Transactions Briefs

HAVA: Heterogeneous Multicore ASIP for Multichannel Low-Bit-Rate Vocoder Applications

Zhenqi Wei, Peilin Liu, Rongdi Sun, Jun Dai, Zunquan Zhou, Xiangming Geng, and Rendong Ying

***Abstract*— As are widely used in military and security fields, multiple channels of low-bit-rate vocoders are required to perform on embedded devices efficiently. We propose HAVA, a multicore Application Specific Instruction Set Processor for multichannel low-bit-rate vocoders with real-time performance. To provide both flexibility and efficiency, HAVA integrates two types of processing cores and a shared-memory core on a 2-D-mesh on-chip network. Adopting a single-Instruction Set Architecture heterogeneous multicore architecture, HAVA cuts down the real-time performance requirement of vocoders by over 40% compared with other platforms. By leveraging the on-chip network for intercore communication, HAVA can perform multichannel vocoders with a marginal efficiency loss. The chip implementation of HAVA is finished in a 40-nm CMOS technology and it dissipates 149 mW at 100-MHz operating frequency for four channels of encoders.**

***Index Terms*— ASIP, heterogeneous multicore, low-bit-rate vocoder, network-on-chip (NoC).**

I. Introduction

Vocoders at low bit rates are commonly used in military and security fields. For the implementation of vocoders in embedded systems, some solutions use general purpose (GP) DSPs to accelerate the execution of vocoder programs, while others adopt commercial chips dedicated to a certain vocoder standard. As the bit rate goes lower, the efficiency improvement gained from GP processors becomes less significant due to the increasing computing complexity of vocoder algorithms. Dedicated vocoder chips usually lack scalability for the advancement of algorithms. Moreover, the demand for multichannel processing in practical use also brings challenges to the design of processors due to the huge memory usage of codebook data in vocoder algorithms.

In order to provide both flexibility and efficiency, we propose HAVA, a heterogeneous multicore ASIP for low-bit-rate vocoder applications with real-time efficiency and multichannel processing capability. The high efficiency of HAVA is gained as a result of threefold contribution: 1) a very-long-instruction-word (VLIW) pipeline with an instruction set extension (ISE) in HAVA processing cores (HPCs) to accelerate computing-intensive operations within vocoders; 2) a single-ISA heterogeneous multicore architecture integrating two types of HPCs and an HAVA shared-memory core (HMC) in an 2-D-mesh on-chip network; and 3) a frame-level optimization approach for vocoders running on HAVA. We finish the chip implementation of HAVA in a 40-nm CMOS technology. The evaluation results show that the real-time performance requirement of low-bit-rate vocoders on HAVA is saved by over 40% compared with commercial DSPs and GP processors. Besides, HAVA supports at most four channels of encoders or eight channels of decoders concurrently.

This brief is organized as follows. Section II gives a brief introduction of low-bit-rate vocoder algorithms and prior embedded solutions. Section III presents the heterogeneous multicore architecture design of HAVA. In Section IV, the chip implementation results of HAVA are demonstrated. Finally, the conclusion is drawn in Section V.

II. Background and Related Works

At bit rates below 4 kb/s, most vocoders extract and compress only the key perceptual features by encoding the parameters of a linear model for speech, instead of encoding the entire waveforms [1]. Three models have been extensively researched to attain significant performance improvements compared with the traditional linear predictive coding, which are mixed excitation linear prediction (MELP), sinusoidal coding (SC), and waveform interpolation. MELP with enhancement (MELPe) is adopted as the NATO vocoder standard at 2400/1200/600 bit/s [2], and it is also the basis of vocoders at 300 bit/s or lower bit rates [3], [4]. The vocoder software running on HAVA employs a superframe structure based on MELPe [4] to reduce algorithm delay.

