All-digital energy-constrained controller for general-purpose accelerators and CPUs

Davide Zoni*, Luca Cremona and William Fornaciari, *Senior Member IEEE*

***Abstract*—Considering the energy-cap problem in battery-powered devices, DVFS and power gating represent the de-facto state-of-the-art actuators. However, the limited margin available to reduce the operating voltage, the impossibility to massively integrate such actuators on-chip. together with their actuation latency force a revision of such design methodologies. We present an all-digital architecture and a design methodology that can effectively manage the energy-cap problem for CPUs and accelerators. Two quality metrics are put forward to capture the performance loss and the energy budget violations. We employed a vector processor supporting 4 hardware threads as representative usecase. Results show an average performance loss and energy cap violations limited to 2.9% and 3.8%, respectively. Compared to solutions employing the DFS actuator, our all-digital architecture improves the energy-cap violations by 3x while maintaining a similar performance loss.**

***Index Terms*—Energy-constrained design, low power, digital design, RTL design, multi-core, power management.**

I. Introduction

Energy efficiency is the major obstacle to the evolution of computing devices, including those for the mobile market. Unfortunately, the performance-constrained strategies employed in servers and datacenters to minimize the energy consumption under performance constraints coming from the applications, cannot be readily applied to mobile platforms. In fact, the latter exhibit an opposite optimization problem, i.e., maximizing the battery life, hence making the *energy-constrained* optimization methodologies the de-facto choice to offer optimized performance while complying with the energy budget. In the last decade, *energy-constrained* optimizations have been extensively investigated in mobile platforms. Some examples are leveraging standard actuators, e.g., Dynamic Voltage and Frequency Scaling (DVFS) [1], [2] and power gating [3], [4], also implementing ever more efficient microarchitectures, e.g. ARM big.LITTLE. However, the technology scaling dramatically reduces the benefit of the DVFS actuator, that, in addition, can be hardly integrated on-chip and show a too large actuation latency to sustain the energy-constrained methodologies for battery-powered devices.

Main contributions - This paper describes a fresh view on the design of a *control-based, all-digital, energy-constrained* management scheme for general purpose processors and accelerators, leveraging all-digital actuators and monitors. The proposed solution has been compared against a traditional Dynamic Frequency Scaling (DFS) actuator considering a representative vector CPU with hardware multi-threading support usecase. We report three contributions with respect to the state-of-the-art. First, the energy constraint can be configured through dedicated memory mapped registers, thus decoupling the low-level actuation details specific to each device from the high-level policy. Second, we modify the halt mechanism used for in-circuit debug of programmable CPUs to obtain a new, all-digital actuator with a fast actuation dynamic, i.e. few clock cycles, low area and easily implemented on-chip. Third, we propose two quantitative metrics to capture the performance loss and the energy cap violations. We experimentally prove that such metrics can be more informative than the total energy consumption one when dealing with energy-cap problems.

Background - The all-digital energy-constrained controller is a discrete time-domain component that, thanks to its standard-cell implementation, ensures fine granularity and scalability. Such solution is becoming a standard approach to control the dynamic power and thus the energy consumption of a device when the voltage cannot be further reduced. It is made of three digital parts; *i)* online power monitor, *ii)* actuator, and *iii)* energy-constrained controller.

Online power monitor - The state-of-the-art reports several digital implementations of the online power monitor to read out, at run-time, the estimated power consumption of the computing platform. The execution time of the application is split in equally fixed-length time-windows and, for each of them, a power estimate, representing the average power consumption spent, is sampled. Such indirect estimate methods first identify a power model of the target and then feed it with the system statistics to periodically deliver a power estimate. We note that all the indirect estimate methods leverage the relationship between the power consumption and an abstraction of the internal switching activity of the circuit. [5] proposes a software version of the digital power monitor that uses the architectural performance counters, while [6] proposes to model the power state of the system from the toggle count, i.e., switch between logical level "0" and "1" in CMOS technology, of selected internal gate-level signals monitored at run-time by means of binary counters. This work focuses on the digital controller design and relies on the online power monitor design proposed in [6].

