Microelectronics Journal 66 (2017) 84–88

Contents lists available at ScienceDirect

Microelectronics Journal

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

An improved design and simulation of low-power and area efficient parallel binary comparator

Chang Chua*, R.B.N. Kumar

Intel Technology India Pvt Ltd., India

ARTICLE INFO

Keywords:
Binary comparator
CMOS technology
Pass transistor logic
Power
Area

ABSTRACT

This paper presents a new low-power and area-efficient parallel binary comparator design based on prefix tree structure. Due to its wide usage in central processing units, optimizing binary comparator for low power applications are need of the hour. A novel EX-OR-NOR gate is used in proposed binary comparator as pre-encoder to reduce area, power and delay. The simulation results performed using CADENCE for CMOS 180 nm – technology. The paper proposes two binary comparator architectures with improved performance. The proposed architecture result in a power reduction upto 25%, area (number of transistors) reduces upto 36% and improves the delay performance 27% compared to existing technique.

1. Introduction

Binary comparator is a digital circuitry, which compares the two binary numbers and generates the binary outputs interms greater than, less than or equal. Binary comparator consists of two input operands and three outputs as shown in Fig. 1.

Even through realization of the architecture is simple, it has wide usage in digital building blocks like central processing unit (associate memories, argument comparison block, queue buffers), test circuit as signature analysis [1] and micro controllers [2]. Binary comparator can be used as decoding circuitry for address mapping for I/O storage devices. This leads to improving the design aspects like optimization of area, power and performance of the binary comparator.

Increase in the number of input bits of the binary comparator results in exponential increase in the area for a binary comparator [3]. Conventional binary comparators are designed using subtractive method (2's complement), which incorporate flat adders [4], resulting in slower response. Alternatively replacing flat adders with fast adder result in marginal improvement in the delay performance [5]. Slow response of the conventional comparator can overcome by tree structure. Tree structure in the comparator design provide improve delay performance. However power dissipation along with delay continuous to be a bottleneck, while designing a binary comparator [6]. However, this is not admissible for wide range of applications due to $\log_2 N$ comparison levels. Further, it is not possible to compute the equality (not supported). Recently many papers are proposed or reducing the switching activity [9].

For reducing dynamic power loss in switching activity, clock signal has been introduced for circuitry [7]. However, this method suffers from loading of clock signal for higher number of bits at the input. Alternatively bitwise comparison can be used, in which bits are compares from MSB to LSB only when the MSB bits are equal [8]. But it has limitation in which synchronizes with bits is leading to parasitic capacitance and increased power dissipation.

In this paper, a low-power and area-efficient N-bit parallel binary comparator is proposed. Which is designed using conventional EX-OR gates and novel EX-OR-NOR gates. The designs interconnection are arranged in a way to optimize the parasitic capacitance in the circuitry. The proposed algorithm enhance the performance and reduces the area and power consumption compared with existing algorithms.

The remaining paper is organised as follows. In Section 2, the existing technique to design the parallel binary comparator is explained. Section 3, presents basic definitions. Section 4 introduces the proposed for N-bit parallel binary comparator design based on prefix tree and a novel EX-OR-NOR as a pre-encoder. Simulation results and graphs are presented in Section 5. Section 6 concludes the paper with key remarks.

2. Existing parallel binary comparator

Existing parallel binary comparator [10] consists of comparison resolution module and decision module as shown in Fig. 2. In comparison resolution module, it uses prefix tree structure of five hierarchy stages to deal with different input bit-width and output has two N-bit buses, which is called as left and right buses. The outputs from the bus are given to decision module to decide the output of binary comparator with the help of decision-making OR-networks.

* Corresponding author.
E-mail address: changchuaki@gmail.com (C. Chua).

http://dx.doi.org/10.1016/j.mejo.2017.06.003
Received 18 February 2017; Received in revised form 14 April 2017; Accepted 7 June 2017

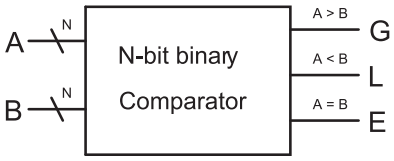

Fig. 1. Generic N-bit binary comparator.

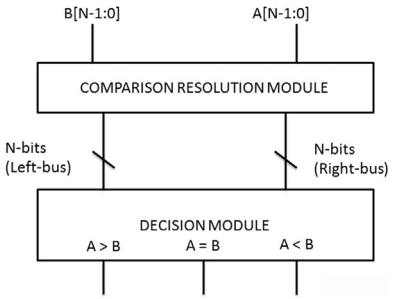

Fig. 2. Existing binary comparator architecture.

