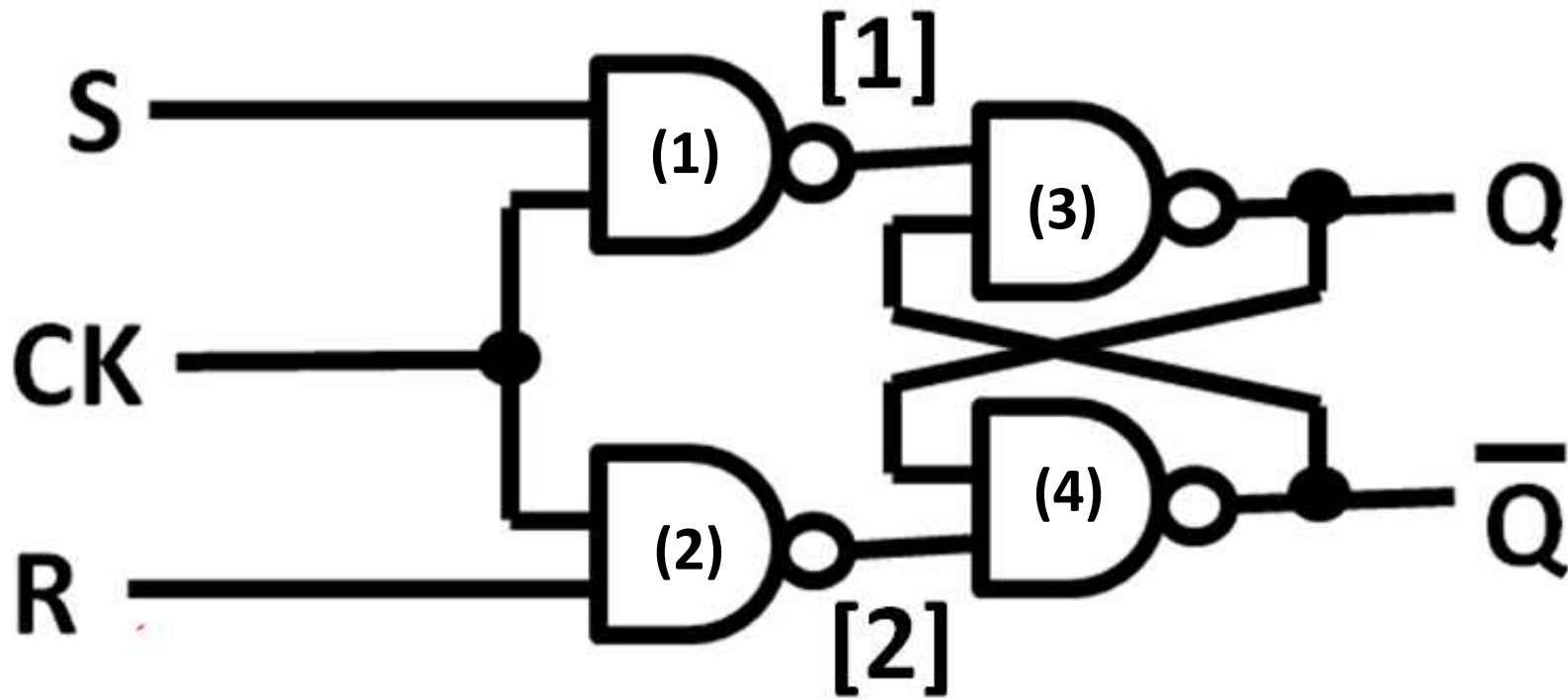
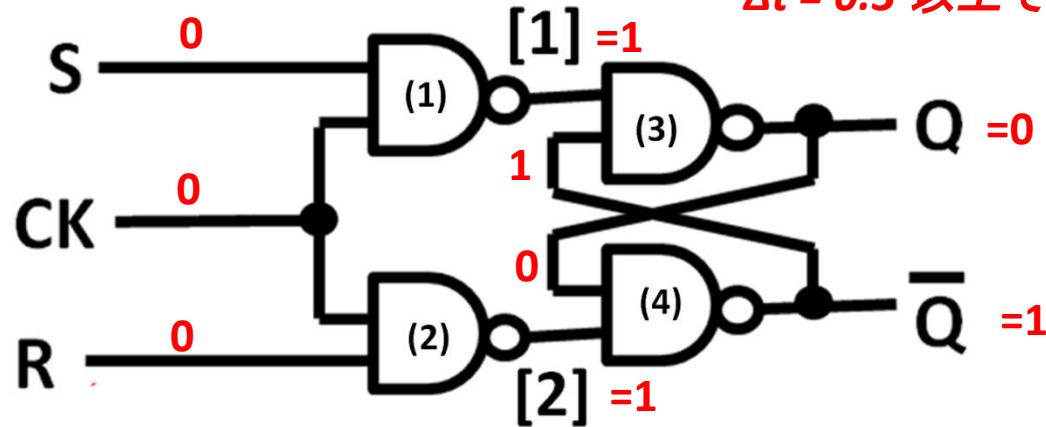


