FPGA Clocking

- Clock-related issues:
  - distribution
  - generation (frequency synthesis)
  - Deskew
  - multiplexing
  - run-time programming
  - domain crossing

- Clock-related constraints

Clock Distribution

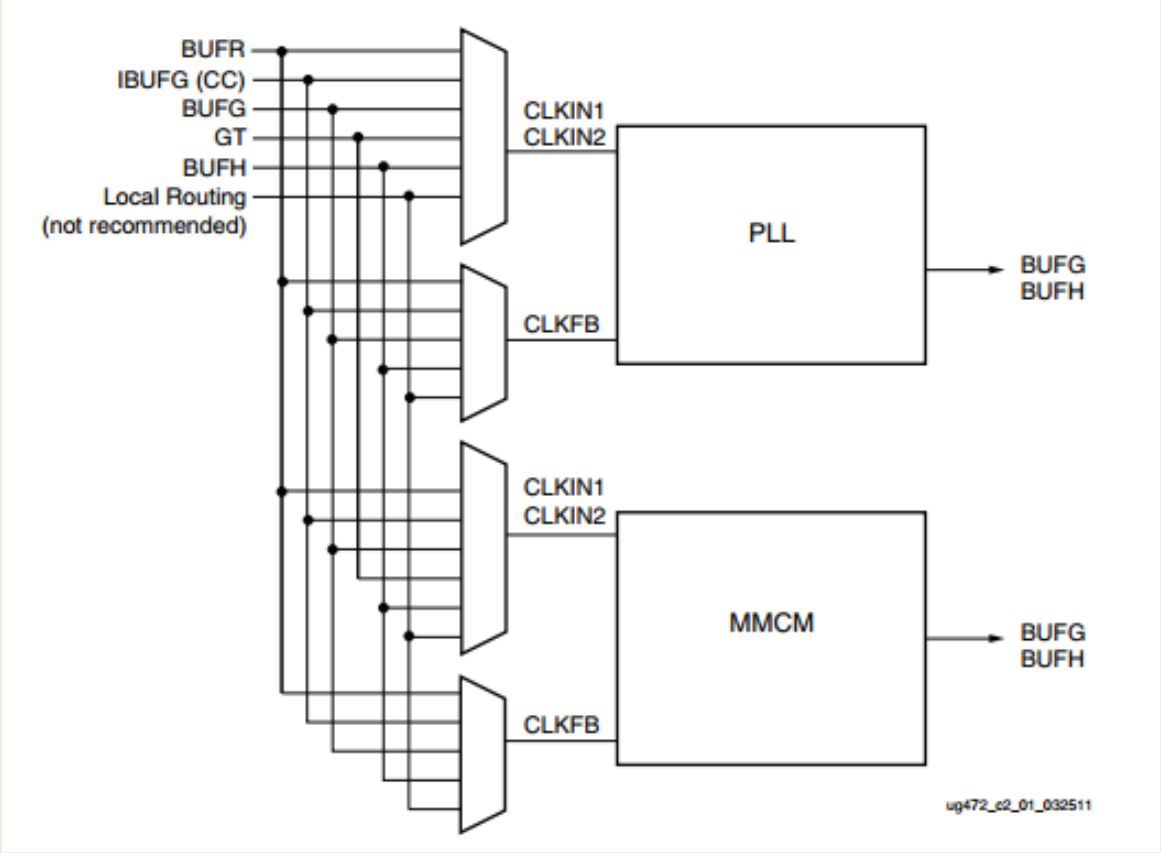
- Device split into halves (top/bottom) and clock regions (rectangular, uniform size)
- Different clock buffer types:
  - BUFG can clock any FF within the top or bottom of device (top/bottom BUFGs)
  - BUFR clocks a single clock region
  - BUFMR clocks up to 3 clock regions
  - BUFH clock a horizontal clock spine within a region

