2EC202-FPGA Based System Design

Department of Electronic and Communication Engineering

School of Technology

Nirma University

Special Assignment

Topic : Analog To Digital Converter

Roll Number : 22bec042 (Neha Golani)

Submitted to: Prof. Purvi Patel

Abstract : This report explores the interfacing of the ADC0809 analog-to-digital converter IC with an FPGA DE2 kit, aiming to seamlessly integrate analog signals into digital systems. The ADC0809's features, including resolution and sampling rate, complement the FPGA DE2's flexibility for digital logic implementation. The interface design is detailed, covering connections, signal flow, and hardware setup considerations. FPGA programming aspects, such as Verilog or VHDL code snippets, are discussed alongside software interface configurations. Testing procedures ensure accuracy, and results analysis underscores the importance of this interface in bridging analog-digital realms for various applications, providing a comprehensive guide for engineers and enthusiasts alike.

Keywords :

ADC0809, Verilog HDL, FPGA, Conversion, Analog signal ,Hardware Setup Signal Flow.

Literature Survey/State of the Art Technology Available :

- **High-Speed ADCs with FPGA Interfaces**: Advanced ADCs, such as those with high resolutions (e.g., 16-bit or higher) and fast sampling rates (in the GHz range), are commonly interfaced with FPGAs to handle high-speed data acquisition tasks in fields like telecommunications, radar, and medical imaging.
- **Mixed-Signal System-on-Chip (SoC) Solutions:** Modern SoCs integrate analog components (including ADCs) with digital processing units (FPGAs or microprocessors) on a single chip, enabling compact and power-efficient systems for applications like IoT sensors and portable devices.
- **JESD204B/C Interface Standard:** JESD204B/C is a high-speed serial interface standard widely adopted for connecting ADCs (and DACs) to FPGAs and other digital processing units. It offers benefits like reduced pin count, simplified board layout, and enhanced data throughput.
- **FPGA Development Tools:** Manufacturers like Xilinx and Altera (now part of Intel) offer comprehensive FPGA development tools, such as Xilinx Vivado and Altera Quartus Prime, which include libraries and IP cores for interfacing with ADCs of various types and specifications.
- **Application-Specific Reference Designs:** Many semiconductor companies provide reference designs and application notes specifically tailored for interfacing their ADC products with FPGAs, offering guidelines, code examples, and performance optimizations for different use cases.

- **Open-Source Projects and Communities:** Online platforms like GitHub host open-source projects related to ADC-FPG A interfacing, where developers share code, designs, and insights into implementing efficient and reliable interfaces for various ADC models and FPGA boards.
- **Research Papers and Publications:** Academic and industry research papers cover topics such as high-speed data acquisition, real-time signal processing, FPGA-based control systems, and optimization techniques for ADC-FPGA interfaces, providing valuable knowledge and advancements in the field.

Limitations or Drawbacks of Currently Available Technology :

