

**SpinSense:**

**Motor Speed Monitoring using Hall Effect Sensor**

**TEAM:**

**UNDER ACHIEVERS**

S.Mahammad Saleem - SSIT0035

C.V.Jaya Prakash - SSIT0041

P.Vasu - SSIT0042

**ABSTRACT**

The project we have done, **SpinSense:Motor Speed Monitoring using Hall Effect Sensor**,is a Verilog-based digital RPM measurement designed to be implementedon **FPGA(Edge Artix-7) board**. The main aim of this project is to constantly monitor the motor speed with the help of a **Hall Effect Sensor**, which could detect the existence of a magnetic field. The Hall Effect Sensor gives a **HIGH output**(output 1) when it **detects the magnet** attached to the rotating motor. these outputs,when generated, are then counted as ‘**pulses’**. The ‘pulses’ are counted per a second(a module is created for a 1-second timer) and used to calculate the **RPM** of the motor. This RPM is then displayed in a **4-digit 7-segment Display** on the FPGA board. In case the speed exceeds the threshold limit, a **buzzer** warning is triggered. The motor used here can be powered using an external power source or with the help of the FPGA board directly(prefer based on the power supply options we have).

1



**INTRODUCTION**

In modern electromechanical systems, accurate and real-time monitoring of motor performance is essential for ensuring efficiency, safety, and longevity. Traditional monitoring systems often rely on bulky and expensive equipment, making them impractical for small-scale or embedded applications. SpinSense is a compact and cost-effective solution designed to monitor the speed (RPM) and optionally the direction of rotation of a motor using a Hall effect sensor.

The Hall effect sensor is a non-contact magnetic sensor capable of detecting changes in magnetic fields. By strategically placing a magnet on a motor shaft and positioning a Hall sensor nearby, each rotation of the shaft generates a detectable pulse. By counting these pulses over time, SpinSense calculates the motor's RPM. With additional sensor configuration, it can also determine the direction of rotation.

This system is ideal for applications in robotics, electric vehicles, industrial automation, and smart maintenance systems. It enables users to gather essential performance data in real time, which can be used for diagnostics, predictive maintenance, and closed-loop control.

SpinSense combines simplicity with functionality, offering a lightweight solution that can be implemented on platforms like Arduino, Raspberry Pi, or FPGA boards such as the Edge Artix 7, making it suitable for both academic and industrial projects.

2



**COMPONENTS :**

1. FPGA board(Edge Artix-7)
2. Hall Effect Sensor
3. Magnet
4. HW Battery(9V)
5. DC/Gear Motor
6. Jumper Wires

**Edge Artix-7 FPGA Board**

The Edge Artix 7 is a high-performance, low-power Field Programmable Gate Array (FPGA) development board based on the Xilinx Artix-7 series. Designed for embedded system developers, students, and researchers, it offers a flexible platform for implementing custom digital logic, signal processing, and hardware-accelerated algorithms.



The board features a Xilinx XC7A35T or XC7A100T FPGA, which belongs to the Artix-7 family known for providing an excellent balance of performance, power efficiency, and logic density. With support for high-speed I/O, block RAM, DSP slices, and clock management tiles, the Edge Artix 7 is well-suited for applications in digital signal processing, motor control, communication systems, and real-time hardware interfacing.

**Hall Effect Sensor**

The Hall effect sensor is a magnetic field sensor that operates based on the Hall effect principle—discovered by Edwin Hall in 1879. When a current-carrying conductor or semiconductor is placed in a magnetic field perpendicular to the current, a voltage (known as Hall voltage) is generated perpendicular to both the current and the magnetic field. This voltage can be measured and used to detect the presence and strength of the magnetic field.



Hall effect sensors are widely used in modern electronics for non-contact switching, positioning, speed detection, and current sensing.

**Working Principle:**

A Hall effect sensor detects magnetic fields.



When a magnet is brought near the sensor, the magnetic field causes the sensor to produce a digital or analog voltage output.



