

数字集成电路静态时序分析基础

邸志雄 博士, zxdi@home.swjtu.edu.cn

西南交通大学信息科学与技术学院

CONTENT

Part 6: Robust Verification

On-Chip Variations

1. Setup time check
2. Hold time check

On-Chip Variations

In general, the process and environmental parameters may not be uniform across different portions of the die.

Due to process variations, identical MOS transistors in different portions of the die may not have similar characteristics. These differences are due to process variations within the die.

On-Chip Variations

These differences can arise due to many factors, including:

- ❑ i. IR drop variation along the die area affecting the local power supply.
- ❑ ii. Voltage threshold variation of the PMOS or the NMOS device.
- ❑ iii. Channel length variation of the PMOS or the NMOS device.
- ❑ iv. Temperature variations due to local hot spots.
- ❑ v. Interconnect metal etch or thickness variations impacting the interconnect resistance or capacitance.

The PVT variations described above are referred to as **On-Chip Variations (OCV)** and these variations can affect the wire delays and cell delays in different portions of the chip.

On-Chip Variations

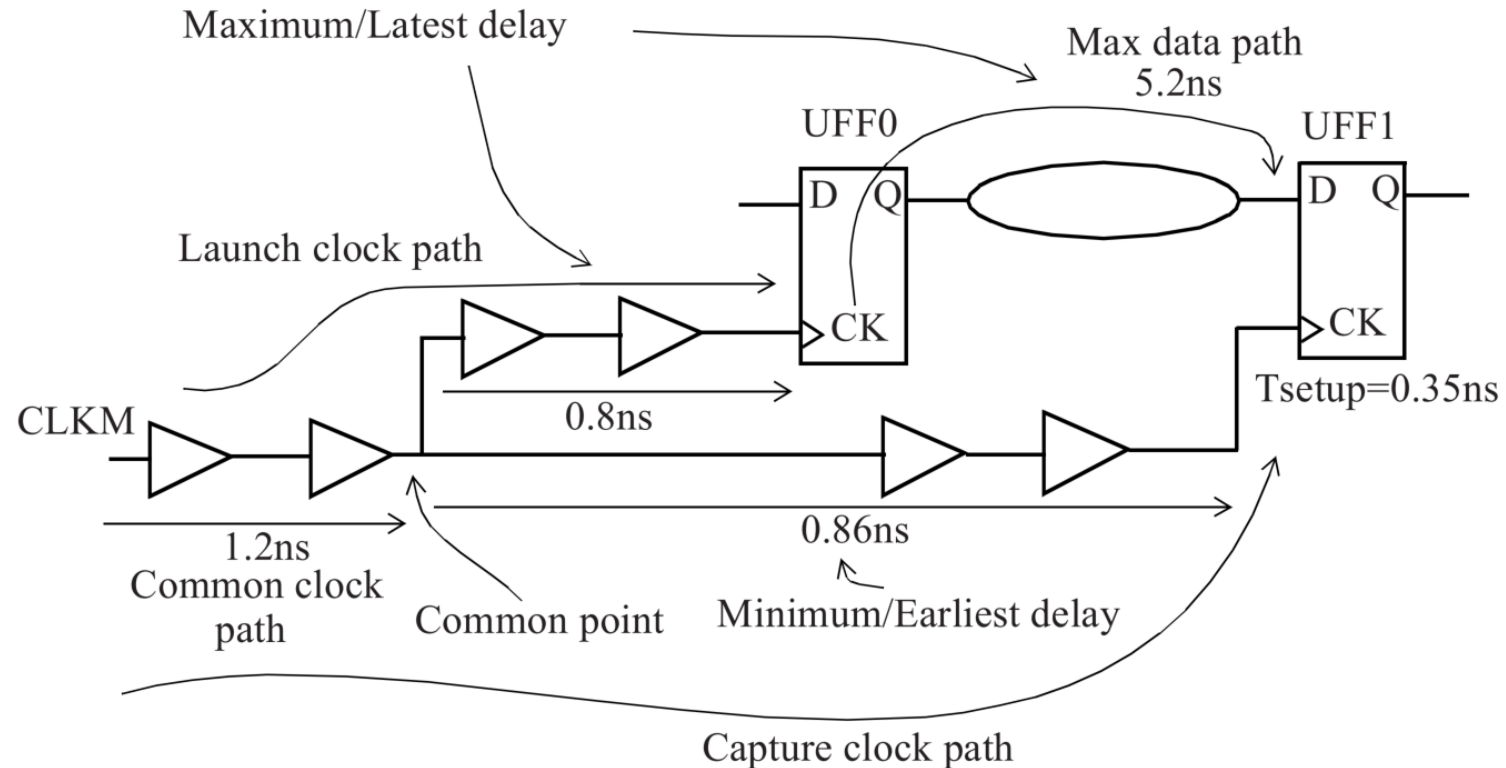
Since the clock and data paths can be affected differently by the OCV, the timing verification can model the OCV effect by making the PVT conditions for the launch and capture paths to be slightly different.

The STA can include the OCV effect *by derating the delays of specific paths*, that is, by making those paths faster or slower and then validating the behavior of the design with these variations.

The cell delays or wire delays or both can be derated to model the effect of OCV.

On-Chip Variations

We now examine how the OCV derating is done for a setup check.