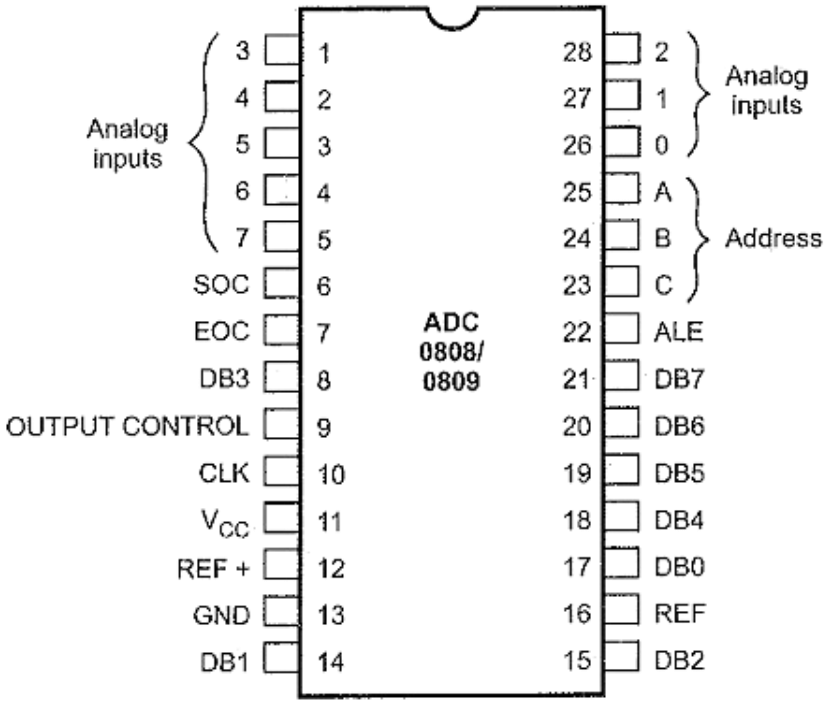
- **Complexity and Design Effort:** Designing and implementing a robust ADC-FPGA interface often requires substantial expertise in both analog and digital domains, making it challenging for novice designers or small-scale projects.
- **Signal Integrity Challenges:** High-speed ADCs demand meticulous attention to signal integrity issues like signal degradation, jitter, and noise, especially in long-distance PCB traces or when using high-speed serial interfaces like JESD204B/C.
- **Power Consumption:** Some high-resolution and high-speed ADCs consume significant power, which can be a limitation in power-constrained applications or portable devices where energy efficiency is crucial.
- **Cost:** Advanced ADCs with superior specifications (e.g., high resolution, fast sampling rates) can be expensive, adding to the overall system cost, especially in mass production or cost-sensitive applications.
- **Compatibility and Interoperability:** Ensuring compatibility and interoperability between different ADC models, FPGA boards, and development tools from various manufacturers can be challenging, leading to integration issues and additional development effort.
- **Firmware and Software Complexity:** Writing and optimizing firmware or software for FPGA-based ADC interfaces require in-depth knowledge of FPGA programming languages (Verilog or VHDL) and signal processing algorithms, which can be time-consuming and error-prone.
- **Testing and Validation:** Verifying the functionality, accuracy, and reliability of an ADC-FPGA interface often necessitates thorough testing procedures, including simulation, hardware testing, and calibration, which can be resource-intensive.
- **Scalability:** Scaling ADC-FPGA interfaces for applications requiring multiple channels or distributed data acquisition systems can pose scalability challenges, requiring careful system architecture and synchronization mechanisms.

- **Real-Time Processing Constraints:** Real-time processing of high-speed ADC data within FPGA resources may encounter limitations in terms of processing throughput, latency, and resource utilization, especially for complex signal processing tasks.

Proposed Solution/Methodology :

- **Simplified Design Tools and Libraries:** Develop user-friendly design tools and libraries tailored for ADC-FPGA interfaces, abstracting complex hardware details and providing pre-verified building blocks for common tasks like data acquisition, signal processing, and interfacing protocols.
- **Power-Optimized ADC Selection:** Evaluate and select ADCs with lower power consumption while meeting the required resolution and sampling rate specifications, prioritizing energy-efficient designs for portable or battery-powered applications.
- **Signal Integrity Analysis and Design Guidelines:** Provide comprehensive guidelines, simulation models, and design best practices for ensuring signal integrity in high-speed ADC-FPGA interfaces, addressing issues like impedance matching, routing, and noise reduction.
- **Interoperability Standards:** Advocate for industry-wide interoperability standards or frameworks for ADC-FPGA interfaces, promoting compatibility between different ADC and FPGA models, and facilitating seamless integration and interchangeability.
- **Automated Testing and Verification Tools:** Develop automated testing tools and methodologies for verifying ADC-FPGA interfaces, including simulation environments, self-test routines, and calibration procedures to streamline validation efforts and reduce manual testing time.
- **Scalable Architectures:** Design scalable architectures and communication protocols for multi-channel ADC systems and distributed data acquisition setups, incorporating synchronization mechanisms, data aggregation techniques, and efficient data transfer protocols.
- **Real-Time Processing Optimization:** Implement optimized algorithms and hardware-accelerated processing modules within FPGAs to handle real-time processing of high-speed ADC data, leveraging parallel processing, pipelining, and resource utilization techniques for improved performance.
- **Collaborative Development and Open Standards:** Foster collaboration among industry stakeholders, academia, and open-source communities to share knowledge, develop open standards, and contribute to the advancement of ADC-FPGA interface technologies, promoting innovation and knowledge exchange.

