Martin Skriver
maskr@mmmi.sdu.dk
RMEMB1 fall 2018

Lab assignment

Finish assignment 1 and 2 and make a short journal (max 10 pages) documenting the lab work. The journal should be uploaded along with the source code (ONLY .vhd files) to blackboard no later than three days before next lecure.

IMPORTANT: DO ONLY USE 3.3 V WHEN WORKING WITH THE SoC.

Materials:

- 74HC14 schmitt trigger inverter
- TE0722 propeller SoC board
- TE0790 XMOD programmer
- Breadboard

Assignment 1: A simple oscillator

Oscillators necessary for almost every computing devices. This assignment we will make a simple and very poor clock generator (*figure 1*) using three schmitt trigger inverters.

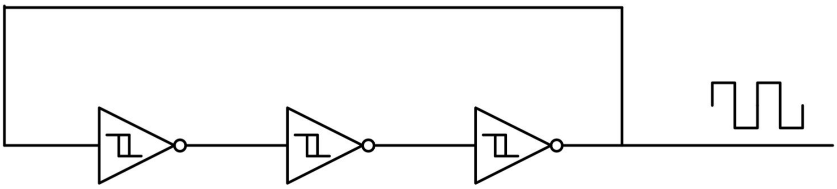

Figure 1: A simple square wave generator using schmitt trigger inverters.

- Place the TE0722 propeller board and the schmitt trigger inverter IC on the breadboard. To get an optimal circuit make sure the IC is place as close as possible to the propeller board.
- Build the circuit shown on figure 1. using the inverter IC. Make sure you have short wires, propper decoupling, grounding unused inputs and use 3-3.3 V as supply voltage.
- Measure frequency, rise time and propagation delay.
- Now try replacing one of the inverters with an inverter build in the PL of the SoC.
- Compare rise time, propagation delay and frequency with the results from before.
- Identify and document how the inverter is build in hardware on the SoC

Assignment 2: Clock divider

It is often necessary to divide a clock signal since the internal clock frequency is often way to high for a PWM generator or communication like I2C and UART.

- Build a clock divider module that takes the oscillator as input and returns a 4-bit vector containing the clock signal divided by 2, 4, 8 and 16.
- Show in the Vivado simulation the behavioral of you implemented module. Use colors and markers to help the reader understand what you are showing.
- Under “RTL Analysis”->”Open Elaborated Design” select the “I/O Planning” view and port the output vector from you module to output pins on the SoC. Make sure you are using “I/O Std” “LVCMOS33” and “Slew Type” “SLOW”.
- Run the implementations steps by clicking “generate bitstream” and program the hardware.
- Show using the oscilloscope that your clock divider is working as expected.
- Document in some details the implemented hardware with schematics and placing of components on the SoC.