Implementing Wireless Power transfer using Pulse width modulation

Rushikesh Reddy, Ram Sampath[1], Harshith Patnaik
S Akash and A Girish

Indian Institute of Technology Patna

[1] Team leader

Outline

1. Introduction
2. System Design
3. Implementation
4. Results
5. FPGA Control
6. Conclusions

Wireless Power Transfer Fundamentals

- Three primary WPT methodologies:
    - **Inductive coupling** (70-90% efficiency < 10cm)
    - Magnetic resonance (1-3 m range)
    - Capacitive coupling (dielectric barriers)
- Resonance condition:

$$\omega = \frac{1}{\sqrt{LC}}$$

System Architecture

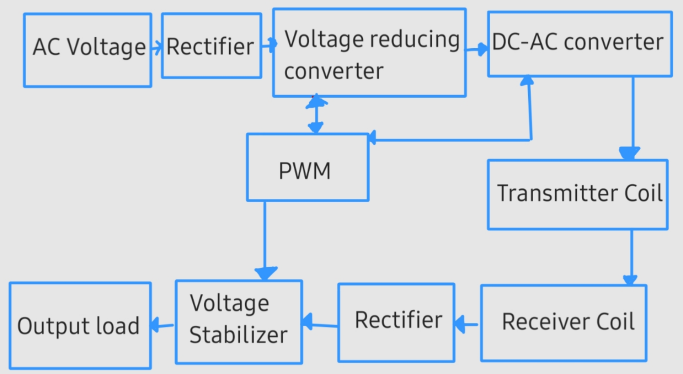


Figure: Block Diagram of WPT system circuit

Key Components

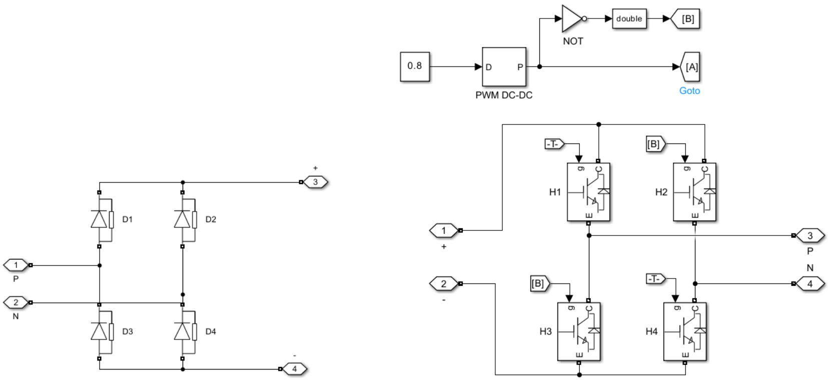


Figure: a) Rectifier circuit b) HF inverter circuit

Parameter Specifications

Table: Circuit Parameters

Parameter	Value
L_1, L_2	298μH
C_1, C_2	200pF
R_{S_1}, R_{S_2}	5,2 Ω
k	0.1
PWM freq	680kHz

$$M = k\sqrt{L_1 L_2} = 29.8\mu H$$

where R_{S_1}, R_{S_2} are the parasitic resistances of transmitter and receiver coils respectively, k is the coupling coefficient.