NAND型Flip-Flop回路RSFF()の動作説明

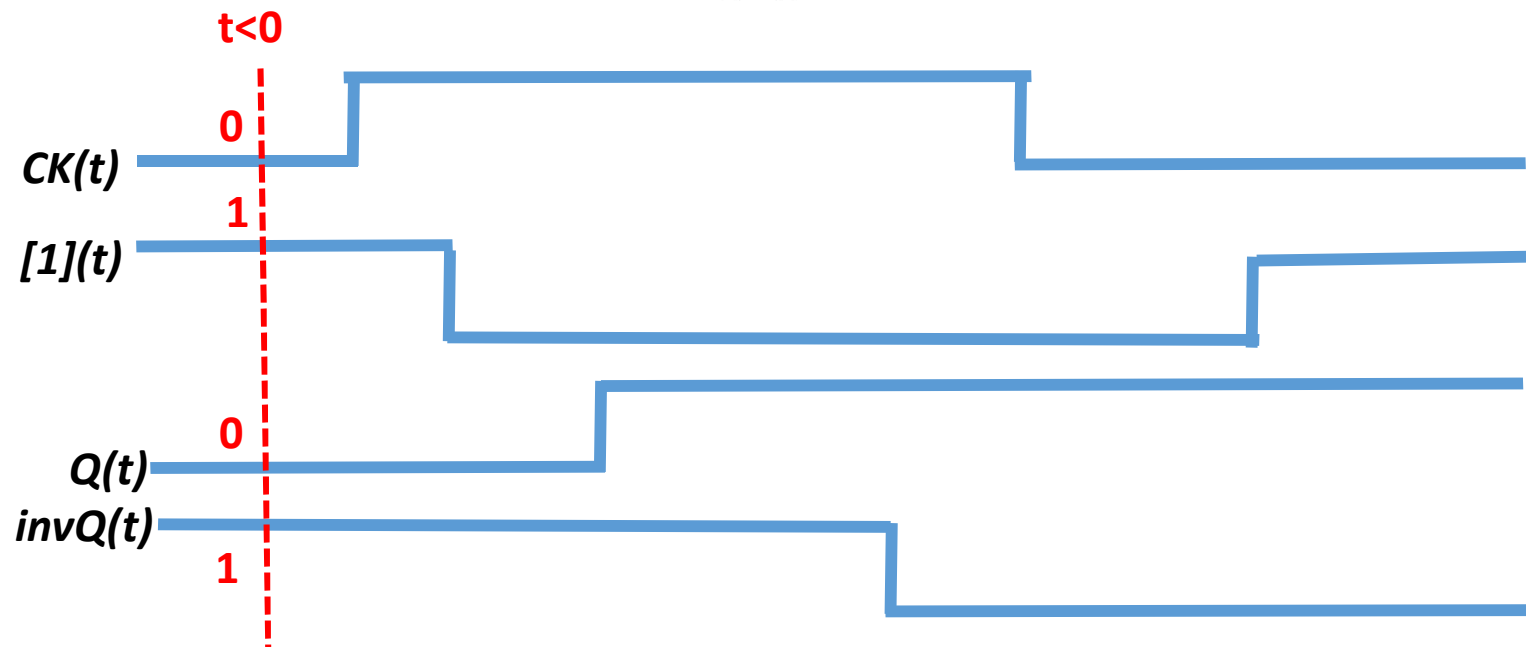


NAND型Flip-Flop回路RSFF()の動作説明

$\Delta t = 0.3$ 以上で RSFF() は動作する。

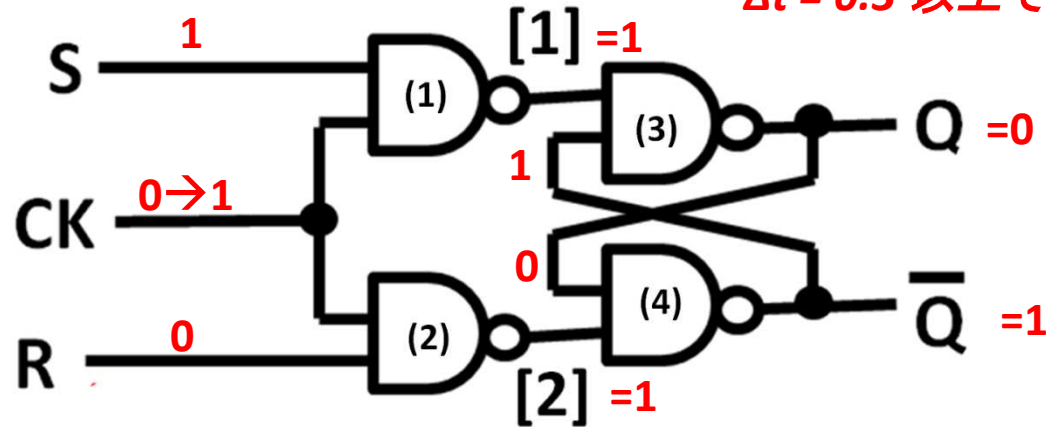


Clock Pulse の幅は少なくとも $\Delta t = 0.3$ 以上必要となる。



NAND型Flip-Flop回路RSFF()の動作説明

$\Delta t = 0.3$ 以上で RSFF() は動作する。

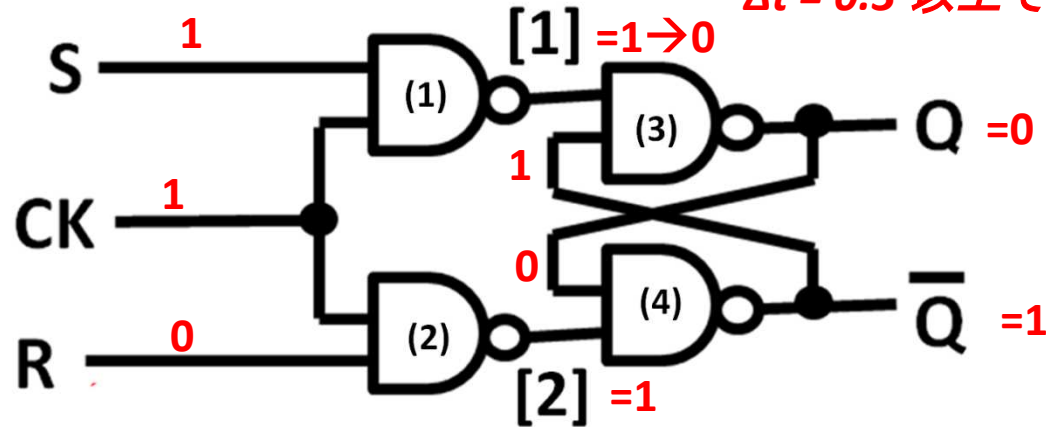


Clock Pulse の幅は少なくとも $\Delta t = 0.3$ 以上必要となる。

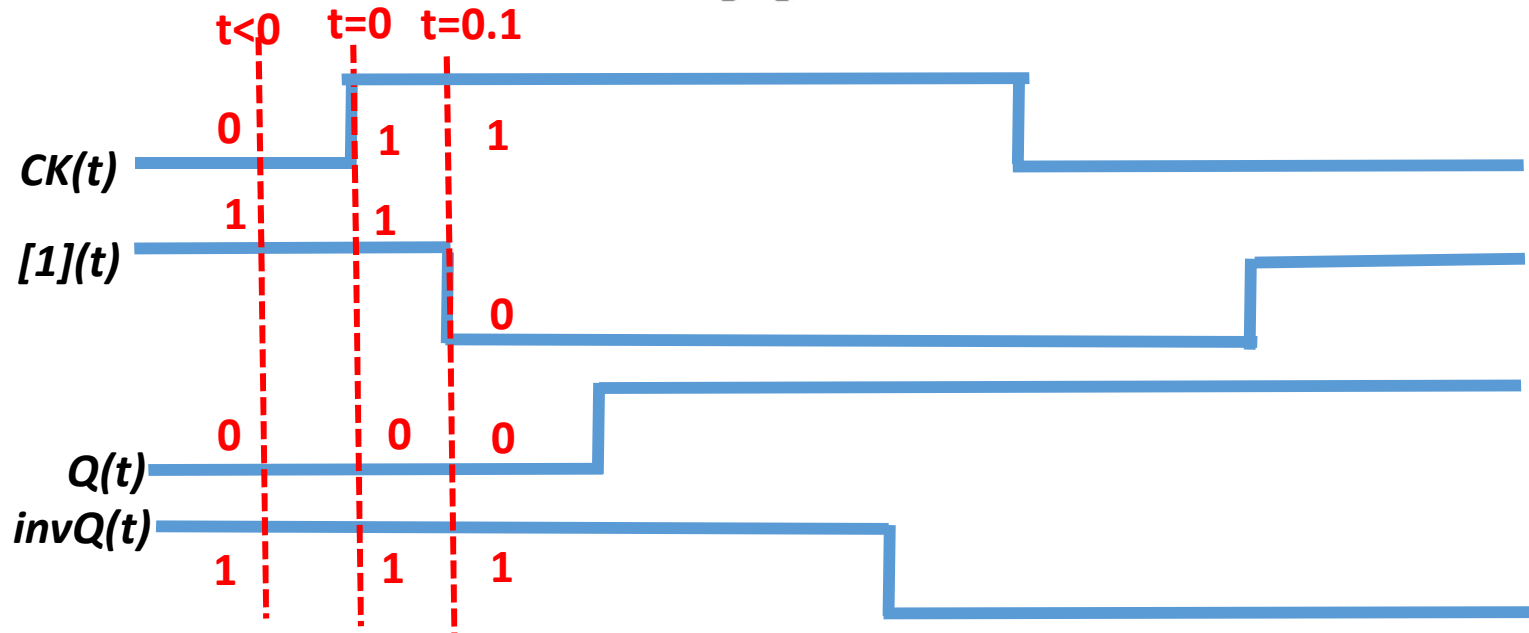


NAND型Flip-Flop回路RSFF()の動作説明

$\Delta t = 0.3$ 以上で RSFF() は動作する。

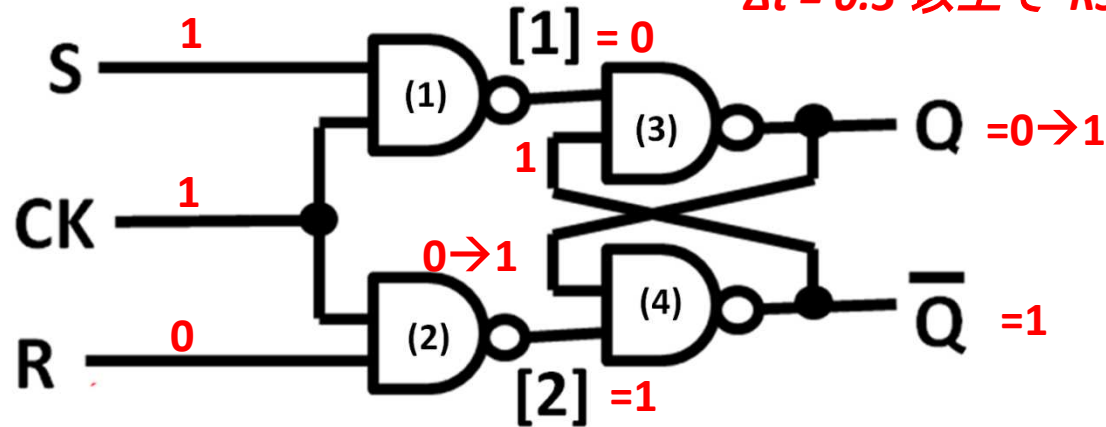


Clock Pulse の幅は少なくとも $\Delta t = 0.3$ 以上必要となる。

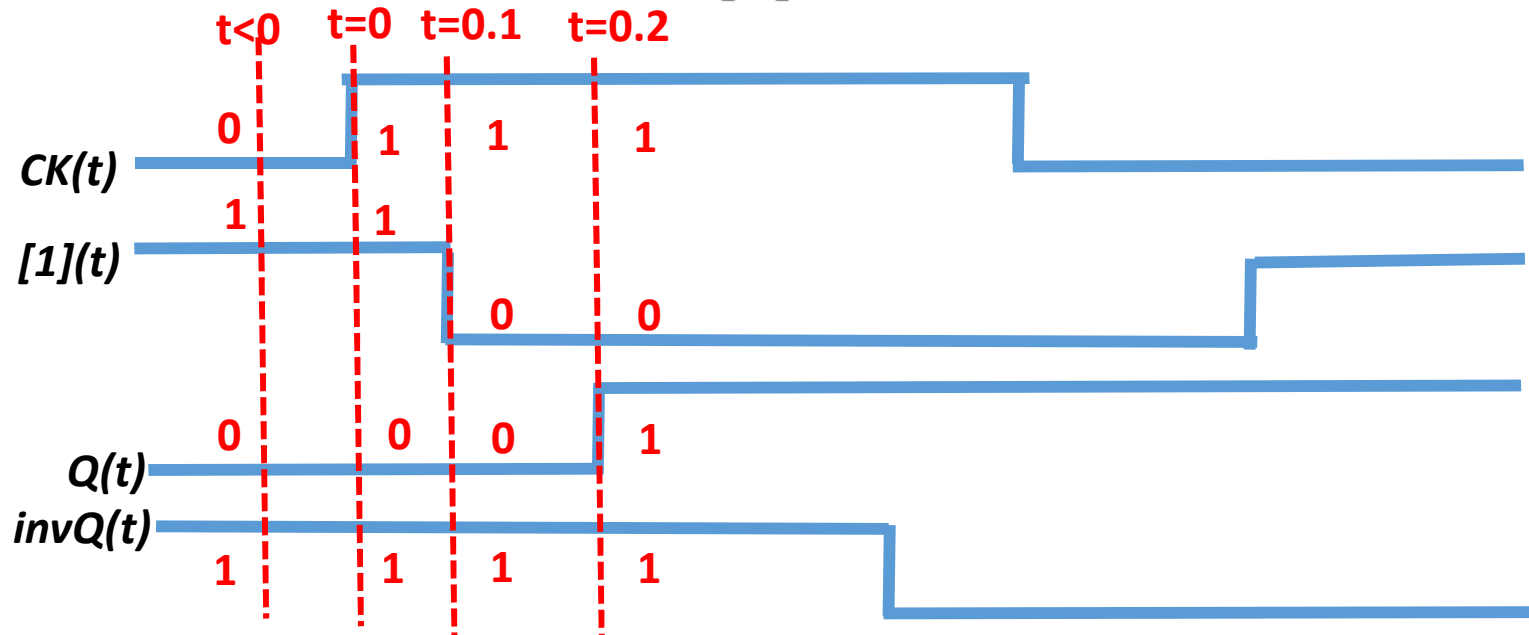


NAND型Flip-Flop回路RSFF()の動作説明

$\Delta t = 0.3$ 以上で RSFF() は動作する。

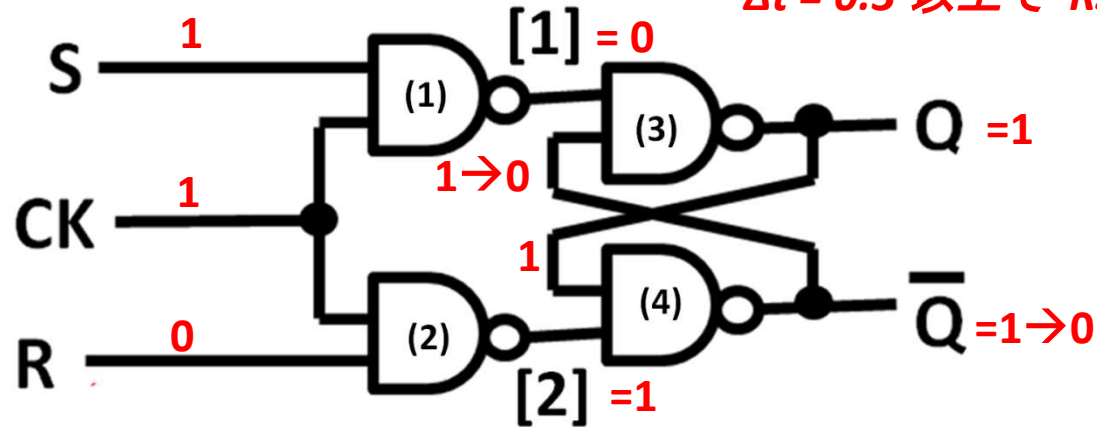


Clock Pulse の幅は少なくとも $\Delta t = 0.3$ 以上必要となる。

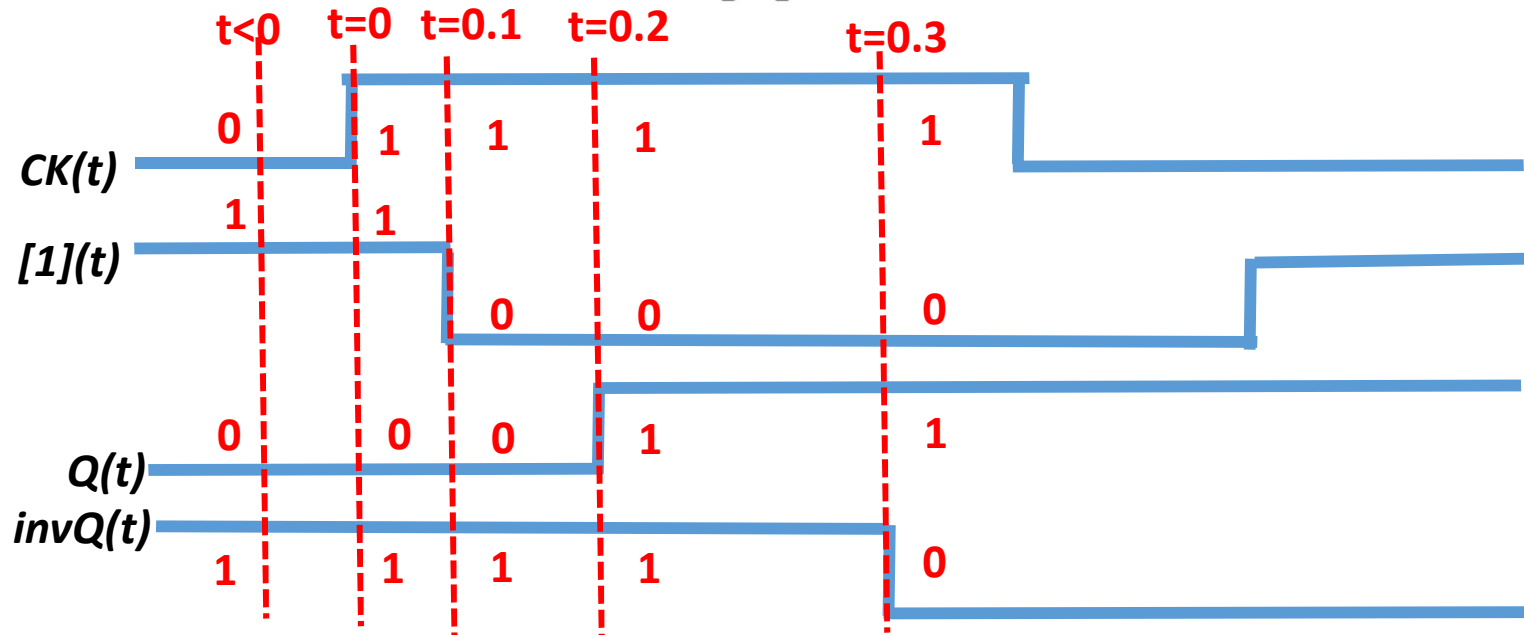


NAND型Flip-Flop回路RSFF()の動作説明

$\Delta t = 0.3$ 以上で RSFF() は動作する。

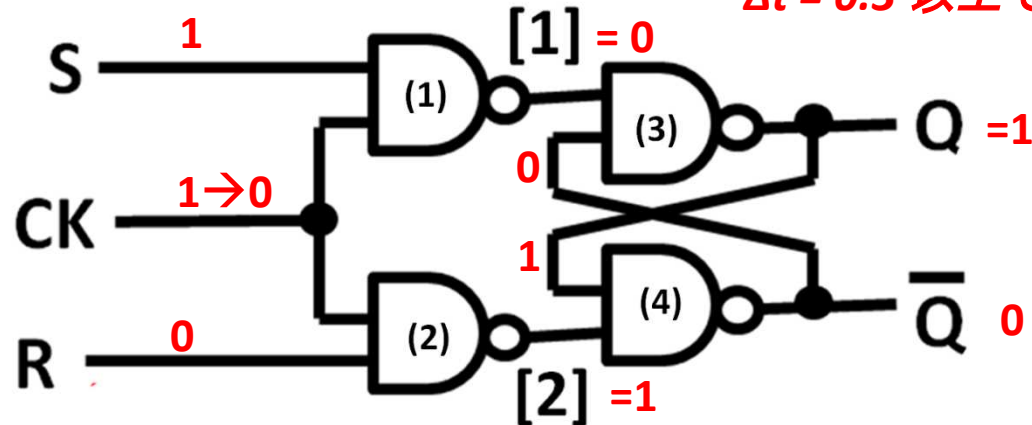


Clock Pulse の幅は少なくとも $\Delta t = 0.3$ 以上必要となる。

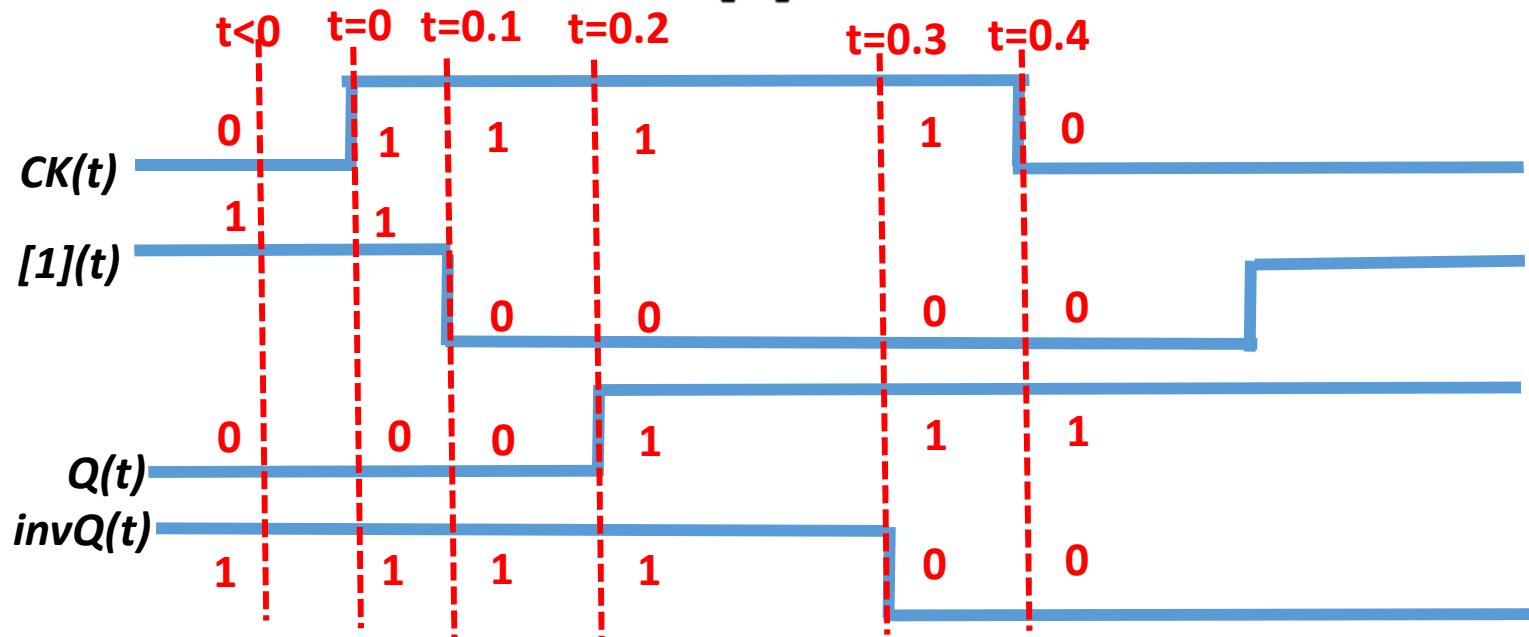


NAND型Flip-Flop回路RSFF()の動作説明

$\Delta t = 0.3$ 以上で RSFF() は動作する。

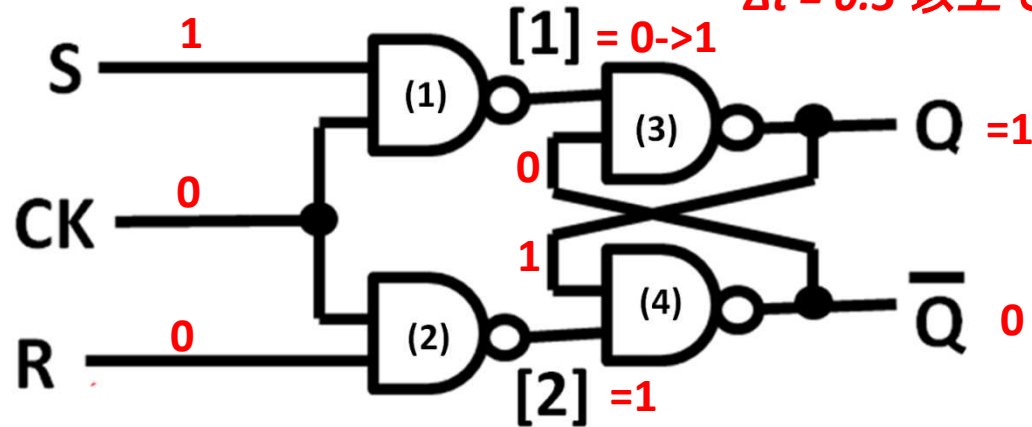


Clock Pulse の幅は少なくとも $\Delta t = 0.3$ 以上必要となる。

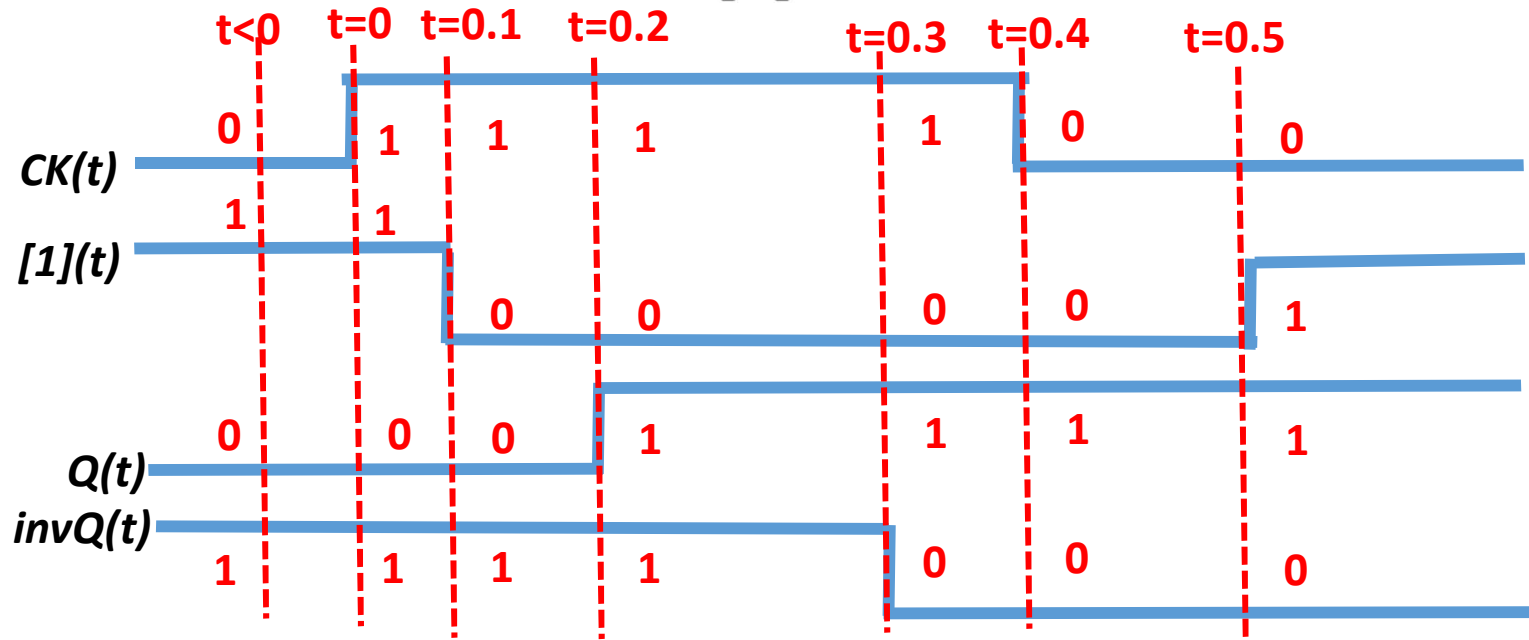


NAND型Flip-Flop回路 RSFF()の動作説明

$\Delta t = 0.3$ 以上で RSFF() は動作する。

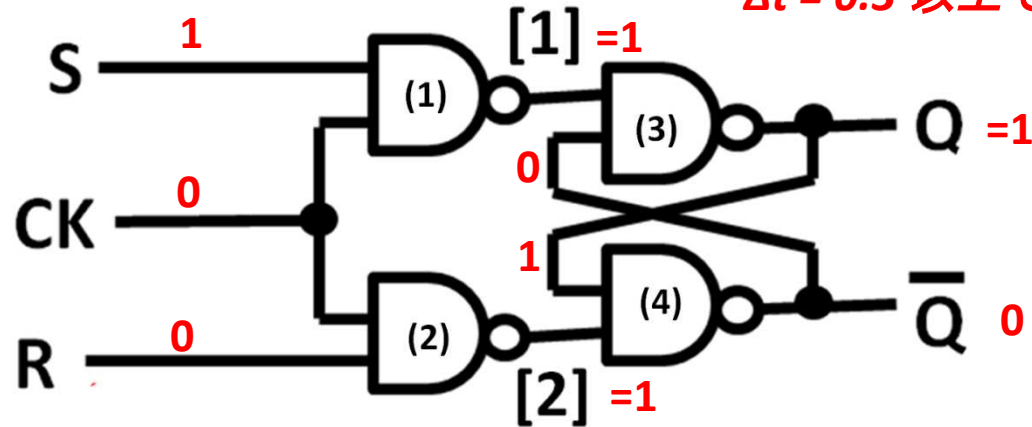


Clock Pulse の幅は少なくとも $\Delta t = 0.3$ 以上必要となる。

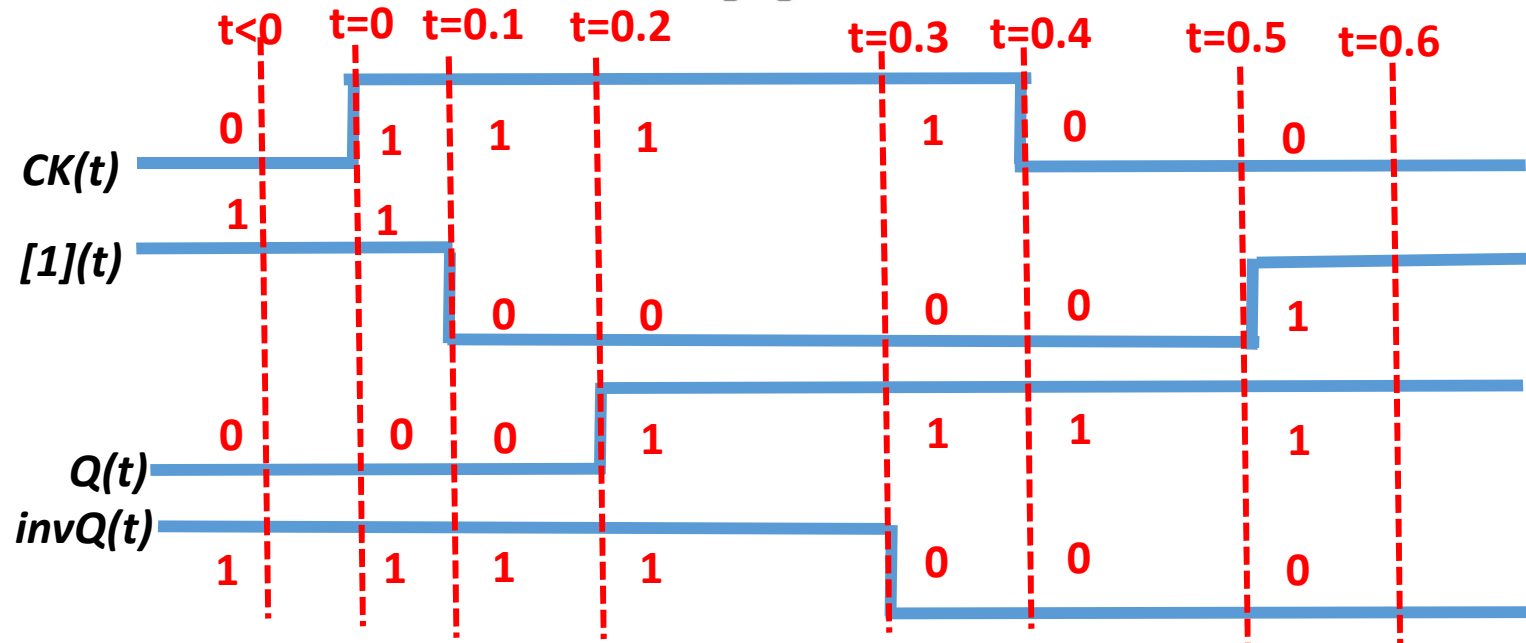


NAND型Flip-Flop回路RSFF()の動作説明

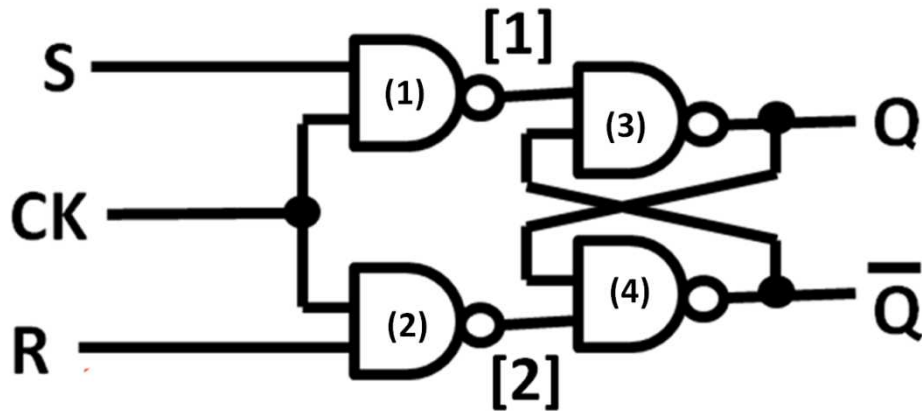
$\Delta t = 0.3$ 以上で RSFF() は動作する。



Clock Pulse の幅は少なくとも $\Delta t = 0.3$ 以上必要となる。



NAND型Flip-Flop回路 RSFF()の定義



禁止
入力 →

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

/****** DCDL Code *****/

```
define RSFF( ){ input CK,R,S ; output Q,invQ ;
```

```
memory QQ,invQQ ;
```

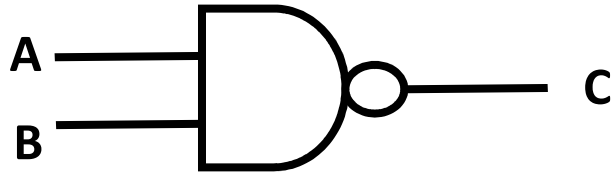
```
[S][CK]NAND(1)--->[1] ; [CK][R]NAND(2)--->[2] ;
```

```
[1][invQQ]NAND(3)--->[Q] ; [QQ][2]NAND(4)--->[invQ] ;
```

```
[Q]--->[QQ] ; [invQ]--->[invQQ] ; }
```

*****/

NAND回路 NAND()の定義



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

```
/****** DCDL Code *****/
```

```
define NAND( ){ input A,B ; output C ; memory CC;
```

```
    C=CC;  CC= 1 - A*B ; }
```

```
***** C-program Source Code *****/
```

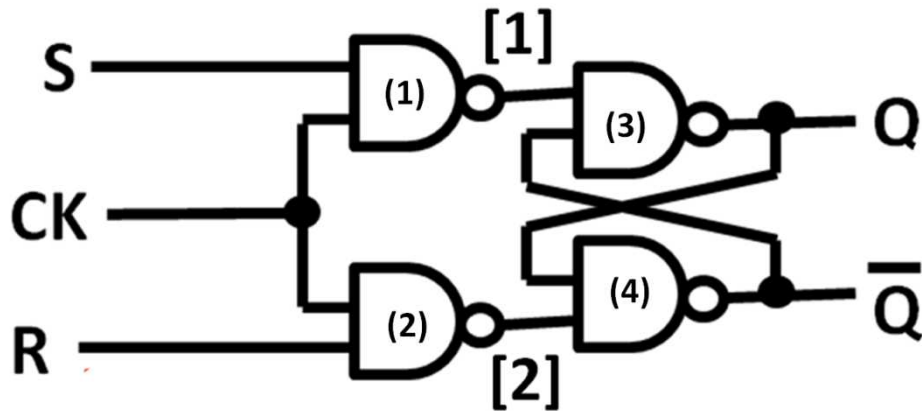
```
int NAND(void){
```

```
    NAND_C[iNAND]= NAND_CC[iNAND];
```

```
    NAND_CC[iNAND]=1-NAND_A[iNAND]*NAND_B[iNAND]; return 0; }
```

```
/****** */
```

