A MODEL OF A TEMPERATURE-CONTROLLED VARIABLE-SPEED PROCESSOR

A. Merchant[*], *B. Melamed*[*], *E. Schenfeld*[†] and *B. Sengupta*[*]

Abstract

One limitation on the operating speed of electronic circuits is the rate at which the chip packaging can dissipate heat. In CMOS technology, the heat generated by a processor is approximately proportional to its clock rate. This paper examines the idea of using a variable-speed processor (VSP) that can be operated at a high clock speed, and then slowed down to a lower speed before heat accumulation destroys the circuit. Under a workload consisting of bursts of work alternating with idle periods (corresponding to cache misses or other delays), this results in a higher average operating speed. This paper shows the optimality of a bang-bang control for the clock rate. It also examines an easier-to-implement policy that estimates the junction temperature through an upper bound, and uses this to control the clock rate. Closed-form expressions are derived for the mean rate of instructions executed by a VSP using each control method. Numerical studies show that both policies give substantial improvements in performance over a single speed processor. Furthermore, the studies suggest that a VSP with a maximum clock rate of 2–4 times that of the single speed processor would suffice to obtain the bulk of the performance improvement. In many cases, the average throughput gain is on the order of 40–60%, without exceeding thermal limits.

Keywords: variable speed processor, throughput gain, thermal modeling, performance analysis, Markov processes, differential equations

---

[*] NEC USA, C&C Research Laboratories, 4 Independence Way, Princeton, NJ 08540

[†] NEC Research Institute, 4 Independence Way, Princeton, NJ 08540

1 Introduction

Heat generation is an intrinsic phenomenon associated with any form of work. In the operation of mechanical, electrical and biological systems, heat generation and removal can be of critical importance. For example, the engine of a car generates heat, which is dissipated by the radiator. Without the proper operation of the radiator, the engine would overheat and the the car would not work. Similarly, electronic circuits generate heat as a part of their operation, which must be dissipated. The most common method of heat removal is forced-air cooling of the package in which the electronic component is placed. Other methods include liquid-based heat-removal technologies, which are used in supercomputers to dissipate large amounts of heat generated in relatively small volumes.

One solution that has been used is to cool the chip package using an external cooling device. Such a scheme was used by Everex Systems, Inc. for an Intel 80486, specified to run at a maximum of 25 MHz clock speed, to achieve a clock rate of 50 MHz. Using a Velox Computer Technology Inc., a Santa Clara, California based start-up, product named Icecap, the CPU was cooled to 0^0 Celsius [1].

The performance of electronic circuits is directly related to the amount of heat they generate: high-speed circuits typically generate more heat than slower speed circuits, and therefore require more expensive packaging and heat removal methods capable of dissipating heat faster. The choice of the packaging and the heat removal method is a trade-off between the cost and the performance of the system. The design of the circuit takes into account the "heat budget" provided by this combination, based on an estimate of some worst-case or average-case conditions under which the circuit is to operate. For instance, as an extreme case, it might be assumed that the circuit is to operate continuously, for long periods of time. However, a more realistic assumption is that the actual work load has a more variable profile: from continous operation for periods of time to bursty operation, with periods of work alternating with idle or wait periods.

In some VLSI technologies (e.g., CMOS), it is possible to switch quickly between an operational mode and a non-operational mode in which heat generation is negligible. For example, in static-CMOS design, heat generation, as well as the performance, is directly proportional to the clock speed. In some cases, the maximum clock speed that can be used is limited by the heat budget imposed by the packaging and the heat removal methods chosen, since higher clock speeds would generate heat faster than the package can dissipate it. In such cases, it is possible to improve the performance of the circuit by varying the clock rate dynamically. Since the workload is, in many cases, bursty, the circuit can be moved to a non-operational mode during pauses in its operation (i.e., idle or wait periods.) Little or no heat is generated in this mode, thereby allowing the circuit to cool below the maximum operating temperature. Thus, when work arrives, a faster-than-normal clock speed can be used for some period of time, switching to a slower clock speed before heat accumulation destroys the circuit. Thus, pauses are parlayed into higher speeds, when possible, with an overall gain in the throughput of the variable speed processor (VSP) as compared to the corresponding single-speed processor (SSP). Note that this gain is predicated on the nature of the processor workload (the sequence of work-pause durations). Intuitively, overly long workbursts will result in little or no gain; but short workbursts punctuated by pauses can result in substantial gains. Fortunately, the latter pattern is fairly common in applications, where frequent brief pauses correspond to cache misses followed by memory access or network delays. The concept of regulating the heat generation and removal under variable work/pause loads has been proposed in [6].

In this paper, we assume that the VSP circuitry is a processor implemented in static-CMOS technology. The amount of heat generated is directly proportional to the clock speed of the processor; for a clock speed of 0 Hz, no heat is generated. We assume that processor packaging was designed to dissipate heat in a *continuous operation* mode of the processor at some prescribed clock rate. However, the processor can be run at higher clock rates for limited periods, provided that the temperature does not exceed operating limits. This means that the speed at which the processor can operate is bounded by the ability of its packaging to remove heat, and not by other factors in the design (such as, internal critical-path delays). This could happen, for instance, when packing processors intended for liquid cooling or high throughput air cooling, in a cheaper and/or denser package.

The rate of the clock can be changed according to the particular activity a circuit performs at any given time. When the circuit is switched to a lower clock rate, the balance of heat generation gains an appropriate credit. Given a certain heat capacity (based on the characteristics of the chip and package), it is possible to switch the clock to a higher rate, if enough heat savings were made. While operating at this higher clock rate, the heat budget account is decreased to account for the extra heat generated.

Power managment circuits were used in the past. Such a power chip set was introduced for example by Intel Corp. [5], primarily intended for the laptop system market. To conserve battery power, the design is fully power-aware through power management features built into the microprocessor. Every piece of logic can be powered down, increasing battery life by up to 50 %. In this paper we study a different idea then one used for *power managment.* Our goal is different in that it is primarily a performance oriented. We increase the performance that a CPU will achieve by changing its clock to a higher rate. Our efforts to save heat production are not for the sake of preserving battery life, but for getting a higher performance, depending on the job the CPU is doing. We do not shut circuits down (i.e. operating in "stand-by mode"), but change their clock rate, maintaining them fully operational.

An example of this mode of operation is shown in Figure 1. Suppose that the maximum temperature of the processor (the junction temperature) for safe operation is T_1 and that at this temperature, the packaging of the processor dissipates heat at the rate H_1. Let R_1 be the clock rate at which the circuit generates heat at rate H_1. Clearly, R_1 is the highest clock rate at which the circuit can be operated continously while maintaining the junction temperature at or below T_1. Therefore, a SSP with this packaging would operate at clock rate R_1 or lower. Now consider a VSP which has two clock rates, R_1 and R^*, where $R^* > R_1$. This higher clock rate, R^*, may be dictated by other technological constraints, such as the delays in the

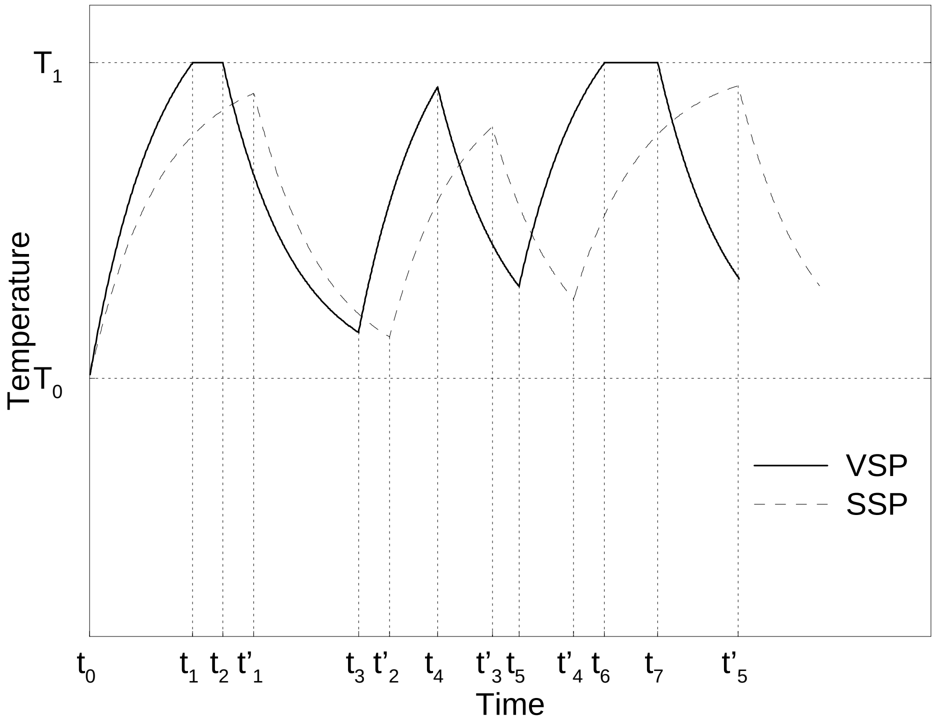

Figure 1: Sample path of temperature for VSP and SSP

critical path of the processor. For illustrative purposes, we assume the policy that the clock rate is R^* if the junction temperature is less than T_1, and is R_1 if the junction temperature equals T_1. The solid line in Figure 1 shows the variation in junction temperature for the VSP under a workload of bursts of work alternating with idle periods. Thus, in the figure, the VSP is operated at clock rate R^* in the intervals (t_1, t_2),(t_4, t_5) and (t_6, t_7); it is operated at clock rate R_1 in the intervals (t_2, t_3) and (t_7, t_8). The intervals (t_3, t_4) and (t_5, t_6) are idle periods in the workload during which the processor cools. In the same figure, the dotted line represents the temperature of a SSP with clock rate R_1. The first work period of the SSP is (t_1, t_2'). Here, $t_2' > t_3$ since the average clock rate of the SSP is lower than that of the VSP. The length of the idle periods (t_3, t_4) for the VSP and (t_2', t_3') for the SSP are identical (since we assume that both are subjected to the same processing workload), but no their times of occurrence. It is clear, at least in this example, that when operating under the two-speed policy, the amount of work completed (throughput) by the VSP is greater than that done by the SSP in any period of time. The foregoing discussion motivates the following questions: (1) What is the form of the optimal operating policy which maximizes the throughput? (2) What is the actual value of the throughput for the optimal policy, and how does it compare with the SSP? (3) Finally, since it is difficult to detect when the junction temperature reaches T_1 (at which point the clock rate must be reduced in the VSP), what is a good way to decide when to change the clock rate without actually measuring the temperature? The next three sections address these questions, while Section 5 presents some numerical results. Section 6 suggest a simple way to implement the VSP and discuss additional applications. Section 7 wraps up the paper with some concluding remarks.

2 Optimality of the Bang-Bang Control Policy

Consider a processor, constructed in some hardware technology, including its packaging. The underlying technology constrains the processor to clock speeds in the range $[0, R^*]$, where R^* is the maximal allowed clock speed. Similarly, the temperature of the processor is constrained to vary in a fixed range $[T_0, T_1]$, where T_0 is the ambient temperature, and T_1 is the maximal temperature that allows safe operation, commensurate with the maximal clock rate R^* and the circuitry's packaging technology. The processor is presented with an infinite workload sequence of the form $B_1, I_1, B_2, I_2, \ldots$, where B_n denotes the number of clock cycles in the n-th workburst, and I_n denotes the number of idle clock cycles in the n-th pause (due to, say, a cache miss or resource contention.) A common workload sequence will be referred to as a *benchmark*. To equalize the initial conditions, we also assume that the processor is started at time 0 at the ambient temperature, and that the work completed at time 0 is nil. A clock cycle is called *busy* if the processor is executing $B_1, B_2, \ldots$, and *idle* if it idles in some I_n.

Define the following functions:

$$\begin{aligned}
\Theta(t) &\equiv \text{processor temperature at time } t \\
W(t) &\equiv \text{number of (non-idle) clock cycles executed by the processor in } [0,t) \\
S(t) &\equiv \text{clock rate at time } t \\
H(t) &\equiv \text{heat generated by the processor in } [0,t) \\
L(t) &\equiv \text{heat dissipated by the processor in } [0,t) \\
X(t) &\equiv \text{state of the processor at time } t \ (0 = \text{idle or } 1 = \text{ busy})
\end{aligned}$$

With this notation, the cumulative amount of heat generated by time t is

$$H(t) \;=\; h\,W(t), \tag{1}$$

where h denotes the amount of heat generated per clock cycle. Allowing for heat dissipation, the heat energy stored in the processor circuit at time t is

$$H(t) - L(t) \;=\; h\,W(t) - L(t) \;=\; q[\Theta(t) - \Theta(0)] \tag{2}$$

where q is the heat capacity of the circuit package. From Eq. (2), we have

$$\Theta(t) \;=\; \frac{h\,W(t) - L(t)}{q} \;+\; \Theta(0), \tag{3}$$

and

$$L(t) \;=\; h\,W(t) - q[\Theta(t) - \Theta(0)]. \tag{4}$$

The rate of heat dissipation for the processor circuit is (see [2, p.313])

$$\frac{dL(t)}{dt} \;=\; c[\Theta(t) - T_0], \tag{5}$$

where c is the heat conductance for the processor circuit packaging mechanism. Recalling that R_1 is the clock speed at which the rate of heat generation equals that of heat dissipation at the maximum operating temperature T_1, we have $h\,R_1 \;=\; c[T_1 - T_0]$, and hence

$$R_1 \;=\; \frac{c[T_1 - T_0]}{h}.$$

The bang-bang control policy (which we later show to be optimal), utilizes the three clock rates, $R_0 \equiv 0$ (idle), R_1 (normal rate), and $R_2 \equiv R^*$ (turbo rate), and is given by

$$S^*(t) \;=\; \begin{cases} 0 & \text{if } X(t) = 0 \text{ (idle)} \\ R_1 & \text{if } X(t) = 1 \text{ (busy), and } \Theta(t) = T_1 \\ R^* & \text{if } X(t) = 1\text{ , and } \Theta(t) < T_1. \end{cases} \tag{6}$$

Let S denote any control policy, such that $S \neq S^*$. We now compare the work completed under the control policy S^*, using the bang-bang control policy (6), with that completed by any other control policy S, over any time interval $[0, t)$. To distinguish the state variables under S^*, we consistently mark them with an asterisk, e.g., $\Theta^*(t)$, $W^*(t)$, $H^*(t)$, $L^*(t)$ and $S^*(t)$ denote, the temperature, work completed, heat generated, heat dissipated, and the clock rate, respectively, for S^* at time t. Finally, let $W^*(0) = W(0) = 0$, $\Theta^*(0) = \Theta(0) = T_0$ and $L^*(0) = L(0) = 0$ be the common initial conditions. Before proving the optimality of $S^*(t)$, we need to establish a few simple facts.

Lemma 1 *Let S be any control policy whose maximal clock speed R satisfies $R < R^*$. Then, there exists a neighborhood $(0, \delta)$, such that*

1. $W^*(t) > W(t), \quad t \in (0, \delta)$

2. $\Theta^*(t) > \Theta(t), \quad t \in (0, \delta)$

3. $L^*(t) > L(t), \quad t \in (0, \delta)$

Proof: The first assertion follows from the fact that there is an interval $(0, \delta)$, in which S^* runs at a higher clock rate than S, since both are started on workload B_1 at the ambient temperature. For the second assertion, differentiate Eq. (2) with respect to t, and substitute Eq. (5) into it. This results in a first-order ordinary differential equation

$$q \frac{d\Theta(t)}{dt} + c\,\Theta(t) = \frac{dH(t)}{dt} + cT_0, \quad t \in (0, \delta).$$

Denoting the forcing function on the right by $r(t)$, the solution of this equation is

$$\Theta(t) = \Theta(0)\, \exp(-ct/q) + \int_0^t \exp(-c(t-s)/q)\, r(s)\, ds.$$

But from the first assertion and Eq. (1), $r^*(t) > r(t) > 0$ for $t \in (0, \delta)$, where $r^*(t) = dH^*(t)/dt + cT_0$. The second assertion follows, because $\Theta^*(0) = \Theta(0) = T_0$, by assumption. Finally, the third assertion follows from the second one in view of Eq. (5), and the initial conditions $L^*(0) = L(0) = 0$. ■

We are now in a position to prove the main optimality result.

Theorem 1 *Let S be any control policy whose maximal clock speed, R, satisfies $R < R^*$. Then, $W^*(t) \geq W(t)$ for $t \geq 0$, for any common benchmark.*

Proof: We prove the theorem by contradiction. Suppose there is a time point $t > 0$, such that the opposite holds, namely, $W^*(t) < W(t)$. By the first assertion in Lemma 1 and by continuity of W and W^*, there is a time point t_0 such that $W^*(t_0) = W(t_0)$ but $W^*(t) < W(t)$, for t in a right neighborhood of t_0. Define

$$t_W = \inf\{t > 0 : W^*(t) < W(t)\} < \infty \tag{7}$$

to be the first such point. Since $W^*(t_W) = W(t_W)$, it follows from Eq. (7) that S must be busy in a right neighborhood of t_W. We first establish that S^* must also be busy in a right neighborhood of t_W. Suppose, conversely, that S^* is idling in I_k, while simultaneously S is busy in a right neighborhood of t_W. Since $W^*(t_W) = W(t_W)$, it must be the case that t_W is precisely the time point when S^* finished idling in I_k and started working on B_{k+1}. But this implies that S^* is idling to the left of t_W. It follows that both S and S^* are idling in I_k to the left of t_W, with starting idling times t_k and t_k^*, respectively, such that $t_k < t_k^*$. But at time t_k^*, we simultaneously have: (i) $W(t_k^*) = W^*(t_k^*)$; (ii) S^* is busy on B_k in a left neighborhood of t_k^*; and (iii) S is idle in I_k in the same left neighborhood. These three facts imply that $W^*(t) < W(t)$, in a left neighborhood of t_k^*. Since $t_k^* < t_W$, this contradicts the definition of t_W in Eq. (7), and we conclude that both S and S^* are busy in a right neighborhood of t_W. It follows that the only way that $W^*(t) < W(t)$ in that right neighborhood is if

$$\Theta^*(t_W) = T_1 > \Theta(t_W). \tag{8}$$

Now, from Eq. (3), we get by subtraction,

$$\Theta^*(t_W) \; - \; \Theta(t_W) \; = \; \frac{L(t_W) - L^*(t_W)}{q}. \tag{9}$$

Combining Eq. (8) and (9), we deduce

$$L(t_W) > L^*(t_W). \tag{10}$$

From Eq. (10) and the third assertion in Lemma 1, there must be a time point $s_0 \in (0, t_W)$, such that $L^*(s_0) = L(s_0)$, by continuity of L and L^*. The point where this happens for the first time to the left of t_W is defined by

$$t_L \; = \; \sup\{t < t_W : \; L(t) > L^*(t)\}. \tag{11}$$

Now, from Eq. (4), we get by subtraction,

$$L(t) - L^*(t) \; = \; h[W(t) - W^*(t)] \; - \; q[\Theta(t) - \Theta^*(t)]. \tag{12}$$

But in the interval (t_L, t_W), we have simultaneously,

$$L^*(t) < L(t), \quad W^*(t) \geq W(t), \quad t \in (t_L, t_W). \tag{13}$$

Combining Eqs. (12) and (13), we conclude,

$$\Theta^*(t) > \Theta(t), \quad t \in (t_L, t_W). \tag{14}$$

Hence, from Eqs. (5) and (14),

$$L^*(t_W) - L^*(t_L) > L(t_W) - L(t_L). \tag{15}$$

Since $L(t_L) = L^*(t_L)$ from Eq. (11), it follows that Eq. (15) implies $L^*(t_W) > L(t_W)$. This contradicts Eq. (10) and the proof is complete. ■

3 A stochastic model for the VSP

In this section, we construct a stochastic model of the work performed by the optimal VSP, S^*, of Eq. (6). Since we only deal here with the optimal control policy S^*, we shall simplify the notation, by dropping all asterisks from quantities associated with S^*.

As in the last section, we assume that the workload is a benchmark $B_1, I_1, B_2, I_2, \ldots$, where B_n represents the length of a sequence of instructions, and I_n represents an idle period. We assume that the I_n's are independent and identically distributed (*iid*) exponential random variables with mean $1/\lambda$, and the B_n's are *iid*, geometrically distributed random variables with mean $1/\mu$; for simplicity, we shall approximate the time to execute an instruction sequence of length B_n at a clock rate of γ instructions/second as an exponential random variable with mean $1/(\mu\gamma)$. As noted in the previous section, the VSP generates a fixed amount of heat h per clock cycle; without loss of generality, we let this be the unit of heat. Also without loss of generality, we let $T_0 = 0$.

To model the circuit, we now redefine the processor's state $X(t)$ at time t to have three possible values (instead of two earlier):

state 0 the "idle" state ($S^*(t) = 0$)

state 1 the "turbo speed" state ($S^*(t) = R^*$)

state 2 the "normal speed" state ($S^*(t) = R_1$)

Define the following quantities:

$$\begin{aligned} r_i(\theta) &\equiv \text{rate of change of temperature in state } i\text{, at temperature } \theta \\ \beta_i &\equiv \text{rate of heat generation in state } i \\ \gamma_i &\equiv \text{clock rate in state } i \\ \mu_i &\equiv \text{rate of a transition from state } i \text{ to state 0 for } i = 1, 2. \end{aligned}$$

Eqs. (3) and 5) imply that the rate of temperature change in state i, at temperature θ is

$$r_i(\theta) \;=\; [\beta_i - c\,\theta]/q,$$

where,

$$\beta_i \;=\; \begin{cases} 0, & i = 0 \\ h\,\gamma_i, & i = 1,\; 2 \end{cases}$$

since the heat generated during a work cycle is h. Thus,

$$r_0(\theta) \;=\; -c\,\theta/q \tag{16}$$

$$r_1(\theta) \;=\; [h\gamma_1 - c\,\theta]/q \tag{17}$$

$$r_2(\theta) \;=\; [h\gamma_2 - c\,\theta]/q \tag{18}$$

Since, in state 2, the temperature is constant at $\theta = T_1$,

$$r_2(T_1) \;=\; [h\gamma_2 - c\,T_1]/q \;=\; 0.$$

Finally,

$$\mu_i \;=\; \mu\,\gamma_i.$$

Next, define

$$\begin{array}{rcl@{\qquad}rcl} \Phi_0(\theta,t) & \equiv & \text{Prob}[X(t) = 0,\; \Theta(t) \le \theta], & \Phi_0(\theta) & \equiv & \lim_{t\to\infty} \Phi_0(\theta,t) \\ \Phi_1(\theta,t) & \equiv & \text{Prob}[X(t) = 1,\; \Theta(t) \le \theta], & \Phi_1(\theta) & \equiv & \lim_{t\to\infty} \Phi_2(\theta,t) \\ \Phi_2(t) & \equiv & \text{Prob}[X(t) = 2], & \Phi_2 & \equiv & \lim_{t\to\infty} \Phi_2(t) \end{array}$$

and assume that the limits $\Phi_0(\theta)$, $\Phi_1(\theta)$ and Φ_2 all exist. Note that $(X(t), \Theta(t); t \geq 0)$ is a Markov process [4]. Our goal is to construct the corresponding backward equations for Φ_0, Φ_1 and Φ_2. Since the amount of work, B_j, is exponentially distributed with mean $1/\mu$, and the idle periods, I_j, are exponentially distributed with mean $1/\lambda$, we have the following transition probabilities in a small interval $(t, t+\delta)$, with $i \rightarrow j$ denoting a transition from state i to state j:

$0 \rightarrow 1$: $\delta\,\lambda + o(\delta)$

$0 \rightarrow 2$: $o(\delta)$

$1 \rightarrow 0$: $\delta\,\mu_1 + o(\delta)$, if $\Theta(t) \leq T_1 - r_1(T_1)\delta$

$1 \rightarrow 2$: $1 - o(1)$, if $\Theta(t) > T_1 - r_1(T_1)\delta$

$2 \rightarrow 0$: $\delta\,\mu_2 + o(\delta)$

$2 \rightarrow 1$: $o(\delta)$

Note that the $o(\delta)$ terms cover the possibility of multiple transitions, e.g., a transition from state 0 to state 1 to state 2.

Lemma 2 *The backward equations for the limiting state probabilities, $\Phi_0(\theta)$, $\Phi_1(\theta)$, and Φ_2, are given by*

$$r_0(\theta)\,\Phi_0'(\theta) = -\lambda\,\Phi_0(\theta) + \mu_1\,\Phi_1(\theta), \quad \theta < T_1 \tag{19}$$

$$r_1(\theta)\,\Phi_1'(\theta) = -\mu_1\,\Phi_1(\theta) + \lambda\,\Phi_0(\theta), \quad \theta < T_1 \tag{20}$$

$$\mu_2\,\Phi_2 = r_1(T_1)\,\Phi_1'(T_1). \tag{21}$$

subject to the boundary condition

$$\Phi_0(T_1) + \Phi_1(T_1) + \Phi_2 = 1. \tag{22}$$

Proof: For state 0,

$$\Phi_0(\theta, t+\delta) \;=\; [1-\lambda\,\delta]\,\Phi_0(\theta - r_0(\theta)\delta, t) \;+\; \mu_1\,\delta\,\Phi_1(\theta, t) \;+\; o(\delta), \quad \theta < T_1$$

Rearranging terms above, we get

$$\frac{\Phi_0(\theta, t+\delta) - \Phi_0(\theta, t)}{\delta} + \frac{\Phi_0(\theta, t) - \Phi_0(\theta - r_0(\theta)\delta, t)}{\delta} =$$

$$-\lambda\,\Phi_0(\theta - r_0(\theta)\delta, t) \;+\; \mu_1\,\Phi_1(\theta, t) \;+\; o(1).$$

Taking limits for $\delta \to 0$ yields

$$\frac{\partial \Phi_0(\theta, t)}{\partial t} + r_0(\theta)\,\frac{\partial \Phi_0(\theta, t)}{\partial \theta} \;=\; -\lambda\,\Phi_0(\theta, t) \;+\; \mu_1\,\Phi_1(\theta, t), \quad \theta < T_1, \quad t > 0.$$

Next, take limits for $t \to \infty$. Since $\lim\limits_{t\to\infty} \dfrac{\partial \Phi_0(\theta, t)}{\partial t} = 0$, we get Eq. (19).

Similarly, for state 1,

$$\Phi_1(\theta, t+\delta) \;=\; \Phi_1(\theta - r_1(\theta)\delta, t)[1-\mu_1\,\delta] \;+\; \lambda\,\delta\,\Phi_0(\theta, t) \;+\; o(\delta), \quad \text{for } \theta < T_1$$

which yields Eq. (20) after taking limits as before. Finally, for State 2,

$$\Phi_2(t+\delta) \;=\; \Phi_2(t)[1-\mu_2\,\delta] \;+\; [\Phi_1(T_1, t) - \Phi_1(T_1 - r_1(T_1)\delta, t)] \;+\; o(\delta)$$

yields Eq. (21) in an analogous way. Note that the second term on the right-hand side is the probability of the event that the temperature is close to T_1 The condition in Eq. (22), is a consequence of the fact that the state must be either of 0, 1, or 2. ■

We are now in a position to solve for the Φ_i in terms of the incomplete beta function $B_x(a, b) \equiv \int_0^x u^a\,(1-u)^b\,du,\ 0 \le x \le 1, a > 0, b > 0.$

Theorem 2 *The solution for Eqs. (19)–(21), subject to Eq. (22), is given by*

$$\Phi_0(\theta) \;=\; \int_0^\theta \Phi_0'(u)\,du \;=\; K\frac{\beta_1^{(\lambda+\mu_1)/(qc)}}{c^{\lambda/(qc)}} B_{c\theta/\beta_1}\left(\lambda/(qc) - 1, \mu_1/(qc)\right) \tag{23}$$

$$\Phi_1(\theta) \;=\; \int_0^\theta \Phi_1'(u)\,du \;=\; K\frac{\beta_1^{(\lambda+\mu_1)/(qc)}}{c^{\lambda/(qc)}} B_{c\theta/\beta_1}\left(\lambda/(qc), \mu_1/(qc)-1\right) \tag{24}$$

$$\Phi_2 \;=\; \frac{r_1(T_1)\,\Phi_1'(T_1)}{\mu_2} \;=\; \frac{Kcq}{\mu_2} T_1^{\lambda/(qc)} \left(\beta_1 - cT_1\right)^{\mu_1/(qc)} \tag{25}$$

where K is determined by the boundary condition (22) as

$$\begin{aligned} K \;=\; \Bigg[& \frac{\beta_1^{(\lambda+\mu_1)/(qc)}}{c^{\lambda/(qc)}} B_{cT_1/\beta_1}\left(\lambda/(qc)-1, \mu_1/(qc)\right) + \frac{\beta_1^{(\lambda+\mu_1)/(qc)}}{c^{\lambda/(qc)}} B_{cT_1/\beta_1}\left(\lambda/(qc), \mu_1/(qc)-1\right) \\ & + \frac{cq}{\mu_2} T_1^{\lambda/(qc)} (\beta_1 - cT_1)^{\mu_1/(qc)} \Bigg]^{-1} \end{aligned} \tag{26}$$

Proof: Adding Eq. (19) to Eq. (20) gives

$$r_0(\theta)\,\Phi_0'(\theta) \;=\; -r_1(\theta)\,\Phi_1'(\theta). \tag{27}$$

Differentiating Equation (19) with respect to θ, and substituting $\Phi_1'(\theta)$ from Eq. (27) into the result, yields

$$\begin{aligned} r_0'(\theta)\,\Phi_0'(\theta) \;+\; r_0(\theta)\,\Phi_0''(\theta) \;&=\; -\lambda\,\Phi_0'(\theta) \;+\; \mu_1\,\Phi_1'(\theta) \\ &=\; -\lambda\,\Phi_0'(\theta) \;-\; \frac{\mu_1\,r_0(\theta)}{r_1(\theta)}\,\Phi_0'(\theta). \end{aligned}$$

Viewing this equation as a first-order differential equation in $\Phi_1'(\theta)$, we can write

$$\begin{aligned} \Phi_0''(\theta) \;&=\; -\left[\frac{r_0'(\theta)}{r_0(\theta)} \;+\; \frac{\lambda}{r_0(\theta)} \;+\; \frac{\mu_1}{r_1(\theta)}\right]\Phi_0'(\theta) \\ &=\; -\left[-\frac{\lambda q - c}{c\,\theta} \;+\; \frac{\mu_1 q}{[\beta_1 - c\,\theta]}\right]\Phi_0'(\theta), \qquad \text{from equations (16–17)} \end{aligned}$$

the formal solution for an equation of the form $f'(\theta) = g(\theta)\,f(\theta)$ is $f(\theta) = K\,\exp(\int g(x)\,dx)$. Hence, the solution for $\Phi_0'(\theta)$ is

$$\begin{aligned} \Phi_0'(\theta) \;&=\; K\,\exp\left(\frac{\lambda q - c}{c}\,\ln(\theta) \;+\; \frac{\mu_1\,q}{c}\,\ln(\beta_1 - c\theta)\right) \\ &=\; K\,\theta^{\lambda\,q/c-1}(\beta_1 - c\theta)^{\mu_1\,q/c}. \end{aligned}$$

Next, from Eq. (27),

$$\Phi_1'(\theta) \;=\; Kc\,\theta^{\lambda q/c}(\beta_1 - c\theta)^{\mu_1 q/c-1}.$$

Since $\Phi_0(0) = \Phi_1(0) = 0$, this now leads to the result. ■

We conclude this section by noting that the performance measure of control policy S is the mean rate of work cycles completed under it, namely,

$$\omega_S \;=\; \gamma_1\,\Phi_1(T_1) \;+\; \gamma_2\,\Phi_2, \tag{28}$$

and this measure will be used to compare the performance of a VSP and a SSP.

4 Controlling the VSP with Linear Temperature Gradients

We begin this section with the observation that it is quite difficult to monitor the temperature of a processor circuitry so as to determine what clock rate to use. A more practical alternative is to conservatively approximate the temperature by assuming that the rate of temperature change is constant in each state; this would permit us to estimate the temperature using a counter which increments or decrements the temperature appropriately every clock cycle.

In this section, we compute the mean rate of work, using one such scheme. Let $\hat{\Theta}(t)$ denote our estimate of the temperature $\Theta(t)$, and compute $\hat{\Theta}(t)$ using the following rule:

$$\hat{\Theta}(0) \;=\; T_1$$

$$\hat{\Theta}'(t) \;=\; \begin{cases} l_0, & \text{if } T \le \hat{\Theta}(t) < T_1 \text{ and } X(t) = 0 \\ l_1, & \text{if } T \le \hat{\Theta}(t) < T_1 \text{ and } X(t) = 1 \\ 0, & \text{if } \hat{\Theta}(t) = T_1 \text{ and } X(t) = 1 \\ 0, & \text{if } \hat{\Theta}(t) = T \text{ and } X(t) = 0 \end{cases}$$

where $l_0 = r_0(T)$ and $l_1 = r_1(T)$ for some $T \in [0, T_1]$. T is a parameter called the *base temperature,* which may be chosen in order to optimize performance. Assuming that $\Theta(0) \le$

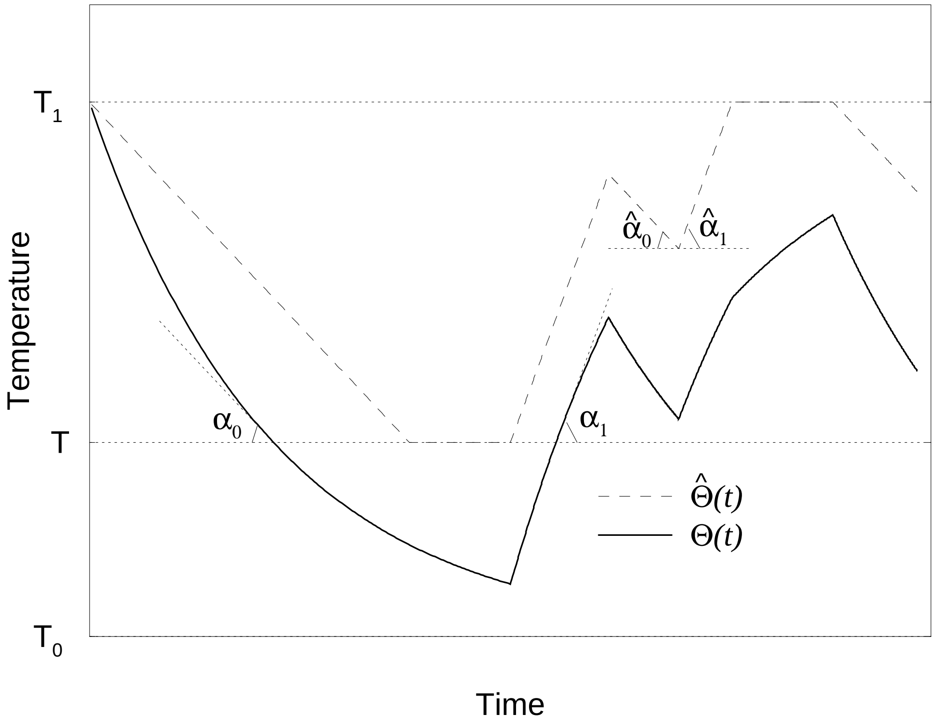

Figure 2: Sample paths of $\Theta(t)$ and $\hat{\Theta}(t)$

$T \le T_1$, it is not difficult to show that if we use the optimal policy with $\hat{\Theta}(t)$ as the estimator of $\Theta(t)$, then $T_1 \ge \hat{\Theta}(t) \ge \Theta(t)$. This is shown in Figure 2, which depicts sample paths for $\Theta(t)$ and $\hat{\Theta}(t)$. For this illustration, we assumed that $\Theta(0) = T_1$ and that the processor has just become idle at time 0. Note that the slopes of the two sample paths are related by $\alpha_0 = \hat{\alpha}_0$ and $\alpha_1 = \hat{\alpha}_1$. From this figure, it is clear that $\hat{\Theta}(t) \ge \Theta(t)$ for all $t \ge 0$.

