

# 論理回路学

## 8. 前半の復習 Part.2

佐藤証 西9-613

akashi.satoh@uec.ac.jp

<http://satoh.cs.uec.ac.jp/ja/lecture/CircuitDesign/index.html>

# 10進数⇔2進数変換

## 10進数→2進数

2) 77	余り	余り
2) 38	...	1
2) 19	...	0
2) 9	...	1
2) 4	...	1
2) 2	...	0
1	...	0

↑ 余りを下から並べる

$$77 = 1001101_{(2)}$$

## 2進数→10進数

1	0	0	1	1	0	1 <sub>(2)</sub>	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	$2^0 \times 1 = 1$
							$2^1 \times 0 = 0$
							$2^2 \times 1 = 4$
							$2^3 \times 1 = 8$
							$2^4 \times 0 = 0$
							$2^5 \times 0 = 0$
							$2^6 \times 0 = 0$
							<u>+ = 64</u>
							77

# 2進数の加減算

**27+77=104**

27=0011011<sub>(2)</sub>

77=1001101<sub>(2)</sub>

```

  0011011
+ 1001101
-----
 1101000
    
```

1101000<sub>(2)</sub>  
=64+32+8=104

**104-77=27**

104=01101000<sub>(2)</sub>

-77=10110011<sub>(2)</sub>

```

  01101000
+ 10110011
-----
 00011011
    
```

00011011<sub>(2)</sub>  
=16+8+2+1=27

**27-77=-50**

27=00011011<sub>(2)</sub>

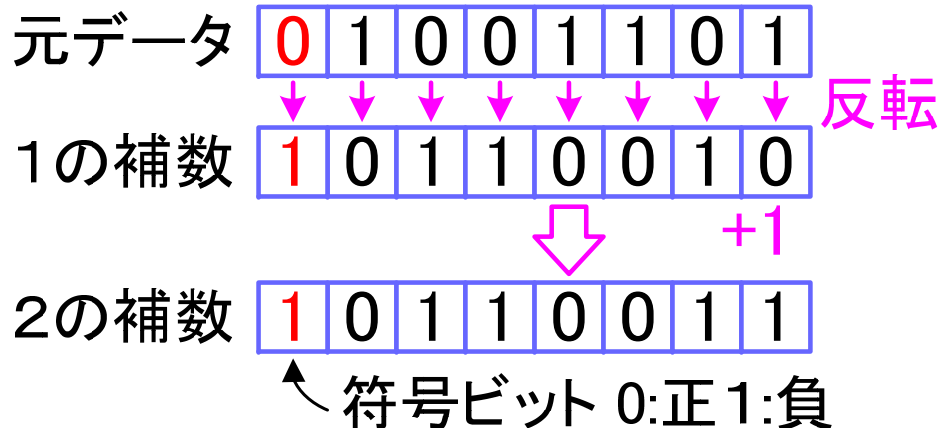
-77=10110011<sub>(2)</sub>

```

  00011011
+ 10110011
-----
 11001110
    
```

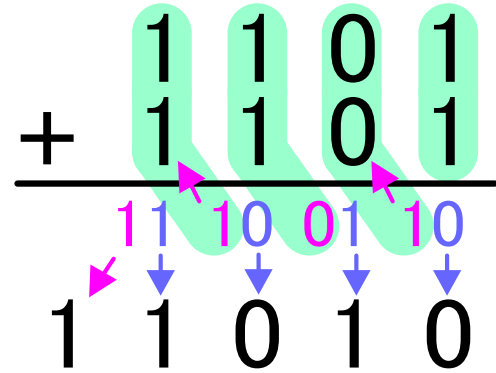
00110010<sub>(2)</sub> ← 2の補数  
=32+16+2=50

