Vivado从此开始（To Learn Vivado From Here）

Vivado
从此开始
To Learn Vivado From Here
高亚军 编著
· 内容翔实全面：涵盖Vivado所有基本功能
· 讲解深入浅出：结合大量案例，帮助读者加强对基本概念的理解
· 描述图文并茂：给出具体操作步骤，易于快速动手实践
中国工信出版集团
電子工業出版社

本书围绕Vivado四大主题

- 设计流程
- 时序约束
- 时序分析
- Tcl脚本的使用

作者：高亚军（Xilinx战略应用高级工程师）

- 2012年2月，出版《基于FPGA的数字信号处理（第1版）》
- 2012年9月，发布网络视频课程《Vivado入门与提高》
- 2015年7月，出版《基于FPGA的数字信号处理（第2版）》
- 2016年7月，发布网络视频课程《跟Xilinx SAE学HLS》

- 内容翔实全面：涵盖Vivado所有基本功能
- 讲解深入浅出：结合大量案例，帮助读者加强对基本概念的理解
- 描述图文并茂：给出具体操作步骤，易于快速动手实践

XILINX

ALL PROGRAMMABLE™

Setting False Path

Lauren Gao

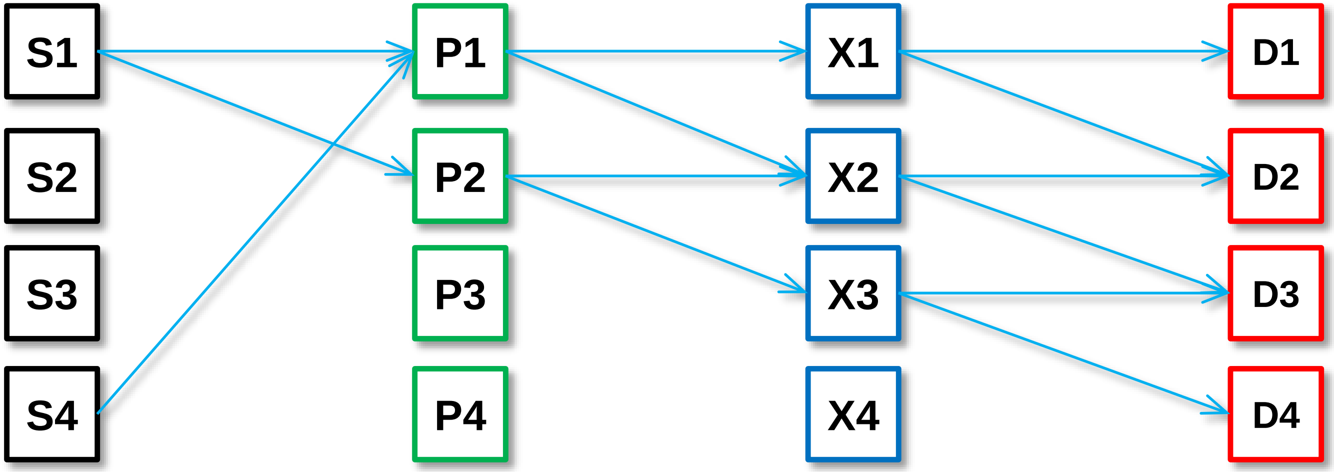
What and Why?

- **What is False Path**
  - A path that does exist in the design but does not play a part in the operation, so it's not necessary to include it in the timing analysis
    - is not functional
    - does not need to be timed
- **Why False Path Exceptions**
  - Remove invalid timing paths
    - Static signals driven by configuration registers
  - Save time and resources
    - Skip false path optimization

set_false_path

- **set_false_path** [**-setup**] [**-hold**] [**-rise**] [**-fall**] [**-reset_path**] [**-from** *args*] [**-rise_from** *args*] [**-fall_from** *args*] [**-to** *args*] [**-rise_to** *args*] [**-fall_to** *args*] [**-through** *args*] [**-rise_through** *args*] [**-fall_through** *args*] [**-quiet**] [**-verbose**]