To compute the mean rate of instruction cycles executed using this policy, we proceed as in the previous section. Define the state function

$$Y(t) = \begin{cases} 0, & \text{if } T \le \hat{\Theta}(t) < T_1, \text{ processor idle} \\ 1, & \text{if } T_1 \le \hat{\Theta}(t) < T_1, \text{ processor busy} \\ 2, & \text{if } \hat{\Theta}(t) = T_1, \text{ processor busy} \\ 3, & \text{if } \hat{\Theta}(t) = T, \text{ processor idle} \end{cases}$$

and the probability distributions

$$F_i(x,t) = \text{Prob}[Y(t) = i, \hat{\Theta}(t) \le x], \quad i \in \{0,1\}$$

$$\begin{aligned} F_i(x) &= \lim_{t\to\infty} F_i(x,t), \quad i \in \{0,1\} \\ G_i(t) &= \text{Prob}[Y(t) = i], \quad i \in \{2,3\} \\ G_i &= \lim_{t\to\infty} F_i(t), \quad i \in \{2,3\} \end{aligned}$$

Writing the backward equations for these probability functions yields,

$$-l_0 F_0'(x) = -l_0 F_0'(T) + \lambda F_0(x) - \mu_1 F_1(x) \tag{29}$$

$$l_1 F_1'(x) = \lambda G_3 - \mu_1 F_1(x) + \lambda F_0(x) \tag{30}$$

$$\mu_2 G_2 = l_1 F_1'(T_1) \tag{31}$$

$$\lambda G_3 = -l_0 F_0'(T) \tag{32}$$

Additionally, we have the boundary condition

$$F_0(T_1) \; + \; F_1(T_1) \; + \; G_2 \; + \; G_3 \; = \; 1.$$

Combining equations (29),(30) and (32), we have $-l_0 F_0'(x) = l_1 F_1'(x)$. Let $\eta(x) = l_1\, F_1'(x) = -l_0\, F_0'(x)$ and $\psi \equiv (\lambda/l_0 + \mu_1/l_1)$. Then

$$\eta'(x) \; = \; -\mu_1\, F_1'(x) \; + \; \lambda\, F_0'(x) \; = \; -[\mu_1/l_1 \; + \; \lambda/l_0]\eta(x) \; = \; -\psi\, \eta(x).$$

Therefore,

$$\begin{aligned} \eta(x) &= C\, \exp(-\psi x) \\ F_0(x) &= \int_T^x \frac{-C}{l_0} \exp(-\psi u)\, du \; = \; -\frac{C}{l_0\, \psi}\, [\exp(-\psi T) - \exp(-\psi x)] \\ F_1(x) &= \int_T^x \frac{C}{l_1} \exp(-\psi u)\, du \; = \; \frac{C}{l_1\, \psi}\, [\exp(-\psi T) - \exp(-\psi x)] \\ G_2 &= \frac{\eta(T_1)}{\mu_2} = \frac{C\, \exp(-\psi T_1)}{\mu_2} \\ G_3 &= \frac{\eta(T)}{\lambda} = \frac{C\, \exp(-\psi T)}{\lambda}. \end{aligned}$$

Parameter	Denoted	Value
package heat capacity	q	0.08541 Joules/^oC
heat generated per cycle	h	1.36364 Joules/cycle
package heat conductance	c	0.08182 Watts/^oC
normal clock rate		33 MHz
maximum junction temperature		100^oC
maximum ambient temperature		50^oC

Table 1: Prototype processor parameter values

Here C can be determined by normalizing,

$$C = \left[\frac{\exp(-\psi T)}{\lambda} + \frac{\exp(-\psi T_1)}{\mu_2} + \frac{[\exp(-\psi T) - \exp(-\psi T_1)]}{-l_0\,\psi} + \frac{[\exp(-\psi T) - \exp(-\psi T_1)]}{l_1\,\psi} \right]^{-1}.$$

The average rate of instruction cycles executed is given by

$$\omega_S \;=\; \gamma_1\,F_1(T) \;+\; \gamma_2\,G_2.$$

5 Numerical results

This section presents some numerical results comparing the performances of a SSP (hereafter referred to as the *baseline* SSP), a VSP using the optimal bang-bang control (hereafter, to be referred to as the *optimal* VSP), and a VSP using a linear gradient control (hereafter referred to as the *linear* VSP). The prototype processor characteristics are based on an Intel DX486 microprocessor [3] running at 33 MHz: this is assumed to be the clock rate at which the rate of heat dissipation by the package equals the rate of heat generation by the circuit. In our VSP prototype, however, we assume that a faster, *turbo* clock speed is available. Relevant physical parameters of the prototype and the environment are given in Table 1. In the graphs below, the metric of performance is normalized throughput, which is the ratio of the mean number of instruction-cycles performed per second to the normal clock rate, under the workload specified. (Clearly, the normalized work rate can be at most unity for stable operation; otherwise the rate of heat generation exceeds the rate of heat dissipation.)

Figure 3 displays the effect of the base temperature, T, on the linear VSP, for three workloads. The normalized throughputs for the optimal VSP and the baseline SSP are also given for the three workloads; since the choice of T does not affect these, the throughputs do not change with T. The workloads were chosen so that the throughput of the baseline SSP is the same for all three cases; thus, the normalized throughputs for all three workloads constitute an identical line, in the case of the baseline SSP. For the first worload, which consists of short workbursts followed by short idle periods, the performance of the linear VSP matches that of the optimal VSP when the base temperature is chosen to be close to the maximum operating temperature — between 94^oC and 98^oC. For the second workload, with longer workbursts and idle periods, the optimal choice of T is lower, around 93^oC, and yet lower for the third workload, with long workbursts and idle periods. We assume that a workload of short to intermediate length workbusts and idle periods is more likely than one with long workbursts and idle periods, and select $T = 95^oC$ for the remaining plots.

