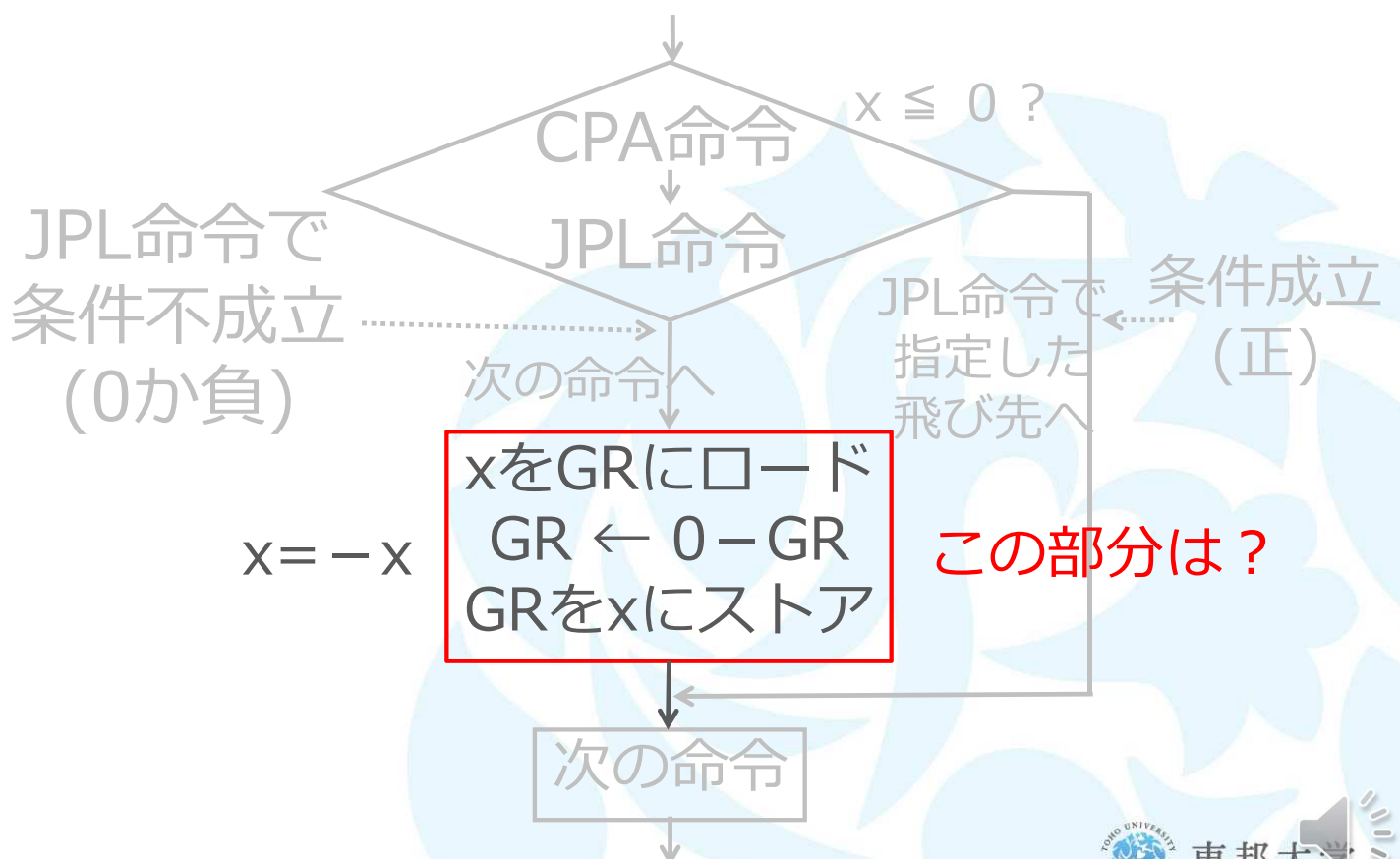


## 条件分岐とIF文(2)



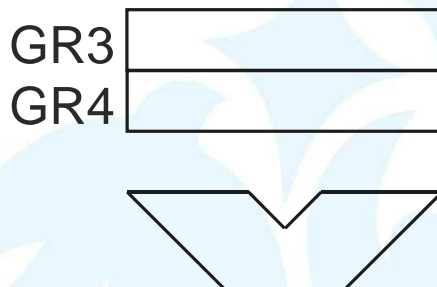
### 命令に分解したイメージ



# 命令に分解したイメージ

$$X = -X$$

0 をGR4にロード  
x をGR3にロード  
GR4 ← GR4 - GR3  
GR4をxにストア



メモリ

x	5
ZERO	0

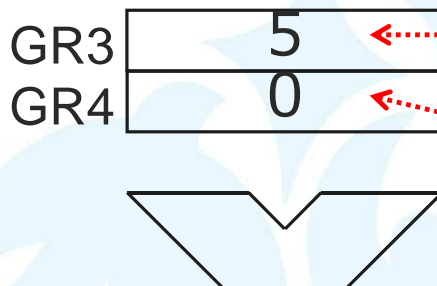
「計算式と代入」と同じ原理

2

# 命令に分解したイメージ

$$X = -X$$

0 をGR4にロード  
x をGR3にロード  
GR4 ← GR4 - GR3  
GR4をxにストア



メモリ

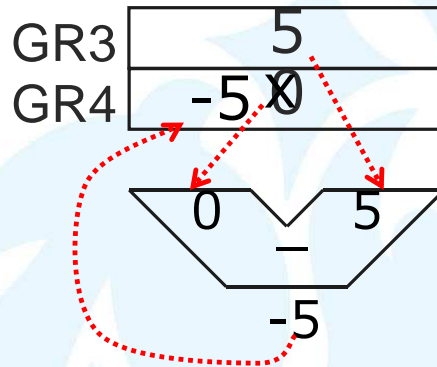
x	5
ZERO	0

3

# 命令に分解したイメージ

$$X = -X$$

0をGR4にロード  
xをGR3にロード  
GR4 ← GR4 - GR3  
GR4をxにストア



メモリ

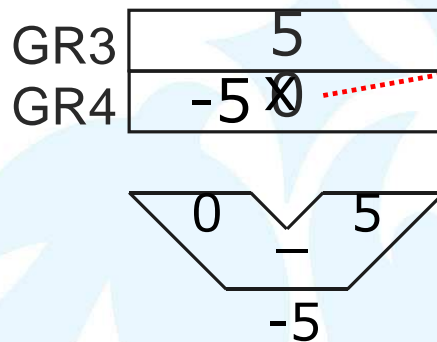
x	5
ZERO	0

4

# 命令に分解したイメージ

$$X = -X$$

0をGR4にロード  
xをGR3にロード  
GR4 ← GR4 - GR3  
GR4をxにストア

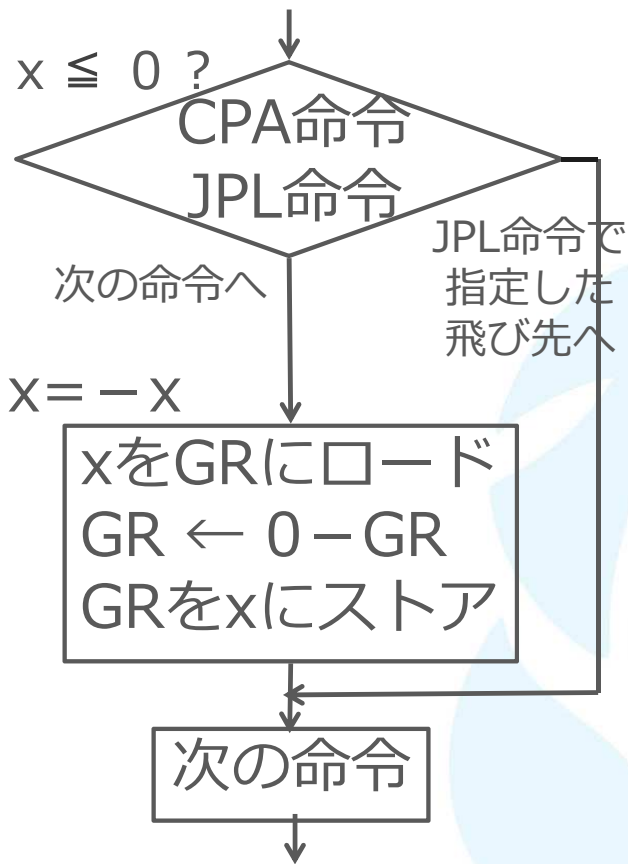


メモリ

x	-5
ZERO	0

5

# 全体のプログラムとしては

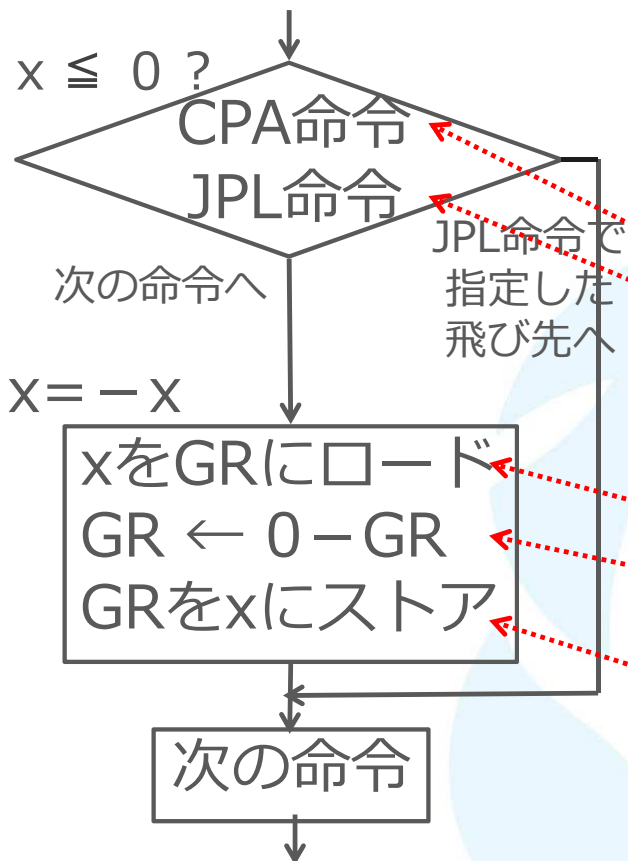


