Design of High Performance Magnitude Comparator

[1]Kailash Chandra Rout, [2]Sushmita Rath, [3]Belal Ali
[1]Asst. Prof, Department of ECE
[2]Student, Department of ECE
[3]Student, Department of ECE
Gandhi Institute For Technology, Bhubaneswar, 752054

***Abstract*- A single-cycle tree-based binary comparator is realized in a 90-nm, 1.2-V CMOS process is presented in this paper. This novel comparator architecture is specifically designed for faster operation. This brief presents a detailed performance and power analysis of various state-of-the-art comparator designs across two CMOS technologies. At 90-nm technology, with a speed of 43.5ps and 180-nm technology, with a speed of 217.5ps, the proposed design demonstrates 56.936µw and 329µw power, respectively. In addition, the proposed work is 43.13% faster than existing design.**

Index Terms—Binary comparator, digital arithmetic.

I. INTRODUCTION

A binary comparator has always been an important block in an arithmetic logic unit and also has extensive applications in many digital systems, such as decoding of the x86 instructions. The recent emergence of multiple-input–multiple output (MIMO) technology for next-generation communication systems has further aggravated the need for low-power high-performance comparators, because MIMO decoding algorithms require extensive iterations of binary number comparison.

Conventionally, a high-speed adder is the choice of design for high-performance binary comparison, at the cost of both power consumption and area. The heavy pipelining (3.5 clock cycles) requirement for ANT logic, however, made this design unsuitable for a single-cycle operation. A single-cycle, two-phase comparator that relies on priority encoders to decode the first unequal bit away from the most significant bit (MSB). A different priority-encoding (parallel-MSB-checking) algorithm along with a new priority encoder design and a MUX-based comparator structure is proposed in [4]. This implementation achieves superior delay performance compared to previous works, at the expense of both power dissipation and increase in the number of transistors.

All of the aforementioned works achieve high-performance operations using dynamic logic. While dynamic logic has demonstrated superior performance, as compared with static logic, it is not suitable for low-power operation because its data activity factor α is always 0.5. On the other hand, static logic has an empirical α of close to 0.1, making it advantageous in terms of power consumption. In addition, a higher stack height is also less attractive in a deep sub micrometer process, where the VDD/V_t ratio is lower compared with an earlier technology, because transistors will exit the saturation mode sooner and be forced to operate in the linear region.

Recently, tree-based comparators are proposed, where a tree structure, similar to the carry-merge tree of a parallel-prefix adder, is used to facilitate the comparison process. A tree-based comparator is theoretically one of the fastest schemes since the delay to compare two-bit numbers only depends on the logarithm of **N.** This tree structure comparator with a pre-encoding scheme to achieve a maximum stack height of two is proposed.

EXISTING COMPARATOR DESIGN

A brief description of the design principles of existing comparators is provided here.

A binary comparator compares two numbers and produces an output if one number is greater than, less than or equal to other number. If A and B are two numbers then the expression for different format can be derived using the truth table given Table 1.

For this designing, the Domino CMOS logic is used. In domino CMOS logic only non- inverting structures can be implemented using domino CMOS. Also charge sharing between the dynamic stage output node and the intermediate nodes of NMOS logic block may cause erroneous output. Fig. below shows a schematic diagram for 2-bit comparator

A	B	B_{big}	B_{less}	EQ
0	0	0	0	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

Table 1: Truth Table for comparison of two numbers

From Table,

Bbig=B.A'

Bless=A.B'

EQ=AΘB

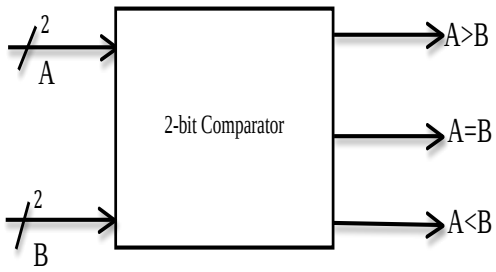

Fig 1: Block Diagram for binary Comparator

Let's consider two bits

A=a1, a0

B=b1, b0

And these two bits A and B comparison can be realized with

B_{big}= a_1'. b_1+ (a_1Θb_1). a_0'. b_0 …….(1)

Equal= (a_1Θ b_1). (a_0 Θ b_0)……...(2)

Fig 2(a) and (b) below shows the schematic diagram of 2-bit comparator.

