

实验四 PS/2

PS/2 是个人计算机串行 I/O 端口的一种标准，名称的由来是因为它在 IBM PS/2(Personal System/2)机器上首次使用，连接 PS/2 键盘和 PS/2 鼠标。PS/2 端口的连接器的名称是 Mini-DIN，它有 6 个针/孔(一边是针、一边是孔)，其中的两个未被使用，其余 4 个分别是时钟、数据、VCC 和 GND。有些计算机或 FPGA 板只有一个连接器，但把两个未被使用的孔用来提供第二套的时钟和数据。这样，在外面使用一个 Y 形的分叉器，就能同时连接键盘和鼠标了。

1 PS/2 键盘

当你按下一个一般的键，键盘侧送出相应键的扫描码;松开时，送出 F0 接着又是扫描码。前者称为 Make Code，后者称为 Break Code。不一般的键，也称扩展键在二者最前面又加上 E0。

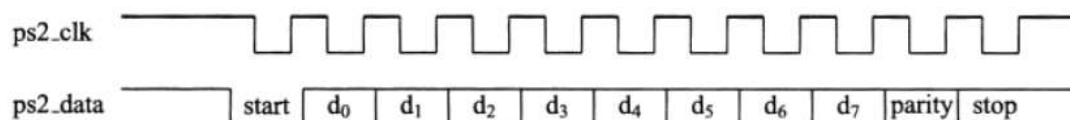
比如，如果你按下“L”键再放开，则送出 4BF04B(扫描码是 4B)。

再比如，如果你按下“Delete”键再放开，则送出 E071E0F071(扩展键)。

如果你长时间按下“L”键再放开，则送出 4B4B. . . 4B F0 4B。

多个键可以同时被按下，比如先按左“Shift”(扫描码为 12)、再按“L”、放开“L”、再放开“Shift”，则送出 12 4B F0 4B F0 12。

以上的数字均是十六进制的。键盘送给主机的数据以字节为单位。发送时以串行方式进行，与 UART 的格式类似，但有同步的时钟信号，另外规定使用奇校验时序见下图，时钟高电平时键盘开始送数据，时钟低电平时主机开始读数据。



以下的代码只负责接收由键盘送来的数据，到底是什么键交由软件处理。我们专门预备了一个 8 字节的缓冲区，以防数据丢失。首先检测时钟的下降沿，然后开始逐位接收数据并放入缓冲区。缓冲区实际上是一个先进先出的队列，配备有写指针和读指针。当队列不空时，送出 ready 信号;当队列溢出时，送出 overdow 信号。

```

module ps2_keyboard (clk, clrn, ps2_clk, ps2_data,
                    rdn, data, ready, overflow, count);
    input clk, clrn, ps2_clk, ps2_data;
    input rdn;
    output [7:0] data;
    output ready;
    output overflow;
    output [3:0] count; // internal signal, for test
    reg overflow;        // fifo overflow
    reg [3:0] count;      // count ps2_data bits
    reg [9:0] buffer;     // ps2_data bits
    reg [7:0] fifo[7:0]; // data fifo
    reg [2:0] w_ptr, r_ptr; // fifo write and read pointers

    // detect falling edge of ps2_clk
    reg [2:0] ps2_clk_sync;
    always @ (posedge clk) begin
        ps2_clk_sync <= {ps2_clk_sync[1:0], ps2_clk};
    end
    wire sampling = ps2_clk_sync[2] & ~ps2_clk_sync[1];
    always @ (posedge clk) begin
        if (clrn == 0) begin
            count <= 0;
            w_ptr <= 0;
            r_ptr <= 0;
            overflow <= 0;
        end else if (sampling) begin
            if (count == 4'd10) begin
                if ((buffer[0] == 0) && // start bit
                    (ps2_data) && // stop bit
                    (~buffer[9:1])) begin // odd parity
                    fifo[w_ptr] <= buffer[8:1]; // kbd scan code
                    w_ptr <= w_ptr + 3'b1;
                    overflow <= overflow |
                        (r_ptr == (w_ptr + 3'b1));
                end
                count <= 0; // for next
            end else begin
                buffer[count] <= ps2_data; // store ps2_data
                count <= count + 3'b1; // count ps2_data bits
            end
        end
        if (!rdn && ready) begin
            r_ptr <= r_ptr + 3'b1;
            overflow <= 0;
        end
        assign ready = (w_ptr != r_ptr);
        assign data = fifo[r_ptr];
    end
endmodule

