

## How we accelerated the design and verification processes with the Blue Pearl Software

Melike Atay Karabalkan, ElectraIC

Blue Pearl Software Inc. provides design automation software for ASIC, FPGA and IP RTL verification. At the same time, with its ability to work integrated with Xilinx Vivado, it speeds up the verification and debugging time of FPGA and IP designs.

In this study, first we will talk about how we integrated Blue Pearl Software's Visual Verification Suite into our FPGA design and verification processes. Also, we will try to show the positive effect of the suite on the project calendar using real life examples.

### 1. The Visual Verification Suite and HDL Creator Usage in a Project Lifecycle

A project lifecycle starts with RTL Design phase. The 'RTL Design Phase' refers to the stage of designing the architecture of the entire system. At this stage, a general block diagram of the system is generated first. Each module to be defined in the system is shown in this block diagram. Secondly, the input-outputs of the modules, clock domain information and lower level block diagrams are created. At this stage, Finite State machines and flow charts are defined in a document called the "Micro-Architecture Document" (MAD). RTL Design phase is completed when the MAD is frozen.

After RTL Design, RTL Coding phase starts. In RTL Coding phase, HDL Creator is used as an editor, formatter and real time analyze checker. HDL Creator provides real-time syntax and style code checking inside an intuitive, easy-to-use full featured editor. Unlike standard editors, HDL Creator provides advanced real-time file analysis to find and fix complex issues as you code, such as compilation dependencies and missing dependencies. In addition, HDL Creator provides advanced design views to help you understand and debug as you code.[1].

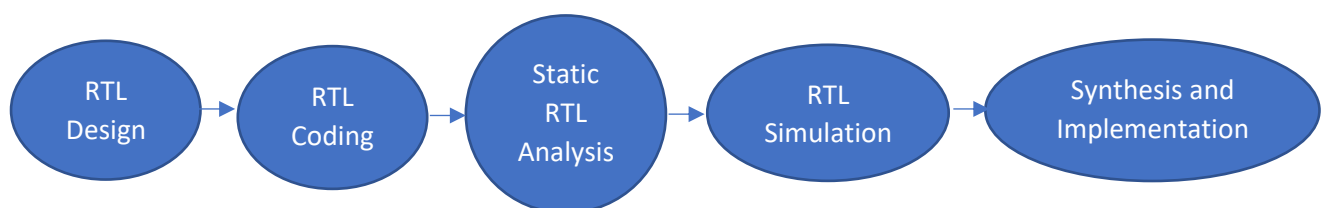


Figure 1 : FPGA Design and Verification Flow

Static RTL Analysis begins after RTL Coding phase completion. In this phase, we prefer to launch the Visual Verification Suite from within the Xilinx Vivado Design Suite. There are three flows to use the Blue Pearl' tools with Vivado. First flow, is to install a \*.tcl file with `bpsvvs::generate_bps_project` tcl command to setup a design in Visual Verification Suite. Second flow, is to install a \*.tcl file and call the Visual Verification tools from Vivado with `bpsvvs::launch_bps` to setup a design in suite. Third flow, is to export Vivado implementation results through `bpsvvs::update_vivado_into_bps` tcl

command. In order to use the suite through Vivado, it is enough to install from Xilinx TCL store as shown in the Figure 2.

