DELCA: DVFS Efficient Low Cost Multicore Architecture

Shoumik Maiti
Synopsys Inc.
Chandler, AZ, U.S.A.
shoumik.maiti@synopsys.com

Sudeep Pasricha
Colorado State University
Fort Collins, CO, U.S.A.
sudeep@colostate.edu

ABSTRACT
The energy efficiency of dynamic voltage and frequency scaling (DVFS), a popular technique used for energy and thermal management, is substantially reduced in deep sub-micron (DSM) technology nodes. This can be attributed to a limited voltage scaling range in DSM nodes as well as the observation that circuits designed for highest performance are inherently power hungry and energy inefficient at other operating points. This paper proposes a DVFS efficient low-cost multicore architecture (DELCA) for dark silicon that is energy efficient and has much lower cost and faster design cycle time compared to custom designed heterogeneous or dynamically reconfigurable multicores. We also propose a runtime flexible application degree of parallelism based scheduling scheme (FlexDoP). Experimental results indicate that DELCA + FlexDoP can improve energy efficiency by 24.4% and enhance throughput by 16.5% compared to conventional multicores for a given power budget.

Keywords: DVFS; dark silicon; multicores; run time scheduling

1. INTRODUCTION
DVFS is used to achieve high energy efficiency and reduce thermal emergencies in multicore processors across all segments of computing [1]. Typically with DVFS, the supply voltage and frequency of a processor core is lowered during the non-compute-intensive phases of a workload. But the potential power and energy savings from DVFS are substantially reduced in deep sub-micron (DSM) technology nodes because *(i)* there is reduced head-room between threshold voltage (V_{th}) and nominal (NOM) operating voltage (V_{dd}), and *(ii)* gate sizes and V_{th} of underlying circuits that are optimized for highest performance do not change with operating conditions [2], [11]. The increased design guard bands required for process variations [22], [23] in DSM also significantly reduce the energy efficiency of DVFS. One possible solution to overcome the limited DVFS headroom and increase energy efficiency in DSM nodes is to increase the voltage scaling range to encompass near threshold computing (NTC) and turbo boost (TB) modes of operation [3], [13]. It has been shown that the NTC mode of operation is highly energy efficient and TB mode realizes high throughput for a given power budget. However, cores designed to operate in NTC mode (with operating voltage close to V_{th}) are not energy efficient in the NOM and TB modes because they require higher design margin to overcome sensitivity to process variations at low operating voltages [4], [5]. Other techniques to enhance the efficiency of DVFS include using heterogeneous multicores where separate cores are used for optimal performance at various modes of operation [6] or using dynamically reconfigurable cores which change their microarchitecture based on the mode of operation [7]. However, such approaches increase the intellectual property (IP) cost, development and verification time, and substantially delay time-to-market. There has thus been an increased emphasis in the semiconductor industry to develop low cost integrated circuits due to commoditization of hardware and increased pricing pressure.

The rising power density and likelihood of thermal hotspots due to technology scaling in recent years have also caused emerging multicores to be severely power limited and given rise to the dark/dim silicon phenomenon, where a significant fraction of a chip is operated in low power mode or is shut down to satisfy the chip power budget [8]. Technology scaling has also enabled packing of more processor cores on a chip, which introduces a new paradigm to optimize energy efficiency and throughput in multicores: by exploiting the degree of parallelism (DoP) of applications [9].

This paper addresses the unique challenge of designing ultra-low cost multicore processors that opportunistically use the dark silicon chip budget and DSM transistor characteristics to achieve high DVFS efficiency and also reduce the time-to-market. We propose a novel DVFS efficient low-cost multicore architecture (*DELCA*) that uses homogeneous cores for TB, NOM and low voltage (LV) modes of operation, where the cores are synthesized separately for each operating mode to achieve energy efficiency across the extended DVFS range. We select the lowest voltage of our DVFS range well below the NOM voltage but higher than NTC to reduce high design margins required for NTC operation [7]. We use the extra transistor budget due to dark silicon to provision the multicore system with banks of energy efficient cores specially synthesized for TB, NOM, and LV modes of operation and use them based on the power and performance requirements of incoming applications at runtime. With *DELCA*, there is no need to design special heterogeneous or reconfigurable cores for different modes of operation, which significantly reduces development time and IP cost. As we will see later, *DELCA* also does not require expensive per core DVFS controllers as each bank of cores operates at a fixed voltage and frequency and still achieves significant improvement in energy efficiency and throughput over conventional multicore systems. We also propose a system level flexible DoP application scheduling framework (FlexDoP) for use with *DELCA* that selects the available cores from TB/NOM/LV banks and picks the optimal DoP of incoming applications at runtime to achieve high energy efficiency and throughput for a given power budget. Our novel contributions in this paper are as follows:

- We present a CAD flow to synthesize homogeneous cores from the ground up that significantly enhances the DVFS efficiency in the extended LV to TB range of operation;
- We devise a low cost and DVFS efficient DELCA architecture targeted towards dark silicon which achieves substantial energy efficiency and throughput enhancement without the overheads of designing special heterogeneous or dynamically reconfigurable cores, or per core voltage regulators and synchronization FIFOs of fine-grained DVFS systems;
- We also design a system-level flexible DoP runtime scheduling framework (FlexDoP) for use with DELCA that selects available cores from the TB/NOM/LV banks and also selects the op-

GLSVLSI '17, May 10-12, 2017, Banff, AB, Canada

DOI: http://dx.doi.org/10.1145/3060403.3060422

timal DoP of incoming applications at runtime to enhance energy efficiency and throughput for a given power budget;
- Lastly, we quantify the benefits of DELCA + FlexDoP over conventional DVFS systems for PARSEC [10] benchmarks.

2. RELATED WORK

There are several prior works that focus on enhancing the efficiency of DVFS in multicores. Khang et al. [11] propose context aware multi-mode synthesis, micro-architectural changes, and selective block replication to achieve energy efficiency in the DVFS operating range. However context aware multi-mode design does not possess the full benefits of a ground up core synthesis targeted for different operating points. Moreover micro-architectural changes increase design and verification cost, and increase time-to-market. Gopireddy et al. [7] propose voltage scalable cores that operate in an energy efficient manner across the DVFS range by using different voltages for storage and logic elements and a pipeline architecture that dynamically reconfigures during low performance mode of operation. This variable latency pipeline also increases the development cost and time-to-market. Maiti et al. [3] and Cai et al. [13] have studied the efficiency of extended range DVFS encompassing the TB and NTC mode of operation with process variation aware thread to core mapping in multicores with fine grained voltage frequency control. But as the cores are not optimized for each mode of operation the energy savings are sub optimal and the per core voltage/frequency control increases the design cost.

Heterogeneous multicores [6] and application specific co-processors have also been proposed to improve the energy efficiency of computation. Both of these approaches increase the IP cost of a chip and design and verification overheads. Moreover application specific cores are not energy efficient for other computations beyond which they are specially designed. Chakraborty et al. [2] proposed homogenous cores synthesized for specific operating conditions in a multicore system. However their work does not consider the extended DVFS range encompassing the TB and LV mode and dark silicon transistor budget. Bokhari et al. [12] propose replication of voltage/frequency optimized routers in networks-on-chip (NoCs) and using the most energy efficient router at runtime based on traffic load. But such replication complicates wire routing and flow control. Kapadia et al. [24] and Raparti et al. [25] studied reliability issues in dark silicon DSM technology nodes.

To the best of our knowledge, this is the first work that targets the design of ultra-low cost and fast time-to-market energy efficient multicore processors for the dark silicon era. DELCA uses the dark silicon budget to add energy efficient cores synthesized for target operating points. At runtime, cores and DoP of incoming applications are selected to achieve high energy efficiency and throughput. Our approach has much lower cost and development cycle time than heterogeneous multicores or dynamic reconfigurable processors.

The rest of the paper is organized as follows. Section 3 gives the details of the CAD flow to design energy efficient homogeneous cores for an extended DVFS range. Section 4 describes our proposed *DELCA* multicore architecture. Section 5 explains the FlexDoP runtime scheduling framework for *DELCA*. Section 6 presents the experimental results and Section 7 concludes the paper.

3. HOMOGENEOUS DVFS CORE SYNTHESIS

A DVFS-capable processor core needs to work reliably at all voltage-frequency operating points in the DVFS range. In a conventional design flow, a DVFS core is synthesized to ensure that it meets the timing constraints at the highest performance (highest voltage and frequency operating point) and the lower voltage and frequency operating points in the DVFS scaling range are selected such that all the required timing constraints are met at these conditions. However the speed of each timing path scales differently with voltage. This scaling depends on the logic depth, type of threshold voltage (V_t) and gate lengths of cells in the timing path and the length of interconnects. Due to these dependencies, the critical timing path changes in each mode of operation [11]. The performance of a DVFS-capable core designed using a conventional synthesis flow is sub-optimal at lower voltage and frequency.

To obtain the best performance at all operating points, EDA vendors have developed a multi-corner multi-mode (MCMM) synthesis flow where all the design operating modes and timing constraints are made available to the synthesis tool and the design is optimized simultaneously for all operating conditions [14]. The design obtained using an MCMM synthesis flow is also not energy optimal at different operating points, compared to the DVFS-capable cores that are synthesized specifically for the operating condition, due to the mix of cell sizes and V_t which DVFS cannot alter.

