RD27 Internal Note ...
26 July 1995

O. Buyanov

CENTRAL TRIGGER PROCESSOR:

Working Paper on Critical Components Design for the RD27 Project.

CERN, ECP Division, Geneva, Switzerland

Introduction

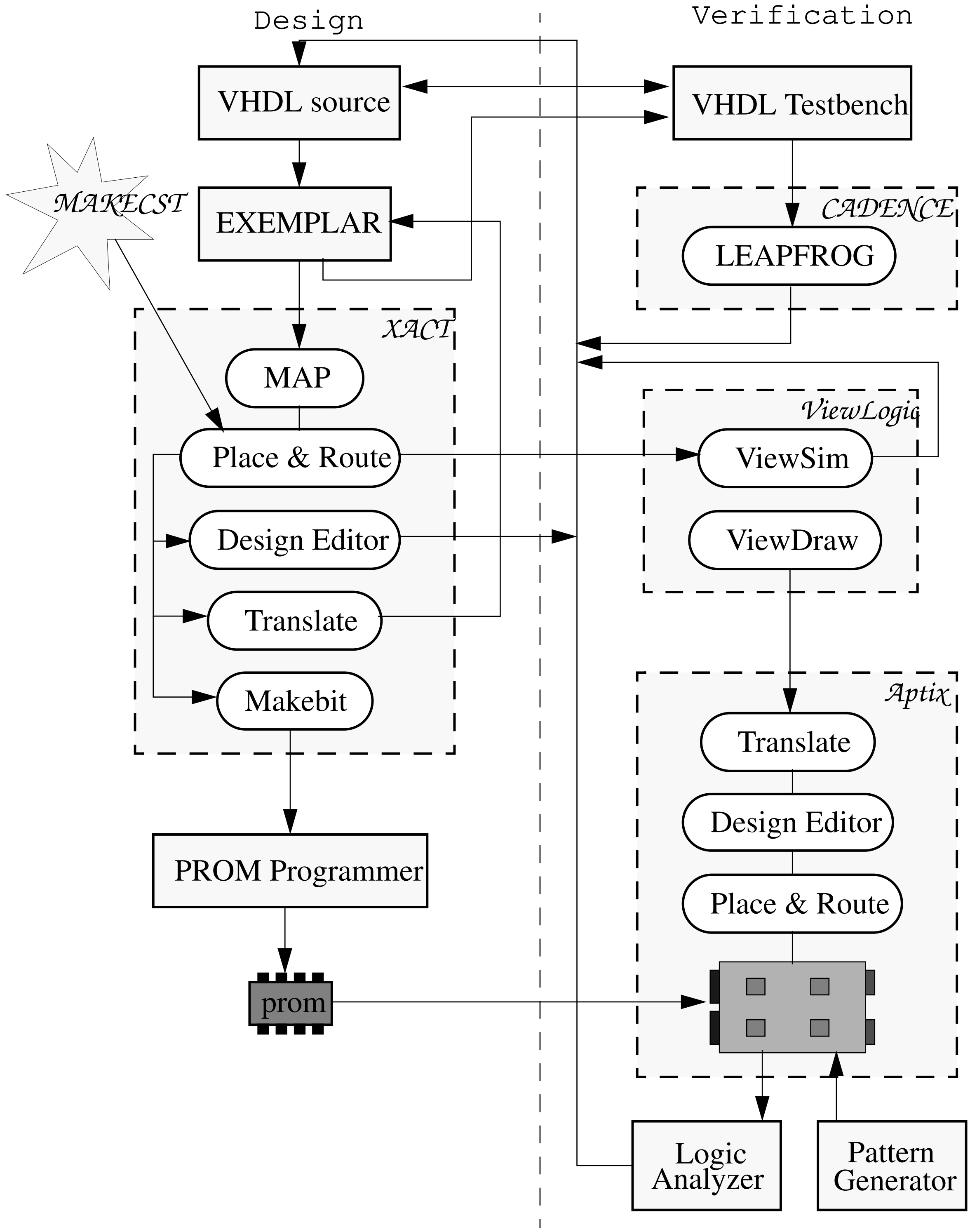
The work on critical components design for the Central Trigger Processor has been done for the RD27 Project (1, 2) during the period from 1.07.94 to 26.07.95.

All files referenced in this paper can be found into the directory /user/buyanov/CTP at CAE_SUN cluster.

1. FPGA Design Flow.

The whole design and verification flow used for critical components design is shown on Fig. 1.

Figure 1: FPGA Design Flow.



Its design part contains the following processes:

1. Synthesis of a VHDL source by EXEMPLAR synthesizer into XILINX netlist format.
2. Standard design flow for XACT development system.

