Digital Design LU

Protocol

Lab Exercise I

Andreas Ciachi, Matr. Nr. 1029176

e1029176@student.tuwien.ac.at

Vienna, October 24, 2016

Structural modeling

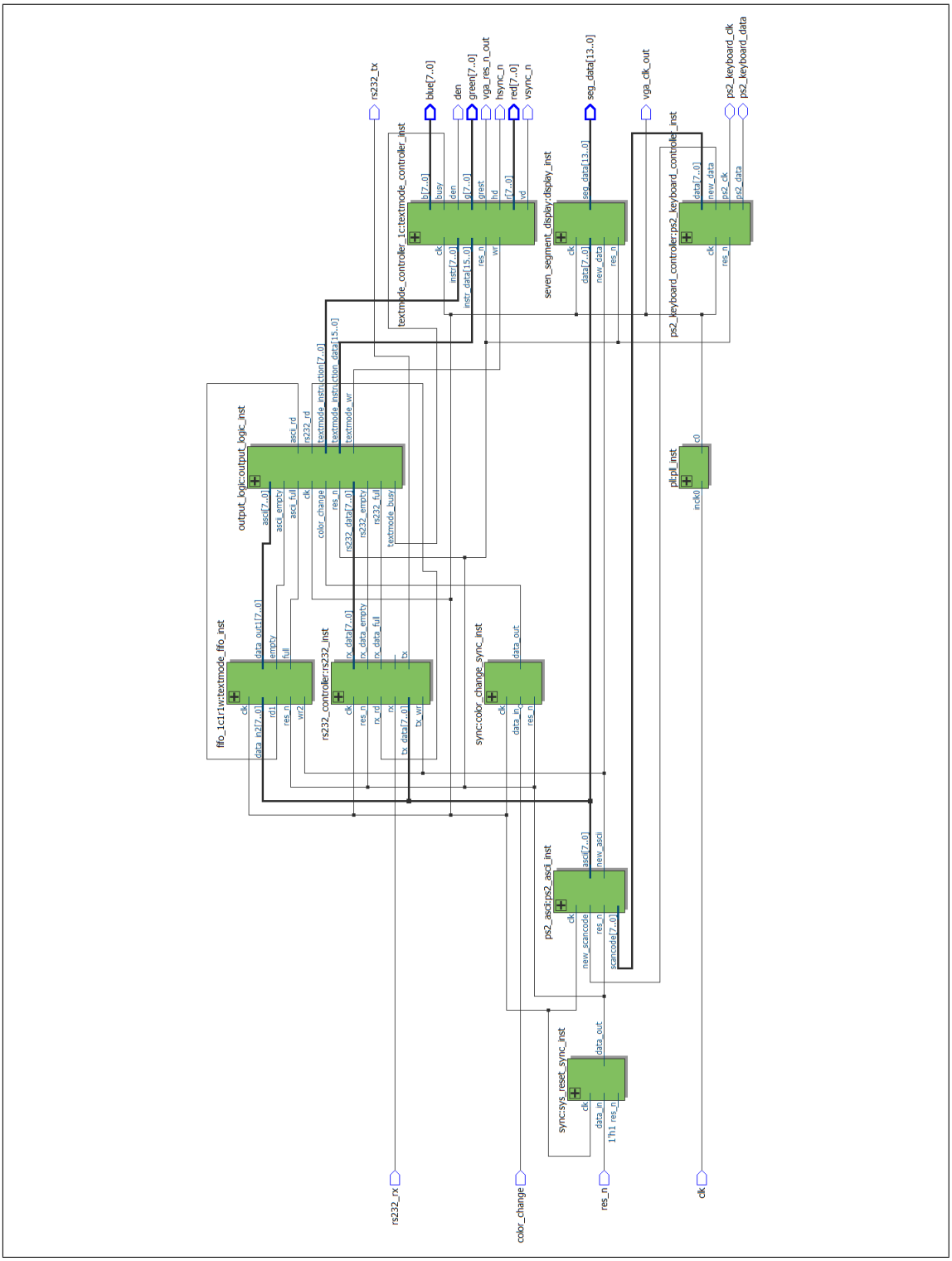

Figure 1: Screenshot of the top level RTL netlist viewer design incl. RS232.