Several implementations of low-bit-rate vocoders based on embedded processors have been presented. Liu *et al.* [5] and Pang *et al.* [6] optimize MELPe vocoders at 2.4 and 1.2 kb/s on ARM9 and ARM11 processors, respectively. Commercial DSP platforms are also used to implement vocoders, such as ADI ADSP-21065L for wideband SC at 7.8 kb/s [7], Texas Instruments Inc. (TI) TMS320VC5509A for MELP at 2.4 kb/s [8], and DVSI AMBE-3000 for AMBE+2 algorithm at 2∼2.4 kb/s [9]. AMBE-3003 [10], which is based on TI TMS320F2811 core with highly optimized programs running on it, supports up to three channels of vocoders at 2∼9.6 kb/s. To the best of our knowledge, HAVA is the first chip implementation of a vocoder processor supporting multichannel processing of vocoders at 300 bit/s.

III. Design of HAVA

A. VLIW Pipeline With ISE in HPC

By analyzing the fixed-point C program of MELPe 300-bit/s vocoder, we find that the execution time of the vocoder is mostly consumed by compound operations coupled with postprocessing of results and addresses for digital signal processing of speech data, which are invoked by most internal functions all over the application for tens of million times. According to the profiling results gained from an MIPS32/Linux platform, Table I lists the top ten time-consuming operations, which together occupy ∼80% of the total run time. In order to save the execution time by using a single instruction to complete a compound operation, that usually takes multiple instructions in a GP processor, we propose an ISE based on a 32-bit MIPS-like instruction set and a dedicated data path with an autoaddressing unit to perform multiple ways of the compound operations in parallel. Slightly different from [11] and [12], which have presented design details, the ISE data path is adjust to four internal stages, including two stages for multiplication, one for addition, and one for postprocessing. Intermediate results of the

Manuscript received June 25, 2015; revised September 24, 2015 and December 2, 2015; accepted December 14, 2015. Date of publication January 18, 2016; date of current version June 23, 2016.

The authors are with Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: awaylovemusic@sjtu.edu.cn; liupeilin@sjtu.edu.cn; srd92@163.com; daijun19900319@163.com; zqzhou.363@gmail.com; xmgeng@sjtu.edu.cn; rdying@sjtu.edu.cn).

Digital Object Identifier 10.1109/TVLSI.2015.2509459

TABLE I
TOP TEN TIME-CONSUMING OPERATIONS IN MELPe 300-bit/s VOCODER

Operations in Enc	Time	Operations in Dec	Time
32-bit Add w/ Sat[1]	27.8%	32-bit Add w/ Sat	22.1%
32-bit Left Shift w/ Sat	14.8%	16-bit Mul w/ Sat	13.4%
16-bit MAC[2] w/ Sat	12.8%	16-bit MAC w/ Sat	8.0%
Multi-Stage VQ[3]	5.2%	Harmonic Synthesis	7.5%
16-bit Inner Product	4.7%	32-bit Sat	7.4%
16-bit Right Shift w/ Sat	4.2%	FIR Filtering	6.6%
16-bit FFT	3.6%	32-bit Left Shift w/ Sat	5.1%
16-bit Mul w/ Sat	3.0%	Post Filtering	3.2%
32-bit Right Shift w/ Sat	2.7%	32-bit Sub w/ Sat	3.1%
2nd-Order IIR Filtering	2.3%	16-bit Add w/ Sat	3.0%
Total	**81.8%**	**Total**	**79.4%**

[1] Saturation
[2] Multiple-And-Accumulate
[3] Vector Quantization

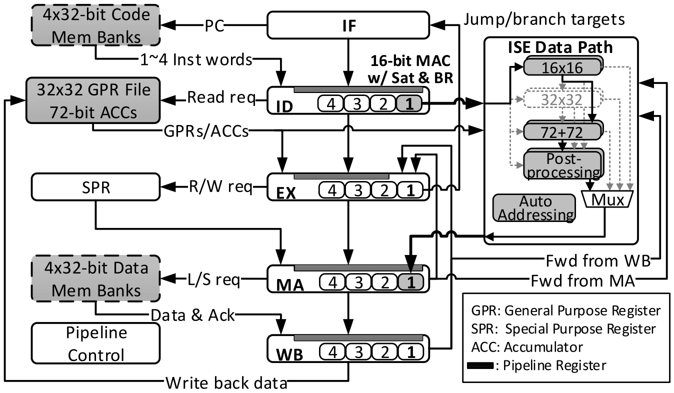


Fig. 1. VLIW pipeline of HPC.