Table 1: Percentage of Low V_t (LVT) cells, avg. drive strength, and power saved using targeted synthesis for various voltage/frequency operating points with a 16nm FinFET standard cell library

Operating point	Mode	% of LVT cells	Avg. drive strength	% Power saved
1.1V 4.0 GHz	TB	28.70	1.54	-
0.88V 3.0 GHz	-	9.32	1.44	17.67
0.80V 2.5 GHz	NOM	6.87	1.36	26.21
0.72V 2.0 GHz	-	5.67	1.40	38.4
0.64V 1.5 GHz	-	8.67	1.67	35.14
0.59V 1.0 GHz	LV	9.68	1.79	32.70

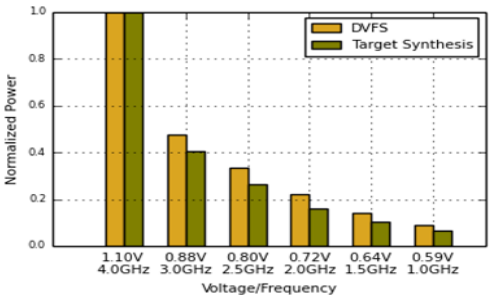


Fig. 1: Normalized power of processor core synthesized for various target operating points vs. applying DVFS on core synthesized for highest performance operating point as done in conventional synthesis flows

To highlight the inefficiency of a conventional synthesis flow for DVFS-capable cores, we have synthesized a proprietary ARM® NEON™ [15] like core used in the ARM® Cortex®-A8 [16] processor using commercial 16nm FinFET standard cell library with multi-V_t cells. We have used Synopsys® Design Compiler Graphical™ [17] to synthesize for operations across the extended DVFS range from 1.1V (TB mode) to 0.59V (LV mode) and employed Synopsys® PrimeTime PX™ [18] to analyze power.

Table 1 and Figure 1 summarize the synthesis results. Table I shows that the core synthesized for highest performance (TB mode) has a higher number of low V_t cells (28.70%) compared to the same core synthesized for lower performance LV mode (9.68%). The low V_t cells have higher leakage than standard and high V_t cells. The average drive strength (taken from standard cell library) in the high performance (TB) mode is also significantly higher than for cores synthesized for other DVFS operating points (NOM mode). Cells with higher drive strength dissipate more dynamic and leakage power than low drive strength cells. Figure 1 shows that a significant amount of power (17.67% - 38.40%) may be saved by synthesizing cores for each operating point in the DVFS spectrum, instead of applying DVFS on a core that is synthesized for highest performance, as done in conventional synthesis flows. This is due to the fact that DVFS cannot change underlying circuits. Cores synthesized for higher per-

formance have higher drive strength cells which affects both the dynamic and leakage power. Such cores also have a considerably higher percentage of low V_t cells that increase leakage power, which is a significant portion of total power in DSM nodes.

Table 1 also shows that the average drive strength in the cores synthesized for lower performance (e.g., LV mode) is higher, to achieve the desired performance at the lower operating voltage. This is because the speed of CMOS transistors does not scale proportionally with voltage but the delay degrades much rapidly beyond a certain voltage. However the netlists in LV mode still have a significantly lower percentage of low V_t cells than the high performance TB mode that results is substantial savings in leakage power.

In summary from our synthesis experiments we conclude that significant improvements in energy efficiency may be achieved if we can include cores synthesized for each operating point in a DVFS capable multicore processor. This is the premise of our proposed DELCA architecture that is described in the next section.

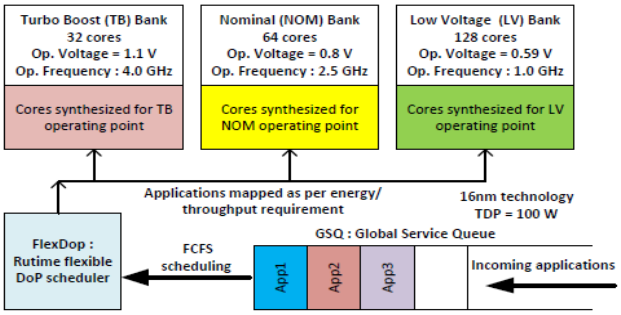


Fig. 2: Proposed DELCA + FlexDoP architecture

4. PROPOSED DELCA ARCHITECTURE

The DVFS Efficient Low Cost Multicore Architecture (*DELCA*) aims to: *(i)* increase DVFS efficiency by using energy optimal cores for each operating point; *(ii)* effectively use the dark silicon budget in deeply scaled technology nodes to provision cores for energy efficiency across different operating points while eliminating costly per-core DVFS controllers; and *(iii)* provide a lower-cost, faster time-to-market, and more energy efficient solution than expensive custom heterogeneous or dynamically reconfigurable core designs.

As discussed earlier, the increased power density in DSM technology nodes has led to the dark/dim silicon phenomenon where an increased fraction of the chip is shut-down/run at a low power mode to meet the chip thermal design power (TDP) budget. It is observed in [8] that about 50% of the chip needs to be powered off in 8nm technology to meet the power budget. While there is some improvement after transitioning to FinFET technology from planar bulk CMOS, dark silicon still accounts for a significant chip area [19]. This provides an interesting opportunity to effectively use the dark silicon budget to develop low cost DVFS-efficient multicores.

In DELCA, we propose to use replication of homogeneous cores synthesized for TB, NOM, and LV modes of operation (as described in Section 3) to substantially increase the DVFS efficiency in DSM technology nodes. We use the available dark silicon budget to provision the multicore processor with banks of homogeneous cores specially synthesized for TB, NOM, and LV modes of operation. The details of the architecture are illustrated in Fig. 2. The global service queue (GSQ) buffers all the incoming applications and a global runtime scheduler (FlexDoP, discussed in more detail in Section 5) selects the degree of parallelism (DoP) for incoming applications and maps the applications to the available cores.

The synthesis of cores for specific operating points enables DELCA multicore processors to achieve high energy efficiency across the entire DVFS range without the overhead of custom design. Several prior works have suggested the use of heterogeneous multicores and also use of dynamic reconfigurable cores to enhance the DVFS efficiency. These techniques have been widely adopted and have good use in specialized applications. However this substantially increases the IP cost and time-to-market because of increased design and verification effort for each specialized processor core and thus the approach is not suitable for low-cost applications.

It should also be noted that DELCA does not need any expensive per core voltage regulators, DVFS controllers, and synchronization FIFOs that are common in fine grained DVFS systems. Each bank (NTB/NOM/LV) works at a specific voltage and frequency and needs only a single voltage regulator and clock generator per bank. As the applications are only mapped within a single bank, there is no need to include synchronization FIFOs between the cores of banks. This further simplifies the design and reduces cost. DELCA addresses this pressing need for low-cost and faster time-to-market multicore architectures designed for energy efficiency.

Table 2: Baseline core architecture in DELCA

Parameters	Values
Core Model	Intel®-X86 Gainestown®
L1 Caches	Private 32 KB, 4-way SA, LRU
L2 Caches	Private 256 KB, 4-way SA, LRU
L3 Caches	Shared 32 MB, 16-way SA, LRU
Technology	16 nm FinFET

The choice of 3 banks is motivated by the potential percentage of dark silicon in DSM nodes [8], [19]. Higher energy savings may be obtained by adding more banks with finer voltage/frequency control but this would increase the amount of dark silicon. The number of cores in TB, NOM, and LV banks is selected based on the baseline core power at these operating modes and an available chip TDP budget of 100 Watts. Table 2 describes the baseline core architecture. The TB bank is provisioned with 32 cores, NOM bank is provided with 64 cores, and the LV bank has 128 cores. TB cores are useful to maximize throughput for time critical applications for a given power budget. These cores can also be used to execute serial applications which do not benefit from increasing DoP. The applications that benefit from moderate parallelism and have relaxed deadlines are ideal for NOM banks to achieve high throughput and good energy efficiency. The large number of cores in the LV bank in DELCA can be used to execute applications in an energy efficient and highly parallel manner, without sacrificing throughput.

From the discussion above, to fully exploit the DELCA architecture, there is a need for a runtime scheduling framework that can select the DoP of incoming applications and map them to the appropriate cores in the TB/NOM/LV banks, to achieve throughput and energy targets as well as ensuring that the dark silicon TDP budget is not violated. We propose a runtime scheduler (FlexDoP) to address this need. This scheduler is discussed in the next section.

5. FLEXDOP SCHEDULING FRAMEWORK

5.1 Motivation for DoP based scheduling

The execution of multi-threaded workloads has become common with the widespread adoption of multicore processors. Increasing the degree of parallelism (DoP or number of threads) for an application does not necessarily improve performance. At higher DoPs, the serial part of the application tends to dominate the execution time and inter-thread communication affects its energy efficiency.

To analyze the impact of DoP on application performance and energy efficiency, we profiled several PARSEC benchmarks on our baseline core architecture from Table 2. Figures 3(a)-(b) show the normalized execution time and normalized energy of five PARSEC benchmarks with DoP varying from 2 to 32 on our baseline cores. The execution time and energy of higher DoPs have been normalized

with respect to the baseline DoP of 2. We used the Sniper [20] architectural simulator for this study and power numbers are obtained from the McPAT tool [21]. It can be observed that for some PARSEC benchmarks such as *blackscholes* and *canneal*, the execution time and energy reduces with increasing DoP as these applications benefit from parallelization and the minimum execution time and energy is obtained for a DoP of 32. However for other benchmarks such as *streamcluster* and *swapoptions*, the execution time and energy metrics deteriorate as DoP increases beyond a certain value, due to increased communication overheads between threads. The *streamcluster* benchmark achieves the minimum execution time at a DoP of 8 and minimum energy consumption at a DoP of 4. Similarly, *swapoptions* attains a minimum execution time and energy at DoP of 16 on the target architecture. The serial part of the application dominates in the *freqmine* benchmark where execution time and the energy progressively deteriorates with increasing DoP.

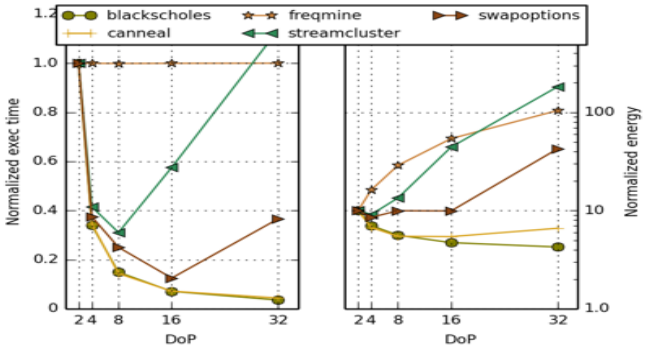


Fig. 3: (a) Normalized execution time of PARSEC benchmarks vs. DoP on baseline core architecture; (b) Normalized energy consumption of PARSEC benchmarks vs. DoP; on baseline core architecture

From these results, it can be concluded that for each application, there is an optimal DoP for highest performance and an optimal DoP for the best energy efficiency. This observation motivates us to develop the FlexDoP runtime scheduler that works on the DELCA architecture to select the DoP of incoming applications and map them to the available cores in the TB/NOM/LV banks, with the aim of achieving maximum throughput for time critical applications and minimum energy for applications that have relaxed deadlines.

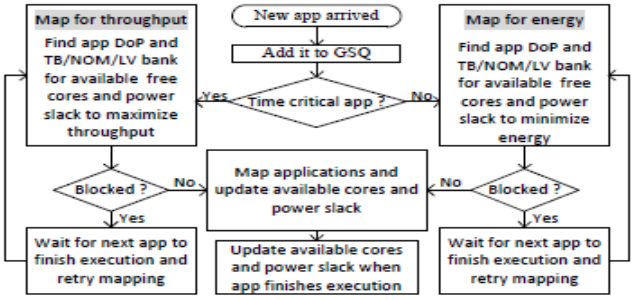


Fig. 4: FlexDoP runtime scheduler

5.2 FlexDoP Runtime Scheduling Framework

All incoming parallel applications in DELCA are buffered in the global service queue (GSQ; Figure 2)**.** The FlexDoP runtime scheduler processes the applications from the head of service queue on a first come first serve (FCFS) basis. The applications are restricted to be scheduled in only one of the TB/NOM/LV banks, which ensures that all the threads of a job are run at the same frequency, thereby eliminating the need for synchronization between the banks. The FlexDoP runtime scheduler keeps track of the available cores in each of the three banks and also the available power slack based on the fixed TDP budget. The scheduler is assumed to have access to application profile information related to the execution time and average power dissipation of each application on the TB/NOM/LV banks of DELCA for the DoPs of 2, 4, 8, 16 and 32. Each application also has a parameter that indicates if it is time critical or not.

Figure 4 summarizes the operation of the FlexDoP scheduler. At runtime, FlexDoP picks the application at the head of the GSQ. If the application is time critical, FlexDoP finds the best combination of application DoP and the corresponding available cores from the TB/NOM/LV banks such that throughput is maximized. If the application is not time critical, FlexDoP selects the best combination of application DoP and free cores to minimize energy.

It should be noted that FlexDoP finds the best DoP for the application being mapped at runtime on the target architecture. This depends on the applications that are in progress as they impact the free cores and available power slack. FlexDoP maps the application to maximize throughput or minimize energy based on application type as soon as a valid mapping can be done that does not violate the TDP. While it may be possible to obtain better throughput or energy efficiency for the application being mapped if FlexDoP waits for more cores of certain types and power slack to become available and then selecting a different DoP. However, this typically increases the wait time of applications and reduces the overall throughput of the system. If no cores are available or there is not enough power slack to map the incoming application, FlexDoP is blocked. It waits for enough scheduled jobs to finish and free-up cores and power slack to become available such that the job can be mapped.

Algorithm 1: Map for throughput

inputs: App from head of GSQ
App profile: exec. time, power info for all DoP and core types
System State: free cores in all banks and power slack

```
1.  blocked ← true , maxTpDoP ← null, maxTpBank ← null
2.  maxThroughput ← 0, mapPower ← null
3.  for bank in {TB,NOM,LV} :
4.      for dop in {2,4,8,16,32} :
5.          if (cores available and mapping power < power slack) :
6.              if (throughput for this mapping > maxThroughput) :
7.                  maxTpDoP ← dop, maxTpBank ← bank
8.                  blocked ← false
9.                  maxThroughput ← throughput for this mapping
10.                 mapPower ← power for this mapping
```

outputs: blocked, maxTpDoP (app DoP), maxTpBank (bank to which app is mapped) , mapPower (power for this mapping)

Algorithm 2: Map for energy

inputs: App from head of GSQ
App profile: exec. time, power info for all DoP and core types
System State: free cores in all banks and power slack

```
1.  blocked ← true , minEnDoP ← null, minEnBank ← null
2.  minEnergy ← ∞, mapPower ← null
3.  for bank in {TB,NOM,LV} :
4.      for dop in {2,4,8,16,32} :
5.          if (cores available and mapping power < power slack) :
6.              if (energy for this mapping < minEnergy) :
7.                  minEnDoP ← dop, minEnBank ← bank
8.                  blocked ← false
9.                  minEnergy ← energy for this mapping
10.                 mapPower ← power for this mapping
```

outputs: blocked, minEnDoP (app DoP), minEnBank (bank to which app is mapped), mapPower (power of this mapping)