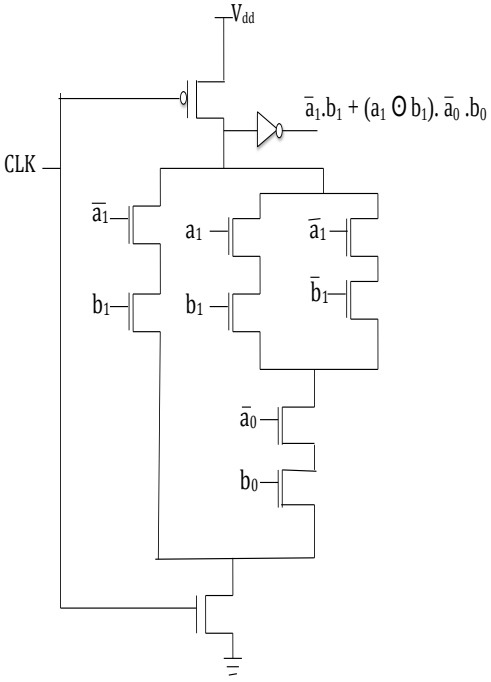

Fig 2(a): Schematic diagram for Existing Comparator (Bbig)

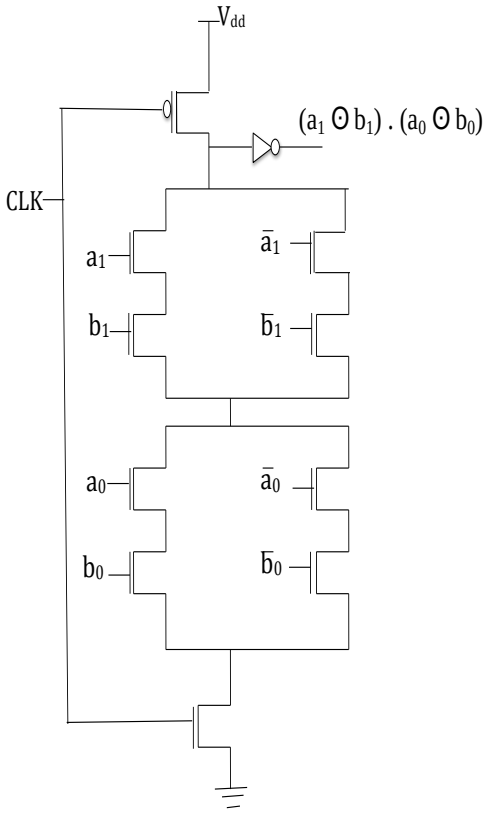

Fig 2(b): Schematic diagram for Existing Comparator (EQ)

TREE BASED COMPARATOR

A tree-based comparator that utilizes dynamic Manchester adders to reduce the number of critical stages to two. A static implementation of this design inevitably incurs performance and area penalties. For instance, a static Manchester adder requires an additional delete signal and has a tall PMOS transistor stack.

The proposed high-performance tree-based comparator is inspired by the fact that G (generate) and P (propagate) signals can be defined for binary comparisons, similar to the G and P signals for binary additions. Hence, the following key observation is made: A binary comparator is essentially a subset of the carry-merge tree in a parallel-prefix adder, where only the final carryout signal is necessary to interpret the result.

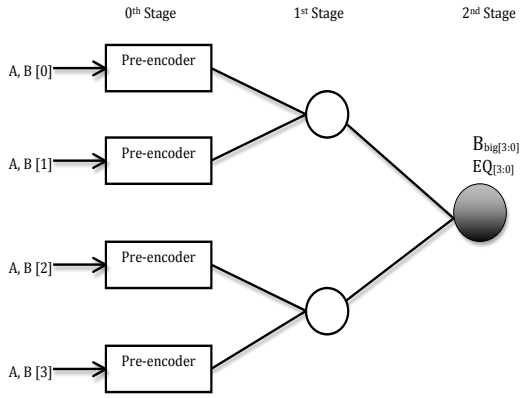

Fig3(a): Block diagram Representation of 4-bit Comparator

Let's consider 4-bit binary numbers $(A_3 \ldots A_0)$& $(B_3 \ldots B_0)$ Then

$B_{big[3:0]}=A3'.B_3+(A3 \odot B3).A_2'.B_2 +(A3 \odot B3).(A2 \odot B2).A1'.B1 +(A3 \odot B3).(A2 \odot B2).(A1 \odot B1)A0'.B_0$ ……(3)

$EQ[3:0] =(A3 \odot B3).(A2 \odot B2).(A1 \odot B1).(A0 \odot B0)$ ……(4)

Equation (3) may not be suitable for high performance operation when implementing with static logic, due to the tall transistor stack height and a complicated XNOR gate. An encoding scheme is employed to mitigate this problem.

