International Journal of Electronics Signals and Systems

Volume 3 | Issue 1 Article 13

July 2013

LCD DISPLAY CONTROLLER APPLICATION BY DYNAMICALLY RECONFIGURABLE PLL USING NIOS II

ARCHANA S. BHAMBORE
Department of Electronics (VLSI) Shri. Ramdeobaba Kamla Nehru Engineering College, Nagpur, India,
archana_bhambore@rediffmail.com

R.R. HARKARE
Department of Electronics, Shri. Ramdeobaba Kamla Nehru Engineering College, Nagpur, India,
rrharkare13@rediffmail.com

LCD DISPLAY CONTROLLER APPLICATION BY DYNAMICALLY RECONFIGURABLE PLL USING NIOS II

ARCHANA S BHAMBORE[1] & R.R.HARKARE[2]

[1]Department of Electronics (VLSI) Shri. Ramdeobaba Kamla Nehru Engineering College, Nagpur, India
[2]Department of Electronics, Shri. Ramdeobaba Kamla Nehru Engineering College, Nagpur, India
Email:archana_bhambore@rediffmail.com, rrharkare13@rediffmail.com

Abstract— The Altera Cyclone family of FPGA provides the ability to perform run time reconfiguration which is known as Dynamic Reconfiguration. This paper concentrates on how to take a complex System-on-Chip design into three different components: fixed hardware, reconfigurable hardware and software, each handled by dedicated sub flow. The method to dynamically reconfigure the pll and application used to see variations in frame rate using LCD Display Controller with FPGA is presented. This flow can be considered a part of a general methodology that can be exploited for the implementation on a complex System on Programmable Chip. An example of application it consists utilizes the reconfigurable clocks generated by PLL and video processing in the reconfigurable module. The architecture of the proposed application will presented in this paper were prototyped using a Cyclone III Starter Board, which is based on Nios II Embedded Evaluation Kit, Cyclone III Edition.

Keywords— *FPGA, Dynamic Reconfiguration, VGA Controller, Video Processing, SOPC.*

I. INTRODUCTION

The process of digital systems design has changed dramatically since the introduction of Field Programmable Gate Arrays. The use of FPGAs has been classified broadly into three main categories: rapid prototyping, system implementation, and dynamically reconfigurable subsystems.

Modern Systems on Chip (SoCs) are subject to rapid changes concerning their functionality and other requirements. In order for them to react more flexibly to environmental changes or new tasks, a simple method must be found to allow such systems to adapt to their surroundings. One possible concept for this is based on the assumption that only those parts of the system (system components) which affected by the new tasks or environment must be updated. Following this idea a little further, one can imagine that certain system components could be replaced during run-time, while the remaining, unaffected parts of the system remain fully operational. A developer could use one chip for different tasks and switch between them during run-time. Thus the so called Dynamic Partial Reconfiguration (DPR) leads efficiency of the application development.

A solution for an efficient and reliable reconfigurable clock generator is presented in this paper. The paper begins by providing an overview of PLL in Section II. Section III presents an introduction to dynamically reconfigurable logic and explains the motivation for using it. It reviews the application of dynamically reconfigurable logic and explains how the present state of FPGA technology is impeding the full investigation of more complex applications of dynamic reconfiguration. It also highlights briefly the need for more consistent terminology when describing dynamically

reconfigurable systems. Section IV describes a pll reconfiguration scenario in cyclone III fpga and implementation of it. Section V gives the details of the application to see variations in frame rate with LCD display controller using the dynamically reconfigurable PLLs with existing FPGA design tools. Section six concludes the paper.

II. PLL OVERVIEW

A phase-locked loop (PLL) is a closed-loop frequency-control system based on the phase difference between the input clock signal and the feedback clock signal of a controlled oscillator. Figure 1 shows a simplified block diagram of the major components in a PLL. The main blocks of the PLL are the phase frequency detector (PFD), charge pump, loop filter, voltage controlled oscillator (VCO), and counters, such as a feedback counter (M), a pre-scale counter (N), and post-scale counters(C).