The worst condition for setup check occurs when the launch clock path and the data path have the OCV conditions which result in the largest delays, while the capture clock path has the OCV conditions which result in the smallest delays.

On-Chip Variations

For this example, here is the setup timing check condition; this does not include any OCV setting for derating delays.

$$\text{LaunchClockPath} + \text{MaxDataPath} \leq \text{ClockPeriod} + \text{CaptureClockPath} - T_{\text{setup_UFF1}}$$

This implies that the minimum clock period = $\text{LaunchClockPath} + \text{MaxDataPath} - \text{CaptureClockPath} + T_{\text{setup_UFF1}}$

On-Chip Variations

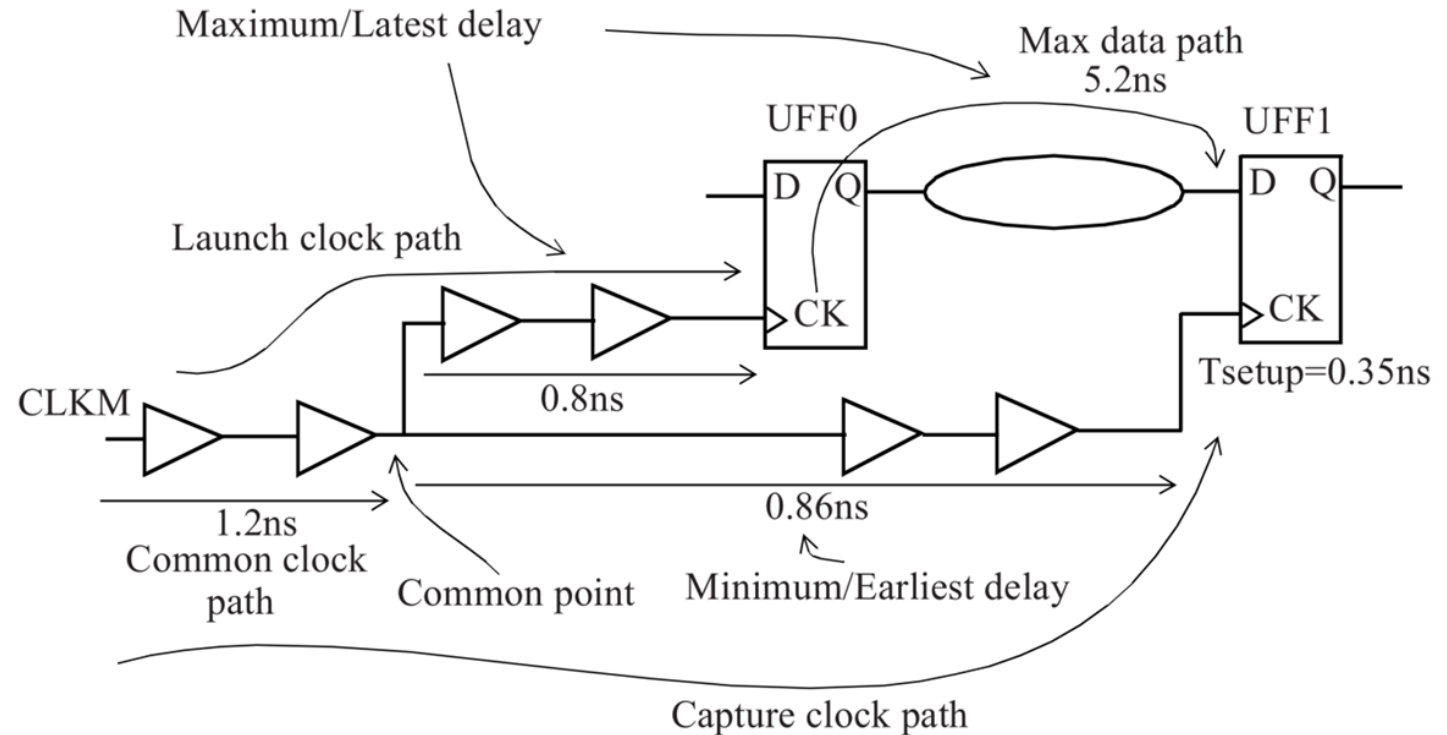
From the figure,

$$\text{LaunchClockPath} = 1.2 + 0.8 = 2.0$$

$$\text{MaxDataPath} = 5.2$$

$$\text{CaptureClockPath} = 1.2 + 0.86 = 2.06$$

$$T_{\text{setup_UFF1}} = 0.35$$



This results in a minimum clock period of:

$$2.0 + 5.2 - 2.06 + 0.35 = 5.49\text{ns}$$

On-Chip Variations

The above path delays correspond to the delay values without any OCV derating. Cell and net delays can be derated using the `set_timing_derate` specification.

For example, the commands:

```
set_timing_derate -early 0.8
```

```
set_timing_derate -late 1.1
```

On-Chip Variations

Derate the minimum/shortest/early paths by -20% and derate the maximum/longest/latest paths by +10%.

- ❑ Long path delays (for example, data paths and launch clock path for setup checks or capture clock paths for hold checks) are multiplied by the derate value specified using the -late option.
- ❑ Short path delays (for example, capture clock paths for setup checks or data paths and launch clock paths for hold checks) are multiplied by the derate values specified using the -early option.

If no derating factors are specified, a value of 1.0 is assumed

On-Chip Variations

The derating factors apply uniformly to all net delays and cell delays.

