Technology Characterization Model and Scaling for Energy Management

Harshil Goyal(✉) and Vishwani D. Agrawal(✉)

Department of Electrical and Computer Engineering, Auburn University,
Auburn, AL 36849, USA
{hzg0007,agrawvd}@auburn.edu

Abstract. We present a low-cost methodology to find the highest energy efficiency operating conditions (voltage and frequency) for a processor with given performance requirements. Taking a black box approach, we start with processor specifications: nominal voltage, clock frequency, thermal design power (TDP), maximum frequency and maximum power, and a knowledge of the device technology. To determine the behavior of the processor, we use a small model circuit that can be economically but accurately simulated in Spice to learn the delay and energy characteristics of the technology. We simulate the model with random vectors to determine power consumption profiles and critical path delays at several voltages. Comparisons between the model data and processor specifications provide scale factors for area, voltage, nominal frequency and maximum frequency. We then optimize the operating modes of the processor for highest cycle efficiency (clock cycles per unit energy). An illustration considers a processor with 3.3 GHz clock, 1.2 V nominal voltage, and 95 W thermal design power. Several optimization scenarios are possible. Observing that the clock is power constrained, we reduce the voltage to 0.92 V, keeping the clock at 3.3 GHz, which now becomes structure-constrained. This gives a 127% higher cycle efficiency over the nominal operation. For highest performance, we set the voltage to 1.1 V and increase the clock to 4.5 GHz while holding power unchanged at 95 W. This gives 38% higher cycle efficiency than the nominal operation. The highest cycle efficiency, ten times greater than the nominal, occurs for subthreshold voltage operation at 0.35 V and 36 MHz.

Keywords: Microprocessor · Power management · Managing performance · Energy efficiency · Subthreshold operation

1 Introduction

Most VLSI chips, including microprocessors, come with prescribed operating conditions, found in specifications supplied by the manufacturer. While such specifications serve a majority of users they are not optimized for specific applications. For example, a portable application must conserve energy without compromising too much on performance. In a remote sensing system, energy, not performance, may be paramount.

A. Sengupta et al. (Eds.): VDAT 2019, CCIS 1066, pp. 679–693, 2019.
https://doi.org/10.1007/978-981-32-9767-8_56

Application-specific optimization of operation requires power-delay characterization of the chip at various supply voltages. Although this could be done through either actual hardware experiment or entire chip simulation, both options present difficulties. Experimental set up is expensive. Simulation requires a complete model of the chip, often not available from the manufacturer. Even if a simulation model is available, accurate timing and power analysis can be expensive.

This paper presents an inexpensive alternative that a user can adopt with reasonable effort and with readily available information. One needs the specifications of the processor and knowledge of its semiconductor technology. Relevant data from specifications include nominal voltage, clock frequency, thermal design power (TDP), maximum frequency (f_{max}) and peak power (P_{max}).

The *main idea* is to simulate a *model* digital circuit at the circuit-level using the technology of the target device. The model can be any circuit of convenient size. The simulator can be any efficient version of Spice with accurate technology models. In our illustration of the methodology, we use HSPICE [1] and, in the absence of actual models, employ the *predictive technology model* (PTM) [2]. The simulation is repeated at several voltages over the entire range in which the device can function. The results are then matched with the device specification data to determine suitable scale factors for size (area), voltage and frequency. In a typical example discussed in the paper the area factor is 7.34×10^5, which results in tremendous savings in computation costs over those of full device simulation.

Section 2 examines relevant papers on power and performance, highlighting the principal differences in the work presented in this paper. Section 3 outlines relevant definitions. Section 4 describes modeling and simulation used in this low-cost learning methodology applied to the processor black box. Section 5 continues with an example of the Intel Sandy Bridge processor, deriving five (both static and dynamic) power management scenarios. Section 6 provides a study of processors in 45 nm, 32 nm and 22 nm *bulk* and *high*-K CMOS technologies. Results in Sects. 4 and 6 are based on a predictive technology model (PTM) [2] only to illustrate the methodology and allow a preview of the type of results expected if real models were available. Section 7 summarizes the main ideas. Section 8 outlines proposals for future research.

2 Background and Experimental Methods

With consumption of desktop microprocessors reaching 130 W, power has emerged as a major challenge facing microprocessor designers [4,9]. A microprocessor must deliver the highest possible performance while keeping power consumption within reasonable limits. Recent theoretical and experimental investigations aim at managing the power (energy and temperature) and delay (speed) of CMOS circuits. In this section, we discuss some recent work.

Wong *et al.* [21] examine changes when moving from Intel's 32 nm planar (32 nm Sandy Bridge processor: Core i5-2500K) to 22 nm Tri-Gate (22 nm Ivy

Bridge processor: i5-3570K) process by comparing their power, voltage, temperature, and frequency. Power consumption is measured using a multimeter on the 1.2 V power connector to record current and voltage. Biostar TZ77MXE motherboard allows adjustment of processor frequency and voltage. The processor's operating voltage is measured by the on-board IT8728F chip. Temperature is measured using core-temp (CPU on-die temperature sensors), reporting the temperature of the hottest core. The results from this experiment indicate that the 22 nm Ivy Bridge has significantly lower static (leakage) power consumption over 32 nm Sandy Bridge, but only shows a small reduction in dynamic power. Ivy Bridge requires higher voltage increase for the same frequency increase, leading to more difficult overclocking but saves power at lower (standard) speeds. In addition to the CMOS process changes, the thermal resistance of Ivy Bridge increased over Sandy Bridge, perhaps due to the change from solder to polymer thermal interface material between the die and the heat spreader.

Sankari [13] developed two proactive thermal aware approaches, PTAS (Proactive Thermal Aware Scheduler) and PTFM (Proactive Thermal aware scheduling with Floating point and Memory access rates), which reduce CPU temperature by predicting the temperature gradient from the rate of change in the CPU temperature, floating point access rate and memory access rate. Their experiments were conducted on desktop and laptop machines with the Ubuntu operating system. They ran eight SciMark benchmarks: Fast Fourier Transform (FFT), Jacobi Successive Over-Relaxation (SOR), Dense Unit Factorization (LU), Sparse Matrix Multiplication (Sparse), FFT-Large, SOR-Large, LU-Large, and SparseLarge. For PTAS, the reductions in peak temperature were 2 °C to 4 °C. The reduction in peak/average temperatures on a laptop were 3 °C to 5 °C/5 °C. Corresponding penalties in the schedule length were between 15% and 30%. For PTFM, there was a decrease in peak/average CPU temperatures: 3 °C to 6 °C/6 °C for small benchmarks and 3 °C to 6 °C/5 °C for large benchmarks. The schedule length penalties were less than 2% to 10%. The corresponding results in peak/average temperature on a laptop were 3 °C to 6 °C/6 °C.

Travers [16] reduces power usage by splitting tasks between several cores. Energy reduced to half with performance greater than that of a single core. Ye *et al.* [22] introduce learning-based dynamic power management (DPM) for multicore processors. Using task allocation they manipulate idle periods on processor cores to achieve a better trade-off between power consumption and system performance. Ghasemazer *et al.* [7] minimize total power consumption of a chip multiprocessor (CMP) while maintaining a target average throughput. They use coarse-grain dynamic voltage and frequency scaling (DVFS), task assignment at the CMP level, and fine-grain DVFS based on closed-loop feedback control at the core level.

Cited here are just a few examples. In contrast, our approach relies on electrical (non-functional) characteristics of the processor. This makes the power-performance management independent of any specific application. We could term it *macro-management*, as opposed to micro-management that involves internal hardware details of the processor, as well as, is dependent on the specific software being run. Benefits and weaknesses of the present approach are brought out in Sects. 4 through 8.

3 Energy Metrics

Although power-delay product (PDP) [10] has been a popular metric for energy, we use cycle efficiency [15], which is a meaningful measure of computation per unit of energy.

3.1 Cycle Efficiency

We consider clock cycle as a unit of computational work. It has two dimensions, time and energy, and is characterized by the time per cycle (TPC) often referred to as clock period and the energy per cycle (EPC). Inverses of these parameters are frequency, $f = 1/TPC$, and *cycle efficiency*, $\eta = 1/EPC$ [15]. Clearly, f and η are numbers of cycles per unit of time and energy, respectively.

To make the operation fast we increase f, thereby reducing the time to execute a clock cycle, and to make the operation efficient we increase η by reducing the energy used in a cycle. Suppose, a program running on a processor takes c clock cycles to execute. Then we have,

$$\text{Execution time} = \frac{c}{f} \tag{1}$$

$$\text{Energy consumed} = \frac{c}{\eta} \tag{2}$$

where, η is cycle efficiency of the processor in cycles per joule. Equation 1 gives the performance in time as,

$$\text{Performance in time} = \frac{1}{\text{Execution time}} = \frac{f}{c} \tag{3}$$

Similarly, Eq. 2 gives the energy performance as,

$$\text{Performance in energy} = \frac{1}{\text{Energy consumed}} = \frac{c}{\eta} \tag{4}$$

Clearly, cycle efficiency (η) characterizes the energy performance in a similar way as frequency (f) characterizes the time performance. These two performance parameters are related to each other by the power being consumed, as follows:

$$\text{Power} = \frac{f}{\eta} \tag{5}$$

For a computing task, f is the rate of execution in time and η is the rate of execution in energy. Taking the automobile analogy, f is analogous to speed in miles per hour (mph) and η is analogous to miles per gallon (mpg).

Table 1. Intel i5 Sandy Bridge 2500 K processor specifications [11].

Technology node	32 nm
Voltage range, V_{dd}	1.2–1.5 V
Nominal base frequency, f_{TDP}	3.3 GHz
Overclock frequency, f_{max}	5.01 GHz
Thermal design power, TDP	95 W
Peak power, P_{Peak}	132 W

3.2 Two Power Limits: Thermal and Peak

- *Thermal design power* (TDP) is the maximum average power in watts the processor dissipates when operating at base (nominal) frequency under a manufacturer defined, high complexity workload.
- *Peak power* is the maximum power dissipated by the processor under the worst case conditions - at the maximum core voltage, maximum temperature and maximum signal loading conditions.

4 Technology Characterization and Scaling

Technology characterization describes the electrical behavior of a circuit in terms of voltage, power, frequency, energy and time (performance). For a processor, such characterization allows one to estimate its frequency and cycle efficiency as functions of the supply voltage. These data are then used to manage the operation of the processor.

We use a ripple carry adder (RCA) circuit as a model for technology characterization. In general, any circuit of reasonable size can be used for this purpose. Simulation of the RCA is carried out using HSPICE [1], with suitably selected input vectors to determine the critical path delay, power consumption and a minimum energy point. This approach differs from that of the co-called “canary software” [14] or “canary circuit” [20], which predict an impending failure of the full scale system. However, a similarity is the predictive behavior of our model circuit, whose analysis characterizes the full-scale processor, sometimes even for future technologies.

To illustrate the proposed methodology, we use the Intel i5-2500K processor [3]. A full scale gate or transistor-level circuit model was not available to us. Even otherwise, it would be too complex for detailed simulation at various voltages. We will only use the operational data, such as, voltage, maximum clock frequency and power consumption, available from the published specification datasheet of the processor. These are extracted in Table 1, where the technology of the device is specified as well.

The main idea in the presented low-cost methodology is to first learn the voltage-power-speed behavior of the technology by simulating a reasonably small *model circuit and, then, derive scale factors to scale up the model data to represent the actual device (such as a processor) in terms of voltage, size, nominal frequency and maximum frequency. Finally, the scaled data allows energy and speed trade-offs, both in static and dynamic operation scenarios.*

We define four scale factors to scale the model circuit data up to the processor [8]:

1. Voltage Scale Factor (σ): It accounts for the difference between the voltage at which adder (RCA) is simulated and the processor supply voltage,

$$\sigma = \frac{V_{dd}^2\ (Processor)}{v_{dd}^2\ (Adder)} \tag{6}$$

2. Area Scale Factor (β): It represents the relative size of processor to that of the adder (RCA),

$$\beta = \frac{TDP}{\sigma[(e_{dyn} \times f_{TDP}) + p_{stat}]} \tag{7}$$

   where e_{dyn} and p_{stat} are dynamic energy and static power of the model. Since e_{dyn} is a function of signal activity in the model, the difference in activities of the model circuit and the processor is implicit in the area scale factor. We have, therefore, not used a separate scale factor for activity.
3. Nominal Frequency Scale Factor (δ): It is the ratio of processor's nominal frequency to adder's maximum frequency at rated voltage, e.g., $V_{dd} = 1.2\,\text{V}$, and is used to find suitable frequency for processor at any supply voltage,

$$\delta = \frac{f_{nomVdd}(Processor)}{f_{maxVdd}(Adder)} \tag{8}$$

4. Maximum Frequency Scale Factor (γ): It is the ratio of processor's maximum frequency to adder's maximum frequency at rated voltage, e.g., $V_{dd} = 1.2\,\text{V}$, and is used to find the maximum (structural or critical path) frequency for processor at any supply voltage,

$$\gamma = \frac{f_{maxVdd}\ (Processor)}{f_{maxVdd}\ (Adder)} \tag{9}$$

4.1 Nominal, Structure-Constrained and Power-Constrained Frequencies

Three frequencies, f_{nom} (nominal or base frequency), f_{max} (maximum, structure-constrained or critical path frequency) and f_{TDP} (power constrained frequency), are determined by scaling adder data. This also results in energy per cycle (EPC) and cycle efficiency (η) for each frequency. We will express frequencies in gigahertz (GHz), or billion cycles per second.

Processor's *base* or *nominal* clock frequency is specified in manufacturer's datasheet. It is the frequency at which TDP is defined. We calculate the nominal frequency, f_{nom} as:

$$f_{nom} = \delta \times f_{max}(Adder) \tag{10}$$

In a structure constrained system, f_{max} is limited by the critical path delay of the circuit. Therefore,

$$f_{max} = \gamma \times f_{max}(Adder) \tag{11}$$

In a power constrained system [17,18], the frequency f_{TDP} is limited by the maximum allowable power for the circuit [8]

$$f_{TDP} = \frac{TDP - \sigma\beta p_{stat}}{\sigma\beta e_{dyn}} \tag{12}$$

where TDP is thermal design power of processor at given f_{TDP} and rated voltage, σ is voltage scale factor, β is adder to processor area scale factor, p_{stat} is the static power of the adder, and e_{dyn} is the dynamic energy of the adder.

At any voltage, the highest clock frequency is [8],

$$f_{opt} = min(f_{TDP}, f_{max}) \tag{13}$$

4.2 Energy per Cycle and Cycle Efficiency for Processor

The energy per cycle for the processor for the nominal frequency and overclock/maximum frequency for a any given V_{dd} is defined by:

$$EPC_{nom} = \frac{TDP}{f_{nom}} \tag{14}$$

$$EPC_{F_0} = \frac{P_{dyn}}{f_{nom}} + \frac{P_{static}}{F_0} \tag{15}$$

Equation 15 defines the energy per cycle EPC_{F_0} for any given frequency F_0 of processor where F_0 lies in the range, $f_{nom} \leq F_0 \leq f_{max}$. In this case, $F_0 = f_{max} = 5.01$ GHz. Therefore, we call EPC_{F_0} as EPC_{fmax}, i.e., energy per cycle for maximum frequency allowed to run the system at a given voltage. As we know, cycle efficiency $\eta = 1/EPC$, therefore, from Eqs. 14 and 15 we can define cycle efficiency for the processor as:

$$\eta = \frac{1}{EPC_{nom}} \tag{16}$$

$$\eta_0 = \frac{1}{EPC_{F_0}} \tag{17}$$

where η is defined as nominal cycle efficiency and η_0 as cycle efficiency for any given frequency F_0 in the range, $f_{nom} \leq F_0 \leq f_{max}$. Here, $EPC_{F_0} = EPC_{f_{max}}$, therefore, we call η_0 as peak cycle efficiency.

All parameters defined above are used in the next section that illustrates the proposed power management method. We show how one can optimize time and energy of a processor based on the performance and efficiency requirements of the user.

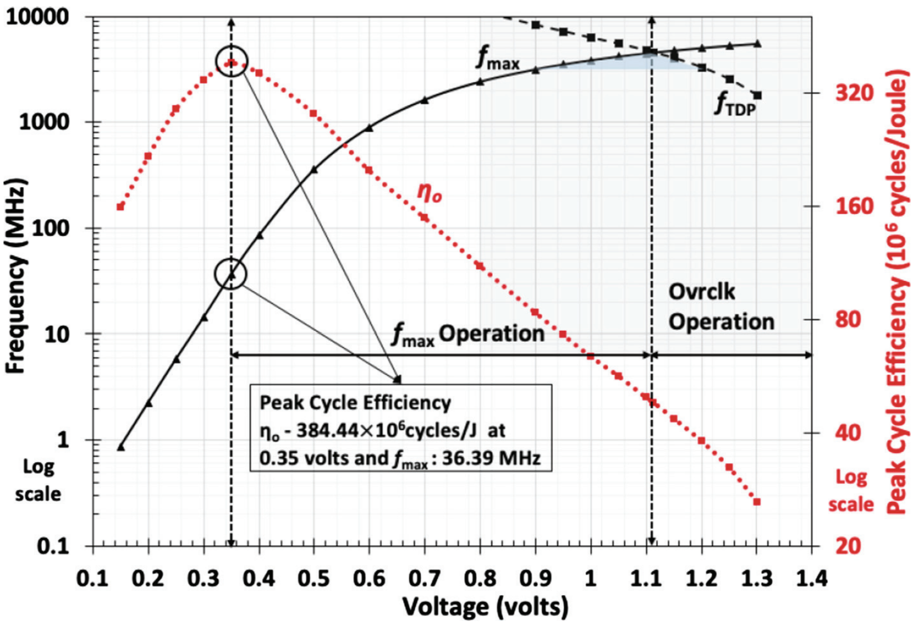

Fig. 1. Plot showing highest performance with respect to clock f_{max} at 1.1 V, highest efficiency at 0.35 V, and overclock operation (1.1 V–1.4 V) for Intel i5 Sandy Bridge 2500K processor. At the nominal voltage 1.2 V, the two frequencies f_{TDP} and f_{max} match the processor specifications of Table 1.

5 Power Management

Power management provides a system solution to boost the processor frequency to values higher than the nominal value, whenever required as per performance criteria. For workloads that are not operating at the cooling/power supply limits this can often result in real performance increase. The focus of this experiment is to evaluate the benefits of the proposed methodology and not necessarily assess the capability of any particular device.

We consider all aspects necessary for time and energy optimization, such as: (a) What will be the most energy-efficient point for the processors that requires low power, ruling out high performance as a main criteria explained through Fig. 1 (b) When is it possible to operate a processor at a higher clock speed

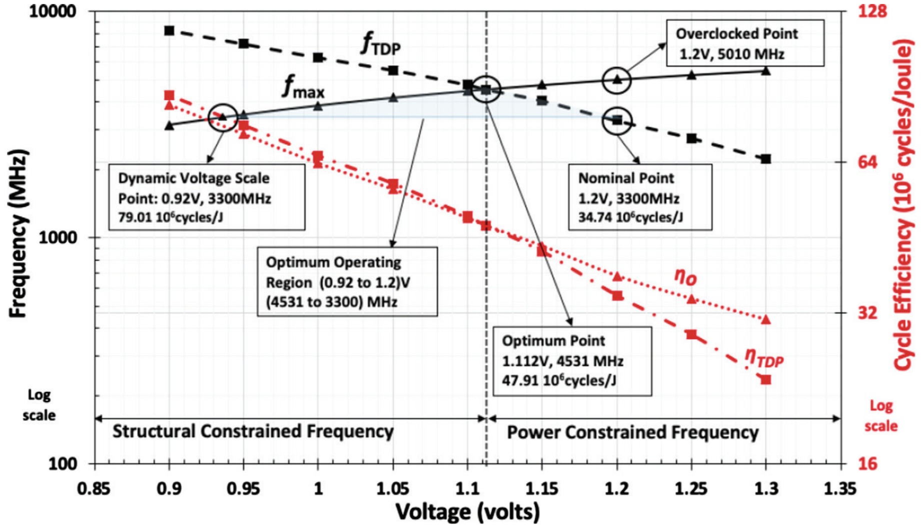

Fig. 2. Scaled version of Fig. 1 from 0.85 V to 1.4 V showing Intel i5 Sandy Bridge 2500K processor's calculated scaled curves of f_{max} and f_{TDP} at various voltages. At the nominal voltage 1.2 V, the two frequencies f_{TDP} and f_{max} match the processor specifications of Table 1.

without exceeding the power limits explained through Fig. 2 And (c) the value of doing so explained with five scenarios in Table 2. Using the processor performance counters to measure execution events of applications, we identify the characteristics that determine the extent of performance benefits in terms of time and energy from higher as well as lower clock frequencies and those characteristics that cause the application to become power-limited. Consider a program that executes in two billion clock cycles ($c = 2 \times 10^9$). Five scenarios of Table 2 are:

Nominal Operation: For nominal conditions, $V_{dd} = 1.2$ V and clock frequency $f = 3.3$ GHz, Figs. 1 and 2 indicate that operation is power-constrained. Cycle efficiency $\eta_{TDP} = 34.74 \times 10^6$ cycles/J. Power consumption = 95 W, which agrees with the processor specification shown in Table 1 and can also be calculated from Eq. 5. The execution time of the two billion clock cycle program is 0.61 s from Eq. 1 and the total energy consumed by the program is 57.57 J from Eq. 2.

Overclock Operation: Overclocking a microprocessor refers to faster than the nominal clock speed, prescribed for sustained operation. Overclocking is a popular technique for getting a performance boost from the system, without acquiring faster hardware. Overclocking can be sustained only for short bursts because CPU will have increased heating. One may also employ additional cooling. This scenario also uses 1.2 V and 80% of the program is executed at 3.3 GHz clock, but the remaining 20% of the program is executed at an overclock frequency of 5.0 GHz, which is the highest frequency the critical path will allow at 1.2 V (Fig. 2). Thus, the power exceeds TDP for 20% of time. Note

Table 2. Managing the processor operation for time and energy used by a program requiring two billion clock cycles ($c = 2 \times 10^9$).

Operating modes	V_{dd} (volts)	Clock frequency f (MHz)	Cycle efficiency η (Mc/J)	Average power $\frac{f}{\eta}$ (watts)	Execution time $\frac{c}{f}$ (seconds)	Total energy $\frac{c}{\eta}$ (J)
Nominal Operation	1.2	3300	34.74	95	0.61	57.57
Overclocked operation with 20% overclocking	1.2	3300 (80%) 5010 (20%)	27.792 + 7.602 = 35.394	95 132	0.485 + 0.0798 = 0.57	46.06 + 10.52 = 56.58
Highest performance operation	1.112	4531	47.91	95	0.44 (−28%)	41.75 (−28%)
Dynamic voltage scaling (DVS) operation	0.92	3300	79.01	41.77 (−56%)	0.61 (0%)	25.31 (−56%)
Most energy efficient operation	0.35	36.39	384.45	0.0946	54.96	5.20

that power increase from 95 W to 132 W is not proportional to the frequency ratio, because only dynamic power increases, leaving static power unchanged. Cycle efficiency η_{TDP} at 3.3 GHz is 34.74×10^6 cycles/J and η_0 at 5010 MHz is 38.01×10^6 cycles/J. The execution time is reduced to 0.57 s and total energy consumption is slightly lower at 56.58 J. We do not observe a significant reduction in execution time or total energy in this scenario despite higher power consumption. To illustrate this, let f_1 and f_2 be two frequencies such that:

$$f_1 \leq f_{max}; \quad \text{at } V_1 \text{ (Rated voltage)}$$

$$f_2 \leq f_{max}; \quad \text{at } V_2 \; (V_2 \leq V_1)$$

Let x be the fraction of time f_1 is used, where $0 \leq x \leq 1$. Therefore, to maximize average performance, we *maximize* $\{xf_1 + (1-x)f_2\}$ under the following constraint:

$$x(E_{dyn}f_1 + P_{stat}) + (1-x)\frac{V_2^2}{V_1^2}(E_{dyn}f_2 + P_{stat}) \leq TDP \tag{18}$$

where, E_{dyn} and P_{stat} are dynamic energy per cycle and static power for processor at rated voltage. The voltage ratio V_2/V_1 will be denoted by σ; this is unlike the scale factor of Eq. 6 since no model circuit is involved. Now, TDP for the processor is expressed as:

$$TDP = E_{dyn} \cdot f_{TDP} + P_{stat} \tag{19}$$

Solving Eq. 19 for f_{TDP}, we get:

$$f_{TDP} = \frac{TDP - P_{stat}}{E_{dyn}} \tag{20}$$

Therefore, from relations 18 and 20, we derive an upper bound for $\{xf_1 + (1-x)f_2\}$ as follows:

$$xf_1 + \sigma^2(1-x)f_2 \leq f_{TDP} + \frac{(1-\sigma^2)(1-x)P_{stat}}{E_{dyn}} \tag{21}$$

Here, f_1 and f_2 have an upper bound of f_{max}, the critical path frequency, which is generally higher than f_{TDP} at the rated voltage (Fig. 2). However, an overall performance higher than f_{TDP} would be possible only when the last term on the right hand side is positive, requiring $\sigma < 1$, which implies dual voltage operation. One the other hand, if $\sigma = 1$ the average performance does not exceed the performance at a single frequency f_{TDP}, because inequality 21 reduces to:

$$xf_1 + (1-x)f_2 \leq f_{TDP} \tag{22}$$

Performance optimization with single frequency is discussed next.

Highest Performance Operation: If we let $x = 1$ in inequality 22, then $f_1 < f_{TDP}$ and its optimum value is $f_1 = f_{opt} = f_{TDP}$. The only way to increase f_{TDP} is to reduce voltage. However, f_{opt} must not exceed the critical path frequency f_{max}. In this scenario we find optimum voltage, frequency and cycle efficiency ($V_{ddopt}, f_{opt}, \eta_{opt}$). From Fig. 2 we determine $V_{dd} = 1.112$ V and clock frequency $f = 4.531$ GHz, which give cycle efficiency $\eta_{opt} = 47.91 \times 10^6$ cycles/J. This is a single clock operation where the processor is operated at maximum frequency (f_{max}) giving the highest performance. The power consumption is no more than 95 W (TDP) and the program execution time reduces to 0.44 s and total energy consumed is 41.75 J. Thus, we observe 28% reduction in both energy consumption and execution time.

Dynamic Voltage Scaling (DVS) Operation: There have been a number of efforts over the years examining the implementation and effectiveness of dynamic voltage and frequency scaling for saving power in embedded systems [12]. Performance-oriented explorations include attempts to quantify and/or reduce the performance loss encountered in an energy-saving adoption of DVS. In contrast, our fourth scenario targets performance increase from DVS in a power-constrained environment. Here the program can execute at the rated frequency, which is 3300 MHz, by decreasing the voltage to 0.92 V. Now, $\eta_0 = 79.01 \times 10^6$ cycles/J as obtained from Fig. 2. The power consumption is 41.77 W, but the program execution time is 0.61 s, the same as that obtained at the rated voltage. However, total energy consumed is reduced to 25.31 J. Here, we see performance enhancement in terms of energy and not time, therefore, when the criterion is lower energy and not higher speed, this type of operation is appropriate.

Highest Energy Efficiency Operation: The fifth scenario is derived for highest cycle efficiency. Figure 1 shows minimum energy operation at $V_{dd} = 0.35$ V and frequency 36.39 MHz. This is subthreshold voltage operation [19]. When a program executes at this low voltage, it gives cycle efficiency $\eta_0 = 384.45 \times 10^6$ cycles/J, which is the peak cycle efficiency for this processor. The power consumption for this type of execution is 0.0946 W, or 94.6 mW, but the program execution time is increased to 54.96 s and the energy consumption is the lowest at 5.20 J.

Table 3. Specified (nominal) operation and optimized high efficiency mode at the specified performance for Intel processors characterized using a predictive technology model (PTM [2]).

CMOS technology	Intel microprocessor	Manufacturer's specification				
		Nominal operation			High efficiency	
		f_{TDP} MHz	V_{dd} V	η_{TDP} Mc/J	V_{dd} V	η_0 Mc/J
45 nm *Bulk*	Core2 Duo T9500	2600	1.25	74.29	1.07	108.58
45 nm *High*-K	Core2 Duo T9500	2600	1.25	74.29	0.79	350.91
32 nm *Bulk*	Core i5 2500K	3300	1.20	34.74	0.92	79.01
32 nm *High*-K	Core i5 2500K	3300	1.20	34.74	0.67	267.57
22 nm *Bulk*	Core i7 3820QM	2700	0.80	60.00	0.70	96.22
22 nm *High*-K	Core i7 3820QM	2700	0.80	60.00	0.61	137.65

Table 4. Maximum performance (highest frequency) and minimum energy (highest efficiency) modes for Intel processors characterized using a predictive technology model (PTM [2]).

CMOS technology	Intel microprocessor	Optimized operation					
		Maximum speed			Minimum energy		
		V_{ddopt} V	f_{opt} MHz	η_{opt} Mc/J	V_{dd} V	$f_{\eta 0}$ MHz	η_0 Mc/J
45 nm *Bulk*	Core2 Duo T9500	1.200	2920	82.28	0.35	33.51	829.29
45 nm *High*-K	Core2 Duo T9500	1.226	3120	89.08	0.30	304.48	1795.00
32 nm *Bulk*	Core i5 2500K	1.112	4531	47.91	0.35	36.39	384.45
32 nm *High*-K	Core i5 2500K	1.155	4940	51.77	0.30	414.2	953.81
22 nm *Bulk*	Core i7 3820QM	0.771	3494	75.46	0.38	177.3	213.99
22 nm *High*-K	Core i7 3820QM	0.760	3626	80.38	0.30	332.6	375.76

6 Summary

Performance and energy optimization data for processors in *bulk* and *high*-K technologies using 45 nm, 32 nm and 22 nm transistor sizes, respectively, are given in Tables 3 and 4 [8]. We make following observations:

Optimizing Nominal Operation (Table 3, columns 5 and 7): For nominal clock frequency, optimized efficiency is always higher than the efficiency for the specified operation. This is accomplished by lowering the supply voltage.

Bulk vs. ***High*-K:** *High*-K consistently has higher frequency (Table 4), as well as higher cycle efficiency (Table 4 and high efficiency mode in Table 3), perhaps due to the reduced leakage.

Performance Optimization (Table 4, columns 3–5): Clock rate can be increased by suitably lowering the voltage, but the efficiency (Table 4, column 5) drops below the maximum achievable at the nominal clock rate (Table 3, columns 7). Still, this efficiency is superior to the rated specification (Table 3, columns 5).

Energy Optimization (Table 4, columns 6–8): Highest efficiency is achieved in the subthreshold voltage range, and is almost an order of magnitude higher than that for the rated specification (Table 3, column 5 or 7), even though the performance in the sub-threshold voltage region (Table 4, columns 7) is reduced almost by an order of magnitude compared to all other operating modes.

In interpreting the available information on the specifications and structure of these processors, we made several assumptions. Hence the data and observations presented here may not exactly represent the behavior of Intel processors. Notably, the outcome of this investigation is a *methodology* for performance and energy optimization.

7 Conclusion

We explored how power management affects the energy and performance of a processor. The proposed method is entirely a simulation based evaluation that accomplishes the goal of performance and energy optimization. Some observations are:

1. Highest performance mode has a superior sustained clock rate than the rated (nominal or specified) clock rate.
2. Highest efficiency at rated clock needs voltage lowering.
3. Performance is enhanced by overclocking, which may require raising voltage whenever frequency is increased.
4. Highest efficiency operation without performance bound uses subthreshold voltage and clock in megahertz range.

Strengths of our methodology are low-cost, simplicity and generality. The black box approach works with minimal details from the datasheet of the processor. The result serves a wide variety of applications. It also allows us to evaluate technologies before a processor chip becomes available. Technology evaluation is through circuit level (Spice) simulation and is more accurate than a coarse evaluation normally done when whole processor is simulated.

Some results on the Sandy Bridge processor have been verified against those obtained in experiments [21]. Still a weakness of the method is lack of application-specific customization where other methods may work better.

8 Future Work

Energy efficiency continues to be a major issue [5]. The present work creates an optimization framework. The simplicity of our *black box macro-modeling approach* makes it useful to many users. However, we must acknowledge areas where work still needs to be done. Besides giving solutions to existing problems, it opens the door for other research venues:

1. Analysis and simulation in this work ignored process variability that is important in nanometer technologies.
2. Although thermal design power (TDP) and peak power may take some heating effects into account, certain applications can produce severe hot spots on the chip. Fine grain thermal management is an area for research.
3. We notice that energy efficiency increases as voltage drops. For given performance, operating voltage should be the lowest to allow that frequency. This suggests further exploration of the near (but above) threshold range of V_{dd} where increased energy efficiency may be possible with only minor loss of performance [6].
4. Operation in the sub-threshold voltage region [19] may be sensitive to the thermal as well as other types of noises. Reliability of such operation requires study.
5. Signal activity of the ripple carry adder (RCA) need not be the same as in the processor. Any difference in the activity is implicitly compensated for by adjustment of the area scale factor. Alternatively, a separate scale factor can account for different activities in the two circuits.

References

1. HSPICE Signal Integrity User Guide: Synopsys Inc., 700 East Middlefield Road, Mountain View, CA 94043 (2010)
2. Predictive Technology Model. Nanoscale Simulation and Modeling (NIMO) Group, Arizona State University (2012). http://ptm.asu.edu/
3. Intel core i5–2500k processor specifications (2016). http://ark.intel.com/products/52210/Intel-Core-i5-2500K-Processor-6M-Cache-up-to-3_70-GHz. Accessed 20 Feb 2016
4. Annavaram, M., Grochowski, E., Shen, J.: Mitigating Amdahl's law through EPI throttling. In: Proceedings of 32nd IEEE International Symposium on Computer Architecture (ISCA), pp. 298–309 (2005)
5. Borkar, S., Chien, A.A.: The future of microprocessors. Commun. ACM **54**(5), 67–77 (2011)
6. Dreslinski, R.G., Wieckowski, M., Blaauw, D., Sylvester, D., Mudge, T.: Near-threshold computing: reclaiming Moore's law through energy efficient integrated circuits. Proc. IEEE **98**(2), 253–266 (2010)
7. Ghasemazar, M., Pakbaznia, E., Pedram, M.: Minimizing the power consumption of a chip multiprocessor under an average throughput constraint. In: Proceedings of 11th International Symposium on Quality Electronic Design (ISQED), pp. 362–371, March 2010
8. Goyal, H.: Characterizing processors for time and energy optimization. Master's thesis, Auburn University, Auburn, Alabama, USA, August 2016
9. Grochowski, E., Ronen, R., Shen, J., Wang, H.: Best of both latency and throughput. In: Proceedings of the International Conference on Computer Design, pp. 236–243 (2004)
10. Rabaey, J.M., Chandrakasan, A.P., Nikolic, B.: Digital Integrated Circuits, vol. 2. Prentice-Hall, Englewood Cliffs (2002)
11. Rotem, E., Naveh, A., Rajwan, D., Ananthakrishnan, A., Weissmann, E.: Power-management architecture of the Intel micoarchitecture code-names Sandy Bridge. IEEE Micro **32**, 20–27 (2012)

12. Rubio, J., Rajamani, K., Rawson, F., Hanson, H., Ghiasi, S., Keller, T.: Dynamic processor over-clocking for improving performance of power-constrained systems. Technical Report RC23666 (W0507-124), IBM Research Division, Austin Research Laboratory, Texas, USA, July 2005
13. Sankari, A.: Proactive thermal-aware scheduling. Ph.D. thesis, Auburn University, Auburn, Alabama, USA, December 2014
14. Sartori, J., Kumar, R.: Software canaries: software-based path delay fault testing for variation-aware energy-efficient design. In: Proceedings of International Symposium on Low Power Electronics and Design, pp. 159–164 (2014)
15. Shinde, A.: Managing performance and efficiency of a processor. Master's thesis, Auburn University, Auburn, Alabama, USA, December 2012
16. Travers, M.: CPU power consumption experiments and results analysis of Intel i7-4820K. Technical Report NCL-EEE-MICRO-TR-2015-19, Newcastle University, Newcastle-upon-Tyne, NE17RU, UK, June 2015
17. Venkataramani, P.: Reducing ATE test time by voltage and frequency scaling. Ph.D. thesis, Auburn University, Auburn, Alabama, USA, May 2014
18. Venkataramani, P., Sindia, S., Agrawal, V.D.: A test time theorem and its applications. J. Electron. Test.: Theory Appl. **30**(2), 229–236 (2014)
19. Wang, A., Calhoun, B.H., Chandrakasan, A.P.: Sub-Threshold Design for Ultra Low-Power Systems. ICIR. Springer, Boston (2006). https://doi.org/10.1007/978-0-387-34501-7
20. Wang, J., Calhoun, B.H.: Techniques to extend canary-based standby V_{DD} scaling for SRAMs to 45 nm and beyond. IEEE Jour. Solid-State Circ. **43**(11), 2514–2523 (2008)
21. Wong, H.: A comparison of Intel's 32nm and 22nm core i5 CPUs: power, voltage, temperature, and frequency (2012). http://blog.stuffedcow.net/2012/10/intel32nm-22nm-core-i5-comparison. Accessed 13 Nov 2015
22. Ye, R., Xu, Q.: Learning-based power management for multicore processors via idle period manipulation. IEEE Trans. Comput. Aided Des. **33**(7), 1043–1055 (2014)