Mini Project - Mouse Interface

Maryam Hemmati

Department of Electrical, Computer and Software Engineering
University of Auckland

email: m.hemmati@auckland.ac.nz

COMPSYS 305-Digital Systems Design

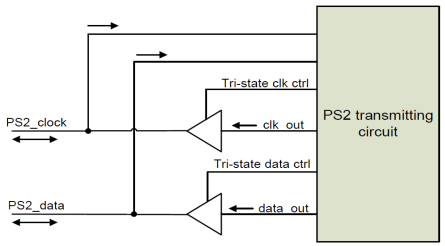
7 April 2022

Content

PS/2 Mouse Interface

The PS/2 device interface was originally developed by IBM and is used by many mice and keyboards.

- The PS/2 mouse and keyboard implement a **bidirectional synchronous** serial protocol.
- The interface consists of two bidirectional signals:
  - **Clock**
  - **Data**



PS/2 Mouse Interface

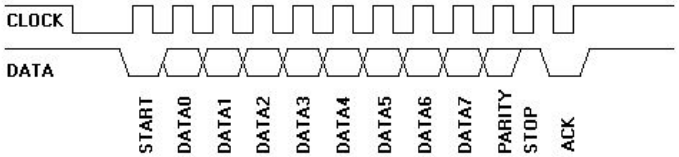
There are three different bus states:

- **Idle**
  - Both *Data* and *Clock* lines are kept **high**.
  - This is the only state where the mouse is allowed to begin transmitting data.
- **Inhibit**
  - The host may inhibit communication at any time by pulling the *Clock* line low.
  - The host then pulls *Data* low and releases *Clock* to enter *Host Request-to-Send*
- **Host Request-to-Send**
  - This state signals the device to start generating clock pulses for the host to device communication.

The clock signal is always generated by the mouse, unless the host wants to inhibit mouse transmission in order to send a command.

PS/2 Mouse Interface

- Data is transmitted **one byte** at a time. Each byte is sent in a sequence consisting of 11-12 bits. These bits are:
  - **1 start bit**: Always '0'
  - **8 data bits**: Least significant bit first.
  - **1 parity bit**: Odd parity
  - **1 stop bit**: Always '1'
  - **1 acknowledge bit**: Host-to-device communication only

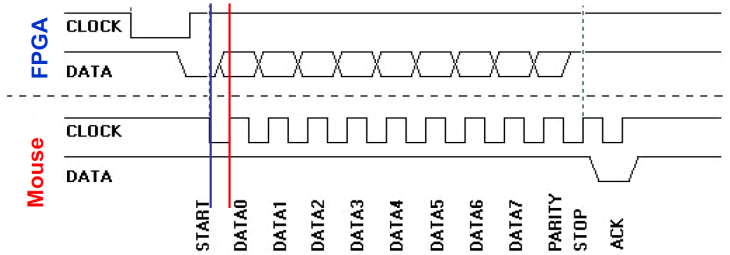


Host (DE0 board) drives the data line when **sending commands** to the mouse.

Mouse drives the data line when **sending data** to the host (DE0 board).

PS/2 Mouse Interface - Host to Device Communication

- The clock signal is always generated by the mouse unless in *Inhibit* state.
- Data is shifted out by the host at clock **falling edge** and is read by the mouse at clock **rising edge**.
- An acknowledgment bit is sent from the mouse to the host.



The diagram shows the separated clock and data signals generated by host and mouse side which can be superimposed.

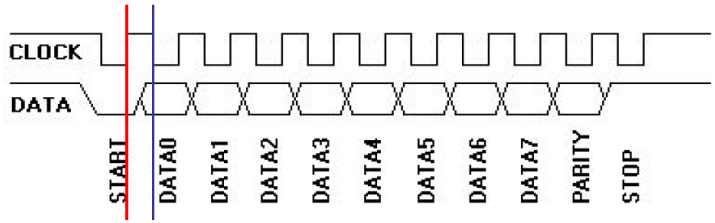
PS/2 Mouse Interface - Host to Device Communication

The steps that host must follow to send data to a PS/2 device:

1. Bring the Clock line low for at least 100 microseconds.
2. Bring the Data line low.
3. Release the Clock line.
4. Wait for the mouse to bring the Clock line low.
5. Set/reset the Data line to send the first data bit.
6. Wait for the mouse to bring Clock high.
7. Wait for the mouse to bring Clock low.
8. Repeat steps 5-7 for the other seven data bits and the parity bit.
9. Release the Data line.
10. Wait for the mouse to bring Data low.
11. Wait for the mouse to bring Clock low.
12. Wait for the mouse to release Data and Clock.

PS/2 Mouse Interface - Device to Host Communication

- The clock signal is generated by the mouse.
- The data bit is written by the mouse at clock **rising edge** and is read by the host at clock **falling edge**.
- No acknowledgment bit is required.



PS/2 Mouse Interface

Mouse Modes

- **Reset Mode**
  - Initialization and self test is done.
- **Stream Mode**
  - Mouse transmits three data packets.
  - Host must transmit **0xF4** command to the mouse to initiate data reporting.
- **Remote Mode**
  - Host requests movement data packets.
- **Wrap Mode**
  - This is diagnostic mode.