If an application scenario warrants different derating factors for cells and nets, the `-cell_delay` and the `-net_delay` options can be used in the `set_timing_derate` specification.

```
# Derate only the cell delays - early paths by -10%, and  
# no derate on the late paths:  
set_timing_derate -cell_delay -early 0.9  
set_timing_derate -cell_delay -late 1.0
```

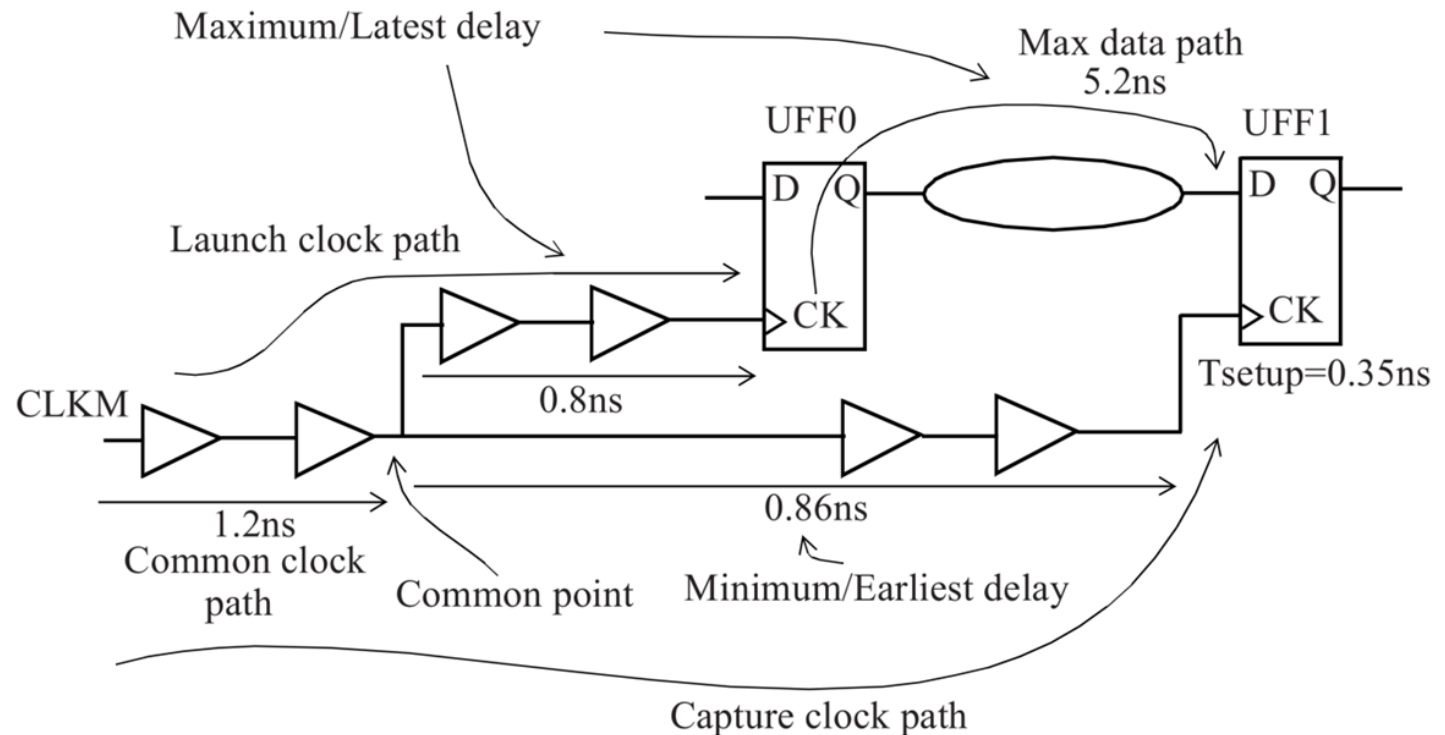
On-Chip Variations

We now apply the following derating to the example.

```
set_timing_derate -early 0.9
```

```
set_timing_derate -late 1.2
```

```
set_timing_derate -late 1.1 -cell_check
```



On-Chip Variations

With these derating values, we get the following for setup check:

$$\text{LaunchClockPath} = 2.0 * 1.2 = 2.4$$

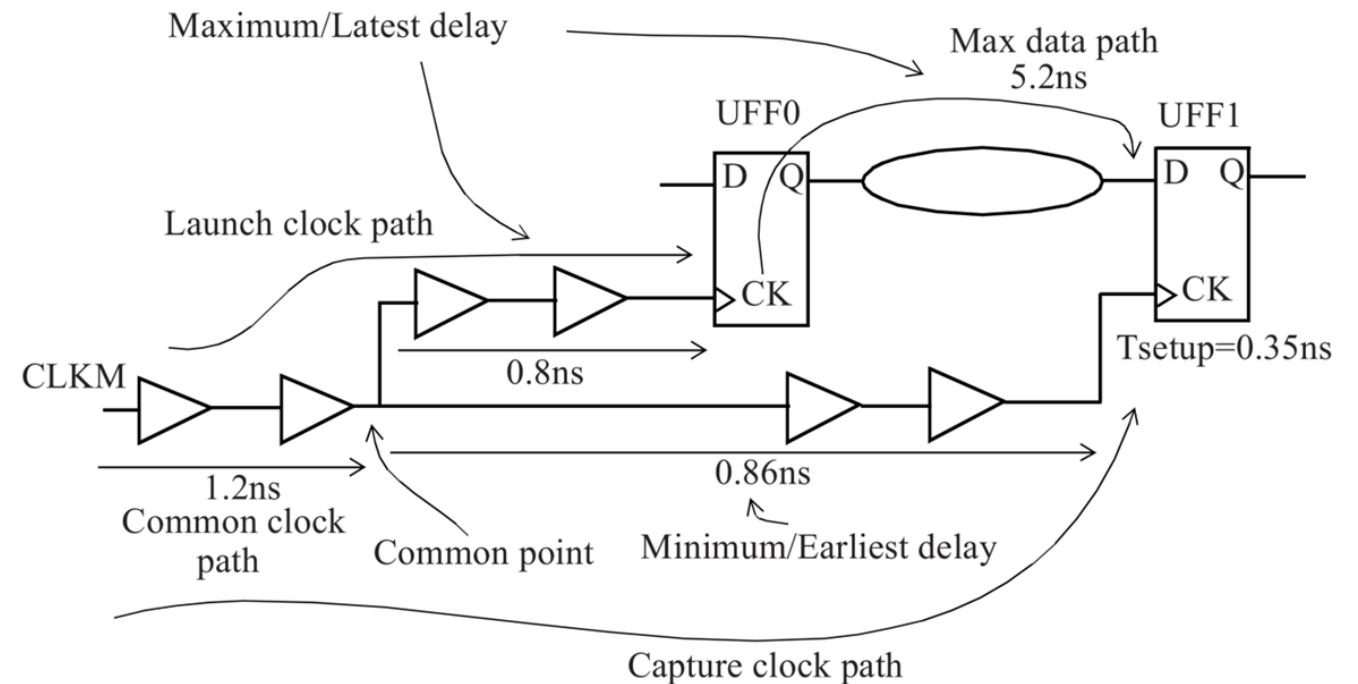
$$\text{MaxDataPath} = 5.2 * 1.2 = 6.24$$

$$\text{CaptureClockPath} = 2.06 * 0.9 = 1.854$$

$$\text{Tsetup_UFF1} = 0.35 * 1.1 = 0.385$$

This results in a minimum clock period of:

$$2.4 + 6.24 - 1.854 + 0.385 = 7.171\text{ns}$$



On-Chip Variations

In the setup check above, there is a discrepancy since the common clock path of the clock tree, with a delay of 1.2ns, is being derated differently for the launch clock and for the capture clock.

This part of the clock tree is common to both the launch clock and the capture clock and should not be derated differently.

On-Chip Variations

Applying different derating for the launch and capture clock is overly pessimistic as in reality this part of the clock tree will really be at only one PVT condition, either as a maximum path or as a minimum path (or anything in between) but never both at the same time.

On-Chip Variations

The pessimism caused by different derating factors applied on the common part of the clock tree is called Common Path Pessimism (CPP) which should be removed during the analysis.

CPPR, which stands for Common Path Pessimism Removal, is often listed as a separate item in a path report. It is also labeled as Clock Reconvergence Pessimism Removal (CRPR).

On-Chip Variations

CPPR is the removal of artificially induced pessimism between the launch clock path and the capture clock path in timing analysis.

If the same clock drives both the capture and the launch flip-flops, then the clock tree will likely share a common portion before branching.

On-Chip Variations

The common point is defined as the output pin of the last cell in the common portion of the clock tree.

$$\text{CPP} = \text{LatestArrivalTime@CommonPoint} - \text{EarliestArrivalTime@CommonPoint}$$

On-Chip Variations

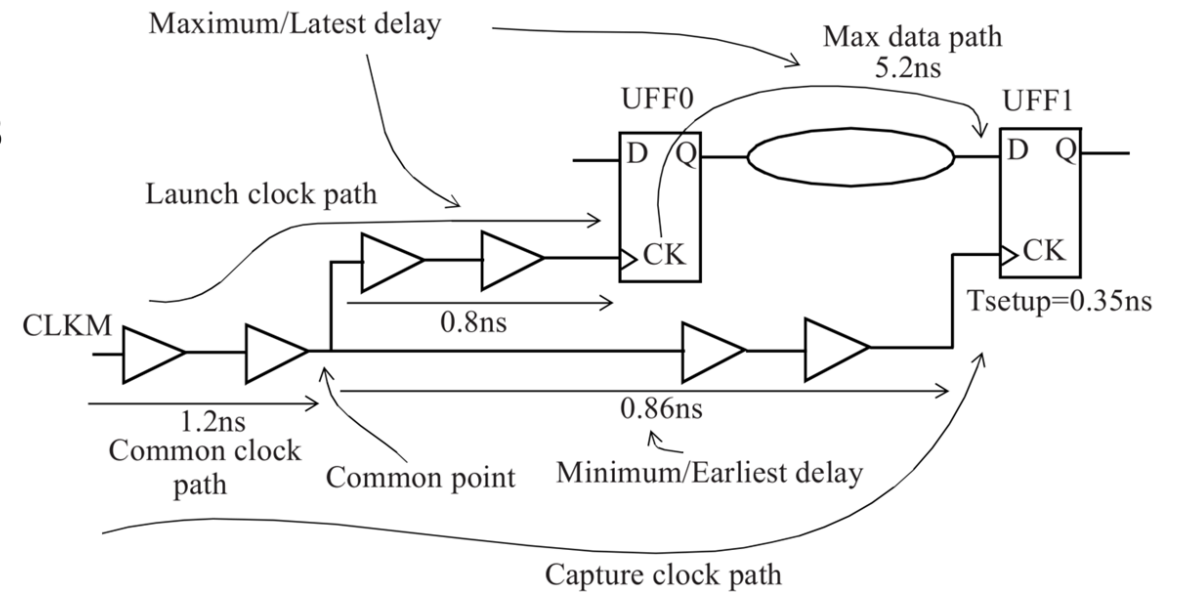
The Latest and Earliest times in the above analysis are in reference to the OCV derating at a specific timing corner - for example worst-case slow or best-case fast.

