Configurable processors: a new method for SoC design

ECE 742 Digital Systems-on-Chip Design

Qiao Gao

CWID:A20282211

10/29/2012

Abstract

This paper provides an introduction on a latest SoC design method: configurable processor, which has been widely used for the sake of its lower cost, lower power consumption, higher efficiency for ASIC or Embedded system design and better performance in many practical application aspects such as video decoders and DSP. To make the overview more clear, we take Xtensa architecture IP core as an example to describe what a configurable processor structure is like and how it help designers to simplify the design flow and get approach to an even better function than before. Xtensa is made by Tensilica and it is one of the most popular configurable processor cores around the world. Xtensa architecture provides a brand new design technology for both hardware and software designers and it is so convenient that it leads to a great impact on SoC design since 1999 when it is firstly built.

This paper will give answers to the questions below by providing details about Xtensa:
1. What is configurable processor.
2. What can a configurable processor do.
3. Why we need configurable processor in our design.
4. How it can improve the performance of our design.

Keywords: Configurable, Processor, Xtensa

1 Introduction

1.1 History of microprocessor development:

Microprocessor evolution have largely change the way people live. As we all know, with the great development of information technology, microprocessors have been widely used on almost every kind of electronic products, such as personal computer, cell phone, video camera etc.. However, this is just a small part where microprocessors are used. In a larger scale, microprocessors perform fully automatic functions in a car, an air conditioner or even a microwave oven to keep it working appropriately and safely. This kind of application is known as embedded system, which even occupies most market share of microprocessor usage.

Steve Leibson and James Kim divided microprocessor evolution into four eras[1]. During the 1970s, microprocessors grew from 4-bit logic-replacement devices to 16- and 32-bit designs that paved the way for PCs and workstations. Explosive growth in 32-bit chips wiped out the minicomputer in the 1980s, which also saw the appearance of digital signal processors and other specialized architectures. Reduced-instruction-set computing dominated the 1990s; even stalwart complex-instruction-set computing cores such as the x86 evolved into disguised RISC architectures, and micro-processors became an integral part of mainframes and supercomputers. Moreover, they described the latest technology, the configurable processor, as the fourth era. Configurable processor architecture is an expansion of RISC architecture, which provides designers an opportunity to change the processor structure corresponding to the functions they want to implement.

1.2 About configurable processor:

Configurable processor design method is a full-featured, configurable processor tool package composed by a predefined base processor core and a design tool environment. All of these can be changed according to specific application requirements. In other words, compared to some traditional fixed core architecture, configurable processor can be changed by designer' s will, either cut off some useless functions or add some new functions. More specifically, designers can add, delete or modify processor features like memory, external bus width, handshake protocol or commonly used processor external equipment. By doing this, designers can get the processor' s register transfer level (RTL) description, which can be easily synthesized and used to do designs on next level.

For instance, designers can tune the processor's instruction set architecture to the application's characteristics. Similarly, they can optimize the processor pipeline or data path components for critical code segments in specific application domains. The resulting processors remain programmable and, in principle, can run any application, but they are optimized for a targeted application domain. [2]

Nikil Dutt and Kiyoung Choi thought that even though the idea of customizing processors is appealing, it opens up new challenges: customization of the entire software tool chain, including compilers, simulators, debuggers, and so on; synthesis of processor models; and validation and verification of generated processor designs.[2] Xtensa architecture solved these questions very well, which is the main reason it had been so popular.

1.3 About Xtensa:

As the configurable processor technology provides a synthesizable RTL core finally, designers can easily combine it into a system-on-a-chip design, download it into FPGAs or achieve it by different processes. While these are all advantages of the use of synthesizability processors in SoC design, as described by Ricardo E. Gonzalez, Xtensa is build to exploit these advantages and even streamline extensibility and configurability.[3]

The Xtensa instruction set architecture (ISA) enables configurability, minimizes code size, reduces power dissipation, and maximizes performance. [3] The recent Xtensa architecture is based on Dataplane Processor Units (DPUs), which has some prominent features:[4]

Data throughput: Xtensa processor communicate with outer structures through FIFO like process other than traditional bus interface.

