A Low Power 8-bit Magnitude Comparator With Small Transistor Count Using STATIC CMOS Logic

Bhaskara Rao Doddi, B.N. Srinivasa Rao, R. Prasad Rao
Avanthi Institute of Engineering and Technology, Visakhapatnam.

Abstract — COMPARATOR is the basic module in digital system. It is widely used in communication and calculation areas. Traditional comparator circuit is based on truth table leads to high power consumption, low speed and increased area. The main objective of this paper is to provide new low power, area solution for very large scale integration (VLSI) designers. At circuit level, STATIC CMOS logic style can give better results over others when we design efficiently. In this project the proposed comparator has been designed by using STATIC CMOS 180nm TECHNOLOGY. Layout for comparator has been implemented by using Tanner tool.

Index Terms — logic block, logic carrying block, magnitude comparator, static cmos.

I. INTRODUCTION

Low density parity check (Ldpc) was designed by gallager in 1962[1]. Implementation of ldpc decoder consumes more power, comparator is important module in decoder and it is also used in digital system C-H HUANG[2] developed priority encoder based on logic and module. S-W CHENG[3] employed conditional sum adder to design efficient comparator. J-Y KIM and H-J YOO[4] proposed a design without arithmetic operation which is even more efficient. Now we need to design a comparator by following our own technique such that low power and low area is achieved. We have designed comparator by using STATIC CMOS logic style, it has advantage of low power consumption but dis-advantage of area with respect to number of transistors, so here we will have the challenge to design the comparator with less number of transistors. Logic style like PTL(Pass Transistor Logic) has advantage of less number of transistors it has the problem of logic level degradation.

II. MAGNITUDE COMPARATOR

Comparator is going to compare 8 bits of A(A7 to A0) and 8 bits of B(B7 to B0) and decides whether AGTB(A>B) or ALTB(A<B) or AEQB(A=B), if we can design the circuit for any two outputs then we can arrange the third one but which two outputs to design is the concern, if we try to design AEQB(A=B) as one of the outputs might not be a good idea, because we need to check all the 8-bits of A and B then only we can say AEQB(A=B) is true or not, so circuit has been designed for AGTB(A>B), ALTB(A<B). now how can circuit be designed for AGTB(A>B), ALTB(A<B) we need to know what are all the conditions for AGTB(A>B), ALTB(A<B) to become true, if (A7>B7) then AGTB(A>B) should be true, so we have designed a sub-circuit for this condition and if (A7<B7) then ALTB(A<B) should be true, so we have designed a sub-circuit for this condition so till now we have designed 1-bit comparator which is logic block which has the two sub-circuits.

Now we need to design 2-bit comparator here if (A6>B6) and ALTB(A<B) is not true in 1-bit comparator , AGTB(A>B) is true in 1-bit comparator then AGTB(A>B) should be true so we have designed a sub-circuit for this condition and if (A6<B6) and AGTB(A>B) is not true in 1-bit comparator , ALTB(A<B) is true in 1-bit comparator then ALTB(A<B) should be true so we have designed a sub-circuit for this condition for this logic carrying block was designed.which has the two sub-circuits.

Now we need to design 3-bit comparator here if (A5>B5) and ALTB(A<B) is not true in 2-bit comparator , AGTB(A>B) is true in 2-bit comparator then AGTB(A>B) should be true and if (A5<B5) and AGTB(A>B) is not true in 2-bit comparator , ALTB(A<B) is true in 2-bit comparator then ALTB(A<B) should be true for this condition logic carrying block was reused.

Now we need to design 4-bit comparator here if (A4>B4) and ALTB(A<B) is not true in 3-bit comparator , AGTB(A>B) is true in 3-bit comparator then AGTB(A>B) should be true and if (A4<B4) and AGTB(A>B) is not true in 3-bit comparator , ALTB(A<B) is true in 3-bit comparator then ALTB(A<B) should be true for this condition logic carrying block was reused.

Now we need to design 5-bit comparator here if (A3>B3) and ALTB(A<B) is not true in 4-bit comparator , AGTB(A>B) is true in 4-bit comparator then AGTB(A>B) should be true and if (A3<B3) and AGTB(A>B) is not true in 4-bit comparator , ALTB(A<B) is true in 4-bit comparator then ALTB(A<B) should be true for this condition logic carrying block was reused.