$$\text{LatestArrivalTime@CommonPoint} = 1.2 * 1.2 = 1.44$$

$$\text{EarliestArrivalTime@CommonPoint} = 1.2 * 0.9 = 1.08$$

This implies a CPP of: $1.44 - 1.08 = 0.36\text{ns}$

With the CPP correction, this results in a minimum clock period of: $7.171 - 0.36 = 6.811\text{ns}$



On-Chip Variations

Applying the OCV derating has increased the minimum clock period from 5.49ns to 6.811ns for this example design.

This illustrates that the OCV variations modeled by these derating factors can reduce the maximum frequency of operation of the design.

On-Chip Variations

Analysis with OCV at Worst PVT Condition

If the setup timing check is being performed at the worst-case PVT condition, no derating is necessary on the late paths as they are already the worst possible.

However, derating can be applied to the early paths by making those paths faster by using a specific derating, for example, speeding up the early paths by 10%.

On-Chip Variations

Analysis with OCV at Worst PVT Condition

A derate specification at the worst-case slow corner may be something like:

```
set_timing_derate -early 0.9
```

```
set_timing_derate -late 1.0
```

```
# Don't derate the late paths as they are already the slowest,
```

```
# but derate the early paths to make these faster by 10%.
```

On-Chip Variations

Analysis with OCV at Worst PVT Condition

```
set_timing_derate -early 0.9
```

```
set_timing_derate -late 1.0
```

The above derate settings are for max path (or setup) checks at the worstcase slow corner; thus the late path OCV derate setting is kept at 1.0 so as not to slow it beyond the worst-case slow corner.

On-Chip Variations

Analysis with OCV at Worst PVT Condition

Here is the setup timing check path report performed at the worst-case slow corner. The derating used by the late paths are reported as Max Data Paths Derating Factor and as Max Clock Paths Derating Factor.

The derating used for the early paths is reported as Min Clock Paths Derating Factor.

On-Chip Variations

Analysis with OCV at Worst PVT Condition

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

Max Data Paths Derating Factor : 1.000

Min Clock Paths Derating Factor : 0.800

Max Clock Paths Derating Factor : 1.000

Point	Incr	Path
-----	-----	-----
clock CLKM (rise edge)	0.000	0.000
clock source latency	0.000	0.000
CLKM (in)	0.000	0.000 r
UCKBUF0/C (CKB)	0.056	0.056 r
UCKBUF1/C (CKB)	0.058	0.114 r
UFF0/CK (DF)	0.000	0.114 r
UFF0/Q (DF) <-	0.143	0.258 f
UNOR0/ZN (NR2)	0.043	0.301 r
UBUF4/Z (BUFF)	0.052	0.352 r
UFF1/D (DF)	0.000	0.352 r
data arrival time		0.352

On-Chip Variations

Analysis with OCV at Worst PVT Condition

clock CLKM (rise edge)	10.000	10.000	
clock source latency	0.000	10.000	
CLKM (in)	0.000	10.000	r
UCKBUF0/C (CKB)	0.045	10.045	r
UCKBUF2/C (CKB)	0.054	10.098	r
UFF1/CK (DF)	0.000	10.098	r
clock reconvergence pessimism	0.011	10.110	
clock uncertainty	-0.300	9.810	
library setup time	-0.044	9.765	
data required time		9.765	

data required time		9.765	
data arrival time		-0.352	

slack (MET)		9.413	

On-Chip Variations

Analysis with OCV at Worst PVT Condition

The cell **UCKBUF0** is on the **common** clock path, that is, on both the capture clock path and the launch clock path.

Since the common clock path cannot have a different derating, the difference in timing for this common path, $56\text{ps} - 45\text{ps} = 11\text{ps}$, is corrected separately.

CONTENT

Part 6: Robust Verification

On-Chip Variations

1. Setup time check
2. Hold time check

On-Chip Variations

OCV for Hold Checks

If the PVT conditions are different along the chip, the worst condition for hold check occurs:

when the launch clock path and the data path have OCV conditions which result in the smallest delays, that is, when we have the earliest launch clock, and the capture clock path has the OCV conditions which result in the largest delays, that is, has the latest capture clock

On-Chip Variations

OCV for Hold Checks

The hold timing check is specified in the following expression for this example.

$$\text{LaunchClockPath} + \text{MinDataPath} - \text{CaptureClockPath} - T_{\text{hold_UFF1}} \geq 0$$

