Programmable Logic Design
Grzegorz Budzyń
Lecture 10: FPGA clocking schemes
Plan

• Introduction
• Definitions
  – Clock skew
  – Metastability
• FPGA clocking resources
  – DCM
  – PLL
Introduction

• One of the most important steps in the design process is to identify how many different clocks to use and how to route them
Introduction

• As larger designs are implemented in FPGAs, it is likely that many of them will have multiple data paths running on multiple clocks.
• An FPGA design that contains multiple clocks requires special attention.
• Issues to focus on are:
  – maximum clock rates and skew,
  – maximum number of clocks,
  – asynchronous clock design,
  – clock/data relationships.
Definitions
Definitions

- The first step in any FPGA design is to decide what clock speed is needed within the FPGA.
- The fastest clock in the design will determine the clock rate that the FPGA must be able to handle.
- The maximum clock rate is determined by the propagation time, P, of a signal between two flip-flops in the design.
- If P is greater than the clock period, T, then when the signal changes at one flip-flop, it doesn't change at the next stage of logic until two clock cycles later.
Definitions

Source: [1]
Definitions

• The *propagation time* is the sum of:
  – the hold time required for the signal to change at the output of the first flip-flop,
  – the delay of any combinatorial logic between stages,
  – the routing delay between stages,
  – the set-up time for the signal going into the flip-flop at the second stage.
Clock Skew

- The clock *skew*, $S$, is the maximum delay from the clock input of one flip-flop to the clock input of another flip-flop.
Clock Skew

- For the circuit to work properly, the skew must be less than the propagation time between the two flip-flops. Otherwise we have **short-path problem**.
- Each clock used in an FPGA design, no matter the rate of the clock, must have low skew.
Clock Skew - reduction

- The short-path problem is created by the existence of an unacceptably large clock skew
- Minimizing the clock skew is the best approach to reduce the risk of short-path problems
- The best way to minimize the clock skew is to use hardware resources like Delay Locked Loop or global clock networks
Clock Skew - reduction

- In cases where designs include multiple clock domains, there may not be enough low-skew global resources in the targeted FPGA for all the external/internal clock signals
- Therefore, regular routing resources and buffers are used to build clock trees for the clock network – possibility of a noticable clock skew
Clock Skew - reduction

• Methods of manual clock skew reduction:
  – *Adding Delay in Data Path*:
    • the data path propagation time must be greater than the clock skew

\[ T_{cq1} + T_{rdq1} + (n \times T_{BUF}) + T_{suq2} > T_{ckq2} \]
Clock Skew - reduction

• Methods of manual clock skew reduction:
  – *Clock Reversing*:
    • the clock signal arrives at the clock port of the sink (receiving) register sooner than the source (transmitting) register
Clock Skew - reduction

• Methods of manual clock skew reduction:
  – *Alternate Phase Clocking* - Clocking on alternate edges:
    • sequentially adjacent registers are clocked on the opposite edges of the clock
Clock Skew - reduction

- Methods of manual clock skew reduction:
  - *Alternate Phase Clocking* - Clocking with two phases:
    - a set of adjacent registers are alternately clocked on two different phases of the same clock
Metastability

• One of the most serious problems associated with multiple clock designs is when two stages of logic are combined using asynchronous clocks.

• Asynchronous logic can create metastable states that can seriously degrade the performance of the design or completely destroy the functionality.
Metastability

• A metastable state is created when the flip-flop's timing requirements (setup and hold times) are violated.

• The resulting output of the flip-flop is unknown, and can make the entire design nondeterministic.

• If one stage of logic asynchronously feeds data to another, it is difficult, if not impossible to meet the set-up and hold-time requirements of the flip-flop.
Metastability

Signal transition occurs after clock edge and minimum $t_H$:
Ball lands on the old data side.

Signal transition meets register $t_{SU}$ and $t_H$:
Ball lands on the new data side.

Signal violates register $t_{SU}$ or $t_H$:
Ball balance at top of hill or takes too long to reach the bottom. Output is metastable and violates $t_{CO}$.

Source: [6]
Metastability

Source: [6]
Metastability

• One of the metastability solution is the double-registering technique.
• Data coming into the first flip-flop is asynchronous with the clock, so the first flip-flop will almost certainly go metastable.
• However, the second flip-flop will never go metastable as long as the length of metastability is less than the period of the clock. (Unfortunately, FPGA vendors rarely publish metastability times, though they are typically less than the sum of the set-up and hold time of the flip-flop.)
Metastability

Source: [5]
Metastability

• If the clock is not too fast to meet normal timing constraints, it is probably not going to propagate metastable states in a circuit shown.

• Even though the output of the first flip-flop can be used as long as all of the paths out go to flip-flops clocked by the same clock, it is generally good practice to use a circuit such as shown to isolate metastability to one short line.