ADC0809 IC Pin Configuration :



Explanation of each pin of the ADC0809:

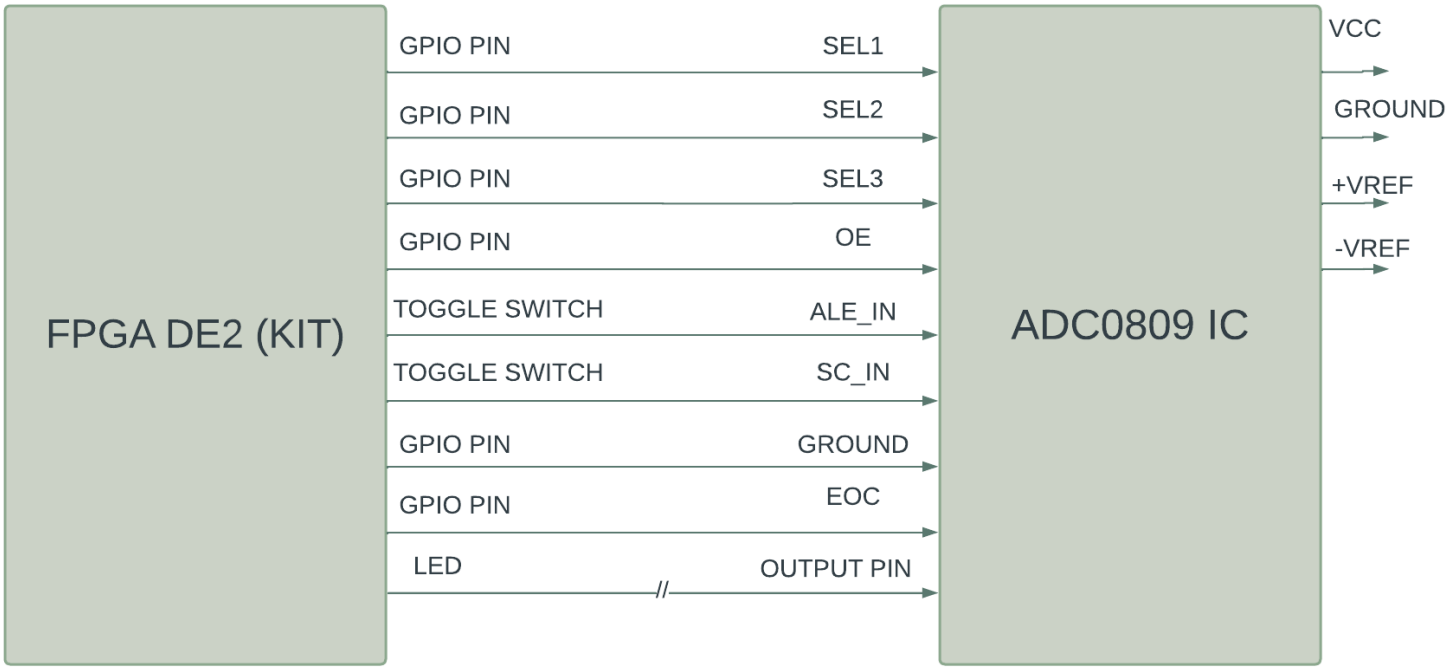
The ADC0809 is an 8-bit successive approximation analog-to-digital converter (ADC) integrated circuit (IC).

1. **VCC :**
    a. This pin is connected to the positive power supply voltage (+Vcc).
    b. It typically operates within the range of +4.5V to +6.3V.
    c. Provides power to the internal circuitry of the ADC.
2. **GND :**
    a. This pin is connected to the ground or 0V reference.
    b. It serves as the ground connection for the ADC's internal circuitry and external components.