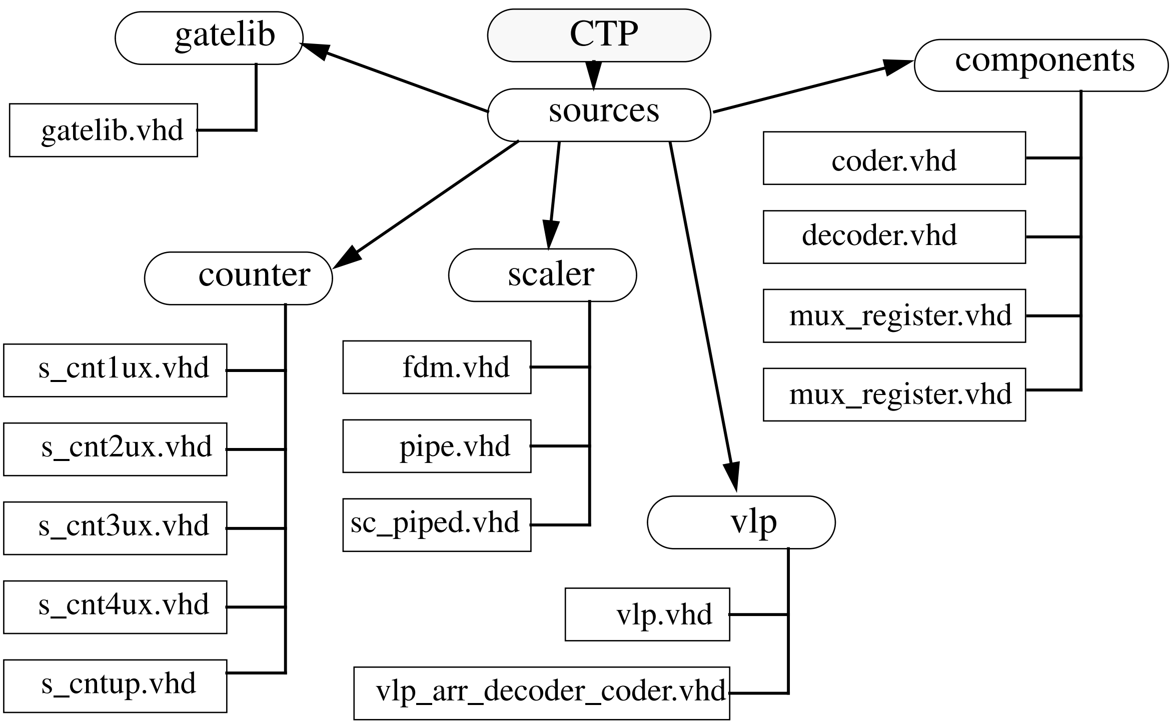
It might be efficient in some cases to use the special program called MAKECST (v1.0 Feb. 1995 (c) by O.Buyanov) to perform automatic placement of multiple parts of the design as an addition to the standard XACT tools.

Duringdifferent stages of the design it is important to verify the validity of the current step. The useful option is the ability to translate from the netlist which comes from EXEMPLAR or may be obtained from XACT using LCA2XNF for instance, to the intermediate VHDL code. This is offered by EXEMPLAR and resulting gate-level code might be simulated using the same test-bench VHDL model as for the initial source code. A problem is that during the synthesis process it appears to be difficult to control the correctness of the synthesis result by referring only to the brief EXEMPLAR's report. The postponed verification on the later stages (using ViewSim) in case of wrong synthesis results may lead to a significant waste of time.

Hints:

1. For post-synthesis simulation the mode "REFORMAT" should be used in EXEMPLAR. INPUT DESIGN: Name.xnf, OUTPUT DESIGN: Name.vhd; INPUT FORMAT: XNF, OUTPUT FORMAT: VHDL.
2. For the compilation of the code by LEAPFROG the simulation package EXEMPLAR.VHD should be used and compiled first. This package can be found in $EXEMPLAR/DATA.

Figure 2. CTP Sources Directory Structure.



At Fig.2 the sources directory structure is shown. For a consistency reason it is strongly recommended to keep this structure and have only one place for each source file rather having several copies of commonly used files in different design directories. If it is neccesary, a better solution seems to have symbolic links to the only source file from different locations.

2. SCALER

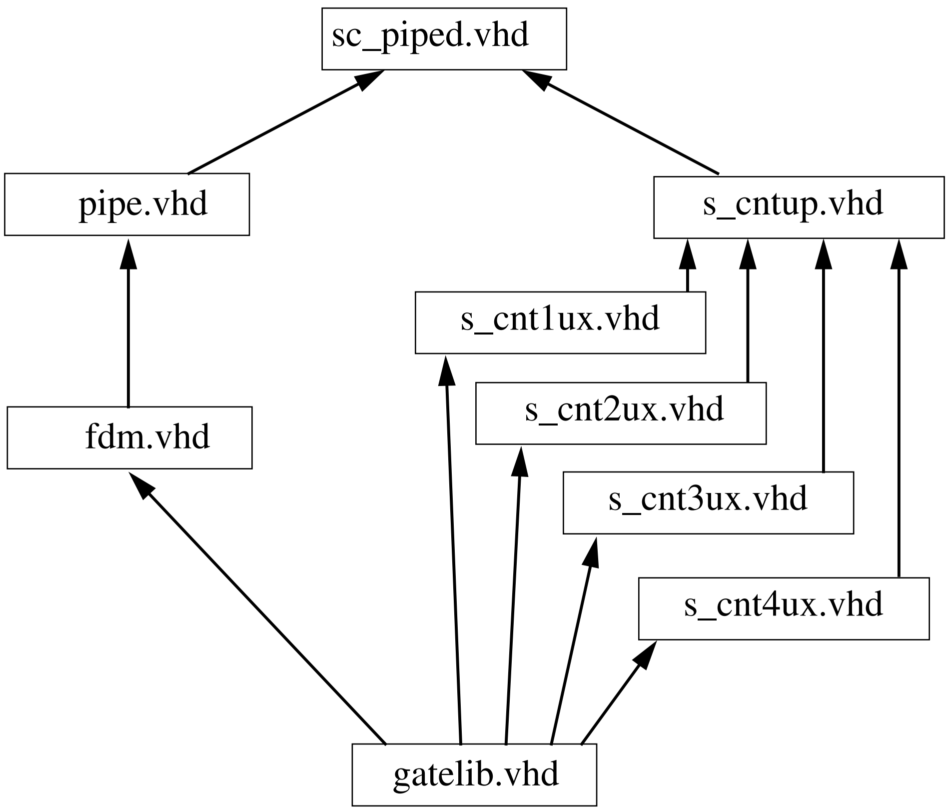
SCALER design is one of the most complex critical components and exploits all varieties of available technics to increase the perfomance characteristics. To achieve higher perfomance and density the following constraints have been used on different steps of the design flow:

1. Structural constraints
2. Placement constraints
3. Timing constraints.

While placement and timing constraints are quite usual terms for synthesis and development tools used, the name “structural constraints” has to be clarified. For the “scalable” style of the current design it is true, that depending on the chosen technology (target device) the counter, as a main part of a scaler, may be built very efficiently for a very limited size. For XILINX FPGAs of 3000 series this limit is only 3 bit. When the size is growing, the perfomance is decreasing very rapidly. To solve this problem the structure of the resulting counter is limited to have only 3-bit slices or less and it is called here “structural constraints”.