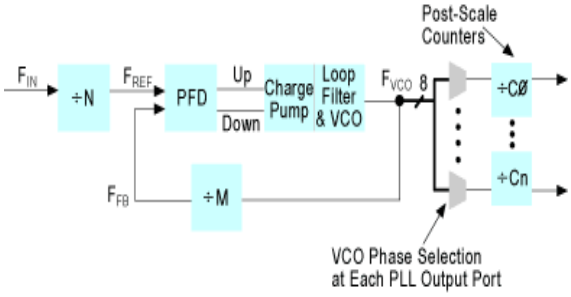


Fig. 1 Block Diagram of a PLL

The output frequency of the PLL is equal to the VCO frequency (F_{VCO}) divided by the post-scale counter (C).

In the form of equations:

- $F_{REF} = F_{IN} / N$
- $F_{VCO} = F_{REF} \times M = F_{IN} \times M/N$
- $F_{OUT} = F_{VCO} / C = (F_{REF} \times M) / C = (F_{IN} \times M) / (N \times C)$

III.DYNAMIC RECONFIGURATION

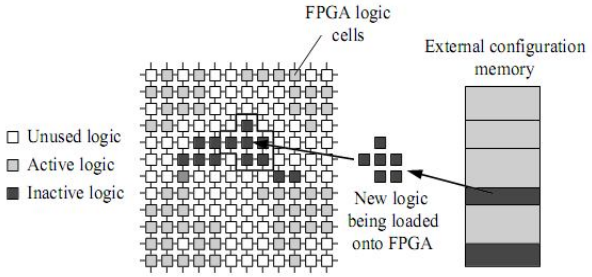


Fig. 2 Dynamic Reconfiguration

The term (Re-) Configurable Computing refers to computers that can modify their hardware circuits during system execution.

Dynamic reconfiguration is defined as the selective updating of a subsection of an FPGA's programmable logic and routing resources while the remainder of the device's programmable resources continue to function without interruption. It is equivalent to partial, run-time reconfiguration of an active array. Such FPGAs are examples of dynamically reconfigurable logic devices. The critical point in this definition is that the FPGA does not have to be halted in order to have its function partially reconfigured. Figure 2 is a simplified representation of dynamic reconfiguration in progress. Several subcircuits are shown resident on the FPGA array, but only one is to be reconfigured. The operation of the appropriate subcircuit is suspended and only those logic cells that need to be modified are overwritten with new configuration data. The other active subcircuits continue to function throughout the reconfiguration period.

Dynamic reconfiguration offers important benefits for the implementation of reconfigurable systems. Dynamically reconfigurable FPGAs offer the fastest possible way to change an active FPGA circuit since only those parts that need to be reconfigured are interrupted. This results in faster overall system operation. The smaller configuration bitstreams also require less external memory for storage. The profile of the circuitry that is active on the array may be adjusted dynamically to match the requirements of the algorithm being executed thus optimising the ratio of active circuitry to available silicon and potentially reducing the overall number of components required by a design. Moreover, the opportunities for deploying dynamic reconfiguration are increasing as the gate counts of individual FPGAs continues to improve. As larger FPGAs become available, the complexity of the systems that can be integrated into a single FPGA increases. Consequently, the probability that the execution of subsets of the logic will be mutually time-exclusive is also likely to increase. The benefits of faster reconfiguration, and hence faster overall system speed, and reduced component count thus become more apparent with increasing part capacity[3].

III.PLL RECONFIGURATION SCENARIO IN CYCLONE III FPGA

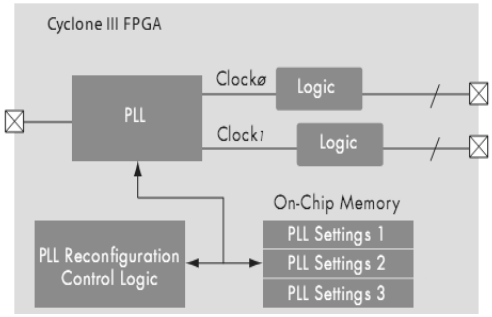


Fig. 3 PLL Reconfiguration Scenario

The PLL reconfiguration feature can dynamically adjust the PLL parameters to lock to a very wide spectrum of input clock frequencies. This is particularly useful in display applications where clock rates vary based on the system and the refresh rate chosen. In this case, Altera-provided IP is used to control the reconfiguration and automatically adjust the PLL parameters to lock onto the given input frequency[5].

