Design of SENIOR: A Case Study Using NoGap

Per Karlström, Wenbiao Zhou, and Dake Liu

1 Introduction

The design and implementation of a new Application Specific Instruction-set Processor (ASIP) processor is usually the result of a substantial design effort, more details about the Application Specific Instruction-set Processor (ASIP) design process can be found in [10]. There are a number of different software tools that relaxes the design effort in one way or another. However all these tools forces the designer into a predefined architecture template. This limitation in design flexibility often makes designers of novel ASIP processors and programmable accelerators revert back to an Hardware Description Language (HDL), e.g. Verilog or VHDL. HDLs offers full design flexibility at the register transfer level, but the flexibility comes at the cost of increased design complexity. All details, e.g. register forwarding and/or pipeline control, has to be handled manually.

This paper will present a case study, where we have used Novel Generator of Accelerators and Processors (NoGap) to design an advanced Reduced Instruction Set Computing (RISC) processor with Digital Signal Processor (DSP) extensions, called SENIOR. The System Verilog code generated by NoGap, was successfully synthesized and targeted to both FPGA and Application Specific Integrated Circuit (ASIC) flows.

As the focus of this paper is not on NoGap the reader is referred to previous publications about NoGap for more detailed information [5, 7, 4, 6].

2 Related Work

A number of tools such as LISA [12], EXPRESSION [2], nML [1], MIMOLA [9], ArchC [11], and ASIP Meister [3], tries to help ease the effort needed to design a
processor. They all have strengths and shortcomings. More details about how they compare to NoGap can be found in the related work section of [8].

3 SENIOR Architecture

The SENIOR processor is a single issue RISC processor with DSP extensions, based on the Harvard architecture. SENIOR is divided into a data path and sequence path. The data path consists of a single precision 16 bit Arithmetic Logic Unit (ALU), a double precision Multiply And Accumulate (MAC) unit, two data memories with dedicated address generated units, a condition checker, flag computation unit, a loop controller, and some data store units including a register file consisting of 32 16 bit registers, 18 16 bit special registers, and four 40 bit MAC register including eight guard bits. The sequence path contains a program counter, a Program Counter-Finite State Machine (PC-FSM) and an instruction memory.

The SENIOR processor has two kinds of pipelines, one normal pipeline and one longer pipeline. The different pipeline architectures are also illustrated in Fig. 1 and Fig. 2.

4 NoGap Common Language (NoGapCL) for SENIOR

NoGapCL is the default NoGap facet used to construct the Micro Architecture Structure Expression (Mase), Micro Architecture Generation Essentials (Mage) and Control Architecture STructure Language (Castle) descriptions, more details about NoGapCL can be found in [8].

4.1 Mage in SENIOR

Mage Functional Units (FUs) are used to describe leaf module with functionality, like most of the components in Fig. 1 An example can be seen in in Listing 1