Fit into hardware design flow: the final processor core which is generated by Xtensa design flow can be easily simulated and verified by common hardware design tools, which make it convenient to involve the processor into traditional SOC design.

Processing speed: Xtensa processor provides much higher process speed on a large number of common algorithms.

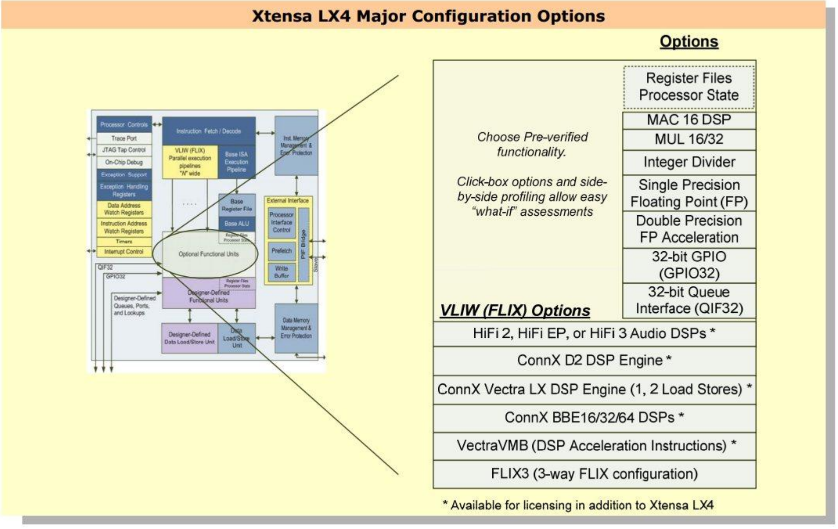
Customization challenges: Xtensa configurable processor can be easily customized by designers who are not familiar with processor architectures.

2 Overview of configurable processor

2.1 What can a configurable processor do

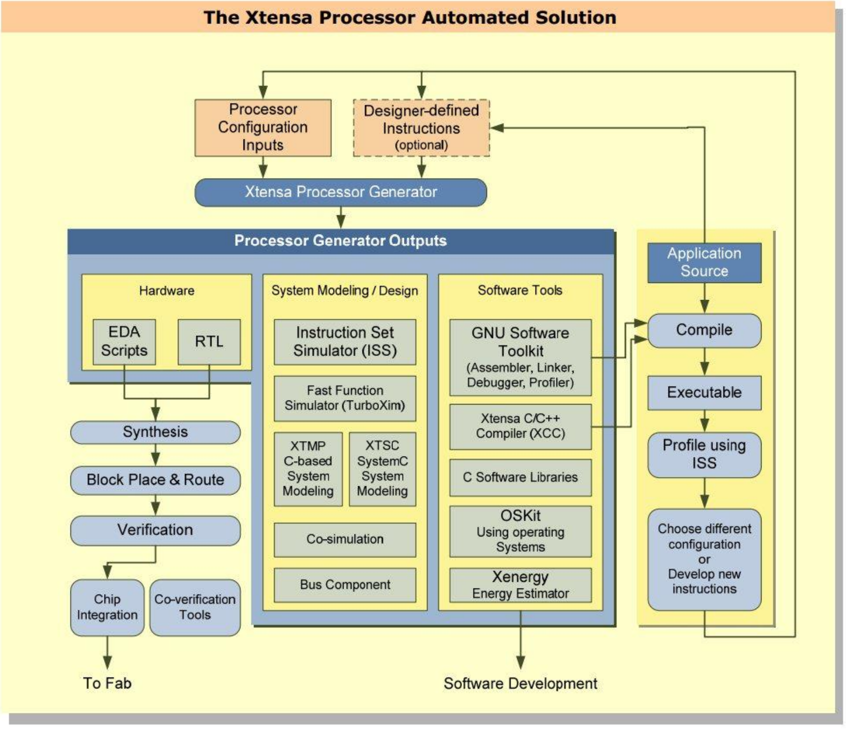
By taking Xtensa configurable processor as an example, we can clearly indicate what tasks a configurable processor can achieve.