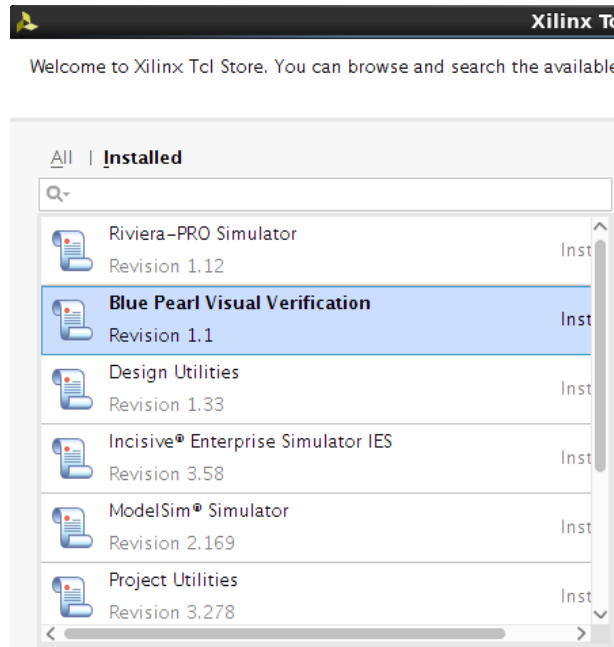


Figure 2 : Xilinx TCL Store – Install Blue Pearl

After the Visual Verification installation phase, we perform static RTL analysis sequence recommended by Blue Pearl as shown in Figure 3. After these steps, suite’s Clock Domain Crossing (CDC) and Automatic SDC (Synopsis Design Constraints) Generation capabilities are used to analyze CDC issues and paths (multi-cycle paths and false paths). When these stages are completed, RTL Simulation and Synthesis phases are applied as usual.

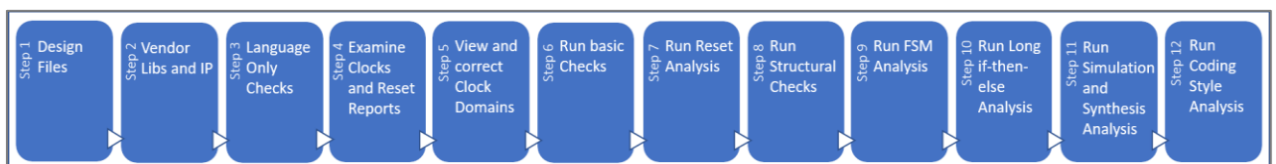


Figure 3 : Recommended Static Analysis Steps

## 2. Positive Effects of Blue Pearl Software tools on Project Calendars

The Visual Verification Suites includes very powerful features such as RTL Analysis, Automatic SDC Generation, CDC Analysis, Advanced Clock Environment, as explained briefly in section 1. This section discusses a completed project without using the Blue Pearl tools. The effect of the tools has been demonstrated in problems encountered during the project that have a significant impact on the project plan.

In terms of giving an idea about the size of the project, there are multiple interfaces with the outside world and intensive signal processing algorithms. Also, there are 5 different clock domains inside the FPGA. Xilinx Kintex-7 FPGA utilization was almost 100%. Figure 4 shows the resource consumption results of the project produced by Xilinx Vivado in the Visual Verification Suite's Management Dashboard. Resource utilization can be transferred from Vivado to the Management Dashboard to generate trends during the project schedule.

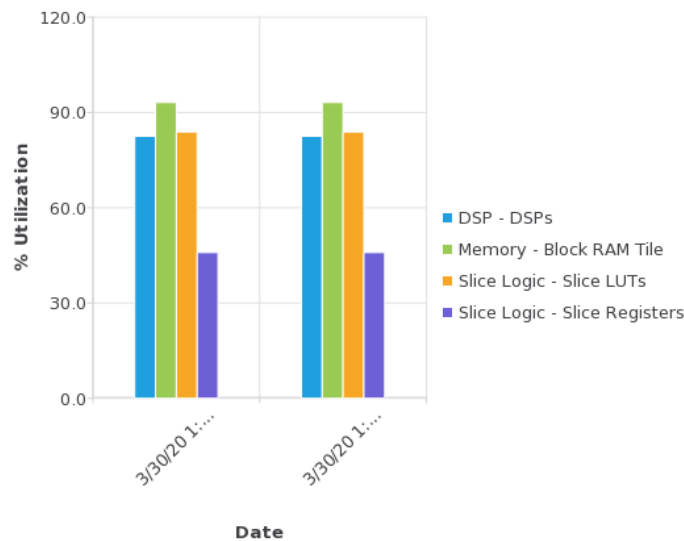


Figure 4 : Referenced Design – Resource Utilization Table

## 2.1 Case 1: Uninitialized FFs

The Visual Verification Suite's Analyze RTL module has lots of RTL checks to analyze the RTL code. One of the most useful check is examining reset issues, including asynchronous vs. synchronous resets and, in the case of asynchronous resets, ensuring that de-assertion is synchronized with the clock to avoid recovery time violations.

