

ASIC Implementation of Lossless High Speed Serial Compression Using X-Match Process

T. Subha Sri Lakshmi

¹CVR College of Engineering, Department of ECE, Hyderabad, INDIA

Email :

Abstract— The paper presents a unique Very Large Scale Integrated Circuit structural design for high-speed data compressor design which implements the X-Match process. The top level block diagram mainly consists of 5 units, namely, First in First out (FIFO), Match unit logic, CAM (content addressable memory) Comparator, X-match Unit, and Output-stage (DE-x-match) unit. The content-addressable memory (CAM) unit produces a set of hit signals which identify those positions whose symbols in a specified window are the same as the input symbol. These hit signals are then passed to the X-match unit which determines both match length and location to form the kernel of compressed data. These two items are then passed to the output-stage unit for packetisation before being sent out. Logic density increases have made feasible the implementation of multiprocessor systems which are able to meet the intensive data processing demands of highly concurrent systems. This design involves trade off that affects the compression performance, latency, and throughput. The design is implemented in ASIC. Ncvlog simulator is used for simulation, RTL Compiler is used for Schematics and to get reports like area, power, timing, and SOC Encounter tool is used for Synthesis (Floor Plan, Partition, Routing, Pre CTS & Post CTS Synthesis, Clock Tree Synthesis) and finally the GDS II file – a complete chip fabrication is obtained.

Index Terms—FIFO, Match Logic Unit, Cam Comparator, X-Match Pro, De-X Match Pro, ASIC Cadence- RTL Compiler, SOC Encounter.

I. INTRODUCTION

There are drawbacks in the existing parallel compression techniques, like latency, high space, high power etc. So in order to eliminate those drawbacks and to achieve excellent redundancy and compression of data this high speed serial data compression has been adopted. The basic idea of this paper is to implement X-Match and De X-Match blocks of the compressor in Hardware Description Language and testing them on ASIC Cadence Platform using TSMC 45 nm technology libraries. [1] The paper includes the compression of either sequential data or improperly ordered data. The block diagram of high speed serial data compression is shown in the figure 1. The architecture mainly consists of 5 units namely, FIFO, Match logic unit, CAM, comparator, X-match Unit, and Output stage (DE-x-match) unit. The CAM unit produces a set of hit signals which identify those positions whose symbols in a specified window are the same as the input symbol. These hits signals are then passed to the X-match

unit which determines both match length and location to form the kernel of compressed data.

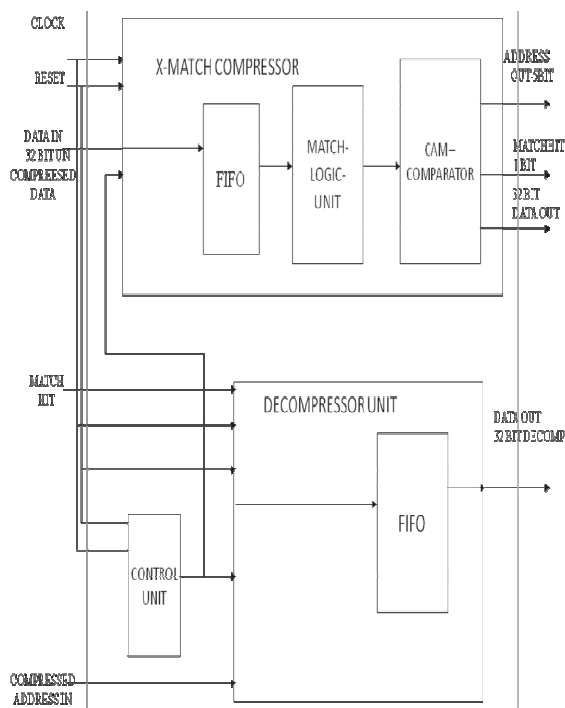


Fig 1: Block Diagram of High Speed Serial Data Compression

II. BLOCKS OF SERIAL DATA COMPRESSION

A) Match Logic Unit

Consider the match logic unit to compare the binary numbers containing two bits each, as shown in figure 2. A1, A0 and B1, B0 are the two 2-bit binary numbers. These are connected to the two exclusive-or gates; in which the LSB's of A0, B0 are given to the first XOR gate and MSB's of A1, B1 are given to the second XOR gate .

