# ASIC IMPLEMENTATION OF A DATA COMPRESSOR BASED ON HUFFMAN CODING IN 130nm TECHNOLOGY

EDWIN GILBERTO CARREÑO LOZANO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
BUCARAMANGA
2016

# ASIC IMPLEMENTATION OF A DATA COMPRESSOR BASED ON HUFFMAN CODING IN 130nm TECHNOLOGY

Autor:
EDWIN GILBERTO CARREÑO LOZANO

Trabajo de Grado para optar al título de
Ingeniero Electrónico

Director:
Ph.D. Elkim Felipe Roa Fuentes
Codirector:
M.SC Héctor Iván Gómez Ortiz

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICOMECÁNICAS
ESCUELA DE INGENIERIAS ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
BUCARAMANGA
2016

CONTENT

Pag.

LIST OF FIGURES

P*ag.*

LIST OF TABLES

Pag.

## RESUMEN

TITULO: Implementación de un CIAE (Circuito Integrado de Aplicación Específica) basado en la codificación de Huffman para la compresión de datos en tecnología de 130nm[1].

AUTOR: Edwin Gilberto Carreño Lozano[2].

PALABRAS CLAVE: Circuitos Digitales, Codificación de Huffman, Compresión de Datos, Síntesis de Circuitos Integrados.

ALICE (A Large Ion Collider Experiment) es un proyecto financiado por el CERN (Conseil Européen pour la Recherche Nucléaire) cuyo objetivo es estudiar la interacción de la materia bajo densidades de energía extrema, permitiendo recrear las condiciones bajo las cuales fue dado el origen del universo. La realización del experimento consiste en la colisión de iones pesados de plomo a gran velocidad; así, en el momento exacto del choque es liberada una cantidad extrema de energía que hace que aparezcan nuevas partículas. Finalmente, la información relacionada con las nuevas partículas es colectada por un conjunto de sensores y sistemas electrónicos para un posterior análisis.

Actualmente el experimento tiene como objetivo aumentar la cantidad de datos capturados para tener más información acerca de los eventos generados tras una colisión. Los datos capturados viajan a través de líneas de transmisión hasta las centrales de información donde estos son almacenados. Sin embargo, aumentar la cantidad de datos trae consigo un aumento en la probabilidad de perder información debido al limitado ancho de banda del canal de comunicaciones.

A través de un esquema de compresión de datos, basado en la codificación de Huffman, se ha desarrollado en este documento un módulo capaz de conservar la información colectada con una menor cantidad de datos para su representación. Este compresor de datos fue diseñado en tecnología CMOS (Complementary Metal Oxide Semiconductor) de 130nm pensando en ser compatible con los circuitos integrados utilizados en el experimento. Además, en este trabajo se


[1] Trabajo de Grado modalidad en investigación.

[2] Facultad de Ingenierias Fisicomecánicas Escuela de Ingenierias Eléctrica, Electrónica y Telecomunicaciones. Director: Ph.D. Elkim Felipe Roa. Codirector: M.Sc Hector Ivan Gomez Ortiz

presenta una segunda versión del compresor, pensando en un eficiente uso de potencia y área.

## ABSTRACT

TITLE: ASIC Implementation of a Data Compressor Based on Huffman Coding in 130nm Technology[3].

AUTHOR: Edwin Gilberto Carreño Lozano[4].

KEYWORDS: Data compression, Digital circuits, Integrated circuit synthesis, Huffman encoding.

ALICE (A Large Ion Collider Experiment) is a project funded by CERN (Conseil Européen pour la Recherche Nucléaire) whose objective is to study the interaction of matter at extreme energy densities, allowing recreate the conditions under which the origin of the universe was given. Conducting the experiment consists in the collision of heavy ions of lead at high speed; so, at the exact time of the collision is liberated an extreme amount of energy that makes new particles appear. Finally, the information related to the new particles is collected by an array of sensors and electronic systems for further analysis.

Currently the experiment aims to increase the amount of data captured for more information about the events generated after a collision. The captured data travel through transmission lines to the central information where they are stored. However, increasing the amount of data entails an increase in the probability of losing information due to limited bandwidth communications channel.

Through a data compression scheme based on Huffman coding it has been developed in this document a module capable of retaining the information collected with less data for representation. This data compressor was designed in CMOS (Complementary Metal Oxide Semiconductor) technology 130nm thinking of being compatible with the integrated circuits used in the experiment. In addition, this paper a second version of the compressor, considering efficient use of power and area is presented.

[3] Degree Project.

[4] Faculty of Physics Mechanics Engineering. Electrical, Electronics Engineering and Telecommunications School. Director: Ph. D. Elkim Felipe Roa. Codirector: M. Sc Héctor Ivan Gómez Ortiz.

# INTRODUCTION

In the installations of ALICE (A Large Ion Collider Experiment)[5] at CERN (Conseil Européen pour la Recherche Nucléaire) there are electronic systems that can detect and capture information relating to the collision of heavy nuclei (lead-lead collisions). The information captured in the experiment allows the scientists to study the conditions under which the origin of the universe was given. With a larger amount of data captured in real time, more information related with the events generated in the collision is collected. In this way, the quantity and quality of information are critical to the success of the experiment.

Taking into account the above, the acquisition of more information is necessary to improve the experiment. Consequently, the data readout rate increases in the chamber where the collisions occur. However, an important consequence of increasing the amount of data is the possible loss of information due to limited bandwidth in the communication channels. Currently, the Integrated Circuits (ICs) involved in the experiment do not have a data compression scheme that allows mitigating the impact of handling a larger amount of data in a transmission line.

For prevent data loss, this paper proposes a data compression scheme based on Huffman coding. The Huffman coding is a lossless method for compress data and can be easily implemented in software or hardware, with the advantage of using an optimal (shortest expected length) prefix code for a given data distribution [1], [2]. With this in mind, the data compression strategy is designed in CMOS 130nm technology thinking about the possibility of being implemented in the integrated circuits involved in the experiment. Finally, this paper shows a second version of the compression scheme, seeking to optimize power and area.

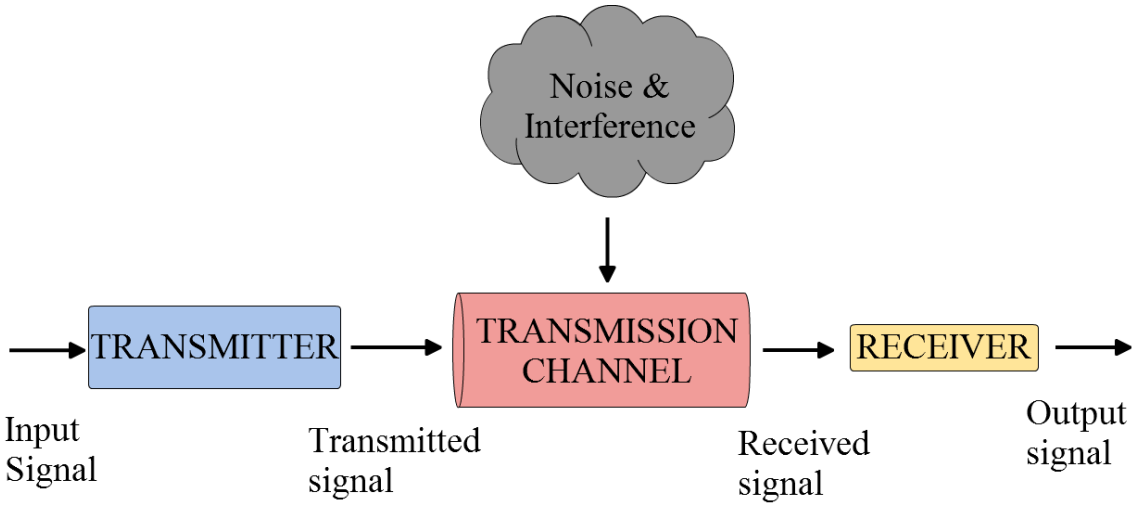
[5] For more information about ALICE project, visit their web page: http://alice-collaboration.web.cern.ch/

## 1. PROBLEM STATEMENT AND PURPOSE OF DATA COMPRESSOR.

Figure 1. Basic model of a communication system. shows a communication system composed of a transmitter, a channel and a receiver. Also, a communication system model must include a signal generator that represents interference effects and random noise affecting the channel. Therefore, the transmission of information depends on the efficiency on each one of the components which are part of the system, including good rejection of unwanted signals by the channel.