A. Implementing PLL Reconfiguration In Cyclone III Devices

PLLs use several divide counters and different voltage-controlled oscillator taps to perform frequency synthesis and phase shifts. In Cyclone III PLLs, you can reconfigure the counter settings and dynamically shift the phase of the PLL output clock. You can also change the charge pump and loop filter components, which dynamically affect the PLL bandwidth. You can use these PLL components to update the clock frequency, PLL bandwidth, and phase shift in real time, without reconfiguring the entire FPGA.

Applications that operate at multiple frequencies can benefit from PLL reconfiguration in real time. PLL reconfiguration is also beneficial in prototyping environments, allowing you to sweep PLL output frequencies and adjust the clock output phase at any stage of the design. For example, a system generating test patterns is required to generate and transmit patterns at 50 or 100 MHz, depending on the device under test. Reconfiguring the PLL components in real time for this example allows you to switch between the two output frequencies within a few microseconds.

The following PLL components are reconfigurable in real time:

■ Pre-scale counter (N)
■ Feedback counter (M)
■ Post-scale output counters (C0-C4)
■ Dynamically adjust the charge pump current (ICP) and loop filter components (R, C) to facilitate on-the-fly reconfiguration of the PLL bandwidth

Figure 4 shows how PLL counter settings can be adjusted dynamically by shifting their new settings into a serial shift-register chain or scan chain. Serial data is fed into the scan chain via the scandata port and the shift registers are clocked by the scanclk port. Serial data is shifted through the scan chain as long as the scanclkena signal stays asserted. After the last bit of data is clocked, asserting the reconfiguration state machine signal, configupdate, for at least one scanclk cycle causes the PLL configuration bits to be synchronously updated with the data in the scan registers. The scan chain can also be initialized or changed using a memory initialization file in the hexadecimal (.hex) file or memory initialization file (.mif) format [7].

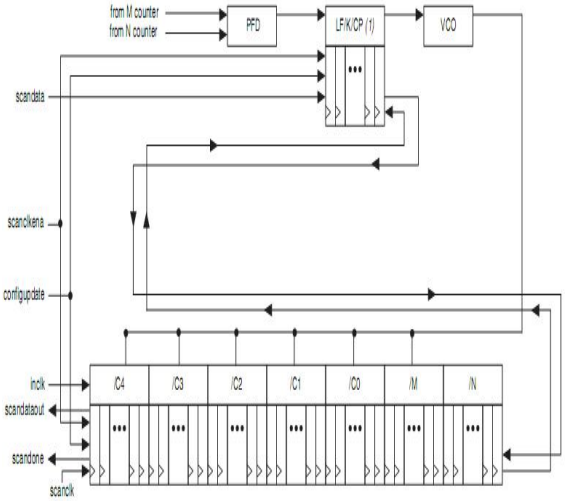


Fig. 4 PLL Reconfiguration Scan Chain

IV. LCD APPLICATION OF DYNAMICALLY RECONFIGURABLE LOGIC

Video systems almost always include an embedded processor and a memory subsystem to manage the video frames in the external memory. The SOPC Builder system tool provided by Altera greatly simplifies embedded system design. This tool includes a library of elements such as soft core processors (Nios® II), interfaces, memory, bridge, and IP cores. It also features a connectivity GUI and generator to automatically wire up arbitrated and streaming bus systems.

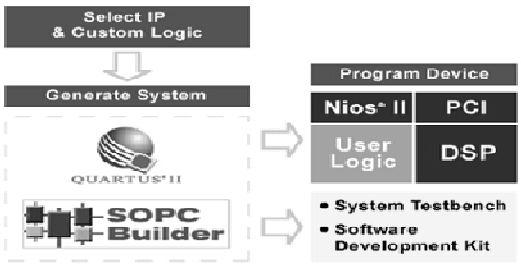
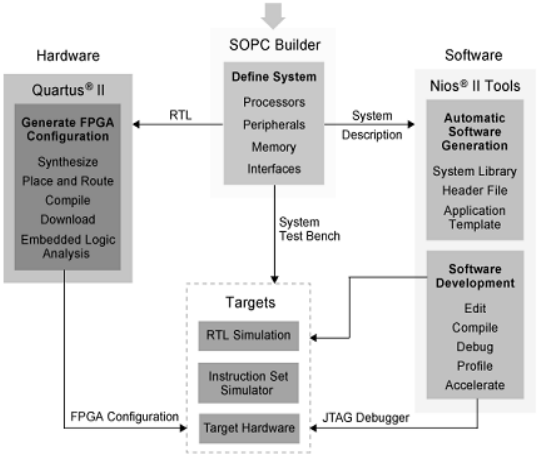


