ISSN:2229-6107

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY

E-mail :
editor.ijpast@gmail.com
editor@ijpast.in

www.ijpast.in

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY
IJPAST

ISSN 2229-6107 www.ijpast.in

Vol 11, Issue 4. Nov 2021

Frequency Synthesis with an FPGA Using a Partial Reconfiguration-Based Method

Y. Pratap Kumar[1], Sd. Shakeera Sulthana[2], Y. Santhosh Kumar[3], K. Suresh[4] ,

Abstract

Dedicated FPGA clock managers are increasingly required to get clocks for high-speed, high-density devices. The variety of uses they may serve is directly proportional to their adaptability and programmability. Modern FPGA designs, like the Xilinx Vertex Series, provide partial and dynamic reconfiguration at run time. Because the FPGA fabric's configuration data is mutable at runtime, the system may be tailored to the requirements of an application by swapping out individual pieces of an existing hardware design. A novel method of Digital Frequency Synthesis using FPGA clock management is described in this paper. In the suggested approach, a DCM primitive's frequency synthesis is performed by way of the Fabric's reconfigurable module and its Dynamic Reconfiguration Port (DRP). The design draws attention to Partial Reconfiguration based design techniques, which allow a DCM's clock frequency to be reconfigured dynamically to meet the changing requirements of a running application. On-the-fly frequency and phase adjustments are both quick and accurate. First, a Virtex-5 FPGA board is used to mimic the proposed design before it is really developed and tested in the lab.

Key words :

FPGA, Digital Clock Manager, PR Flow, Reconfiguration Module, Virtex-5, Software Defined Radio.

Introduction

Field Programmable Gate Arrays (FPGAs) are growing in size and complexity, making clock distribution a critical concern. Keeping high clock speeds has gotten more challenging as the complexity and size of integrated circuits continue to increase at an exponential pace. The addition of clock managers to FPGAs [1-4], [17] helps to address clock distribution issues and enhances the devices' adaptability and usefulness. Designers may adjust clock synchronization between components operating at various frequencies by multiplying and dividing the clock frequencies on and off chip. Software Defined Radio (SDR) is one example of a waveform-based application; in this case, the waveform design engineer may need multiple frequencies at runtime depending on the waveform of the chosen communication standard [1].SDR implementation is made possible by the unique characteristics of FPGAs. High-speed interfaces, embedded DSP blocks, phase-locked loops, and

Assistant Professor 1[23,4,5],
Mail Id : prathapkumar231@gmail.com, Mail Id : shakirasulthana.sayed@gmail.com Mail
Mail id : santhonani@gmail.com Mail Id : sssureshk1@gmail.com
Department of EEE,
Swarna Bharati Institute of Science and Technology (SBIT),
Pakabanda Street,Khammam TS, India-507002.

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY
IJPAST

ISSN 2229-6107 www.ijpast.in

Vol 11, Issue 4. Nov 2021

large memory capacities are all examples. The military also greatly benefits from the system's flexibility to execute updates and reconfiguration of waveforms in the field. With the rise of Software Defined Radio (SDR) and Cognitive Radio (CR), framework compliance with low power design and Dynamic Reconfigurability is becoming more important [4].

The Circuit Manager for Digital Clocks

FPGAs with built-in DCMs are superior than other options because they eliminate clock management issues in high-speed design. DCMs simplify design and reduce the cost of development and testing for systems requiring clocking services such as frequency synthesis, phase alignment, board disked, compliance with source synchronous interface standards, clock-data synchronization, and clock switching. As can be seen in Figure 1(a), it may function as a clock delay locked loop, a digital frequency synthesizer, a digital phase shifter, and a digital spread spectrum. Input clock frequencies may be multiplied, divided, and phase-shifted using DCM's clock management system. By connecting the CLKFB input to a node on the global clock network powered by the CLK0 output of the same DCM, on-chip synchronization may be established. The suggested design makes use of the DCM_ADV Primitive, seen in figure 1(b), which has access to all DCM functionalities. Many on-chip DCM circuits with zero propagation delay and lowest jitter performance[5-6], [17] are provided by the Xilinx-Virtex-5 utilized in this application.