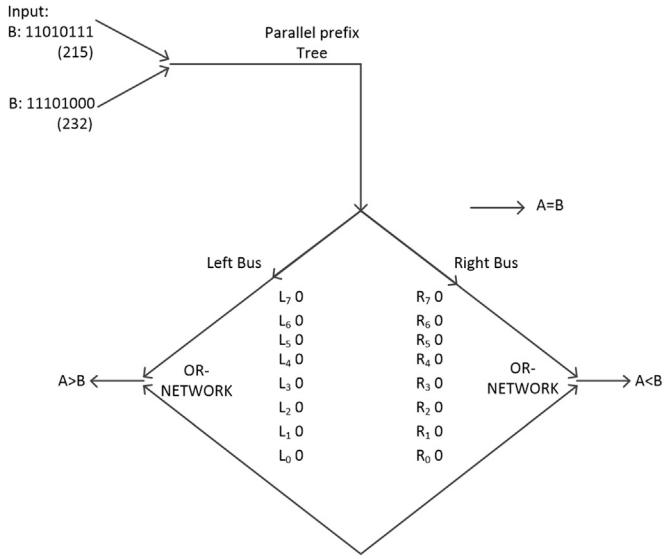

Fig. 3. Example of an 8-bit comparison using parallel prefix tree [10].

Table 1
Basic logic of binary comparator.

Condition	G	L	E
A > B	1	0	0
A < B	0	1	0
A = B	0	0	1

The parallel prefix tree is designed based on the principle of asynchronous bit-wise comparison of two input binary operand, which is given at the input. Consider a case of two input binary operands A, B with a size of N bits, starting from 0th bit is least significant bit (LSB) to (N-1)th bit is most significant bit (MSB). Prefix tree structure consists of encoder in which comparison start from MSB to LSB and result is stored into N-bit left and right buses. In parallel prefix tree, the switching activities have been reduced by terminating the comparison of LSB, when the MSB's are not equal, So either of two buses has only one high bit [10]. The decision module take input of left and right buses with the help of two OR-networks and decides which binary input is greater or less.

For example, an 8-bit comparison of two binary operands A = 11010111 and B = 11101000 is shown in Fig. 3. Firstly pre-encoder compares A and B and gives output to next stage. The two N-bit buses are given to decision module [10]. MSB bits $A_7 = 1$ and $B_7 = 1$ gives as L_7 and $R_7 = 0$, $A_6 = B_6 = 1$ gives as L_6 and $R_6 = 0$, $A_5 = 0$ and $B_5 = 1$ gives as $L_5 = 0$ and $R_5 = 1$ and terminates comparison an leaves the remaining left and right buses as zero regardless of input operands. As right bus consist of a high bit, it can be concluded that A is less than B.

The above mentioned parallel binary comparator is easy to implement and since it has parallel architecture overall delay of the binary comparator is less. But since there are two separate buses for checking greater than and less than condition, the area and power dissipation is high. To overcome these limitations, the paper proposes single bus comparator architecture for checking greater than and less than condition.

3. Basic definitions

In this section, we present the basic idea and definitions of power, area and delay.

3.1. Power

The power of a logic circuit is the summation of individual power of each gate of the circuit. Suppose, a circuit consists of n gates and individual power of those gates are p_1, p_2,…., p_n, respectively. Then, the power (P) of the circuit is

$$P = \sum_{i=1}^{n} p_i \tag{1}$$

As we can obtain current rating of each gate by using CMOS 180 nm technology, we can calculate power requirement of each transistor using the formula P=V * I. Here, P refers to the power, V refers to the voltage and I refers to the current.

3.2. Area

The area of a logic circuit is the summation of individual area of each gate of the circuit. Suppose, a circuit consists of n gates and area of those n gates are a_1, a_2,…., a_n. Then by using above definition, area (A) of that circuit is

$$A = \sum_{i=1}^{n} a_i \tag{2}$$

Given the above definition the area of a circuit can be calculated easily by obtaining area of each individual gate using CMOS 180 nm technology.

3.3. Delay

The delay of a logic circuit is the maximum number of gates in a path from any input line to any output line. This definition is based on the following two assumptions: firstly, each gate performs the computation in one unit time. This means that every gate in the given circuit will take the same amount of time for internal logic operations. Secondly, all inputs to the circuit are known before the computation begins. In this work, computation time for each gate is obtained using CMOS 180 nm technology, and delay is calculated as the sum of delays of each gate through the path that causes maximum delay. This means that the internal structure and each operation of the gate are known before the calculation. Interconnect delay is playing a significant role as semiconductor technology is marching towards lower sized chips. The interconnect delay (t_{id}) can be calculated as [11]

$$t_{id} = (3.56{*}K{*}L^2{*}\rho{*}\epsilon)/(\lambda^2{*}n) \tag{3}$$

Here, K is the dielectric constant of metal, L is the maximum interconnect wire length, ρ is the wire resistivity, ϵ is the permittivity, λ is