按下按键就会通过SIM900A发送”sos“短信并附上一小时以内的运动轨迹以及目前位置

要实现这个功能，您需要使用带有GPS模块和SIM900A模块的FPGA板来实现。建议使用Altera Cyclone IV系列FPGA，GPS模块和SIM900A模块需要进行驱动和编程，以确保它们的正常工作。

程序可以检测按下的按钮并通过SIM900A发送"SOS"短信，并将最近一小时内的运动轨迹和当前位置包括在短信中。

module sos\_send (

input clk, // 输入时钟信号

input reset\_n, // 复位信号

input button, // 按钮

output reg led, // LED指示灯

output reg sim\_tx // SIM900A 发送数据序列

);

// 定义常量和参数

parameter BAUD\_RATE = 9600;

localparam GPS\_DATA\_LENGTH = 20;

localparam SMS\_MAX\_LENGTH = 160;

localparam SOS\_TEXT = "SOS - HELP NEEDED! \n";

localparam ERROR\_MSG = "ERROR: UNABLE TO RETRIEVE GPS DATA \n";

// 定义寄存器和变量

reg [31:0] timer = 0; // 按钮计时器

reg [31:0] gps\_timer = 0; // GPS 模块计时器

reg [31:0] gps\_div; // GPS 模块波特率调整

reg [7:0] buffer[GPS\_DATA\_LENGTH-1:0]; // GPS 数据缓存

reg [7:0] sms\_buffer[SMS\_MAX\_LENGTH-1:0]; // 短信缓存

reg [7:0] sms\_counter = 0; // 短信计数器

reg [7:0] buffer\_count = 0; // GPS 数据计数器

reg [1:0] state = 0; // 状态机，用来保存当前状态6

reg gps\_ready = 0; // GPS 连接状态

reg sms\_sent = 0; // 短信发送状态

reg gps\_read = 0; // GPS 数据读取状态

reg can\_send = 0; // 是否可以发送短信的状态

reg tx\_ready = 0; // SIM900A 发送缓冲区状态

// 状态机

always @(posedge clk, negedge reset\_n) begin

if (~reset\_n) begin

state <= 0;

gps\_timer <= 0;

gps\_read <= 0;

buffer\_count <= 0;

sms\_counter <= 0;

timer <= 0;

sms\_sent <= 0;

gps\_ready <= 0;

can\_send <= 0;

end else begin

case (state)

0: begin // 空闲状态

timer <= 0;

led <= 0;

if(button == 0) begin

state <= 1;

timer <= 0;

end

end

1: begin // 按钮按下状态

timer <= timer + 1;

led <= 1;

if(timer >= 10000000) begin

state <= 0;

end

if(button == 1) begin

state <= 0;

end

end

2: begin // 检查GPS状态状态

gps\_timer <= gps\_timer + 1;

if(gps\_timer >= 1000000) begin

gps\_ready <= 1;

sms\_counter <= 0;

state <= 3;

end

end

3: begin // 读取GPS数据状态

if(~gps\_read) begin

buffer[buffer\_count] <= SIM900A\_DATA\_IN;

buffer\_count <= buffer\_count + 1;

if(buffer[buffer\_count-1] == 0x0A && buffer[buffer\_count-2] == 0x0D) begin

gps\_ready <= 0;

gps\_read <= 1;

buffer\_count <= 0;

sms\_counter <= 0;

state <= 4;

end

end

end

4: begin // 拼接短信并预处理状态

sms\_buffer <= SOS\_TEXT;

sms\_counter <= strlen(SOS\_TEXT);

if(~gps\_ready) begin

state <= 5;

end else begin

for(i=0; i<buffer\_count; i=i+1) begin

sms\_buffer[sms\_counter+i] <= buffer[i];

end

sms\_counter <= sms\_counter + buffer\_count;

buffer\_count <= 0;

if(sms\_counter + 14 > SMS\_MAX\_LENGTH) begin

state <= 5;

end else begin

state <= 2;

end

end

end

5: begin // 发送错误信息状态

sms\_buffer <= ERROR\_MSG;

sms\_counter <= strlen(ERROR\_MSG);

can\_send <= 1;

gps\_read <= 0;

state <= 6;

end

6: begin // 等待SIM900A发送缓存区准备状态

if(tx\_ready) begin

if(can\_send) begin

sms\_sent <= 1;

for(i=0; i<strlen(sms\_buffer); i=i+1) begin

sim\_tx <= sms\_buffer[i];

end

sim\_tx <= 0x1A; // 发送结束

end

state <= 0;

end

end

endcase

end

end

// 波特率调整

always @(posedge clk, negedge reset) begin

if(~reset) begin

gps\_div <= 0;

end else begin

if(gps\_div >= clk/BAUD\_RATE) begin

gps\_div <= 0;

gps\_read <= 1;

end else begin

gps\_div <= gps\_div + 1;

gps\_read <= 0;

end

end

end

// SIM900A 发送缓存区状态检查

always @(posedge clk, negedge reset) begin

if(~reset) begin

tx\_ready <= 0;

end else begin

if(sim\_tx == 0) begin

tx\_ready <= 1;

end else begin

tx\_ready <= 0;

end

end

end

endmodule

