PS/2 Mouse Tutorial

Introduction

This tutorial details a method for receiving input from a PS/2 mouse or USB to PS/2 mouse adapter using the DE1-SoC. A PS/2 mouse driver, adapted from example code from Altera and written in Verilog, is packaged along with this tutorial.

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*: 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.

Driver Interface

- 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, *e.g.*, 20%)
- 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*

* The width of this vector is the minimum number of bits needed to represent the larger of `WIDTH` or `HEIGHT`. This could be expressed in Verilog as `$clog2(WIDTH>HEIGHT ? WIDTH : HEIGHT)`.

- Top-Level Inouts – should be connected to top-level pins of the same name
    - PS2_CLK – clock inout
    - PS2_DAT – data inout

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.