Reconfigurability-aware Structural Mapping for LUT-based FPGAs

Karel Bruneel and Dirk Stroobandt
Department of Electronics and Information Systems
Ghent University
Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
{kbruneel, dstrooba}@elis.ugent.be

Abstract

In many applications, subsequent tasks differ only in a specific set of parameters. Because of their reconfigurability, FPGAs (Field Programmable Gate Arrays) can be configured with an optimized configuration every time these parameter values change. This results in configurations that are smaller and faster than their generic counterparts. Unfortunately, the overhead involved in generating these configurations at run-time with conventional tools is very large. However, if the incoming tasks only differ in a set of parameter values, the use of Tunable LUT (TLUT) circuits can drastically reduce this overhead. A TLUT circuit is a LUT circuit in which the truth tables of the LUTs are expressed as a function of a set of parameters. At run-time the truth tables for a specific set of parameter values can rapidly be calculated by evaluating these functions. Up to now TLUT circuits had to be designed manually resulting in a huge design cost. This paper introduces TMAP2, a software tool based on conventional structural mapping that automatically generates a TLUT circuit starting from an arbitrary Boolean circuit. We have tested TMAP2 on a set of 12 micro-benchmarks and we show a substantial reduction in both the circuits area and maximum depth compared to conventional implementations.

1 Introduction

Due to the inherent reconfigurability of SRAM-based FPGAs, the configuration of an FPGA can change at run-time, called *run-time reconfiguration* (RTR). RTR systems can use this property to execute an incoming task in three steps. First, an FPGA configuration that is optimized for executing the given task is generated. Second, this configuration is loaded in the FPGA and third, the task is executed by the FPGA. An RTR system is a heterogeneous systems containing at least an FPGA and a configuration manager (CM), generally a CPU. The CM is responsible for generating the optimized FPGA configurations and reconfiguring the FPGA.

Run-time reconfiguration is used because for some applications the number of resources needed to execute an incoming task (FPGA and CM resources) is smaller when exploiting RTR than using a conventional FPGA implementation. This is only true if the CM resources needed to generate an optimized configuration and reconfigure the FPGA are at least compensated by the reduction in FPGA resources.

For most applications the number of CM resources is too large and the RTR implementation is inefficient. However, if the incoming tasks only differ in a set of parameter values, the use of tunable LUT (TLUT) circuits can drastically reduce the CM resources [2]. A TLUT circuit is a LUT circuit in which the truth tables of some of the LUTs are expressed as a function of a set of parameters. We call this function the tuning function of the TLUT. The truth tables for a specific set of parameter values is easily found by evaluating the tuning functions.

Because the structure of a TLUT circuit does not change with the parameter values, placement and routing can be done at compile-time. Therefore, changing the parameter values at run-time only requires the CM to evaluate the tuning functions, which is extensively less computationally expensive than a full place and route. Moreover, since only the truth table bits have to be reconfigured the reconfiguration cost is also drastically reduced for FPGAs suporting partial reconfiguration.

Throughout this paper we will use a 4:1 multiplexer as example. The multiplexer has four data inputs (I_0, I_1, I_2 and I_3), two select inputs (S_0 and S_1) and one output (O). Figure 1 shows a TLUT circuit for the multiplexer where the select inputs are chosen as parameter inputs. The figure shows both the LUT structure, that is fixed for all possible sets of parameter values and the tuning functions, that express the truth tables of the LUTs as function of the parameters. The TLUT implementation uses only two LUTs to implement the multiplexer while a conventional FPGA