LD GR4, ZERO	GR4 ← 0
LD GR3, X	GR3 ← x
CPA GR3, GR4	比較
JPL L1	x > 0なら L1へ飛ぶ
LD GR4, ZERO	GR4 ← 0
LD GR3, X	GR3 ← x
SUBA GR4, GR3	GR4 - GR3 → GR4
ST GR4, X	X ← GR4

L1: 次の命令



# 全体のプログラムとしては

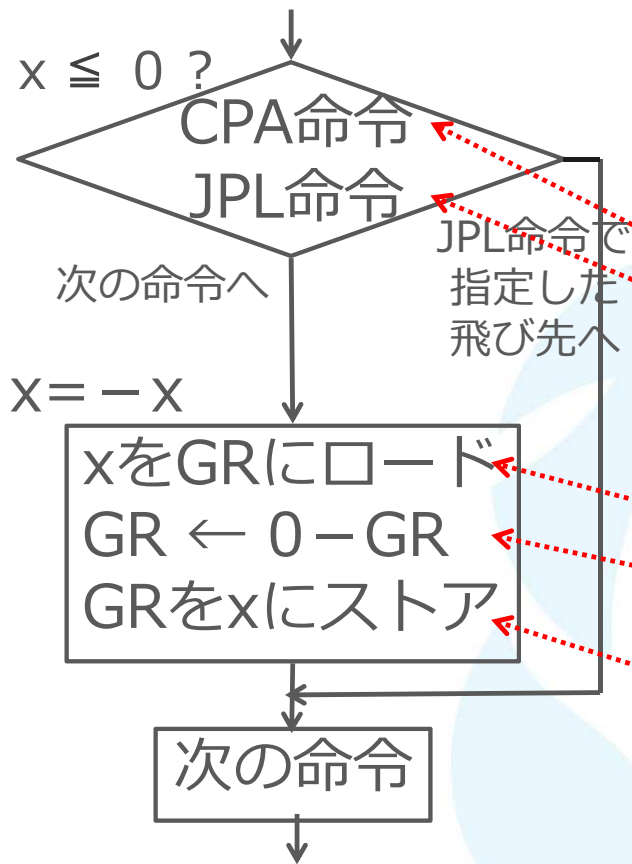


LD GR4, ZERO	GR4 ← 0
LD GR3, X	GR3 ← x
CPA GR3, GR4	比較
JPL L1	x > 0なら L1へ飛ぶ
LD GR4, ZERO	GR4 ← 0
LD GR3, X	GR3 ← x
SUBA GR4, GR3	GR4 - GR3 → GR4
ST GR4, X	X ← GR4

L1: 次の命令



# 全体のプログラムとしては



比較のためのレジスタ設定

```
LD GR4, ZERO    GR4 ← 0
LD GR3, X        GR3 ← x
```

```
CPA GR3, GR4    比較
JPL L1          x > 0なら
                L1へ飛ぶ
```

引き算のためのレジスタ設定

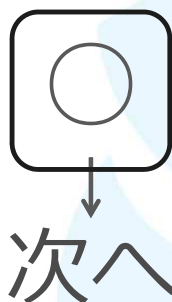
```
LD GR4, ZERO    GR4 ← 0
LD GR3, X        GR3 ← x
SUBA GR4, GR3    GR4 - GR3
                  → GR4
```

```
ST GR4, X        X ← GR4
```