3



**VERILOG SOURCE CODE:**

module top\_module(

input clk,

input reset,

input in,

output [3:0] an,

output [7:0] ca,

output buzzer

);

wire [15:0] pulse;

wire [31:0] rpm;

RPM\_measure rpm\_m (.clk(clk),.reset(reset),.in(in),.pulse(pulse),.rpm(rpm),

.buzzer(buzzer\_signal) );

assign buzzer=buzzer\_signal;

seven\_seg display\_m (.clk(clk),.rpm(rpm),.an(an),.ca(ca) ); endmodule

module timer(

input clk,

input reset,

output reg second

);

reg [31:0] count;

always @(posedge clk or posedge reset) begin if (reset) begin

count <= 0;

second <= 0;

end

else if (count == 50\_000\_000-1) begin

count <= 0;

second <= 1;

end

else begin

count <= count + 1;

second <= 0;

end

end

endmodule

4



module hall\_sensor(

input clk,

input reset,

input in,

output reg out);

reg prev;

always @(posedge clk or posedge reset) begin if (reset) begin

prev <= 0;

out <= 0;

end else begin

out <= in & ~prev;

prev <= in;

end

end

endmodule

module RPM\_measure(

input clk,

input reset,

input in,

output reg [15:0] pulse,

output reg [31:0] rpm,

output buzzer);

wire hall\_edge;

wire second;

reg buzzer\_reg=0;

hall\_sensor hs (.clk(clk),.reset(reset),.in(in),.out(hall\_edge) ); timer tm (.clk(clk),.reset(reset),.second(second));

always @(posedge clk or posedge reset) begin if (reset) begin

pulse <= 0;

rpm <= 0;

buzzer\_reg<=0;

end else begin

if (hall\_edge) begin

pulse <= pulse + 1;

if(pulse\*60>500)

buzzer\_reg<=1;

else

buzzer\_reg<=0;

end

5



if (second) begin

rpm <= pulse \* 60;

pulse <= 0;

end

end

end

endmodule

module seven\_seg(

input clk,

input [31:0] rpm,

output reg [3:0] an,

output reg [7:0] ca

);

reg clk\_seg = 0;

reg [31:0] count = 0;

reg [1:0] sel = 0;

always @(posedge clk) begin

if (count == 100\_000-1) begin

count <= 0;

clk\_seg <= ~clk\_seg;

end else

count <= count + 1;

end

reg [3:0] digit [3:0];

always @(posedge clk) begin

digit[3] <= rpm / 1000;

digit[2] <= (rpm % 1000) / 100;

digit[1] <= (rpm % 100) / 10;

digit[0] <= rpm % 10;

end

wire [7:0] seg [3:0];

display d0(.clk(clk), .value(digit[0]), .ca(seg[0])); display d1(.clk(clk), .value(digit[1]), .ca(seg[1])); display d2(.clk(clk), .value(digit[2]), .ca(seg[2])); display d3(.clk(clk), .value(digit[3]), .ca(seg[3]));

6



always @(posedge clk\_seg) begin

case (sel)

2'b00: begin an <= 4'b0001; ca <= seg[3]; end

2'b01: begin an <= 4'b0010; ca <= seg[2]; end

2'b10: begin an <= 4'b0100; ca <= seg[1]; end

2'b11: begin an <= 4'b1000; ca <= seg[0]; end endcase

sel <= sel + 1;

end

endmodule

module display(

input clk,

input [3:0] value,

output reg [7:0] ca

);

always @(posedge clk) begin

case (value)

4'd0: ca <= 8'b11000000;

4'd1: ca <= 8'b11111001;

4'd2: ca <= 8'b10100100;

4'd3: ca <= 8'b10110000;

4'd4: ca <= 8'b10011001;

4'd5: ca <= 8'b10010010;

4'd6: ca <= 8'b10000010;

4'd7: ca <= 8'b11111000;

4'd8: ca <= 8'b10000000;

4'd9: ca <= 8'b10010000;

endcase

end

endmodule

7



**Modules Explanantion:**

**1. top\_module**

Connects all submodules.



Takes clk, reset, and Hall sensor signal in.



Outputs:



an: Enable signals for 4-digit 7-segment display.



ca: Segment data for each digit.



buzzer: High if RPM exceeds the limit.



**2. hall\_sensor**

Detects rising edge from the Hall effect sensor.



Converts level-based signal to pulse-based edge signal (out), used to count rotations.



**3. timer**

Generates a 1-second pulse from a 50 MHz clock.



Used to time the measurement window for RPM calculations.



**4. RPM\_measure**

Counts the number of Hall pulses within 1 second.



Multiplies by 60 to get RPM (since 1 second of pulses × 60 = 1 minute equivalent).



Activates a buzzer if RPM exceeds 500.



Outputs:



pulse: pulse count within a second.



rpm: calculated RPM.



buzzer: logic high if RPM > 500.



**5. seven\_seg**

Splits the 4-digit RPM value into individual decimal digits.



Multiplexes the digits to drive a 4-digit common anode 7-segment display. Uses a fast internal clock for digit switching to simulate all digits being lit simultaneously.



**6. display**

Converts each 4-bit decimal digit into a 7-segment display code.