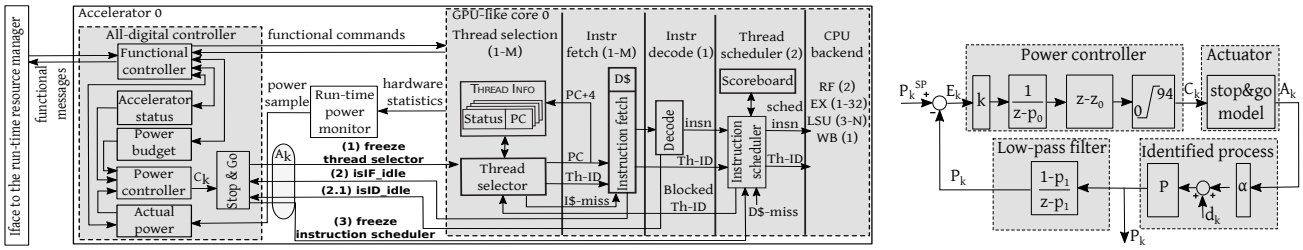
Digital actuator - To overcome the above-mentioned limitations of the DVFS, we propose a cheap, all-digital power actuator that is highly integrable on-chip and guarantees low actuation latency. We also compare our solution with the standard Dynamic Frequency Scaling (DFS) mechanism.

* Corresponding and leading author: Davide Zoni

Davide Zoni, Luca Cremona and William Fornaciari were with the Dipartimento di Elettronica informazione e Bioingegneria, Politecnico di Milano, Milano, Italy. E-mail: {davide.zoni,luca.cremona,william.fornaciari}@polimi.it

The work was supported by the EU H2020 "RECIPE" Project (801137)

Manuscript received October 2018; revised YYY; accepted YYY.



(a) Architectural view of the energy controller and the front-end of the considered accelerator (b) Block diagram of the energy controller

Fig. 1: Architectural view of the proposed all-digital power-cap system and block diagram of the closed-loop controller.

Digital controller - Traditionally, the energy-constrained solutions are global components placed either into the OS or into global resource managers [7]. However, the complexity of current embedded systems critically impacts on the effectiveness of the actuation due to both compute and communication latencies. Our solution delivers a fast and cheap digital controller-actuator scheme suitable for both host CPUs and accelerators, with a message interface to interact with OSs or run-time managers for integration in complex platforms.

II. All-digital energy-constrained design

Figure 1a details the proposed all-digital controller. We note that the proposed energy-cap methodology actually constrains the average power consumption that is sampled at a fixed time interval, i.e., once for each time window (k), by the online power monitor. The *Functional controller* interacts with the global resource manager or the Operating System (OS) by means of *functional messages*. Such messages are parsed and actuated via the *functional commands* interface. The global resource manager and the OS can neither directly observe nor control the accelerator. They can only read out its status and dynamically assign a power budget through the memory mapped *Accelerator status*, *Actual power* and *Power budget* registers. The *Accelerator status* register is also updated when a new application is kicked off or terminated. The *Functional controller* delivers the same interface to communicate with the controller regardless of whether the latter controls either the host CPU or the accelerators.

The *Power controller* takes three inputs, *i)* the status of the computing resources, *ii)* the power budget, and *iii)* the actual power consumption, to produce the actuation signal (C_k), for each time window k, used as the input to the *Stop&Go* actuator. We designed the *Stop&Go* actuator tailored to the considered GPU-like accelerator [6] by using a 3-phase protocol with a stop and resume latency limited to 4 clock cycles. To stop the processing elements, the protocol pauses the *Thread Selector* and observes when the fetch stage (*Instr fetch*) is empty - at most 2 clock cycle delay. The *Instr Decode* is then stalled during the third clock cycle. Finally, the *Instruction scheduler* is blocked in the subsequent clock cycle, i.e., forth one, since the ID/EX inter-stage registers are single slot queues. In contrast, the *Instruction scheduler* features an 8-instruction queue for each one of the four hardware-implemented threads. We avoid stalling the pipeline stages after the *Instruction scheduler* to allow the commit of the on-the-fly instructions completed during the stop period of the CPU. This fact motivates negligible energy consumptions during the stop periods, i.e., less than 0.0001% of the energy consumption of the considered CPU.

A. Model identification and control design

Figure 1b depicts the discrete time-domain closed-loop of the proposed power controller. It implements a programmable control-based PID and a low pass filter, and it actuates on the process (P) that models the power consumption of the system. The set point (P_k^{SP}) represents the energy-cap dynamically imposed by the OS or the resource manager for each time window k. The actual power consumption (P_k) is backward propagated via a low pass filter to generate the error signal (E_k) which is then fed to the controller. The control signal C_k drives the *Stop&Go* actuator by directly impacting the controlled variable, i.e., power state of the system, via the actuation variable (A_k). The d_k quantity models the non-controllable disturb on power consumption.

