

Mouse Tutorial

This tutorial details a method for receiving input from a PS/2 mouse or USB to PS/2 mouse adapter using the DE1-SoC. Along with this tutorial is packaged a PS/2 mouse driver written in Verilog. The driver was adapted from example code from Altera. This document will not explain electrical or timing specifications of the PS/2 interface and deals exclusively with the interface between the supplied driver and the user's designs.

Functional Description

The PS/2 protocol supplies, at intervals, displacements of the mouse from its position in the previous interval. To obtain an absolute position, the displacements must be integrated over time, which is the essential function of the supplied mouse driver. A typical use case anticipated for the driver is to use the mouse to select one of a relatively small number of discrete locations arranged in a rectangular grid. Since the resolution of the mouse's movement is potentially much higher than desired in such cases, the integrated X and Y displacements are binned into a smaller number of X and Y discrete locations. It is also desirable to have some amount of hysteresis. That is, if the mouse's position lies near the boundary between two of the discrete locations, small jitters back and forth across that boundary ought not change which bin is selected. The mouse should clearly move well out of one bin and into the interior of the next before the driver responds to the movement. So, there are four parameters which characterize the behavior of the driver. The dimensions of the rectangular grid of discrete locations is described by width and height parameters. A third parameter controls the size of each bin, that is the number of maximum resolution mouse “ticks” between discrete locations. A fourth parameter specifies the size of the no-man's-land along the boundary between bins, how far beyond the boundary of one discrete location the mouse must move into the interior of the next location before the selected bin changes.

Driver Interface

A description of the parameters, inputs, outputs, and inouts of the driver is given below. Users should connect the inouts PS2_CLK and PS2_DAT to top level pins of the same name.

- Parameters
 - WIDTH – the number of bins wide
 - HEIGHT – the number of bins high
 - BIN – the width of each bin in mouse “ticks” (the distance a tick represents varies for each mouse)

- HYSTERESIS – the width of the no-man's-land between bins in mouse “ticks” (this should be some small fraction of BIN, for example 20%)
- User Inputs
 - clock– 50 MHz clock
 - reset – active high reset signal
- User Inputs
 - start – transmits instructions to the mouse to begin, active high
 - reset – active high asynchronous global reset
 - CLOCK_50 – 50 MHz clock
- User Outputs
 - button_left – active high left mouse button down
 - button_right – active high right mouse button down
 - button_middle – active high middle mouse button down
 - bin_x – unsigned integer representing selected X bin*
 - bin_y – unsigned integer representing selected Y bin*
- Top Level Inouts
 - PS2_CLK – clock inout
 - PS2_DAT – data inout

* = the width of this vector is just wide enough to contain the larger of the two dimensions, WIDTH or HEIGHT. That is, it is the base 2 logarithm of whichever is larger, WIDTH or HEIGHT, rounded up to the nearest integer. This is expressed in Verilog as `$clog2(WIDTH>HEIGHT?WIDTH:HEIGHT)`.

Driver code adapted from Altera by Kyle Gagner working with Jesse Liston and Professor Scott Hauck
 Tutorial written by Kyle Gagner
 Driver design validated by Logan Adams