# *Asynchronous Sub-Threshold Ultra-Low Power Processor*

Ron Diamant, Ran Ginosar
Dept. of Electrical Engineering, Technion IIT
Haifa, Israel
rond@tx.technion.ac.il, ran@ee.technino.ac.il

Christos Sotiriou
Dept. of Electrical Engineering, University of Thessaly
Thessaly, Greece
csotiriou@gmail.com

***Abstract - Ultra low power VLSI circuits may enable applications such as medical implants, sensor networks and "things" for IoT. Aggressive supply voltage scaling is known to significantly improve power consumption and efficiency, but incurs both performance degradation and high delay variations. We illustrate that the most energy efficient operating point of a pipelined MIPS CPU lies in the deep sub-threshold region. We investigate the optimal selection of technology node, process variant and transistor type, and compare synchronous and asynchronous designs.***

***We identify the optimal performance/power ratio design point for the 28nm high-k metal-gate high-performance process with high VT transistors and a bundled-data asynchronous design style to efficiently accommodate delay variations. We illustrate a 7.4× power efficiency improvement potential for the CPU, coupled with a reduction in power consumption by more than one thousand, relative to a synchronous CPU operating at nominal voltage. The asynchronous sub-threshold MIPS CPU designed in this work is compared with other commercial and research CPUs, and is shown to achieve superior power efficiency.***

## I. Introduction

Ultra-Low Power (ULP) processors dissipate extremely low power, *e.g.*, 1 micro-Watt. They are useful when powered by energy scavenging or by extremely small batteries, such as in implanted devices.

Typical use areas of ULP processors include implanted medical devices, such as cardiac pacemakers, which are presently powered by implanted batteries and may harvest their energy from the human body in the future, wireless sensor networks, in which hundreds or thousands of sensors are deployed to remote location without capability of battery replacement and wearable devices which attempt to achieve long battery lifetimes, such as smart wrist watches where the user wishes to attend to the battery very rarely. This paper focuses mainly on ULP processors for medical implanted devices.

This paper is organized as follows. Section II describes related work, and its application to this paper. Section III describes the selection of process node and variant, and the selection of the most suitable transistor type for ultra-low power circuits. Section IV describes the design and benchmark of a synchronous and asynchronous processor, and the comparison between them. Section V compares the processor developed in this work to published and commercial results of other ultra-low-power processors. Section II summarizes the work, and suggests possibilities for further research

## II. Related Work

Power efficiency techniques in VLSI circuits are discussed in [1][2] and sub-threshold concepts are formulated in [3]. Sub-threshold circuits were demonstrated in several studies, *e.g.*, [4][5][6][7].

Process scaling is discussed in [8] and also in [9][10] for circuits operating in sub-threshold region. Delay variations incurred by operating digital circuits in sub-threshold region were studied in [11].

Asynchronous VLSI design [12][13] is employed in this paper as a means of dealing with delay variations made worse by sub-threshold operation. In particular, bundled-data asynchronous design [14][15] is applied using a de-synchronization flow [16][17]. Other asynchronous design techniques, such as Quasi-Delay-Insensitive [5][18][19] and Carrier-Sense Completion-Detection [20], have also been proposed for mitigating delay variations.

## III. Technology Selection

When considering CMOS fabrication technology for ultra low power use, three factors should be studied: process node (indicated as feature size), process variant (high performance or low power) and transistor type (high/standard/low threshold voltage).

### *A. Process Node Selection*

Dynamic power is expected to decrease with scaling. For scaling factor α, gate capacitance and (with Dennard scaling in effect) supply voltage both decrease by α and consequently dynamic energy $C_g V_{DD}^2$ is reduced by $\alpha^3$ [8]. However, in deep sub-micron technologies, actual scaling proceeds slower than that. Because of increasing sub-threshold leakage, threshold voltage is kept higher than indicated by scaling and hence supply voltage cannot be reduced by α [8]. Still, newer process nodes facilitate lower dynamic energy per operation. Leakage, on the other hand, typically increases with technology. We seek the process node that provides minimal total energy, in presence of this contrast.

Aiming at ultra-low power operation, [9] compared minimum energy per operation of a benchmark multiplier

9781467394192/15/$31.00 ©2015 IEEE

circuit for process nodes from 0.25μm down to 32nm, and concluded that technology scaling was desirable due to lower minimum energy per operation. However, [9] reported that the more advanced technology nodes (45, 32nm) suffered of excessive leakage power.

To compare different process nodes, we have SPICE simulated a FO4 inverter (Figure 1) over several process nodes (using PTM models [21]) and the full range of supply voltage from nominal downwards. FO4 inverter measurements were shown to correlate well to full circuit measurements across different process nodes [22]. Performance-to-power ratios are drawn in Figure 2. The FO4 inverter has been simulated using both minimum size and 4× size transistors, yielding essentially the same results.

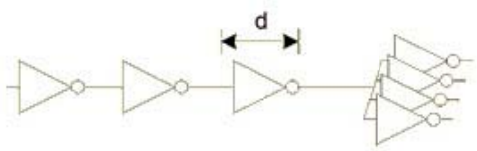


**Figure 1: FO4 test circuit, used for process selection**

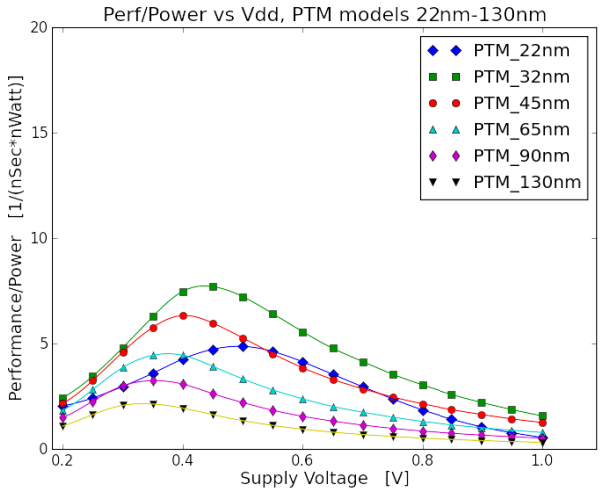


**Figure 2: FO4 performance/power ratio as a function of supply voltage for several PTM technology nodes (higher is better)**

Figure 2 demonstrates that technology scaling is beneficial in terms of optimal performance/power down to 32nm. However, the 22nm PTM node demonstrates a deteriorated performance/power. Similar results had been shown in [9].

The same experiment was repeated with PTM models that employ High-κ / Metal-Gate (HKMG) in the 45, 32 and 22nm process nodes (Figure 3). Clearly, the most advanced process node (22nm) achieves the highest performance/power ratio. Similar conclusions were reported in [10], which concluded that low-power variants such as Fully-Depleted Silicon-on-Insulator and High-κ / Metal-Gate (commonplace in contemporary nano-scale technology nodes) managed to decrease leakage power by one to two orders of magnitude, making such variants preferred candidates for ultra-low power circuits. It is likely that newer process nodes, with FinFET transistors, will produce even better results, due to their improved speed and lower leakage consumption.

In following sections, we select an actual high-K metal gate 28nm HP process node and employ its fab models in our simulations. That specific process node has been selected because we have had access to the technology files and transistor models for this process node.

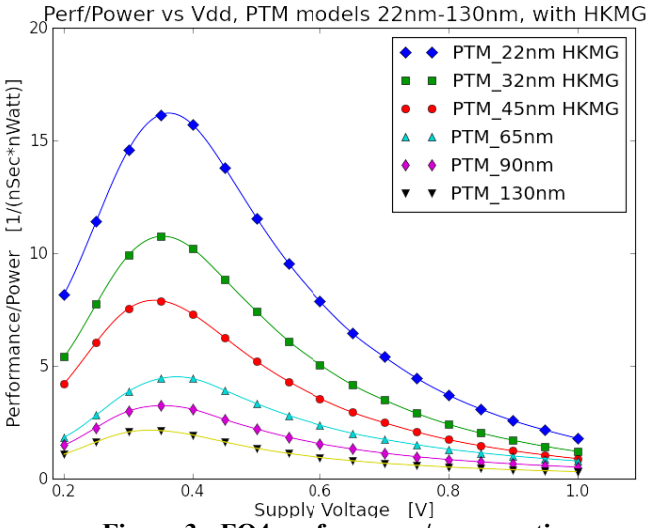


**Figure 3: FO4 performance/power ratio in 22,32,45nm High-κ / Metal-Gate PTM technologies**

*B. Process Variant Selection*

Deep sub-micron process nodes typically include at least two variants, typically including LP and HP, tuned to deliver low power and high performance, respectively.

We repeated the FO4 experiment for the 22nm PTM HKMG process node, comparing two different process variants, LP and HP. The results suggest that while LP is more power efficient than HP in nominal voltage operation, HP becomes more efficient for sub-threshold operation (Figure 4). Further research is required on these findings.

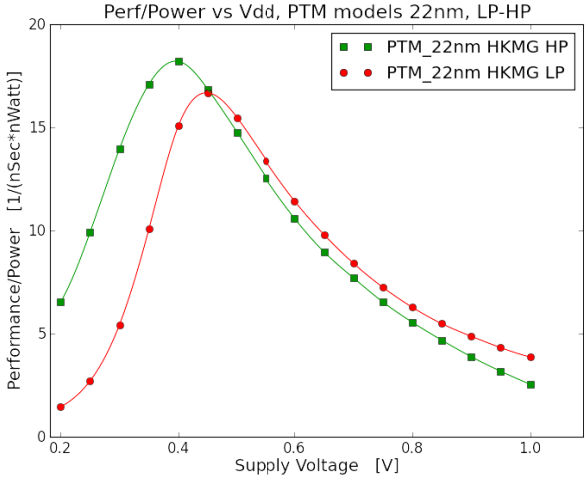


**Figure 4: FO4 performance/power ratio in 22nm LP and HP PTM technology variants**

### C. *Transistor Type Selection*

After selecting the process node and variant, one also needs to select which transistor type(s) to use. Typically, three transistor types are available - high, standard and low $V_T$, named HVT, SVT and LVT, respectively.

HVT transistors are expected to achieve superior performance/power in sub-threshold regime, due to their improved leakage power control.

We have repeated the FO4 experiment with the three transistor types mentioned above (Figure 5). As expected, HVT transistors demonstrated the best performance/power ratio, 1.89 times better than LVT and 1.32 times higher than SVT.

