

# Automatic Currency Denomination Conversion System

<sup>1</sup>Chinababu Vanama

<sup>1</sup>Assistant professor

<sup>1</sup> Department of Electronics and Communication Engineering

<sup>1</sup>Rajiv Gandhi University of Knowledge and Technologies, Ongole Campus, A.P, India.

**Abstract**—The Automatic Currency Denomination Conversion System is designed to perform automatic conversion between currency notes and coins using Verilog HDL. The system helps in reducing manual effort and improves accuracy in currency handling applications such as vending machines, ticket counters, shopping malls, and banking systems. The proposed system supports two modes of operation namely Note-to-Coin conversion and Coin-to-Note conversion. In Note-to-Coin mode, the inserted note is converted into equivalent coin denominations. In Coin-to-Note mode, the inserted coins are collected, processed, and converted into the minimum number of currency notes using priority-based denomination logic. The complete system is designed and simulated using Xilinx ISE software. Digital components such as registers, counters, timers, and arithmetic logic are used for denomination calculation and output generation. The simulation results verify the correct functioning of the system under different operating conditions. The proposed system provides fast, accurate, and efficient denomination conversion with reduced human intervention. The project can be further extended using FPGA implementation, currency validation techniques, IoT monitoring, and digital payment integration for real-time automated financial applications.

**Index Terms**— Verilog HDL, Automatic Currency Denomination Conversion, Coin to Note, Note to Coin, Xilinx ISE, journals.

## I. INTRODUCTION

In many public and commercial places, currency transactions often become difficult due to the unavailability of exact change. Manual denomination handling increases waiting time, causes calculation errors, and reduces transaction efficiency, especially in crowded environments such as ticket counters, vending machines, shopping malls, and public transport systems. Traditional cash handling methods also face problems such as improper denomination distribution, dependency on human operators, delay in transaction processing, and difficulty in managing large numbers of currency exchanges during peak hours. In some cases, inaccurate denomination handling may also lead to customer inconvenience and interruption in financial transactions.

With the increasing demand for automation in financial systems, faster and more reliable denomination handling methods are required. Automated currency processing helps in reducing human effort, improving transaction accuracy, minimizing delays, and providing smooth real-time financial operations.

## II. APPLICATIONS

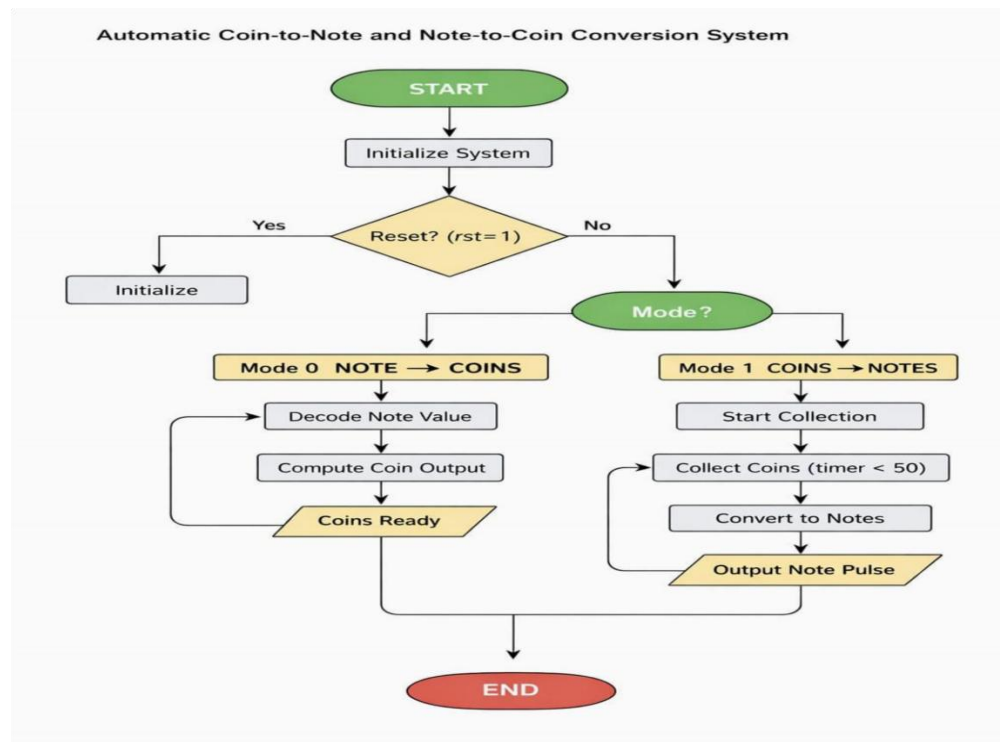
It is used in automated financial and commercial applications where quick and accurate denomination conversion is required. In many public places, people often face difficulty due to the unavailability of exact currency denominations. The conventional method of denomination exchange mainly depends on manual handling, which requires more time and may lead to calculation errors. By employing this project, the currency inserted by the user can be converted automatically into the required denomination with improved accuracy and reduced human effort. The system can be effectively used in vending machines, shopping malls, Ticket counters, railway stations, banking systems, and automated kiosks where fast denomination handling is necessary. This project also helps in improving transaction efficiency and reducing delays in crowded environments. The design is cost effective, reliable, and suitable for real time automated currency handling applications.



