Vivado从此开始（To Learn Vivado From Here）

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

Setting Multicycle Path Constraint

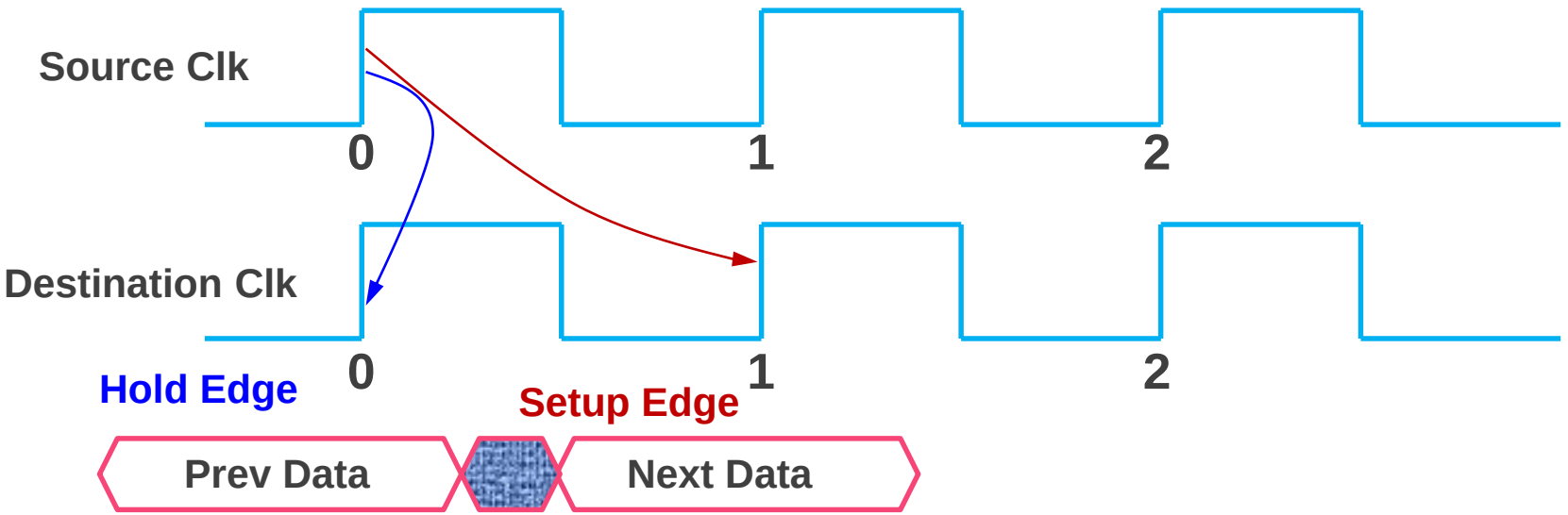
Lauren Gao

Why Multicycle Path Exceptions

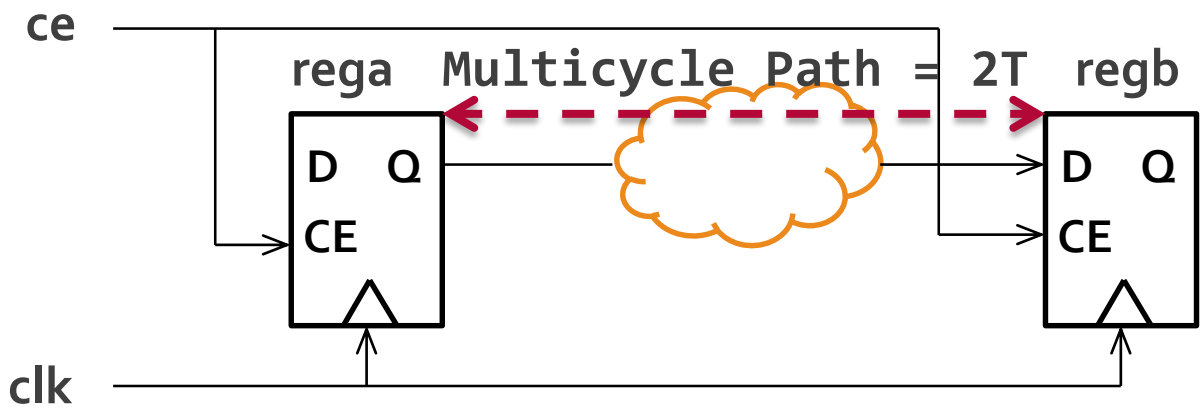
- **Relaxes timing requirements**
  - Some logic paths need more than 1 cycle to propagate to next sequential cell
  - Allows tools to focus on real critical paths
- **It effects both setup and hold timing**