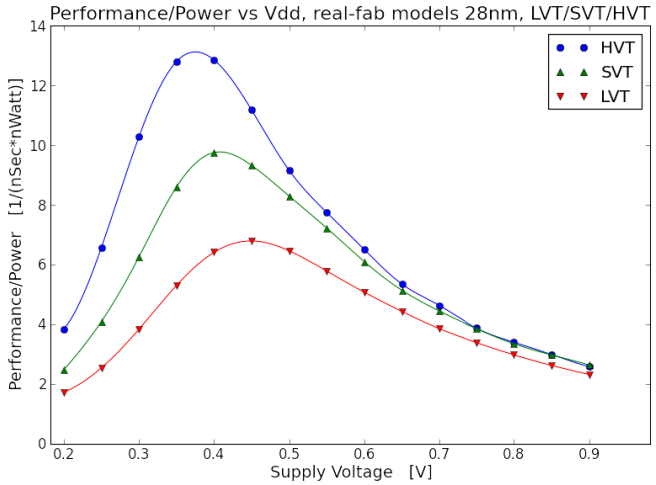


**Figure 5: FO4 performance/power ratio in HVT, SVT and LVT fab-model transistors**

## IV. Design Style Selection

Comparing different design styles is more challenging than comparing technology nodes and variants, partly due to lack of tools for some of the design styles. We avoided hand-crafted circuit design; rather, for each candidate design style, we defined the benchmark circuit in behavioral RTL, and applied standard EDA tools for logic synthesis and (where necessary) clock tree synthesis. The synthesized circuits were then compared using low-level SPICE simulations, to estimate performance and power in the most reliable manner. Results are mathematically adjusted to accommodate process variations, caused mainly by random dopant fluctuations, based on [11]. Averaging effects are also taken into account, due to transistor stacking and logic depth [22]. Mathematical adjustment was preferred over Monte Carlo SPICE simulations due to the long simulation times of the benchmark CPU circuit. Delay variations are computed as follows:

- Confidence interval values of the log-normally distributed sub-threshold propagation delay, for each $V_{DD}$ value, are adopted from [11] (we assume the same confidence interval values as [11], due to the lack of any other data)
- The standard deviation of the underlying normally distributed threshold voltage is extracted mathematically using the $3\sigma$ confidence interval formula of [11] $CI = e^{3\sigma} - e^{-3\sigma}$.
- The measured gate-delay is treated as the mean value of a log-normal distribution (*i.e.*, $T_{PD} = e^{\mu+0.5\sigma^2}$, where both $\mu$ and $\sigma$ relate to the underlying normal distribution).
- Similarly, the gate delay standard deviation is $\sqrt{e^{\sigma^2}-1} \cdot e^{\mu+0.5\sigma^2}$, which can be calculated based on $T_{PD}$ and $CI$.
- As a first order approximation, we assume the gates in the critical path are independent and identically distributed, and thus a $k$-gates critical path delay is distributed normally (central limit theorem), with standard deviation of $\sqrt{k} \cdot \left(\sqrt{e^{\sigma^2}-1}\right) \cdot e^{\mu+0.5\sigma^2}$.

### A. *Benchmark Circuit Selection*

A simple processor was selected as the benchmark circuit. The processor was sufficiently simple so that we could fully define its behavioral RTL for each of the alternative design styles, and we could fully simulate it using SPICE in reasonably short time. Two key design choices were made, the ISA and the level of pipelining.

#### 1) *Instruction Set Architecture (ISA)*

CISC architecture allows instructions of varying width, enabling dense code and saving power on instruction fetching. However, decoding and control may consume higher energy per operation. While RISC architecture typically incurs lower code density (incurring higher power for instruction fetching), the simplicity of instruction decoding enables simpler, smaller and faster instruction decoding, requiring less power. As suggested by [24], we employ RISC architecture with a small number of instructions, combined with data and instruction width of only eight bits, which simplifies the design and increases code density In addition, this architecture can be easily adapted and optimized to the application, and it is easier to design, synthesize and analyze it with SPICE.

#### 2) *Pipelining*

We follow the simplified MIPS architecture of [5], where three micro-architectures are proposed: single-cycle, multi-cycle and pipelined. Fine-grained pipelining may not be power efficient, due to the leakage consumed by pipeline registers, and due to a shorter critical path, resulting in more significant effect of delay variations (which are averaged over a small number of gates). For these reasons, it would appear that single-cycle architecture should be used: there are no leaking pipeline registers and the longer critical path helps average delay variations over more gates. However, performance of single-cycle processors is typically much lower than of pipelined ones, resulting in deteriorated power efficiency. We selected a coarse-grained 5-stage pipelined RISC (Figure 6), because leakage overhead was negligible in our very small circuit and performance was enhanced, improving the performance/power ratio. To simplify this study, the data and instruction memories are implemented using flip-flops rather than SRAM.