Algorithms 1 and 2 summarize the map for throughput and map for energy mapping schemes used by FlexDoP (Figure 4). If there are N_b banks in the system and N_d possible DoPs for applications our mapping algorithm requires $O(N_b \cdot N_d)$ comparisons to select the DoP and banks. In the case of DELCA, N_b is 3 and we assume a maximum value of N_d to be 5. Thus we need to perform at most 15 comparisons

before application mapping, which makes our approach very feasible for fast runtime implementation. Our algorithm is also very scalable for higher number of core banks and application DoP counts.

6. EXPERIMENTS

6.1 Experimental setup

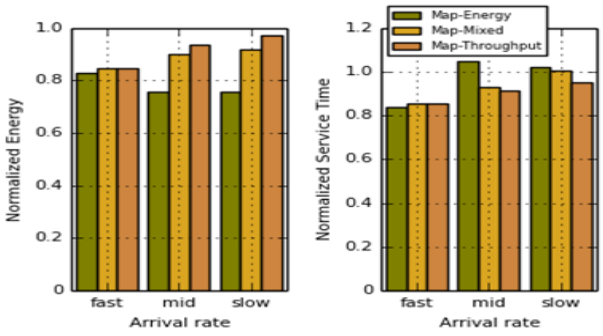
We evaluate a DELCA processor with three separate banks of cores (one bank each for TB, NOM, and LV modes of operation) as shown in Figure 2. The baseline core architecture is as described in Table 2, with a private L1, L2 cache per core, and a shared L3 last level cache (LLC). The cores in each bank are synthesized and hardened for their respective target operating conditions, for energy efficiency. The TB bank has 32 cores, NOM bank has 64 cores, and the LV bank has 128 cores. Each bank operates at a fixed voltage and frequency as follows: *(i)* TB core: V_{dd} = 1.1V and F = 4.0 GHz; *(ii)* NOM core: V_{dd} = 0.8V and F = 2.5 GHz and *(iii)* LV core: V_{dd} = 0.59V and F = 1.0 GHz. We also account for voltage level shifters and synchronizing FIFOs between the shared L3 cache and the banks. A 2D mesh NoC is used to communicate between the cores in each bank. The incoming applications are mapped on a single bank, so there is no communication overhead between the banks for any mapped application. The application also stays in the same bank for its entire execution and hence there is no latency due to task migration or dynamic change of voltage and frequency. To save power, unused cores are kept in deep sleep mode.

The following PARSEC [10] benchmarks are used in our simulation based analysis: *blackscholes, bodytrack, canneal, deadup, facesim, ferret, fluidanimate, freqmine, raytrace, streamcluster, swapoptions*, *vips* and *x264*. Each PARSEC benchmark is profiled on the base NOM cores of DELCA (Table 2) for the 16nm technology node. The execution time and power numbers on NOM cores for DoPs of 2, 4, 8, 16 and 32 are obtained using Sniper [20] micro architectural simulations and McPAT [21] respectively. The simulation time and power numbers on other cores are obtained using the frequency differences and power scaling factors obtained from synthesis experiments (similar to as discussed in section 3).

For our experiments, we generated a workload consisting of a random sequence of 1000 PARSEC benchmark applications. We varied the arrival time of the applications to simulate high, medium and low loading of the multicore system. During fast arrival (high loading) the applications arrive randomly much faster than their mean best case execution time. For medium arrival (medium loading) the applications arrive randomly at the rate of mean execution time. Similarly, for slow arrival (low loading) the applications arrive randomly much slower than their worst case execution time. For each arrival rate we have also generated three separate set of workloads of 1000 random applications where *(i)* none of the applications are time critical and all are mapped for energy efficient execution; *(ii)* 50% of the applications are time critical and are mapped for throughput and the remaining 50% are mapped for energy efficiency; and *(iii)* all applications are time critical and mapped for throughput. The dark silicon TDP budget was assumed to be 100W.

6.2 Experimental results

For comparison purposes, we created a reference system called *DVFSM* where all cores of *DELCA* are replaced by highest performance TB cores (i.e., all the cores are synthesized and hardened for the TB target operating mode). This reference system represents the approach used in conventional state-of-the-art homogenous DVFS-based systems with extended range DVFS as discussed in [3], [13]. The energy consumption and application service time of our proposed *DELCA* architecture are compared with *DVFSM* for workloads with fast/mid/slow application arrival and mixed time criticality. All application mappings are performed using the FlexDoP runtime scheduler for both *DELCA* and *DVFSM*. For a fair comparison, *DVFSM* is assumed to have the same number of cores as *DELCA*, operating at TB/NOM/LV modes and with the same TDP of 100W. The cores in *DVFSM* are not synthesized for target operating conditions (synthesized for TB mode); instead the cores use DVFS to operate at NOM/LV modes.