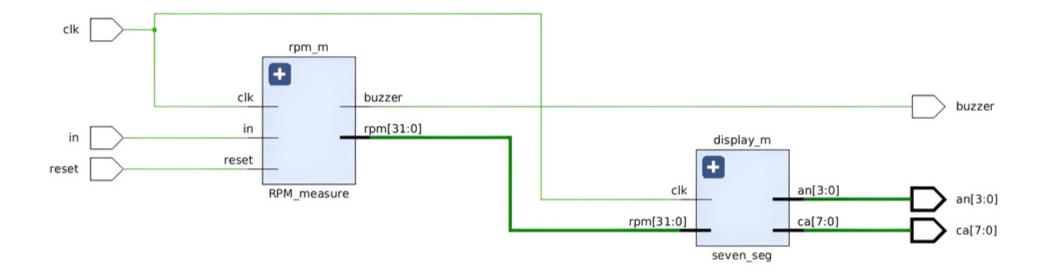
ca controls which segments are turned on to show numbers 0–9.



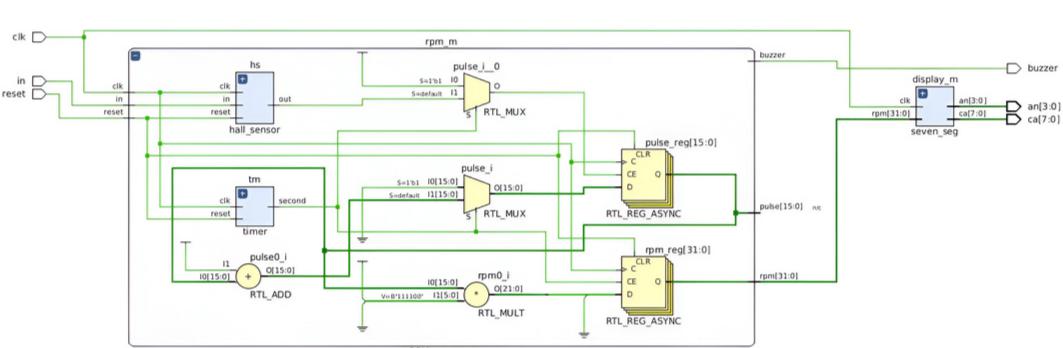
8



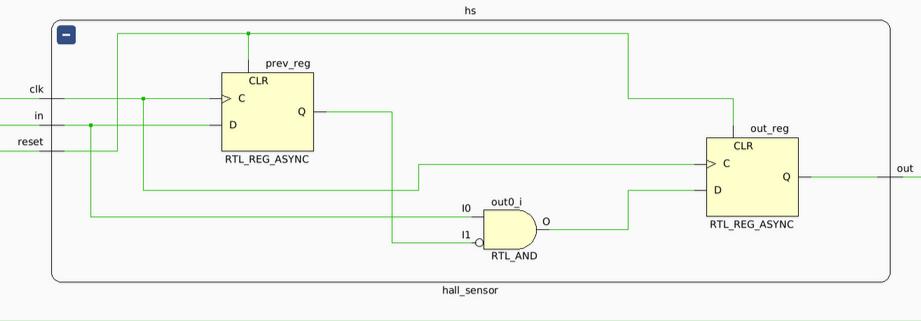
**Schematic Diagrams:**



Simple Schematic Diagram



Total Schematic Diagram



Hall Effect Sensor Schematic Diagram

9



**Simulation:**



Note: *The Outputs* ***an*** *and* ***ca*** *doesn’t show any value since the 7-segment display is assigned to show the RPM value which are at an interval of 1 second(50MHz) which cannot be shown but can be seen in hardware procedure*

**Timing Constraints:**

set\_property PACKAGE\_PIN **K12** [get\_ports buzzer]

