Panoptic DVS: A Fine-Grained Dynamic Voltage Scaling Framework for Energy Scalable CMOS Design

Mateja Putic[#], Liang Di[*1], Benton H. Calhoun[#], and John Lach[#]
[#]*University of Virginia,* [*]*Intel Corporation*
{mp3t, bcalhoun, jlach}@virginia.edu, liang.di@intel.com

***Abstract*— The energy efficiency of a CMOS architecture processing dynamic workloads directly affects its ability to provide long battery lifetimes while maintaining required application performance. Existing scalable architecture design approaches are often limited in scope, focusing either only on circuit-level optimizations or architectural adaptations individually. In this paper, we propose a circuit/architecture co-design methodology called Panoptic Dynamic Voltage Scaling (PDVS) that makes more efficient use of common circuit structures and algorithm-level processing rate control. PDVS expands upon prior work by using multiple component-level PMOS header switches to enable fine-grained rate control, allowing efficient dithering among statically scheduled algorithms with sub-block energy savings. This way, PDVS is able to achieve a wide variety of processing rates to match incoming workload as closely as possible, while each iteration takes less energy to process than on architectures with coarser levels of rate control. Measurements taken from a fabricated 90nm test chip characterize both savings and overheads and are used to inform PDVS synthesis decisions. Results show that PDVS consumes up to 34% and 44% less energy than Multi-VDD and Single-VDD systems, respectively.**

I. Introduction

REDUCTION of energy consumption for CMOS circuit architectures continues to be an important area of research. Portable systems demand extended battery lifetimes, high performance systems have reached thermal limits due to increased power densities, and “green” computing initiatives have prioritized energy efficiency as a first class system metric. In recent years, the use of multiple supply voltage rails (Multi-VDD) has been introduced to enable systems with strict latency and throughput requirements to still reduce energy by assigning lower voltages to components performing non-critical operations. Header/footer switches are also widely employed to power-/ground-gate off components when they are not being used, which helps to reduce wasted leakage energy. Dynamic voltage scaling (DVS) has also become commonplace, enabling systems to adapt to dynamic workloads and battery availability by operating at the slowest rate while still meeting performance requirements. Each of these techniques provides benefits individually, but no framework currently exists that combines all of them to approach the limits of energy efficiency.

For systems that process dynamic incoming workloads, the lowest energy operating point is achieved when the processing rate matches the rate of the incoming workload. Often the rate of the incoming workload is evaluated by the occupancy of an input buffer. The energy efficiency of DVS architectures is evaluated by the range of processing rates that they can achieve and by the latency and energy overhead of transitions between voltages. The *Panoptic Dynamic Voltage Scaling* (PDVS) architecture described in this work addresses both of these criteria through fine-grained control of processing rates.

PDVS adds *spatial* granularity by introducing header switches at the level of individual arithmetic components that select one from a small number of VDD rails. This allows the algorithm to dictate the processing rate and ultimately the energy use of individual operations. With this level of control, the slack of individual operations within a schedule can be exploited to lower the energy of execution of a single, static schedule. The *temporal* granularity of PDVS enables fast switching among static schedules of various latencies so that the processing rate can be matched as closely as possible to the dynamic workload. The greatest novelty of this approach is that the combination of low switching overhead together with fine grained header switches allows PDVS to meet a wide variety of processing rates, each with less energy per iteration, in comparison to similar techniques, such as traditional DVS or Multi-VDD. In effect, PDVS combines Multi-VDD, header switches, and DVS in a framework in which the whole is greater than the sum of the parts and the limits of energy efficiency can be approached.

These benefits are provided at only the cost of structures that are already commonplace in modern CMOS architectures: header switches and multiple supply voltage rails. Typically, header switches are used to reduce the leakage energy of idle components, and multiple supply rails are used to provide the aforementioned dynamic energy savings in non-critical components. PDVS uses them together (one header switch per voltage rail per component) to dynamically select each component's voltage. This *sub-block* voltage control enables PDVS to achieve the same schedules and dynamic energy as Multi-VDD – results in this paper show up to 16% total energy savings in comparison to Single-VDD in a fixed rate system – with fewer components due the ability to use the same component to perform operations at different voltages in the schedule. This header switch approach to voltage control also provides the rapid, energy-efficient transitions in processing rate, enabling local voltage dithering (LVD) [2] and rapid adjustments to dynamic workloads to exceed the energy savings of traditional Single-VDD DVS [3] by up to 44% in the workloads evaluated. The benefits of PDVS are most prominent in systems requiring the implementation of time-wise mutually exclusive functions on the same hardware. For all results, the energy and delay overhead of voltage transitions and the operation delay overhead imposed by the header

[1] Work performed while author was at the University of Virginia.