On-Chip Variations

OCV for Hold Checks

Applying the delay values in the Figure 10-2 to the expression, we get (without applying any derating):

$$\text{LaunchClockPath} = 0.25 + 0.6 = 0.85$$

$$\text{MinDataPath} = 1.7$$

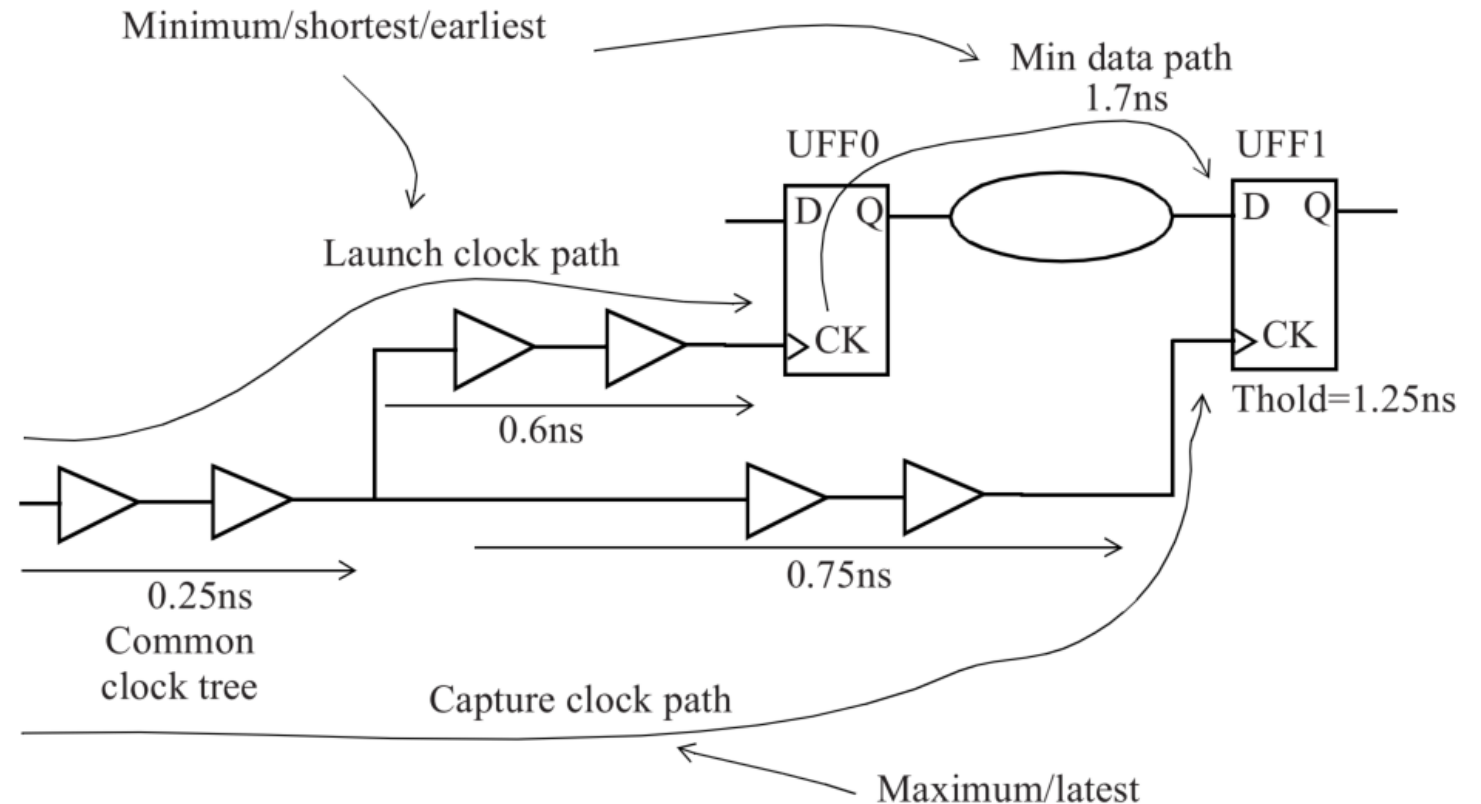
$$\text{CaptureClockPath} = 0.25 + 0.75 = 1.00$$

$$\text{Thold_UFF1} = 1.25$$

This implies that the condition is:

$$0.85 + 1.7 - 1.00 - 1.25 = 0.3\text{ns} \geq 0$$

which is true, and thus no hold violation exists.



On-Chip Variations

OCV for Hold Checks

Applying the following derate specification:

```
set_timing_derate -early 0.9
```

```
set_timing_derate -late 1.2
```

```
set_timing_derate -early 0.95 -cell_check
```