compound operations are latched by interstage registers of the data path, instead of repeatedly being saved to General Purpose Register file and then being read out between multiple instructions in GP processors, which improve the execution efficiency of the critical operations within the vocoder application. Besides, leveraging the stage-skipping technique [12], the ISE data path allows the execution of extended instructions to skip unnecessary stages to further save cycles. As shown in Fig. 1, for example, the second stage for 32×32 multiplication is skipped when the data path is performing an instruction of 16-bit MAC with saturating (Sat) and bit-reverse addressing for fast Fourier transform.

Further analysis of the profiling results reveals that ~30% of total instructions have no dependence with their contexts. Thus, we apply VLIW technology to the canonical five-stage RISC pipeline to provide instruction-level parallelism. As shown in Fig. 1, the issue width of HPC pipeline is expanded to four ways. According to the compile-time scheduling of instructions, the first slot in the Instruction Decode stage will always have a valid instruction, while the other three slots in the following stages may be injected with bubbles to save power consumption. Considering both the hardware overhead of the ISE data path (23% area and 16% power of the HPC pipeline) and the scheduling of VLIW slots (one slot for the ISE at most), we permit only the first slot of the pipeline to feed extended instructions in the ID stage to the ISE data path and receive operation results in the Memory Access stage.

The implementation of the compiler for HPCs with various issue widths is based on the LLVM compiler infrastructure [13]. The HAVA compiler supports not only the original instruction set but also part of the ISE, such as fixed-point scaling instructions. It is necessary to embed assembly instructions into C codes to fully utilize the high efficiency of the ISE. Using processor designer [14], we implement a cycle-accurate instruction set simulator for the proposed VLIW pipeline. According to the execution cycles of MELPe 300-bit/s vocoder acquired from the simulator, we achieve a 2.2× efficiency gain after the ISE is adopted.

B. Single-ISA Heterogeneous Multicore Architecture of HAVA

Since the codebook of MELPe 300-bit/s vocoder contributes half of the total memory cost for one channel, we make the codebook data in an HMC be shared between all HPCs for multichannel processing to reduce the total memory size of HAVA. We integrate all cores in an on-chip network with 2-D mesh topology instead of bus-based architecture, since the latter will result in severe bus contention issues when the shared codebook data are accessed by increasing vocoder channels concurrently. As demonstrated in Fig. 2(a), the HMC is located in the center of a 3 × 3 mesh network-on-chip (NoC) and is surrounded by four big HPCs and four little HPCs. Thus, the distance for intercore communications is limited within two hops during the processing of multichannel vocoders, which is explained in Section III-C. Fig. 2(b) shows that inside an HPC, the VLIW pipeline is connected to several units for intercore communications, which are all globally addressable and thus can be accessed by other HPCs via a network interface. As shown in Fig. 2(c), the HMC allows at most four incoming requests to access its memory banks or to configure any channel of a Direct Memory Access (DMA) engine simultaneously. With proper scheduling of requests by a synchronization controller (SC), the HMC can provide a maximal data throughput of 4 × 32 bits per cycle. The SC supports various synchronization schemes, including spinlock, barrier, and semaphore with low latency, which is presented in [15].

HAVA adopts packet switching and circuit switching in two subnetworks Pnet and Cnet of the 2-D-mesh NoC, respectively. Routers [see Fig. 2(d)] use XY routing algorithm to direct flits in the Pnet for transferring small amounts of data, while crossbars [see Fig. 2(e)] forward bulk data for DMA transfers in the Cnet according to their connection configuration, which is similar to [16].

