Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

Metro-on-FPGA: A feasible solution to improve the congestion and routing resource management in future FPGAs

A. Belghadr, A. Jahanian*

Electrical and Computer Engineering Department, Shahid Beheshti University, G.C., Velenjak, Tehran 19839-63113, Iran

ARTICLE INFO

Article history:
Received 25 July 2012
Received in revised form
21 July 2013
Accepted 27 July 2013
Available online 19 August 2013

Keywords:
Asynchronous serial link
FPGA
Routing congestion

ABSTRACT

Asynchronous serial transceivers have been recently used for data serializing in large on-chip systems to alleviate the routing congestion and improve the routability. FPGAs have considerable potential for using the asynchronous serial transmission but they have serious challenges to use this technology. In this paper, we present a new FPGA architecture corresponding with a new routing algorithm to use the asynchronous data serializing technique in modern FPGAs. Experimental results show that allocated routing tracks and routing congestion can be reduced considerably (18.81% and 48.73%, respectively) by using the asynchronous data serializing without any performance degradation in cost of reasonable overhead in area and power consumption. The resulting improvements will increase for larger and more complex FPGAs.

1. Introduction

In recent years, technology node of FPGAs has dramatically shrunk and also their logic and interconnect capacity have improved considerably. On the other hand, number of embedded processors and capacity of programmable modules in FPGAs has been increased. As a result, coming FPGAs will be too congested and routability of a them will be a serious concern [1]. In an over-congested design, many wires have to be detoured that may cause to generate large number of long wires. These long wires occupy many of routing resources and have a great impact on routability and performance of the design and may even result in crosstalk and manufacturability problems.

At the same time, routing of FPGAs is more restricted than ASICs because FPGA routing resources are limited by the number of channels and switches [2]. It is worthwhile to note that congestion does not affect the performance directly but in a congested design, which has many detoured long wires, performance may be reduced. In this situation, congestion reduction is very important in FPGAs because it can improve routability and performance.

An effective method for congestion reduction is minimizing the number of used routing tracks for each net. As the number of allocated routing track of nets decreases, more routing resource will be saved for other FPGA wires and consequently, routing congestion can be reduced. It is worth noting that track reduction does not increase the performance explicitly but it can improve the performance indirectly by decreasing the routing congestion level. Moreover, about 80%–90% of area and power in a modern FPGA is dedicated to the routing resources [3]. Therefore, track reduction which may reduce routing congestion, can improve FPGA routing resource usage and also power consumption of used routing switches.

Asynchronous serial links have been used in many of 2D and 3D ASIC circuits to reduce routing congestion by serializing the parallel wires into serial lines [4].

In [5], some asynchronous techniques for serial transmission in NoC structures were introduced and their applications were described. Asynchronous circuits can be implemented by various approaches such as QDI, GasP and wave pipelining [6–9].

Authors of [10] proposed a serial link transceiver for global on chip communications working at 3 Gb/s per wire in a standard 0.18 μm CMOS process with low crosstalk-induced delay variability. They implemented pure electrical link without any repeater in higher layers of metal (M6 and M4) to convey the NoC information.

Teifel and Manohar [11] presented a high-speed and clock-less serial link transceiver for inter-chip communication in asynchronous VLSI systems. They used a token-ring architecture that eliminates complex clock generation and synchronization circuitry. Their receiver dynamically self-adjusts its sampling rate to match the bit rate of the transmitter. Their experiments showed that their transceiver operates at up to 3 Gb/s in 0.18 μm CMOS technology.

Authors of [12] proposed an asynchronous serializing for ASIC circuits. Experiments and also analyses show that routability and congestion can be improved considerably in cost of slightly

* Corresponding author. Tel.: +98 21 66873190; fax: +98 21 22431804.
E-mail addresses: ar.belghadr@mail.sbu.ac.ir (A. Belghadr), jahanian@sbu.ac.ir (A. Jahanian).
URL: http://faculties.sbu.ac.ir/~jahanian (A. Jahanian).

0167-9260/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.vlsi.2013.07.002

increasing of the area and total wire-length by asynchronous serializing the parallel wires. In addition, authors of [12] showed that power consumption is not increased considerably by data serializing.

In [13], a comparison between wave pipelining and surfing source-synchronous schemes in the presence of power supply and crosstalk noise was made. Their results show that wave pipelining can operate at rates as high as 5 Gbps for short links, but it is very sensitive to noise in longer links and must run much slower to be reliable. In contrast, surfing achieves a stable operating bit rate of 3 Gbps and is relatively insensitive to noise.

In this paper, we proposed a new design methodology for using the asynchronous data serializing in FPGAs to reduce routing wire segment usage and congestion without any performance degradation. This mechanism is called *Metro-on-FPGA* (*MoF*) in this paper. In this methodology, a new architecture is proposed for FPGA consisting of some asynchronous serial transceivers. Then, a post-routing algorithm is proposed to update the FPGA routing to use these transceivers in order to multiplex parallel wires to serial wires and reduce routing tracks usage. In this algorithm, near-optimal set of non-critical and sufficiently long wire segments are selected and serialized at the signal sources and then, they are de-serialized into parallel signals at the destinations after serial transmission. The main contribution of this paper is improving the routing congestion of design by reducing the number of required routing tracks without any clock frequency degradation.