L1: 次の命令



## 命令に変換するとどうなるかわかりましたか？



もう少しいろいろな例を見てみましょう

10

もう少しいろいろな例を見てみましょう

```
if (x ≥ 0) x = x + 1  
else x = x - 1
```

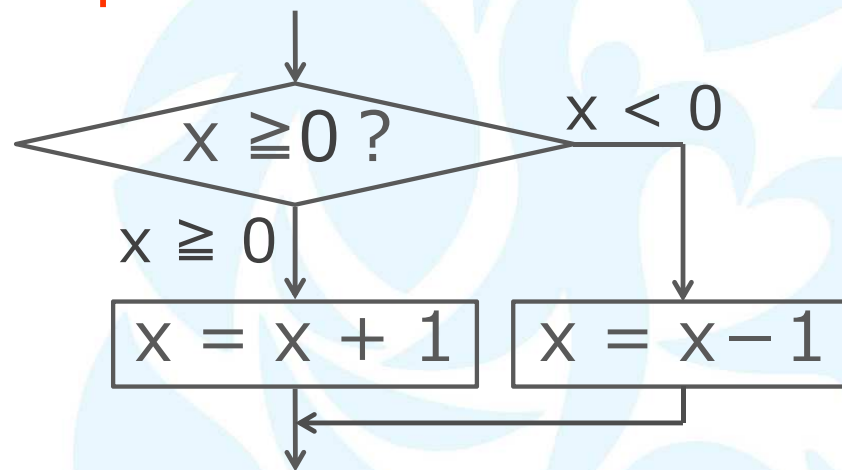
11



もう少しいろいろな例を見てみましょう

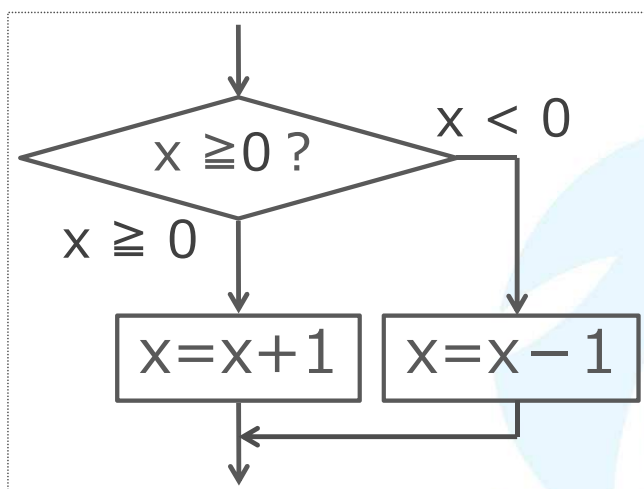
if ( $x \geq 0$ )  $x = x + 1$   
else  $x = x - 1$

フローチャート  
を描くと



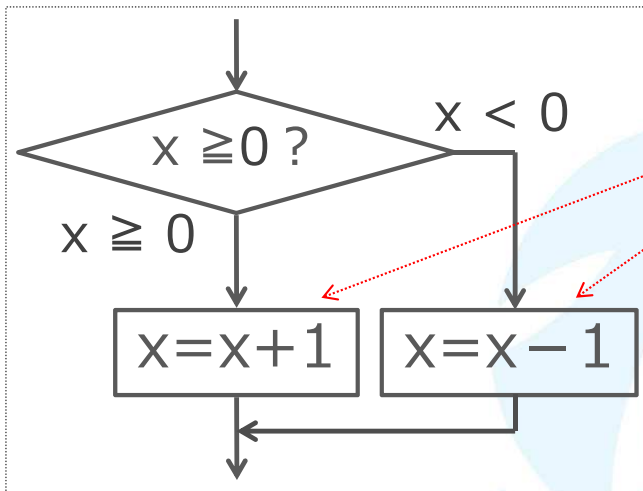
12

これを命令の流れに書き直すには



13

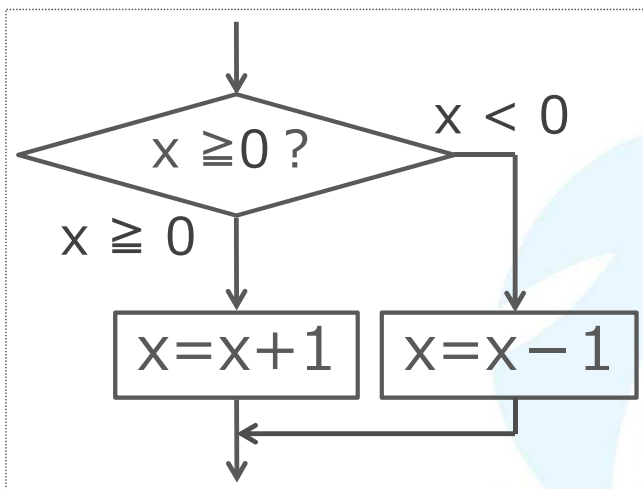
これを命令の流れに書き直すには



横に2列になる所がある



これを命令の流れに書き直すには



横に2列になる所がある



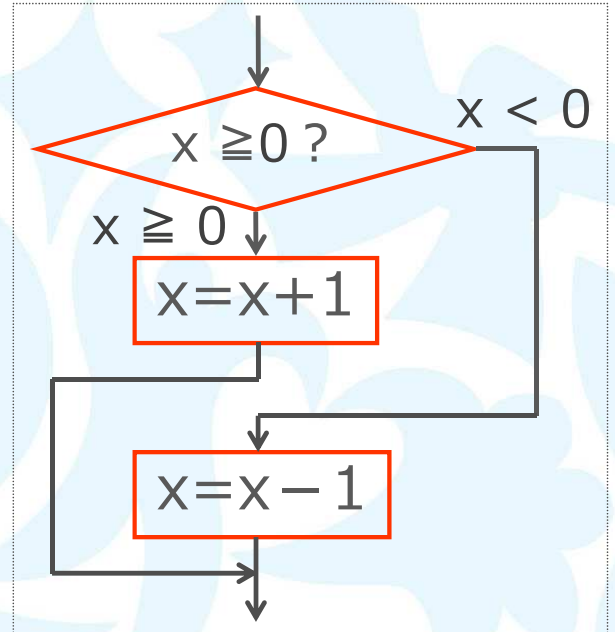
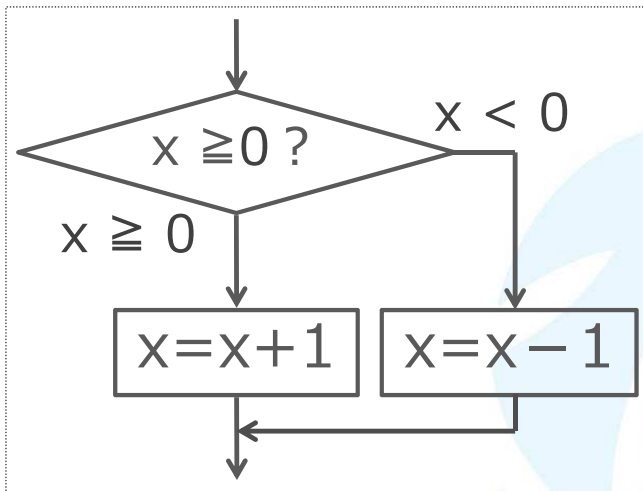
1列の流れに書き換える



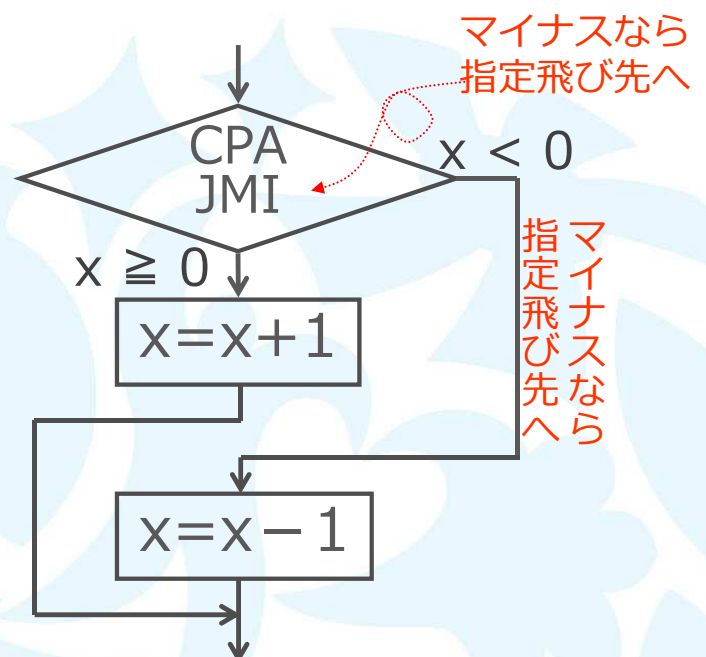
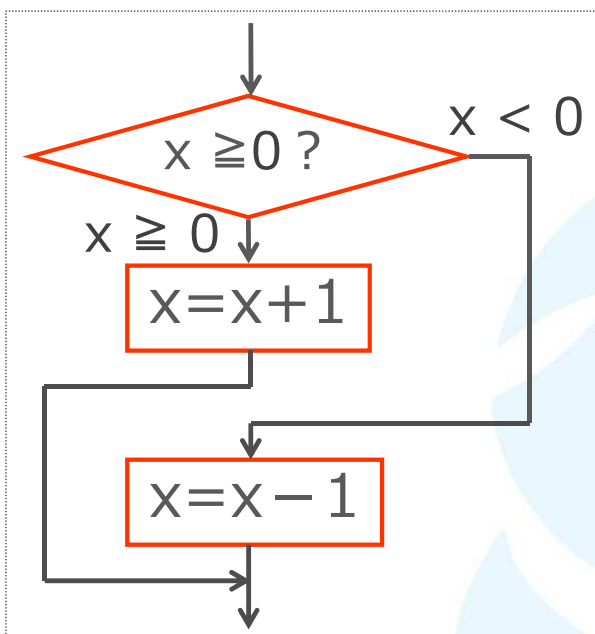


