

Javaプログラマ育成コース

第一回 コンピュータの原理

ソフトシンク株式会社

代表取締役

周 順彩

zhousc@soft-think.com

目次

- プログラマに必要な能力は？
- 上級プログラマになるには
- コンピュータは何？
- コンピュータの構成は？
- 2進数
- 論理回路
- プログラム
- 練習課題
- 次回の予習タスク

プログラマに必要な能力は？

プログラマに必要なの能力ーロジカルシンキング

◆ ロジカルシンキングとは？

ズバリ論理思考のこと。論理的にものを考えること！

物事の**因果関係を推理**しながら そうなる原因／なるべき結果／問題を解決する方法・・・
を**導き出す**

◆ ロジカルシンキング方程式

- ① AであればBである。
- ② BであればCである。

なので、

- ③ AであればCです。

◆ ロジカルシンキングの事例

プログラマに必要なの能力ーロジカルシンキング

◆ ロジカルシンキング力を高める方法

- 「これは〇〇だ→なぜならば、××だから→…」のクセを身に付ける
- 事実と事実ではないものを分けてみる
 - Javaはプログラム言語である
 - JavaScriptはもプログラム言語である
 - JavaScriptはJavaから派生している
- 物事を分析・分類する
 マインドマップ
- 言葉の曖昧さをなくす
- ものを疑う習性
- 結論から話す
- 考えたことをもの(文章、図面・・・)にする⇒頭で考えるだけではだめ！

