MetaCore: A Configurable & Instruction-Level Extensible DSP Core

Jin-Hyuk Yang, Byoung-Woon Kim, Sung-Won Seo, Sang-Jun Nam, Chang-Ho Ryu, Jang-Ho Cho, and Chong-Min Kyung
Department of Electrical Engineering
KAIST
Taejon, 305-701, Korea
Tel: +82-42-866-0700
Fax: +82-42-866-0702
e-mail: {jhyang,kyung}@dalhara.kaist.ac.kr

Abstract—The design of application-specific processor for DSP applications is not only driven by low-cost requirements, but also needs to fulfill constraints in terms of short development time. MetaCore, a configurable and instruction-level extensible DSP core, can effectively satisfy these requirements by providing a high degree of customization to the given application. In this paper, we present architectural concept of MetaCore and the MetaCore development platform. And we also present the first-generation MetaCore product version, MDSP16, which is a 16-by-16 fixed-point DSP processor. MDSP16 was implemented using 0.6μm TLM CMOS technology and has 50 MIPS peak performance.

I. Introduction

The state-of-the art VLSI technology now allows for the integration of a DSP core and dedicated components on a single chip. There are two types of DSP cores which are offered by a number of DSP suppliers. One is fixed-architecture DSP cores such as Motorola’s 16-bit DSP cores, the Texas Instruments cDSP and the DSP cores from SGS-Thomson, etc. The fixed-architecture DSP cores offer a very limited degree of customization in that just memory dimension and peripherals can be tuned to a specific application. The other is configurable DSP cores such as EPICS from Phillips and Clarkspar’s CD2100 DSP core. In contrast to relative fixed-architecture DSP cores, the configurable DSP cores can be further customized to a specific application in terms of word length, register file size and instruction set. The instruction set customization in the configurable DSP cores is achieved by removing less useful instructions from pre-defined instruction set. This feature can help application-specific DSP designers reduce silicon area while achieving the required performance level. However, there is no way to extend instruction set in order to improve performance.

We designed a configurable and instruction-level extensible DSP core, which is called MetaCore. The major contribution of MetaCore is instruction-level extensibility at application-specific processor designer level. In this paper, we present architectural concept of MetaCore and MetaCore development platform. And we also present the first-generation MetaCore product version, which is called MDSP16.

II. The MetaCore Architecture and Development Platform

MetaCore architecture has been designed to meet the basic computational, configurability and instruction-level extensibility requirements. The result is shown in Fig. 1. As shown in Fig. 1, the core processor performs the basic computations. The heart of core processor consists of three execution units operating in parallel. Those are the fixed-point data arithmetic logic unit (DALU), the memory address generation unit (AGU) and the program control unit (PCU). The DALU performs MAC (Multiply-Add and ACCumulation) operation and various arithmetic logic operations. The AGU performs effective address calculations necessary to indirect address data operands in memory. The address ALUs in AGU calculate three addressing modes generally used in DSP algorithms, which are linear, modulo and bit-reverse. The PCU supports branching, zero overhead loop control, subroutine control and exception handling. The off-core elements consists of memories, peripherals and external interface.

The core processor and off-core elements have parallel processing capability. During the execution of arithmetic logic or data transfer operation, operand fetch from memory and operand address pointer update are performed in parallel. The standard format with an example of arithmetic logic instruction is as follows:

[standard format] acc. = [acc. (+,-)] S1 op S2;

[example of MAC] a0 = a0 + ar1++ * ar2++;

In standard format, S1 and S2 include address pointer and its update rule. The optional part [(accumulator, +,-)] is used for MAC instructions. Example shows a MAC instruction, consisting of two arithmetic operations (multiply/addr), a double operand fetch from memory pointed by ar1 and ar2 and a double operand address pointer update (ar1 = ar1+1, ar2 = ar2+1).

The core processor and off-core elements can be customized to a given application by tuning MetaCore parameter set, including data and address word length, on-chip memory size, datapath bit-width, and register file (accumulator and address pointer) size, etc. Those instructions and addressing modes which are hardly used, or not used at all in a certain application can be removed from a derived product version.

In MetaCore architecture, the most distinguishing feature is EALU (Extended ALU) interface. EALU interface supports instruction-level extensibility to satisfy the requirement for more computing power and/or reducing power consumptions in a certain application area. It has a pair of source operand buses and a destination bus at which hardware functional blocks can be added. An application-specific processor designer can easily extend instruction set by only describing extended instructions information which are needed by MetaCore development platform to generate the application processor. The information include instruction encodings and control signals to activate functional blocks which execute the instruction. The extended instructions designed by application-specific processor designer must meet the standard format.

Fig. 1. MetaCore architecture
In order to fulfill short development time requirement, MetaCore development platform offers a set of software tools and HDL generator. Software tools include instruction set simulator, assembler, C-language compiler. Software tools have re-targetability to support configurability and instruction-level extensibility. The HDL generator includes parameterized HDL hardware model, hardware information database and application-specific processor's HDL code generation software (SMART).

The methodology to develop an application-specific processor using MetaCore development platform is depicted in Fig. 2. An application-specific processor's HDL model is derived from MetaCore platform by tuning MetaCore parameter set and describing information about extended instructions. SMART assigns hardware parameters to MetaCore HDL model's parameter variables for generating HDL model of core processor and off-core elements. And SMART generates HDL model of decoding logic and EALU part. The decoding logic is a mapping table between instruction encoding and hardware control signals. As describe above, instruction set extension can be done only by specifying target architecture description and updating databases of MetaCore development platform. Once HDL model is generated, an application-specific processor can be synthesized using available standard-cell libraries, datapath libraries and memory compilers. The HDL model is also being used for functional verification of the application-specific processor.

III. The First-Generation MetaCore Product Version: MDSP16

The first-generation MetaCore product version, MDSP16, is a 16-by-16 fixed-point DSP processor. The MDSP16 was implemented using 0.6μm TLM CMOS technology and has 50 MIPS peak performance. We achieved a high-performance DSP processor and verified application-specific processor design methodology using MetaCore development platform through MDSP16 example.

TABLE I

<table>
<thead>
<tr>
<th>Hardware Configuration</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>arithmetic</td>
<td>fixed-point</td>
</tr>
<tr>
<td>datapath word length</td>
<td>36 bit</td>
</tr>
<tr>
<td>number of accumulators</td>
<td>2</td>
</tr>
<tr>
<td>addressing mode</td>
<td>linear/modulo/bit-reverse</td>
</tr>
<tr>
<td>memory and peripherals</td>
<td></td>
</tr>
<tr>
<td>program memory size</td>
<td>32KB</td>
</tr>
<tr>
<td>data memory size</td>
<td>4KB, 4KB</td>
</tr>
<tr>
<td>data memory word length</td>
<td>16 bit</td>
</tr>
<tr>
<td>peripherals</td>
<td>serial port[2], timer</td>
</tr>
</tbody>
</table>

MDSP16 is a 16-by-16 fixed-point DSP processor. The MDSP16 was implemented using 0.6μm TLM CMOS technology and has 50 MIPS peak performance. We achieved a high-performance DSP processor and verified application-specific processor design methodology using MetaCore development platform through MDSP16 example.


core configuration

References


IV. Conclusion

MetaCore is a DSP architecture model which can be customized to specific application domains. MetaCore architecture and efficient development platform fulfill low-cost and short development time requirements. The first-generation MetaCore product version, MDSP16, was developed using MetaCore design platform.