7-Segment SPI Slave Display

Conor Dooley

April 25, 2019

Verilog Interface

```
module Nexys4Display (
input   rst_low_i,
input   block_clk_i,
input   spi_sclk_i, //idle low, posedge active, < 2.5 MHz
input   spi_ss_i,   //idle high
input   spi_mosi_i, //idle high
output  spi_miso_o, //idle high
output  [7:0] segment_o,
output  [7:0] digit_o
);
```

Display Behaviour

The display has 10 8-bit registers to control which digits appear on the screen. Register 0 acts as a bitmask to control which digits of the 7-Segment display are enabled. Registers 1 through 8 contain the value to display in the corresponding location, with index 1 appearing at the right hand side of the board and index 8 on the far left. Register 9 controls the radix point. These registers can be written to via SPI and at startup all will contain zero.
Each bit in Register 0 and 9 is active high. Writing all zeros to these registers will disable all digits or radices respectively, as will be the case at start-up. The segments of the display are not controlled by programmer directly, but rather tells the display which number to display in a given digit by writing to the appropriate register. The display then converts the value contained in the register to segments required to display it. Registers 1 through 8 are those that control the digits, and in accordance with the other registers are also 8 bits wide. To set the digit, an unsigned integer should be written to the relevant address. Of these 8 bits, only the lower 4 bits are used to compute the digit. The state of the upper bits is ignored. Table 1 shows how each register value is interpreted.

Register Value (uint8_t)	Digit
0 → 9	0 → 9
10	Minus Sign
11 → 15	Blank

Table 1: Segment Decoding

SPI Behaviour

The display is be controlled over SPI with the signals `spi_sclk_i`, `spi_ss_i` and `spi_mosi_i`, `spi_miso_o` all of which are idle high. The data is clocked on the positive edge of `spi_sclk_i`. The display is clocked at 6.25 MHz. The maximum SPI clock frequency is 2.5 MHz.
The display implements a 16 bit long transation, containing addressing and command information as well as the register value. Transfers are expected to be upper byte first, MSB first. There is no ability to reconfigure this. The 16 bit long message is broken up as follows: Only one command is implemented, `0x1`, which serves as the write com-

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
command				address				display value							

mand. This allows capabilty of the device to be extended later without impacting already legact implemenations. The address portion of the message is the 4 bit unsigned integer representation of the register number.