Fig. 5 SOPC-Based Design Flow for Video Datapath

A. Nios II Embedded Design Suite

The Nios® II Embedded Design Suite (EDS) is a collection of components and tools used to develop embedded software for the Nios II processor, including Nios II Software Build Tools for Eclipse based on the familiar Eclipse development environment. The Nios II IDE is also included to provide support for legacy designs.



Nios II is an embedded processor system that can be configured by the user, meaning the actual hardware of the processor is easily customized for a particular application through the SOPC Builder feature of the Quartus II and SOPC Builder FPGA design software.

Cyclone III family offers a wide range of capability, including density, on-board memory, and I/O, that designers can make use of in order to design the right application for their end product.

A. VGA Controller

The VGA controller core displays images by creating the timing signals required by VGA compatible monitors attached to the VGA port on the DE-series board, or the Terasic LCD screen with touch panel. Video is produced by displaying frames (or images) in rapid succession. In a typical video, frames are displayed between 30 and 120 times per second. A frame is a two-dimensional array of pixels as depicted in Figure 6.

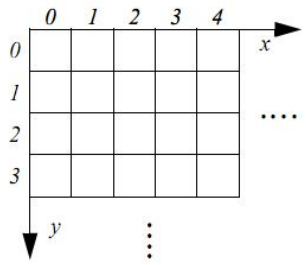


Fig. 6 Video frame's screen layout

The resolution of a frame is defined as the number of pixels in the x and y axes. An example resolution is 640x480,which has 640 pixels across the x axis and 480 pixels down the y axis, as shown in Figure 7. Therefore, each pixel location in a frame can be identified by an (x,y) coordinate, with (0,0) being in the top-left corner [8].

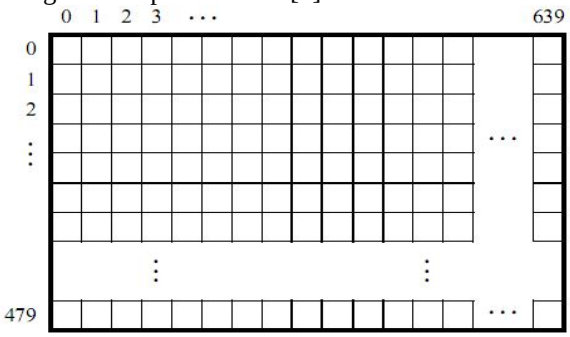


Fig. 7 A frame with a 640x480 resolution.

VGA Controller, creates the timing information required by the LCD daughtercards or by the on-board VGA DAC.

The VGA Controller generates the timing signals required for the VGA DAC and LCD daughtercards, including horizontal and vertical synchronization signals. The timing information generated by the VGA Controller produces screen resolutions of 800x480 pixels for the VGA DAC and LCD Touchscreen. To generate the timing information correctly, a 50 MHz clock has to be provided to the VGA Controller. This is the default clock in the SOPC Builder software and on the DE boards. In addition, a 25 MHz clock signal must be supplied to the VGA DAC or the LCD daughtercards. For the VGA DAC, this clock can be supplied via the VGA_CLK pin. The External Clocks for DE Board Peripherals core, also provided by the Altera University Program, can generate the required 25 MHz clock; see its documentation for more details.

B. LCD Design Example:

Figure 8 shows an example of a typical application that benefits from the improved Cyclone III PLLs. The system drives a LCD display and integrates functions, such as timing controller, video processing, and a memory controller into a single Cyclone III FPGA.

