Ultra Low Energy Microcontroller Architectures

Aditya Tandon, *at3g10@sotn.ac.uk, University of Southampton*

***Abstract*—The abstract goes here.**

I. Introduction

THIS report looks into low power and energy techniques for processor architectures

March 12, 2014

II. Asynchronous design

In a field of sensor networks asynchronous processors and microcontrollers have been gaining popularity as they lead to energy efficient designs. The basic principle is that these designs function without a global clock and hence reduce the number of unwanted switching activities in the circuit [1], [2]. To compensate for no clock, the designs usually employ extra hardware for a handshaking protocol [1]. To further conserve energy these designs are event driven [1], [2], [3]. In such a system, the controller is mostly in a state of sleep until it is asked to perform a computation by an event. After performing the task the controller goes back to a sleep state thereby minimizing its active energy [1], [2]. Research has shown that there is no necessary software overhead for these systems due to their event driven nature [1], [2]. This is because these microcontrollers are used for a set of pre-defined tasks and can be simplified. For example, interrupts can be processed as events and there is no overhead to handle concurrent tasks [1], [2]. Designs can be further simplified by employing an in-order design which reduces the amount of hardware and therefore the amount of energy [2], [4].

It can be observed that event driven architectures should have a minimal transition time from a sleep state to an active state. The SNAP/LE architecture addresses this concern by employing an event queue [1]. This resembles a FIFO handler and tasks are executed if there is an event token present in the queue. If there is a token, the appropriate event handler associated with the token is looked up and the task is executed. After executing the task the processor goes into a 'sleep' state if there is no token present [1]. The time taken by the processor to transition between an active and sleep state is the time taken for a token to go through the event queue [1]. The length of the queue can be optimized so that this process is in the order of tens of nanoseconds and therefore this procedure saves energy and is also efficient [1].

Typically these designs can be modularised and stress can be taken off the microcontroller by employing hardware accelerators [1], [2]. Hempstead et al. [2] used the microcontroller only for computational intensive tasks and a separate event processor was employed which was effectively a hard-coded state machine to handle events which required light computation. This reduced the active and leakage power of the main controller. Another type accelerator used was the Message Coprocessor which was responsible for forming and forwarding incoming messages from the radio unit. Timing operations commonly found in wireless applications can be handled by a Timer Coprocessor and therefore can lead to a simplistic, energy efficient implementation for the microcontroller at the expensive of some additional hardware [2]. An architectural implementation of such a system can be seen from Figure 1.

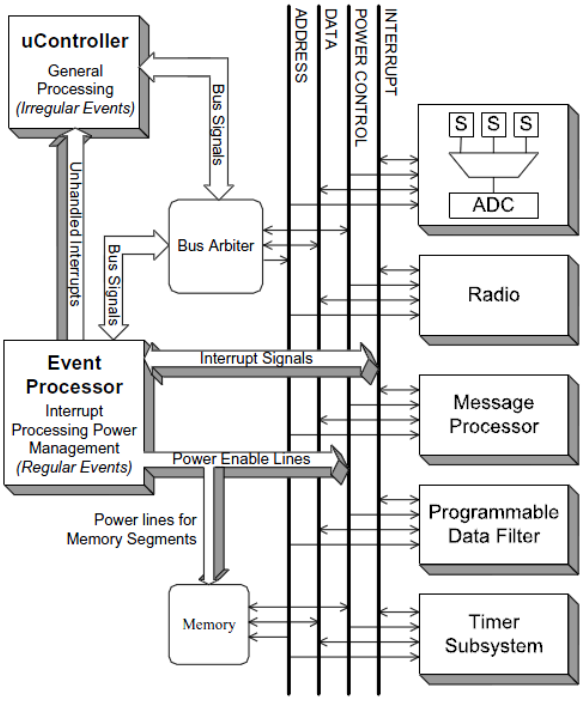

Fig. 1. Event driven design (reproduced from [2]).

The steps taken in desynchronization involve replacing the clock tree with local, asynchronous controllers and converting the flip-flops into latches [3]. This was demonstrated by Neechi et al. [3] where a synchronous AVR microcontroller was used as a template to create an asynchronous version. It was found that the asynchronous controller was about 5 times more energy efficient than the synchronous one [3]. The amount of energy that these prototype processors have taken to execute a particular instruction has been in the range of 10pJ 14pJ assuming an operating voltage of 1.2V and about 2.7pJ/instruction at 0.54V [1], [3], [4].

