A Review on Digital Binary Comparator

[1]Vikas Agrawal,[2]Shweta Agrawal
[1]M.tech Scholar [2]Assistant Professor
[1]Electronics & Communication, VLSI, SRCEM, Banmore , Morena ,India

Abstract— In era of electronics comparator is most important digital combinatorial circuit which compares two digital or analog signals, but outputs always digital. With enhancement of technology optimization of digital circuit design improving day by day at the cost of complexity .In this review paper some thoughts on performance of dynamic comparators is discussed. The purpose is to find the high speed, low power and minimum area of dynamic comparator design. There are number of techniques to design digital comparators. By using different logic styles of comparators it is used accordance with application in specific need.

Index Terms— Comparator, Low Power , MIMO, Pre- encoder.

I. INTRODUCTION

The comparison of two n-bit bit numbers is a critical operation for almost all digital systems. A comparator compares two n-bit values to determine which is greater, or if they are equal. In general it is used to compare two inputs. Comparators are broadly classified into Analog and Digital comparators. However in this brief what is concerned is about the digital comparator. The digital comparator is further classified into Total (Full) comparators and Equality comparators. In full comparators, given two n-bit binary numbers A and B, they are able to separately recognize the three possible conditions i.e. A > B, A < B and A = B. In equality comparators, as the name suggests, they only indicate equality when both the inputs are equal. Comparators find their applications in many Digital Signal Processors. It has been an important logic block in an ALU and have extensive applications such as decoding of x86 instructions. It also finds applications in MIMO (Multiple Input Multiple Output) decoding algorithms require extensive iterations of binary number comparison.

II. EXISTING COMPARATOR DESIGNS

In order to achieve a very high throughput, the approach proposed in [1] uses 2-phase clocking dynamic logic with all N transistors (ANT) blocks. In the ANT logic there are N blocks, the threshold voltage of transistor is variable depending on operation of entire block. Such a 64-bit transistor requires 1890 transistors and produces the correct result within 3.5 clock cycles. The main disadvantage of [1] is that it can only be implemented with heavy pipelining. Some popular microprocessors like ARM often need to execute a comparison instruction within a single clock cycle. The latches used to form the pipelines increase the circuit complexity and power consumption of the ANT comparator. A single cycle, two-phase clocking architectures are presented in [2] and [3].These comparators use a priority-encoding algorithm in [2] and a parallel MSB checking method is exploited in [3]. The latter is 22% faster than [2] but it requires 88% more transistors. In order to increase the achievable speed, a modification of MSB checking algorithm used in [3] has been proposed in [4] in which a MUX based structure has been used. This architecture is basically designed for high fan-in comparators and exhibits highest computational speed but requires 3386 transistors. The design in [4] is not suitable for static logic implementation due to tall transistor stack height. In [5] a high performance tree based comparator is proposed wherein generate (G) and propagate (P) signals can be used for binary comparisons.

III. 64-BIT TREE BASED COMPARATOR

The basic comparison operation between two n-bit numbers A and B can be performed by a simple addition operation. That is, when A is greater than or equal to B, the addition operation between A and 2's complement of B generates a carry signal equal to 1.When A is less than B, the carry signal is 0. The low-cost addition architectursaes such as ripple-carry adders drastically reduce the operating speed. On the other hand, high-speed adders increase the hardware complexity. Due to this reason the design of efficient comparators doesn't employ addition logic. The circuit, for comparing two n-bit numbers, has 2n inputs and 2^{2n} entries in the truth table. The 64-bit binary comparator [5] compares two numbers each having 64 bits (A63 to A0 & B63 to B0). Therefore in this arrangement the truth table has 128 inputs & 2128 entries. The tree based comparator is designed with the logic that the generate (G) and propagate (P) signals can be employed for binary comparisons.

A. Design Principle

A two 2-bit binary number (A1A0 & B1B0) comparison can be realized with:

$$Bbig = A1'\,B1 + (A1 \parallel B1)'\,(A0'\,B0) \quad (1)$$

$$EQ = (A1 \parallel B1)'\,(A0 \parallel B0)' \quad (2)$$

The three comparison signals are checked with the following conditions:
For B>A: Bbig =1 and EQ=0; For A>B: Bbig =0 and EQ=0; For A=B Bbig =0 and EQ=1.