NAND型Flip-Flop回路 RSFF()の定義



禁止
入力 →

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

```

/***** DCDL Code *****/

define RSFF( ){ input CK,R,S ; output Q,invQ ;

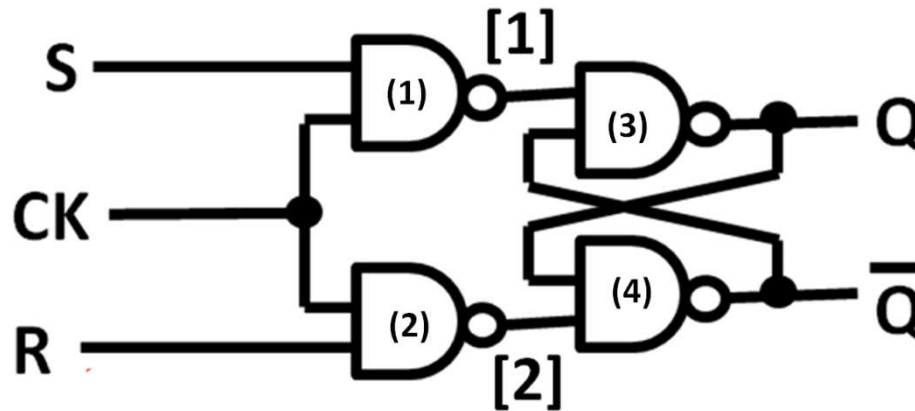
                memory QQ,invQQ ;

                [S][CK]NAND(1)--->[1] ; [CK][R]NAND(2)--->[2] ;
                [1][invQQ]NAND(3)--->[Q] ; [QQ][2]NAND(4)--->[invQ] ;

                [Q]--->[QQ] ; [ invQ]--->[invQQ] ; }

*****/
    
```

NAND型Flip-Flop回路 RSFF()の定義



***** DCDL Code *****

```
define RSFF(){ input CK,R,S ; output Q,invQ ;

    memory QQ,invQQ ;

    [S][CK]NAND(1)--->[1] ; [CK][R]NAND(2)--->[2] ;
    [1][invQQ]NAND(3)--->[Q] ; [QQ][2]NAND(4)--->[invQ] ;

    [Q]--->[QQ] ; [invQ]--->[invQQ] ; }
```

*****/

*****C-program Source Code *****/

```
int RSFF(void){

    iNAND=1; NAND_A[iNAND]= RSFF_S[iRSFF] ;
             NAND_B[iNAND]= RSFF_CK[iRSFF] ;
             NAND() ;
             RSFF_1[iRSFF]= NAND_C[iNAND] ;

    iNAND=2; NAND_A[iNAND]= RSFF_CK[iRSFF] ;
             NAND_B[iNAND]= RSFF_R[iRSFF] ;
             NAND() ;
             RSFF_2[iRSFF]= NAND_C[iNAND] ;

    iNAND=3; NAND_A[iNAND]= RSFF_1[iRSFF] ;
             NAND_B[iNAND]= RSFF_invQQ[iRSFF] ;
             NAND() ;
             RSFF_Q[iRSFF]= NAND_C[iNAND] ;

    iNAND=4; NAND_A[iNAND]= RSFF_Q[iRSFF] ;
             NAND_B[iNAND]= RSFF_2[iRSFF] ;
             NAND() ;
             RSFF_invQ[iRSFF]= NAND_C[iNAND] ;

    RSFF_Q[iRSFF] = RSFF_Q[iRSFF] ;

    RSFF_invQ[iRSFF]= RSFF_invQ[iRSFF];

    return 0; }
```

*****/

Clock CK(t) の定義

DCDL Code

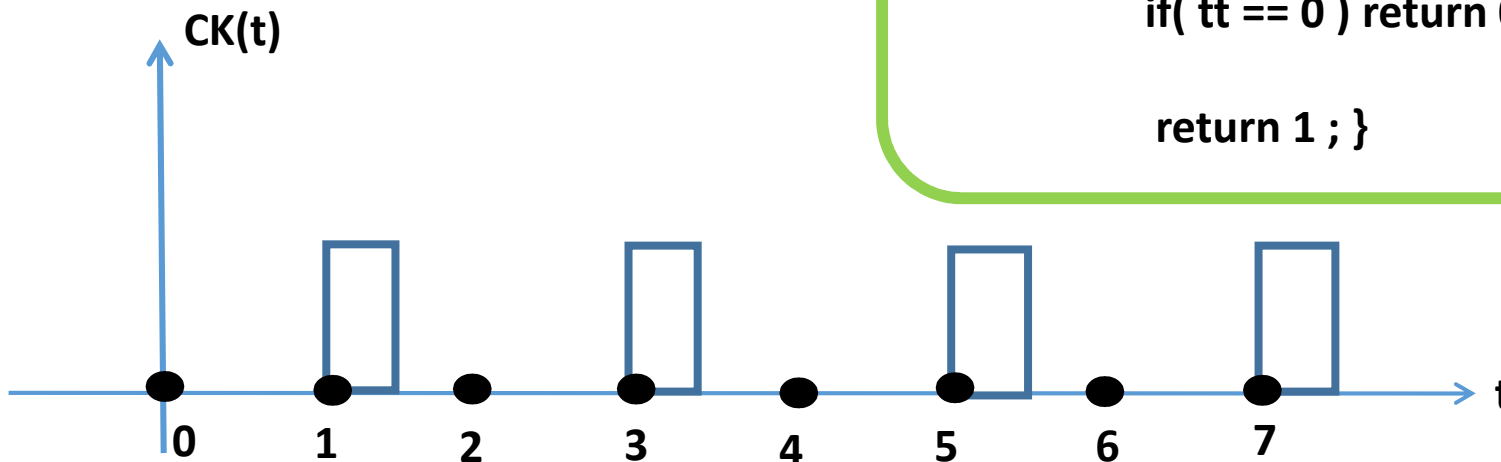
```
define CK(t) { input t ; output CK ;  
  
    CK = t - (t/2)*2 ; }
```

時間 t が整数で奇数の場合にのみ短いパルスがCK波形にでます。それ以外ではCK波形は常に off です。

入力変数 t は共通(*common*)変数として最初に宣言されているとします。

C-program Source Code

```
int CK( void ) { int tt ;  
  
    tt = t - 2*(t/2) ;  
  
    if( tt == 0 ) return 0 ;  
  
    return 1 ; }
```



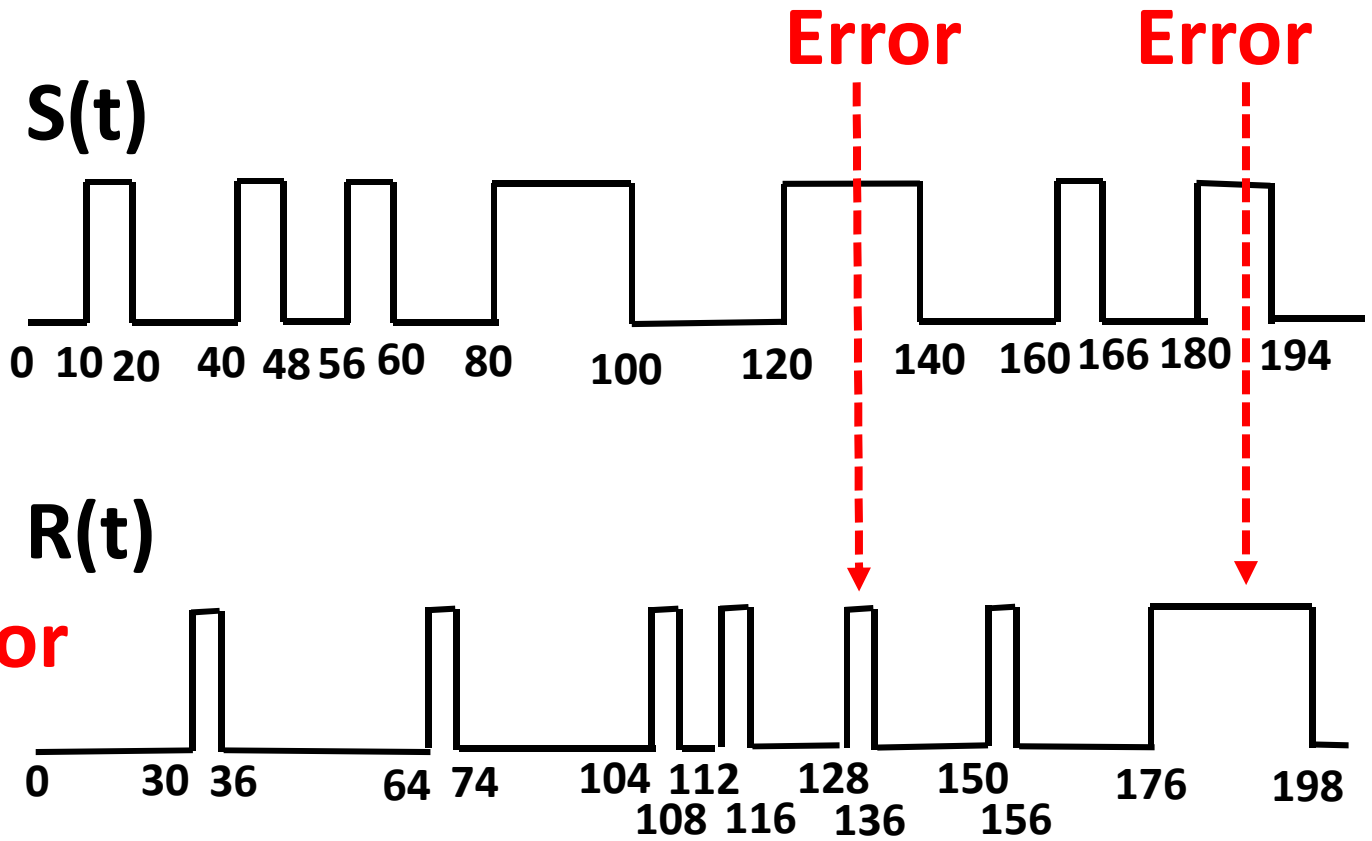
●実際のDCDLの計算では、CKパルス波形の幅は $\Delta t = 0.4$ に設定しています。

```

inRSFF_1 - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V)
*** input for RSFF_1 ***
t S R
10000 0 0
10010 1 0
10020 0 0
10030 0 1
10036 0 0
10040 1 0
10048 0 0
10056 1 0
10060 0 0
10064 0 1
10074 0 0
10080 1 0
10100 0 0
10104 0 1
10108 0 0
10112 0 1
10116 0 0
10120 1 0
10128 1 1
10136 1 0
10140 0 0
10150 0 1
10156 0 0
10160 1 0
10166 0 0
10176 0 1
10180 1 1
10194 0 1
10198 0 0
10200 0 0

```

入力信号 $S(t)$ と $R(t)$ の定義
 Input File Name = inRSFF_1.txt



Error

Error

●わざと 入力信号に Error を入れています。
 入力信号 $S(t)$ と $R(t)$ の値は必ず 時間 t が
 偶数值整数の時のみに変化することにします。

```

inRSFF_1 - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V)
*** input for RSFF_1 ***
  t   S   R
10000 0   0
10010 1   0
10020 0   0
10030 0   1
10036 0   0
10040 1   0
10048 0   0
10056 1   0
10060 0   0
10064 0   1
10074 0   0
10080 1   0
10100 0   0
10104 0   1
10108 0   0
10112 0   1
10116 0   0
10120 1   0
10128 1   1
-----Error
10136 1   0
10140 0   0
10150 0   1
10156 0   0
10160 1   0
10166 0   0
10176 0   1
10180 1   1
-----Error
10194 0   1
10198 0   0
10200 0   0

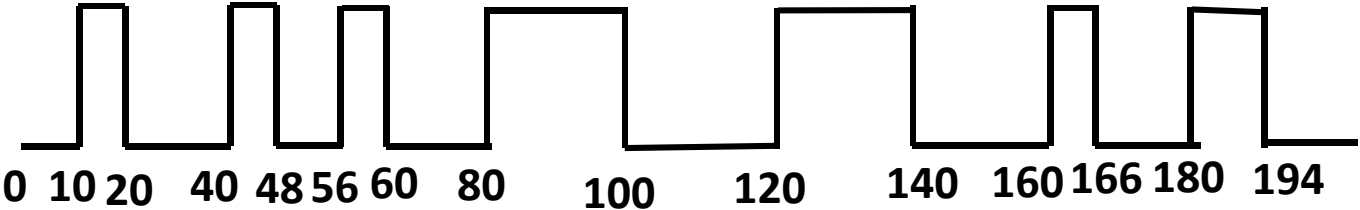
```

入力信号関数 $S(t)$ の定義
 Input File Name = inRSFF_1.txt

あらかじめ、 $SS[i]$ for $i=1$ to NT ; の値を求めます。

Sampling Number = $NT = 200$;

$S(t)$



```

int S( void ) { int i ;

for (i=1;i<NT+1;i++) {

    if( t>=TT[NT+1-i] ) {

        if( SS[NT+1-i]==1) return 1 ; return 0; } }

return 0; }

```



```

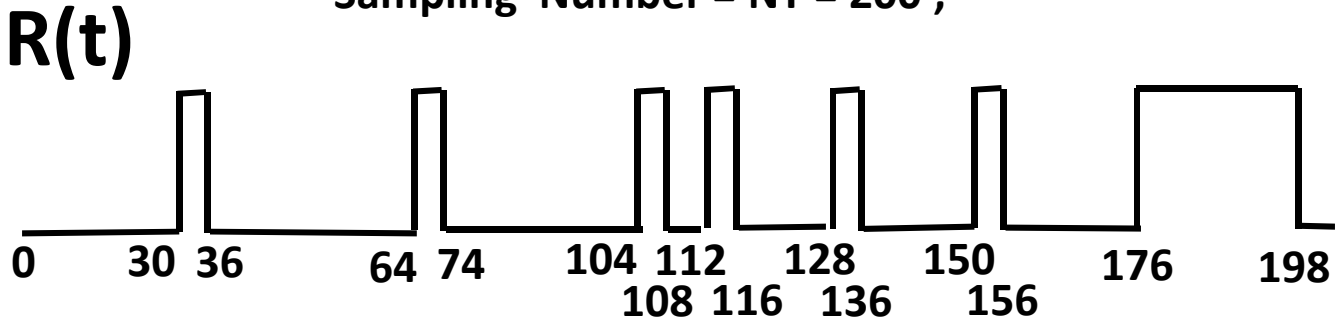
inRSFF_1 - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V)
*** input for RSFF_1 ***
  t   S   R
10000 0   0
10010 1   0
10020 0   0
10030 0   1
10036 0   0
10040 1   0
10048 0   0
10056 1   0
10060 0   0
10064 0   1
10074 0   0
10080 1   0
10100 0   0
10104 0   1
10108 0   0
10112 0   1
10116 0   0
10120 1   0
10128 1   1
-----Error
10136 1   0
10140 0   0
10150 0   1
10156 0   0
10160 1   0
10166 0   0
10176 0   1
-----Error
10180 1   1
10194 0   1
10198 0   0
10200 0   0

```

入力信号関数 R(t) の定義
Input File Name = inRSFF_1.txt

あらかじめ、RR[i] for i=1 to NT ; の値を求めます。

Sampling Number = NT = 200 ;



```

int R( void ) { int i;

for (i=1;i<NT+1;i++) {

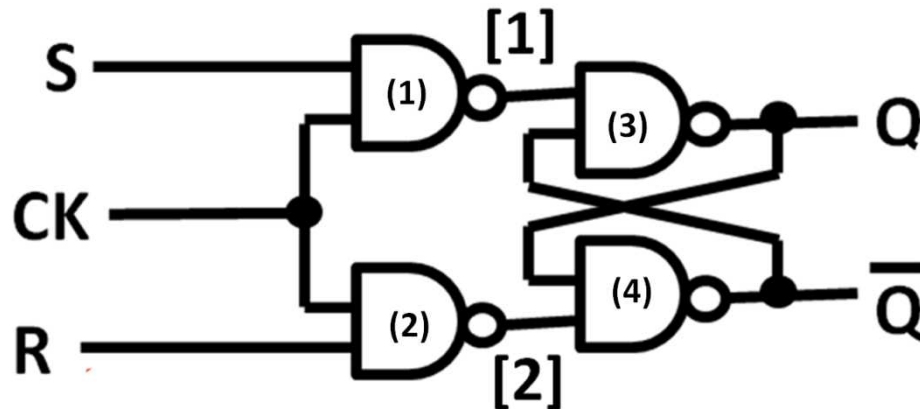
    if( t>=TT[NT+1-i] ) {

        if( RR[NT+1-i]==1) return 1; return 0; } }

return 0; }

```


Subroutine Function int inputRSFF(void)の定義



禁止
入力 →

R	S	CK	Q	\bar{Q}
0	0	⌚	Q_0	\bar{Q}_0
0	1	⌚	1	0
1	0	⌚	0	1
1	1	⌚	0/1	0/1
-	-	⌚ 以外	Q_0	\bar{Q}_0

/****** DCDL Code *****/

```
define main( ){ input CK,R,S; output Q,invQ;
```

```
[CK][R][S] RSFF(1) -> [Q][invQ] ;
```

```
for( t = 0 to 200 ) print t,CK,S,R,Q,invQ,QQ,invQQ,[1],[2] ; }
```

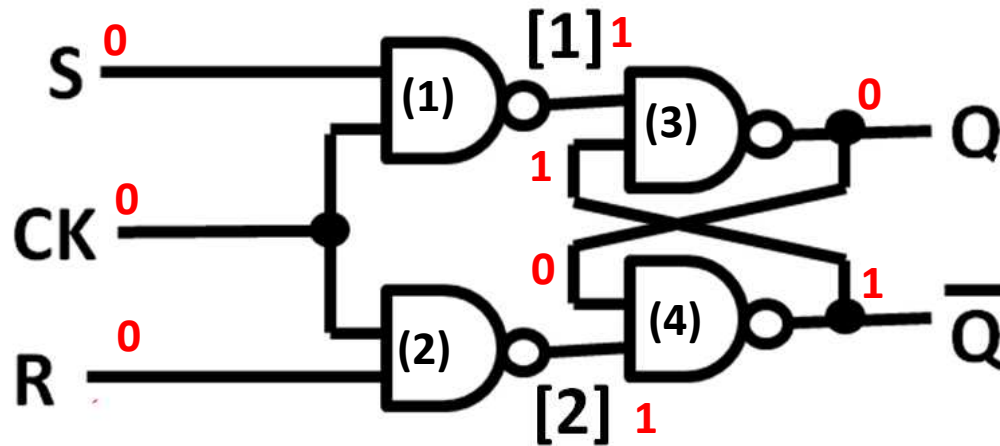
*****/

●DCDL CodeをC-program Source Codeに変換したものが RSFF_1C.txt となります。

Input File Name = inRSFF_1.txt を使って、C-program Source Code(RSFF_1C.txt)を実行することにより、Output File Name = outRSFF_1.txt に計算結果が出力されます。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

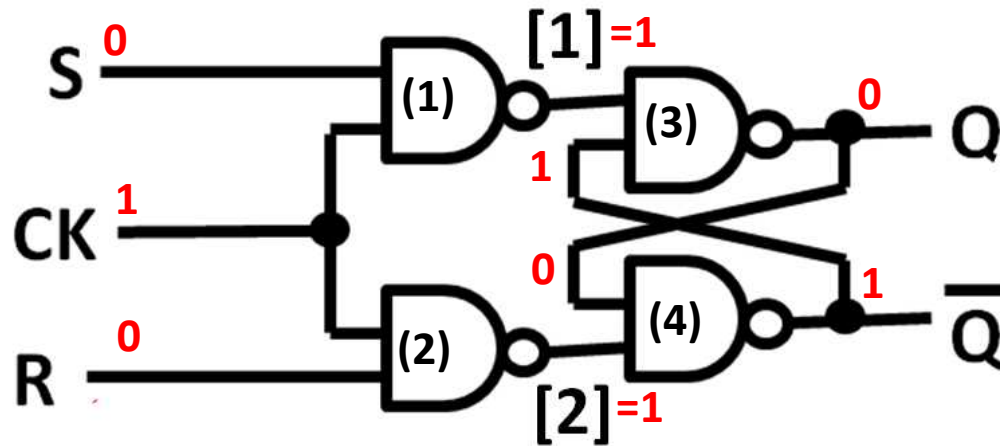
time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10000.0	0	0	0	0	1	0	1	1	1
10001.0	1	0	0	0	1	0	1	1	1
10001.5	0	0	0	0	1	0	1	1	1
10003.0	1	0	0	0	1	0	1	1	1
10003.5	0	0	0	0	1	0	1	1	1
10005.0	1	0	0	0	1	0	1	1	1
10005.5	0	0	0	0	1	0	1	1	1
10007.0	1	0	0	0	1	0	1	1	1
10007.5	0	0	0	0	1	0	1	1	1
10009.0	1	0	0	0	1	0	1	1	1
10009.5	0	0	0	0	1	0	1	1	1
10010.0	0	1	0	0	1	0	1	1	1