B) Operation

If the two numbers are equal, then the outputs of the two XOR gates are at 0 levels. These outputs are inverted and applied to the AND gate. This causes the AND gate output as the HIGH (1) level. If the two numbers are not equal, then the outputs of the two XOR gates are at 1

level. These outputs are inverted and applied to the AND gate. This causes the AND gate output as at LOW (0) level. The basic building elements of a comparator are XOR, NOT and AND gate. The Match Logic Unit circuit is also known as “Comparator”.

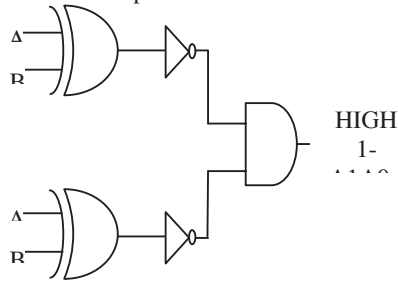


Fig 2: Structure of 2 – bit Match Logic Unit

C) Content Addressable Memory

The CAM behaves as a SRAM which itself can perform READ OR WRITE operation with a given address and data but also performs matching operations to generate hit signals as shown in figure 3. Matching asserts a match line output for each word of the CAM that contains a specified key. A common application of CAM is translation look aside buffers in microprocessors supporting virtual memory. The virtual address is given as a key to the TLB CAM. If this address is in the CAM, the corresponding match line is asserted. This match line can serve as the word line to access as RAM containing the associated physical address, as shown in figure 4.

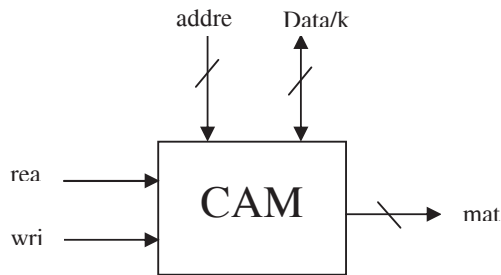


Fig 3: Structure of CAM

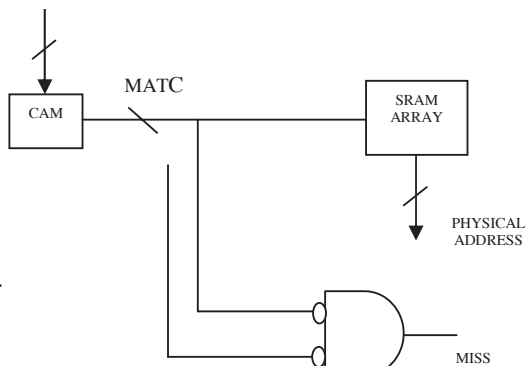


Fig 4: TLB using CAM Cell

Figure 5 shows another CAM cell design with one transistor less. N1 and N2 perform an XOR of the key and

cell data. If the value is not accepted, N3 is switched on to pull down the word line. However, the gate of N3 sees a degraded high logic level. Figure 6 shows a complete 4*4 CAM array structure. Like an SRAM, it consists of an array of cells, a decoder, and column circuitry. However, each row also produces a dynamic match line. The match lines are pre-charged with the clocked pCMOS transistors. The miss signal is produced with a distributed pseudo-NMOS NOR.

