MY PROJECTS

ADHIRAJ MANDAL

ELECTRONICS AND TELECOMMUNICATION ENGINEERING STUDENT

TABLE OF CONTENTS

AHB2APB BRIDGE RTL DESIGN

Objective:

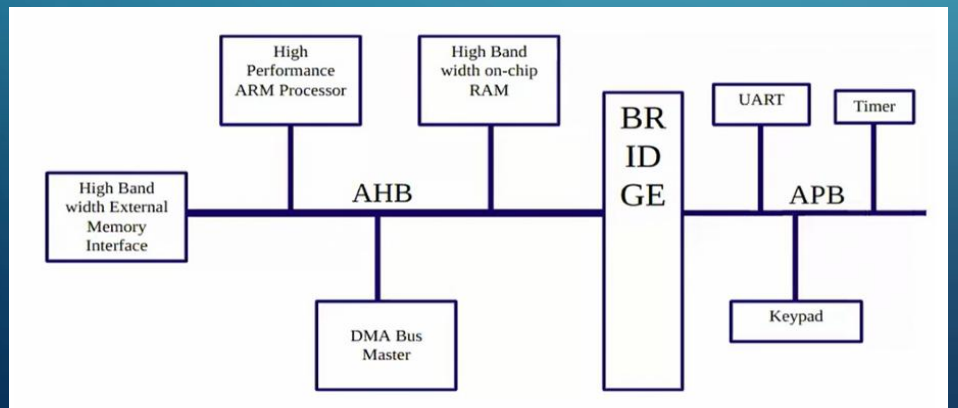
- To design and implement an **AHB to APB Bridge** for seamless communication between high-speed AHB peripherals and low-speed APB devices in SoC systems.

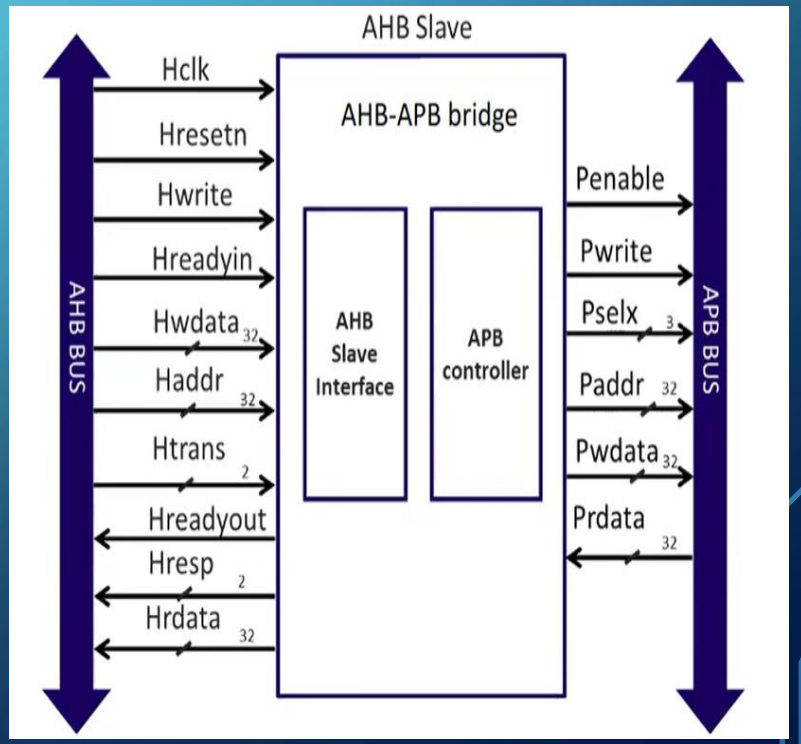
Architecture Overview:

- The bridge consists of two main blocks —
- **AHB Slave Interface:** Captures and validates AHB transactions, extracting address, data, and control signals.
- **APB Controller:** Converts validated AHB signals into APB protocol signals (PADDR, PWRITE, PENABLE, PSELx, PWDATA).
- Both blocks are integrated in a **top-level bridge module** that ensures end-to-end data transfer and synchronization between the two buses.

Key Features & Implementation:

- Implemented RTL design for both AHB Slave and APB Controller.
- Verified design through **simulation and waveform analysis.**
- Performed **synthesis** to validate timing and resource utilization.
- Designed using **AMBA AHB and APB protocols.**