In our case, the FPGA design is connected to a host that controls multiple systems. It is very important to reset the design properly in cases where there is an error in any of the units in the system. On the other hand, the system has extensive signal processing algorithms as mentioned above, it takes too long to run the tests determined in the simulation environment. In this case, long iterations may be needed to see the behavior of the system in reset state. It is also a very time-consuming process to analyze these types of behaviors in the laboratory environment. In such complex designs, it can take weeks to catch the problematic module with Vivado ILAs (Integrated Logic Analyzers). However, adding different ILAs also changes the design considerably. Also, it causes serious timing problems.

Figure 5 shows the results generated by Analyze RTL for the "RST\_SEQ\_INT" check analyze. This says that the signal 'send\_char\_val' is not assigned to initial value under reset condition. It can be seen from the figure below that there is no assignment under the reset control for the 'send\_char\_val' signal. When the analysis results are examined in detail, it is possible to catch many problems without losing time before moving to simulation or laboratory environment.

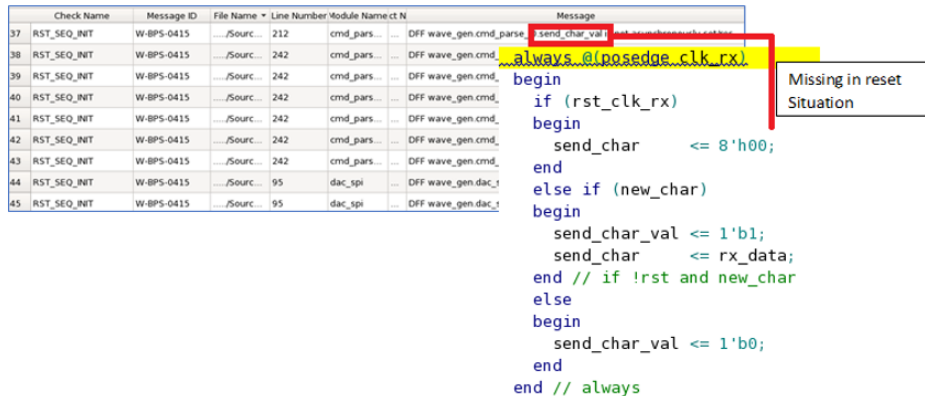


Figure 5 : BPS RST\_SEQ\_INT Check Results

## 2.2 Case 2: Multicycle Path

Giving the most accurate timing constraints to the system is an iterative process even for designers experienced in complex designs. The designer creates the correct constraints iteratively according to the information provided by the timing reports [2].

However, there is always possibility of missing points from time to time. This can lead to designs that are far from optimum in terms of timing, power consumption and area. Therefore, it is very important to create the right timing constraints. The Visual Verification Suite offers a way to detect false paths and multicycle paths in the design.

In our project, there are cascade-connected decimation filters in the signal processing blocks, which is mentioned briefly above. Figure 6 shows the change of input and output signals in a filtering process with 1: 2 decimation. As seen from the figure, the startpoint 's\_axis\_rx\_tdata' changes, it is not changing again for 2 clock cycles and the endpoint 'm\_axis\_tx\_data' captures the result on the 2<sup>nd</sup> clock cycle after the start point changes.

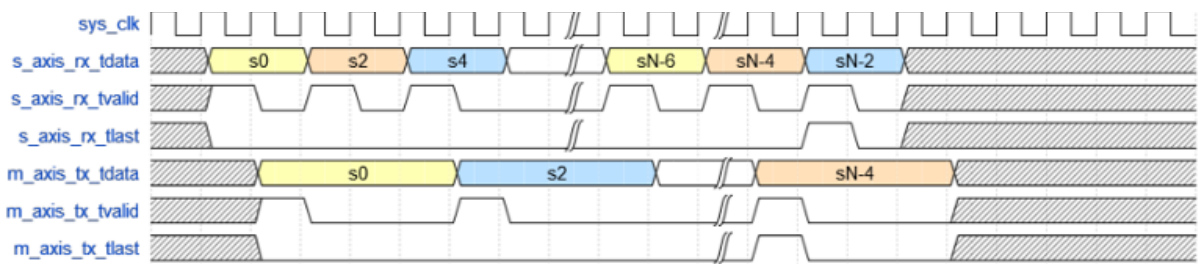


Figure 6 : Decimation Filter Timing Diagram

When the paths of the project are examined with suite, it is observed that there are deficiencies in the path constraints of the decimation filters. The following figure provides a screenshot of the results produced by the suite's path analysis trailer. The path analysis trail showed that multicycle path has occurred for decimation filters.

