Optimal creation of clock divider for various frequency

Lakshmi M
Department of ECE *organization (VTU)*
East Point College of Engineering and Technology*(VTU)*
Bengaluru,India

Dr.Chandrappa D N
Associate Professor
Department of ECE
East Point College of Engineering and Technology
Bengaluru,India

Dr.Anita R
Associate Professor
Department of ECE
East Point College of Engineering and Technology
Bengaluru,India

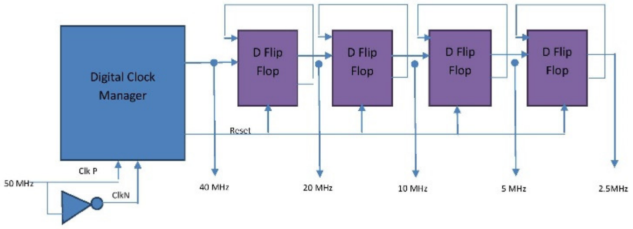
Dr.Yogesh G S
Vice Principal & HOD
Department of ECE
East Point College of Engineering and Technology
Bengaluru,India

***Abstract*— Frequency dividers are important in frequency synthesizers that utilize Phase Locked Loops (PLL's) circuits. The implementation of double modulus presales improves the flexibility of designs in synthesizers. Choosing an appropriate division ratio is influenced by the synthesizer's channel spacing and frequency range. Various methods for division in electronic systems are available including injection locked frequency dividers (ILFD)circuits, complementary ILFD circuits, flip flop-based divider circuits, dual modulus divider circuits, and modular dividers circuits. Each of these methods comes with its own set of benefits and drawbacks, such as jitter, less frequency tuning range, larger circuit size due to additional L-C tank circuits, high power consumption, and low amount of quality factor. This research addresses specific challenges related to clock dividers and proposes a novel design that employs a multiple digital frequency divider using D flip flops. The architectural framework relies on a phase-shifting mechanism with a D flip flop that efficiently regulates the division ratio. The study incorporates a preliminary phase-shifting melody in combination with a Digital Clock Manager (DCM). The auto-tuning procedure outlined aims to adjust the phase difference between two differential clock signals. By deliberately inducing metastability in one or more flip flops, this approach leverages a digital clock manager in a clock divider to counteract metastability effects and minimize jitter across a range of tuning frequencies. Additionally, it is notable that the logic size and power consumption necessary for this design are considerably reduced.**

Keywords—DCM , metastability , jitter , Xilinx , modelim

I. *INTRODUCTION*

methodology



The Digital Clock Manager (DCM) receives the input clock of the proposed design, which begins at 50 Mhz and is converted to differential signals of Clk_P and Clk_N. DCM simulations indicate that its output will have dropped to 40MHz. With this 40MHz input clock, a proposed design for D Flip Flop-based clock divider circuits is provided. In this instance, each D flip flop input will be created using the feedback input of the Qbar output, and the Q output will serve as the input for the next D flip flop clock input. We will repeat the same procedure, and it was demonstrated that we can create five different clock frequencies using the previously specified design.

Frequency Divider:

Fig. 1 shows a basic flip-flop based divider. As can be seen in Fig. 1, at Q1 output, the clock signal is divided by two. Then, at the second DFF, Q1 is divided by two, and the total division factor is four, resulting in Fin/4. Table 1 displays this structure's truth table. Table 1 suggests that there are two logical zero states and two logical one states in the division factor of 2, or Q1. For Q2 and the division by 4 factor, the same conclusion is drawn. Four logical zero states and four logical one states are what it ought to have. The division factor is shown by the number of zero and one states, which is an important aspect of them. For instance, the divided signal includes three logical zero states and three logical one states if the input signal is to be divided by a factor of 3. These numbers relate to an input signal that exists in two states: zero and one. The truth table is broken down by components 3, 6, and 7 in Table 2.

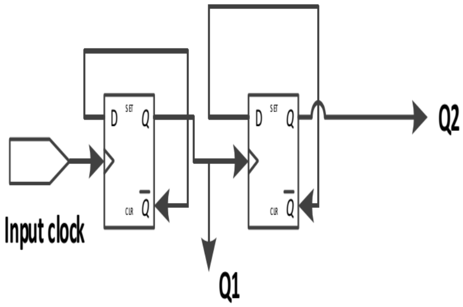


Figure 1: Clock divider divided into two and four parts

input	output	
clock	Q1	Q2
0	0	0
1	0	0
0	1	0
1	1	0
0	0	1
1	0	1
0	1	1
1	1	1

Table 1 : Truth table of Fig.1

Inputs			Outputs			Continued					
A	B	C	Q3	Q6	Q7	A	B	C	Q3	Q6	Q7
0	0	0	0	0	0	1	1	1		1	1
0	0	1	0	0	0	0	0	0		1	1
0	1	0	0	0	0	0	0	1		1	1
0	1	1	1	0	0	0	1	0		1	1
1	0	0	1	0	0	0	1	1		1	1
1	0	1	1	0	0	1	0	0			1
1	1	0		1	0	1	0	1			1

Table 2: Divide by Three, Six, and Seven Truth Table

D Flip Flop

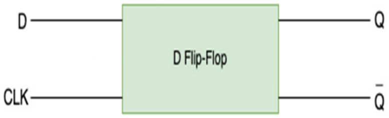


Fig 2 : D flipflop

Flip Flop is an electrical device, or more accurately, a type of memory element, that has the capacity to store a single bit of data. There are two states for a flip flop: "SET" and "RESET." Binary values 0 and 1 are used to represent those states. Until it receives a signal to switch to the opposite state, the flip flop stays in its current state. The flip flop may be "triggered" to change states by a clock or pulse signal.

A single bit of data can be stored in an electronic device called a D flip flop, often referred to as a "data flip flop" or "delay flip flop."Asynchronous or synchronous D flip flops are available. The clock single needed for D flip flops in their synchronous form, but not in their asynchronous form. The D flip flop is controlled by a clock input and two inputs: data and data. The data is moved to the flip flop's output when the clock input is high, and the output is retained in its initial state when the clock input is low.

D flip flop has two outputs (Q and Q') and one input (D)

The following is how D Flip Flop functions fundamentally:

The flip flop retains its present state and ignores the D input when the clock signal is low.

The flip flop samples and saves the D input when the clock signal is high.

The flip flop's Q output displays the value that was previously input into the D input.

Q will be 0 if D = 0.

Q will be 1 if D = 1.

The flip flop's Q output is a complement to its Q' output.

Q' will be 1 if Q = 0.

Q' will be 0 if Q = 1.

Clock Divider:

A digital circuit or component that receives an input clock signal and outputs a lower-frequency clock signal is called a clock divider, sometimes referred to as a frequency divider. It is a fundamental component of digital electronics and is frequently employed in a variety of applications where lowering the clock frequency is necessary for timing or synchronizing disparate system components that are operating at different speeds.

Typically, a clock divider operates as follows:

1. **Clock input**: An input clock signal, sometimes referred to as the reference clock, is fed into the clock divider. This input clock signal is produced by a crystal oscillator or another source and has a greater frequency.

2**Division Ratio**: The division ratio, which establishes the amount by which the input clock frequency will be split, is specified by the user or the circuit designer. For instance, if the division ratio is set to 10, and the input clock is 100 MHz, the resultant output clock frequency is 10 MHz.

3. **Divide-by-N Counter**: A divide-by-N counter is usually located inside the clock divider circuit. The division ratio is represented by the value of N. When the counter reaches N counts, it produces an output pulse by counting the rising or falling edges of the input clock signal. The frequency is effectively divided by N by this pulse.

4. **Output Clock**: The signal that is obtained by dividing the input clock is known as the output clock. The division ratio determines its lower frequency.

Modelsim - Altera

Assumptions

presumptively you know how to utilize your operating system. Additionally, you must be used it Microsoft windows 2000/XP, Open windows, OSF/Motif, CDE, KDE, or GNOME. Additionally, we presume that you are conversant in the language (VHDL, Verilog, etc.) used to write your design and/or test bench. This article is not intended to assist with learning HDL concepts and techniques, even if ModelSim TM is a great tool for this purpose.

Basic Simulation Flow:

The fundamental procedures for modeling a design in ModelSim are depicted in the diagram below.

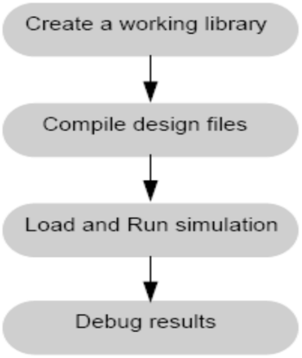


Figure 3: Basic Simulation Flow

Overview Lab for Basic Simulation Flow [Building the Working Library]

Every design in ModelSim is collected into a library. The usual way to launch a new simulation in ModelSim is to create a functioning library called "work." This is because the compiler uses "work" as the default library name when determining where to save compiled design units.Putting Your Design Together : compiling Once the working library has been established, put your design units into it. All supported platforms support the ModelSim library format. Without requiring you to recompile your design, it can emulate your design across all platforms. In order to load the simulator with your design and run it after the design has been compiled, invoke the simulator on a configuration, entity/architecture pair, or top-level module (Verilog) (VHDL).

The simulation time is set to zero, assuming the design loads correctly, and you enter a run command to start the simulation.

Examining Your Outcomes: debugging

If the outcomes are not what you were hoping for, you can investigate the root of the issue with ModelSim's extensive debugging environment.

Project Flow

A project is a way to gather data about an HDL design that is being tested or specified. ModelSim projects are not required, although they can make using the program easier. They can also be used to organize files and establish simulation parameters. The essential processes for simulating a design within a ModelSim project are depicted in the following diagram.

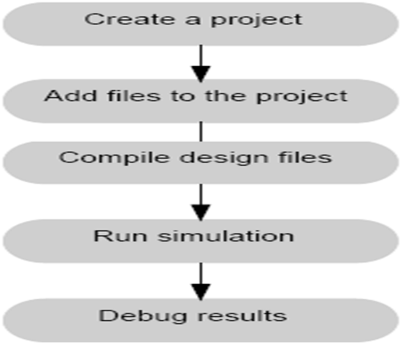


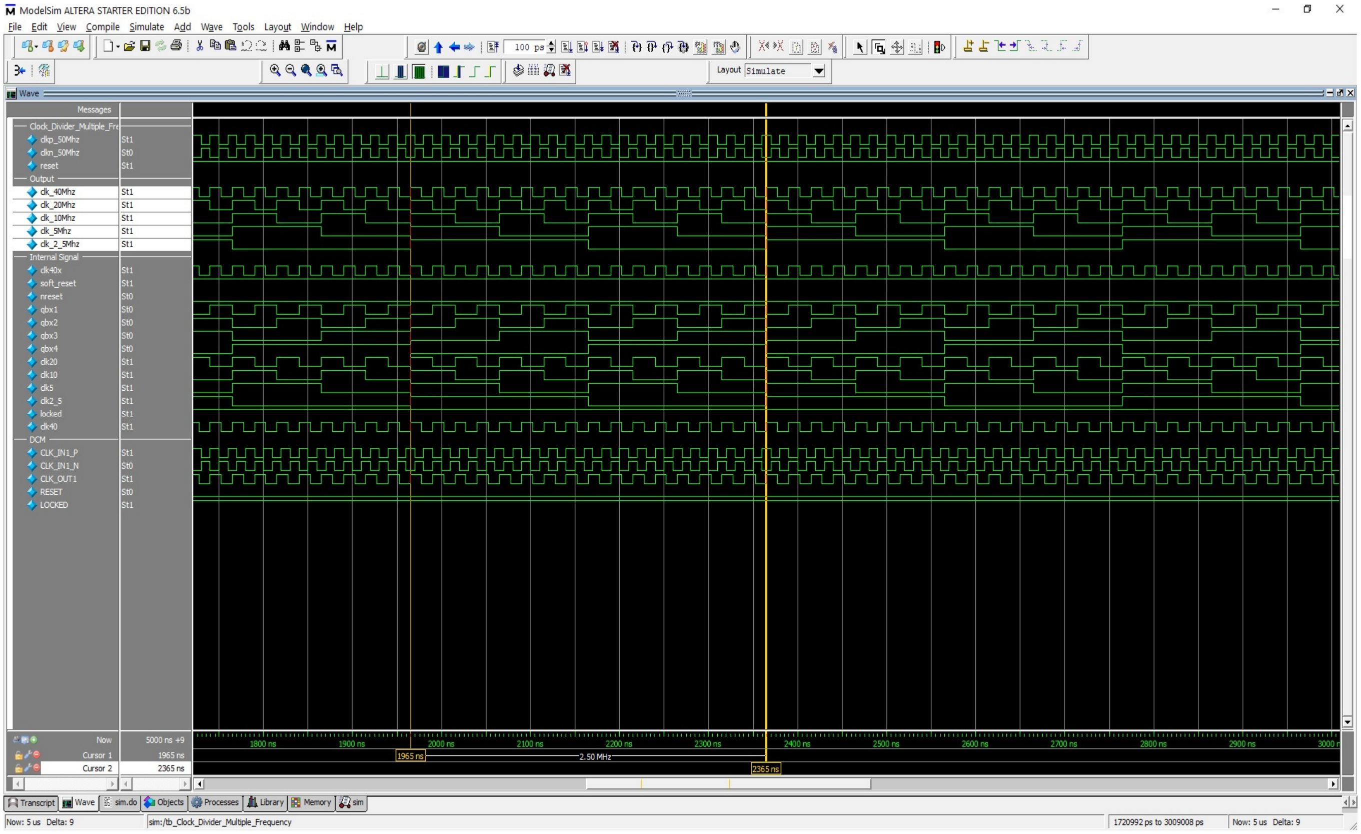
Figure 4: Project flow is shown

Debugging tools:

Numerous tools for design analysis and debugging are provided in ModelSim. In subsequent lessons, several of these instruments are examined.

- By employing project
- Using various libraries in your work.
- Placing breakpoints and advancing by the source code
- Observing wave patterns and recording durations
- Observing and starting up recollections
- Generating stimulation using the Waveform Editor.
- Making simulation automated

Result


ModelSim ALTERA STARTER EDITION 6.5b
File Edit View Compile Simulate Add Wave Tools Layout Window Help
Layout Simulate
Wave
Transcript Wave sim.do Objects Processes Library Memory sim
1800 ns 1900 ns 2000 ns 2100 ns 2200 ns 2300 ns 2400 ns 2500 ns 2600 ns 2700 ns 2800 ns 2900 ns 3000 n
1965 ns
2.50 MHz
2365 ns

```
Messages
— Clock_Divider_Multiple_Fre
    clkp_50Mhz      St1
    clkn_50Mhz      St0
    reset           St1
— Output
    clk_40Mhz       St1
    clk_20Mhz       St1
    clk_10Mhz       St1
    clk_5Mhz        St1
    clk_2_5Mhz      St1
— Internal Signal
    clk40x          St1
    soft_reset      St1
    nreset          St0
    qbx1            St0
    qbx2            St0
    qbx3            St0
    qbx4            St0
    clk20           St1
    clk10           St1
    clk5            St1
    clk2_5          St1
    locked          St1
    clk40           St1
— DCM
    CLK_IN1_P       St1
    CLK_IN1_N       St0
    CLK_OUT1        St1
    RESET           St0
    LOCKED          St1

Now        5000 ns +9
Cursor 1   1965 ns
Cursor 2   2365 ns

Now: 5 us  Delta: 9    sim:/tb_Clock_Divider_Multiple_Frequency    1720992 ps to 3009008 ps    Now: 5 us  Delta: 9
```

Conclusion:

Finally Concept of DCM (Digital Clock Manager) reduces the effects of metastability and jitter across multiple tuning frequencies. Furthermore, it is important to notice that the logic size and power consumption required for the operation are significantly reduced.

References:

[1] “A 1.8V, Sub-mW, Over 100% Locking Range, Divide-by-3 and 7 Complementary-Injection-Locked 4 GHz Frequency Divider” YungChung Lo, Hsien-Pu Chen, Jose Silva-Martinez and Sebastian Hoyos

[2] “90Nm – CMOS based Low-Power 256/257 Dual Modulus Percale”, Gore Shukracharya S., AkotkarRohitKishor, JadhavGeeta A.

[3] Y.-F. Kuo and R.-M. Weng, “5 GHz low power frequency synthesizer with dual-modulus counter,” IET Circuits, Devices and Systems, vol. 3, no. 6, pp. 365 –375, Dec. 2009

[4] X. Lei, Z. Wang, K. Wang, and X. Wang, “A high speed low power pulse swallow frequency divider for drm/dab frequency synthesizer,” in International Conference on Wireless Communications Signal Processing, 2009, Nov. 2009, pp. 1 –4.

[5] L. Zhang, B. Chi, Z. Wang, H. Chen, and E. Wu, “A low power 440- mhz pulse-swallow-divider combination synchronization-asynchronism hybrid frequency divider,” in IEEE International Midwest Symposium on Circuits and Systems, 2006, vol. 2, Aug. 2006, pp. 566 –568

[6] “A 5/6-bit Multi-Modulus Frequency Divider in 0.13um CMOS Technology” SitiMaisurah M. H., Member, IEEE, NazifEmran F., Norman FadhilIdham M. and A. I. Abdul Rahim

[7] H. R. Rategh and T. H. Lee, “Superharmonic injection-locked frequency dividers,” IEEE J. Solid-State Circuits, vol. 34, pp. 813–821, 1999.

[8] H. Zhang, H. gang Yang, J. Zhang, and F. Liu, “High-speed programmable counter design for pll based on a delay partition technique,” in IEEE International Symposium on Radio-Frequency Integration Technology, 2009, Dec. 2009, pp. 100 –103

[9] R. HOSSAIN, L. D. WRONSKI AND A. ALBICKI, “LOW POWER DESIGN USINGDOUBLE EDGE TRIGGERED FLIP-FLOPS,” IEEE TRANS. VLSI SYSTEMS, VOL.2, NO.2, PP.261-265, JUNE 1994