FPGA Clocking

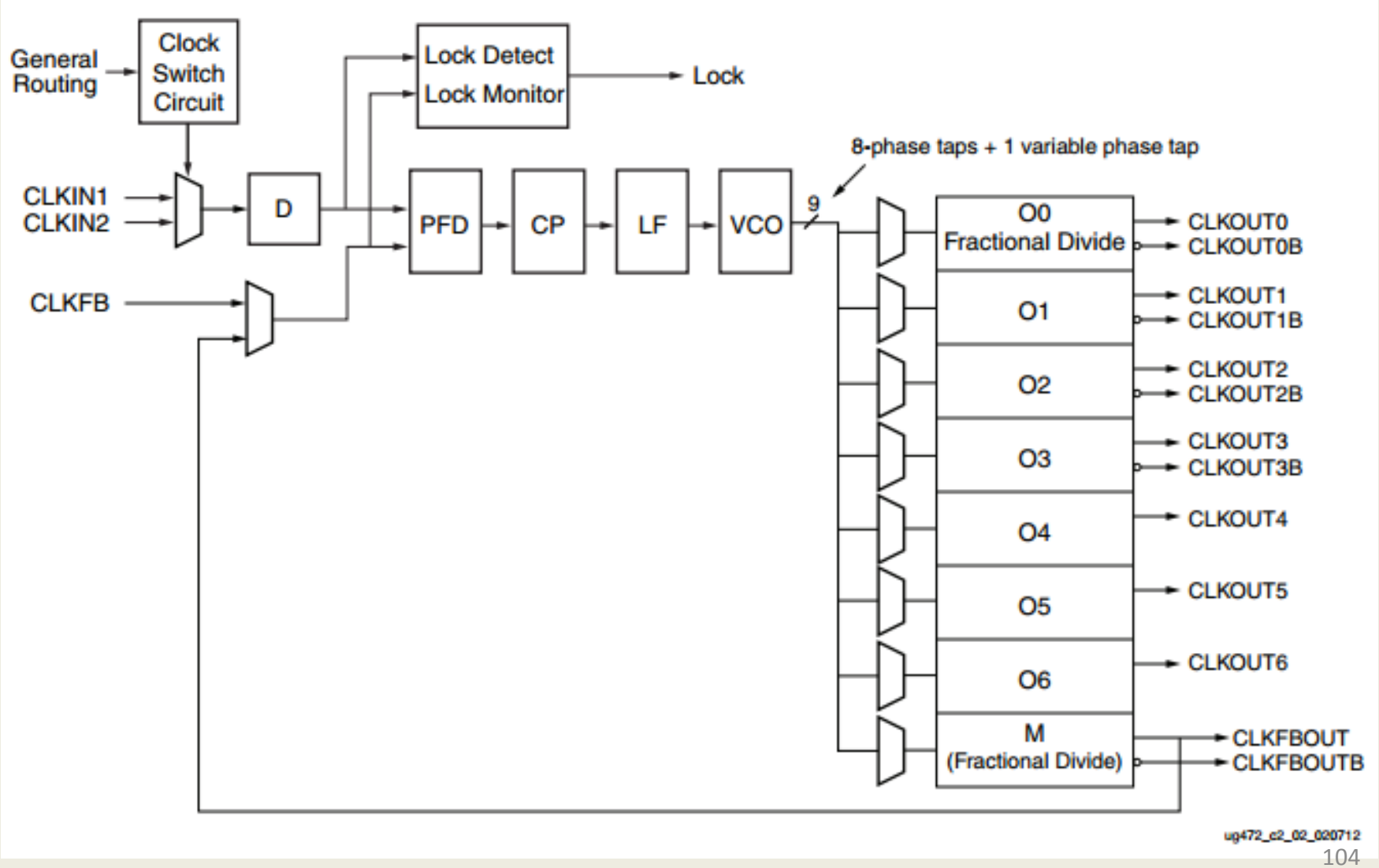
- Clock generation (frequency synthesis)
  - Uses “Clock Management Tiles” which consist of:
    - PLL/DCM (Frequency Synthesis)
    - MMCM (Advanced PLL with phase control)
  - Clock input from PCB must use “Clock capable pins” of FPGA
    - Differential pairs
    - Single-ended clocks always use P side

Frequency Synthesis

- Common use case: generate all design clocks from single input clock (crystal oscillator)



