A 100 GOPS ASP Based Baseband Processor for Wireless Communication

Zhu Ziyuan, Tang Shan, Su Yongtao, Han Juan, Sun Gang, Shi Jinglin
Wireless Communication Technology Research Center, Institute of Computing Technology, Chinese Academy of Sciences
Beijing Key Laboratory of Mobile Computing and Pervasive Device
Beijing, China
zhuziyuan@ict.ac.cn

***Abstract*—This paper presents an ASP (application specific processor) with 512-bit SIMD (Single Instruction Multiple Data) and 192-bit VLIW (Very Long Instruction Word) architecture optimized for wireless baseband processing. It employs optimized architecture and address generation unit to accelerate the kernel algorithms. Based on the ASP, a multi-core baseband processor is developed which can work at 2x2 MIMO and 20 MHz physical bandwidth configuration for LTE inner receiver and meet requirements of Category 3 User Equipment (CAT3 UE). Furthermore, a silicon implementation of the baseband processor with 130nm CMOS technology is presented. Experimental results show that the baseband processor provides 100 GOPS computing ability at 117.6MHz.**

Keywords—Application Specific Processor; VLIW; AGU; Baseband processor; LTE

I. INTRODUCTION

As wireless communications have been developing rapidly in recent years [1-2], baseband processor with high complexity and multiple modes have become the norm in the high-end mobile market. However, designing such a sophisticated baseband processor is almost an impossible mission, with regard to the fact that a huge amount of signal processing tasks must be conducted with a limited budget of power consumption. E.g., the workload of the baseband processing for LTE could be higher than 100 giga operations per second (GOPS). For LTE-A, the value could be nearly 500 GOPS [3]. The increasing needs on computing capability does not loosen the power requirement. Moreover, in order to reduce the cost, extending the life-time of a baseband chip by increasing its flexibility is becoming a major trend.

Traditionally, multi-mode baseband processor for mobile device integrates different hardware accelerators for different communication standards. This method leads to significant increase of hardware cost and complexity, which is not manageable in the coming future. At the same time, DSP based baseband processing is widely used in wireless base station, where the power consumption is not a critical constraint. The flexibility of DSP makes different wireless communication standards could be implemented and integrated easily. These advantages encourage the research on designing more efficient DSPs and the programmable baseband processor which can satisfy the flexibility requirements of the complex and multi-mode mobile devices. From 2003, SandBridge Ltd. proposed a series of DSPs named 'Sandblaster' as SDR (Software-Defined Radio) solution for 3G and 4G standards [4]. In 2005, NXP presented a vector processor named 'EVP' supporting WLAN, UMTS and other 3G standards [5]. The micro-architecture and instruction set architecture (ISA) of EVP have many features optimized for wireless baseband processing. Different from traditional DSP, the processors with optimized architecture and ISA for specific applications are always classified to ASP [6]. For the timing/power critical task of wireless signal processing, ASP can be finely tuned to meet the performance requirements in a low power budget. Based on ASP optimized for 3G wireless protocols, in 2007, University of Michigan developed a fully programmable SDR platform named 'SODA' [7]. Followed in 2008, ARM Ltd. released a commercial SDR prototype named 'Ardbeg' based on SODA architecture [8]. Both EVP and SODA provide a relative high performance by using SIMD and VLIW technologies which improves data parallelism and instruction parallelism respectively. Continued in 2010, [9] present an upgraded version of 'SODA' named 'SODA-II' by improving the memory access architecture.

Different from previous works, in this paper we present our attempt of designing an efficient ASP architecture which focuses on the SIMD/VLIW architecture and the memory access optimization. Furthermore, a LTE baseband processor is build base on the ASPs. The rest of the paper is organized as following. In section II, the architecture of proposed ASP and the optimization strategy are described. Followed by section III, a baseband processor is build and its performance for LTE inner receiver are shown. Finally, the silicon implementation of the ASP and the baseband processor are given in Section IV.

II. THE ASP ARCHITECTURE

In this paper, the proposed ASP is named BP-ASP (Baseband Processing ASP) for convenience. The architecture of BP-ASP is VLIW and SIMD mixed. For the data parallelism, BP-ASP has several 512-bit SIMD data paths (DP) and each of them supports 32-lane 16-bit or 16-lane 32-bit computing. For the instruction parallelism, BP-ASP has 192-bit VLIW instruction word length supporting six instructions dissipation in parallel. Furthermore, BP-ASP has a structured address generation unit which provides powerful and flexible memory address generation ability.

This work is funded by National Science and Technology Major Project 2011ZX03003-003-02 and 2012ZX03001007-004.