978-1-4244-5028-2/09/$25.00 ©2009 IEEE

switches were included. These overheads were measured from a test chip manufactured in a 90nm process and were used to inform the PDVS synthesis decisions. The scheduling strategy employed to minimize iteration energy in this work is heuristic in nature and may benefit from several related works that address this topic formally [17-20].

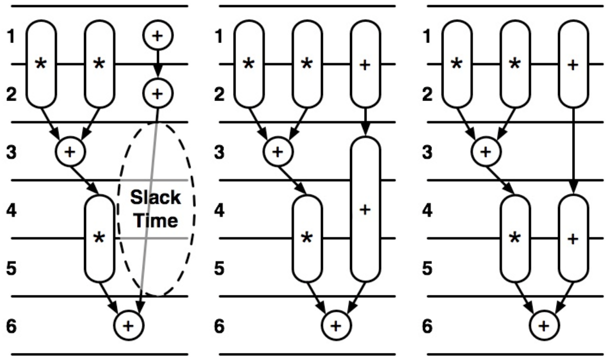
This paper is organized as follows. Section 2 discusses related work, and Section 3 introduces the PDVS architecture and design methodology. Section 4 presents the overheads measured from the test chip, which were used to identify the break-even times for voltage transitions. The results detailed in Section 5 reveal the benefits of PDVS in single-rate, multi-rate, and multi-function usage scenarios. Section 6 concludes and suggests future work.

II. Related Work

The continued increase in leakage current as a component of total system power due to down-scaling of CMOS technology has been a major factor in the proliferation of power gating switches in modern design [8,9,11,14]. These switches allow systems to switch off or “gate” leakage current when a circuit block is not being used. The total savings achieved by gating is only substantial when leakage power dominates total power.

Multi-VDD architectures incur the area overhead of routing three voltage rails on chip but potentially provide significant energy savings by enabling components to operate at different speeds. Consider the dataflow graph (DFG) in Figure 1 implemented on a synchronous dataflow architecture, where each control step (c-step) corresponds to one clock cycle. Assuming a multiplication and an addition at the highest source voltage (VDDH) take two and one c-steps, respectively, the DFG can be executed in a minimum of six c-steps. However, the two additions not on the DFG's critical path could be executed at lower voltages and consume less energy per operation without affecting the DFG latency. The minimum dynamic energy schedule is shown in Figure 1b, but this requires three VDDs and three adders (the VDDH adder used in c-step 3 can be re-used in c-step 6). Figure 1c shows an alternate schedule that sacrifices some dynamic energy relative to the optimal schedule but can be implemented with only two VDDs and two adders.

DVS lowers the supply voltage [1,15] to reduce active power dynamically when reduced processing rates are possible. This approach usually applies to the entire chip (global DVS) using an external dc-dc converter to change VDD at the power pads [10,12]. Some approaches move the dc-dc converter and controller onto the chip itself but still modulate the voltage of the whole architecture [7]. Delays of voltage transitions using a dc-dc converter are on the order of tens to hundreds of microseconds [1,6,16], meaning that it cannot keep up with rapid changes in workload and limits the number of processing rates that can be achieved. A fine-grained header switch implementation similar to PDVS uses synthesis techniques to minimize energy and delay but does not use IC measurements for VDD-switching overhead and does not compare results to static multi-VDD [17]. While global DVS allows for flexible energy scalability, the magnitude of energy and delay overheads of global supply switching mean that the frequency of VDD changes must be low so that power savings from periods spent at the lower voltage offset the overhead.



(a) Adder idles during slack in time steps 3-5.

(b) Filling slack lowers active energy, but requires one more adder.

(c) An alternative solution saves less energy bus uses one adder.

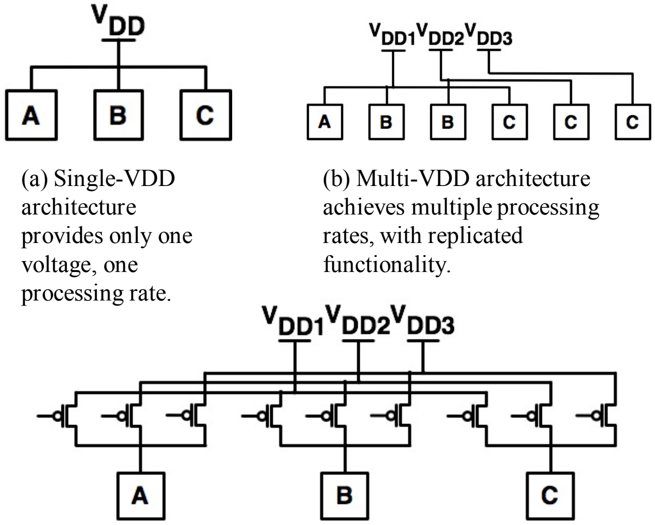
Figure 1. Slack-fill for static sub-block energy savings.