The file hierarchy for SCALER design is presented at Fig.3.

Figure 3. Design Hierarchy for SCALER Synthesis.



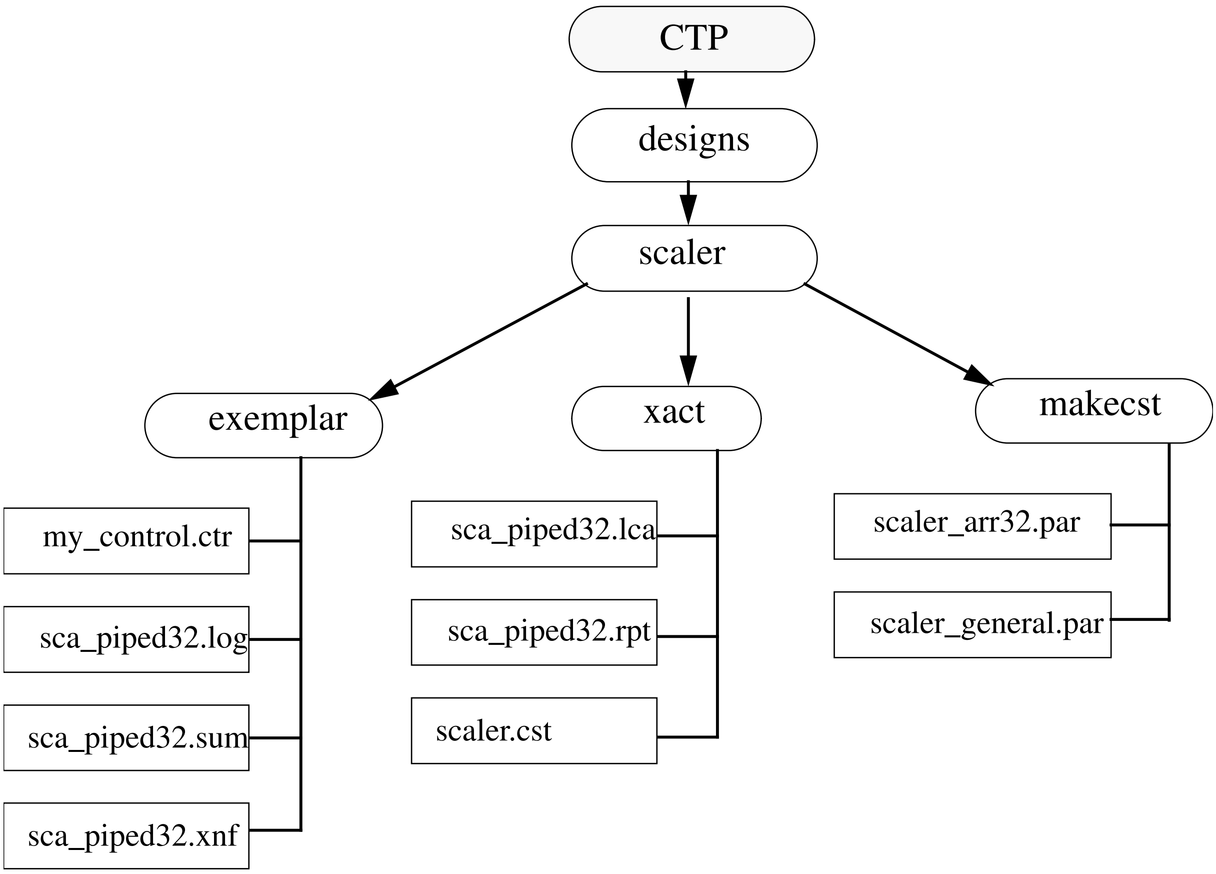
At the bottom of hierarchy we can see a gate level library **gatelib.vhd,** which is also an element of “structural constraints”. After investigation of the behaviour of the synthesis tool, and due to the limited information available as support documentation for synthesis using EXEMPLAR, it is found that for more predictability/reliability of synthesis results it would be better to have a library of elements for a specific design and on this basis create a technology independent design, hiding the technology dependance in the library.

Nevertheless, the current SCALER description is far away from the idea of complete technology independance. Some additional improvement of the code will be needed to achieve this goal. For the time being, the SCALER described on the very low level and dependance may appear even on the COUNTER level, assuming that 3 bit slicing is good enough probably only for XC3xxx technology.

Hints:
A further point for the “architectural constraints” issue may be found in the fact that the very limited control on the optimization process in the synthesis tool (EXEMPLAR) leads to the perfomance loss already during the synthesis phase. For such cases, if the tool appears to be “too smart” we must tell it: “Don’t optimize” and profit from it. It is found that it attempts (unefficiently) to optimize (or even treat) structures like: XOR2, AND5 etc.

For practical usage, all neccessary information related to the design stages can be found in the design directory (Fig. 4.) as well as the resulting netlists and design files.

Figure 4. Scaler Design Directory Structure.

