OASIS® Application to Re–Synthesis of FPGA Designs as Standard Cell ASICs

Krzysztof Kozminski

MCNC Center for Microelectronics, P.O.Box 12889, 3021 Cornwallis Road Research Triangle Park, North Carolina 27709–2889
e–mail: kk@mcnc.org, tel. (919)–248–1969

Abstract

This paper presents the ongoing work on the standard cell–based silicon compiler OASIS. A recent extension of the OASIS’ capabilities permits the designer to migrate FPGA designs into standard cell technology. Performing FPGA re–synthesis part of OASIS on real–life designs reinforces the advantages of design capture on algorithmic level over the most commonly used schematic capture design entry.

FPGA Design Entry Into OASIS

OASIS is a cell–based design system with four major components illustrated in Fig. 1. The design is captured in Logic–III, a Pascal–like programming language that combines both functional and structural descriptions. Logic synthesis algorithms are used to transform functional descriptions of finite state machines and decoding logic into circuit structures. Data–path circuits are assembled hierarchically from the descriptions of subcomponents [1]. The resulting hierarchical netlist can be simulated, used to determine the testability of the circuit and generate tests, and to synthesize the circuit layout.

Some of the current work on the OASIS development concentrates on adding new interfaces to the existing system. Among others, an effort is being made to provide designers with the capabilities to:

• implement OASIS designs as FPGAs, and
• convert existing FPGA designs to the standard cell–based designs.

The motivation behind the above goals is to facilitate use of OASIS in a popular methodology of implementing circuit prototypes as Field–Programmable Gate Arrays (FPGA) before committing the design to costlier semi–custom design. The FPGAs used in this process usually are either Xilinx™ or Actel FPGA chips.

Fig. 1 Data flow in the OASIS System

This paper describes the methodology and implementation of the process used for re–synthesis of the existing FPGA designs using Xilinx chips.

Data Flow for FPGA Re–Synthesis

The software tools used to link FPGA re–synthesis to the OASIS System and the data–flow among them are shown in Fig. 2. The design flow was designed to maximally use the existing tools already implemented in other subsystems of OASIS. The only new software tool required for the task was the XNF netlist assembler/pruner (XNF is the standard format used to describe Xilinx FPGA designs [2]).

The OASIS’ data flow template was readily extended to incorporate the relations from Fig. 2. The OASIS’ data flow manager DECOL could be used without modifications [3].

The task of the front–end program enabling OASIS to read Xilinx netlists is more than a simple netlist conversion. Since the number of gates that can be used in a design implementable as a single FPGA is relatively small, we are typically dealing with multi–chip FPGA systems.
which can be easily fabricated as a single standard cell IC. Thus, the XNF netlist assembler creates a hierarchical description of a digital circuit, the parts of which correspond to single FPGA chips. The connectivity among the component parts is extracted by comparing names of the I/O signals of the individual chips, based on the following paradigms:

- a signal occurring in two or more component chips is internal to the system, and
- a signal occurring in only one of the components is an I/O signal to/from the system.

The process determining the system I/O signals can be influenced by additional connectivity data supplied by the user.

The output of the XNF netlist assembler is a Logic-II code which is the usual input format used by OASIS. In this circuit description, each of the component chips is further subdivided into a part containing all sequential elements and a part containing all combinational elements. This enables separate processing of these parts without enforcing the OASIS' clocking methodology (two-phase, non–overlapping, non–gated clock).

The usual data flow of OASIS is amended with two additional processing steps for the combinational components extracted from the FPGA chips. Since typical front end of a FPGA design system is a schematic capture program [2], manually designed components may contain redundancies and the logic gates used in a design, while appropriate for a FPGA chip, may not be present in the standard cell library. Consequently, two steps are "borrowed" from the other parts of OASIS and used to optimize the FPGA designs without changing their functions. The first step is testability analysis of the circuit. If the design is found to contain any untestable (redundant) faults, these faults and the associated circuitry are removed using algorithms described in [4,5]. Typically, this produces a highly testable, irredundant circuit. This circuit is then implemented using modules available in the desired standard cell library [6]. For the purpose of the re-implementation experiments described in the next section, the redundancy removal stage may be bypassed to obtain direct technology mapping from the original design.

The net effect is that each FPGA chip is converted into a minimized, technology–mapped netlist which subsequently can be treated in the same manner as a netlist synthesized from behavioral specification during the normal OASIS operation. The standard OASIS' netlist assembler
is used to produce a hierarchical netlist corresponding to the central point in the OASIS data flow in Fig. 1; this netlist is then used for simulation, test generation and layout synthesis.

The future work will add additional optimization steps performing partitioning and re-synthesis of large combinational and sequential netlists [7,8].

Experiments.

The FPGA re-synthesis subsystem of OASIS was tested on several examples of digital systems implemented as Xilinx FPGA chips. Each chip was synthesized twice: with, and without the redundancy removal in the OASIS data flow. Typical results are shown in the tables below. The example used here was a system consisting of two FPGA arrays. The results in the two tables were obtained with two different technology mapping programs and two different target libraries.

<table>
<thead>
<tr>
<th></th>
<th>Direct technology mapping</th>
<th>Redundancy removal before technology mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>#gates</td>
<td>1873</td>
<td>1752</td>
</tr>
<tr>
<td>#nets</td>
<td>1889</td>
<td>1770</td>
</tr>
<tr>
<td>Delay</td>
<td>35.7</td>
<td>35.6</td>
</tr>
<tr>
<td>Design area</td>
<td>21.70 mm$^2$</td>
<td>19.95 mm$^2$</td>
</tr>
<tr>
<td>Fault Coverage</td>
<td>94.60%</td>
<td>99.43%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Direct technology mapping</th>
<th>Redundancy removal before technology mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>#gates</td>
<td>1192</td>
<td>1108</td>
</tr>
<tr>
<td>#nets</td>
<td>1308</td>
<td>1185</td>
</tr>
<tr>
<td>Delay</td>
<td>39.67</td>
<td>37.4</td>
</tr>
<tr>
<td>Design area</td>
<td>21.31 mm$^2$</td>
<td>19.89 mm$^2$</td>
</tr>
<tr>
<td>Fault Coverage</td>
<td>95.96%</td>
<td>99.47</td>
</tr>
</tbody>
</table>

The sample results point out to a weakness in the schematic-capture based design entry. Particularly, when testability is not of concern, designs may contain large portions of redundant circuitry, providing significant opportunity for use of redundancy removal and circuit optimization software.

Conclusions.

This paper presented a unique design path for direct migration of FPGA designs into standard cell-based IC chips. The implementation of the ideas presented here provides designers with a full power of the OASIS System, in particular with capabilities to reduce the circuit complexity, increase testability, quickly re-target designs into varying families of standard cells, optimize the circuit for minimal delay time or for minimal area.

References