Usually in a communication system, it is necessary to increase the amount of information to be transmitted in the shortest possible time in order to avoid losing important content in real time. Thus, the increment of data for being transmitted implies enhancing the frequency of transmission. The frequency enhancement in the case of digital signals is carried out decreasing the duration of the rectangular pulse that represents a bit (Binary Digit).

*Figure 1. Basic model of a communication system.*




The immediate consequence of reducing the duration of one bit can be seen by reviewing the Fourier transform of a rectangular pulse. In Figure 2 decreasing the pulse width disperses the energy concentration in a higher bandwidth compared with the original signal in Figure 3. Indeed, a higher bandwidth increases the

possibility of losing information due to the fact that the channel bandwidth could ignore part of the new frequency content of the signal.

*Figure 2. Narrow rectangular pulse and Fourier transform; the bandwidth of signal f(t) increases when time of bit decreases.*

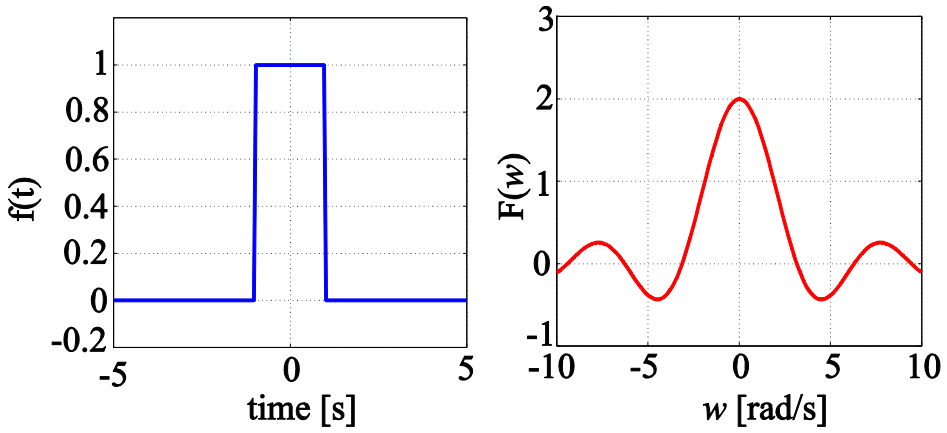
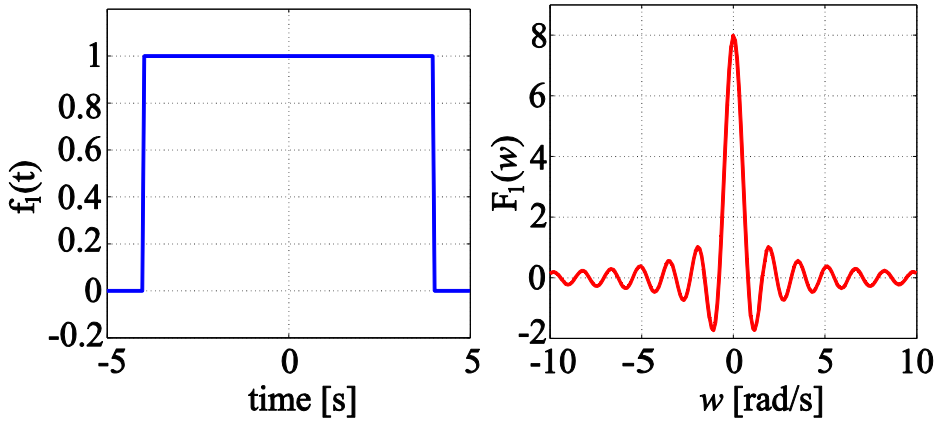



*Figure 3. Rectangular pulse and spectral distribution of energy.*




A first solution of the problem, which implies increasing the data rate, could be to improve the transmission channel in order to increase the bandwidth limit. However, from an economic point of view, it is not very profitable to replace a transmission line each time that increasing the data readout is required. Because of the cost involved, it is necessary to seek other strategies to increase data rate by avoiding wasting existing resources or the installation of new and more expensive ones.

Considering the fundamentals of information theory, a much more appropriate solution is to reduce the redundancy present in the data. Avoiding redundancy allows the transmission line to transport the same content with less data, a process known as encoding from source or data compression. Data compression also has another advantage: the amount of memory required to store the information decreases due to the data that represents the information coming in a smaller set than the original one (information plus redundancy).

*Figure 4. Two uncompressed strings of 10-bits each one, in this case are needed 10-bits to represent the value '0'.*

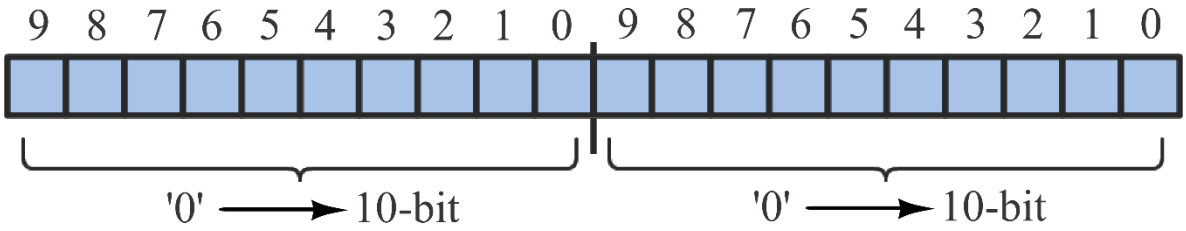
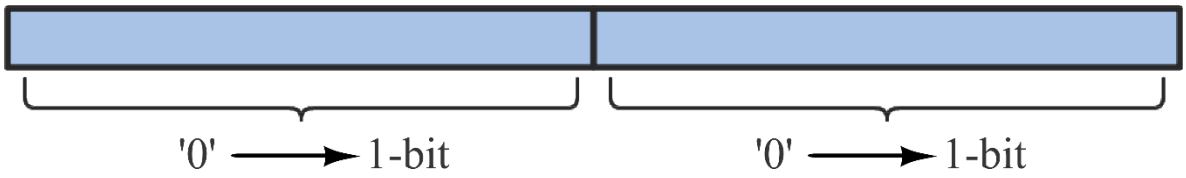



*Figure 5. Strings after compression, the value '0' is represented by 1-bit.*

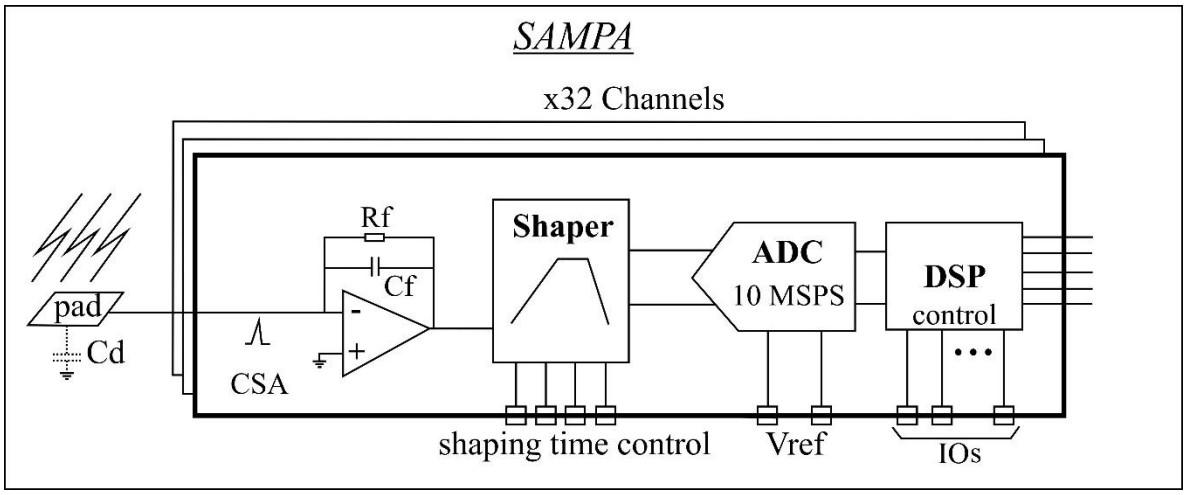



The Figure 4 and Figure 5 shows an example of two bits streams that convey the same information; the content is the value '0'. The string shown in the Figure 4 requires 10-bits to represent the information, while the string in the Figure 5 requires only 1-bit. This example shows that data compression reduces the amount of elements to represent the information without losing content. Note that the two data strings in both figures have the same baud rate, because the symbol '0' is transmitted in the same time. According to the Figure 4 and Figure 5 the energy of the signal is distributed differently in the frequency domain. Finally, it is shown that data compression can increase or maintain the same transmission rate thereby reducing the possibility of losing information by the bandwidth limit imposed by the channel.

Given the benefits of data compression and applying an Application Specific Integrated Circuits (ASIC) design methodology, a circuit for data compression compatible with the SAMPA chip in [3] was designed. In Figure 6 is shown the SAMPA chip (created by the Universidade de Sao Paulo), it has been designed for the acquisition, filtering and digital conversion of signals that contains information about events in particle collisions in the ALICE project; the front-end of SAMPA

chip was created by Hugo Hernandez in [4]. Through the SAMPA chip, it is possible to increase the sampling rate, improving resolution of the experiment. As a consequence, the amount of data sent in a transmission line increases, raising the possibility of losing information.

*Figure 6. SAMPA system block diagram.*




*Technical design report for the upgrade of the ALICE read-out & trigger system [5], [4].*

## 2. HUFFMAN COMPRESSOR ARCHITECTURE.

The data compressor designed in [3] and the proposed improvement made in this document use the Huffman coding as a method of lossless compression. Both works have been designed in a 130nm CMOS technology.

### 2.1. LOSS-LESS HUFFMAN COMPRESSION ALGORITHM

The Huffman coding is a prefix code that is used for lossless data compression. This algorithm is optimal because the length of the compressed data ($L$) is not greater than 1-bit of the lower bound imposed by the entropy ($H(x)$) [1], as follows:

$$H\ x\ \leq L\ \leq H\ x\ +1 \quad (1)$$

Currently, the Huffman algorithm is widely used to compress data. The algorithm compresses the data by assigning shorter code-words to the most frequent data, whereas the remaining data have longer code-words [2], [6]. Due to the variable length of the code-words generated, this algorithm has
high computational requirements. The challenge of this problem is to design and to implement a solution that combines the compression performance using the Huffman algorithm and an implementation into a small and suitable footprint layout in the multi-channel SAMPA chip. In this study we present an approach to solving this two issues.

### 2.2. STATISTICAL DATA ANALYSIS

The construction of a compression algorithm implies the understanding of the data and its relationship with the precedent hardware; i.e. number of bits at the input, data transmission rate, DC coupling and other characteristics. In Figure 7 there are 110 milliseconds of real particle collision data, 1099 packets with around 1000 collision-samples each one. These samples were taken at a collision rate of 10 kHz and 10 MHz of sampling frequency. Considering the upgrade of the experiment, the future packets will be slightly different according to the increase of the collision rate to 50 kHz. However, the sampling frequency will be the same. The data has been shaped by 5495 packets with around 200 samples each one.

*Figure 7. DSP raw data input.*

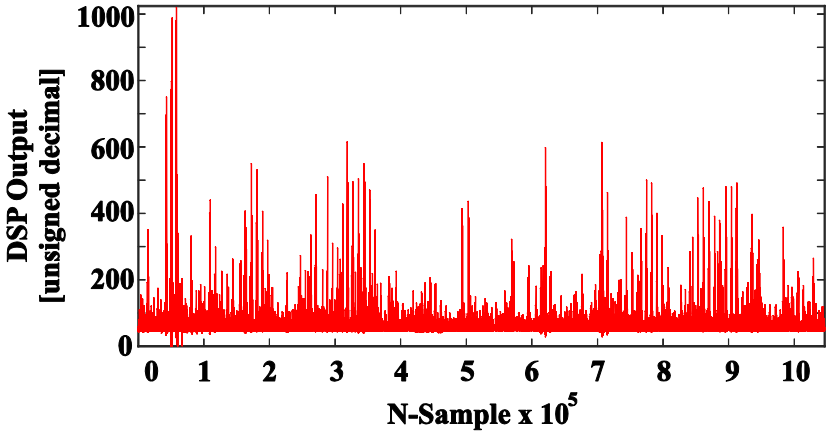



*Figure 8. Differential encoding for data input (previous incoming data is subtracted from the current one).*

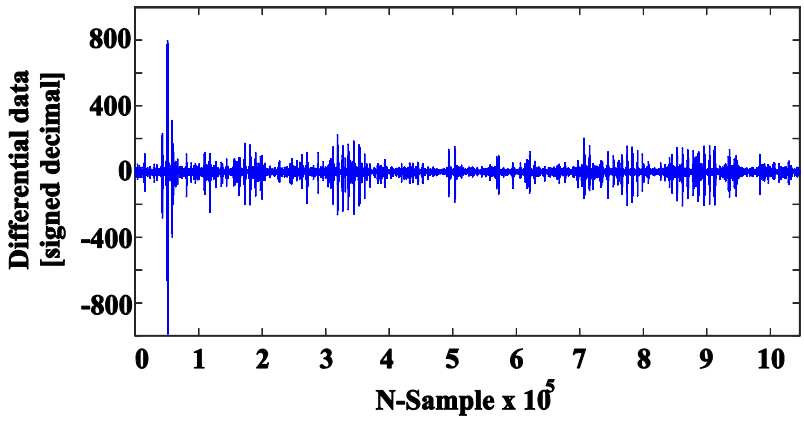
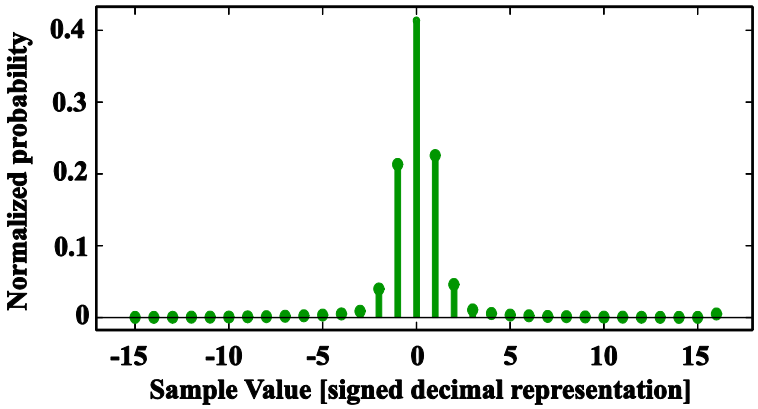


*Figure 9. Probability distribution of data collisions.*




Since data is slowly changing and baselines are different for each channel and chip, we applied differential encoding. First, previous incoming data is subtracted from the current-one. With differential encoding, we were able to get a data stream that is independent from the baseline and therefore we enabled the reuse of the Huffman encoder for more channels. Figure 8 shows the result for DC balancing.

To apply Huffman coding, a probabilistic study of raw data is necessary. The probabilistic study has been done with the data variation obtained in the differential encoding output. The results are shown in the Figure 9, note that approximately 99% of the information is within the -15 to +15 range. Therefore, the Huffman coding strategy only encodes the values within the mentioned range and when the difference between the two samples is outside, the compressor assigns a special header, other than Huffman coding, in front of the raw data. This header notifies the receiver that the following 11-bits will be raw data.

Using the results of the probabilistic study, we created a Huffman tree based on one fact: increasing the length of the longest code increases the compression ratio. A Huffman tree coded in MATLAB gives a compression factor of 4.15, but the longest code has 13-bits. There is a tradeoff between the best compression ratio and the maximum length of the longest code.

In this case, using a code with 13-bits implies more hardware, power consumption and area. So, the best choice is to reduce the longest code, and then to reduce the compression ratio as well. The final proposed solution has a Huffman tree with a compression ratio of 3.89 and the longest code becomes 9-bits long. Table 1 shows a portion of the Huffman dictionary. Before sending the data stream, the Huffman code is compacted into a buffer memory of 10-bits wide.

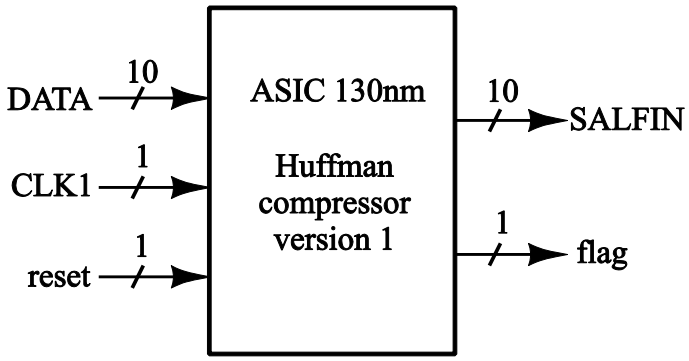
*Table 1. Huffman compressor dictionary*

| Data | Probability | Huffman Code | Code-length (bits) |
|---|---|---|---|
| 0 | 0.41352 | 0 | 1 |
| 1 | 0.22571 | 100 | 3 |
| -1 | 0.21323 | 101 | 3 |
| 2 | 0.04606 | 1100 | 4 |
| -2 | 0.03987 | 1101 | 4 |
| Out of range | 0.005099 | 111000 | 6 |

## 2.3. HUFFMAN COMPRESSOR, VERSION 1

The design methodology used in both works is top-down; it allows the designer to focus on the behavior of the overall circuit and then build it from smaller components or modules. Finally, the overall circuit is represented as a white box containing inputs and outputs as shown in the Figure 10. Each module within the overall circuit has a specific function that allows the design to be more flexible with a higher level of abstraction. The flexibility of this methodology is evident when it is required to improve some specific module without affecting others.

*Figure 10. Module diagram of Huffman compressor, version 1.*




The data compressor designed in [3] is an ASIC with three inputs (DATA: 10-bit, CLK1 and reset has 1-bit width) and two outputs (SALFIN: 10-bits and flag: 1-bit) as shown Figure 10. The circuit uses only a clock signal with a frequency of 100 MHz to handle all existing sequential logic. Finally, to synchronize the DSP output with the proposed data compressor a signal of 10 MHz is needed. This signal is created from a frequency divider and 100 MHz clock signal (CLK1 input).

The circuit behavior is divided into three main events: the first of them is responsible for taking data from the DSP and making a subtraction between the current data and the immediately preceding. The reason for this subtraction is to remove the dependency of any Direct Current (DC) signal. The second step consists in encoding the result of subtraction according to a dictionary based on the Huffman algorithm. Finally, the Figure 11 shows a processor composed of a Finite State Machine (FSM) and a Datapath. Both of them are responsible for packaging 10-bit frames with the contents of the compressed data. The Figure 12 shows the general scheme of the Huffman compressor and data traffic from its inputs to its outputs.

*Figure 11. Dedicated microprocessor scheme, finite state machine (blue) + datapath (black) [7].*

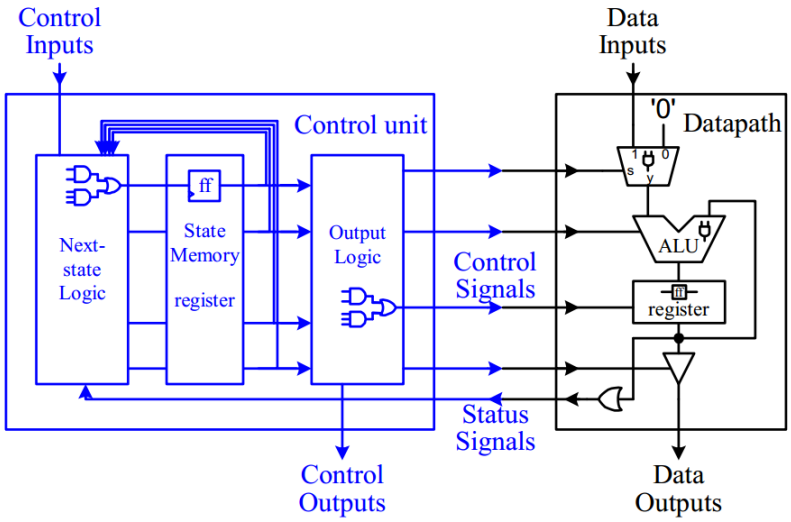
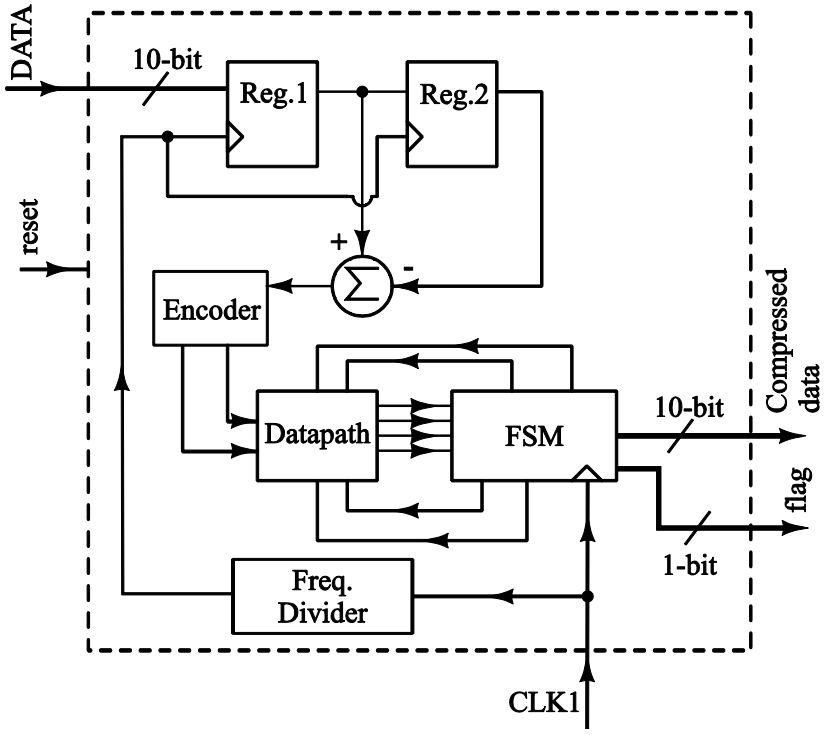


*Figure 12. General scheme of the Huffman compressor implemented on chip, version 1.*




2.4. HUFFMAN COMPRESSOR, VERSION 2

The objective of improving the circuit designed in [3] is to reduce power consumption and area, parameters which are related to good energy management of a low power electronic system. Power and area depend on the complexity of the operations performed by the compressor. A large number of operations increases the amount of hardware used for executing them. Therefore, a first step to improve the circuit involves an analysis of the power consumption on each one of the modules that make up the circuit.