プログラマに必要なの能力ーその他

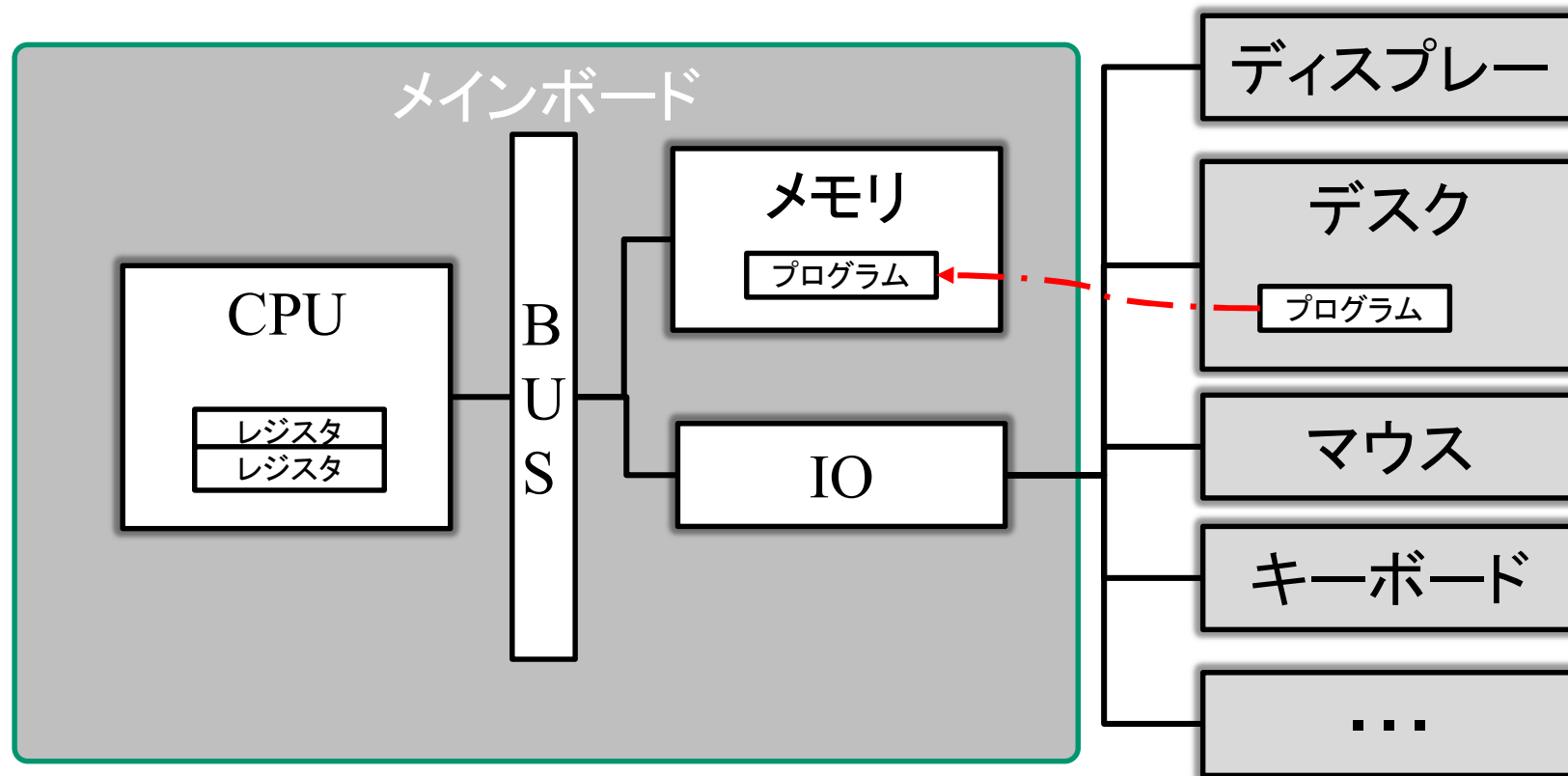
- ◆ コミュニケーション力
- ◆ 勉強力

◆ 努力！

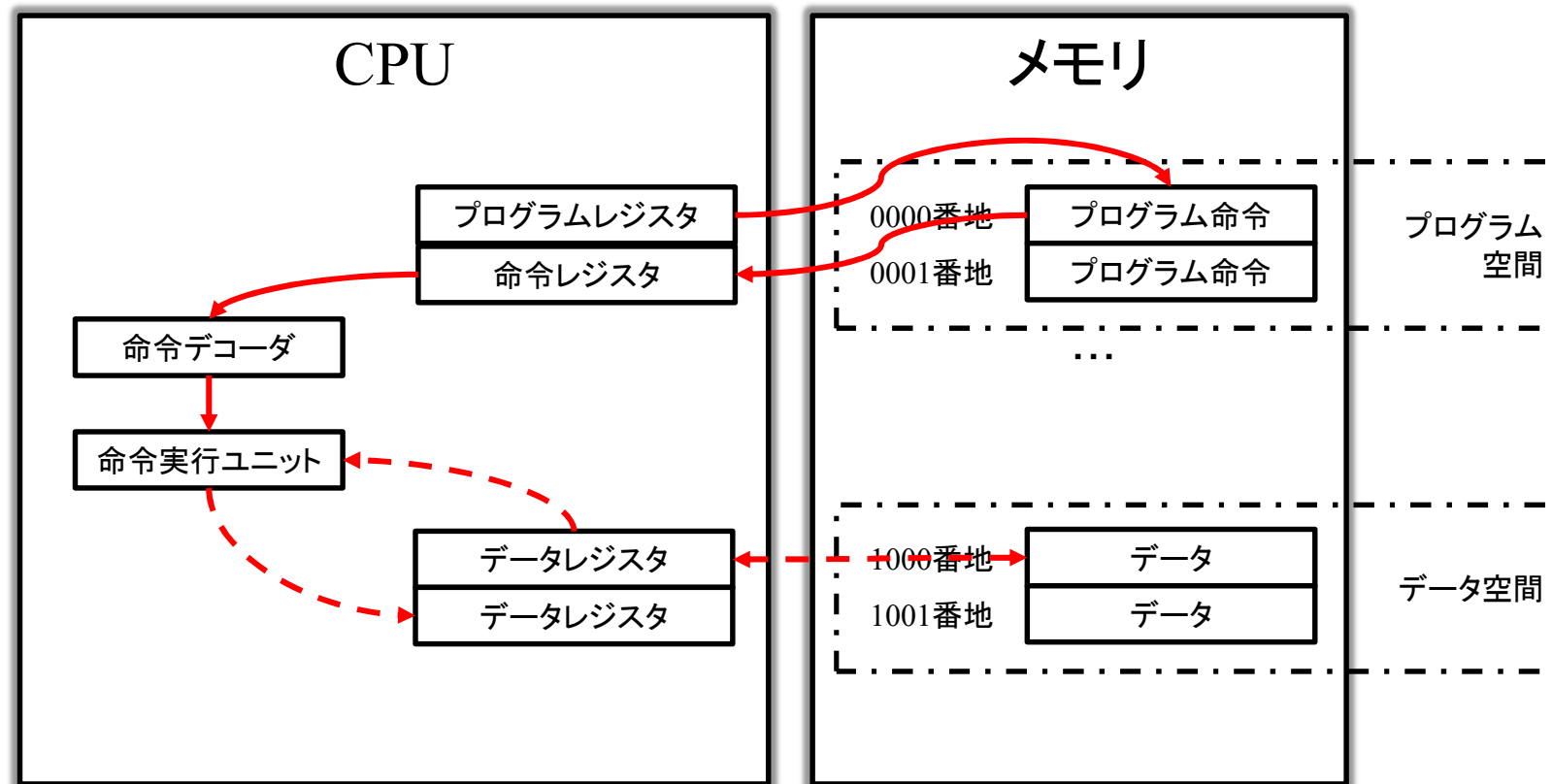
コンピュータとは？

簡単に言えば、自動計算器のこと

コンピュータの構成？



CPUの構成



進数

◆ 10進数

1、10、100、1000、10000...

◆ その他の進数

- 12進数(12:ダース、12*12:グロス)
- 秒、分、時
- 週
- 日、月、年
- ...

◆ コンピュータで使っている進数は？

2進数

なぜ2進数？

◆ 10進数の足し算、掛け算(九九表)

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1									10
2	2									11
3	3									12
4	4									13
5	5									14
6	6									15
7	7									16
8	8									17
9	9	10	11	12	13	14	15	16	17	18

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0									9
2	0									18
3	0									27
4	0									36
5	0									45
6	0									54
7	0									63
8	0									72
9	0	9	18	27	36	45	54	63	72	81

◆ 2進数の足し算、掛け算(一一表)

	0	1
0	0	1
1	1	10

	0	1
0	0	0
1	0	1

2進数、8進数、16進数

10進数	2進数	8進数	16進数
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

2進数桁(ビット)	表せる数
1	2(0, 1)
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10(1K)	1,024
20(1M)	1,048,576
30(1G)	1,073,741,824