The encoding equation is given as

$G[i]=A[i]'.B_{[i]}$……..(5)

$EQ[i]=(A[i] \odot B[i])$……(6)

Where i=0…3.The radix-2 comparison in equation (3) and (4) can be then simplified to

$B_{big[2j+1:2j]} = G_{[2j+1]}+ EQ_{[2j+1]}.G_{[2j]}$…(7)

$EQ_{[2j+1:2j]} = EQ_{[2j+1]} . EQ_{[2j]}$……..(8)

Where j=0…1, G[i] signal is used for B[i]>A[i] and EQ[i] signal is used for A[i]=B[i].

$B_{big[3:0]}=A_3'.B_3+(A_3 \odot B_3).A_2'.B_2 +(A_3 \odot B_3).(A_2 \odot B_2).A_1'.B_1 +(A_3 \odot B_3).(A_2 \odot B_2).(A_1 \odot B_1).A_0'.B_0$

$=A_3'.B_3+(A_3 \odot B_3)[A_2'.B_2+(A_2 \odot B_2) [A_1'.B_1+(A_1 \odot B_1).A_0'.B_0]]$

$=G_3+EQ_3 [G_2+EQ_2 [G_1+EQ_1.G_0]]$

$=B_{big[3:2]}+EQ_{[3:2]}.B_{big[1:0]}$…….....…(9)

Finally, B_{big} and EQ in a N-bit comparator are computed using

$B_{big[N-1:0]}= G_{N-1}+ \sum_{k=0}^{N-2}(Gk . \prod_{m=k-1}^{N} EQm)$

……(10)

$EQ_{[N-1:0]}= \prod_{m=0}^{N-1} EQm$.....(11)

Fig. below shows the circuit diagram of different stages of tree structure.