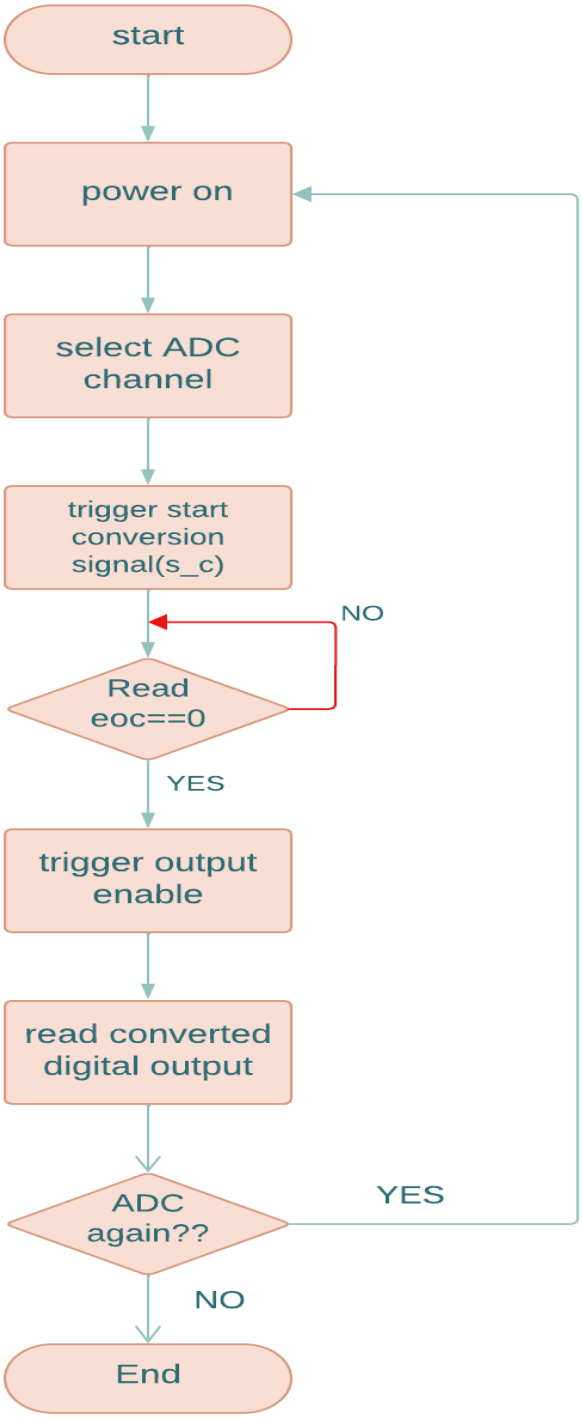
3. **+Vref , -Vref :**
    a. The Vref pin is used to provide the reference voltage for the ADC conversion.
    b. It sets the maximum voltage level that can be converted to a digital value (e.g., +Vref for maximum input , -Vref for minimum input).

4. **CS (Chip Select,(A,B,C):**
    a. The CS pin is used to enable/disable the ADC for conversion.
    b. When CS is high (logical 1), the ADC is enabled and ready to convert analog input.
5. **IN0-IN7 (Analog Input Channels, Pins):**
    a. These pins are the analog input channels where the voltage to be converted is applied.
    b. IN0 corresponds to the least significant bit (LSB), and IN7 corresponds to the most significant bit (MSB) of the digital output.
6. **ALE (Address Latch Enable):**
    a. The ALE pin is used for multiplexing the address and control signals during a conversion cycle.
    b. It latches the address and control signals when transitioning from high to low.
7. **EOC (End of Conversion):**
    a. The EOC pin indicates the end of the conversion cycle.
    b. It goes high (logical 1) when a conversion is complete and the digital data is ready for reading.
8. **OE (Output Enable):**
    a. The OE pin controls the output buffer for the digital data.
    b. When OE is low (logical 0), the digital output data is enabled and can be read from the ADC's output pins.
9. **START (Start Conversion):**
    a. The START pin initiates a conversion cycle when transitioning from high to low.
    b. It serves as an alternative to the WR pin for starting a conversion process.
10. **DB0-DB7 (Data Bus):**
    a. These pins form the data bus of the ADC0809, collectively transmitting the 8-bit digital output data.
    b. Each pin corresponds to one bit of the digital output data, with DB0 being the least significant bit (LSB) and DB7 being the most significant bit (MSB).