The equation (1) is similar to carry signal generation in binary addition. For instance consider the following carry generation:

$Cout = AB + (A \parallel B)Cin$

This implies that Cout can be written as:

$$Cout = G+P\,Cin \qquad (3)$$

Where A and B are binary inputs, Cin and Cout are the carry inputs and carry outputs, and G and P are generate and propagate signals respectively.

Comparing (1) and (3) we have: $G1 = A1'\,B1$

$EQ1 = (A1 \parallel B1)'$

$Cin = (A0'\,B0)$ for Bbig.

The equation (1) may not be suitable for high performance operation due to complicated XNOR operation. An encoding scheme is used to solve this issue. The encoding equation is given as:

$$G[i] = A[i]'\,B[i];\ EQ[i] = (A[i] \parallel B[i])' \qquad (4)$$

Where i= 0...63. The comparison in (1) and (2) can be simplified to:

$$Bbig\ [2j+1:2j] = G\,[2j+1] + EQ\,[2j+1]\,G\,[2j] \qquad (5)$$

$$EQ\,[2j+1:2j] = EQ\,[2j+1]\,EQ\,[2j] \qquad (6)$$

Where j= 0...31.

Therefore, Greater and EQ in a 64-bit comparator can be computed using:

$$Bbig\ [63:0] = G63 + \sum_{k=0}^{62} \left(Gk \cdot \prod_{m=k+1}^{63} EQm \right) \qquad (7)$$

$$EQ[63:0] = \prod_{m=0}^{63} EQm \qquad (8)$$

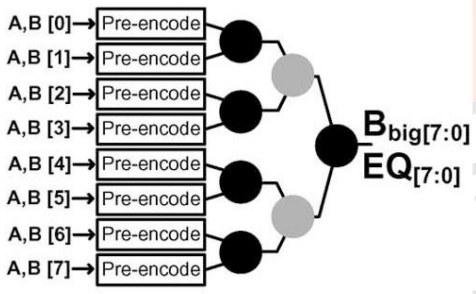


Fig.1 8-bit tree diagram of comparator

B. *Architecture*

Fig. 1 demonstrates an 8-bit version of the proposed tree based comparator.Fig.2 shows the Pre-encode circuitry. 64-bit comparator is here designed in 7 stages. In the 0th stage, modified pass transistor logic style circuitry is employed to produce "less than" & "equal to" outputs. The outputs of 0th stage act as inputs of 1st stage. In 1st stage, CMOS circuitry is employed to produce inverse inputs for stage 2nd. In 2nd stage, CMOS circuitry is employed again to produce actual inputs for stage 3rd. Now, according to tree structure given in Fig. 1, circuitry of first stage is used for third stage. Similarly, for fourth stage, circuitry of second stage is employed. For the fifth stage first stage circuitry is employed. For sixth stage the second stage circuitry employed.

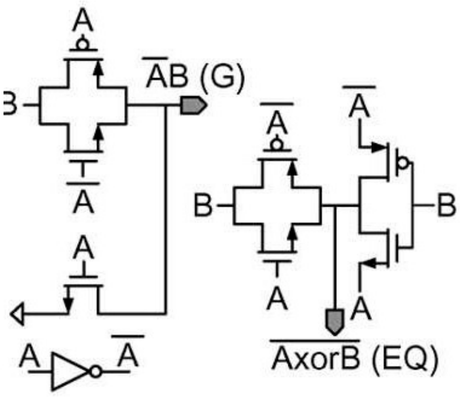


Fig.2 Pre-encode circuitry

IV. MODIFIED 64-BIT COMPARATOR

The proposed design strategy uses a hierarchical design of a fast 64-bit comparator is shown in Fig. 3, which is composed of eight 8-bit comparators and one final 8-bit zero/one comparator. The 64 bits are divided into eight bytes which are evaluated at the same time, and then the 8-bit comparator produces the final output signal.Fig.4, Fig.5 and Fig 6 shows the simulations for modified 64 bit comparator which depicts A<B,A=B and A>B respectively. Fig.7, Fig.8 and Fig 9 shows the simulations for modified 64 bit comparator which depicts A<B,A=B and A>B respectively

.