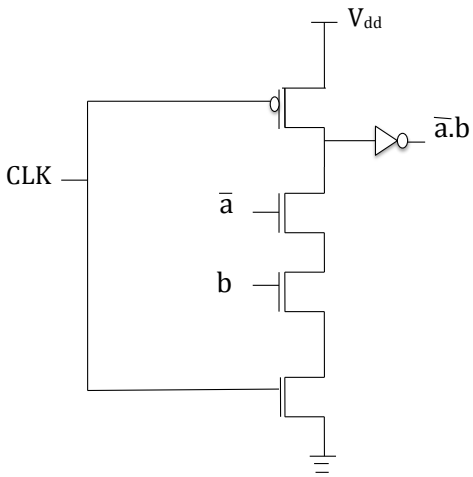

Fig 3(b): Pre-encoder stage for B_{big}

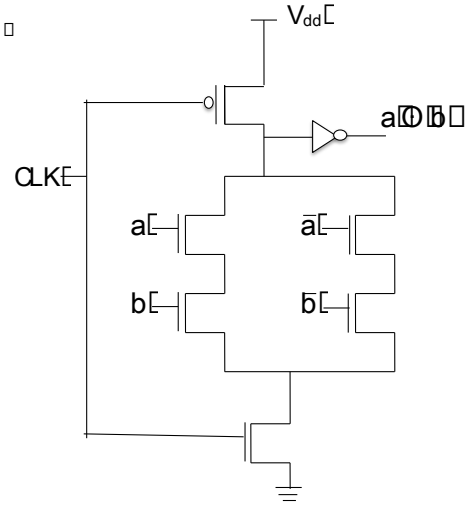

Fig 3(c): Pre-encoder stage for EQ

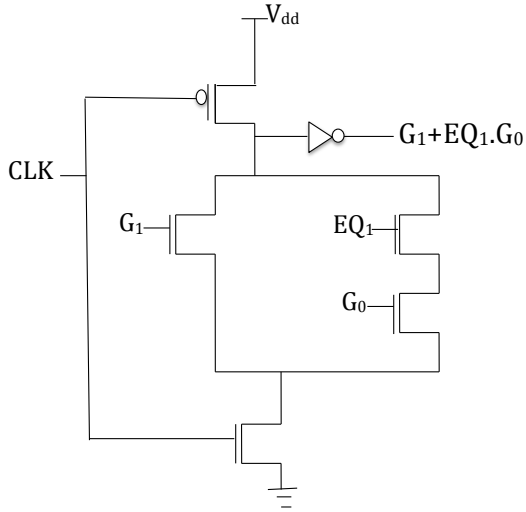

Fig 3(d): Circuit diagram for stage 1(B_{big})

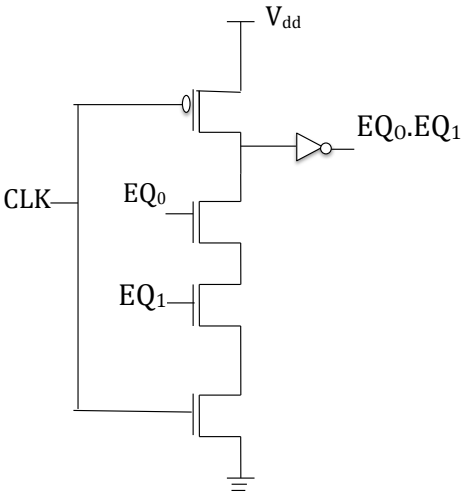

Fig 3(e): Circuit diagram for Stage 1(EQ)

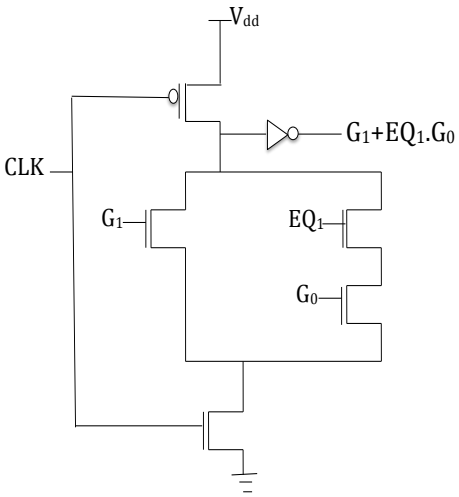

Fig 3(f): Circuit diagram for Stage 2(B_{big})

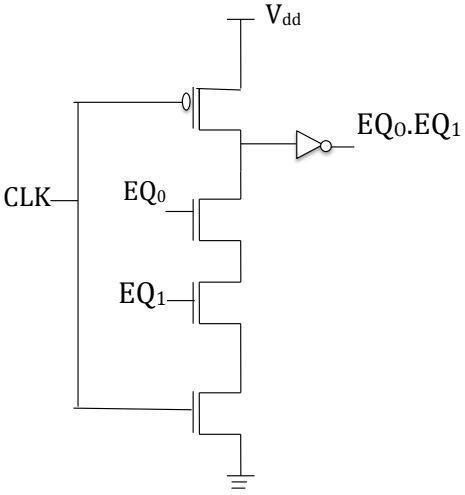

Fig 3(g): Circuit diagram for Stage 2(EQ)