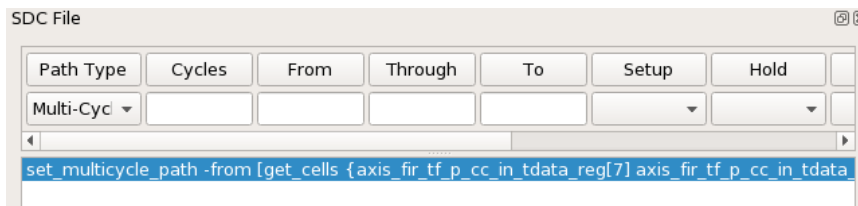


Figure 7 : Path Analyse Results of BPS

In case of missing constraints, the result obtained from the Xilinx Vivado timing report is given in the Figure 8. (These are the results without adding any pipeline register)

Summary	
Name	Path 1
Slack	-3.713ns
Source	axis_fir_tf_p_cc_in_tdata_reg[1]/C (rising edge-triggered cell F
Destination	m_axis_tdata_reg[31]/D (rising edge-triggered cell FDRE clocke
Path Group	n_sys_clk
Path Type	Setup (Max at Slow Process Corner)
Requirement	25.000ns (n_sys_clk rise@25.000ns - n_sys_clk rise@0.000ns)
Data Path Delay	28.684ns (logic 13.063ns (45.541%) route 15.621ns (54.459%))
Logic Levels	73 (CARRY4=47 LUT1=1 LUT2=24 LUT3=1)

Figure 8 : Timing Report before constraining the MultiCycle Path

The results obtained by adding the correct constraints are given in Figure 9. (These are the results without adding any pipeline register)

Name	Path 1
Slack	3.829ns
Source	fir_tf_p_cc_ddc_1/mac/gn_mac[16].mac/Num/CLK
Destination	m_axis_tdata_reg[31]/D (rising edge-triggered cell
Path Group	n_sys_clk
Path Type	Setup (Max at Slow Process Corner)
Requirement	25.000ns (n_sys_clk rise@25.000ns - n_sys_clk rise@0
Data Path Delay	20.892ns (logic 12.358ns (59.152%) route 8.534ns (40.
Logic Levels	57 (CARRY4=34 LUT2=22 LUT3=1)

Figure 9 : Timing Report before constraining the MultiCycle Path

According to reports, it can be seen that there is a serious improvement in worst negative slack after adding constraints. The effect of timing problems resulting from the lack of these constraints on the Project Schedule is given in Section 3.

### 2.3 Case 3: Clock Domain Crossing Issues

Especially in complex FPGA designs, where communication with different devices around the FPGA is intensive, multiple clock domains are needed. In this case, it is mandatory to cross from one clock domain to another clock domain at many points inside the FPGA.

Clock domain crossings can be two types in RTL designs.[3]

1. Synchronous CDC
2. Asynchronous CDC

In synchronous CDC types, a relationship can be defined between clocks in terms of frequency and phase. Data crossing between these types of clocks are known as synchronous CDC. In asynchronous CDC types, there is no relationship between clocks those are interacting each other. Asynchronous CDC paths can cause metastability, data loss and data incoherency problems. These asynchronous points, that cannot be captured in the synthesis tools, cause problems that may take weeks to be found in the laboratory environment.

The Visual Verification Suite offers the capability to visualize clocks and asynchronous clock domain crossings in RTL designs via its Advanced Clock Environment (ACE). ACE summarizes data paths between various clocks and groups clocks into clock domains. It is used before running a CDC analysis to see if clocks are not in the intended domains and make corrections before in-depth CDC analysis.