*Figure 13. Power breakout of Huffman compressor version 1, subdivided in modules.*

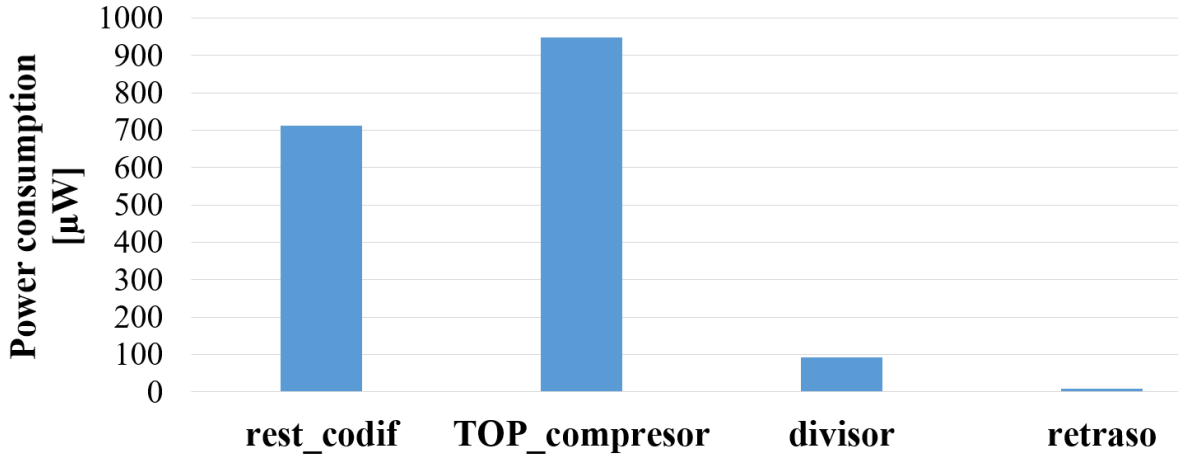



Figure 13 shows the power breakout of the Huffman compressor circuit in its first version. Modules that consume more power are the rest_codif and Top_compressor (FSM + Datapath), responsible for the differential coding and data packing respectively. Considering the above, optimization is focused on reducing the complexity of these processing blocks.

Data packing consists of taking encoded data and placing them in a buffer of 10-bit length and then transmitting data when the buffer is completely full. The 10-bit buffer gradually fills their positions according to the length of the encoded data, the case existing where the buffer is filled first before loading the data completely, i.e. when the code length is 17-bit. Given the above, 7-bit would be left out until the buffer empties its contents again and starts the process anew. Figure 14 shows the FSM in the first version of the compressor and it consists of 10 states for packaging and storage of data.

*Figure 14. Finite state machine for compressing data, version 1.*

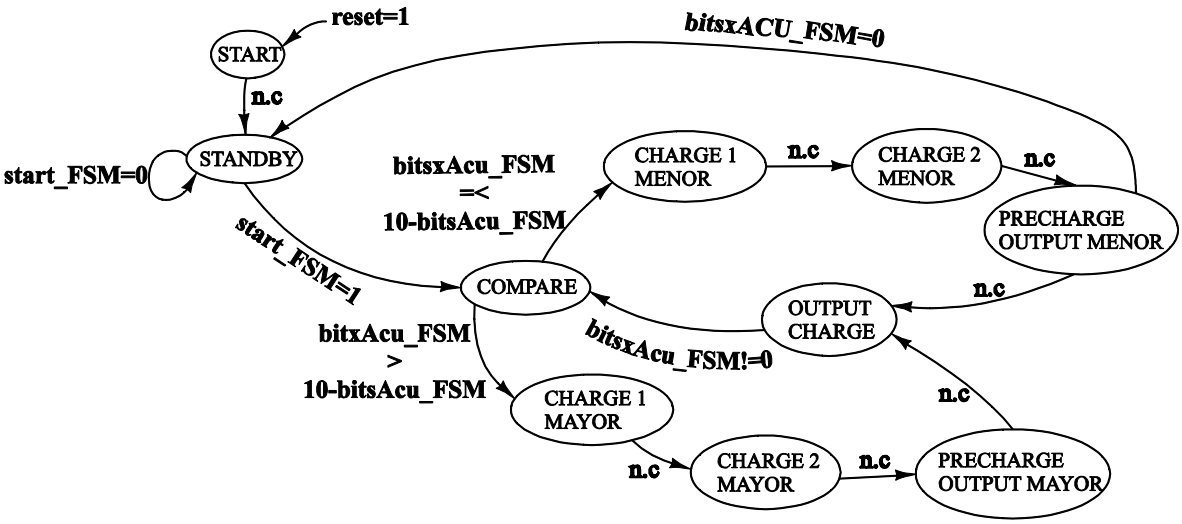
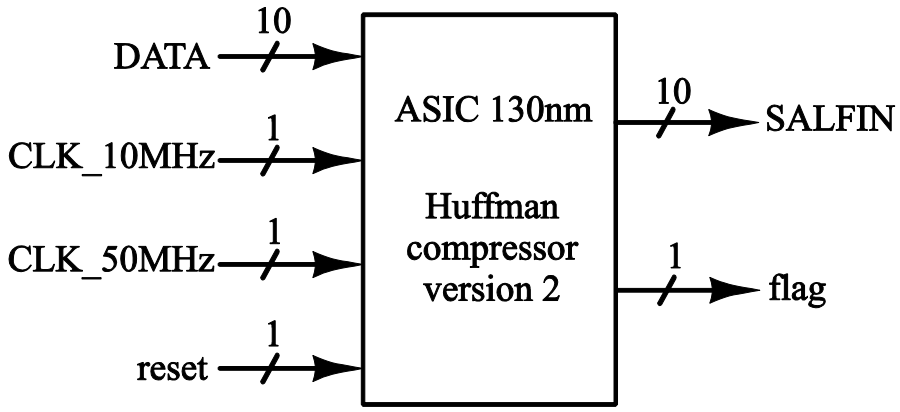



*Figure 15. Module diagram of Huffman compressor, version 2.*




With the existence of two clock signals with different frequency (100 MHz and 10 MHz) it is preferable to assign two clock inputs instead of one. So it is no longer necessary a divider block that generates the lower frequency signal (10 MHz) based on the 100 MHz clock signal. Figure 15 shows the composition of the main module with modifications. Although it is true this change is not relevant for power consumption, the design is more flexible with independent clock signals.

The analysis of the FSM and its optimization have led to a decrease to five states (see Figure 16), reducing the use of hardware for data control. Some states could be compacted, retaining the same purpose of the processor (package and

storage). The decreased number of states also helped to reduce the number of logical components that are part of the datapath, a fact which results in a decrease of power and area. The data related to area, power consumption and number of standard cells will be reported in the next section. Finally, the Figure 17 shows the reduced scheme for the datapath with the FSM in the second version of the Huffman compressor. A datapath is a set of registers and combinational elements governed by a FSM and its purpose is to control the data flow in each state. The "big datapath" is composed by Datapath and Shift register (see Figure 17). Both ("Big datapath" and FSM) are part of the process of storing and charging data before sending the compressed data to the output.