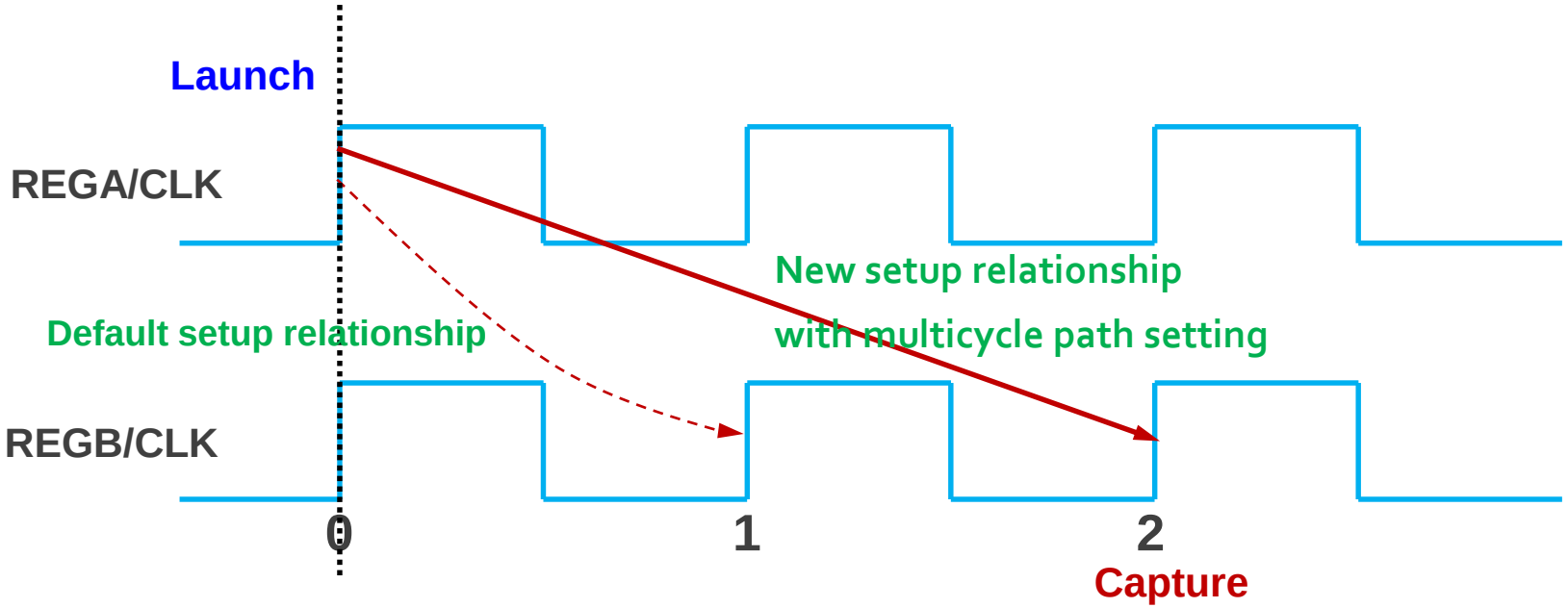
Default Setup and Hold Analysis

- **Setup check is performed at destination:**
  - 1 clock cycle after the launch clock edge
- **Hold check is performed at destination:**
  - `1-1 = 0` clock cycles after the launch clock edge