### III. OBJECTIVE

The main objective of this project is to develop an Automatic Currency Denomination Conversion System using Verilog HDL. In many public and commercial places, manual denomination handling requires more time and may lead to transaction delays and calculation errors. The denomination problems mainly occur due to the unavailability of exact change, improper denomination distribution, and dependency on manual cash handling systems. To overcome this problem, here is a project namely Automatic Currency Denomination Conversion System, used for automatic conversion between notes and coins. Using this project, currency denominations can be converted quickly and accurately with reduced human effort. This project is designed using Verilog HDL and simulated using Xilinx ISE software for efficient and reliable automated currency handling.

### IV. FLOW CHART



## V. SOFTWARE IMPLEMENTATION

Software used: Xilinx ISE

### Xilinx ISE:

The Xilinx Integrated Software Environment (ISE) is a software development tool used for the design, simulation, synthesis, and implementation of digital systems using Hardware Description Languages such as Verilog HDL and VHDL. The software is widely used for FPGA and VLSI- based digital circuit development applications. Xilinx ISE provides an efficient platform for designing and testing digital logic circuits before real-time hardware implementation. It supports behavioral simulation, waveform generation, timing analysis, synthesis, and debugging of digital systems. The software enables designers to verify the functionality and performance of the system accurately.

### Verilog Design and Simulation

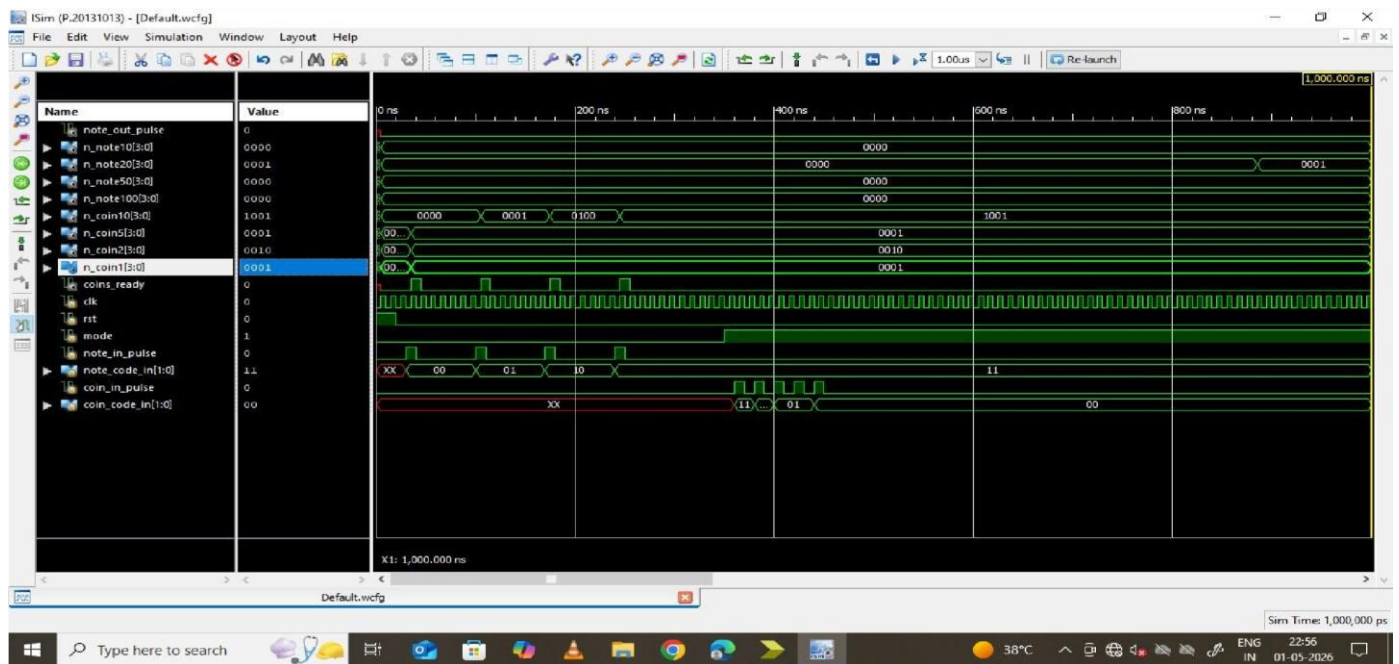
Xilinx ISE allows users to write, compile, simulate, and analyse Verilog HDL programs in an integrated environment. The simulation tools help in checking the correctness of registers, counters, timers, arithmetic logic, and control signals used in the project. The waveform analysis feature provides clear visualization of input and output operations during simulation. The software also supports RTL schematic generation and logic optimization for efficient hardware implementation. Due to its reliability and flexibility, Xilinx ISE is widely used in academic projects, FPGA development, and digital system design Applications.