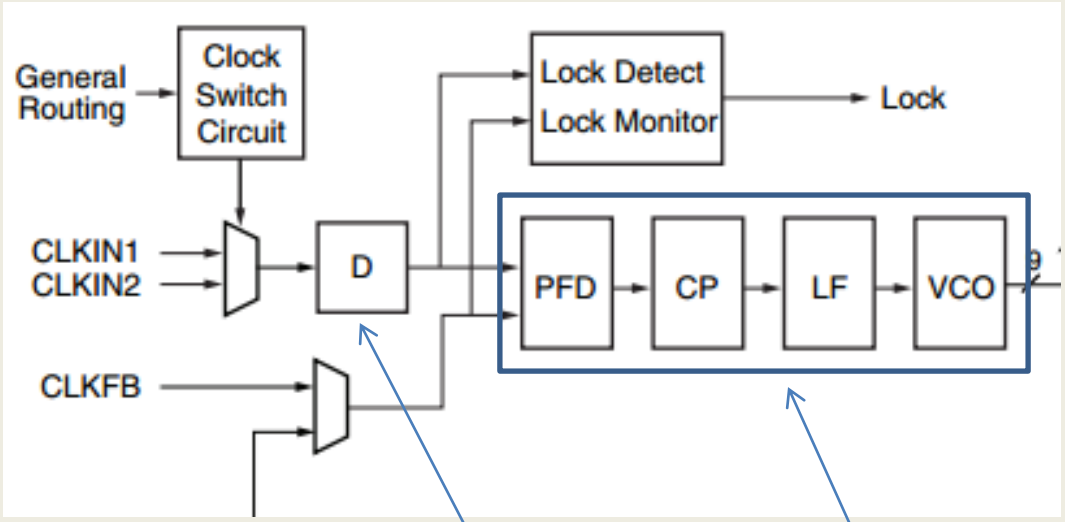
Frequency Synthesis with MMCM



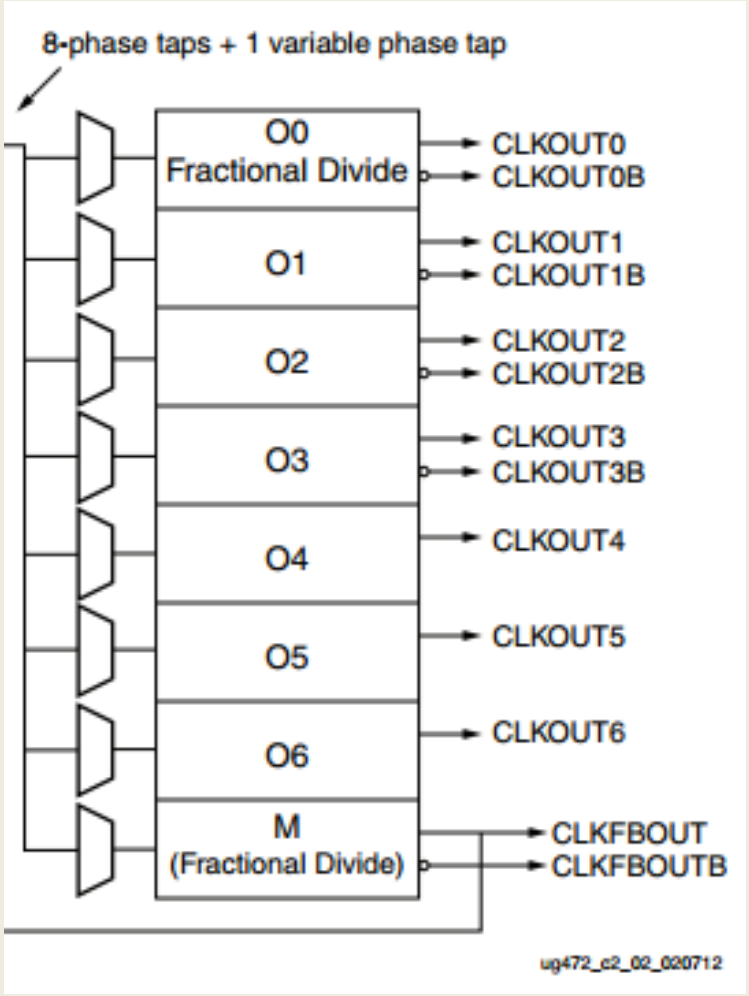
Frequency Synthesis with MMCM

MMCM Input Block: VCO frequency generator

- $F_{VCO}=F_{CLKIN}*(M/D)$
- F_{VCO} is limited, not all M/D values are valid



Frequency Synthesis with MMCM



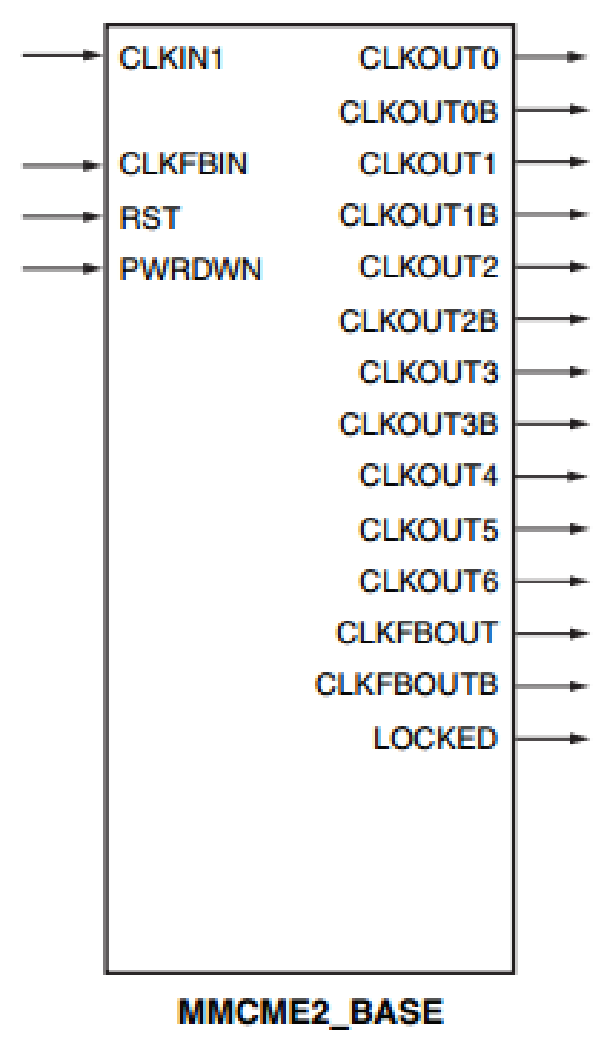
Output divider block:

- Phase shift
- Frequency division

PLL lacks the following:

- Variable Phase Tap
- CLKOUTxB
- CLKFBOUTB
- Fractional dividers

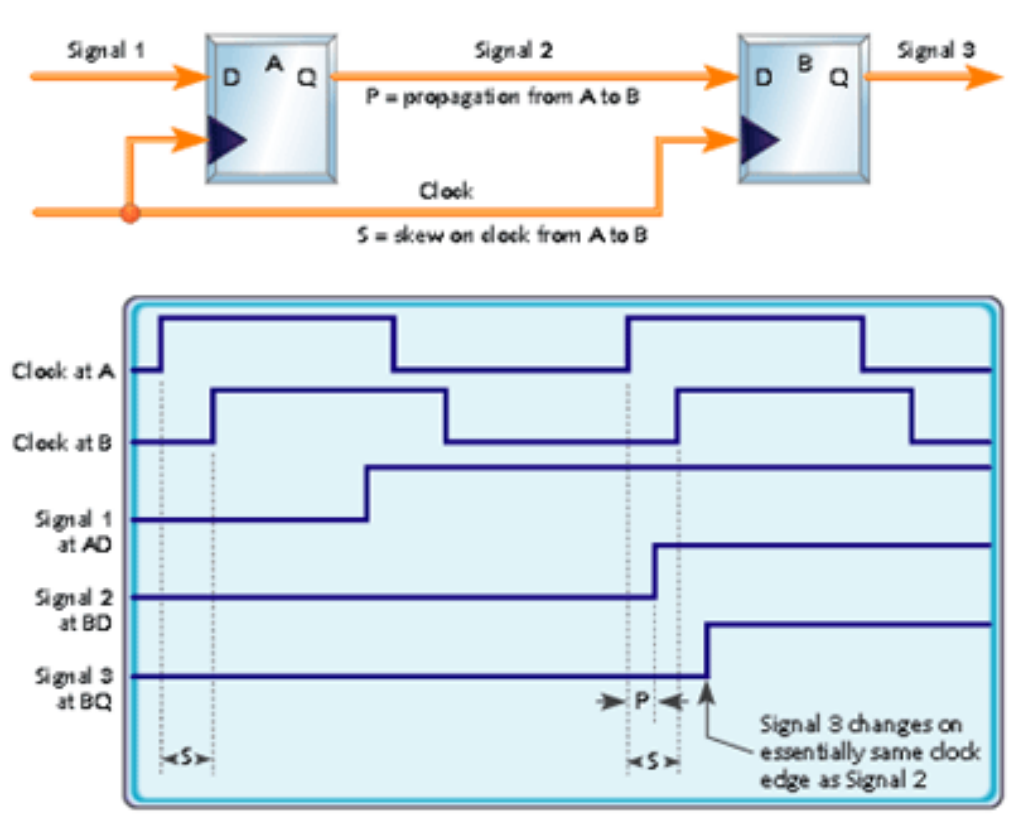
Frequency Synthesis with MMCM




```
 MMCME2_BASE #(
     .CLKFBOUT_MULT_F(),     // (2.000-64.000)
     .CLKOUTx_DIVIDE(),      // (1-128)
     .CLKOUT0_DIVIDE_F(),    //  (1.000-128.000)
     .CLKOUTx_DUTY_CYCLE()  // (0.01-0.99)
     .DIVCLK_DIVIDE(),        // (1-106)
) MMCME2_BASE_inst (
     .CLKOUTx(),
     .CLKOUTxB(),
     .CLKFBOUT(),
     .CLKFBOUTB(),
     .LOCKED(),
     .CLKIN1(),
     .PWRDWN(),
     .RST(),
     .CLKFBIN());
```

Clock deskew with MMCM

- Skew is delay in clock arrival at FFs, which is caused by clock routing (buffers, wires)