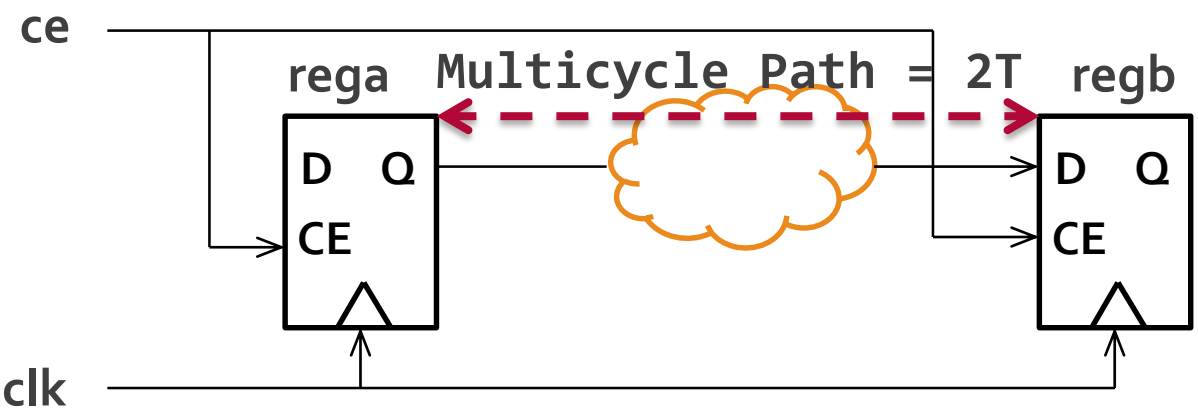


Multicycle Path : An Example

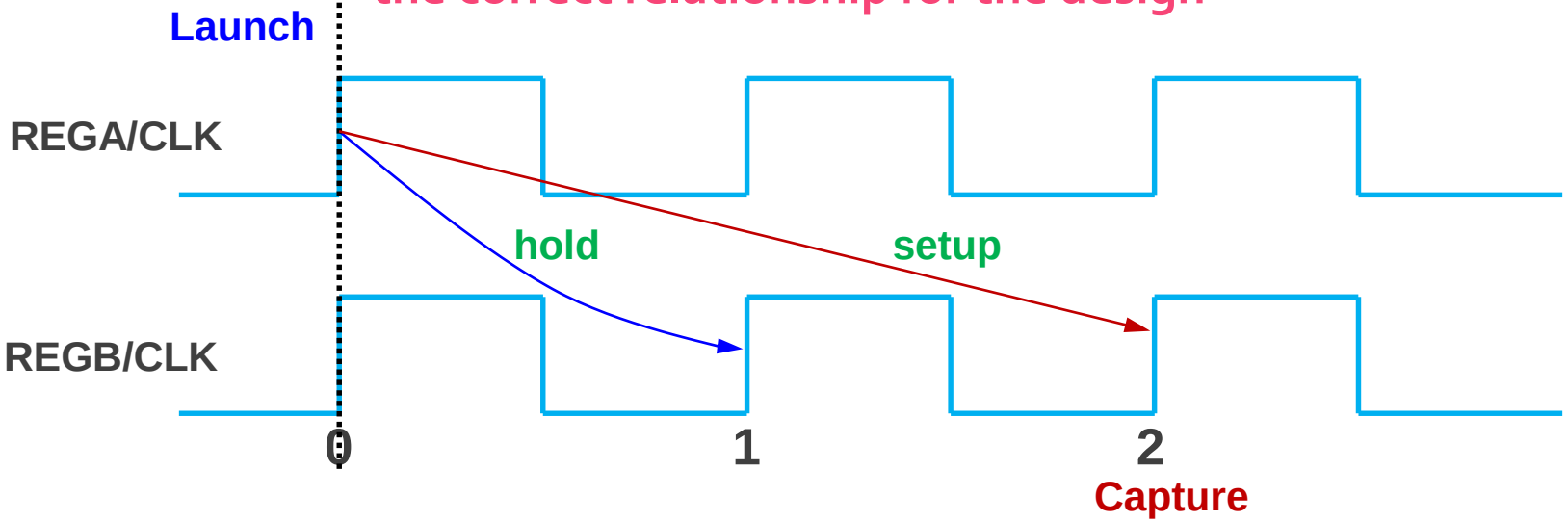




Multicycle Path : An Example

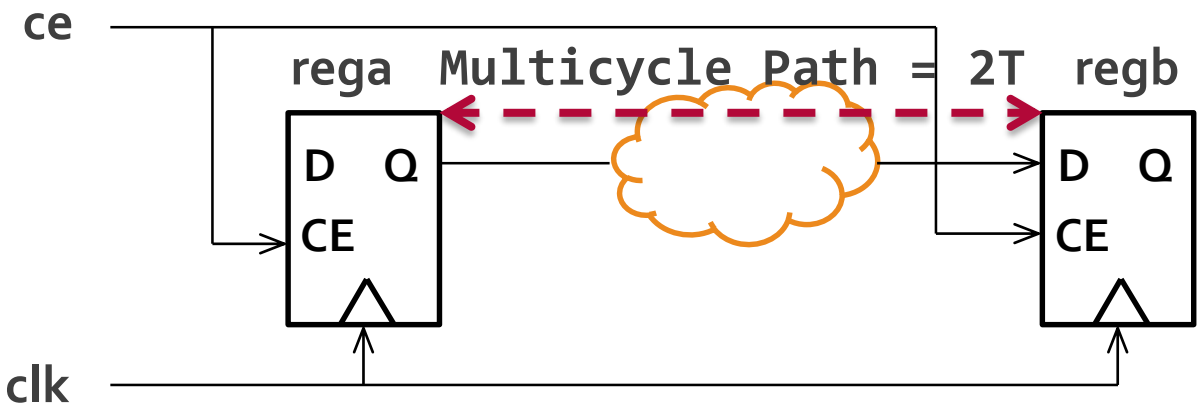