Quality metrics. We define two quality metrics capturing the performance loss and the energy cap violations for energy-constrained optimizations. Both metrics are piece-wise defined for constant set point values and are limited between 0 and P^{SP}. The *overflow metric* (OVF) sums up the magnitudes of the energy budget violations (see Equation 1). The *efficiency metric* (EFF) measures the performance loss induced by a non-zero control action when the power consumption is below the assigned energy budget (see Equation 3). For each epoch k, Gap_k (see Equation 2) measures the minimum positive value between two quantities: *i)* the difference between the power cap and the actual power ($P_k^{SP} - P_k$), and *ii)* the proportional increase in the actual power in case the control action is equal to zero, i.e. ($\frac{TP*P_k}{TP-C_k} - P_k$). We consider a temporal resolution (TP) of 100 clock cycles for the entire system.

$$OVF = \frac{1}{P^{SP}} * \frac{\sum_{k=1}^{\#samples} max(0, P_k - P^{SP})}{\#samples} \quad (1)$$

$$Gap_k = min\Big((P_k^{SP} - P_k), (\frac{TP * P_k}{TP - C_k} - P_k)\Big) \quad (2)$$

$$EFF = \frac{1}{P^{SP}} * \frac{\sum_{k=1}^{\#samples} max(0, Gap_k)}{\#samples} \quad (3)$$

Model identification of the process. In the literature, the power consumption is modeled by using an integral formulation [8]. We experimentally found that such description is accurate if the temporal resolution of the sampling process

is in the order of milliseconds and above. Starting from our experimental evaluation, we propose a novel power model to design the power controller. Such power model is defined in Eq. 4, it works at microsecond time resolution granularity and it corresponds to the *Identified process* in Figure 1b.

$$P_k = d_k + \alpha \times A_{k-1} \tag{4}$$

The *stop&go* command (A_k) represents the controllable input, while the d_k term is the uncontrollable disturb of the system that represents the power variations following any event that is not a direct consequence of our actuator. The α parameter in Eq. 4 models the relationship between the controllable input and the power consumption. The model of the process belongs to the family of deterministic auto-regressive ones with exogenous input (ARX) [9]. We employed standard identification techniques to determine α by considering different benchmarks (see Table I).

Controller design. The controller is fed with the error signal (E_k) and it outputs the control signal for the actuator. To offer a very fine grain actuation, we fix at 2us the time resolution for our system. This corresponds to 100 clock cycles considering a 50MHz synthesized clock frequency. We employ the low pass filter to smooth the power consumption measure that, from the experimental data at 2us time resolution, was found to contain high frequency components. Experimental results have led to the selection of p_1 equal to 0.32. The output signal of the controller (C_k) is limited between 0 and 94 of the total 100 clock cycles epoch since 6 clock cycles are used at the beginning of each time window to compute and apply the actuation. The actuator takes 4 clock cycles to stop the pipeline and it is modeled as a pure delay component. The low pass filter and the power controller take one clock cycle each to compute their output values. We noted that the stop period, if any, is imposed at the beginning of each time window starting from cycle 7, i.e., immediately after the actuation is made available by the controller.

TABLE I: Experimentally identified α values for different benchmarks and number of threads. For each level of parallelism, i.e., number of threads, the selected k value (blue boxes) ensures asymptotic stability of the closed-loop system.

Benchmark	1 thread		2 threads		4 threads	
	α	k_{max}	α	k_{max}	α	k_{max}
conv layer	0.67	6.17	0.96	4.31	1.33	3.11
fft	0.74	**5.59**	1.28	**3.23**	1.97	**2.10**
fir	0.67	6.17	1.14	3.63	1.74	2.38
gauss	0.70	5.90	1.18	3.50	1.82	2.27
matrix mul fl	0.68	6.08	1.09	3.79	1.74	2.38

The controller implements a PID control-based transfer function for which the proportional, the integral and the derivative parts can be configured through memory mapped registers. We present a controller design to balance the EFF and OVF metrics. In particular, we define an integral control to ensure a zero error at steady state ($p_0 = 1$) and we set the derivative contribution to ensure enough responsiveness of the controller to the power variations ($z_0 = 0.75$). Then, we determine k to ensure the closed-loop stability through the root locus analysis method. The allowed k values are constrained by the α value that varies depending on the benchmark. We conservatively choose k to ensure the closed-loop system stability. Table I reports the sets of α and k values for each combination of benchmark and number of threads. For the sake of validation, we identify three clusters for α depending on the number of threads. Thus, we select a k value for each cluster, also assuming that such value is set in the controller anytime a new application starts.

III. Experimental results

We validate the proposed all-digital controller by considering an in-order GPU-like processor featuring an 8-stage pipeline with 4 hardware threads and a vectored integer and floating point architecture with 16 parallel functional units each [6]. The complete system is made of the accelerator and the global resource manager. These are connected by means of a point to point Wishbone compliant link that features 32-bit address and data buses, and that allows the communication with an instance of the BBQ [7] run-time resource manager.