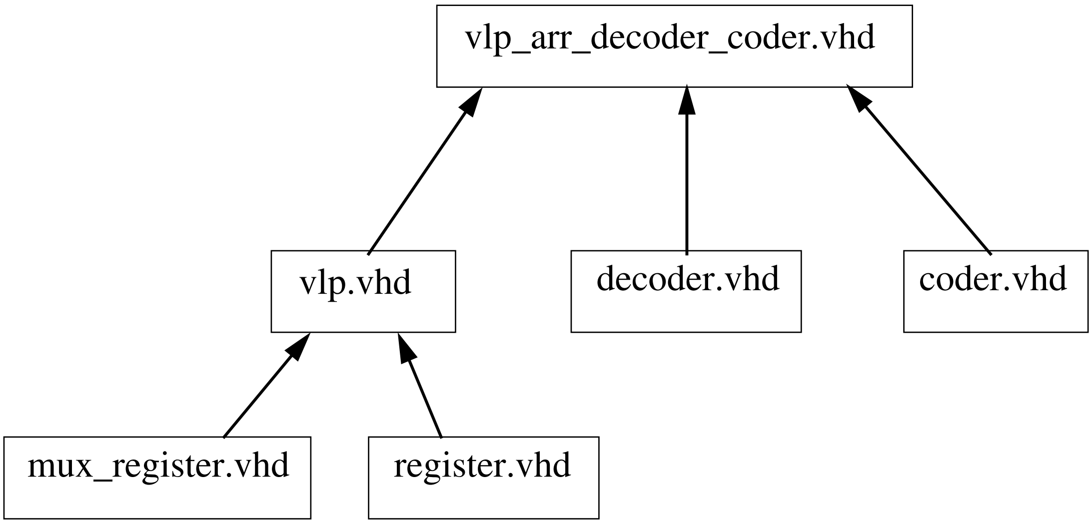


Synthesis phase switches and presettings are stored into the logfile in the EXEMPLAR directory. For the XACT design tool, the constraint file, needed on the Place&Route stage is placed in the XACT directory. This file is generated by MAKECST tool (Appendix 1), its source description to produce the constraint file can be found in the MAKECST directory.

3. Variable Length Pipeline

The Variable Length Pipline (VLP) design is based on the mixed behaviour/RTL style of component description. For the VLP, which is rather simple and regular shift-register type device this style produces satisfactory perfomance characteristics.

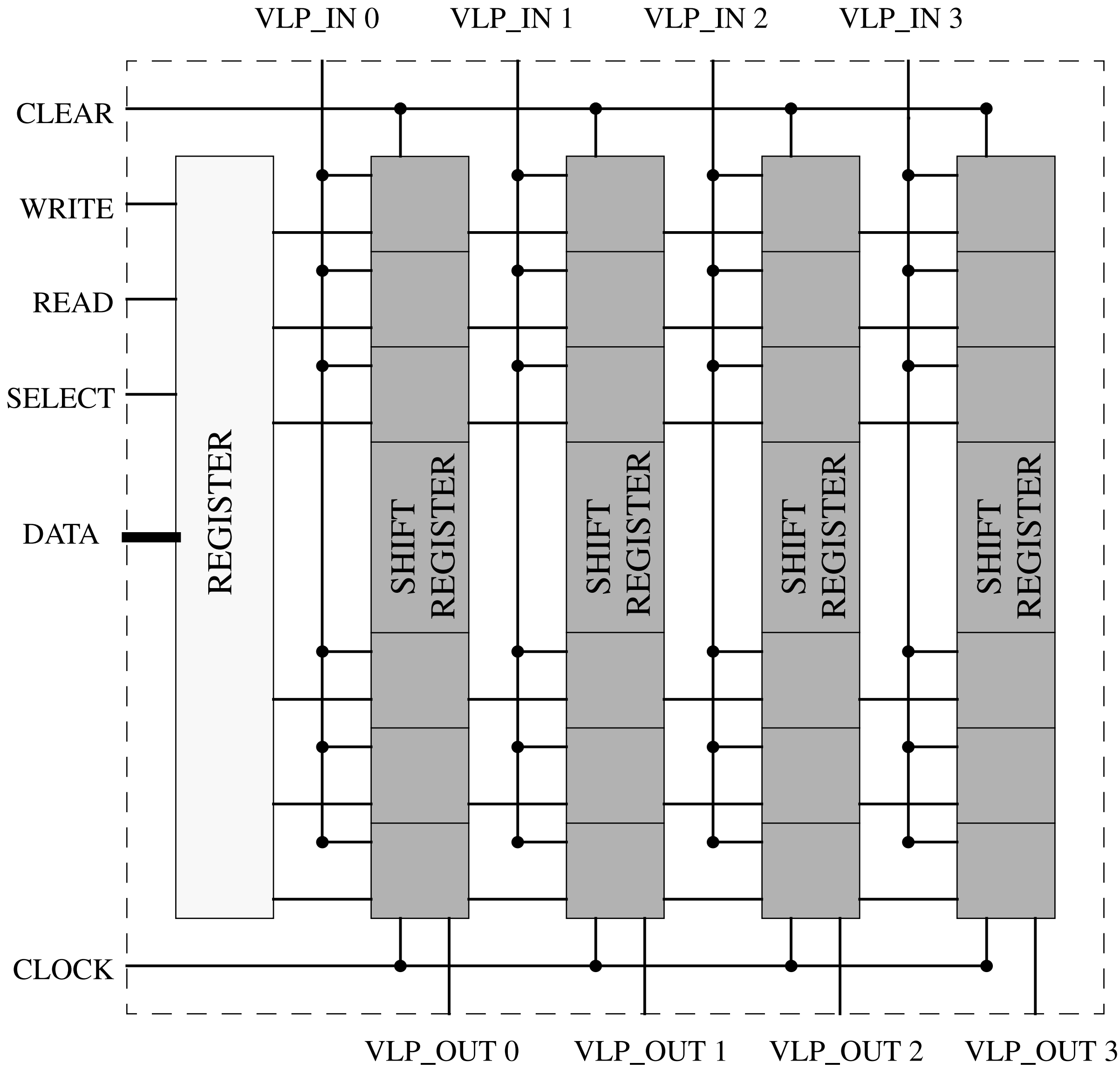
Figure 5. Design Hierarchy for VLP Synthesis.