32	4,294,967,296
40(1T)	1,099,511,627,776
64	18,446,744,073,709,551,616

10進数と2進数の変換

◆ 10進数から2進数への変換

順次2で割ってそのあまりを並べる。

$$\begin{array}{l} 13/2=6\cdots 1 \\ 6/2=3\cdots 0 \\ 3/2=1\cdots 1 \\ 1/2=0\cdots 1 \end{array} \quad \uparrow$$

結果:1101

◆ 2進数から10進数への変換

2進数の各桁はその下位桁の2倍であるので、各桁の係数をかけたものを足し合わせる。

$$1101(2進)=1*2^3+1*2^2+0*2^1+1*2^0=1*8+1*4+0*2+1*1=13$$

2進数で負の数を表す

- ◆ コンピュータは人間と同じように数の前に符号をつけて表現するのではなく、2進数の最上位桁(MSB:most significant bit)を符号ビット(sign bit)として、負の数を表す

符号なし2進数		符号付き2進数	
2進数	10進数	2進数	10進数
0000	0	<u>0</u> 000	0
0001	1	<u>0</u> 001	1
0010	2	<u>0</u> 010	2
0011	3	<u>0</u> 011	3
0100	4	<u>0</u> 100	4
0101	5	<u>0</u> 101	5
0110	6	<u>0</u> 110	6
0111	7	<u>0</u> 111	7
1000	8	<u>1</u> 000	-8
1001	9	<u>1</u> 001	-7
1010	10	<u>1</u> 010	-6
1011	11	<u>1</u> 011	-5
1100	12	<u>1</u> 100	-4
1101	13	<u>1</u> 101	-3
1110	14	<u>1</u> 110	-2
1111	15	<u>1</u> 111	-1

2進数で文字を表す

◆ キャラクタコード

コンピュータが直接扱えるのは2進数だけなので、文字も2進数で表すこととなる。そこで、コンピュータの利用者が共通的に使用できるように、規格化された文字コードのセットをキャラクタコード(character code)といいます。

