Using Xilinx Planahead to create physical constraints

Author: Vegard Endresen

1 Introduction

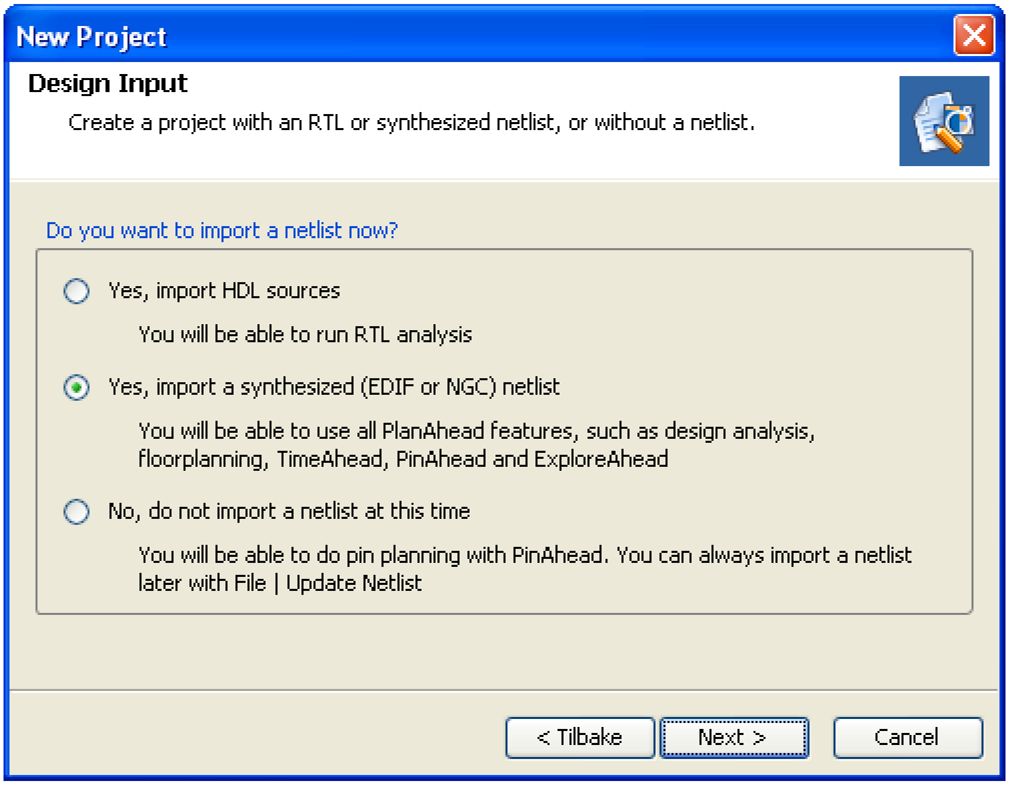
This tutorial will explain the steps to create physical constraints for a Platform Studio project in Xilinx Planahead and how to export these constraints back into the Platform Studio project. Xilinx Design Suite 10.1 is used in this tutorial.

2 Steps

After completing design and synthesis in Platform Studio, open Xilinx Planahead.

2.1 Create a new Planahead project

Create a new Planahead project by *File -> New Project*. Follow the wizard until the window in figure 2.2.1 appears. Select import a synthesized (EDIF or NGC) netlist and click next.



New Project

Design Input

Create a project with an RTL or synthesized netlist, or without a netlist.

Do you want to import a netlist now?