初期値の設定
@ t = 0

CK=S=R=0;
[1]=[2]=1;
Q=0: invQ=1;
QQ=0: invQQ=1;
とします。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

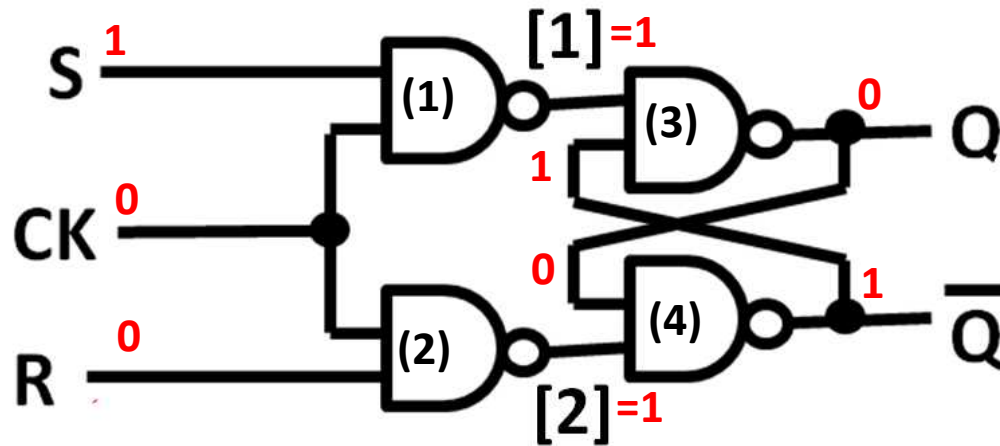
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10000.0	0	0	0	0	1	0	1	1	1
10001.0	1	0	0	0	1	0	1	1	1
10001.5	0	0	0	0	1	0	1	1	1
10003.0	1	0	0	0	1	0	1	1	1
10003.5	0	0	0	0	1	0	1	1	1
10005.0	1	0	0	0	1	0	1	1	1
10005.5	0	0	0	0	1	0	1	1	1
10007.0	1	0	0	0	1	0	1	1	1
10007.5	0	0	0	0	1	0	1	1	1
10009.0	1	0	0	0	1	0	1	1	1
10009.5	0	0	0	0	1	0	1	1	1
10010.0	0	1	0	0	1	0	1	1	1

CK=1 になっても R=S=0 なので [1]=[2]=1 のままで、Q と invQ の値には変化ありません。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

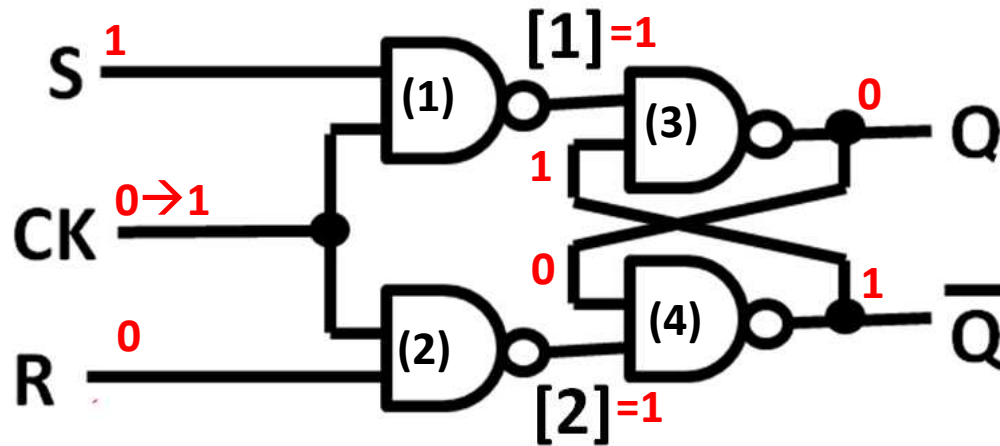
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10000.0	0	0	0	0	1	0	1	1	1
10001.0	1	0	0	0	1	0	1	1	1
10001.5	0	0	0	0	1	0	1	1	1
10003.0	1	0	0	0	1	0	1	1	1
10003.5	0	0	0	0	1	0	1	1	1
10005.0	1	0	0	0	1	0	1	1	1
10005.5	0	0	0	0	1	0	1	1	1
10007.0	1	0	0	0	1	0	1	1	1
10007.5	0	0	0	0	1	0	1	1	1
10009.0	1	0	0	0	1	0	1	1	1
10009.5	0	0	0	0	1	0	1	1	1
10010.0	0	1	0	0	1	0	1	1	1

S=1 になっても
CK=0 なので
[1]=[2]=1 のままで、
Q と invQ の値には
変化ありません。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

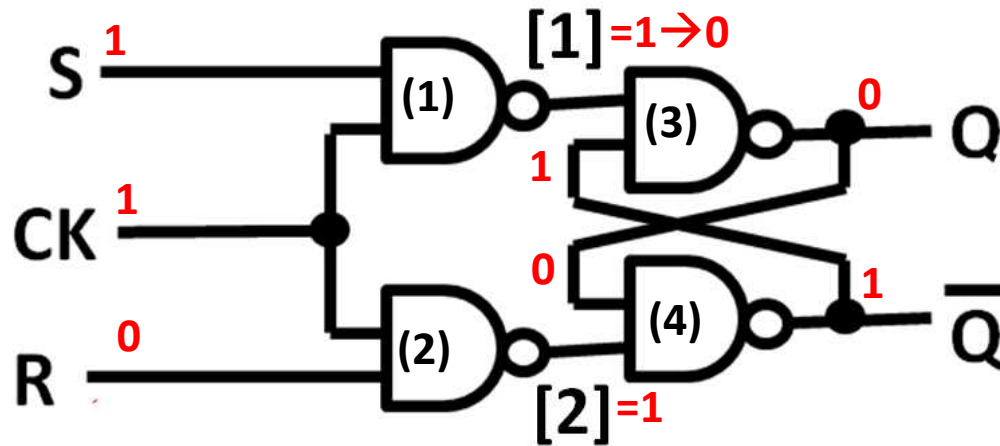
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10009.5	0	0	0	0	1	0	1	1	1
10010.0	0	1	0	0	1	0	1	1	1
10011.0	1	1	0	0	1	0	1	1	1
10011.1	1	1	0	0	1	0	1	0	1
10011.2	1	1	0	1	1	1	1	0	1
10011.4	1	1	0	1	0	1	0	0	1
10011.5	0	1	0	1	0	1	0	0	1
10011.6	0	1	0	1	0	1	0	1	1
10013.0	1	1	0	1	0	1	0	1	1
10013.1	1	1	0	1	0	1	0	0	1
10013.5	0	1	0	1	0	1	0	0	1
10013.6	0	1	0	1	0	1	0	1	1

CK=1 になったので S=1 の値が取り込まれます。NAND(1)回路のGate遅延があるので、[1] の値はまだ 1 です。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

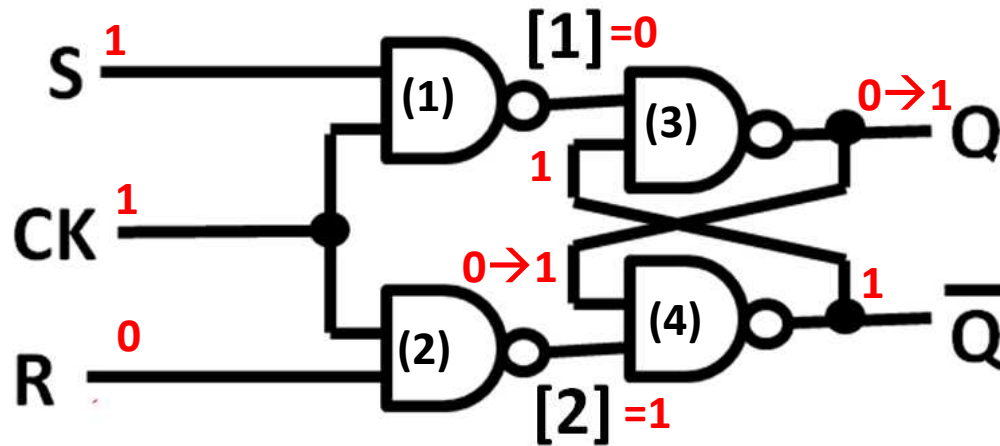
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10009.5	0	0	0	0	1	0	1	1	1
10010.0	0	1	0	0	1	0	1	1	1
10011.0	1	1	0	0	1	0	1	1	1
10011.1	1	1	0	0	1	0	1	0	1
10011.2	1	1	0	1	1	1	1	0	1
10011.4	1	1	0	1	0	1	0	0	1
10011.5	0	1	0	1	0	1	0	0	1
10011.6	0	1	0	1	0	1	0	1	1
10013.0	1	1	0	1	0	1	0	1	1
10013.1	1	1	0	1	0	1	0	0	1
10013.5	0	1	0	1	0	1	0	0	1
10013.6	0	1	0	1	0	1	0	1	1

NAND(1)回路のGate遅延が
終わり、[1]=0 になります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

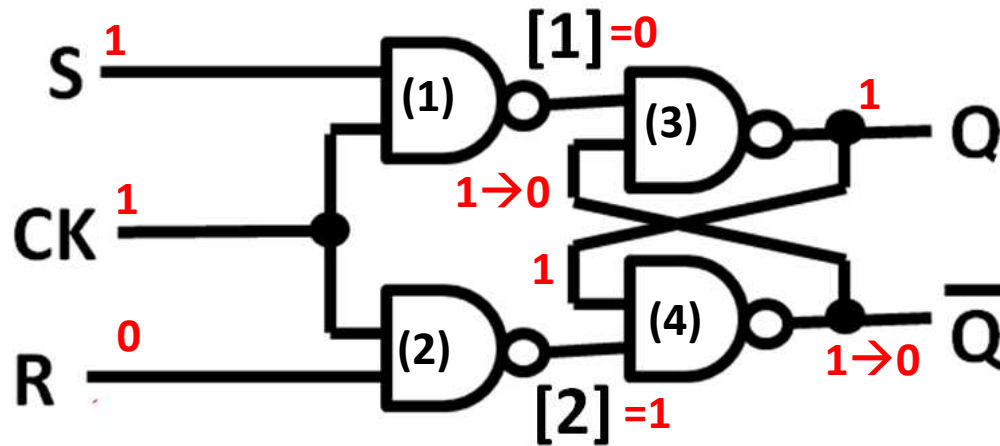
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10009.5	0	0	0	0	1	0	1	1	1
10010.0	0	1	0	0	1	0	1	1	1
10011.0	1	1	0	0	1	0	1	1	1
10011.1	1	1	0	0	1	0	1	0	1
10011.2	1	1	0	1	1	1	1	0	1
10011.4	1	1	0	1	0	1	0	0	1
10011.5	0	1	0	1	0	1	0	0	1
10011.6	0	1	0	1	0	1	0	1	1
10013.0	1	1	0	1	0	1	0	1	1
10013.1	1	1	0	1	0	1	0	0	1
10013.5	0	1	0	1	0	1	0	0	1
10013.6	0	1	0	1	0	1	0	1	1

NAND(3) の出力 Q が、
Q = 1 となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10009.5	0	0	0	0	1	0	1	1	1
10010.0	0	1	0	0	1	0	1	1	1
10011.0	1	1	0	0	1	0	1	1	1
10011.1	1	1	0	0	1	0	1	0	1
10011.2	1	1	0	1	1	1	1	0	1
10011.4	1	1	0	1	0	1	0	0	1
10011.5	0	1	0	1	0	1	0	0	1
10011.6	0	1	0	1	0	1	0	1	1
10013.0	1	1	0	1	0	1	0	1	1
10013.1	1	1	0	1	0	1	0	0	1
10013.5	0	1	0	1	0	1	0	0	1
10013.6	0	1	0	1	0	1	0	1	1

NAND(4)の出力 が、
invQ=0 となります。

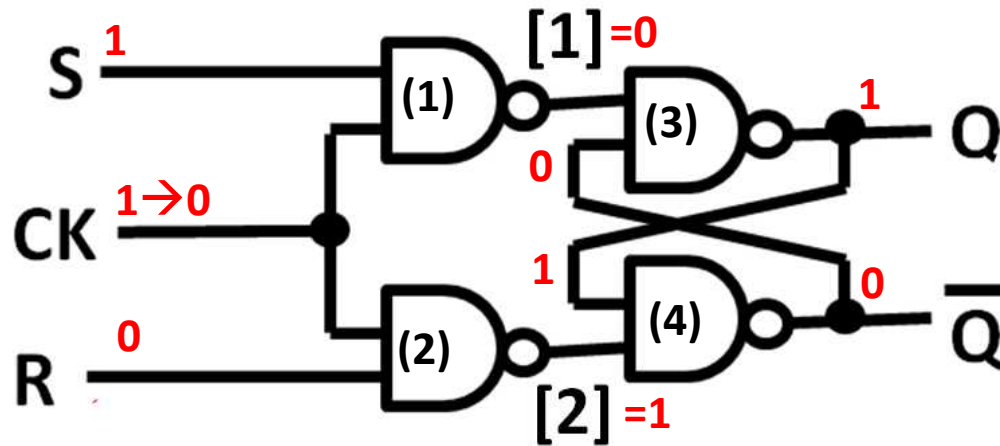
これでs=1の入力信号が
RSFF回路の中を伝搬して
状態が(Q,invQ)=(0,1)から
(Q,invQ)=(1,0)になりました。

CKパルスの幅は少なくとも
この伝搬遅延時間が終了
する迄の、 $\Delta t=0.3$ が必要
となります。

ここでは dt=0.4 の場合の
計算結果を掲示しています。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

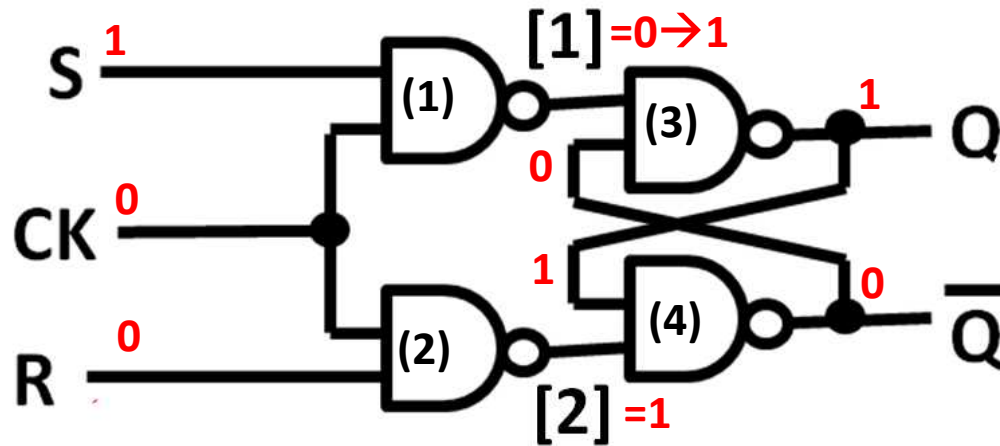
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10009.5	0	0	0	0	1	0	1	1	1
10010.0	0	1	0	0	1	0	1	1	1
10011.0	1	1	0	0	1	0	1	1	1
10011.1	1	1	0	0	1	0	1	0	1
10011.2	1	1	0	1	1	1	1	0	1
10011.4	1	1	0	1	0	1	0	0	1
10011.5	0	1	0	1	0	1	0	0	1
10011.6	0	1	0	1	0	1	0	1	1
10013.0	1	1	0	1	0	1	0	1	1
10013.1	1	1	0	1	0	1	0	0	1
10013.5	0	1	0	1	0	1	0	0	1
10013.6	0	1	0	1	0	1	0	1	1

CK=0 になっても、内部状態には、変化はありません。
(Q,invQ)=(1,0) を維持します。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

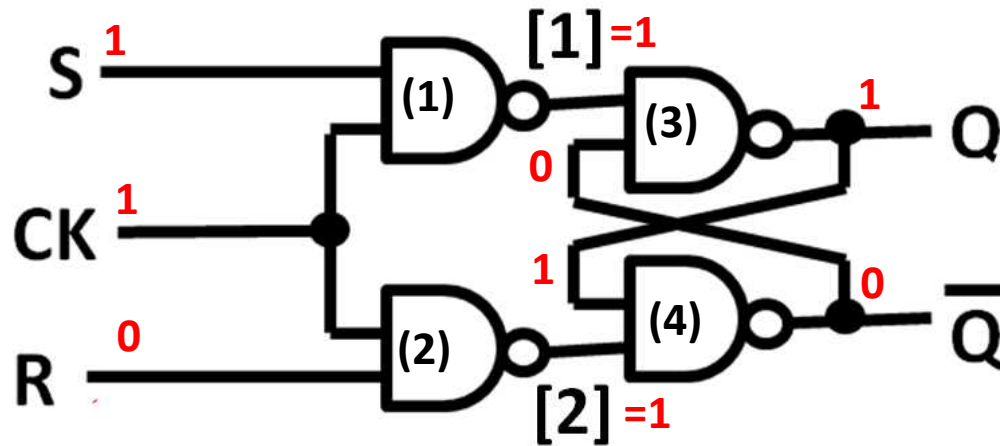
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10009.5	0	0	0	0	1	0	1	1	1
10010.0	0	1	0	0	1	0	1	1	1
10011.0	1	1	0	0	1	0	1	1	1
10011.1	1	1	0	0	1	0	1	0	1
10011.2	1	1	0	1	1	1	1	0	1
10011.4	1	1	0	1	0	1	0	0	1
10011.5	0	1	0	1	0	1	0	0	1
10011.6	0	1	0	1	0	1	0	1	1
10013.0	1	1	0	1	0	1	0	1	1
10013.1	1	1	0	1	0	1	0	0	1
10013.5	0	1	0	1	0	1	0	0	1
10013.6	0	1	0	1	0	1	0	1	1

NAND(1)の出力が [1]=0 から [1]=1 に変化します。これで [1]=[2]=1となり、最終安定状態になります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

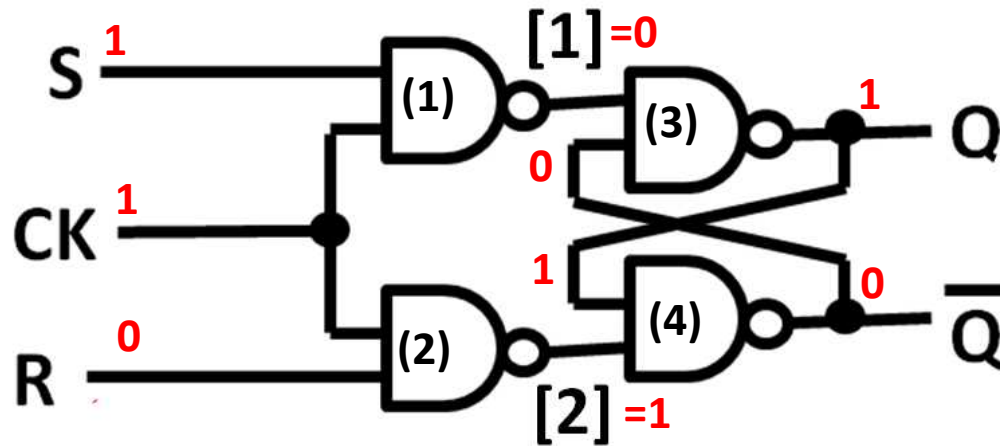
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10009.5	0	0	0	0	1	0	1	1	1
10010.0	0	1	0	0	1	0	1	1	1
10011.0	1	1	0	0	1	0	1	1	1
10011.1	1	1	0	0	1	0	1	0	1
10011.2	1	1	0	1	1	1	1	0	1
10011.4	1	1	0	1	0	1	0	0	1
10011.5	0	1	0	1	0	1	0	0	1
10011.6	0	1	0	1	0	1	0	1	1
10013.0	1	1	0	1	0	1	0	1	1
10013.1	1	1	0	1	0	1	0	0	1
10013.5	0	1	0	1	0	1	0	0	1
10013.6	0	1	0	1	0	1	0	1	1

また周期的に CK=1 になりますが、s=1 のままなので内部状態には、変化はありません。(Q,invQ)=(1,0) を維持します。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