On-Chip Variations

OCV for Hold Checks

$$\text{LaunchClockPath} = 0.85 * 0.9 = 0.765$$

$$\text{MinDataPath} = 1.7 * 0.9 = 1.53$$

$$\text{CaptureClockPath} = 1.00 * 1.2 = 1.2$$

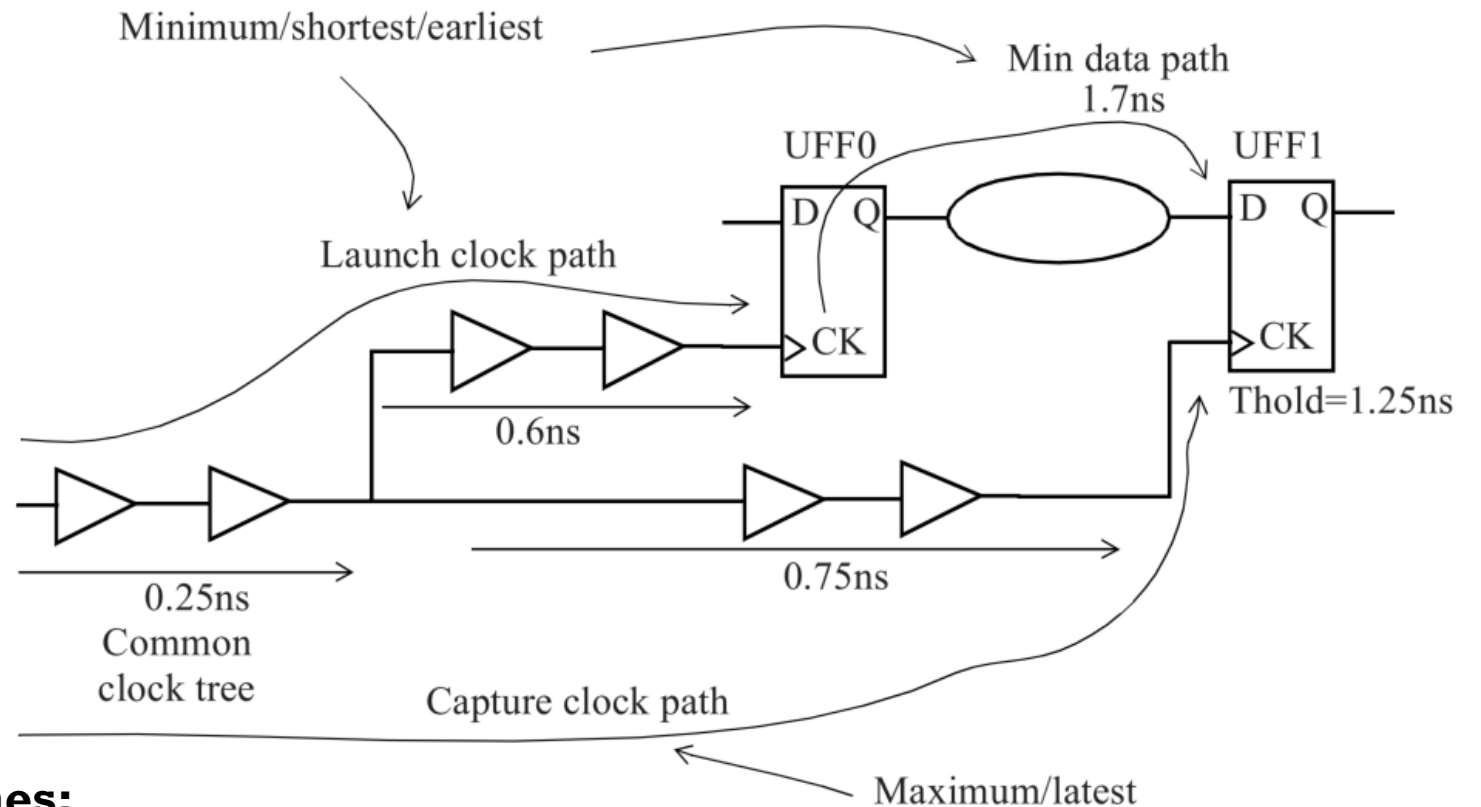
$$\text{Thold_UFF1} = 1.25 * 0.95 = 1.1875$$

Common clock path pessimism:

$$0.25 * (1.2 - 0.9) = 0.075$$

The hold check condition then becomes:

$$0.765 + 1.53 - 1.2 - 1.1875 + 0.075 = -0.0175\text{ns} < 0$$



On-Chip Variations

OCV for Hold Checks

In general, the hold timing check is performed at the best-case fast PVT corner. In such a scenario, no derating is necessary on the early paths, as those paths are already the earliest possible.

However, derating can be applied on the late paths by making these slower by a specific derating factor, for example, slowing the late paths by 20.

On-Chip Variations

OCV for Hold Checks

A derate specification at this corner would be something like:

set_timing_derate -early 1.0

set_timing_derate -late 1.2

Don't derate the early paths as they are already the

fastest, but derate the late paths slower by 20%.