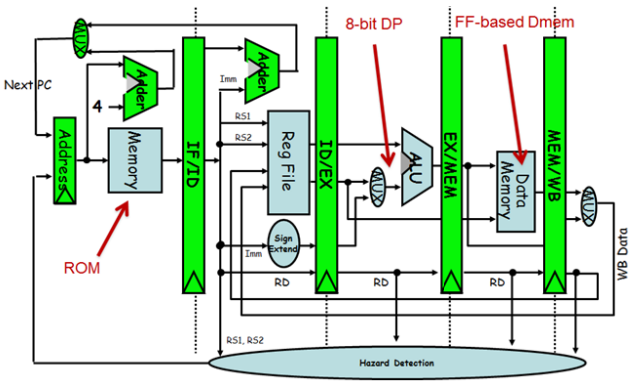


**Figure 6: Synchronous 8-bit datapath, 5-stage pipelined MIPS processor, used as a benchmark circuit.**

*3) Benchmark Program*

Performance and power were measured by SPICE simulations of the execution of a simple benchmark program, containing ALU, branch and memory access operations (Figure 7). In addition, the benchmark program helps identify the maximum operating frequency per each voltage level, by iterations, as follows. The program intentionally generates several pipeline hazards, since the hazard detection logic is known to be part of the CPU's critical path, and thus should lead to logic errors when the clock frequency is set too high for the given supply voltage. Such logic errors are easily detected and indicate repeating the simulation at lower frequency. The same benchmark program is used to measure performance and power for all compared designs, in order to assure fair comparison. It is noted that in general, the critical path might change with the scaling of $V_{DD}$ , which requires the benchmark program to test various different paths, and not only focus on the critical path that was identified using STA tools in nominal voltage.

```
LOOP1:  ADDI    R1,R0,8
        ADDI    R2,R1,8
        ADDI    R3,R2,8
        ADD     R4,R2,R3
        STORE   R4,R1,2
        LOAD    R5,R1,2
        SUB     R6,R4,R5
        BZ      R6, LOOP1
        ADDI    R7,R7,1
```

**Figure 7: Benchmark Program**

*B. Synchronous Design*

The synchronous processor was described in behavioral Verilog RTL, synthesized using Synopsys DC and a commercial 28nm HP standard cell library. The clock tree was manually created and inserted into the synthesized netlist, and subsequently the Verilog netlist was converted to a SPICE netlist by NetTran. Synopsys PrimeTime STA tool identified the maximum operating frequency at nominal voltage. Standard libraries are not calibrated for extremely low supply voltages, and thus the maximum operating frequency for lower voltages was found by a laborious trial-and-error process as follows:

- Set supply voltage.
- Assume clock frequency *f*.
- Simulate the execution of the benchmark program on the SPICE netlist using *f*.
- Examine values of registers, data-memory and program counter in each cycle. If erroneous, reduce frequency f and repeat.
- Repeat for all desired levels of supply voltage.

This experiment flow is qualitatively described in Figure 8. Note that ideally we should have re-synthesized the circuit and repeated STA for every voltage level, to optimize the circuit (and its power and performance) for each voltage. However, this is unrealistic because there are no reliable library models for voltage levels below threshold.

The performance/power results are shown in Figure 9, suggesting a potential for power reduction and for improvement in power efficiency at the expense of performance degradation. The charts include variation bars: as mentioned in section IV, delay variation is calculated based on voltage-dependent variance [11] averaged over logic depth of $k$ gates, and the confidence interval is $\pm 3\sigma$. Performance is the inverse of the critical path delay multiplied by IPC.

When lowering the supply voltage from its nominal 0.9V down to the most power efficient point of 0.24V, power efficiency is improved by 8.5× for nominal delays, and by 6.2× when adding cycle time margin to accommodate for delay variations. Figure 10 shows the underlying charts of cycle time (inverse performance, multiplied by IPC) and average power. Note that while delay variations are clearly visible on the cycle time chart, power variations are negligible because power is averaged over the entire circuit of 2700 gates, whereas critical path delays are computed over paths of only about 20 gates.

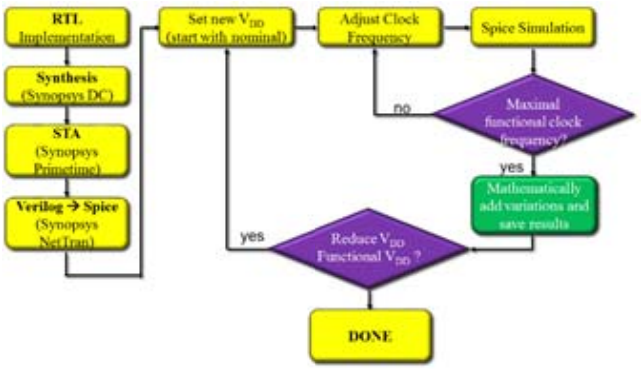


**Figure 8: Synchronous design experiment flow**

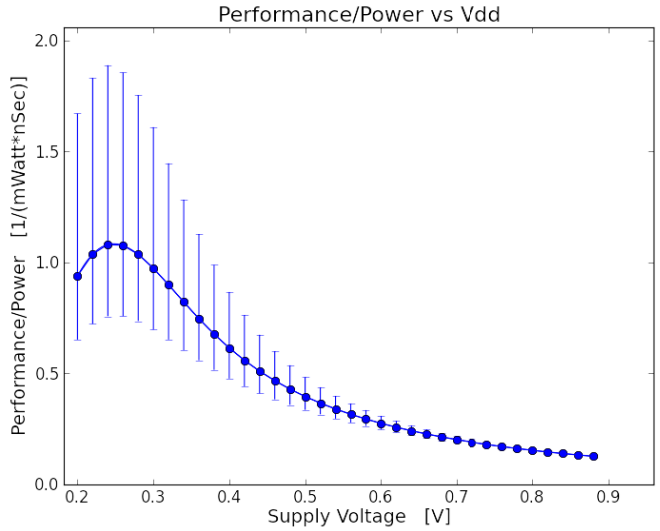


**Figure 9: Power-Efficiency *versus* $V_{DD}$ in synchronous MIPS, 28nm HKMG, HVT ($V_T$=0.6V). As voltage drops from 0.9V down to 0.24V, nominal performance/power ratio enhances 8.5× but only 6.2× when worst-case timing is employed due to expected variations ($\pm 3\sigma$ cycle time error bars are shown; they appear asymmetrical due to inversion: performance ~ 1/cycle time).**