# これを命令の流れに書き直すには

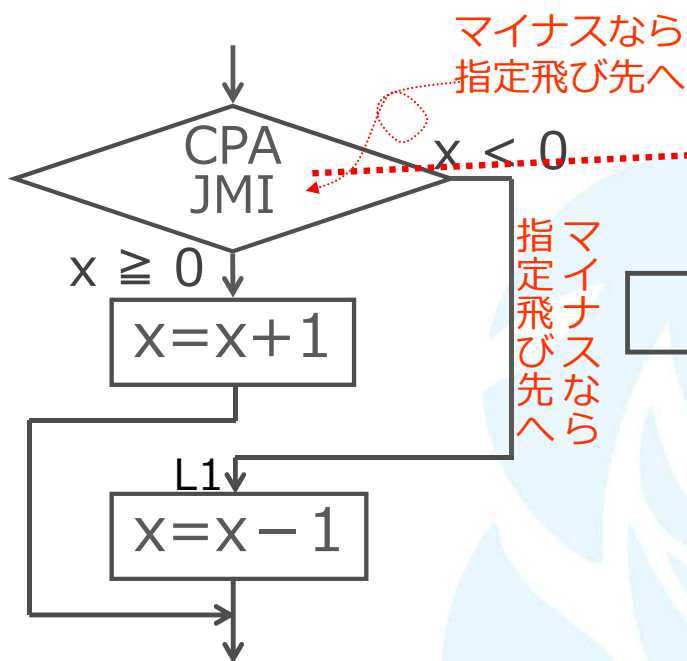
1列の流れ～横に並んでいない



# これを命令の流れに書き直すには



# これを命令の流れに書き直すには



```
LD  GR3, ZERO
LD  GR4, X
CPA GR3, GR4  (0-X)
JMI L1        負→L1
```

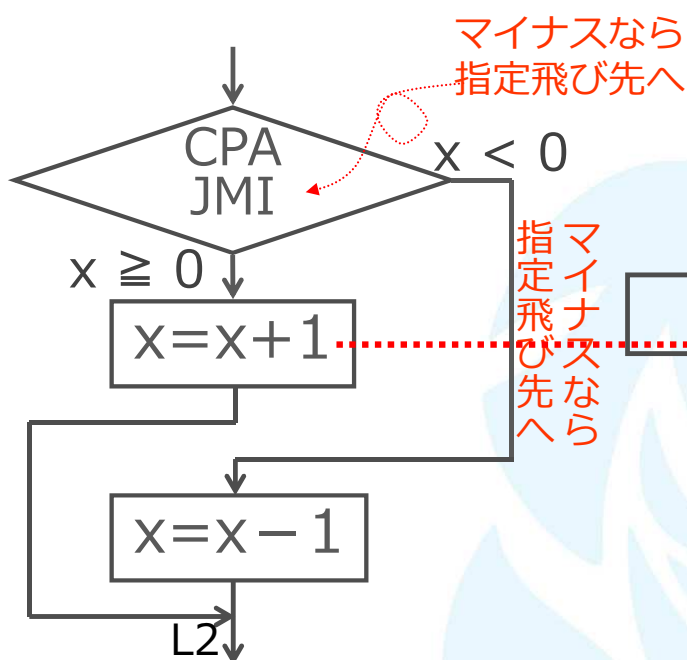
```
LD  GR3, ONE
ADDA GR4, GR3 (X+1)
ST  GR4, X
JUMP L2
```

```
L1 LD  GR3, ONE
SUBA GR4, GR3 (X-1)
ST  GR4, X
```

L2 次の命令



# これを命令の流れに書き直すには



```
LD  GR3, ZERO
LD  GR4, X
CPA GR3, GR4  (X-0)
JMI L1        負→L1
```

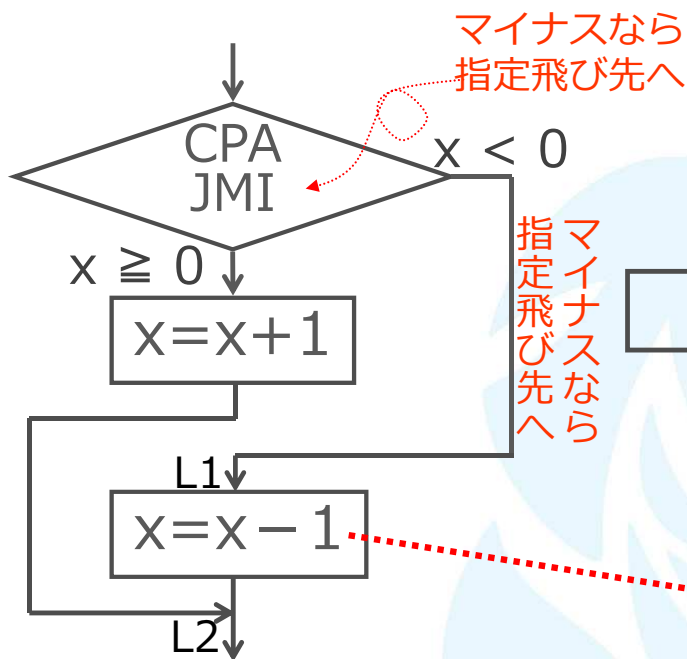
```
LD  GR3, ONE
ADDA GR3, GR4 (X+1)
ST  GR3, X
JUMP L2
```

```
L1 LD  GR3, ONE
SUBA GR4, GR3 (X-1)
ST  GR4, X
```

L2 次の命令



# これを命令の流れに書き直すには



```

LD   GR3, ZERO
LD   GR4, X
CPA  GR3, GR4   (X-0)
JMI  L1         負→L1
    
```

```

LD   GR3, ONE
ADDA GR3, GR4   (X+1)
ST   GR3, X
JUMP L2
    
```

