Lab in System on Chip Integrated Design

Final Project Report

4108064101 杜冠廷

1. Topic: KT Gray Image Compression Encoder

2. Abstract:

KT Gray Image Compression Architecture combines the specifications of both JPEG and JPEG2000 image compression. This method not only avoids some drawbacks of JPEG, such as mosaic distortion, but also retains several advantages of JPEG2000, such as real-time decoding. Compared to JPEG2000, it boasts lower computational complexity, and is also much more suitable for hardware implementation.

In this project, I utilized Zedboard for the hardware implementation of the encoder and then developed the decoder using MATLAB.

3. Architecture:

A、 System:

a. Hardware design, which includes DWT, quantization, difference calculation, and Huffman coding.

b. Communication protocol, which involves AXI4-Stream and AXI4-Lite.

c. Software design, which includes reading/writing data from an SD card.

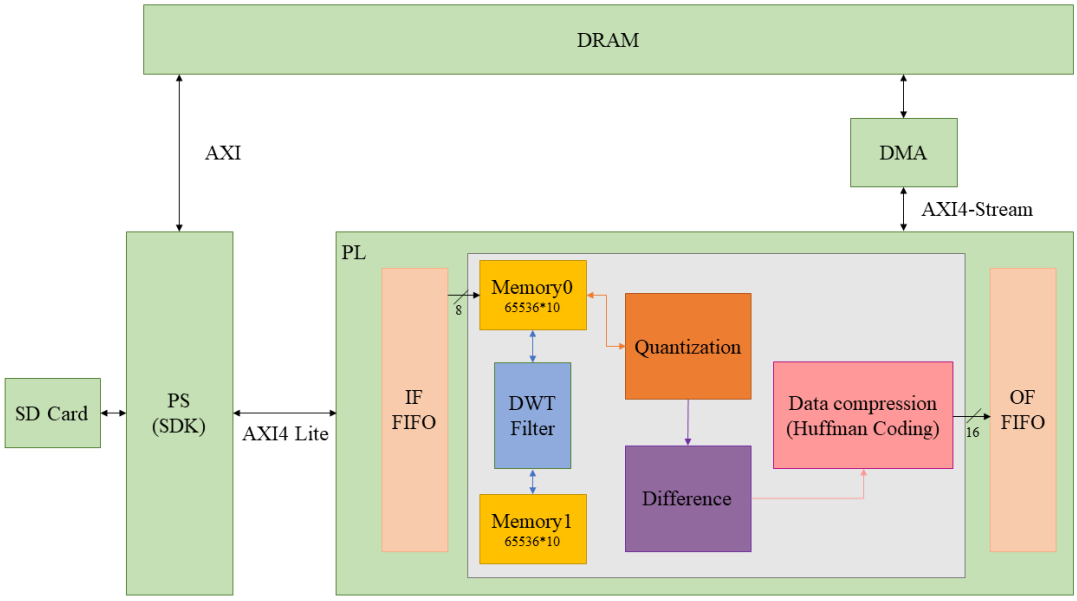


Figure 1 System Architecture

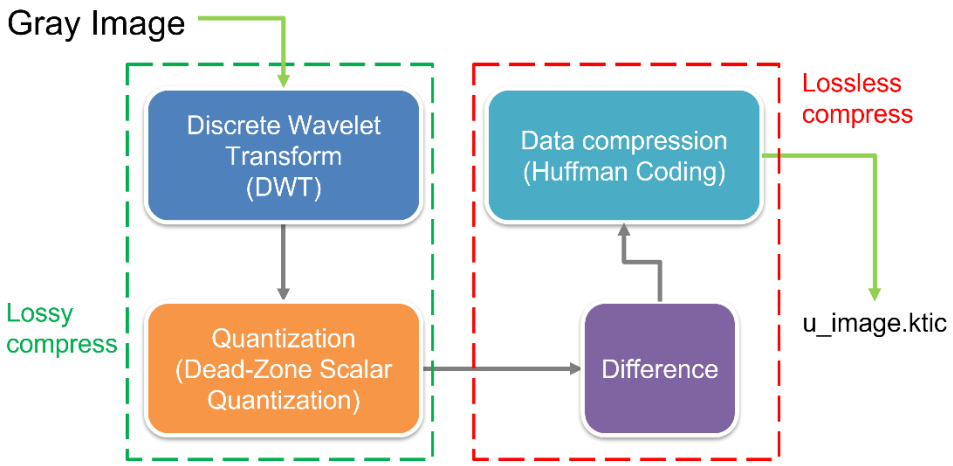


Figure 2 Encoder Architecture

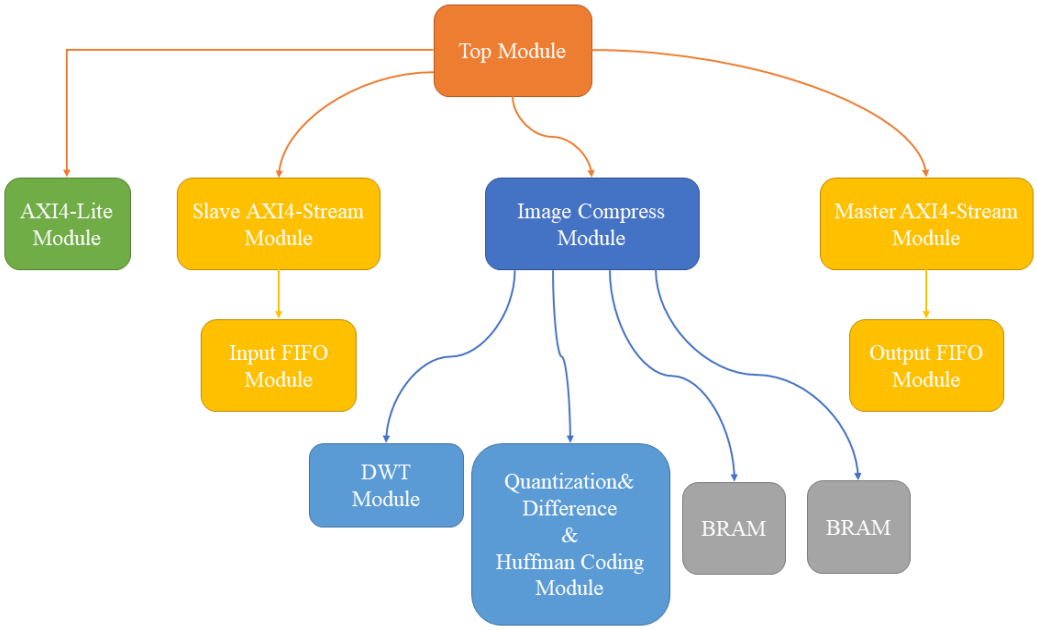


Figure 3 Design Module

B、 Hardware Design:

a. Layer 1: DWT layer(Discrete Time Wavelet Transform)

b. Layer 2: Quantization layer(Dead-Zone Scalar Quantization)

In this layer, we will quantize the data by discarding some bits to achieve data compression.

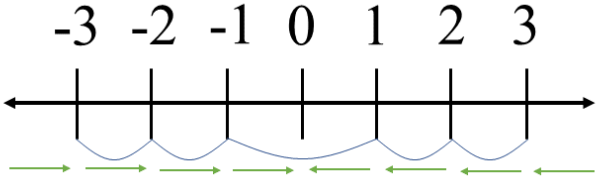


Figure 4 Dead-Zone Scalar Quantization

c. Layer 3: Difference layer

In this layer, we first divide the data processed in the previous steps into four regions: LL, HH, LH, and HL. Then, for each region, we subtract each column of data from its left column and record the difference. This step aims to concentrate the frequency of data occurrence.