set\_property IOSTANDARD **LVCMOS33** [get\_ports buzzer] set\_property IOSTANDARD **LVCMOS33** [get\_ports clk] set\_property IOSTANDARD **LVCMOS33** [get\_ports in] set\_property IOSTANDARD **LVCMOS33** [get\_ports reset] set\_property IOSTANDARD **LVCMOS33** [get\_ports {an[3]}] set\_property IOSTANDARD **LVCMOS33** [get\_ports {an[2]}] set\_property IOSTANDARD **LVCMOS33** [get\_ports {an[1]}] set\_property IOSTANDARD **LVCMOS33** [get\_ports {an[0]}] set\_property IOSTANDARD **LVCMOS33** [get\_ports {ca[7]}] set\_property IOSTANDARD **LVCMOS33** [get\_ports {ca[6]}] set\_property IOSTANDARD **LVCMOS33** [get\_ports {ca[5]}] set\_property IOSTANDARD **LVCMOS33** [get\_ports {ca[4]}] set\_property IOSTANDARD **LVCMOS33** [get\_ports {ca[3]}] set\_property IOSTANDARD **LVCMOS33** [get\_ports {ca[2]}] set\_property IOSTANDARD **LVCMOS33** [get\_ports {ca[1]}] set\_property IOSTANDARD **LVCMOS33** [get\_ports {ca[0]}] set\_property PACKAGE\_PIN **N11** [get\_ports clk] set\_property PACKAGE\_PIN **T15** [get\_ports in] set\_property PACKAGE\_PIN **L5** [get\_ports reset] set\_property PACKAGE\_PIN **G4** [get\_ports {an[3]}] set\_property PACKAGE\_PIN **G5** [get\_ports {an[2]}] set\_property PACKAGE\_PIN **E1** [get\_ports {an[1]}] set\_property PACKAGE\_PIN **F2** [get\_ports {an[0]}] set\_property PACKAGE\_PIN **H1** [get\_ports {ca[7]}] set\_property PACKAGE\_PIN **H2** [get\_ports {ca[6]}] set\_property PACKAGE\_PIN **J4** [get\_ports {ca[5]}] set\_property PACKAGE\_PIN **J5** [get\_ports {ca[4]}] set\_property PACKAGE\_PIN **H4** [get\_ports {ca[3]}] set\_property PACKAGE\_PIN **H5** [get\_ports {ca[2]}] set\_property PACKAGE\_PIN **G1** [get\_ports {ca[1]}] set\_property PACKAGE\_PIN **G2** [get\_ports {ca[0]}]

10



**Applications:**

Real-time monitoring of motor speed in conveyor belts, pumps, and industrial machinery.



Monitoring wheel or motor RPM to estimate vehicle speed.



Smart alerts (e.g., buzzer) if speed exceeds safe operating limits.



RPM measurement in small-scale wind turbines to track generation efficiency.