Path Specification #1



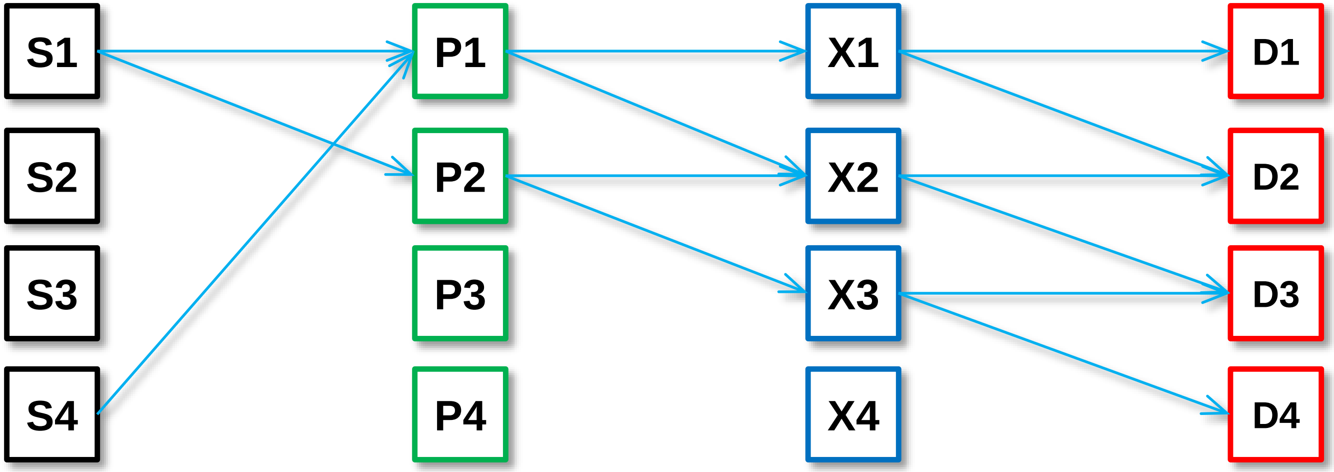
```
set_false_path –from S1
```

```
S1 → P1 → X1 → D1
S1 → P1 → X1 → D2
S1 → P1 → X2 → D2
S1 → P1 → X2 → D3
S1 → P2 → X2 → D2
S1 → P2 → X2 → D3
S1 → P2 → X3 → D3
S1 → P2 → X3 → D4
```

```
set_false_path –through P1
```

```
S1 → P1 → X1 → D1
S1 → P1 → X1 → D2
S1 → P1 → X2 → D2
S1 → P1 → X2 → D3
S4 → P1 → X1 → D1
S4 → P1 → X1 → D2
S4 → P1 → X2 → D2
S4 → P1 → X2 → D3
```