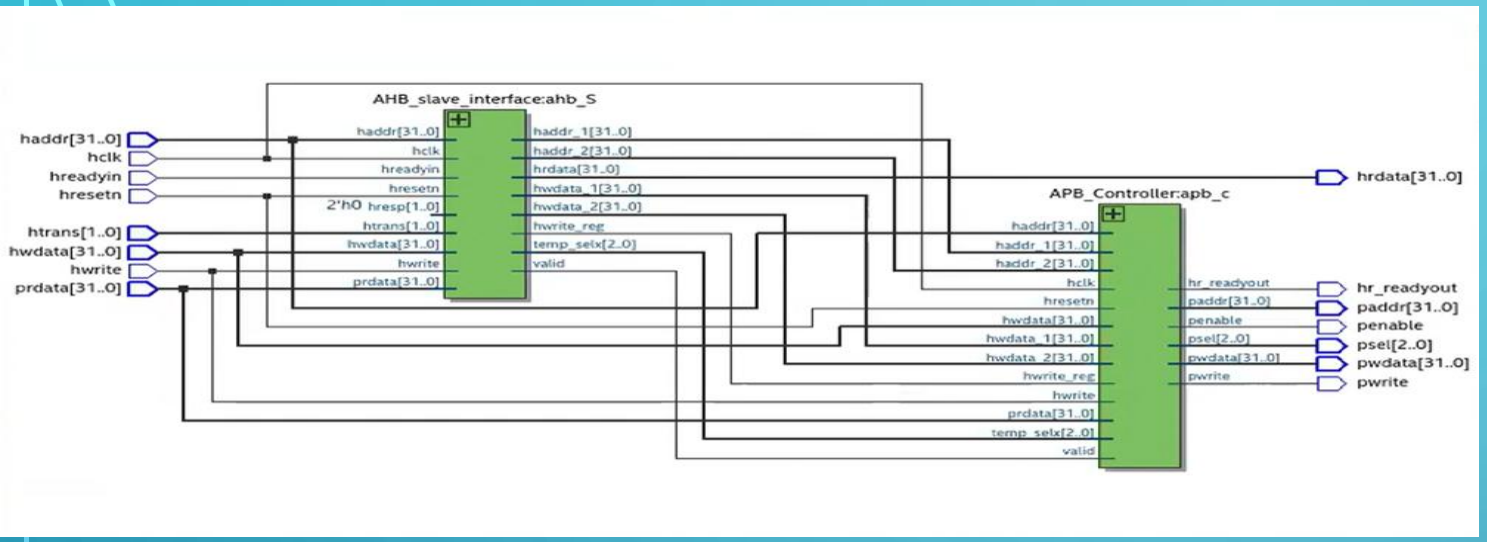


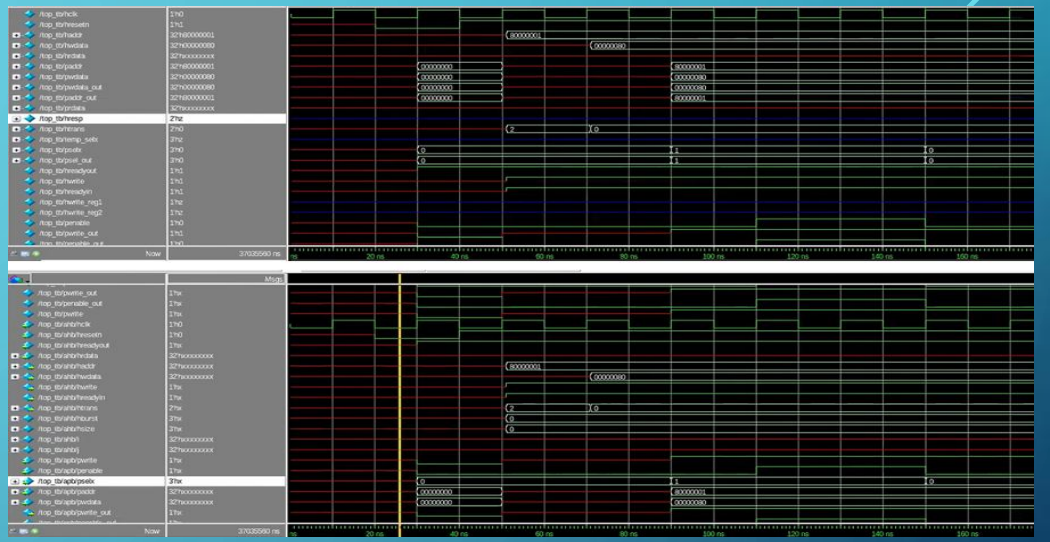


Outcome & Learning:

- Successfully designed and verified an efficient bridge interface.
- Gained hands-on experience in **bus protocol conversion, digital design,** and **RTL verification** using Verilog.

Organization: Maven Silicon **Role:** Intern **Tools Used:** Verilog HDL, ModelSim, Quartus Prime





aceic™ Accelerate Collaborate Excel | MAVEN SILICON Centre of Excellence in Semicon

CERTIFICATE OF COMPLETION

This is to certify that Mr./Ms. **Adhiraj Mandal** *a student of* **B.TECH** *from* **Indian Institute of Engineering Science and Technology Shibpur, West Bengal** *& has successfully completed VLSI Design Internship Program from* **04-12-2024** *to* **18-01-2025**

During the internship program with us, they worked on **AHBtoAPB Bridge RTL Design** *project.*

We wish them all the best for their future endeavours.