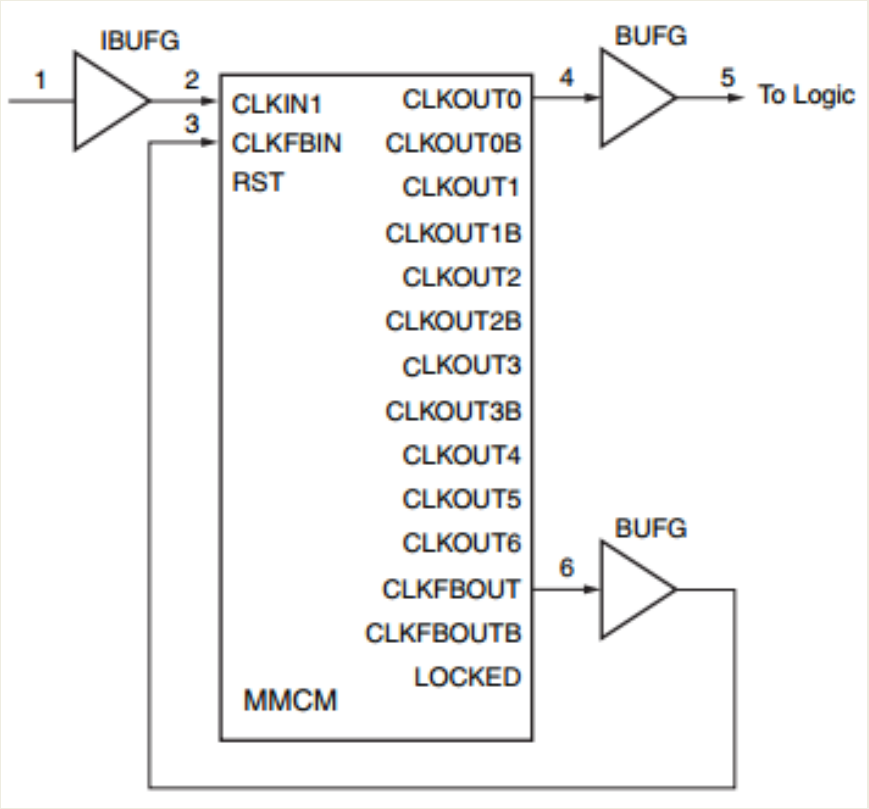


Clock deskew with MMCM

- Use clock buffer in same clock region of FPGA to drive CLKFBIN
- Output clock buffers

must be in the same clock region relative to their function:

  - BUFG same half
  - BUFH same clock region



Dynamic MMCM Reconfiguration

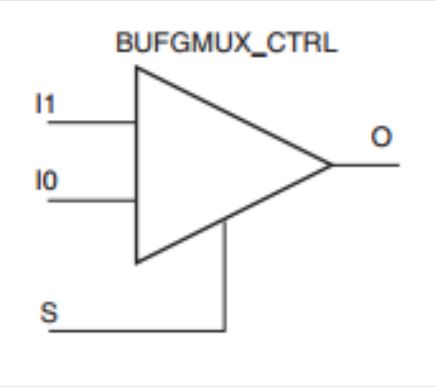
- MMCMs have run-time (dynamic) reconfiguration ports
- Reconfigurable parameters:
  - Phase
  - Divide Counters
  - Duty cycle
- Configuration through multiple registers and addressable access port (see XAPP 888 in further reading)

