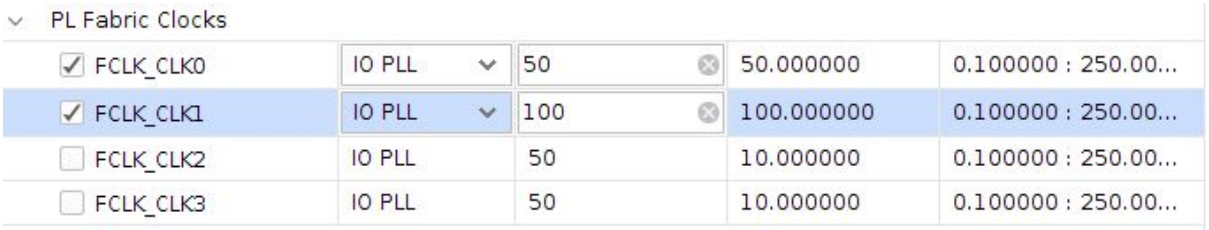
Martin Skriver
maskr@mmmi.sdu.dk

Guide for flashing TE0722 propeller board

1. Create a new RTL project for the “ZYNQ-7 TE0722” board and create a block design.
    a. Under project manager in the menu settings you can change the project target language from Verilog to VHDL.
2. Add the ZYNQ7 Processing System by clicking on the ‘+’ sign in the Design window and search for ZYNQ.
3. Select “Run Block Automation” to let Vivado setup your processing system.
4. Double click on the processor IP and go to the menu “Peripheral I/O Pins”. Vivado should have set up all peripherals while running block automation except setting UART 0 external. Set UART 0 to Peripheral pin 10 and 11.

5. In Clock Configuration under PL Fabric Clocks select 0 and 1 and set the PL frq to the frequency you want in your PL design.

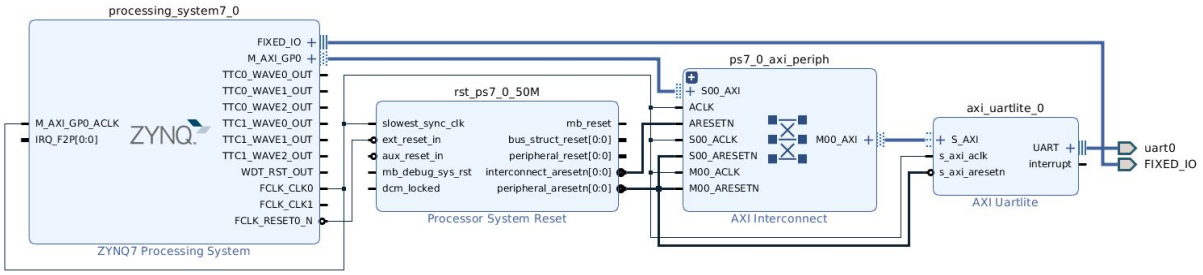


6. Click “OK” to get back to the block design.
7. Click the ‘+’ sign and add an AXI Uartlite IP

8. Click “Run Connection Automation” and select “All Automation”.

9. The block design should now look similar to this one:

Martin Skriver
maskr@mmmi.sdu.dk



10. Add a VHDL module to the design and connect FCLK_CLK1 to the clock input to the module. (e.g. “rgb_test”)
11. Finish the design by making connections external, creating a wrapper and set output pins.
    a. R: J15, G: L14, B: K12
    b. Rx: P14, Tx: P15
12. Click “Generate bitstream” under “program and debug”. (It would be a good idea to make a copy of the project now to avoid doing all this again in future projects)
13. Click “File -> Export -> Export hardware” and select “Include bitstream” in the Export hardware window.
14. Open SDK by clicking “File -> Launch SDK”.
15. From SDK click “File -> New -> Application project”.
16. Name project and leave the rest as it is. Click “Next”.
17. Chose Zynq FSBL (First stage bootloader) and click finish.
18. Open the main file in the FSBL project and locate the following code:

```
#ifdef XPAR_PS7_DDR_0_S_AXI_BASEADDR

    /*
     * DDR Read/write test
     */
    Status = DDRInitCheck();
```

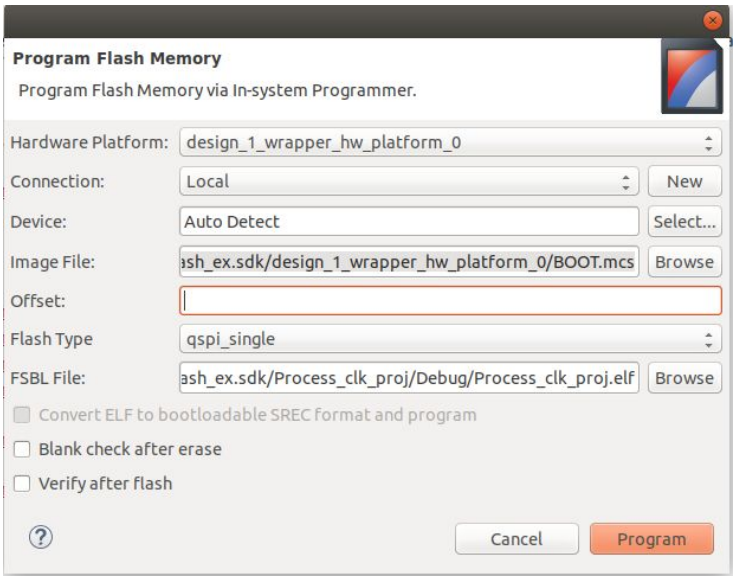
Replace it by this:

```
#define XPAR_PS7_DDR_0_S_AXI_BASEADDR 0
#ifdef XPAR_PS7_DDR_0_S_AXI_BASEADDR

    /*
     * DDR Read/write test
     */
    //Status = DDRInitCheck();
```

19. In the menu select Xilinx-> Create Boot Image
20. Chose a directory for the “Output BIF file” and select output format “MCS”.
21. Click “Add” and in the window chose browse and select the .elf file created in subdirectory <your_project/Debug/ELF_FILE>. Partition type should be bootloader and the rest should be left blanc.
22. Click add again and find the bit file in the wrapper (platform) directory. Choose partition type datafile and leave the rest blanc.
23. Click “Create Image”.
24. Connect the programmer and SoC
25. Click Xilinx -> Program flash memory and browse the paths to the mcs file and the elf file and click Program.

Martin Skriver
maskr@mmmi.sdu.dk



26. You can now program the PL from Vivado and use the clock signal without programming the processor.