○ Yes, import HDL sources

You will be able to run RTL analysis

● Yes, import a synthesized (EDIF or NGC) netlist

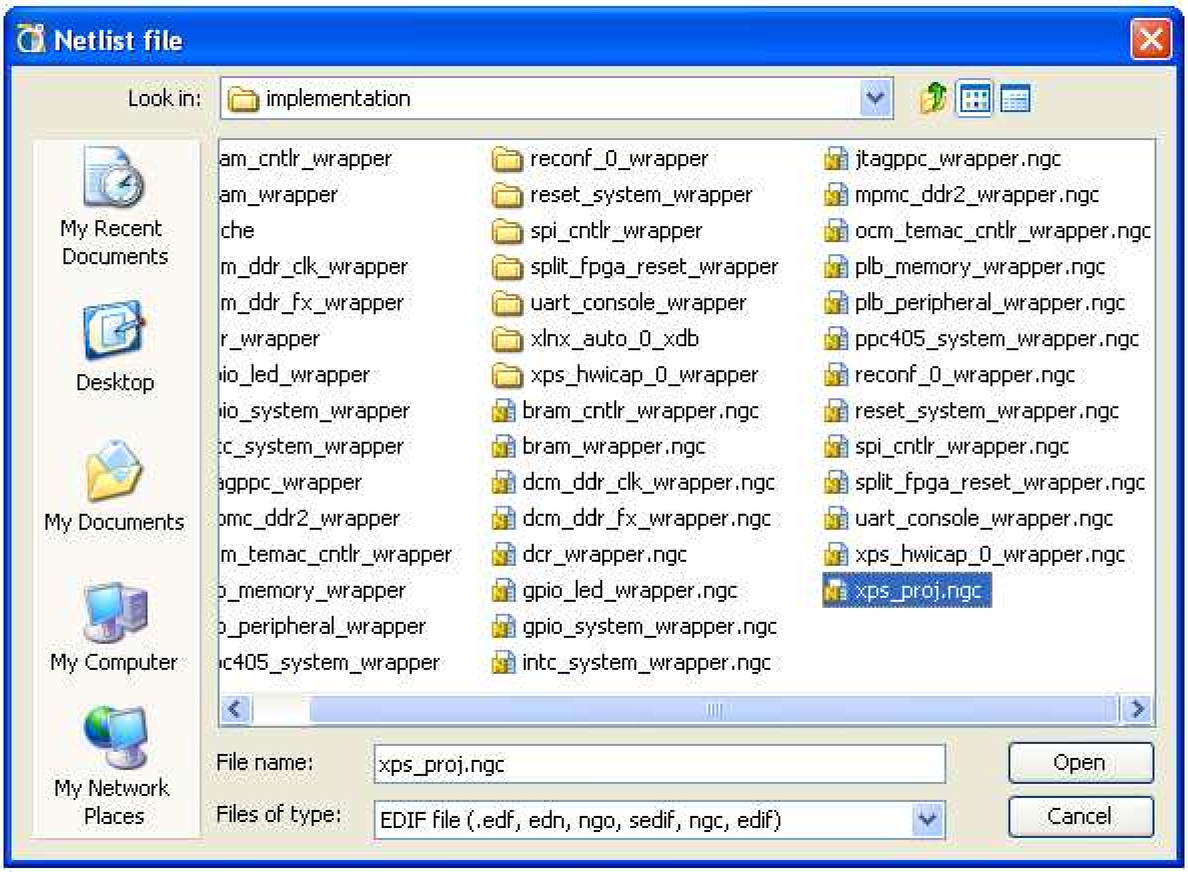
You will be able to use all PlanAhead features, such as design analysis, floorplanning, TimeAhead, PinAhead and ExploreAhead

○ No, do not import a netlist at this time

You will be able to do pin planning with PinAhead. You can always import a netlist later with File | Update Netlist

Figure 2.1.1 Imports

In the next window click the button to the right on the *Netlist File* line. This will open a file browser. In the file browser navigate to the */implementation* folder of the Platform Studio project in question and select the top level NGC file as shown in figure 2.2.2.



Look in: implementation

File name: xps_proj.ngc

Files of type: EDIF file (.edf, edn, ngo, sedif, ngc, edif)

Figure 2.2.2 NGC file select

After selecting the top level NGC file, click next and the netlists from the Platform Studio project will be read. In the following windows select FPGA product family and part. For instance, the Suzaku-sz410 card's product family is *Virtex 4* and part is *xc4vfx12sf363-10.*