III. Panoptic DVS Framework

Previous works have shown that the energy-optimal processing rate for a block of data is equivalent to the average workload of that data [3,7]. While the PDVS approach does not by itself determine this rate, it provides an architecture that can scale processing rates of individual operations correspondingly. PDVS can be used in concert with existing approaches for workload estimation [21,22] to implement a fully dynamic, application responsive energy scalability. This section details the PDVS architecture and how it is able to approach the energy-optimal processing rate at multiple functional granularities more closely than existing techniques.

To achieve maximum savings of active energy, work should be processed as slow as possible while still meeting performance requirements, even when finishing early and entering a low power sleep state is an option. This applies at multiple levels of functional granularity. At the finest level of design, the workload of the non-critical path in Figure 1 should be performed as slowly as possible without changing the critical path length. More coarsely, if a variable number of DFG iterations must be performed in a fixed amount of time, then a processing rate should be chosen such that the maximum amount of available time is used. This reduction in processing rate corresponds to a reduction in supply voltage, achieving quadratic energy savings, in contrast to only linear savings when power gating is used.

An architecture that provides only one power supply, such as Single-VDD shown in Figure 2a, has limited flexibility. All components must operate at the same voltage, and DVS must alter the voltage on a large capacitance rail. As discussed in Section 2, Multi-VDD (Figure 2b) provides the opportunity for sub-block energy savings when operations have timing slack, but the permanent assignment of voltages to components prevents a single physical component from executing operations at different rates, often requiring additional components to achieve the minimum energy schedules. In addition, Multi-VDD suffers from similar DVS inefficiencies as Single-VDD. As shown in Figure 2c, PDVS uses components that are each able to switch between several available supply voltages, enabling a single arithmetic component to implement multiple sequential operations within a single algorithm iteration at various processing rates. With this fine spatial voltage control granularity, PDVS is able to achieve the same schedules as Multi-VDD, but often with fewer components. In addition, the header switches provide fine temporal voltage control granularity, as components are able to quickly switch processing rates for both intra-DFG changes and coarser-grained changes in workload. This section explores PDVS's capabilities in three contexts: achieving reduced energy (and area) schedules given a single rate of operation, adapting to dynamic workloads and enabling LVD under multiple rates of operation, and implementing energy- and area-efficient systems capable of executing multiple time-wise mutually exclusive functions.



(a) Single-VDD architecture provides only one voltage, one processing rate.

(b) Multi-VDD architecture achieves multiple processing rates, with replicated functionality.

(c) PDVS architecture proposed in this work, header switches enable dynamic scaling of processing rates of individual components.

Figure 2. CMOS architectures used in comparison.

A. Single-Rate Operation

Consider again the DFG in Figure 1a. Existing Single-VDD and Multi-VDD high-level synthesis algorithms are able to determine minimum energy schedules given a latency (and often area) requirement. With Single-VDD, Figure 1a already provides the optimal schedule and only requires two multipliers and one adder. As discussed in Section 2, Multi-VDD could implement either schedule in Figures 1b and 1c based on energy vs. area tradeoffs. PDVS implements the schedules in Figures 1b and 1c with two and one adders, respectively. To achieve the latter, the adder must switch quickly between the middle voltage level (VDDM) and the low voltage level (VDDL), and the fine-grained temporal voltage scaling capabilities of PDVS make this possible. However, the additional energy required to make such voltage transitions does increase the total energy of this implementation, but Section 4 shows that this overhead is small and that the reduced leakage provided by fewer components is a significant benefit.

B. Multi-Rate Operation

Now consider the dynamic workload scenario in which a variable number of DFG iterations (reflected as the normalized workload axis in Figure 3) must be performed in a given amount of time. Without any DVS capabilities, an architecture would be forced to operate at the maximum rate and enter a low-power sleep mode when it finishes early. As shown in Figure 3, this provides only linear energy savings (e.g., a workload that is half of the maximum is executed with half of the maximum energy). The sub-block energy savings provided by Multi-VDD and PDVS in the previous subsection are reflected by the downward shift of this curve, but the lack of DVS would still provide linear savings for reduced workloads.