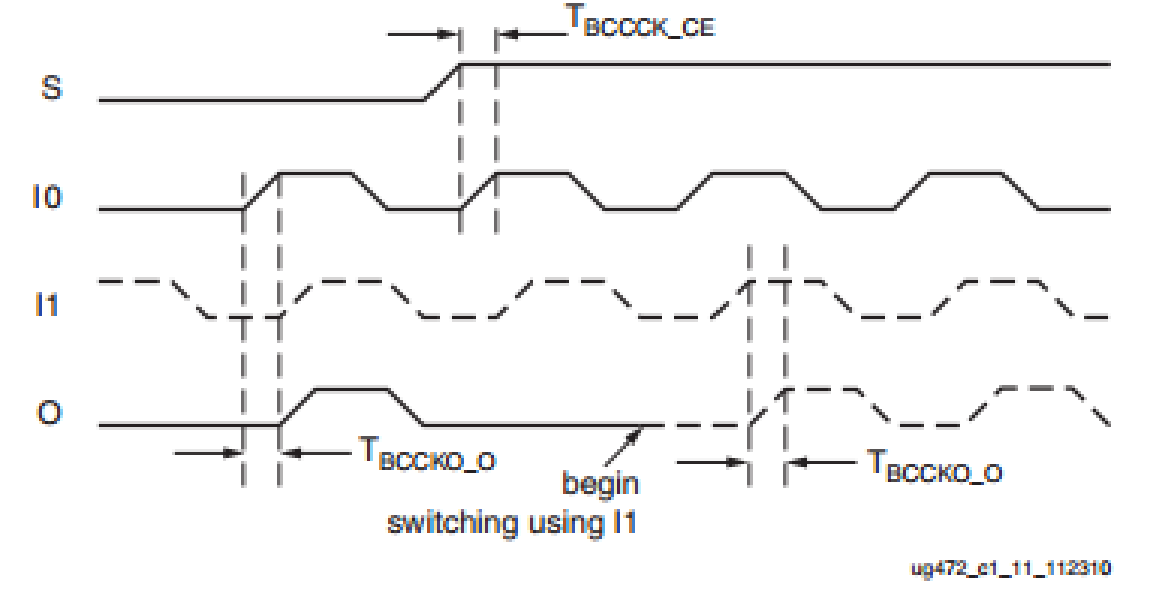
Clock Multiplexing

- Many applications require clock multiplexing:
  - In-circuit debugging (to avoid domain crossing)
  - Dynamic frequency scaling
  - Clock redundancy (switch away from dead clock)
- FPGA clock multiplexers (2:1) implemented with BUFGMUX_CTRL primitive

Clock Multiplexing

- Clock switch latency: max 3 clock cycles of the slower clock
- Glitchless output



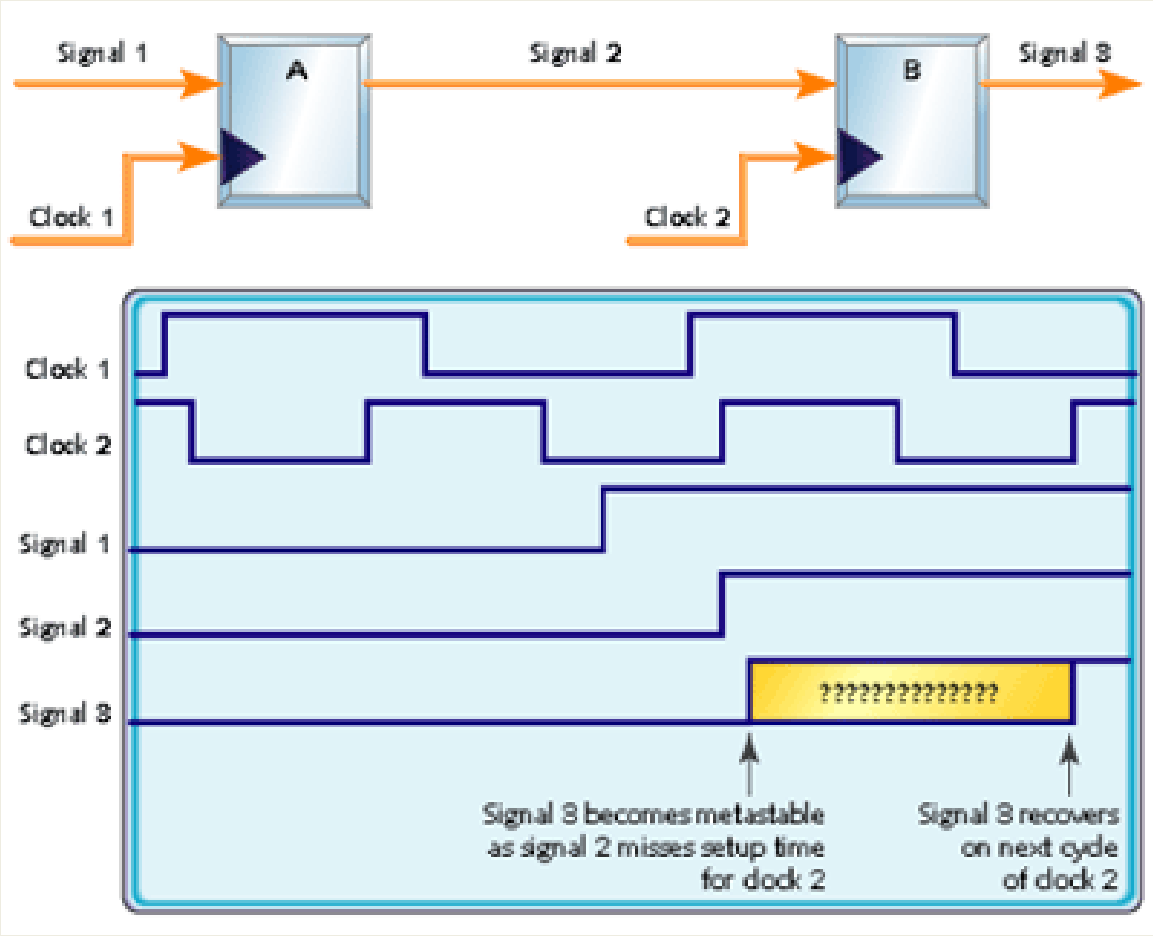


Clock-Related Constraints

- Timing constraints:
  - Period: guides timing analysis with regard to a periodic signal (clock)
    - NET “<clock_name>" TNM_NET = “<group_name>";
    - TIMESPEC <spec_name> = PERIOD “<group_name>" <period> <unit> HIGH/LOW <percentage> %;
  - OFFSET IN/OUT
  - FROM:TO