C. Frame-Level Optimization of Vocoders on HAVA

As mentioned in Section III-B, HAVA integrates two types of HPCs in the on-chip network. The little HPC has exactly the same ISA and core architecture as the big one except that its VLIW issue width is reduced from four to two and the size of its local memory is halved, which saves 46% circuit area and 29% power consumption. Accordingly, the two-issue HPC loses 15% efficiency for MELPe 300-bit/s vocoder compared with the four-issue architecture. Based on the careful analysis of the algorithm structure, we apply both big and little HPCs to perform the encoder program collaboratively for better performance, which is demonstrated in Fig. 3. In a little HPC, once a superframe finishes part of noise preprocessing, which consumes 30% time of a single-big-HPC solution, the processed data are sent to a neighboring big HPC by means of DMA transfer and mailbox's message passing. Then, the big HPC will perform parameter extraction (45% of total time) frame by frame, and then vector quantization (25% of total time). With usage efficiencies of 51% for the little HPC and 84% for the big HPC, the task allocation for the two HPCs yields a 1.4× performance speedup compared with the single-core solution. For the decoder with only half task load of the encoder, one HPC at full usage efficiency is sufficient to achieve real-time processing.

More HPCs can also be allocated for single channel of low-bit-rate vocoders to process frames in parallel, which can provide higher performance at the cost of lower usage efficiency in average.

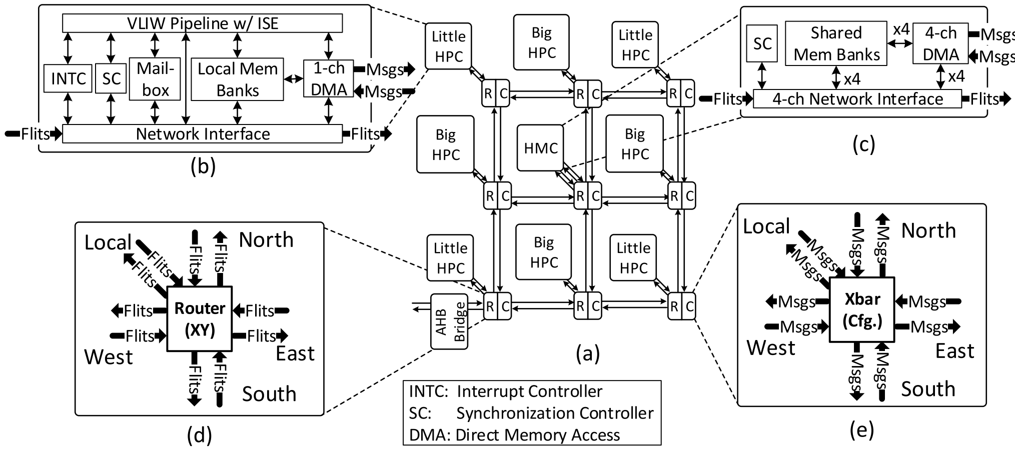


Fig. 2. Heterogeneous multicore architecture of HAVA. (a) 3 × 3 mesh NoC. (b) HPC including VLIW pipeline and units for intercore communication. (c) HMC including memory banks and units for intercore communication. (d) Five-ported router in Pnet. (e) Five-ported crossbar in Cnet.

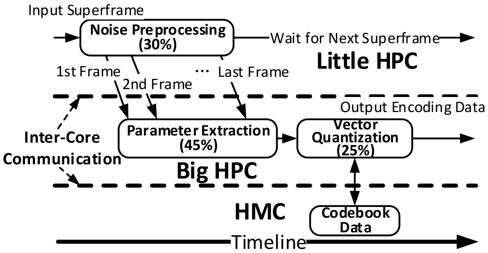


Fig. 3. Task allocation of MELPe encoder on a big and a little HPC.