Implementation details: timing, area and overheads. Starting from the SystemVerilog description of the entire system, we employ Vivado 2018.2 for the synthesis and for the place and route (PAR) passes targeting a *Xilinx Artix 7 100t* FPGA chip. At the considered operating frequency of 50 MHz the accelerator has a slack of +1.46 ns. In contrast, the low pass filter and the power controller are not limiting the operating frequency since they show a slack of +18.77 ns and +12.39 ns, respectively. Compared to the entire system, the controller has an area overhead of 0.44% (LUT), 0.11 (FFs) and its power overhead is limited to 0.9%. The low pass filter and the controller use cheap shift and add operations, in place of complex and high latency floating point operations. The low pass filter and the controller take 1 clock cycle each to compute the corresponding output signal. The *Stop&Go* actuator takes 4 clock cycles to stop or to resume the pipeline. Its actuation is limited between 0 and 94. For the sake of comparison, we also implement a DFS actuator using the *Xilinx* MMCM and resynchronization blocks [10]. The use of a DFS is justified by the impossibility to effectively scale the voltage [11]. The DFS takes 60 clock cycles to lock to the new frequency and we modeled it in the control-theoretical framework as a pure delay [10]. Such DFS delay limits the impact of the actuation to less than 38 cycles for each epoch, with negative effects on both EFF and OVF. Note that such degradation is due to the latency gap between the application and the actuation dynamics, i.e., applications can trigger fast power spikes.

Overflow and efficiency results. We used 5 WCET [12] applications reported in Table I and results are collected by simulating the post-PAR netlist of the entire design. For each application, we presented three implementations by using a different number of threads, i.e., 1, 2 and 4, to completely stress the computing platform. Figure 2 reports the quantitative results for the EFF and OVF metrics defined in Section II as well as the standard total energy metric by considering the same system employing either the DFS or the *Stop&Go* actuators. Results consider a static scenario, where each application is executed to completion and the

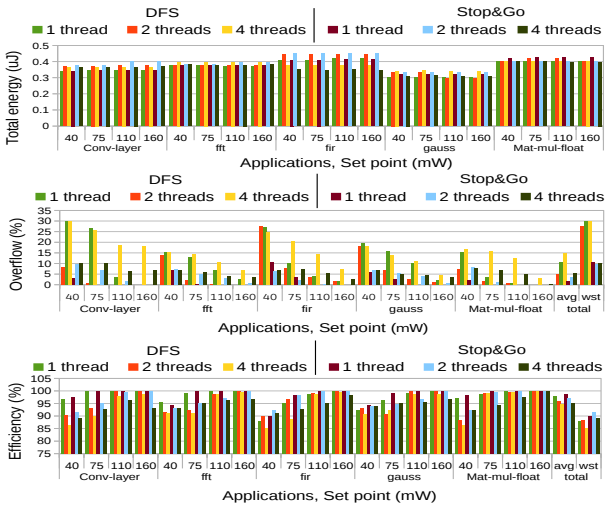


Fig. 2: Total energy, efficiency and overflow results considering different benchmarks and set point values.

set point is constant. We selected four representative set points, i.e., 40mW, 75mW, 110mW and 160mW, starting from the collected power consumption of each instance in the application set. The controller employing the *Stop&Go* actuators shows an average efficiency and performance loss of 95% and 5.4%, respectively. The worst case is the *conv layer* application using 4 threads which shows an efficiency limited to 89% and a performance loss equal to 10.2%. The reduced efficiency and higher overflow of the worst case depend on the characteristics of the application and not to an incorrect design of the controller. In fact, *conv layer* uses 4 threads and remains stable up to a k equal to 3.11, while k equal to 2.11 is required to ensure a theoretical closed-loop stability to all the 4-thread applications (see Table I). The smaller k determines a lower actuation strength for *conv layer* and a consequent small degradation of the quality metrics. Differently, the controller employing the DFS actuator shows a slightly reduced EFF metric, i.e., less than 1% penalty. However, the use of DFS negatively affects the OVF metrics that degrades between 2.5x and 3x on average compared to the same scenario using the *Stop&Go* actuator. We also note that, in the energy-constrained scenarios, the EFF and OVF metrics are more informative that the standard total energy or Energy-Delay-Product (EDP) ones. Considering each benchmark, Figure 2 shows that the energy consumption remains stable for each set point value without reporting any energy violation. In contrast, the EFF and OVF metrics highlight the quality of the time-wise energy-graph for each application and set point value.