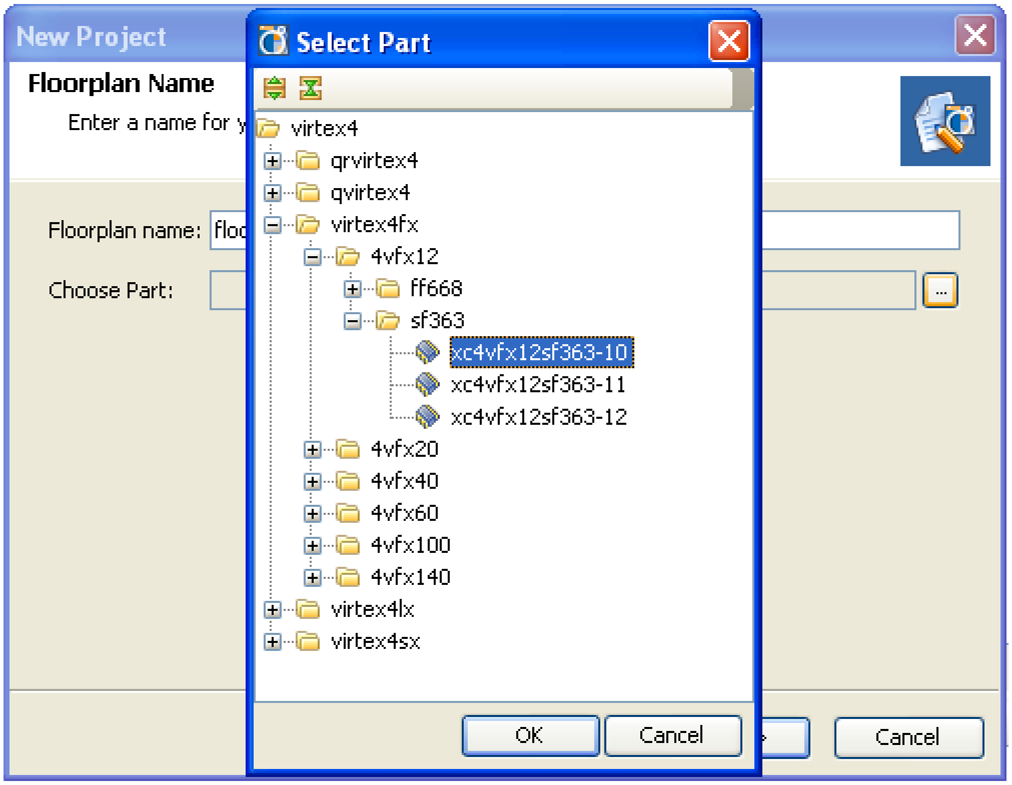
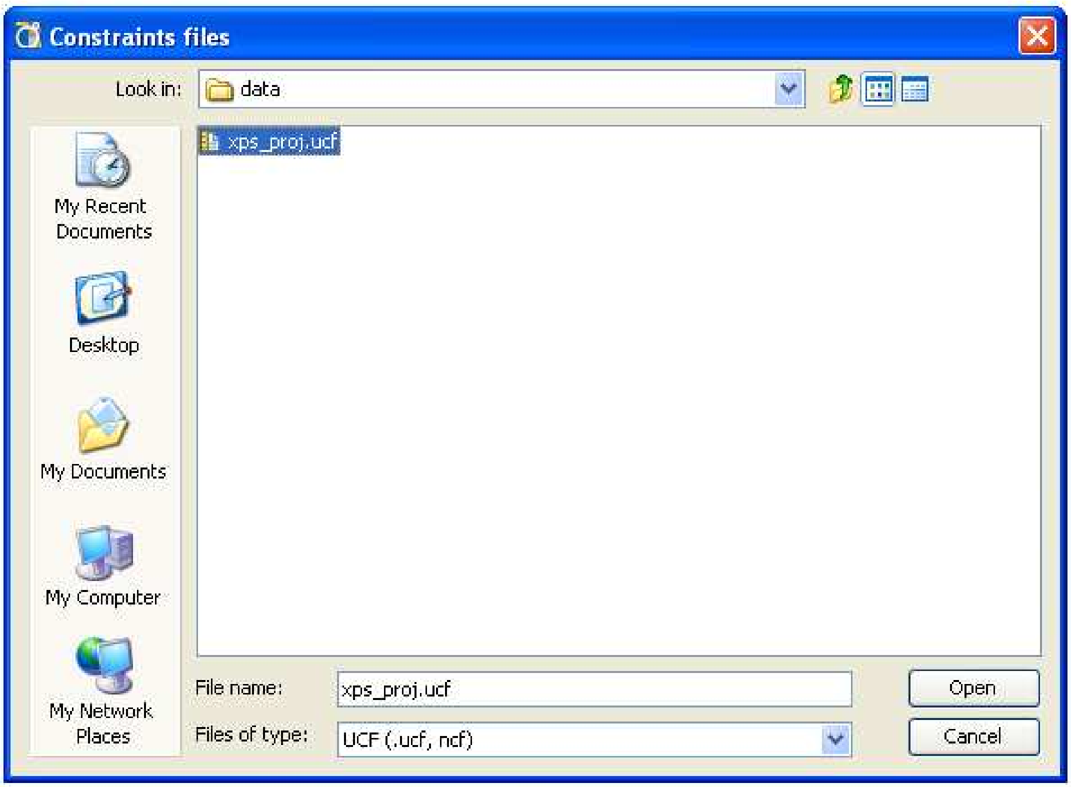