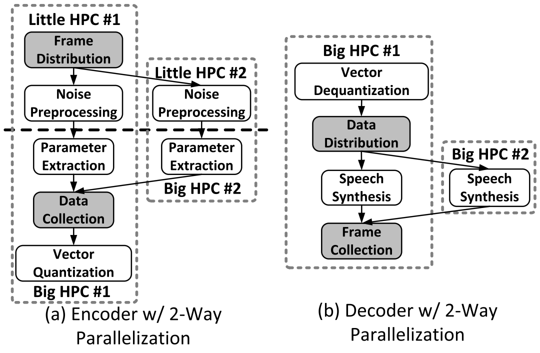


Fig. 4. Frame-level parallelization for single-channel vocoder.

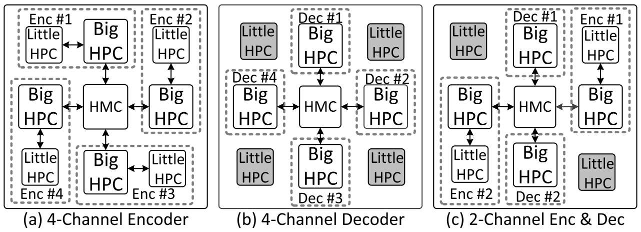


Fig. 5. Support for multichannel vocoders in HAVA.

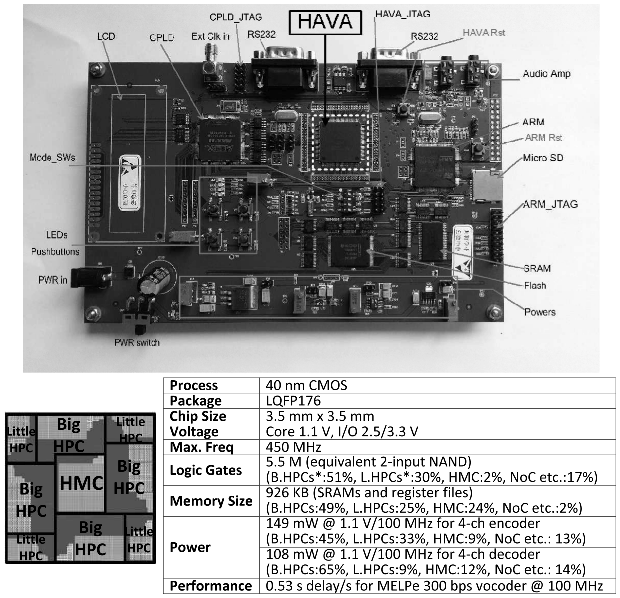
Fig. 4 shows the frame-level parallelization of MELPe encoder and decoder running on multiple big and little HPCs. For a single-channel encoder, input frames contained in a superframe are distributed to little HPCs using DMA transfer before noise preprocessing, while after the parameter extraction task, the processed data are collected from big HPCs. Similarly, in a single-channel decoder, multiple HPCs are used to perform speech synthesis for different frames after the vector dequantization task. Leveraging the packet-circuit switching on-chip network of HAVA, we limit the overhead of data distribution and collection between HPCs within 7% of total processing time. The performance improvement gained from frame-level parallelization is evaluated in Section IV.

The heterogeneous multicore architecture of HAVA allows it to support multichannel processing of low-bit-rate vocoders with most flexibility. Fig. 5 presents various task allocations of on-chip cores to perform a four-channel encoder, a four-channel decoder, and two channels of encoder and decoder, respectively. The codebook data stored in memory banks of the HMC can be simultaneously accessed by all HPCs using memory access instructions or DMA transfers within a two-hop distance. Idle HPCs drawn as gray boxes in Fig. 5 can be turned OFF using the clock gating technique to save energy consumption. In order to minimize the average distance of intercore communications for multichannel vocoders based on software reconfiguration, we first manually allocate for each channel of the encoder a big HPC and a neighboring little HPC, and then reconfigure the rest HPCs with preference for big ones to perform the decoder with one core per channel. Considering the usability of HAVA in embedded systems, we add a dedicated Advanced High-Performance Bus (AHB) bridge at one of the available router ports in HAVA's Pnet to support AMBA AHB-Lite compatible protocol, which is shown in Fig. 2(a). In our test platform, an ARM Cortex M1 is used to interact with HAVA during speech processing.