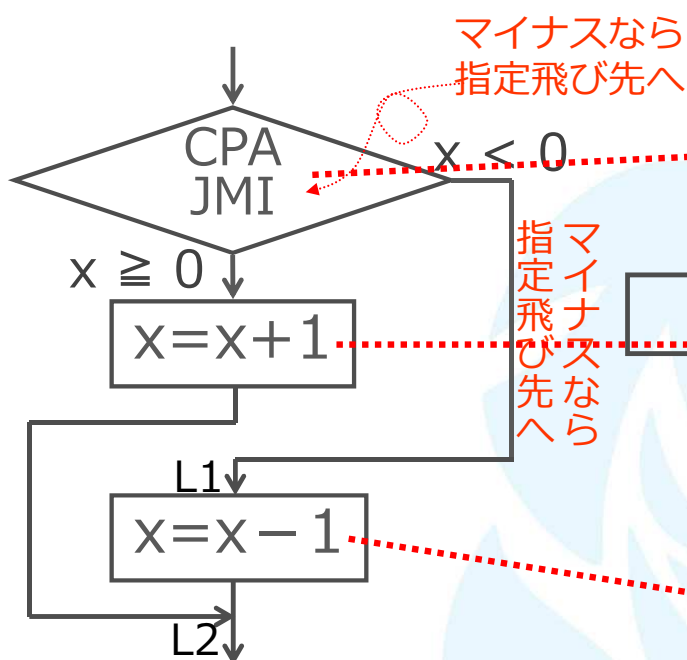
```

L1 LD   GR3, ONE
   SUBA GR4, GR3 (X-1)
   ST   GR4, X
    
```

L2 次の命令



# これを命令の流れに書き直すには



```

LD   GR3, ZERO
LD   GR4, X
CPA  GR3, GR4   (X-0)
JMI  L1         負→L1
    
```

```

LD   GR3, ONE
ADDA GR3, GR4   (X+1)
ST   GR3, X
JUMP L2
    
```

```

L1 LD   GR3, ONE
   SUBA GR4, GR3 (X-1)
   ST   GR4, X
    
```

L2 次の命令



<完成>

# 条件分岐をまとめると

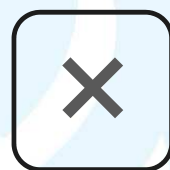
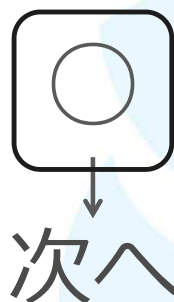
条件分岐は CPA+JPL (や CPA+JMI) で  
作ることが出来る

条件 (  $>$  とか  $\geq$  とか  $<$  とか  $\leq$  など ) は  
JPL, JMI, JZE, JNE を選んで作る

左右2つの経路があるときは、縦に並べて  
JUMP命令などでつなぐ必要がある

22

if 文 (条件分岐) から機械命令へ  
変換の仕方がわかりましたか？



23

では、演習問題で試してみましよう

24

## 演習問題

```
if ( x > 0 ) x = x + x  
else x = 0
```

25

# 演習問題

```
if ( x > 0 ) x = x + x  
else x = 0
```

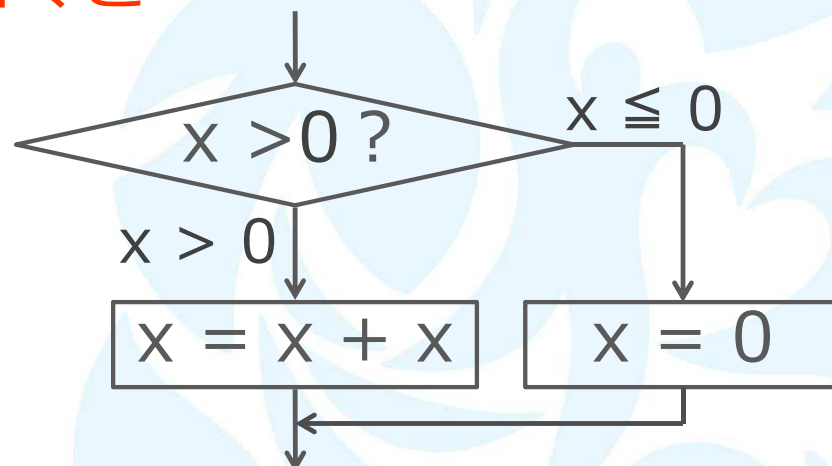
流れ図を描くと

26

# 演習問題

```
if ( x > 0 ) x = x + x  
else x = 0
```

流れ図を描くと

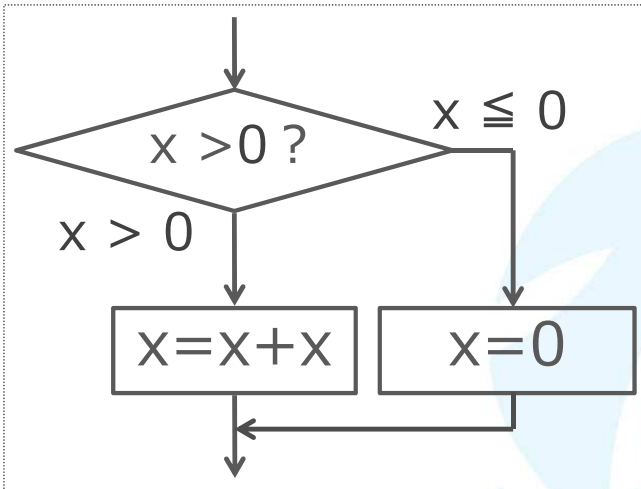


前の例題と同じようですね

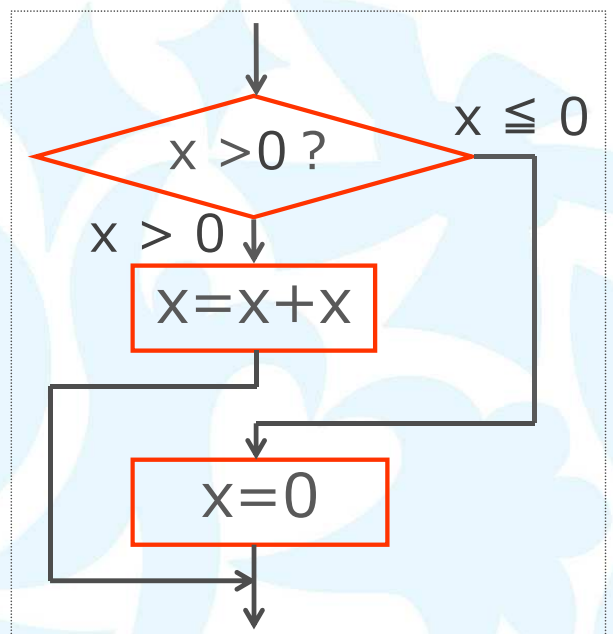
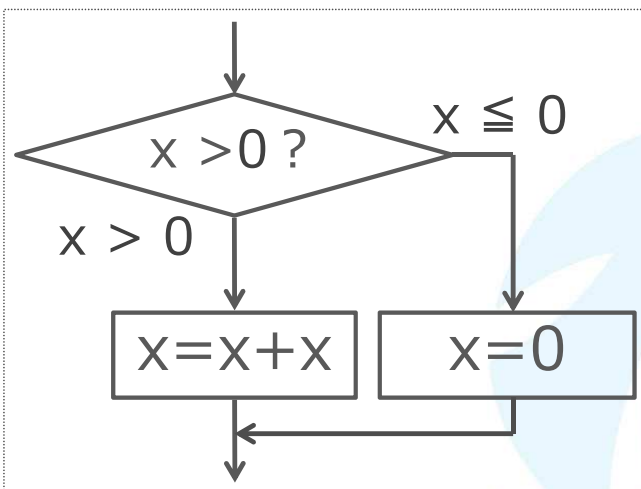
27