Clock domain crossing

- Problem: direct crossing leads to metastability

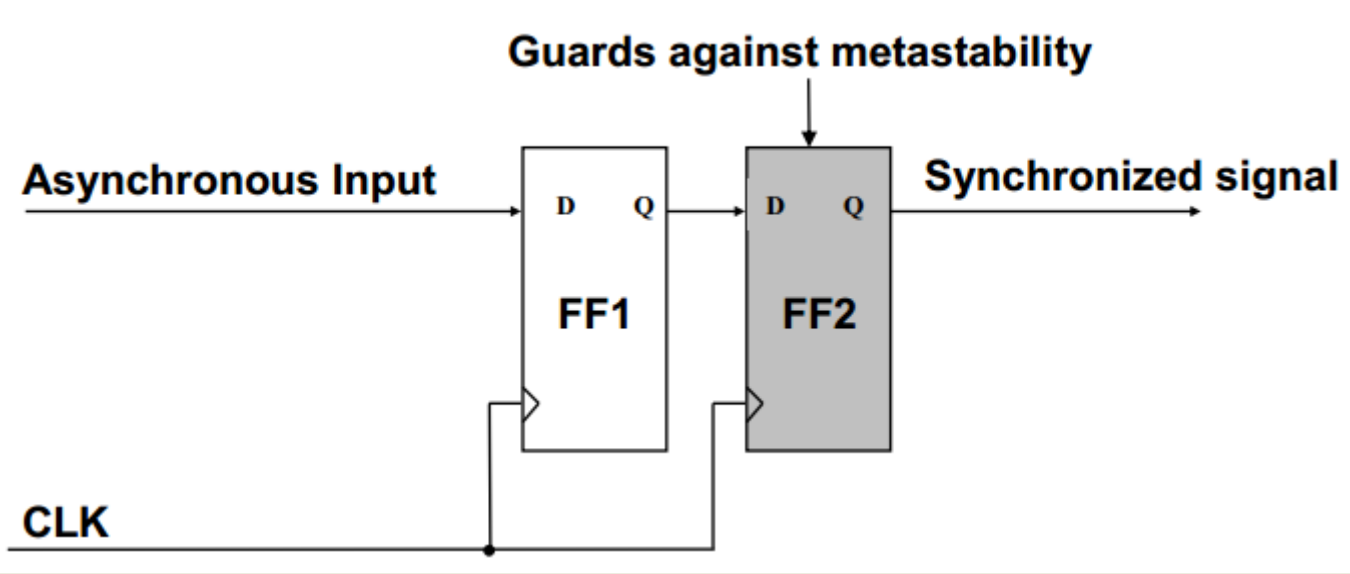


Clock domain crossing

- FPGAs have rapid average metastability recovery (ps)
- But recovery is unbounded (probability is non-null that the metastable state will last a given time T)
- Design goal: achieve a desired MTBF, given the recovery parameters of the flip-flop

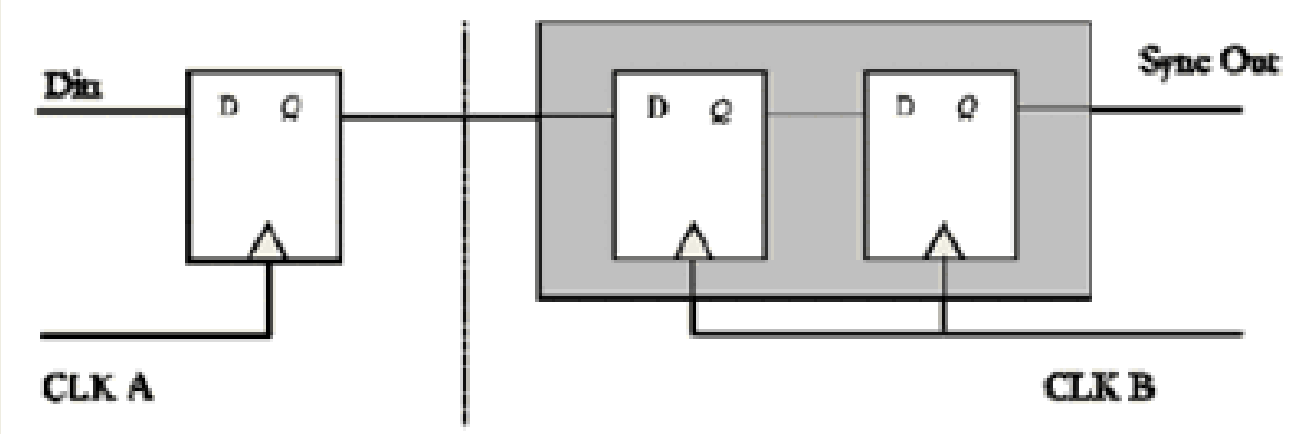
Clock domain crossing

- Synchronizer: Multiple FFs at the receiving end guard against metastability
- Number of FFs a function of desired MTBF and ratio of clock frequencies