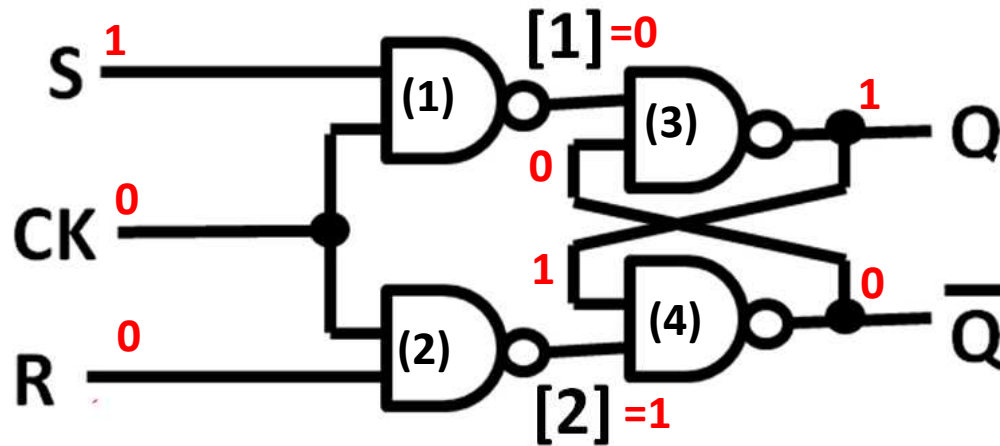
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10009.5	0	0	0	0	1	0	1	1	1
10010.0	0	1	0	0	1	0	1	1	1
10011.0	1	1	0	0	1	0	1	1	1
10011.1	1	1	0	0	1	0	1	0	1
10011.2	1	1	0	1	1	1	1	0	1
10011.4	1	1	0	1	0	1	0	0	1
10011.5	0	1	0	1	0	1	0	0	1
10011.6	0	1	0	1	0	1	0	1	1
10013.0	1	1	0	1	0	1	0	1	1
10013.1	1	1	0	1	0	1	0	0	1
10013.5	0	1	0	1	0	1	0	0	1
10013.6	0	1	0	1	0	1	0	1	1

CK=1 の値がNAND(1)の出力を[1]=0 としますが、内部状態には、変化はありません。(Q,invQ)=(1,0)を維持します。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

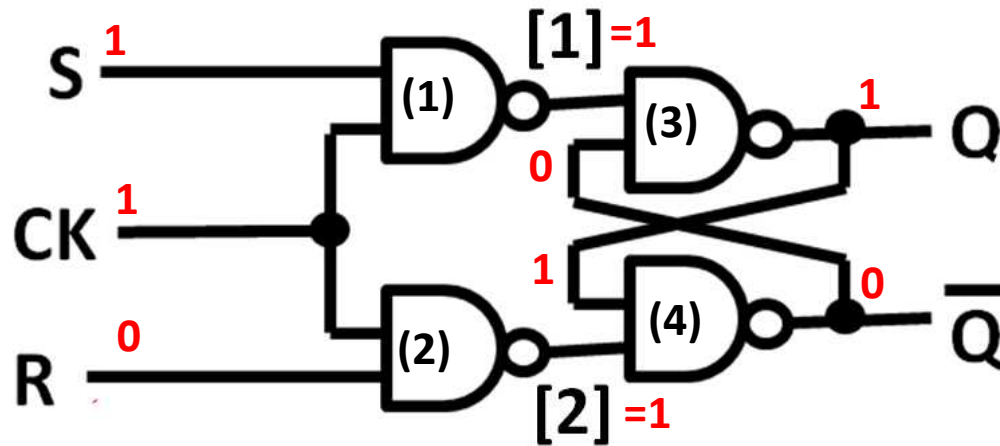
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10009.5	0	0	0	0	1	0	1	1	1
10010.0	0	1	0	0	1	0	1	1	1
10011.0	1	1	0	0	1	0	1	1	1
10011.1	1	1	0	0	1	0	1	0	1
10011.2	1	1	0	1	1	1	1	0	1
10011.4	1	1	0	1	0	1	0	0	1
10011.5	0	1	0	1	0	1	0	0	1
10011.6	0	1	0	1	0	1	0	1	1
10013.0	1	1	0	1	0	1	0	1	1
10013.1	1	1	0	1	0	1	0	0	1
10013.5	0	1	0	1	0	1	0	0	1
10013.6	0	1	0	1	0	1	0	1	1

再びCK=0 となりますが、内部状態には変化はありません。(Q,invQ)=(1,0)を維持します。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

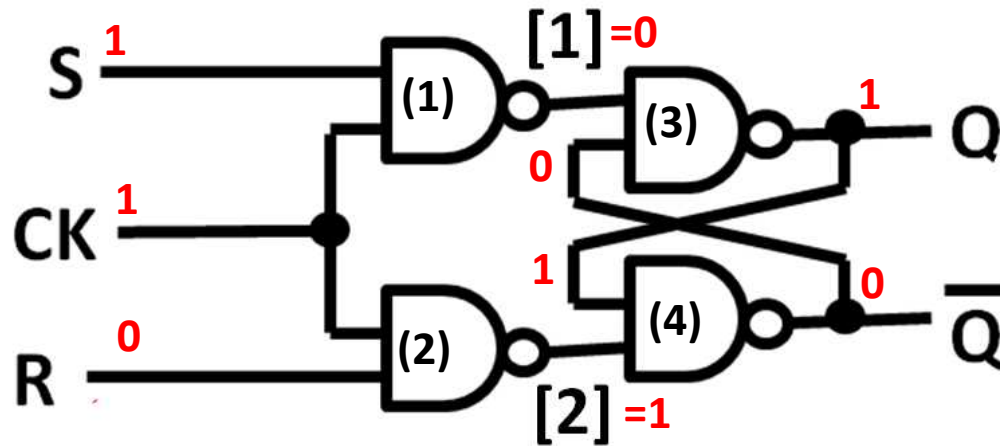
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10009.5	0	0	0	0	1	0	1	1	1
10010.0	0	1	0	0	1	0	1	1	1
10011.0	1	1	0	0	1	0	1	1	1
10011.1	1	1	0	0	1	0	1	0	1
10011.2	1	1	0	1	1	1	1	0	1
10011.4	1	1	0	1	0	1	0	0	1
10011.5	0	1	0	1	0	1	0	0	1
10011.6	0	1	0	1	0	1	0	1	1
10013.0	1	1	0	1	0	1	0	1	1
10013.1	1	1	0	1	0	1	0	0	1
10013.5	0	1	0	1	0	1	0	0	1
10013.6	0	1	0	1	0	1	0	1	1

CK=0 の値が内部伝搬して NAND(1)の出力を再び [1]=1 とします。再び、[1]=[2]=1 となり、内部状態には、変化はありません。(Q,invQ)=(1,0)を維持します。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

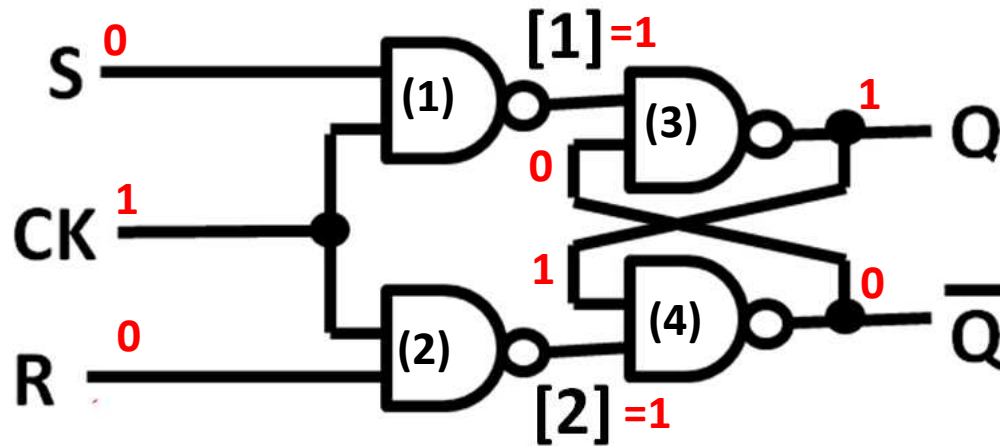
time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10013.5	0	1	0	1	0	1	0	0	1
10013.6	0	1	0	1	0	1	0	1	1
10015.0	1	1	0	1	0	1	0	1	1
10015.1	1	1	0	1	0	1	0	0	1
10015.5	0	1	0	1	0	1	0	0	1
10015.6	0	1	0	1	0	1	0	1	1
10017.0	1	1	0	1	0	1	0	1	1
10017.1	1	1	0	1	0	1	0	0	1
10017.5	0	1	0	1	0	1	0	0	1
10017.6	0	1	0	1	0	1	0	1	1
10019.0	1	1	0	1	0	1	0	1	1
10019.1	1	1	0	1	0	1	0	0	1

時刻が奇数整数の場合
周期的に CK=1 になりますが
S=1 のままなので、内部状態
に変化はありません。

(Q,invQ)=(1,0) を維持します。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10019.0	1	1	0	1	0	1	0	1	1
10019.1	1	1	0	1	0	1	0	0	1
10019.5	0	1	0	1	0	1	0	0	1
10019.6	0	1	0	1	0	1	0	1	1
10020.0	0	0	0	1	0	1	0	1	1
10021.0	1	0	0	1	0	1	0	1	1
10021.5	0	0	0	1	0	1	0	1	1
10023.0	1	0	0	1	0	1	0	1	1
10023.5	0	0	0	1	0	1	0	1	1
10025.0	1	0	0	1	0	1	0	1	1
10025.5	0	0	0	1	0	1	0	1	1
10027.0	1	0	0	1	0	1	0	1	1

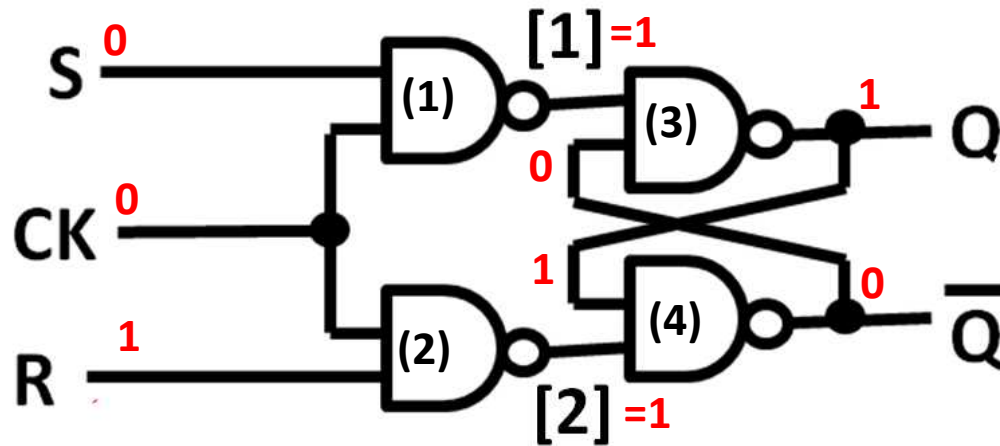
S=0 になりました。

S=R=0 ですのでCKの値に関係なく内部状態には変化はありません。

(Q,invQ)=(1,0) を維持します。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10027.0	1	0	0	1	0	1	0	1	1
10027.5	0	0	0	1	0	1	0	1	1
10029.0	1	0	0	1	0	1	0	1	1
10029.5	0	0	0	1	0	1	0	1	1
10030.0	0	0	1	1	0	1	0	1	1
10031.0	1	0	1	1	0	1	0	1	1
10031.1	1	0	1	1	0	1	0	1	0
10031.2	1	0	1	1	1	1	1	1	0
10031.4	1	0	1	0	1	0	1	1	0
10031.5	0	0	1	0	1	0	1	1	0
10031.6	0	0	1	0	1	0	1	1	1
10033.0	1	0	1	0	1	0	1	1	1

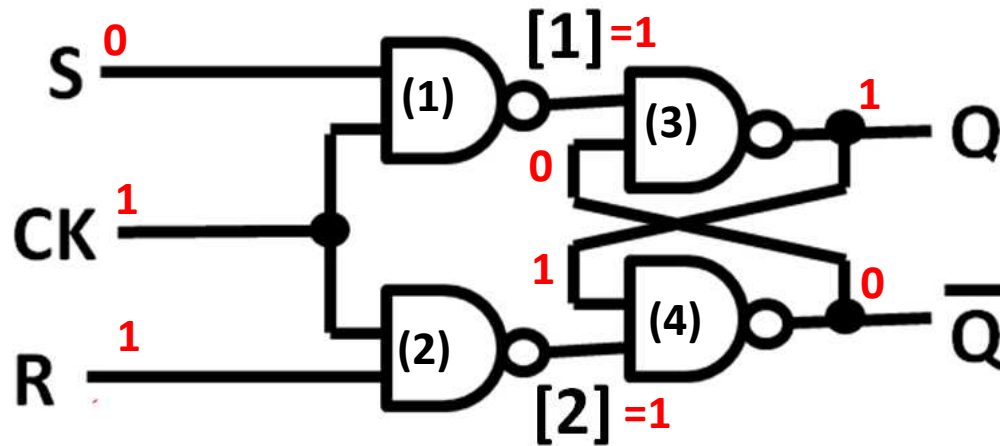
今度は、R=1 になりました。
CK=0 なのでまだ内部状態には変化はありません。

[2]=1 のままです。

(Q,invQ)=(1,0) を維持します。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

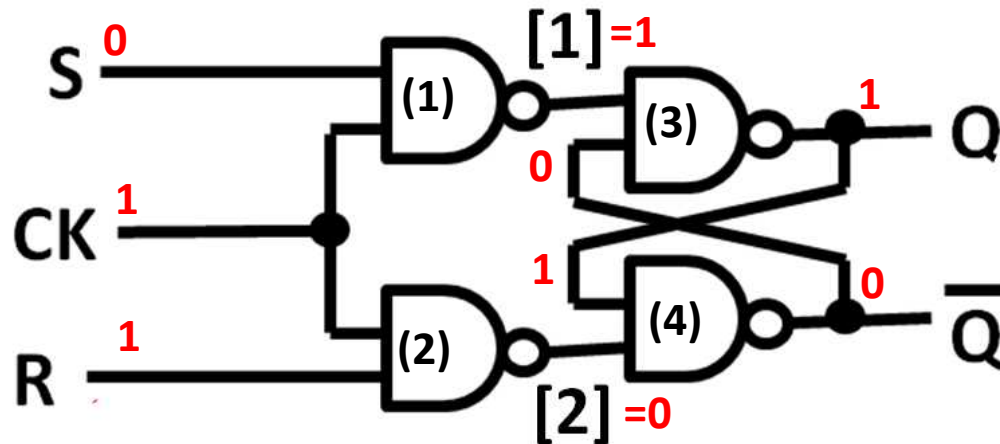
time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10027.0	1	0	0	1	0	1	0	1	1
10027.5	0	0	0	1	0	1	0	1	1
10029.0	1	0	0	1	0	1	0	1	1
10029.5	0	0	0	1	0	1	0	1	1
10030.0	0	0	1	1	0	1	0	1	1
10031.0	1	0	1	1	0	1	0	1	1
10031.1	1	0	1	1	0	1	0	1	0
10031.2	1	0	1	1	1	1	1	1	0
10031.4	1	0	1	0	1	0	1	1	0
10031.5	0	0	1	0	1	0	1	1	0
10031.6	0	0	1	0	1	0	1	1	1
10033.0	1	0	1	0	1	0	1	1	1

CK=1 となり、NAND(2)回路の出力[2]に信号が伝搬しますが、この時点ではまだNAND(2)回路にGate遅延があるので 出力は [2]=1 のままです。

(Q,invQ)=(1,0) を維持します。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10027.0	1	0	0	1	0	1	0	1	1
10027.5	0	0	0	1	0	1	0	1	1
10029.0	1	0	0	1	0	1	0	1	1
10029.5	0	0	0	1	0	1	0	1	1
10030.0	0	0	1	1	0	1	0	1	1
10031.0	1	0	1	1	0	1	0	1	1
10031.1	1	0	1	1	0	1	0	1	0
10031.2	1	0	1	1	1	1	1	1	0
10031.4	1	0	1	0	1	0	1	1	0
10031.5	0	0	1	0	1	0	1	1	0
10031.6	0	0	1	0	1	0	1	1	1
10033.0	1	0	1	0	1	0	1	1	1

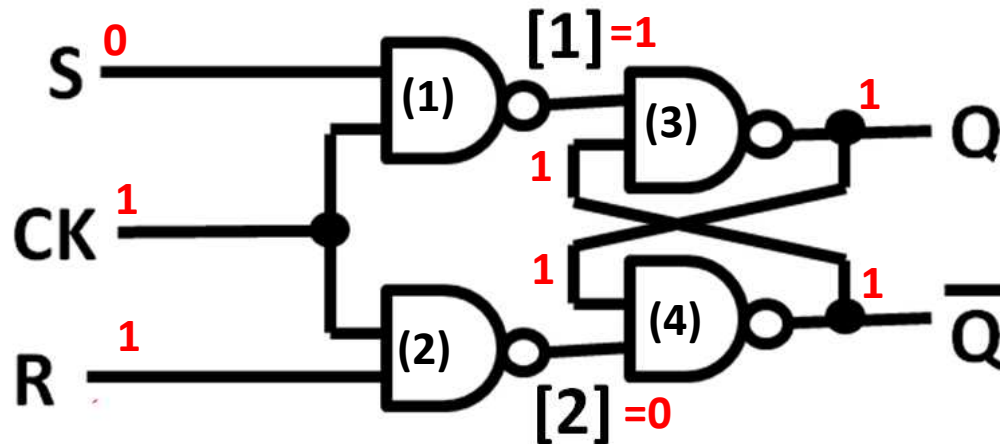
CK=1 の値が、NAND(2)回路の出力[2] に信号が伝搬しました。[2]=0 となります。

まだNAND(3)回路とNAND(4)回路の出力には遅延があるので信号は伝搬していません。

まだ(Q,invQ)=(1,0) を維持します。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10027.0	1	0	0	1	0	1	0	1	1
10027.5	0	0	0	1	0	1	0	1	1
10029.0	1	0	0	1	0	1	0	1	1
10029.5	0	0	0	1	0	1	0	1	1
10030.0	0	0	1	1	0	1	0	1	1
10031.0	1	0	1	1	0	1	0	1	1
10031.1	1	0	1	1	0	1	0	1	0
10031.2	1	0	1	1	1	1	1	1	0
10031.4	1	0	1	0	1	0	1	1	0
10031.5	0	0	1	0	1	0	1	1	0
10031.6	0	0	1	0	1	0	1	1	1
10033.0	1	0	1	0	1	0	1	1	1

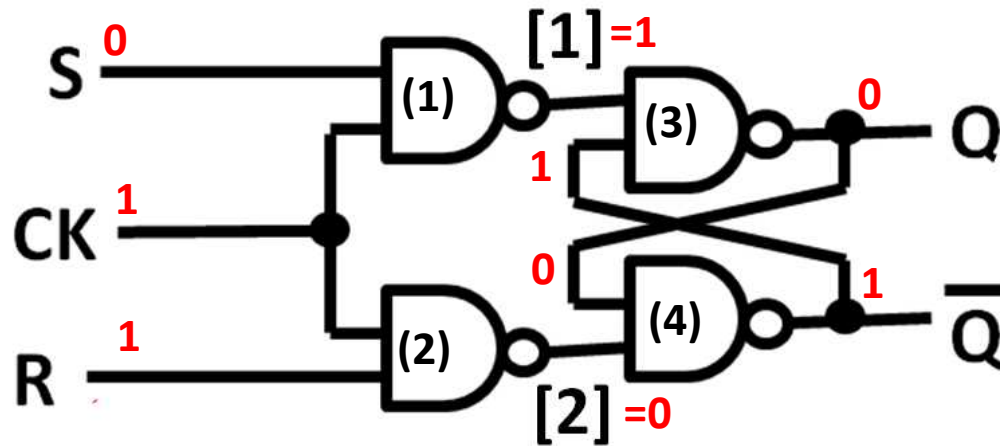
NAND(4)回路の出力 invQ がここでやっと遅延していた信号が伝搬して invQ=1 となります。

NAND(3)回路の出力には、まだ遅延があるので信号は伝搬していません。

瞬間的に (Q,invQ)=(1,1) となっていますが、NAND(3)回路の遅延時間が過ぎるとすぐに最終値の (Q,invQ)=(0,1) になります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

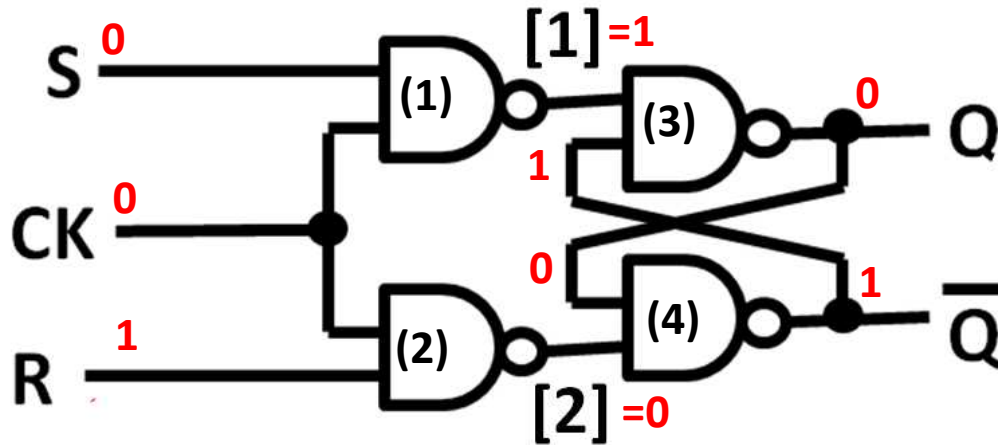
time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10027.0	1	0	0	1	0	1	0	1	1
10027.5	0	0	0	1	0	1	0	1	1
10029.0	1	0	0	1	0	1	0	1	1
10029.5	0	0	0	1	0	1	0	1	1
10030.0	0	0	1	1	0	1	0	1	1
10031.0	1	0	1	1	0	1	0	1	1
10031.1	1	0	1	1	0	1	0	1	0
10031.2	1	0	1	1	0	1	0	1	0
10031.4	1	0	1	0	1	0	1	1	0
10031.5	0	0	1	0	1	0	1	1	0
10031.6	0	0	1	0	1	0	1	1	1
10033.0	1	0	1	0	1	0	1	1	1

