

# Digital System Design SS2024

---

## Lab 3: JPEG-Encoder

Report submitted by:

**Soumya Ranjan Sabat**  
Matrikelnr: 1127993

**Azaz Hassan Khan**  
Matrikelnr: 1128032

**July/12/2024**

## Content

1	Introduction .....	3
1.1	Verification of DCT-IP.....	3
1.1.1	Draw a block diagram of the VHDL module,„ DCT_AAN”! Briefly explain with the help of IP documentation how to calculate the 2D DCT of an 8x8 matrix .....	3
1.1.2	A test bench for the DCT-IP core is also available. Explain the functions of the three components „TEST_DCT“, „DCT_BEH“ and „BMP_Generator“! What function does the process „ERROR_CALC“ have? .....	4
2	Verification of the Custom IP.....	4
2.1.1	Verify the functionality of the custom IP using the provided test bench.Simulate at least for 20μs (step 4). What is delay (number of clock cycles) between first data input and first data output? How many clock cycles does it take to read in one input matrix? Give min. and max. value of mean square error for this simulation! Attach simulation snapshots to verify your findings! 4	
3	Integration of DCT-IP into JPEG encoder project .....	6
3.1	Instantiate the DCT_AAN core .....	6
3.2	Get C-Program of JPEG-encoder .....	7
3.3	Screen shot of statistics shown in serial terminal and Compare BMP to JPEG pictures .....	8

# 1 Introduction

As FPGAs grow in size and complexity, and design timelines become tighter, leveraging third-party IP and design reuse is becoming essential. We utilize a powerful feature within the Vivado Design Suite known as the Vivado IP integrator. This tool allows you to create intricate system designs by instantiating and interconnecting IP from the Vivado IP catalog or other sources.

In this lab, our objective is to analyze a DCT-IP core and synthesize it onto FPGA programmable logic. The goal is to leverage the hardware implementation of the Discrete Cosine Transform (DCT) for rapid JPEG conversion. We employ a Zynq device from Xilinx, a System on Chip (SoC) that integrates an ARM dual-core Cortex-A9 Processing System with FPGA programmable logic.

The workflow involves creating an IP on FPGA programmable logic and interfacing it with a C++ project running on the ARM processor, culminating in a hardware/software co-design. The primary aim of this co-design is performance optimization.

The Discrete Cosine Transform (DCT) is widely used in signal and image processing, particularly for the lossy compression of images, such as JPEG encoding. Various FPGA implementations of the DCT exist. For this lab session, we are utilizing a DCT-IP provided by Unicores Systems, which is freely available via [opencores.org](http://opencores.org).

## 1: Top level RTL design

### 1.1 Verification of DCT-IP

#### 1.1.1 Draw a block diagram of the VHDL module, "DCT\_AAN"! Briefly explain with the help of IP documentation how to calculate the 2D DCT of an 8x8 matrix

Task-Description: Describe meaning with the help of IP documentation how to calculate 2D DCT.

Elaboration: Below is the implementation of a 2-D DCT for an 8x8 matrix. The output of the first stage undergoes transposition before being processed by the second stage. Specifically, the output of DCT8AAN1 is connected to DCT8AAN2. The DCT8AAN1 module performs the DCT on each row of the data matrix. The DCT\_BUF then converts the data from DCT8AAN1 to the format required by the next module. Subsequently, DCT8AAN2 performs the DCT on each column of the matrix. The output is then passed to another DCT\_BUF, which transposes the data matrix obtained from DCT8AAN2 to produce the final desired output.

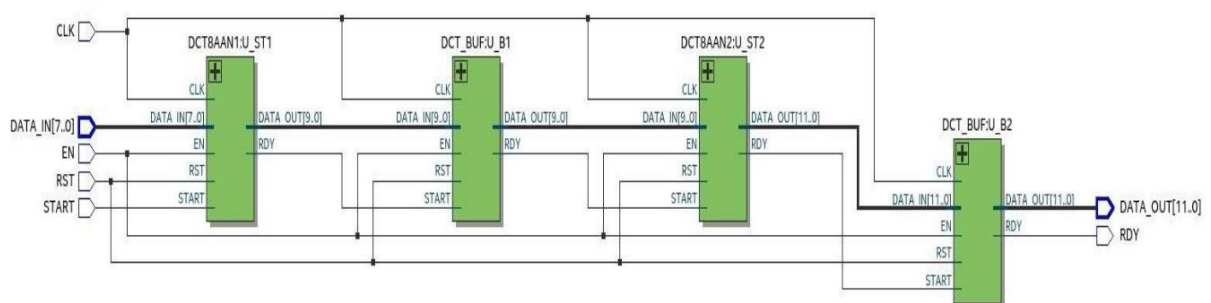


Fig 1.1 DCT\_AAN

**1.1.2 A test bench for the DCT-IP core is also available. Explain the functions of the three components „TEST\_DCT“, „DCT\_BEH“ and „BMP\_Generator“! What function does the process „ERROR\_CALC“ have?**

Task-Description: Explain TEST\_DCT, DCT\_BEH, BMP\_Generator and function ERROR\_CALC.

