A 52mW 1200MIPS Compact DSP for Multi-Core Media SoC

Shih-Hao Ou, *Tay-Jyi Lin*, *Chao-Wei Huang*, *Yu-Ting Kuo*,
Chie-Min Chao, *Chih-Wei Liu*, and *Chein-Wei Jen*
Department of Electronics Engineering
National Chiao Tung University, Taiwan

Abstract - This paper presents a DSP core for multi-core media SoC, which is optimized to execute a set of signal processing tasks very efficiently. The fully-programmable core has a *data-centric* instruction set and a corresponding *latency-insensitive* micro-architecture, where the hardware design is optimized concurrently with its automatic software generator. The proposed DSP core has 3X performance (in cycles) of those found in commercial dual-core application processors with similar computing resources. The silicon implementation in UMC 0.18μm 1P6M CMOS technology operates at 314MHz and consumes only 52mW average power.

I. Introduction

The computations of next-generation communication devices can no longer be satisfied with a single processor at reasonable cost. Dual-core processor with a RISC and a DSP is a popular solution to meet the computation demands. TI OMAP [1], which consists of an ARM core and a DSP core, is a well-known example. The ARM core is responsible for control-oriented tasks such as the user interfaces, system coordination, and protocol stack, while the DSP core takes care of the computation-intensive tasks such as baseband processing, data transformation and so on. However, most dual-core processors have redundant components because they are constructed with existing processor cores, which are designed for standalone uses. In this project, we design a DSP core from scratch, while the RISC core is kept unchanged for software compatibility. The design goals include multi-core configuration, compactness, and low power. Moreover, automatic software generation from high-level specification is of the same importance. This paper is organized as follows. The processor design is first described in Section II. Section III summarizes the chip specification and silicon implementation of the compact DSP core. Finally, Section IV concludes this work.

II. Processor Design

A. Data-centric ISA

Considering the generic 4-way VLIW processor, conventional ISA (instruction set architecture) [2] specifies operations performed by each functional unit and additionally where to get the corresponding source/destination operands and some control information. We call aforementioned ISA operation-centric. In contrast to the operation-centric ISA which controls the functional units, our data-centric ISA is designed to be responsible for data generation directly. In other words, the instructions specify the required data by each functional unit in each current iteration and take care of the returned results from all functional units in previous iteration. In addition, the operation and other control information are carried with instructions. Take our addition instructions for example. The assembly syntax is described as: `Rd=DS, (Rs>>+Rt>>)>>;` that is interpreted as "***store the DS datum into the register Rd***, and ***add*** the ***datum in the register Rs and the register Rt together***". Those ">>" marks enable one-bit shift for input alignment or output normalization.

B. Latency-insensitive microarchitecture

Fig. 1 depicts the DSP core's micro-architecture consisting of the data generator and 4 sites which are the adder, the multiplier, the shifter and the control unit respectively. The data generator is made up of the 4-by-4 crossbar switch and those DRF (distributed register file). Additionally, the DSP is equipped with the banked data memory served as ping-pong buffer to reduce the communication overhead between the DSP core and host processor. Because the micro-architecture is tightly coupled with the data-centric ISA by which the operands is scheduled, the processor absolutely does not require some complex forwarding-path found in conventional VLIW processors between functional units. Therefore, the micro-architecture features latency-insensitive characteristic. Besides, the micro-architecture is so modular and thus simplifies replacement and collocation of functional modules with different latency. Thus, only simple modification in the software generator is required to adapt to different hardware configurations without altering the other hardware blocks. By the way, as the technology advances rapidly, the modular micro-architecture which localizes the interconnection is an effective solution to the increasing wire delay & on-chip interconnection overhead.

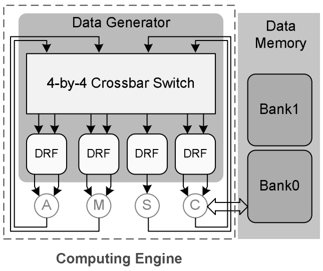

Figure 1. Computing engine

C. Automated Code Generation

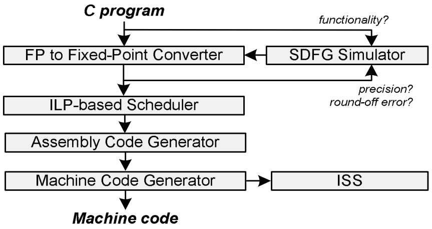

Figure 2. Code generation flow

Fig. 2 depicts the automatic code generation flow of our DSP core. First, the FP SDFG [3] can be derived from the C descriptions via the SUIF compiler. The SDFG simulator is bit-true and supports both the FP and the fixed-point arithmetic, and the designers can develop and verify their DSP algorithms easily. Once the functionality of the FP SDFG is verified to be correct, the FP-to-fixed-point converter translates the FP SDFG into a fixed-point one by applying static analysis which is based on worst-case range analysis [4]. The operations of the fixed-point SDFG are scheduled

0-7803-9451-8/06/$20.00 ©2006 IEEE.

with the integer linear programming (ILP). Finally, the fixed-point executables can be generated and simulated by the machine code generator and the ISS (instruction set simulator) respectively.