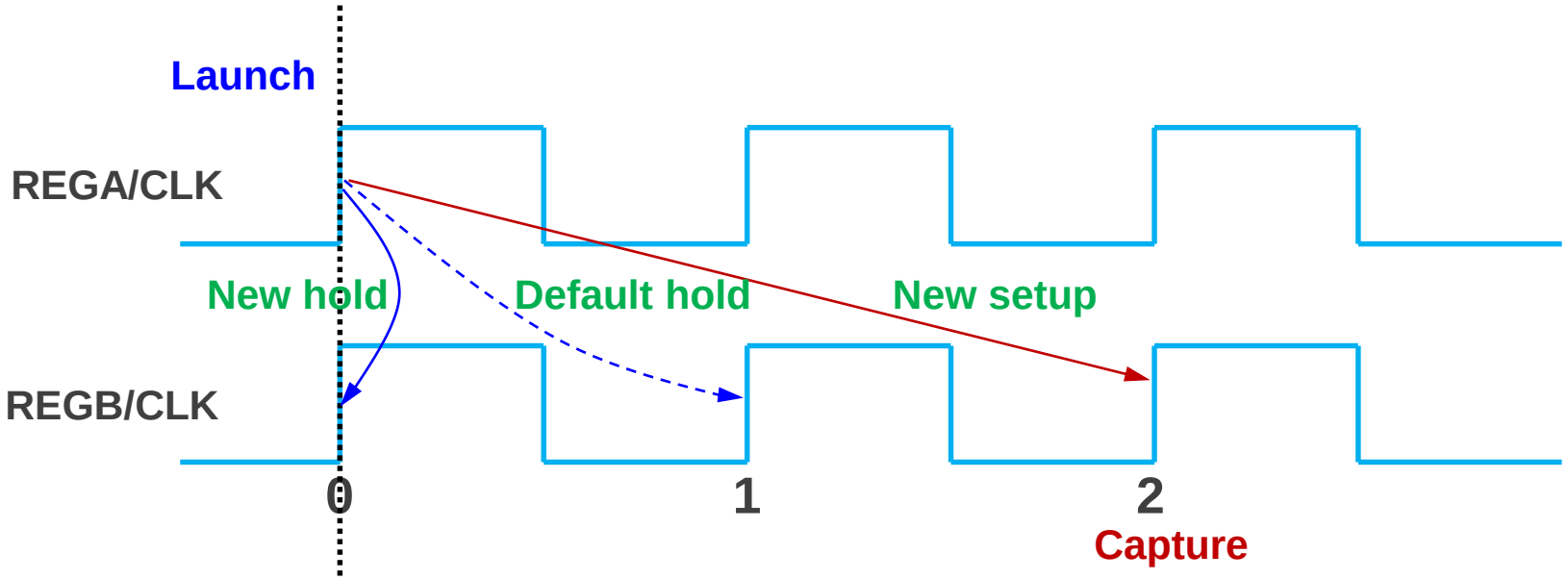
The hold relationship shown beside is probably not the correct relationship for the design



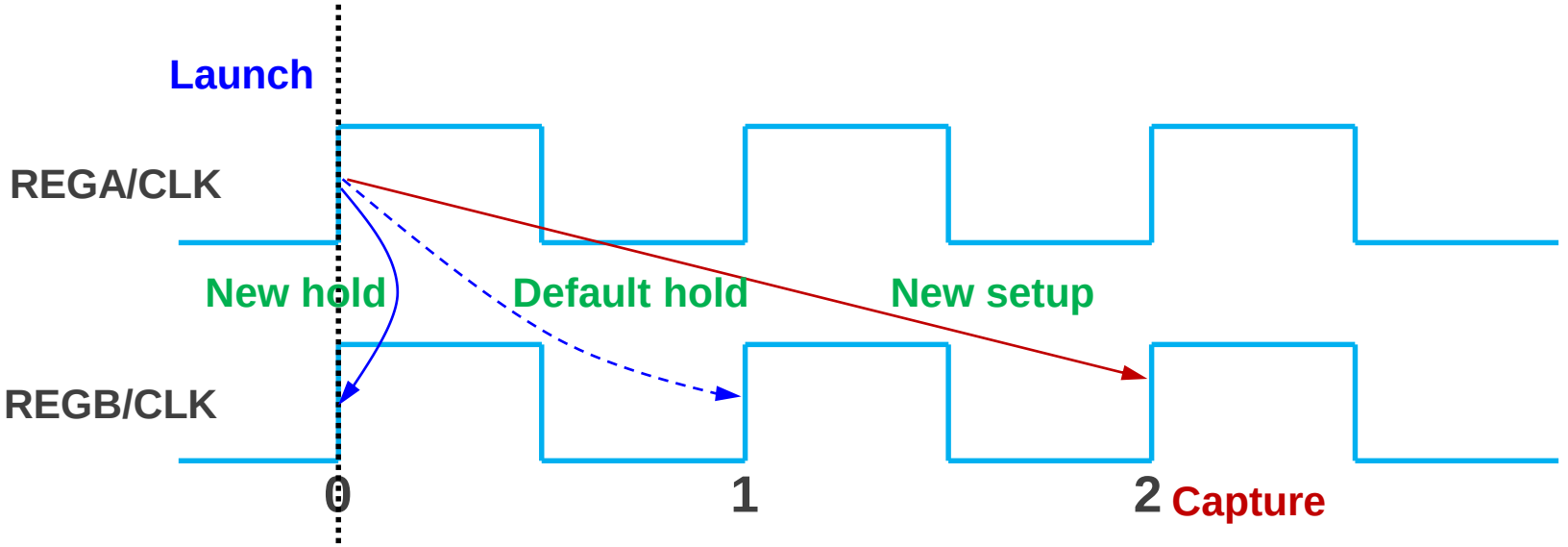
Hold Check Is Based on Setup Check!