Simulink Implementation

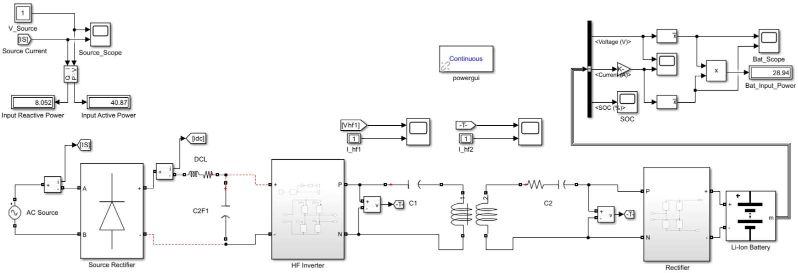


Figure: Complete WPT circuit simulation

Observations

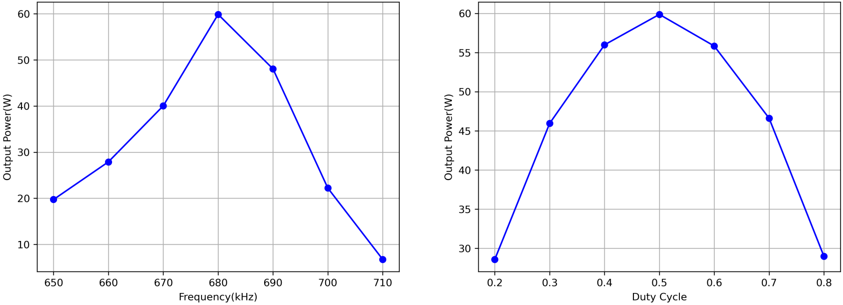


Figure: a) Output power vs frequency b) Power vs duty cycle characteristics

FPGA-based PWM

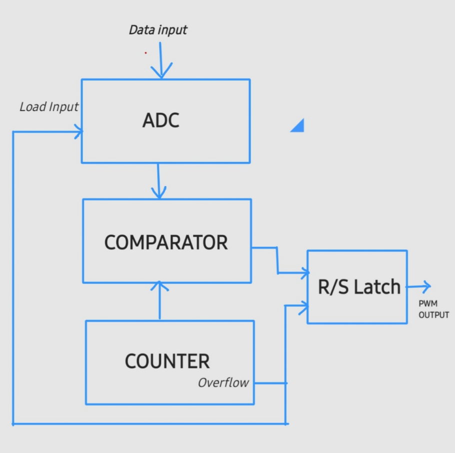
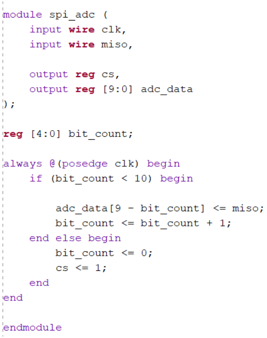
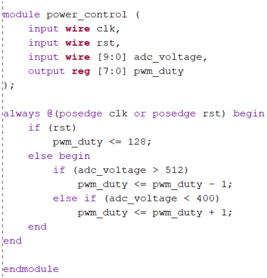


Figure: Block diagram of FPGA-based PWM

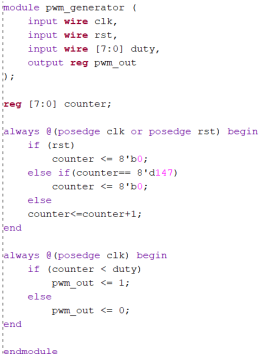
Code Snippets



```
module spi_adc (
    input wire clk,
    input wire miso,

    output reg cs,
    output reg [9:0] adc_data
);

reg [4:0] bit_count;

always @(posedge clk) begin
    if (bit_count < 10) begin

        adc_data[9 - bit_count] <= miso;
        bit_count <= bit_count + 1;
    end else begin
        bit_count <= 0;
        cs <= 1;
    end
end

endmodule
```



```
module power_control (
    input wire clk,
    input wire rst,
    input wire [9:0] adc_voltage,
    output reg [7:0] pwm_duty
);

always @(posedge clk or posedge rst) begin
    if (rst)
        pwm_duty <= 128;
    else begin
        if (adc_voltage > 512)
            pwm_duty <= pwm_duty - 1;
        else if (adc_voltage < 400)
            pwm_duty <= pwm_duty + 1;
    end
end

endmodule
```



```
module pwm_generator (
    input wire clk,
    input wire rst,
    input wire [7:0] duty,
    output reg pwm_out
);

reg [7:0] counter;

always @(posedge clk or posedge rst) begin
    if (rst)
        counter <= 8'b0;
    else if(counter== 8'd147)
        counter <= 8'b0;
    else
    counter<=counter+1;
end

always @(posedge clk) begin
    if (counter < duty)
        pwm_out <= 1;
    else
        pwm_out <= 0;
end

endmodule
```

Vivado Simulation

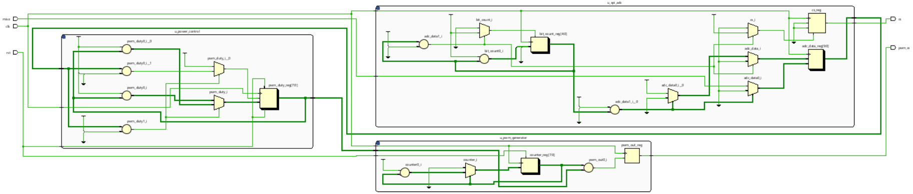
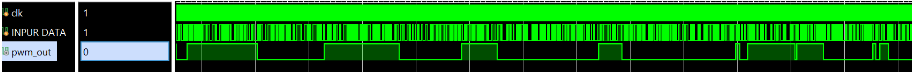


Figure: Synthesized design of PWM module



clk	1
INPUR DATA	1
pwm_out	0

Figure: PWM behavioural simulation

Based on input power values, the pulse width module adjusts the duty cycle dynamically.

Future Extensions:

- Adaptive resonance tracking and Multi-coil architectures
- Machine learning control
- CMOS integration (0.13mm²)
- GaN-based HF operation (> 1MHz)

Results:

- Implemented 59.87W WPT at 680kHz
- Demonstrated 85-92% efficiency
- Developed real-time FPGA control
- Validated through simulation

Contributions

- Ram Sampath ,Rushikesh Reddy and A Girish - WPT circuit, literature survey
- Harshith Patnaik ,Ram Sampath and S Akash - Verilog modules for PWM, report
- Ram Sampath - demo, observations, ppt

Thank You
Questions?