This paper is organized as follows. Section 2 describes the basics of asynchronous serial transmission in ASICs. In Section 3, the Metro-on-FPGA concept is proposed. Section 4 explains the proposed architecture corresponding with the presented algorithm for using the serial communication in FPGAs. Experimental results are reported and discussed in Section 5 and finally, Section 6 concludes the paper.

2. Asynchronous serial transmission in ASIC design flow

The idea of asynchronous data serializing in regular ASIC design methodology is proposed in [12]. In this section, this concept is briefly described to facilitate the presentation process of the contribution. In the conventional physical design flow, terminals of each net are connected via a dedicated wire. Serializing the parallel nets in congested circuits can be an effective technique for congestion reduction. The main idea is to automatically find non-critical and sufficiently long wires and serialize them using asynchronous serial transmission hardware. This mechanism is similar to the Metro system in large cities that conveys many passengers in one trip to reduce traffic congestion.

Authors of [12] used the fast serial communication mechanism proposed in [14,15] for the serial link module. This serial link transceiver uses low-latency synchronizers at the source and sinks with a two-phase non-return zero (NRZ) asynchronous protocol that allows non-uniform delay intervals between successive bits. The acknowledgement of transmission is returned only once per word, rather than bit by bit enable pipelined transmission as shown in Fig. 1.

In [14], a high data rate shift-register structure for asynchronous serializer and de-serializer has been proposed. It operates at about one fan-out 4 inverter (FO4) delay and can be fully implemented in CMOS technology. Authors of [14] have used their shift register as the serializer and de-serializer in a bit-serial on-chip communication link. Main advantage of the serial link of [14] is that it eliminates the clock and replaces flip-flops by lower latency latches, but incurs handshake overhead due to having to acknowledge each data transfer. This problem is alleviated by generating acknowledgement signal at the word level, rather than bit by bit, enabling multiple bits in a pipelined manner over the serial channel. It is worth noting that control signals *T* and *TN* in [14] should be generated using multi-phase clock generator.

A MoF cluster with n wires consists of 2n pipelined asynchronous shift registers to serialize and also de-serialize the data. These registers are named as XL in [14]. Each XL contains 12 transistors with about 3.78 μm^2 area in 32 nm technology. It was shown that the proposed circuit can send and receive each bit at a data cycle of a single FO4 (fan-out of 4) inverter delay (about 10 ps in 32 nm technology).

Inserting an asynchronous module in a synchronous circuit may change the functionality of the whole system due to the violation of the setup and hold time requirements of synchronous registers. In this section, a feasible solution for interfacing the asynchronous transmission module and a synchronous circuit with a global clock is presented. In the proposed interface, on each rising edge of the system clock, the asynchronous serial link starts to convey a copy of signals at the source towards the destination and this operation is repeated for all input signals. At the end of the operation, a data-acknowledge signal is returned back to the sender. The process of serializing, transmission and de-serializing is initiated by the activation of start signal and the completion of the whole transmission process is indicated by activating the data-ack signal. The signaling of the synchronous system after inserting the asynchronous serial link is shown in Fig. 2.

In this mechanism, the clock period must be extended by the delay of serializing/de-serializing hardware if the paths containing the serialized wires do not have enough delay slack. In other words, if the paths of the selected wires have enough delay slack,

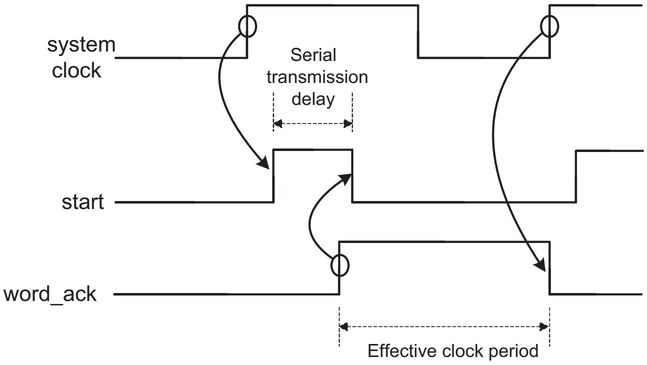

Fig. 2. Signaling of serial link transmission.

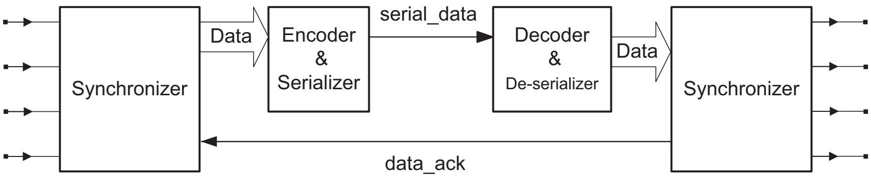

Fig. 1. Global structure of used serial transmission hardware.