Multicycle Path : An Example





Set_multicycle_path




```
set_multicycle_path -from [get_cells rega] -to [get_cells rega] –setup -end 2
set_multicycle_path -from [get_cells regb] -to [get_cells regb] –hold -end 1
```

***语句中数字的含义：**

对于-setup：表示该多周期路径所需要的时钟周期个数；

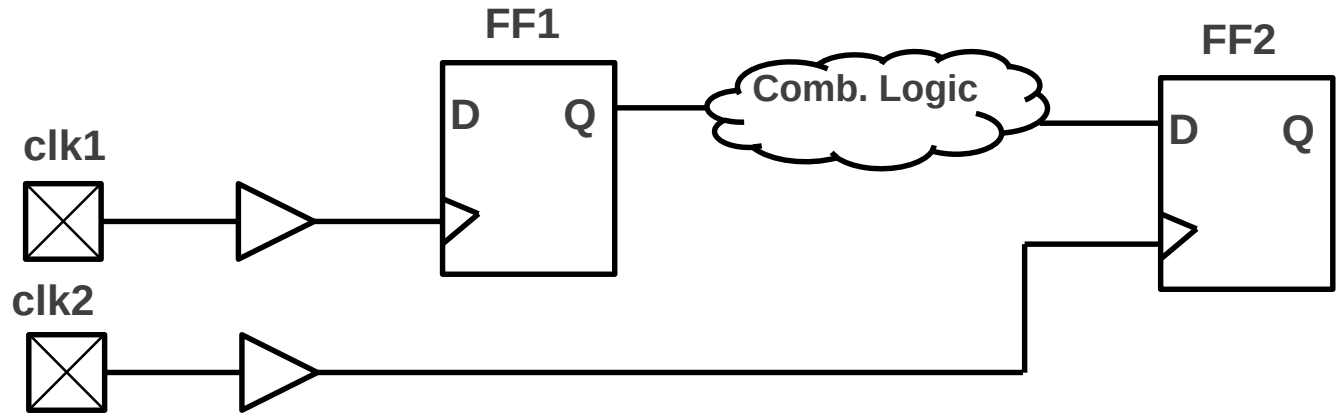
对于-hold：表示相对于缺省捕获沿（图中的Default hold）,实际捕获沿（图中的New hold）应回调的时钟周期个数