```

2 PS/2 鼠标

PS/2 键盘送扫描码给主机。与此类似,PS/2 鼠标送移动信息及按钮的状态信息给主机。下图给出的是三按钮鼠标(中间按钮带一小轮的那种)的信息格式,共有四个字节。如果是两按钮的鼠标,只有前三个字节。

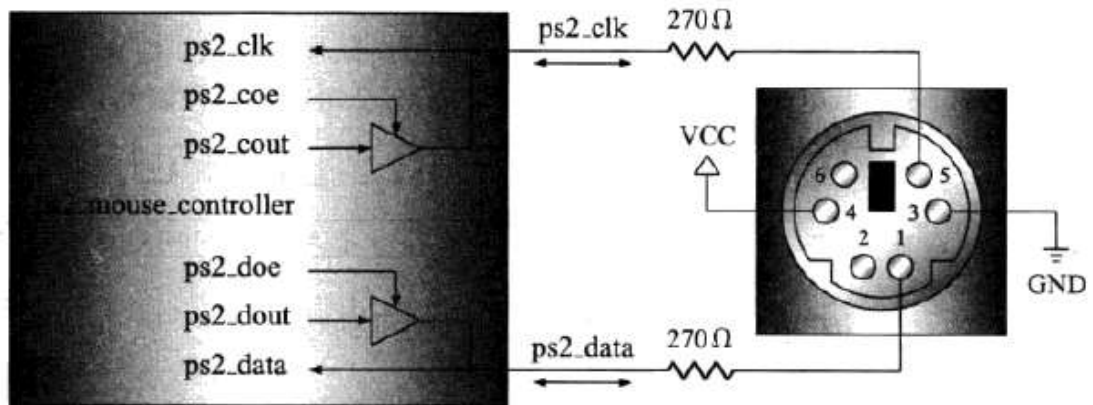
	7	6	5	4	3	2	1	0
Byte 1	Y 上溢	X 上溢	Y_8	X_8	1	中按钮	右按钮	左按钮
Byte 2	X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0
Byte 3	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
Byte 4	Z_7	Z_6	Z_5	Z_4	Z_3	Z_2	Z_1	Z_0

鼠标的左右移动量用 9 位补码表示的带符号数 $X_8 \cdots X_0$ 表示,鼠标向左移为负,右移为正;上下移动量用 9 位补码表示的带符号数 $Y_8 \cdots Y_0$ 表示,鼠标向下移为负,上移为正;小轮子的转动用 $Z_8 \cdots Z_0$ 表示。第一个字节中除了 X_8 和 Y_8 ,其他位的意义如下: $X(Y)$ 上溢表示水平(垂直)移动量超出了表数范围,左、中、右按钮按下时相应的位为 1。

主机可以发送命令给鼠标来选择不同的工作方式。最常用的是所谓的流方式(Stream Mode):当移动鼠标或按下按钮时,鼠标立即送出连续的数据。加电后,鼠标本身进行内部测试,送出一个字节的 AA,表示已经通过了测试。然后送出 00,表示自己是一个 PS/2 鼠标。主机收到 AA 00 后,要送出 F4(Enable),以允许鼠标进入流方式工作状态。鼠标收到 F4 后,送出 FA 作为回答。这时鼠标处在流方式工作状态,可以发出通常的数据包了。

注意,如果你使用 FPGA,刚加电时 FPGA 芯片还没被初始化,无法检测到 AA00。FPGA 可以忽略这两个字节的数据,直接送出 F4 并检测 FA。FPGA 可以在任何时候送出 FF(SofReset)给鼠标,令其进行初始化。鼠标收到这个命令后,送出 FA 作为回答,然后进行内部测试,再送出 AA 00(与加电时的动作相同)。