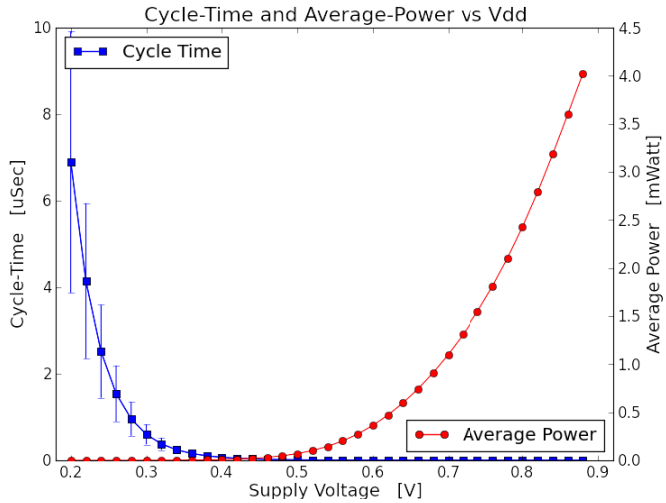


**Figure 10: Cycle time and average power *versus* $V_{DD}$ in synchronous MIPS, 28nm HKMG, HVT ($V_T$=0.6V).**

Power dissipation breakdown for the synchronous MIPS processor demonstrates the increased dominance of leakage power at low voltages (Figure 11).

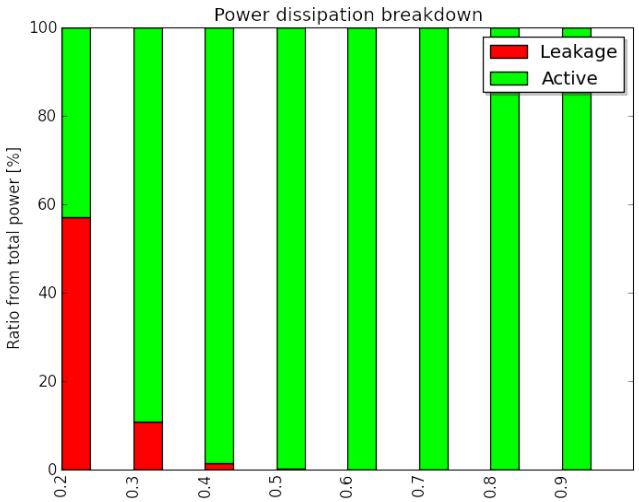


**Figure 11: Power dissipation breakdown for a synchronous MIPS processor**

### *C. Asynchronous Bundled-Data Design*

Bundled-data asynchronous circuits [12][13] can be implemented by hand-crafted design or using dedicated tools [14]. Alternatively, they can be derived by de-synchronization from synchronous circuits [16][17], facilitating the use of standard EDA tools and libraries. We implemented a pipeline-oriented de-synchronization tool following [17] and used it to convert the synchronous circuit of the previous section into a four-phase bundled-data asynchronous processor.

The de-synchronized circuit requires low-skew buffered latch-enable trees, C-element cells and matched delay lines, unavailable in the commercial standard cell library. The low-skew buffered latch-enable trees were inserted manually onto the de-synchronized netlist. The C-elements were implemented using the carry-out function of Full Adder cells, where carry-out is fed back to carry-in. The feedback path is delay-annotated to enable simulations by logic simulators, helping to accelerate the early debug stage. SPICE was used for timing and power measurements. While better circuits may be devised for C-elements, using full-adders helped avoiding circuit design and relying solely on standard cell libraries. Our MIPS processor needed a small number of C-elements, so there was no need to optimize them.

The matched delay lines were asymmetric, to ensure quick "return-to-zero" phase of the four-phase bundled data handshake. Additionally, the delay line was designed to enable post-silicon tuning, in order to avoid excessive margins imposed in naïve bundled-data design (Figure 12).

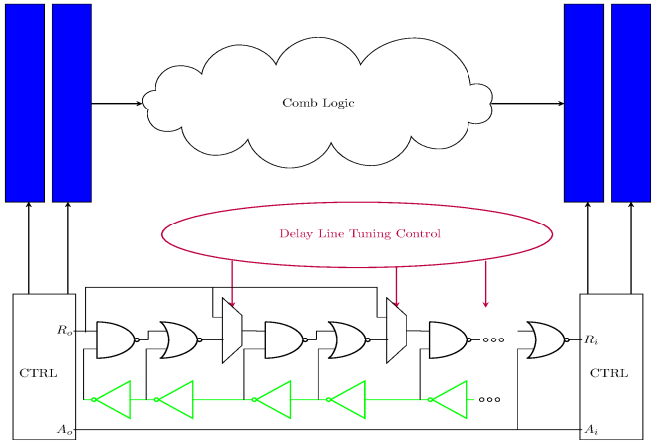


**Figure 12: Asymmetric delay line, with post-silicon delay line tuning.**

It is possible to further improve the performance of bundled-data asynchronous circuits by using data-dependent speculative completion [15], to achieve better than critical-path delay performance.

