J. Low Power Electron. Appl. **2015**, *5*, 57-68; doi:10.3390/jlpea5020057

OPEN ACCESS

Journal of
Low Power Electronics and Applications
ISSN 2079-9268
www.mdpi.com/journal/jlpea

Article

A Robust Ultra-Low Voltage CPU Utilizing Timing-Error Prevention [†]

Markus Hiienkari [1],*, Jukka Teittinen [1], Lauri Koskinen [1], Matthew Turnquist [2], Jani Mäkipää [3], Arto Rantala [3], Matti Sopanen [3] and Mikko Kaltiokallio [4]

[1] Technology Research Center, University of Turku, Joukahaisenkatu 1C, 20520 Turku, Finland; E-Mails: jukka.teittinen@utu.fi (J.T.); lauri.koskinen@utu.fi (L.K.)

[2] Department of Micro and Nanosciences, Aalto University, Otakaari 5A, 02150 Espoo, Finland; E-Mail: matthew.turnquist@aalto.fi

[3] VTT Technical Research Centre of Finland, Tietotie 3, 02150 Espoo, Finland; E-Mails: jani.makipaa@vtt.fi (J.M.); arto.rantala@vtt.fi (A.R.); matti.sopanen@vtt.fi (M.S.)

[4] TDK, Keilaranta 8, 02601 Espoo, Finland; E-Mail: mikko.kaltiokallio@epcos.com

[†] This paper is an extended version of our paper published in IEEE S3S Conference 2014.

* Author to whom correspondence should be addressed; E-Mail: markus.hiienkari@utu.fi; Tel.: +358-40-5631816 .

Academic Editor: David Bol & Steven A. Vitale

Received: 20 February 2015 / Accepted: 10 April 2015 / Published: 17 April 2015

Abstract: To minimize energy consumption of a digital circuit, logic can be operated at sub- or near-threshold voltage. Operation at this region is challenging due to device and environment variations, and resulting performance may not be adequate to all applications. This article presents two variants of a 32-bit RISC CPU targeted for near-threshold voltage. Both CPUs are placed on the same die and manufactured in 28 nm CMOS process. They employ timing-error prevention with clock stretching to enable operation with minimal safety margins while maximizing performance and energy efficiency at a given operating point. Measurements show minimum energy of 3.15 pJ/cyc at 400 mV, which corresponds to 39% energy saving compared to operation based on static signoff timing.

Keywords: timing-error prevention (TEP); Ultra-Low Power (ULP); digital CMOS; clock stretching; near-threshold; variability; energy-efficiency

1. Introduction

With constantly tightening power budgets and incentives to add new features, energy consumption has become one of the most important aspects of portable electronics. The minimum energy consumption of digital logic is traditionally achieved by utilizing a supply voltage which is below the transistor threshold voltage [1]. Operation in this sub-threshold -region has several practical limitations such as radically reduced performance and exponentially increased variability.

Recent manufacturing processes have pushed the minimum energy point (MEP) towards the threshold voltage, which mitigates performance and variance concerns and makes near-threshold operation much more attractive and practical. Near-threshold systems may still need to satisfy temporary high throughput requirements, which can be satisfied by utilizing dynamic voltage and frequency scaling (DVFS). In energy-constrained systems, supply voltage also varies slowly with time (battery, solar cell, *etc.*), creating additional challenges for ensuring efficient and reliable operation with a dynamic performance target. Despite the challenges, near-threshold computing has recently gained increasing amount of attention due to its suitability to IoT and other extremely energy-limited applications which yet have moderate performance requirements.

The goal of this research is to study applicability of timing-error prevention (TEP) to an ultra-low voltage CPU. We present two variants of a 32-bit RISC microprocessor which can operate reliably at near-threshold voltages with minimal safety margins. The CPU cores are optimized for design compatibility and Ultra-Dynamic Voltage Scaling (UDVS) [2] respectively. They are both placed on same die, which is manufactured in 28 nm CMOS technology.

Section 2 of this article describes more closely the operation at ultra-low voltages with background theory and examples, and presents the concepts of adaptive timing. The design process of the CPU cores is described in Section 3, and Section 4 presents the measurement results and analysis. Finally, Section 5 concludes the article.

2. Ultra-Low Voltage Operation

The goal of this section is to make the reader familiar with the principles and incentives of ultra-low voltage operation. This is followed by a description of the most important challenges, and two possible solutions to overcome them.

2.1. Motivation and Challenges

For digital static CMOS logic, when the performance constraints allow, the straightforward solution to energy minimization is to lower the operating voltage all the way to the MEP. This point has been proven to exist around 0.2–0.4 V depending on various factors [3]. With older process nodes, this is in the subthreshold operation region and for newer process nodes, in the near-threshold region. Figure 1 shows the MEP for 65 nm and 28 nm low-power (LP) bulk processes with different activity factors. On the 28 nm process, MEP resides near the threshold voltage. With lower leakage and higher V_t, the 65 nm process has MEP at approximately 100 mV lower voltage which resides in subthreshold region.