Figure 2.1.3 Part selection

Finally in the *import constraint window*, select *add*. Go to the */data* directory in the Platform Studio project directory and select xps_proj.ucf as shown in figure 2.1.4. Finish the wizard by clicking next in the following windows.



Look in: data

File name: xps_proj.ucf

Files of type: UCF (.ucf, ncf)

Figure 2.1.4 UCF file select

2.2 Creating physical constraints

After creating a new project the Planahead workspace should appear. To create physical constraints for a set of modules they need to be placed in a Pblock (Physical Block). One way of doing this is selecting some set of modules from the *Netlist window*, right click and select *Draw Pblock* and draw a rectangle in the *device window*. A Pblock containing the selected modules should now be created and displayed in the device window as in figure 2.2.1.

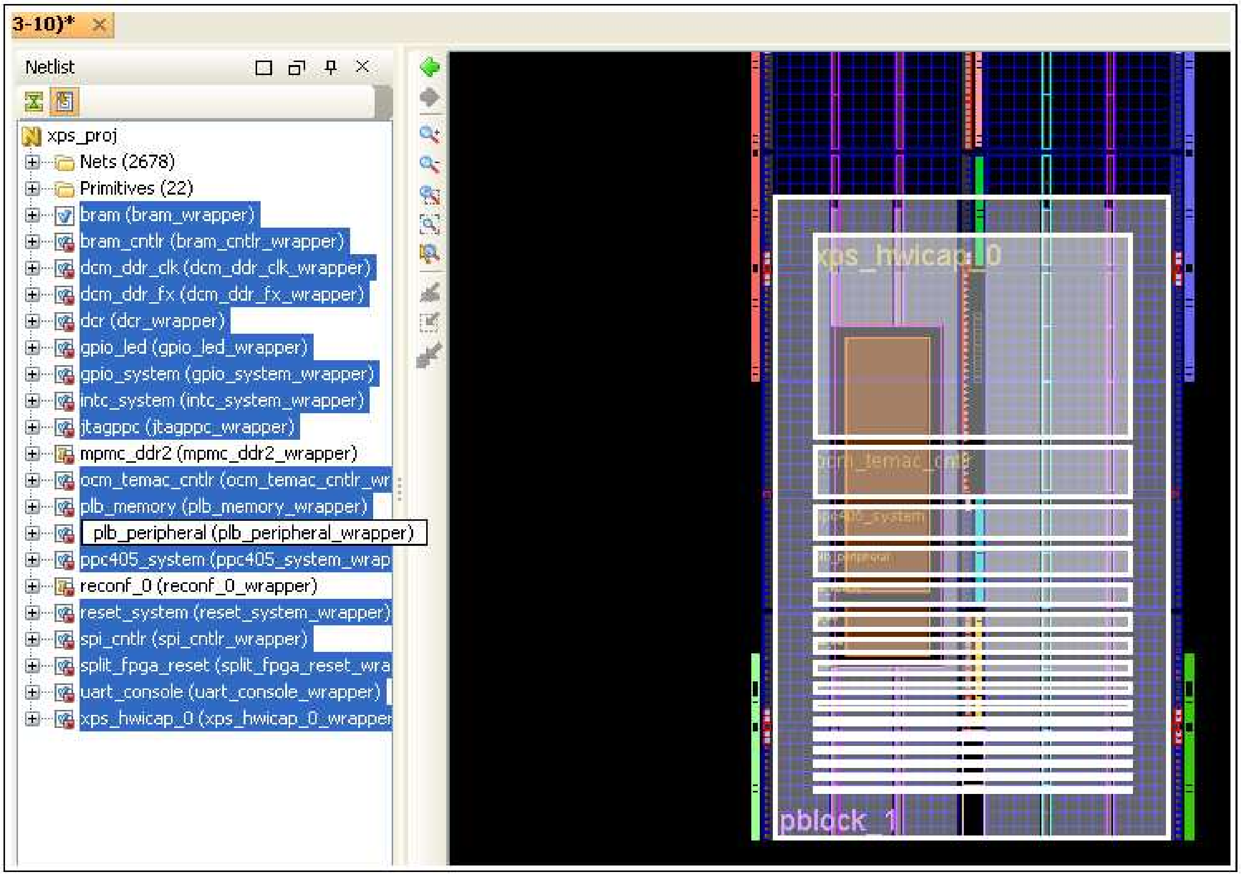
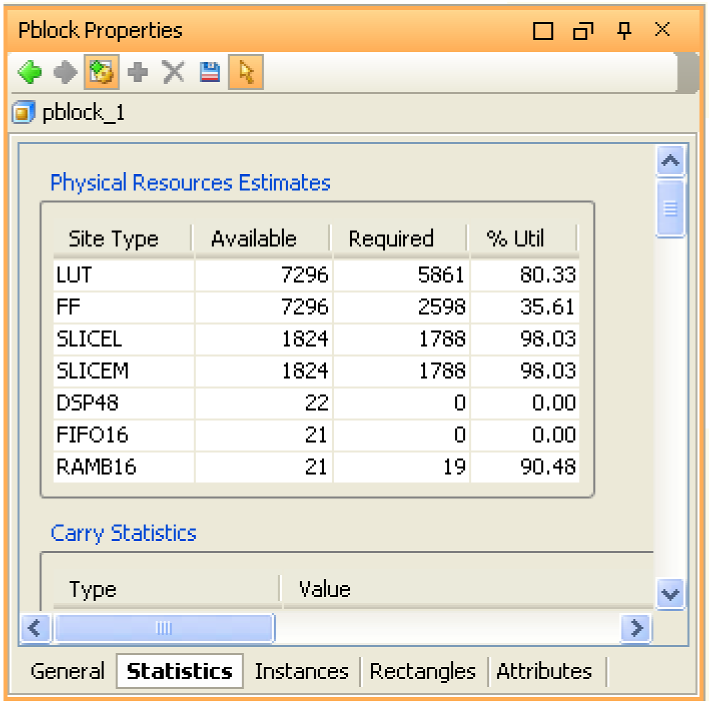

3-10)*
Netlist
xps_proj
Nets (2678)
Primitives (22)
bram (bram_wrapper)
bram_cntlr (bram_cntlr_wrapper)
dcm_ddr_clk (dcm_ddr_clk_wrapper)
dcm_ddr_fx (dcm_ddr_fx_wrapper)
dcr (dcr_wrapper)
gpio_led (gpio_led_wrapper)
gpio_system (gpio_system_wrapper)
intc_system (intc_system_wrapper)
jtagppc (jtagppc_wrapper)
mpmc_ddr2 (mpmc_ddr2_wrapper)
ocm_temac_cntlr (ocm_temac_cntlr_wr
plb_memory (plb_memory_wrapper)
plb_peripheral (plb_peripheral_wrapper)
ppc405_system (ppc405_system_wrap
reconf_0 (reconf_0_wrapper)
reset_system (reset_system_wrapper)
spi_cntlr (spi_cntlr_wrapper)
split_fpga_reset (split_fpga_reset_wra
uart_console (uart_console_wrapper)
xps_hwicap_0 (xps_hwicap_0_wrapper
xps_hwicap_0
ocm_temac_cntlr
ppc405_system
plb_peripheral
pblock_1

