GAUT – A Free and Open Source High-Level Synthesis Tool

Philippe COUSSY, Ghizlane LHAIRECH-LEBRETON, Dominique HELLER, Eric MARTIN

Lab-STICC/CNRS – Universite de Bretagne-Sud– France
http://lab-sticc.fr/www-gaut

Abstract

GAUT is an open source High-Level Synthesis tool. From a bit-accurate C/C++ specification it automatically generates a RTL architecture described in VHDL that can be used by commercial logical synthesis tools like ISE (Xilinx), Quartus (Altera). GAUT also generates TLM and CABA SystemC simulation models for the SocLib virtual prototyping platform.

1. Introduction

In the SoCs context, the traditional IC design methodology relying on EDA tools used in a two stages design flow – a VHDL/Verilog RTL specification, followed by logical and physical synthesis– is no more suitable [1][2]. However, the increasing complexity and the data rates of DSP applications require efficient hardware implementations like dedicated accelerators or coprocessors. Thus actual SoC embedded DSP cores need new ESL level tools in order to raise the specification abstraction level up to the «algorithmic one » [3]. Algorithmic descriptions enable an IC designer to focus on functionality and target performances rather than debugging RTL. Designers spend more time exploring the design space with multiple “what if” scenarios. They obtain a range of implementation alternatives, from which they select the architecture providing the best power/speed/gate count trade-off. CatapultC from Mentor Graphics, Cynthesizer from Forte or PICO from Synfora are EDA software tools enabling to capture such C/C++/SystemC-based algorithmic design entries and synthesize them into an equivalent RTL specification. GAUT is an academic and open source HLS tool dedicated to DSP applications.

2. GAUT, a HLS tool

GAUT [4][5] takes as input a C/C++ description of the algorithm that has to be synthesized where Algorithmic C™ class library from Mentor Graphics can be used. This allows the designer to specify signed and unsigned bit-accurate integer and fixed-point variables by using bit accurate integer and fixed-fixed data types [6]. The mandatory constraints are the throughput (specified through an initiation interval II which represents the constant interval between the start of successive iterations) and the clock period. Optional design constraints are the memory mapping [7] and I/O timing [8][9]. The architecture of the hardware components that GAUT generates is composed of three main functional units: a processing unit PU, a memory unit MEMU and a Communication & Interface Unit COMU (see Figure 1). The PU is a datapath composed of logic and arithmetic operators, storage elements, steering logic and a controller (FSM). The MEMU is composed of memory banks and their associated controllers. The COMU includes a synchronization processor and an operation memory which allow to have a GALS / LIS communication interface [10].

As described in Figure 2, GAUT first synthesizes the Processing Unit. Then it generates the Memory Unit [7][11] and the Communication Unit [10][12]. During the design of the PU, GAUT initially selects arithmetic operators and after targets their best use according to the design constraints and objectives. Then GAUT processes the registers and memory banks, which are part of the memory unit. The register’s optimization, which is done before the memory optimization, is based on prediction techniques. The communication paths will then be optimized, followed by the optimization of the address generators of the memory banks dedicated to the application being considered. The communication interface is generated next by using the I/O timing behavior of the component [8]. GAUT has been successfully used to design several complex circuits from the telecommunication domain (see [13] and [14] for example).

To validate the generated architecture, a test bench is automatically generated to apply stimulus to the design and to analyze the results. The stimulus can be incremental, randomized or user defined values allowing automatic comparison with the initial algorithmic specification (i.e. the “golden” model). The processing unit can be verified alone. In this case, the memory and communication units are generated as VHDL components whose behavior is described as a Finite State Machine with Data path. GAUT generates not only VHDL models but also scripts necessary to compile and simulate the design with the Modelsim simulator. It can also compare the results of two simulations (produced by different timing behaviors (I/O, pipeline…)).

GAUT also addresses the design of multi-mode / multi-standard architectures [15]. Given a unified description of a set of time-wise mutually exclusive tasks and their associated throughput constraints, a single RTL hardware architecture optimized in area is generated [16].

GAUT supports hierarchical synthesis and will generate multiple clock domain architecture for low-power design on FPGA [17].
GAUT generates an IEEE P1076 VHDL file. The VHDL file is an input for commercial, off the shelf, logical synthesis tools like ISE/Foundation from Xilinx, Quartus from Altera or Design Compiler from Synopsys. GAUT generates a VHDL test-bench and is seamlessly interfaced with Modelsim from Mentor Graphics. SystemC simulation models can be automatically generated from GAUT. These models are Cycle Accurate Bit Accurate (CABA) and can be used in the SocLib platform. More information of this open platform for virtual prototyping of multi-processors system-on-chip can be found at [18].

GAUT also supports FSL (Fast Simplex Link) interfaces which allow to connect customized IP to the MicroBlaze soft processor from Xilinx.

### References


[18] https://www.soclib.fr/