◆ ASCII

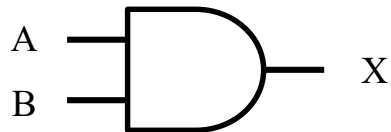
ASCII(アスキー、American Standard Code for Information Interchange)は、現代英語や西ヨーロッパ言語で使われるラテン文字を中心とした文字コード。1963年6月17日に、American Standards Association(ASA、後のANSI)によって制定された。

ASCIIは7ビットコードである。つまり、情報を表すのに7桁の2進数(10進数では0～127)を用いる。

文字	ASCII	10進数	16進数	説明
NUL	000 0000	0	0x0	ヌル文字
HT	000 1001	9	0x9	水平タブ
LF	000 1010	10	0xA	改行
CR	000 1100	12	0xD	復帰(キャリッジリターン)
SP	010 0000	32	0x20	スペース
0	011 0000	48	0x30	ゼロ
A	100 0001	61	0x3D	A
a	110 0001	97	0x61	a

論理回路—AND回路

- ◆ コンピュータは演算回路で計算するが、演算回路は論理回路の組み合わせで作られる。
- ◆ AND回路
複数の入力信号のいずれかが1のときのみ出力が1になる。



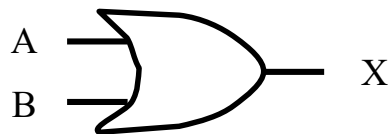
$$X = A \cdot B$$

A	B	X
0	0	
0	1	
1	0	
1	1	

論理回路—OR回路

◆ OR回路

複数の入力信号のいずれかが1のときに出力が1になる。



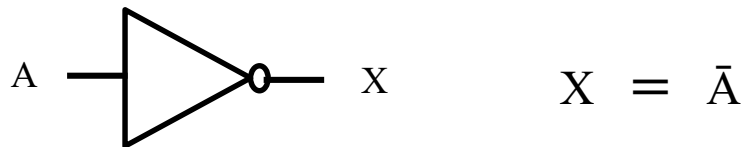
$$X = A + B$$

A	B	X
0	0	
0	1	
1	0	
1	1	

論理回路—NOT回路

◆ NOT回路

入力信号を反転して出力する。

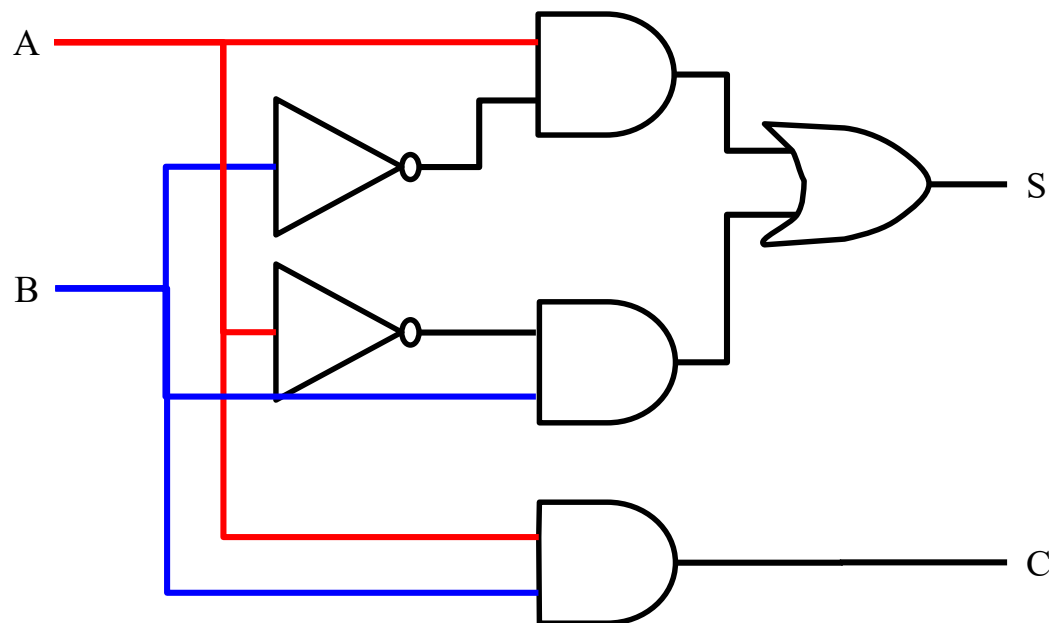


A	X
0	
1	

論理回路一半加算器

◆ 半加算器

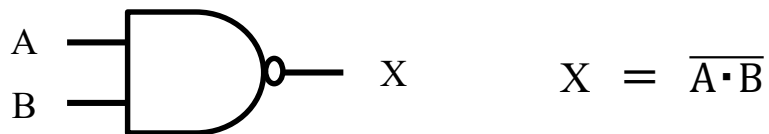