In the relevant design, there are multiple interfaces (ADCs, DDR3 Chips, PCIe etc.) connected to the FPGA. Each interface has different reference clock source, so inside FPGA there are 5 different clock domains interacting each other. As a result, ACE shows 5 different clock domain groups for our design. Also, ACE reports 1110 asynchronous paths between these 5 clock domains.

- *ddr3\_clk\_group* : DDR3 Interface Clock
- *adc\_clk\_12\_group* : ADC Group -1
- *adc\_clk\_34\_group* : ADC Group - 2
- *pcie\_clk\_group* : PCIe Interface Clock
- *sys\_clk\_group* : System Clock of FPGA

After analyzing clock domains and paths in ACE, asynchronous paths are analyzed with the suite's CDC visualizer by performing CDC analysis. The example of the asynchronous CDC that we spent the most time in this 1110 asynchronous paths during the test phase is given in *figure 10* as an example. The signal '*write\_process\_completed*' at *ddr3\_clk\_group* domain is used to indicate that write operation to DDR3 external memory have been completed. On the other hand, this signal is checked in another module that operates at *sys\_clk\_group* domain without any prevention. This clock domain crossing is shown as 'unsynchronized CDC' in the CDC report.

The frequency of *usr\_clk* (*ddr3* clock group) is 200 MHz and the frequency of *sys\_clk* is 125 MHz. Whenever a new '*write\_process\_completed*' is generated, it may not be captured by the *sys\_clk* domain in the very first cycle because of metastability. If each transition on the source signal is captured in the destination domain, data is not lost. In order to ensure this, the source data should remain stable for some minimum time, so that the setup and hold time requirements are met with respect to at least one active edge of destination clock [4].

