

プログラム内蔵方式 (von Neumann 型)

プログラム・データを共に
メモリ (主記憶装置) 上に置く

実行の流れ: 以下を繰り返す

- プログラムを一命令ずつ読み出す
- 命令の実行

命令の形式: 命令の種類 + 番地

必要な構成要素

- 主記憶装置 (main memory)
- プログラムカウンタ (PC)
- 命令レジスタ (IR)
- アキュムレータ (Acc)(汎用レジスタ)
- フラグレジスタ (FR)

- 番地解読回路・命令解読回路
- 演算回路 (補助演算回路)

- パルス発生器

命令の形式: 命令の種類 + 番地

命令の種類:

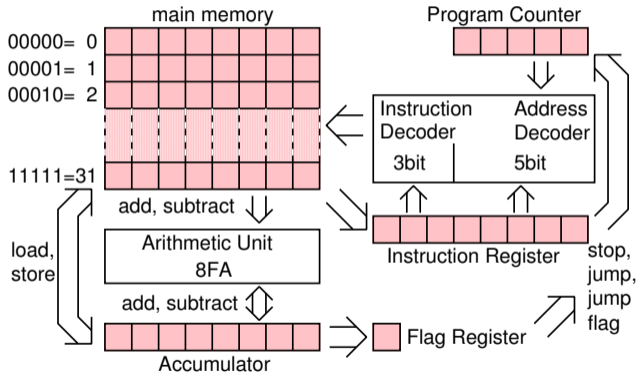
- 主記憶からアキュムレータへの読出し (load)
- アキュムレータから主記憶への書込み (store)
- 演算 (add, subtract)
- 実行制御 (jump, jump flag)
条件分岐の判断 → フラグレジスタ
- 終了 (stop)

→ これらを論理回路で実装すれば良い

説明用の簡易モデルの仕様

- 1 **word** = 8 **bit**(= 1 **byte**)
 - 一度に扱うデータの大きさ
 - 主記憶 1 番地分・**Acc**・**IR**・演算回路の大きさ
- 命令部: 3 **bit**、番地部: 5 **bit**
 - ★ 命令: 二進 3 桁に符号化
load, store, add, subtract,
jump, jump flag, stop.
 - ★ 番地: 0 番地から 31 番地まで
 - 主記憶 32 **byte**、**PC** 5 **bit**

説明用の簡易モデルの仕様



説明用の簡易モデルによる計算の実行

予め

- 主記憶にプログラム・データを
- PC に実行開始番地 (通常 0 番地) を
それぞれ書き込んでおいて、

パルスが発生させると動作する。

プログラム・データ: bit 列 (機械語)

説明用の簡易モデルの機械語

命令の符号化 (encoding)

命令	番号	符号化
stop	1	001
load	2	010
store	3	011
add	4	100
subtract	5	101
jump	6	110
jump flag	7	111

例: $87 + 26$ を計算する

	番地	機械語		アセンブリ言語	
		二進	十六進		
0	00000	01000100	0x44	load	A
1	00001	10000101	0x85	add	B
2	00010	01100110	0x66	store	C
3	00011	00100000	0x20	stop	0
4	00100	01010111	0x57	A 87	
5	00101	00011010	0x1a	B 26	
6	00110	00000000	0x00	C 0	

アセンブリ言語 (assembly language)

アセンブリ言語 (assembly language):

機械語と一対一対応する符丁 (ニーモニック) で
プログラムを記述したもの

アセンブラ (assembler):

アセンブリ言語で記述されたプログラムを
機械語に直すプログラム

→ CPU の設計 (アーキテクチャ) によって
それぞれ異なる (互換性がない)

例: $87 + 26$ を計算する (実際の動作)

	番地	機械語 二進	アセンブリ 言語
0	00000	01000100	load A
1	00001	10000101	add B
2	00010	01100110	store C
3	00011	00100000	stop 0
4	00100	01010111	A 87
5	00101	00011010	B 26
6	00110	00000000	C 0

PC: 00000 **IR:** ----- **Acc:** -----

例: $87 + 26$ を計算する (実際の動作)