Node Name	Direction	Location	I/O Bank	I/O Standard
blue[7]	Output	PIN_V24	5	3.3-V LVTTL
blue[6]	Output	PIN_P25	6	3.3-V LVTTL
blue[5]	Output	PIN_U27	5	3.3-V LVTTL
blue[4]	Output	PIN_P26	6	3.3-V LVTTL
blue[3]	Output	PIN_R23	5	3.3-V LVTTL
blue[2]	Output	PIN_U22	5	3.3-V LVTTL
blue[1]	Output	PIN_R22	5	3.3-V LVTTL
blue[0]	Output	PIN_V28	5	3.3-V LVTTL
clk	Input	PIN_Y2	2	3.3-V LVTTL
color_change	Input	PIN_M21	6	3.3-V LVTTL
den	Output	PIN_V27	5	3.3-V LVTTL
green[7]	Output	PIN_V26	5	3.3-V LVTTL
green[6]	Output	PIN_L21	6	3.3-V LVTTL
green[5]	Output	PIN_R27	5	3.3-V LVTTL
green[4]	Output	PIN_L22	6	3.3-V LVTTL
green[3]	Output	PIN_R28	5	3.3-V LVTTL
green[2]	Output	PIN_N25	6	3.3-V LVTTL
green[1]	Output	PIN_V23	5	3.3-V LVTTL
green[0]	Output	PIN_N26	6	3.3-V LVTTL
hsync_n	Output	PIN_P21	5	3.3-V LVTTL
ps2_keyboard_clk	Bidir	PIN_G6	1	3.3-V LVTTL
ps2_keyboard_data	Bidir	PIN_H5	1	3.3-V LVTTL
red[7]	Output	PIN_J26	6	3.3-V LVTTL
red[6]	Output	PIN_T25	5	3.3-V LVTTL
red[5]	Output	PIN_L27	6	3.3-V LVTTL
red[4]	Output	PIN_T26	5	3.3-V LVTTL
red[3]	Output	PIN_L28	6	3.3-V LVTTL
red[2]	Output	PIN_U25	5	3.3-V LVTTL
red[1]	Output	PIN_V25	5	3.3-V LVTTL
red[0]	Output	PIN_U26	5	3.3-V LVTTL
res_n	Input	PIN_M23	6	3.3-V LVTTL
rs232_rx	Input	PIN_G12	8	3.3-V LVTTL
rs232_tx	Output	PIN_G9	8	3.3-V LVTTL
seg_data[13]	Output	PIN_U24	5	3.3-V LVTTL
seg_data[12]	Output	PIN_U23	5	3.3-V LVTTL
seg_data[11]	Output	PIN_W25	5	3.3-V LVTTL
seg_data[10]	Output	PIN_W22	5	3.3-V LVTTL
seg_data[9]	Output	PIN_W21	5	3.3-V LVTTL
seg_data[8]	Output	PIN_Y22	5	3.3-V LVTTL
seg_data[7]	Output	PIN_M24	6	3.3-V LVTTL
seg_data[6]	Output	PIN_H22	6	3.3-V LVTTL
seg_data[5]	Output	PIN_J22	6	3.3-V LVTTL
seg_data[4]	Output	PIN_L25	6	3.3-V LVTTL
seg_data[3]	Output	PIN_L26	6	3.3-V LVTTL
seg_data[2]	Output	PIN_E17	7	2.5 V
seg_data[1]	Output	PIN_F22	7	2.5 V
seg_data[0]	Output	PIN_G18	7	2.5 V
vga_clk_out	Output	PIN_R21	5	3.3-V LVTTL
vga_res_n_out	Output	PIN_P28	6	3.3-V LVTTL
vsync_n	Output	PIN_U28	5	3.3-V LVTTL
<<new node>>				

Figure 2: Screenshot of pin assignments.