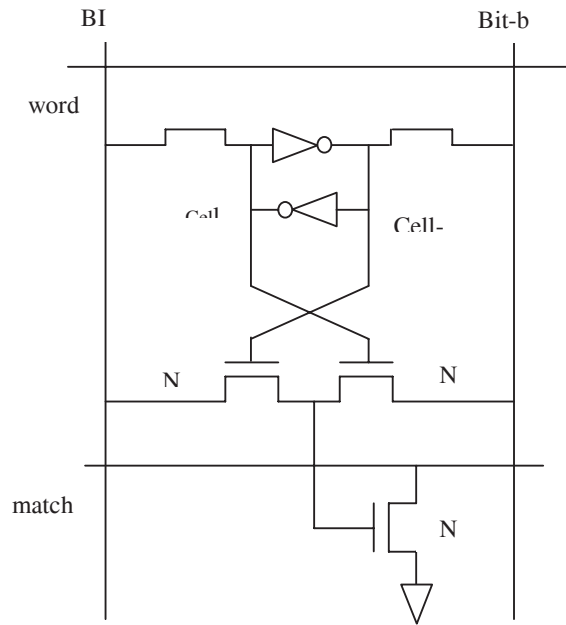


Fig 5: CAM Cell Implementation

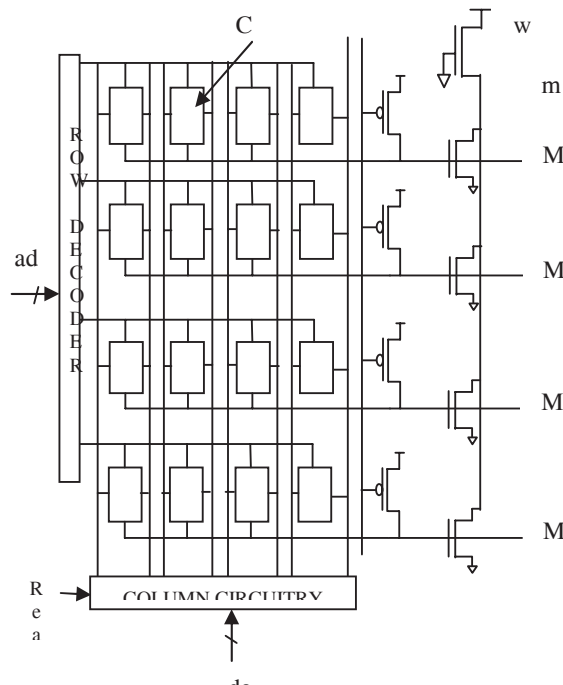


Fig 6: 4*4 CAM Array

D) Cam Comparator

The CAM Comparator consists of 4 input signals i.e., clock, reset, start bit and data input. 4 output signals signal address, address, data output and match hit output as shown in the figure 7. The reset and start consists of 1 bit data information and whereas data input and data output consists 32-bit. The signal address, address consists 6 bit. The input to the CAM is given through the data input signal, then it will store temporarily in signal address and finally it will store permanently in address field, after storing in address field then perfect data output will be fetching outside of the CAM comparator cell. [5]



From the above example at first clock we take input as 2 then when we click on 2nd clock .The data will be stored temporarily in 1.But it will be showing output 2. When we click on 3rd clock the data will be stored permanently in address then it shows perfect output 2.

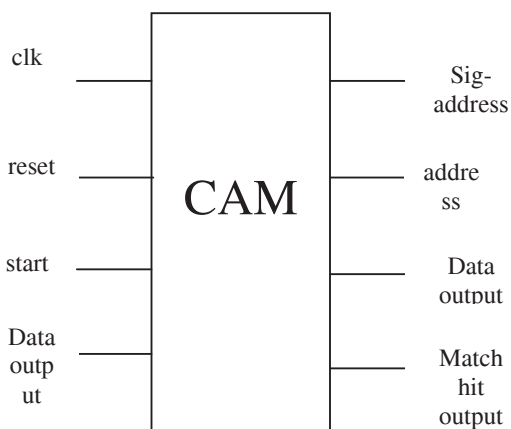


Fig 7: CAM Comparator Cell Implementation

E) Xmatch Pro Operation

The basic use of x match is to store given data. Here clock, reset, start and data in are present which act as input signals as shown in the figure 8. Reset and start are having one bit input, but data input is having 32 bits. Here we assign '0' for the reset and start next input is given to data input. Here data output, and x match signals act as output. For the given data input, the respective output is obtained. This data is stored in any of the 32 bit address lines.

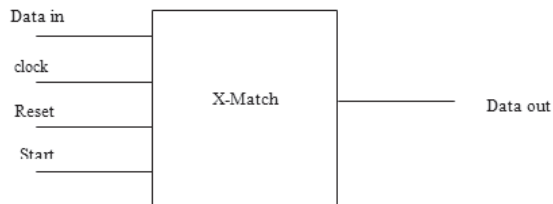


Fig 8: XMatch Pro Block

F) DE X-Match Pro Operation

De-x match is completely opposite to that of x-match in which the data will be compressed and the corresponding address will come as an output i.e , instead of getting compressed data at the output address will be displayed on it. But here de-x match acts as a receiver. The output of x-match will act as an input of de-x match. i.e address in the x-match will act as an input of de-x match . When we apply input (address) at the de-x match the output will be data.

