Multicycle && Half-cycle && False paths

2022年6月9日　14:22

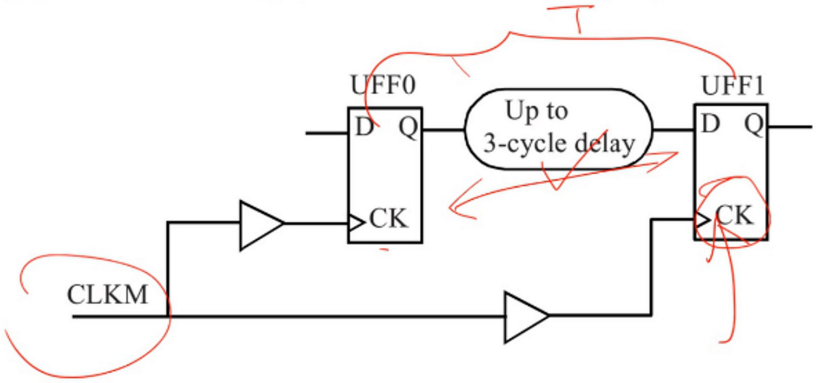
Multicycle && Half-cycle && False paths

Multicycle paths

由于max delay超过了一个周期从而产生多周期的时序路径，对于多周期时序路径，需要设置
多周期的时序约束，而不是沿用原来的单周期时序约束

Multicycle Paths

In some cases, the combinational data path between two flip-flops can take more than one clock cycle to propagate through the logic. In such cases, the combinational path is declared as a multicycle path.
Even though the data is being captured by the capture flip-flop on every clock edge, we direct STA that the relevant capture edge occurs after the specified number of clock cycles.



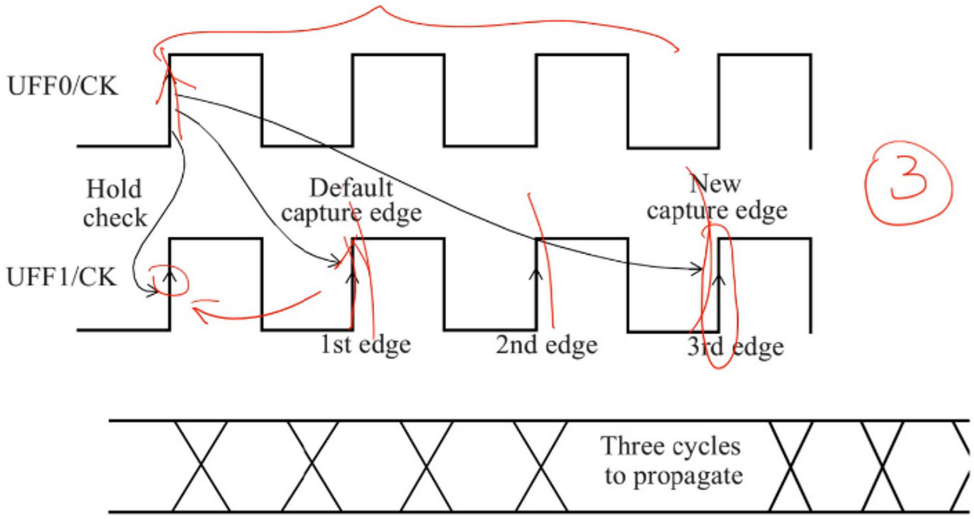
设置setup check的时间点在时钟的3周期之后
这里指定了get_pins

Multicycle Paths

Since the data path can take up to three clock cycles, a setup multicycle check of three cycles should be specified.
The multicycle setup constraints specified to achieve this are given below:

```
create_clock -name CLKM -period 10 [get_ports CLKM]
set_multicycle_path 3 -setup \
  -from [get_pins UFF0/Q] \
  -to [get_pins UFF1/D]
```

Multicycle Paths



setup check at 3T

Multicycle Paths

```
Startpoint: UFF0 (rising edge-triggered flip-flop clocked by CLKM)
  Endpoint: UFF1 (rising edge-triggered flip-flop clocked by CLKM)
  Path Group: CLKM
  Path Type: max

  Point                                             Incr       Path
  ------------------------------------------------------------------
  clock CLKM (rise edge)                           0.00       0.00
  clock network delay (propagated)                 0.11       0.11
  UFF0/CK (DFF )                                   0.00       0.11 r
  UFF0/Q (DFF ) <-                                 0.14       0.26 f
  UNOR0/ZN (NR2  )                                 0.04       0.30 r
  UBUF4/Z (BUFF  )                                 0.05       0.35 r
  UFF1/D (DFF )                                    0.00       0.35 r
  data arrival time                                           0.35
```