The VLP description is represented by a three-level hierarchy of components (Fig. 5). The VLP itself is built of shift registers which allow the loading of input data into any stage of the register with an individual control signal for load (Fig. 6). This solution makes it possible to avoid multiplexing of input or output signals and allows to create a real generic structure, fully synchronous, and of high perfomance, independent of the VLP length and minimum “length” less or equal to 1 Clock period.

The four shift registers are composed of 4 bit VLP slices with an individual “length value” register. This means that all 4 bits of input data for a slice will be delayed by the VLP for the same number of Clock periods.

Figure 6. VLP Layout.



The VLP interface, shown on Fig. 7allows both read and write data to each 4 bit slice register through the internal bus, containing the number of bits equal to the length of the VLP. Such a solution minimizes the number of decoders, required otherwise for each “length register” in case of binary storage value. The only cost for it is one additional encoder, needed for the readout of the stored value. One more decoder is used to decode the address of a choosen individual slice register to interact with.

A VLP design directory is built similar to the SCALER directory. Since the design process doesn’t involve the detailed placement constraints, it is not required to use the MAKECST utility.

Figure 7. VLP Control Interface

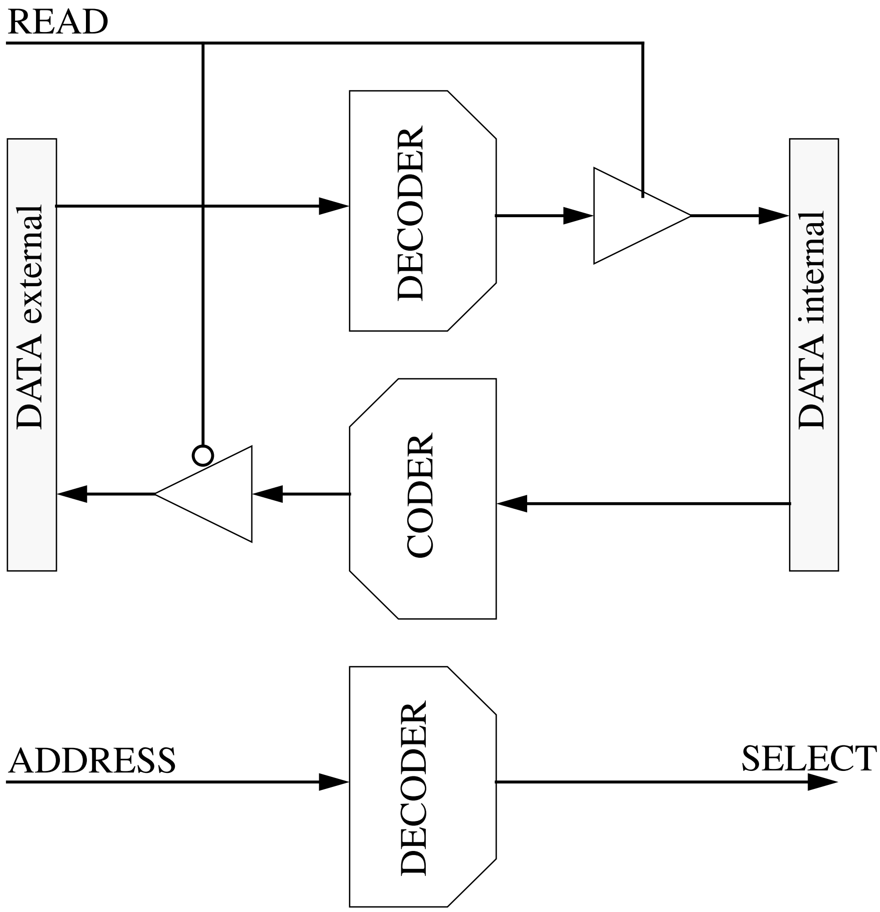
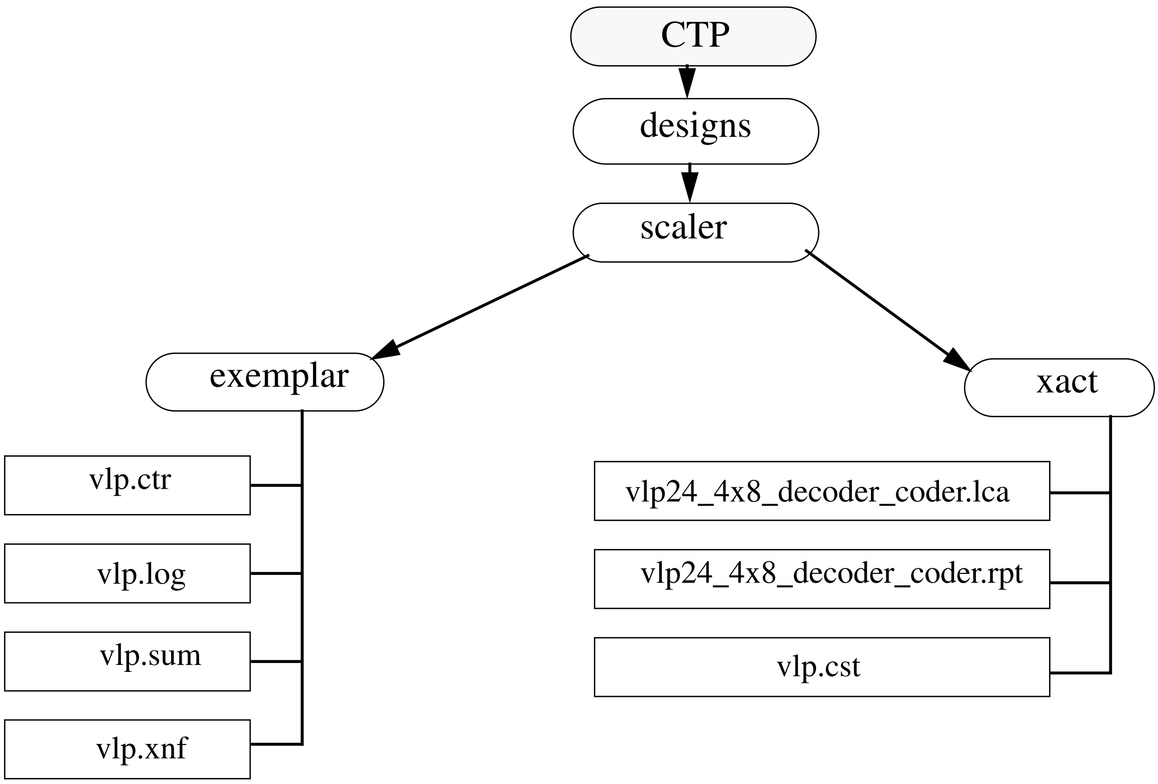