• That way, it is less likely that a future change to the circuit will unintentionally use the metastable line in nonclocked logic.
FPGA clocking resources (Spartan 6)
Clocking resources

• Each Spartan-6 FPGA device includes 16 high-speed, low-skew global clock resources
• Each Spartan-6 FPGA also provides 40 ultra high-speed, low-skew I/O regional clock resources
• These resources are used automatically by the Xilinx tools
Clocking resources

• The Spartan-6 FPGA clock resources consist of:
  – Clock networks:
    • Global clock network driven by global clock multiplexers (BUFGMUX):
      – can multiplex between two global clock sources or be used as a simple BUFG clock buffer.
    • I/O regional clock networks driven by I/O clock buffers (BUFIO2) as well as PLL clock buffers (BUFPLL):
      – used to drive clocks routed only on the I/O regional clock network with much higher performance than the global clock network
Clocking resources

• The Spartan-6 FPGA clock resources consist of:
  – Connections:
    • Global clock input pads (GCLK)
    • Global clock multiplexers (BUFG, BUFGMUX)
    • I/O clock buffers (BUFIO2, BUFIO2_2CLK, BUFPLL)
    • Clock routing buffers (BUFH)
  – Auxiliary blocks:
    • CMT – Clock Management Tiles
    • DCM – Digital Clock Management
      – DLL – Delay Locked Loop
      – DFS – Digital Frequency Synthesiser
      – DPS – Digital Phase Synthesiser
    • PLL – Phase Locked Loop
Global Clocking Infrastructure

Source: [2]
Clock routing buffers (BUFH)
I/O Clocking infrastructure
Clock inputs (GCLK)

- Each Spartan-6 FPGA has:
  - Up to 32 global clock inputs located along four edges of the FPGA.
  - Eight dedicated clock inputs in the middle of each edge of the device.
  - Eight BUFIO2 clocking regions
Clock inputs (GCLK)
Global clock multiplexers BUFGMUX

- Each BUFGMUX primitive is a 2-to-1 multiplexer

- The BUFGMUX multiplexes two clock signals while eliminating any timing hazards by switching from one clock source to a completely asynchronous clock source without glitches
Global clock multiplexers BUFGMUX
Global clock multiplexers BUFG

- The BUFG clock buffer primitive drives a single clock signal onto the clock network.
- The BUFG is essentially the same as a BUFGMUX, without the clock select mechanism.
I/O clock buffers - BUFIO2

- The BUFIO2 takes a GCLK clock input and generates two clock outputs and a strobe pulse

- IOCLK – normal output
- DIVCLK – divided output (by 1,2,3,4,5,6,7,8)
- SERDESSTROBE - clock network output used to drive IOSERDES2
I/O clock buffers - BUFIO2
I/O clock buffers - BUFPLL

- The BUFPLL is intended for high-speed I/O routing to generate clocks and strobe pulses for the SERDES primitives.

- The BUFPLL aligns the SERDESSSTROBE to the IOCLK.
- SERDESSSTROBE is a divided PLLIN signal.
## Available CMT, DCM, and PLL Resources

<table>
<thead>
<tr>
<th>Device</th>
<th>Number of CMTs</th>
<th>Number of DCMs</th>
<th>Number of PLLs</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC6SLX4</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>XC6SLX9</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>XC6SLX16</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>XC6SLX25</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>XC6SLX25T</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>XC6SLX45</td>
<td>4</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>XC6SLX45T</td>
<td>4</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>XC6SLX75</td>
<td>6</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>XC6SLX75T</td>
<td>6</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>XC6SLX100</td>
<td>6</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>XC6SLX100T</td>
<td>6</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>XC6SLX150</td>
<td>6</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>XC6SLX150T</td>
<td>6</td>
<td>12</td>
<td>6</td>
</tr>
</tbody>
</table>
DCM Summary

• Digital Clock Managers (DCMs) provide advanced clocking capabilities to Spartan FPGA applications.
• DCMs integrate advanced clocking capabilities directly into the global clock distribution network.
DCM Summary

• DCM solves a variety of common clocking issues, especially in high-performance, high-frequency applications:
  – Eliminates clock skew, either within the device or to external components, to improve overall system performance and to eliminate clock distribution delays.
  – Phase shifts a clock signal, either by a fixed fraction of a clock period or by incremental amounts.
DCM Summary

– Multiplies or divides an incoming clock frequency or synthesizes a completely new frequency by a mixture of static or dynamic clock multiplication and division
– Conditions a clock, ensuring a clean output clock with a 50% duty cycle.
– Filters clock input jitter
DCM Summary