半加算器 (Half Adder) は1桁の2進数、AとBを加算して和はSに、加算の結果生じるキャリーがCに出力する回路。



A	B	C	S
0	0		
0	1		
1	0		
1	1		

論理回路—NAND回路

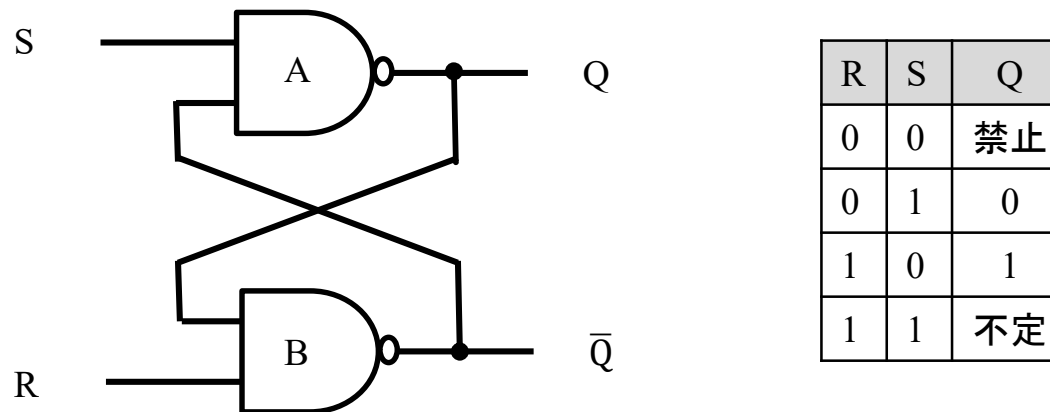
- ◆ NANDはNot of ANDの意味で、AND回路にNOT回路を加えたのと同じ働きになる。



A	B	$A \cdot B$	X
0	0	0	
0	1	0	
1	0	0	
1	1	1	

フリップ・フロップ回路 (flip-flop)

◆ NAND回路を利用して1ビットのデータを記憶する回路。



R(reset)とS(set)の入力は普段はどちらも1にしておく。この場合、出力Qが不定で、0にも1にもなることができ、その状態を永久に保持する。

入力Sを0にすると、NAND回路の真理値表によると、二つの入力がいずれも1のときだけ出力が0になるので、Qは1になる。QはBの入力でもあるので、Bの出力の \bar{Q} は0になる。

\bar{Q} はAの入力でもあるので、そのあと、Sが1に戻るとしても、Aのもう一つの入力が0であるため、Aの出力のQが1のままとなり、つまり記憶することができる記憶回路になる。初期状態でRだけ0にすると、その逆で、記憶値が0になる。

プログラム

- ◆ メモリ上に2進数で書かれた命令がプログラム

- ◆ 機械語プログラム

CPUが実行できる2進数の命令そのもので書かれたプログラム。

- ◆ アセンブリプログラム

人間が機械語でプログラムを書くのは大変なので、機械語を人間が分かりやすい記号(加算ならADD、出力ならOUT)で表記するものがアセンブリ言語。

アセンブリ言語で書かれたプログラムをコンピュータでわかる機械語に変換するのはコンパイラ。

練習課題

◆ 4bitのCPUの機械語命令とアセンブリ語コマンドを設計しよう

- 加算のみ
- プログラムがメモリの0番地から配置
- 4bitのため、メモリ上はMAX16命令またはデータしか保持できない

次回の予習タスク

- ◆ Javaが何？
- ◆ Javaのデータ型、変数定義
- ◆ Javaの演算
- ◆ Javaの制御文