Date: 30-01-2025
Place: Bengaluru

MSUID: MS/23-24/5794

MAVEN SILICON SOFTECH PVT. LTD. BANGALORE

SIVAKUMAR P R
FOUNDER & CEO, MAVEN SILICON

LOW POWER UART AND TIMER IP BLOCK FOR SOC INTEGRATION IN VERILOG

Overview:

- Designed and verified a **low-power UART (baud generator, TX/RX)** and a **programmable Timer IP** in synthesizable Verilog for integration into a custom SoC. The project focused on modular IP design, efficient communication, and accurate timing control while ensuring low power and high reliability.

Key Features & Components:

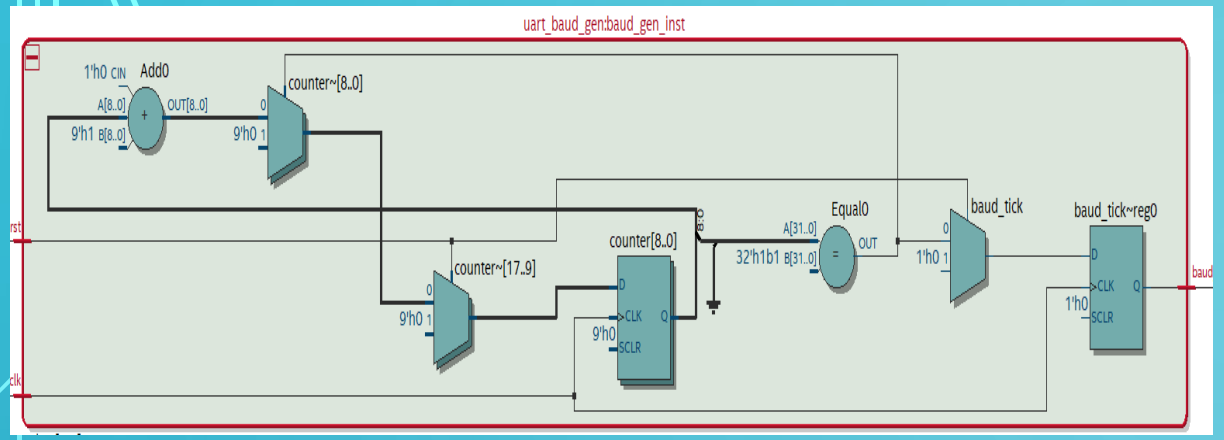
- **UART Module:** Implemented baud rate generator, transmitter, and receiver with configurable parameters.
- **Timer IP:** Designed programmable timer with start/stop, reset, and interrupt generation functionality.
- **Integration:** Added SoC wrapper for UART + Timer connection with **memory-mapped registers, IRQ lines,** and **clock-gating** for power efficiency.
- **Verification:** Developed **self-checking testbenches** for each module to validate functionality and timing accuracy.

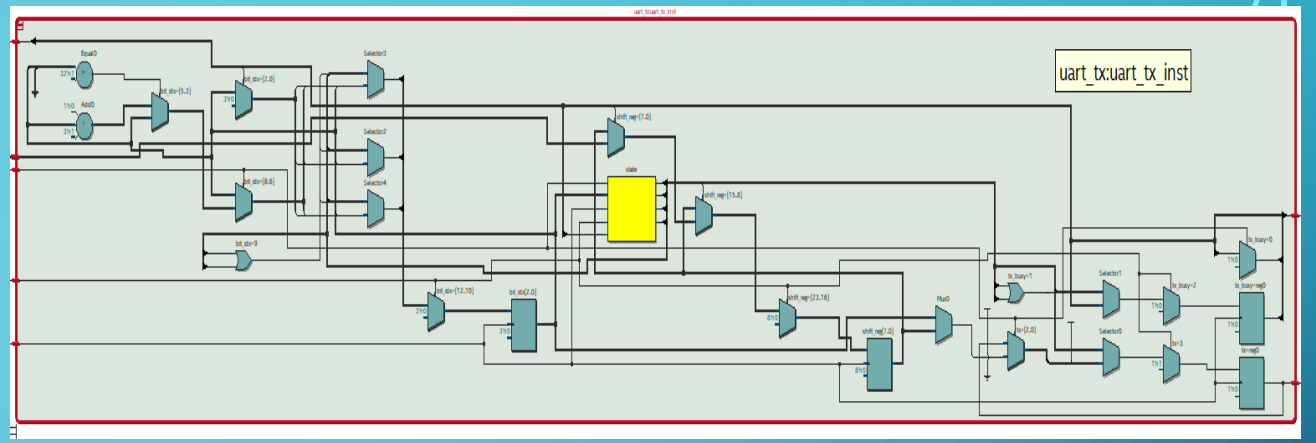
Tools & Technologies:

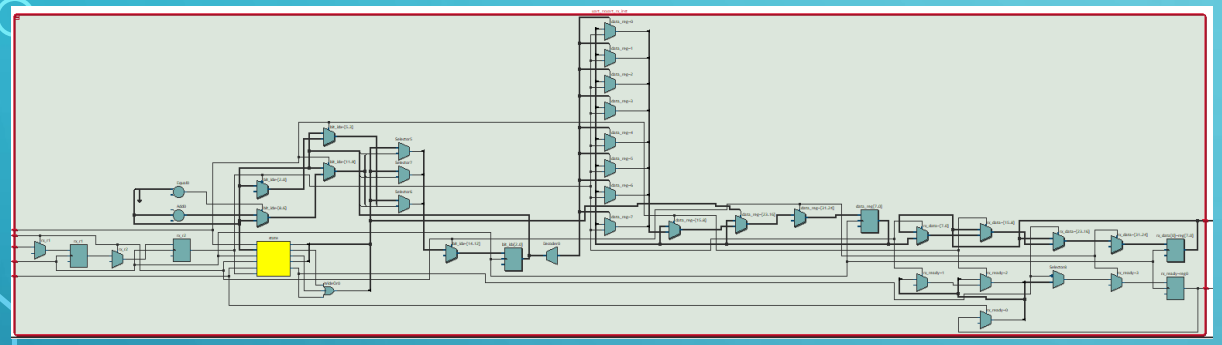
- Verilog HDL | Quartus Prime (Synthesis) | ModelSim (Simulation & Verification)

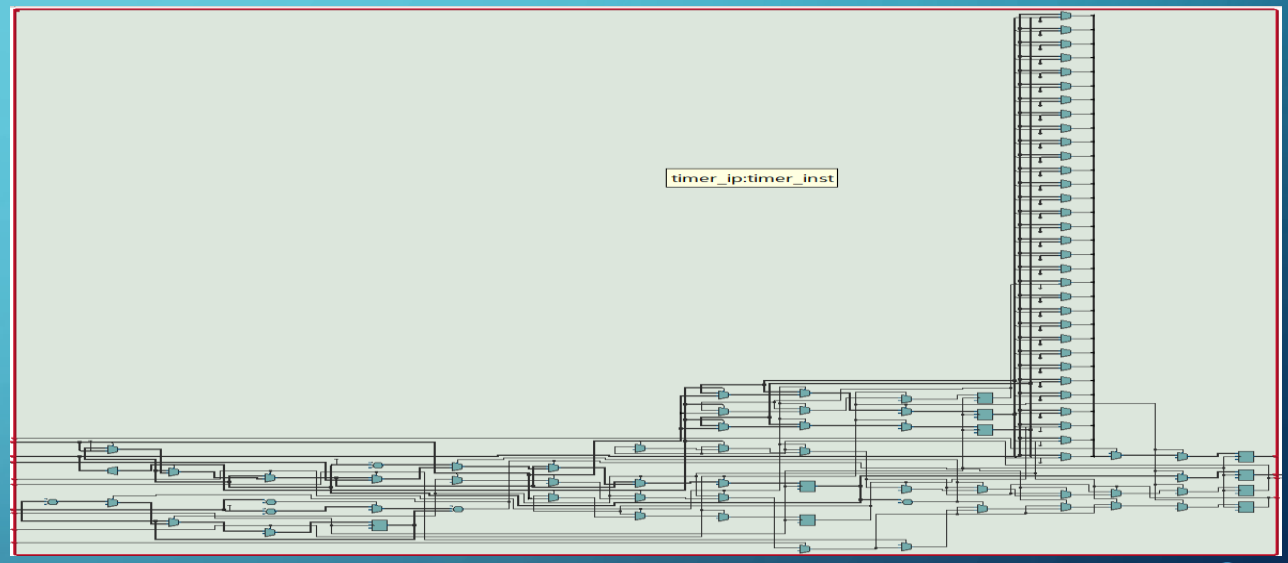
Outcome:

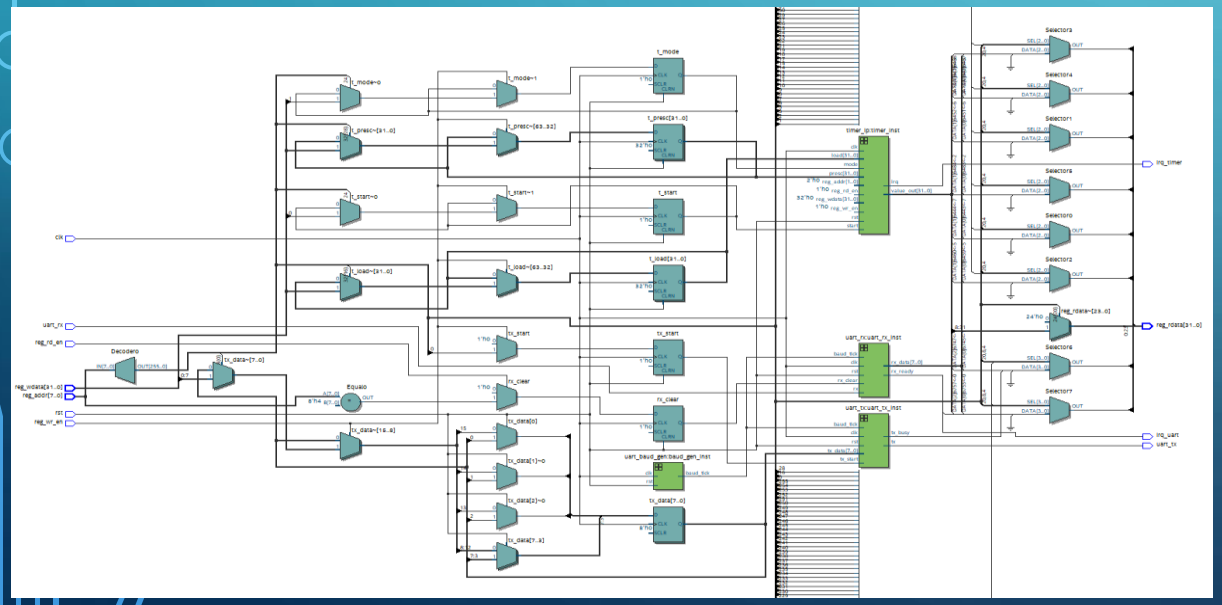
- Successfully implemented and verified synthesizable UART and Timer IP blocks ready for SoC integration, demonstrating strong skills in digital design, HDL coding, and functional verification.