Simulation

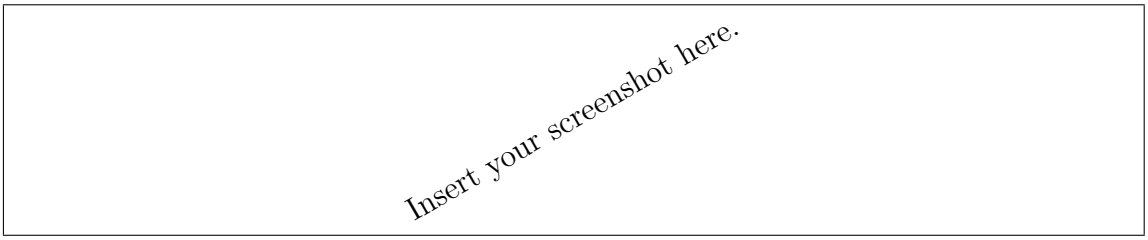

Figure 3: Three characters propagating from input to the displays

Table 1: Timing measurements

Time	Value
First transition of character input on PS2 to ASCII character output	80 ns
ASCII character output to seven segment display update	40 ns
ASCII character output to textmode instruction	200 ns
ASCII character output to RS232 transmission start*	160 ns
1/Display frame rate (*vsync_n* period)	22.133.760 ns (22,1 ms or 45 fps)

* Sidenode: The (additional) measurement for the RS232 transmission was done separately from the measurement for textmode controller, since both interfere with each other.

Question: Different propagation delays: How long is the transition time you measured when the seven segment display output bus changes its value and multiple signals toggle?

Answer: ...

Question: Describe how the bug in the *ps2_ascii* component affects the design.

Answer: ...

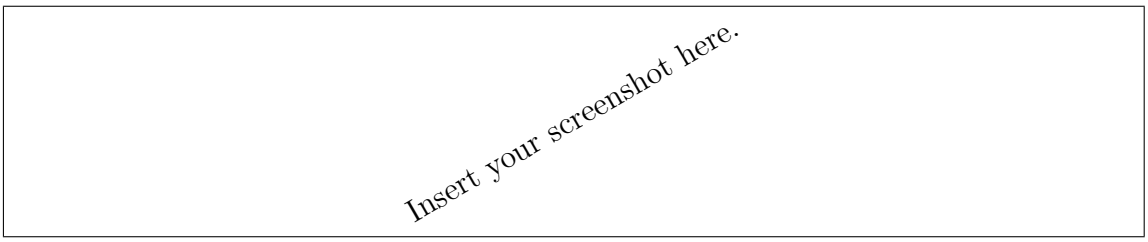

Figure 4: Different propagation delays on the seven segment display bus.

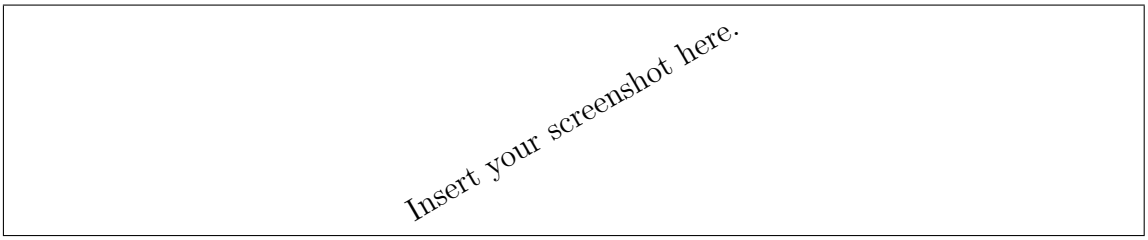

Figure 5: Screenshot of a simulation showing the reception of a whole UART frame.

Behavioral Modeling

Question: Which baudrate did you use for the above simulation? How long should the transmission take for the whole frame (including start and stop bit)? What is the time you measured in the simulation (not including the stop bit)?

Answer: I've chosen 9600 baud as transmission speed since it is widely spread and well supported. One baud consists of 10 bit, start/stop and 8 data bits, which means, 96000 bit are transmitted in 1 second. A frame should take about $\frac{1}{96000bit} = 10.47\mu s$ to transfer.

Table 2: Resouce usage of the serial module (including all submodules).