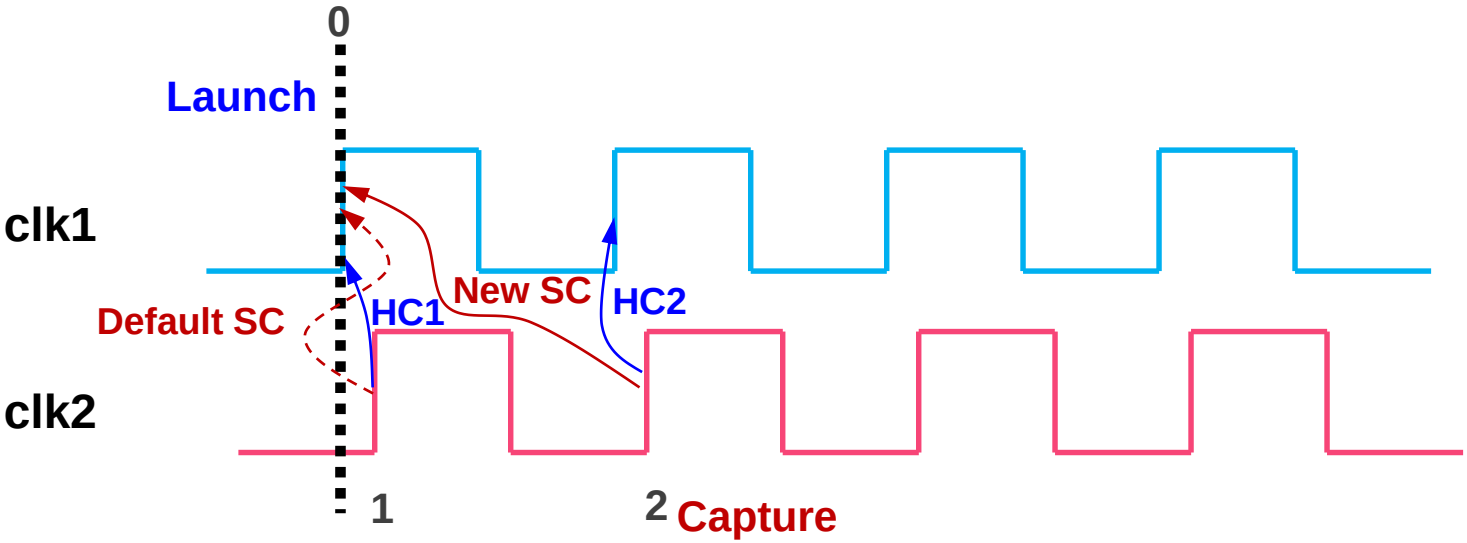
***参考时钟周期的选取：**

-end表示参考时钟为捕获端（收端）所用时钟，对于-setup缺省为-end

-start表示参考时钟为发送端（发端）所用时钟，对于-hold缺省为-start

App 1 :
Same Frequency Clocks with Destination Clock Positive Offset

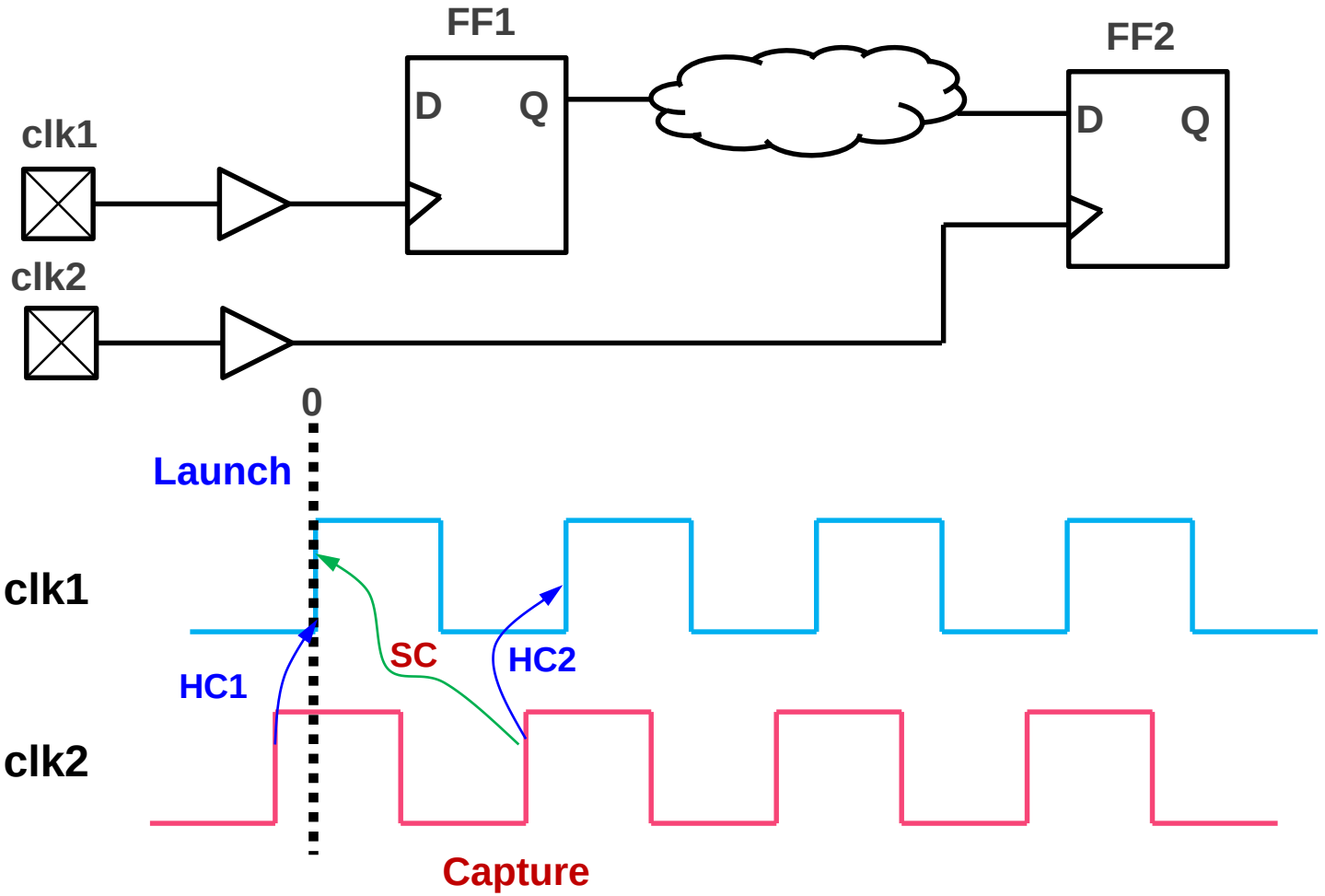





```
set_multicycle_path -from [get_clocks clk1] -to [get_clocks clk2] -setup -end 2
```

In this example, the default hold analysis returns the preferred hold requirements and no multicycle hold exceptions are required

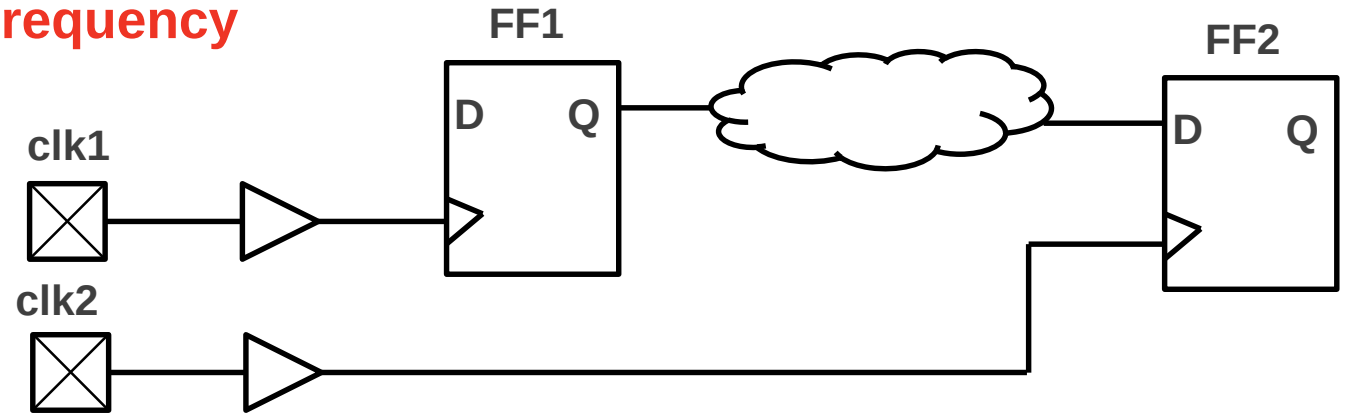
App 2 :
Same Frequency Clocks with Destination Clock Negative Offset