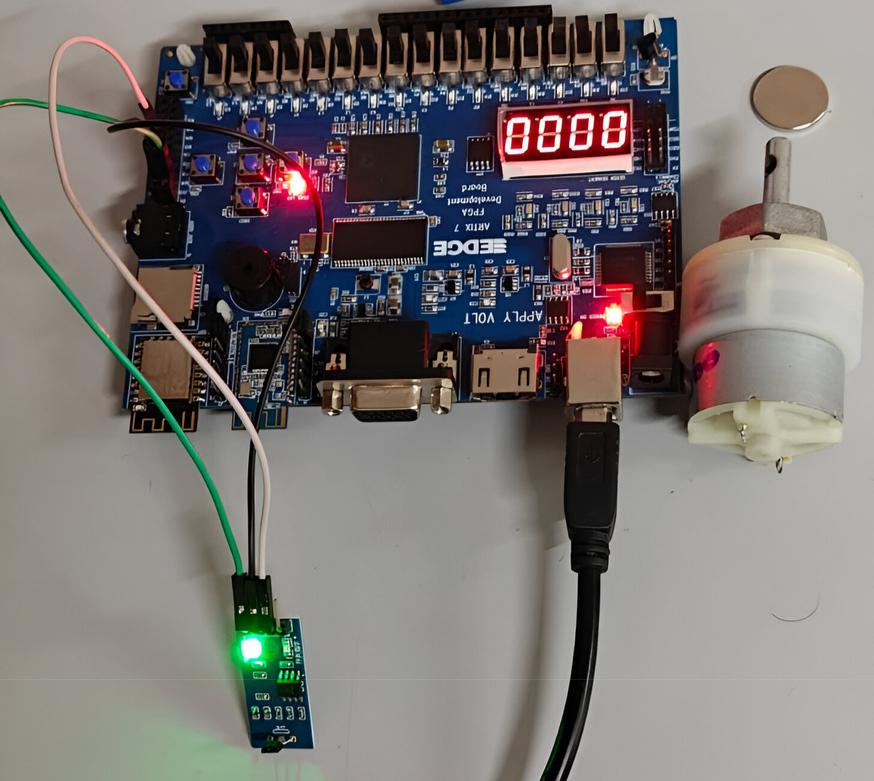


**Challenges Faced:**

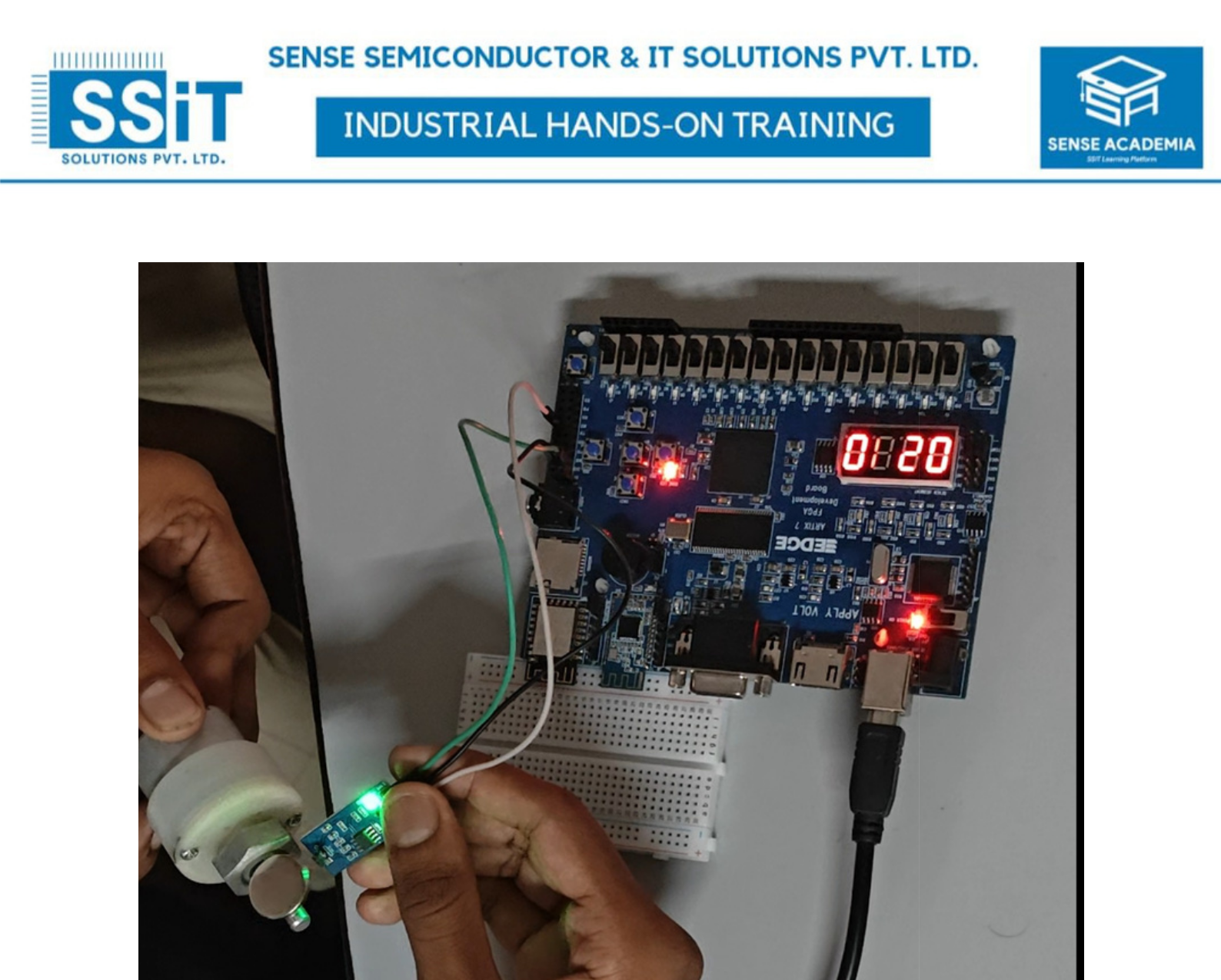
Integrating a DC Motor to the FPGA board seems quiet challenging to us which is why we decided to use an external source powered motor for the detection. There were multiple constraints errors but we managed to solved them. Hall Effect Sensor is a new thing to us so it took awhile to know about it.



**CIRCUIT DIAGRAM AND CONNECTIONS:**



11



**Conclusion:**

The SpinSense project successfully demonstrates a real-time motor monitoring system using a Hall effect sensor interfaced with an FPGA (Edge Artix 7). The system accurately detects rotational pulses, calculates the RPM (revolutions per minute), and displays the value on a 7-segment display. Additionally, a buzzer alert is triggered if the RPM exceeds a predefined threshold, ensuring immediate response to potential over speed conditions.

Overall, SpinSense demonstrates the practical application of digital design and sensor interfacing in real-world electromechanical systems, showcasing the power and flexibility of FPGAs in modern embedded systems.

The experience we gained during the process of making this project is very essential and can be utilised for future projects.

12