Figure 2.2.1 Pblock

A Pblock contains the FPGA resources that are within its borders. To verify that a Pblock contains enough resources for the modules assigned to it, select the Pblock and inspect the *Physical Resources Estimates* table in the *Properties* window. If resource utilization for all types of resources is less than or equal to 100%, there are enough resources within the Pblock to accommodate the assigned modules. If not some action must be taken to ensure that resource demands are met. Figure 2.2.2 shows the *Physical Resources Estimates* table for pblock 1 in figure 2.2.1, which contains enough resources for its modules.



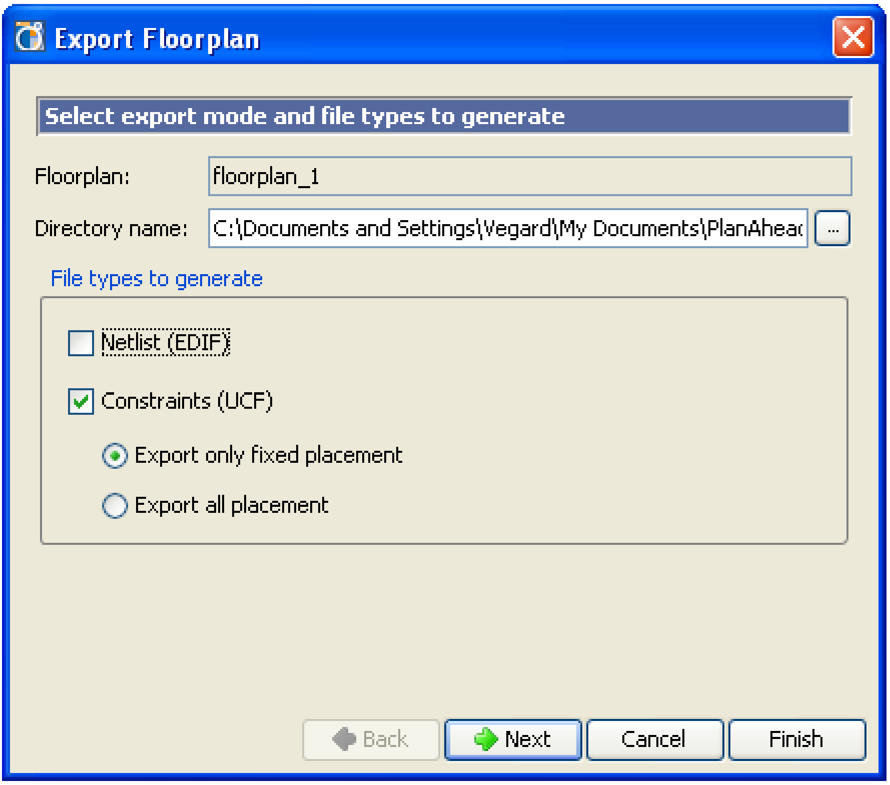
Site Type	Available	Required	% Util
LUT	7296	5861	80.33
FF	7296	2598	35.61
SLICEL	1824	1788	98.03
SLICEM	1824	1788	98.03
DSP48	22	0	0.00
FIFO16	21	0	0.00
RAMB16	21	19	90.48

Figure 2.2.2 Physical Resources Estimates

Planahead enables reallocation through *assign* and *unassign*, which can be used to move modules between existing Pblocks to meet resource demands. It is also possible to resize and move a Pblock, thus changing the number of resources it contains. By partitioning a full design into Pblocks a custom layout can be created.

2.3 Exporting constraints from Planahead

After completing partitioning a design into Pblocks, the constraints can be exported to file by *File -> Export Floorplan*. Deselect EDIF under *files to generate* and select *Export only fixed placement* in the dialog that appears. Next click finish. The generated UCF file can be found in the Planahead project directory and is named top.ucf by default.