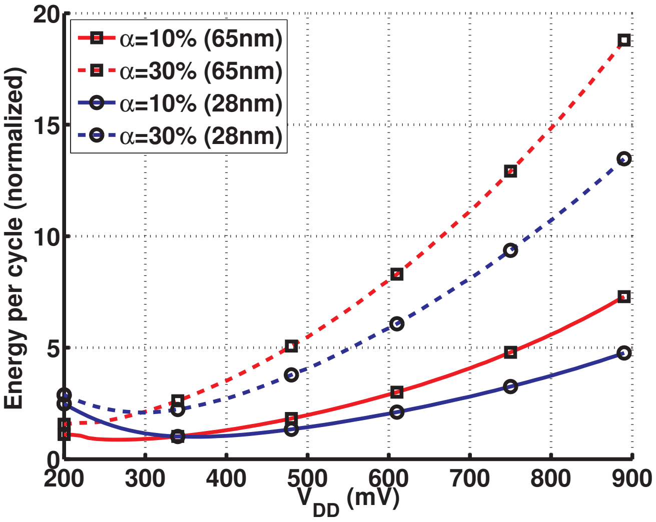


Figure 1. The minimum-energy points for 65 nm and 28 nm low-power (LP) bulk processes using low-V_t (LVT) transistors. α denotes the activity factor. The test circuit here is a 13 stage minimum-size negative-AND -gate (NAND) ring oscillator (based on a circuit presented in [3]).

However, the advantages of operating at the MEP are easily lost due to higher device variability. In subthreshold, process, supply voltage, temperature, and aging (PVTA) induced variance is amplified in the subthreshold region due to exponential dependency of the subthreshold current [3]:

$$I_{ds,sub} = I_0 \exp\left(\frac{V_{gs} - V_t}{nV_{th}}\right)\left(1 - \exp\left(\frac{-V_{ds}}{V_{th}}\right)\right) \tag{1}$$

In Equation (1), I_0 is the drain current when $V_{gs} = V_t$; V_t is the threshold voltage; n is the subthreshold slope factor; and V_{th} is the thermal voltage. While near-threshold operation mitigates the exponential dependency, increased variance still remains when compared to nominal operating voltages.

The theory is verified with a simulation of an example timing path from a test circuit in 28 nm CMOS. The path is simulated with 100 Monte Carlo runs, and Table 1 shows resulting nominal, minimum and maximum delays for near-threshold voltages. The results demonstrate the design challenge of near-threshold logic, as local variation can cause up to 30% delay deviation from nominal. Even without the exponential dependence on transistor parameters of subthreshold, robust operation clearly demands overly large design margins or individual post-fabrication measurements of the components; the former negating the minimum energy operation, and the latter increasing production costs considerably.

Table 1. Delay of an example critical path in 28 nm CMOS with 100 Monte-Carlo iterations.

V_{DD}	T_{avg} (ns)	T_{min} (ns)	T_{max} (ns)	σ (ns)
0.3 V	305	237 (–22.2%)	393 (+28.9%)	32.8 (10.8%)
0.4 V	40.1	31.9 (–20.4%)	52.0 (+29.7%)	3.9 (9.7%)
0.5 V	6.63	5.57 (–16.0%)	8.20 (+23.7%)	0.5 (7.3%)
0.6 V	1.90	1.71 (–10.0%)	2.17 (+14.2%)	0.09 (4.7%)

2.2. Adaptive Timing Methods

Here, we present 2 alternative solutions for solving the issue of large timing margins. Several researches have successfully applied these methods to a number of test chips.

2.2.1. Timing-Error Detection

Timing-Error Detection (TED)-based systems have been shown to be effective in largely removing the variation-incurred timing margins [4–6]. The achieved lower margins have then enabled energy savings (*i.e.*, lower V_{DD} [4]) or a higher yield [5]. The TED methodology is based on having the system operate at a voltage and frequency point in which the timing of critical paths fails intermittently. These failed timing occurrences are detected and handled. The overhead of detection and handling has to be lower than the energy savings resulting from lower V_{DD} or higher frequency. A classic example of a TED system is the instruction replay, where an instruction with a failed timing path is replayed [4].

2.2.2. Timing-Error Prevention

Timing-Error prevention (TEP) is a variant of TED which utilizes time borrowing (TB). When a system is timed for zero TB at normal operation, fractionally late signals can be tolerated without errors occurring in the system. However, TB sets timing requirements on the subsequent pipeline stages; namely the cumulative TB of consecutive stages must never exceed the TB window and therefore careful design time planning is required. Combining TED with TB into TEP conceives a system which can tolerate late coming signals, but which does not require special arrangements with regards to stage lengths. The method works as follows: When a late signal arrives, time borrowing occurs normally. Time borrow events (TBE) are detected with latches or special time borrow flip-flops. A recovery is necessary to prevent borrowed time from accumulating and generating a timing error. This can be done by moving the clock phase [7] or gating the clock on per-stage basis [8].

Since late signals are allowed with both TED and TEP technique, setup-timing margins are eliminated thereby allowing energy reduction. When TEP is integrated into a dual-phase latch pipeline, the resulting system can not only tolerate late signals, but does not require additional hold buffers on fast paths [8]. This is a large advantage compared to a traditional TED system. As shown in Table 1, minimum delay uncertainty is also clearly worsened at near-threshold voltages, which would significantly increase the demand for extra hold buffers on fast paths. A dual-phase latch pipeline can also be driven by 2 non-overlapping clocks, which increases skew tolerance and decreases general requirement for hold buffers.

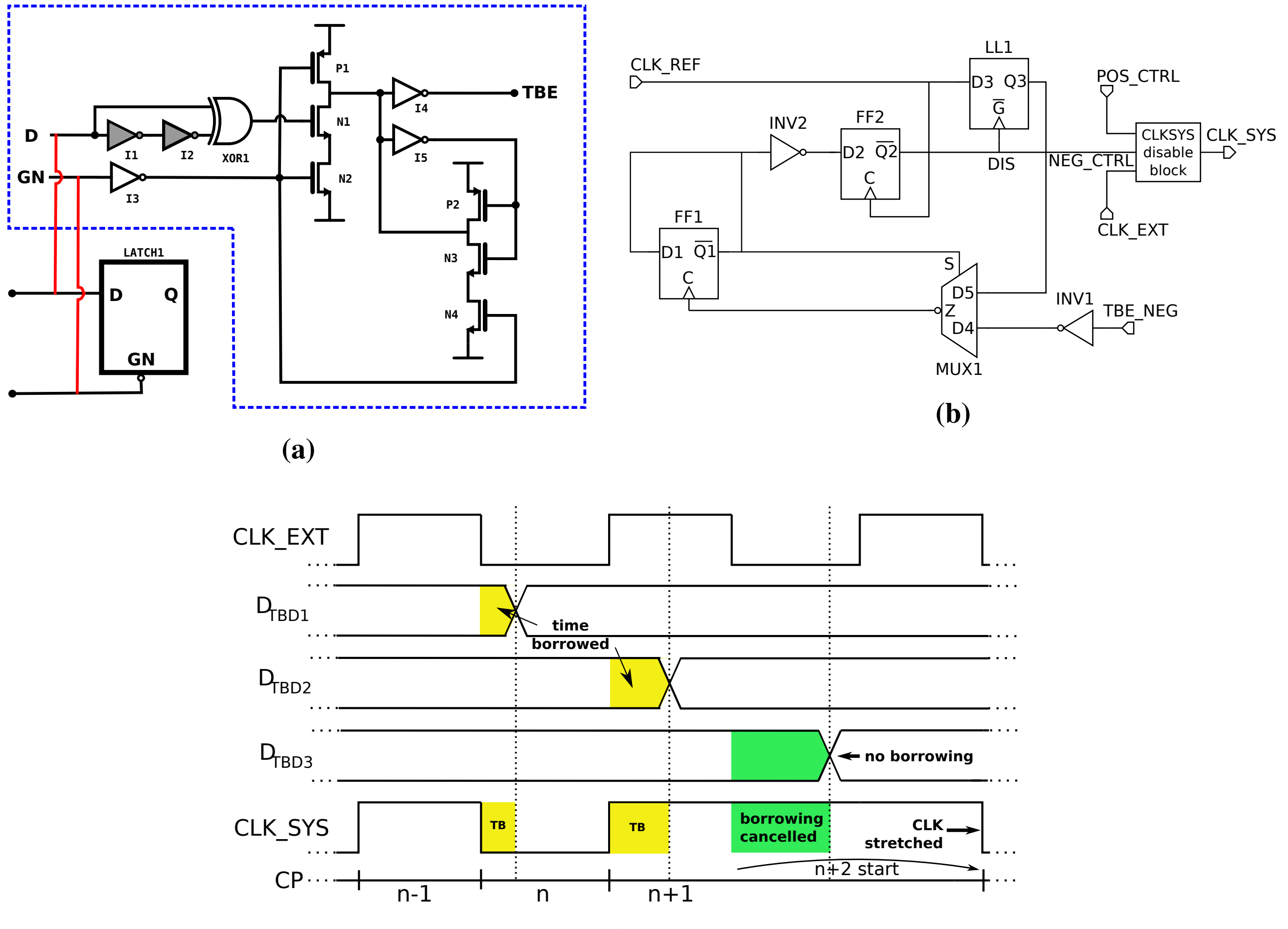
With a balanced pipeline, a TEP system has margins but only when they might be required (when time borrowing is detected). Importantly, the system is a zero margin device at design time and adaptive margin device at runtime.

3. Design of a 32-bit RISC CPU with TEP

The CPU cores of our system are modified versions of a freely available, open-core LatticeMico32 CPU [9]. LM32 is a configurable medium-scale RISC microprocessor with 6 pipeline stages, full

GCC toolchain support and sufficient performance (1.14 DMIPS/MHz, 1.83 Coremark/MHz) even for demanding sensor network applications.