978-3-9815370-0-0/DATE13/©2013 EDAA

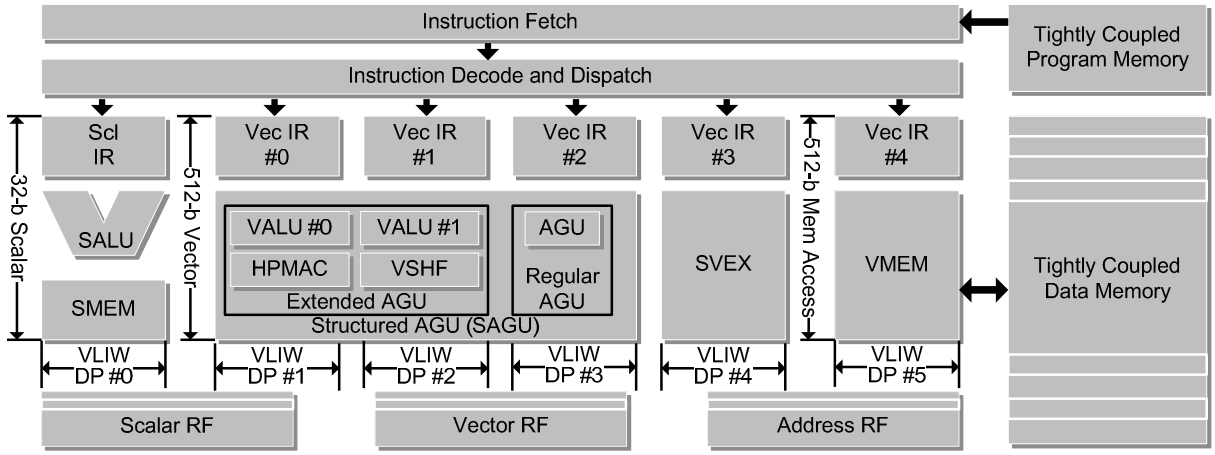

Fig. 1. BP-ASP Architecture block diagram

The block diagram of BP-ASP is shown in Fig. 1. BP-ASP has a 6 data-path (DP) VLIW architecture where the VLIW DP #0 (the first VLIW data-path) is used for scalar computation and program flow control, VLIW DP #1, #2 and #4 are used for vector computation, and VLIW DP #3 and #5 are used for memory access. The program and data memories are implemented with tightly coupled memory (TCM). There are three register files in BP-ASP architecture, the scalar data RF (Scl-RF), the vector data RF (Vec-RF) and the address RF (Addr-RF). The functional units are located in the different data paths, including scalar arithmetic/logic unit (SALU), scalar memory access unit (SMEM), vector arithmetic/logic unit (VALU), high performance MAC unit (HPMAC), vector shuffle unit (VSHF), address generation unit (AGU), vector memory access unit (VMEM) and scalar/vector exchange unit (SVEX). The function of these units are listed in TABLE I. The AGU, VALU, HPMAC, VSHF and the dedicated address register file, Addr-RF, can be organized as a structured address generation unit (SAGU), as shown in Fig. 1.

BP-ASP employs two technologies to enhance the computing performance for baseband processing: HW/SW co-optimized VLIW architecture and SAGU. The design principles and details will be shown in the following sections.

A. HW/SW Co-optimized VLIW Architecture

By profiling the kernel algorithms of baseband processing, it reveals that most of the algorithms could be decomposed into five primitive operations: vector computation, vector shuffle/alignment, vector load/store, vector/scalar data exchanging, and program flow control. These primitive operations can be corresponded to different sets of functional units in BP-ASP, as listed in TABLE II. Base on the primitive operation classification, a straightforward VLIW architecture and the corresponding functional unit allocation are shown in Fig. 2. This architecture is a 5 data paths VLIW machine which provides an instruction parallelism of 5 and a data parallelism of 2080-bit (32-bit for DP #0 and 512-bit for DP #1, DP #2, DP #4, DP #5, respectively). On the other hand, the read port/write port (rp/wp) numbers of the Vec-RF are taken as the major consideration of area and power efficiency. The straightforward VLIW architecture needs a 5 read ports/4 write ports Vec-RF.

TABLE I. FUNCTIONAL UNIT DESCRIPTION IN BP-ASP

Functional Units	Function Descriptions
VALU	512-bit vector arithmetic and logic operation.
HPMAC	512-bit vector multiplication, multiply and accumulate. Radix-2 or radix-4 butterfly.
VSHF	Vector Shuffling and reordering with a certain pattern.
AGU	Address calculations for vector memory access.
VMEM	Vector load and store.
SVEX	Broadcast a scalar data to a vector register; Downsize the vector register to scalar register.
SALU	Branching and program flow control.
SMEM	Scalar load and store.