Elaboration:

**TEST\_DCT:** This is the test bench for the DCT IP. It instantiates the BMP\_GENERATOR, DCT\_AAN, and DCT\_BEH modules to verify the functionality of the DCT IP. The BMP\_GENERATOR creates the data pattern, and the DCT is calculated using two methods within this module. The first method utilizes the DCT\_AAN module, while the second method employs the DCT\_BEH module.

**DCT\_BEH:** This module serves as the behavioral model of the DCT processor, performing 2D DCT calculations using floating-point arithmetic.

**BMP\_GENERATOR:** This module generates an 8x8 matrix to serve as input data for the DCT modules.

**ERROR\_CALC:** This block computes the difference between the outputs of the DCT\_AAN and DCT\_BEH modules. The results of these differences are represented by the ERROR signals.

**Q2a. Explain the meaning of the following output signals: DCT[11:0], DCT\_STD[11:0], ERROR, QUADMEAN!**

Task-Description: Explain DCT[11:0], DCT\_STD[11:0], ERROR, QUADMEAN.

Elaboration:

**DCT[11:0]:** DCT[11:0] is the signal mapped to the output signal data\_out[11:0] generated by the module "DCT\_AAN".

**DCT\_STD[11:0]:** DCT\_STD[11:0] is the signal mapped to the output signal data\_out[11:0] generated by the module "DCT\_BEH".

**ERROR:** The ERROR signal represents the deviation between the two DCT output signals, DCT[11:0] and DCT\_STD[11:0].

**QUADMEAN:** The QUADMEAN signal represents the quad mean value of the deviation error derived from the ERROR signal.

## **2 Verification of the Custom IP**

**2.1.1 Verify the functionality of the custom IP using the provided test bench. Simulate at least for 20μs (step 4). What is delay (number of clock cycles) between first data input and first data output? How many clock cycles does it take to read in one input matrix? Give min. and max. value of mean square error for this simulation! Attach simulation snapshots to verify your findings!**

Task-Description: Analysis of the waveform.

Elaboration:



Figure 1.2 Simulation of custom IP

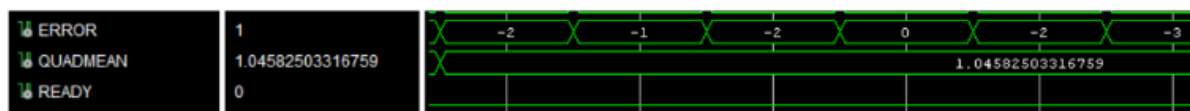


Figure 2.3 Quad Mean Error Max and Min

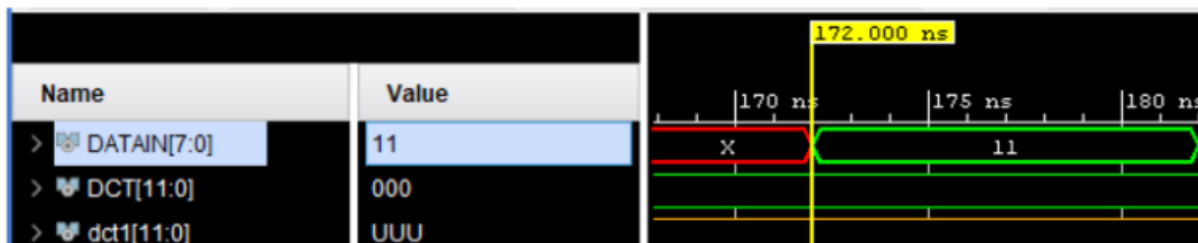


Figure 3.4 First input signal



Figure 4.5 First input signal

**First Input Signal:**

- Arrival Time: 172 ns

**First Output Signal:**

- Arrival Time: 1505 ns

**Difference:**

- Time Difference: 1333 ns

**Clock Period:**

- Duration: 10 ns

**Clock Cycles:**

- Number of Clock Cycles between First Input and First Output: Approximately 133 cycles

