システムソフトウェア講義の概要

1. 計算機システムの復習：中央演算処理装置(CPU)，プログラムの実行，主記憶装置，補助記憶装置
2. オペレーティングシステムとは：CPU，主記憶装置，補助記憶装置などの抽象化
3. CPUの抽象化１：スレッドとプロセス，割り込み
4. CPUの抽象化２：CPUの割り当てアルゴリズム
5. 主記憶の抽象化：アドレス空間と仮想記憶
6. 補助記憶装置の抽象化：ファイルシステム
7. 演習問題
8. プログラミングシステムの概要，文法とそのクラス，字句解析と正規文法
9. 正規表現からの非決定性オートマトンの生成、決定性オートマトンへの変換
10. 字句解析用オートマトン生成ソフトウエアの実際
11. 構文解析と導出，文脈自由文法の構文解析法：LL構文解析
12. 文脈自由文法の構文解析法：LR構文解析
13. コンパイラ-コンパイラと構文解析の実際
14. 演習問題
15. 講義の総括と試験

計算機アーキテクチャ

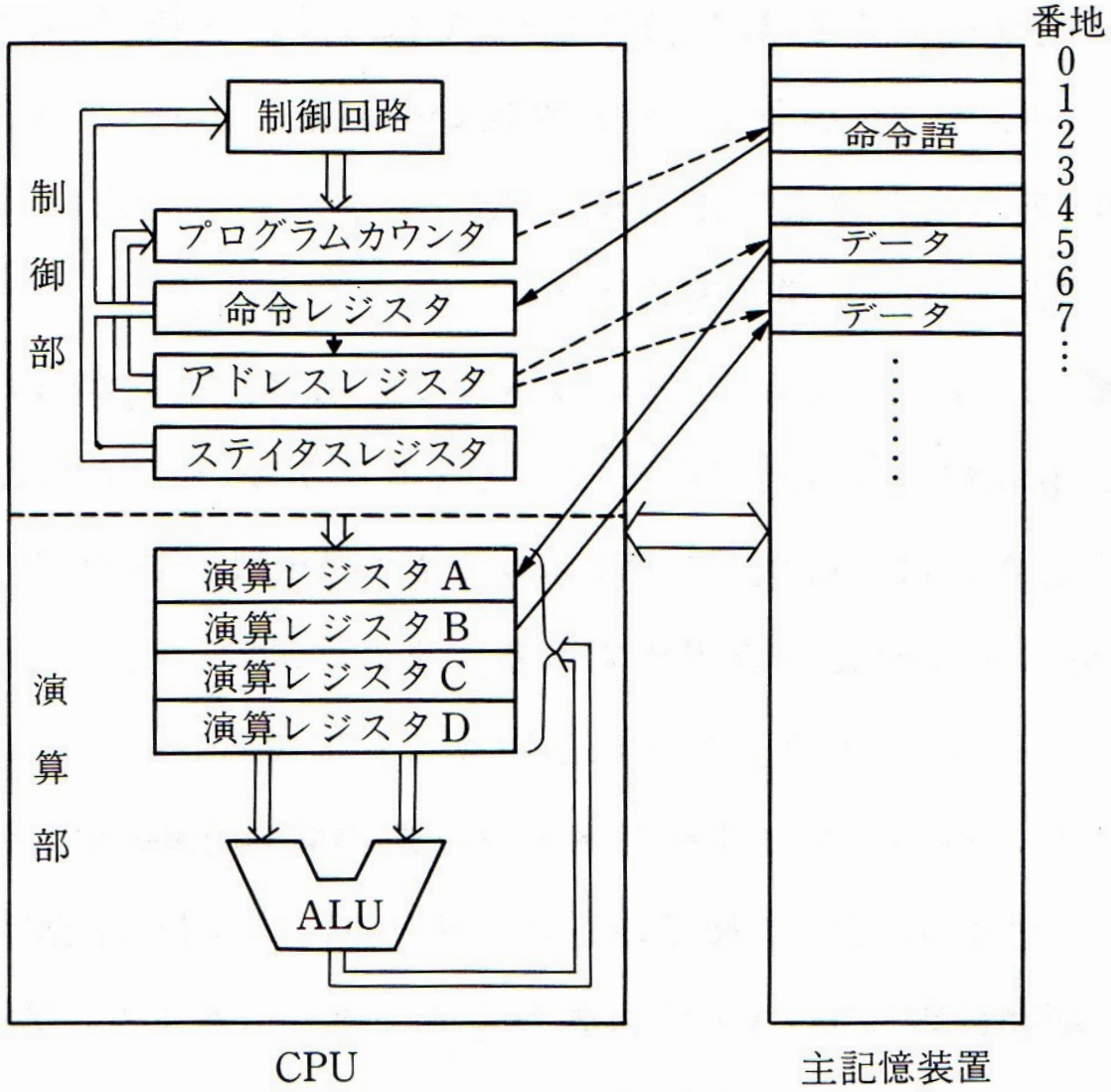


図 3.1　コンピュータの基本構造

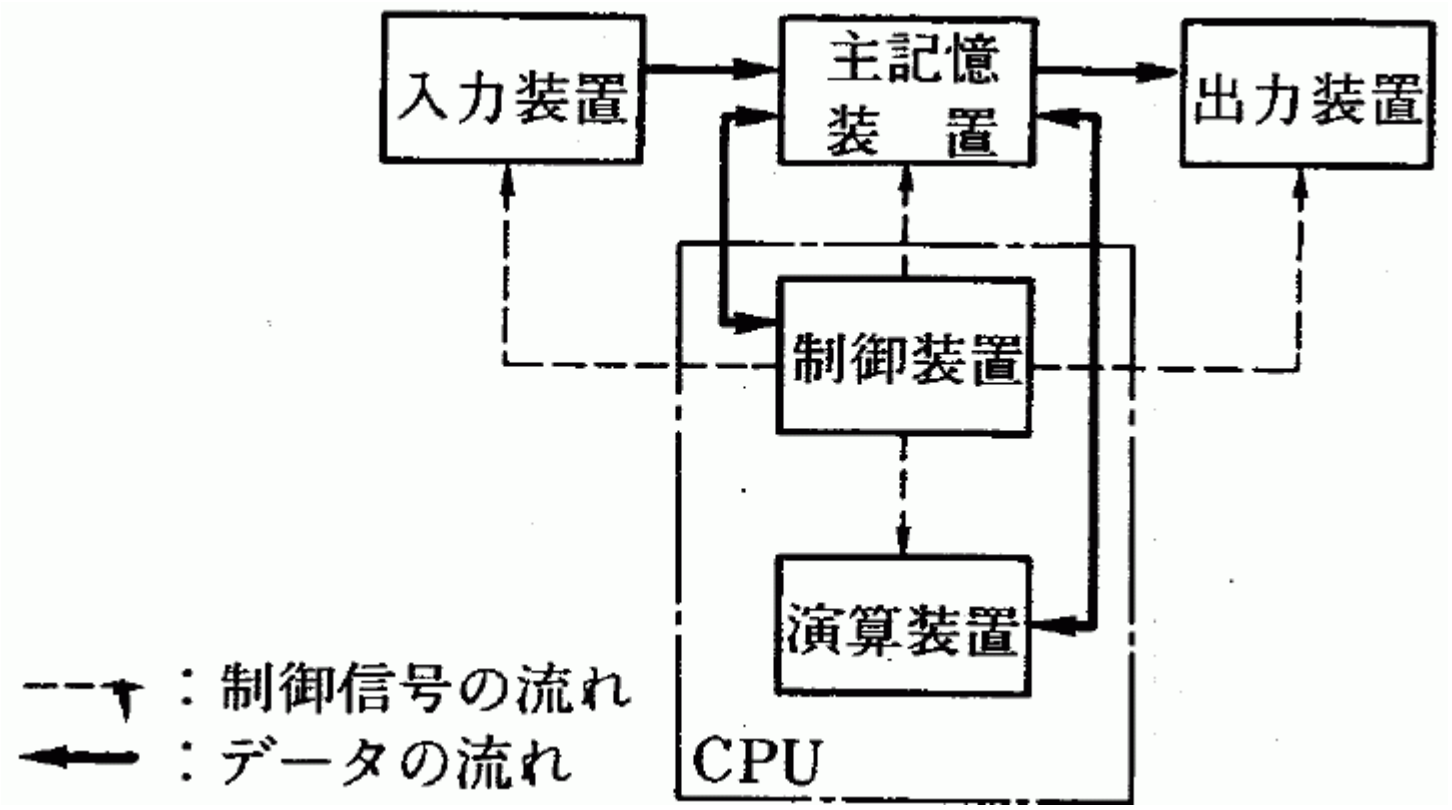
用語と質問

- CPU
  - 制御部
    - 制御回路
    - プログラムカウンタ
    - ステイタスレジスタ
    - 命令レジスタ
    - アドレスレジスタ
  - 演算部
    - 演算レジスタ群
    - ALU
- Fetch-Execution Cycle (命令読み取り実行サイクル)
- フォン・ノイマン型計算機
  - プログラムをメモリ上に置く