TABLE II. PRIMITIVE OPERATIONS AND THE CORRESPONDING ASP UNITS

Primitive Operations	Dedicated Functional Units
Vector Computation	VALU, HPMAC
Vector Alignment	VSHF
Vector Load/Store	VMEM, AGU
Vector/Scalar exchange	SVEX
Program Flow Control	SALU, SMEM

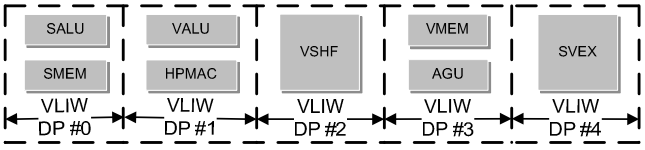

Fig. 2. A straightforward VLIW architecture of BP-ASP

```
for ( i = 0; i<n; i++) {
    C[i] = A[i] + B[i];
    D[i] = conj( C[i] );
}
```
(a)

```
for ( i = 0; i<n; i++) {
    store( A[i], address );
    address = address + x;
}
```
(b)

Fig. 3. Two pseudo code pieces in baseband processing

```
C[0] = A[0] + B[0];
for ( i = 1; i<n;  i++) {
  C[i] = A[i] + B[i] || D[i-1] = conj( C[i-1] );
}
D[n-1] = conj( C[n-1] );
```
(a)

```
for ( i = 0; i<n;  i++) {
  store( A[i], address ) ||  address = address + x;
}
```
(b)

Fig. 4. The optimized code pieces

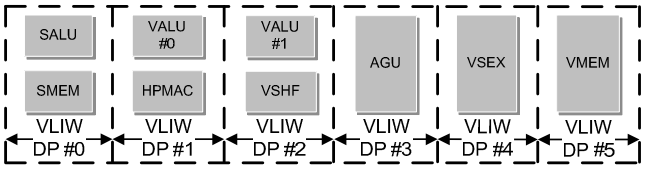

Fig. 5. The optimized VLIW architecture

In order to optimize the straightforward VLIW architecture for the minimum hardware cost, the functional unit can be further split into separate VLIW data-paths for higher parallelism in the case of keeping Vec-RF rp/wp numbers unchanged. By exploring the assembled codes of the baseband processing software, it is found that there are large numbers of consequent VALU operations and memory access operations in loops. Two representative pseudo codes pieces are shown in Fig. 3. , where the variables A[i], B[i], C[i] and D[i] are vector data, the symbol conj( ) represents conjugation operation. These two pieces of codes cannot be executed in parallel, because the consequent instructions can only share the functional unit in sequence as the time division multiplexing in the straightforward VLIW architecture. To promote the Fig.3.a code performance, we further split the VALU unit into two sub unit, named VALU #0 and VALU #1. The VALU #0 unit supports all the dual-operand instructions, e.g. addition, logic AND, logic OR. The VALU #1 unit supports the rest of single-operand instructions, e.g. logic NOT and complex conjugation. By moving the VALU #1 to the different VLIW data-path (DP #2), the Fig.3.a could be optimized in parallel, as illustrated in Fig.4.a. To promote the Fig.3.b code performance, VMEM unit and AGU unit are separated into two different VLIW data paths. The code can be optimized as shown in Fig.4.b. The notation '||' in Fig. 4. represents VLIW instruction concatenation symbol which means that the operations beside the symbol are executed in parallel.

With the functional unit re-allocation discussed previously, the optimized VLIW architecture is obtained, as shown in Fig. 5. This is a 6 data paths VLIW machine which provides more parallel capability for data computing, address calculation and memory access. For the quantitative analysis, the 2048 point FFT of the LTE OFDM kernel algorithms is tested. The computation cycle on straightforward architecture is 3384 while the value is 2352 on the optimized architecture. This result shows that there is 33% improvement for 2048-point FFT with the optimized VLIW architecture.

B. Structured Address Generation Unit

Baseband processing is characterized by intensive data memory access. It reveals that there are two types of memory address calculations: the deterministic patterns of address generations and the complex address calculations. Previously works always focus on the hardware AGU for the deterministic address calculations pattern [10]. In this section, a structured address generation unit (SAGU) optimized for baseband processing is designed for BP-ASP.