On-Chip Variations

OCV for Hold Checks

In the example of Figure 10-2,

LatestArrivalTime@CommonPoint

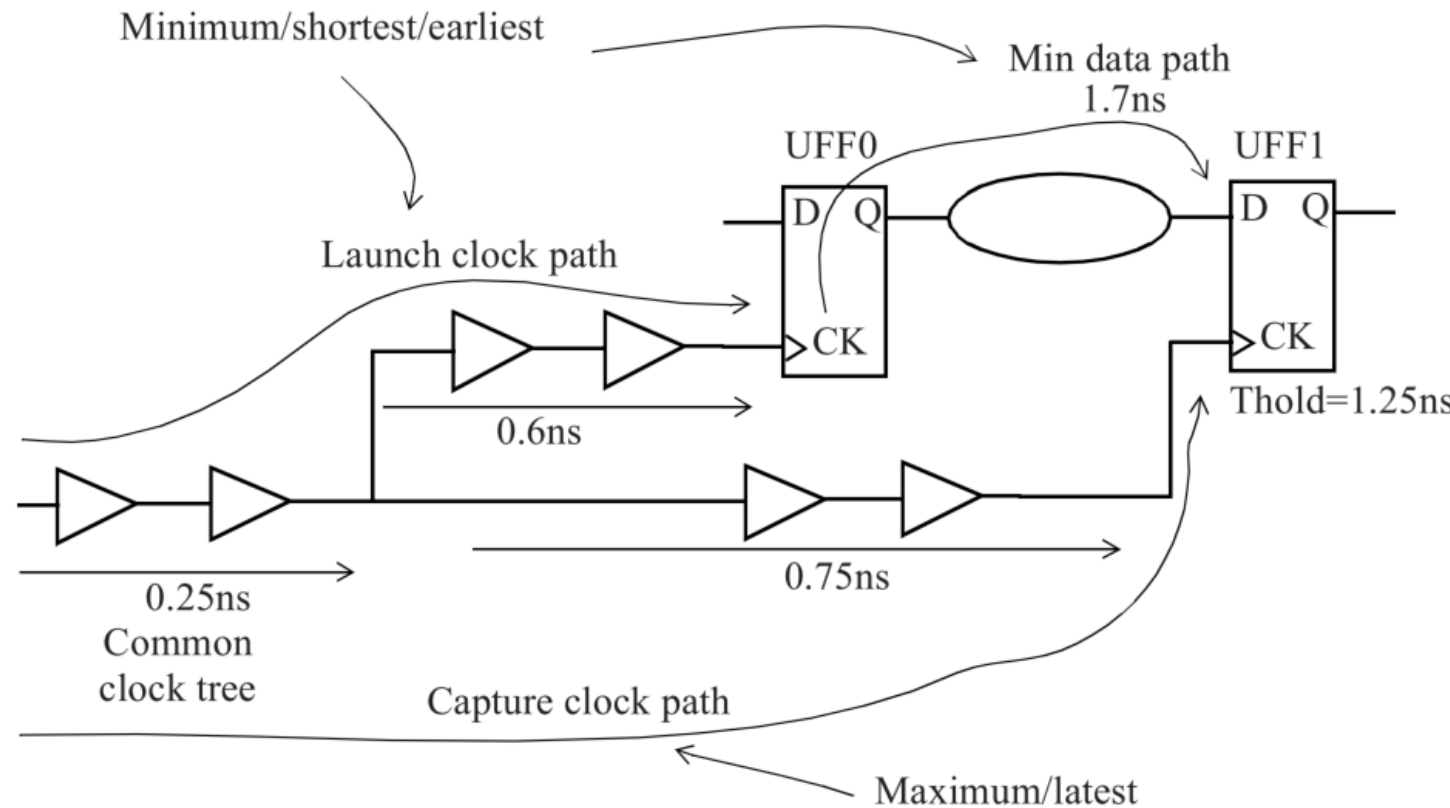
$$= 0.25 * 1.2 = 0.30$$

EarliestArrivalTime@CommonPoint

$$= 0.25 * 1.0 = 0.25$$

This implies a common path pessimism of:

$$0.30 - 0.25 = 0.05\text{ns}$$



On-Chip Variations

OCV for Hold Checks

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

Min Data Paths Derating Factor : 1.000

Min Clock Paths Derating Factor : 1.000

Max Clock Paths Derating Factor : 1.200

Point	Incr	Path
-----	-----	-----
clock CLKM (rise edge)	0.000	0.000
clock source latency	0.000	0.000
CLKM (in)	0.000	0.000 r
UCKBUF0/C (CKB)	0.056	0.056 r
UCKBUF1/C (CKB)	0.058	0.114 r
UFF0/CK (DF)	0.000	0.114 r
UFF0/Q (DF) <-	0.144	0.258 r
UNOR0/ZN (NR2)	0.021	0.279 f
UBUF4/Z (BUFF)	0.055	0.334 f
UFF1/D (DF)	0.000	0.334 f
data arrival time		0.334

On-Chip Variations

OCV for Hold Checks

clock CLKM (rise edge)	0.000	0.000	
clock source latency	0.000	0.000	
CLKM (in)	0.000	0.000	r
UCKBUF0/C (CKB)	0.067	0.067	r
UCKBUF2/C (CKB)	0.080	0.148	r
UFF1/CK (DF)	0.000	0.148	r
clock reconvergence pessimism	-0.011	0.136	
clock uncertainty	0.050	0.186	
library hold time	0.015	0.201	
data required time		0.201	

data required time		0.201	
data arrival time		-0.334	

slack (MET)		0.133	

On-Chip Variations

OCV for Hold Checks

Notice that the late paths are derated by +20% while the early paths are not derated. See cell UCKBUF0.

Its delay on the launch path is 56ps while the delay on the capture path is 67ps - derated by +20%.

UCKBUF0 is the cell on the common clock tree and thus the pessimism introduced due to different derating on this common clock tree is, $67\text{ps} - 56\text{ps} = 11\text{ps}$, which is accounted for separately on the line clock reconvergence pessimism.

参考书目

- Static Timing Analysis for Nanometer Designs: A Practical Approach. J. Bhasker, Rakesh Chadha. Springer Science Business Media, LLC 2009.
- 集成电路静态时序分析与建模. 刘峰, 机械工业出版社. 出版时间: 2016-07-01.



谢谢聆听！

个人教学工作主页<https://customizablecomputinglab.github.io/>