Multicycle Paths

```
clock CLKM (rise edge)                          30.00      30.00
clock network delay (propagated)                 0.12      30.12
clock uncertainty                               -0.30      29.82
UFF1/CK (DFF )                                             29.82 r
library setup time                              -0.04      29.78
data required time                                         29.78
-----------------------------------------------------------------
data required time                                         29.78
data arrival time                                          -0.35
-----------------------------------------------------------------
slack (MET)                                                29.43
```

Notice that the clock edge for the capture flip-flop is now three clock cycles away, at 30ns.

由于max delay超过了1个周期从而产生多周期的时序路径，对于多周期时序路径，需要设置多周期的时序约束，而不是沿用原来的单周期时序约束，在这种情况下，
如果指定capture edge的时间点是3个周期之后进行setup check，使max delay小于3个周期，从而满足setup time的要求；那么默认在capture edge上的hold check的时间点应该是（3-1）=2个周期，这意味着，min delay至少需要大于2个周期才能满足hold time的要求，这个情况过于严苛，但实际上，即便max delay被认为超过了1个周期（这里是3个周期之后），min delay可能不会大于2个周期，对于这种情况，我们必须将capture edge上的hold check的时间点往前推2个周期，即默认值2-前推值2 = 0

```
set_multicycle_path 2 -hold -from [get_pins UFF0/Q] \
-to [get_pins UFF1/D]
```

这也意味着，如果不指定

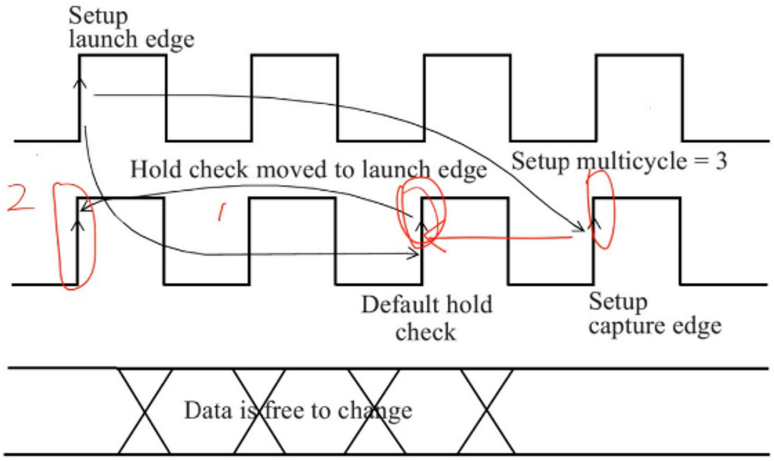
```
set_multicycle_path 2 -hold -from [get_pins UFF0/Q] \
-to [get_pins UFF1/D]
```

那么会取capture edge默认hold check的时间点（3-1）=2个周期

Multicycle Paths

In most designs, a multicycle setup specified as N (cycles) should be accompanied by a multicycle hold constraint specified as N-1 (cycles).

```
set_multicycle_path 2 -hold -from [get_pins UFF0/Q] \
  -to [get_pins UFF1/D]
```



hold check by moving 0

Multicycle Paths

```
Startpoint: UFF0 (rising edge-triggered flip-flop clocked by CLKP)
  Endpoint: UFF1 (rising edge-triggered flip-flop clocked by CLKP)

Path Group: CLKP
Path Type: min

Point                                              Incr       Path
-----------------------------------------------------------------------
clock CLKP (rise edge)                             0.00       0.00
clock source latency                               0.00       0.00
CLKP (in)                                          0.00       0.00 r
UCKBUF4/C (CKB  )                                  0.07       0.07 r
UFF0/CK (DFF )                                     0.00       0.07 r
UFF0/Q (DFF ) <-                                   0.15       0.22 f
UXOR1/Z (XOR2  )                                   0.07       0.29 f
UFF1/D (DFF )                                      0.00       0.29 f
data arrival time                                             0.29
```