*Figure 16. Finite state machine for compressing data, version 2.*

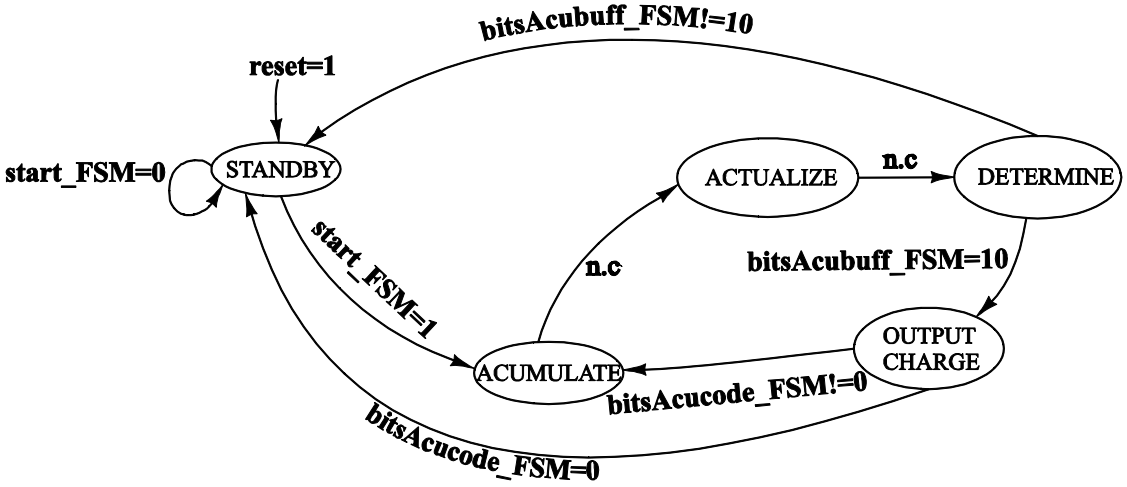
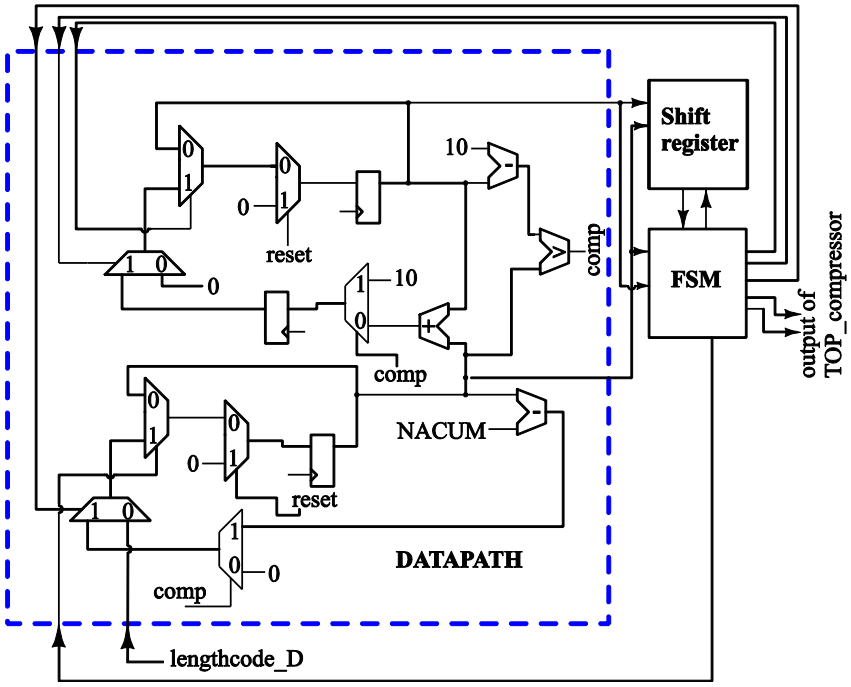


*Figure 17. Example scheme for datapath with the finite state machine, both are part of the Huffman compressor in its second version.*




Another important improvement is that by reducing system complexity due to the reduced number of states, a high speed clock signal for processing is no longer necessary. The new clock signal has a low frequency compared with the previous one in version 1, resulting in a saving of dynamic power consumption in the system. Finally, this improved version meets the same objective of the first version with the advantage of consuming less power and area.

# 3. RESULTS AND CHIP IMPLEMENTATIONS

## 3.1 STATE-OF-THE-ART

There are now many published materials about data compression, thinking about improving the use of system resources. Most of them are based on FPGA and only few propose the design of the compressor in an integrated circuit approach. In this section we made an overview of some important works reported in technical literature.

Patauner [8] proposes a data compressor for S-ALTRO chip (System on Chip installed in ALICE experiment). This development is based on vector quantization and Huffman coding; the data compressor was designed to be implemented in an ASIC, but there are no results of power consumption or occupied area in the integrated circuit. Continuing, another project for compressing data on communications is proposed by Rigler [9], where the implementation was made through FPGA and the application of compression algorithms like LZ77 and Huffman coding.

Chen [10] proposes a solution for low power consumption in a Wireless Body Sensor Network, Chen includes an adaptive algorithm for saving power and an encoder based on Huffman coding. Chen demonstrate a saving of power consumption around of 99.164% using encoding, the reason is that data frame sent is shorter than original one and this fact represents less use of system resources. MCU and the encoder have a power consumption of 150 $\mu W$, the core area is around of 134K $\mu m^2$ synthesized using 180nm CMOS process and the frequency of operation is 100 MHz.

Another project of Chen [11] is a MCU (Micro Control Unit) with a reconfigurable filter and it is implemented in 180 nm CMOS process with an area of 76K $\mu m^2$ and power consumption of 5.8 mW. Power consumption increases because MCU have more functions that previous project, i.e. UART interface.

Taking into account previous projects intended for healthcare applications, Chen [12] proposes an encoder for ECG signals. Encoding data decreases power consumption and allows to transmission line to drive more data with limited bandwidth; the VLSI implementation intended for compress ECG signals has a power consumption of 36.4 $\mu W$, core area of 46K $\mu m^2$, frequency of operation is 100MHz and was implemented in 180nm technology.

An upgrade of the ECG encoder [12] is proposed in [13] which includes Huffman coding and Golomb-Rice algorithm for compressing ECG signals. The power consumption and area decrease due to adaptive fuzzy predictors. The power consumption is 30 $\mu W$ and the area is 33.9K $\mu m^2$ in 180nm CMOS

technology. Finally, continuing with the ECG signals compression, Chen presents in [14] a data compressor in 90nm technology which increases the compression factor to 2.84 comparing to previous works: 1.92 [10] **<** 2.59 [11] **<** 2.43 [12] **<** 2.56 [13]. The power consumption increases to 201 $\mu W$ with operation frequency of 201MHz and a reduced area of around 6000 $\mu m^2$.

The published documents presented here are the most approximate to the proposed project, but FPGA based technology still remains dominant in research publications. There are few papers published in the field of ASIC compression that are actually taken to tape-out (physical implementation of the device designed, all this made by an integrated circuit manufacturer). Therefore, the work realized in [13] and its improve in this document aims to perform a design process oriented to the physical implementation.

## 3.2 RESULTS SUMMARY

Both architectures were synthesized in 130nm CMOS technology. The power consumption, area and number of gates used allow us to compare both designs, the integrated circuit in [3] and the improved version. Finally, a comparative with similar work will be reported in Table 2.

Table 2 reports of the main features of both compressors, using typical corner analysis at 25 °C and maximum frequency of operation. It is important to note a decrease of power consumption up to 50%; the number of standard cells also decreases to 37% compared with the first version. The strongest influence on the power decrease has to do with the reduction of the operating frequency and the number of states required for execution of circuit operations.

*Table 2. Summary report Huffman compressor (version 1 & 2)*

| **Parameters** | [1] | [2] | [3] | [4] | [5] | [6] | [7] |
|---|---|---|---|---|---|---|---|
| **Process[nm]** | 130 | 130 | 90 | 180 | 180 | 180 | 180 |
| **Area K [$\mu m^2$]** | 7.15 | 4.9 | 6 | 33.9 | 46 | 76 | 134 |
| **Power [mW]** | 1.97 | 0.99 | 0.2 | 0.03 | 36.4 | 5.8 | 0.15 |
| **Std. Cells (K)** | 0.71 | 0.41 | 1.9 | 2.71 | 3.57 | 3.77 | 13.4 |
| **Max. Freq.[MHz]** | 333 | 333 | - | - | - | - | - |
| **Clk. Rate[MHz]** | 160 | 50 | 200 | 100 | 100 | 100 | 100 |

The Table 3 and Table 4 show the area report, power consumption and number of standard cells for each of the modules that make up the first and second version of the compressor respectively. Both are operating at nominal clock rate. It is

important to note that the highest power consumption is related to the instance involving the state machine and the datapath, which are the components responsible for packaging and storing encoded data.

