FROM QUARTUS TO VPR: CONVERTING HDL TO BLIF WITH THE TITAN FLOW

Kevin E. Murray, Scott Whitty, Suya Liu, Jason Luu, Vaughn Betz

Department of Electrical and Computer Engineering, University of Toronto, Ontario, Canada
email: {kmurray,jluu,vaughn}@eecg.utoronto.ca, {scott.whitty,suya.liu}@mail.utoronto.ca

ABSTRACT

Realistic benchmarks are important for FPGA Architecture and CAD evaluation. This paper provides a demo illustrating how designs described in HDL can be converted to BLIF using the Titan flow, and used in academic CAD tools.

1. INTRODUCTION

One of the challenges of FPGA CAD and architecture research is utilizing realistic benchmark designs. This has traditionally been a difficult process, typically requiring significant modification to the original design. The Titan flow [1] enables real designs to be synthesized using Altera's Quartus II, and converted to the standard Berkeley Logic Interchange Format (BLIF), commonly used by academic FPGA logic synthesis and physical design tools.

This paper is organized as a tutorial illustrating how the HDL to BLIF conversion is performed within the Titan flow.

2. INITIAL SETUP

Download the demo archive from the Titan page at `http://www.eecg.toronto.edu/~vaughn/software.html` then create the environment as shown in Listing 1.

Listing 1: Creating Demo Environment

```
$ tar -xzf titan_hdl_to_blif_demo.tar.gz #Extract
$ cd titan_hdl_to_blif_demo/vqm2blif/
$ make #Compile vqm2blif
$ cd ../bitcoin_small/ #Move to design directory
$ ls
bitcoin_small.qpf  bitcoin_small.qsf  hdl
```

3. VQM GENERATION

To generate BLIF, a Verilog Quartus Map (VQM) file must first be produced by Quartus II.

To enable VQM generation, hidden variables must be added to Quartus settings file (`.qsf`) *before* synthesis is performed. This is shown in Listing 2 for the Stratix IV family.

Listing 2: Assignments added to the `.qsf` for VQM Generation

```
set_global_assignment -name INI_VARS
    "qatm_force_vqm=on;vqmo_gen_sivgx_vqm=on"
```

The VQM is generated by first synthesizing and merging the design, and finally writing out the VQM. The commands are shown in Listing 3.

Listing 3: Synthesizing & Generating the VQM

```
$ quartus_map bitcoin_small
$ quartus_cdb bitcoin_small --merge
$ quartus_cdb bitcoin_small --vqm=bc_small.vqm
```

In some cases a design (such as `bitcoin_small`) contains encrypted IP. The offending IP can be identified in the Quartus `.map.rpt` file, as shown in Listing 4, and must be removed to generate a plain-text VQM.

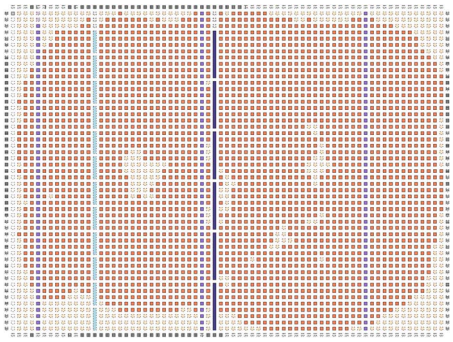

Fig. 1: Final VPR placement of the `bitcoin_small` design.

Listing 4: Identifying Encrypted IP

```
$ grep 'Encrypted' bitcoin_small.map.rpt
sld_rom_sr.vhd ; yes ; Encrypted Megafunction
sld_hub.vhd ; yes ; Encrypted Megafunction
```

For the `bitcoin_small` design, a modified top level HDL file that removes the encrypted IP has been provided. The top level of the design can be modified by changing the `TOP_LEVEL_ENTITY` assignment in `bitcoin_small.qsf` to `top_unencrypted`. Repeating Listing 3 should now generate an unencrypted VQM.

4. VQM TO BLIF CONVERSION

The plain-text VQM can now be converted to BLIF, using the `vqm2blif` tool, as shown in Listing 5. Advanced usage is described in `vqm2blif`'s documentation.

Listing 5: VQM to BLIF conversion

```
$ ../vqm2blif/vqm2blif.exe -vqm bc_small.vqm
     -arch ../test_arch.xml -out bc_small.blif
```

5. RUNNING VPR

The generated BLIF file can now be used in academic CAD tools that read BLIF. Figure 1 shows the placement generated by VPR [2], created with the command shown in Listing 6.

Listing 6: VPR command to generate Figure 1

```
$ vpr ../test_arch.xml bc_small.blif
    --timing_analysis off
```

6. CONCLUSION

In conclusion, we have shown how to convert a circuit from HDL to BLIF using the Titan flow, and used it in an academic CAD tool. Quartus II synthesized the HDL and generated a VQM file which was converted to BLIF, with the `vqm2blif` tool. This flow offers an easy way to bring realistic designs into academic CAD flows with minimal modifications.

REFERENCES

[1] K. E. Murray, S. Whitty, S. Liu, J. Luu, and V. Betz, "Titan: Enabling large and complex benchmarks in academic CAD," in *FPL*, 2013.

[2] V. Betz and J. Rose, "VPR: A new packing, placement and routing tool for FPGA research," in *FPL*, 1997, pp. 213–222.