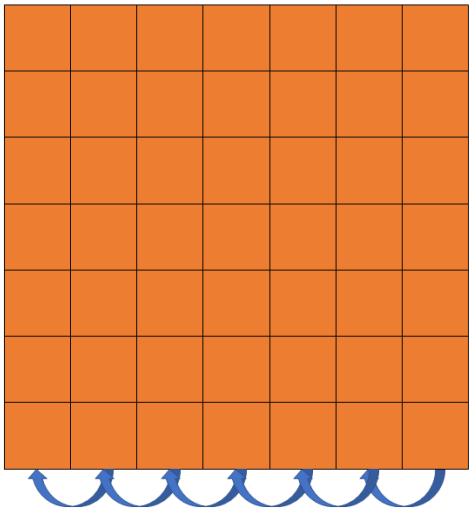


Figure 5 Difference

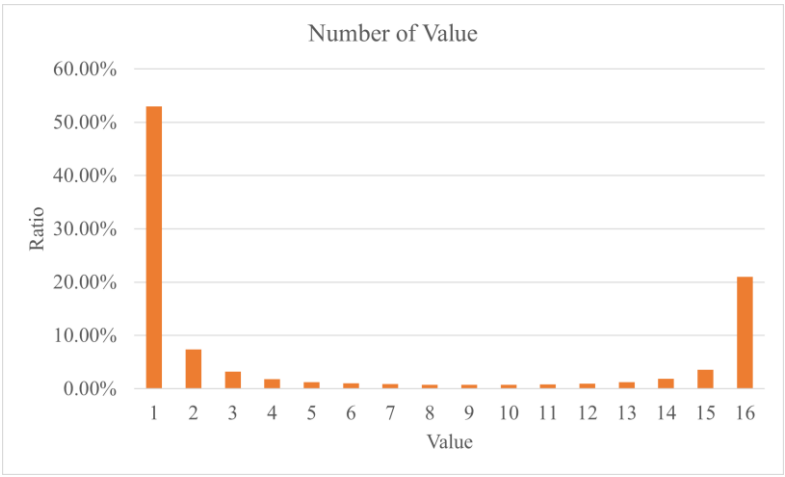


Figure 6 Frequency of Data Occurrence

d. Layer 4: Data compression layer

In this layer, we first divide the data of each pixel into groups of 4 bits and apply Huffman coding to each group individually. As a result, the number of data after this layer will be M times the original. However, due to Huffman coding, the total number of output bits will be fewer than the input.

The pre-analysis reveals that the frequencies of data occurrences are mostly the same, so we predefine the format of Huffman coding to

reduce complexity and speed up the computational speed of the architecture.

Table 1 Huffman Table

Value	Huffman code	Value	Huffman code
0000	0	1000	111111111111111
0001	110	1001	1111111111111110
0010	11110	1010	111111111110
0011	1111110	1011	11111111110
0100	11111110	1100	111111110
0101	1111111110	1101	111110
0110	111111111110	1110	1110
0111	11111111111110	1111	10

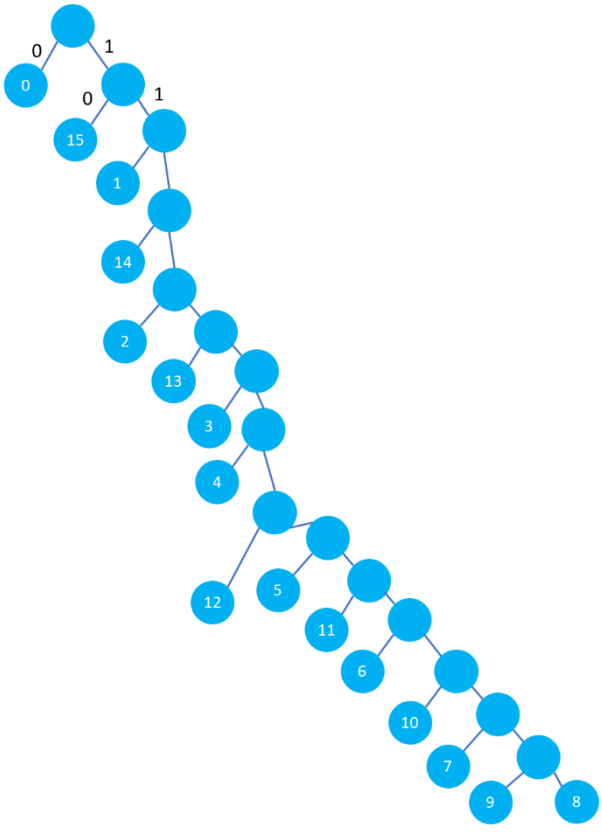


Figure 7 Huffman Tree

4. Specification:

A、 Input image size: 256*256*8 (pixel*pixel* pixel width)

B、 Output data size: 256*512*N (N is unknown)

Hint: Huffman coding is a variable-length data compression technique.