*Table 3. Report Huffman compressor version 1*

| **Instance name** | **Area [$\mu m^2$]** | **Power [mW]** | **Standard Cells** |
|---|---|---|---|
| **TOP_1** | **6594** | **0.724** | **609** |
| *Top_compresor* | 3576 | 0.385 | 382 |
| *recodif* | 2533 | 0.219 | 193 |
| *divisor* | 177 | 0.040 | 13 |
| *retraso* | 32 | 0.0034 | 2 |

*Table 4. Report Huffman compressor version 2*

| **Instance name** | **Area [$\mu m^2$]** | **Power [mW]** | **Standard Cells** |
|---|---|---|---|
| **TOP_1** | **4989** | **0.332** | **446** |
| TOP_compresor | 3529 | 0.299 | 329 |
| recodif | 1416 | 0.0205 | 112 |

Finally, Figure 18 shows a graphic report of resource usage. The improved version betters the first implementation in all specifications studied. The improved implementation allows obtaining a comprised version, optimizing the usage of resources. In Figure 19 is shown the layout implementation for the Huffman compressor in its version 1, it is evident to see that most space in the circuit is occupied by the FSM + Datapath, as it is shown in Table 3.

*Figure 18. Comparison between version 1 and its upgrade (version 2)*

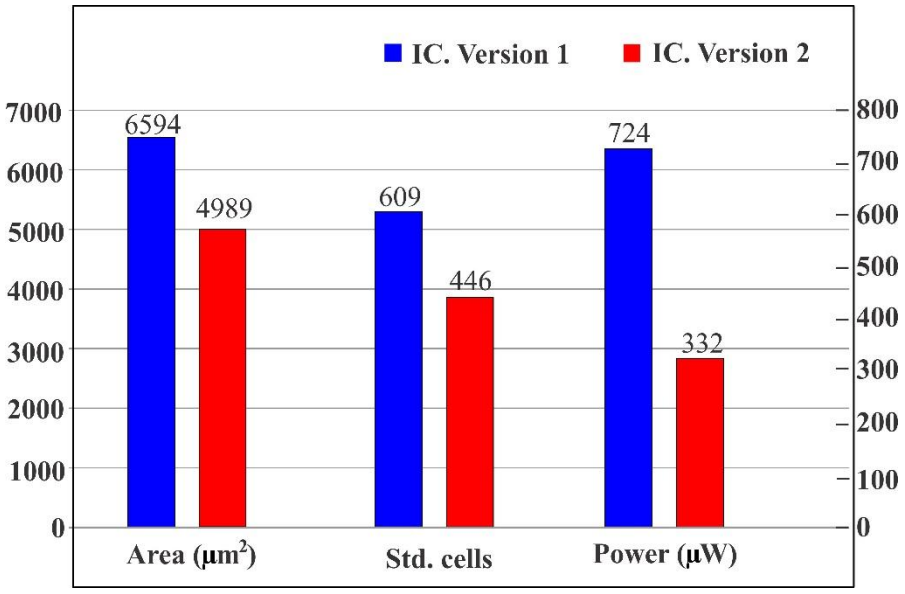
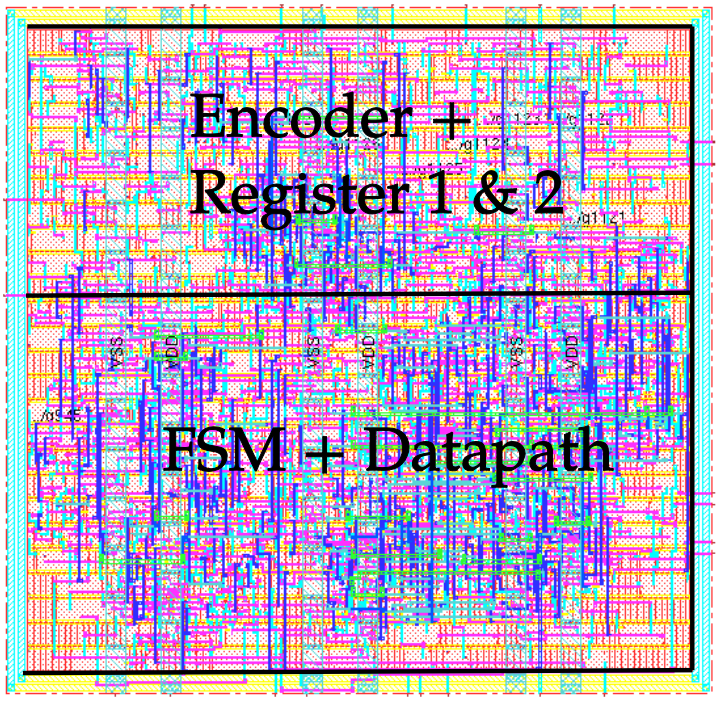



*Figure 19. Layout of Huffman compressor version 1 in 130nm CMOS technology*



# 4. CONCLUSIONS

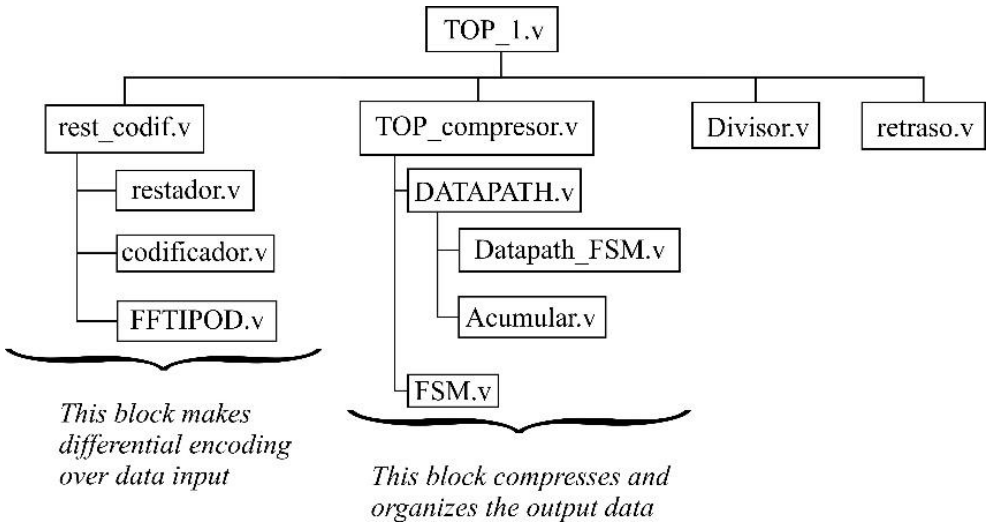
A Huffman encoding scheme with small footprint has been proposed. The resulting compression ratio enables the increase of data rate density about four fold in current electronics at the LHC (Large Hadron Collider) in the ALICE project. The analysis in power and area consumption show that an optimized finite state machine with reduced operations and less states results in saving power and area. Furthermore, the tradeoff between code-length and compression ratio should be considered in order to achieve the required compression ratio with low resources implementation. Reducing the longest code length allows saving area but this reduction decreases the compression ratio. The synthesized and layout results of the proposed scheme show that it is possible to implement a compression scheme inside the readout chips of the ALICE project at CERN with low cost and without data lost. Consequently, full-data is later processed in data centers.