– Mirrors, forwards or rebuffers a clock signal, often to deskew and converts the incoming clock signal to a different I/O standard. For example, forwarding and converting an incoming LVTTL clock to LVDS.
– Free-running oscillator
– Spread-spectrum clock generation
# DCM Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>DCM Signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCMs per device</td>
<td>Four to 12 DCMs, depending on device size. See Table 2-1.</td>
<td>All</td>
</tr>
<tr>
<td>Clock input sources</td>
<td>GCLK input, BUFG output, cascaded DCM or PLL output (within the same CMT)</td>
<td>CLKin</td>
</tr>
<tr>
<td>Frequency synthesizer output</td>
<td>Multiply CLKin by the fraction (M/D) where M = [2..256], D = [1..256] when using the DCM_CLKGEN primitive</td>
<td>CLKFX, CLKFX180</td>
</tr>
<tr>
<td>Clock divider output</td>
<td>Divide CLKin by 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 9, 10, 11, 12, 13, 14, 15, or 16</td>
<td>CLKDV</td>
</tr>
<tr>
<td>Clock doubler output</td>
<td>Multiply CLKin frequency by 2</td>
<td>CLK2X, CLK2X180</td>
</tr>
</tbody>
</table>
## DCM Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock conditioning, duty-cycle correction</td>
<td>Always provided on most outputs.</td>
<td>All</td>
</tr>
<tr>
<td>Quadrant phase-shift outputs</td>
<td>0° (no phase shift), 90° (¼ period), 180° (½ period), 270° (¾ period)</td>
<td>CLK0, CLK90, CLK180, CLK270</td>
</tr>
<tr>
<td>Half-period phase-shift outputs</td>
<td>Output pairs with 0° and 180° phase shift, ideal for DDR applications</td>
<td>CLK0, CLK180, CLK2X, CLK2X180,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CLKFX, CLKFX180</td>
</tr>
<tr>
<td>Variable phase-shifting</td>
<td>Allows DCM clock outputs to adjust phase shift during operation</td>
<td>PSEN, PSINCDEC, PSCLK, PSDONE</td>
</tr>
<tr>
<td>General purpose DCM operation indicators</td>
<td>Number of DCM clock outputs connected to general-purpose interconnect</td>
<td>STATUS, LOCKED</td>
</tr>
</tbody>
</table>
# DCM in Xilinx families

<table>
<thead>
<tr>
<th>Function</th>
<th>Virtex-5 FPGAs(1)</th>
<th>Spartan-3E and Extended Spartan-3A FPGAs</th>
<th>Spartan-6 FPGAs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design primitive</td>
<td>DCM_BASE</td>
<td>DCM_SP</td>
<td>DCM_SP</td>
</tr>
<tr>
<td></td>
<td>DCM_ADV</td>
<td></td>
<td>DCM_CLKGEN</td>
</tr>
<tr>
<td>Distinct DLL operating frequency ranges</td>
<td>Two: Low and High</td>
<td>One</td>
<td>One</td>
</tr>
<tr>
<td>Distinct DFS operating frequency ranges</td>
<td>Two: Low and High</td>
<td>One</td>
<td>One</td>
</tr>
<tr>
<td>Variable phase-shift increment or decrement unit</td>
<td>$1/256$th of CLKIN period (degrees)</td>
<td>DCM_DELAY_STEP between 15 to 35 ps (time)</td>
<td>DCM_DELAY_STEP See the Spartan-6 FPGA data sheet</td>
</tr>
<tr>
<td>DCM $V_{CCAUX}$ voltage supply</td>
<td>2.5V</td>
<td>2.5V or 3.3V</td>
<td>2.5V or 3.3V</td>
</tr>
<tr>
<td>Jitter reduction with expense of phase alignment</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Dynamic programming of frequency multiplication and division</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Generation of spread-spectrum clocks</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
DCM Block Diagram

Source: [1]
Delay-Locked Loop

- The Delay-Locked Loop (DLL) provides an on-chip digital deskew circuit that effectively generates clock output signals with a net zero delay.
- The deskew circuit compensates for the delay on the clock routing network by monitoring an output clock, from either the CLK0 or the CLK2X outputs.
Delay-Locked Loop

• The DLL effectively eliminates delay from the external clock input port to the individual clock loads within the device.
• The well-buffered global network minimizes the clock skew on the network caused by loading differences.
Delay-Locked Loop - functioning

- A DLL in its simplest form inserts a variable delay line between the external clock and the internal clock.
- The clock tree distributes the clock to all registers and then back to the feedback pin of the DLL.
- The control circuit of the DLL adjusts the delays so that the rising edges of the feedback clock align with the input clock.
DLL Block Diagram - functioning
Digital Frequency Synthesizer

• The Digital Frequency Synthesizer (DFS) provides a wide and flexible range of output frequencies based on the ratio of two user-defined integers, a multiplier (CLKFX_MULTIPLY) and a divisor (CLKFX_DIVIDE).
• The output frequency is derived from the input clock (CLkin) by simultaneous frequency division and multiplication.
• The DFS feature can be used in conjunction with or separately from the DLL feature of the DCM.
Phase Shift