質問

- CPUが読み取るプログラムは何語で書かれているか？
- プログラムはメモリのどの番地に置かれるのか？

入出力装置も含めたシステム



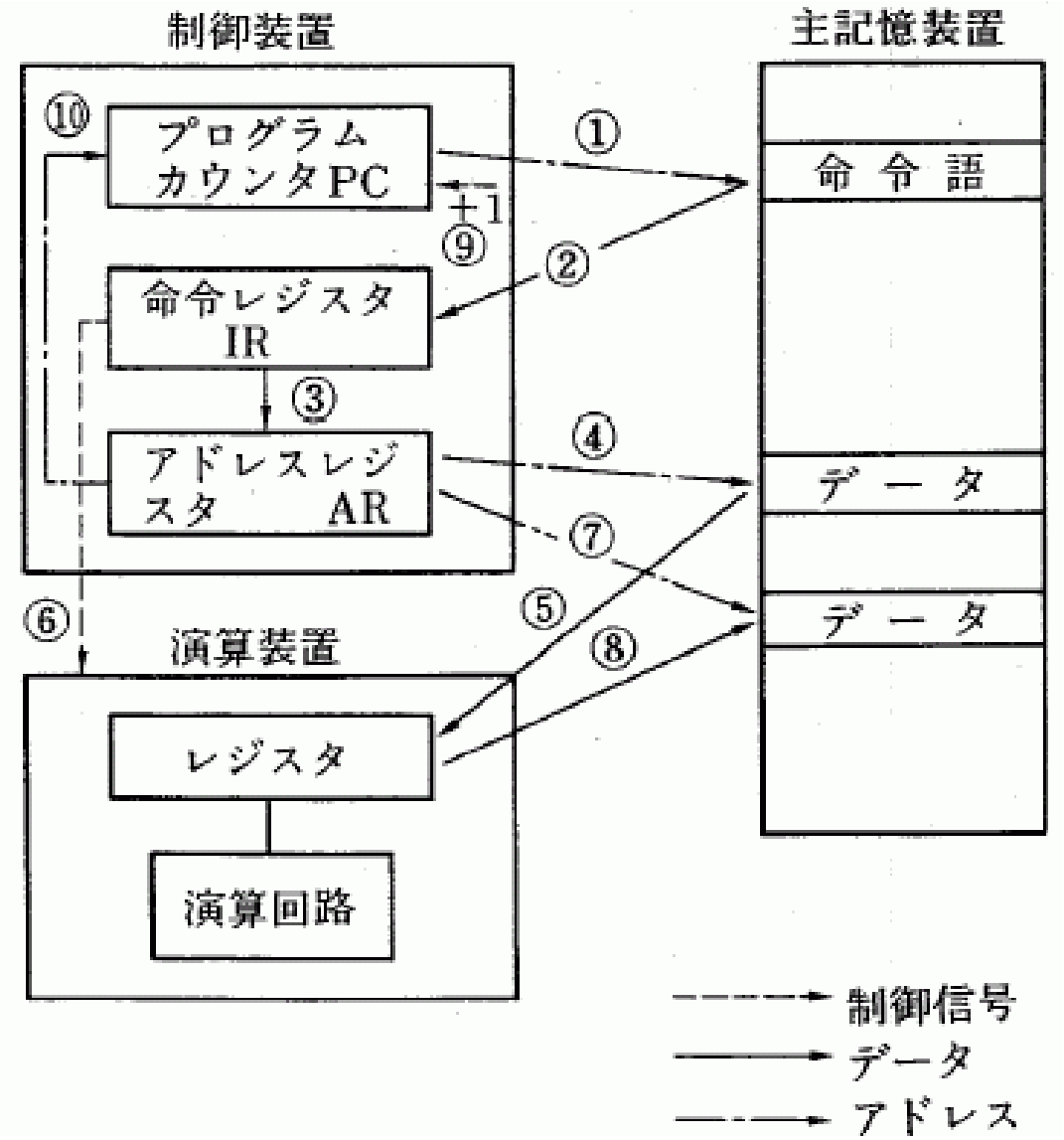
CPUの構造と基本動作

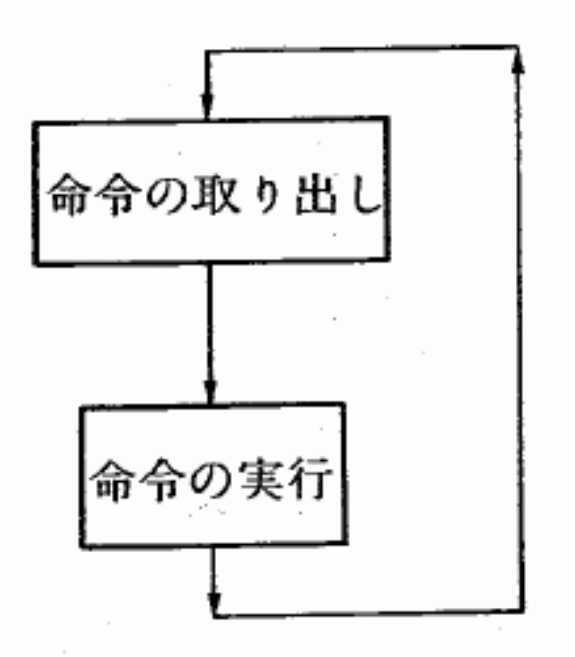
パイプライン

条件分岐

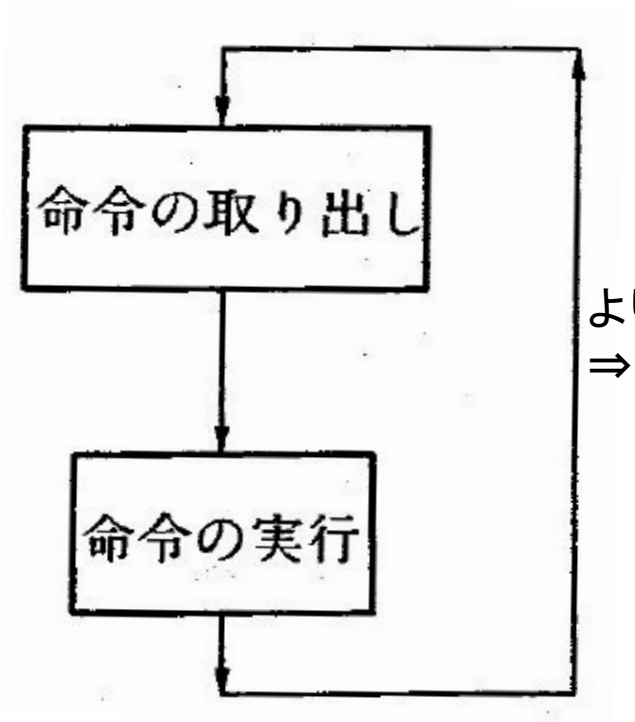
割り込み

命令の実行過程


制御装置
主記憶装置
⑩
プログラム
カウンタPC
+1
⑨
①
②
命令レジスタ
IR
③
アドレスレジスタ
AR
④
⑦
⑤
⑧
⑥
命令語
データ
データ
演算装置
レジスタ
演算回路
制御信号
データ
アドレス


命令の取り出し
命令の実行

命令の実行過程



より詳細には
⇒

命令の読み取り

命令の解読

データの読み取り

命令の実行

結果の書き戻し

各段階で動いている場所は違う

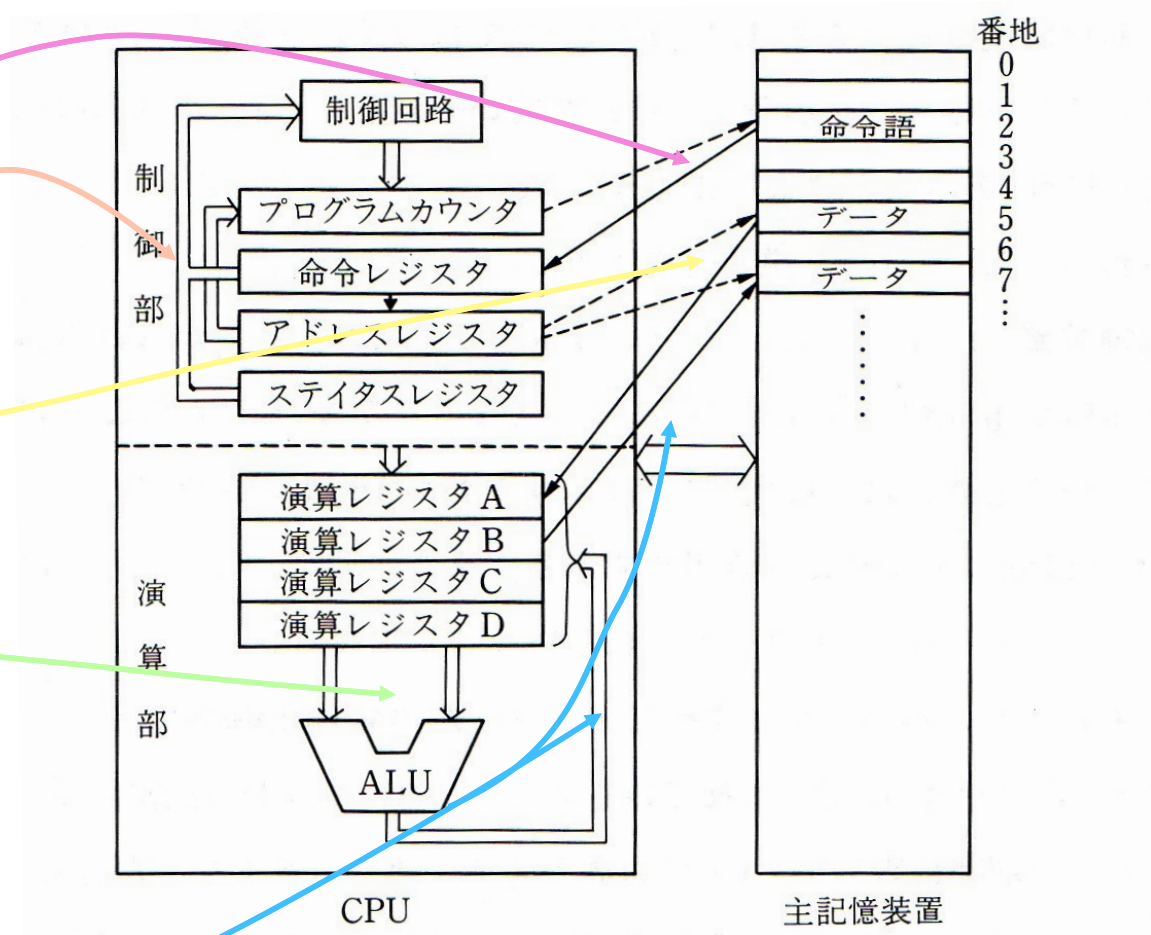
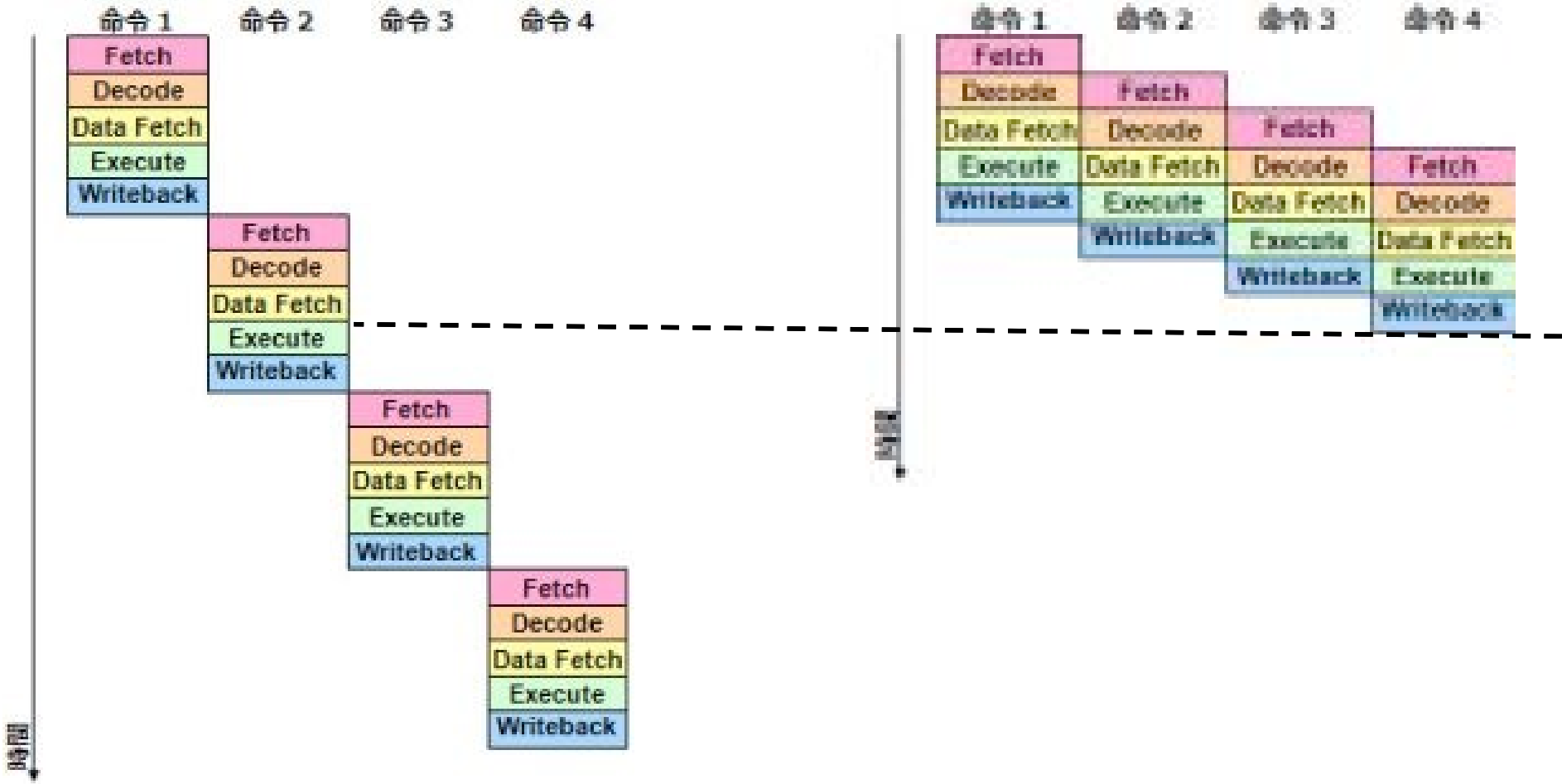


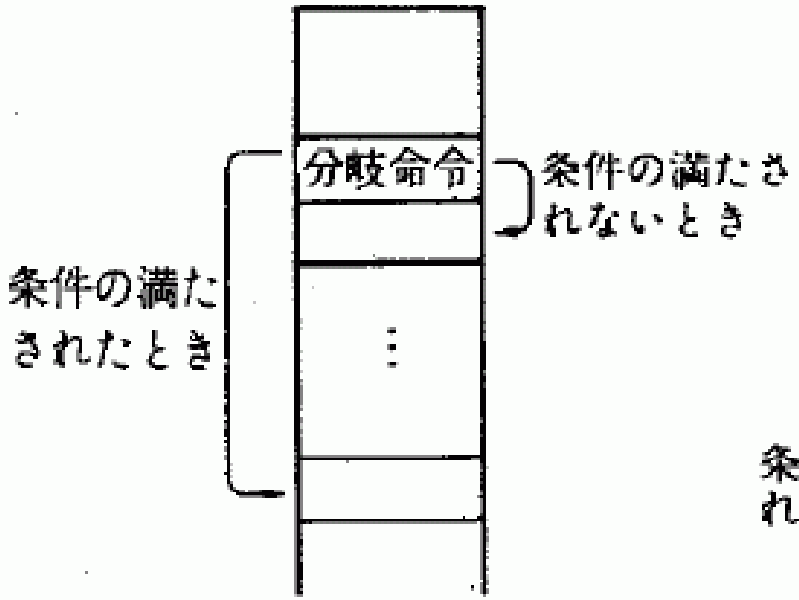
図 3.1　コンピュータの基本構造

パイプライン処理

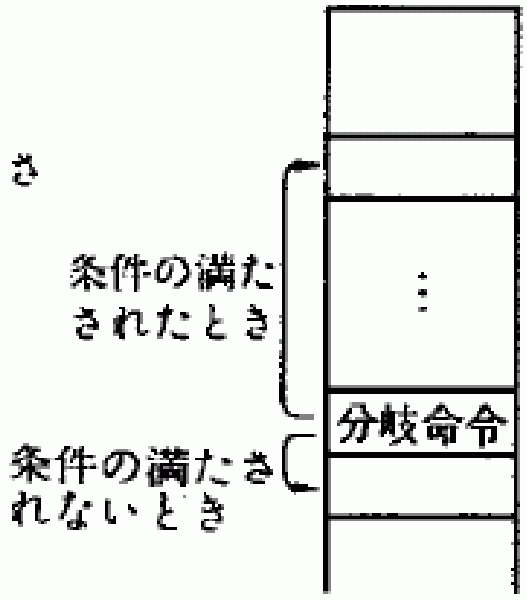


CPUの各部分を並列に動かす．異なる命令に関する処理が同時に進む．

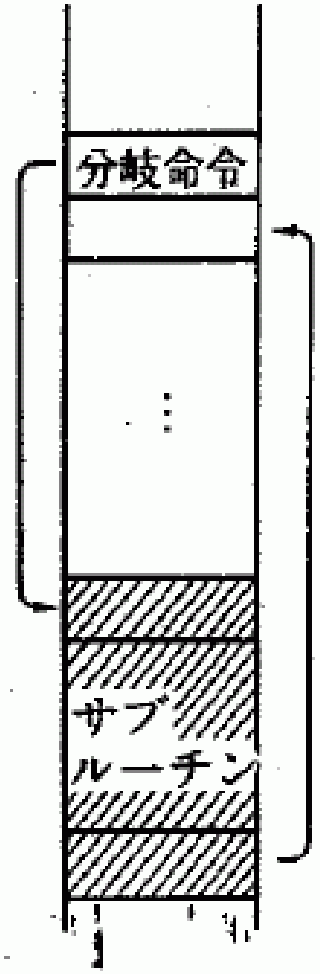
プログラムの制御構造



(a)　ジャンプ



(b)　ループ



(c)　サブルーチン

条件分岐がなければ. . .

- ある計算メカニズムが万能チューリングマシンと同じ計算能力をもつとき、その計算モデルはチューリング完全（Turing-complete）と呼ぶ.
- 条件分岐がなければチューリング完全とはならない.

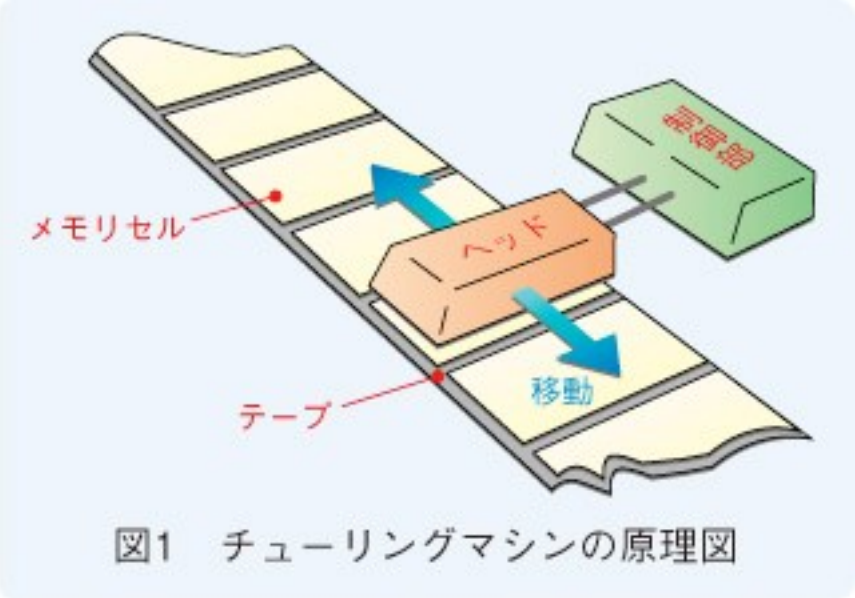
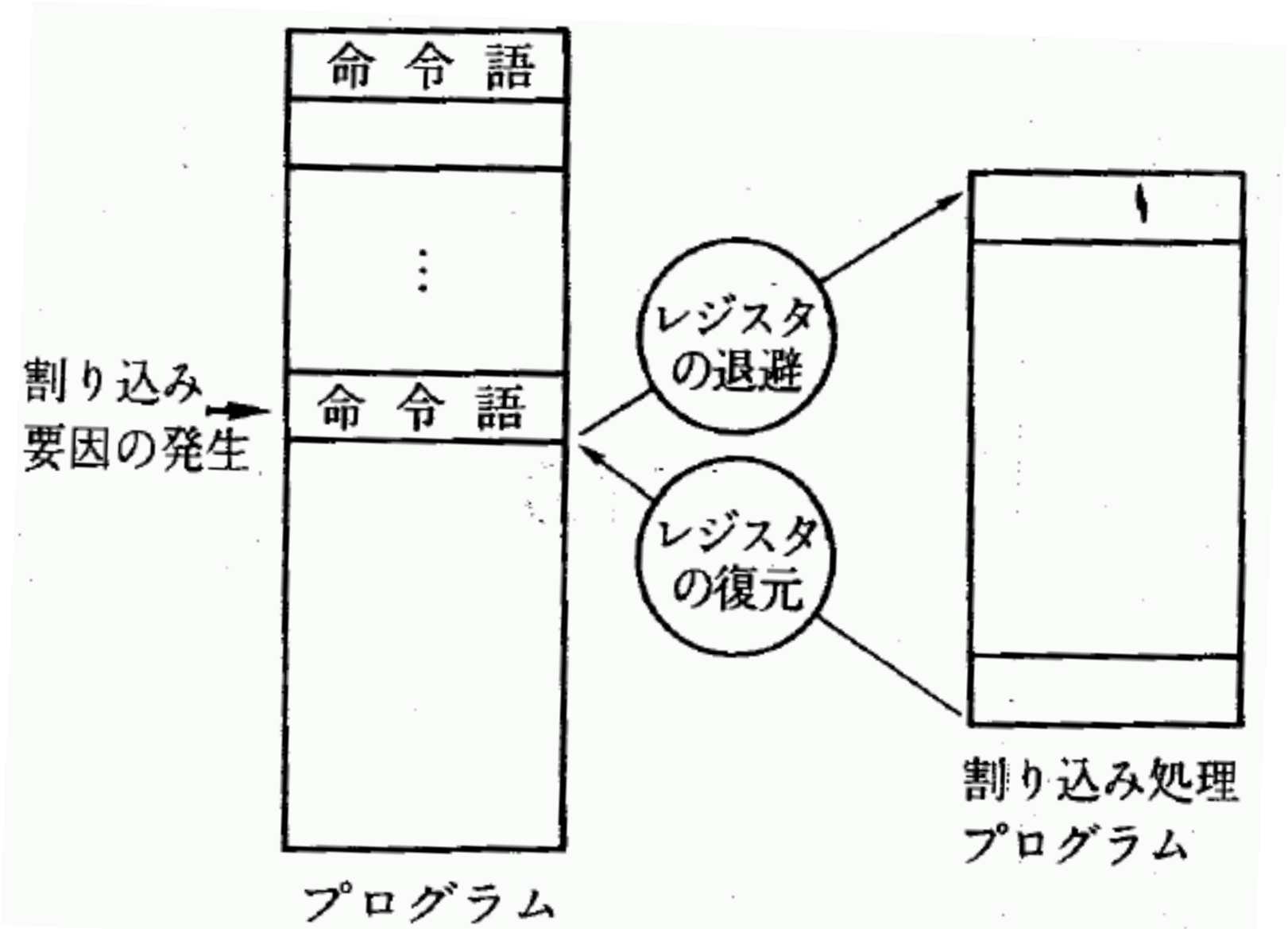