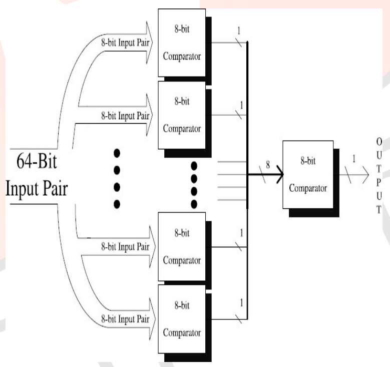
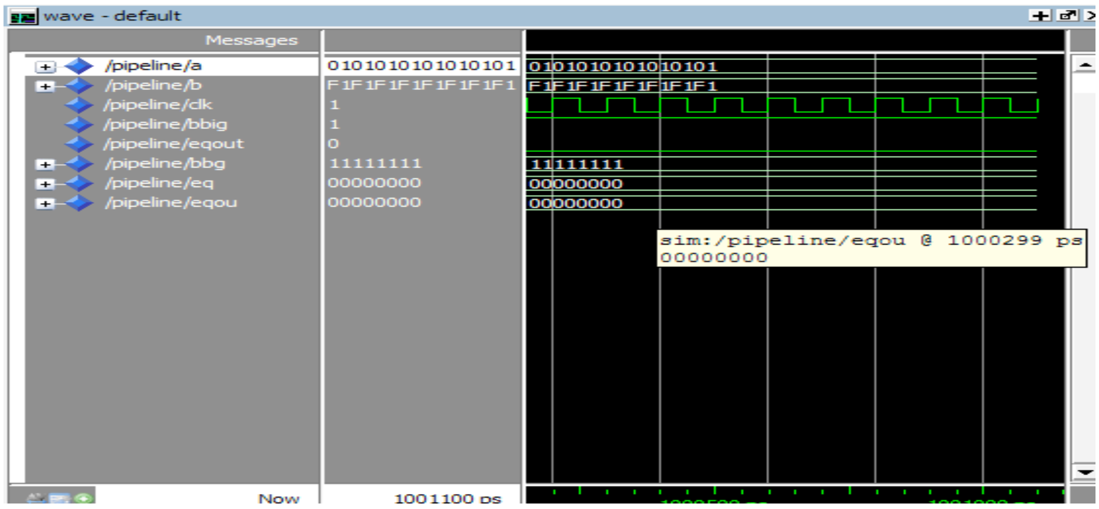


Fig.3 Modified 64-bit Comparator

V. SIMULATIONS AND COMPARISONS

Fig.4 A<B



```
/pipeline/a       0101010101010101
/pipeline/b       F1F1F1F1F1F1F1F1
/pipeline/clk     1
/pipeline/bbig    1
/pipeline/eqout   0
/pipeline/bbg     11111111
/pipeline/eq      00000000
/pipeline/eqou    00000000

sim:/pipeline/eqou @ 1000299 ps
00000000

Now               1001100 ps
```