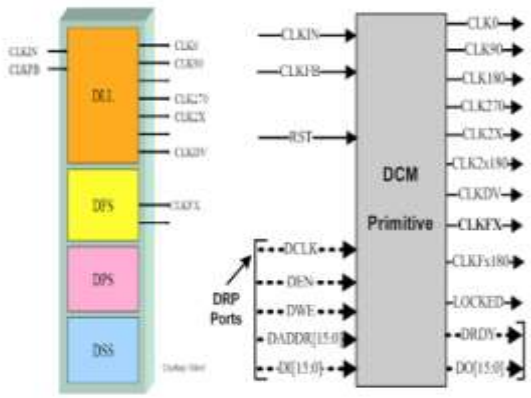

Figure 1 (a) DCM Block; (b) DCM Primitive Block

DFS stands for "digital frequency synthesizer."

The suggested architecture makes use of the DFS function of the clock management. The DCM's clock multiplication and division operations allow for a large variety of output clock frequencies to be synthesized from a single input clock [7]. The input frequency may be multiplied, divided, or even multiplied and divided in a wide variety of ways, all of which are made possible by the Frequency Synthesizer. The CLKFX and CLKFX180 outputs, together with the CLKFX_MULTIPLY and CLKFX_DIVIDE properties, allow for a frequency synthesizer in which the CLKIN input may be any multiple or division. If feedback is sent to the DLL's CLKFB input, then every CLKFX_MULTIPLY cycle of CLKFX and every CLKFX_DIVIDE cycle of CLKIN will result in CLKFX and CLKIN being in phase with one another. This equation describes CLKFX's frequency.

$$CLK\,FX = F_{CLKIN} * CLKFX\_Multiply(M) / CLKFX\_DIVIDE(D)$$

The CLKFX and CLKFX180 outputs are fully compatible with each other. The CLKFX180 frequency is the CLKFX frequency shifted by 180 degrees. The duty cycle of CLKFX and CLKFX180 is a constant 50%.

DRP stands for "dynamic reconfiguration port."

As shown in Figure 1(b), the DCM's clock output frequency may be dynamically adjusted through the port labelled "Dynamic Reconfiguration Port" [7]. The initial DCM parameters may be updated via the DRPs without having to feed a fresh bit stream into the FPGA. To alter the presently configured phase shift, frequency, or frequency mode, Dynamic Reconfiguration allows for the modification of DCM properties. Each functional block has a dynamic reconfiguration port built in in, which makes setup a breeze. The clock outputs may be adjusted on the fly while the design is in operation. Variables such as frequency, division by, and multiplication by may all be adjusted on the fly.

Techniques for performing a partial reorganization

When an FPGA supports partial reconfiguration (PR), only a subset of its logic may be changed after initial setup. Partial Dynamic Reconfiguration (PDR) [8-9],[17] is a method that has become available as a result of advancements in FPGA technology that enable the designer to update/reconfigure just a certain area of the internal structure of the FPGA at run-time. By reusing and reconfiguring hardware cores (IP blocks), PDR makes it feasible to construct dynamic systems that

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY
IJPAST

ISSN 2229-6107 www.ijpast.in

Vol 11, Issue 4. Nov 2021

respond to the requirements of the application in real time. This keeps the FPGA functioning without jeopardizing the security of the applications using the sections of the FPGA that are not being changed. When doing a partial reconfiguration, there are two primary methods [8]: Partial Reconfiguration Based on Modules: Modular Design Flow is the foundation for this approach. Reconfigurable modules (RM) are the building blocks from which the whole design may be assembled. A Reconfigurable Partition (RP) is a block of computer memory in which reconfigurable modules (swappable logic) are stored. Partially reconfiguring Xilinx devices is where this technique shines. Modifying a design slightly and then creating a new bit stream based on the differences between the two designs is difference-based partial reconfiguration.

suggested Experimental Layout

In this study, we offer a framework for doing such dynamic reconfiguration of Digital Clock Managers through the Partial reconfiguration method. This method of clock scaling requires careful planning and construction of the system. Since it is not possible to install a DCM primitive in the Reconfigurable zone of an FPGA, this particular primitive was instantiated in the Static region [3, 4]. Partial Reconfiguration is required to build implementations of the same system with various DCM setups. Two key parts, the DCM module in the static zone and the DCM controller module in the reconfigurable region, make up the logic design approach provided here. The RTL schematics of the design components are shown in Figure 2. Using the Xilinx Virtex-5 board ML506 and the XC5VSX50T- FFG1136 device, the suggested architecture for run-time Reconfigurability of DCM was developed and confirmed.