Path Specification #2



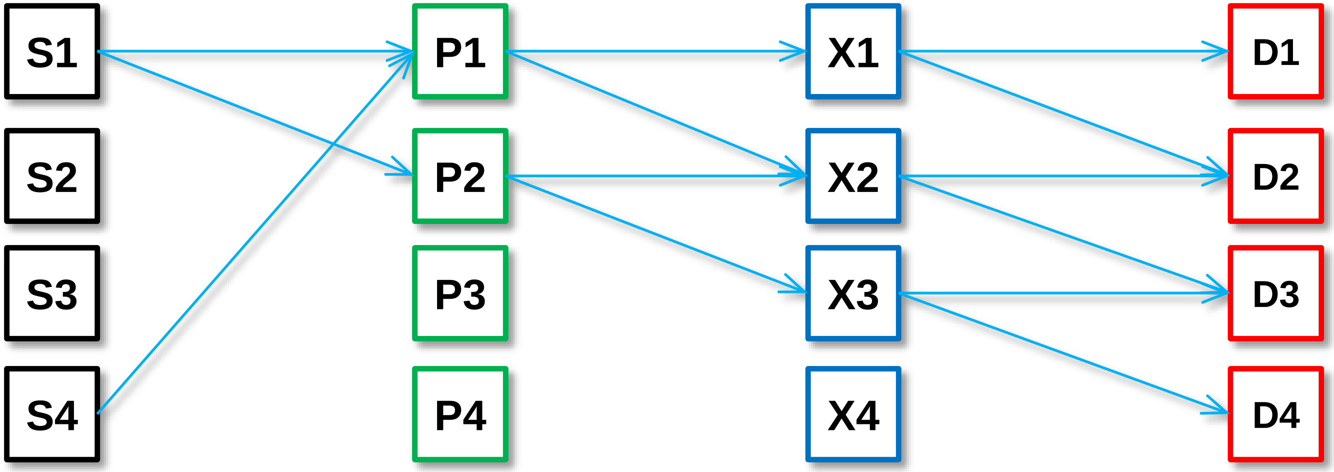
```
set_false_path -to D1
```

S1 → P1 → X1 → D1
S4 → P1 → X1 → D1

```
set_false_path -from S1 -through X1
```

S1 → P1 → X1 → D1
S1 → P1 → X1 → D2

Path Specification #3



```
set_false_path -from S1 -through {X1, X2}
```

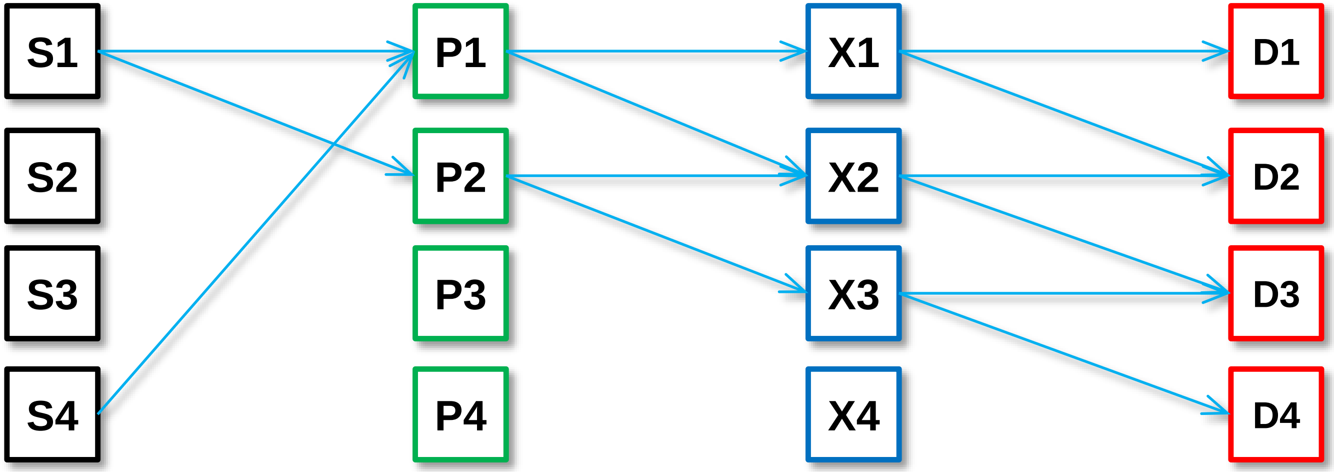
⇕

```
set_false_path -from S1 -through X1
set_false_path -from S1 -through X2
```

S1 → P1 → X1 → D1
S1 → P1 → X1 → D2
S1 → P1 → X2 → D2
S1 → P1 → X2 → D3
S1 → P2 → X2 → D2
S1 → P2 → X2 → D3

Paths starting from S1 and passing through either of (X1 or X2)