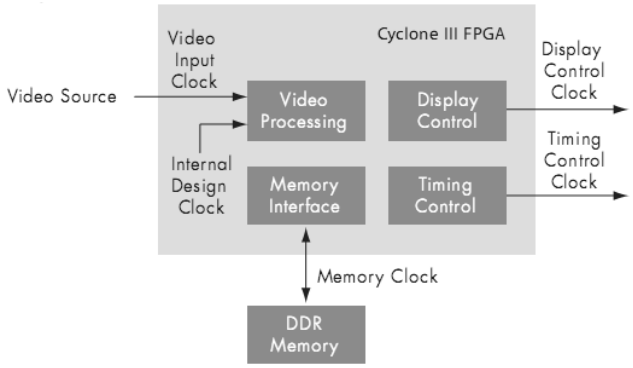


Fig. 8 LCD Display Design Example

This example takes advantage of several features of the Cyclone III PLL. The first is the ability to drive multiple clocks. This application has multiple clock domains that the FPGA must generate. It must be able to take in the clock frequency from the video source and generate clocks for the memory interface, the internal logic for the FPGA, and any multiple clocks the LCD display may require. Depending on the size and resolution of the LCD, the panel may have multiple clocks that need to be driven. Using the numerous outputs and multiple PLLs will allow you to generate all the clocks needed internally and externally.

The second feature is reconfigurability. In many cases, the clock rates are unknown. The video source may come in different formats with different clock frequencies, or the refresh rate and resolution of the LCD may differ from one panel to another or change on the fly. The wide frequency range of the PLL allows it to accommodate a number of different sources, as well as switch and readjust frequencies on the fly. In other FPGA architectures, the whole FPGA would need to be reconfigured to produce the right clocks, but this reconfiguration takes time and requires special design consideration. The reconfigurability of the Cyclone III PLL insures that this takes place smoothly and with minimum downtime[5].

VI. CONCLUSION

In this paper, we proposed a general methodology that can be exploited for the implementation on a complex System on Programmable Chip and we used a video processing for realization. The results show that the flow provides an effective and low cost approach to the partial dynamic reconfiguration and mixed hardware-software execution problems. Its strength lies on the introduction of the partial dynamic reconfiguration degree of freedom at design time. The proposed flow organizes the input specification into three different components: hardware, reconfigurable hardware and software, managed by proper portion of the methodology.

REFERENCES

[1] Radu M. Secareanu, David Albonesil, and Eby G. Friedman' Motorola, Inc., "A Dynamic Reconfigurable Clock Generator," Semiconductor Products Sector, Digital DNATM Laboratories, Tempe, AZ 85284 luniversity of Rochester Department of Electrical and Computer Engineering, Rochester, NY 14627-0231

[2] P. Lysaght, J. Dunlop, "Dynamic Reconfiguration of FPGAs," In: W. Moore, W. Luk (eds.), More FPGAs, Abingdon, Oxford, UK, pp. 82–94, 1994.

[3] "A Simulation Tool For Dynamically Reconfigurable Field Programmable Gate Arrays," Patrick Lysaght and Jon Stockwood

[4] "A Dynamically Reconfigurable Computing Model for Video Processing Applications ,"G.Alonzo Vera,Daniel Llamocca,Marios S.Pattichis, James Lyke,978-1-4244-5827-1/09/$26.00©2009 IEEE

[5] Altera Corporation, "Gain Flexibility And Increased Integration With Advanced Cyclone III Fpga PLLs," White Paper , October 2007, Ver. 1.0

[6] Timothy Carl Bronson , David Andrew Thorndike, "Phase Locked Loop Reconfiguration," US Patent, Sept.16 2003

[7] Altera Corporation, "Implementing PLL Reconfiguration in Cyclone III Devices," www.altera.com September 2011

[8] Altera Corporation - University Program,"Video IP Cores for Altera DE Series Boards," July 2010.

[9] "Phase-Locked Loop (ALTPLL) Megafunction User Guide," www.altera.com November 2009

[10] "Digital Logic and Microprocessor Design With VHDL," a book by Enoch O. Hwang La Sierra University, Riverside

[11] Altera Corporation, "Nios II 3C25 Microprocessor with LCD Controller Data Sheet," March 2009