As described in [10,11], we have enhanced the CPUs with critical path monitoring combined with timing-error prevention. Our TEP system enables time borrowing on all paths. Time borrow events are detected with time borrow detector (TBD) circuits, which are integrated into critical path latches as illustrated in Figure 2a. The circuit indicates when TB occurs, and TBEs are combined and propagated to a clock control circuit, which is responsible for error prevention. The clock control circuit is a state machine consisting of a small number of flip-flops and logic gates (Figure 2b). It is able to shift the global clock phase after a TBE as illustrated in Figure 2c. Therefore, the TEP system prevents the stacking of TB, which could otherwise lead to a timing error. In addition to per-cycle timing error prevention, the real-time feedback from critical paths (combined TBEs) is output from the chip, allowing it to be used in tuning the microprocessor voltage/frequency in order to minimize die characterization effort.



(a)

(b)

(c)

Figure 2. An overview of the timing-error prevention system. Based on [11] by the authors of this article. (**a**) The time borrow detector (TBD) circuit; (**b**) the clock control circuit (negative branch in detail); (**c**) a high-level illustration of adaptive clocking, where clock stretching prevents time borrow accumulation. © 2015 IEEE.

Since the unmodified LM32 is a standard edge-sensitive pipeline, additional design steps are required to make it suitable for TEP. Out of the compatible sequential cells, pulse-latches and TB flip-flops would be the most straightforward to integrate into an edge-sensitive design, but they would add minimum-delay constraints into fast paths and raise robustness concerns at near-threshold voltages. Standard latches are readily available in foundry design kits and have no robustness issues, but an edge-sensitive design needs to be fully transformed into a dual-phase latch pipeline to avoid half-cycle minimum delay requirement for critical paths. We created 2 CPU variants (CPU1 and CPU2) with dual-phase latch pipelines to study the applicability and effectiveness of different implementation strategies. As shown in Table 2, different design techniques were applied to the 2 CPUs. The fully custom TBD cell has 3.2× area of a minimum-sized latch, and the total TBD area overhead specified in the table is the percentage of combined TBD area out of standard cell area. The area overheads of OR-tree and clock control block are insignificant in comparision and are not listed.

Table 2. Design targets and results for the CPUs.

Design attribute	**CPU1**	**CPU2**
Pipeline implementation	Latch transformation + retime	RTL rewrite
Memory type	SRAM	Latch array
Supply voltage (mV)	400	400
Frequency target (MHz)	0.89	0.67
Sequential cells	3623 latches	2006 latches
Time-borrow detectors	272	256
TBD area overhead (%)	7.5	10.9
Die area (excl. memory) (mm^2)	0.030	0.022

3.1. CPU1

CPU1 uses automatic latch transformation and EDA-tool backed fixed-phase retiming [12] during synthesis, which is illustrated (for a single flip-flop) in Figure 3. The flip-flops are split into 2 latches, and the whole design is retimed in order to balance the paths. As the method is mostly automatic, it has been demonstrated to be applicable in larger designs [8]. However, there are some problems in practice. The balancing performed by the EDA tools is not guaranteed to be optimal—a carefully optimized flip-flop pipeline may turn into a pipeline where the path between the original master and slave latch becomes consistently shorter than the paths between the latch pairs, reducing overall efficiency. Also, some of the tools are not able to retime designs with multiple voltage domains. The method is also prone to increase clock network complexity, as the number of registers in the design increases.

As the target of CPU1 is to demonstrate applicability of TEP in a typical RISC processor, the core is supplemented with standard on-chip SRAM memory. This creates a new challenge, since it adds a hard edge-sensitive block inside otherwise level-sensitive design. As with [8], our design employs a SRAM wrapper, which moves reads and writes to the falling clock edge to ensure that SRAM input signals are always valid. However, this reduces the allowable time on the paths into subsequent latches. It also prevents usage of TBD latches at the end of these paths, since the data always transitions during their transparent phase, which would generate a possibly false positive. However, the vendor-provided SRAM in our design operates in a separate power domain of higher voltage due to retention requirements, making it significantly faster than the core logic and thereby eliminating the issue of stricter timing requirements. Based on simulations, a core voltage up to 0.5 V was shown to cause no SRAM performance bottleneck in the system.