IV. Implementation Results

A test platform for the chip implementation of HAVA is shown in Fig. 6, along with the on-die layout and the performance summary of the chip. As a multicore ASIP used for more than low-bit-rate vocoders, HAVA has a chip size of 12 mm^2 in a 40-nm CMOS technology and can run at a maximal frequency of 450 MHz. The 3 × 3 mesh on-chip network of HAVA contributes 17% logic gates and 13% power consumption. The HMC uses about one quarter of the total on-chip memory and it occupies 19% chip area. Compared with the AMBE-3003 [10] which supports at most three channels of 2-kb/s vocoders at 1.1 V/150 MHz with restrictions in its use, HAVA can perform multiple channels of 300-bit/s vocoders in various scenarios at 1.1 V/100 MHz with much higher flexibility and low-power consumption. The processing delay of single-channel vocoder is 0.53 s for 1-s speech.

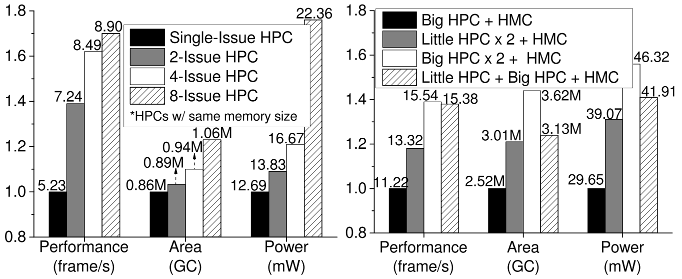
We make a comprehensive comparison between HPCs applying various VLIW issue widths for design exploration. It should be



Process	40 nm CMOS
Package	LQFP176
Chip Size	3.5 mm x 3.5 mm
Voltage	Core 1.1 V, I/O 2.5/3.3 V
Max. Freq	450 MHz
Logic Gates	5.5 M (equivalent 2-input NAND) (B.HPCs*:51%, L.HPCs*:30%, HMC:2%, NoC etc.:17%)
Memory Size	926 KB (SRAMs and register files) (B.HPCs:49%, L.HPCs:25%, HMC:24%, NoC etc.:2%)
Power	149 mW @ 1.1 V/100 MHz for 4-ch encoder (B.HPCs:45%, L.HPCs:33%, HMC:9%, NoC etc.: 13%) 108 mW @ 1.1 V/100 MHz for 4-ch decoder (B.HPCs:65%, L.HPCs:9%, HMC:12%, NoC etc.: 14%)
Performance	0.53 s delay/s for MELPe 300 bps vocoder @ 100 MHz

*B.HPC: Big HPC; L.HPC: Little HPC

Fig. 6. Test platform, on-die layout, and performance summary of HAVA.



(a) (b)

Fig. 7. Performance and cost evaluation of HPCs. (a) HPCs with various issue widths. (b) Encoders with various solutions.

TABLE II

PERFORMANCE OF MELPe VOCODERS ON DIFFERENT PLATFORMS

MELPe Vocoder	Platform	Memory Size (KB)	Real-Time Perf. (MCPS)[1]
300 bps	TI C55@100 MHz	262	90
	TI C64@500 MHz	285	81
	Intel E2160[2]@2.0 GHz	226	289
	HAVA@100 MHz	**262**	**53**
600 bps	TI C54@140 MHz [2]	180	103
	Intel E2160@2.0 GHz	188	225
	HAVA@100 MHz	**225**	**46**
1.2 Kbps	TI C54@140 MHz [2]	166	98
	ARM11@482 MHz [6]	N/A	(820 ms)
	HAVA@100 MHz	**187**	**41 (410 ms)**
2.4 Kbps	TI C54@140 MHz [2]	80	79
	TI C55@200 MHz [8]	N/A	62
	ARM9@100 MHz [5]	N/A	95
	HAVA@100 MHz	96	**37**

[1] Cycles per frame / frame length
[2] Using single thread of single core

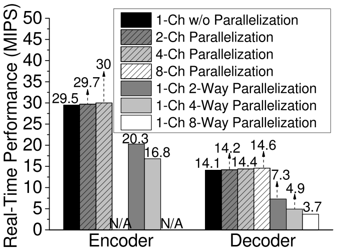


Fig. 8. MELPe 300-bit/s vocoders with various parallelization methods.