III. Low power design techniques

A. Low power memory

Over the years researchers have come up with different techniques to mitigate energy loss due to memory. A proposal for an adaptive cache for mobile processors could help reduce power [8]. The L2 caches on mobile phones have been found to have access patterns that are not correlated or balanced and therefore there is scope to dynamically adjusting the cache to match the application using it [8]. The compiler does an offline analysis of the application before run-time to determine parameters such as global average miss rates and access rates. Cache access is also monitored during run-time and the run-time information in conjunction with the offline material is used to enlarge or decrease the size of the cache dynamically depending on the need [8]. This proves useful as memory is used much more efficiently. Also, the leakage power is reduced as there are fewer idle cells. This technique was found to give a 13% - 29% reduction in power consumption (using benchmark programs) but there was a small trade-off for speed and area to incorporate this [8].

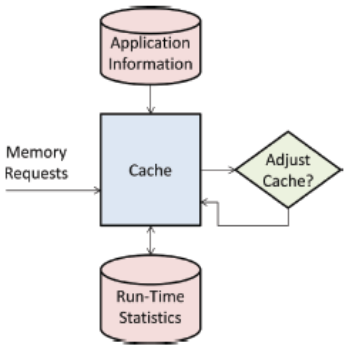

Fig. 2. Adaptive cache design (reproduced from [8]).

Another technique to reduce power is memory compression [5]. Here, a part of the data in volatile memory is compressed in order to reduce the number of logic elements that have to be self-refreshed when a device is turned into a low power state. Rest of the memory can be powered off and therefore the battery life is extended [5]. When there is a request to put a device into a low power state, the memory compression logic takes blocks from a designated memory, compresses it using a compression algorithm and then stores these blocks back to memory. A decompression procedure is followed when the device is in the active state [5]. The compression logic can be implemented in hardware or software and induces some overhead on the battery while performing compression but the power saved by compressing data outweighs this overhead [5].

Non-volatile memories are another area of interest for power reduction and even in performance enhancement [6]. Resistive Random Access Memory (RRAM) is a piece of memory that could be used as a substitute for SRAMs on mobile devices [6]. It was found that an RRAM with a crosspoint structure that uses a diode as a select cell reduces leakage current. This is because the resistance goes up as voltage decreases and the leakage current paths are cut-off [6]. This structure is also area efficient as multi-layered structures can be made [6]. Investigation is still going into this area as peripheral circuit design is harder if RRAMs are used but it could be used as a technique to reduce energy and power [6].

Many other techniques can be used to reduce power. An Intel processor for mobile devices reduces the leakage power in the L2 cache [7]. The data arrays in the cache continue to be in the sleep mode until a Hit signal is generated. Even when there is a hit, only the relevant data array is charged so that it can be activated while the other arrays continue to be in the low power mode [7]. This is a memory partitioning technique and is widely used to mitigate leakage power [2], [7].

B. Subthreshold libraries

- Minimum energy point analysis [9]
- Design of subthreshold libraries [9]

IV. Power management schemes

A. Power management

- Hierarchical power domains [10]
- Reduce sleep to active mode transition [11]
- Digitally adaptive Low Drop Oscillator to control current drive. Discrete load adaptive scheme [11]
- Intel Architecture processor for mobile internet devices [7]
- Power efficient algorithms [7]
- Different power states and power gating [7]

B. DVFS

- Choosing correct operating point: frequency and right number of cores [12]
- Temperature aware DVFS in mobile devices [13]
- General concept [3]

V. Other techniques

- Adaptive Body Bias: FBB to improve performance and RBB to reduce power. Only applied to parts of the chip [14]
- Heterogeneous multicores and understanding the connections between hardware and software [15] [16]
- Low power FPGA on mobile phones: power gating per tile, low leakage SRAM and other techniques to reduce power [17]

VI. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

REFERENCES

[1] V. Ekanayake, C. Kelly, IV, and R. Manohar, "An ultra low-power processor for sensor networks," *SIGARCH Comput. Archit. News*, vol. 32, no. 5, pp. 27–36, Oct. 2004. [Online]. Available: http://doi.acm.org/10.1145/1037947.1024397

[2] M. Hempstead, N. Tripathi, P. Mauro, G.-Y. Wei, and D. Brooks, "An ultra low power system architecture for sensor network applications," *SIGARCH Comput. Archit. News*, vol. 33, no. 2, pp. 208–219, May 2005. [Online]. Available: http://doi.acm.org/10.1145/1080695.1069988

[3] L. Necchi, L. Lavagno, D. Pandini, and L. Vanzago, "An ultra-low energy asynchronous processor for wireless sensor networks," in *Asynchronous Circuits and Systems, 2006. 12th IEEE International Symposium on*, March 2006, pp. 8 pp.–85.