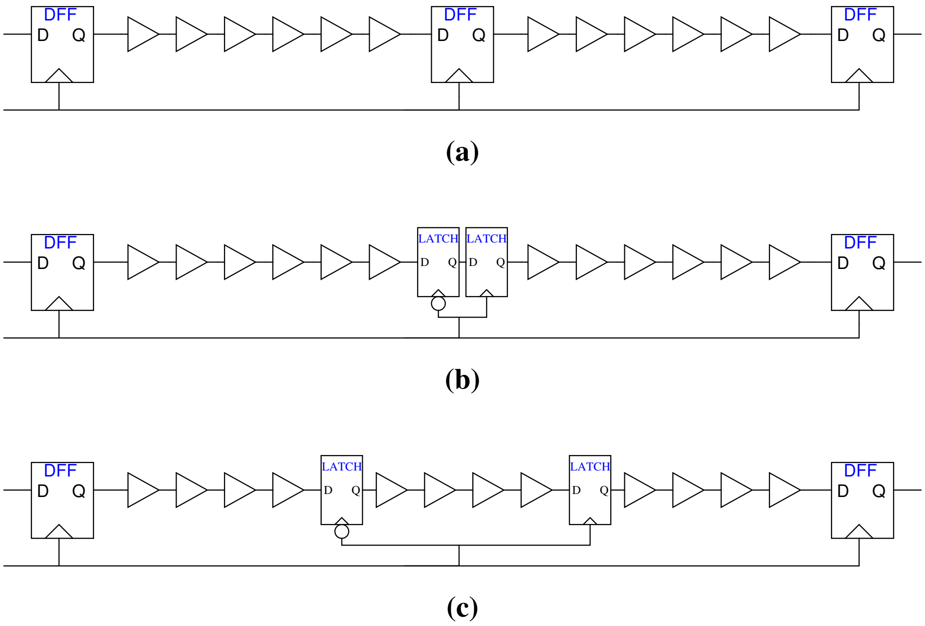


Figure 3. An example of latch transformation and retiming. This method is used in CPU1 implementation. (**a**) A standard pipeline with flip-flops; (**b**) The center flip-flop is split into 2 latches; (**c**) Retiming applied.

3.2. CPU2

A more optimal starting point for a TEP implementation would be a CPU design which has two-phase latch pipeline to start with, and a more flexible memory interface. In order to study the benefits of such base design, we created another variant of the core. CPU2 is manually transformed at register-transfer level (RTL) into dual-phase system before synthesis, and it uses a small latch array memory instead of SRAM. The former removes EDA tool limitations as the registers inside the design are synthesized directly to latches. The latter property removes the need of wrapper logic and memory path length constraining. Moreover, due to the lack of SRAM retention issues, the system is implemented in a single voltage domain. This relaxes voltage scaling limitations, and thus allows operation at an ultra-wide voltage range.

Figure 4 illustrates the RTL rewrite process through a simple 3-stage pipeline example. The edge-sensitive processes are changed into level-sensitive, and consecutive stages are assigned to alternating clock phases. Any paths starting and terminating on a stage with the same polarity—including the case where source and target register is the same (*pc_f* in the example)—must be supplemented by an additional synchronizing latch stage. Despite the additional latches, the total clock network load will be smaller than the original flip-flop design due to latch being smaller and simpler than a flip-flop. Changing a pipeline stage to operate on a single phase halves the achievable clock frequency, but also cuts the pipeline latency into half. It should be noted that conditional constructs in a level-sensitive block will explicitly synthesize into clock-gated latches. This is in contrast to an edge-sensitive block, where a similar construct would synthesize to a circular feedback path by default, unless clock-gating was enabled in the synthesis tool.

```
// These will synthesize into flip-flops
reg [31:0] pc_f;
reg [31:0] inst_f;
reg [9:0] x_result;
reg [9:0] oper_x_0;
reg [9:0] oper_x_1;

// Fetch, decode and execute stages
always @(posedge clk)
begin
    inst_f <= #1 memory[pc_f];
    pc_f <= #1 pc_f + 4;
    oper_x_0 <= #1 inst_f[9:0];
    oper_x_1 <= #1 inst_f[19:10];
    x_result <= #1 oper_x_0 + oper_x_1;
end
```

(a)

```
// These will synthesize into latches
reg [31:0] pc_f;
reg [31:0] pc_f_DEL;
reg [31:0] inst_f;
reg [9:0] x_result;
reg [9:0] oper_x_0;
reg [9:0] oper_x_1;

// Fetch stage (POSITIVE phase)
always @(*)
if (clk == `TRUE)
begin
    inst_f <= #1 memory[pc_f_DEL];
    pc_f <= #1 pc_f_DEL + 4;
end

// Syncronizing latch stage for pc_f
always @(*)
if (clk == `FALSE)
    pc_f_DEL <= #1 pc_f;

// Decode stage (NEGATIVE phase)
always @(*)
if (clk == `FALSE)
begin
    oper_x_0 <= #1 inst_f[9:0];
    oper_x_1 <= #1 inst_f[19:10];
end

// Execute stage (POSITIVE phase)
always @(*)
if (clk == `TRUE)
    x_result <= #1 oper_x_0 + oper_x_1;
```

(b)

Figure 4. An example of register-transfer level (RTL) rewrite for dual-phase latch pipeline. This method is used in CPU2 implementation. (**a**) Standard edge-sensitive pipeline; (**b**) equivalent level-sensitive pipeline.