Multicycle Paths

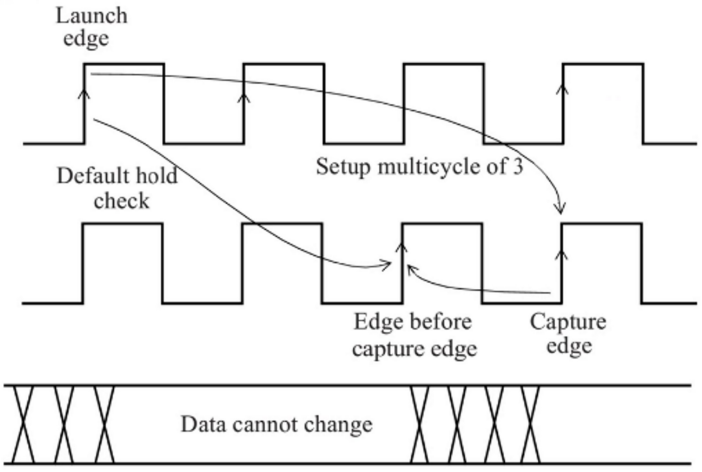
2 - hold

clock CLKP (rise edge)	**0.00**	0.00
clock source latency	0.00	0.00
CLKP (in)	0.00	0.00 r
UCKBUF4/C (CKB)	0.07	0.07 r
UCKBUF5/C (CKB)	0.06	0.13 r
UFF1/CK (DFF)	0.00	0.13 r
clock uncertainty	0.05	0.18
library **hold** time	0.01	0.19
data required time		0.19
data required time		0.19
data arrival time		-0.29
slack (MET)		0.11

Multicycle Paths

What happens when a multicycle setup of N is specified but the corresponding N-1 multicycle hold is missing?

In such a case, the hold check is performed on the edge one cycle prior to the setup capture edge.



hold check by default 2T if setup check at 3T

Multicycle Paths

```
Startpoint: UFF0 (rising edge-triggered flip-flop clocked by CLKM)
  Endpoint: UFF1 (rising edge-triggered flip-flop clocked by CLKM)
  Path Group: CLKM
  Path Type: min

  Point                                              Incr       Path
  -----------------------------------------------------------------
  clock CLKM (rise edge)                             0.00       0.00
  clock source latency                               0.00       0.00
  CLKM (in)                                          0.00       0.00 r
  UCKBUF0/C (CKB  )                                  0.06       0.06 r
  UCKBUF1/C (CKB  )                                  0.06       0.11 r
  UFF0/CK (DFF )                                     0.00       0.11 r
  UFF0/Q (DFF ) <-                                   0.14       0.26 r
  UNOR0/ZN (NR2  )                                   0.02       0.28 f
  UBUF4/Z (BUFF  )                                   0.06       0.33 f
  UFF1/D (DFF )                                      0.00       0.33 f
  data arrival time                                             0.33
```

N -setup
N-1 -hold

Multicycle Paths

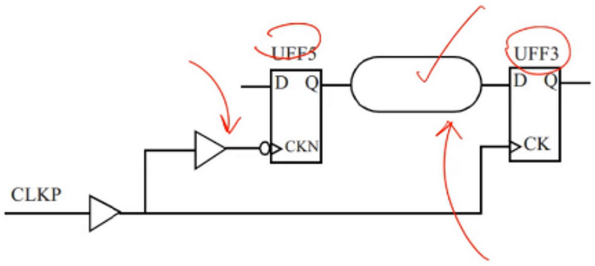
```
  clock CLKM (rise edge)                            20.00      20.00
  clock source latency                               0.00      20.00
  CLKM (in)                                          0.00      20.00 r
  UCKBUF0/C (CKB  )                                  0.06      20.06 r
  UCKBUF2/C (CKB  )                                  0.07      20.12 r
  UFF1/CK (DFF )                                     0.00      20.12 r
  clock uncertainty                                  0.05      20.17
  library hold time                                  0.01      20.19
  data required time                                           20.19
  -----------------------------------------------------------------
  data required time                                           20.19
  data arrival time                                            -0.33
  -----------------------------------------------------------------
  slack (VIOLATED)                                             -19.85
```