Figure 1 shows the major configuration options on the latest version Xtensa configurable processor, the Xtensa LX4. We can see from this figure that Xtensa processor have been optimized on many common functions such as audio and video decoders and DSP. By choosing the parts which is needed for different designs, designers can customized processors of their own with unique instructions for the algorithms they use.



Xtensa LX4 Major Configuration Options
Options
Register Files
Processor State
MAC 16 DSP
MUL 16/32
Integer Divider
Single Precision Floating Point (FP)
Double Precision FP Acceleration
32-bit GPIO (GPIO32)
32-bit Queue Interface (QIF32)
Choose Pre-verified functionality.
Click-box options and side-by-side profiling allow easy "what-if" assessments
VLIW (FLIX) Options
HiFi 2, HiFi EP, or HiFi 3 Audio DSPs *
ConnX D2 DSP Engine *
ConnX Vectra LX DSP Engine (1, 2 Load Stores) *
ConnX BBE16/32/64 DSPs *
VectraVMB (DSP Acceleration Instructions) *
FLIX3 (3-way FLIX configuration)
* Available for licensing in addition to Xtensa LX4

Figure 1: Tensilica offers the widest range of configurable functional units for the Xtensa LX4 DPU[5]



The Xtensa Processor Automated Solution
Processor Configuration Inputs
Designer-defined Instructions (optional)
Xtensa Processor Generator
Processor Generator Outputs
Hardware
EDA Scripts
RTL
System Modeling / Design
Instruction Set Simulator (ISS)
Fast Function Simulator (TurboXim)
XTMP C-based System Modeling
XTSC SystemC System Modeling
Co-simulation
Bus Component
Software Tools
GNU Software Toolkit (Assembler, Linker, Debugger, Profiler)
Xtensa C/C++ Compiler (XCC)
C Software Libraries
OSKit Using operating Systems
Xenergy Energy Estimator
Application Source
Compile
Executable
Profile using ISS
Choose different configuration or Develop new instructions
Synthesis
Block Place & Route
Verification
Chip Integration
Co-verification Tools
To Fab
Software Development

Figure 2: Tensilica's proven methodology automates the creation of customized processors and matching software tools[5]

Figure 2 shows the design flow of using a Xtensa processor. It can be noticed that Xtensa can automatically generate not only synthesizable RTL level processor core, but also the total software develop environment according to the processor structure we defined, which means we can use the configurable processor as a conventional fixed core processor to develop software but with the instructions we have customized to make our designs perform better.

2.2 Why we need configurable processor in our design

The popular fixed instruction set architecture (fixed-ISA) embedded microprocessor architecture design starting in the eighties of the last century, such as ARM, MIPS and PowerPC processors, which were designed as separate chips. The implementation on this framework of many algorithms is satisfactory, but with the development of electronic technology, SoC based embedded systems are being used more frequently and higher data process efficiency is demanded in many applications, such as HD video decoder and high-speed DSP for telecommunication systems. As a result, designers often want to improve running speed on some key parts in the hardware design.

However, this problem can be solved by adding external RTL module since RTL units can provide very high speed for they are designed only to achieve specific algorithms. Usually, RTL modules are originally described in Verilog or VHDL code, so they can only be used after passing compiling and synthesis. Though these modules do work in very high efficiency, they still have some disadvantages:

1. The speed of the dataflow between processor core and external RTL modules will be limited by the width of bus of the processor.
2. The design will get much more complicated, because the simulation and verification of a new RTL design is very complex and take a long time. What' s more, it will lead to more possibility on mistakes to make the initialize modules and interface modules for the RTL design
3. It will cost more than single fixed core design since many of the modules which are used in this design may cannot be reused in the next design.

Actually, with the development of the consumer electronics industry, it is very hard to integrated such a large number of high-speed functions by adding RTL modules. Though the recent fixed processors are often come with a very high frequency, it is still hard to meet the demand some high speed implementation. Moreover, Higher frequency means more power consumption.

However, configurable processors provide us a perfect solution on these problems. By adding algorithm-specific instructions, the whole system can run in a high efficiency and much lower frequency. Furthermore, many common modules have already been provided that designers can just add them to the core directly, which can largely decrease time on verification steps.

2.3 How it can improve the performance of our design

We can also use Xtensa architecture as an instance.

2.3.1 The Xtensa architecture break through the input / output bottleneck

In order to improve the I/O bandwidth, configurable processor must overcome the bus bottleneck, which is a long existing problem since the time Intel introduced the first commercial microprocessor 4004 in 1971. In this structure, each processor have to communicate with the rest of the design components through the system bus and the dataflow on the bus is controlled by the load/store components. Due to the inherent characteristics of the bus, at any time, it allows only a small portion of the data communication on the bus and the processor. Furthermore, the load/storage units, the processor internal execution units and the processor local memory exchange data with each other on the limited bus of the same kind. Such kind of single, one direction for once processor bus characteristics severely limit the throughput of the microprocessor system.

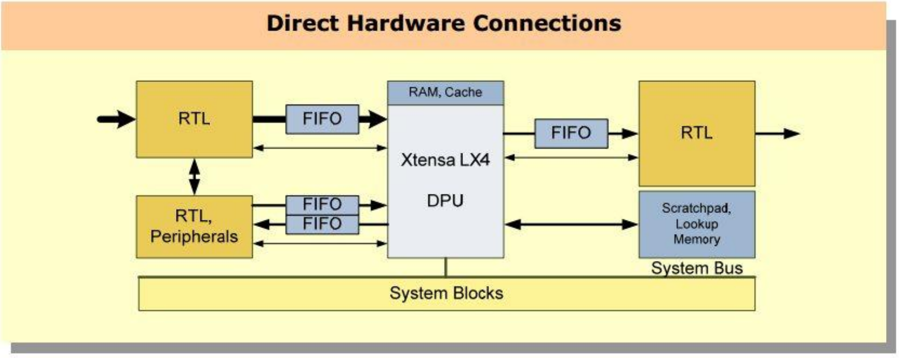


Figure 3: Example of direct FIFO and Port connections using TIE Queues and TIE Ports[5]

In order to overcome the inherent limitations of the processor bus, Tensilica Xtensa processors have been designed with a new feature which eliminates bus bottleneck problem forever. This new feature is called the TIE (Tensilica Instruction Extension)port and queue technology.[3] Using the TIE port and queue technology, designers can define up to 1024 ports directly connected with the Xtensa processor execution units, as it is shown in figure 3. This technique results in 350,000 Gbits/sec[5] on speed of information exchange between the system and the Xtensa processor, which can fully satisfy all of the processor's input/output bandwidth demands and system requirements in RTL design.

2.3.2 The Xtensa architecture improves computing performance

Three techniques have been used to create optimized Xtensa processor configurations: operator fusion; single instruction, multiple data (SIMD) vectorization; and flexible-length instruction extensions (FLIX).[1]

Xtensa processor gives system-on-chip designers the ability to define new instructions, based on the multi-operation instructions technology that multi-operation instructions can be used as a new instruction. The technology which combine different operation as a new instruction is called operands fusion. Operand fusion technology can effectively improve the computational performance of the microprocessor. In addition, the system-on-chip designers can add SIMD (Single Instruction Multiple Data) instructions to any Xtensa processors. SIMD instructions can perform the same operation on multiple data elements at the same time, as it is shown in figure 4. This technology can also improve the computational performance of the microprocessor significantly.