[4] B. Warneke and K. Pister, "An ultra-low energy microcontroller for smart dust wireless sensor networks," in *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, Feb 2004, pp. 316–317 Vol.1.

[5] S. Balasundaram, "Increasing the battery life of a mobile computing system in a reduced power state through memory compression," Dec. 20 2007, uS Patent App. 11/450,214. [Online]. Available: http://www.google.com/patents/US20070291571

[6] P. Chiu, P. Lu, and X. Z, "Energy efficiency enhancement in mobile processor memory design using emerging nonvolatile memory," University of Berkley, Tech. Rep., 2013. [Online]. Available: http://www.eecs.berkeley.edu/ pfchiu/EE241_midtermReport.pdf

[7] G. Gerosa, S. Curtis, M. D'Addeo, B. Jiang, B. Kuttanna, F. Merchant, B. Patel, M. Taufique, and H. Samarchi, "A sub-2 w low power ia processor for mobile internet devices in 45 nm high-k metal gate cmos," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 1, pp. 73–82, Jan 2009.

[8] G. Bournoutian and A. Orailoglu, "Application-aware adaptive cache architecture for power-sensitive mobile processors," *ACM Trans. Embed. Comput. Syst.*, vol. 13, no. 3, pp. 41:1–41:26, Dec. 2013. [Online]. Available: http://doi.acm.org/10.1145/2539036.2539037

[9] A. Wang and A. Chandrakasan, "A 180-mv subthreshold fft processor using a minimum energy design methodology," *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 1, pp. 310–319, Jan 2005.

[10] T. Hattori and et al., "Hierarchical power distribution and power management scheme for a single chip mobile processor," in *Proceedings of the 43rd Annual Design Automation Conference*, ser. DAC '06. New York, NY, USA: ACM, 2006, pp. 292–295. [Online]. Available: http://doi.acm.org/10.1145/1146909.1146986

[11] M. Lueders, B. Eversmann, J. Gerber, K. Huber, R. Kuhn, M. Zwerg, D. Schmitt-Landsiedel, and R. Brederlow, "Architectural and circuit design techniques for power management of ultra-low-power mcu systems," pp. 1–1, 2013.

[12] A. Carroll and G. Heiser, "Mobile multicores: Use them or waste them," in *Proceedings of the Workshop on Power-Aware Computing and Systems*, ser. HotPower '13. New York, NY, USA: ACM, 2013, pp. 12:1–12:5. [Online]. Available: http://doi.acm.org/10.1145/2525526.2525850

[13] J. Kim, Y. Kim, and S. Chung, "Stabilizing cpu frequency and voltage for temperature-aware dvfs in mobile devices," pp. 1–1, 2013.

[14] G. Gammie and et al., "A 45nm 3.5g baseband-and-multimedia application processor using adaptive body-bias and ultra-low-power techniques," in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, Feb 2008, pp. 258–611.

[15] R. Ramirez, E. Rubio, and A. Viveros, "Energy consumption in mobile computing," in *Electronics, Communications and Computing (CONIELECOMP), 2013 International Conference on*, March 2013, pp. 132–137.

[16] C. H. K. van Berkel, "Multi-core for mobile phones," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '09. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2009, pp. 1260–1265. [Online]. Available: http://dl.acm.org/citation.cfm?id=1874620.1874924

[17] T. Tuan, S. Kao, A. Rahman, S. Das, and S. Trimberger, "A 90nm low-power fpga for battery-powered applications," in *Proceedings of the 2006 ACM/SIGDA 14th International Symposium on Field Programmable Gate Arrays*, ser. FPGA '06. New York, NY, USA: ACM, 2006, pp. 3–11. [Online]. Available: http://doi.acm.org/10.1145/1117201.1117203