これを縦一列に書き直すと

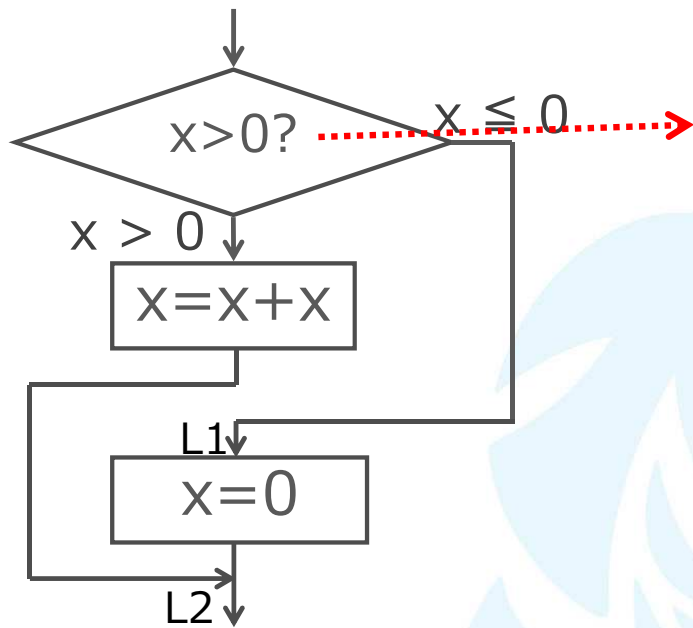


これを縦一列に書き直すと



ここまで前の例題と同じやり方です

# これを命令の流れに書き直すには



CPA GR4, GR3 (x-0)  
JMI L1 負→L1

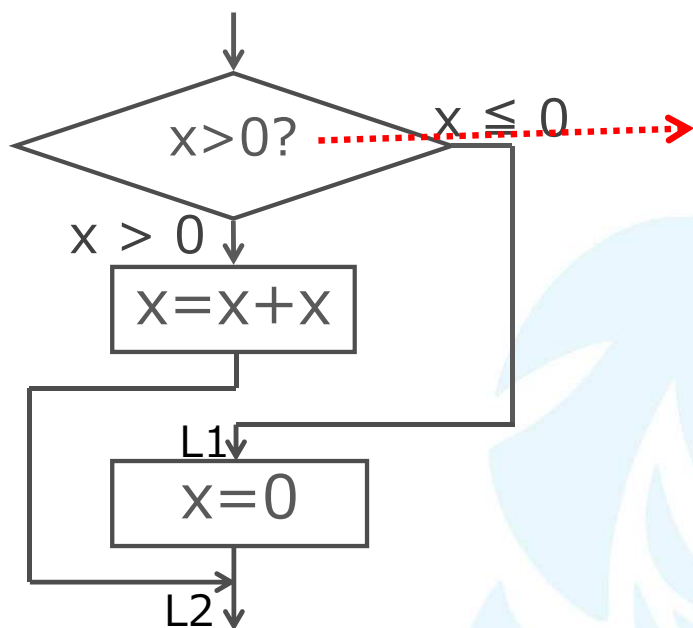
でいいだろうか？

x=0の時を考えてみよう

30



# これを命令の流れに書き直すには



CPA GR4, GR3 (x-0)  
JMI L1 負→L1

でいいだろうか？

x=0の時を考えてみよう

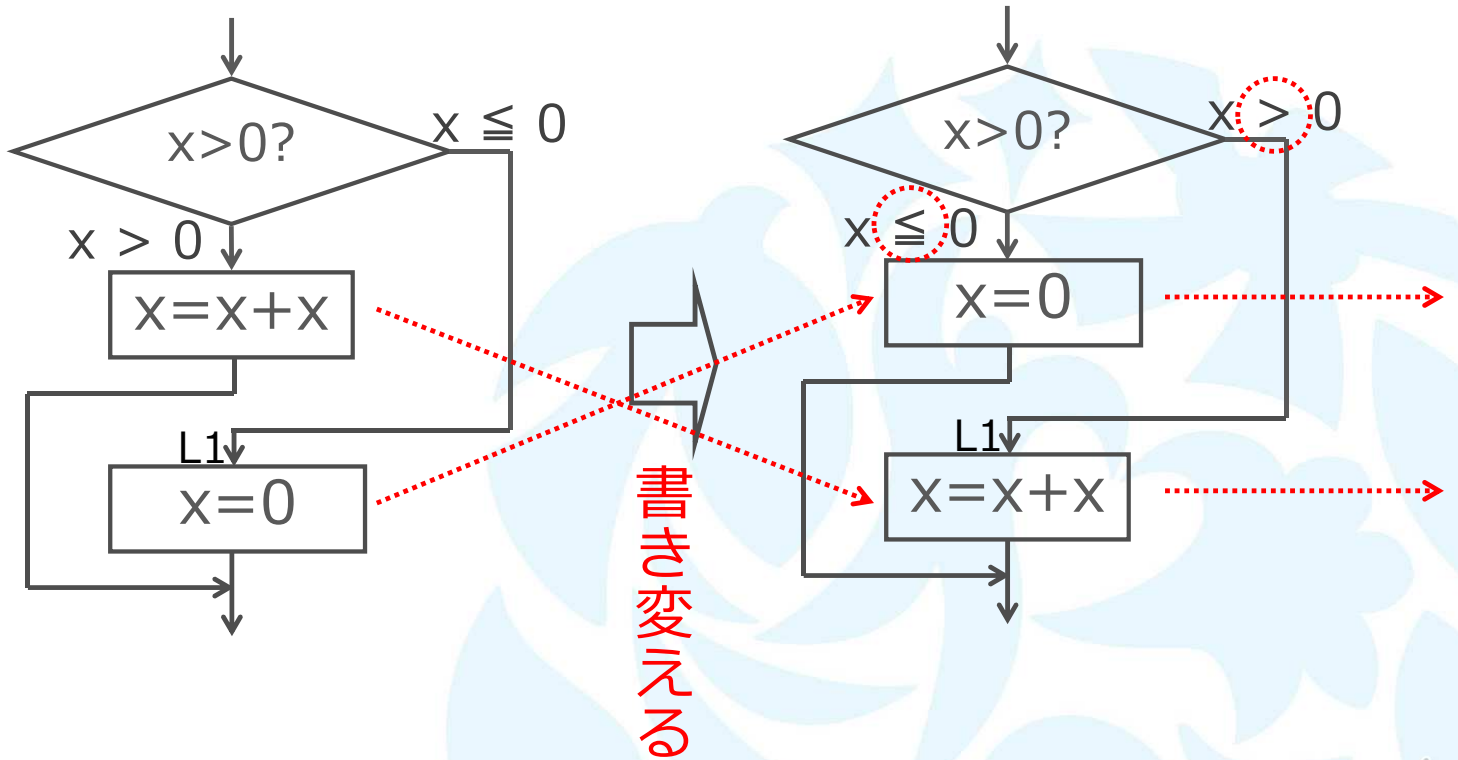
流れ図ではL1へ飛ぶ  
JMIだと0の時は  
ジャンプしない

流れ図と違う

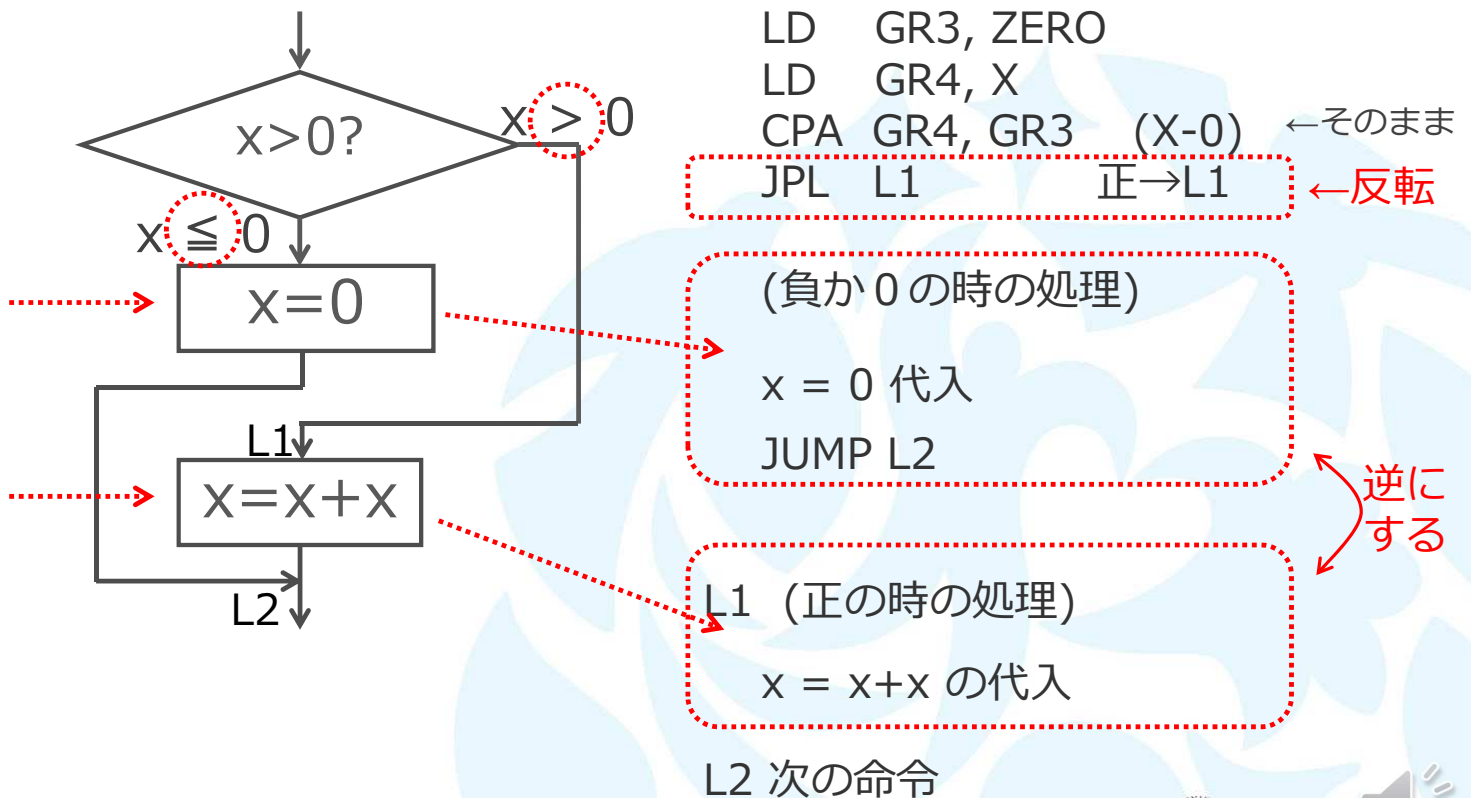
31



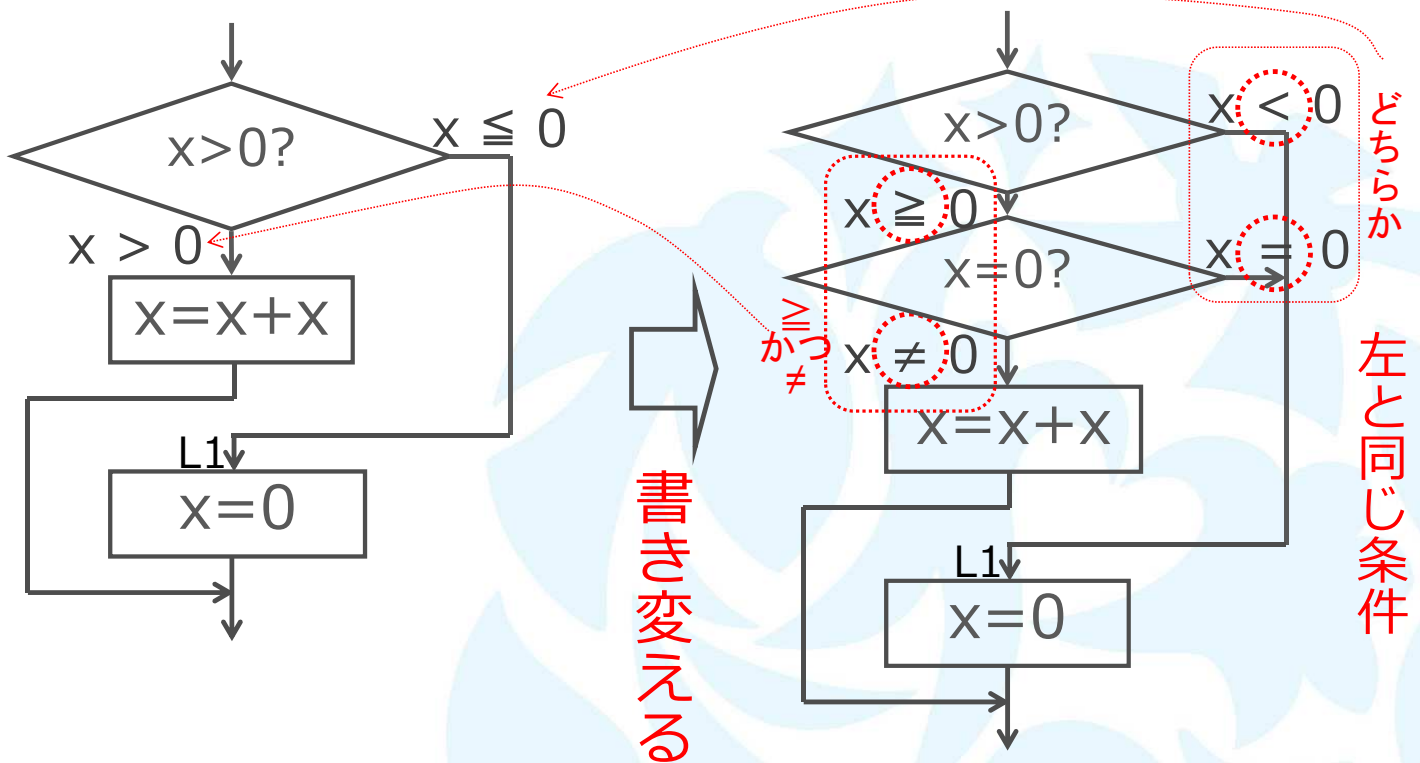