TABLE III

CYCLE COUNTS OF BENCHMARKS ON DIFFERENT PLATFORMS

Benchmark	ARM926EJ-S	TI C6416	Big HPC
2048-p MDCT	152 K	70 K	38 K
512-p FFT	100 K	49 K	27 K
5.1-ch Downmix	2.7 K	2.2 K	0.9 K
AAC LC Decoder	178 K	278 K	98 K

noted that all compared HPCs running at 100 MHz have same memory size in order to perform a complete 300-bit/s vocoder application. In Fig. 7(a), the performance and the cost of a single-issue HPC are scaled to one for clarity. Compared with a single-issue core, an HPC adopting a four-way VLIW pipeline (big HPC) gains a 1.62× performance speedup at the cost of 10.1% area and 21.4% power overhead, which achieves the best tradeoff between performance improvement and hardware cost among different design alternatives.

As explained in Section III-C, the MELPe 300-bit/s encoder can run on two HPCs in a pipeline fashion. Fig. 7(b) shows the efficiency speedup and hardware overhead of various dual-HPC solutions compared with the baseline with one big HPC and one HMC. Two little HPCs provide 18% performance gain, while the other two solutions have similar speedups of about 1.4×. Since the parameter extraction task dominates the total execution time, allocating the noise preprocessing task on a big HPC does not bring more improvements, but incurs 16% more area and 11% more power costs than the dual-HPC solution with two types of HPCs. Thus, we decide to use one little and one big HPCs for the encoder application.

Table II shows the memory and performance requirements of single-channel vocoders for real-time processing on various platforms, which all apply the MELPe algorithm with corresponding length of superframe for vocoder cases at 300∼2400 bit/s. For the 300-bit/s case, we apply the same coding style and compiler optimization on different platforms as much as possible to ensure the fairness of comparisons. The real-time performance requirements in terms of million cycles per second are determined by the architectures of various platforms regardless of their variance on working frequency or process technology. As explained in Section III-C, two heterogeneous HPCs with six VLIW ways in all are used for a single-channel encoder, while one big HPC with four ways is used for a single-channel decoder. Without frame-level parallelization, HAVA outperforms TI DSPs and ARM processor by over 40% for vocoders at bit rates ranging from 300∼2400 bit/s. The memory cost of MELPe vocoders on HAVA is slightly higher than other platforms, and codebook data occupy over half of the memory.

With the help of DMA engines and SCs for high-efficient intercore communication, the HMC can offer maximal data throughput for all working HPCs when they acquire different codebook data for encoding or decoding tasks simultaneously. The performance results of multichannel vocoders (bars with slashes) given in Fig. 8 indicate that the sharing of codebook data between multiple channels leads to insignificant performance degradation since most access conflicts can be avoided by well-scheduled requests to the HMC. As explained in Section III-C, frame-level parallelization for single-channel vocoders can achieve higher performance. In the same figure, an encoder with four-way parallelization decreases 42.3% performance requirement of an encoder baseline with two HPCs, while an eight-way decoder only needs 25% instructions of a nonparallelized decoder. An eight-channel encoder and a single-channel encoder with eight-way parallelization both require 16 HPCs with two cores per channel (way), so they are not compared with other solutions using at most eight HPCs.

As shown in Table III, a big HPC saves more than 45% cycles of other platforms when performing selected benchmarks, which also proves the high efficiency of HAVA for audio and speech processing.