Path Specification #4



```
set_false_path -through P1 -through X1
```

S1 → P1 → X1 → D1 *S1 → P1 → X1 → D2*
S4 → P1 → X1 → D1 *S4 → P1 → X1 → D2*

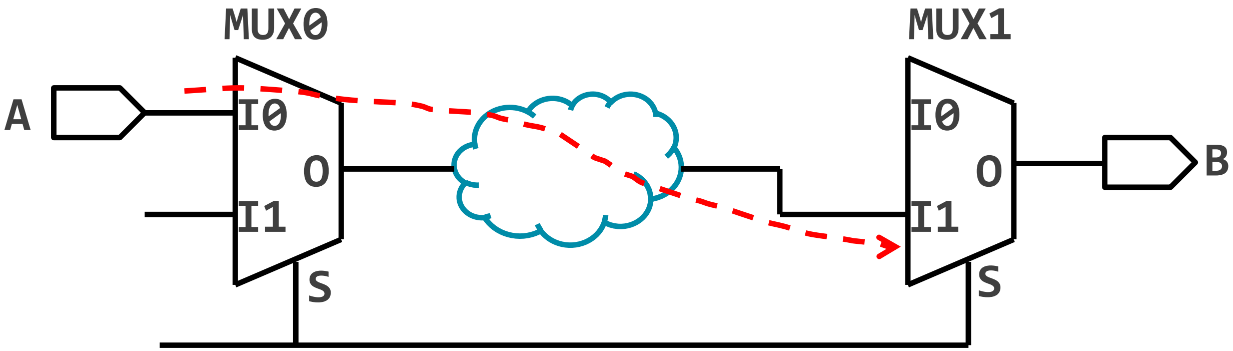
- `-through P1 -through X1` ≠ `-through X1 -through P1`
- `-through {X1, X2}` ≠ `-through X1 -through X2`

When -through is specified multiple times, it indicates that each of the –through have to be satisfied independently

Path Specification #4

- `set_false_path -from CLK1` means all paths originating from
  - All sequential elements triggered by `CLK1`
  - And all input ports constrained with respect to `CLK1`
- Transition Specification
  - `-rise_from`
  - `-fall_from`
  - `-rise_through`
  - `-fall_through`
  - `-rise_to`
  - `-fall_to`
  - `-rise : impacts only rising paths`
  - `-fall : impacts only falling paths`