Export Floorplan

Select export mode and file types to generate

Floorplan: floorplan_1

Directory name: C:\Documents and Settings\Vegard\My Documents\PlanAheac

File types to generate

☐ Netlist (EDIF)

☑ Constraints (UCF)

● Export only fixed placement

○ Export all placement

Figure 2.3.1 Export to UCF

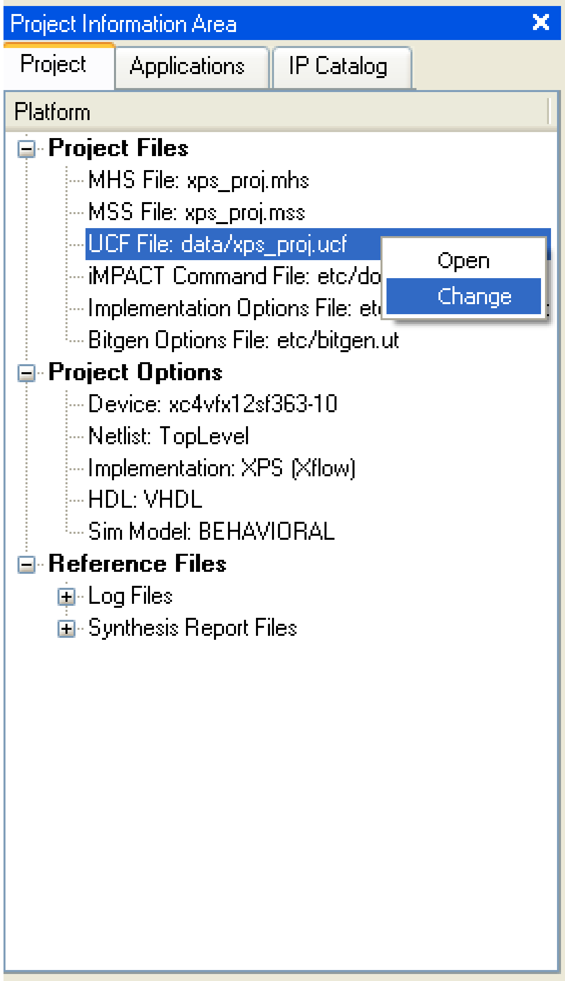
2.4 Importing constraints in Platform Studio

To use the constraints generated in Planahead in a Platform Studio project, top.ucf from the previous section should be used. The UCF file for a Platform studio project, xps_proj.ucf, is located in its */data* directory and can also be found in the *Project Information Area* -> *Project* tab -> *UCF file* in Platform Studio. Ideally one should only need to change the Platform Studio project's UCF file from xps_proj.ucf to top.ucf to import the constraints created in Planahead, but I have had to create a workaround as replacing xps_proj.ucf with top.ucf does not work for me.

When running place and route of a Platform Studio design using a UCF file generated by Planahead, I experience an error with diagnostics: *Conflicting Vcc voltages in bank I/O Bank 8*. Planahead actually modifies the original constraints from the Platform Studio project's UCF file, which are imported during creation of a Panahead project, and causes PAR to fail. A solution is to manually combine the placement constraints created in Planahead with the original constraints from the Platform Studio project's UCF file as follows.

Create a new file called merge.ucf in the */data* directory of the Platform Studio project, copy the region containing constraints generated by PlanAhead from top.ucf and the entire content of xps_proj.ucf into merge ucf. I have attached a UCF folder to this tutorial, containing three UCF files: xps_proj.ucf, top.ucf and merge.ucf. They demonstrate the process of merging UCF files as mentioned here. If changing the Platform Studio project's UCF file to top.ucf works for you, then stick to that and disregard the merging trick.

To change the Platform Studio UCF file, go to the *Project Information Area*, right click *UCF file* and click change as shown in figure 2.4.1 and select the UCF file that is to be used as shown in figure 2.4.2.