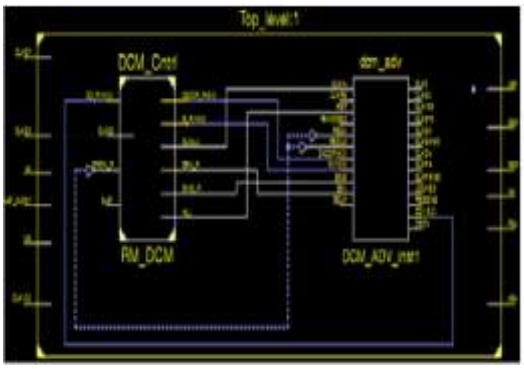

Figure 2 RTL schematic Diagram

The following is the established order for the experimental particulars: The Modules and their hierarchical structure are described in Section 5.1. The outcomes of Design and Simulation are discussed in Section 5.2. The implementation procedure based on Partial Reconfiguration is presented in Section 6. The acquired findings and their quantitative comparison are discussed in Section 7. The implementation overview and Future design considerations are separated into their own sections (8 and 9).

Module and Organizational Structure

The hierarchical netlist is shown in Figure 3. The Top and DCM modules are part of the design's static zone, which means that they continue to function normally when the other modules are being changed [10].

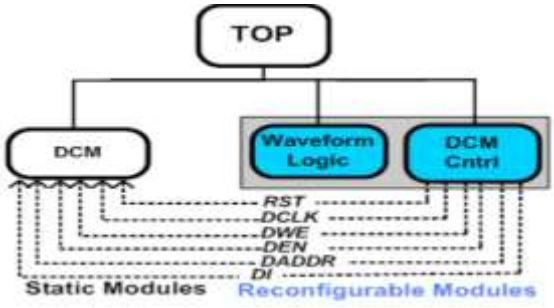

Figure 3 Design Hierarchy Flow

The DCM component of the Static Region's reconfiguration is managed by the DCM_Cntrl module through the DRP port interface. The top-level design is synthesized in sections, with each section later being included into Xilinx's Plan ahead tool. Its unique features include the generation of partial reconfiguration files, which are required for partial reconfiguration [9], and the designer's ability to actualize a modular idea with specified placement limitations.

Efforts Made in Design and Simulation

The architecture used here utilizes both the FPGA's (a) Dynamic reconfiguration port of DCM and (b) Partial Reconfiguration (PR) flow technologies. For Glitch-free and timing-accurate operation, the DCM control Logic for reconfigurable Partition has been built in VHDL utilizing the State machine-based method. Before putting the suggested segmented design into hardware, we used Xilinx ISE software to do a behavioural simulation of the proposed DRP control logic. The circuit is emulated in a Xilinx simulator from VHDL code. The suggested architecture is implemented when the design of the Digital clock management circuit

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY
IJPAST

ISSN 2229-6107 www.ijpast.in

Vol 11, Issue 4. Nov 2021

has been confirmed. Using the readily accessible FPGA templates, we additionally validate the DCM block shown in Figure 1. Frequency Synthesizer output CLKFX from the DCM is the primary focus of the design implementation. For clock synthesis, you may use the multiplier and divider values specified in the vendor's datasheet [7]. The procedures below are part of the simulation result displayed in Figure 4 for reconfiguring the DCM through its DRP port. Here is how the process will unfold:

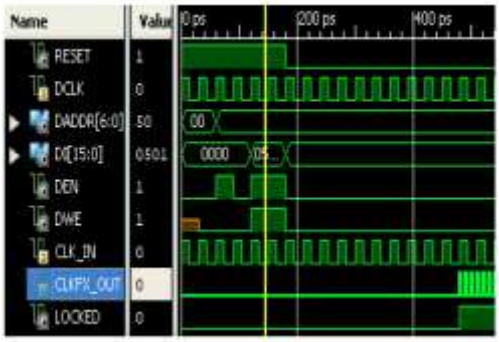

Figure 4 Simulation results of DRP Logical Block

Step 1: The DCM must be held in reset by activating input RESET while changing the M/D values.
Step 2: The rising edge of DCLK signal is the timing reference for all the other port signals.
Step 3: This signal DEN enables read and write port operations. It should only be pulsed for one DC cycle.
Step 4: When DWE and DEN is high, it enables a write operation to the port .It should only be pulsed one DCLK cycle.
Step 5: The value on DADDR bus specifies the individual cell that is written or read on the next cycl DCLK. The address is presented in the cycle that DEN is active.
Step 6: The value on DI bus is the data that is written to the addressed cell.
Step 7: When LOCKED signal is asserted, indicates that clock is generated and stable.

. Module-based Implementation vs.

Partial-Reconfiguration-Based Implementation

The Partial Reconfiguration capability of the FPGA is used in this unique strategy. This method of clock scaling requires very particular system architecture. Since it is not possible to insert a DCM primitive in the Reconfigurable zone of an FPGA, one of these primitives was instantiated in the Static region [12–14]. Partial Reconfiguration is required to build implementations of the same system with various DCM setups. Floor planning, physical constraints, and partial bit files are prepared with the help of the Plan Ahead tool, and the design is synthesized using Xilinx ISE. The PR flow [8, 15] is used to split the system into fixed and dynamic sections, as shown in Figure 5(a). Depending on the use case, different bit files will be created for each RP. Downloading one of several partial bit files—Frequency_LF.bit, Frequency _HF.bit, or Frequency_VHF.bit—based on the relevant Waveforms application modifies the functionality provided in Reconfigurable Partitions.

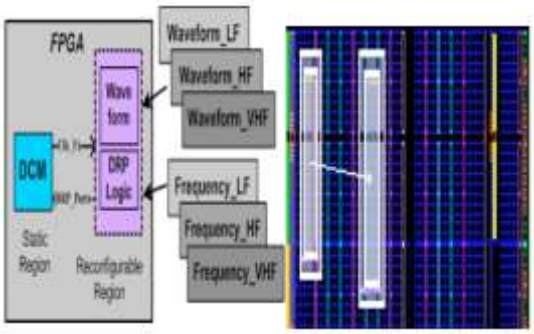

Figure 5 (a) Reconfigurable Partition Blocks; (b) Pblocks of Reconfigurable Partition

Figure 5(b) depicts the partitioning of the FPGA fabric into the physical design reconfigurable blocks (Pblock). For each Pblock, partial bit files were created after considering various configurations [8, 15].

Implementation Based on Distinctions

In order to produce a partial bit stream containing just the differences between the two designs, the Difference Based design flow requires the Full Bit file and the updated Native Circuit Description, NCD file as inputs. A new NCD file was created so that the internal clock synthesis registers could be modified in an FPGA editor. This design process works well when only small adjustments need to be made and there is no need to reorganize any major functionality. FPGA Editor provides a high-level view of the routed design. Due to its ability to isolate the DCM reconfiguration phase, Difference Based techniques are preferable since they result in lower partial bit streams and much reduced reconfiguration time. However, it's not a good fit for a setup that makes extensive use of Reconfigurable Modules.

Changes in Organization Only Partial

Reconfiguration Time as Observed

One of the innovative benefits of the suggested implementation flow is a short setup time. One PRM's area was measured to be 25 KB on this test system. Table 1 displays the estimated Partial Reconfiguration Time (PRT) using the formula [14]:

$$PRT = PRM\ (Bits) / CCLK(Hz) * BusWidth(Bits)$$

Table 1 Measured Reconfiguration Time

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY
IJPAST

ISSN 2229-6107 www.ijpast.in

Vol 11, Issue 4. Nov 2021

Configuration mode	Max Clock rate	Data width	Configuration Time (PRT)
Select Map	100 Mhz	16 bit	125 usec
JTAG	66 Mhz	1 bit	3030 usec

Observed Power Reduction

Total Power consumption was measured using XPower Analyzer available with Xilinx ISE Environment. The Fmax for the tested design is 100 MHz. Table 2 compares the measured power dissipation, using the proposed technique and traditional method.

Table 2 Measured Power Consumption at Fmax=100 MHz

On-Chip	Traditional	Proposed
Clocks	2.79	0.24
Logics	0.12	0.00
Signals	0.31	0.00
IOs	68.22	0.00
DCM	68.00	33.49
Quiescent	704.43	703.10
Total	843.87	736.83

A reduction of 107mW of power obtained with the proposed implementation flow. The PR based design and size of the bit file leads to decrease in power consumption.

Implementation summary

The Virtex-5 SXT FPGA included in the ML506 Evaluation platform has been used to test the DRP Reconfiguration Logic we outlined before [10]. On the other hand, any Xilinx FPGA with DCM_ADV basic characteristics may be used to implement it. Except for blocks that vary per FPGA fabricant, like the DCM, the whole FPGA's logic was specified in generic VHDL code. These building blocks were made by either instantiating component primitives or using the Xilinx Coregent tool. Design synthesis and simulation are performed in the Integrated Software Environment (ISE). Functional, post-place-and-route simulation, and bit file creation were all accomplished with the help of Xilinx's Plan Ahead tool. The high performance mandated by the time limitations was attained by the usage of FSM-based VHDL coding approaches.

conclusion

With the introduction of reconfigurable architectures, users now have the chance to install and implement a wide variety of novel programs. The presented framework aids in making full use of the DCM resources already present on FPGA, which may reduce the requirement for further hardware. However, this article focuses on how a design cantered on hardware and software reuse is crucial to the success of these applications.

References

[1] Steven W. Cox, Joel A. Seely, "Rapid SDR waveform development in FPGAS using DSP builder", CF-SDR031505-1.0.

[2] Kevin Skey, John Bradley, Karl Wagner, "A Reuse Approach For FPGA-Based SDR Waveforms", Case No. 06-0796.

[3] Katarina Paulsson,et al., ",Exploitation of Run-Time Partial Reconfiguration for Dynamic Power Management in Xilinx Spartan III-based Systems."

[4] Altera, FPGA Run-Time Reconfiguration: Two Approaches: A White Paper, WP-01055-1.0, March 2008, ver. 1.0.

[5] Majd Ghazi Batarseh, et al., Window-Masked Segmented Digital ClockManager-FPGA-Based Digital Pulsewidth Modulator Technique, IEEE transactions on power electronics, vol. 24, no. 11, November 2009.

[6] Xilinx, Virtex-5 FPGA User Guide, UG190 (v5.3), www.xilinx.com May 17, 2010, Page 50. [7] Xilinx, Virtex-5 Configuration Guide UG191, www.xilinx.com, August 20, 2010, Page 106.

[8] Xilinx, Partial Reconfiguration User Guide UG702 (v12.3), October 5, 2010, Page 17& 27&103.

[9] Julien Delorme, et al., "A FPGA partial reconfiguration design approach for cognitive radio based on NoC architecture", IEEE,2008.

[10] Xilinx, ML505/ML506/ML507 Evaluation Platform, UG347 (v3.0) www.xilinx.com, May 19, 2008.

[11] J. Jones, M. Stettler , "Dynamic Reconfiguration and Incremental Firmware Development in the Xilinx Virtex 5",2008.

[12] Ross Hymel, Alan D. George , "Evaluating Partial Reconfiguration for Embedded FPGA Applications", 2004.

[13] P. Sedcole, et al., "Modular dynamic reconfiguration in Virtex FPGAs", IEE Proceeding on Digit. Tech., Vol. 153, No. 3, May 2006.

[14] Bjorn Osterloh,et al., "Dynamic Partial Reconfiguration in Space Applications",2009 NASA/ESA Conference on Adaptive Hardware and Systems.

[15] Wang Lie, Wu Feng-yan , "Dynamic partial reconfiguration in FPGAs" ,Third International Symposium on Intelligent Information Technology Application,2009,IEEE.

[16] John Huie, et al., "Synthesizing FPGA Cores For Software-Defined Radio", CF-SDR031405-1.0.

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY
IJPAST

ISSN 2229-6107 www.ijpast.in

Vol 11, Issue 4. Nov 2021

[17] Ian Brynjolfson and Zeljko Zilic, "FPGA Clock Management for Low Power" , McGill University, 2008.