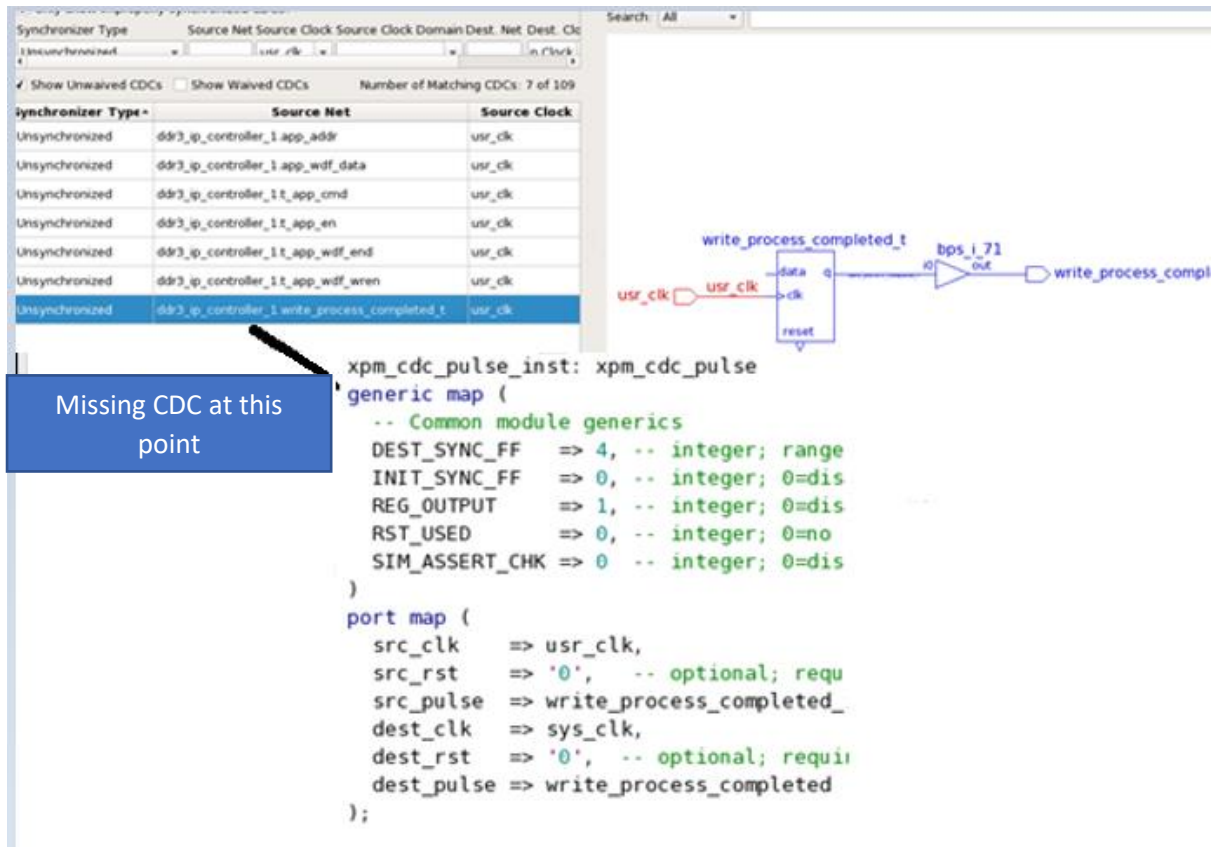


Figure 10 : BPS CDC Analyse Results

In the project, Xilinx Parameterized modules are preferred for CDC problems. Xilinx Parameterized Macros can be used as a Clock Domain Crossing solution for 7 series, UltraScale and UltraScale+ FPGAs. When you select the Xilinx version as shown in *Figure 11* before running analysis with the Visual Verification Suite, the XPM library is also recognized when loading the design.

This path is not seen in the asynchronous CDC report when the analysis is repeated after instantiating the XPM module as in *Figure 12*.

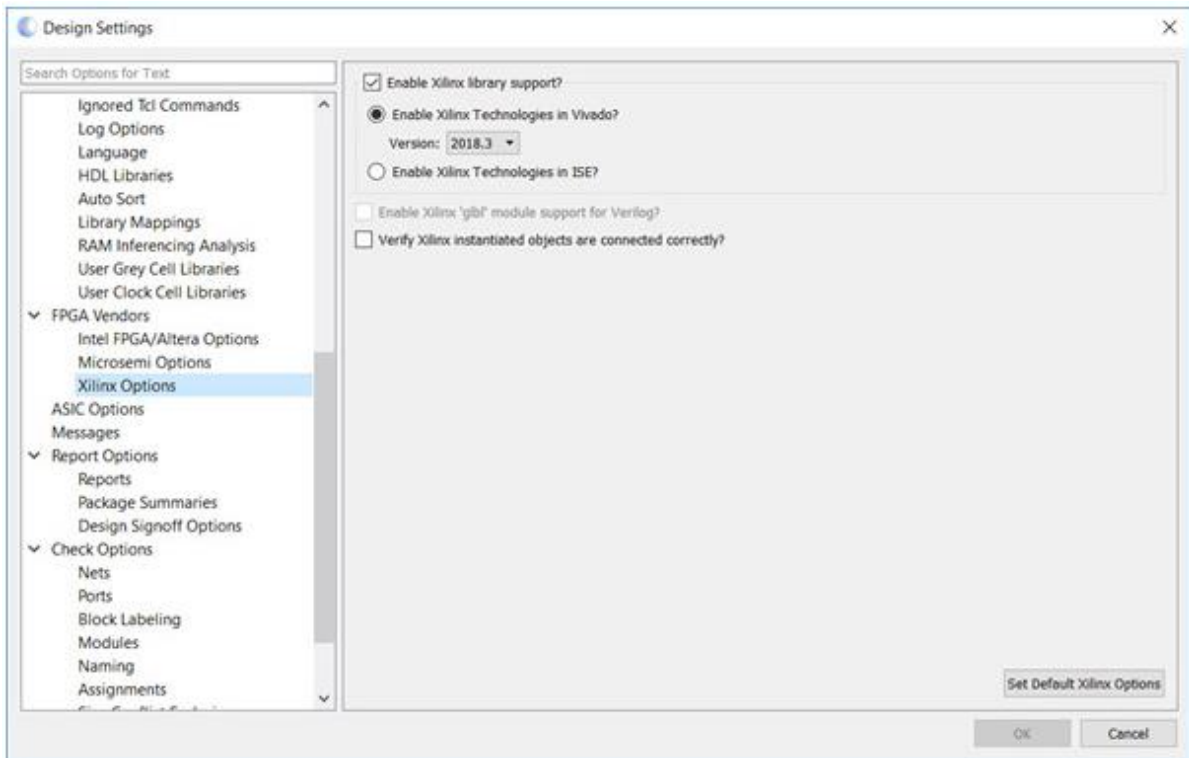


Figure 11 : Design Settings – FPGA Vendors

CDC File

Only show improperly synchronized CDCs?

