ECE 570 Final Report

Low-Power Architectures and Design Techniques

Joe Crop, Mohsen Nasroullahi, and Robert Pawlowski

March 15, 2010

Contents

List of Figures

List of Tables

1 Introduction

This paper explores various techniques that are known to reduce power within a system. After the survey of low-power architecture techniques, two simulations were performed. First, a simulation on the pairing of sub-threshold operation with asynchronous logic. Second, a simulation on the pairing of sub-threshold operation and multi-core processors.

2 Current Work in Low-Power Architectures

2.1 Introduction

There have been many architectural-level methods to date that have been proposed to reduce the power of a design. Some options such as gray-coding provide a small power decrease, especially with respect to the overhead in the implemented hardware to achieve it. However, options such as power gating can provide a much more significant power decrease if implemented properly. The following Section will explore current techniques for lowering power within a digital architecture. With that said, there are two types of power in a system: dynamic and static. Dynamic power is the power consumed via bits switching within a system. Static power is consumed by the inherant leakage of a CMOS device.

2.2 Currently Known Low-Power Architecture Techniques

2.2.1 Partitioning of large data buses and wires

It has been shown that reducing the size and length of large data buses can reduce power in a system [3]. By reducing the length of wires and the amount of connection points to them the total capacitance of each link can be lowered. Because dynamic power takes on the form $P = \frac{1}{2} \times C \times V^2 \times f$, if the capacitance is lowered power scales proportionately.

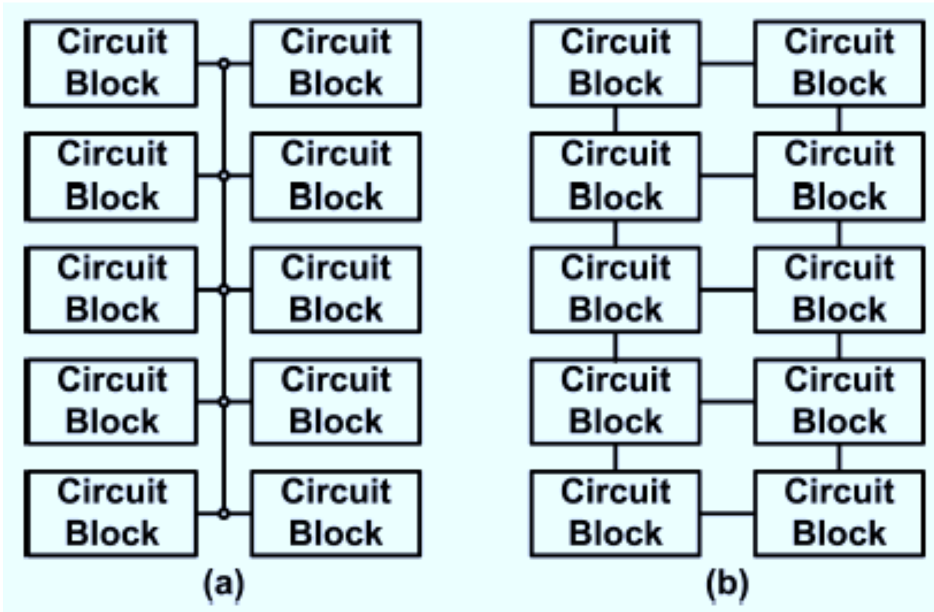


Figure 1 Illistration of bus paritioning of an example design [4]

2.2.2 Gray-Coding

The Gray-coding technique is a scheme in which the bits have been rearranged so their sequential increase will only cause a single bit transition. By reducing the bit switching, dynamic power will reduce significantly. This technique can only be used where you there is sequential bus access [4].

By utilizing Gray-coding the energy consumption due to switching can be reduced. The figure below shows the energy improvements simulated by [4] in 2009.

Gray	BCD	Decimal	Gray	BCD	Decimal
0000	0000	0	1100	1000	8
0001	0001	1	1101	1001	9
0011	0010	2	1111	1010	10
0010	0011	3	1110	1011	11
0110	0100	4	1010	1100	12
0111	0101	5	1011	1101	13
0101	0110	6	1001	1110	14
0100	0111	7	1000	1111	15

Figure 2 Table of conversions from binary to gray-code [4]