	LC Combinationals	LC Registers	Memory
Absolute number % of whole design % of whole FPGA resources			

Measurement

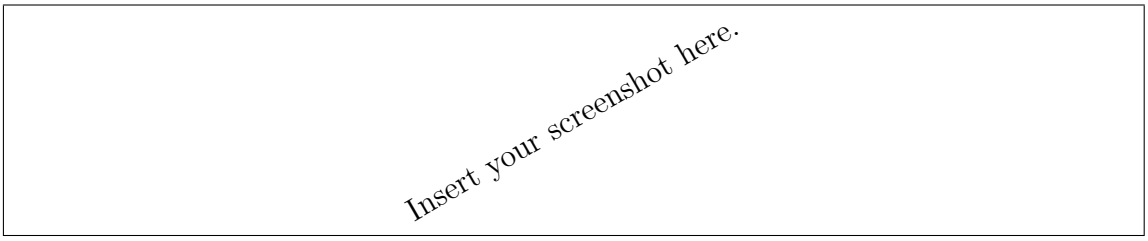

Figure 6: Screenshot of a measurement showing the duration of a whole UART frame sent from the FPGA to the computer.

Question: Which baudrate did you use for the hardware implementation? How long should the transmission take for the whole frame (including start and stop bit)? What is the time you measured (not including the stop bit)?

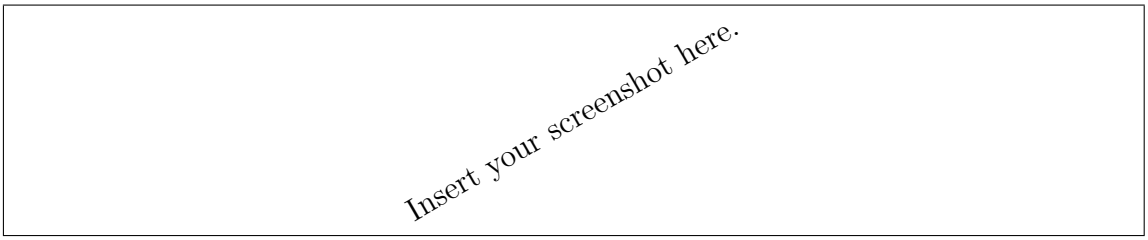

Figure 7: Screenshot of a display timing measurement showing the seventh visible pixel row.

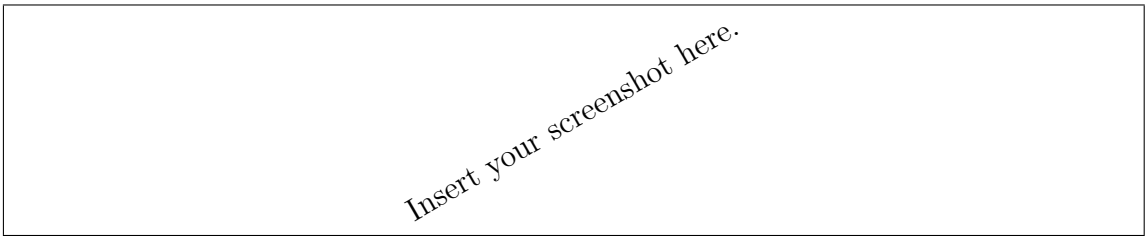

Figure 8: Screenshot of the trigger setting for finding the seventh row.

Answer: ...

Question: How long is the hsync-to-hsync interval you measured?

Answer: ...

Question: What is the sampling rate you used for this measurement?

Answer: ...

Question: What is the uncertainty you have to add to the measured time when sampling with this sample rate? ($\pm$? ns)

Answer: ...

Question: What is the maximum frequency that you could reliably display as a switching waveform at this sample rate (assuming a 50% duty cycle)?

Answer: ...

Feedback & Comments

By answering the optional questions below you can give us feedback and help us to further improve this lab course. Your answers will not influence your grading!

Question: How many hours did you need to solve this lab exercise? Please give us a rough estimate.

Answer:

Question: Were there any annoying problems you encountered (e.g. bugs in tools, flaws in the task description or documentation, etc.)?

Answer:

Question: Other remarks?

Answer: