Microprocessors and Microsystems 40 (2016) 88–101

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Design space exploration for device and architectural heterogeneity in chip-multiprocessors

Ying Zhang[a,*], Samuel Irving[a], Lu Peng[a], Xin Fu[b], David Koppelman[a], Weihua Zhang[c,d], Jesse Ardonne[a]

[a] Division of Electrical & Computer Engineering, School of Electrical Engineering and Computer Science, Louisiana State University, Baton Rouge, LA 70803, United States
[b] Department of Electrical and Computer Science, University of Houston, Houston, TX 77004, United States
[c] Software School, Fudan University, 201203, China
[d] Parallel Processing Institute, Fudan University, Shanghai, 201203, China

ARTICLE INFO

Keywords:
Heterogeneity
Cost efficiency
Energy efficiency

ABSTRACT

As we enter the deep submicron era, the number of transistors integrated on die is exponentially increased. While the additional transistors largely boost the processor performance, a repugnant side effect caused by the evolution is the ever-rising power consumption and chip temperature. It is widely acknowledged that the shortage of power supplied to a processor will be a major hazard to sustain the generational performance scaling, if the processor design is to follow the conventional approach. To utilize the on-chip resources in an efficient manner, computer architects need to consider new design paradigms that effectively leverage the advantages of modern semiconductor technology. In this paper, we address this issue by exploiting the device-heterogeneity and two-fold asymmetry in the processor manufacturing. We conduct a thorough investigation on these design patterns from different evaluation perspectives including performance, energy-efficiency, and cost-efficiency. Our observations can provide insightful guidance to the design of future processors.

1. Introduction

Processor manufacturers have been able to double transistor count and performance for each new product generation in past decades, as predicted by Moore's Law. However, as we enter the deep submicron era, the continuous decrease of the transistor supply and threshold voltage at each new technology node, known as Dennard Scaling has stalled [18,28], leading to an ever-increasing power density on modern processors. On the other hand, the maximum processor power consumption should always be enclosed within a reasonable envelope, regardless of manufacturing technology due to physical constraints such as heat dissipation and power delivery. Given these limitations, a large portion of the integrated transistors on a future processor must be significantly underclocked or even turned off in order to satisfy power constraints and maintain a safe working temperature. This phenomenon, which has been termed "dark silicon" [18], is recognized as one of the most critical constraints preventing us from obtaining commensurate performance benefits from increasing the number of transistors.

The problem might become exacerbated as Moore's Law continues to dominate processor development. According to the ITRS roadmap [5], the percentage of the chip that cannot be turned on is exponentially expanding with each generation, and up to 93% of all transistors on a chip would be forced inactive in a few years from now. Therefore, seeking new design dimensions to efficiently utilize chip-level resources including power and area is important for us to obtain sustainable performance improvements in the future. In this paper, we conduct a comprehensive assessment of new design dimensions with special concentration on heterogeneity in the early stage of processor manufacturing.

Our target processor is a chip multiprocessor (CMP) with a fixed power and area budget. The first dimension that will be evaluated is *device heterogeneity*. Since the gap between power requirement and supply capability is essentially caused by the slow improvement in a Complementary Metal–Oxide–Semiconductor (CMOS) device's switch power, emerging low-power materials might be used to fabricate processors in order to illuminate the dark area. However, many power-saving devices manufactured with nano-technology manifest a series of drawbacks such as long switch delay [21]. Due to this limitation, it is inappropriate to use such devices to completely replace the traditional CMOS in processor

* Corresponding author.
E-mail addresses: ying.esz.zhang@gmail.com (Y. Zhang), sirvin1@lsu.edu (S. Irving), lpeng@lsu.edu (L. Peng), xfu6@uh.edu (X. Fu), koppel@lsu.edu (D. Koppelman), zhangweihua@fudan.edu.cn (W. Zhang), jardon2@lsu.edu (J. Ardonne).

http://dx.doi.org/10.1016/j.micpro.2015.07.012
0141-9331/

manufacturing. Instead, integrating cores made of different materials on the same die emerges as an attractive design option. A few works have justified the feasibility of a hybrid-device CMP at the circuit level [24,31,33]. On the other hand, *architectural heterogeneity* (e.g., including both big and small cores on a processor) has proven to be an effective way to improve energy efficiency [25]. Therefore, jointly applying device and architectural heterogeneity becomes a promising option compared to conventional designs, hence the second design dimension "two-fold heterogeneity". The third aspect considered in this study is the *operating voltage/frequency* (*v/f*) of processors since it significantly impacts the processor power and thermal characteristics. Finally, the last factor that will be taken into consideration is a recently proposed technique "*computational sprinting*" [28] which allows the system to temporarily exceed the thermal-design power constraint in a burst fashion. In general, by evaluating the described dimensions in detail, we attempt to summarize a set of "principles" that can guide the design of processors in the next generation and beyond. The following is a list of the main observations made in this study.

- We demonstrate that the on-chip resources can be more efficiently utilized by using diverse materials in the chip fabrication. By integrating more cores made of slower power-saving devices and less cores built with faster yet power-consuming devices, more processor cores can be booted up, thus delivering better energy- and cost-efficiency.
- We explore processor designs with two-fold heterogeneity with regards to both manufacturing devices and core architectures. We show that by building complex out-of-order cores using power-saving devices while in conjunction with small in-order cores using relatively power-consuming material, we are able to deliver extra energy- and cost-efficiency benefits.
- We examine the impact of the voltage/frequency setting on the overall performance, energy- and cost-efficiency of the target processor. Our evaluations demonstrate that the most promising design pattern remains the same (i.e., building big cores with power-saving devices and small cores with faster devices) although appropriately setting the operating voltage/frequency can effectively increase the performance and efficiency of other configurations.
- We enable the computational sprinting technique on the target system and investigate its implication on the design pattern selection. The results show that this technique is capable of delivering better performance and execution efficiencies than regular configurations. Moreover, as for the distribution of the extra power in the sprinting phase, an "even" distribution (i.e., increase the frequency of all cores by an amount) is more preferable than "prioritized" distribution which gives all extra power to a few cores (e.g., the big cores).

2. Related work

The problem of power supply shortage for activating transistors (i.e., dark silicon) emerges as an increasingly important issue that jeopardizes the scaling of Moore's Law in the deep submicron era and beyond. For this reason, researchers recently started to investigate this problem and propose several solutions. Esmaeilzadeh et al. [18] use an analytical model to predict processor scaling for the next few generations and show that the percentage of unused transistors will be expanding as manufacturing technology keeps shrinking. Turakhia et al. [36] propose an iterative optimization based approach to investigate the optimal number of cores of each type with given area and power budget for heterogeneous CMPs, where cores with different architectures are made of identical devices. Hardavellas et al. [19] pay specific attention to the server processors and perform an exploration of throughput-oriented processors. Systems built with near-threshold voltage processors (NTV) [14] are also effective approaches.

As for the hybrid device study, Saripalli et al. [31] discuss the feasibility of technology-heterogeneous cores and demonstrate the design of mix-device memory. Wu et al. [38] presents the advantage of hybrid-device cache. Kultursay [24] and Swaminathan [33] respectively introduce a few runtime schemes to improve performance and energy efficiency on CMOS-TFET hybrid CMPs. Our work deviates from the aforementioned in that we conduct a more comprehensive study in the early stage of processor manufacturing. We propose to utilize architectural and device heterogeneity simultaneously to optimally utilize the on-chip resources and balance the performance, energy consumption and total cost. Additionally, in comparison to our previous work [42], this study extends the investigation to more important design factors and aims at drawing more comprehensive conclusions.

3. Methodology

3.1. Metrics

In this section, we describe metrics for the evaluation of different configurations. Note that we characterize multiple aspects including performance, energy efficiency, thermal features and cost-efficiency for each design configuration in order to make a comprehensive investigation.

We choose the total execution time for performance evaluation. For energy-efficiency and thermal features, we use energy-delay product (ED) and peak temperature for assessment. Besides these three extensively discussed metrics, we also include cost-efficiency as the fourth factor for investigation. In this work, we mainly concentrate on the operating cost which is essentially determined by the temperature during execution. The cost efficiency is defined as MIPS/dollar, a widely used metric in computer engineering studies that quantifies the efficiency in delivering performance at a specific cost [6,37,38]. The cooling cost is computed based on a model introduced in a prior work [41]:

$$C_{\text{cooling}} = K_c t + c \tag{1}$$

Note that both K_c and c are cooling cost parameters. K_c is a coefficient associated with the temperature and c is a fitted parameter dependent on the temperature range as well. In general, this cost is determined by the peak temperature achieved during execution. Note that K_c is a variable which is highly related to the steady temperature. High temperature t corresponds to a larger coefficient K_c and results in higher cooling cost consequently. Characterizing the cost-efficiency is necessary for computer architects to identify the optimal design configurations, thus deserving careful consideration.

3.2. Simulation environment and workloads

We use a modified SESC [29], a widely used cycle-accurate simulator for architectural study, to conduct our investigation. We choose McPat 1.0 [26] for power and area estimation and Hotspot 5.0 [32] for temperature calculation. Note that we assume the technology is 22 nm in this work, thus we set the system budget based on an Intel Ivy Bridge processor [3]. The area of the target chip should not exceed 100 mm^2 and the maximal power consumption is 60 W.

Recall that our design space includes configurations which integrate both big and small cores on the same chip. For this purpose, we assume a complex out-of-order core and a simple in-order core whose parameters are summarized from recent commercial processors [3,4,20] and are listed in Table 1. Given these conditions,

Table 1
Architectural parameters for system components.

Component	Parameter	Value
Big core	Pipeline type	Out-of-order
	Processor width	4
	ALU/FPU	4/4
	ROB/RF	160/160
	L1I cache size	32 KB
	L1D cache size	32 KB
	L1 associativity	4
	Area	7.6 mm^2
	Peak power	5.6 W (High-K at 3.0 G) 4.8 W (NEMS-CMOS)
Small core	Pipeline type	In-order
	Processor width	1
	ALU/FPU	1/1
	L1I cache size	8 KB
	L1D cache size	8 KB
	L1 associativity	2
	Area	1.97 mm^2
	Peak power	1.1 W (High-K at 3.0 G) 0.8 W (NEMS-CMOS)
Other parameters	L2 cache size	4 MB
	L2 associativity	8
	Cache block size	32 B
	L2 area	3 mm^2/MB
	L2 power	0.8 W/MB
	Interconnect area	4mm^2
	Interconnect power	5 W
	Other SOC components area	23 mm^2
	Other SOC components power	11 W
	Technology	22 nm
	Voltage/frequency (High-K)	1.1 V/3.0 GHz 0.95 V/2.5 GHz
	Total chip area	100mm^2
	TDP	60 W
	Technology	22 nm

the number of cores that can be accommodated is determined by the following expressions:

$$\text{Area constraint}: N_b \times A_b + N_s \times A_s + A_{\text{all other}} \leq 100 \quad (2)$$

$$\text{Power constraint}: N_b \times P_b + N_s \times P_s + P_{\text{all other}} \leq 60 \quad (3)$$

where variables N_b and Ns denote the number of big cores and number of small cores respectively. Constants A_b and P_b indicate the area and peak power for a big core as listed in Table 1. Similar interpretations apply to other symbols such as A_s and P_s .

Note that conducting a comprehensive exploration of such complex heterogeneous chip multiprocessor systems will inevitably introduce several non-trivial issues that deserves careful investigation. First, calculating the cost-efficiency necessitates temperature estimation which is highly dependent on the chip layout. For each architectural configuration in this paper, we evaluate four types of layouts which pair or scatter the core/L2 in different fashion as described in [27], and choose the one leading to the lowest average temperature as the final target design.

The second issue is to choose an appropriate set of applications for the evaluation. The workloads used in the study are based on the specific architecture in study. When both big and small cores are integrated, we consider "heterogeneous" workloads to be more appropriate for the investigation and thus use combinations of programs from SPEC CPU 2006 for the evaluations; on the other hand, for architectural configurations that are identical across all cores (in the study of device heterogeneity), multi-threaded programs are also used for the assessment. For parallel applications, the number of threads for execution always equals to the core count of the underlying CMP and all programs are executed until completion in order to guarantee that identical tasks are performed. We choose a total of 10 programs from SPLASH-2, PARSEC [7] and ALPBench for the simulation. The reason for not including other workloads is that their intrinsic characteristics (e.g., requiring 2^n threads) prohibit the execution on many configurations. As for the SPEC mixes, each of them includes 30 individual programs (the maximum core count in all evaluated configurations). We simulate 100 million instructions after fast-forwarding the initial 1.5 billion for each individual program within a mix. This also ensures that identical tasks are performed across different configurations. Note that when the core count is less than 30, part of programs will be launched after some cores finish their tasks assigned earlier. A subtle issue that deserves more description is the thread-to-core mapping when a multi-program workload run on a heterogeneous chip multi-processor. We adopt a recently proposed heterogeneity-aware scheduler [12] and apply it to this study for all multi-program executions. By doing so, we expect the reported performance and efficiencies associated with each design configuration represent its full potential. Therefore, the conclusions drawn from the observation would be more convincing. Also, considering that program features such as memory intensity determine the computation efficiency on heterogeneous CMPs, we briefly classify the programs from SPEC CPU 2006 into two categories, namely computation-intensive and memory-intensive, based on their L2 miss ratios. Table 2 lists all selected benchmarks used in this study.

4. Device Heterogeneity

4.1. New devices and architectural implication

The slight improvement in transistor power density is caused by the physical characteristics of metal–oxide–semiconductor field-effect transistors (MOSFET). Due to this limitation, it is intuitive to recognize that breakthroughs in semiconductor technology are the solution to the power shortage problem. In this work, we consider two representative devices, namely High-K dielectric [1,2] and hybrid Nano-electro-mechanical-switch-CMOS (NEMS-CMOS) [9–11] for the investigation.

4.1.1. High-K dielectric

High-K dielectric refers to a device that replaces the silicon dioxide in semiconductor manufacturing. It is capable of greatly suppressing the gate leakage compared to conventional devices. This makes High-K dielectric a promising material for future processor's manufacturing given that gate leakage is observed to be an increasingly important leakage mechanism with the continuous MOSFET downscaling [8,15,39]. As introduced by many leading semiconductor manufacturers, High-K is likely to be the de-facto choice for deep submicron fabrication [1,2].

4.1.2. NEMS-CMOS

The NEMS material, on the other hand, is built as a physical switch and thus not limited by the drawbacks of MOSFET. Fig. 1

Table 2
Selected applications for simulation.

Category	Benchmark suite	Applications (Kernels)
Homogeneous	SPLASH-2	Barnes, FMM, Radix, Raytrace, Water-spatial, waterNS
	PARSEC	Blackscholes, Swaptions
	ALPBench	MPGDec, MPGEnc
Heterogeneous	Computation-intensive	h264, dealII, namd, spcrand, sjeng, omnetpp, gobmk, hmmer, bzip2
	Memory-intensive	mcf, libquantum, milc, leslie3d, perlbench, lbm, soplex, astar

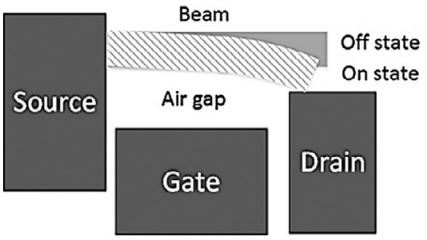


Fig. 1. Architecture of a typical NEMS device.

visualizes the architecture of a typical NEM device. As illustrated in the figure, the device consists of a movable beam which is connected to the source, gate and drain electrode. The movement of the beam is determined by the balance between: (1) the electrostatic force that pulls the beam toward the gate, and (2) the elastic force of the bended beam. When the voltage between the gate and the source reaches a threshold value (i.e., the pull-in voltage), the electrostatic force begins to exceed the elastic force. Consequently, the beam contacts the drain, forming a conductive channel between the source and drain. On the other hand, the beam is physically separated from the drain when the device is in the "off" state. This endows the material with a promising near-zero-leakage feature. However, the NEMS material demonstrates a significantly longer switch delay compared to conventional devices [21]. To benefit from the zero-leakage feature manifested by NEMS, researchers propose to combine NEMS and CMOS together and make a more attractive device. Recent studies have demonstrated the effectiveness of this hybrid device in reducing energy in different scenarios [9,10,13,12,17,35,40]. In this work, we adopt the NEMS-CMOS design proposed in [13] and consider it as the second material for processor manufacturing. Note that the power and performance features of this hybrid NEMS-CMOS device given in [13] are derived at 90 nm technology. Though no 22 nm NEM switch has been fabricated, it is the belief of the community that 22 nm NEMS devices will be realized in the near future [5,9,16,40]. We extrapolate from projections of the hybrid device to assert that the overall performance/power metrics of the material are still achievable at a 22 nm scale.

Another issue that needs to be considered is the operating frequency of NEMS device because it may be constrained by the mechanical component. However, recent studies have presented the design and applications of NEMS device in GHz range [22,34], which eliminates the concern on the operating frequency and confirms that NEMS device can be used in future high-performance processors.

To project the performance of a 22 nm NEM transistor, we use scaling techniques based on [34] by adjusting the component parameters such as the beam and air gap dimensions to control the on-state current (I_{on}) and off-state leakage current (I_{off}). The I_{on} and I_{off} can be projected following the approach in [10] which predicts the performance of NEM transistors at 65 nm technology by scaling the gap and beam thickness. We adopt a similar strategy to conduct the projection at 22 nm scale using the values reported in [10,40], both suggesting that the leakage current of NEMs devices has decreased significantly since the 90 nm simulations.

As for the power features, we use SPICE to prove that the hybrid NEMS/CMOS device proposed in [13] can offer the same power improvements over purely CMOS design as long as the ratio of NEMS to CMOS leakage current is the same or less than was used in [13]. The schematic of the dynamic OR gate used for the simulation is shown in Fig. 2 . Note that this assumption on the ratio is fairly reasonable considering the significantly decreased leakage power on NEM transistors manufactured with technology newer than 90 nm [10,40].

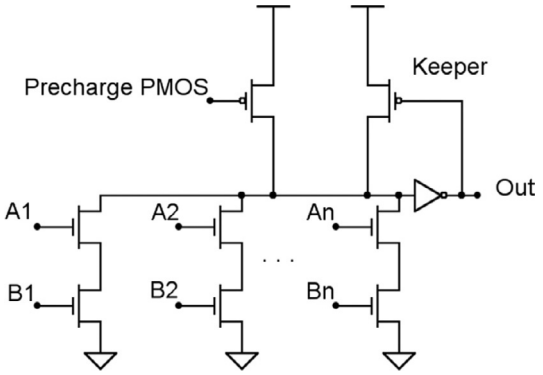


Fig. 2. Dynamic OR gate used for SPICE simulation.

Table 3
Features of materials considered in this work.

Material	Features
High-K	Reduce leakage power to 20% of the dynamic power
NEMS-CMOS	OR gate: 20% higher delay, reducing 60% switching power
	SRAM cell: 25% higher delay, saving 85% leakage energy

Based on these analyses, we can safely conclude that the performance/power feature of the hybrid NEMS/CMOS device proposed in [13] is still achievable at 22 nm technology node. Table 3 lists the important features of two materials considered in this study [1,13,21]. Note that the percentage of savings for both materials are with respect to standard CMOS process. As can be noted from the table, NEMS-CMOS and High-K materials deliver distinct tradeoff between performance and power, implying that an appropriate combination of High-K cores and NEMS-CMOS cores on the same chip would produce a processor that works more efficiently than a CMP using one device exclusively. Furthermore, it is important to note that our investigations in this work can be generalized to scenarios where different devices are used. For instance, Tunnel-FET (TFET) cannot match the performance of CMOS under normal voltage, but it is beneficial for power saving [31], thus introducing similar trade-offs between performance and power.

4.2. Results analysis

4.2.1. Performance and energy efficiency

We consider two categories of CMPs to characterize the impact of device selection. The first group of chip-multiprocessors is composed of big out-of-order cores while the ratio of High-K cores to NEMS-CMOS cores is varying. Based on the power and area constraints depicted in Section 3.2, the total number of big cores that can be accommodated on die is either 7 or 8. When all cores are manufactured with High-K, the power constraint restricts the maximal number of cores to be 7 although there is enough space for an extra core; as more NEMS-CMOS cores are integrated to replace High-K cores, the area constraint becomes the determinative factor and confines the core count to be 8. In contrast, when all cores are small in-order ones, the core count is always limited by the area constraint and should not exceed 30.

We run both multi-threaded and multi-program workloads with these configurations for evaluation. Fig. 3(a) plots the average performance, energy, and energy-efficiency of multi-threaded and computation-intensive multi-program workloads. The notation *x*H_*y*N means a total of *x* High-K cores and *y* NEMS-CMOS cores are installed. Also recall that the performance is measured in execution time, thus smaller values indicate better performance. As can be observed, in the "big" category, the execution time

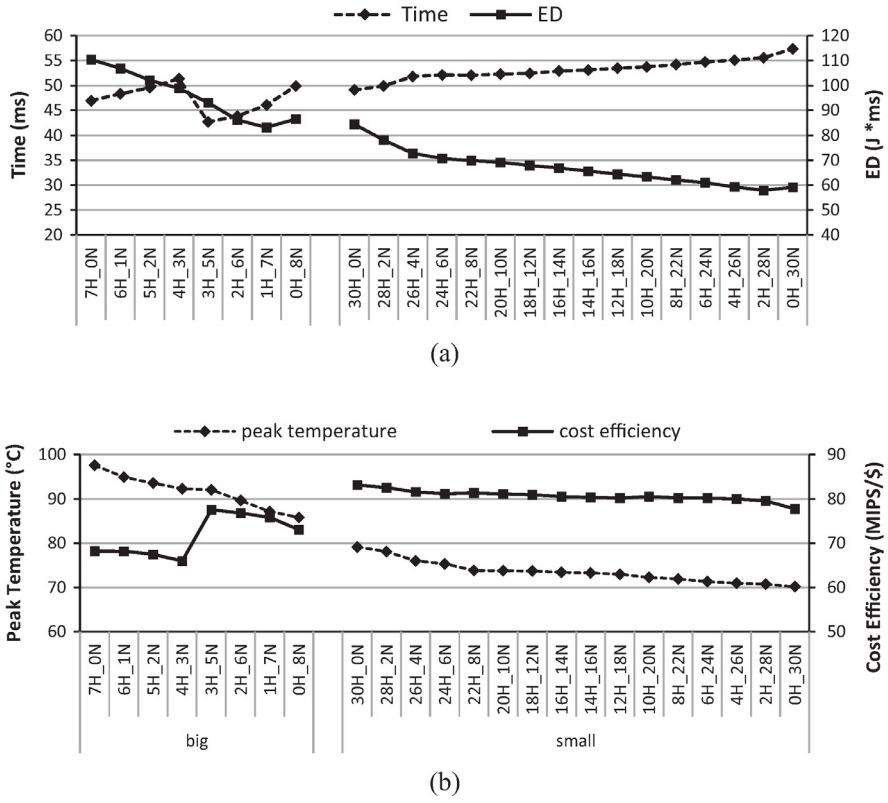


Fig. 3. Execution results of multi-threaded applications and computation-intensive "heterogeneous" workloads running on mix-device CMPs: (a) average execution time and ED (b) average peak temperature and cost efficiency.

gradually increases at first and demonstrates a significant reduction from 4H_3N to 3H_5N, after which the curve rises again. The reason for the performance degradation (e.g., from 7H_0N to 4H_3N, and the segment between 3H_5N and 0H_8N) is that NEMS-CMOS cores execute at a lower rate than their High-K counterparts; therefore, increasing the number of NEMS-CMOS cores tends to increase the overall execution time. The performance improvement at 3H_5N comes from the extra core in this configuration, with which the applications are executed with one more thread. As for the "small" category, the execution time gradually increases as more NEMS-CMOS cores are included since the core count is fixed to 30 irrespective of the manufacturing device.

The energy consumption demonstrates a different variation from the performance change. In the big core category, the total energy dissipation is generally decreased as more power-saving NEMS-CMOS cores are integrated. This is because the average per-core power consumption is reduced and it outweighs the performance degradation introduced by the slower cores. In addition, similar to the trend of performance variation, there is a reverse change from the 4H_3N to 3H_5N configuration. The reason of the energy increment at this point is that the total power increase outweighs the performance benefit, thus leading to slight more energy consumption. The trend in the small core category is relatively more stable. Generally speaking, including more small cores is helpful to save the total energy despite the execution time gradually increases.

The energy-efficiency variation is similar to the change of energy consumption. In general, the energy-delay product is decreasing as more NEMS-CMOS cores are added. This is because that the energy saving from the NEMS-CMOS cores outweighs the corresponding performance degradation while running these applications, thus using more such cores is beneficial to improving the energy-efficiency. The only exception is observed at the switch from 1H_7N to 0H_8N in the "big" category (or 2H_28N to 0H_30N in "small"), where the energy-delay demonstrates a slight increase. This is due to the fact that the performance degradation contributes more to the variation of ED for programs with long serial phase. This is particularly noticeable for multi-threaded applications. With the 0H_8N configuration, the sequential stages are executed on the NEMS-CMOS cores, thus resulting in significant performance loss and higher ED. We also examine the execution of memory-intensive multi-program workloads. The evaluation results generally corroborate the effectiveness of device heterogeneity in delivering better performance-energy tradeoff.

In summary, for a CMP which only consists of big cores, including relatively more NEMS-CMOS cores and a few faster High-K cores is preferable to building a chip with processor cores made entirely using a single device. For the small-core-oriented architecture, the highest energy-efficiency is delivered by the configuration 2H_28N, meaning the optimal balance between performance and energy consumption is also achieved on a CMP with a large number of NEMS-CMOS cores and a few High-K cores.

4.2.2. Thermal feature and cost efficiency

Peak temperature and cost-efficiency are another two important metrics to evaluate a design configuration. We demonstrate the results of these two features for the proposed configurations in Fig. 3(b). As shown in the figure, the temperature drops significantly as we employ more power-saving NEMS-CMOS big cores. Therefore, the coolest chip is the one where all cores are manufactured with NEMS-CMOS. With respect to cost-efficiency, lower temperature results in a lower cooling cost. This means that we are essentially trading off "performance" for "low cost" when we replace a NEMS-CMOS core for a High-K core. In this scenario, the cost-efficiency reaches the peak value at 1H_7N where the performance and cost can be optimally balanced. Note that the increment of cost-efficiency from 4H_3N to 3H_5N is resulted from the performance boost. The curve corresponding to the "small"

category is smoother. This is because the in-order cores consume much less power than the big cores and thus generate less heat. This results in relatively mild temperature variation across configurations. The cost-efficiency does not have a large variance when we change manufacturing devices. Nevertheless, it is still reasonable to conclude that hybrid-device CMPs outperform chips built with a single device alone.

4.2.3. Case study

To further understand the performance scaling trend shown in Fig. 3, we choose a representative application (MPGEnc) from the program set for analysis and demonstrate the results in Fig. 4. Note that we only show the results of using CMPs with big cores. The MPGEnc benchmark implements a parallel version of MPEG-2 encoder. In this application, the threads are forked and joined at the beginning and end of each frame. Each thread is responsible for encoding a set of macroblocks of a frame while thread 0 always operates on its dedicated buffer. The tasks assigned to each thread are not identical, thus the time spent by each thread will vary. Plot (a) demonstrates the performance and ED scaling while Plot (b) shows the active cycles of each core during the execution of this program with four configurations. The total execution time is determined by the main thread running on the first processor (P0), and the performance of the parallel stage can be estimated from the active cycles of P1. Since the number of threads is increased from 7 to 8, the 3H_5N configuration takes much less time than 4H_3N to finish the encoding due to acceleration in parallel stage, hence the remarkable performance improvement at 3H_5N. For the latter three configurations where the core counts are identical, the performance degradation is caused by decreasing the number of faster cores (High-K). For example, the 1H_7N organization includes only one High-K core (P0) while three such cores are equipped in 3H_5N; as a consequence, the parallel stage needs more time to complete on the CMP configured as 1H_7N, thus lowering the overall performance. On the other hand, the performance degradation from 1H_7N to 0H_8N essentially stems from the slow execution of the sequential stage. This is especially critical for programs with long initialization and finalization.

5. Two-fold heterogeneity

5.1. Performance, energy, and energy efficiency

Prior studies have demonstrated the advantages of architecturally asymmetrical chip multi-processors for energy-efficiency improvement. In light of these advantages, it is natural for us to consider a design pattern in which both device-heterogeneity and architectural asymmetry are jointly adopted, hence the name "two-fold heterogeneity". In this section, we consider a set of configurations where both the material and complexities are different among integrated cores. We assess two kinds of organizations: one with big High-K cores and small NEMS-CMOS cores and vice versa.

Fig. 5(a) plots the performance scaling of computation-intensive programs with these two design patterns. The upper labels on the horizontal axis correspond to the first architecture in which big cores are made of High-K and small cores are manufactured with NEMS-CMOS (mix0 or *x*HB_*y*NS); accordingly, the lower labels correspond to the opposite architecture which includes big NEMS-CMOS and small High-K processors (mix1 or *x*NB_*y*HS). As can be observed, configurations from the second category, namely *x*NB_*y*HS, always outperform their counterparts from the first category. This can be explained by two aspects. First, since NEMS-CMOS cores are relatively power-saving, the second design pattern accommodates more processors when the core count is power-limited. For this reason, the total number of cores is larger in the *x*NB_*y*HS designs, thus these configurations take shorter time to finish executing the program combination. This corresponds to the scenarios where the number of big cores is no smaller than 6. Second, as the constraining factor shifts from power to chip area, the core counts in both design patterns become identical (from 5B_11S). In this situation, the global execution time basically depends on the performance of small cores as they are in the majority. For instance, in the 2B_23S configuration, how fast the programs run on small cores determines the overall performance in essence, because the number of small cores is remarkably larger than that of big cores. Since those in-order processors are made of High-K, the chips designed with the second pattern still offer better performance.

The energy consumption for the two sets of design options are shown in Fig. 5(b). In general, the variations of the two curves are not monotonic. Processors designed with the mix1 pattern consume less energy than the counterparts in mix0 when the number of big cores is between 3 and 7, while mix0 design options are more energy-saving when the number of big cores is no more than 2. Among all the evaluated options, the 4NB_15HS in the mix1 category turns out to be the optimal configuration from the energy consumption perspective. Fig. 5(c) demonstrates the variation of the energy-efficiency for the same program set running with considered configurations. Note that the interplay between the performance/energy of different cores makes the ED variation non-monotonic. For both blending patterns, we note that the energy-delay product gradually decreases at first until the minimal value is reached at 4NB_15HS, after which the efficiency is decreasing. More specifically, the *x*NB_*y*HS delivers better energy-efficiency than the *x*HB_*y*NS when the configuration is varied from 8 big cores to 3 big cores. This is due to the shorter execution time and lesser energy consumption on big NEMS-CMOS cores. As small cores begin dominating the chip in 2B_23S and beyond, their relatively large energy consumptions mitigates the performance benefits and make the ED rise again.

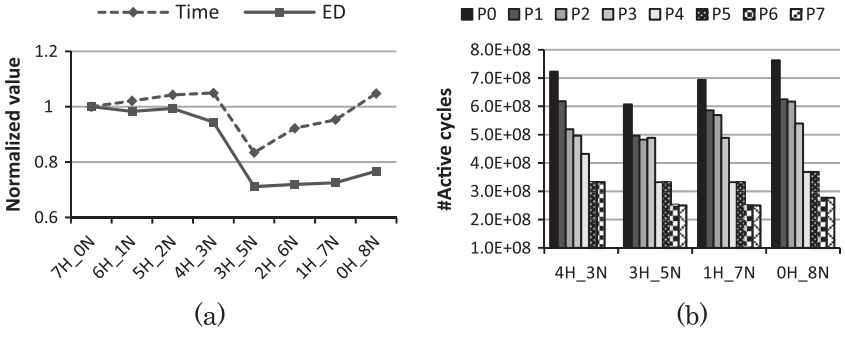
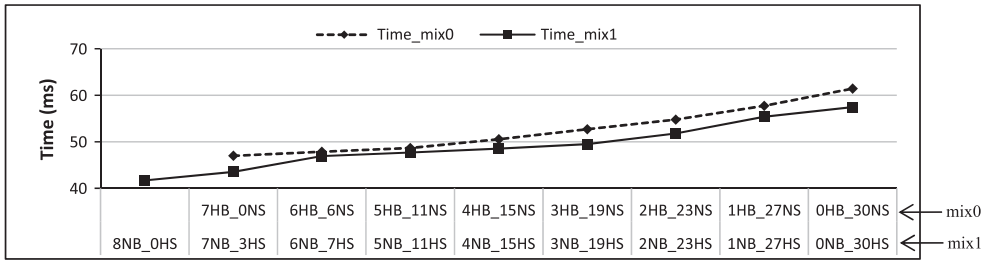
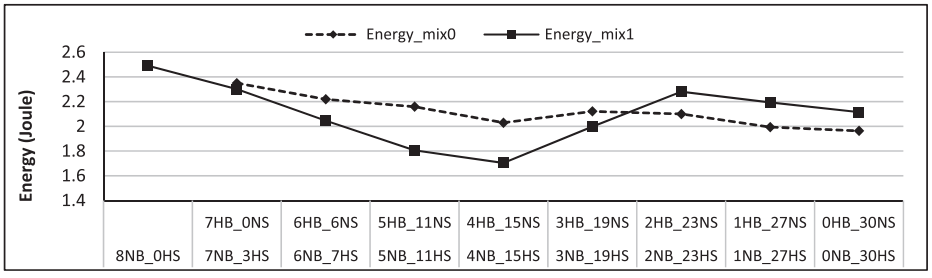


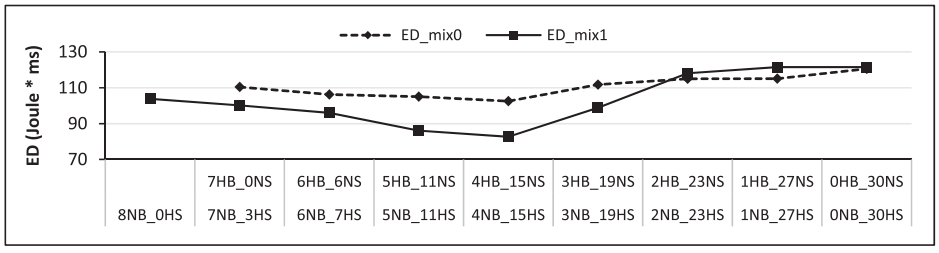
Fig. 4. Execution information of *MPGEnc*: (a) execution time and ED (b) per-core active cycles while running with selected configurations.



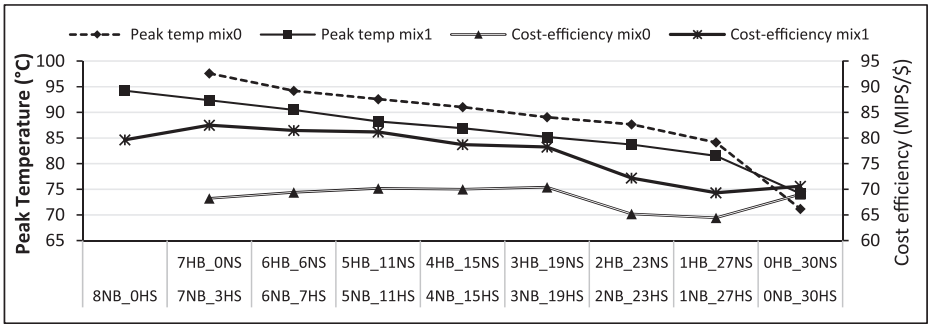
(a)



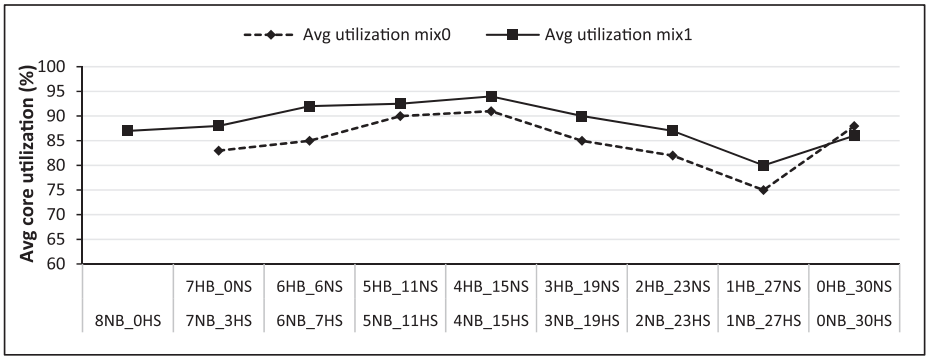
(b)



(c)



(d)



(e)

Fig. 5. Execution results of computation-intensive workloads running on mix-device heterogeneous CMPs: (a) execution time, (b) energy consumption, (c) energy-delay product, (c) peak temperature and cost-efficiency and (e) average core utilization.

5.2. Thermal feature, cost efficiency, and utilization degree

Fig. 5(d) plots the peak temperature and cost-efficiency of these two-fold heterogeneous CMPs while running computation-intensive workloads. As we have observed previously, NEMS-CMOS cores result in lower temperature than High-K cores and small cores are much cooler than big ones. Consequently, the second design pattern (i.e., *x*NB_*y*HS) tends to be cooler than its alternative (*x*HB_*y*NS), because the hotspot on die which is usually located in the out-of-order processor has lower temperature. Recall that the *x*NB_*y*HS also delivers better performance. Therefore, its cost-efficiency is significantly higher than that offered by *x*HB_*y*NS configurations. As can be seen, for computation-intensive workloads, the cost-efficiency reaches the peak value at 7NB_3HS configuration, which improves the efficiency by 20.9% compared to the 7HB_0NS case. For memory-intensive workloads, (graphs not shown due to space limitation), the optimal configuration outperforms the baseline case by up to 66.7%. In conclusion, our observations made in this section demonstrate that the mix1 design paradigm (*x*NB_*y*HS, or big NEMS-CMOS cores along with small High-K cores) stands as the optimum among all evaluated configurations, since it can more efficiently balance the execution performance, energy consumption, and total cost.

For a chip-multiprocessor that integrates a large number of cores, the core utilization degree is another important metric to evaluate the execution behavior. Inefficient usage of the computation resources may lead to longer execution time and lower energy-/cost-efficiency. As we mentioned in Section 3.2, we employ a recent proposed heterogeneity-aware scheduler on the target platform to utilize all cores in an effective and balanced fashion. Fig. 5(e) plots the average core utilization of all design points while running computation-intensive workloads. As can be observed, all design points from the mix0 and mix1 category reach a utilization degree above 75% while the peak utilization is greater than 90%, implying that no cores stay idle for long time. However, configurations falling to the mix1 category generally demonstrates slightly higher utilization than those from mix0. This is because the performance difference between the architectural big and small cores in mix1 is relatively smaller (i.e., big cores are built with slower NEMS-CMOS device). Another interesting observation is that on CMPs where small cores dominate (e.g., 2HB_23NS, 1NB_27HS), the average utilization is obviously lower. This is because the computation-intensive programs are assigned to run on the big cores more frequently, leading to relatively lower utilization on small cores. Nevertheless, the utilization result demonstrates the effectiveness of the proposed heterogeneous design pattern.

We have shown that mixed-device heterogeneous CMPs are beneficial for improving the energy- and cost-efficiency for computation-intensive workloads. Now let us shift our concentration to memory-intensive workloads in order to further justify the conclusion that the design paradigm mix1 is globally optimal. Fig. 6(a) shows the performance comparison between mix0 and mix1. Generally, we observe a similar trend that the mix1 design paradigm is more preferable than mix0 by delivering better performance. However, compared with the scaling behavior shown in Fig. 5(a), Fig. 6(a) demonstrates that memory-intensive workloads favor more small cores, hence favoring a larger total number of cores, for shorter execution time. The reason is that running memory-bound programs on big cores will not significantly accelerate the execution as opposed to computation-intensive workloads. Therefore, executing more programs concurrently can effectively reduce the time for completing all tasks compared to running them sequentially on a few big cores. To more clearly demonstrate the difference across all configurations and illustrate the benefit of two-fold heterogeneity, we choose the most energy-efficient configurations from five design patterns, namely High-K for all cores, mix0, mix1, and NEMS-CMOS for all cores, and make comparison among these material-dependent optima. As can be seen from Fig. 6(b), the most energy-efficient configuration in the mix1 category outperforms the optimal High-K CMP by 17% in energy-efficiency with a less than 4% performance loss. We also make the comparison for computation-intensive programs and draw a similar conclusion that mix1 demonstrates remarkable benefits over other design patterns in terms of energy-efficiency.

Fig. 7 plots the thermal and cost-efficiency results for memory-intensive workloads running on mixed-device heterogeneous CMPs. Not surprisingly, the mix1 design paradigm results in a cooler chip than mix0 in most cases, thus delivering up to 66.7% higher cost-efficiency compared to the baseline configuration. Our conclusion is that building big out-of-order cores with NEMS-CMOS and manufacturing small in-order cores with High-K is able to achieve the optimal balance between performance, energy consumption, and total cost also holds for the memory-intensive applications.

5.3. Case study

The average results demonstrated in prior subsections show the general trend of performance/efficiency variation with different design configurations. However, it is also worthwhile to take a closer look at the scaling at a finer granularity to further understand the underlying rationale. In this subsection, we will focus on two workloads and use their execution results to exemplify the correlation between workload characteristics and configuration selection.

In order to make a more thorough investigation, we concentrate on memory-intensive multi-program workloads for analysis in this subsection given that a case study based on a representative computation-intensive multi-threaded application is shown in Section 4.2.3. While the two workloads used for analysis are both categorized as memory-intensive, they differ in the off-chip memory access intensity, implying that their sensitivities to the core count/type variation are different. For simplicity, we use MWL1 and MWL2 to denote the two workloads respectively. MWL1 is composed of the applications with high L2 miss rates, and each of them issues notably higher off-chip memory accesses than the applications included in MWL2.

Fig. 8 shows the performance, energy-efficiency and cost-efficiency of these two workloads when they are executed on configurations of the mix1 design pattern. As can be seen, the consensus of the scaling trends of MWL1 and MWL2 is that they both prefer executing on processors with relatively more cores instead of a few powerful big cores. This is similar to the observation made in Fig. 6 which demonstrates the average results. The reason is given in Section 5.2. However, MWL2 performs better with the configurations to the right end of the curve, i.e., CMPs with more cores, while the optimal configuration for MWL1 has relatively fewer cores. This is not hard to understand considering the workload characteristics described above. Since each individual program in MWL1 generates substantial off-chip memory requests, the shared bus is likely to get congested while there are more active cores (i.e., more simultaneously running programs). In this situation, appropriately reducing the number of cores is beneficial alleviating the bus congestion and leads to a better balance between concurrency and shared resource utilization. On the contrary, MWL2 favors more cores because the benefit from higher parallelism outweighs the degradation due to bus contention. Nevertheless, this does not change our conclusion drawn from Sections 5.1 and 5.2 that two-fold heterogeneity can more effectively utilize the power and area resources.

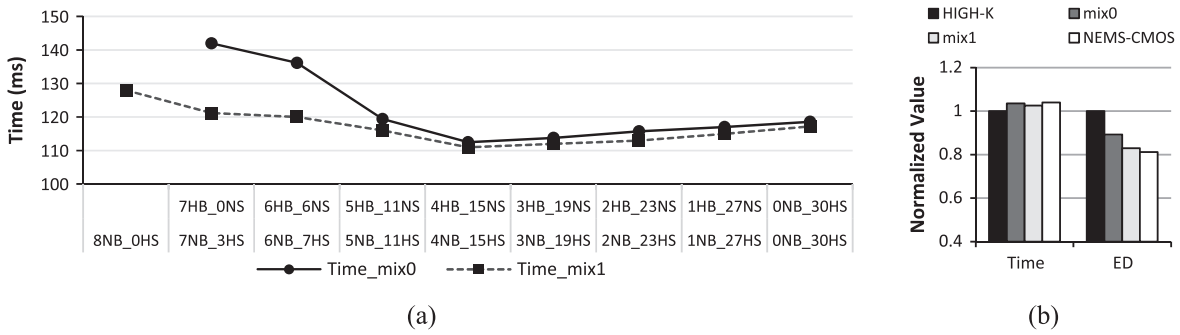


Fig. 6. Execution information for memory-intensive workloads running on mix-device heterogeneous CMPs: (a) execution time and (b) comparison among material-dependent optimal configurations.

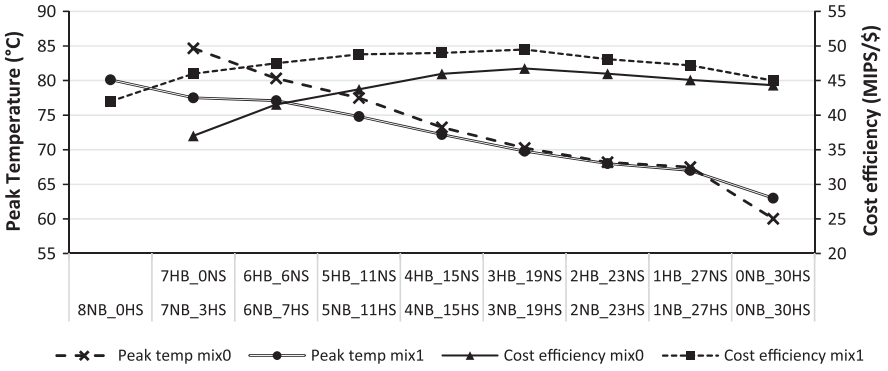


Fig. 7. Peak temperature and cost-efficiency of memory-intensive workloads running on mix-device heterogeneous CMPs.

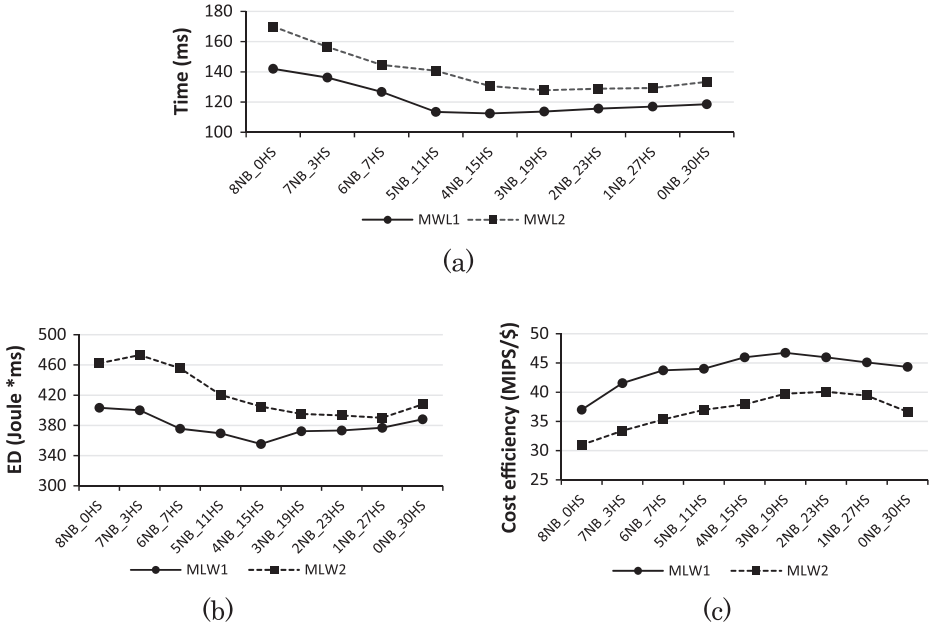


Fig. 8. Execution results of two memory-intensive workloads running on mix1 mix-device heterogeneous CMPs: (a) execution time, (b) energy-delay product and (c) cost-efficiency.

6. Exploiting heterogeneity under varying voltage/frequency

The analysis made in previous sections all assumes a single voltage/frequency on both types of cores. However, considering the strong correlation between the operating point (i.e., v/f pair) and the performance, power and temperature on the target processor, it is essential for us to further evaluate the design patterns by taking the operating voltage/frequency into account for a thorough investigation. To highlight the necessity of such assessments, let us recall the features of High-K and NEMS-CMOS for a brief comparison. As can be derived from Table 3, compared to a High-K core, a NEMS-CMOS core operated at the same supply voltage might provide impressive power saving with less than 30% performance degradation when executing certain workloads. Also, the relatively

large power consumption of High-K big cores results in a CMP with fewer cores integrated in many configurations (e.g., 7HB_0NS in mix0). These could result in a biased preference on the NEMS-CMOS device. Therefore in this section, we assess another *v*/*f* level for High-K cores in order to conduct a fair comparison. Note that the study presented in this section is performed under two conditions. First, we do not change the voltage/frequency setting for NEMS-CMOS cores because that the performance of NEMS material tends to be limited by its mechanical structure, which is out of the scope of this paper. Second, for the purpose of clarity, we only change the operating points on big cores since their power consumption usually contributes more to the total power.

We conduct the new assessment on two-fold heterogeneous CMPs. To ensure that the entire chip area can be effectively utilized without violating the power constraint, we lower the operating voltage/frequency of High-K big cores to 0.95 V/2.5 G [30], at which the total power consumption just meets the power budget while no area is left inactive. We use mix2 to denote this set of configurations.

Fig. 9(a) plots the variation of energy-efficiency while computation-intensive workloads are running with different configurations. Note that it is more reasonable to use the energy-delay squared product (ED^2) for energy efficiency evaluation when voltage/frequency scaling is taken into consideration [23], therefore we demonstrate the plot of ED^2 for all the evaluated configurations. As can be observed, the mix1 design pattern still outperforms all other configurations by delivering the lowest ED^2 at 6NB_7HS. This corroborates our conclusion drawn in the previous section that building big cores with NEMS-CMOS while manufacturing small cores with High-K is a promising design pattern. On the other hand, appropriately setting the operating voltage/frequency on a CMP following the HB_NS paradigm is effective to increase its energy-efficiency. Specific to the configurations in mix2, we observe that the ED^2 values are significantly smaller than those corresponding to the mix0 CMPs. This is attributed to the considerable reduction in dynamic power of big cores. Moreover, when the number of big cores is greater than 5, the mix2 design pattern can accommodate more cores than mix0, thus shortening the total execution time. As the big core count keeps decreasing, the CMP is gradually dominated by small cores. Therefore, the overall energy-efficiencies achieved on mix2 CMPs converge to those of mix0 processors (e.g., 2HB_23NS to 0HB_30NS).

We also make comparison among the material-dependent optimal configurations of the five design patterns, namely High-K for all cores, mix0, mix1, mix2, and NEMS-CMOS for all cores. We normalize the execution time and ED^2 to those corresponding to the optimal High-K processor and demonstrate the result in Fig. 9(b). As can be observed, the CMP 6NB_7HS with mix2 design pattern obviously outperforms all other design options from the energy-efficiency perspective. As for the HB_NS configurations, those with High-K big cores running at 2.5 GHz deliver better energy-efficiency than those from mix0 when the big core count is no more than 5. This is due to the significant power savings on big cores running at a lower frequency. When the number of NEMS-CMOS small cores becomes overwhelming, the energy-efficiency of mix0 and mix2 design options becomes comparable. However, both mix0 and mix2 configurations trail mix1in terms of energy-efficiency in most cases. We also study the memory-intensive workloads and observe a similar phenomenon. Therefore, we can make the following conclusion based on the investigations: building big cores with a comparatively power-saving material (NEMS-CMOS) and manufacturing small cores with faster High-K device (i.e., mix1 or NB_HS) is the most attractive design paradigm. For the alternative pattern HB_NS, appropriately setting the voltage/frequency of High-K cores according to the workload features is necessary to yield better usage of the on-chip resources.

The conclusion also holds from the perspective of thermal feature and cost-efficiency. To demonstrate this, we identify the most cost-efficient configurations from the five design patterns and plot their respective efficiencies and peak temperatures in Fig. 9(c). We do not show the specific temperature/efficiency-configuration curve for the purpose of clarity. As can be noted from the figure, the optimal configuration from mix1 (with respect to cost-efficiency) remarkably surpasses its competitors by leading to the most

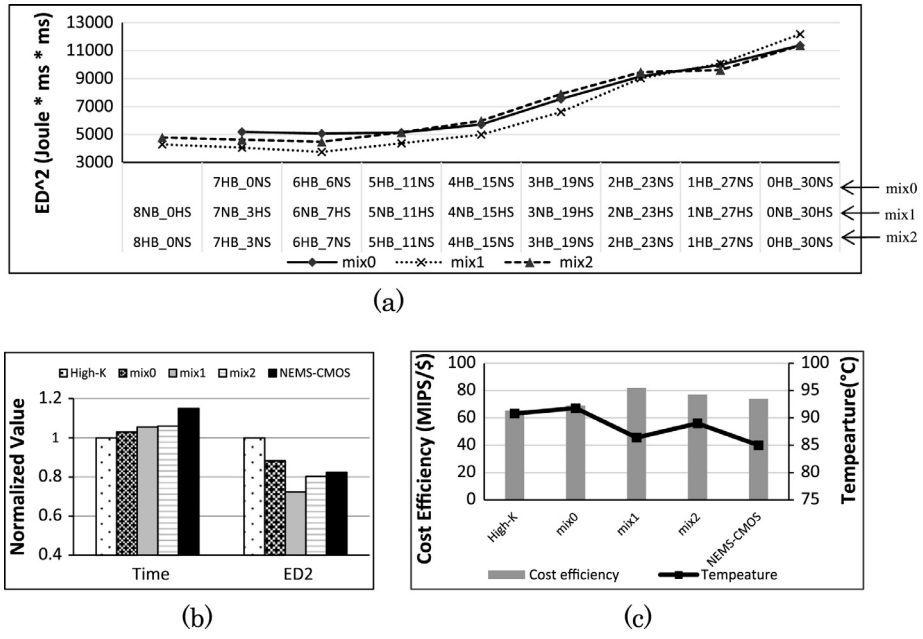


Fig. 9. Execution results of computation-intensive workloads running on mix-device heterogeneous CMPs with varying operating voltage/frequency: (a) average ED^2, (b) performance and ED^2 comparison among material-dependent optimal configurations and (c) peak temperature and cost-efficiency comparison among material-dependent optimal configurations.

desirable balance between performance and cost due to the low temperature on the big cores. For CMPs with HB_NS architecture, decreasing the operating point on High-K big cores is able to cool down the entire chip; however, it still trails the optimal configuration in mix1. In general, the NB_HS (mix1) stands as the most promising paradigm to effectively utilize the on-chip physical resources.

7. Optimal design configuration with computational sprinting

As we introduced in Section 1, the ever-widening gap between the processor power requirement and insufficient supply forces transistors to either operate at a low voltage/frequency or is able to activate only a portion of transistors at a time. Computational sprinting [28] is proposed to alleviate this problem by breaking the TDP limit in a burst fashion. By employing appropriate materials with high thermal capacitance and integrating a well-designed heat spreading network, a computational sprinting enabled system is capable of "illuminating" all transistors temporarily or overclocking the active cores within a short execution period, in order to increase the system responsiveness without causing irrecoverable reliability issues due to exceeding TDP. Obviously, involving this technique will impact the performance and efficiencies of the underlying hardware because the number of active cores and core types can be different from that under regular operating condition. In this section, we apply the computational sprinting technique to our target processor and observe its implication on the design pattern selection.

Given the investigations conducted in previous sections, we will apply computational sprinting on the *x*NB_*y*HS design paradigm since it delivers the best trade-off among the important design goals in general. We adopt the design strategy introduced in [28] to configure our target system with computational sprinting capability. Specifically, we assume that phase-change material is used for heat storage while the heat spreading and power distribution network proposed in [28] are integrated. Note that our simulation platform is modified accordingly to mimic these changes. The system is able to provide 10 W extra power for a 0.5 s bursty execution period. In other words, the maximal power consumption of the system can be up to 70 W for a short time. Considering that all dark area can be utilized with the *x*NB_*y*HS design pattern (i.e., mix1 paradigm), we will use the extra power to overclock the running cores. We feed this value into our power model and derive the sprinting configurations as listed in Table 4. Note that these configurations are categorized into two groups. The first group of settings boost all processor cores in an even fashion such that the frequency of each core is increased by the same amount. With this setting, all cores can be sprinted to 3.12 GHz. In the second group, we prioritize the big cores by further increasing their frequencies while keeping the remaining cores unchanged. By doing so, up to 4 big cores can be overclocked to 3.2 GHz. Meanwhile, all other cores are still running at 3 GHz. Note that on CMPs from 3NB_19HS to 0NB_30HS where integrates relatively few big cores, we alternatively give all power to small cores and up to 12 small cores can be overclocked to 3.2 GHz. In the following paragraph, we refer the first group of configurations as mix3 and the second group as mix4.

Table 4
Configurations with computational sprinting.

Category	Configuration
Even distribution (mix3)	All cores are boosted to 3.12 GHz
Prioritized distribution (mix4)	4 big NEMS-CMOS big cores are overclocked to 3.2 GHz bursty (from 8NB_0HS to 4NB_15HS) 12 High-K small cores are boosted to 3.2 GHz bursty (from 3NB_19HS to 0NB_30HS)

Fig. 10 demonstrates the performance, energy consumption, energy-efficiency, and cost-efficiency while running computation-intensive workloads on all *x*NB_*y*HS processors. This corresponds to all configurations from mix1, mix3 and mix4 categories. We do not include the mix2 category for comparison since mix1 outperforms mix2 by delivering better performance and efficiencies as presented in Section 7. Note that ED^2 is used for energy-efficiency evaluation since different frequencies are involved in the experiments [22]. We first make a comparison between computational-sprinting-enabled processors (i.e., mix3 and mix4) and processors running under regular conditions (mix1). Unsurprisingly, all processors from mix3 and mix4 outperform those from mix1 from the performance perspective. This essentially confirms the effectiveness of computational sprinting which aims to increase the system responsiveness by exceeding the TDP and boosting core frequencies temporarily. On the other hand, the energy consumption and energy-efficiency both show non-monotonic variation in all three groups of configurations. Note that none of these three design paradigms show a consistent advantage in terms of lower energy consumption (or smaller energy-delay squared product) than the other two, because the relative saving in execution time and power consumption is varying across the configurations. For example, in the big core dominant platforms (e.g., 7NB_3HS), the performance boost from computational sprinting outweighs the increase in power consumption, thus mix3/mix4 processors consume less energy and delivers better energy-efficiency than mix1. However, this is not the case in configurations such as 3NB_19HS, since the performance improvement from sprinting is not able to mitigate its larger power consumption. Nevertheless, we should note that the most energy-efficient configurations (i.e., those leading to the smallest energy-delay product) from mix3/mix4 still outperform that from mix1, meaning that appropriate configurations with computational sprinting can improve the energy-efficiency. Moreover, the most energy-efficient configurations in mix3 and mix4 are 7NB_3HS and 8NB_0HS respectively while the optimal one in mix1 is 4NB_15HS. This implies that integrating more big cores is preferable in computational sprinting enabled systems. This is reasonable considering that computation-intensive workloads are used in this evaluation, since those workloads are able to obtain more benefits from big cores. On the other hand, the cost-efficiency variations of mix3 and mix4 show a similar trend as that of mix1. It initially rises gradually to the peak value and then decreases. This is due to a similar reason that the optimal trade-off between the performance and total cost is achieved at that particular point as discussed in Section 5.2.

It is not surprising to see that computational sprinting enabled processors display an obvious advantage over regular processors. However, it is also interesting to compare mix3 and mix4 in order to understand the most efficient way to distribute the extra power among all cores. Therefore, in this paragraph, we concentrate on the comparison of mix3 and mix4 because they represent two typical approaches to distribute power for overclocking. As can be observed from Fig. 10(a), the 8NB_0HS configuration in mix4 (i.e., 4 big cores run at 3.2 GHz in bursty fashion) results in the shortest execution time among all options. This is also because computation-intensive workloads are able to get more benefits from the accelerated big cores. As the configurations shift to small-core dominant patterns such as 3NB_19HS, the mix3 design shows its advantage by taking less time to complete the workload because most cores are running faster than the counterparts in mix4. However, the mix3 design paradigm shows a consistent advantage over mix4 from the energy perspective. More specifically, the 7NB_3HS in mix3 configuration consumes the least

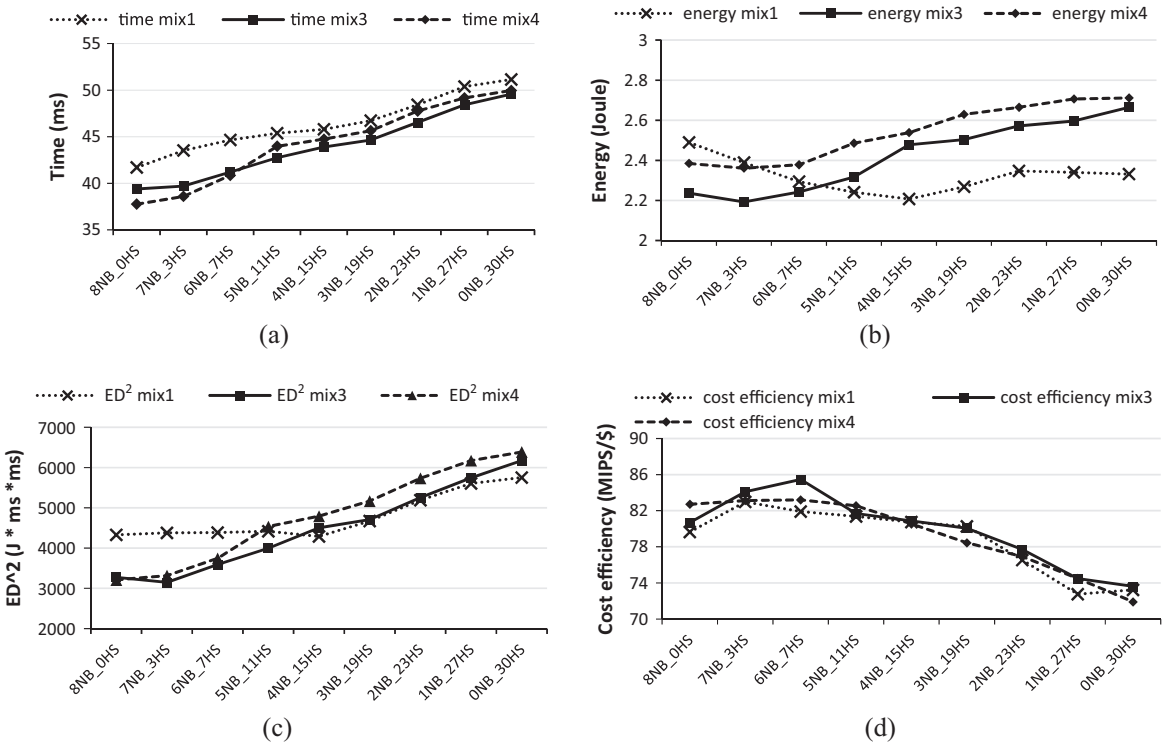


Fig. 10. Execution results of computation-intensive "heterogeneous" workloads running on *x*NB_*y*HS CMP with computation sprinting capability: (a) execution time, (b) energy consumption, (c) ED^2 and (d) cost efficiency.

energy for the workload execution and delivers the optimal energy-efficiency. This is mainly because of its relatively short execution time and lower average power. When the number of small cores is gradually increasing, the total execution time is significantly prolonged and results in the increase of energy consumption and energy-delay product. As for the cost-efficiency, the highest value is delivered at 6NB_7HS in the mix3 category. Note that while the best performance is achieved at 8NB_0HS in mix 4, its high cooling cost due to the four overclocked big cores (3.2 GHz) largely mitigates the performance benefit and results in relatively lower cost-efficiency.

We also run memory-intensive workloads and conduct the same investigation. We observe that the optimal energy- and cost-efficiency are also given by configurations from mix3, while small-core dominant configurations are preferred due to higher thread-level parallelism. Therefore, based on the aforementioned analysis, we can make a conclusion that employing the computational sprinting technique provides noticeable benefits by boosting the performance and execution efficiencies compared to regular situations. In addition, spreading the extra power to all cores in a homogenous fashion is a better option than distributing it among a few powerful cores in order to deliver higher energy- and cost-efficiency.

8. Conclusion

As dark silicon has begun to hazard the scaling of Moore's Law and prohibits us benefiting from the increasing number of transistors, new design technologies are in high demand to address this problem. This is especially important in the early stage of processor manufacturing where issues such as architectural organization and device selections need to be carefully considered. For this purpose, our work evaluates a series of design configurations by exploiting the device heterogeneity and architectural asymmetry in the processor manufacturing. Our evaluation results demonstrate that building heterogeneous chip multiprocessors with different materials is more preferable than conventional designs since it can efficiently utilize chip level resources and deliver the optimal balance among performance, energy consumption and cost.

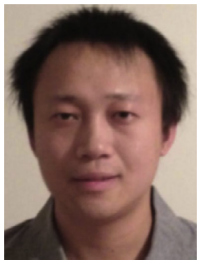
Acknowledgement

This work is supported in part by NSF Grants CCF-1017961 and CCF-1422408. We acknowledge the computing resources provided by the Louisiana Optical Network Initiative (LONI) HPC team. Finally, we appreciate invaluable comments from anonymous reviewers.

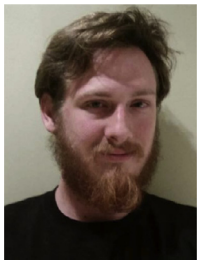
References

[1] Intel Corporation, High-K and Metal Gate Transistor Research, 2012. <http://www.intel.com/pressroom/kits/advancedtech/doodle/ref_HiK-MG/high-k.htm>.
[2] Intel Corporation, High-K + Metal Gate Transistor Breakthrough on 45 nm Microprocessor, 2007. <http://download.intel.com/pressroom/kits/45nm/Press45nm107_FINAL.pdf>.
[3] Intel Corporation, Intel CoreTM i7-3770 Processor. <http://ark.intel.com/products/65719/>.
[4] Intel Corporation, Intel Atom Processor E680. <http://ark.intel.com/products/52497/Intel-Atom-Processor-E680-512K-Cache-1_60-GHz>.
[5] International Technology Roadmap for Semiconductors. <http://www.itrs.net/>.
[6] D.H. Bailey, RISC microprocessor and scientific computing, in: SC'93, Portland, OR.
[7] C. Bienia, S. Kumar, K. Li, PARSEV vs. SPLASH-2: a quantitative comparison of two multithreaded benchmark suites on chip-multiprocessors, in: IISWC'08, Seattle, WA.
[8] H. Chang, S.S. Sapatnekar. Prediction of leakage power under process uncertainties, in: ACM TODAES, 2007.
[9] C. Chen et al., Nano-electro-mechanical relays for FPGA routing: experimental demonstration and a design technique, in: DATE'12, Dresden, Germany.
[10] S. Chong et al., Integration of nanoelectromechanical (NEM) relays with silicon CMOS with functional CMOS-NEM circuit, in: IDEM'11, Washington, DC.

[11] S. Chong et al., Nanoelectromechanical (NEM) relays integrated with CMOS SRAM for improved stability and low leakage, in: ICCAD' 09, San Jose, CA.
[12] K.V. Craeynest, A. Jaleel, L. Eeckhout, P. Narvaez, J. Emer, Scheduling heterogeneous multi-cores through performance impact estimation (PIE), in: ISCA'12, Portland, OR.
[13] H.F. Dadgour, K. Banerjee, Design and analysis of hybrid NEMS-CMOS circuits for ultra low-power applications, in: DAC'07, San Diego, CA.
[14] R.G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, T. Mudge, Near-threshold computing: reclaiming Moore's law through energy efficient circuit, in: Proceedings of the IEEE Special Issue on Ultra-low Power Circuit Technology, 2012.
[15] W.M. Elgharbawy, M.A. Bayoumi, Leakage sources and possible solutions in nanometer CMOS technologies, in: IEEE Circuits and System Magazine, 2005.
[16] M. Enachescu, M. Lefter, A. Bazigos, A.M. Ionescu, S.D. Cotofana, Ultra low power NEMFET based logic, in: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), May 2013.
[17] M. Enachescu, G.R. Voicu, S.D. Cotofana, Leakage-enhanced 3D-stacked NEMFET-based power management architecture for autonomous sensors systems, in: ICSTCC'11.
[18] H. Esmaeilzadeh, E. Blem, R.St. Amant, K. Sankaralingam, D. Burger, Dark silicon and the end of multicore scaling, in: ISCA'11, San Jose, CA.
[19] N. Hardavellas, M. Ferdman, B. Falsafi, A. Ailamaki, Toward dark silicon in servers, in: IEEE Computer Society, 2011.
[20] K.R. Hoffman, P. Hegde. ARM Cortex-A8 vs. Intel Atom: Architectural and Benchmark Comparisons. Technical Report, University of Texas at Dallas, 2009.
[21] R. Jammy, Materials, process and integration options for emerging technologies, in: SEMATECH/ISMI symposium, 2009.
[22] M.W. Jang, M. Lu, T. Cui, S.C. Campbell, A 1.6 GHz NEMS actuator built from carbon nanotube layer by layer composite films, in: Device Research Conference, June 2009.
[23] S. Kaxiras, M. Martonosi, Computer Architecture Techniques for Power-Efficiency, Morgan & Claypool Publishers, 2008.
[24] E. Kultursay, K. Swaminathan, V. Saripalli, V. Narayanan, M.T. Kandemir, S. Datta, Performance enhancement under power constraints using heterogeneous CMOS-TFET multicores, in: CODES + ISSS'12, Montreal, Canada.
[25] R. Kumar, K.I. Farkas, N.P. Jouppi, Parthasarathy Ranganathan, Dean M. Tullsen, Single-ISA heterogeneous multi-core architectures: the potential for processor power reduction, in: MICRO'03, San Diego, CA.
[26] S. Li, Jung Ho Ahn, Richard D. Strong, Jay B. Brockman, Dean M. Tullsen, Norman O. Jouppi, McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures, in: MICRO'09, New York, NY.
[27] M. Monchiero, R. Canal, A. Gonzalez, Design space exploration for multicore architectures: a power/performance/thermal view, in: ICS'06.
[28] A. Raghavan et al., Computational sprinting, in: HPCA'12, New Orleans, LA.
[29] J. Renau et al., SESC Simulator, 2006.
[30] S. Rusu et al., A 45nm 8-core enterprise Xeon processor, in: A-SSCC'09, Taipei, Taiwan.
[31] V. Saripalli, G. Sun, A. Mishra, Y. Xie, S. Datta, V. Narayanan, Exploiting heterogeneity for energy efficiency in chip multiprocessors, in: IEEE Transactions on Emerging and Selected Topics in Circuits and Systems, 2011.
[32] K. Skadron, M.R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, D. Tarjan, Temperature-aware microarchitecture, in: ISCA'03, San Diego, CA.
[33] K. Swaminathan, E. Kultursay, V. Saripalli, V. Narayanan, M. Kandemir, S. Datta, Improving energy efficiency of multi-threaded applications using heterogeneous CMOS-TFET multi-cores, in: ISLPED'11, Fukuoka, Japan.
[34] M. Tabib-Azar, S.R. Venumbaka, K. Alzoubi, D. Saab, 1 V, 1 GHz NEMS switches, IEEE Sensors, 2010, 1–4, Nov. 2010, pp. 1424–1426.
[35] Y. Taur, T. Ning, Fundamentals of Modern VLSI Devices, Cambridge University Press, 2013.
[36] Y. Turakhia, B. Raghunathan, S. Garg, D. Marculescu. HaDeS: architectural synthesis for heterogeneous dark silicon chip multiprocessors, in: DAC'13, Austin, TX.
[37] R. Weerasekera, L.-R. Zheng, D. Pamunuwa, H. Tenhunen, Extending systems-on-chip to the third dimensions: performance, cost and technological tradeoffs, in: ICCAD'07, San Jose, CA.
[38] X. Wu et al., Hybrid cache architecture with disparate memory technologies, in: ISCA'09, Austin, TX.
[39] Y.-C. Yeo, T.-J. King, C. Hu, MOSFET gate leakage modeling and selection guide for alternative gate dielectrics based on leakage considerations, in: IEEE Transactions on Electron Devices, 2003.
[40] U. Zaghloul, G. Piazza, 10–25 nm piezoelectric nano-actuators and NEMS switches for multivolt computational logic, in: MEMS'13, Taipei, Taiwan.
[41] J. Zhao, X. Dong, Y. Xie, Cost-aware three-dimensional (3D) many-core multiprocessor design, in: DAC'10, Anaheim, CA.
[42] Y. Zhang, L. Peng, X. Fu, Y. Hu, Lighting the dark silicon by exploiting heterogeneity on future processors, in: DAC'13, Austin, TX.

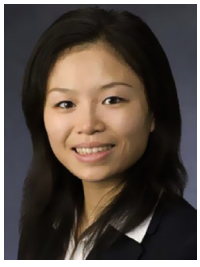


Ying Zhang received the Bachelor's and Master's degree in Electronics and Information Engineering from Huazhong University of Science and Technology, China, in June 2006 and 2008. He received his PhD degree in Electrical and Computer Engineering from Louisiana State University in 2013. He is currently working as a performance architect in Intel Corporation. His research interests include heterogeneous system design and processor reliability. He also has interests in GPU architecture.



Samuel Irving received Bachelor's degrees in both Computer Science and Electrical Engineering from Louisiana State University in December 2011. He is currently enrolled in the Electrical and Computer Engineering PhD program at LSU. His research interests include machine learning, big data analytics, and heterogeneous architecture design.

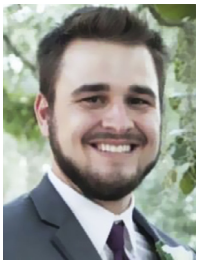
Lu Peng is currently an Associate Professor with the Division of Electrical and Computer Engineering at Louisiana State University. He received the Bachelor's and Master's degrees in Computer Science and Engineering from Shanghai Jiao Tong University, China. He obtained his Ph.D. degree in Computer Engineering from the University of Florida in Gainesville in April 2005. His research focus on memory hierarchy system, reliability, power efficiency and other issues in CPU design. He also has interests in Network Processors. He received an ORAU Ralph E. Powe Junior Faculty Enhancement Awards in 2007 and a Best Paper Award from IEEE International Conference on Computer Design in 2001. Dr. Peng is a member of the ACM and the IEEE Computer Society.



Xin Fu received the Ph.D. degree in Computer Engineering from University of Florida in 2009. She was a NSF Computing Innovation Fellow with the Computer Science Department, University of Illinois at Urbana-Champaign from 2009 to 2010. From 2010 to 2014, she was an Assistant Professor at the Department of Electrical Engineering and Computer Science, University of Kansas. She joined the Electrical and Computer Engineering Department, University of Houston since Fall 2014. Her research interests include computer architecture, high-performance computing, energy-efficient computing, hardware reliability and variability, and emerging technologies. Dr. Fu is a recipient of 2014 NSF Faculty Early CAREER Award, 2012 Kansas NSF EPSCoR First Award, and 2009 NSF Computing Innovation Fellow.

David Koppelman received his Ph.D. in Computer Engineering from RensselaerPolytechnic Institute. He is currently an Associate Professor in the Department of Electrical and Computer Engineering at Louisiana State University. His interests include parallel computation and computer architecture.

Weihua Zhang received the Ph.D degree in computer science from Fudan University in 2007. He is currently an associate professor of Parallel Processing Institute, FudanUniversity. His research interests are in compilers, computer architecture, parallelization and systems software.



Jesse Ardonne received his Bachelor's degree in Engineering Technology from Southeastern Louisiana University in May 2012. He is currently pursuing his PhD at Louisiana State University. His research interests include computer architecture and cache management policies.