図1　チューリングマシンの原理図

【万能チューリングマシン】
任意のチューリングマシンの動作を再現するもの

割り込み



割り込みはどの様な時に使われる？

- 外部イベントの検出：

  例：ボタンが押された，HDDに対するデータの転送が完了した，外部から通信メッセージが届いた．等

  外部イベント検出の方式

  –ポーリング　イベントが起きていないかどうかをCPUが常に監視する

  –割り込み　　CPUに対して割り込み信号が入ることにより，通常の処理を中断して割り込み処理に移る．割り込み処理後は元の処理を再開する．

ポーリングv.s.割り込み

- 御用聞きに回る三河屋と，電話で注文を受けて作って配達するPIZZA屋さん．どっちが賢い？
  - ポーリングはCPUの無駄遣い．
  - 割り込みは，発生時までCPUを別用途で使える．

計算機に対する命令

＝プログラム

機械語とアセンブリ言語1

```
0000
0000  B8 ---- R
0003  8E D8
0005  B4 09
0007  BA 0000 R
000A  CD 21
000C  B4 4C
000E  CD 21
0010
0000
0000  0D 0A
0002  48 65 6C 6C 6F 2C
      20 77 6F 72 6C 64
      21
000F  0D 0A 24
0012
```

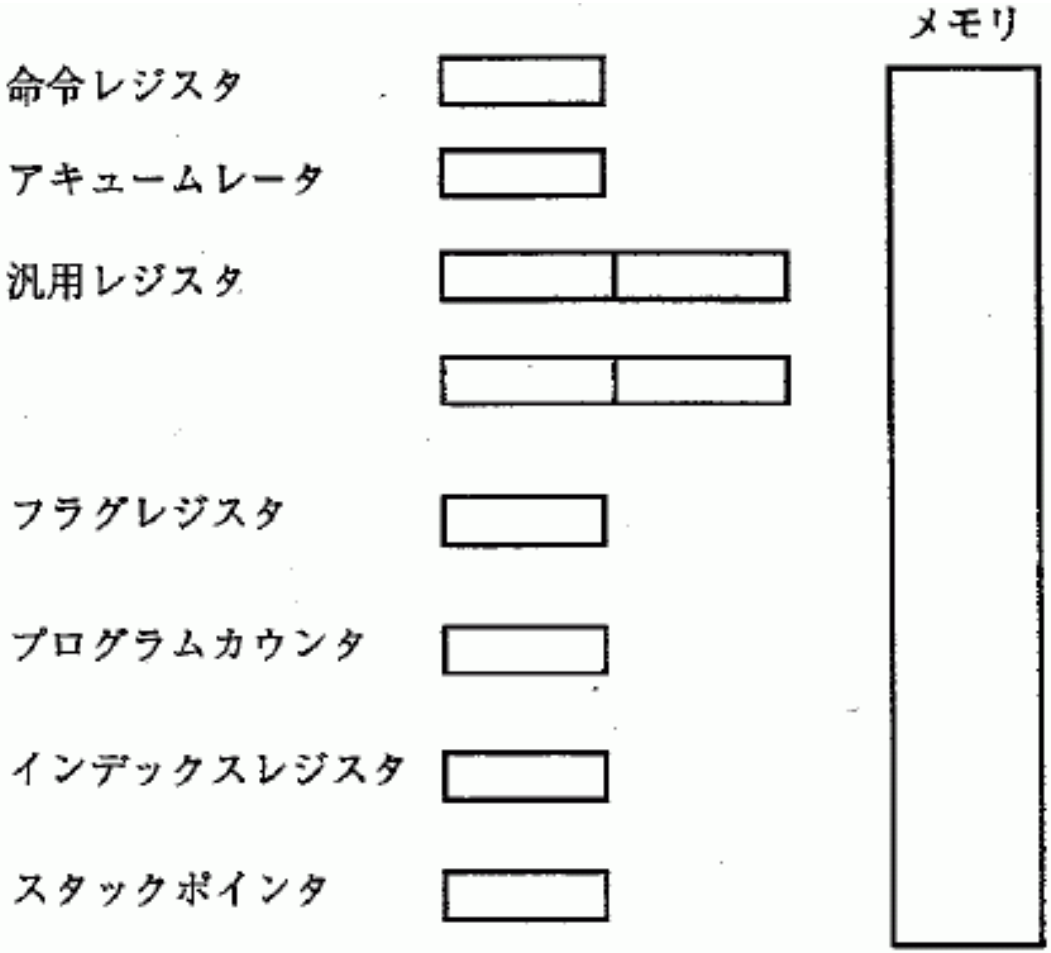
（a） 機械語によるプログラム

```
CODE   SEGMENT
       ASSUME CS : CODE, DS : DATA,
       SS : STACK
       MOV     AX, DATA
       MOV     DS, AX
       MOV     AH, 9
       MOV     DX, OFFSET MSG
       INT     21H
       MOV     AH, 4CH
       INT     21H
CODE   ENDS
DATA   SEGMENT
MSG    DB      CR, LF
       DB      'Hello, world ! '
       DB      CR, LF,'$'
DATA   ENDS
```

（b） アセンブリ言語によるプログラム

図 3.3 機械語のプログラムとアセンブリ言語のプログラム

機械語の実行過程




```
MOV AX, 3F2AH          (B8 3F2A)
MOV AL, [0103H]        (A0 0103)
MOV [0105H], AX        (A3 0105)
MOV [0105H], 203FH
ADD AX, BX
INT 21H
JL A
CALL KEYBOARD
```

16進数3F2A をAXレジスタに代入
0103番地の内容をALレジスタに代入
AXレジスタの内容を0105番地代入
・・・

高級言語

表 3.1　代表的な高級言語の例

FORTRAN	PL/I
DO 10 I=1,10000 READ*,X IF (X.GT.MAX) MAX=X 10 CONTINUE	L: DO I=1 TO 10000; GET LIST(X); IF X>MAX THEN MAX=X; END L;
BASIC	**COBOL**
10 FOR I=1 TO 10000 20 INPUT X 30 IF X>MAX THEN MAX=X 40 NEXT I	PROCEDURE DIVISION. L. ACCEPT X. IF X> MAX MOVE X TO MAX. GO TO L.
C	**APL**

for(i=1;i<=10000;i++){
scanf("%d", &x);
if(x>max) max=x;
}
	X←3 1 17 5 10 4 9 7 14 6
	⌈/X

PASCAL	**LISP**
for i:=1 **to** 10000 **do** **begin** read(x); **if** x>max **then** max:=x **end**;	(defun what-day(day) (cond ((member day '(月 火 水 木 金)) '平日)((member day '(土 日)) '週末)(t 'エラー)))
ALGOL	**PROLOG**
for i:=1 **until** 10000 **do** **begin** ininteger(x); **if** x>max **then** max:=x **end**;	human(Socrates). mortal(X) :- human(X).