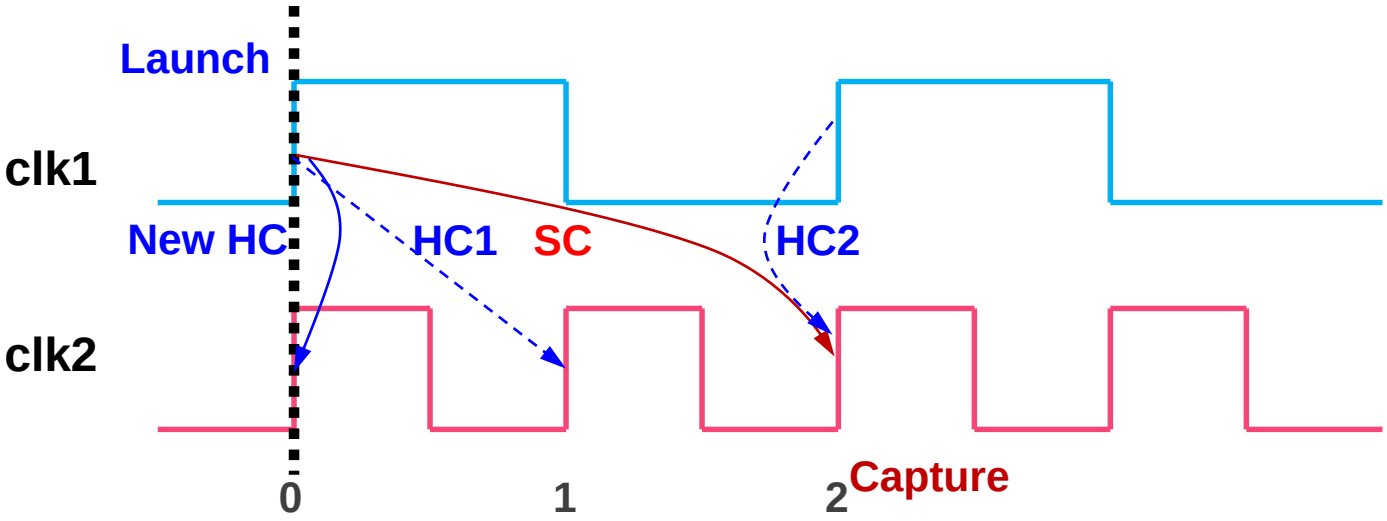


In this example, the default setup and hold analysis returns the preferred requirements and no multicycle exceptions are required

App 3 :