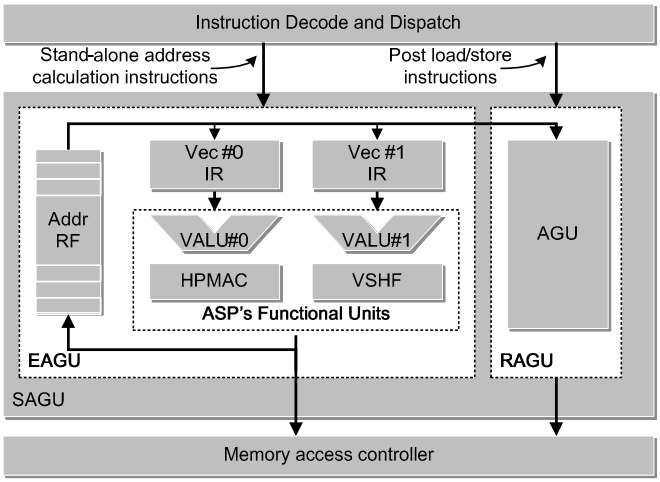

Fig. 6. Block diagram of SAGU

The structure of SAGU is shown in Fig. 1. and the block diagram of SAGU is shown in Fig. 6. SAGU includes two sub-unit: the regular-AGU (RAGU) and the extended-AGU (EAGU). The RAGU is a simple hardware accelerator used to update the memory access address automatically for the deterministic address generations patterns, just like the traditional designs [10]. The EAGU is constructed by 3 types of resources: shared ASP pipeline resources, shared ASP functional units and Addr-RF. The Addr-RF is an additional register file different to the traditional processor architecture which is dedicated for the address processing. By sharing the ASP resources, especially the powerful computing units, the extended AGU has the capability of complex address calculations.

Besides the hardware design, the instruction set architecture (ISA) is also optimized. The related instructions for address generation include two sub sets: post load/store instructions and stand-alone address calculation instructions. The former is designed for the automatically address updating for the deterministic address generation pattern. The latter is designed for complex address calculation.

With SAGU and the corresponding address calculation instructions, the address generation operations of BP-ASP can be depicted in Fig. 6. For the address generation related instructions, the instruction decode and dispatch unit distinguishes the stand-alone address calculation instructions and the post load/store ones. For the post load/store instructions, the address updating is calculated by the RAGU with deterministic pattern. For the stand-alone address calculation instructions, the address generation process includes the following steps. Firstly, the ASP read the source address registers from Addr-RF. Secondly, the source address operands are sent to the functional computing unit of EAGU. Finally, the calculated result from the functional unit is sent to memory access controller as the effective address and simultaneously wrote back to Addr-RF for further calculations.

Profiling result shows that a considerable amount of code is avoided with SAGU, which is used to manipulate the memory access address. The overall algorithm performance is improved at a relatively small hardware cost.

III. BP-ASP BASED MULTI-CORE BASEBAND PROCESSOR

Based on the proposed BP-ASP, a triple-core baseband processor is presented whose system architecture is shown in Fig. 7. To evaluate the performance of the presented baseband processor, the LTE inner part of the downlink receiver is mapped on it. The inner receiver implements LTE equalization, de-interleaving, and channel estimation which are suitable for ASP profiling. The modulation and coding is set as 16-QAM and 1/2 Turbo. The detail profiled information, including average power consumption, executes clock cycles of the kernel algorithms, are shown in TABLE III.

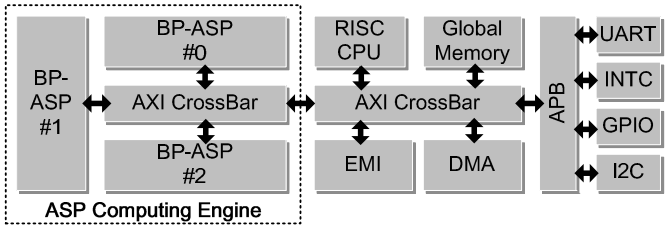

Fig. 7. The overview of the BP-ASP based baseband processor

The profiling results show that the proposed baseband processor meets the requirement of LTE bandwidth from 1.4MHz to 20MHz. The downlink data rate is 100 Mb/s and the uplink data rate is 50Mb/s, which meets the throughput and latency requirements of CAT3 UE.