主机与鼠标之间的信号线有两条:时钟信号 ps2_clk 和数据信号 ps2_data。它们与 PS/2 插口的连接关系见图 15.17。由于鼠标使用它们来接收命令和发送数据,因此这两条线是双向的。在鼠标控制器一侧可以使用三态门来控制主机数据的输出:当 ps2_coe (Clock Output Enable)为 1 时,主机时钟信号 ps2_cout 经三态门送到 ps2_clk 线上,为 0 时三态门输出为高阻;当 ps2_doe (DataOutput Enable)为 1 时,主机数据信号 ps2_dout 经三态门送到 ps2_data 线上,为 0 时三态门输出为高阻。

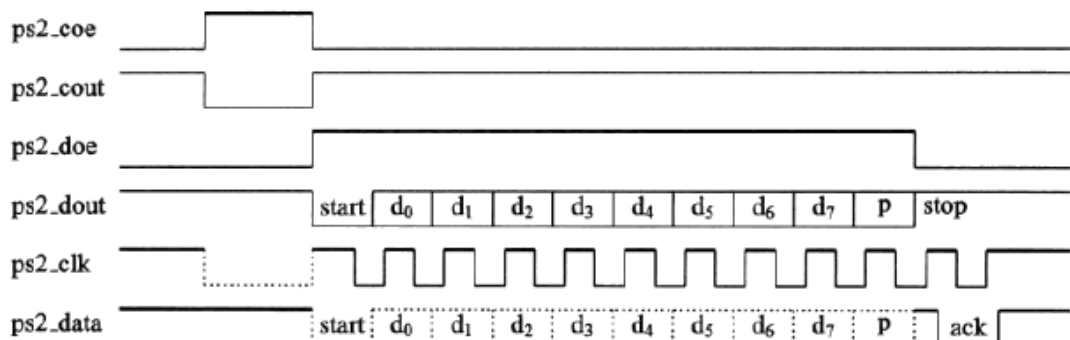


双向信号和三态门控制的 Verilog HDL 描述可以使用以下的例子。当使能信号为 1 时，送出二值信号；为 0 时，送出高阻信号。注意第一行的信号类型是 inout (input and output, 双向)。

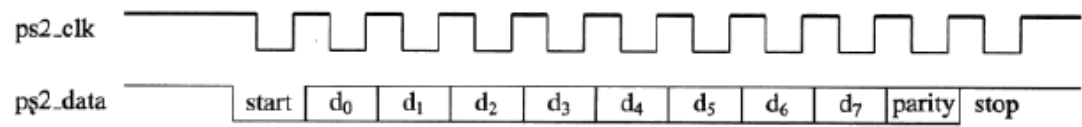
```
Inout ps2_clk, ps2_data; //bi-directional signals
assign ps2_clk = ps2_coe?ps2_cout:1'bz;
assign ps2_data=ps2_doe ?ps2_dout :1'bz;
```

鼠标从主机接收数据(命令)的时序如图 15.18 所示。图中 ps2_clk 和 ps2_data 的虚线部分由主机送出，实线部分由键盘送出。注意图中最上面的 4 个信号是鼠标控制器的内部信号。

首先，由主机送出低电平的 ps2_cout 到 ps2_clk，告诉鼠标，主机要发送数据了。然后释放 ps2_clk 并把起始位经由 ps2_dout 送至 ps2_data。鼠标检测到起始位的下降沿后，开始驱动 ps2_clk，送出一个负脉冲。主机检测到 ps2_clk 的负脉冲后，送出数据的最低位(d0)。然后依次送出数据的其他位，时钟仍由键盘侧提供。当奇校验位送出后，主机释放 ps2_data 线(由提拉电阻送出停止位)。鼠标在收到停止位后，送出一个低电平的 ACK 到 ps2_data 线。然后鼠标要忙着执行刚收到的命令(比如软复位命令 FF)，并告诉主机执行结果(比如送出 FA AA 00)。



鼠标送数据给主机的时序见下图。时钟信号由鼠标侧提供。时钟高电平时标开始送数据，时钟低电平时主机开始读数据，与 PS/2 键盘接口控制器相同。



根据上述原理，仿照键盘代码完成设计。