Informatics and Mathematical Modelling
Technical University of Denmark

02207 : Advanced Digital Design Techniques

Exercise of Retiming

LAB 3

Group *dt07*

Markku Eerola (s053739)

Rajesh Bachani (s061332)

Josep Renard (s071158)

November 27, 2007

Contents

1 Introduction

This document is report of the third exercise on DTU course Advanced Digital Design. In this exercise we studied the concept of retiming using digit recurrence division implementation with radix-4 and carry-save adder.

In the introductory section we will briefly explain the concept of retiming, the original circuit and the retimed circuit. In the next section we will explain how the retimed circuit was implemented ie. what changes we made to the original circuit. In the last two sections we will present the power reports and cell counts of the two designs and discuss the results.

1.1 Retiming

Retiming is an optimizing technique where structural location of registers is manually moved without affecting the functionality of the circuit in order to improve its performance. This is done either by removing a register from each input to a block and adding a register to each output, or by adding registers to the inputs and removing registers from the outputs. In our case the motivation for retiming was to create slack on a non-critical path, and to have the synthesizer substitute HS cells with LL cells on this path thus lowering the overall power dissipation in the whole circuit. According to the lecture slides the circuit we were studying should gain approximately 30% power savings from this kind of retiming.

1.2 Simple Design for Division

The original design upon which we aimed to improve with the retiming is presented in figure 1.

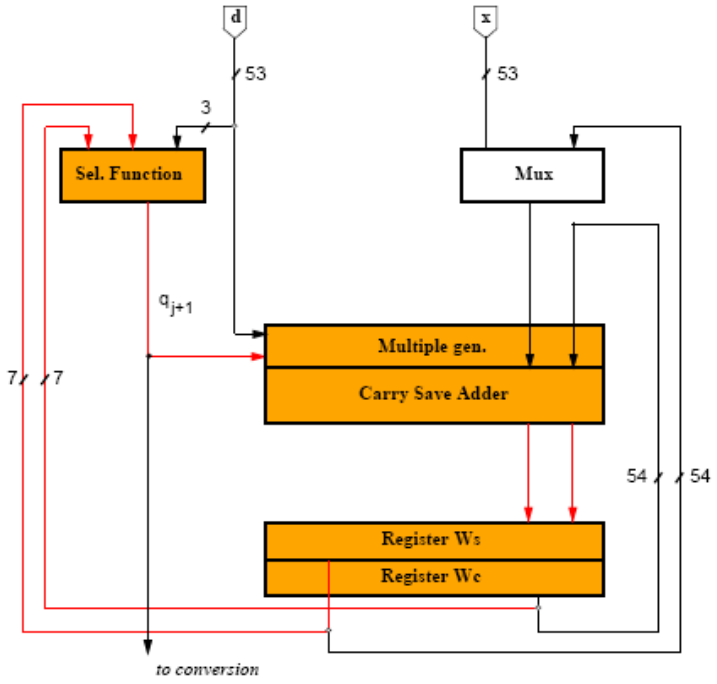

Figure 1: Digit recurrence division

The Sel. function -block implements the quotient digit selection function. The selection function determines a 4-bit quotient digit using 3 most significant bits of the divisor d and 7 most significant bits from the results stored in registers Ws and Wc.

The MUX block selects the input for the divisor multiplication between the dividend, which is used only in the initialization phase of the division algorithm, and the result of the substraction of the quotient digit/divisor multiplication result from the dividend. The substraction result is stored in register Ws.

The Multiple gen. -block implements the divisor multiplication ie. it multiplies the divisor d with the 4-bit quotient digit. This block is basically a multiplexer.

The Carry Save Adder -block implements the substraction of the result of the divisor multiplication from the dividend. The substraction is done with a carry-save adder as the name of the block suggests.

The registers Ws and Wc store the carry and the sum from the carry-save adder respectively.

The critical path of this circuit is marked with red arrows in the figure 1.

1.3 Design for Division using Retiming

The retimed design is presented in figure 2.

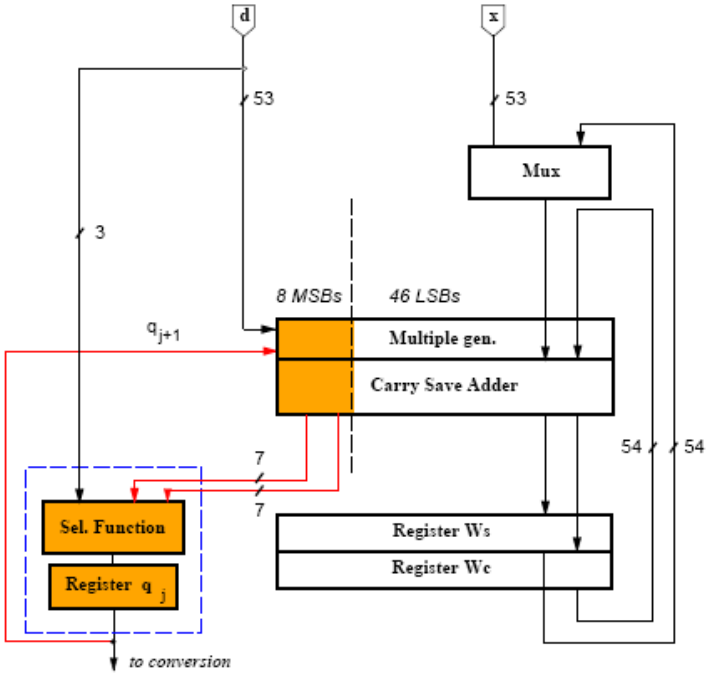

Figure 2: Digit recurrence division retimed

Since only the most significant bits of sum and carry are used in the quotient digit selection it makes sense to separate the most significant bits from the least significant bits to separate structural slices. This is done in the VHDL code by disconnecting the quotient digit selection block from the W registers thus *removing registers from the inputs* and adding new registers to the output of the block. This frees the W registers from the most significant slice. By also separating the implementation of the most significant bits of the adder and the multiplie generation from the implementation for the less significant bits more of the design is freed from the critical path.

These changes do not affect the functionality of the circuit, but by dividing the implementation to most significant and least significant slices we get two paths. The critical path is marked with red arrows in the figure 2. The delay on the critical path is T(SEL)+T(reg q)+T(mux)+T(CSA). The delay on the non-critical path is at maximum T(reg W)+T(mux)+T(CSA). From this it can be seen that the non critical path has some slack which the synthesizer should be able to use to optimize the least significant slice for power, namely by replacing HS cells with LL cells.

1.4 Authors by Section

- *Rajesh Bachani*
- *Josep Renard*
- *Markku Eerola*