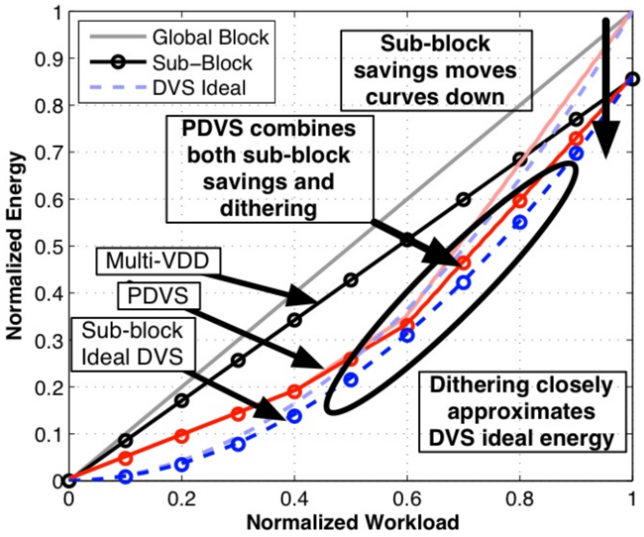


Figure 3. PDVS achieves sub-block energy savings and closely tracks ideal DVS with dithering.

DVS with either Single- or Multi-VDD provides the ability to adjust the processing rate based on the workload but with some limitations. If the voltage can be selected from a continuous range and there is no energy or delay overhead to make the voltage transition (DVS ideal), quadratic energy savings can be achieved, with or without the sub-block savings based on the number of VDDs. Voltage dithering [7] provides a nearly continuous average processing rate by switching between quantized rates after a percentage of iterations determined by the workload. However, the long settling time for a dc-dc converter limits dithering.

PDVS combines the benefits of sub-block savings with practical dithering to approach the energy-optimal processing rate. Instead of changing the voltages on large rails, header switches provide fast and efficient voltage transitions for each component, enabling the system to effectively switch processing rates by implementing schedules with variable latencies.

C. Multi-Function Operation

The benefits are PDVS are greatest with single-rate and multi-rate operation profiles are required across multiple functions. For such a system, an architecture must be capable of implementing any schedule from a given set of functions. The architecture that minimizes dynamic energy would simply be the full implementation of each function individually, but that would incur significant area overhead. If the functions are time-wise mutually exclusive, a flexible architecture that maps each function onto the same components is possible. For such an architecture, the benefits of PDVS over Single-VDD are the same as in the previous two subsections. However, the benefits over Multi-VDD are significantly larger than before, as disparate functions may require different numbers of components at each voltage level. This results in either a significant area overhead or the inability of Multi-VDD to implement the minimum energy schedule. Due to its flexibility, PDVS is still able to do so with only the maximum number of components required for a single function. As a result, PDVS is able to save both dynamic energy (due to more energy-efficient schedules) and area and static energy (due to fewer components) over Multi-VDD.

IV. Overhead of PDVS

Based on the analysis in the previous section, PDVS can approach the energy-optimal processing rate more closely than any previous architecture. However, the flexibility of PDVS comes with an overhead. For example, PDVS can implement the schedule in Figure 1c with only one adder, but there are finite energy and delay costs to switching between voltages, which may make this schedule energy-inefficient or un-implementable within the given latency constraints. The magnitudes of the voltage switching energy and delay overheads must be known in order to derive the conditions under which switching to a lower voltage is efficient. These conditions can then be used to inform architecture-level synthesis algorithms. We fabricated a test PDVS architecture in 90nm bulk CMOS to accurately measure these overheads. The fabricated platform served as a testbed for investigating the merits of fine-grained header switches in low power design. A more involved implementation of this architecture would meet the physical design challenges of multiple supply routing and packaging and was not the focus of this study, but the physical design strategy of this architecture is documented in [4].

Figure 4 shows the architecture of the PDVS test chip. The design uses a 32-bit Kogge-Stone adder and a 32-bit Baugh-Wooley multiplier, with only the least significant 32 bits of the product fed back. A linear-feedback shift register (LFSR) provides pseudo-random operands for testing. The output of each component feeds into a respective register that stores the result until the next active clock edge. Level converters interface outputs of the arithmetic components to the registers, which operate at the highest VDD.

Figure 5 shows a die photo of the chip. It is clear from this photo the relatively small area overhead introduced by the header switches and level shifter. Global voltage rails were sized sufficiently large so that the sizing of the header switches would dominate the switching delay and energy. While PDVS structures and additional supply rails increase design area, the PDVS solution is more area efficient in comparison to a Multi-VDD version that requires additional functional units to provide operation at the various voltages.

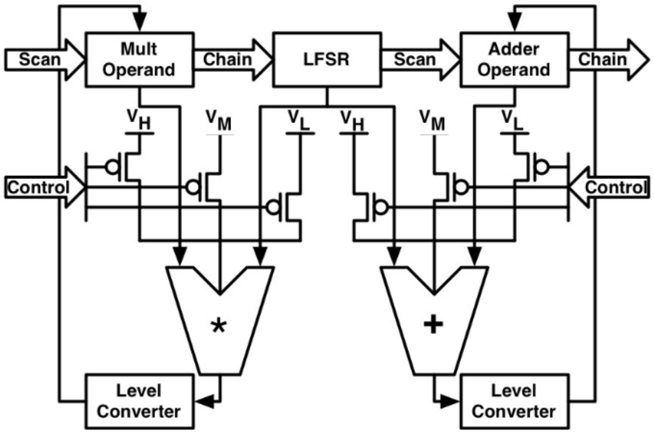


Figure 4. Top-level PDVS architecture implemented in 90nm bulk CMOS test chip.

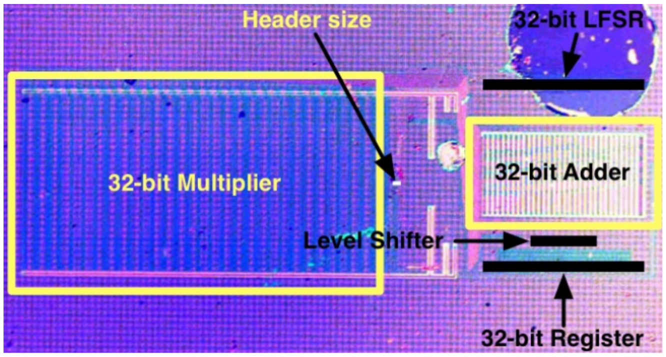


Figure 5. Die photo showing adder and multiplier.

A. Break-even Calculation

There are two sources of overhead in PDVS that influence how algorithms are scheduled on the architecture: energy and delay. The overheads result from additional header switches, the level converters, and the charging of virtual VDD nodes of components following a voltage switch. When a voltage switch occurs, the gates of header switches must be charged or discharged by control signals. The time to charge header gates results in a delay overhead, and the charge delivered to the switches and the virtual rail results in energy overhead. Both of these overheads are proportional to the header width, with smaller headers having lower voltage switching energy overhead but higher voltage switching delay and propagation delay penalty for the underlying component [4].

The delay of transitioning to a higher voltage is critical, as the virtual VDD rail must be charged up to the higher voltage. By properly sizing the header switches, the delay overhead can be reduced to less than one cycle [4]. Delay overheads factor into scheduling decisions by limiting voltage transition speed, and consequently, the amount of slack that can be saved within a schedule. As a result, an operation may not be able to run at the lowest possible processing rate, even though just enough slack time exists. However, this only occurs in schedules with very limited slack.

The point at which the energy overhead of switching is mitigated by processing at a lower energy rate is called the break-even time. An efficient transition occurs if the cost of switching to a lower energy rate, processing for a period of time, then switching back, is lower than remaining at the higher rate for the same amount of time. This concept is described by:

$$N \leq \frac{E_{ovh}}{E_H - E_L} \qquad (1)$$

where N is the number of clock cycles, E_{ovh} is the lumped, round-trip overhead energy, and E_H and E_L are the energy per cycle at the high and low voltage, respectively.

To measure energy overhead due to voltage switching on the test chip, headers were switched continuously between two voltages while the arithmetic components were idle so that the energy due to overhead transients would dominate. The switching frequency was made high enough to ensure that the overhead due to switching dominated.

B. Summary of Measured Quantities

Table 1 shows a summary of measured values for the fabricated 3-rail PDVS architecture. The voltage levels used in this study are chosen such that addition operations at each of the three levels complete in approximately integer multiples of each other, as the duration of one c-step corresponded to the critical path delay of the adder at 1.0V. Energies and delays (normalized to number of c-steps i.e. adder cycles) of multiplication operations are then determined for these voltages. Additional leakage energy was measured separately and was factored in based on the fraction of the cycle that the multiplier spends idle after execution is complete. This is shown in Table 1 as “Add’l Leakage per Op” for the multiplier. Since multiplier operations were normalized to adder operations, there is no additional leakage for the adder, because it finishes in exact multiples of adder cycles – hence the “Total Energy per Op” equals the “Active Energy per Op”.

The calculation for break-even time, based on these numbers and Equation 1, has shown that the time required to mitigate the energy cost of switching between any two voltages in the test architecture is less than the latency of a single operation for both the adder and the multiplier. This means the energy of completing one execution at a low voltage plus the round-trip energy of switching down and switching back up is less than one execution at the highest rate. The low energy overhead therefore justifies frequent down-switching in PDVS to take advantage of sub-block energy savings, dithering, and rapid changes in workload. The more often that a down-switch can be scheduled, the lower the average voltage that is used, and consequently, the lower the total execution energy.

Table 1. Summary of measured quantities.

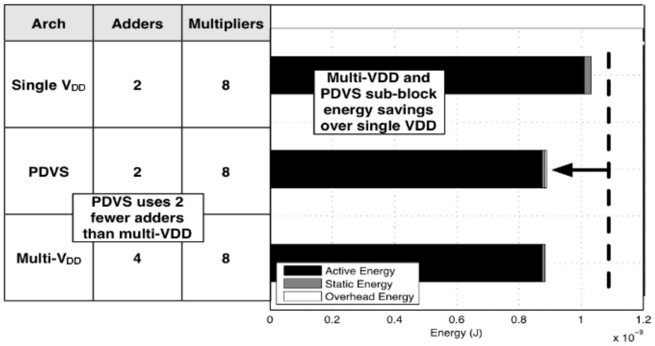
VDD (V)	1.0	0.78	0.68
	Adder		
Delay (ns)	0.43	0.85	1.28
Active Energy per Op (pJ)	2.43	0.78	0.49
Total Energy per Op (pJ)	2.43	0.78	0.49
Nearest Delay (Adder cycles)	1	2	3
Switching Overhead (pJ)	-	1.12	1.61
Breakeven Cycles	-	0.55	0.71
	Multiplier		
Delay (ns)	2.96	4.31	6.50
Active Energy per Op (pJ)	61.28	35.59	28.38
Add’l Leakage per Op (pJ)	0.0013	0.0107	0.0055
Total Energy per Op (pJ)	61.29	35.6	28.38
Nearest Delay (Adder cycles)	7	11	16
Switching Overhead (pJ)	-	6.71	9.67
Breakeven Cycles	-	1.22	1.39

V. RESULTS

We integrated the measured overhead data into a heuristic scheduling algorithm that reduces active energy by exploiting slack in existing algorithms, represented as DFGs. The algorithm determines when a voltage switch is appropriate for the PDVS system. Since each VDD switch consumes overhead energy, we used a heuristic binding algorithm to assign DFG operations to specific components with the goal of reducing the number of VDD switches for each component. We selected four typical DSP algorithms (ARLattice, DiffEQ, Elliptical filter, and FIR) as benchmarks to compare Single-VDD, Multi-VDD, and PDVS architectures across single-function single-rate, single-function multi-rate, and multi-function multi-rate scenarios. As described in Section 3c, PDVS provides the most substantial savings for multi-function multi-rate systems.

A. Single-Function Single-Rate

In single-function single-rate (SFSR) mode, we created distinct Single-VDD, Multi-VDD, and PDVS architectures for each of the DSP algorithms. We used our scheduling and binding algorithms to re-optimize each schedule at a given rate. Figure 6 shows energy and area requirements of the FIR benchmark for one SFSR point. Multi-VDD and PDVS save energy over Single-VDD by assigning operations that have slack in the DFG to lower voltage components. The cost of these energy savings is an increase in area (e.g. 5 to 6 extra adders for the FIR schedule). PDVS uses less area (and thus less leakage power) than Multi-VDD, however, because it can reuse some components (1 adder in this case) at different voltages at different times in the DFG. The slight energy overhead that comes from switching VDD to this component is included in the figure.



Arch	Adders	Multipliers
Single V_{DD}	2	8
PDVS	2	8
Multi-V_{DD}	4	8

Figure 6. Single-Function Single-Rate comparison for the FIR benchmark. PDVS gives the energy benefit of Multi-VDD with minimal energy overhead and less area.

The general trends from Figure 6 hold for different DFGs and for different choices of the single rate. As expected, schedules with the most slack allow for the most energy savings from sub-block VDD assignment. The PDVS architecture saves 16% versus Single-VDD for the FIR DFG. Very little slack in the DiffEQ DFG leads to essentially zero savings over Single-VDD. SFSR energy for PDVS often slightly exceeds Multi-VDD (e.g. by 0.85% at most for the DiffEQ DFG) because of the overhead of switching the voltage of individual components. Lower overall rate requirements generally produce less energy savings, since the difference between voltages in the Multi-VDD and PDVS architectures is smaller.

B. *Single-Function Multi-Rate*

In the single-function multi-rate (SFMR) configuration, we compared energy and area for the three architectures executing each single DFG but at variable rates. Single- and Multi-VDD architectures can only implement DVS by using the dc-dc converter to change a global VDD rail. (Multi-VDD could support DVS by fully replicating the DFG datapath at each voltage level, but we omit that case due to its large area overhead.) Because this takes so long (100s of microseconds), responses to changes of workload on that timescale can be effectively considered as discrete SFSR points. Our SFMR comparison assumes that changes in the required rate occur more rapidly.

Figure 7 shows the resulting comparison of energy versus variable rate for practical implementations given the fine timescale. Although the Multi-VDD architecture saves energy relative to Single-VDD due to the fine spatial granularity of sub-block voltage assignment, both single- and Multi-VDD architectures can only respond to rate changes by operating at the fastest rate and then shutting down. In contrast, PDVS combines sub-block energy savings with efficient energy response to rate changes, proving energy savings of up to 34% and 44% less energy than Multi-VDD and Single-VDD systems, respectively. The figure shows how this fine temporal granularity of PDVS allows for voltage dithering between the discrete processing rates, which causes the energy curve to closely approximate the ideal curve. When dithering for the most common workloads, between the lowest rate (VL) and the highest rate (VH), PDVS is always within 35% of the optimum PDVS energy curve across all benchmarks. Each schedule at the three rates is implemented with the same number of components as the corresponding schedule in SFSR.

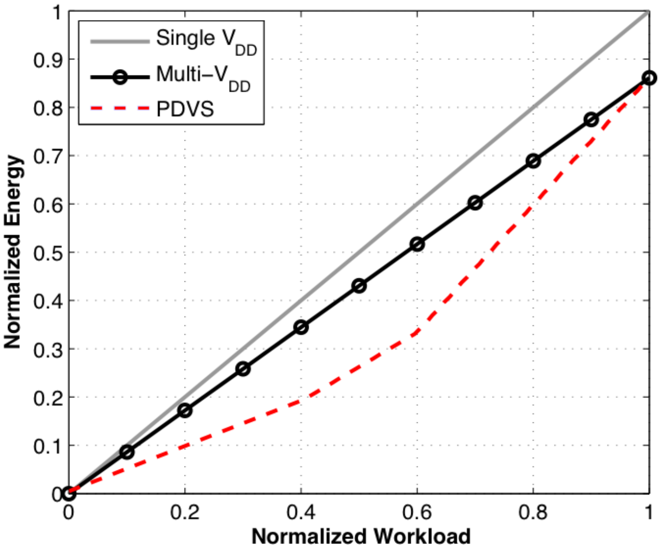


Figure 7. A comparison of energy budgets for single-function multi-rate operation across rapidly changing dynamic workloads for AR Lattice.

C. *Multi-Function*

Since most real-world systems implement a wide variety of temporally mutually exclusive functions, our final comparison examines multi-function architectures for Single-VDD, Multi-VDD, and PDVS that can run all four of the example DFGs. Each architecture has enough components to implement any individual DFG. Components that are unused during execution of a given DFG consume leakage energy (although PDVS - and any architecture with component-level header switches - could power gate unused components). Although the results are specific to the combination of functions, the flexibility of PDVS adds to the energy and area benefits of SFSR and SFMR.

In order to achieve the same optimal dynamic energy schedules for each function as were achieved in SFSR, Multi-VDD would require significant additional area, even when combining the functions into one architecture, as different functions have different numbers of components at each voltage. In fact, in many scenarios, the extra components were so significant that the additional leakage energy started to dominate the dynamic energy savings of implementing optimal energy schedules. As a result, the more efficient implementation for multifunction Multi-VDD in terms of total energy is to implement sub-optimal schedules with fewer components. Conversely, PDVS is able to change the voltages of components based on the function being executed and is therefore able to achieve the optimal energy schedules from SFSR without significant area overhead.

When multi-function is combined with multi-rate, PDVS represents the only architecture that both benefits from sub-block energy savings and is able to dither between processing rates in order to approach the optimal energy operation for dynamic workloads.

VI. Conclusion

In this study, we have presented the benefits and tradeoffs of using fine-grained header switches for a dynamic low energy CMOS architecture. With measurements collected from a test architecture fabricated in 90nm silicon, we have confirmed that the overhead energy and delay of switching processing rates is mitigated in less than one operation and one control cycle, respectively, as opposed to the large overheads associated with changing the voltages on chip-wide rails, as is done in traditional DVS. Evaluation benchmark algorithms were scheduled to minimize energy while taking these findings into account. PDVS introduces fine spatial granularity to achieve sub-block energy savings, lowering the total energy needed to complete a single iteration of a scheduled algorithm. With fine temporal granularity, PDVS is able to dither among static schedules to achieve rates that closely approximate the average incoming workload, with sub-block energy savings. The combination of these two strategies results in superior active energy performance as algorithm-level slack permits, superior static energy performance due to increased resource utilization enabled by multi-modal components, and implementations with fewer components than Single-VDD or Multi-VDD. PDVS essentially combines existing energy management techniques and commonplace circuit structures in such a way that the whole is greater than the sum of the parts, enabling the system to approach the optimal energy operation for both static and dynamic (in rate and function) workloads.