The experiment flow of the asynchronous design was significantly simpler than the synchronous design flow, due to the self-adjusting properties of asynchronous design. The following asynchronous circuit experiment flow is also shown schematically Figure 13:

- Set supply voltage.
- Simulate the execution of the benchmark program on the SPICE netlist.
- Examine values of registers, data-memory and program counter in each cycle. If erroneous, modify delay lines and re-start the experiment (we did not encounter such erroneous behavior).
- Repeat for all desired levels of supply voltage.

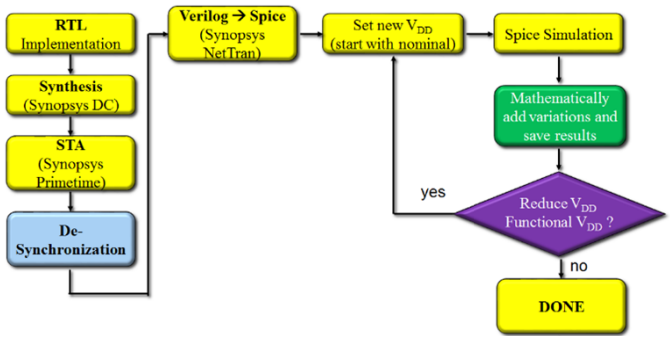


**Figure 13: Asynchronous design experiment flow**

The measured performance/power ratio is displayed in Figure 14. When lowering the supply voltage from its nominal 0.9V down to the most power efficient point of 0.28V, power efficiency is improved by 6.8× for nominal delays and by 5.4× when adding delay lines timing margin to accommodate for delay variations.

By tuning the delay-lines of the manufactured circuit, it is possible to operate the processor at actual speed and avoid excessive timing margins. CPU speed is the average of the maximum critical-path delay over all pipeline stages, which are approximately normally distributed, as discussed in section IV. This may result in binning after delay-line tuning, in order to keep only the faster (and more power efficient) manufactured devices. If the screening criteria is, for example, faster than average performance, the screened devices can reach more than 6.8× power efficiency improvement relative to nominal voltage operation of the bundled-data processor, and 7.4× power efficiency improvement relative to nominal voltage operation of the synchronous processor. The performance/power ratio achieved by the desynchronized circuit with delay-line (plotted in green in Figure 14) is higher than the worst-case performance/power achieved by synchronous design, when taking delay variations into account. The worst-case performance/power of the synchronous design is indicated by the bottom edges of the error bars in Figure 14.

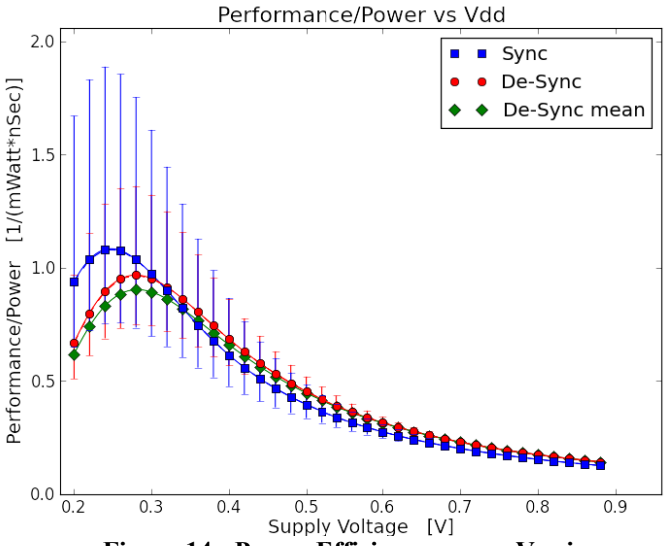


**Figure 14: Power-Efficiency versus $V_{DD}$ in Asynchronous (de-synchronized) MIPS CPU, 28nm HKMG, HVT ($V_T$=0.6V).**
**Nominal Power-Efficiency is shown together with error bars reflecting variations. Mean Power-Efficiency achievable by delay-line tuning and binning is added to demonstrate self-adapting properties of Asynchronous design. Power-Efficiency achieved by Synchronous design is also plotted for reference.**

### *D. Asynchronous Quasi-Delay-Insensitive Design*

Quasi-delay-insensitive (QDI) asynchronous circuits [12][13] are typically implemented by hand-crafted design. There is no known method for de-synchronizing a QDI circuit from a synchronous one. We have implemented the QDI circuit using SystemVerilog CSP and Proteus backend flow [18]. We were unable to perform SPICE simulations on the QDI circuit, due to the lack of SPICE views for the Proteus QDI libraries (now intellectual property of Intel).

Nevertheless, QDI is estimated to be a less suitable design style for ULP CPUs, due to higher power consumption. While completion detection eliminates delay margins and may enhance performance, active power is increased due to additional required toggles in QDI circuits relative to synchronous and bundled-data design styles. In addition, leakage power is also expected to increase due to larger QDI cells [25]. We estimate that the increase in power is more significant than the possible enhancement in performance, so that QDI performance/power ratio is less favorable than synchronous and bundled-data designs.

## V. Results

Aggressive power scaling is suggested in this work as a method of improving power efficiency. Moreover, bundled-data design style with delay-line tuning, and possibly data-dependent delay lines is proposed as a method for dealing with delay variations in sub-threshold domain, and thus significantly improving power efficiency. Figure 15 compares the methods used in this work and other known and widely used power reduction methods – power gating and heterogeneous computing.

Data from real ARM big.LITTLE heterogeneous CPU systems is gathered from [26] and used for this comparison. In order to compare two independent CPU systems, performance and power are normalized to 1 at normal operating conditions (*i.e.*, nominal voltage, CA15 'big' processor). It is shown that supply voltage scaling is more power efficient than power gating, where performance and power are linearly dependent on the relative time the processor is active. In contrast, Heterogeneous computing might achieve superior power efficiency results, however it is limited to a narrow performance and power range. Heterogeneous computing and aggressive supply voltage scaling may be combined together and used as complementary technologies.

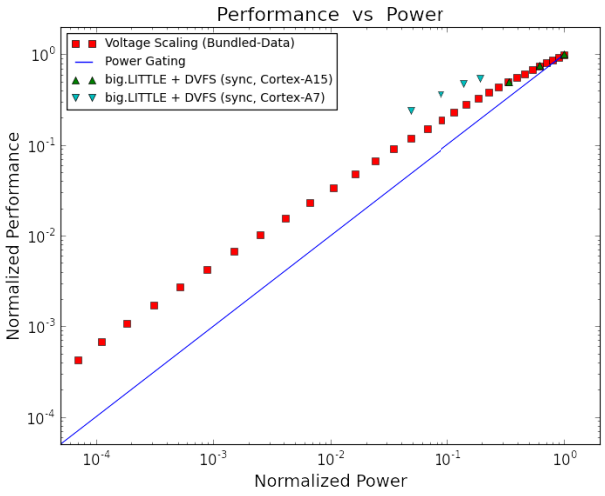


**Figure 15: comparison of known power savings techniques.**

Several ultra-low power systems have been designed and researched in recent years. The systems differ significantly, in process node (65nm to 250nm), supply voltage (300mV to 1V), clock frequency (100KHz to 23MHz), architectural choices (GPP, Accelerator-based) and design style (synchronous, QDI). Figure 16 compares performance and power of the different processors. The processor designed in this work is shown to achieve superior power efficiency, partly due to the selected process node.

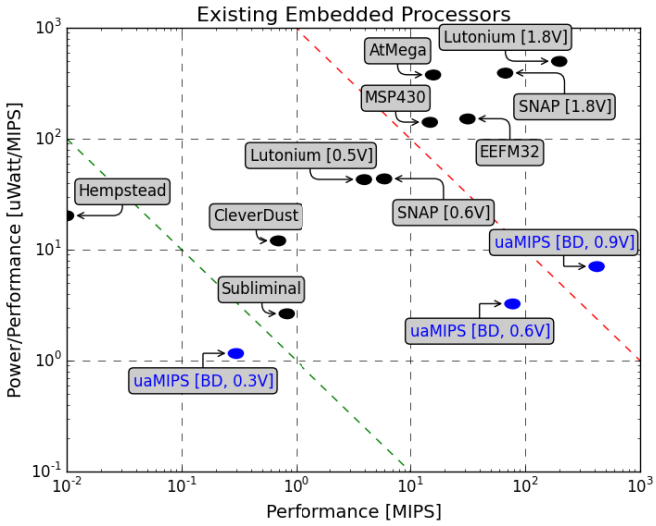


**Figure 16: The Arena of ULP processors [5][6][7][27][28][29]. The De-Synchronized uaMIPS (Micro-Watt Asynchronous MIPS) Processor designed in this work is compared to other ULP processors**

## II. Discussion

ULP processors require computer architects to re-evaluate many of the design choices that were made in recent years, when focus was mainly on achieving maximum performance.

In this work, it has been shown that when implementing ULP processors, it is beneficial to select an advanced process node and use its HP variant, with a selected leakage-control-technology (*e.g.*, HKMG) and use mostly HVT cells. Future work may also compare FD-SOI [30] and FinFET technologies, or other new emerging technologies, to select the most suitable technology for such processors. In addition, future work might research for the reasons why the HP variant achieved superior power efficiency *versus* the LP variant in sub-threshold regime.

Synchronous and Bundled-data design styles have been compared, and bundled-data is suggested as more suitable for ULP processors, due to its ability to deal with delay variations through post-silicon delay-line tuning, and thus avoid the excessive timing margins imposed on a synchronous design operating at ultra-low voltages. QDI design style had been argued to not be suitable for ULP circuits. Future work may implement a QDI processor and support or refute this argument, and also add other design styles to the comparison, such as Carrier-Sense Completion-Detection (CSCD). Future work may also repeat the experiment with SRAM-based implementation for the instruction and data memories, which is typically used in common-day processors.

Finally, further investigation is deserved on Bundled-Data implementation with data-dependent speculative-completion, which might noticeably improve performance and power efficiency of bundled-data circuits operating at sub-threshold regime.

## REFERENCES


[1] J.F. Christmann, E. Beigne, C. Condemine, N. Leblond, P. Vivet, G. Waltisperger and J. Willemin, "Bringing robustness and power efficiency to autonomous energy harvesting microsystems," Int. Symp. Asynchronous Circuits and Systems, 2010

[2] J. Rabaey, "Low-power design essentials," Springer, 2009

[3] A. Wang, B.H. Calhoun and A.P. Chandrakasan, "Sub-threshold Design for Ultra Low-Power Systems," Springer, 2006

[4] M. Alioto, "Ultra-low power VLSI circuit design demystified and explained: A tutorial," Circuits and Systems I: Regular Papers, IEEE Transactions, 2012

[5] K. Clinton, V. Ekanayake, and R. Manohar, "SNAP: A sensor-network asynchronous processor," Int. Symp. Asynchronous Circuits and Systems, 2003

[6] M. Hempstead, D. Brooks, and G. Wei, "An accelerator-based wireless sensor network processor in 130 nm CMOS," IEEE J. Emerging and Selected Topics in Circuits and Systems, 2011

[7] Z. Bo, L. Nazhandali, *et al*., "A 2.60 pJ/Inst subthreshold sensor processor for optimal energy efficiency," Int. Symp. VLSI Circuits, 2006

[8] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," IEEE Micro, 2005

[9] D. Bol, R. Ambroise, D. Flandre and J. Legat, "Interests and limitations of technology scaling for subthreshold logic," Int. Symp. VLSI Systems, 2009

[10] G. Streel and D. Bol, "Scaling Perspectives of ULV Microcontroller cores to 28nm UTBB FDSOI CMOS," J. Low Power Electronics and Applications, 2014

[11] T. Gemmeke, A. Maryam and T. G. Noll, "Noise margin based library optimization considering variability in sub-threshold," PATMOS, 2013

[12] J. Sparsø and S. Furber, "Principles Asynchronous Circuit Design," Kluwer Academic Publishers, 2002

[13] P.A. Beerel, R.O. Ozdag and M. Ferretti, "A Designer's Guide to Asynchronous VLSI," Cambridge University Press, 2010

[14] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, and A. Yakovlev, "Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers," XI Conf. Design of Integrated Circuits and Systems, 1996

[15] S. M. Nowick, K. Y. Yun, P. A. Beerel, and A.E. Dooply, "Speculative completion for the design of high-performance asynchronous dynamic adders," Int. Symp. Asynchronous Circuits and Systems, 1997

[16] A. Branover, R. Kol and R. Ginosar, "Asynchronous Design By Conversion: Converting Synchronous Circuits into Asynchronous Ones," DATE, 2004

[17] J. Cortadella, A. Kondratyev, L. Lavagno and C. Sotiriou, "De-synchronization: Synthesis of Asynchronous Circuits from Synchronous Specifications", ICCAD, 2006

[18] P. A. Beerel, G. D. Dimou, and A. M. Lines, "Proteus: An ASIC Flow for GHz Asynchronous Designs", DATE, 2011

[19] A. J. Martin, "Programming in VLSI: from communicating processes to delay insensitive circuits," in Developments in Concurrency and Communication, UT Year of Programming Series, Addison Wesley, 1990

[20] O. C. Akgun, "Energy Efficiency Enhance method of Sub-threshold Digital CMOS: Modeling, Technology Selection, and Architectural Exploration", PhD dissertation, EPFL, Switzerland, 2009

[21] W. Zhao and Y. Cao, "Predictive Technology Model (PTM)," URL: http://ptm.asu.edu, 2008

[22] D. Harris, R. Ho, G. Wei, and M. Horowitz, "The fanout-of-4 inverter delay metric." Unpublished Manuscript. http://odin. ac. hmc. edu/harris/research (1997).

[23] D. Blaauw, C. Kaviraj, S. Ashishand and L. Scheffer, "Statistical timing analysis: From basic principles to state of the art," IEEE Trans. CAD, 2008

[24] L. Nazhandali, M. Minut, B. Zhai, J. Olson, T. Austin and D. Blaauw, "A second-generation sensor network processor with application-driven memory optimizations and out-of-order execution", ACM Int. Conf. Compilers, Architectures and Synthesis for Embedded Systems, 2005

[25] P. Beerel, private communication, 2014

[26] P. Greenhalgh, "Big.LITTLE Processing with ARM Cortex-A15 & Cortex-A7", ARM Holdings white paper, 2011

[27] R. Manohar, private communication, 2014

[28] Raval, K. Rajkumar, C. H. Fernandez and C. J. Bleakley, "Low-power TinyOS tuned processor platform for wireless sensor network motes," ACM Trans. Design Automation of Electronic Systems, 15(3), 2010.

[29] A. J. Martin, M. Nystrom *et al.*, "The Lutonium: A sub-nanojoule asynchronous 8051 microcontroller," Ninth Int. Symp. on Asynchronous Circuits and Systems, 2003.

[30] M. Gautschi, D. Rossi and L. Benini, "Customizing an open source processor to fit in an ultra-low power cluster with a shared L1 memory," 24th Great Lakes Symp. VLSI, 2014.