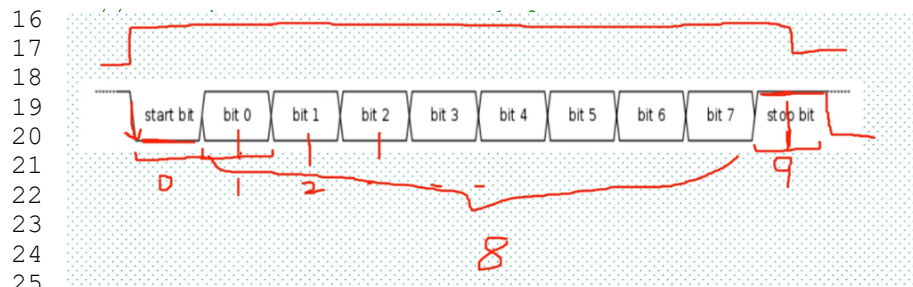


```

1 //*****Copyright
2 (c)*****//
3 //技术支持: www.openedv.com
4 //淘宝店铺: http://openedv.taobao.com
5 //关注微信公众平台微信号: "正点原子", 免费获取FPGA & STM32资料。
6 //版权所有, 盗版必究。
7 //Copyright(C) 正点原子 2018-2028
8 //All rights reserved
9 //-----
10 // File name:          uart_recv
11 // Last modified Date: 2018/4/19 19:56:36
12 // Last Version:       V1.1
13 // Descriptions:       uart串口接收模块
14 //-----
15 // Created by:         正点原子
16 // Created date:       2018/1/29 10:55:56

```

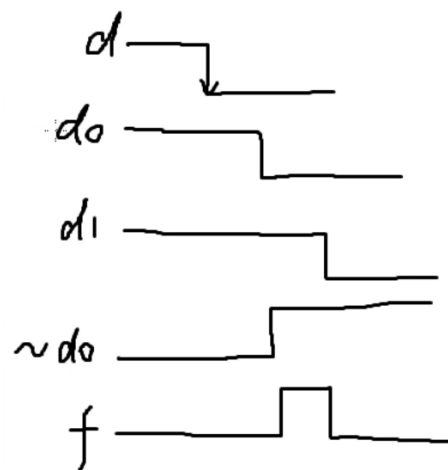


这幅图是rx\_cnt具体含义, 每帧数据计数从0-9变化

```

26 //*****
27 /
28 module uart_recv(
29     input          sys_clk,          //系统时钟
30     input          sys_rst_n,        //系统复位, 低电平有效
31
32     input          uart_rxd,          //UART接收端口
33     output reg     uart_done,         //接收一帧数据完成标志信号
34     output reg [7:0] uart_data       //接收的数据
35 );
36
37 //parameter define
38 parameter CLK_FREQ = 50000000;      //系统时钟频率
39 parameter UART_BPS = 9600;          //串口波特率
40 localparam BPS_CNT = CLK_FREQ/UART_BPS; //为得到指定波特率,
41                                         //需要对系统时钟计数BPS_CNT次
42
43 //reg define
44 reg [15:0] clk_cnt;                 50 000 000/115200 = 434.02778
45 reg [3:0] rx_cnt;                   400多个时钟周期完全不用担心
46 reg rx_flag;                        //系统时钟计数器
47 reg [7:0] rxdata;                   //接收数据计数器
48                                     //接收过程标志信号
49                                     //接收数据寄存器
50
51 //wire define
52 wire start_flag;
53
54 //*****
55 /**          main code
56 //*****
57 //捕获接收端口下降沿(起始位), 得到一个时钟周期的脉冲信号
58 assign start_flag = uart_rxd_d1 & (~uart_rxd_d0);
59
60 //对UART接收端口的数据延迟两个时钟周期
61 always @(posedge sys_clk or negedge sys_rst_n) begin
62     if (!sys_rst_n) begin
63         uart_rxd_d0 <= 1'b0;
64         uart_rxd_d1 <= 1'b0;
65     end
66     else begin
67         uart_rxd_d0 <= uart_rxd;
68         uart_rxd_d1 <= uart_rxd_d0;
69     end
70 end

```



为什么不用担心错位呢? 因为是边沿检测, 作为起始, 只相差两个时钟周期而已, 后面又取中间时刻

```
71 //当脉冲信号start_flag到达时, 进入接收过程
72 always @(posedge sys_clk or negedge sys_rst_n) begin
73     if (!sys_rst_n)
74         rx_flag <= 1'b0;
75     else begin
76         if(start_flag) //检测到起始位
77             rx_flag <= 1'b1; //进入接收过程, 标志位rx_flag拉高
78         else if((rx_cnt == 4'd9) && (clk_cnt == BPS_CNT/2))
79             rx_flag <= 1'b0; //计数到停止位中间时, 停止接收过程
80         else
81             rx_flag <= rx_flag;
82     end
83 end
84
85 //进入接收过程后, 启动系统时钟计数器与接收数据计数器
86 always @(posedge sys_clk or negedge sys_rst_n) begin
87     if (!sys_rst_n) begin
88         clk_cnt <= 16'd0;
89         rx_cnt <= 4'd0;
90     end
91     else if ( rx_flag ) begin //处于接收过程
92         if (clk_cnt < BPS_CNT - 1) begin
93             clk_cnt <= clk_cnt + 1'b1;
94             rx_cnt <= rx_cnt;
95         end
96         else begin
97             clk_cnt <= 16'd0; //对系统时钟计数达一个波特率周期后清零
98             rx_cnt <= rx_cnt + 1'b1; //此时接收数据计数器加1
99         end
100     end
101     else begin //接收过程结束, 计数器清零
102         clk_cnt <= 16'd0;
103         rx_cnt <= 4'd0;
104     end
105 end
106
107 //根据接收数据计数器来寄存uart接收端口数据
108 always @(posedge sys_clk or negedge sys_rst_n) begin
109     if ( !sys_rst_n)
110         rxdata <= 8'd0;
111     else if(rx_flag) //系统处于接收过程
112         if (clk_cnt == BPS_CNT/2) begin //判断系统时钟计数器计数到数据位中间
113             case ( rx_cnt )
114                 4'd1 : rxdata[0] <= uart_rxd_d1; //寄存数据位最低位
115                 4'd2 : rxdata[1] <= uart_rxd_d1;
116                 4'd3 : rxdata[2] <= uart_rxd_d1;
117                 4'd4 : rxdata[3] <= uart_rxd_d1;
118                 4'd5 : rxdata[4] <= uart_rxd_d1;
119                 4'd6 : rxdata[5] <= uart_rxd_d1;
120                 4'd7 : rxdata[6] <= uart_rxd_d1;
121                 4'd8 : rxdata[7] <= uart_rxd_d1; //寄存数据位最高位
122             default::;
123             endcase
124         end
125         else //根据位计数写入对应的寄存器就好了
126             rxdata <= rxdata;
127     else
128         rxdata <= 8'd0;
129 end
130
131 //数据接收完毕后给出标志信号并寄存输出接收到的数据
132 always @(posedge sys_clk or negedge sys_rst_n) begin
133     if (!sys_rst_n) begin
134         uart_data <= 8'd0;
135         uart_done <= 1'b0;
136     end
137     else if(rx_cnt == 4'd9) begin //接收数据计数器计数到停止位时
138         uart_data <= rxdata; //寄存输出接收到的数据
139         uart_done <= 1'b1; //并将接收完成标志位拉高
140     end
141     else begin
142         uart_data <= 8'd0;
```

```
143         uart_done <= 1'b0;  
144     end  
145 end  
146  
147 endmodule
```