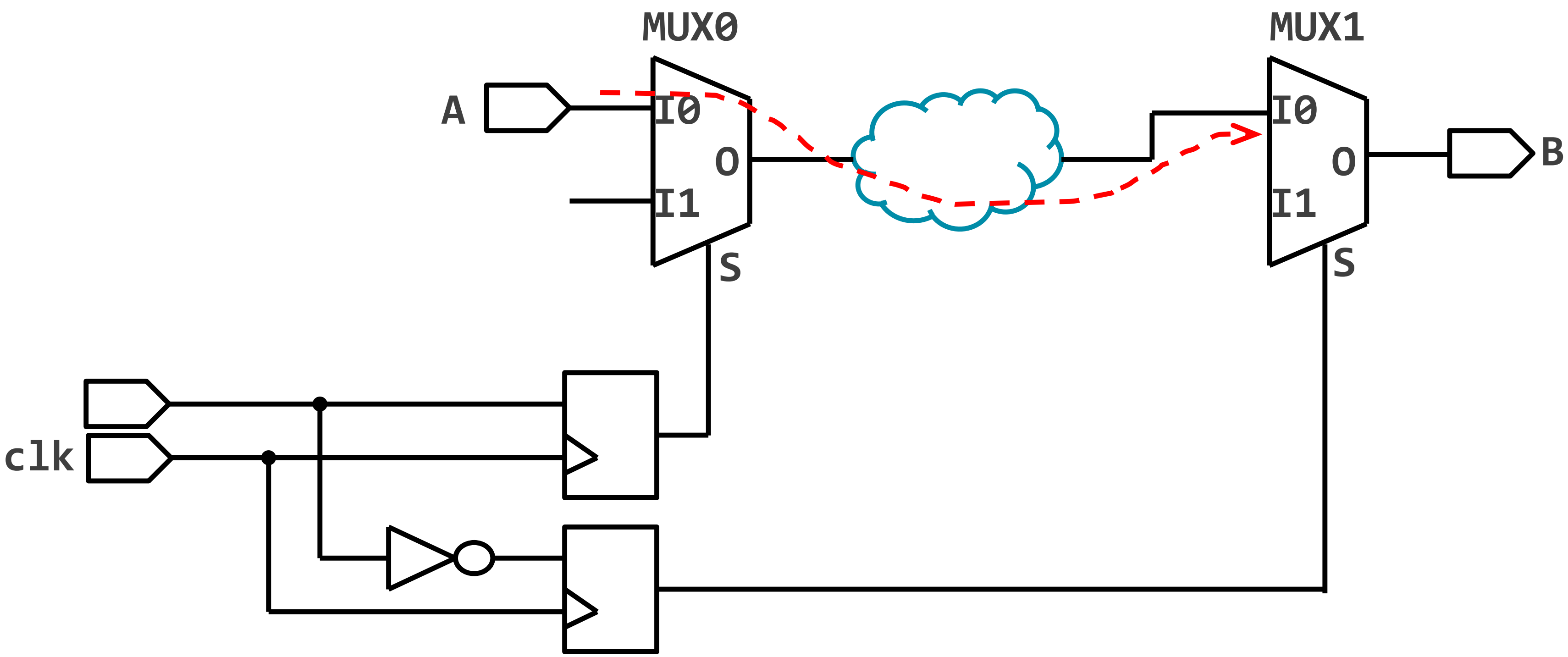
Types of False Paths: Combinational False Path



```
set_false_path -from [get_ports A] -through \
[get_pins mux0/I0] -through [get_pins mux1/I1] -to [get_ports B]
```

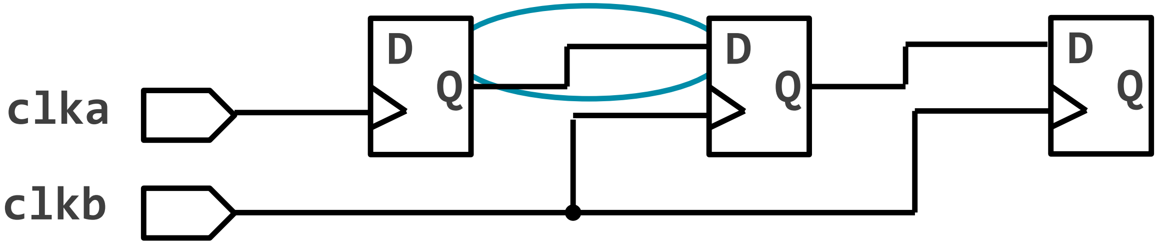
```
set_false_path -through [get_pins MUX0/I0] \
-through [get_pins MUX1/I1]
```

Types of False Paths: Sequential False Path



```
set_false_path -through [get_pins MUX0/I0] \
-through [get_pins MUX1/I0]
```

Types of False Paths: Asynchronous Domain Crossings



```
set_false_path -from [get_clocks clka] -to [get_clocks clkb]
set_false_path -from [get_clocks clkb] -to [get_clocks clka]
```