PS/2 Mouse Interface - Initialization

- At power up, the mouse will go to **Reset Mode**.
  - It runs a self test and sends the message **0xAA**, which indicates that the test is passed.
  - It then sends **0x00** which is the ID for standard PS/2 mouse.
  - If the PS/2 mouse is connected before the FPGA device is programmed this data can be ignored in the interface design.
- In **Stream Mode**, the streaming should be enabled by a command from host.
  - Command **0xF4** is sent by the host.
  - Acknowledge is sent by the mouse.
  - Mouse continuously transmits three data packets which include the movement information as well as the pushbutton status.
  - Mouse sends data at a default sampling rate of 100 samples per second.

PS/2 Mouse Interface - Data Packet Format

Bits	MSB (7)	6	5	4	3	2	1	LSB (0)
Byte 1	Yo	Xo	Ys	Xs	1	0	R	L
Byte 2	X7	X6	X5	X4	X3	X2	X1	X0
Byte 3	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

- **X7-X0**: Moving distance of X-axis (horizontal move).
  - moving right = positive values
  - moving left = negative values
- **Y7-Y0**: Moving distance of Y-axis (vertical move).
  - moving up = positive values
  - moving down = negative values
- **Xo**: X data overflow bit (1 = overflow)
- **Yo**: Y data overflow bit (1 = overflow)
- **Xs**: X data sign bit (1 = negative)
- **Ys**: Y data sign bit (1 = negative)
- **L**: Left button status bit (1 = button pressed, 0 = released)
- **R**: Right button status bit (1 = button pressed, 0 = released)

PS/2 Mouse Interface

- Byte 1 of the packet contains button status information and X, Y motion sign, and overflow bits.
  - 8-bit distance value together with the sign bit covers the range of -256 to +255.
  - If the movement is too fast the overflow bits are set.
- Byte 2 and 3 contain the value of relative X, Y movement.
  - These values should be added to the current cursor position.
  - Two 10-bit registers are used to store the cursor position within the screen range of 640x480.
  - The registers should be updated once a new packet of data is received.
- The cursor is usually initialized to the center of a video screen at power up.

PS/2 Mouse Interface - Mouse Component

```
ENTITY MOUSE IS
   PORT( clock_25Mhz, reset             : IN std_logic;
         SIGNAL mouse_data              : INOUT std_logic;
         SIGNAL mouse_clk               : INOUT std_logic;
         SIGNAL left_button, right_button: OUT std_logic;
         SIGNAL mouse_cursor_row        : OUT std_logic_vector(9 DOWNTO 0);
         SIGNAL mouse_cursor_column     : OUT std_logic_vector(9 DOWNTO 0));
END MOUSE;
```

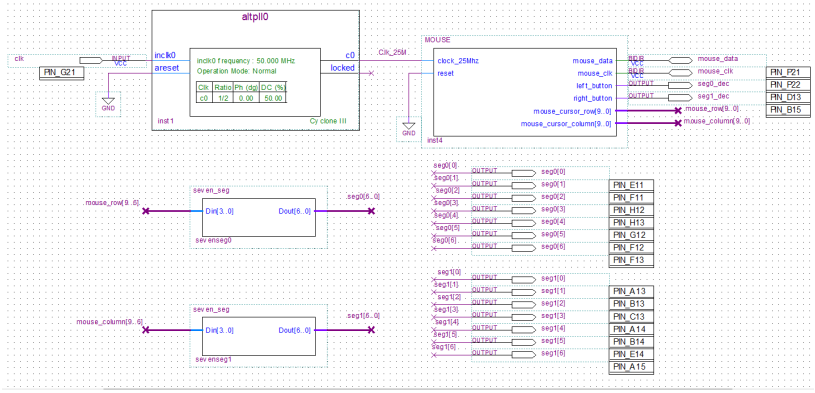
```
mouse_cursor_row <= cursor_row;
mouse_cursor_column <= cursor_column;

          -- tri_state control logic for mouse data and clock lines
MOUSE_DATA <= 'Z' WHEN MOUSE_DATA_DIR = '0' ELSE MOUSE_DATA_BUF;
MOUSE_CLK <=  'Z' WHEN MOUSE_CLK_DIR = '0' ELSE MOUSE_CLK_BUF;
```

```
    WITH mouse_state SELECT
-- Mouse Data Tri-state control line: '1' DE0 drives, '0'=Mouse Drives
        MOUSE_DATA_DIR  <=  '0' WHEN INHIBIT_TRANS,
                            '0' WHEN LOAD_COMMAND,
                            '0' WHEN LOAD_COMMAND2,
                            '1' WHEN WAIT_OUTPUT_READY,
                            '0' WHEN WAIT_CMD_ACK,
                            '0' WHEN INPUT_PACKETS;
-- Mouse Clock Tri-state control line: '1' DE0 drives, '0'=Mouse Drives
    WITH mouse_state SELECT
        MOUSE_CLK_DIR   <=  '1' WHEN INHIBIT_TRANS,
                            '1' WHEN LOAD_COMMAND,
                            '1' WHEN LOAD_COMMAND2,
                            '0' WHEN WAIT_OUTPUT_READY,
                            '0' WHEN WAIT_CMD_ACK,
                            '0' WHEN INPUT_PACKETS;
```


PS/2 Mouse Interface Example

Try this simple example and see how you can see the row and column values of your cursor on seven segment displays.



Summary

- We looked at the PS/2 mouse interface and discussed about its communication protocol.
  - Host to device communication for sending the commands.
  - Device to host communication for sending the data.

Acknowledgment

- Some figures/notes are taken from or inspired by the
  - CS305 Lecture notes by Muhammad Nadeem, 2019
  - EETimes Tutorial : Linear Feedback Shift Registers by Max Maxfield, 2006
  - PS/2 Mouse/Keyboard Protocol by Adam Chapweske, 1999