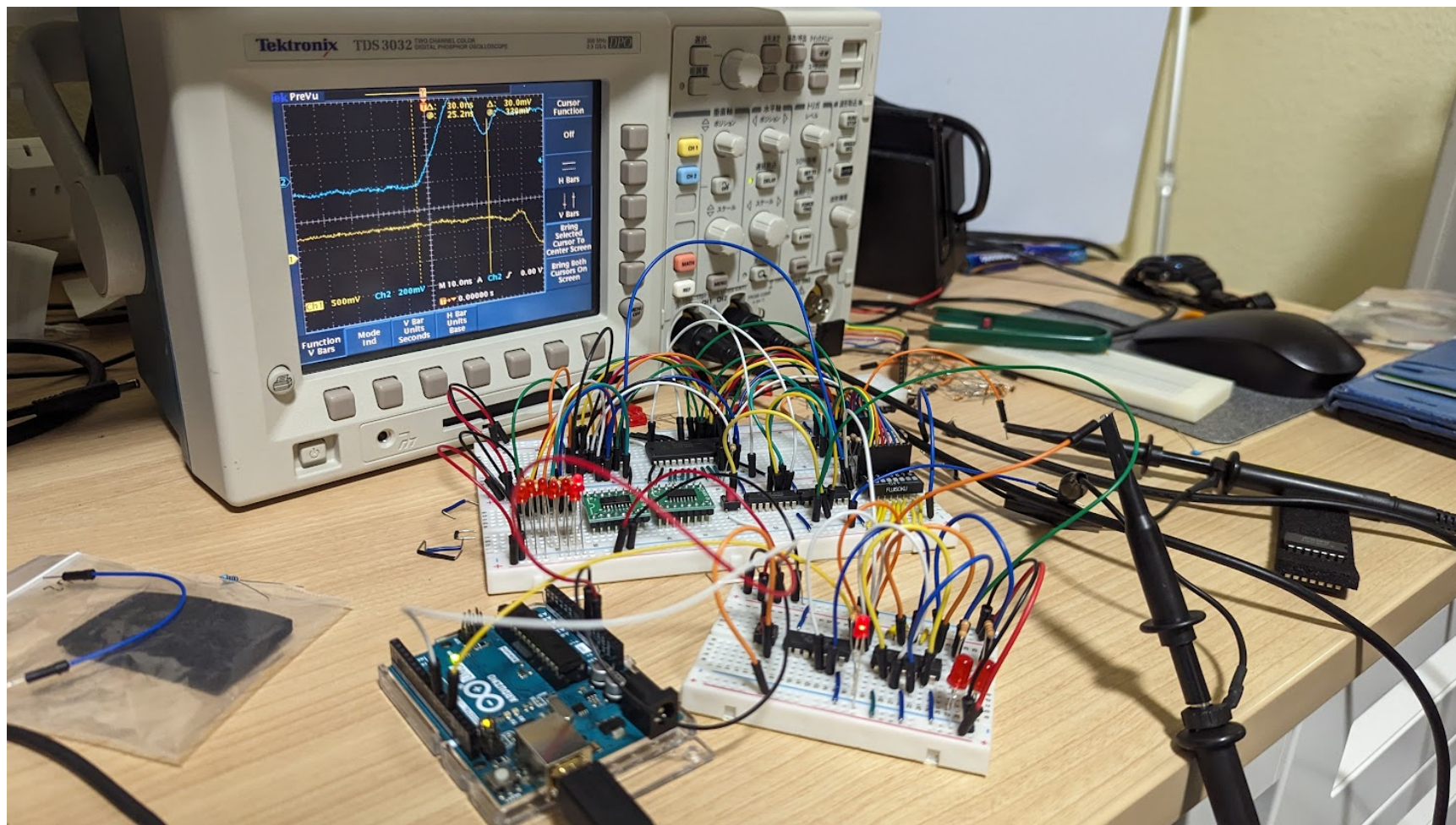
*2 Device is continuously selected, CS = OE = V_{IL}.

*3 Address valid prior to or coincident with CS transition low.

*4 Transition is measured at the point of ±500mV from steady state voltage with specified Load II in Fig. 2.