- 減算は2の補数を加える

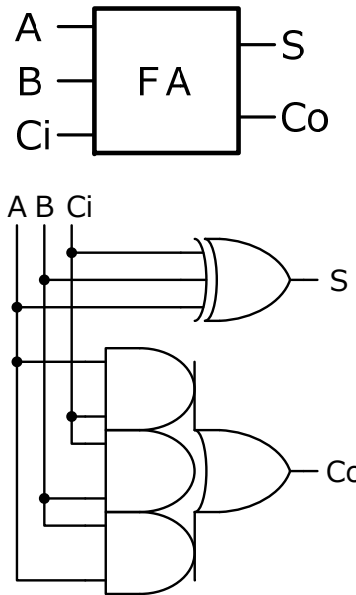


# 全加算器と半加算器

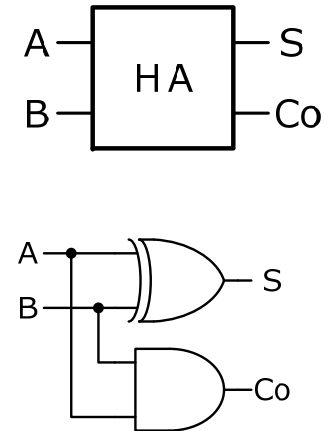
- 2進数の加算では各桁で、2bitの入力と1bitの桁上がりの計3bitの加算(全加算器)が必要
- 最下位は桁上りがないので2bitの加算(半加算器)でよい