### VERILOG HDL:

Verilog HDL is a Hardware Description Language used for modelling, designing, and implementing digital electronic systems. It is commonly used in FPGA and VLSI design for describing both the structural and behavioral operation of digital circuits. In this project, Verilog HDL is used to design the Automatic Currency Denomination Conversion System. The language helps in Implementing denomination calculation, timer-based control, value accumulation, mode selection, and output generation logic in a Simple and efficient manner.

Verilog HDL supports modular design methodology, which improves flexibility, scalability, and reusability of the system. It also Provides fast simulation and accurate verification of digital circuits before hardware implementation. Due to these advantages, Verilog HDL is widely used for modern digital system and FPGA-based application development

## VI. SIMULATION AND OUTPUT



**VII. SIMULATION CODE****Main Program**

```

`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 12:24:36 03/24/2026
// Design Name:
// Module Name: currency
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////
module notecurrency(
input clk,
input rst,
input mode,
// NOTE INPUT
input note_in_pulse,
input [1:0] note_code_in,
// COIN INPUT
input coin_in_pulse,
input [1:0] coin_code_in,
// NOTE OUTPUT
output reg note_out_pulse,
output reg [1:0] note_code_out,
// NOTE COUNTS
output reg [3:0] n_note10,
output reg [3:0] n_note20,
output reg [3:0] n_note50,
output reg [3:0] n_note100,
// COIN OUTPUT
output reg [3:0] n_coin10,
output reg [3:0] n_coin5,
output reg [3:0] n_coin2,
output reg [3:0] n_coin1,
output reg coins_ready
);
//-----
reg [7:0] value_reg;
reg [7:0] rem_val;
reg [7:0] timer;
reg collecting; // collecting coins phase
reg done; // conversion done flag
//-----
// NOTE DECODE
//-----
wire [7:0] note_val =
(note_code_in==2'b11)?100:
(note_code_in==2'b10)?50:
(note_code_in==2'b01)?20:10;
//-----