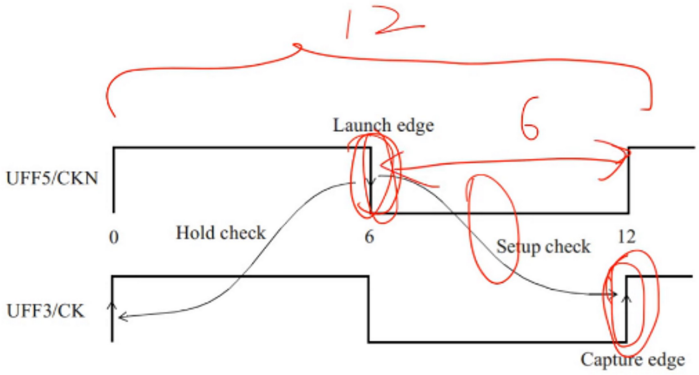
N-setup N-1-hold

Half-cycle paths

Half-Cycle Paths

If a design has both negative-edge triggered flip-flops (active clock edge isfalling edge) and positive-edge triggered flip-flops (active clock edge is rising edge), it is likely that half-cycle paths exist in the design.





过了6，或者说等了半个周期才开始发送数据，使data arrive time大大延后了，这样也许会导致setup time不满足的情况
falling edge triggered is different

setup check under half-cycle paths

Half-Cycle Paths

```
Startpoint: UFF5 (falling edge-triggered flip-flop clocked by CLKP)
  Endpoint: UFF3 (rising edge-triggered flip-flop clocked by CLKP)
  Path Group: CLKP
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock CLKP (fall edge)                   6.00       6.00
  clock source latency                     0.00       6.00
  CLKP (in)                                0.00       6.00 f
  UCKBUF4/C (CKB  )                        0.06       6.06 f
  UCKBUF6/C (CKB  )                        0.06       6.12 f
  UFF5/CKN (DFN  )                         0.00       6.12 f
  UFF5/Q (DFN  ) <-                        0.16       6.28 r
  UNAND0/ZN (ND2  )                        0.03       6.31 f
  UFF3/D (DFF )                            0.00       6.31 f
  data arrival time                                   6.31
```

Half-Cycle Paths

clock CLKP (rise edge)	**12.00**	12.00	
clock source latency	0.00	12.00	
CLKP (in)	0.00	12.00	**r**
UCKBUF4/C (CKB)	0.07	12.07	r
UFF3/CK (DFF)	0.00	12.07	r
clock uncertainty	-0.30	11.77	
library **setup** time	-0.03	11.74	
data required time		11.74	
data required time		11.74	
data arrival time		-6.31	
slack (MET)		5.43	

Note the edge specification in the Startpoint and Endpoint.

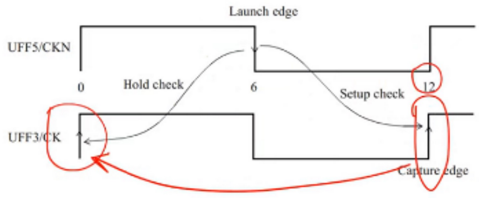
The falling edge occurs at 6ns and the rising edge occurs at 12ns. Thus, the data gets only a half-cycle, which is 6ns, to propagate to the capture flip-flop.

对于hold check，半周期路径会使数据等了半个周期才发送，但是hold check是在0，
launch是在6，因此hold time可以满足

hold check under half-cycle paths

Half-Cycle Paths

While the data path gets only half-cycle for setup check, an extra half-cycle is available for the hold timing check. Here is the hold timing path.



Startpoint: UFF5 (**falling edge-triggered** flip-flop clocked by CLKP)
Endpoint: UFF3 (**rising edge-triggered** flip-flop clocked by CLKP)
Path Group: CLKP
Path Type: **min**

Point	Incr	Path
clock CLKP (fall edge)	**6.00**	6.00
clock source latency	0.00	6.00
CLKP (in)	0.00	6.00 f
UCKBUF4/C (CKB)	0.06	6.06 f
UCKBUF6/C (CKB)	0.06	6.12 f
UFF5/CKN (DFN)	0.00	6.12 f
UFF5/Q (DFN) <-	0.16	6.28 r
UNAND0/ZN (ND2)	0.03	6.31 f
UFF3/D (DFF)	0.00	6.31 f
data arrival time		6.31

两个芯片互跨，使用半周期让hold time更容易满足；setup很好修，但是hold难修

Half-Cycle Paths

clock CLKP (rise edge)	0.00	0.00
clock source latency	0.00	0.00
CLKP (in)	0.00	0.00 r
UCKBUF4/C (CKB)	0.07	0.07 r
UFF3/CK (DFF)	0.00	0.07 r
clock uncertainty	0.05	0.12
library **hold** time	0.02	0.13
data required time		0.13
data required time		0.13
data arrival time		-6.31
slack (MET)		6.18

The hold check always occurs one cycle prior to the capture edge. Since the capture edge occurs at 12ns, the previous capture edge is at 0ns, and hence the hold gets checked at 0ns.