- 論理回路シミュレータと仕様が異なりすぎて雲行きが怪しくなる

クロックと遅延回路の評価

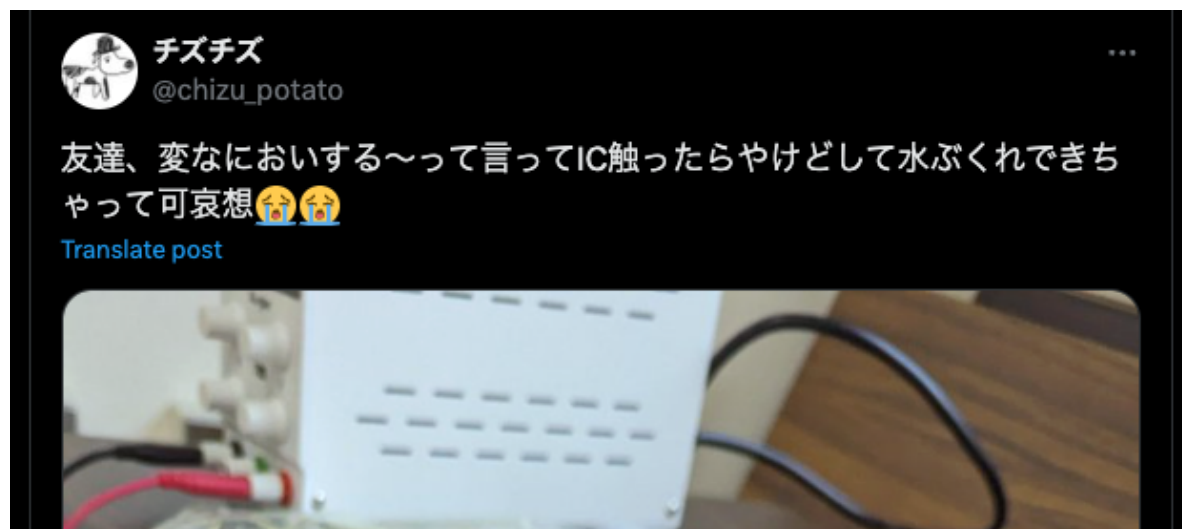


ハードウェアに実装するうえで困ったこと

- マルチプレクサに頼った設計
- 動作の掌握に2週間かかったRAM
- ICを焼きまくった日々
- 入手性の悪いIC、意味のわからないIC

謎の発熱

- RAMだけが、原因不明の発熱に悩まされた (n=3)
- 中々再現ができなかったが、浮かせた状態が危険ということがわかった
- 直流安定化電源のノイズも危ない
- 大体掌握



ハードウェアに実装するうえで困ったこと

- マルチプレクサに頼った設計
- 動作の掌握に2週間かかったRAM
- ICを焼きまくった日々
- 入手性の悪いIC、意味のわからないIC

入手性の悪いIC

- 秋葉原で探しても全然見つからない
 - オンラインショップで買おう
- 入手できるICを探そう
 - あるなら74シリーズを選ぼう
 - レジスタにはカウンタ回路がちょうどいい（要出典）