	番地	機械語 二進	アセンブリ 言語
0	00000	01000100	load A
1	00001	10000101	add B
2	00010	01100110	store C
3	00011	00100000	stop 0
4	00100	01010111	A 87
5	00101	00011010	B 26
6	00110	00000000	C 0

PC: 00001 **IR:** 01000100 **Acc:** -----

例: $87 + 26$ を計算する (実際の動作)

	番地	機械語 二進	アセンブリ 言語
0	00000	01000100	load A
1	00001	10000101	add B
2	00010	01100110	store C
3	00011	00100000	stop 0
4	00100	01010111	A 87
5	00101	00011010	B 26
6	00110	00000000	C 0

PC: 00001 **IR:** 01000100 **Acc:** 01010111

例: $87 + 26$ を計算する (実際の動作)

	番地	機械語 二進	アセンブリ 言語
0	00000	01000100	load A
1	00001	10000101	add B
2	00010	01100110	store C
3	00011	00100000	stop 0
4	00100	01010111	A 87
5	00101	00011010	B 26
6	00110	00000000	C 0

PC: 00002 **IR:** 10000101 **Acc:** 01010111

例: $87 + 26$ を計算する (実際の動作)

	番地	機械語 二進	アセンブリ 言語
0	00000	01000100	load A
1	00001	10000101	add B
2	00010	01100110	store C
3	00011	00100000	stop 0
4	00100	01010111	A 87
5	00101	00011010	B 26
6	00110	00000000	C 0

PC: 00002 **IR:** 10000101 **Acc:** 01110001

例: $87 + 26$ を計算する (実際の動作)

	番地	機械語 二進	アセンブリ 言語
0	00000	01000100	load A
1	00001	10000101	add B
2	00010	01100110	store C
3	00011	00100000	stop 0
4	00100	01010111	A 87
5	00101	00011010	B 26
6	00110	00000000	C 0

PC: 00003 **IR:** 01100110 **Acc:** 01110001

例: $87 + 26$ を計算する (実際の動作)

	番地	機械語 二進	アセンブリ 言語
0	00000	01000100	load A
1	00001	10000101	add B
2	00010	01100110	store C
3	00011	00100000	stop 0
4	00100	01010111	A 87
5	00101	00011010	B 26
6	00110	01110001	C 113

PC: 00003 **IR:** 10000101 **Acc:** 01110001

例: $87 + 26$ を計算する (実際の動作)

	番地	機械語 二進	アセンブリ 言語
0	00000	01000100	load A
1	00001	10000101	add B
2	00010	01100110	store C
3	00011	00100000	stop 0
4	00100	01010111	A 87
5	00101	00011010	B 26
6	00110	01110001	C 113

PC: 00004 **IR:** 00100000 **Acc:** 01110001

例: $87 + 26$ を計算する (実際の動作)

	番地	機械語 二進	アセンブリ 言語
0	00000	01000100	load A
1	00001	10000101	add B
2	00010	01100110	store C
3	00011	00100000	stop 0
4	00100	01010111	A 87
5	00101	00011010	B 26
6	00110	01110001	C 113

PC: 00000 **IR:** 00100000 **Acc:** 01110001

例: *A* と *B* とを入れ替える

0		load	A
1		store	C
2		load	B
3		store	A
4		load	C
5		store	B
6		stop	0
7		A	15
8		B	5
9		C	0

Acc: -----

例: A と B とを入れ替える

0		load	A
1		store	C
2		load	B
3		store	A
4		load	C
5		store	B
6		stop	0
7		A	15
8		B	5
9		C	0

Acc: 00001111

例: A と B とを入れ替える

0		load	A
1		store	C
2		load	B
3		store	A
4		load	C
5		store	B
6		stop	0
7		A	15
8		B	5
9		C	15

Acc: 00001111

例: A と B とを入れ替える

0		load	A
1		store	C
2		load	B
3		store	A
4		load	C
5		store	B
6		stop	0
7		A	15
8		B	5
9		C	15

Acc: 00000101

例: A と B とを入れ替える

0		load	A
1		store	C
2		load	B
3		store	A
4		load	C
5		store	B
6		stop	0
7		A	5
8		B	5
9		C	15

