

Tasks

1. Breakdown resources and tasks
 - a. Download PMOD documentation from Digilent, schematic of PMOD, and SSD1331 OLED controller reference manual
 - b. Determine I/O of PMOD
 - c. Determine how to startup, turn off, and reset controller
 - d. Determine different available commands
2. FPGA MOSI SPI interface
 - a. Drive FPGA as master to control chip select, MOSI, and slave clock
 - b. Should send 1 byte of data, or 2 hex values at a time
 - c. Should drive D/C (data/command control) bit at start of every byte sent
3. MOSI Byte Buffer
 - a. Take in X bytes to transmit concurrently, with individual D/C bits for each
4. Fill Screen
 - a. Set start/end row/column as screen range
 - b. Fill with background color
5. Display ASCII
 - a. Create table for ASCII values
 - b. Set start/end address based on ASCII dimensions
 - c. Increment through each ASCII display of 8x8 bit pixels
 - d. Restart once done automatically

Nbit_MOSI_SPI

Function

- Load 1 byte of data and serial transmit from MSB to LSB over o_MOSI
- Drive o_CS and o_DC on neg edge so will be stable for slave to read

Parameter

- WIDTH: default 8, N bits to transmit in one set

Input

- I_RST: asynchronous reset
- I_SCK: clock for slave interface, has to transmit on it
- I_DATA: byte to transmit to OLED interface slave (can either be command or data)
- I_START: begin transmitting byte in i_DATA
- I_DC: D/C control to be updated on negedge at same time as data on MOSI line

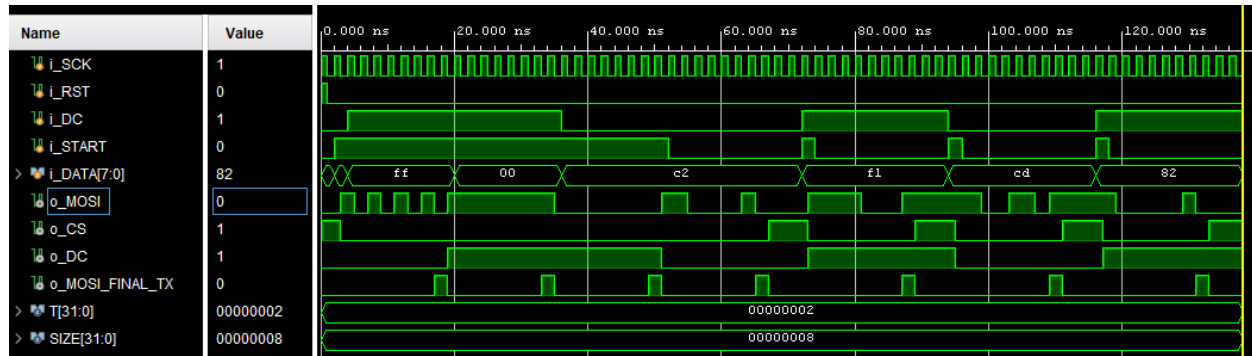
Output

- O_MOSI: Output 1 bit at a time of i_DATA, starting with MSB
- O_CS: chip select, active low
- O_DC: D/C to control data vs. command for the OLED controller
- O_MOSI_FINAL_TX: asserted when last bit transmitted

States

- Idle: waiting for i_START to be asserted once to transmit byte
- Transmit: transmit byte, stay in state if i_START asserted by end, otherwise back to idle

Nbit_MOSI_SPI_tb



Nbyte_MOSI_SPI_buffer

Take in X inputs parallel, serially load them and drive start of master SPI MOSI

Parameters

- WIDTH: number of bits to transmit in one set
- N: max # of sets that can be parallel loaded