4. Measurement Results and Discussion

Based on the simulations and estimated average activity, the CPUs were synthesized and place & routed at the estimated MEP of 400 mV. With the exception of TBD and level shifters (in CPU1), all gates were from vendor standard cell library, which was re-characterized at the target operation point. As shown in Table 3 and explained in Section 3.2, the frequency of CPU2 is lower due to its pipeline structure, but its area is also smaller as shown in the chip microphotograph in Figure 5.

Silicon measurements of 6 test chips verified correct operation for both CPUs at their signoff frequencies, but due to safety margins and inaccuracies in timing libraries, optimal energy or performance was not achieved. This was evidenced by studying timing feedback from critical paths, which showed that TEP was not activated. Further testing verified that the CPUs could be run at significantly higher frequencies until time borrowing started to occur. Since the clock stretching halves the effective frequency when activated, the optimal frequency was set so that time borrow events only occurred rarely.

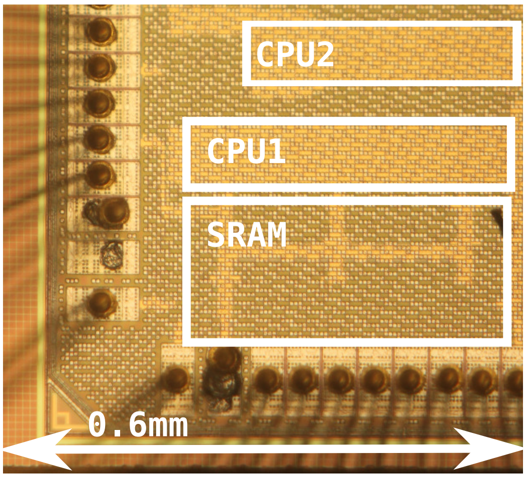


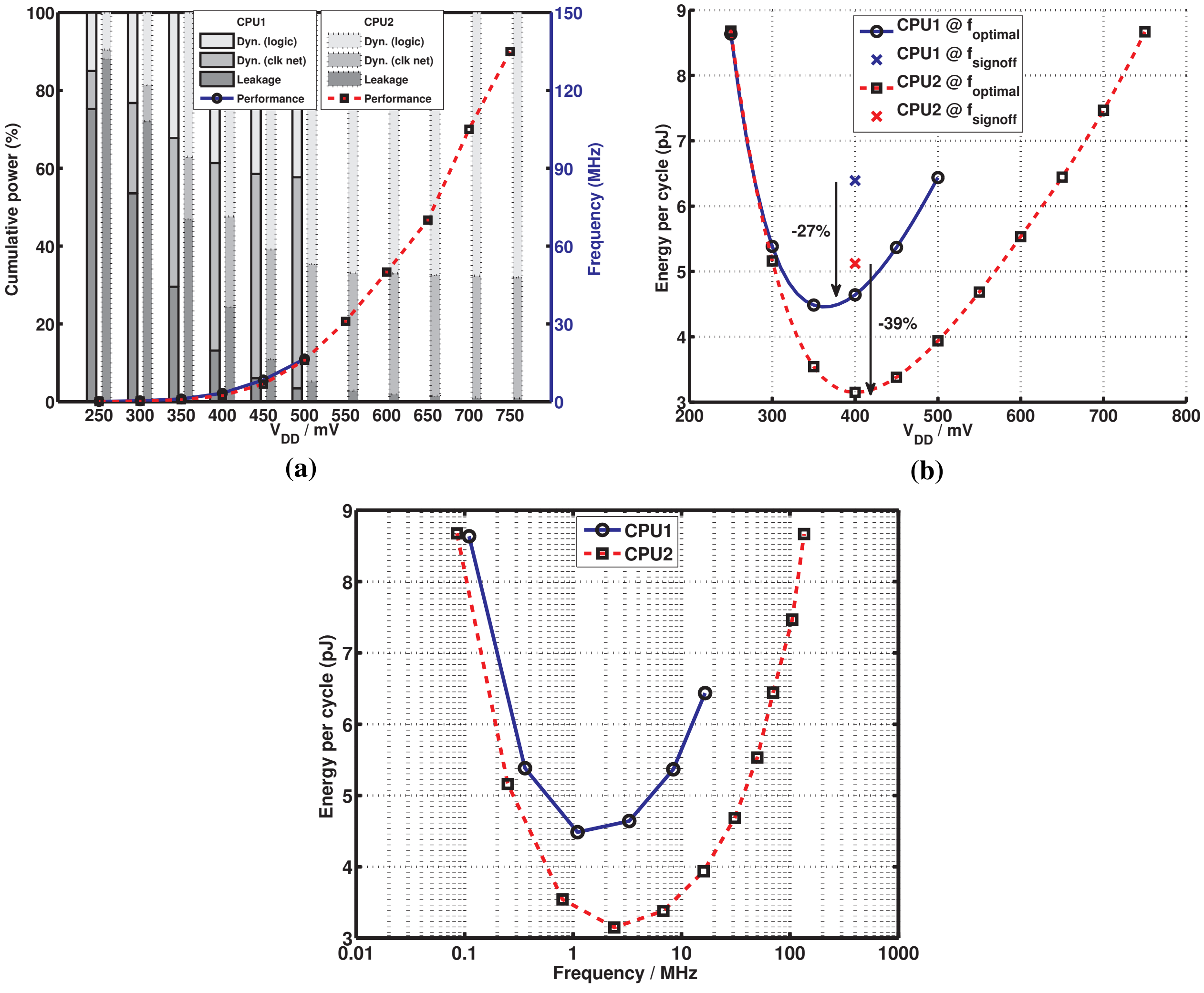
Figure 5. Test chip microphotograph. Based on [10] by the authors of this article. © 2015 IEEE.