5. Result:

A、 Demo:

a. Execution time: 13453 μs/per operations.

```
AXI DMA Initial

--- axi dma initial ---

PL Start
PL Start success
READ_DATA0 Initial
SD Card Reading...
--- read data done ---
SD Card Reading Done!
image_compress_IP0 Initial

Start image_compress_IP0...
AXI DMA Transfer success 0

--- TX ifmap done ---
AXI DMA Transfer success 0

--- RX ofmap done ---
image_compress_IP0 done!
Time used: 12145 us
image_compress_IP0 Ending
Output datas are correct !!!
```

Figure 8 SDK Terminal

b. The average compression ratio is approximately 31%.

c. Average PSNR (Peak Signal Noise Ratio) above than 42dB.

d. Compare:

Original image

Image after compression and decoder

Figure 9 Barara Gray Image

Original image

Image after compression and decoder

Figure 10 Goldhill Gray Image

Original image

Image after compression and decoder

Figure 11 Lena Gray Image

B、 Implementation Result

a. Block Design

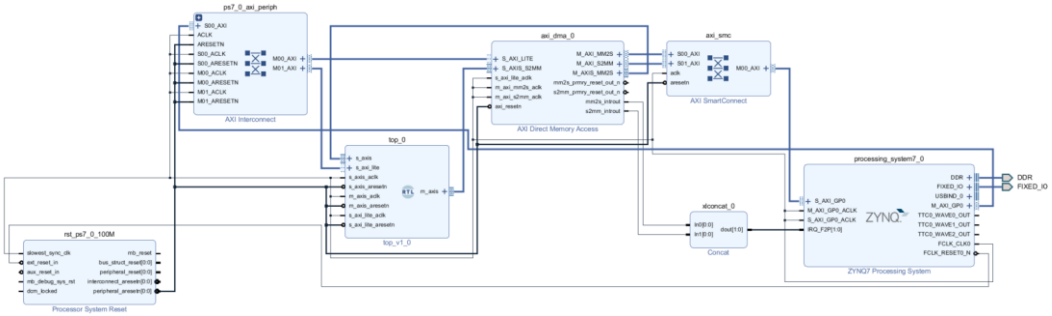


Figure 12 Block Design Diagram

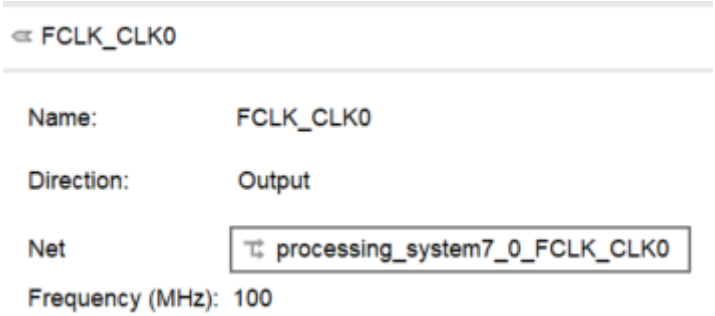


Figure 13 Clock Frequency

b. Synthesis and Implementation Report:

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	1.114 ns	Worst Hold Slack (WHS):	0.016 ns	Worst Pulse Width Slack (WPWS):	3.750 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	26108	Total Number of Endpoints:	26108	Total Number of Endpoints:	8718

All user specified timing constraints are met.

Figure 14 Timing Summary

Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**1.748 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**45.2°C**
Thermal Margin:	39.8°C (3.3 W)
Effective ϑJA:	11.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