Synchronizer Type: [Dropdown] Source Net: [Dropdown] Source Clock: usr\_clk Source Clock Domain: [Dropdown] Dest. Net: n Clock Dest. Clock Domain: [Dropdown]

Show Unwaived CDCs  Show Waived CDCs Number of Matching CDCs: 6 of 110

Source Net	Source Clock	Source Clock Domain	Dest Net	Dest Clock
ddr3_ip_controller_1.app_addr	usr_clk	usr_clk_group	ddr3_ip_controller_1...	Unknown Clock
ddr3_ip_controller_1.app_wdf_data	usr_clk	usr_clk_group	ddr3_ip_controller_1...	Unknown Clock
ddr3_ip_controller_1.t_app_cmd	usr_clk	usr_clk_group	ddr3_ip_controller_1...	Unknown Clock
ddr3_ip_controller_1.t_app_en	usr_clk	usr_clk_group	ddr3_ip_controller_1...	Unknown Clock
ddr3_ip_controller_1.t_app_wdf_end	usr_clk	usr_clk_group	ddr3_ip_controller_1...	Unknown Clock
ddr3_ip_controller_1.t_app_wdf_wren	usr_clk	usr_clk_group	ddr3_ip_controller_1...	Unknown Clock

Figure 12 : BPS CDC Analyse Results: After XPM Instantiation

### 3. Results



While the problems listed above led to iterations in simulations and timing violations, they also caused long debug times in the lab environment. The total time spent for the three cases is shown in the table below. As can be seen, the identification of missing constraints improves the timing produced by the synthesis tools and the optimizations related to the place & route. Total time spent for timing optimizations of the relevant module in the project is reported as 2 worker/week as in the table. The time to solve the uninitialized FFs problems in the simulation and setup environment is reported as approximately 2 worker/week. Since clock domain crossing problems cause unstable conditions in the laboratory environment, it took about 3 worker/week to detect these problems (and we were lucky to detect them, usually it will not be possible to detect them as those are metastability issues and random).

With this table, it is clear that when Blue Pearl’s Visual Verification Suite is used effectively, a gain of approximately 7 worker/week can be obtained for a project of this density. The project was completed with 108 worker/week when the suite was not used, where it could have been completed in 101 worker/week when used.

The time gain will be much more in case of a functional safety design like DO-254 as there will be more things to be checked there.

Table 1 : Problem Fixing Time -Table

Case	Results	Effort Saving
Case-1: Uninitialized FFs	<ul style="list-style-type: none"> <li>• Long Iterations in simulation environment</li> <li>• Requires Hardware Debug</li> </ul>	2 worker/week
Case-2: Multicycle path deficiency	<ul style="list-style-type: none"> <li>• Long Iterations to meet timing</li> </ul>	2 worker/week
Case-3: Clock Domain Crossing Issues	<ul style="list-style-type: none"> <li>• Requires Hardware Debug</li> </ul>	3 worker/week

## References

- [1] <https://www.BluePearlsoftware.com/hdlcreator/>
- [2] ‘Basics of Multi-Cycle and False Paths’, 08.07.2014, <https://www.edn.com/basics-of-multi-cycle-false-paths/>
- [3] Preetam, Isukametla & Mazumder, P. & Kumar, T. & Krishna, S & Kumawat, Renu. (2015). Design and verification of Ethernet, VME IP core using ACE and CDC. 194-198. 10.1109/ECS.2015.7124891.
- [4] ‘Understanding Clock Domain Crossing Issues’ By Saurabh Verma, Ashima S. Dabare, Atrenta 12.24.2017, <https://www.eetimes.com/understanding-clock-domain-crossing-issues/#>