NAND(3)回路の遅延時間が過ぎたのでNAND(3)回路の出力Q が Q=0 となります。最終値の (Q,invQ)=(0,1) になりました。

CK パルスの幅はここでも少なくともこの伝搬遅延時間が終了する迄の、少なくとも $\Delta t=0.4$ が必要となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10027.0	1	0	0	1	0	1	0	1	1
10027.5	0	0	0	1	0	1	0	1	1
10029.0	1	0	0	1	0	1	0	1	1
10029.5	0	0	0	1	0	1	0	1	1
10030.0	0	0	1	1	0	1	0	1	1
10031.0	1	0	1	1	0	1	0	1	1
10031.1	1	0	1	1	0	1	0	1	0
10031.2	1	0	1	1	1	1	1	1	0
10031.4	1	0	1	0	1	0	1	1	0
10031.5	0	0	1	0	1	0	1	1	0
10031.6	0	0	1	0	1	0	1	1	1
10033.0	1	0	1	0	1	0	1	1	1

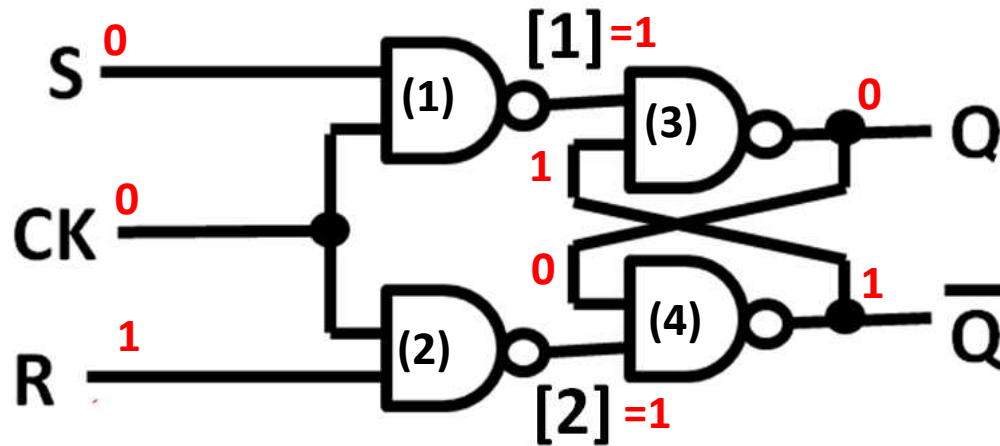
CK=0 となり入力信号(R,S) が切斷されます。

まだNAND(2)回路のGate遅延がありますので[2]=0 ですがすぐに次の瞬間に[2]=1 になります。

出力値は (Q,invQ)=(0,1) のままです

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10027.0	1	0	0	1	0	1	0	1	1
10027.5	0	0	0	1	0	1	0	1	1
10029.0	1	0	0	1	0	1	0	1	1
10029.5	0	0	0	1	0	1	0	1	1
10030.0	0	0	1	1	0	1	0	1	1
10031.0	1	0	1	1	0	1	0	1	1
10031.1	1	0	1	1	0	1	0	1	0
10031.2	1	0	1	1	1	1	1	1	0
10031.4	1	0	1	0	1	0	1	1	0
10031.5	0	0	1	0	1	0	1	1	0
10031.6	0	0	1	0	1	0	1	1	1
10033.0	1	0	1	0	1	0	1	1	1

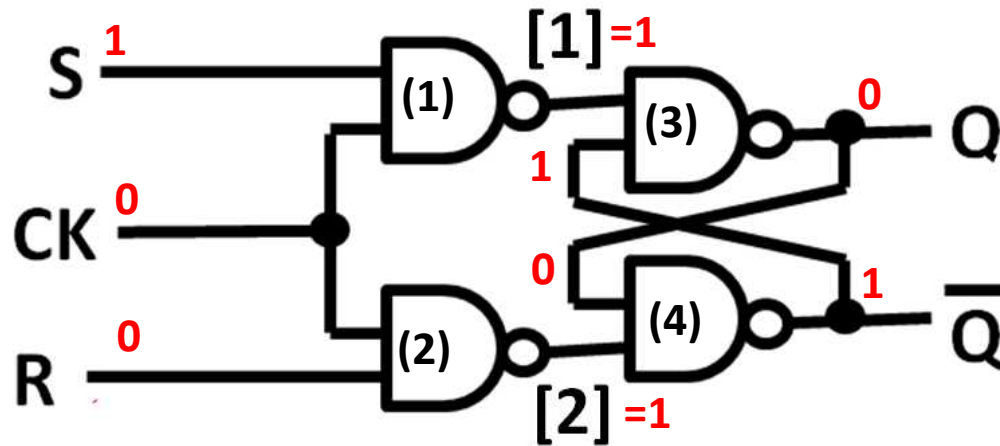
NAND(2)回路のGate遅延が
終わり、[2]=1 になりました。

これで最終安定状態となります。

(Q,invQ)=(0,1) を維持します。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

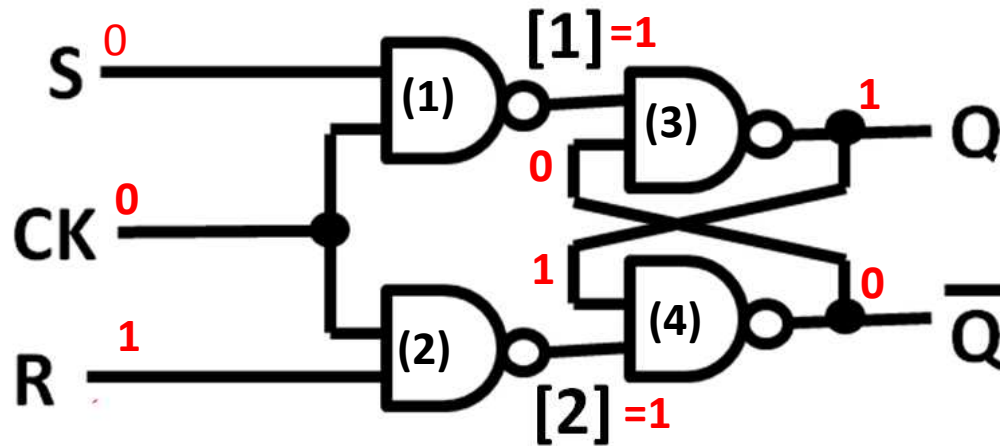
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10035.6	0	0	1	0	1	0	1	1	1
10036.0	0	0	0	0	1	0	1	1	1
10037.0	1	0	0	0	1	0	1	1	1
10037.5	0	0	0	0	1	0	1	1	1
10039.0	1	0	0	0	1	0	1	1	1
10039.5	0	0	0	0	1	0	1	1	1
10040.0	0	1	0	0	1	0	1	1	1
10041.0	1	1	0	0	1	0	1	0	1
10041.1	1	1	0	0	1	0	1	0	1
10041.2	1	1	0	1	1	1	1	0	1
10041.4	1	1	0	1	0	1	0	0	1
10041.5	0	1	0	1	0	1	0	0	1

再び s=1 となり、その後 CK=1 となり、Q=1となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

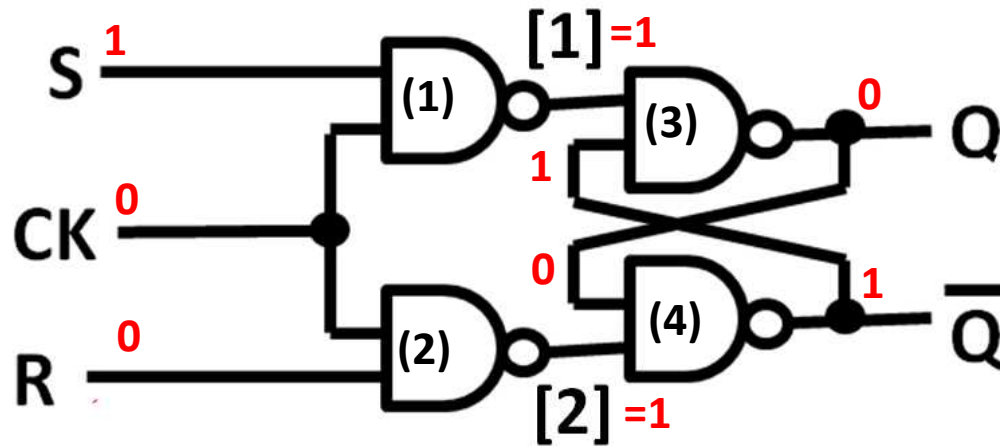
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10063.0	1	0	0	1	0	1	0	1	1
10063.5	0	0	0	1	0	1	0	1	1
10064.0	0	0	1	1	0	1	0	1	1
10065.0	1	0	1	1	0	1	0	1	1
10065.1	1	0	1	1	0	1	0	1	0
10065.2	1	0	1	1	1	1	1	1	0
10065.4	1	0	1	0	1	0	1	1	0
10065.5	0	0	1	0	1	0	1	1	0
10065.6	0	0	1	0	1	0	1	1	1
10067.0	1	0	1	0	1	0	1	1	1
10067.1	1	0	1	0	1	0	1	1	0
10067.5	0	0	1	0	1	0	1	1	0

今度は R=1 となり、その後 CK=1 となり、Q=0 ; invQ=1 となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

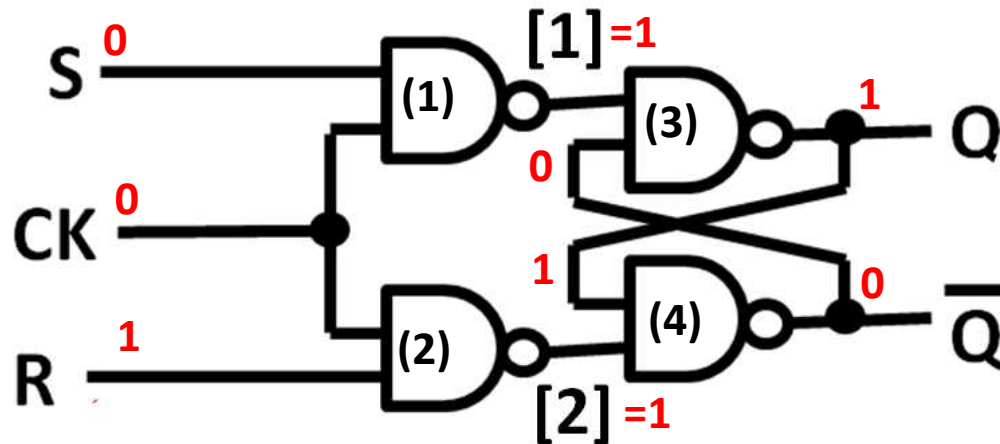
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10077.0	1	0	0	0	1	0	1	1	1
10077.5	0	0	0	0	1	0	1	1	1
10079.0	1	0	0	0	1	0	1	1	1
10079.5	0	0	0	0	1	0	1	1	1
10080.0	0	1	0	0	1	0	1	1	1
10081.0	1	1	0	0	1	0	1	1	1
10081.1	1	1	0	0	1	0	1	0	1
10081.2	1	1	0	1	1	1	1	0	1
10081.4	1	1	0	1	0	1	0	0	1
10081.5	0	1	0	1	0	1	0	0	1
10081.6	0	1	0	1	0	1	0	1	1
10083.0	1	1	0	1	0	1	0	1	1

再び s=1 となり、その後
CK=1 となり、Q=1となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

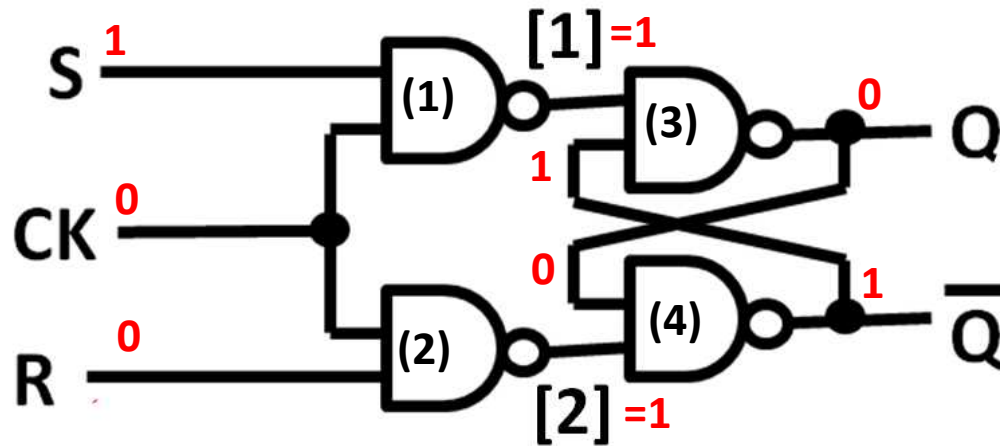
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10103.0	1	0	0	1	0	1	0	1	1
10103.5	0	0	0	1	0	1	0	1	1
10104.0	0	0	1	1	0	1	0	1	1
10105.0	1	0	1	1	0	1	0	1	1
10105.1	1	0	1	1	0	1	0	1	0
10105.2	1	0	1	1	1	1	1	1	0
10105.4	1	0	1	0	1	0	1	1	0
10105.5	0	0	1	0	1	0	1	1	0
10105.6	0	0	1	0	1	0	1	1	1
10107.0	1	0	1	0	1	0	1	1	1
10107.1	1	0	1	0	1	0	1	1	0
10107.5	0	0	1	0	1	0	1	1	0
10107.6	0	0	1	0	1	0	1	1	1

再び、R=1 となり、その後 CK=1 となり、Q=0; invQ=1 となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

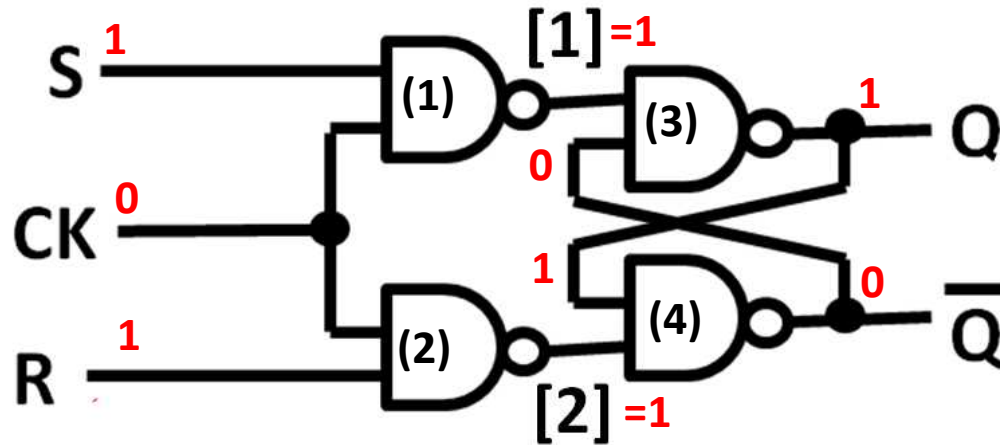
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10119.0	1	0	0	0	1	0	1	1	1
10119.5	0	0	0	0	1	0	1	1	1
10120.0	0	1	0	0	1	0	1	1	1
10121.0	1	1	0	0	1	0	1	1	1
10121.1	1	1	0	0	1	0	1	0	1
10121.2	1	1	0	1	1	1	0	0	1
10121.4	1	1	0	1	0	1	0	0	1
10121.5	0	1	0	1	0	1	0	0	1
10121.6	0	1	0	1	0	1	0	1	1
10123.0	1	1	0	1	0	1	0	1	1
10123.1	1	1	0	1	0	1	0	0	1
10123.5	0	1	0	1	0	1	0	0	1

再び s=1 となり、その後
CK=1 となり、Q=1となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q ₀	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q ₀	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

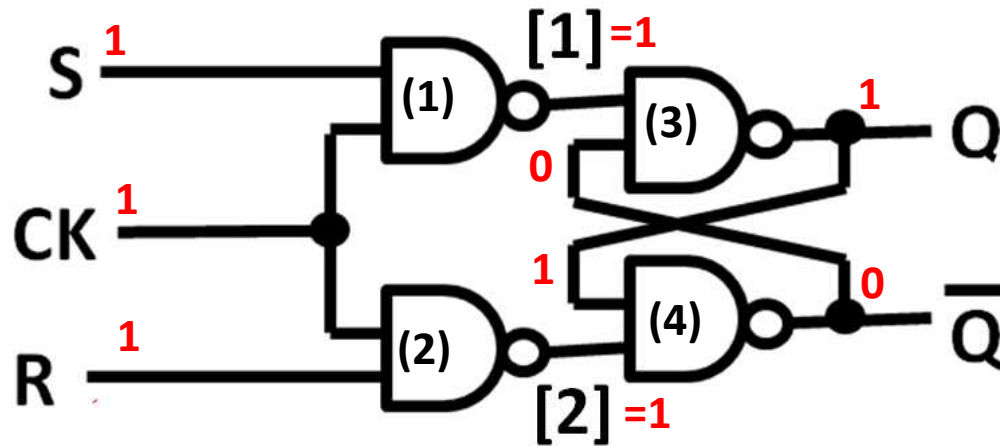
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10127.1	1	1	0	1	0	1	0	0	1
10127.5	0	1	0	1	0	1	0	0	1
10127.6	0	1	0	1	0	1	0	1	1
10128.0	0	1	1	1	0	1	0	1	1
10129.0	1	1	1	1	0	1	0	1	1
10129.1	1	1	1	1	0	1	0	0	0
10129.2	1	1	1	1	1	1	1	0	0
10129.5	0	1	1	1	1	1	1	0	0
10129.6	0	1	1	1	1	1	1	1	1
10129.7	0	1	1	0	0	0	0	1	1
10129.9	0	1	1	1	1	1	1	1	1
10130.1	0	1	1	0	0	0	0	1	1

そこでわざと、S=R=1 としてみました。まだCK=0 なので S=R=1 の値は内部に取り込まれていません。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

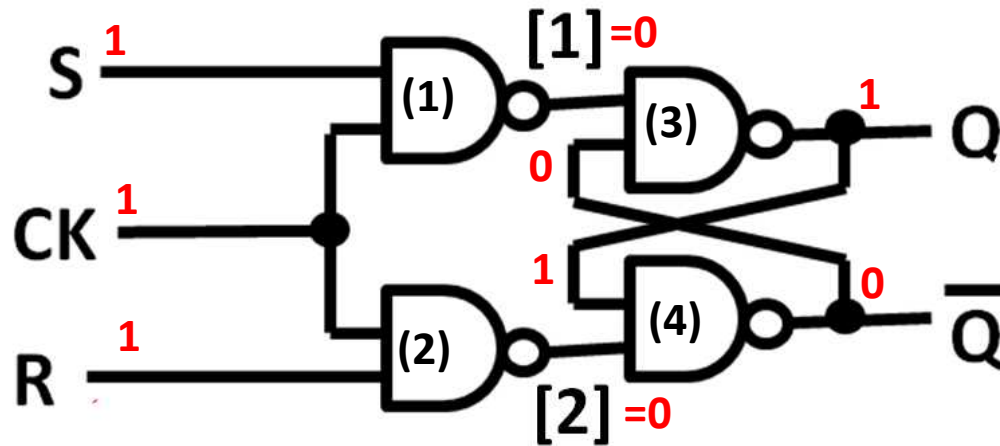
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10127.1	1	1	0	1	0	1	0	0	1
10127.5	0	1	0	1	0	1	0	0	1
10127.6	0	1	0	1	0	1	0	1	1
10128.0	0	1	1	1	0	1	0	1	1
10129.0	1	1	1	1	0	1	0	1	1
10129.1	1	1	1	1	1	1	1	0	0
10129.2	1	1	1	1	1	1	1	0	0
10129.5	0	1	1	1	1	1	1	0	0
10129.6	0	1	1	1	1	1	1	1	1
10129.7	0	1	1	0	0	0	0	1	1
10129.9	0	1	1	1	1	1	1	1	1
10130.1	0	1	1	0	0	0	0	1	1

CK=1 となりました。
S=R=1 の値が内部に
取り込まれました。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