```

```

// COIN DECODE
//-----
wire [7:0] coin_val =
(coin_code_in==2'b11)?10:
(coin_code_in==2'b10)?5:
(coin_code_in==2'b01)?2:1;
//-----
always @(posedge clk)
begin
if(rst)
begin
value_reg <= 0;
rem_val  <= 0;
timer    <= 0;
collecting<= 0;
done      <= 0;
n_coin10 <= 0;
n_coin5  <= 0;
n_coin2  <= 0;
n_coin1  <= 0;
n_note10 <= 0;
n_note20 <= 0;
n_note50 <= 0;
n_note100 <= 0;
coins_ready  <= 0;
note_out_pulse <= 0;
end
else
begin

note_out_pulse <= 0;
coins_ready  <= 0;

//-----
// MODE 0 : NOTE → COINS (same)
//-----
if(mode==0 && note_in_pulse)
begin
rem_val = note_val;

n_coin10 = 0;
n_coin5  = 0;
n_coin2  = 0;
n_coin1  = 0;

if(rem_val >= 10)
begin
n_coin5 = 1;
n_coin2 = 2;
n_coin1 = 1;
rem_val = rem_val - 10;
end

n_coin10 = rem_val / 10;

coins_ready <= 1;
end

//-----
// MODE 1 : COINS → NOTES (FIXED)

```

```

//-----
else if(mode==1)
begin
//-----
// START COLLECTION
//-----
if(!collecting && !done)
begin
if(coin_in_pulse)
begin
collecting <= 1;
timer <= 0;
value_reg <= coin_val;
end
end

//-----
// COLLECTING PHASE
//-----
else if(collecting)
begin
if(timer < 50)
begin
timer <= timer + 1;

if(coin_in_pulse)
value_reg <= value_reg + coin_val;
end
else
begin
collecting <= 0;
done <= 1;
end
end

//-----
// CONVERSION PHASE
//-----
else if(done)
begin
rem_val = value_reg;

n_note100 <= rem_val / 100;
rem_val = rem_val % 100;

n_note50 <= rem_val / 50;
rem_val = rem_val % 50;

n_note20 <= rem_val / 20;
rem_val = rem_val % 20;

n_note10 <= rem_val / 10;
rem_val = rem_val % 10;

//-----
// OUTPUT PRIORITY
//-----
if(n_note100>0)
note_code_out <= 2'b11;

```

```

    else if(n_note50>0)
        note_code_out <= 2'b10;
    else if(n_note20>0)
        note_code_out <= 2'b01;
    else if(n_note10>0)
        note_code_out <= 2'b00;

    if(n_note10 || n_note20 || n_note50 || n_note100)
        note_out_pulse <= 1;

    //-----
    // RESET SYSTEM
    //-----
    value_reg <= 0;
    timer <= 0;
    done <= 0;
end
end
end
endmodule

```

### Test bench Program

```

`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 03/24/2026 11:56:13 PM
// Design Name:
// Module Name: Allcurrency_tb
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////
module Allcurrency_tb;
    reg clk;
    reg rst;
    reg mode;
    // NOTE INPUT
    reg note_in_pulse;
    reg [1:0] note_code_in;
    // COIN INPUT
    reg coin_in_pulse;
    reg [1:0] coin_code_in;
    // OUTPUTS
    wire note_out_pulse;
    wire [1:0] note_code_out;
    wire [3:0] n_note10;
    wire [3:0] n_note20;
    wire [3:0] n_note50;