Circuit Layout Implementation

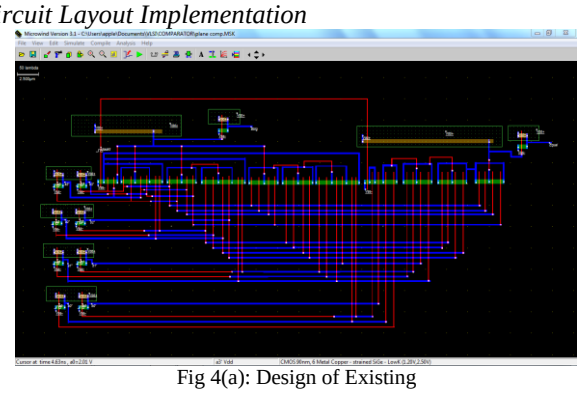

Fig 4(a): Design of Existing Comparator Architecture

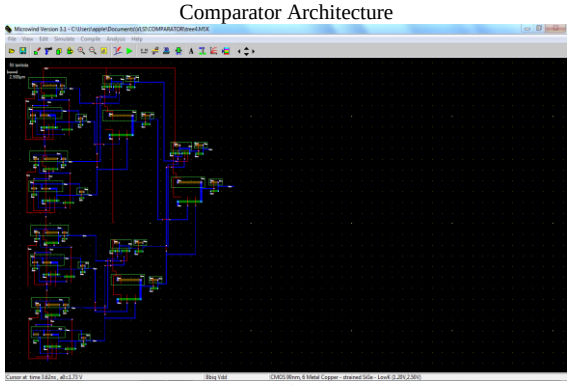

Fig 4(b): Design of Tree Comparator Architecture

Simulation of Comparator Architecture

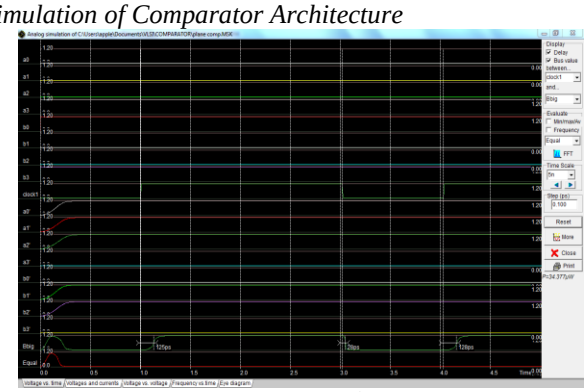

Fig 5(a): Simulation of Existing Comparator Architecture

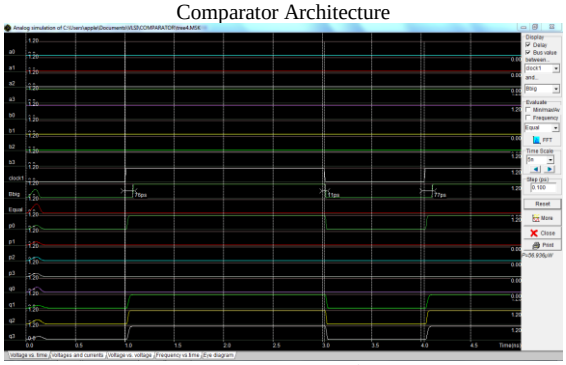

Fig 5(b): Simulation of Tree Comparator Architecture

All simulation runs are done at layout level in the MICROWIND 3.1 design environment using two different CMOS technologies at nominal supply voltage at 27° C.

SIMULATION & COMPARISION

After simulation of 65nm, 90nm and 180nm designs,

final results are obtained for delay and power consumption and are shown in Table 2-3-4 respectively. Simulations have been carried out at these three technologies in MICROWIND 3.1.

Design	Power Consumption (μW)	Propagation Delay (ns)
Existing	5.739	102
Modified	11.034	71

Table 2: Simulation data using 65nm Technology

Design	Power Consumption (μW)	Propagation Delay (ns)
Existing	34.377	76.5
Modified	56.936	43.5

Table 3: Simulation data using 90nm Technology

Design	Power Consumption (μW)	Propagation Delay (ns)
Existing	173	297.5
Modified	329	217.5

Table 4: Simulation data using 180nm Technology

Comparator Architecture	Increased Speed (%)
Using 65nm	30.39
Using 90nm	43.13
Using 180nm	26.89

Table 5: Comparison of speed

CONCLUSION

A single-cycle radix-2 tree-based comparator has been proposed in this work. This design is demonstrated for high-performance operations, as compared with existing architecture and gives an improved delay performance of 43.13% than the existing comparator architecture.

ACKNOWLEDGEMENT

We are thankful to the Dean and HOD, Department of Electronics & Communication Engineering for providing us necessary permission to carry out this work.

REFERENCES

[1] S. M. Kang, Y. Leblebici, 'CMOS Digital Integrated Circuits: Analysis & Design', TATA McGraw- Hill Publication, 3e, 2003.
[2] M. Mano, 'Digital Electronic Circuit', TATA McGraw- Hill Publication, 3e, 2003.
[3] P. Chuang, M. Sachdev, D. Lii, "A Low-Power High-Performance Single-Cycle Tree-Based 64-Bit Binary Comparator"IEEE Trans. Cir. & syst. II, vol.59, no.2, pp.489-493, Feb. 2012.
[4] 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