Launch Power Constraint Advisor to find and fix invalid switching activity

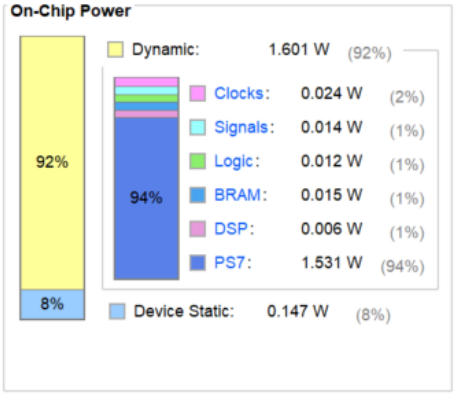
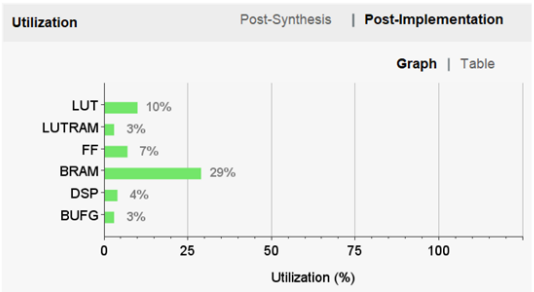


Figure 15 Power Summary



Utilization Post-Synthesis | **Post-Implementation**

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	5459	53200	10.26
LUTRAM	538	17400	3.09
FF	7656	106400	7.20
BRAM	41	140	29.29
DSP	9	220	4.09
BUFG	1	32	3.13

Name	Slice LUTs (53200)	Slice Registers (106400)	F7 Muxes (26600)	F8 Muxes (13300)	Slice (13300)	LUT as Logic (53200)	LUT as Memory (17400)	Block RAM Tile (140)	DSPs (220)	Bonded IOB (200)	PHY_CONTROL (4)	BUFIO (16)
image_compress_wrapper	10.26%	7.20%	0.83%	0.83%	19.07%	9.25%	3.09%	7.20%	29.29%	4.09%	100.00%	3.13%
image_compress_i (image_com	10.26%	7.20%	0.83%	0.83%	19.07%	9.25%	3.09%	7.20%	29.29%	4.09%	0.00%	3.13%
axi_dma_0 (image_compres	2.61%	1.86%	0.00%	0.00%	4.46%	2.39%	0.69%	1.86%	3.57%	0.00%	0.00%	0.00%
axi_smc (image_compress_i	3.97%	2.63%	0.00%	0.00%	6.37%	3.30%	2.05%	2.63%	0.00%	0.00%	0.00%	0.00%
processing_system7_0 (ima	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	3.13%
ps7_0_axi_periph (image_co	1.02%	0.62%	0.00%	0.00%	1.81%	0.90%	0.35%	0.62%	0.00%	0.00%	0.00%	0.00%
rst_ps7_0_100M (image_co	0.03%	0.03%	0.00%	0.00%	0.08%	0.03%	<0.01%	0.03%	0.00%	0.00%	0.00%	0.00%
top_0 (image_compress_top	2.63%	2.06%	0.83%	0.83%	6.65%	2.63%	0.00%	2.06%	25.71%	4.09%	0.00%	0.00%
xlconcat_0 (image_compres	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Figure 16 Utilization