Figure 8. VLP Design Directory Structure.

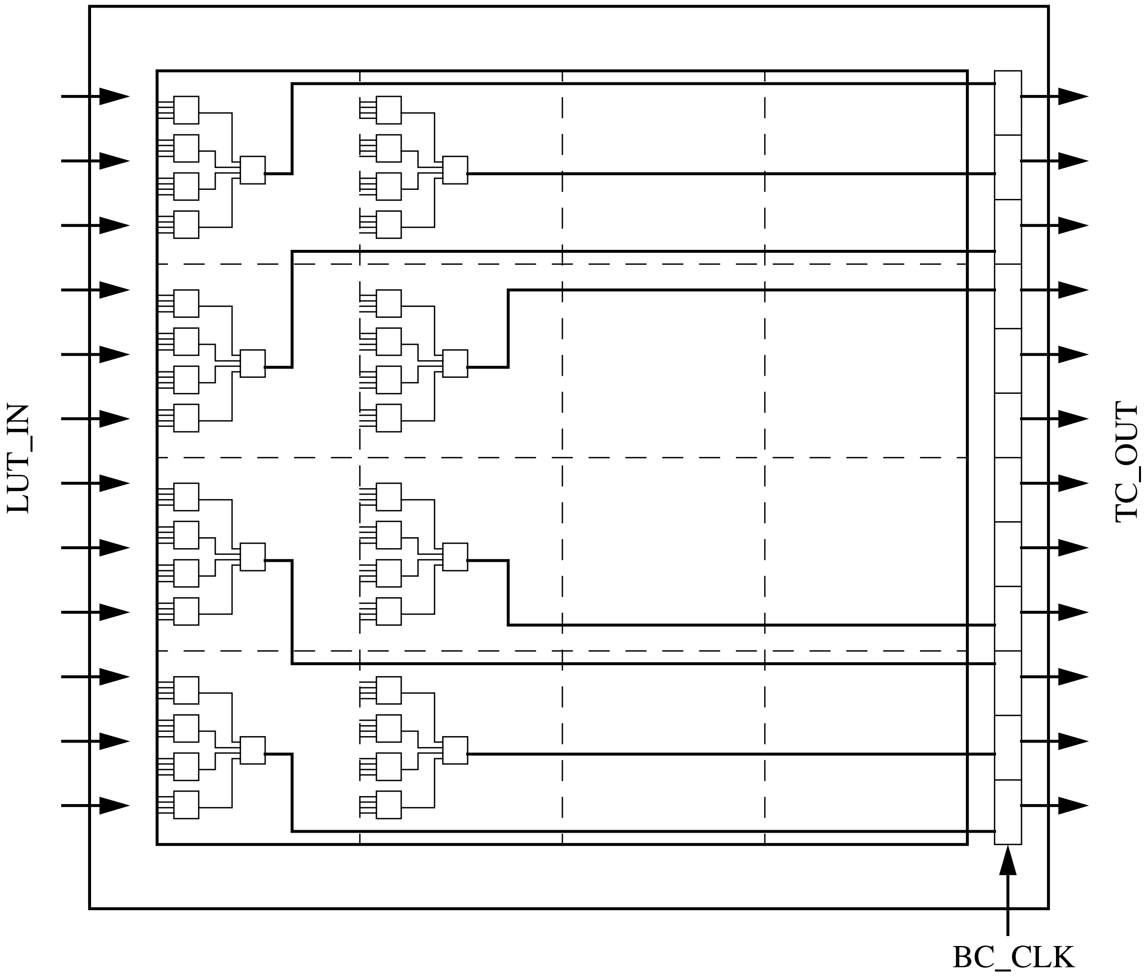


4. Look-Up Table

As a possible solution for the Look-Up Table (LUT) implementation using XILINX FPGA technology the following idea may be taken into consideration.