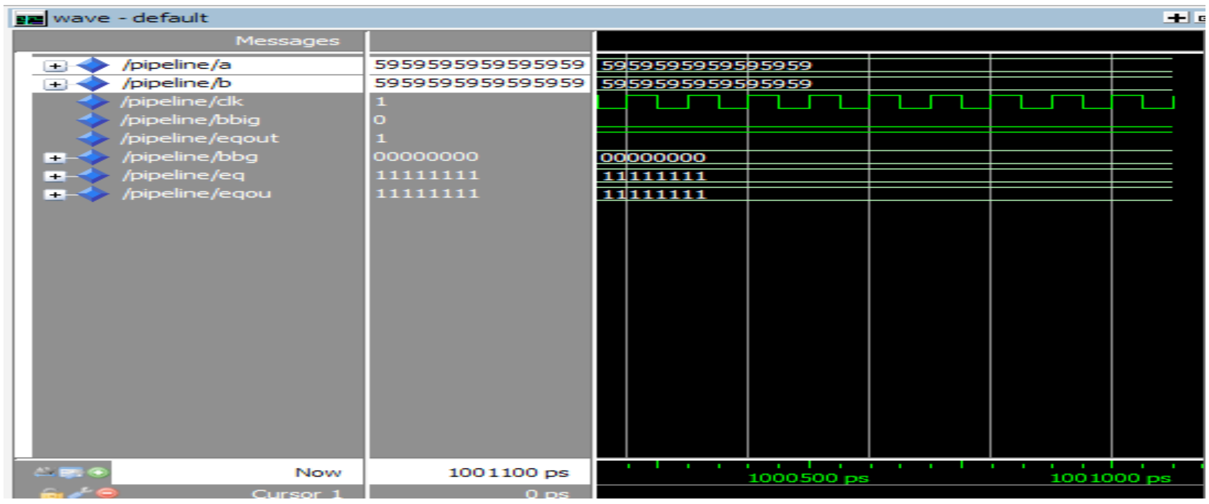


Fig.5 A=B

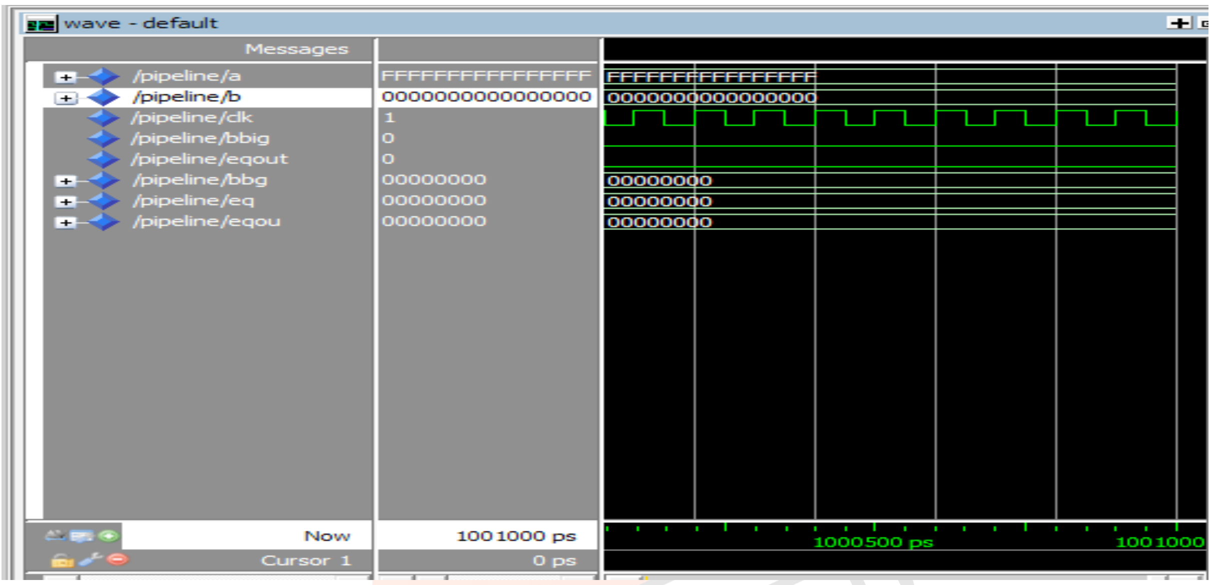
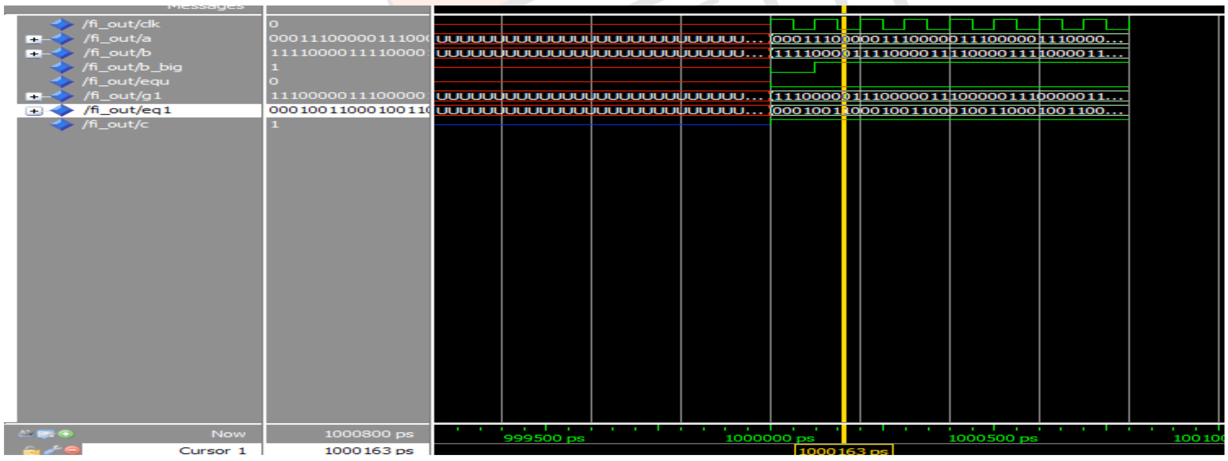