As static power continues to rise due to the down-scaling trend of CMOS technology, PDVS will provide a strong approach for mitigating the increase in energy loss due to leakage. The DFGs presented in this work are very similar, consisting only of multiplication and addition operations. Energy reduction opportunities for multipliers, the greatest consumer of energy, were limited due to lack of slack. The benefits of PDVS would increase for DFGs with heterogeneous operations. With more variability of slack, more opportunities would exist to save energy by scheduling heavy energy consumers at a lower voltage.

Acknowledgements

This work was supported in part by a UVA FEST award, the NSF under grant No. EHS-0410526, and the Southeastern Center for Electrical Engineering Education.

References

[1] T. Burd, T. Pering, A. Stratakos, and R. Brodersen. A dynamic voltage scaled microprocessor system. JSSCC, 35(11):1571–1580, 2000.

[2] B. Calhoun and A. Chandrakasan. Ultra-dynamic voltage scaling (UDVS) using sub-threshold operation and local voltage dithering. JSSCC, 41(1):238–245, 2006.

[3] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos. Data driven signal processing: an approach for energy efficient computing. ISLPED, pp. 347–352, 1996.

[4] L. Di, M. Putic, B. Calhoun, and J. Lach. Power switch characterization for fine-grained dynamic voltage scaling. ICCD, pp. 605-611, 2008.

[5] V. Gutnik. Variable Supply Voltage for Low Power DSP. Master's thesis, Massachusetts Institute of Technology, May 1996.

[6] V. Gutnik and A. Chandrakasan. An efficient controller for variable supply-voltage low power processing. SVLSI, pp. 158–159, 1996.

[7] V. Gutnik and A. Chandrakasan. Embedded power supply for low-power DSP. TVLSI, 5(4):425–435, 1997.

[8] J. Hu, Y. Shin, N. Dhanwada, and R. Marculescu. Architecting voltage islands in core-based system-on-a-chip designs. ISLPED, pp. 180–185, 2004.

[9] Y. Kanno et al. Hierarchical power distribution with power tree in dozens of power domains for 90-nm low-power multi-CPU SoCs. JSSCC, 42(1):74–83, 2007.

[10] H. Kawaguchi et al. A 0.5V, 400MHz, v/sub 00/-hopping processor with zero-V/sub TH/ FD-SOI technology. ISSCC, vol. 1, pp. 106–481, 2003.

[11] D. Lackey et. al. Managing power and performance for system-on-chip designs using voltage islands. ICCAD, pp. 195–202, 2002.

[12] S. Lee and T. Sakurai. Run-time voltage hopping for low-power real-time systems. DAC, pp. 806–809, 2000.

[13] A. Prodic, D. Maksimovic, and R. Erickson. Design and implementation of a digital PWM controller for a high-frequency switching DC-DC power converter. IECON, vol. 2, pp. 893–898, 2001.

[14] R. Puri, D. Kung, and L. Stok. Minimizing power with flexible voltage islands. ISCAS, vol. 1, pp. 21–24, 2005.

[15] N. Seki et al. A fine-grain dynamic sleep control scheme in mips r3000. ICCD, pp. 612-617, 2008.

[16] G. Wei, O. Trescases, and W. T. Ng. A dynamic voltage scaling controller for maximum energy saving across full range of load conditions. EDSSC, pp. 371–374, 2005.

[17] D. Chen, J. Cong, and J. Xu. Optimal simultaneous module and multivoltage assignment for low power. ACM TODAES, 11(2):362-386, 2006.

[18] D. Chen et al. Optimality study of resource binding with multi-Vdds. DAC, pp. 580-585, 2006.

[19] A. Manzak and C. Chakrabarti. A low power scheduling scheme with resources operating at multiple voltages. TVLSI, 10(1):6-14, 2002.

[20] H. Yang and L. Dung. On multiple-voltage high-level synthesis using algorithmic transformations. DAC, pp. 872-876, 2005.

[21] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos. Data driven signal processing: an approach for energy efficient computing. ISLPED, pp. 347-352, 1996.

[22] Y. Zhang et al. Optimal procrastinating voltage scheduling for hard real-time systems. DAC, pp. 905-908, 2005.