• The Phase Shift (PS) controls the phase relations of the DCM clock outputs to the CLKIN input.
• PS can be used in either fixed phase-shift or variable phase shift modes
• The phase of all nine DCM clock output signals are shifted by a fixed fraction of the input clock period when using the fixed phase shift.
• The PS also provides a digital interface for the FPGA application to dynamically advance or retard the current shift value, called variable phase shift.
Status Logic

• The status logic indicates the current state of the DCM via the LOCKED and STATUS[0] output signals.
• The LOCKED output signal indicates whether the DCM outputs are in phase with the CLKin input.
• The STATUS output signals indicate the state of the DLL and PS operations.
DCM config
LOCKED output behaviour
Spread Spectrum Generation

• Spread-spectrum clock generation (SSCG) is widely used by manufacturers of electronic devices to reduce the spectral density of the electromagnetic interference (EMI) generated by these devices.

• Typical (but expensive!) solutions for meeting EMC requirements involve adding expensive shielding, ferrite beads, or chokes.
Spread Spectrum Generation

- SSCG spreads the electromagnetic energy over a large frequency band to effectively reduce the electrical and magnetic field strengths measured within a narrow window of frequencies.
- The peak electromagnetic energy at any one frequency is reduced by modulating the SSCG output.
Spread Spectrum Generation

• The SSCG commonly defines the following parameters:
  – Frequency deviation, that is the percentage of the input frequency
  – Modulation frequency
  – Spread type: up/down/center
  – Modulation profile, for example, the shape of the triangle

• Typically: 75 MHz (input frequency), ±2.0% center spread, and 75 KHz triangular modulation.
# Spread Spectrum Generation

<table>
<thead>
<tr>
<th>Spread Spectrum Values</th>
<th>Fixed Spread-Spectrum Modes</th>
<th>Soft Spread-Spectrum Modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPREAD_SPECTRUM values</td>
<td>CENTER_LOW_SPREAD</td>
<td>VIDEO_LINK_M0, VIDEO_LINK_M1, VIDEO_LINK_M2</td>
</tr>
<tr>
<td>Additional logic</td>
<td>None</td>
<td>SOFT_SS</td>
</tr>
<tr>
<td>Modulation profile</td>
<td>Triangular</td>
<td>Triangular</td>
</tr>
<tr>
<td>Spread direction</td>
<td>Center</td>
<td>Down</td>
</tr>
<tr>
<td>Spread range</td>
<td>Fixed</td>
<td>See Figure</td>
</tr>
<tr>
<td>( F_{MOD} )</td>
<td>( F_{IN}/1024 )</td>
<td>See Figure</td>
</tr>
<tr>
<td>CLKFX_MULTIPLY</td>
<td>2–32</td>
<td>7</td>
</tr>
<tr>
<td>CLKFX_DIVIDE</td>
<td>1–4</td>
<td>2, 4</td>
</tr>
<tr>
<td>DCM_CLKGEN programming ports</td>
<td>N/A</td>
<td>PROGCLK, PROGEN, PROGDATA, PROGDONE</td>
</tr>
</tbody>
</table>
PLL
PLL Introduction

• The main purpose of PLLs is:
  – to serve as a frequency synthesizer for a wide range of frequencies
  – to serve as a jitter filter for either external or internal clocks in conjunction with the DCMs of the CMT (Clock Management Tile).
PLL Block Diagram

Source: [1]
PLL Block Diagram

• D is a programmable counter for each clock input
• Phase-Frequency Detector (PFD) compares both phase and frequency of the input (reference) clock and the feedback clock
  – The PFD is used to generate a signal proportional to the phase and frequency between the two clocks.
• Charge Pump (CP) and Loop Filter (LF) are used to generate a reference voltage to the Voltage Controlled Oscillator (VCO).
PLL Block Diagram

- The PFD produces an up or down signal to the charge pump and loop filter to determine whether the VCO should operate at a higher or lower frequency.
- When VCO operates at too high of a frequency, the PFD activates a down signal, causing the control voltage to be reduced decreasing the VCO operating frequency.
- When the VCO operates at too low of a frequency, an up signal will increase voltage.
PLL Block Diagram

• The VCO produces eight output phases.
• Each output phase can be selected as the reference clock to the output counters
• The counter M controls the feedback clock of the PLL allowing a wide range of frequency synthesis
PLL Primitives

Source: [1]
Thank you for your attention
References

[1] Tim Behne, „FPGA Clock Schemes”
[6] https://nepp.nasa.gov/mapld_2009/talks/Posters/Landoll_David_mapld09_pres_1.ppt#1019,12,Clock Domain Crossings Guaranteed to Cause Metastability