プログラムの作成から実行まで

コンパイラ

プログラムテキスト

実行プログラム

アセン

リンカ

```
program example(output);
 var i, sum : integer
begin
  sum := 0;
  for i :=1 to 100 do
+i;
  writeln(sum)
end.
```

```
main:
.globl  PASCALMAIN
        .type   PASCALMAIN,@function
PASCALMAIN:
.globl  program_init
        .type   program_init,@function
program_init:
        pushl   %ebp
        movl    %esp,%ebp
        subl    $4,%esp
        call    FPC_INITIALIZEUNITS
        movw    $0,_SUM
        movw    $1,_I
        .balign 4,144
.L7:
        movswl  _SUM,%eax
        movswl  _I,%edx
        addl    %eax,%edx
        movw    %dx,_SUM
        cmpw    $100,_I
        jge     .L6
        incw    _I
        jmp     .L7
```

```
457f 464c 0101 0001 0000 0000 0000 0000
0002 0003 0001 0000 8080 0804 0034 0000
b43c 0000 0000 0000 0034 0020 0002 0028
0005 0004 0001 0000 0000 0000 8000 0804
8000 0804 ab00 0000 ab00 0000 0005 0000
1000 0000 0001 0000 b000 0000 3000 0805
3000 0805 0420 0000 0e00 0004 0006 0000
1000 0000 0000 0000 0000 0000 0000 0000
8959 89e3 40c8 e0c1 0102 83e0 f8e4 f8a3
053d 8908 3c0d 0534 8908 481d 0534 9b08
e3db d99b 002d 0530 3108 e8ed a9d0 0000
                . . .
0000 0000 000b 0000 0001 0000 0006 0000
8080 0804 0080 0000 aa80 0000 0000 0000
0000 0000 0010 0000 0000 0000 0011 0000
0001 0000 0003 0000 3000 0805 b000 0000
0420 0000 0000 0000 0000 0000 0004 0000
0000 0000 0017 0000 0008 0000 0003 0000
3420 0805 b420 0000 09e0 0004 0000 0000
0000 0000 0010 0000 0000 0000 0001 0000
0003 0000 0000 0000 0000 0000 b420 0000
001c 0000 0000 0000 0000 0000 0001 0000
                0000 0000
```

プログラムの例（Loop）

- １から１００までの和を求めるプログラム。

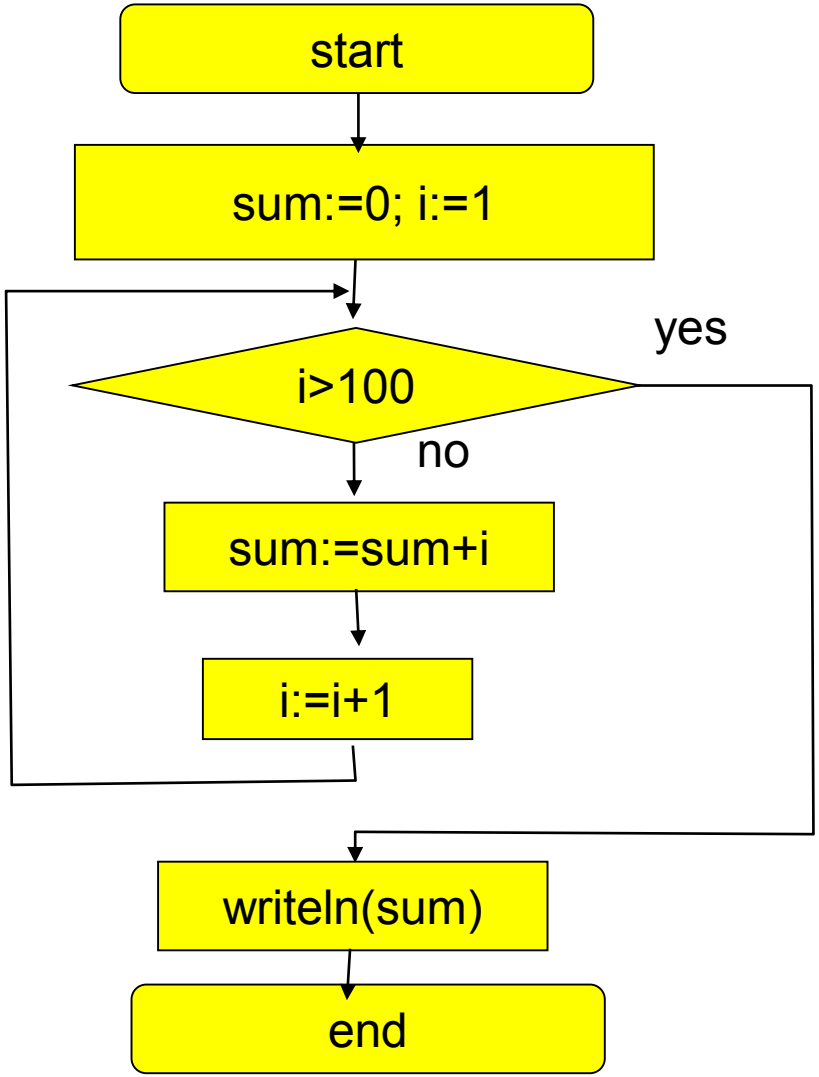
```
program example(output);
 var i, sum : integer;
begin
  sum := 0;
  for i :=1 to 100 do
                sum := sum +i;
  writeln(sum)
end.
```

結果を格納する変数の初期化

この部分をi=1から
i=100まで繰り返す

結果を画面に出力

フローチャートによる計算の記述




```
program example(output);
 var i, sum : integer;
begin
  sum := 0;
  for i :=1 to 100 do
                sum := sum +i;
  writeln(sum)
end.
```

プログラムの例(if 文)

- 最大値を求めるプログラム。

```
program example(input,output);
 var i, x, max: integer;
begin
   x := 0; max := 0;
   for i := 1 to 10 do
   begin
      read(x);
      if x > max then max := x
   end;
   writeln('maximum=',max)
end.
```