**Simulation Details:**

- Total Simulation Duration: 20 microseconds

- Observed Delay between First Input and First Output: 132 cycles

- Cycles to Read One 8x8 Input Matrix: 64 cycles

**Quad Mean Error Values:**

- Minimum Value: 0

- Maximum Value: 1.0458

### 3 Integration of DCT-IP into JPEG encoder project

#### 3.1 Instantiate the DCT\_AAN core

Port map section of DCT\_AAN instantiation -

```
U2: DCT_AAN
  generic map(d_SIGNED => 0, scale_out => 1)
  port map(
    CLK => S_AXI_ACLK,
    RST => not(S_AXI_ARESETN),
    START => start_pulse2,
    EN => slv_reg49(1),
    DATA_IN => dct_data,
    DATA_OUT => dct_core_out,
    RDY => dct_core_rdy
  );
```

Figure 3: DCT\_AAN instantiation

## 3.2 Get C-Program of JPEG-encoder

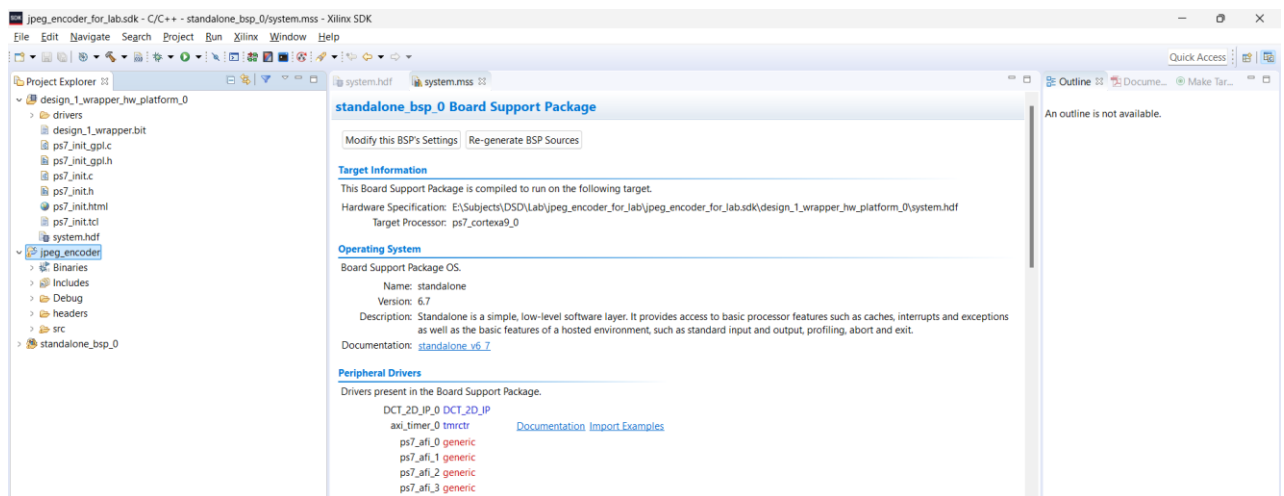


Figure 4: jpeg\_encoder.elf

### 3.3 Screen shot of statistics shown in serial terminal and Compare BMP to JPEG pictures

```
Overview Source
Problems Tasks Console Properties SDK Terminal
Connected to: Serial ( COM4, 115200, 0, 8 )
INFO: Converting image row: 73 of 75
OLED: Status: 98%
INFO: Converting image row: 74 of 75
OLED: Status: 100%
INFO: Converting image row: 75 of 75
INFO: JPEG Image conversion for: /JPG/RAY.JPG finished.
INFO: JPEG File size: 760006 byte
INFO: JPEG compression rate: 100
INFO: JPEG image /BMP/RAY.BMP was encoded in: 23181ms
INFO: I/O time amount: 10179ms
INFO: Bitmap Block extraction time amount: 136ms
INFO: DCT time amount: 2404ms
INFO: Color Conversion time amount: 1251ms
INFO: Quantization time amount: 1782ms
INFO: Huffman encoding time amount: 7334ms
```