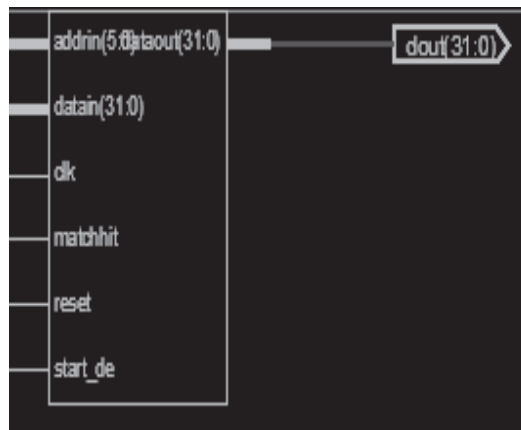


Fig 9: De XMatch Pro Block

G) Top Level Block Implementation

1, 2, 3, 4, 5 as a data input then data output and address output will be appear same as data input i.e., 1, 2, 3, 4, 5. This resembles that whatever the bit size is represented as an input the same will appear at the output without any lossless.

IV. IMPLEMENTATION RESULTS

All the blocks of high speed serial data compression using X Match pro process are implemented by using ASIC Cadence 45 nm technology libraries. Figure 10 shows RTL Schematics of Top level block, Figure 11 shows RTL Schematic of Match Logic Unit, Figure 12 shows RTL Schematic of CAM comparator. Figure 13 shows timing report of Top level block, Figure 14 shows power analysis report of Top level block and Figure 15 shows area analysis and finally the figure 16 shows the IC chip fabrication layout structure which is named as GDS II file.