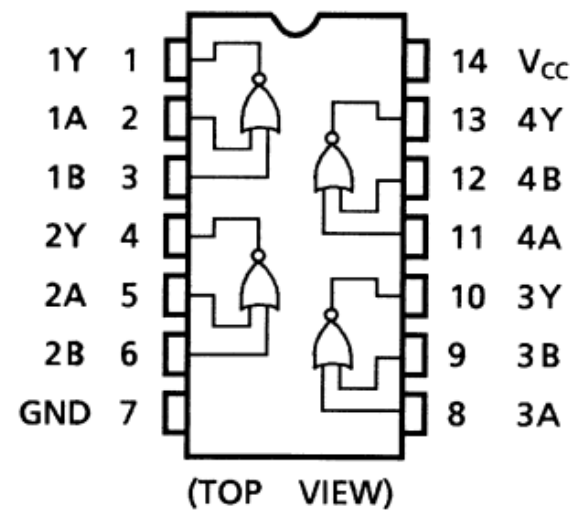
~~このIC・・・何か変・・・~~

- 74HC02って知ってますか？

このIC・・・何か変・・・

- 74HC02って知ってますか？
- NOR回路です

ピン接続図

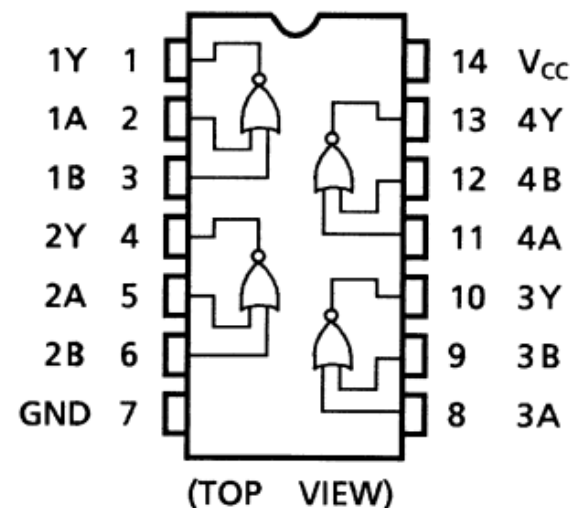


このIC・・・何か変・・・

- 74HC02って知ってますか？
- NOR回路です

**よく見たら素子の向きが逆
下から上！！！！**

ピン接続図



ハードウェアに実装するうえで困ったこと

- マルチプレクサに頼った設計
- 動作の掌握に2週間かかったRAM
- ICを焼きまくった日々
- 入手性の悪いIC、意味のわからないIC
- 1クロックでやることが多すぎる

1クロックでやること

- RAMから値を読み出す
- ALUで足し算する
- RAMに書き込む

1クロックでやること

- RAMから値を読み出す
- ALUで足し算する
- RAMに書き込む



できないのでクロックを分周
R, Wを作った

どうすればよかったか

- マイクロプログラム方式にすればよかった
 - 1つの命令を単純な命令群に置き換える
- 拡張性も上がる！

もりとにーのブログ: Brainfuck CPUをマイクロプログラム方式で実装したお話 <https://tony-moori.blogspot.com/2017/06/brainfuck-cpu.html>

できたもの

- コンポーネント
 - デコーダ
 - Nest, Stackポインタに関する回路
 - クロック発振器
 - ALU
 - RAM(?)

できなかったもの

- ROM (Arduinoで作るだけ)
- 結合

チーム開発

- 全員寮に住んでいた
- 寮では（広義）電熱器具の持ち込みが禁じられている

チーム開発

- 毎日夜点呼後に部屋に集まって作業
- はんだを使う作業は学校の実験室を借りた
- 廃棄になったオシロを借りた

チーム開発

- 部品はスプシで管理
- 誰がどこで何個持っているか

部品名	個数	所有者	memo
TC74HC00	10	Fukum...	
TC74HC02	5	発送待ち	
TC74HC04UAP	10	Fukum...	
TC74HC08AP	10	Fukum...	
TC74HC32AP	12	Fukum...	
TC74HC157AP	10	Fukum...	2入力マルチプレクサ
TC74HC153AP	4	Fukum...	4入力マルチプレクサ
TC4030BP	4		XOR
CD74HC283E	4	Fukum...	Full Adder
TC74HC161AF	7	Fukum...	同期式4ビットカウンタ
I-12066	2	Fukum...	EEPROM
MB84256C-70L	3	Ochi	SRAM (鈴商)
セラミックコンデンサ 0.1μF	50	Ochi	
電解コンデンサ 100μF	100	Ochi	
TC74HC541AP	7	Ochi	3ステートバッファ
TC74HC4066AP	1	Ochi	アナログスイッチ
TC74HC161AF	7	Ochi	同期式4ビットカウンタ
TC74HC74AP	3	Ochi	Dフリップフロップ
ピンヘッダ低オス	28	Ochi	
SOP16ピンDIP変換基盤	14	Ochi	TC74HC161AF用
コネクタ用ハウジング 8P	20	Ochi	
ケーブル用コネクタ	160	Ochi	
ピンヘッダ 1x40	4	Ochi	
耐熱電子ワイヤー 1mm x 10色	1	Ochi	
DIPスイッチ		Fukum...	
TC74HC283P	11	Sugim...	

展望

- I/Oの充実
- 電話でプログラムを実行できないか (?)
- ゲーミングBrainfuck CPU
- ポータブルゲーミングBrainfuckマイコン
- 同人誌に書いて出す

展示ブースでお話しましょう！！！！