# 方法 1) 判定条件をひっくり返す



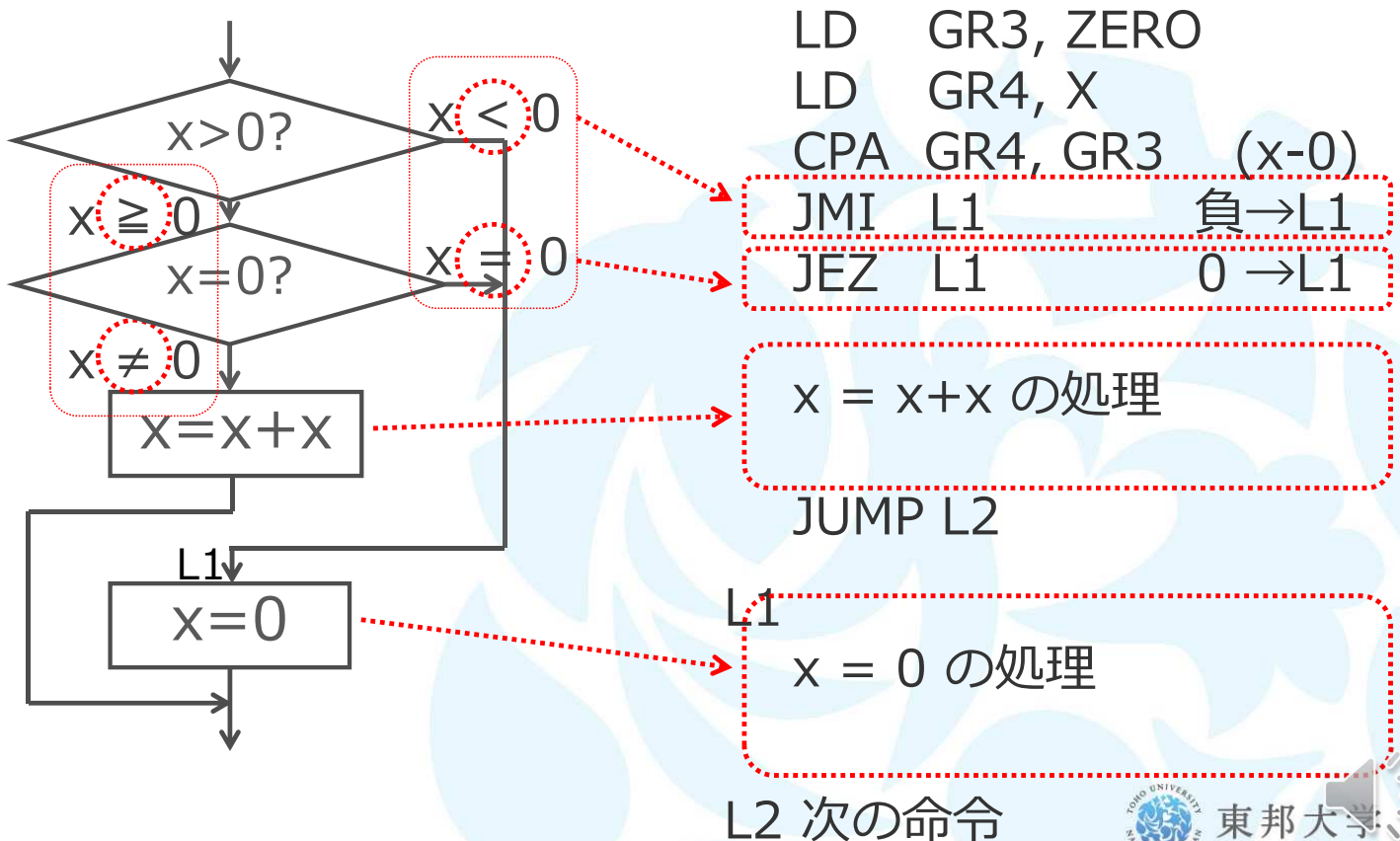
# 方法 1) 判定条件をひっくり返す



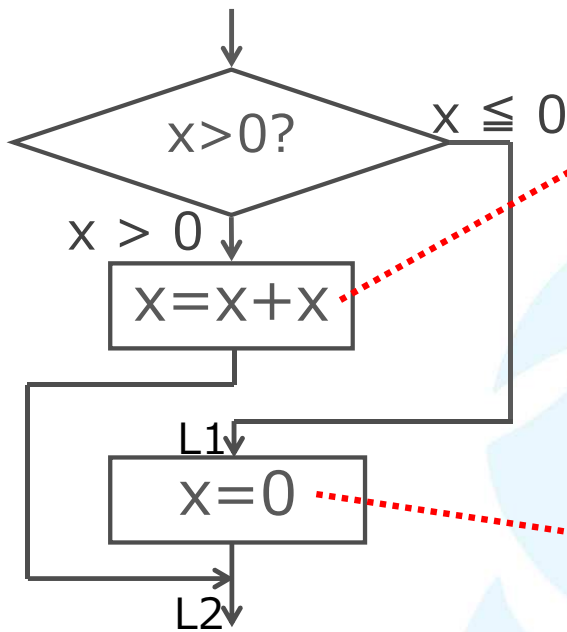
# 方法2) <と=の2回判定



# 方法2) <と=の2回判定



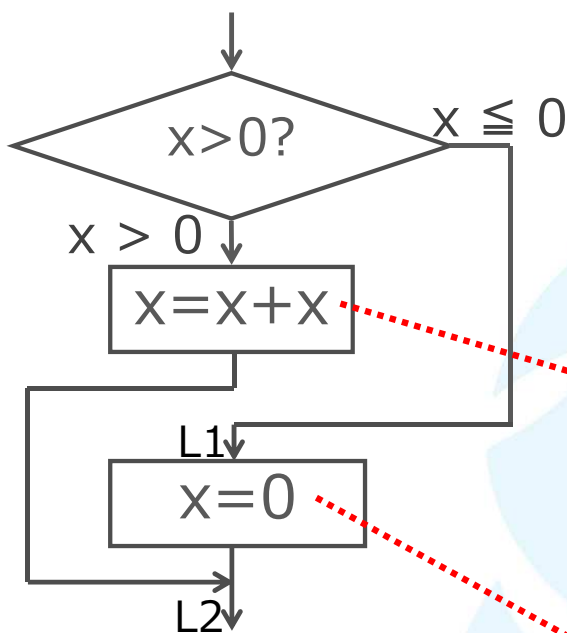
# 下の2つは既にできるはず



```
LD   GR3, X      GR3 ← X
ADDA GR3, GR3    GR3 ← GR3 + GR3
ST   GR3, X      X ← GR3
```

```
LD   GR3, ZERO  GR3 ← ZERO
ST   GR3, X      X ← GR3
```

# 全体を合わせると



```
LD   GR3, ZERO  2番目の方法
LD   GR4, X
CPA  GR4, GR3   (x-0)
JMI  L1         負 → L1
JEZ  L1         0 → L1
```

```
LD   GR3, X      GR3 ← X
ADDA GR3, GR3    GR3 ← GR3 + GR3
ST   GR3, X      X ← GR3
JUMP L2
```

```
L1 LD   GR3, ZERO  GR3 ← ZERO
ST   GR3, X      X ← GR3
```

L2 次の命令



# 追加した話は

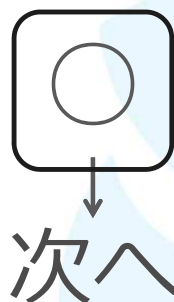
条件分岐での JPL や JMI の選び方は  
必ずしも一通りではない

条件を逆にし、飛び先も逆にすれば、  
同じような事ができる (=の時に注意)

2つ連続で JMI と JZE などを使っていい  
1つ目でジャンプしなかったときに  
2つ目の条件分岐 Jxx が実行される  
(注：CPAは1回だけでよい)

38

if 文（条件分岐）から機械命令へ  
変換の仕方がわかりましたか？



39