Figure 5: Serial Terminal

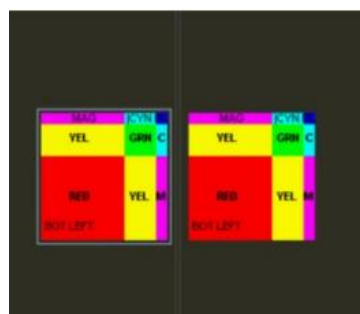


Figure 6: Comparison of FLAG\_B24 image



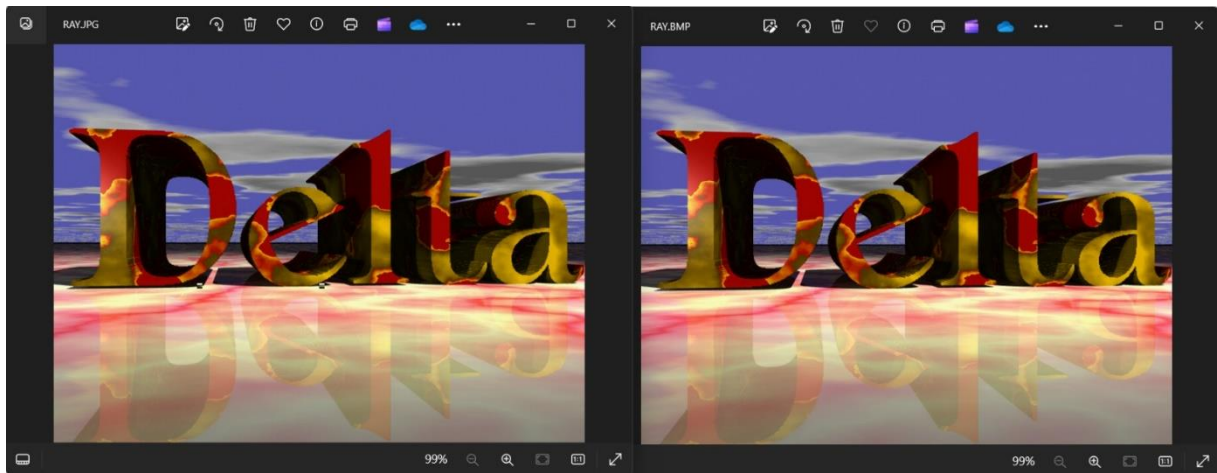


Figure 7: Comparision of RAY Image

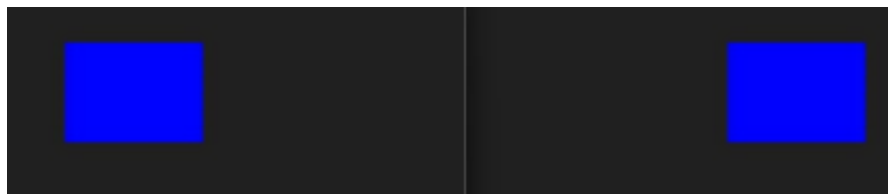


Figure 8: Comparision of BLU Image

## References

- [1] Xilinx, "Clocking Wizard," [Online]. Available: [https://www.xilinx.com/products/intellectual-property/clocking\\_wizard.html](https://www.xilinx.com/products/intellectual-property/clocking_wizard.html).
- [2] Xilinx, "Zynq-7000 Processing System IP," [Online]. Available: [https://www.xilinx.com/products/intellectual-property/processing\\_system7.html](https://www.xilinx.com/products/intellectual-property/processing_system7.html).
- [3] Xilinx, "Processor System Reset Module v5.0 - LogiCORE IP Product Guide," PG164, 2015.
- [4] Xilinx, "AXI4-Stream Interconnect," ISE Design Suite, [Online]. Available: [https://www.xilinx.com/products/intellectual-property/axi4-stream\\_interconnect.html](https://www.xilinx.com/products/intellectual-property/axi4-stream_interconnect.html). [Accessed January 2020].
- [5] Xilinx, "AXI Interconnect," Vivado Design Suite, [Online]. Available: [https://www.xilinx.com/products/intellectual-property/axi\\_interconnect.html](https://www.xilinx.com/products/intellectual-property/axi_interconnect.html). [Accessed January 2020].
- [6] Analog devices, Inc, *ADAU1761 - SigmaDSP Stereo, Low Power, 96 kHz, 24-Bit Audio Codec with Integrated PLL*, One Technology Way, 2009.
- [7] H. E. Electronics, "ZYNQ: Create an I2S Transmitter to Send Audio Signals," [Online]. Available: <http://www.harald-rosenfeldt.de/2017/12/30/zynq-create-an-i2s-transmitter-to-send-audio-signals/>. [Accessed January 2020].
- [8] Wikipedia, "Audio filter," July 2018. [Online]. Available: [https://en.wikipedia.org/wiki/Audio\\_filter](https://en.wikipedia.org/wiki/Audio_filter). [Accessed January 2020].