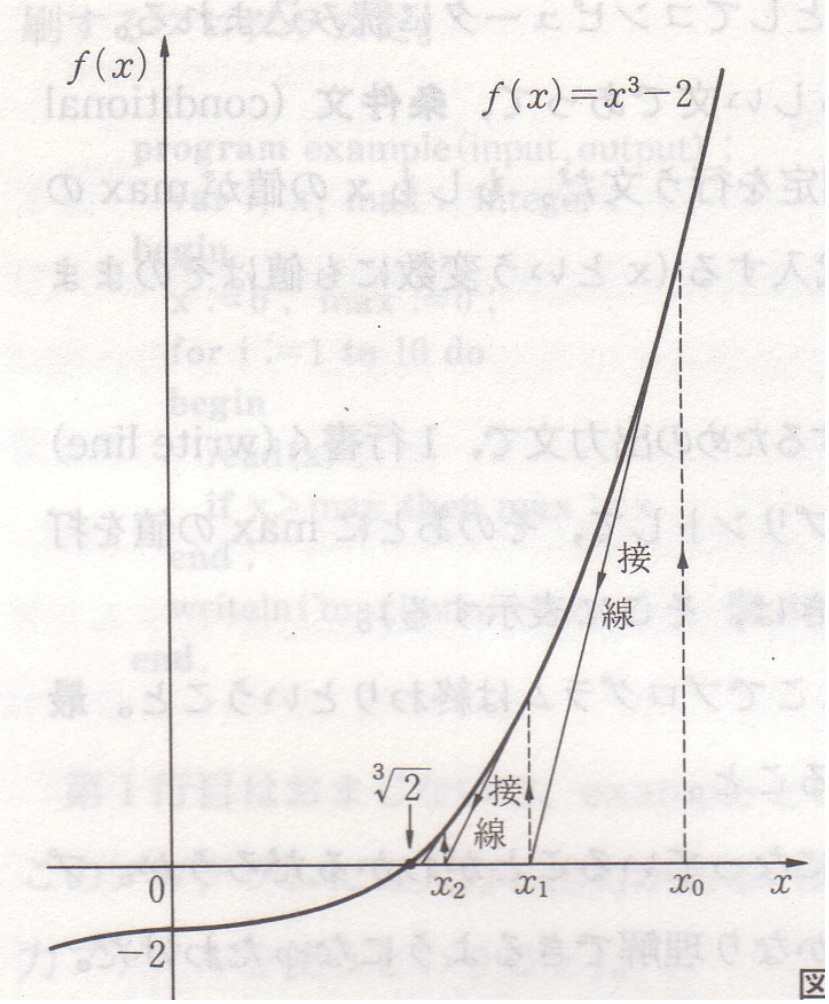
結果を格納する変数の初期化

入力された数値をxに格納する

最大値の更新

結果を画面に出力

プログラムの例(数値計算)




```
program Newton(output);
  var x, x1 : real;
begin
  x := 1.0;
  repeat
    x1 := x;
    x := (2.0/3.0) * (x1+1.0/sqr(x1))
  until abs(x-x1) < 1.0e-11;
  writeln('cubic root of 2=',x)
end.
```

x1²を表している

プログラムの例(数値計算)解説

$f(x)=x^3-2$ の $x=x_1$ での接線の方程式 $g(x)$ を求める

$$g(x)=f'(x_1)(x-x_1)+f(x_1)$$
$$=3x_1^2(x-x_1)+x_1^3-2$$
$$=3x_1^2x-2x_1^3-2$$

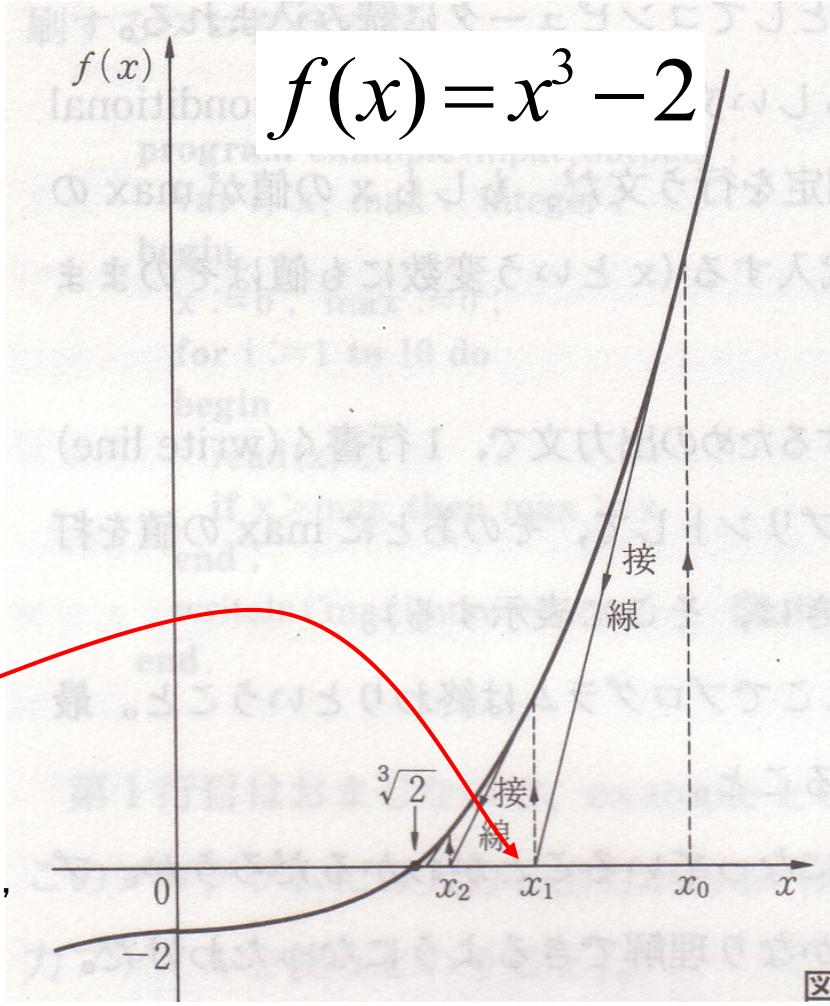
接線が x 軸と交わる位置は,

$$g(x)=3x_1^2x-2x_1^3-2=0$$

から

$$x=\frac{2}{3}\left(x_1+\frac{1}{x_1^2}\right)$$

この x を新たな x_1 として, 上の計算を繰り返すと,
答えが求められる.



この講義がカバーする2つの内容

- OSの話
  - コンピュータのリソース（計算機資源）の細部まで知らなくても使えるようにするための基本ソフトウエア．→リソースの抽象化・仮想化
- 言語処理系の話
  - コンピュータはどうやって「高級言語」を「機械語」に変換しているのか？→字句解析，構文解析

この範囲での他のトピック
次回以降の講義で説明

ＴＳＳ：時分割実行

パイプラインの乱れと投機的実行

物理アドレス空間とユーザアドレス空間

仮想記憶

問題

- メモリを搭載したプリンタに，コンピュータから印字データを送る時に，
  - ポーリングを用いた場合
  - 割り込みを用いた場合

のそれぞれで，動作はどの様になるか文章で説明しなさい．