Now we need to design 6-bit comparator here if (A2>B2) and ALTB(A<B) is not true in 5-bit comparator , AGTB(A>B) is true in 5-bit comparator then AGTB(A>B) should be true and if (A2<B2) and AGTB(A>B) is not true in 5-bit comparator , ALTB(A<B) is true in 5-bit comparator.
then $\text{ALT}(A<B)$ should be true for this condition logic carrying block was reused.

Now we need to design 7-bit comparator here if $(A>1B)$ and $\text{ALT}(A<B)$ is not true in 6-bit comparator, $\text{AGT}(A>B)$ is true in 6-bit comparator then $\text{AGT}(A>B)$ should be true and if $(A1<B1)$ and $\text{AGT}(A>B)$ is not true in 6-bit comparator, $\text{ALT}(A<B)$ is true in 6-bit comparator then $\text{ALT}(A<B)$ should be true for this condition logic carrying block was reused.

Now we need to design 8-bit comparator here if $(A0>B0)$ and $\text{ALT}(A<B)$ is not true in 7-bit comparator, $\text{AGT}(A>B)$ is true in 7-bit comparator then $\text{AGT}(A>B)$ should be true and if $(A0<B0)$ and $\text{AGT}(A>B)$ is not true in 7-bit comparator, $\text{ALT}(A<B)$ is true in 7-bit comparator then $\text{ALT}(A<B)$ should be true for this condition logic carrying block was reused.

There are 2 blocks in this comparator named Logic block and Logic carrying block. Logic block compares MSB(Most Significant Bit) of A and B that is $A7,B7$. Logic carrying block compares MSB(Most Significant Bit)-1 to LSB(Least Significant Bit) of A and B. so for an N-bit comparator there will be $(N-1)$ Logic carrying blocks and 1 Logic block. Logic block, Logic carrying block, Inverter and Nor gate are used in our design.
Logic block has two inputs and two outputs, inputs are A7 ,B7 and outputs are AGTB(A>B) and ALTB(A<B) and if these are “00” means that AGTB(A>B) is TRUE, “11” means that ALTB(A<B) is TRUE, “10” means that decision is postponed to next level, “01” never happens. Logic carrying block(MSB(Most significant bit) -1,MSB(Most significant bit) -3,MSB(Most significant bit) -5,MSB(Most significant bit) -7) has four inputs ainput , binput , AGTB(A>B) and ALTB(A<B) and two outputs AGTB(A>B)OUTPUT and ALTB(A<B) OUTPUT . If AGTB(A>B) is ‘0’ then AGTB(A>B)OUTPUT is ‘1’ and also if ALTB(A<B)OUTPUT is ‘0’ and A6=’1’ and B6=’0’ then also AGTB(A>B)OUTPUT is ‘1’ else AGTB(A>B) OUTPUT is ‘0’. If AGTB(A>B) is ‘1’ then ALTB(A<B) OUTPUT is ‘0’ and also if AGTB(A>B) is ’1’ and A6=’0’ and B6=’1’ then also ALTB(A<B)OUTPUT is ‘0’ else ALTB(A<B)OUTPUT is ‘1’. If (AGTB(A>B)OUTPUT and ALTB(A<B)OUTPUT) are ‘01’ decision is postponed to next level, “00” ALTB(A<B) is TRUE, “11” AGTB(A>B) is TRUE, “10” never happens.

In Logic carrying block(MSB(Most significant bit) -2,MSB(Most significant bit) -4,MSB(Most significant bit) -6) If (AGTB(A>B)OUTPUT and ALTB(A<B)OUTPUT) are “10” decision is postponed to next level, “00” AGTB(A>B) is TRUE, “11” ALTB(A<B) is TRUE, “01” never happens.

In MSB(Most significant bit) -1 Logic carrying block ainput is A6 and binput is B6 and AGTB(A>B) is AGTB(A>B) of Logic Block and ALTB(A<B) is ALTB(A<B) of Logic Block.