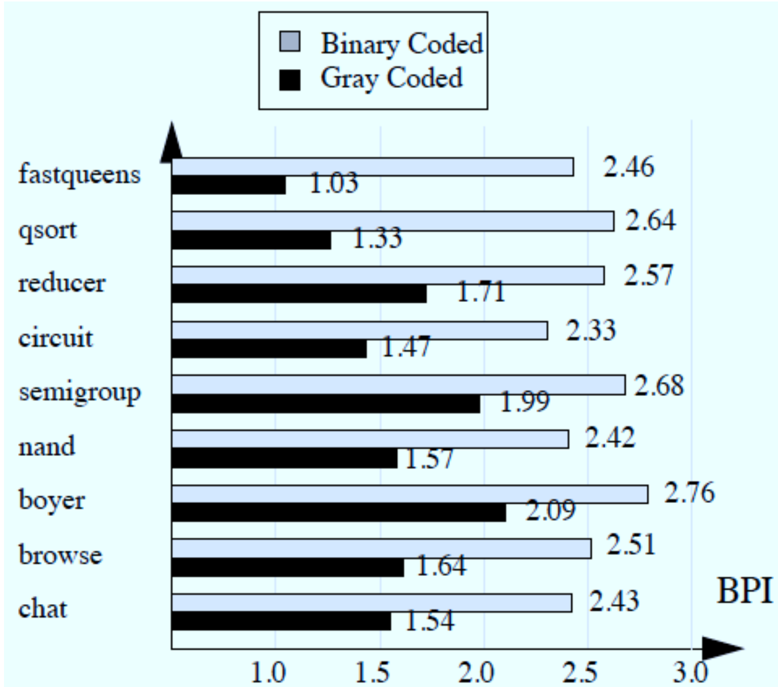


Figure 3 Illustration of energy savings when using Gray-coding vs. BCD [23]

2.2.3 Dynamic Voltage and Frequency Scaling (DVFS)

DVFS is the process of dynamically changing the supply voltage of a system or sub-block of a system depending upon it's work-load [8]. This can be beneficial when the demand of a system is dynamic and the system can be slowed down without lower the perceived quality of performance. In order to determine an appropriate supply voltage based on the needed clock frequency the following equation can be used:

$$delay \propto \frac{V}{(V-V_k)^a} \text{ and } f_{CLK} \propto \frac{(V-V_k)^a}{V}$$

In this equation V is the supply voltage and F_{CLK} is the clock frequency. a ranges from 1 to 2 and V_k depends on the velocity saturation [12].

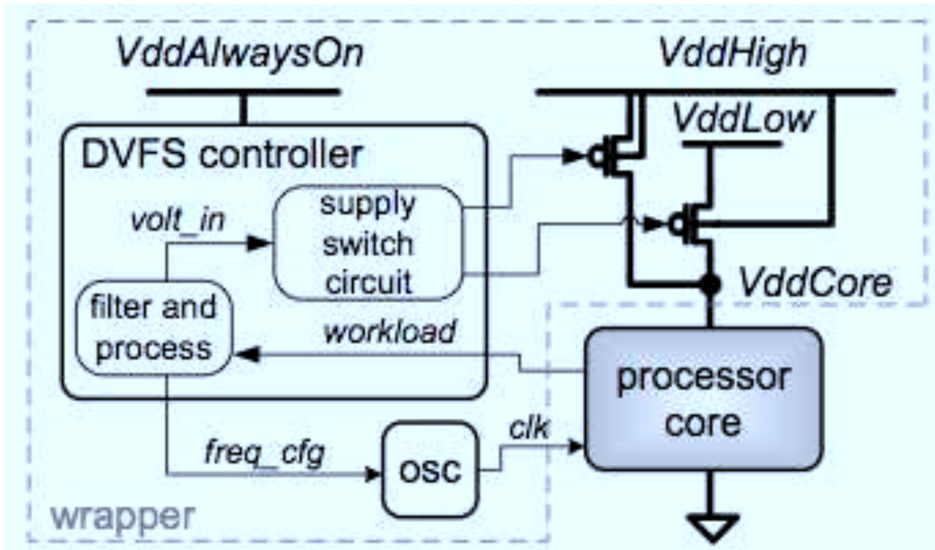


Figure 4 Example DVFS system used in [22]

2.2.4 Power Gating

Power Gating is the process of dynamically turning off blocks of a system when they are not used in order to save static power consumption [7]. This is usually achieved by placing a "sleep transistor" separating the supply node of a block with the supply of the entire system. The primary drawbacks of this method are increased area overhead, additional logic circuit to generate the control signals and possibly slower startup and shutdown times of the block leading to poorer performance.

For successful implementation at of the this technique on have to consider the following parameters:

- Power Gate Size: A choice of proper sized switch for footer or header transistor switches to handle the amount of switching current for avoiding the IR drop across the switches have to be taken into the consideration.

- Gate Control Slew Rate: Power gating efficiency determined by the slew rate of the switching transistors. Large slew rate will cause slower startup and hence degrade the performance. In order to reduce the slew rate buffering of the control signal have to be taken into consideration [18].

- Simultaneous Switching Capacitance: In order to avoid simultaneous current draw from power grid there is a limitation on how many switches can be turned on simultaneously.

- Power Gate Leakage: Since the switches are made of PMOS and NMOS, hence reducing the leakage of these transistors is crucial for energy reduction of the entire system.

To further reduce the power consumption techniques such as fine-grain power gating [16], and coarse-grain power gating [17] can be use.

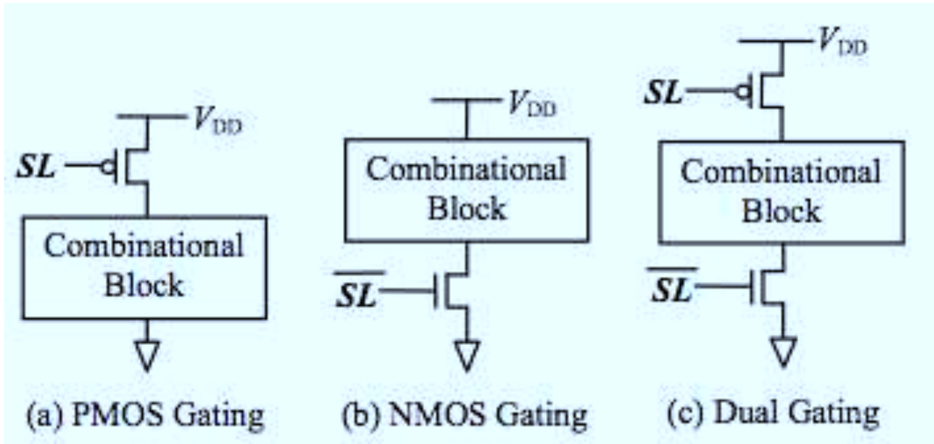


Figure 5 Illustartion of different power gating configurations [21]

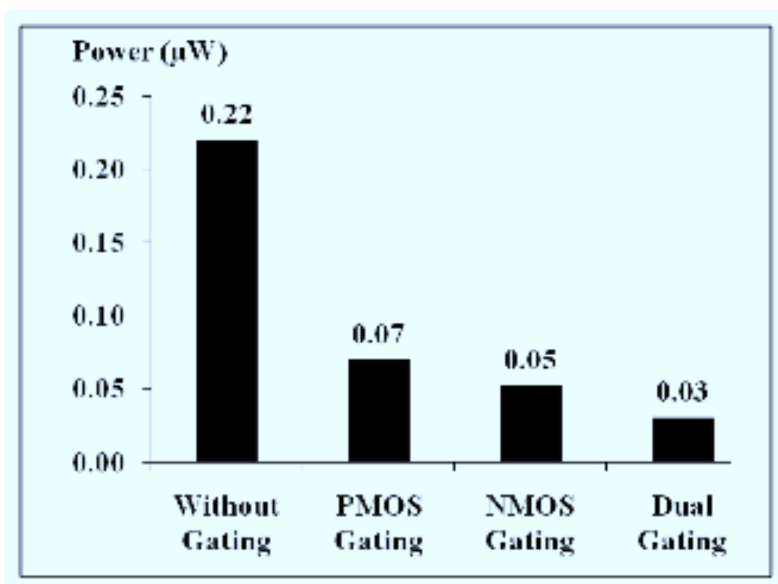


Figure 6 Graph illustrating static power savings while using various power gating techniques [21]

2.2.5 Clock Gating

In most of the current low power architectures the clock distribution tree contribute significantly to the total energy consumption of the processor. The majority of this power comes from the dynamic power dissipation at the latch nodes [19]. Clock Gating is similar to power gating in the respect that it is used to "turn off" the clock to a given block that is unused. This technique is widely used in processors today for reduction in clock power [12]. Clock gating can be implemented in Register Transfer Level (RTL) in three primary groups; system level, sequential logic, and combinational logic.

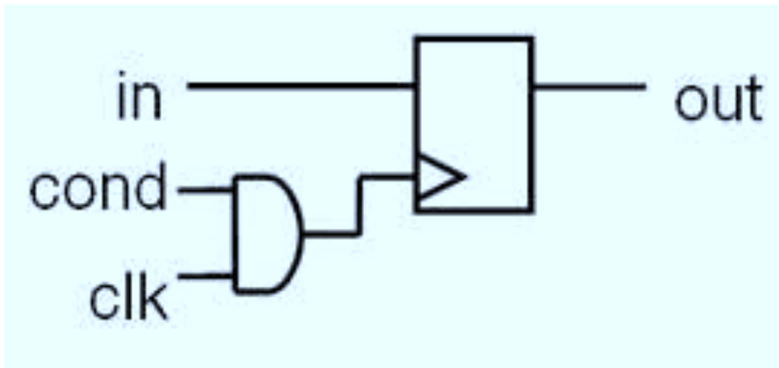


Figure 7 Simple example of clock gating logic

2.2.6 Cold Scheduling

Cold scheduling is a method that is used to reduce power by purposefully changing the order of instructions. The instructions are reordered in a way to reduce the amount of switching activity within the processor and functional units [2].

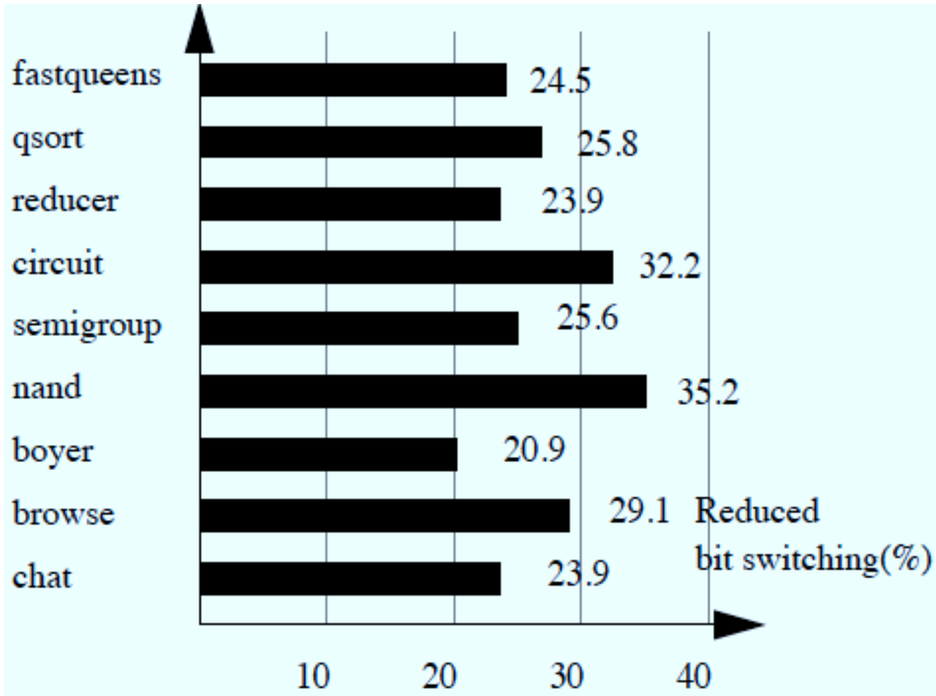


Figure 8 Graph of bit switching reduction when clod scheduling is used [23]

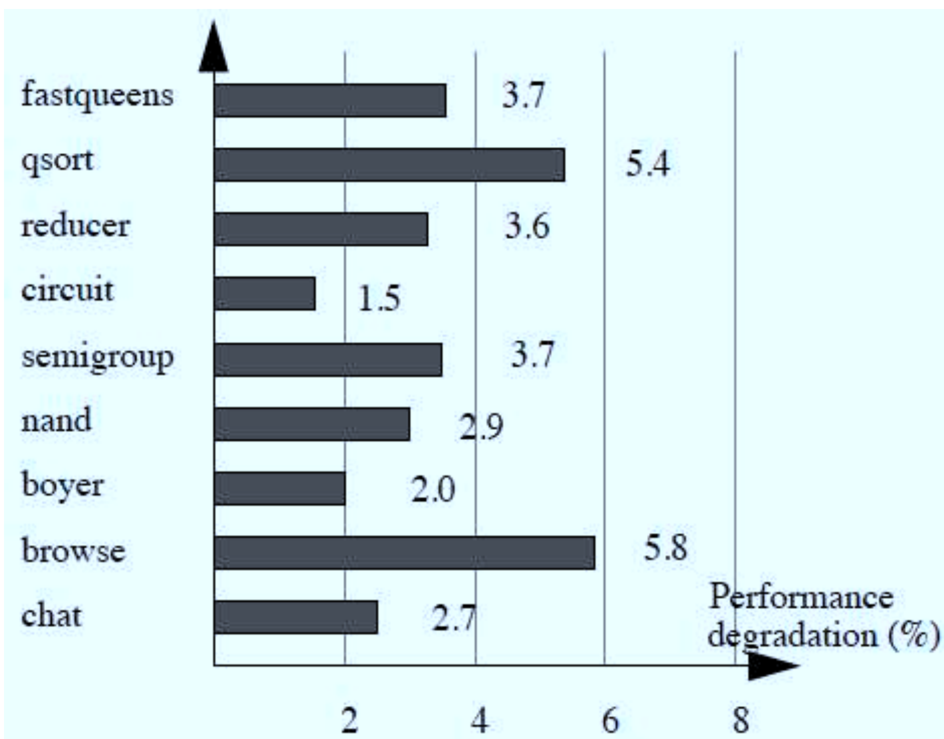


Figure 9 Graph of performance degradation when cold scheduling is used [23]

2.2.7 Sub-Threshold Operation

Sub-threshold operation is the process of simply reducing the supply voltage of a design. Drawbacks of this technique are: slower operating speed, decreased system robustness and chip-to-chip failure rate increase [5]. In addition, the low V_{DD} will degrade the I_{on}/I_{off} ratio which reduce robustness [20].

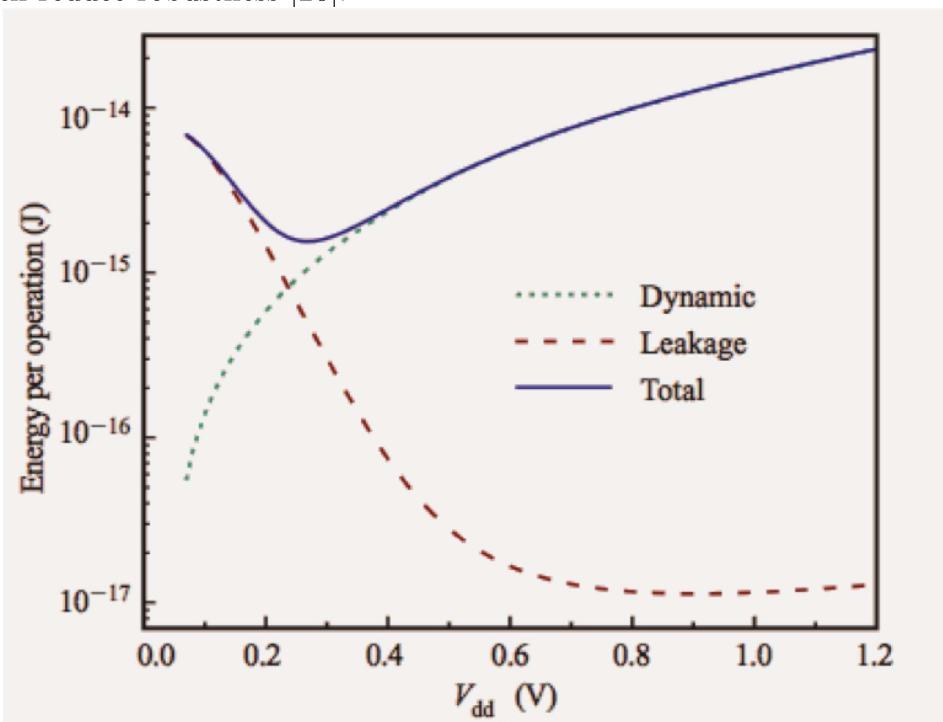


Figure 10 Leakage energy increases, dynamic energy decreases as the total energy falls to an optimal energy point [6]

3 Simulations

3.1 Sub-Threshold Multi-Core —MSP430

The primary goal of this simulation-driven portion of this project is to understand the possibilities of multi-core processing in the sub-threshold region of circuit operation. The question that is set out to answer is the following: **Does a sub-threshold multi-core processor with throughput M utilize less energy than a single-core super-threshold processor with throughput M?**

3.1.1 Existing OpenMSP430 Architecture

In order to evaluate the potential of sub-threshold multi-core processors the OpenMSP430 micro-controller core was used. The core is a fully functional MSP430 equivalent CPU that, for this test, has been synthesized in a 90nm process. The core is completely open source so any aspect of the hardware can be re-written in order to accommodate any modifications needed to alter the core for multi-core processing functionality.

3.1.2 Example Multi-Core Architecture

For the purposes of this simulation an example architecture has been designed to prove the concept. In order to keep things simple a shared-SRAM design has been implemented. All 4 cores in the figure below share an SRAM that is clocked 4 times faster than the speed of the CPUs. This allows for all cores to access the SRAM at non-overlapping time-duplexed slots.

The advantage of this architecture is that is takes advantage of the fact that a sub-threshold circuit cannot operate as fast as a super-threshold circuit. Therefore, there is no inherent performance degradation from the use of this architecture in this configuration.

A couple assumptions need to be made in order to unclutter the possible confusion of testing this architecture:

- The code being executed is perfectly parallelizable, and
- Throughput is proportional to clock speed (for example 1MHz -¿ 1MIPS).

With all the above in mind, two simulations have been carried out. The first simulation uses the proposed architecture and simulates 4 cores in near-sub-threshold. The second simulation runs the sub-threshold cores at their lowest voltage and compares the resultant MIPS/power.

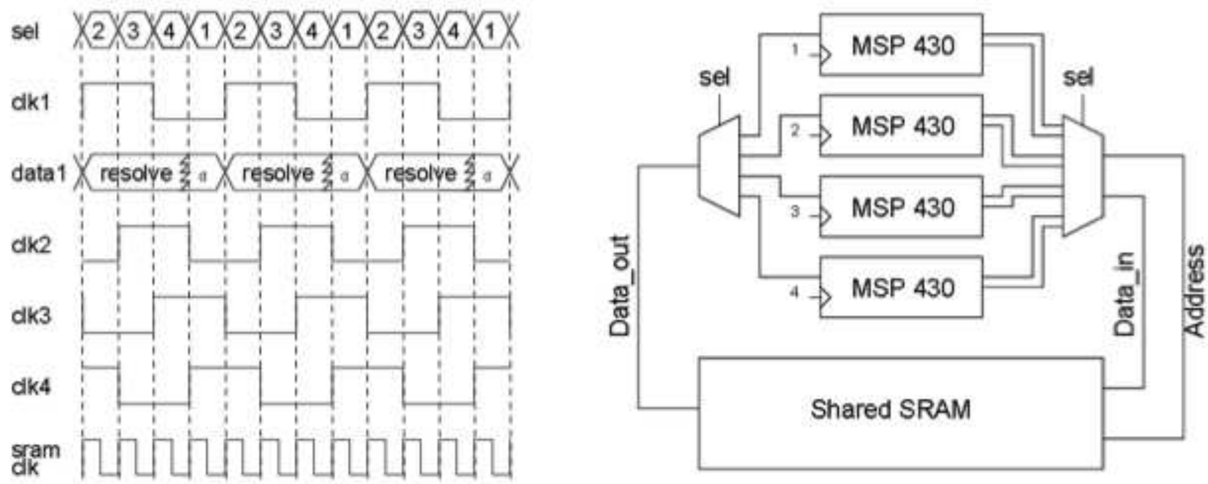


Figure 11 timing diagram along with proposed architecture of 4-core MSP430 process

3.1.3 Simulation Results - 4 cores

In the following simulation, a single core was run at nominal voltage (1.2V) at its maximum speed. That speed was found to be 500MHz. The core was then simulated at 500/4 = 125MHz where the supply voltage was lowered as fas as possible without destroying the functionality of the core. In this case the lowest supply voltage was found to be 0.84V. By finding the energy for each voltage/speed we can determine the possible power savings when the lower speed core is multiplied by 4.

The table below shows the power results from these simulations and gives some brief analysis. It is found that, in this case, using 4 cores in near sub-threshold does save power over 1 core in super-threshold.

	Nominal	Sub-threshold	4-core equivalent
Voltage	1.2V	0.84V	0.84V
Max Speed	500MHz	125MHz	125MHz x 4
Energy (J/comp)	$26.4E^{-12}$	$6.23E^{-12}$	$24.92E^{-12}$
Pros	Small Area	Low-Power	Energy Efficiency
Cons	High-Power	Slow	Large Area, Complexity

Table 1 Results of single-core system versus 4-core system in sub-threshold

3.1.4 Simulation Results - 152 cores

In the second set of simulations, the core's voltage was lowered to the optimal energy point. The energy/computation was drastically lowered. However, the throughput did not scale well putting the maximum frequency at 3.3MHz. This means that the sub-threshold core has to be duplicated 152 times. Not only is this impractical from a data access point of view (i.e. 152-access SRAM), but it turns out that by duplicating the core 152 times the energy/computation is drastically increased.

These results lead to an interesting conclusion. This shows that there must be an optimal point where the energy/throughput reaches a minimum. This point is most likely around 125MHz @ 0.84V in this case. More simulations will have to be carried out in order to find the aforementioned optimal point.

	Nominal	Sub-threshold	152-core equivalent
Voltage	1.2V	0.6V	0.6V
Max Speed	500MHz	3.3MHz	3.3MHz x 152
Energy (J/comp)	$26.4E^{-12}$	$3.24E^{-12}$	$492E^{-12}$
Pros	Small Area	Low-Power	None
Cons	High-Power	Slow, Less Reliable	Large Area, Complexity, IO!

Table 2 Results of single-core system versus 152-core system in sub-threshold

3.2 Sub-Threshold Asynchronous Logic —A Multiplier

In wireless sensor applications, power is a scarce commodity. Operating a circuit in the sub-threshold region [9] enables a designer to implement a circuits functionality with the minimum energy possible. Unfortunately, circuits operating in sub-threshold exhibit wide variations in delay as supply voltage is scaled [9], especially across process variations. Therefore, conventional synchronous timing schemes exhibit large delay spreads across transistor and process mismatches, resulting in impractical usage of sub-threshold circuits in deeply scaled CMOS technologies.

Though multipliers exist using either sub-threshold or asynchronous design methodologies [10-11], none have utilized a fusion of both techniques. In this paper we will first discuss our motivation for the combined design methodology. We will then describe our implementation, and finally, analyze our results.

The primary question asked here is: **Does the pairing of sub-threshold and asynchronous logic reduce the energy consumption over conventional sub-threshold logic design?**

3.2.1 Advantages of Asynchronous Logic

Asynchronous or self-timed circuits possess a number of properties that make them advantageous for sensor applications:

- They are event driven. They wait indefinitely, burning only leakage power until they are provided with an event. Then they wake to perform the desired computation.
- They relieve a designer from designing a high-fanout, time sensitive clock tree to every block of the design. Power is consumed only when a computation is performed, unlike clocked systems that constantly burn power even if they are not used.
- They are not forced to compute on unused data in globally clocked buses. By construction they provide fine grain clock gating.
- They can exploit the fact that the time required to compute a multiplication varies greatly depending on the operands. If the multiplier is zero, the done signal triggers immediately, allowing the start of the next stage of computation. This bypasses all of unnecessary steps that take place in a typical synchronous multiplier that computes an intermediary sum before then throwing it away.

The drawback to asynchronous circuits is the overhead of the control circuitry required to detect the completion of an intermediary computation step, as well as in preventing successive calculations from overwriting a result before it has been written back. In addition, there is design complexity in fully understanding asynchronous design. This paper will show that asynchronous design becomes more attractive in sub-threshold sensors.

3.2.2 Asynchronous Communication Overview

Asynchronous implementation for the multiplier uses "handshakes" to communicate between the different block of the multiplier. Figure 12 illustrates the basic control between the separate stages of the multiplier. Each logic stage of the multiplier has its own control block that will communicate when each stage is ready to receive data, or is done with their computation. As the figure shows, there are data signals that pass between each of the logic blocks and control signals that pass between each of the control blocks. The following steps are the standard communication procedure between three stages of the multiplier.

1. The second logic block is free to receive data, and its control block lets the first stages' control block know that it is ready.
2. The first logic block sends data to the second stage. It also sends a control signal letting it know that it is sending data.
3. The second logic block performs its computation. At this time the second stage control block lets the adjacent stages know that it is busy.
4. The second logic block has finished its computation. Its control block lets the third stage know it is finished while also telling the first stage that it is ready to receive new data.
5. This procedure repeats for each computation.

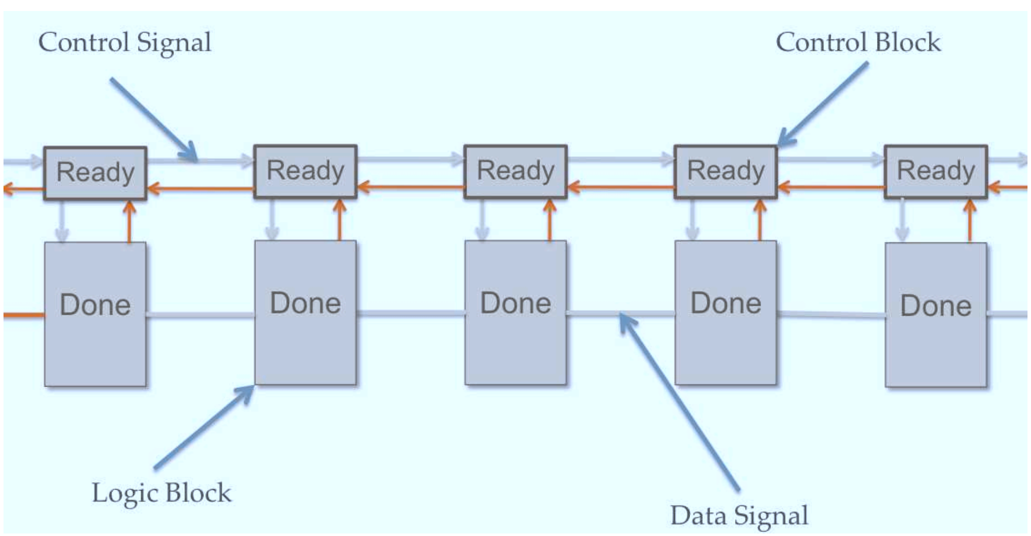


Figure 12 Illustration of asynchronous communication scheme

3.2.3 Advantages of Sub-Threshold Operation

A 3-5X improvement in power consumption can be expected for computation using sub-threshold operation [1], [2]. The biggest drawback to sub-threshold design is unpredictable delay, caused by variation in transistor current. In [1], the deleterious effect of lowering the supply voltage on clock frequency variation is shown, where the $3\sigma/\mu$ clock frequency variation can vary as much as 85% as shown in the figure. In a synchronous system, this poses a significant problem. While in synchronous systems, the slowest sub-block determines the maximum clock frequency, asynchronous systems are tolerant of the maximum delay path and can therefore achieve higher throughput.

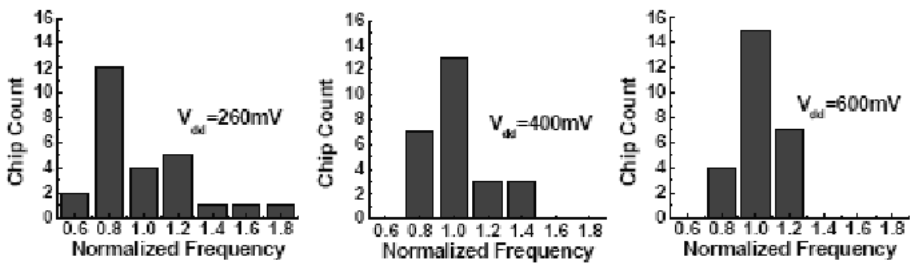
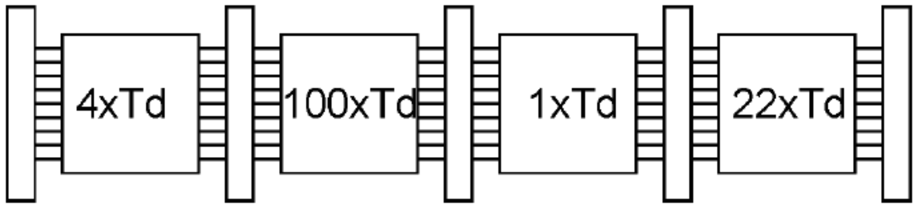


Figure 13 System Frequency Variation Due to Sub-Threshold Operation in Synchronous Circuit [9]

3.2.4 Combined Advantages

To illustrate the primary advantage of pairing sub-threshold operation and asynchronicity an example of a simple pipeline is shown in the figure below. Because of process variation coupled with the effects of running in the sub-threshold region each pipeline stages delay is widely varied. For a typical synchronous system the total pipeline delay is the maximum delay multiplied by each stage because all stages share the same clock. If an asynchronous system is used, the overall pipeline delay will only be the sum delay from all stages. In this particular example the speedup ends up being well over 3X.



$$SyncPipelineDelay = (100 \times 4) \times T_d = 400 \times T_d$$

$$AsyncPipelineDelay = (4 + 100 + 1 + 22) \times T_d = 127 \times T_d$$

Figure 14 Example Pipeline with Variable Block Delays

4 Conclusions

As sown by this report, there are many different low-power architecture design techniques. Partitioning of large data busses and wires, gray-coding, DVFS, power gating, clock gating, cold scheduling, and sub-threshold operation were surveyed. It is important to note that bus partitioning, dynamic frequency scaling, cold scheduling, gray-coding, and clock gating are good for dynamic power redection and dynamic voltage scaling, and power gating are good for static power reduction. Sub-threshold Asynchronous design techniques along with sub-threshold multi-core techniques were porposed and simulated. This work showed that more robust, power-optimal performance can be achieved by using asynchronous logic in the sub-thershold domain with the drawback of increased area. Furthermore, it was found that simple multi-core architectures, while run in sub-threshold can maintain throughput while decreacing the overall power of a processor. These two techniques show a promising future in low-power architectures.