V. Conclusion

In this brief, a multicore ASIP HAVA based on a 3 × 3 mesh NoC is proposed for real-time processing of multichannel vocoders at low bit rates. HAVA occupies 12.25-mm^2 die size in a 40-nm technology and consumes 149 mW at 100 MHz for four channels of MELPe encoders. Compared with other embedded processors, HAVA improves the performance of low-bit-rate vocoders by over 40% with nearly no efficiency loss for multichannel processing. Besides, the multicore architecture of HAVA with high scalability allows it to further reduce 42%∼75% processing time of single-channel vocoders after frame-level parallelization.

References

[1] A. V. McCree, "Low-bit-rate speech coding," in *Springer Handbook of Speech Processing*. Heidelberg, Germany: Springer, 2008, pp. 331–350.

[2] G. Guilmin, F. Capman, B. Ravera, and F. Chartier, "New NATO STANAG narrow band voice coder at 600 bits/s," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2006, p. 1.

[3] A. McCree, K. Brady, and T. F. Quatieri, "Multisensor very low-bit rate speech coding using segment quantization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar./Apr. 2008, pp. 3997–4000.

[4] X. Zou, X. Zhang, and Y. Zhang, "A 300 bps speech coding algorithm based on multi-mode matrix quantization," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2009, pp. 1–4.

[5] S. Liu, Y. Zhang, J. Hong, and J. Tang, "Optimization of 2.4 kbps MELPe based on ARM9," in *Proc. IEEE 15th Int. Symp. Consum. Electron. (ISCE)*, Jun. 2011, pp. 204–207.

[6] X. Pang, L. Li, and X. Zhao, "Optimization of 1.2 kbps MELPe based on ARM11," in *Proc. 5th Int. Conf. Comput. Inf. Sci. (ICCIS)*, Jun. 2013, pp. 1660–1662.

[7] A. P. Q. Unisa and R. C. L. Guevara, "Real-time implementation of wideband sinusoidal speech coder on ADSP-21065L," in *Proc. 16th Int. Conf. Digit. Signal Process.*, Jul. 2009, pp. 1–5.

[8] Y. P. Shi, "Research and implementation of MELP algorithm based on TMS320VC5509A," *Adv. Mater. Res.*, vol. 934, pp. 239–244, May 2014.

[9] J. Luo, H. Zhou, Y. Li, and M. Lu, "The design of underwater speech communication codec system based on AMBE-3000," in *Proc. Int. Conf. Mechatronic Sci., Electr. Eng. Comput. (MEC)*, 2011, pp. 2471–2475.

[10] Digital Voice Systems, Inc. *AMBE-3003 Vocoder Chip*. [Online]. Available: http://www.dvsinc.com/products/a3003.htm, accessed Sep. 3, 2015.

[11] J. Kong *et al.*, "Next-generation consumer audio application specific embedded processor," in *Proc. IEEE 8th Symp. Appl. Specific Process. (SASP)*, Jun. 2010, pp. 1–7.

[12] Z. Q. Wei, P. L. Liu, J. Kong, and R. D. Ying, "Design of arithmetic operation core in embedded processor for high definition audio applications," *Appl. Mech. Mater.*, vol. 538, pp. 289–292, Apr. 2014.

[13] *The LLVM Compiler Infrastructure*. [Online]. Available: http://llvm.org/, accessed Sep. 3, 2015.

[14] Synopsys. *Processor Designer*. [Online]. Available: http://www.synopsys.com/IP/ProcessorIP/asip/processor-designer/, accessed Sep. 3, 2015.

[15] Z. Wei, P. Liu, Z. Zeng, J. Xu, and R. Ying, "Instruction-based high-efficient synchronization in a many-core network-on-chip processor," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2014, pp. 2193–2196.

[16] Z. Yu *et al.*, "An 800 MHz 320 mW 16-core processor with message-passing and shared-memory inter-core communication mechanisms," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2012, pp. 64–66.