In MSB(Most significant bit) -3 Logic carrying block ainput is A4 and binput is B4 and AGTB(A>B) is AGTB(A>B)OUTPUT of MSB(Most significant bit) -2 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-2 Logic carrying block.

In MSB(Most significant bit) -5 Logic carrying block ainput is A2 and binput is B2 and AGTB(A>B) is AGTB(A>B)OUTPUT of MSB(Most significant bit) -4 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-4 Logic carrying block.

In MSB(Most significant bit) -7 Logic carrying block ainput is A0 and binput is B0 and AGTB(A>B) is AGTB(A>B)OUTPUT of MSB(Most significant bit) -6 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-6 Logic carrying block.

In MSB(Most significant bit) -2 Logic carrying block ainput is B5 and binput is A5 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -1 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-1 Logic carrying block and here outputs will be interchanged.

In MSB(Most significant bit) -4 Logic carrying block ainput is B3 and binput is A3 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -3 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-3 Logic carrying block and here outputs will be interchanged.

In MSB(Most significant bit) -6 Logic carrying block ainput is B1 and binput is A1 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -5 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-5 Logic carrying block and here outputs will be interchanged.

In MSB(Most significant bit) -8 Logic carrying block ainput is B0 and binput is A0 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -7 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-7 Logic carrying block.

In MSB(Most significant bit) -7 Logic carrying block ainput is B5 and binput is A5 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -6 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-6 Logic carrying block and here outputs will be interchanged.

AGTB(A>B)OUTPUT is ALTB(A<B)OUTPUT and ALTB(A<B)OUTPUT is AGTB(A>B)OUTPUT. But the block is flipped vertically in our block diagram so AGTB(A>B)OUTPUT is AGTB(A>B)OUTPUT and ALTB(A<B)OUTPUT is ALTB(A<B)OUTPUT.

In MSB(Most significant bit) -4 Logic carrying block ainput is B3 and binput is A3 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -3 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-3 Logic carrying block and here outputs will be interchanged.

In MSB(Most significant bit) -6 Logic carrying block ainput is B1 and binput is A1 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -5 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-5 Logic carrying block and here outputs will be interchanged.

In MSB(Most significant bit) -8 Logic carrying block ainput is B0 and binput is A0 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -7 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-7 Logic carrying block.

In MSB(Most significant bit) -7 Logic carrying block ainput is B5 and binput is A5 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -6 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-6 Logic carrying block and here outputs will be interchanged.

AGTB(A>B)OUTPUT is ALTB(A<B)OUTPUT and ALTB(A<B)OUTPUT is AGTB(A>B)OUTPUT. But the block is flipped vertically in our block diagram so AGTB(A>B)OUTPUT is AGTB(A>B)OUTPUT and ALTB(A<B)OUTPUT is ALTB(A<B)OUTPUT.

In MSB(Most significant bit) -4 Logic carrying block ainput is B3 and binput is A3 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -3 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-3 Logic carrying block and here outputs will be interchanged.

In MSB(Most significant bit) -6 Logic carrying block ainput is B1 and binput is A1 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -5 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-5 Logic carrying block and here outputs will be interchanged.

In MSB(Most significant bit) -8 Logic carrying block ainput is B0 and binput is A0 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -7 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-7 Logic carrying block.

In MSB(Most significant bit) -7 Logic carrying block ainput is B5 and binput is A5 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -6 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-6 Logic carrying block and here outputs will be interchanged.

AGTB(A>B)OUTPUT is ALTB(A<B)OUTPUT and ALTB(A<B)OUTPUT is AGTB(A>B)OUTPUT. But the block is flipped vertically in our block diagram so AGTB(A>B)OUTPUT is AGTB(A>B)OUTPUT and ALTB(A<B)OUTPUT is ALTB(A<B)OUTPUT.

In MSB(Most significant bit) -4 Logic carrying block ainput is B3 and binput is A3 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -3 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-3 Logic carrying block and here outputs will be interchanged.

In MSB(Most significant bit) -6 Logic carrying block ainput is B1 and binput is A1 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -5 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-5 Logic carrying block and here outputs will be interchanged.

In MSB(Most significant bit) -8 Logic carrying block ainput is B0 and binput is A0 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -7 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-7 Logic carrying block.

In MSB(Most significant bit) -7 Logic carrying block ainput is B5 and binput is A5 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -6 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-6 Logic carrying block and here outputs will be interchanged.

AGTB(A>B)OUTPUT is ALTB(A<B)OUTPUT and ALTB(A<B)OUTPUT is AGTB(A>B)OUTPUT. But the block is flipped vertically in our block diagram so AGTB(A>B)OUTPUT is AGTB(A>B)OUTPUT and ALTB(A<B)OUTPUT is ALTB(A<B)OUTPUT.

In MSB(Most significant bit) -4 Logic carrying block ainput is B3 and binput is A3 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -3 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-3 Logic carrying block and here outputs will be interchanged.

In MSB(Most significant bit) -6 Logic carrying block ainput is B1 and binput is A1 and AGTB(A>B) is ALTB(A<B)OUTPUT of MSB(Most significant bit) -5 Logic carrying block and ALTB(A<B) is ALTB(A<B)OUTPUT of MSB(Most significant bit)-5 Logic carrying block and here outputs will be interchanged.
IV. PERFORMANCE ANALYSIS

Implementation of 8-bit magnitude comparator has been done using STATIC CMOS logic style. Table 1 shows AREA comparision of IMPROVED HYBRID and STATIC CMOS. Power dissipation comparison for 8-bit magnitude comparator using IMPROVED HYBRID and STATIC CMOS for various supply voltage(VDD) are shown in TABLE 2. This table clearly shows that STATIC CMOS has very less power dissipation than IMPROVED HYBRID over various supply voltage(VDD). The comparator which we designed using STATIC CMOS logic style uses less number of transistors. It uses 7.7% less area(number of transistors) than IMPROVED HYBRID. Layout Area of the comparator is 33.2 µm x 26.8 µm equal to 889.76 µm² which is far better than many of the comparator’s mentioned in the references.

TABLE 1: COMPARISON OF AREA IN TWO DESIGNS

<table>
<thead>
<tr>
<th></th>
<th>Design in reference (Improved hybrid)</th>
<th>Design in this paper (Static cmos)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area(number of transistors)</td>
<td>154</td>
<td>142</td>
</tr>
</tbody>
</table>

TABLE 2: COMPARISON OF POWER IN TWO DESIGNS

<table>
<thead>
<tr>
<th>VDD(V)</th>
<th>Average power consumption(µ W)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Design in reference</td>
</tr>
<tr>
<td></td>
<td>(Improved hybrid)</td>
</tr>
<tr>
<td>1.4</td>
<td>21.4</td>
</tr>
<tr>
<td>1.2</td>
<td>12</td>
</tr>
<tr>
<td>1</td>
<td>6.05</td>
</tr>
</tbody>
</table>

V. CONCLUSION

Power, area and speed are the three important constraints, even though we want all these things to be effectively achieved it might not be possible always and also we might look for low-power consumption and also low-area depending upon the requirement by sacrificing the performance up to some extent even though we have designed our comparator by using STATIC CMOS logic style which generally requires more number of transistors, we are able to design the comparator by using less number of transistors than IMPROVED HYBRID which is a mix of HYBRID PTL / CMOS logic style. It shows an 8-bit comparator of the proposed technique only needs 142 transistors, the technique which we presented can be easily extendable up to 64-bit.

ACKNOWLEDGEMENT

This material is based upon work supported by the students of Avanthi Institute of Engineering and Technology. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily react the views of AIET.

REFERENCES

Engineering and Technology, Visakhapatnam, Andhra Pradesh, India. He has 6 years teaching and 9 years industrial experience. He has 11 publications in various International Journals. His area of interest is VLSI Semi and full custom design. He guided many projects for B.Tech and M.Tech students.

R. Prasada rao received his B.Tech degree in Electronics and Communication Engineering from Andhra University, Visakhapatnam, India and M.Tech in VLSI System Design from JNT University, Hyderabad, India. He is currently working as an Assistant Professor in Avanthi Institute of Engineering and Technology, Visakhapatnam, Andhra Pradesh, India. He has 8 years teaching experience. He has 2 publications in various International conferences. His area of interest is Digital Electronics, Signals and Systems and Signal Processing. He guided many projects for B.Tech and M.Tech students.