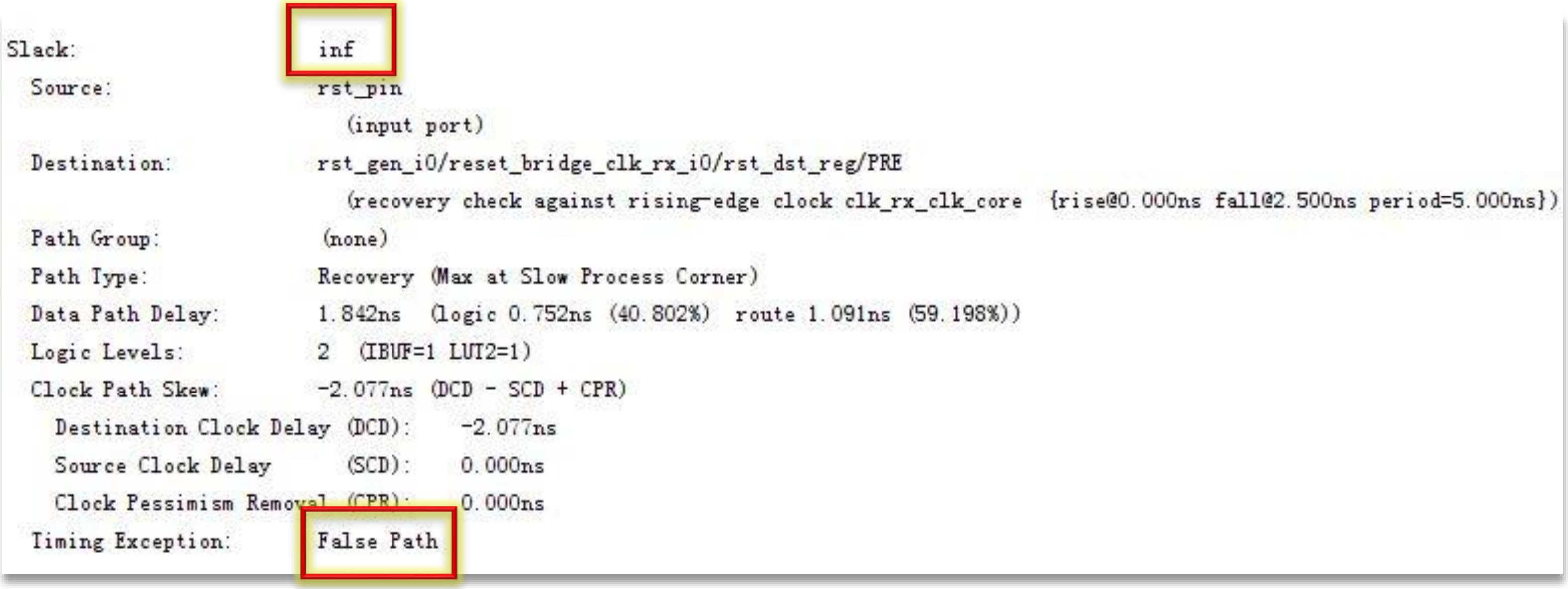
⇕

```
set_clock_groups -asynchronous -group [get_clocks clka] \
-group [get_clocks clkb]
```

False Path Timing Report

```
set_false_path -from [get_ports rst_pin]
```

report_timing -from [get_ports rst_pin]



```
Slack:                  inf
  Source:               rst_pin
                          (input port)
  Destination:          rst_gen_i0/reset_bridge_clk_rx_i0/rst_dst_reg/PRE
                          (recovery check against rising-edge clock clk_rx_clk_core  {rise@0.000ns fall@2.500ns period=5.000ns})
  Path Group:           (none)
  Path Type:            Recovery (Max at Slow Process Corner)
  Data Path Delay:      1.842ns  (logic 0.752ns (40.802%)  route 1.091ns (59.198%))
  Logic Levels:         2  (IBUF=1 LUT2=1)
  Clock Path Skew:      -2.077ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    -2.077ns
    Source Clock Delay      (SCD):    0.000ns
    Clock Pessimism Removal (CPR):    0.000ns
  Timing Exception:     False Path
```

False Path Impact

- **Impact on synthesis**
  - It is usually not needed to use false path exceptions during synthesis except for ignoring CDC paths
- **Impact on implementation**
  - All the implementation steps are sensitive to the false path timing exception