



Contribution ID: 141

Type: Oral

Improving FPGA Timing Closure via Automated Pipeline Placement

Thursday 9 October 2025 09:20 (16 minutes)

The growing capacity of high-end FPGAs enables more powerful algorithms in high-energy physics but introduces new challenges for firmware developers. The largest AMD devices, composed of multiple silicon dies (SLRs), face data transfer timing challenges due to Vivado's placer limitations in large designs. In particular, pipelined buses crossing SLRs often experience poor flip-flop placement, impacting timing and latency. We present a Python tool that automatically generates optimized placement constraints for pipeline registers, equalizing propagation delays of the stages to improve timing closure while minimizing latency, number of pipeline stages, and resource utilization.

Summary (500 words)

LHC experiments are immersed in upgrade projects involving the implementation of significant portions of data processing to top-range FPGAs, in order to achieve the required challenges of HL-LHC. The AMD/Xilinx UltraScale+ family offers substantial improvements over previous generations, providing higher logic switching speeds (enabling increased data processing throughput) and expanded resource availability in its largest devices. One of these devices is the Virtex UltraScale+ VU13P FPGA, which is being extensively used in many CMS system upgrades, such as the first stage of Level-1 trigger generation for the CMS Muon Barrel.

However, implementing such large designs poses significant challenges, one of which is achieving reliable, low-latency data transfers across Super Logic Region (SLR) boundaries. To mitigate this, many systems impose restrictive requirements to ensure signals are processed within the same SLR as their associated gigabit transceivers (GTYs). This leads to increased complexity and higher costs in the optical fiber infrastructure, with patch panels designed to map interconnections at single-link granularity, rather than at the more practical board-level granularity. In our experience, Vivado's placer struggles with FF-pipelined buses in large designs (over 500k FFs), often clustering pipeline stages together or even introducing additional SLR crossings, resulting in timing failures on the longest nets (as illustrated in figure 1).

Contrary to widespread perception, we have observed that data can be transferred across SLRs with minimal latency impact in these devices. In figure 2, we show the minimum achievable latency (a) and maximum clock frequency (b) for pipelines moving data across three SLR crossings, both vertically (blue) and diagonally to the opposite corner of the FPGA (orange). As it can be seen, even the most extreme lengths can be covered with high throughput buses while keeping the latency under 15 ns, which could be compensated by the reduction in fibre optic length if system complexity is reduced. The availability of Super Long Lines (SLLs) should not pose a limitation either, as the total combined input and output bandwidth of all VU13P GTY transceivers (8.4 Tbps) could be accommodated, in the worst-case scenario, by the existing SLL resources when operated at 400 MHz.

To address placement limitations, we have developed a Python tool that automatically generates placement constraints for intermediate FFs in pipelined buses. The tool accounts for propagation delays of various routing resources (both intra- and inter-SLR) and generates a TCL constraint file containing Pblock assignments that equalize the propagation delays of pipeline stages. This approach minimizes the number of stages required, reducing both bus latency and resource utilization.

We believe adopting this methodology at the current stage of the LHC upgrade projects could significantly impact fiber mapping strategies, resulting in reduced system cost, complexity, and required rack space.

This tool is latest feature added to the software suite that we are developing, that allows to address the challenges of achieving top timing performance in the FPGA by ensuring an optimal placement of the modules during the implementation process. This tool provides improved performance with respect to the native AMD/Xilinx existing software.

Author: NAVARRO TOBAR, Alvaro (CIEMAT - Centro de Investigaciones Energéticas Medioambientales y Tec. (ES))

Co-authors: SASTRE ALVARO, Javier (CIEMAT - Centro de Investigaciones Energéticas Medioambientales y Tec. (ES)); PAZ HERRERA, Rolando (CIEMAT - Centro de Investigaciones Energéticas Medioambientales y Tec. (ES))

Presenter: NAVARRO TOBAR, Alvaro (CIEMAT - Centro de Investigaciones Energéticas Medioambientales y Tec. (ES))

Session Classification: Logic

Track Classification: Programmable Logic, Design and Verification Tools and Methods