Fig.6 A >B


Messages
/fi_out/clk 0
/fi_out/a 000111000000111000
/fi_out/b 111100001111000
/fi_out/b_big 1
/fi_out/equ 0
/fi_out/g1 11100000111000000
/fi_out/eq1 000100110001001110
/fi_out/c 1
UUUUUUUUUUUUUUUUUUUUUUUUUUUUU...
Now 1000800 ps
Cursor 1 1000163 ps
999500 ps
1000000 ps
1000500 ps
1000163 ps
Fig.7 A <B

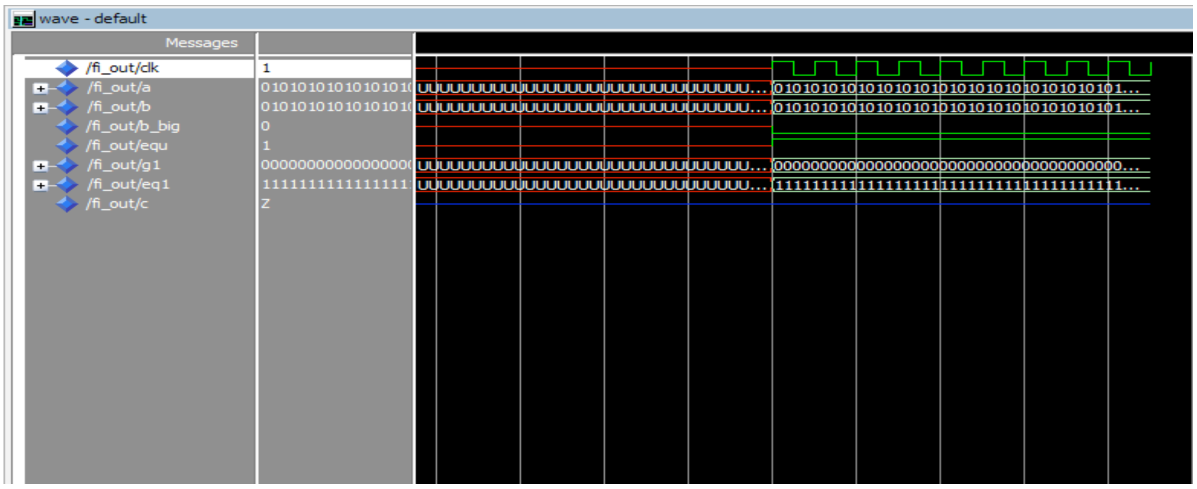

wave - default
Messages
/fi_out/clk 1
/fi_out/a 0101010101010101
/fi_out/b 0101010101010101
/fi_out/b_big 0
/fi_out/equ 1
/fi_out/g1 0000000000000000
/fi_out/eq1 1111111111111111
/fi_out/c Z