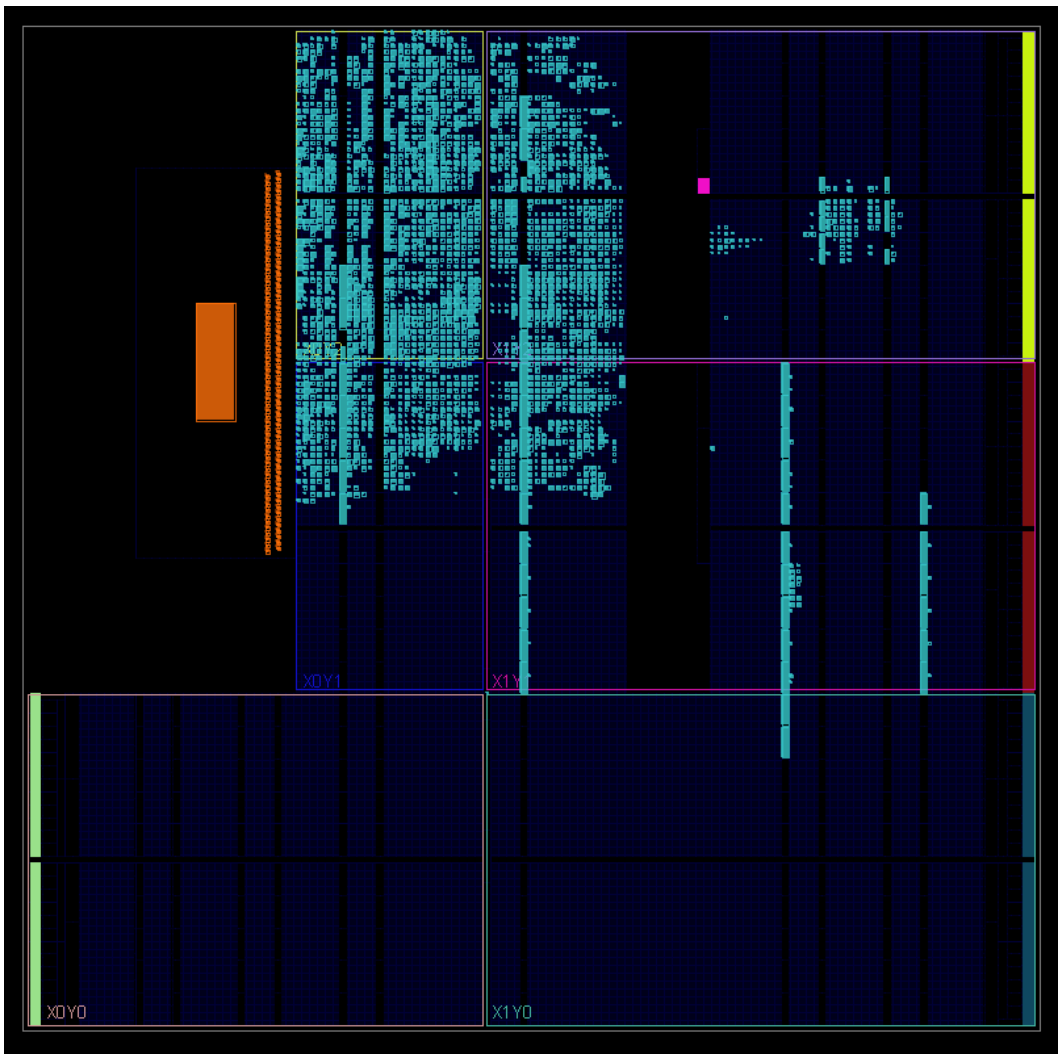


Figure 17 Implemented Visual

6. Acknowledgment:

A、 NCHU EE Undergraduate:

a. Chun-Wei Su

B、 NCHU EE MSPIC Lab. (611B):

a. Kun-Yung Chang

C、 NCHU EE ICs & Systems Lab. (612):

a. Hung-Jui Chang

b. Mo-Hsuan Hsiung

c. Shun-Liang Yeh

d. Chun-Yuan Hsiao

D、 NCHU EE Lab. 716:

a. Hsing-Yao Wang

b. Hsuan-Yu Lin

E、 NCHU EE VSIP-IC Lab. (908):

a. Wen-Chia Yang

7. Reference:

[1] Po-Wei Liu, “The Implementation of Image Compression JPEG2000,” M.S. thesis, Dept. Elect. Eng., DYU, Changhua, Taiwan, 2014.

[2] M.Puttaraju, and Dr.A.R.Aswatha “FPGA Implementation of 5/3 Integer DWT for Image Compression” International Journal of Advanced Computer Science and Applications, Vol. 3, No. 10, 2012

[3] G. K. Khan and A. G. Sawant, "Spartan 6 FPGA implementation of 2D-discrete wavelet transform in Verilog HDL," 2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT), 2016, pp. 139-143, doi: 10.1109/ICAECCT.2016.7942570

[4] Hardware Design of the Discrete Wavelet Transform: an Analysis of Complexity, Accuracy and Operating Frequency Dora M. Ballesteros L. 1, Diego Renza 2 and Luis Fernando Pedraza 3 Received: 28-04-2016 | Accepted: 21-10-2016 | Online: 18-11-2016 PACS: 84.40.Ua; 07.50.Qx doi:10.17230/ingciencia.12.24.6

[5] https://www.cnblogs.com/chengqi521/p/6732999.html

[6] https://www.cnblogs.com/amxiang/p/16543664.html

[7] https://zhuanlan.zhihu.com/p/608277782

[8] https://blog.csdn.net/weixin_38071135/article/details/118581250