Acc: 00000101

例: A と B とを入れ替える

0	load	A
1	store	C
2	load	B
3	store	A
4	load	C
5	store	B
6	stop	0
7	A	5
8	B	5
9	C	15

Acc: 00001111

例: A と B とを入れ替える

0		load	A
1		store	C
2		load	B
3		store	A
4		load	C
5		store	B
6		stop	0
7		A	5
8		B	15
9		C	15

Acc: 00001111

例: A と B とを入れ替える

0	load	A
1	store	C
2	load	B
3	store	A
4	load	C
5	store	B
6	stop	0
7	A	5
8	B	15
9	C	15

Acc: 00001111

例: $C \leftarrow \max\{A, B\}$

0		load	A
1		subtract	B
2		jump flag	M
3		load	A
4		jump	N
5	M	load	B
6	N	store	C
7		stop	0
8	A	5	
9	B	7	
10	C	0	

PC: 00000=**0** **Acc:** -----=**FR:** -

例: $C \leftarrow \max\{A, B\}$

0		load	A
1		subtract	B
2		jump flag	M
3		load	A
4		jump	N
5	M	load	B
6	N	store	C
7		stop	0
8	A	5	
9	B	7	
10	C	0	

PC: 00001 = 1 **Acc:** 00000101 = 5 **FR:** 0

例: $C \leftarrow \max\{A, B\}$

0		load	A
1		subtract	B
2		jump flag	M
3		load	A
4		jump	N
5	M	load	B
6	N	store	C
7		stop	0
8	A	5	
9	B	7	
10	C	0	

PC: 00002=₂ **Acc:** 11111110=₂ **FR:** 1

例: $C \leftarrow \max\{A, B\}$

0		load	A
1		subtract	B
2		jump flag	M
3		load	A
4		jump	N
5	M	load	B
6	N	store	C
7		stop	0
8	A	5	
9	B	7	
10	C	0	

PC: 00101 = 5 **Acc:** 11111110 = -2 **FR:** 1

例: $C \leftarrow \max\{A, B\}$

0		load	A
1		subtract	B
2		jump flag	M
3		load	A
4		jump	N
5	M	load	B
6	N	store	C
7		stop	0
8	A	5	
9	B	7	
10	C	0	

PC: 00110 = 6 **Acc:** 00000111 = 7 **FR:** 0

例: $C \leftarrow \max\{A, B\}$

0		load	A
1		subtract	B
2		jump flag	M
3		load	A
4		jump	N
5	M	load	B
6	N	store	C
7		stop	0
8	A	5	
9	B	7	
10	C	7	

PC: 00111= 7 **Acc:** 00000111= 7 **FR:** 0

例: $C \leftarrow \max\{A, B\}$

0		load	A
1		subtract	B
2		jump flag	M
3		load	A
4		jump	N
5	M	load	B
6	N	store	C
7		stop	0
8	A	5	
9	B	7	
10	C	7	

PC: 00000 = 0 **Acc:** 00000111 = 7 **FR:** 0

例: 15×3 を計算する

0	LOOP	load	B	01001010	4a
1		subtract	D	10101100	ac
2		jump flag	END	11101000	e8
3		store	B	01101010	6a
4		load	C	01001011	4b
5		add	A	10001001	89
6		store	C	01101011	6b
7		jump	LOOP	11000000	c0
8	END	stop	0	00100000	20
9	A	15		00001111	0f
10	B	3		00000101	05
11	C	0		00000000	00
12	D	1		00000001	01

0	LOOP	load	B
1		subtract	D
2		jump flag	END
3		store	B
4		load	C
5		add	A
6		store	C
7		jump	LOOP
8	END	stop	0
9	A	15	
10	B	3	
11	C	0	
12	D	1	

PC: 00001= 1 **Acc:** 00000011= 3 **FR:** 0

0	LOOP	load	B
1		subtract	D
2		jump flag	END
3		store	B
4		load	C
5		add	A
6		store	C
7		jump	LOOP
8	END	stop	0
9	A	15	
10	B	3	
11	C	0	
12	D	1	