The Destination Clock Frequency is a Multiple of the Source Clock Frequency

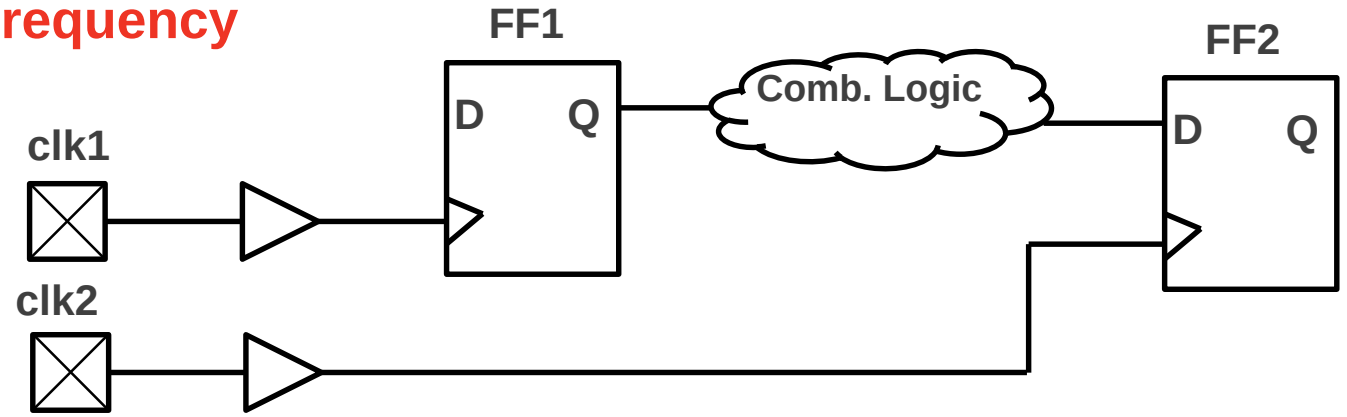


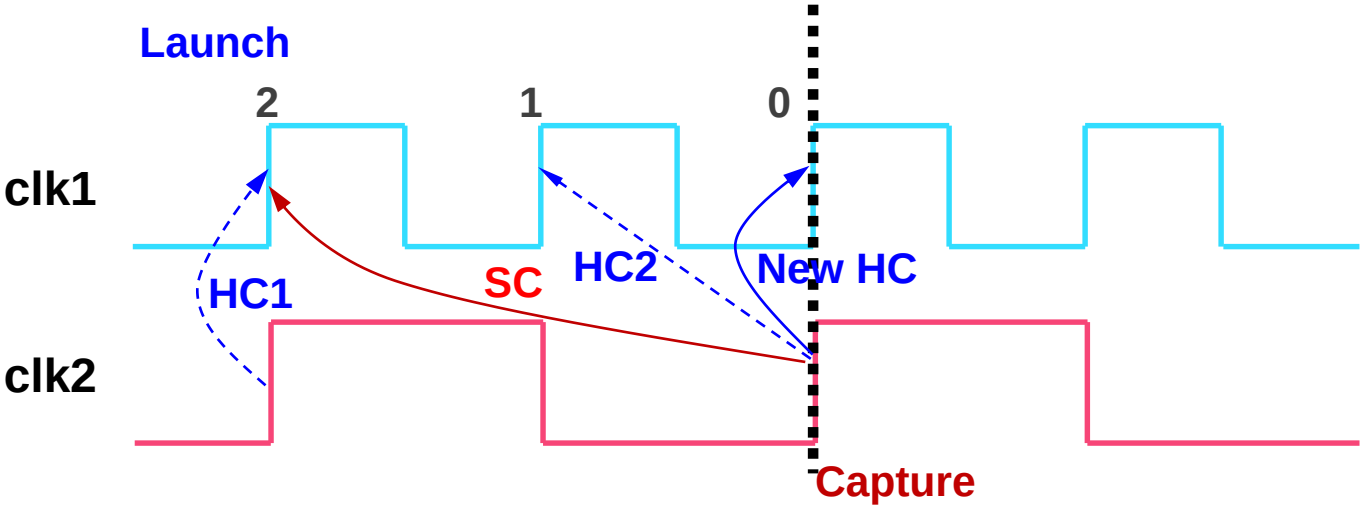



```
set_multicycle_path -from [get_clocks clk1] -to [get_clocks clk2] -setup -end 2
set_multicycle_path -from [get_clocks clk1] -to [get_clocks clk2] -hold -end 1
```

App 4 :

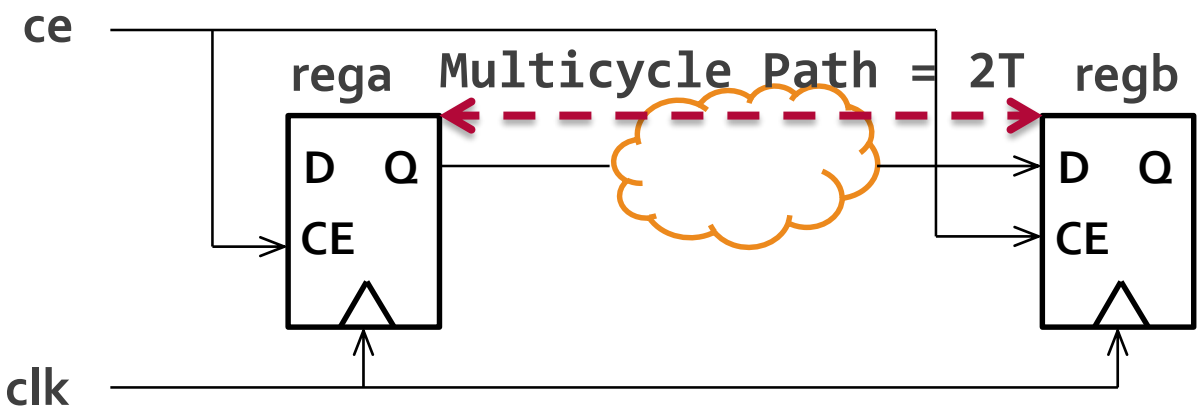
The Source Clock Frequency is a Multiple of the Destination Clock Frequency






```
set_multicycle_path -from [get_clocks clk1] -to [get_clocks clk2] -setup -start 2
set_multicycle_path -from [get_clocks clk1] -to [get_clocks clk2] -hold -start 1
```

App 5: Multicycle Path Using Clock Enable




```
set encells [get_cells -of_objects [get_pins -leaf –filter\
{IS_ENABLE==1} -of_objects [get_nets CE]]]
```

```
set_multicycle_path -from $encells -to $encells -setup 2
set_multicycle_path -from $encells -to $encells -hold 1
```

Summary

- By default, the setup path multiplier is defined with respect to the destination clock (-end)
- By default, the hold path multiplier is defined with respect to the source clock (-start)
- The -start/-end options have no effect when applying a multicycle path constraint on paths clocked by the same clock, or clocked by two clocks having the same waveform, or with no phase shift