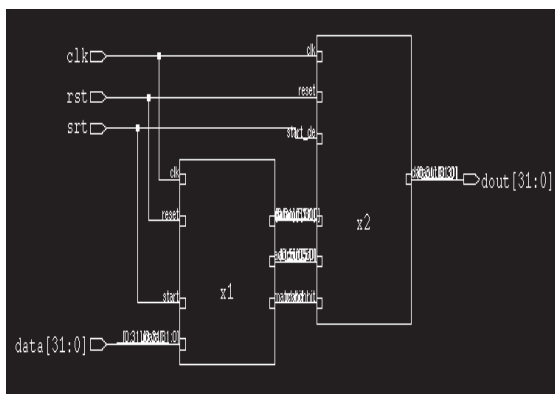


Fig 10: RTL Schematic of Top Level Block

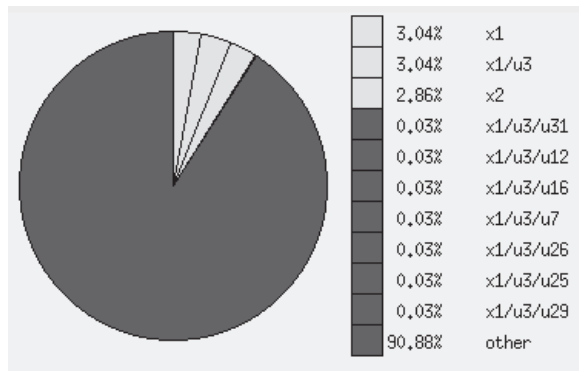


Fig 14: Power Analysis Report of Top Level Block

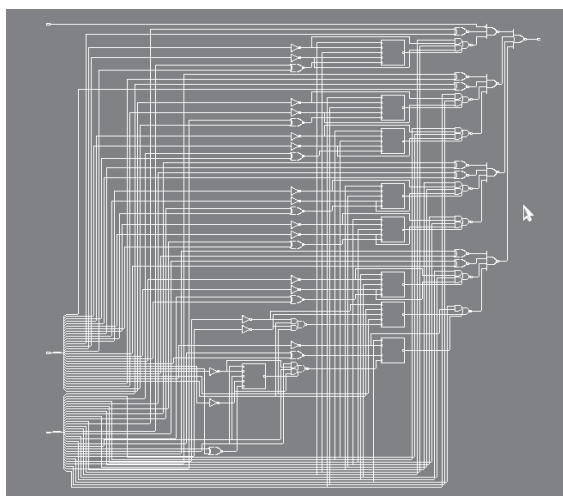


Fig 11: RTL Schematic of Match Logic Unit

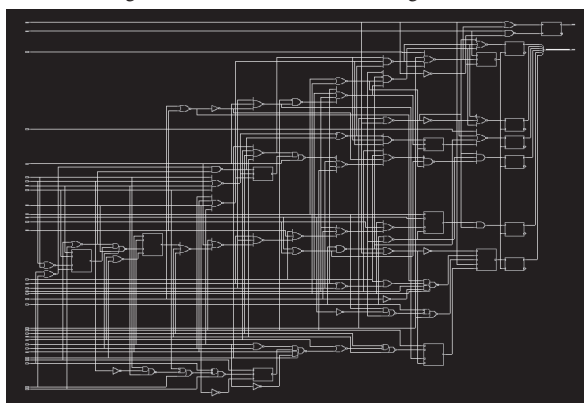


Fig 12: RTL Schematic of CAM Comparator

Endpoint	Stack (ps)	Rise Slew (ps)	Fall Slew (ps)
x1/u3W_addr_reg[31]D	5282.00	22.50	23.00

Fig 13: Timing Report of Top Level Block

report Datapath Area

Generated by: Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1 (Apr 25 2014)
 Generated on: Sep 19 2015 13:23:43
 Module: top32
 Technology library: slow_vdd1v0 1.0
 Operating conditions: PVT_0P9V_125C (balanced_tree)
 Wireload mode: enclosed

Type	Cell Area	Area %
datapath	115.25	0.36
external	0.00	0.00
others	31761.89	99.64
TOTAL	31877.14	100.00

Fig 15: Area Analysis

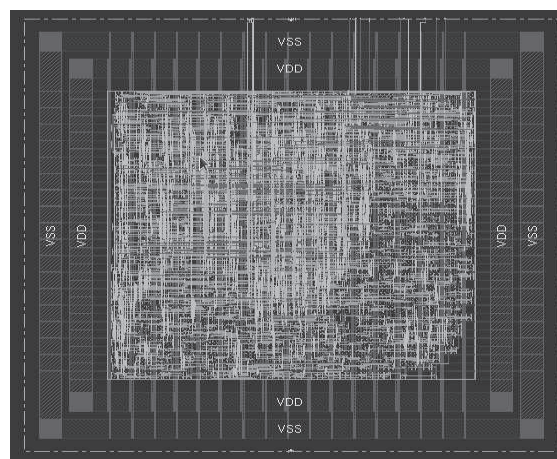


Fig 16: GDS II file of High Speed Serial Data

V.CONCLUSION

All the block sets are verified by ncvhdl simulator and synthesized by using RTL Compiler and finally implemented on SOC Encounter and obtained IC chip layout i.e., GDS II file. The main advantage of Xmatch pro process is high throughput. The improved Compression ratio is achieved in parallel Compression architecture with least increase in latency. The architecture provides inherent scalability in future. The total time required to transmit compressed data is less than that of transmitting

uncompressed data. This can lead to a performance benefit, as the bandwidth of a link appears greater. There is a potential of doubling the performance of storage / communication system by increasing the available transmission bandwidth and data capacity with minimum investment. It can be applied in Computer systems and high performance storage devices. As future work, improving compression for the disk data set by increasing dictionary length and introducing run length coding techniques to the algorithm to improve compression ratio is considered.

References

1. S. Henriques and N. Ranganathan, "High Speed VLSI Design for Lempel-Ziv Based Data Compression," IEEE Trans. Circuits and Systems, vol. 40, no. 2, pp. 90-106, Feb. 1993.
2. Jung and W.P. Burleson, "A VLSI Systolic Array Architecture for Lempel-Ziv Based Data Compression," Proc. IEEE Int'l Symp. Circuits and Systems, pp. 65-68, June 1994.
3. Jung and W.P. Burleson, "Real Time VLSI Compression for High Speed Wireless Local Networks," Proc. Data Compression Conf., Mar. 1995.
4. J.A. Storer and J.H. Rief, "A Parallel Architecture for High Speed Data Compression," J. Parallel and Distributed Computing, vol. 13, pp. 222-227, 1991.
5. C.Y. Lee and R.Y. Yang, "High-Throughput Data Compressor Designs Using Content Addressable Memory," IEE Proc. Conf. Circuits Devices Systems, vol. 142, pp. 69-73, Feb. 1995.