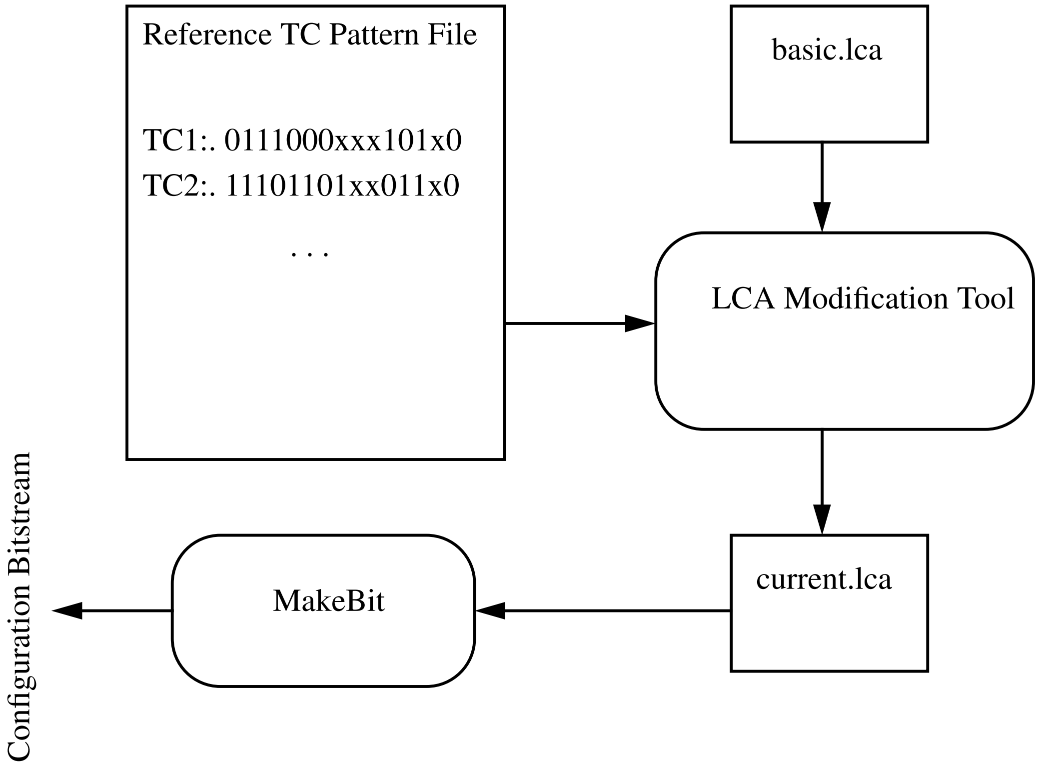
The design contains several identical blocks (Fig. 9) described as an appropriate simple logic functions (eg. AND32). The routing is then optimized to achieve the needed range for the total logic latency, then the routing is fixed. The resulting xxx.lca file than may be used as basis for future modifications according to the logic function for each Trigger Candidate (TC).

Figure 9. Possible Implementation of LUT.



Bearing in mind that actual logic block implementation in the given technology is based on the LUT technique, we can built a real Look-Up Table for each TC. A special LCA Modification Tool is needed to produce the final LUT design file avoiding the long and unpredictable compilation phase using simple text translation technique (some software work is needed). This resulting yyy.lca design file than may be easily transformed into the FPGA configuration bitstream with the standard Makebit utility (Fig. 10).

Figure 10. LUT Programming.



5. PRESCALER

The PRESCALER design is similar to the SCALER design. The PRESCALER design hirarchy is shown at Fig. 11. The next picture (Fig. 12) shows the design directory contents. The PRESCALER design is based on the loadable counter with feedback, described in XAPP 003 (1). With the current version of the FPGA development tool (XACT 5.00) it is not likely to obtain the perfomance, which satisfies the current requirements. The main obstacle is the lack of the mapping control, and therefore it is most likely to produce the appropriate PRESCALER design, basing on the sources developed, with the next version of the XACT Development System.

Table 1: CTP Critical Components Design Results

DESIGN	SIZE	LEVEL	DEVICE	#CLB	%USED	SPEED	TIME
Scaler	32x12	GATE+RTL	3195APC84-3	449/484	92%	63.7 MHz	1:00
VLP	(8x4)x24	BEH+RTL	4013PG223-4	480/576	83%	60.2 MHz	1:49
Prescaler	2x12	GATE+RTL	3142APC84-3	56/144	38%	43.3 MHz	0:02

Figure 11. Design Hierarchy for PRESCALER Synthesis.