# APPENDIX

## HIERARCHICAL SCHEMES FOR HUFFMAN COMPRESSOR (VERSION 1 AND VERSION 2).

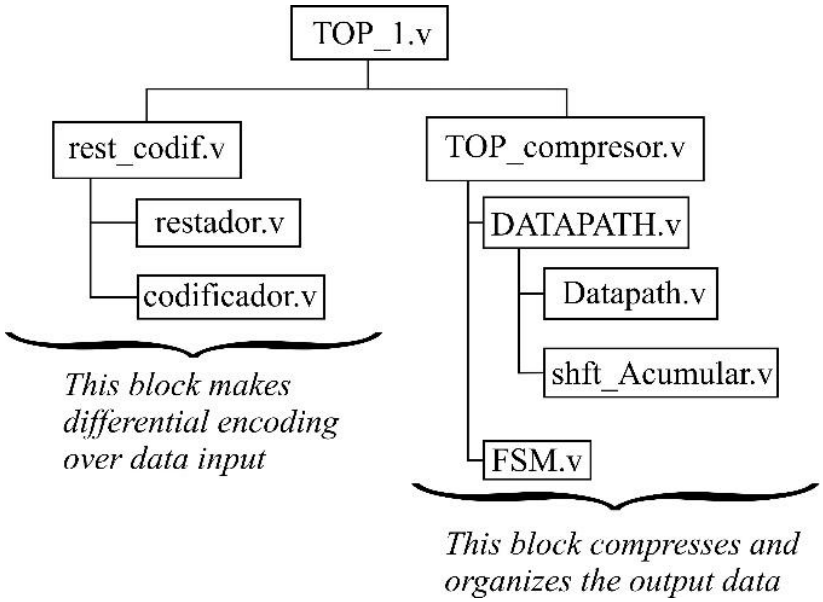
This section presents the hierarchical schemes using real names of the Verilog files that make up the overall circuit for compressing data. In Figure 20 is shown the hierarchical scheme for the Huffman compressor in its first version. Note that, the first version is denser than second version shown in Figure 21.

*Figure 20. Top-down digital design methodology for Huffman compressor version 1*




For improving a module, it is necessary modify only the Verilog file associated. This kind of methodology makes the design more flexible and organized. In both versions of the Huffman compressor, the files TOP_1.v have the instances for differential encoding and the storage/package data before sending to the output.

*Figure 21. Top-down digital design for Huffman compressor version 2.*



# BIBLIOGRAPHIC REFERENCES


[1] T. M. Cover and J. A. Thomas, *Elements of information theory*. New Jersey, USA: John Wiley & Sons, 2012. pp. 112-113, 123-127.

[2] D. Huffman, “A Method for the Construction of Minimum-Redundancy Codes,” *Proceedings of the I.R.E*, vol. 40, sep 1952.

[3] E. G. Carreño *et al.*, “A 3.9 Compression-ratio Huffman Encoding Scheme for the Large Hadron Collider on 65nm and 130nm CMOS Technology,” in *2016 Latin American Symposium on Circuits & Systems (LASCAS)*, (Florianopolis), pp. 347–350, Feb 2016.

[4] H. Hernandez *et al.*, “Configurable Low Noise Readout Front-End for Gaseous Detectors in 130nm CMOS Technology,” in *IEEE International Symposium on Circuits and Systems (ISCAS)*, jul 2015.

[5] T. A. Collaboration, “Technical Design Report for the Upgrade of the ALICE Read-out & Trigger System,” tech. rep.

[6] D. Salomon, *Data Compression: The Complete Reference*. New York, USA: Springer-Verlag, 2007. pp. 74-95.

[7] H. E. O, *Microprocessor Design: Principles and Practices with VHDL*. Brooks/Cole, 2004. Chapter 9.

[8] A. C.Patauner, W.Pribyl, “A Lossless Data Compression Method for an Application in High Energy Physics,” in *Ph.D. Research in Microelectronics and Electronics*, (Cork), pp. 300 – 303, jul 2009.

[9] A. S.Rigler, W.Bishop, “FPGA-Based Lossless Data Compression Using Huffman and LZ77 Algorithms,” in *Canadian Conference on Electrical and Computer Engineering*, (Vancouver, BC), pp. 1235–1238, apr 2007.

[10] S. L. Chen *et al.*, “Wireless Body Sensor Network with Adaptive Low-Power Design for Biometrics and Healthcare Applications,” *IEEE SYSTEMS JOURNAL*, vol. 3, dec 2010.

[11] C. A. Chen *et al.*, “An Efficient Micro Control Unit with a Reconfigurable Filter Design for Wireless Body Sensor Networks(WBSNs),” *IEEE SYSTEMS JOURNAL*, vol. 3, dec 2012.

[12] S. L. Chen *et al.*, “VLSI Implementation of Low-power Cost Efficient

Lossless ECG Encoder Design for Wireless Healthcare Monitoring Application," *ELECTRONICS LETTERS*, vol. 49, jan 2013.

[13] S. L. Chen *et al.*, "Efficient Fuzzy-Controlled and Hybrid Entropy Coding Strategy Lossless ECG Encoder VLSI Design for Wireless Body Sensor Networks," *ELECTRONICS LETTERS*, vol. 49, aug 2013.

[14] S. L. Chen *et al.*, "VLSI Architecture of Lossless ECG Compression Design Based on Fuzzy Decision and Optimization Method for Wearable Devices," *ELECTRONICS LETTERS*, vol. 51, pp. 1409 – 1411, jul 2015.

# BIBLIOGRAPHY


CARREÑO, Edwin, et. al. A 3.9 Compression-Ratio Huffman Encoding Scheme for the Large Hadron Collider on 65nm and 130nm Technology. Florianopolis, Brazil. En: IEEE Latin American Symposium on Circuits & Systems (LASCAS)., 2016. pp. 347-350.

CHEN, Chiung-An, *et. al.* An Efficient Micro Control Unit with a Reconfigurable Filter Design for Wireless Body Sensor Networks (WBSNs). En: Sensors, 2012. pp 16211-16227.

CHEN, S. L. ; LUO, G. A. y LIN, T. L. Efficient Fuzzy-Controlled and Hybrid Entropy Coding Strategy Lossless ECG Encoder VLSI Design for Wireless Body Sensor Networks. En Electronics Letters. 2013. pp.1058 - 1060.

CHEN, Shih-Lun, *et. al* . VLSI Architecture of Lossless ECG Compression Design Based on Fuzzy Decision and Optimization Method for Wearable Devices. En *Electronics Letters*. 2015. pp. 1409-1411.

CHEN, Shih-Lun , et. al. Wireless Body Sensor Network with Adaptive Low-Power Design for Biometrics and Healthcare Applications. En IEEE Systems Journal*.* 2010. pp.398 - 409.

CHEN, S. L. y WANG, J. G. VLSI Implementation of Low-power Cost Efficient Lossless ECG Encoder Design for Wireless Healthcare Monitoring Application. En: Electronics Letters. 2013. pp. 91-93.

COLLABORATION, the ALICE. Technical Design Report for the Upgrade of the ALICE Read-out & Trigger System. 2014.

COVER, Thomas M. Cover y THOMAS, J. A. Elements of Information Theory*.* 2ed. New Jersey, USA. Editorial John Wiley & Sons.2012

HERNANDEZ, Hugo; VAN NOIJE, Wilhelmus y MUNHOZ, Marcelo. Configurable low noise readout front-end for gaseous detectors in 130nm CMOS technology. En: IEEE International Symposium on Circuits and Systems *(ISCAS). 2015 .*

HUFFMAN, David A. A method for the construction of Minimum-redundancy codes. USA. En: Proceedingd of the I.R.E., 1952. pp.1098 - 1101.

PATAUNER, Chistian; PRIBYL, Wolfgang y MARCHIORO, Alessandro. A Lossless Data Compression Method for an Application in High Energy Physics. Cork, Irlanda. En: Ph. D. Research in Microelectronics and Electronics.,2009 p. 300-303.

RIGLER, Suzanne; BISHOP, William y KENNINGS, Andrew. FPGA-Based Lossless Data Compression Using Huffman and LZ77 Algorithms. Vancouver, BC. En: Canadian Conference on Electrical and Computer Engineering. 2007. pp. 1235-1238.

SALOMON, David. Data Compression: The complete reference. 4ed. New York, USA. Editorial Springer-Verlag. 2007.

WANG, E. O. Microprocessor Design: Principles and Practices with VHDL. Brooks/Cole. 2004. Capitulo 9.