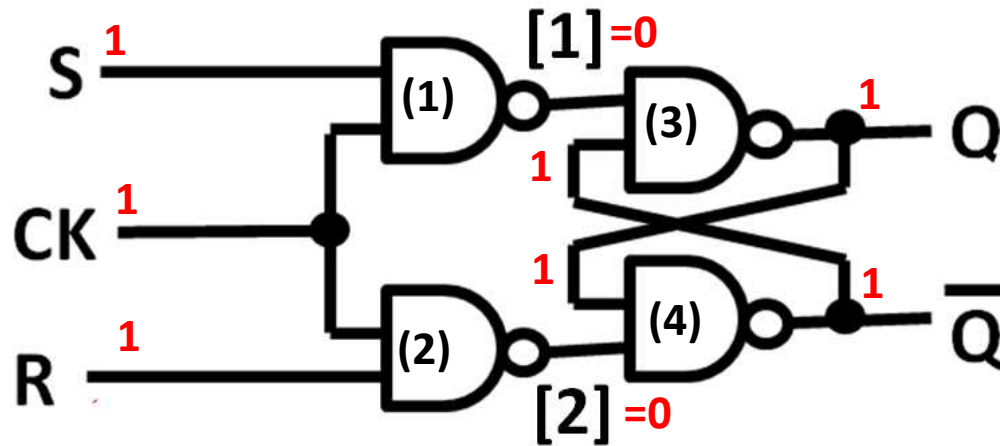
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10127.1	1	1	0	1	0	1	0	0	1
10127.5	0	1	0	1	0	1	0	0	1
10127.6	0	1	0	1	0	1	0	1	1
10128.0	0	1	1	1	0	1	0	1	1
10129.0	1	1	1	1	0	1	0	1	1
10129.1	1	1	1	1	0	1	0	0	0
10129.2	1	1	1	1	1	1	1	0	0
10129.5	0	1	1	1	1	1	1	0	0
10129.6	0	1	1	1	1	1	1	1	1
10129.7	0	1	1	0	0	0	0	1	1
10129.9	0	1	1	1	1	1	1	1	1
10130.1	0	1	1	0	0	0	0	1	1

[1]=[2]=0 となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	1	1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

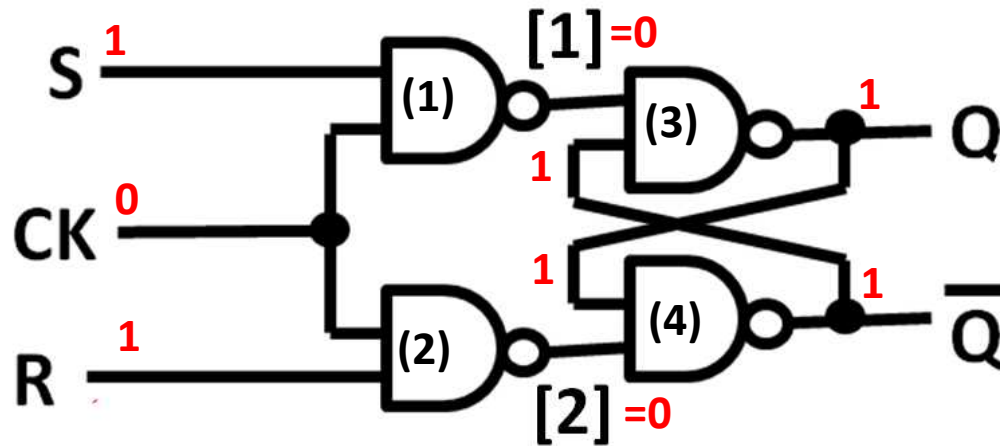
time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10127.1	1	1	0	1	0	1	0	0	1
10127.5	0	1	0	1	0	1	0	0	1
10127.6	0	1	0	1	0	1	0	1	1
10128.0	0	1	1	1	0	1	0	1	1
10129.0	1	1	1	1	0	1	0	1	1
10129.1	1	1	1	1	0	1	0	0	0
10129.2	1	1	1	1	1	1	1	0	0
10129.5	0	1	1	1	1	1	1	0	0
10129.6	0	1	1	1	1	1	1	1	1
10129.7	0	1	1	0	0	0	0	1	1
10129.9	0	1	1	1	1	1	1	1	1
10130.1	0	1	1	0	0	0	0	1	1

invQ=1 となります。

これで、Q=invQ=1
となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	1	1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

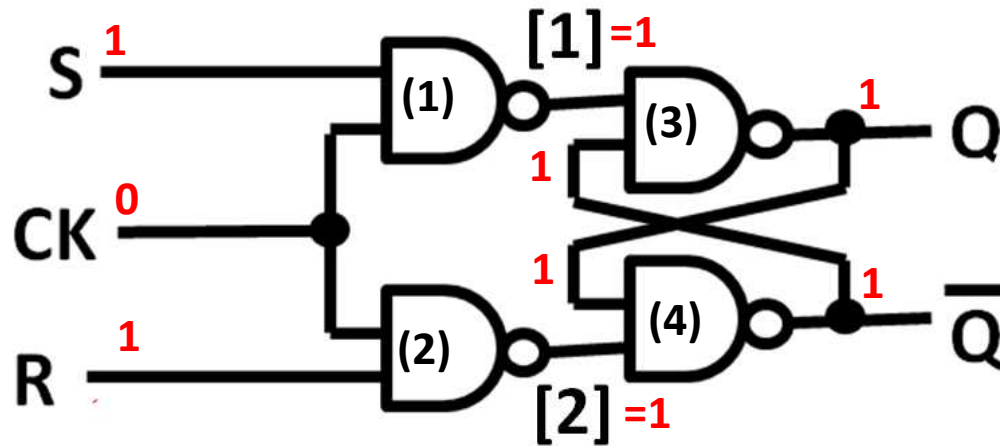
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10127.1	1	1	0	1	0	1	0	0	1
10127.5	0	1	0	1	0	1	0	0	1
10127.6	0	1	0	1	0	1	0	1	1
10128.0	0	1	1	1	0	1	0	1	1
10129.0	1	1	1	1	0	1	0	1	1
10129.1	1	1	1	1	0	1	0	0	0
10129.2	1	1	1	1	1	1	1	0	0
10129.5	0	1	1	1	1	1	1	0	0
10129.6	0	1	1	1	1	1	1	1	1
10129.7	0	1	1	0	0	0	0	1	1
10129.9	0	1	1	1	1	1	1	1	1
10130.1	0	1	1	0	0	0	0	1	1

CK=0 となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	1	1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

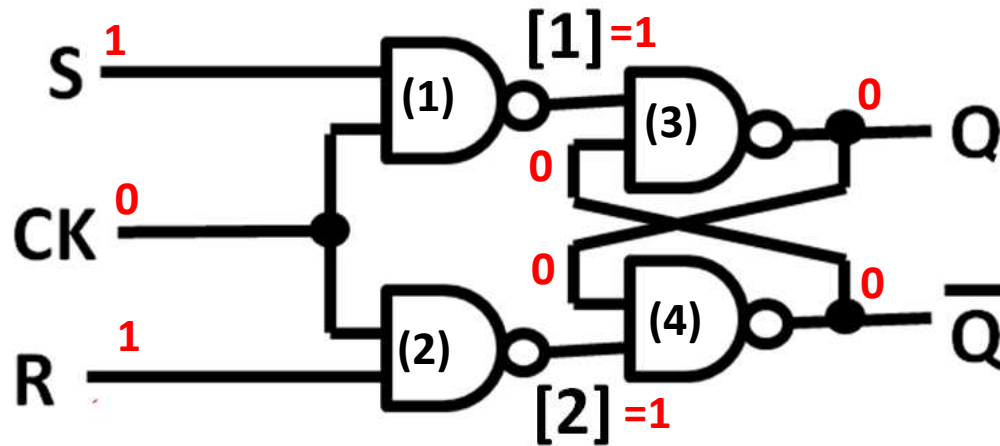
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10127.1	1	1	0	1	0	1	0	0	1
10127.5	0	1	0	1	0	1	0	0	1
10127.6	0	1	0	1	0	1	0	1	1
10128.0	0	1	1	1	0	1	0	1	1
10129.0	1	1	1	1	0	1	0	1	1
10129.1	1	1	1	1	0	1	0	0	0
10129.2	1	1	1	1	1	1	1	0	0
10129.5	0	1	1	1	1	1	1	0	0
10129.6	0	1	1	1	1	1	1	1	1
10129.7	0	1	1	0	0	0	0	1	1
10129.9	0	1	1	1	1	1	1	1	1
10130.1	0	1	1	0	0	0	0	1	1

[1]=[2]=1 となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0	0
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

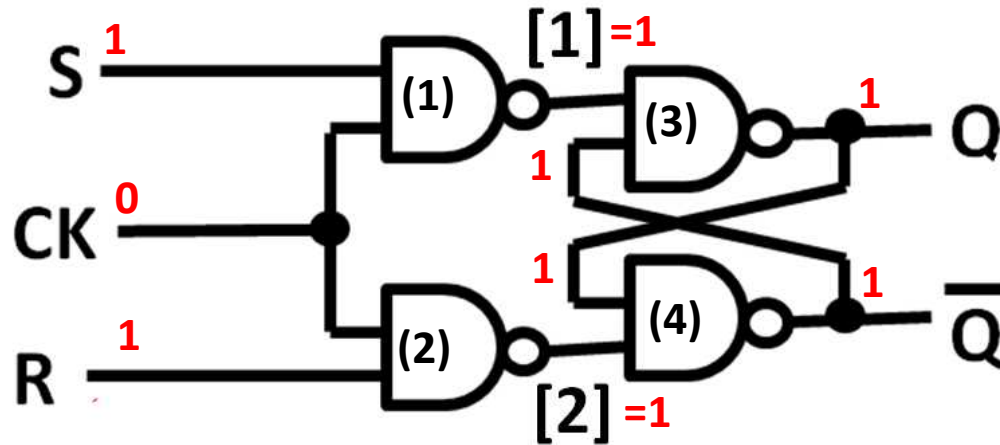
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10127.1	1	1	0	1	0	1	0	0	1
10127.5	0	1	0	1	0	1	0	0	1
10127.6	0	1	0	1	0	1	0	1	1
10128.0	0	1	1	1	0	1	0	1	1
10129.0	1	1	1	1	0	1	0	1	1
10129.1	1	1	1	1	0	1	0	0	0
10129.2	1	1	1	1	1	1	1	0	0
10129.5	0	1	1	1	1	1	1	0	0
10129.6	0	1	1	1	1	1	1	1	1
10129.7	0	1	1	0	0	0	0	1	1
10129.9	0	1	1	1	1	1	1	1	1
10130.1	0	1	1	0	0	0	0	1	1

Q=invQ=0 となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	1	1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

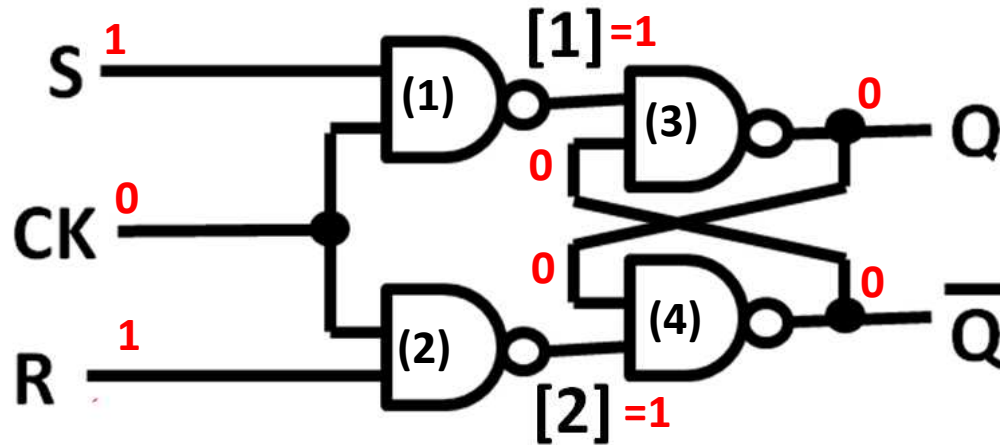
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10127.1	1	1	0	1	0	1	0	0	1
10127.5	0	1	0	1	0	1	0	0	1
10127.6	0	1	0	1	0	1	0	1	1
10128.0	0	1	1	1	0	1	0	1	1
10129.0	1	1	1	1	0	1	0	1	1
10129.1	1	1	1	1	0	1	0	0	0
10129.2	1	1	1	1	1	1	1	0	0
10129.5	0	1	1	1	1	1	1	0	0
10129.6	0	1	1	1	1	1	1	1	1
10129.7	0	1	1	0	0	0	0	1	1
10129.9	0	1	1	1	1	1	1	1	1
10130.1	0	1	1	0	0	0	0	1	1

Q=invQ=1 となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0	0
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

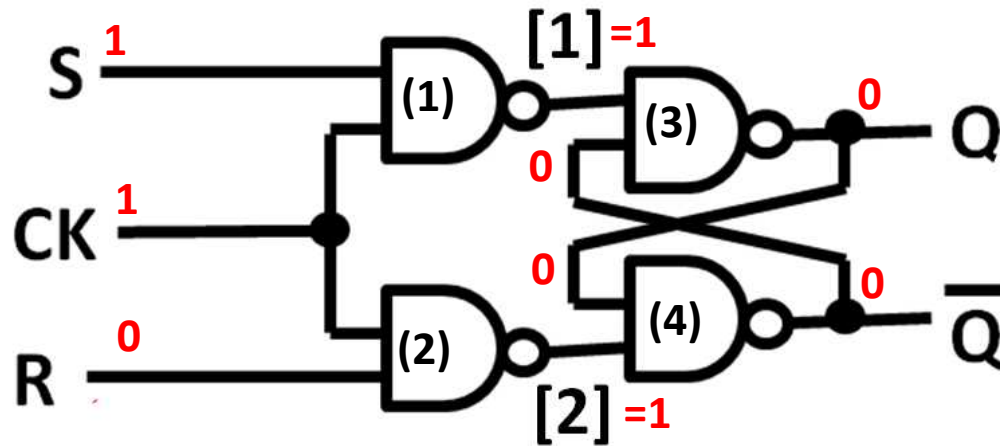
time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10127.1	1	1	0	1	0	1	0	0	1
10127.5	0	1	0	1	0	1	0	0	1
10127.6	0	1	0	1	0	1	0	1	1
10128.0	0	1	1	1	0	1	0	1	1
10129.0	1	1	1	1	0	1	0	1	1
10129.1	1	1	1	1	0	1	0	0	0
10129.2	1	1	1	1	1	1	1	0	0
10129.5	0	1	1	1	1	1	1	0	0
10129.6	0	1	1	1	1	1	1	1	1
10129.7	0	1	1	0	0	0	0	1	1
10129.9	0	1	1	1	1	1	1	1	1
10130.1	0	1	1	0	0	0	0	1	1

Q=invQ=0 となります。

このままではずっと
Q=invQ=0 の状態と
Q=invQ=1 の状態を
永遠に繰り返す状態、
すなわち発振状態と
なります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

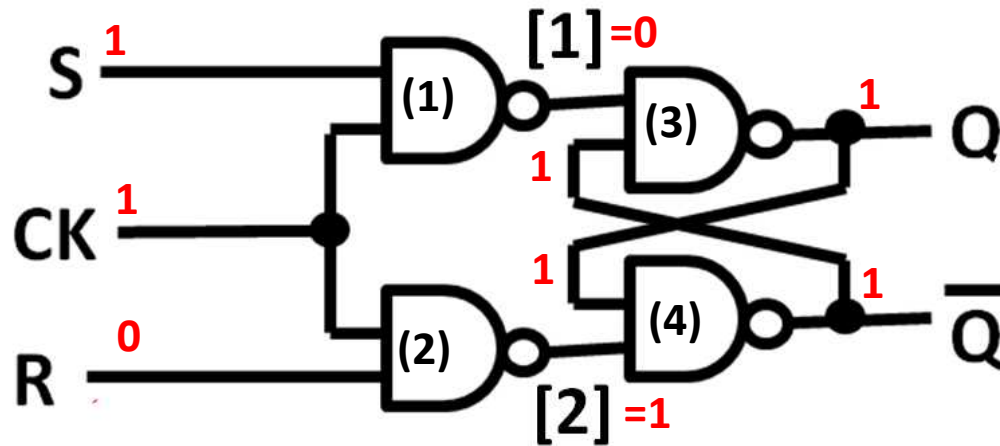
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10136.5	0	1	0	0	0	0	0	1	1
10136.7	0	1	0	1	1	1	1	1	1
10136.9	0	1	0	0	0	0	0	1	1
10137.0	1	1	0	0	0	0	0	1	1
10137.1	1	1	0	1	1	1	1	0	1
10137.3	1	1	0	1	0	1	0	0	1
10137.5	0	1	0	1	0	1	0	0	1
10137.6	0	1	0	1	0	1	0	1	1
10139.0	1	1	0	1	0	1	0	1	1
10139.1	1	1	0	1	0	1	0	0	1
10139.5	0	1	0	1	0	1	0	0	1
10139.6	0	1	0	1	0	1	0	1	1

正常な入力信号
CK=S=1;R=0 のとなり、
これでやっと発振
状態から解放される
こととなります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

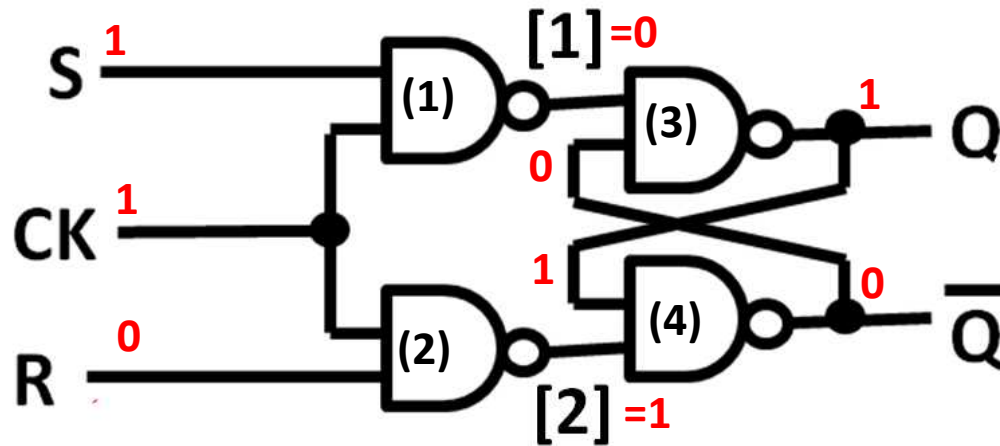
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10136.5	0	1	0	0	0	0	0	1	1
10136.7	0	1	0	1	1	1	1	1	1
10136.9	0	1	0	0	0	0	0	1	1
10137.0	1	1	0	0	0	0	0	1	1
10137.1	1	1	0	1	1	1	1	0	1
10137.3	1	1	0	1	0	1	0	0	1
10137.5	0	1	0	1	0	1	0	0	1
10137.6	0	1	0	1	0	1	0	1	1
10139.0	1	1	0	1	0	1	0	1	1
10139.1	1	1	0	1	0	1	0	0	1
10139.5	0	1	0	1	0	1	0	0	1
10139.6	0	1	0	1	0	1	0	1	1

NAND(1)の出力が
[1]=0 となりました。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

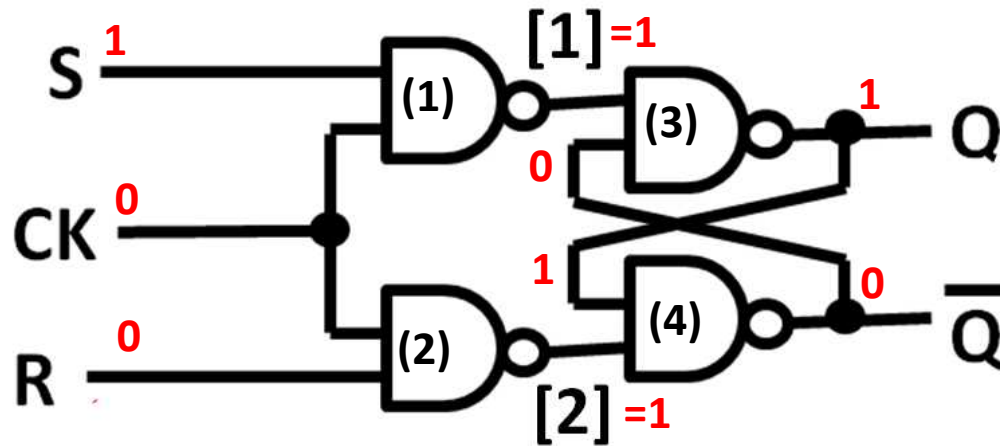
time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10136.5	0	1	0	0	0	0	0	1	1
10136.7	0	1	0	1	1	1	1	1	1
10136.9	0	1	0	0	0	0	0	1	1
10137.0	1	1	0	0	0	0	0	1	1
10137.1	1	1	0	1	1	1	1	0	1
10137.3	1	1	0	1	0	1	0	0	1
10137.5	0	1	0	1	0	1	0	0	1
10137.6	0	1	0	1	0	1	0	1	1
10139.0	1	1	0	1	0	1	0	1	1
10139.1	1	1	0	1	0	1	0	0	1
10139.5	0	1	0	1	0	1	0	0	1
10139.6	0	1	0	1	0	1	0	1	1