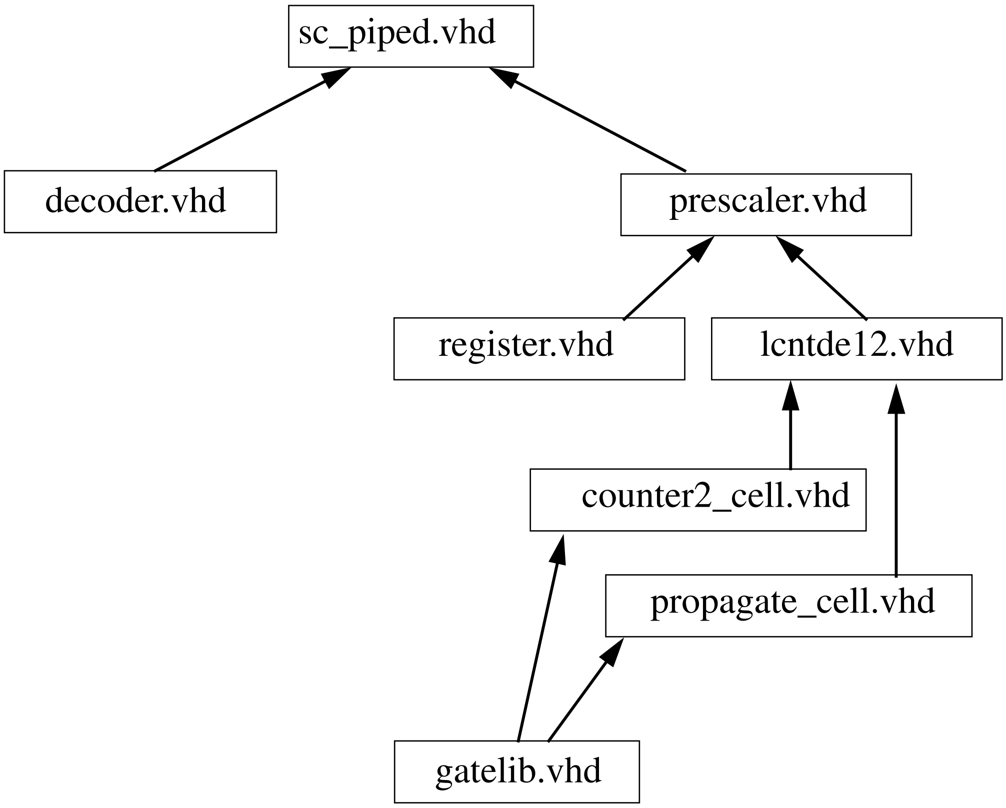
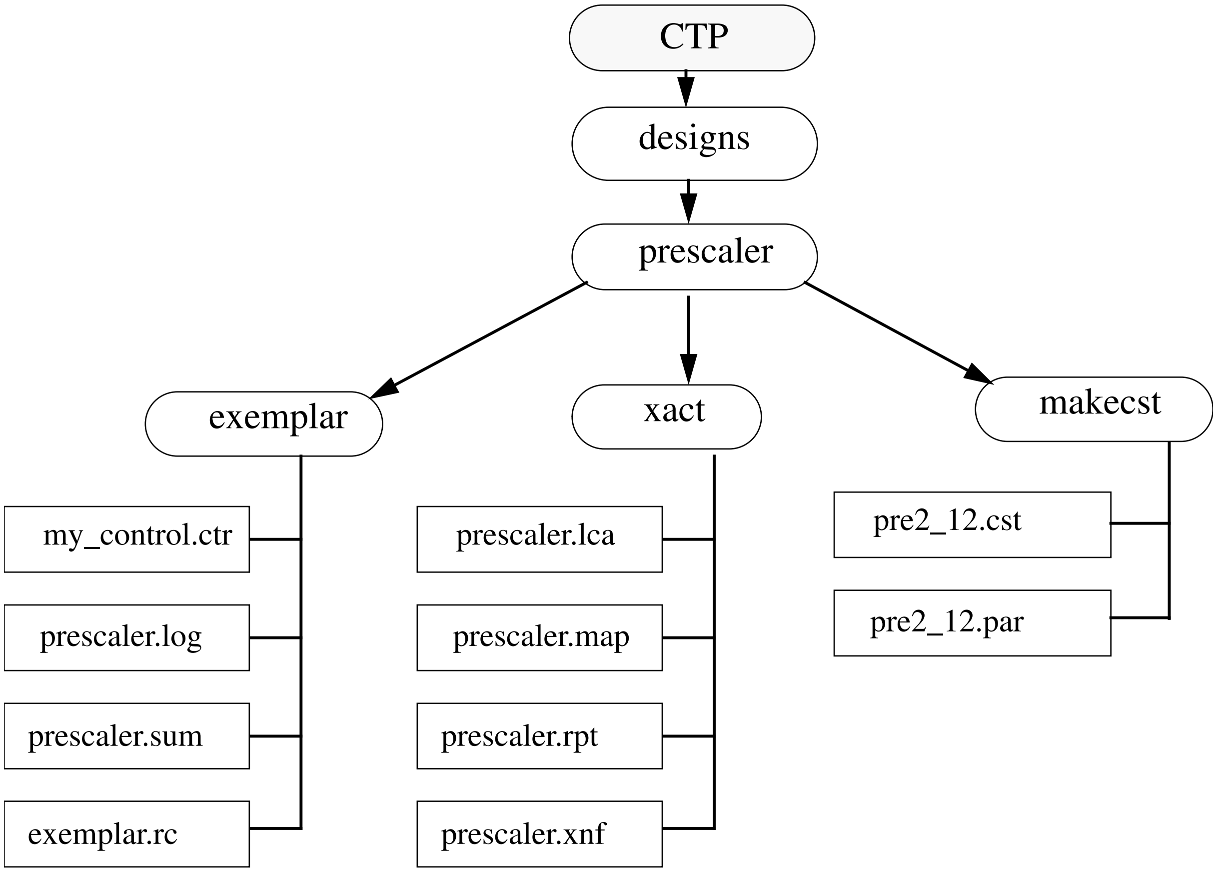


Figure 12. PRESCALER Design Directory Structure.



6. SUMMARY

The resulting characteristics obtained from the design of three CTP's Critical Components are shown in Tab. 1. The initial requirements for the perfomance of the components limited with 40 MHz BC Clock frequency is achived with a good 50% safety margin, which shows that the XILINX FPGA technology is a good candidate to build the CTP.

Acknowledgement

The author thanks H.Wendler for the help in the work on CTP and preparation of this paper.

References

1. RD27 Status Report, CERN/DRDC 94-D52, 9 Jan. 1994.
2. First Level Central Trigger Processor for LHC Experiments, By C.Jacobs et al., RD27 note 39 / ATLAS Internal Note DAQ-NO-033, 16 Dec. 1994.
3. The Programmable Logic Data Book, Xilinx Inc., San Jose, California, USA, 1994.