Table 3. Comparison with state of the art. Based on [10] by the authors of this article. © 2015 IEEE.

Design metric	MIT, JSSC [13]	Sleepwalker [14]	This Work (CPU2)
Technology	65 nm CMOS	65 nm CMOS LP/GP	28 nm CMOS LP
CPU	16-bit MSP430 compatible	16-bit MSP430 compatible	32-bit LatticeMico32
CPU area (mm^2)	0.14	approx. 0.12	0.022
Ext. supply (V)	1.2	1.0–1.2	1.0 V–1.5 V
Int Vdd (V)	0.3–0.6	0.32–0.48	0.25–0.5
Frequency (MHz)	0.43 (25 C)	25	2.4 (0.4 V/25 C)
CPU energy (pJ/cyc)	6–10	2.6	3.15
DMIPS/MHz	0.45	0.45	1.14
DMIPS/mJ (estim.)	60	173	362

Voltage was scaled next to study energy/performance tradeoff at various operation points (Figure 6a,b). The lower bound (250 mV) was based on the simulated functional failure rate as described in [10], while the upper bound of CPU1 was limited by fixed SRAM performance. Voltage of CPU2 could be scaled up to 750 mV, after which further performance improvements were not possible as the externally generated clock became the limiting factor. If an on-chip PLL was used for clock generation, we estimate that CPU2 would operate approximately at 300 MHz with a nominal voltage of 1.0 V.

Figures 6a–c illustrate power distribution, performance and energy of the CPUs running a stress test at the measured voltage/frequency ranges. The dynamic energy consumed by logic is similar for both CPUs as both are inherently same architecture, but the more complicated clock network of CPU1 results to higher total dynamic energy. CPU1 also has higher leakage power, but when integrated over a clock period, the resulting leakage energy is slightly smaller because of higher clock frequency. Thus, when leakage power is not dominant (from 350 mV upwards), CPU2 is over 30% more energy-efficient than CPU1.



(**a**)

(**b**)

(**c**)

Figure 6. Measurement results for CPU1 and CPU2. (**a**) Power distribution and performance at optimal frequency; (**b**) E_{cyc} at optimal and signoff-estimated frequencies; (**c**) energy versus performance.

The frequency of CPU1 ranges from 110 kHz to 16.5 MHz, and its minimum energy point (4.48 pJ/cyc) is at around 350 mV. The ultra-wide voltage scale of CPU2 allows it to operate up to 135 MHz, making it suitable for a wide number of usage scenarios. Minimum energy consumption (3.15 pJ/cyc) is achieved at a near-threshold voltage of 400 mV. Compared to operation based on static signoff timing, TEP-assisted DVFS allows reducing the energy consumption by 27% and 39% for CPU1 and CPU2 respectively.

The results demonstrate that TEP can utilized successfully in a typical RISC processor with a few automated extra steps during the design flow (CPU1). However, largest benefits are achieved with a processor, which is designed TEP in mind (CPU2). Such processor design would ideally have a dual-phase level-sensitive pipeline, and no hard edge-sensitive blocks. The measured energy consumption is compared with state of the art in Table 3, which shows that CPU2 achieves over 2× lower effective energy than competition with smaller area.

5. Conclusions

Ultra-low voltage CPUs are not adopted widely in commercial systems due to complex design and reliability concerns. Shown here, is a simple adaptive technique, timing-error prevention. This technique helps the two presented CPU variants to operate reliably with minimal safety margins in order to ensure optimal performance and energy consumption. The first presented CPU is converted from a standard RISC design to a TEP-compatible system with minimal amount of manual intervention, demonstrating the technique's applicability to contemporary designs. The second CPU variant is specifically designed with ultra-low voltage operation and TEP in mind, enabling higher energy savings and ultra-wide voltage operation from 250 mV to 750 mV. Compared to operation at static signoff-based frequency, the TEP system reduces energy consumption of the presented CPUs by 27% and 39%.

Acknowledgments

Technology Industries of Finland Centennial Foundation and Academy of Finland #270585.