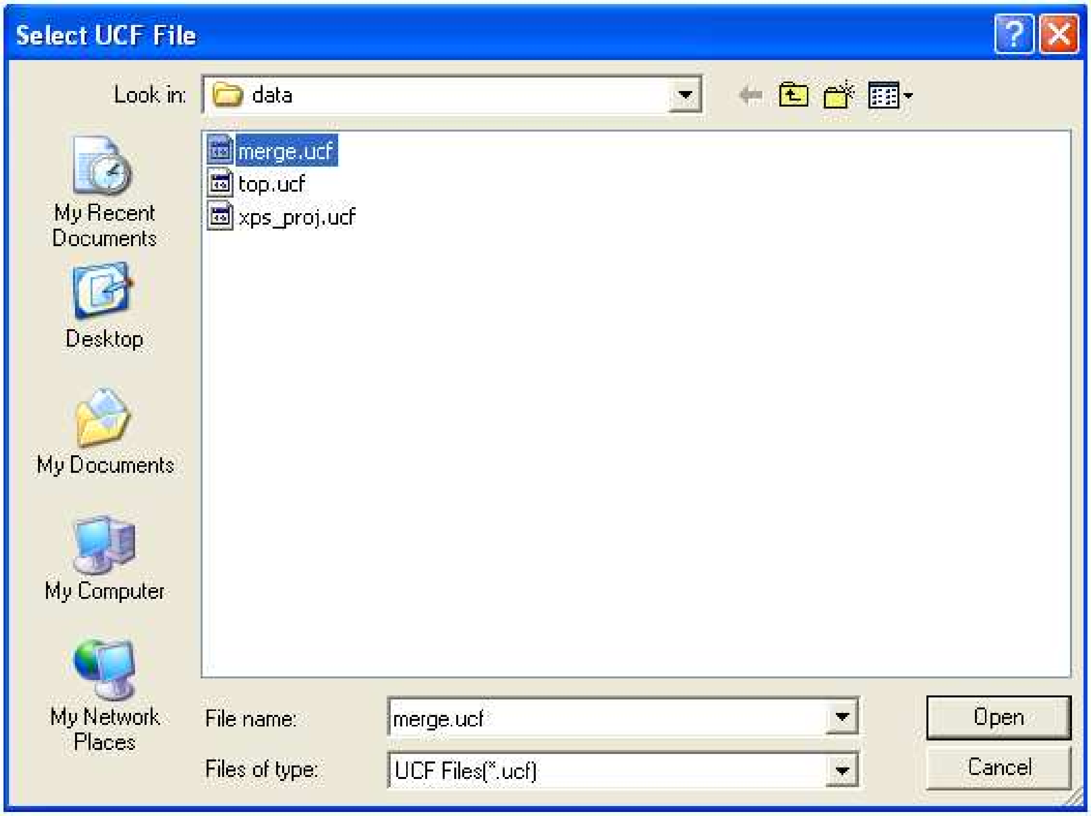


Project Files
- MHS File: xps_proj.mhs
- MSS File: xps_proj.mss
- UCF File: data/xps_proj.ucf
- iMPACT Command File: etc/do
- Implementation Options File: et
- Bitgen Options File: etc/bitgen.ut

Project Options
- Device: xc4vfx12sf363-10
- Netlist: TopLevel
- Implementation: XPS (Xflow)
- HDL: VHDL
- Sim Model: BEHAVIORAL

Reference Files
- Log Files
- Synthesis Report Files

Figure 2.4.1 Changing Platform Studio UCF file



Select UCF File

Look in: data

merge.ucf
top.ucf
xps_proj.ucf

File name: merge.ucf

Files of type: UCF Files(*.ucf)

Figure 2.4.2 Select UCF file

Constraints have now been imported back into Platform Studio and the design can be placed and routed. Make sure to inspect the routed design in FPGA editor afterwards as some restrictions might not work as planned.

3 Final Words

This tutorial shows how to create placement constraints for a Platform Studio project in Planahead and how to export them back into the original project. This is of interest if position of parts of a design must be known, for instance in a reconfigurable design. Any comments or questions to the tutorial can be sent to me at *vegarend@gmail.com*