```

```
wire [3:0] n_note100;
wire [3:0] n_coin10;
wire [3:0] n_coin5;
wire [3:0] n_coin2;
wire [3:0] n_coin1;
```

```
wire coins_ready;
```

```
//-----
// DUT
//-----
```

```
Allcurrency uut (
    .clk(clk),
    .rst(rst),
    .mode(mode),

    .note_in_pulse(note_in_pulse),
    .note_code_in(note_code_in),

    .coin_in_pulse(coin_in_pulse),
    .coin_code_in(coin_code_in),

    .note_out_pulse(note_out_pulse),
    .note_code_out(note_code_out),

    .n_note10(n_note10),
    .n_note20(n_note20),
    .n_note50(n_note50),
    .n_note100(n_note100),

    .n_coin10(n_coin10),
    .n_coin5(n_coin5),
    .n_coin2(n_coin2),
    .n_coin1(n_coin1),
    .coins_ready(coins_ready)
);
```

```
//-----
// CLOCK
//-----
```

```
always #5 clk = ~clk;
```

```
//-----
// MODE 0 OUTPUT DISPLAY
//-----
```

```
always @(posedge coins_ready)
begin
    $display("TIME=%0t ns [MODE0] Coins: 10=%d 5=%d 2=%d 1=%d",
        $time, n_coin10, n_coin5, n_coin2, n_coin1);
end
```

```
//-----
// MODE 1 OUTPUT DISPLAY
//-----
```

```
always @(posedge note_out_pulse)
begin
    $display("TIME=%0t ns [MODE1] Notes: 100=%d 50=%d 20=%d 10=%d",
        $time, n_note100, n_note50, n_note20, n_note10);
end
```

```

//-----
// TEST SEQUENCE
//-----
initial
begin
    clk = 0;
    rst = 1;
    mode = 0;

    note_in_pulse = 0;
    coin_in_pulse = 0;

    #20 rst = 0;

    //-----
    // ? MODE 0 : NOTE ? COINS
    //-----
    $display("==== MODE 0 : NOTE TO COINS =====");

    // ?10
    #10;
    note_code_in = 2'b00;
    note_in_pulse = 1;
    #10 note_in_pulse = 0;

    #50;

    // ?20
    #10;
    note_code_in = 2'b01;
    note_in_pulse = 1;
    #10 note_in_pulse = 0;

    #50;

    // ?50
    #10;
    note_code_in = 2'b10;
    note_in_pulse = 1;
    #10 note_in_pulse = 0;

    #50;

    // ?100
    #10;
    note_code_in = 2'b11;
    note_in_pulse = 1;
    #10 note_in_pulse = 0;

    #100;
    //-----
    // ? SWITCH TO MODE 1
    //-----
    $display("==== MODE 1 : COINS TO NOTES =====");

    mode = 1;
    //-----
    // ?20 INPUT
    //-----

```

```

#10 coin_code_in = 2'b11; coin_in_pulse = 1; #10 coin_in_pulse = 0; //10
#10 coin_code_in = 2'b10; coin_in_pulse = 1; #10 coin_in_pulse = 0; //5
#10 coin_code_in = 2'b01; coin_in_pulse = 1; #10 coin_in_pulse = 0; //2
#10 coin_code_in = 2'b01; coin_in_pulse = 1; #10 coin_in_pulse = 0; //2
#10 coin_code_in = 2'b00; coin_in_pulse = 1; #10 coin_in_pulse = 0; //1
#700;
//-----
// ?120 INPUT
//-----
repeat(12)
begin
    #10 coin_code_in = 2'b11;
    coin_in_pulse = 1;
    #10 coin_in_pulse = 0;
end
#700;
//-----
// ?50 INPUT
//-----
repeat(5)
begin
    #10 coin_code_in = 2'b11;
    coin_in_pulse = 1;
    #10 coin_in_pulse = 0;
end

#1000;

//-----
// END SIMULATION
//-----
end
endmodule

```

### VIII .Conclusion

Thus, the project on Automatic Currency Denomination Conversion System was successfully designed and simulated using Verilog HDL in Xilinx ISE. The system performs efficient bidirectional denomination conversion using digital processing and control logic.

The proposed design provides accurate denomination handling with reduced processing delay and reliable operation for automated currency transaction applications.

### REFERENCES

- [1]. E. Kho and M. Kumar, "Design and Implementation of FPGA based Vending Machine for Integrated Circuit", 2020 IEEE International Conference on Communication and Signal Processing.
  - [2]. M. H. Fuad, R. Yeassin, K. M. M. Hassan, "Design of a Vending Machine Using Verilog HDL and Implementation", European Journal of Electrical Engineering and Computer Science, 2023.
  - [3]. A. T. Saadoon, "Design and Implementation of Binary-To-Decimal Converter Using Verilog HDL", Journal of Engineering, 2013.
  - [4]. Xilinx Inc., "Xilinx Design Tools Documentation", Xilinx Official Documentation.
- "Verilog HDL Tutorials and FPGA Design Applications", IEEE Xplore Digital Library.