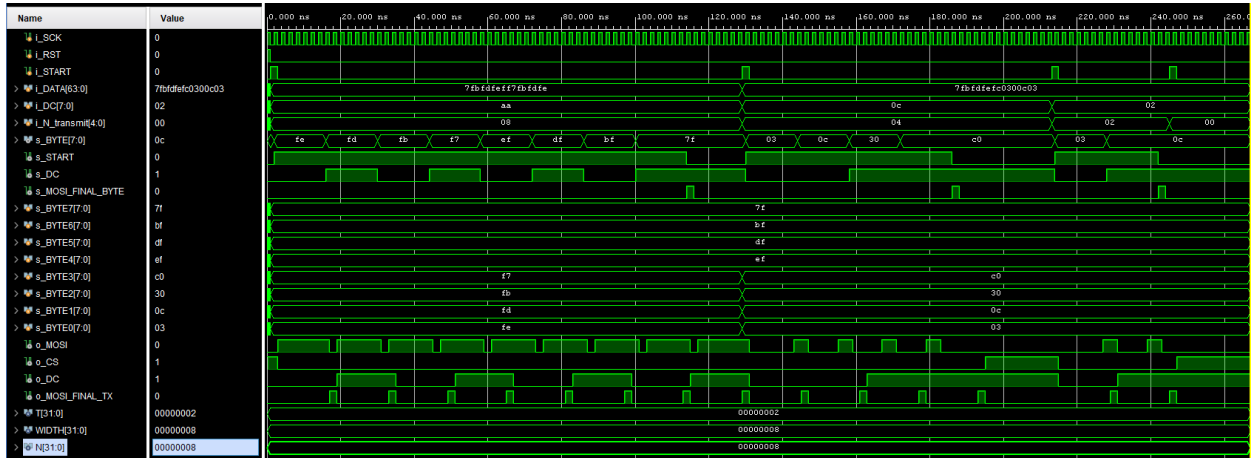
Inputs

- I_SCK: SPI clock
- I_RST: asynch reset
- [WIDTH*N-1:0] i_DATA: 2D bit array holding N sets to transmit over MOSI
- [N-1:0] i_DC: D/C command for each byte
- I_START: start from driver control, initiate parallel load of bytes
- [4:0] i_N_transmit: # of N bytes to transmit on load, will be counted

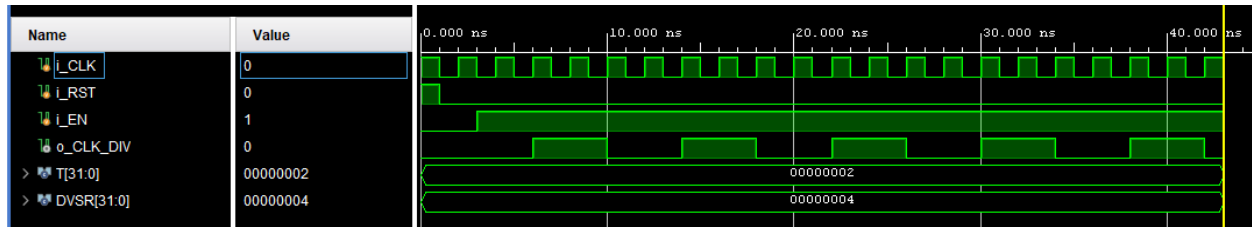
Outputs

- [WIDTH-1:0] o_DATA: byte to feed to master SPI shift module
- O_START: start bit to transmit another byte over MOSI, keep on to keep going
- O_DC: D/C read from input DC for each respective byte
- O_MOSI_FINAL_BYTE: final byte to transmit

Nbyte_MOSI_SPI_buffer_tb



Clock Divider



OLED_Interface

Parameters

- WIDTH: number of bits to transmit in one set
- N: max # of sets that can be parallel loaded

- SCLK_DIVIDER: minimum period for SCK
- WAIT_3_US: # of count to wait 3 us on 100MHz clock
- WAIT_100_MS: # of count to wait 100 ms on 100MHz clock

- NUM_COL: # of pixel columns in OLED array
- NUM_ROW: # of pixel rows in OLED array
- ASCII_COL_SIZE: # of x bits of ASCII char
- ASCII_ROW_SIZE: # of y bits of ASCII char