Block Diagram :



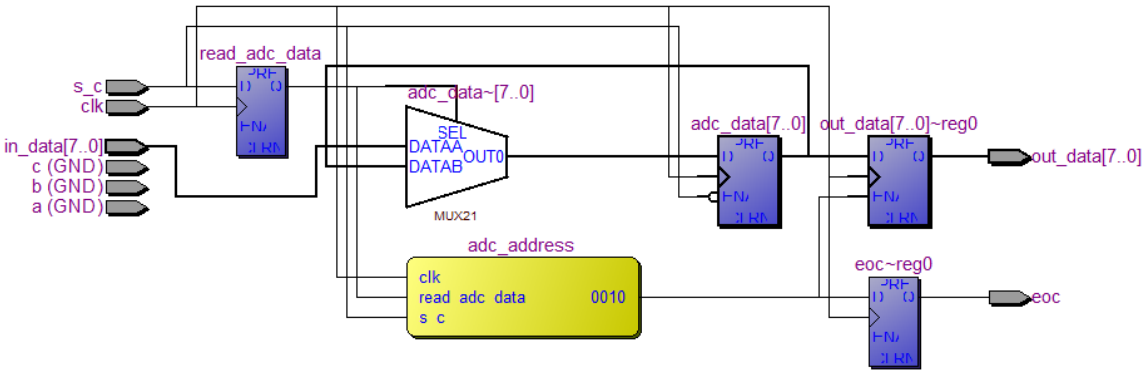
Flow Chart of the Code :



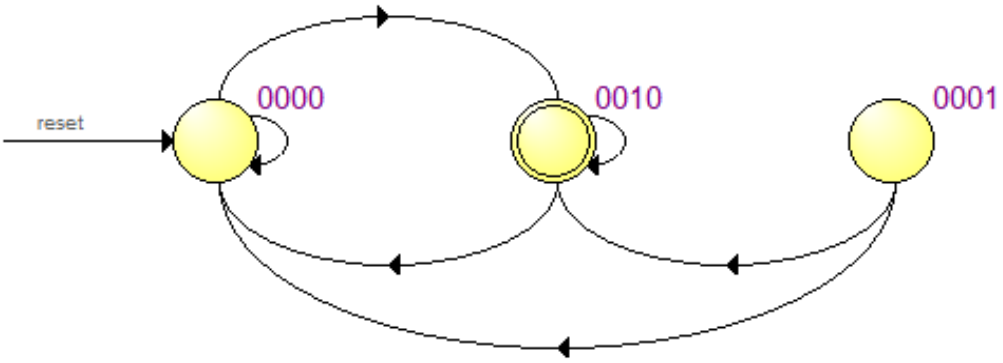
Verilog Code :

```
// a,b,c - select line
//ale_in and ale_out - Address Latch Enable
//sc_in and sc_out - start conversion
//clk_in and clk_out - clock_in (internal clock-50 Mhz),clock_out(divided clock-0.5 khz)
//oe - output enable
//eoc - end of conversion
//in_data - output of ADC ic is using as a input for fpga
//out_data - 8 bit digital output data


module ADC(a,b,c,ale_in,ale_out,sc_in,sc_out,clk_in,clk_out,oe,eoc,in_data,out_data);
input clk_in,ale_in,sc_in;
output reg sc_out,ale_out;
output clk_out;
input [7:0]in_data;
output reg a,b,c,oe;
input eoc;
output reg [7:0]out_data;
reg count=4'b0010;
output reg led;

//clock divider
reg [16:0]out = 0;
reg divide;
reg [31:0] td = 50000 ;
reg [31:0] ctr = 0;
always @(posedge clk_in)
begin
ctr = ctr + 1;
if(ctr==td)
begin
ctr = 0;
if( divide == 1 )
divide = 0 ;
else
divide = 1;
end
end

//assign our actual clock is divided clock
assign clk_out=divide;

always @ (posedge divide)
begin

//for transition
ale_out  =  divide  &  ale_in;
sc_out  =   divide &  sc_in;

//for transition
ale_out  =  divide  &  ale_in;
sc_out  =   divide &  sc_in;

//check end of conversion
if(eoc==0)
count=count-1;    //make a counter for delay time when eoc = 0
if(count==0)
begin
oe=1'b1;
out_data=in_data;
end
else
oe=1'b0;
end
endmodule
```