Clock domain crossing

- Register output from source domain for more predictable timing (increased MTBF)

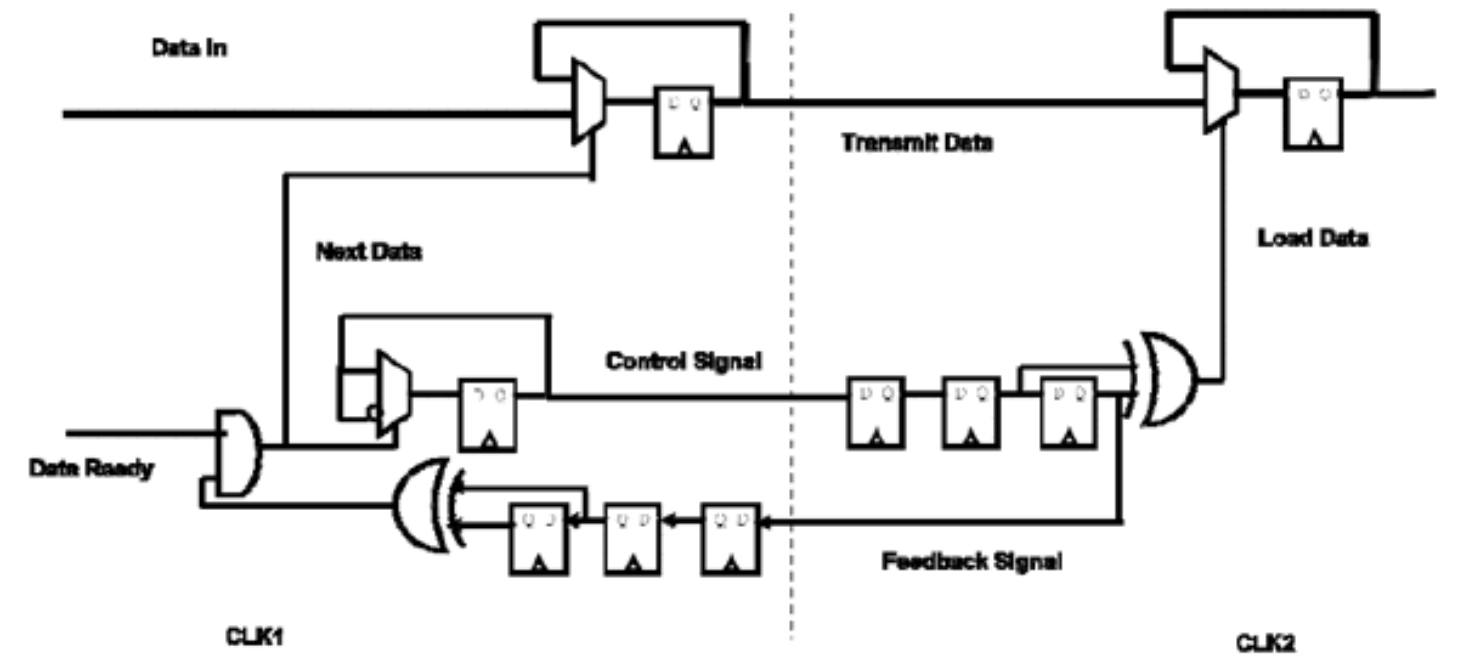


Clock domain crossing

- 2FF synchronizers work well when destination clock is faster than source clock (signal will remain stable for at least one destination clock cycle)
- Fast-to-slow crossing requires closed-loop synchronizer with handshake

Clock domain crossing

- Closed-loop: Control signal crosses into clock domain 2, then back into clock domain 1, and is checked against the reference (high latency)



Further Reading

- Article: Tim Behne, FPGA Clock Schemes
- Xilinx 7-Series Clock Resources User Guide
- XAPP888: MMCM and PLL Dynamic Reconfiguration
- Cliff Cummings: Clock Domain Crossing
- XAPP94: Metastability Recovery
- Xilinx: FPGA Design Techniques
- SOC Central: Clock Domain Crossing Demystified