GreenDroid Mobile Application Processor

Sonal Sarode[1], Ruchika Singh[2]

[1]*E&TC Department, GMCOE, Balewadi, Pune, Maharastra, India.*
Savitribai Phule Pune University.
(E-mail: 27sonal@gmail.com)

[2]*E&TC Department, GMCOE, Balewadi, Pune, Maharastra, India.*
Savitribai Phule Pune University.
(E-mail:ruchikasingh03@gmail.com)

***Abstract*—** As innovation in microprocessor desktop processors are sonly replaced by mobile application processors. These processors support multicore processing and execution, so these processors are more power hungry. In mobile application processors fundamental limiter is dark silicon. In recent work, we are using this dark silicon which is natural evolution of modern mobile processor. We are going to create energy saving cores which are generated automatically using dark silicon area. These automatically generated cores are called as conservation cores, which can reduce consumption of energy. This paper describes GreenDroid, in which conservation cores are used to save energy across hotspots in general purpose smart phone applications.

Keywords*—Smartphone application, Conservation Cores, Dark silicon, Utilization Wall* .

I. INTRODUCTION

Mobiles are recently emerged as fast changing segment of computing platform. The GreenDroid mobile processor is 45nm multicore processor. Android mobile software stack is targeted by GreenDroid mobile processor. And this GreenDroid mobile processor can execute general purpose programs of mobile with very less energy compare to other energy efficient design. This is done by using conservation cores, which are automatically generated, energy reducing, specialized cores.

In this paper, we are attacking on technological problem, which is utilization wall [1]. The utilization wall means, due to power constraint at full frequency the percentage of transistor drops exponentially with each process generation. These directly affect the dark silicon area, which is silicon area of chip; to stay within power budget this area should remain passive. Currently, within 3W power budget only one percent of 32nm chip can switch at full frequency.

With each process generation, power budget increases exponentially, whereas dark silicon gets cheaper exponentially. In this paper, there are two key insights. First, compare to general purpose processor specialized logic can give 10X to 1000X better energy efficiency. Second, to find architectural techniques for dark silicon, this is cheap resource. Using architectural techniques dark silicon can be made more valuable resource and energy efficient. Our approach is to fill specialized cores in dark silicon area of chip. This saves energy in common applications. From code base specialized cores are automatically generated.

II. RELATED WORK

GreenDroid mostly related to specialized accelerators [2] [3]. Accelerators are another class of specialized cores, which are widely used in smart phones and other systems. In accelerator, performance or speed up of program is first goal while energy saving is second goal. While in GreenDroid energy saving is the primary goal and performance is second goal.

Accelerators generally depend upon structure of program for improvement of performance. Accelerators have properties which include moderate or high levels of parallelism, small number of lines of code and branch directions. Even using this structure accelerator requires human guidance.

TABLE I. CLASSICAL VS. LEAKAGE SCALING

Transistor property	Classical scaling	Leakage-limited scaling
ΔVt (threshold voltage)	1/S	1
ΔVDD (supply voltage)	1/S	1
Δ quantity	S^2	S^2
Δ frequency	S	S
Δ capacitance	1/S	1/S
Power	1	S^2
Utilization	1	1/ S^2

The transformation that parallelizing compiler performs same transformation is required to generate accelerators. Accelerator creation requires many years effort. Accelerators are limited in their applicability.

III. UNDERSTANDING THE UTILIZATION WALL

Utilization wall can be demonstrated in two ways. First, extend the CMOS scaling because of leakage limitation on threshold voltage scaling to limit on power scaling. Second with experimental result demonstrate utilization wall [4].

Table I shows with each process generation how transistor properties change. Here scaling factor is denoted by S. The column of classical scaling shows transistor properties changed before 2005. Before 2005 it was possible to scale supply voltage and threshold voltage together. The second column leakage limited scaling shows chip properties. This second column shows properties that we could no longer lower the supply voltage or threshold.

In case of classical scaling, threshold voltage and supply voltage drops by 1/S. While in case of leakage limited scaling, threshold voltage and supply voltage remains constant. In both cases, operating frequency increases by factor S, quantity of transistor increases by S^2 and capacitance drops by 1/S.

As Table I shows power and utilization of silicon resources remain constant in case of classical scaling. While power increases by S^2 and utilization of silicon resources drops by $1/S^2$ in case of leakage limited constant.

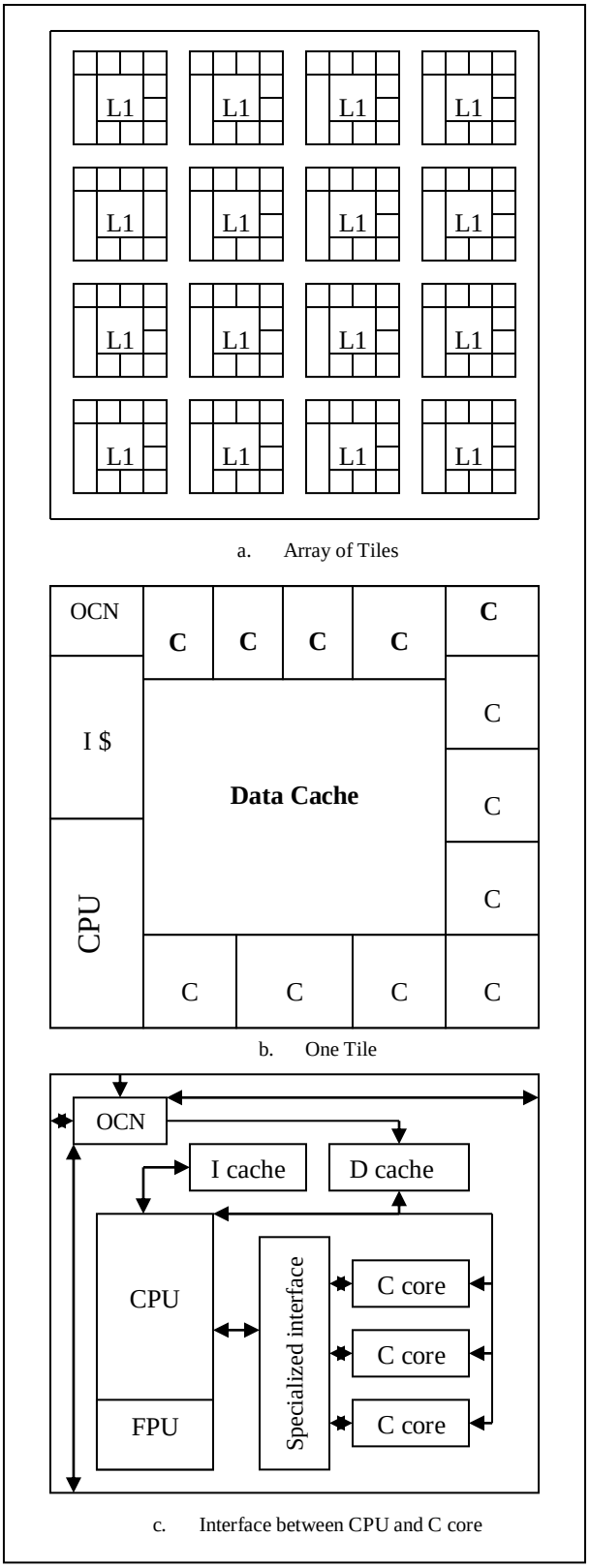
IV. UTILIZATION WALL

Poor CMOS scaling is dictated by utilization wall, improvement in performance of processor are determined by degree at which each process shrink to reduce switching energy of transistor on chip. Improvement in performance of processor does not depend on improvement transistor count or transistor frequency. Underlying energy efficiency is not improving as faster as transistor counts are growing. Due to this phenomenon of dark silicon occurs. Dark silicon is chip silicon area which should remain passive to stay in power budget.

Desktop processor industries decided to build multicore processors and stop scaling clock frequency because of dark silicon problem. With each generation power budget is increasing exponentially and dark silicon is becoming exponentially cheaper.

V. GREENDROID ARCHITECTURE

GreenDroid architecture uses conservation cores or c-cores [1] [3]. These c-cores are energy efficient and specialized processors used to execute frequently used portion of code. GreenDroid processor combines application specific processors which are energy efficient with general purpose processors.



a. Array of Tiles

b. One Tile

c. Interface between CPU and C core

Fig.1: GreenDroid architecture.

Fig.1 shows architecture of GreenDroid processor. GreenDroid consist of an array of tiles. Each tile has one CPU, data cache of 32 Kbytes and On Chip Network (OCN) which is point to point mesh connection [5]. The On Chip Network is used for synchronization and memory traffic. Each tile consists of 8 to 15 c-cores and each tile is unique. Specialized interface and data cache couples the c-cores with CPU. CPU can pass argument by specialized interface to c-cores. Hardware is reconfigured to change application code and context switching is also performed by CPU.

To create GreenDroid, hotspots region is determined [6]. The hotspot is portion of code where more time is spent by processor. These hot spots are automatically transformed to specialized circuit. The cold code is run on CPU; cold code means code that is not executed frequently. While hot code is handled by c-core. As code moves from hot to cold code or cold code to hot code execution also jump from c-core to CPU or CPU to c-core.

Energy saving of overall system is impacted by three things: first energy required for less efficient CPU to run cold code, second energy required in data cache and third energy required by clock and leakage. First problem is solved by executing code on c-cores. Second is resolved by using novel memory optimization. And third is solved by using clock power reducing technique and power gating.

A. Implementation

The CPU in each tile is 32 bits and has seven stage pipeline.

Each CPU has

- A multiplier.
- Floating Point Unit which is single précised.
- An instruction cache of 16 Kbytes.
- A data cache of 32 Kbytes.
- Translation Look-aside Buffer (TLB).

Cache access time set the target frequency of 1.5 GHz and which is frequency for 45nm design. Data cache is shared by CPU and c-cores. And multiplier and FPU are optionally shared by c-cores with CPU, depending on execution of code. To reduce consumption of energy most of tiles and c-cores are power gated.

B. Execution

At runtime, initially application runs on any general purpose CPU but as it enters in hot code region execution is automatically transferred to appropriate c-core. Depending upon c-core used and applications which are active currently execution moves from tile to tile. To reduce static power dissipation clock rating and power rating is used.

VI. TARGETTING THE ANDROID MOBILE

Android is developed by Google, which is open source mobile software stack. Android consist of set of application libraries, a Linux kernel and virtual machine. On the top of Dalvik virtual machine user applications run.

There are several reasons to Android to suit c-cores. First, Android has set applications which are used commonly such as media player, email and Web browser. Typically, virtual machine and application libraries have hot code. For Android application high coverage is achieved by using less number of c-cores which targets virtual machine and application libraries.

In experiment Android workloads include mail client, web browser, media player, map navigation and many applications. According to experiment 95 percent of Android mobile is covered using less number of static instructions that is about 43000 instructions. Out of 95 percent, 72 percent was of virtual machine or library code which was shared between two or more applications.

VII. CREATING CONSERVATION CORE

Each c-core consists of control state machine and data path which is directly derived from code it targets [7]. The structure of C code is mimicked by control components and data path. The data path consists of multiplexer, functional unit and registers. Multiplexers used to implement control decisions, to execute instructions functional units are used and register across clock cycle holds program values. State machine is implemented using control unit which mirrors the code of control flow graph (CFG). It tracks the outcomes of branch during each cycle to determine which hardware block will be active.

Communication occurs via shared data cache between CPU and c-cores. C-cores are constructed using memory interface for application with access pattern. As conventional accelerators can't extract memory parallelism so they can't applications. In absence of memory parallelism energy saving can be attainted by conservation cores.

C code is translated to state machine and hardware schematic. Internal complier represents sample code which correspond hardware. The Control Flow Graph (CFG) is identical to the c core's state machine. In data path to access memory in order to write array has store unit and to read array has load unit.

A. Synthesizing Conservation Cores

Greendroid processor will have different c cores which target different part of Android mobile. It is not possible to design each c core by hand. So we are building tool chain of C/CPP to verilog which convert arbitrary part of code into c core hardware [1].

Loops and key functions in target code are firstly identified by tool chain. Inlining functions and outlining loops are extracted by them. Internal representation of Dataflow graph and control flow graph are generated by complier. Then verilog code for data path and control unit is generated by complier. Function stubs are also generated by complier. To invoke hardware, function stubs can call applications instead of

original functions. Finally, description of c core is generated by complier.

Wider range of C construct can be targeted as c cores mainly focus on power consumption and saving energy instead of parallelism. To speedup applications with memory parallelism and irregular control accelerators struggle. Conservation core can significantly lower the power cost and save energy of such code.

VIII. CONCLUSION

Dark silicon problem becomes worse due to utilization wall in both mobile and desktop processors. We attack on dark silicon problem through GreenDroid and conservation cores. Due to conservation code dark silicon is converted to energy saving and reduce power budget. Conservation cores can be used in key regions of Android mobile using conservation cores to save energy, even in regions which are irregular to control. Dark silicon is a new area which will open new opportunities. About 91 percent of energy consumption of processor is reduced using conservation cores.

IX. ACKNOWLEDGMENT

I would like to thank all department staff of my college for their helpful suggestion and support.

X. REFERENCE

[1] G. Venkatesh, J. Sampson, N. Goulding, S. Garcia, V. Bryksin,J Lugo-Martinez, S. Swanson, and M. B. Taylor, \Conservation Cores: Reducing the Energy of Mature Computations", ASP-LOS, 2010.

[2] D.E. Shaw et al., ‘‘Anton: A Special-Purpose Machine for Molecular Dynamics Simulation,’’ Proc. 34th Ann. Int'l Symp. Computer Architecture (ISCA 07), IEEE CS Press, 2007.

[3] P. Coussy and A. Morawiec, High-Level Synthesis: from Algorithm to Digital Circuit, Springer, 2008.

[4] R. Dennard et al., ‘‘Design of Ion-Implanted MOSFET’s with Very Small Physical Dimensions,’’IEEE J. Solid-State Circuits, vol. 9, no. 5, 1974, pp. 256-268.

[5] J. Sampson, G. Venkatesh, N. Goulding-Hotta, S. Garcia, S. Swanson, and M. B. Taylor, \E_cient Complex Operators for Irregular Codes", HPCA, 2011.

[6] M. Taylor et al., ‘‘The Raw Microprocessor: A Computational Fabric for Software Circuits and General Purpose Programs,’’ IEEE Micro, vol. 22, no. 2, 2002, pp. 25-35.

[7] N. Goulding-Hotta, J. Sampson, G. Venkatesh, S. Garcia, J. Au-ricchio, J. Babb, M. B. Taylor, and S. Swanson, \GreenDroid: AMobile Application Processor: A Mobile Application Processor for a Future of Dark Future", HOTCHIPS 2010.

[8] G. Venkatesh, J. Sampson, N. Goulding-Hotta, S. Kota Venkata, M. B. Taylor, and S. Swanson. \QsCores: Trading Dark Silicon for Scalable Energy E_ciency with Quasi-Speci_c Cores", MI-CRO 2011.

Sonal Sarode recieved the BE degree in Electronics and telecommunication from Savitribai Phule University, Pune in 2008. She was working as lecturer in polytecnic college for 2.5 years. She is currently doing ME in VLSI and Embedded system from Savitribai Phule University,Pune.