TABLE III. INNER RECEIVER PROFILING RESULT ON BP-ASP

Kernel algorithms	Power(mw)	Cycles	Energy(nJ)
Equalization	120.5	847	868
DeInterleave	120.7	2605	2674
Channel Estimation	121.1	1569	1616
Synchronization	122.1	26580	27597
2048-point FFT	166.5	2352	3330

IV. SILICON IMPLEMENTATION

The baseband processor based on BP-ASP is implemented with 130 nm CMOS technology. The work frequency is set to 117.6MHz. The gate count of one BP-ASP is about 1.3 million and the memory usage is 256k byte. The average power consumption is 120 mw. In TABLE IV. different ASPs are compared. The results show that the proposed BP-ASP architecture could provide a high performance with relative low power consumption even in the 130nm technology.

The post layout area of the overall baseband processor is 57.8 mm^2. TCM contribute to almost 70% of this area. One notable problem is that the ultra-wide bit-width of the SIMD processor introduced many difficulties for the chip place & routing process and the cell utilization is just about 60%. As discussed in Section II, the proposed BP-ASP architecture provides 298 operations per cycle at the peak computing capability (2 operations in VLIW DP #0, 200 operations in VLIW DP #1, 32 operations in VLIW DP #2, 32 operations in VLIW DP #4 and 32 operations in VLIW DP #5). Single BP-ASP has a 35 GOPS peak performance. So the triple-core BP-ASP based baseband processor has a 105 GOPS peak computing performance at 117.6MHz working frequency.

TABLE IV. COMPARISON OF DIFFERENT APPLICATION SPECIFIC PROCESSORS

Processor	Architecture	Performance	Area	Power
Sandblaster	SIMD	30 GMACs @ 600MHz	N/A	N/A
NXP EVP	SIMD VLIW	N/A	2 mm^2 @ 90nm	300mw
ARM Ardbeg	SIMD VLIW	40 GOPS @ 300MHz	3.5 mm^2 @ 65nm	500mw
Proposed BP-ASP	SIMD VLIW	35 GOPS @ 117.6MHz	8.4 mm^2 @ 130nm	120mw

V. CONCLUSIONS

In this paper, an efficient ASP architecture is presented, which provides an instruction parallelism of 6, a data parallelism of 2080-bit and more parallel capability for address calculation and memory access. Furthermore, a multi-core baseband processor was built based on the proposed ASP. Finally, a silicon implementation of the baseband processor with 130nm CMOS technology is presented. Experimental results show that the baseband processor can provide more than 100 GOPS computing ability at a relative low frequency.

REFERENCES

[1] L. Tian, Y. Zhou, Y. Zhang, G. Sun and J. Shi, "Resource Allocation for Multicast Services in Distributed Antenna Systems with QoS Guarantees", IET Communications, vol. 6(3), pp. 264–271, Feb.2012.

[2] Y. Zhou, J. Wang, T.S. Ng, K. Higuchi and M. Sawahashi, "OFCDM: a promising broadband wireless access technique", IEEE Communications Magazine, vol. 46, pp. 39-49, Mar. 2008.

[3] C.H. Van Berkel, "Multi-Core for Mobile Phones", invited paper, Design, Automation and Test in Europe DATE'09, 2009.

[4] J. Glossner, D. Iancu, J. Lu, E. Hokenek, M. Moudgill, "A software-defined communications baseband design", IEEE Communications Magazine, vol. 41(1), pp. 120-128. 2003.

[5] K. Van Berkel, F. Heinle, P.P.E. Meuwissen, K. Moerman, M. Weiss, "Vector processing as an enabler for software-defined radio in handheld devices", Eurasip Journal on Applied Signal Processing, vol. 16, pp. 2613-2625. 2005.

[6] K. Keutzer, S. Malik, and A. Richard Newton, "From ASIC to ASIP: The Next Design Discontinuity", IEEE Int. Conf. on Computer Design: VLSI in Computers and Processors, pp 84-90, 2002.

[7] Y. Lin, H. Lee, M. Woh, Y. Harel, S. Mahlke, T. Mudge, C. Chakrabarti, K. Flautner, "SODA: A high-performance DSP architecture for software-defined radio", IEEE Micro, vol. 27(1), pp. 114-123, 2007.

[8] M. Woh, Y. Lin, S. Seo, S. Mahlke, T. Mudge, C. Chakrabarti, R. Bruce, et al., "From SODA to scotch: The evolution of a wireless baseband processor", Proc. of the Annual Int. Symposium on Microarchitecture, MICRO, pp. 152-163. 2008.

[9] H. Lee, C. Chakrabarti, T. Mudge, "A low-power DSP for wireless communications", IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 18 (9), pp. 1310-1322. 2010.

[10] M. A. Shami, A. Hemani, "Address generation scheme for a coarse grain reconfigurable architecture", Proc. of Int. Conf. on Application-Specific Systems, Architectures and Processors, pp. 17-24, 2011.