Fig.8 A =B

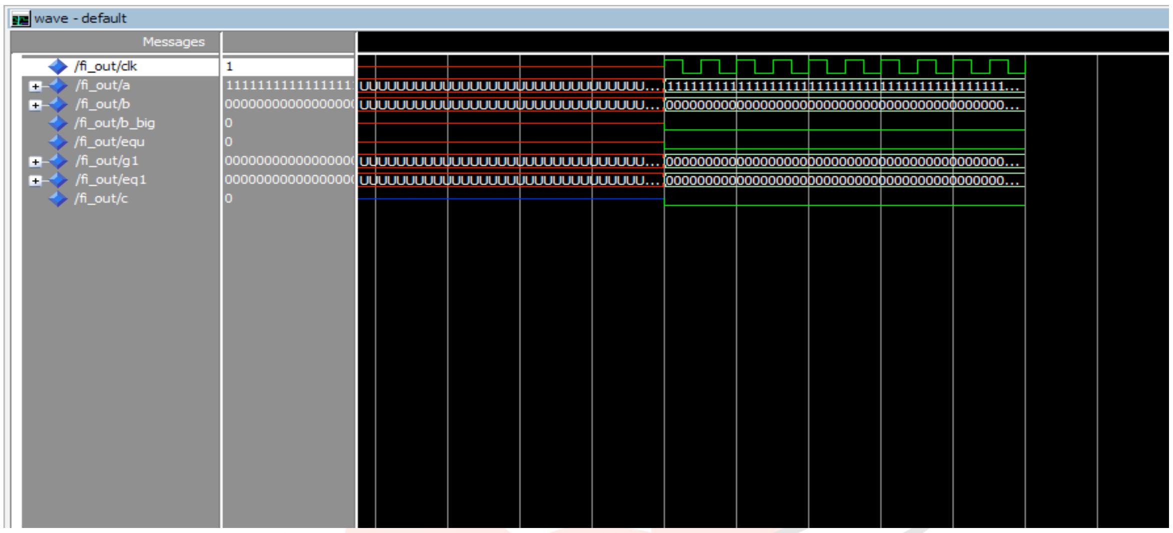

wave - default
Messages
/fi_out/clk 1
/fi_out/a 1111111111111111
/fi_out/b 0000000000000000
/fi_out/b_big 0
/fi_out/equ 0
/fi_out/g1 0000000000000000
/fi_out/eq1 0000000000000000
/fi_out/c 0

Fig.9 A >B

CONCLUSION

With power and area being a limiting factor in high density and high-performance VLSI designs, a great deal of effort has been made to explore low-power and area design options without sacrificing performance. A modified 64 bit comparator is proposed in this paper. Rather than having a general scheme for comparators, different logic should be used for lower fan-in comparators and separate logic for higher fan-in comparators. The entire 64 bit is divided into groups of 8 bits each and given as input to eight comparators and a final comparator gives the output. The proposed method helps to achieve relatively large power savings over a range of supply voltage than other comparators. The comparisons of comparator design are based upon Xilinx design suite 10.1 and using Modelsim6.3 f simulations.

REFERENCES

[1] C.H. Huang and J.S. Wang, "High Performance and power efficient CMOS comparators," IEEE J. Solid- State Circuits, vol.38, no.2, pp.254-262, Feb.2003.
[2] C.C. wang, C.F. Wu and K.C. Tsai, "1 GHz 64-bit high-speed comparator using ANT dynamic logic with two-phase clocking" IEEE Proc. Comput. Digit. Tech., vol.145, no.6, pp.433-436, Nov 1998.
[3] H.M. Lam and C.Y.Tsui,"High performance single clock cycle CMOS comparators," Electron. Lett. vol.42, no.2, pp.75-77, Jan.2006.
[4] H.M. Lam and C.Y.Tsui, "A MUX based high performance single cycle CMOS comparator," IEEE Trans. Circuits Syst. II, Exp.Briefs, vol.54, no.7, pp.591-595, Jul. 2007.
[5] Pierce Chuang, David Li, Manoj Sachdev, "A Low-Power High-Performance Single-Cycle Tree-Based 64-Bit Binary Comparator," IEEE Trans. Circuits Syst. II, Exp Briefs,vol.59, no.2, pp.108-112, Feb. 2012.