Figure 3 shows the execution of two subsequent applications while the set point (red line - P^{SP}) dynamically changes. The y axis reports the power consumption in milliwatts for the two systems by employing either the DFS (blue line - P_{DFS}) or the *Stop&Go* (yellow line - $P_{Stop\&Go}$) actuators. Results reported in Figure 3 highlight a slightly better EFF value when using the *Stop&Go* (worst case 95.7%) in place of the DFS (worst case 95.3%). In contrast, the use of the DFS severely affects the OVF metric that is degraded up to 3x with respect to the system employing the *Stop&Go* mechanism. Such degradation is due to the high latency of the DFS that prevents to timely react when the power consumption exceeds the set point.

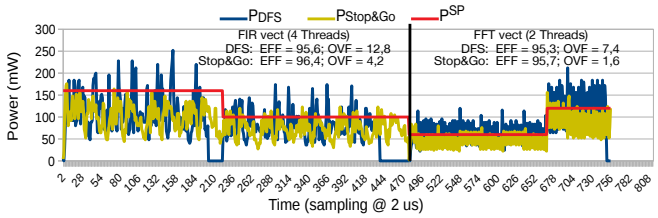


Fig. 3: Applications are entering in sequence and the power cap is dynamically changed by the run-time resource manager.

IV. Conclusions

We presented an all-digital energy-constrained controller and a low-latency actuator for general purpose embedded platforms. The controller is fully programmable via memory-mapped registers, it implements a PID and a low pass filter, which make our design compatible with existing online power monitoring schemes. In addition to EDP and total energy consumption metrics, we introduced two quality metrics to capture the performance loss (EFF) and the intensity of the energy cap violations (OVF). We demonstrated that the latter is better suited to capture the information required to analyze energy-cap scenarios. Results on a representative accelerator show an average performance loss and energy budget violations limited to 2.9% and 3.8%, respectively. The DFS actuator, coupled with the proposed energy controller, shows similar EFF values but reports an OVF degradation between 2.5x and 3x compared to the proposed actuator.

References

[1] R. Begum, D. Werner, M. Hempstead, G. Prasad, and G. Challen, "Energy-performance trade-offs on energy-constrained devices with multi-component dvfs," in *2015 IEEE International Symposium on Workload Characterization*, Oct 2015, pp. 34–43.

[2] R. Xu, D. Mossé, and R. Melhem, "Minimizing expected energy consumption in real-time systems through dynamic voltage scaling," *ACM Trans. Comput. Syst.*, vol. 25, no. 4, Dec. 2007.

[3] D. Zoni, L. Colombo, and W. Fornaciari, "DarkCache: Energy-Performance Optimization of Tiled Multi-Cores by Adaptively Power-Gating LLC Banks," *ACM Trans. Archit. Code Optim.*, vol. 15, no. 2, pp. 21:1–21:26, May 2018.

[4] D. Zoni, A. Canidio, W. Fornaciari, P. Englezakis, C. Nicopoulos, and Y. Sazeides, "Blackout," *J. Parallel Distrib. Comput.*, vol. 104, no. C, pp. 130–145, Jun. 2017.

[5] M. J. Walker, S. Diestelhorst, A. Hansson, A. K. Das, S. Yang, B. M. Al-Hashimi, and G. V. Merrett, "Accurate and stable run-time power modeling for mobile and embedded CPUs," *IEEE TCAD*, 2017.

[6] D. Zoni, L. Cremona, A. Cilardo, M. Gagliardi, and W. Fornaciari, "Powertap: All-digital power meter modeling for run-time power monitoring," *MICPRO*, vol. 63, pp. 128 – 139, 2018.

[7] P. Bellasi, G. Massari, and W. Fornaciari, "Effective runtime resource management using linux control groups with the barbequertrm framework," *ACM TECS*, vol. 14, no. 2, pp. 1–17, 2015.

[8] K. Ma, X. Li, M. Chen, and X. Wang, "Scalable power control for many-core architectures running multi-threaded applications," in *ISCA*, June 2011, pp. 449–460.

[9] L. Ljung, *System identification:theory for the user*. Prentice-Hall, 1987.

[10] Xilinx, "7 Series FPGAs Clocking Resources," https://www.xilinx.com/support/documentation/user_guides/ug472_7Series_Clocking.pdf.

[11] T. R. da Rosa, V. Larra, N. Calazans, and F. G. Moraes, "Power consumption reduction in mpsocs through dfs," in *SBCCI*, 2012.

[12] J. Gustafsson, A. Betts, A. Ermedahl, and B. Lisper, "The malardalen wcet benchmarks: Past, present and future." in *WCET*, ser. OASICS, B. Lisper, Ed., vol. 15, 2010, pp. 136–146.