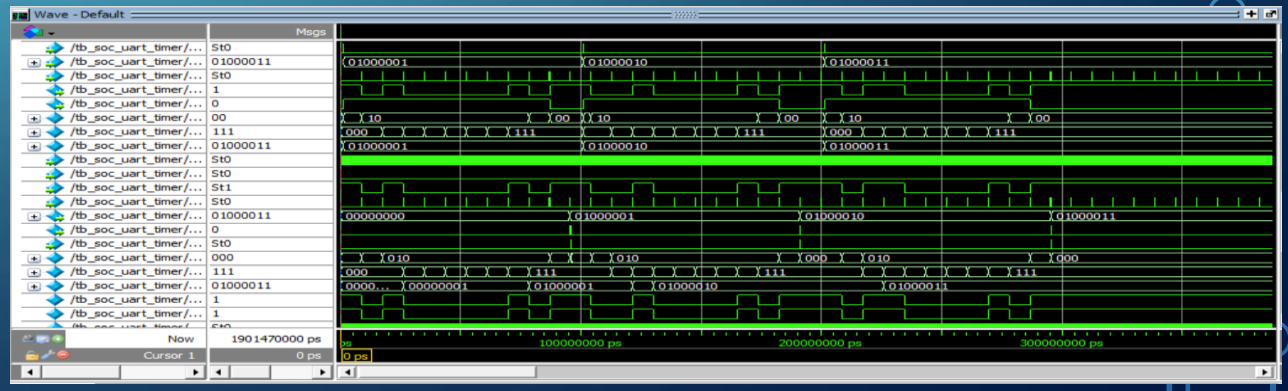

uart_baud_gen:baud_gen_inst
1'h0 CIN
Add0
A[8..0]
OUT[8..0]
9'h1 B[8..0]
counter~[8..0]
9'h0 1
counter~[17..9]
counter[8..0]
Equal0
A[31..0]
32'h1b1 B[31..0]
OUT
baud_tick
1'h0 1
baud_tick~reg0
D
CLK
Q
SCLR
1'h0
rst
clk
baud


uart_tx:uart_tx_inst




timer_ip:timer_inst




Wave - Default
Msgs
/tb_soc_uart_timer/...
Now
1901470000 ps
Cursor 1
0 ps
100000000 ps
200000000 ps
300000000 ps

AI-POWERED HDL GENERATION TOOL FOR ASIC/FPGA RTL DESIGN & TESTBENCH

Overview:

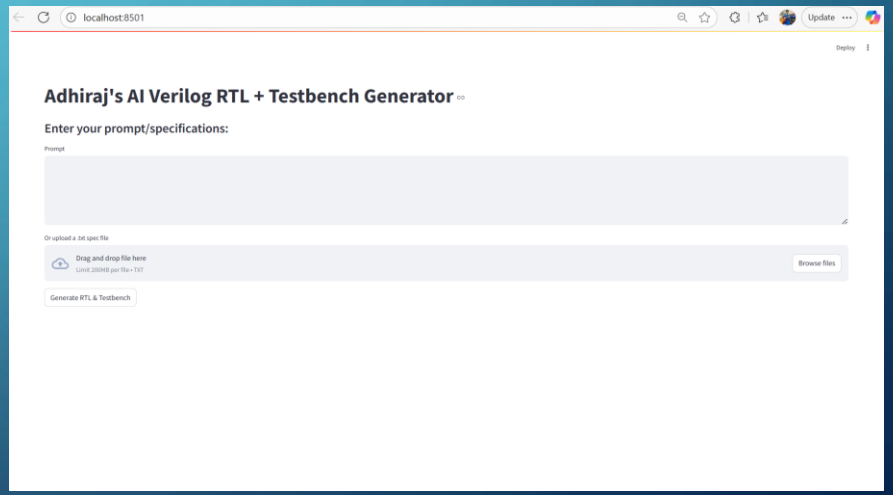
- Developed an **AI-assisted HDL generation tool** that converts **natural-language or JSON specification inputs** into **synthesizable, parameterized Verilog RTL and corresponding self-checking testbenches**.
  The tool leverages **large language models (LLMs)** for intelligent code synthesis and uses a **Streamlit-based front-end** for easy interaction, allowing users to type or upload specs and instantly generate Verilog code.

Key Features:

- Generates **Verilog RTL and Testbench** from plain-text or structured JSON specs
- Supports **modular, parameterized design generation** for ASIC/FPGA front-end flow
- Enables **code download directly** from the UI for easy simulation or synthesis
- Demonstrates **real-world design examples** — ALU, D Flip-Flop, Counter, FIFO, and AXI4-Lite Memory Mapped Timer Peripheral

Architecture:

- **Front-End:** Streamlit interface for user input and output display
- **Back-End:** Python-based LLM integration using OpenRouter API
- **Processing Pipeline:**
  Specification → Prompt Engineering → LLM Code Generation → Output Display



Tools & Technologies:

- Python | Streamlit | Verilog HDL | OpenRouter API | Prompt Engineering | LLMs

Impact:

- Bridges **AI and semiconductor design automation**, accelerating RTL development, reducing manual effort, and demonstrating how **AI can assist VLSI front-end design workflows** in industry applications.

Deploy

Adhiraj's AI Verilog RTL + Testbench Generator

Enter your prompt/specifications:

Prompt

Design a 4-bit ALU in Verilog that supports basic arithmetic and logic operations like addition, subtraction, AND, OR, XOR, shift, increment, and decrement.

Press Ctrl+Enter to apply

Or upload a .txt spec file

Drag and drop file here
Limit 200MB per file • TXT

Browse files

Generate RTL & Testbench

localhost:8501

Deploy

Generated RTL Code

```
module alu_4bit (
    input [3:0] a,
    input [3:0] b,
    input [2:0] op,
    output reg [3:0] result,
    output reg carry
);
always @(*) begin
    case(op)
        3'b000: {carry,result} = a + b;
        3'b001: {carry,result} = a - b;
        3'b010: result = a & b;
        3'b011: result = a | b;
        3'b100: result = a ^ b;
        3'b101: result = ~(a | b);
        default: result = 4'b0000;
    endcase
end
endmodule
```

Download RTL (.v)

Generated Testbench Code

```
module tb_alu_4bit;
reg [3:0] a,b;
reg [2:0] op;
wire [3:0] result;
wire carry;

alu_4bit uut(.a(a),.b(b),.op(op),.result(result),.carry(carry));

initial begin
    a=4'd3; b=4'd5; op=3'b000;
    #10 op=3'b001;
    #10 op=3'b010;
    #10 op=3'b011;
```

localhost:8501

Deploy

Adhiraj's AI Verilog RTL + Testbench Generator

Enter your prompt/specifications:

Prompt

Generate a synthesizable Verilog module implementing an AXI4-Lite memory-mapped timer peripheral based on the specifications mentioned in the specs file attached.

Or upload a .txt spec file

Drag and drop file here
Limit 200MB per file • TXT

Browse files

axi_timer_specs.txt 3.4KB

Generate RTL & Testbench

localhost:8501

Deploy

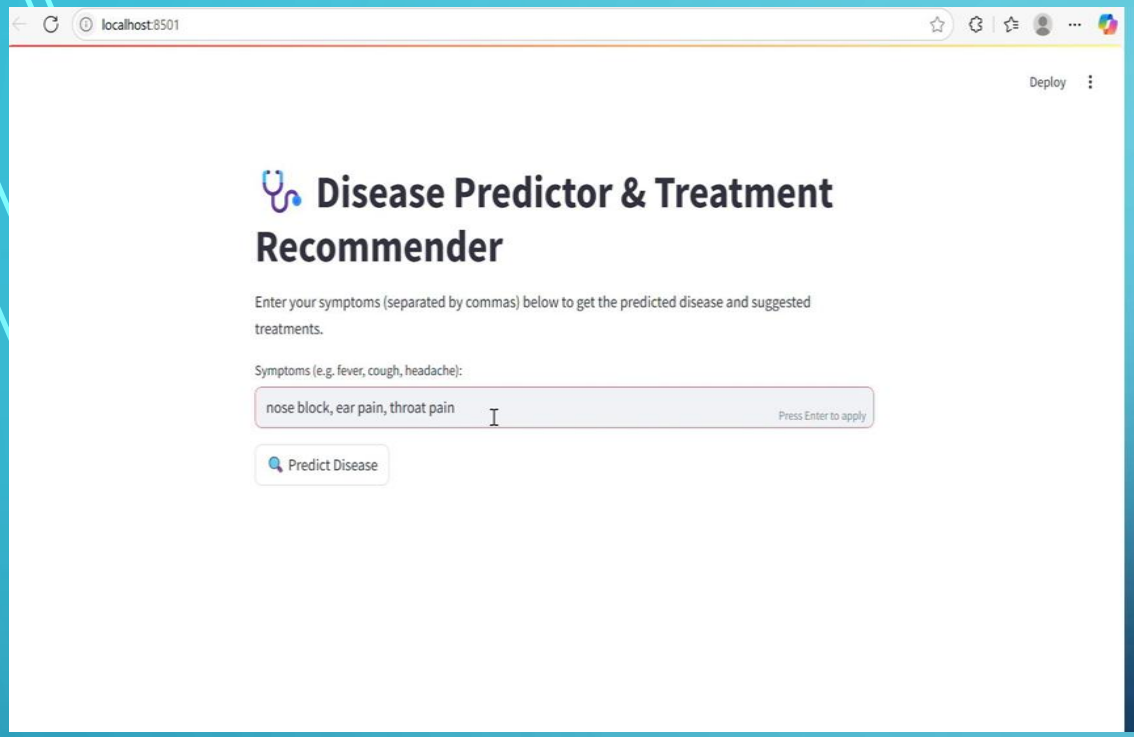
Generated RTL Code

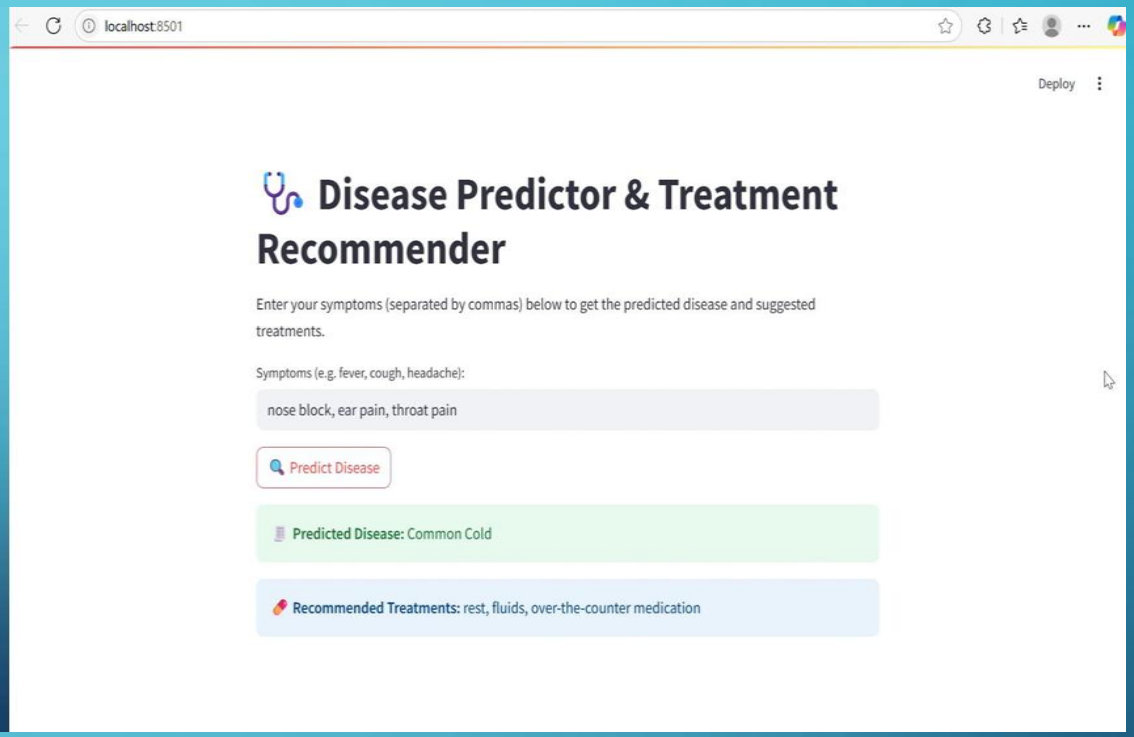
```
module axi_timer #(
    parameter ADDR_WIDTH=4,
    parameter DATA_WIDTH=32,
    parameter PRESCALER_WIDTH=16,
    parameter COUNT_WIDTH=32
)(
    input clk,
    input rst_n,
    output reg irq,
    input [ADDR_WIDTH-1:0] axi_awaddr,
    input axi_awvalid,
    output reg axi_awready,
    input [DATA_WIDTH-1:0] axi_wdata,
    input [3:0] axi_wstrb,
    input axi_wvalid,
    output reg axi_wready,
    output reg [1:0] axi_bresp,
    output reg axi_bvalid,
    input axi_bready,
    input [ADDR_WIDTH-1:0] axi_araddr,
    input axi_arvalid,
    output reg axi_arready,
    output reg [DATA_WIDTH-1:0] axi_rdata,
    output reg [1:0] axi_rresp,
    output reg axi_rvalid,
    input axi_rready
);
    // Simplified timer logic (pre-determined)
    reg [PRESCALER_WIDTH-1:0] prescaler;
    reg [COUNT_WIDTH-1:0] load,count,cmp;
    reg enable, auto_reload, irq_enable;
    reg irq_pending;

    always @(posedge clk or negedge rst_n) begin
        if(!rst_n) begin
            count<=0; prescaler<=0; irq<=0; irq_pending<=0;
        end else begin
            // simple tick increment
```

DISEASE PREDICTOR AND TREATMENT RECOMMENDER

- **Objective:**
  To develop an AI-based system that predicts the most probable disease from user-entered symptoms and recommends suitable treatments.
- **Methodology:**
- Collected and preprocessed a dataset containing diseases, symptoms, and treatments.
- Converted textual symptoms into numerical vectors using **CountVectorizer**.
- Trained a **Random Forest Classifier** to predict diseases and used **LabelEncoder** for mapping labels.
- Created a **treatment map** to link each disease with its respective treatments.
- Saved model components using **joblib** for efficient reuse during prediction.
- **Working Principle:**
  User enters symptoms → model vectorizes input → Random Forest predicts disease → system fetches treatment recommendations.
- **Outcome:**
  Accurately predicts diseases based on symptom patterns and provides treatment suggestions. Designed for easy future integration with a Flask-based web app.
- **Domain:** Machine Learning, Data Science | **Tools:** Python, Pandas, NumPy, Scikit-learn, Flask, VS Code


localhost:8501
Deploy
Disease Predictor & Treatment Recommender
Enter your symptoms (separated by commas) below to get the predicted disease and suggested treatments.
Symptoms (e.g. fever, cough, headache):
nose block, ear pain, throat pain
Press Enter to apply
Predict Disease


localhost:8501
Deploy
Disease Predictor & Treatment Recommender
Enter your symptoms (separated by commas) below to get the predicted disease and suggested treatments.
Symptoms (e.g. fever, cough, headache):
nose block, ear pain, throat pain
Predict Disease
Predicted Disease: Common Cold
Recommended Treatments: rest, fluids, over-the-counter medication


Upflairs
...because winning takes it all
CERTIFICATE
OF COMPLETION
Proudly Presented to
Adhiraj Mandal
B.Tech. IV Sem. | Department of Electronics and Communication Engineering
from Indian Institute of Engineering Science & Technology has successfully completed 60 Days Summer Internship & Training Program on Data Science with Machine Learning & AI from 1st May 2025 to 30th June 2025 organized by Upflairs Pvt. Ltd.
Siddharth Singh
Founder & Director
Upflairs Pvt. Ltd.
Enrollment No. :
UF/0525/6134
MSME
#startupindia
Ministry of MSME, Govt. of India

IMAGE RECOGNITION AND CLASSIFICATION USING DEEP LEARNING

- **Objective:**
  Developed a Convolutional Neural Network (CNN) model to classify images into predefined categories using the CIFAR-10 dataset.
- **Project Overview:**
- Implemented a deep learning model using TensorFlow and Keras for multiclass image recognition.
- Trained and validated the CNN on 60,000 images across 10 classes including cars, cats, dogs, airplanes, fruits, etc.
- Achieved **77% training accuracy** and **70% test accuracy**, optimizing using Adam optimizer and categorical cross-entropy loss.
- Integrated the trained model with a simple Streamlit app for real-time image prediction.
- **Key Learnings:**
- Hands-on experience with CNN architecture, model evaluation, and deployment workflow.
- Improved understanding of dataset preprocessing, feature extraction, and neural network tuning.
- **Tools & Technologies:**
  TensorFlow · Keras · Python · Google Colab · Streamlit

localhost:8501

Deploy

Image Classifier using Deep Learning

Upload an image, and the AI will predict what it is!

Choose an image...

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Browse files

ai_test.jpg 37.3KB

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

localhost:8501

utilize the `use_container_width` parameter instead.

Deploy

Uploaded Image

Prediction: **airplane** (1235.74% confidence)

CERTIFICATE

ISO 9001 2015 CERTIFIED

Corporate Identification Number
U80903DL2020NPL371984

19894000000177074

Ybi Foundation

This is to Certify that

ADHIRAJ MANDAL

has self motivation and willingness to learn, engage in day to day tasks and projects with high level of professionalism in

Data Science and Machine Learning

Internship Period: 2 Months, Completion on: Monday, Jun 30 2025

MINISTRY OF CORPORATE AFFAIRS
GOVERNMENT OF INDIA

ISO 9001Certified

Monday, Jun 30 2025
Dr. Alok Yadav
ADHIRAJ MANDAL

Issue Date:
Monday, Jun 30 2025

Scan QR Code
Check Credential Id for Certificate Verification : CJM3BDK39ABWP

www.ybifoundation.com (+91) 966 798 7711 support@ybifoundation.com

Self generated by intern

THANK YOU

ADHIRAJ MANDAL

B.TECH. ELECTRONICS AND TELECOMMUNICATION ENGINEERING

INDIAN INSTITUTE OF ENGINEERING SCIENCE AND TECHNOLOGY

(IIEST SHIBPUR)

For detailed project reports and project showcase, click on the link below

My_Projects_showcase_portfolio/Projects_showcase.pdf at main · Adhiraj-Mandal/My_Projects_showcase_portfolio