5 References

[1] Paul R. Gray, Paul J. Hurst, Stephen H. Lewis, Robert G. Meyer, "Analysis And Design Of Analog Integrated Circuits," 5Th Ed, Wiley, New York, 2009.

[2] C. Su, C. Tsui, A. Despain, "Low power architecture design and compilation techniques for high performance processors," Proceedings of the IEEE COMPCON, February 1994.

[3] G. Blair, "Designing low-power digital CMOS," Electronics & Communication Engineering Journal, vol. 6, no. 5, pp. 229-236, Oct. 1994.

[4] S. Meliza "Ultra low energy digital logic controller design for wireless sensor networks," M.S. dissertation, Oregon State University, Corvallis Oregon, 2009.

[5] 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, 2005.

[6] S. Hanson, et al., "Ultralow-voltage minimum-energy CMOS," IBM Journal of Research and Development, Vol. 50, pp. 469-90, 2006.

[7] T. Sakurai, "Low power digital circuit design," in Proc. of the 30th European Solid-State Circuits Conf., Sep. 2004, pp. 11-18.

[8] H. Soeleman and K. Roy, "Ultra-low power digital subthreshold logic circuits," in IEEE International Symposium on Low Power Electronics and Design, 1999, pp.94-96.

[9] B. Zhai, et al., "A 2.60pJ/Inst. Subthreshold Sensor Processor for Optimal Energy Effi- ciency," IEEE Symposium on VLSI Circuits (VLSI-Symp), June 2006.

[10] B. Gwee, et al, "A Low-Voltage Micropower Asynchronous Multiplier With ShiftAdd Multiplication Approach," IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 57, No. 7, pp. 1349-1359, July 2009.

[11] C. Singh, et al., "A 4-bit Subthreshold MIPS Processor for Ultra Low Power Applications," [Online document] April. 2008 [2010 Jan 30], Available at HTTP: http://users.ecel.ufl.edu/ csingh/MIPS.pdf, April 2008.

[12] W. Chedid, C. Yu, and B. Lee, "Power Analysis and Optimization Techniques for Energy Efficient Computer Systems," Advances in Computers, Vol. 63, 2005.

[13] William Dally et al. "Stream Processors: Programmability with Efficiency" ACM Queue, March 2004, pp. 52-62.

[14] Kapasi, U. J., Rixner, S., Dally, W. J., Khailany, B., Ahn, J. H., Mattson, P., and Owens, J. D. 2003. "Programmable Stream Processors." Computer 36, 8 (Aug. 2003), 54-62.

[15] Khailany, B., Dally, W. J., Rixner, S., Kapasi, U. J., Owens, J. D., and Towles, B. 2003. "Exploring the VLSI Scalability of Stream Processors." In Proceedings of the 9th international Symposium on High-Performance Computer Architecture(February 08 - 12, 2003). HPCA. IEEE Computer Society, Washington, DC, 153.

[16] Tong Lin; Kwen-Siong Chong; Bah-Hwee Gwee; Chang, J.S.; , "Fine-grained power gating for leakage and short-circuit power reduction by using asynchronous-logic," Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on , vol., no., pp.3162-3165, 24-27 May 2009.

[17] Nair, P.S.; Koppa, S.; John, E.B.; , "A comparative analysis of coarse-grain and fine-grain power gating for FPGA lookup tables," Circuits and Systems, 2009. MWSCAS '09. 52nd IEEE International Midwest Symposium on , vol., no., pp.507-510, 2-5 Aug. 2009.

[18] Ming-Dou Ker; Tzu-Ming Wang; Fang-Ling Hu; , "Design on mixed-voltage I/O buffers with slew-rate control in low-voltage CMOS process," Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on , vol., no., pp.1047-1050, Aug. 31 2008-Sept. 3 2008

[19] Sulaiman, D.R.; , "Using clock gating technique for energy reduction in portable computers," Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on , vol., no., pp.839-842, 13-15 May 2008.

[20] Calhoun, B.H.; Wang, A.; Verma, N.; Chandrakasan, A.; , "Sub-Threshold Design: The Challenges of Minimizing Circuit Energy," Low Power Electronics and Design, 2006. ISLPED'06. Proceedings of the 2006 International Symposium on , vol., no., pp.366-368, 4-6 Oct. 2006.

[21] Tong Lin; Kwen-Siong Chong; Bah-Hwee Gwee; Chang, J.S.; , "Fine-grained power gating for leakage and short-circuit power reduction by using asynchronous-logic," Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on , vol., no., pp.3162-3165, 24-27 May 2009.

[22] Cheng, W.H.; Baas, B.M.; , "Dynamic voltage and frequency scaling circuits with two supply voltages," Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on , vol., no., pp.1236-1239, 18-21 May 2008.

23 C. L. Su, C. Y. Tsui, and A. M. Despain. "Low power architecture design and compilation techniques for high-performance processors." IEEE COMPCON, Feb. 1994.

[24] J. Howard, S. Dighe, Y. Hoskote, et al., "A 48-Core IA-32 Message-Passing Processor with DVFS in 45nm CMOS", ISSCC 2010 Digest of Technical Papers, pp. 108-109, Feb., 2010.