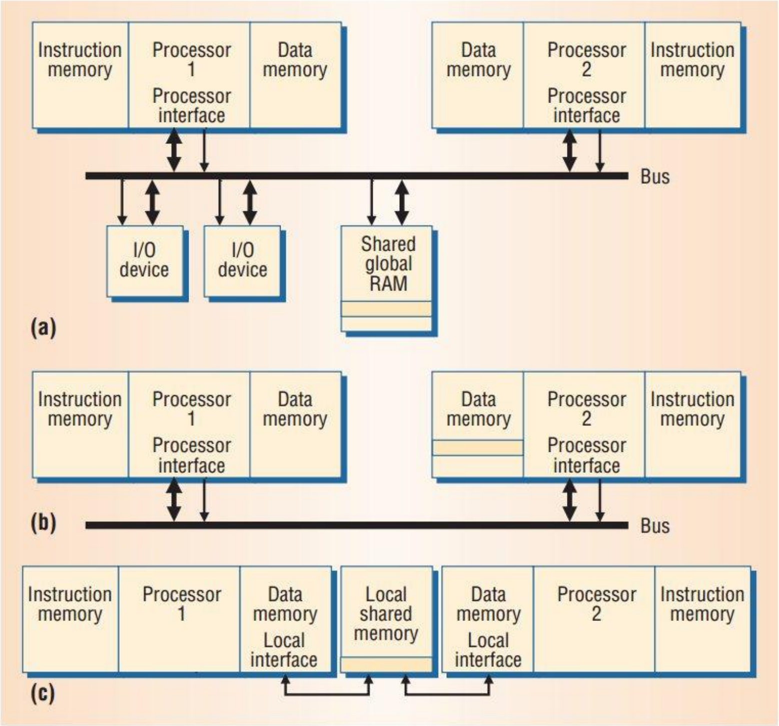


Figure 4: Shared-memory bus topologies using configurable processors. (a) Two processors access global memory over a bus. (b) One processor accesses the local data memory of a second processor over a bus. (c) Two processors share access to local data memory.[1]

Unlike the multiple dependent operations of fused and SIMD instructions, FLIX instructions consist of multiple independent operations. Tensilica's Xtensa C/C++ compiler can pack these operations into a FLIX-format instruction as needed to accelerate code. While fused and SIMD instructions are 24 bits wide, FLIX instructions are either 32 or 64 bits wide to allow the flexibility needed to fully describe multiple independent operations.[1]

3 Result and Discussion

The final discussion focuses on what the future of configurable processor will be like. Nowadays, with the advent of the post-pc era, configurable processors have been widely used on SOC design because functions that consumers demand for electronic devices are changing very quickly. To keep pace with the development, designers have to find more efficient and less cost ways for their design and using configurable processors is even the best choice for them.

As R E. Gonzalez concluded in his article[3] when configurable processor technology had been just get started, configurable processor do open a new era on SoC design. At present, configurable processor has even taken an equal status with general-purpose processor and play an important role in design methods.

Moreover, Steve Leibson and James Kim boldly predicted in 2005[1] that configurable processors will someday replace the conventional design methods. However, N. Dutt and K. Choi thought there is still a long way to go for this brand new design method.[2] As a fact, configurable processor technology is not the only design method that developed fast these year. For example, fixed core processor architectures, such as ARM, have been greatly improved on both working efficiency and IC process. Though it still performs worse on some specific algorithms than configurable processor, it is much more easier to use and can be easily approached by designers on different levels.

In my opinion, configurable processor may cannot replace the traditional design methods in a short time. From a long-term perspective, however, it is so hard to predict the market trend that maybe one day fixed core processor could be substituted totally by configurable one, or they could all be replaced by a newer technology.

4 Conclusion

We have taken Xtensa architecture as an example to provide an overview on a new SoC design method: configurable processor technology and answered the questions we put forward at the beginning of the article. Finally we know that how this technology help designers to simplify design flow and get better performance at result on SoC design

5 Future recommendation

The thought of using configurable processors is adding customizes the instructions for specific algorithms based on a RISC standard processor. Why not go deeper and provide a platform for processor architecture design? What' s more, the optional instructions are so limited. Maybe there should be a better technology that allow us to implement any instruction we want.

6 References

[1]. Steve Leibson and James Kim, "Configurable processors: a new era in chip design," IEEE Computer, vol. 38, no. 7, 2005, pp. 51- 59.
[2]. N. Dutt and K. Choi, "Configurable processors for embedded computing," Computer, vol. 36, no. 1, pp. 120–123, Jan. 2003.
[3]. R E. Gonzalez, "Xtensa: A configurable and Extensible Processor", IEEE Micro, vol. 20, No. 2, March/April2000
[4]. Tensilica. [Online]. Available: http://www.tensilica.com
[5]. Xtensa LX4 Product Brief. [Online]. Available: http://www.tensilica.com/uploads/99/LX4-pdf