(a) **(b)**

Fig. 4: (a) Normalized energy of DELCA Vs DVFSM (b): Normalized Service Time of DELCA Vs DVFSM

Figure 4(a) compares the energy consumption of DELCA with that of DVFSM for the three workloads consisting of 1000 randomly selected PARSEC benchmarks, for fast, mid and slow arrival rates and mixed time criticality. The energy consumption numbers for *DELCA* in the figure are normalized with respect to that of *DVFSM*. The first bar in the clustered bar plot represents the case when all of the applications are non-time critical and mapped for energy efficiency. The second bar is for mixed time critical applications where 50% of the applications are mapped for energy efficiency and the other 50 % are mapped for throughput. All the applications in the third bar are time critical and mapped for throughput.

The cores in *DELCA* are more energy efficient than DVFSM as they are synthesized separately for different target operating conditions. When all the applications are mapped for energy, *DELCA* is 17.7% more energy efficient than DVFSM for fast application arrival rate and 24.2% and 24.4% more energy efficient than *DVFSM* for mid and low application arrival rates respectively. At a lower application arrival rate, more applications may be mapped on energy efficient NOM/LV cores in *DELCA* as they become available more frequently than for high application arrival rates. For a high application arrival rate, *DELCA* is still 15.2% more energy efficient than *DVFSM* when all applications are mapped for throughput. As TB cores are not always available when mapping for throughput and when application arrival rate is high, some applications end up using energy efficient NOM/LV cores. However when all applications are mapped for throughput for a low application arrival rate, more applications are mapped to TB cores and energy usage in *DELCA* is similar to *DVFSM*. Note that the TB mode cores are the same in *DELCA* and *DVFSM*.

Figure 4(b) compares the average service time of an application (service time is defined as the sum of wait time in the global service queue and the run time for the applications) in the *DELCA* architecture with respect to *DVFSM* for the three workloads consisting of 1000 randomly selected PARSEC benchmarks, for fast, mid and slow arrival rates and mixed time criticality. The service time numbers for *DELCA* in the figure are normalized with respect to that of *DVFSM*. The order of bars in the clustered bar graph follows the same order for application mapping as in Figure 4(a).

DELCA achieves 16.5% improvement in service time compared to *DVFSM* when all applications are mapped for energy and 14.9% improvement in service time when all applications are mapped for throughput for fast application arrival rate. During fast application arrival, the lower power of NOM/LV cores in *DELCA* compared to cores operated at NOM/LV modes via DVFS in *DVFSM* enables

more applications to run simultaneously in *DELCA* for a given TDP budget. This in turn reduces the wait times and hence the total service time of the applications. It should be noted that the service improvement is achieved despite the TB/NOM/LV cores running at the same frequencies in both *DELCA* and *DVFSM*. For a medium (mid) application arrival rate, *DELCA* cores are able to cut down on the wait time of the applications and the service time is improved by 8.7% from *DVFSM* for applications that are mapped for throughput. For slow application arrival rates, there are not enough applications to run in parallel, so the service time of *DELCA* is similar to that of *DVFSM*.

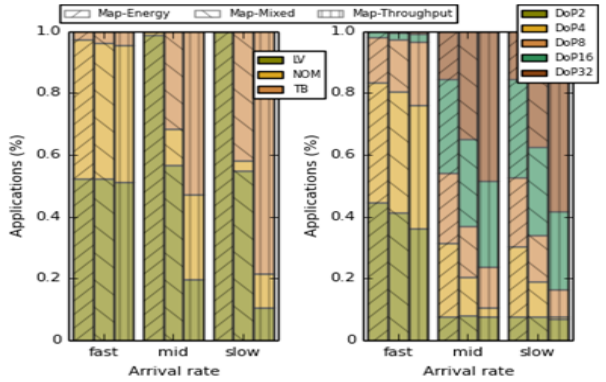


Fig. 5: (a) Core usage in *DELCA* with FlexDoP scheduling (b) DoP usage in *DELCA* with FlexDoP scheduling

To gain more insights into the performance with our proposed *DELCA* architecture, we analyzed the usage of different types of cores from the TB/NOM/LV banks and the use of different application DoPs by FlexDoP for the workloads with fast/mid/slow arrival and mixed time criticality. Figure 5(a) gives the details of the core usage in *DELCA* with FlexDoP scheduling. When applications are mapped for energy, 52.5% of applications are mapped on LV cores for a fast application arrival rate and 100% of the applications are mapped on LV cores for a slow application arrival rate. This behavior is what we expect: LV cores are more preferable for energy efficient execution and more LV cores are available during mapping when applications arrive slowly. Similarly, when the applications are mapped for throughput, TB cores are preferred and as many as 78.4% of the applications are mapped on to TB cores when mapped for throughput for slow application arrival rates.

