

Verilog 串口接收范例程序

给你一个 verilog 的串口接收例子

```
////////////////////////////////////
//                               //
//                               //
////////////////////////////////////
//                               //
//                               //
//                               //
//                               //
//                               //
////////////////////////////////////
module UartRec(
    rst,                //异步复位
    rxd,                //串行数据
    baudClk8x,         //8X 波特率时钟
    irq,                //中断请求
    dataOut             //数据输出
);
input    rst,rxd,baudClk8x;
output   irq;
output[7:0] dataOut;
parameter idle_state=0, verifyStartBit_state=1,
           startBitOk_state=2, bit0_state=3,
           bit1_state=4, bit2_state=5,
           bit3_state=6, bit4_state=7,
           bit5_state=8, bit6_state=9,
           bit7_state=10;
reg [3:0] bitState; //接收状态机
reg [7:0] shifter; //移位寄存器
reg [7:0] dataOut; //输出锁存器
reg [2:0] bitCounter; //采样计数器
reg      irq;

always @(posedge rst or posedge baudClk8x)
begin
    if(rst)
        begin
            bitState<=4'd0;//idle_state;
            bitCounter<=3'd0;
            shifter<=8'd0;
        end
    else
        begin
            if(bitState==idle_state)
                begin
                    if(rxd)
                        bitCounter<=0;
                    else
                        bitCounter<=bitCounter+1;
                    if(bitCounter==verifyStartBit_state)
                        bitState<=startBitOk_state;
                    if(bitCounter==bit7_state)
                        dataOut<=shifter;
                end
            else if(bitState==startBitOk_state)
                begin
                    shifter<=shifter<<1|rxd;
                    bitState<=bit0_state;
                end
            else if(bitState==bit0_state)
                begin
                    shifter<=shifter<<1|rxd;
                    bitState<=bit1_state;
                end
            else if(bitState==bit1_state)
                begin
                    shifter<=shifter<<1|rxd;
                    bitState<=bit2_state;
                end
            else if(bitState==bit2_state)
                begin
                    shifter<=shifter<<1|rxd;
                    bitState<=bit3_state;
                end
            else if(bitState==bit3_state)
                begin
                    shifter<=shifter<<1|rxd;
                    bitState<=bit4_state;
                end
            else if(bitState==bit4_state)
                begin
                    shifter<=shifter<<1|rxd;
                    bitState<=bit5_state;
                end
            else if(bitState==bit5_state)
                begin
                    shifter<=shifter<<1|rxd;
                    bitState<=bit6_state;
                end
            else if(bitState==bit6_state)
                begin
                    shifter<=shifter<<1|rxd;
                    bitState<=bit7_state;
                end
            else if(bitState==bit7_state)
                begin
                    shifter<=shifter<<1|rxd;
                    bitState<=idle_state;
                end
        end
    end
end
```

```

end
else
begin
if(bitState==4'd0)//idle_state
begin
if(!rxd)
begin
bitState<=bitState+1;
bitCounter<=3'd0;
end
end
else if(bitState==4'd1)//verifyStartBit_state:
begin
if(bitCounter==3'd3)
begin
if(!rxd) //校验起始位是否依然有效
bitState<=bitState+1;//4'd2 bit 0 state
else
bitState<=4'd0; //idle_state;

bitCounter<=3'd0;
end
else
bitCounter<=bitCounter+1;
end
else if(bitState==4'd10)//bit7_state
begin
if(bitCounter==3'd7) //返回初始状态,不检查停止位
begin
bitCounter<=3'd0;
bitState<=4'd0;
end
else
bitCounter<=bitCounter+1;
end

else //bitState=2~9 8 个数据位接收
begin
if(bitCounter==3'd7)
begin
shifter<={rxd,shifter[7:1]};
bitCounter<=3'd0;
bitState<=bitState+1;
end
end

```

```
        else
            bitCounter<=bitCounter+1;
        end
    end
end

always @(posedge rst or negedge baudClk8x)
begin
    if(rst)
        begin
            dataOut<=0;
            irq<=0;
        end
    else if(bitState==4'd10)
        begin
            dataOut<=shifter;
            irq<=1;
        end
    else
        irq<=0;
    end
end
endmodule
```