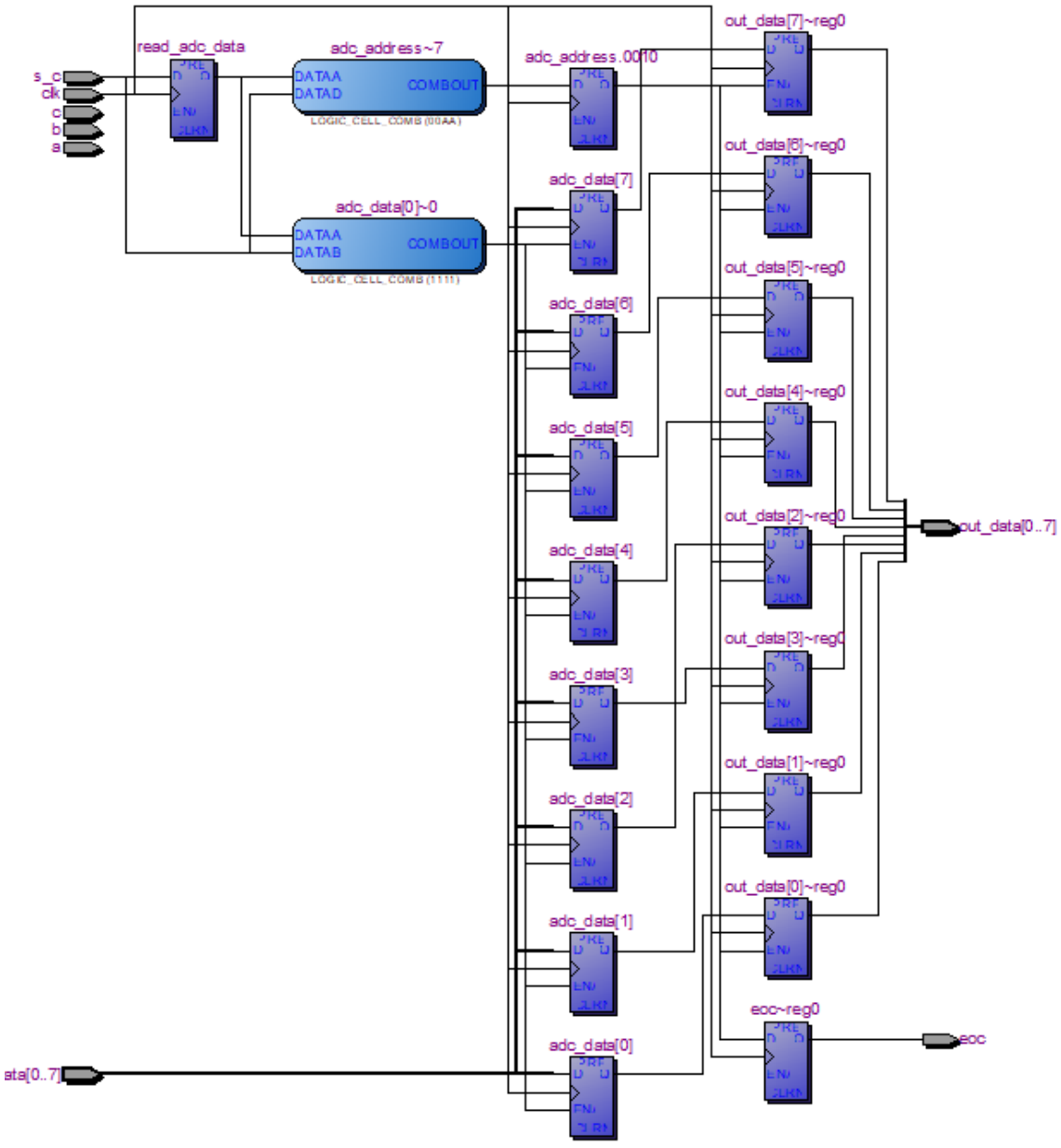
Results in Terms of RTL, TTL:



RTL



STATE DIAGRAM


read_adc_data
adc_address~7
DATAA
DATAD
COMBOUT
LOGIC_CELL_COMB (00AA)
adc_data[0]~0
DATAA
DATAB
COMBOUT
LOGIC_CELL_COMB (1111)
s_c
clk
c
b
a
ata[0..7]
adc_address.0010
adc_data[7]
adc_data[6]
adc_data[5]
adc_data[4]
adc_data[3]
adc_data[2]
adc_data[1]
adc_data[0]
out_data[7]~reg0
out_data[6]~reg0
out_data[5]~reg0
out_data[4]~reg0
out_data[2]~reg0
out_data[3]~reg0
out_data[1]~reg0
out_data[0]~reg0
eoc~reg0
out_data[0..7]
eoc

TTL

RESULTS :

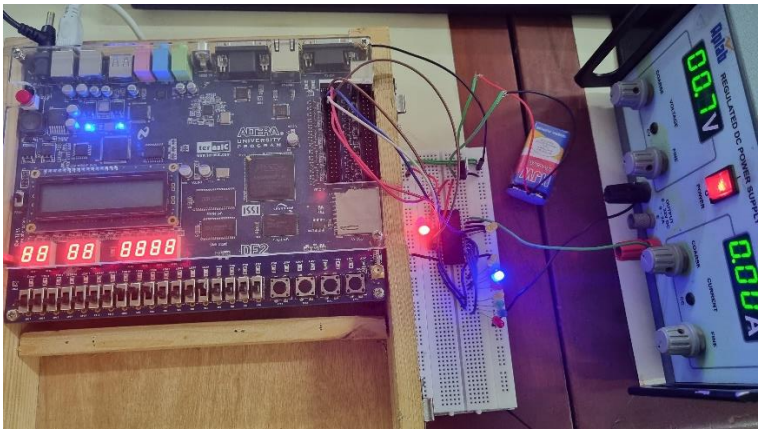


Fig 1.

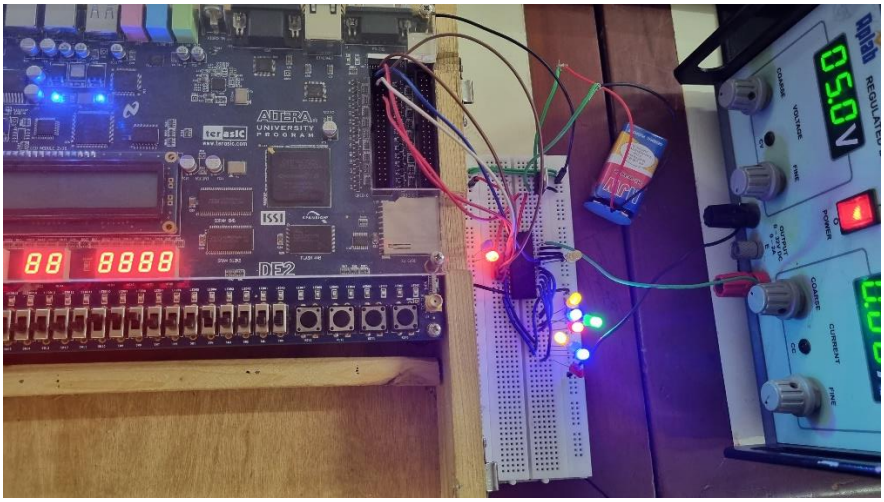


Fig 2.

Conclusion :

The Verilog HDL ADC module we've developed offers a versatile and adaptable solution for integrating ADC capabilities into FPGA-based systems. This module gives users control over the conversion process and data retrieval, making it easier to handle analog signal processing in a wide range of applications.

References :

[1] Roth Jr., C. H., & John, L. K. (2011). Digital Systems Design Using Verilog. Cengage Learning.

[2] Palnitkar, S. (2003). Verilog HDL: A Guide to Digital Design and Synthesis. Prentice Hall.

[3] Smith, Kenneth W. "Analog-to-Digital Conversion." Oxford University Press, 1997

[4] Mazidi, M. A., & McKinlay, R. D. (2013). The 8051 Microcontroller and Embedded Systems: Using Assembly and C. Pearson.