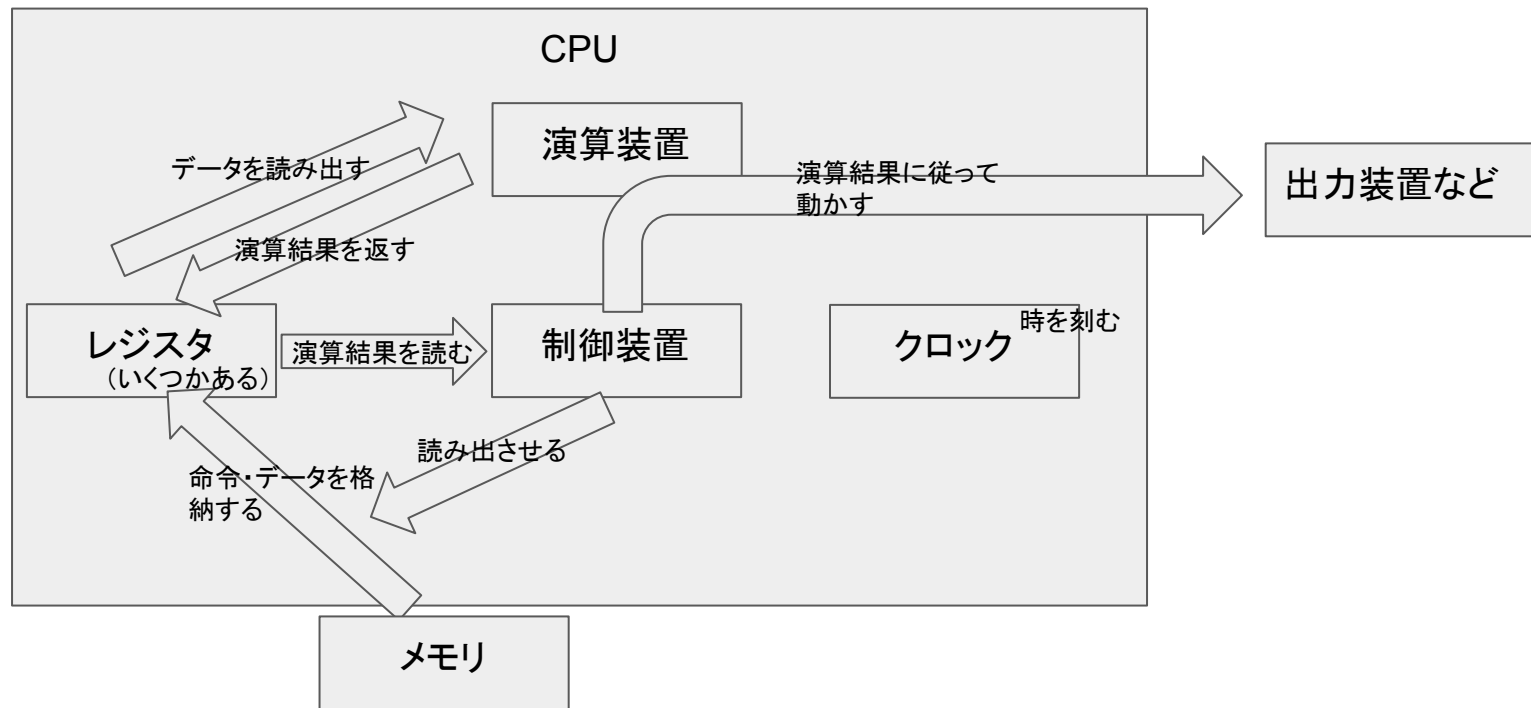


プログラムはどのようにして動いているのか

平田みなみ

CPUの中の動き



レジスタの種類

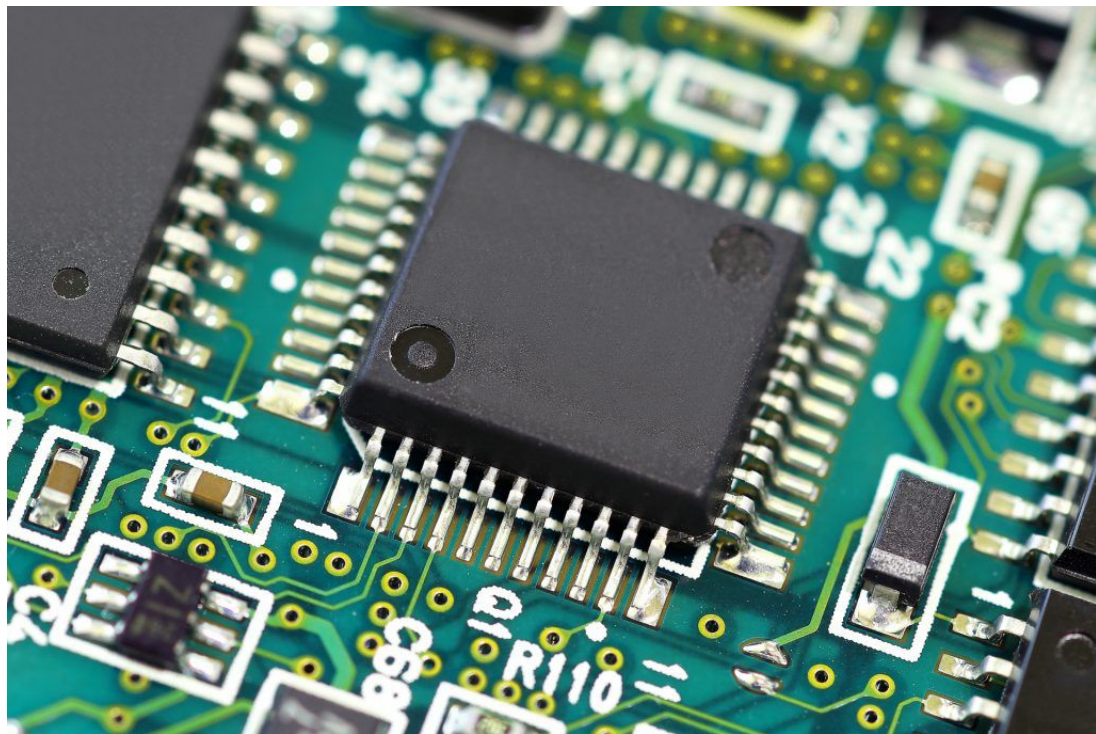
命令レジスタ...メモリから読みだした命令を置く

アドレス部(アドレス〇〇のデータを) 命令部(〇〇せよ)の2パーツに分かれる

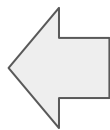
命令アドレスレジスタ(プログラムカウンタ)...命令のアドレスを保持する

アドレスレジスタ...メモリにアクセスする際にアドレスを指定する

CPUやメモリって何でできてるの？



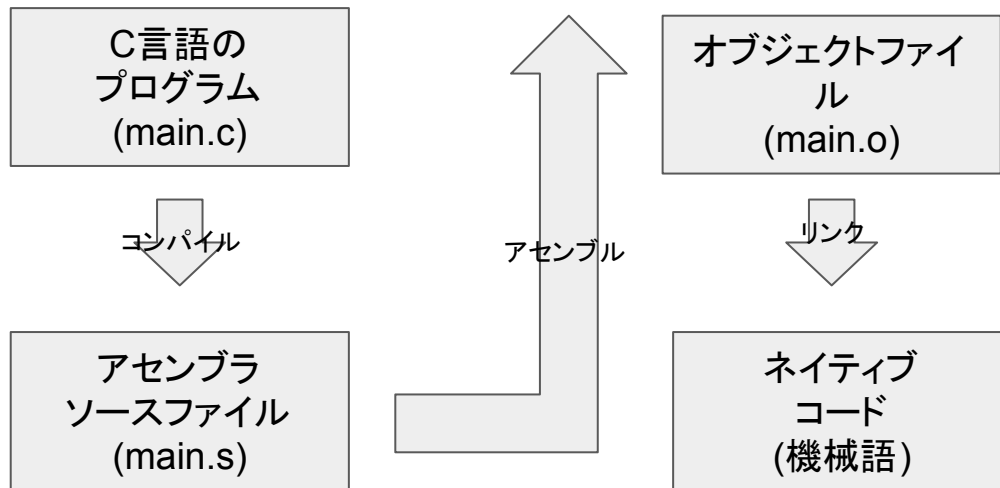
これ



集積回路(Integrated Circuit)とは

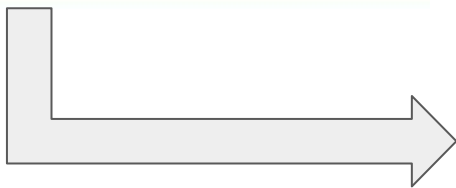
トランジスタ(電氣的なON・OFFをするスイッチ)をたくさん集めたもの
シリコンからできていて電氣を通したり通さなかったりする(半導体)
初期は1チップに数百個のトランジスタしか載せられなかったが、
近年では数百万個搭載できるようになった

C言語の動き方



C言語のhello worldをアセンブリにしてみた

```
1 // Type your code here, or load an
2 #include <stdio.h>
3
4 int main(void)
5 {
6     printf("%s", "hello world");
7
8     return 0;
9 }
```



```
1 .LC0:
2     .string "hello world"
3 .LC1:
4     .string "%s"
5 main:
6     push    rbp
7     mov     rbp, rsp
8     mov     esi, OFFSET FLAT:.LC0
9     mov     edi, OFFSET FLAT:.LC1
10    mov     eax, 0
11    call    printf
12    mov     eax, 0
13    pop     rbp
14    ret
```

どういう意味？

```
1  .LC0:
2      .string "hello world"
3  .LC1:
4      .string "%s"
5  main:
6      push    rbp
7      mov     rbp, rsp
8      mov     esi, OFFSET FLAT: .LC0
9      mov     edi, OFFSET FLAT: .LC1
10     mov     eax, 0
11     call    printf
12     mov     eax, 0
13     pop     rbp
14     ret
```

LC0、LC1...ラベル(メモリ上の場所に名前をつけたもの)

LC0は定数を置く場所

rbp...ベースポインタ(現在の処理のスタックフレームの底のアドレスを記録する)

rsp...スタックポインタ(コールスタックの先頭のアドレス)

rbpをスタックに積む

rspをrbpに代入

OFFSET FLATはラベルを表示するときに前につけるもの

関数printfを呼び出す

rbpをpopする

Javaの動き方