A	B	Ci	Co	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

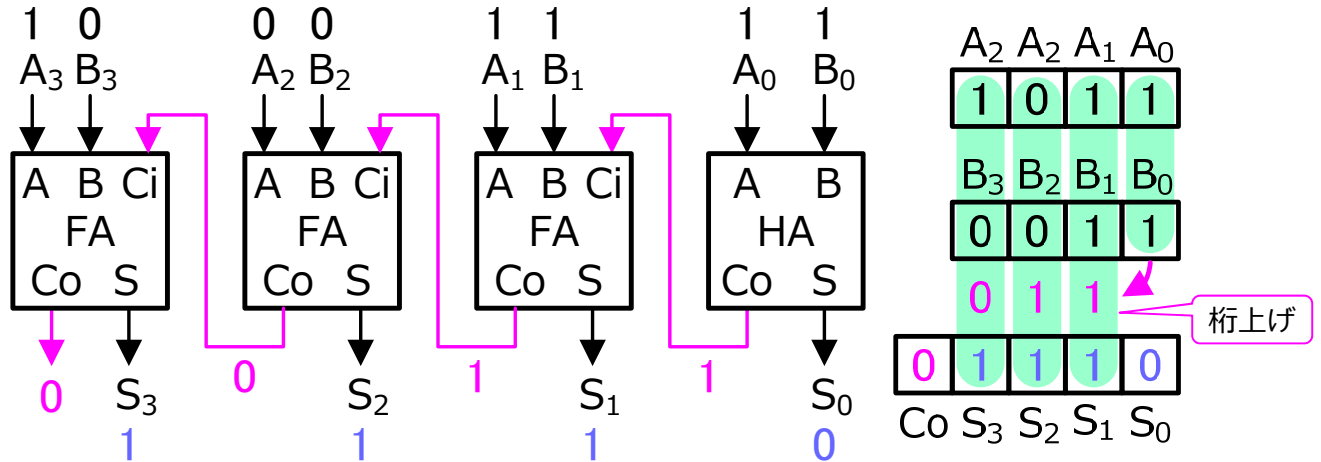


A	B	Co	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



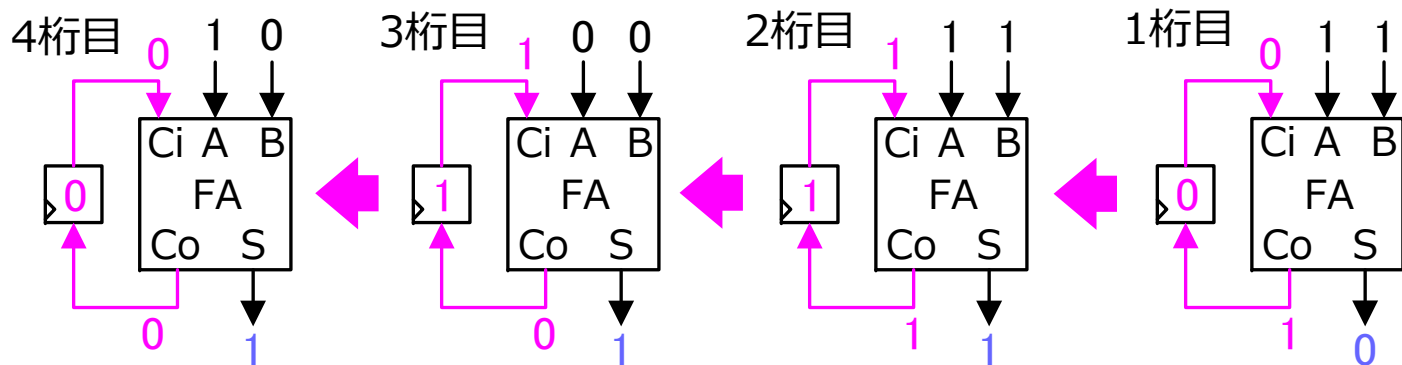
# 複数桁の加算器

## 並列加算方式



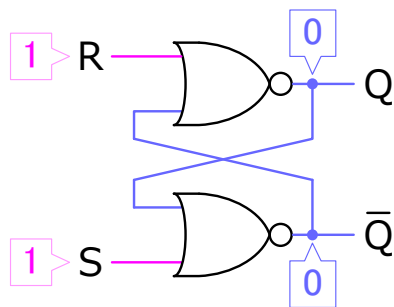
## 直列加算方式

$$1011 + 0011 = 1110$$



# RSフリップフロップ

- $R=S=1$ とすると出力は $Q=\bar{Q}=0$ となる
- 次に $R=S=0$ とすると通常は状態の保持だが、 $S$ が早く0になるか $R$ が早く0になるかによって出力が異なってしまう