D. Performance Evaluation

Several popular DSP kernels are used to evaluate our proposed DSP core. Table 1 outlines the results. In addition, the 2nd & 3rd rows give the reference performances of two commercial DSP cores, both of which have already been integrated in some dual-core processor designs. ADI ADSP-218x has similar computing resources to our DSP core, while TI C'55 has one more MAC unit. The cycle counts are all excerpted from their application notes. The reasons that our DSP has such improvements over conventional DSP can be summarized as follows. First, its data-driven engine and code generator are developed in parallel based on high-level synthesis to extensively exploit the inherent parallelism of DSP algorithms, and the performance can therefore be very close to that of customized ASIC designs. Then, the data-centric ISA & latency-insensitive micro-architecture enable smooth dataflow with its internal crossbar network and relatively plenty registers (note that the complexity is much less than that of a plain register file in the general-purpose processors). Moreover, the four embedded 1-bit shifters in the fixed-point arithmetic units also help the reduction of the execution cycles. We further have several implementations of the 2-D DCT from the independent JPEG group (IJG) to analyze the round-off error of our proposed fixed-point arithmetic. Table 2 summarizes the comparisons. The proposed 16-bit fixed-point even outperforms the hand-optimized 32-bit integer 2D-DCT from IJG. Moreover, our 24-bit fixed-point has about 64dB PSNR, which has the same maximum precision as the single-precision FP (i.e. with the 23-bit mantissa).

TABLE I. PERFORMANCE EVALUATION

	Lattice	Biquad	FFT	2D-DCT
ADSP-218x	32	13	874	2,452
TI C'55	12	5	367	1,082
Proposed DSP	**12**	**16**	**268**	**688**

TABLE II. ROUND-OFF ERROR COMPARISON OF 2D-DCT

	PSNR (dB)	Cycel count
Single-precision FP	--	624
16-bit integer	29.5183	848
32-bit interger	33.8020	656
Proposed 16-bit	**40.0468**	**656**
Proposed 32-bit	**64.1201**	**624**

III. Silicon Implementation

Fig. 3 illustrates the global design flow exercised by the proposed DSP core. First, in the **ISA/Micro-architecture Design** phase, we analyze several popular DSP algorithms to design the ISA and the micro-architecture. The cycle-accurate SystemC model is utilized to be verified with the compiler-generated code. If the resulted performance is not satisfied, the micro-architecture refinement, i.e. latency modification is performed to improve performance. Then in the **RTL Design** phase, we write the fully synthesizable RTL in Verilog and use the SystemC model to cross-check the correctness by HDL simulator. We further manually handcraft assembly codes to account for the corner cases to increase the code coverage. The metrics such as statement, branch, state, and arc all achieve 100%. Finally the **Implementation** phase involves the RTL synthesis and the physical design by means of the *Synopsys Design Compiler* and the *SoC Encounter* respectively. The functional verification of the synthesized gate-level net-list is through the formal equivalence checking with the RTL model. Besides, the timing/area/power consumption estimation is made by the popular estimation tools like *PrimeTime*, *PrimePower*, *nanosim* and so on. Others like the scan-insertion and memory BIST are also implemented to increase testability. If the required performance/cost is not met, one solution is to go back to RTL design, i.e. RTL code quality improvement by *nLint* and the other is again back to micro-architecture refinement.

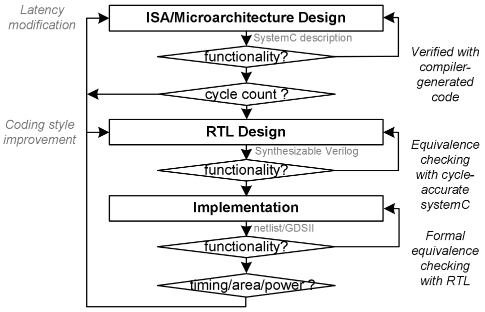

Figure 3. Design flow

The DSP has been implemented and fabricated in UMC 0.18µm 1P6M CMOS technology. Table III summaries the specification and Fig. 4 shows both the chip layout and the die photo. In the post-layout simulation, the DSP core can operate at 314 MHz and consumes only 52mW average power. The chip has already been fabricated and tested to work correctly at 100MHz. The maximum operating frequency is unknown at the time of paper submission due to the limitation of our available *IMS* test machine.

TABLE III. CHIP SEPCIFICATION

Technology	UMC 0.18um 1P6M CMOS
Core size	1.5 x 1.5 mm^2
Transistor/Gate Count	197,655
Power dissipation	52 mW
Max. frequency	314 MHz
On-chip memory size	16KB (8KB data /8KB instruction)

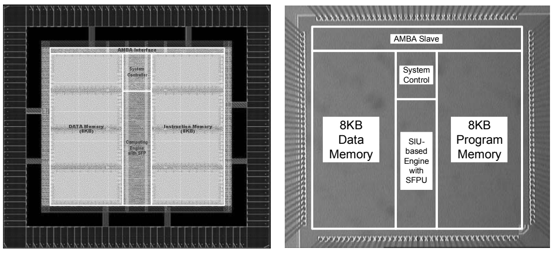

Figure 4. Chip layout (left) & die photo (right)

IV. Conclusion

The paper summaries the design of a compact DSP core for multi-core media SoC, from the definition of its instruction set in C++, the micro-architecture exploration in cycle-accurate SystemC, to the synthesizable RTL implementation. The DSP core with compiler-generated software codes has about 3X performance improvements of those found in commercial dual-core application processors. Its silicon implementation in the UMC 0.18µm CMOS technology achieves 314MHz frequency and consumes only 52mW.

References

[1] *OMAP5910 Dual Core Processor – Technical Reference Manual*, Texas Instruments, Jan. 2003

[2] J. L Hennessy, and D. A. Patterson, *Computer Architecture – A Quantitative Approach*, 3rd Edition, Morgan Kaufmann, 2002

[3] K. K. Parhi, *VLSI Digital Signal Processing Systems – Design and Implementation*, John Wiley & Sons, 1999

[4] S. H. Ou, T. J. Lin, H. Y. Lin,C. M. Chao,C. W. Liu and C. W. Jen, "Lightweight arithmetic units for VLSI digital signal processors," in *Proc. VLSI-TSA-DAT*, Apr. 2005