This effectively adds a half-cycle margin for hold checking and thus results in a large positive slack on hold.

False paths

False Paths

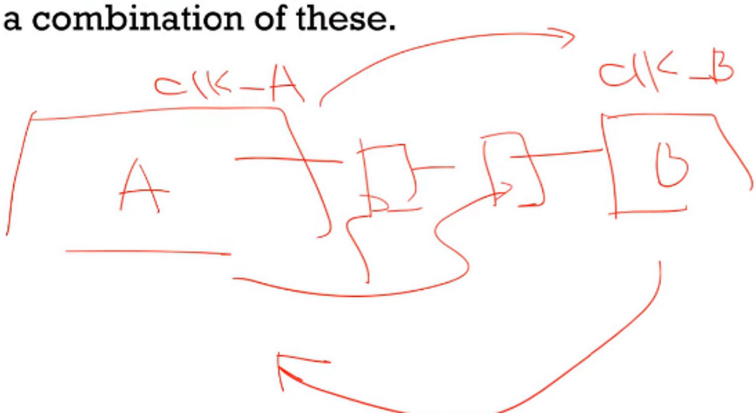
It is possible that certain timing paths are not real (or not possible) in the actual functional operation of the design.

Such paths can be turned off during STA by setting these as false paths.

A false path is ignored by the STA for analysis.

False Paths

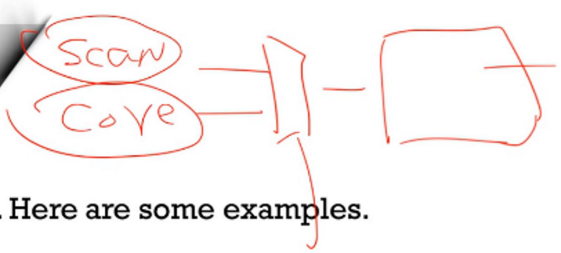
Examples of false paths could be from one clock domain to another clock domain, from a clock pin of a flip-flop to the input of another flip-flop, through a pin of a cell, through pins of multiple cells, or a combination of these.



False Paths

- When a false path is specified through a pin of a cell, all paths that go through that pin are ignored for timing analysis.
- The advantage of identifying the false paths is that the analysis space is reduced, thereby allowing the analysis to focus only on the real paths.
- This helps cut down the analysis time as well.
- However, too many false paths using the through specification can slow down the analysis.

False Paths



A false path is set using the set_false_path specification. Here are some examples.

```
set_false_path -from [get_clocks SCAN_CLK] \
  -to [get_clocks CORE_CLK]
# Any path starting from the SCAN_CLK domain to the
# CORE_CLK domain is a false path.

set_false_path -through [get_pins UMUX0/S]
# Any path going through this pin is false.
```

False Paths

A false path is set using the set_false_path specification. Here are some examples.

```
set_false_path \
  -through [get_pins SAD_CORE/RSTN]]
# The false path specifications can also be specified to,
# through, or from a module pin instance.

set_false_path -to [get_ports TEST_REG*]
# All paths that end in port named TEST_REG* are false paths.

set_false_path -through UINV/Z -through UAND0/Z
# Any path that goes through both of these pins
# in this order is false.
```

定义从时钟A到时钟B的伪路径

False Paths

Few recommendations on setting false paths are given below. To set a false path between two clock domains, use:

```
set_false_path -from [get_clocks clockA] \
 -to [get_clocks clockB]
```

instead of:

```
set_false_path -from [get_pins {regA_*}/CP] \
 -to [get_pins {regB_*}/D]
```

The second form is much slower.

少用through的options

False Paths

- Another recommendation is to minimize the usage of -through options, as it adds unnecessary runtime complexity.
- The -through option should only be used where it is absolutely necessary and there is no alternate way to specify the false path.

如果是真实的虚假路径，就要加入虚假路径约束

False Paths

From an optimization perspective, another guideline is to not use a false path when a multicycle path is the real intent.

If a signal is sampled at a known or predictable time, no matter how far out, a multicycle path specification should be used so that the path has some constraint and gets optimized to meet the multicycle constraint.

If a false path is used on a path that is sampled many clock cycles later, optimization of the remaining logic may invariably slow this path even beyond what may be necessary.