NAND(4)の出力が
invQ=0 となりました。

これで出力が
Q=1; invQ=0 となり
正常状態になりました。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

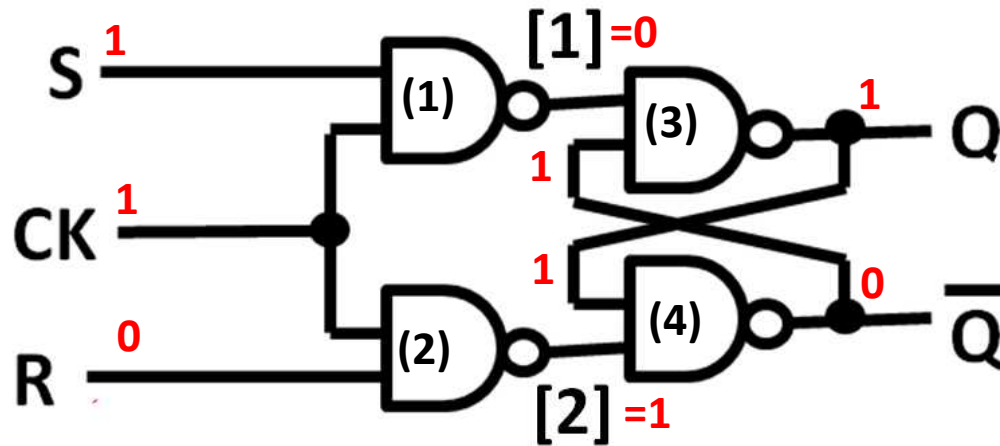
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10149.0	1	0	0	1	0	1	0	1	1
10149.5	0	0	0	1	0	1	0	1	1
10150.0	0	0	1	1	0	1	0	1	1
10151.0	1	0	1	1	0	1	0	1	1
10151.1	1	0	1	1	0	1	0	1	0
10151.2	1	0	1	1	1	1	1	1	0
10151.4	1	0	1	0	1	0	1	1	0
10151.5	0	0	1	0	1	0	1	1	0
10151.6	0	0	1	0	1	0	1	1	1
10153.0	1	0	1	0	1	0	1	1	1
10153.1	1	0	1	0	1	0	1	1	0
10153.5	0	0	1	0	1	0	1	1	0

再び R=1 となり、その後 CK=1 となり、Q=0 ; invQ=1 となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

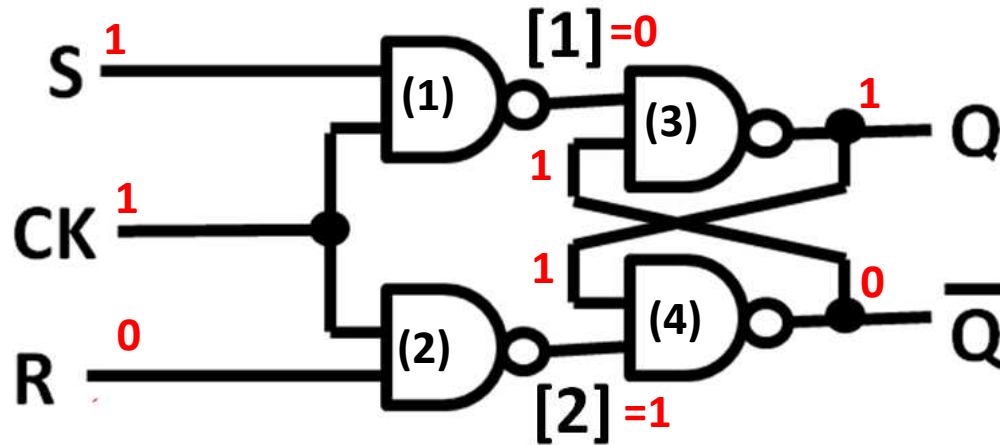
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10157.5	0	0	0	0	1	0	1	1	1
10159.0	1	0	0	0	1	0	1	1	1
10159.5	0	0	0	0	1	0	1	1	1
10160.0	0	1	0	0	1	0	1	1	1
10161.0	1	1	0	0	1	0	1	1	1
10161.1	1	1	0	0	1	0	1	0	1
10161.2	1	1	0	1	1	1	1	0	1
10161.4	1	1	0	1	0	1	0	0	1
10161.5	0	1	0	1	0	1	0	0	1
10161.6	0	1	0	1	0	1	0	1	1
10163.0	1	1	0	1	0	1	0	1	1
10163.1	1	1	0	1	0	1	0	0	1

再び s=1 となり、その後 CK=1 となり、Q=1 ; invQ=0 となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

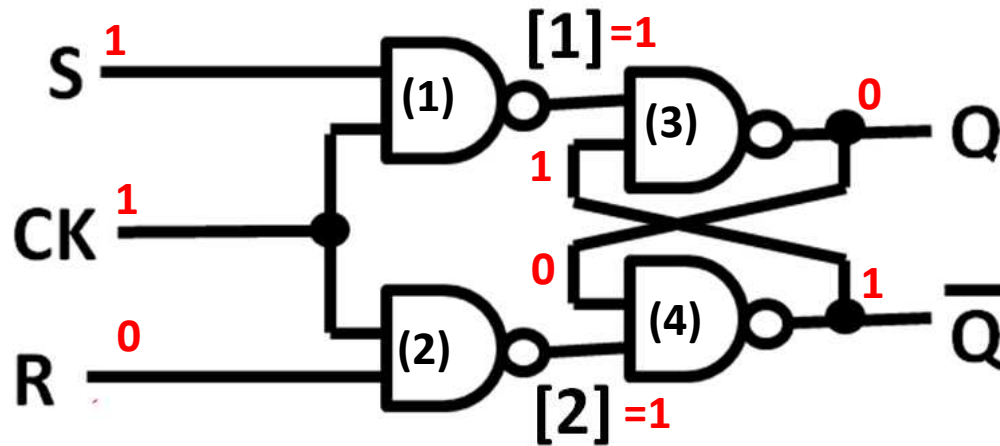
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10175.0	1	0	0	1	0	1	0	1	1
10175.5	0	0	0	1	0	1	0	1	1
10176.0	0	0	1	1	0	1	0	1	1
10177.0	1	0	1	1	0	1	0	1	1
10177.1	1	0	1	1	0	1	0	1	0
10177.2	1	0	1	1	1	1	1	1	0
10177.4	1	0	1	0	1	0	1	1	0
10177.5	0	0	1	0	1	0	1	1	0
10177.6	0	0	1	0	1	0	1	1	1
10179.0	1	0	1	0	1	0	1	1	1
10179.1	1	0	1	0	1	0	1	1	0
10179.5	0	0	1	0	1	0	1	1	0

再び R=1 となり、その後 CK=1 となり、Q=0 ; invQ=1 となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

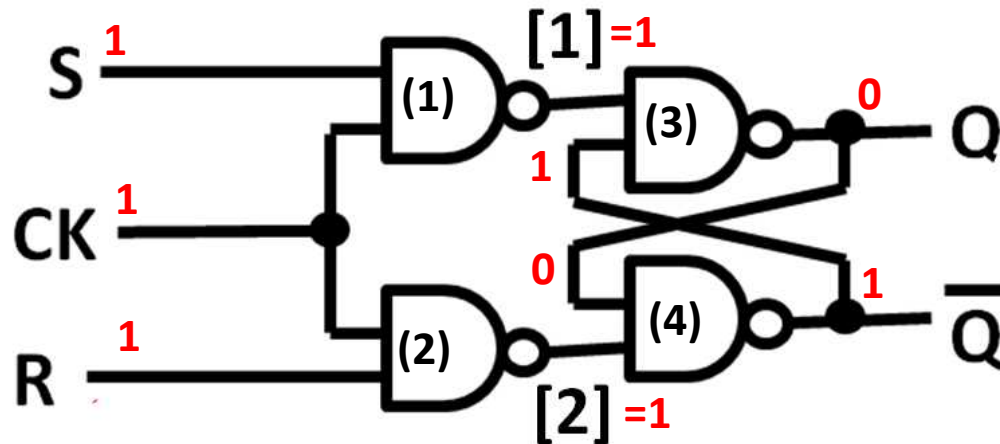
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10179.5	0	0	1	0	1	0	1	1	0
10179.6	0	0	1	0	1	0	1	1	1
10180.0	0	1	1	0	1	0	1	1	1
10181.0	1	1	1	0	1	0	1	1	1
10181.1	1	1	1	0	1	0	1	0	0
10181.2	1	1	1	1	1	1	1	0	0
10181.5	0	1	1	1	1	1	1	0	0
10181.6	0	1	1	1	1	1	1	1	1
10181.7	0	1	1	0	0	0	0	1	1
10181.9	0	1	1	1	1	1	1	1	1
10182.1	0	1	1	0	0	0	0	1	1
10182.3	0	1	1	1	1	1	1	1	1

そこで再びわざと、S=R=1としました。まだCK=0 なのでS=R=1 の値は内部に取り込まれていません。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

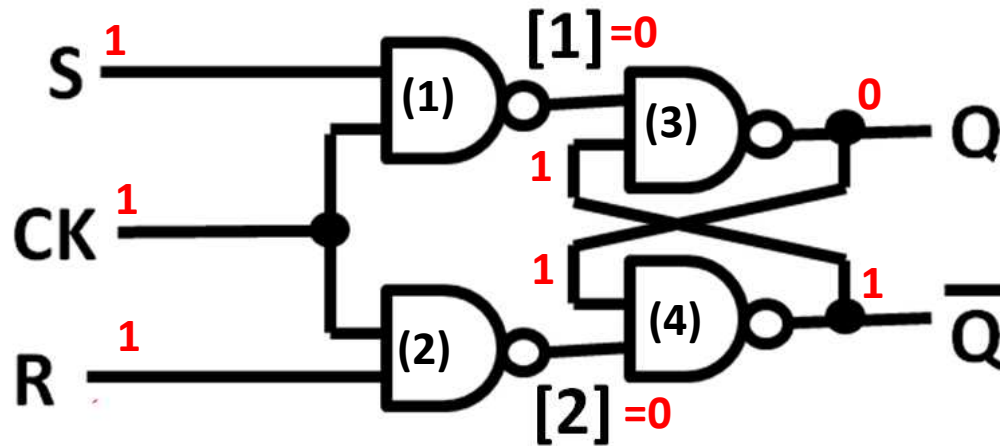
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10179.5	0	0	1	0	1	0	1	1	0
10179.6	0	0	1	0	1	0	1	1	1
10180.0	0	1	1	0	1	0	1	1	1
10181.0	1	1	1	0	1	0	1	1	1
10181.1	1	1	1	0	1	0	1	0	0
10181.2	1	1	1	1	1	1	1	0	0
10181.5	0	1	1	1	1	1	1	0	0
10181.6	0	1	1	1	1	1	1	1	1
10181.7	0	1	1	0	0	0	0	1	1
10181.9	0	1	1	1	1	1	1	1	1
10182.1	0	1	1	0	0	0	0	1	1
10182.3	0	1	1	1	1	1	1	1	1

CK=1 となり、S=R=1 の値
が内部に取り込まれました。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

禁止
入力

$\Delta t = 0.3$ 以上で RSFF() は動作する。

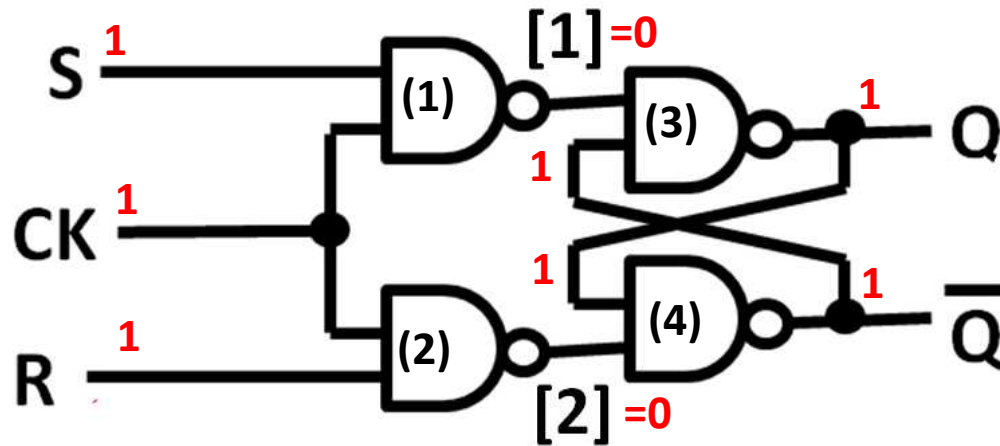
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10179.5	0	0	1	0	1	0	1	1	0
10179.6	0	0	1	0	1	0	1	1	1
10180.0	0	1	1	0	1	0	1	1	1
10181.0	1	1	1	0	1	0	1	1	1
10181.1	1	1	1	0	1	0	1	0	0
10181.2	1	1	1	1	1	1	1	0	0
10181.5	0	1	1	1	1	1	1	0	0
10181.6	0	1	1	1	1	1	1	1	1
10181.7	0	1	1	0	0	0	0	1	1
10181.9	0	1	1	1	1	1	1	1	1
10182.1	0	1	1	0	0	0	0	1	1
10182.3	0	1	1	1	1	1	1	1	1

[1]=[2]=0 となりました。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力 →

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	1	1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

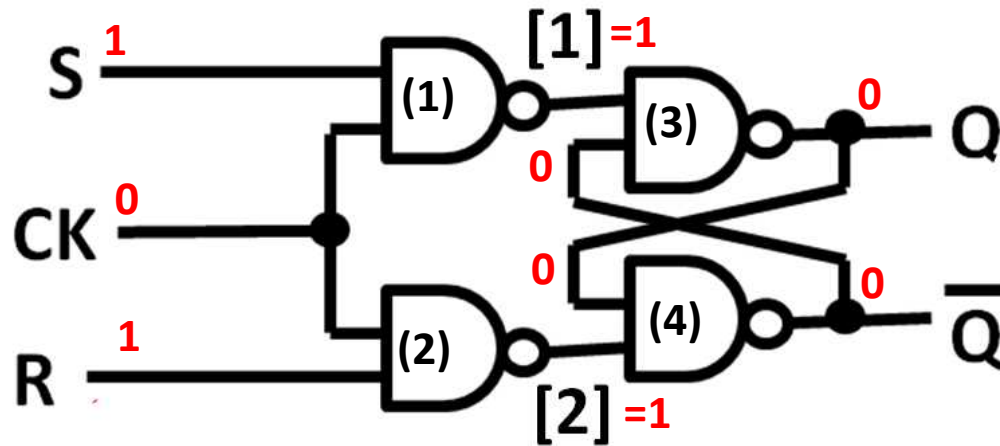
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10179.5	0	0	1	0	1	0	1	1	0
10179.6	0	0	1	0	1	0	1	1	1
10180.0	0	1	1	0	1	0	1	1	1
10181.0	1	1	1	0	1	0	1	1	1
10181.1	1	1	1	0	1	0	1	0	0
10181.2	1	1	1	1	1	1	1	0	0
10181.5	0	1	1	1	1	1	1	0	0
10181.6	0	1	1	1	1	1	1	1	1
10181.7	0	1	1	0	0	0	0	1	1
10181.9	0	1	1	1	1	1	1	1	1
10182.1	0	1	1	0	0	0	0	1	1
10182.3	0	1	1	1	1	1	1	1	1

Q=invQ=1 となり、以後 (1,1) と(0,0) を繰り返し発振状態となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0	0
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

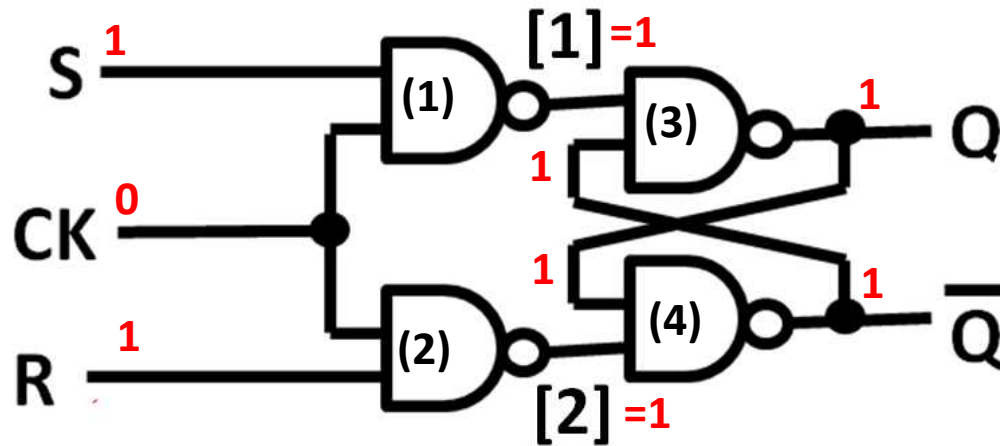
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10179.5	0	0	1	0	1	0	1	1	0
10179.6	0	0	1	0	1	0	1	1	1
10180.0	0	1	1	0	1	0	1	1	1
10181.0	1	1	1	0	1	0	1	1	1
10181.1	1	1	1	0	1	0	1	0	0
10181.2	1	1	1	1	1	1	1	0	0
10181.5	0	1	1	1	1	1	1	0	0
10181.6	0	1	1	1	1	1	1	1	1
10181.7	0	1	1	0	0	0	0	1	1
10181.9	0	1	1	1	1	1	1	1	1
10182.1	0	1	1	0	0	0	0	1	1
10182.3	0	1	1	1	1	1	1	1	1

Q=invQ=0 となり、以後
(1,1) と(0,0) を繰り返し
発振状態となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	1	1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

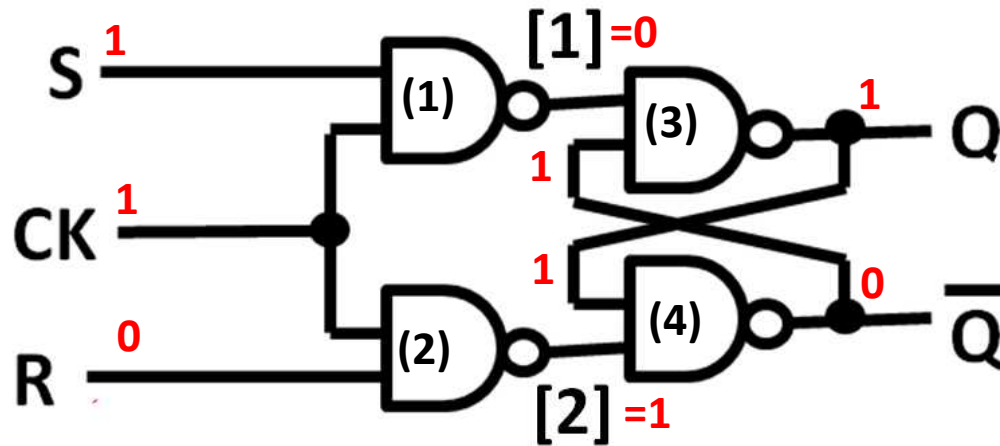
$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10179.5	0	0	1	0	1	0	1	1	0
10179.6	0	0	1	0	1	0	1	1	1
10180.0	0	1	1	0	1	0	1	1	1
10181.0	1	1	1	0	1	0	1	1	1
10181.1	1	1	1	0	1	0	1	0	0
10181.2	1	1	1	1	1	1	1	0	0
10181.5	0	1	1	1	1	1	1	0	0
10181.6	0	1	1	1	1	1	1	1	1
10181.7	0	1	1	0	0	0	0	1	1
10181.9	0	1	1	1	1	1	1	1	1
10182.1	0	1	1	0	0	0	0	1	1
10182.3	0	1	1	1	1	1	1	1	1

Q=invQ=1 となり、以後
(1,1) と(0,0) を繰り返し
発振状態となります。

NAND型Flip-Flop回路 RSFF()の動作解析

計算結果 (outRSFF_1.txt) の説明



禁止
入力

R	S	CK	Q	\bar{Q}
0	0	⌘	Q_0	\bar{Q}_0
0	1	⌘	1	0
1	0	⌘	0	1
1	1	⌘	0/1	0/1
-	-	⌘ 以外	Q_0	\bar{Q}_0

$\Delta t = 0.3$ 以上で RSFF() は動作する。

$\Delta t = 0.4$ の場合

time	CK	S	R	Q	invQ	QQ	invQQ	[1]	[2]
10194.7	0	0	1	1	1	1	1	1	1
10194.9	0	0	1	0	0	0	0	1	1
10195.0	1	0	1	0	0	0	0	1	1
10195.1	1	0	1	1	1	1	1	1	0
10195.3	1	0	1	0	1	0	1	1	0
10195.5	0	0	1	0	1	0	1	1	0
10195.6	0	0	1	0	1	0	1	1	1
10197.0	1	0	1	0	1	0	1	1	1
10197.1	1	0	1	0	1	0	1	1	0
10197.5	0	0	1	0	1	0	1	1	0
10197.6	0	0	1	0	1	0	1	1	1
10198.0	0	0	0	0	1	0	1	1	1

再び R=1 となり、その後 CK=1 となり、Q=0; invQ=1 となり、正常状態に戻ります。