PC: 00010= 2 **Acc:** 00000010= 2 **FR:** 0

0	LOOP	load	B
1		subtract	D
2		jump flag	END
3		store	B
4		load	C
5		add	A
6		store	C
7		jump	LOOP
8	END	stop	0
9	A	15	
10	B	3	
11	C	0	
12	D	1	

PC: 00011= 3 **Acc:** 00000010= 2 **FR:** 0

0	LOOP	load	B
1		subtract	D
2		jump flag	END
3		store	B
4		load	C
5		add	A
6		store	C
7		jump	LOOP
8	END	stop	0
9	A	15	
10	B	2	
11	C	0	
12	D	1	

PC: 00100= 4 **Acc:** 00000010= 2 **FR:** 0

0	LOOP	load	B
1		subtract	D
2		jump flag	END
3		store	B
4		load	C
5		add	A
6		store	C
7		jump	LOOP
8	END	stop	0
9	A	15	
10	B	2	
11	C	0	
12	D	1	

PC: 00101 = 5 **Acc:** 00000000 = 0 **FR:** 0

0	LOOP	load	B
1		subtract	D
2		jump flag	END
3		store	B
4		load	C
5		add	A
6		store	C
7		jump	LOOP
8	END	stop	0
9	A	15	
10	B	2	
11	C	0	
12	D	1	

PC: 00110 = 6 **Acc:** 00001111 = 15 **FR:** 0

0	LOOP	load	B
1		subtract	D
2		jump flag	END
3		store	B
4		load	C
5		add	A
6		store	C
7		jump	LOOP
8	END	stop	0
9	A	15	
10	B	2	
11	C	15	
12	D	1	

PC: 00111= 7 **Acc:** 00001111= 15 **FR:** 0

0	LOOP	load	B
1		subtract	D
2		jump flag	END
3		store	B
4		load	C
5		add	A
6		store	C
7		jump	LOOP
8	END	stop	0
9	A	15	
10	B	2	
11	C	15	
12	D	1	

PC: 00000 = 0 **Acc:** 00001111 = 15 **FR:** 0

0	LOOP	load	B
1		subtract	D
2		jump flag	END
3		store	B
4		load	C
5		add	A
6		store	C
7		jump	LOOP
8	END	stop	0
9	A	15	
10	B	2	
11	C	15	
12	D	1	

PC: 00001= 1 **Acc:** 00000010= 2 **FR:** 0

中略

3 巡回ると

0	LOOP	load	B
1		subtract	D
2		jump flag	END
3		store	B
4		load	C
5		add	A
6		store	C
7		jump	LOOP
8	END	stop	0
9	A	15	
10	B	0	
11	C	45	
12	D	1	

PC: 00001 = 1 **Acc:** 00000000 = 0 **FR:** 0

0	LOOP	load	B
1		subtract	D
2		jump flag	END
3		store	B
4		load	C
5		add	A
6		store	C
7		jump	LOOP
8	END	stop	0
9	A	15	
10	B	0	
11	C	45	
12	D	1	

PC: 00010=**2** **Acc:** 11111111=**-1** **FR:** 1

0	LOOP	load	B
1		subtract	D
2		jump flag	END
3		store	B
4		load	C
5		add	A
6		store	C
7		jump	LOOP
8	END	stop	0
9	A	15	
10	B	0	
11	C	45	
12	D	1	

PC: 01000 = 8 **Acc:** 11111111 = - 1 **FR:** 1

0	LOOP	load	B
1		subtract	D
2		jump flag	END
3		store	B
4		load	C
5		add	A
6		store	C
7		jump	LOOP
8	END	stop	0
9	A	15	
10	B	0	
11	C	45	
12	D	1	

PC: 00000 = 0 **Acc:** 11111111 = - 1 **FR:** 1

演習問題

簡易モデル上でのアセンブリ言語により、
自然数 n に対して、

1 から n までの和
を計算するプログラムを作成せよ

- 実行前 (入力): ラベル A の場所に
数値 n が書き込まれている
- 実行後 (出力): ラベル B の場所に
計算結果の数値が書き込まれている
- ラベル A の場所以外は
入力 n に依存してはならない