Table Of Content

1 General

1.1 Introduction

This project of synthesis, and place and route is not complete. It rather to build an infrastructure for implementing this design using XILINX XST free tool. The project is not complete in that a lot can be improved. For instance applying more constraints in the UCF etc...

1.2 GUI or Command Line Scripts

Working with GUI improves productivity, but makes it difficult to document. Command Line Scripts are text files, which facilitates the documentation and porting to other newer version of the XST tool.

1.3 Download

All files can be downloaded from the download location:download.

1.4 File Suffixes

- Executable scripts use the suffix .unx.
-

2 The Project File

The file, test.prj, lists the verilog files, which are used.

3 The Synthesize Script

3.1 Synthesize File Script

The script is synt.unx. It contains one line “xst -ifn synt.xst -ofn synt.syr”. The output report goes to the file synt.syr. The input which controls the synthesize (part number, top level module etc ...) is synt.xst. The results, which will be used as input to next tools, are placed in the file synt.ngc.

4 Running XILINX Back-end Tools

4.1 UCF

This file is for user constraint. Presently there is not much there, but the main clock set yo 48 Mhz and PADS to FF and FF to PADS.

4.2 NGD

This script creates two new files: synt.ngd and xil.v. The former is to be used by XILINX next tool and the later is a verilog net-list. I use it only for post synthesize simulation as I am not sure how to use ICARUS with SDF.

4.3 Map

After NGD the script map.unx should be run. Its output is an NCD file, test_map.ncd. A few more files are generated during the map process. One of the is the map report, test_map.mrp.

Logic Utilization:

Number of Slice Flip Flops:	2,152 out of	21,504	10%
Number of 4 input LUTs:	5,350 out of	21,504	24%

4.4 PAR

The script par.unx is the next step. This script does two things: it places and routes the design, PAR, and generates a net-list, this time VHDL format using netgen command. The output NCD is directed to test.ncd. From the PAR report I will only list in this document the timing report:

```
--------------------------------------------------------------------------------
  Constraint                                | Requested  | Actual     | Logic
                                            |            |            | Levels
--------------------------------------------------------------------------------
  TS_IFCLK = PERIOD TIMEGRP "clk_48" 20 ns  | N/A        | N/A        | N/A
  HIGH 20%                                  |            |            |
--------------------------------------------------------------------------------
  TSP2F = MAXDELAY FROM TIMEGRP "PADS" TO T | 12.000ns   | 11.960ns   | 12
  IMEGRP "FFS" 12 ns                        |            |            |
--------------------------------------------------------------------------------
  TSF2P = MAXDELAY FROM TIMEGRP "FFS" TO TI | 12.000ns     | 4.615ns      | 1
  MEGRP "PADS" 12 ns                        |              |              |
--------------------------------------------------------------------------------
  TS_dll1 = PERIOD TIMEGRP "dll1" TS_IFCLK  | 20.000ns     | 19.640ns     | 6
  HIGH 50%                                  |              |              |
--------------------------------------------------------------------------------
```

4.5 Trace

Trace generates a detailed timing report, test.twr.

4.6 Bit Generation

This script has never been tested, as I don't have a board to test. I assume that "bitgen -w -f bitgen.ut test.ncd download.bit” would be okay.