- N_COLOR_BITS: # of color bits in pixel transmission. Either 16 or 8 bits

Input

- I_CLK: FPGA clock
- I_RST: asynch reset
- [1:0] I_MODE: display mode select, will determine state machine load
- I_START: load bytes and transmit
- I_TEXT_COLOR: text color of ASCII
- I_BACKGROUND_COLOR: color for pixels not ASCII, background
- I_ASCII: 1 byte ASCII values, filling array of ASCII_COL_SIZE * ASCII_ROW_SIZE

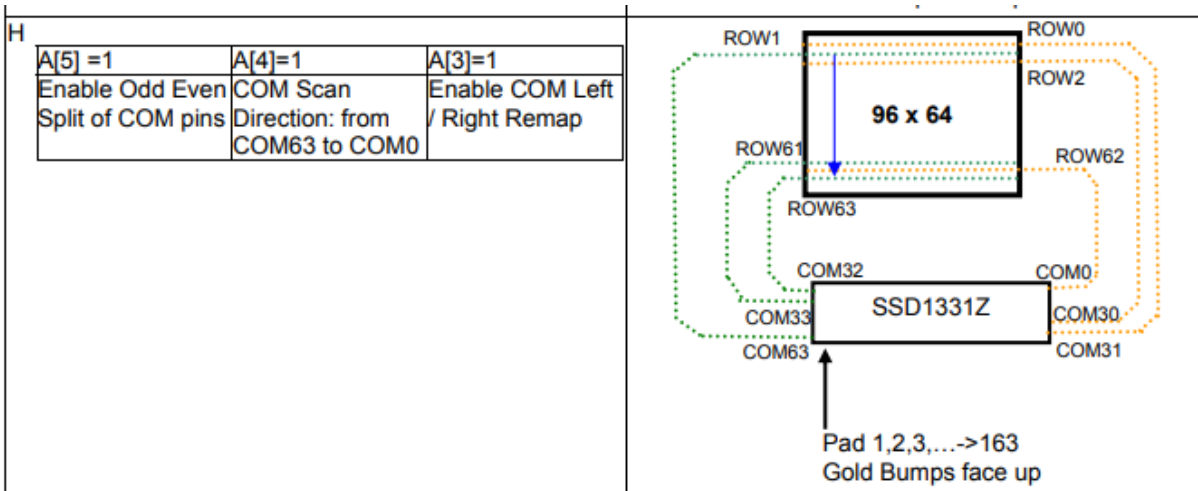
Output

- O_CS: Active low chip select
- O_MOSI: Master out slave in serial bit
- O_SCK: Slave clock
- O_D/C: Data/Command control
- O_RES: power reset
- O_VCCEN: VCC Enable
- O_PMODEN: Vdd Logic Voltage control
- O_MOSI_FINAL_BIT: indicator of final bit in byte transmission
- O_MOSI_FINAL_BYTE: indicator of final byte transmission of SPI byte buffer

Uses the Nbit_MOSI_SPI to send serial bits to OLED display over MOSI. Also drives CS and D/C. This will include a clock divider to count/utilize the FPGA clock and drive O_SCK for the byte buffer and slave OLED display.

Turn on Steps:

1. Apply power to VDD, VDDIO (3V3 supply on FPGA)
2. Set RES pin to low for atleast 3 us
3. Set RES pin to high, set VCCEN to high, set PMODEN to high, wait 3 us
4. After VCC is stable, send command 0xAF for display ON, wait 100 ms
5. Set Display Settings
 - a. Send command byte 0xA0
 - b. Send data byte to update to 256 color format (bit 6 to 1)
 - c. Set display row/col order: A[5:3] = 3b111
 - d. Other bits kept as default settings like in reset



ASCII Draw steps:

1. Load next ASCII value
2. Set start and end row based on ASCII position
 - a. Set Row command, 0x75
 - b. Starting row address (range from 00d to 63d) , data command
 - c. Ending row address (range from 00d to 63d), data command
3. Set start and end column based on ASCII position
 - a. Set Row command, 0x15
 - b. Starting row address (range from 00d to 93d), data command
 - c. Ending row address (range from 00d to 93d), data command
4. Transmit 8x8 bit ASCII
5. Repeat until all ASCII transmitted