Author Contributions

Markus Hiienkari designed and implemented majority of the digital parts of the system, and wrote test software and the manuscript. Matthew Turnquist designed analog blocks residing on the same chip and assisted in the optimization of TBD cell. Jani Mäkipää designed the clock control block, and Jukka Teittinen assisted in layout-related tasks. Matti Sopanen characterized the standard cell libraries used in the design, and Arto Rantala designed pad cells for the chip. Mikko Kaltiokallio helped in generating on-chip power supply network, and Lauri Koskinen was the system developer and coordinated the research project.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Bol, D.; Ambroise, R.; Flandre, D.; Legat, J.D. Interests and Limitations of Technology Scaling for Subthreshold Logic. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2009**, *17*, 1508–1519.
2. Calhoun, B.; Chandrakasan, A. Ultra-Dynamic Voltage scaling (UDVS) using sub-threshold operation and local Voltage dithering. *IEEE J. Solid-State Circuits* **2006**, *41*, 238–245.
3. Wang, A.; Calhoun, B.H.; Chandrakasan, A.P. *Sub-Threshold Design for Ultra Low-Power Systems (Series on Integrated Circuits and Systems)*; Springer-Verlag New York, Inc.: Secaucus, NJ, USA, 2006.

4. Das, S.; Tokunaga, C.; Pant, S.; Ma, W.H.; Kalaiselvan, S.; Lai, K.; Bull, D.; Blaauw, D. RazorII: *In Situ* Error Detection and Correction for PVT and SER Tolerance. *IEEE J. Solid-State Circuits* **2009**, *44*, 32–48.
5. Bowman, K.; Tschanz, J.; Lu, S.; Aseron, P.; Khellah, M.; Raychowdhury, A.; Geuskens, B.; Tokunaga, C.; Wilkerson, C.; Karnik, T.; *et al.* A 45 nm Resilient Microprocessor Core for Dynamic Variation Tolerance. *IEEE J. Solid-State Circuits* **2011**, *46*, 194–208.
6. Mäkipää, J.; Turnquist, M.J.; Laulainen, E.; Koskinen, L. Timing-Error Detection Design Considerations in Subthreshold: An 8-bit Microprocessor in 65 nm CMOS. *J. Low Power Electron. Appl.* **2012**, *2*, 180–196.
7. Chae, K.; Mukhopadhyay, S. A Dynamic Timing Error Prevention Technique in Pipelines with Time Borrowing and Clock Stretching. *IEEE Trans. Circuits Syst. I***2014**, *61*, 74–83.
8. Fojtik, M.; Fick, D.; Kim, Y.; Pinckney, N.; Harris, D.; Blaauw, D.; Sylvester, D. Bubble Razor: Eliminating Timing Margins in an ARM Cortex-M3 Processor in 45 nm CMOS Using Architecturally Independent Error Detection and Correction. *IEEE J. Solid-State Circuits* **2013**, *48*, 66–81.
9. LatticeMico32: Open, Free 32-Bit Soft Processor. Available online: http://www.latticesemi.com/en/Products/DesignSoftwareAndIP/IntellectualProperty/IPCore/IPCores02/LatticeMico32.aspx (accessed on 20 Febuary 2015).
10. Hiienkari, M.; Teittinen, J.; Koskinen, L.; Turnquist, M.; Kaltiokallio, M. A 3.15 pJ/cyc 32-bit RISC CPU with Timing-Error Prevention and Adaptive Clocking in 28 nm CMOS. In Proceedings of the 2014 IEEE Custom Integrated Circuits Conference (CICC), San Jose, CA, USA, 15–17 September 2014; pp. 1–4.
11. Hiienkari, M.; Teittinen, J.; Koskinen, L.; Turnquist, M.; Kaltiokallio, M.; Makipaa, J.; Rantala, A.; Sopanen, M. Ultra-Wide Voltage Range 32-bit RISC CPU with Timing-Error Prevention in 28 nm CMOS. In Proceedings of the 2014 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), Millbrae, CA, USA, 6–9 October 2014; pp. 1–2.
12. Yoshikawa, K.; Hagihara, Y.; Kanamaru, K.; Nakamura, Y.; Inui, S.; Yoshimura, T. Timing Optimization by Replacing Flip-Flops to Latches. In Proceedings of the 2004 Asia and South Pacific Design Automation Conference, ASP-DAC '04, Singapore, 27–30 January 2004; IEEE Press: Piscataway, NJ, USA, 2004; pp. 186–191.
13. Kwong, J.; Ramadass, Y.; Verma, N.; Chandrakasan, A. A 65 nm Sub-Vt Microcontroller with Integrated SRAM and Switched Capacitor DC-DC Converter. *IEEE J. Solid-State Circuits* **2009**, *44*, 115–126.
14. Bol, D.; de Vos, J.; Hocquet, C.; Botman, F.; Durvaux, F.; Boyd, S.; Flandre, D.; Legat, J. SleepWalker: A 25-MHz 0.4-V Sub-mm^2 7-µW/MHz Microcontroller in 65-nm LP/GP CMOS for Low-Carbon Wireless Sensor Nodes. *IEEE J. Solid State Circuits* **2013**, *48*, 20–32.