

# 基礎コンピュータ工学

## 第5章 機械語プログラミング

### (パート8)

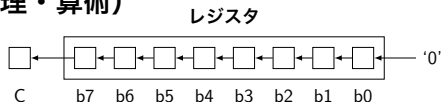
<https://github.com/tctsigemura/TecTextBook>

本スライドの入手：

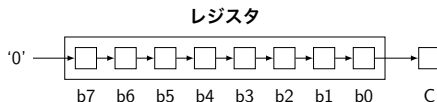


# シフト（桁ずらし）命令

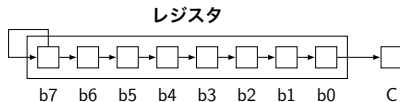
- データの2進数を左右に桁移動する命令のこと。
- TeCは4種類（実質は3種類）の命令を持っている。
- 左シフト（論理・算術）**



- 右シフト（論理）**

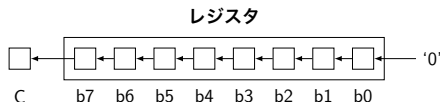


- 右シフト（算術）**



# SHLA (Shift Left Arithmetic) 命令

左算術 (算術 = Arithmetic) シフト命令.  
レジスタの値を左に **1 ビット** ずらす. (シフトする)

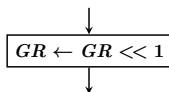


**C フラグ** 上の図のように変化する.

**S フラグ** 結果が負なら 1, それ以外は 0 になる.

**Z フラグ** 結果がゼロなら 1, それ以外は 0 になる.

**フローチャート :** Java のシフト演算子を流用する.



ニーモニック： SHLA GR

命令フォーマット： 1 バイトの長さを持つ.

第 1 バイト	
OP	GR XR
1001 <sub>2</sub>	GR 00 <sub>2</sub>

例：SHLA 命令を実行して確かめる. (イルミネーション?)  
(次のプログラムを G0 を表示したまま STEP 実行する.)

00	10 05		LD	G0, N
02	90	LOOP	SHLA	G0
03	A0 02		JMP	LOOP
05	01	N	DC	1

注：左シフトは  $\times 2$  を計算している.

# SHLL (Shift Left Logical) 命令

左論理（論理＝Logical）シフト命令。  
レジスタの値を左に**1ビット**ずらす。（シフトする）  
（SHLL 命令と SHLA 命令の動作は全く同じ。）

**フラグ** SHLA と同じ

**フローチャート**： SHLA と同じ

**ニーモニック**： SHLL GR

**命令フォーマット**： 1 バイトの長さを持つ。

第1バイト	
OP	GR XR
1001 <sub>2</sub>	<i>GR</i> 01 <sub>2</sub>

# 左シフトを用いた $\times 2$ 計算

## 符号なし数の $\times 2$

0000	0001	(1)
0000	0010	(2)
0000	0100	(4)
0000	1000	(8)
0001	0000	(16)
0010	0000	(32)
0100	0000	(64)
1000	0000	(128)
0000	0000	(ERR)

SHLL 命令はこちら用

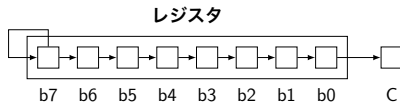
## 符号付き数の $\times 2$

1111	1111	(-1)
1111	1110	(-2)
1111	1100	(-4)
1111	1000	(-8)
1111	0000	(-16)
1110	0000	(-32)
1100	0000	(-64)
1000	0000	(-128)
0000	0000	(ERR)

SHLA 命令はこちら用

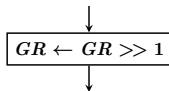
# SHRA (Shift Right Arithmetic) 命令

右算術（算術＝Arithmetic）シフト命令。  
レジスタの値を右に**1ビット**ずらす。（シフトする）



**フラグ** SHLA と同じ

**フローチャート**：Java のシフト演算子を流用する。



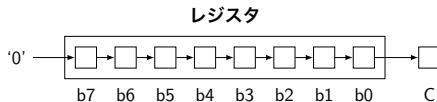
**命令フォーマット**：1 バイトの長さを持つ。

第1バイト	
OP	GR XR
1001 <sub>2</sub>	GR 10 <sub>2</sub>

**注**：SHRA は符号付き数の  $\div 2$  を計算している。

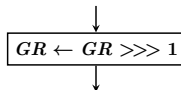
# SHRL (Shift Right Logical) 命令

右論理（論理＝Logical）シフト命令。  
レジスタの値を右に**1ビット**ずらす。（シフトする）



**フラグ** SHLA と同じ

**フローチャート**：Java のシフト演算子を流用する。



**命令フォーマット**：1 バイトの長さを持つ。

第1バイト	
OP	GR XR
1001 <sub>2</sub>	GR 11 <sub>2</sub>

**注**：SHRL は符号なし数の  $\div 2$  を計算している。



# 右シフトを用いた $\div 2$ 計算 (1)

## 符号なし数の $\div 2$

1100	0000	(192)
0110	0000	(96)
0011	0000	(48)
0001	1000	(24)
0000	1100	(12)
0000	0110	(6)
0000	0011	(3)
0000	0001	(1)
0000	0000	(0)

SHRL 命令を使用する

## 符号付き数の $\div 2$

1100	0000	(-64)
1110	0000	(-32)
1111	0000	(-16)
1111	1000	(-8)
1111	1100	(-4)
1111	1110	(-2)
1111	1111	(-1)
1111	1111	(-1)
1111	1111	(-1)

SHRA 命令を使用する

## 右シフトを用いた $\div 2$ 計算 (2)

符号付き正数の  $\div 2$

0100	0000	(64)
0010	0000	(32)
0001	0000	(16)
0000	1000	(8)
0000	0100	(4)
0000	0010	(2)
0000	0001	(1)
0000	0000	(0)
0000	0000	(0)

SHRA 命令使用

符号付き負数の  $\div 2$

1100	0000	(-64)
1110	0000	(-32)
1111	0000	(-16)
1111	1000	(-8)
1111	1100	(-4)
1111	1110	(-2)
1111	1111	(-1)
1111	1111	(-1)
1111	1111	(-1)

SHRA 命令使用

# まとめ

## 学んだこと

- TeC のシフト命令は **1 ビット**シフトする.
- TeC は 4 種類 (実質は 3 種類) のシフト命令を持っている.
- シフト命令はイルミネーション (?) に使用できる.
- **左シフト (論理・算術)** は,  
符号付き・なし兼用の  $\times 2$  計算に使用できる.
- **右シフト (論理)** は, 符号なし数の  $\div 2$  計算に使用できる.
- **右シフト (算術)** は, 符号付き数の  $\div 2$  計算に使用できる.

## 演習

- ビットの右回転 (例題 5 - 5 を参考に)
- シフト命令を使用した「 $\times 7$  の計算」 (例題 5 - 6 を参考に)
- シフト命令を使用した「 $\div 4$  の計算」
- シフト命令を使用した「 $\times 1.5$  の計算」