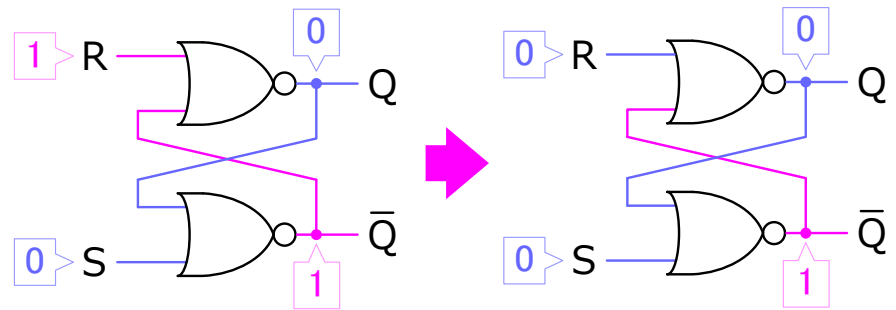


どっちにころぶ  
かわからない

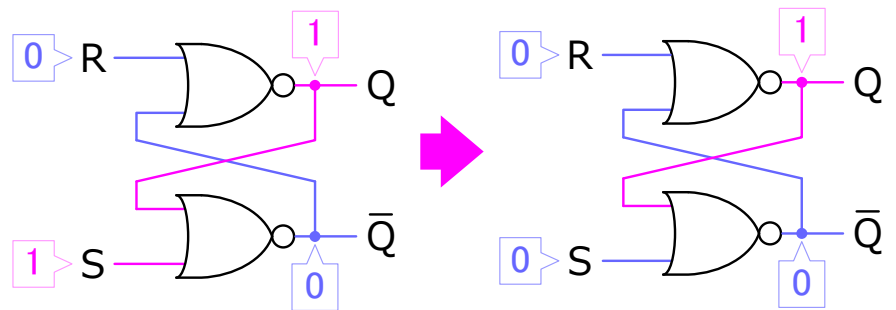
NOR

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

SがRよりも早く0に変化した場合

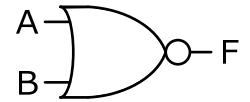


RがSよりも早く0に変化した場合



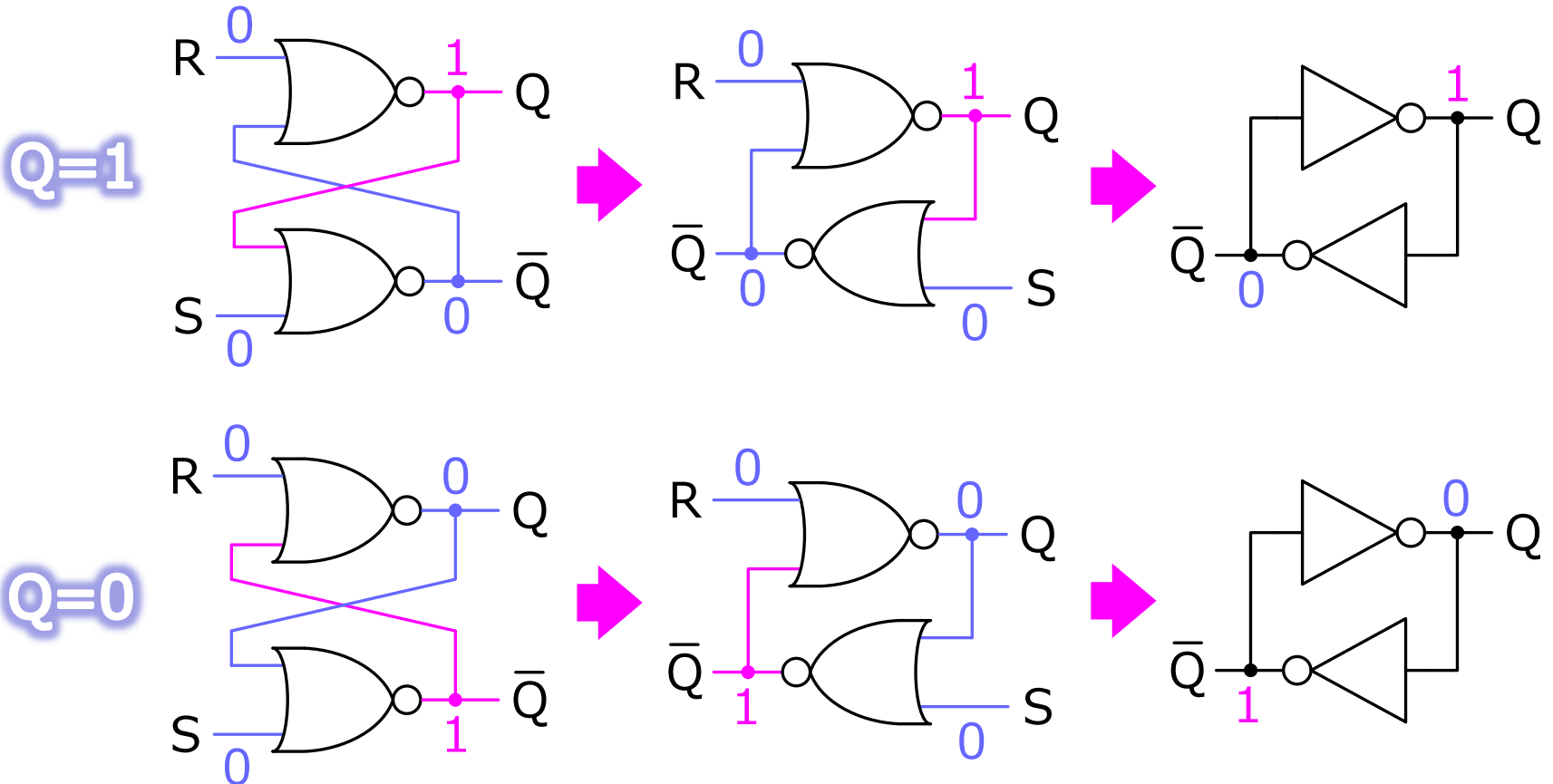
RS-FFで $S=R=1$ は禁止

# RSフリップフロップ



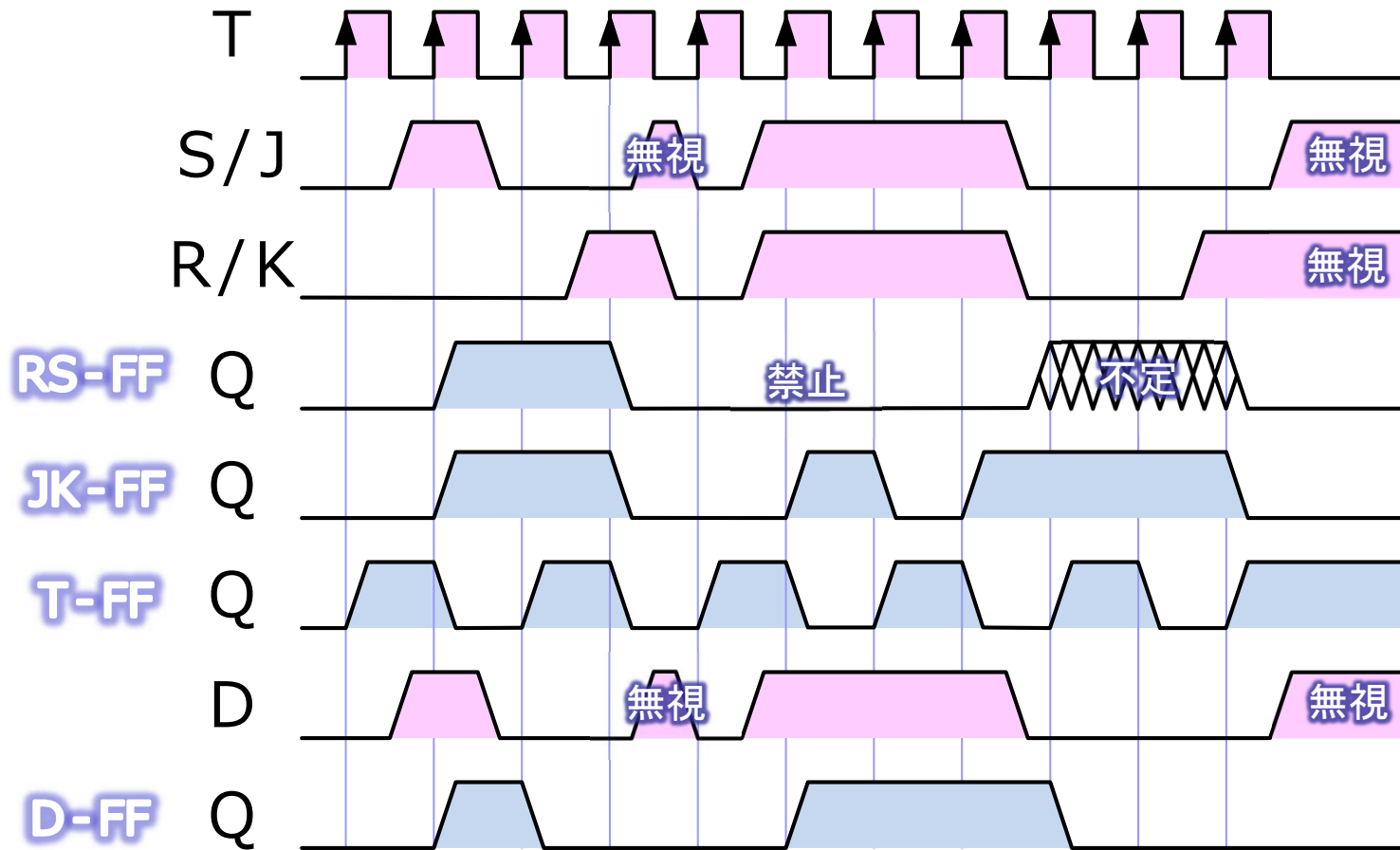
- R=S=0(保持)のときの回路の状態を調べる
- NOR入力的一方が0なので、他方の入力を反転するNOT
- データが 0 → 1 → 0 → 1 と回るループを作っている
- SRAMの1ビットの記憶回路はこのNOTのループ

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0



# クロック入力端子付きFF

- クロック(の立ち上がりまたは立下り)に同期して出力が変化
- クロックが入っていない時の変化は無視される
- 出力の変化には遅延があることに注意





# ゲート入出力遅延

- 入力信号の変化から出力変化までは遅延時間がある