Figure 5(b) gives the details of DoP usage in *DELCA* with FlexDoP scheduling. Each application may be mapped with a DoP of 2, 4, 8, 16, or 32. For a slow application arrival rate, more cores are available during mapping and a significant fraction of the applications use higher DoPs when mapped either for energy or throughput. It can be observed that 58.3% of the applications are mapped with a DoP of 32 when mapped for throughput for a slow application arrival rate. To reduce the wait time for high application arrival rate scenarios and also due to the lower of the number of cores available during mapping, most of the applications use lower DoPs when mapped either for energy or throughput. It can be observed that 83.3% of the applications when mapped for energy and 76.2% of the applications when mapped for throughput use DoP values of 2 and 4 for fast application arrival rate scenarios.

7. CONCLUSION

Emerging multicores are severely power limited in the dark silicon era. DVFS is widely used to achieve performance goals while staying within the chip power budget. However, DVFS efficiency is steadily decreasing in DSM technology nodes due to the reduced headroom between V_{dd} and V_{th}. In addition, DVFS cannot alter the characteristics of underlying circuits designed to operate at the highest performance. There is also an increased focus in the semiconductor industry to design low cost and fast time-to-time to market multicore processors. *DELCA* addresses this unique need to develop ultra low-cost and energy efficient multicore processors that also intelligently leverages the dark silicon transistor budget. Our experimental results show that *DELCA* can achieve up to 24.4% improvement in energy efficiency and 16.5% improvement in throughput over conventional DVFS multicore architectures.

REFERENCES

[1] V. Hanumaiah et al., "Energy-efficient operation of multicore processors by dvfs, task migration, and active cooling", IEEE Trans. on Comp., Feb 2014.

[2] K. Chakraborty et al., "Architecturally Homogeneous Power-Performance Heterogeneous Multicore Systems", IEEE TVLSI, April 2013

[3] S. Maiti et al., "Process variation aware dynamic power management in multicore systems with extended range voltage/frequency scaling", MWSCAS, 2015

[4] R. Dreslinski et al., "Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits", Proc. of IEEE, Feb. 2010

[5] S. Jain et al., "A 280mW-to-1.2V Wide-Operating-Range IA-32 Processor in 32nm CMOS", ISSCC, 2012

[6] big.LITTLE Technology: The Future of Mobile [Online] https://www.arm.com/files/pdf/big_LITTLE_Technology_the_Futue_of_Mobile.pdf

[7] Gopireddy et al., "ScalCore: Designing a core for voltage scalability", HPCA, 2016

[8] H. Esmaeilzadeh et al., "Dark silicon and the end of multicore scaling", ISCA, 2011

[9] N. Kapadia et al., "VARSHA: Variation and reliability-aware application scheduling with adaptive parallelism in the dark silicon era", DATE, 2015

[10] C. Bienna et al., "The Parsec benchmark suite: Characterization and architectural implications", PACT, 2008.

[11] A. Kahng et al., "Enhancing the Efficiency of Energy-Constrained DVFS Designs", IEEE Trans. on VLSI, Oct, 2013

[12] H. Bokhari et al., "Malleable NoC: Dark Silicon inspired adaptive Network-on-Chip", DATE, 2015

[13] E. Cai et al., "Learning-Based Power/Performance Optimization for Many-Core Systems with Extended Range Voltage/Frequency Scaling", IEEE Tran. on CAD, Aug 2016

[14] Synopsys® IC Compiler ™ User Manual [Online] http://www.synopsys.com

[15] ARM® NEON™ Core [Online] http://www.arm.com

[16] ARM® Cortex®-A8 Core [Online] http://www.arm.com

[17] Synopsys® Design Compiler Graphical™ User Manual [Online] http://www.synopsys.com

[18] Synopsys® PrimeTime PX™ User Manual [Online] http://www.synopsys.com

[19] A. Bejestan et al., "Analyzing the Dark Silicon Phenomenon in a Many-Core Chip Multiprocessor under Deeply-Scaled Process Technologies", GLSVLSI, 2015

[20] T. Carlson et al., "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations" SC, 2011

[21] S. Li et al., "The McPAT framework for multicore and manycore architectures: Simultaneously modeling power, area and timing." TACO, 2011

[22] N. Kapadia et al, "VISION: A Framework for Voltage Island Aware Synthesis of Interconnection Networks-on-Chip", GLSVLSI, 2011

[23] N. Kapadia et al, "VERVE: A Framework for Variation-Aware Energy Efficient Synthesis of NoC-based MPSoCs with Voltage Islands", ISQED, 2013

[24] N. Kapadia et al, "ARTEMIS: An Aging-Aware Run-Time Application Mapping Framework for 3D NoC based Chip Multiprocessors", NOCS, 2015

[25] V. Y. Raparti et al, "CHARM: A Checkpoint-based Resource Management Framework for Reliable Multicore Computing in the Dark Silicon Era,", ICCD, 2016