Figure 3 can also be used to study the effect of the granularity of the workload on the performance of the optimal VSP and the linear VSP. A workload consisting of short workbursts, followed by short idle periods is said to have a *fine granularity*. On the other hand, a workload of long workbursts followed by long idle periods has a *coarse granularity*. To vary the granularity of the workload, we varied the length of the workburst while holding fixed the ratio of the workburst length to the idle period length. The throughput of the baseline SSP depends only on this ratio, and thus is unchanged across the three workloads, thereby providing a baseline for the comparison of the performance of the other two processors. We note that the performance of the optimal VSP and the linear VSP improves as the granularity of the workload decreases. This observation agrees with intuition: when the idle periods are short, the temperatures of the optimal and the linear VSP's remain close to the maximum operating temperature, and the rate of heat disspations is close to the maximum rate. Since the average throughput is proportional to the rate of heat dissipated, a workload with fine granularity increases the average throughput. For workloads with longer idle periods, the variation in temperature is larger, and the heat dissipation can at times be low, leading to lower average work rates. This also explains why high values of the base temperature T improve performance for workloads with fine granularity, but not for those with coarse granularity. When the idle periods are short, the actual temperature, $\Theta(t)$, as well as the estimated temperature, $\hat{\Theta}(t)$, stay close to the maximum operating temperature, T_1. In this case, a choice of a large T would improve the performance of the linear VSP, since the estimated temperature gradients, $l_0 = r_0(T)$ and $l_1 = r_1(T)$, closely match the actual temperature gradients. However, when the granularity of the workload is coarse, the temperature variation is large. Furthermore, when the estimated temperature falls below T, the estimated temperature gradients are too conservative, and consequently performance suffers. Viewed differently, the long idle periods in a workload of coarse granularity "waste" the potential speed-up of any VSP, as compared to the baseline SSP.

Figure 4 compares the performance of the optimal VSP and the linear VSP as the turbo speed is increased. The throughput of the baseline SSP (which is unaffected by the turbo speed,) again provides a baseline for the comparison. Based on the discussion above, $T = 95^oC$ was chosen for the linear VSP. A workload of moderate granularity was selected, with workbursts of the average length 10^6 cycles, and average idle periods of $20ms$. For a speed-up factor of 2 (i.e., the turbo clock rate is twice the normal clock rate), the throughput of the optimal VSP is 40% higher than the baseline SSP, whereas the corresponding improvement for the linear VSP is 36%. As the turbo clock speed is increased further, the marginal performace gains become smaller. For speed-up factors of 4, 8 and 16, the throughput improvements over the baseline SSP are, respectively, 57%, 60% and 61% for the optimal VSP and 47%, 50% and 51% for the linear VSP.

In Figure 5, the mean workburst length is held constant at 10^6 cycles, while the mean idle period is allowed to grow. Intuitively, we expect that the normalized throughput would drop, as the mean idle period increases; indeed, the results confirm this intuition. However, the optimal VSP can compensate for the increased idle periods to a considerable extent, so long as the idle periods are not too long. The normalized throughput for the optimal VSP is close to the maximum, 1, for idle periods of mean length not exceeding $10ms$. The performance of

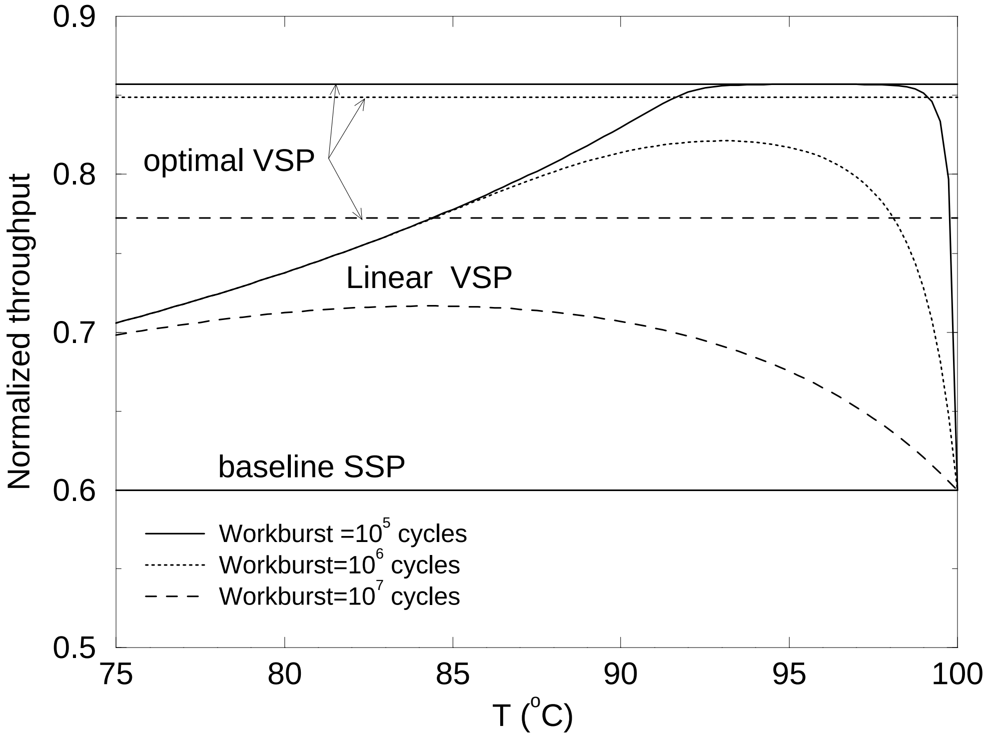

Figure 3: Effects of changing granularity and base temperature on the normalized throughput.

(Turbo clock rate = 66MHz)

the linear VSP is only slightly poorer. For idle periods with mean length $20ms$, the optimal VSP and linear VSP are effectively 42% and 36% faster, repectively, than the baseline SSP. As the idle periods exceed $100ms$, the differences between the optimal VSP, linear VSP and the baseline SSP decay to zero. This, too, is intuitive: as the idle period grows, the limiting normalized throughputs of all three processors should approach zero, regardless of the speed at which the workbursts are executed.

6 Implementation and applications

This section presents a simple implementation of the VSP in any common CPU. Also we suggest some possible applications using the principle of VSP.

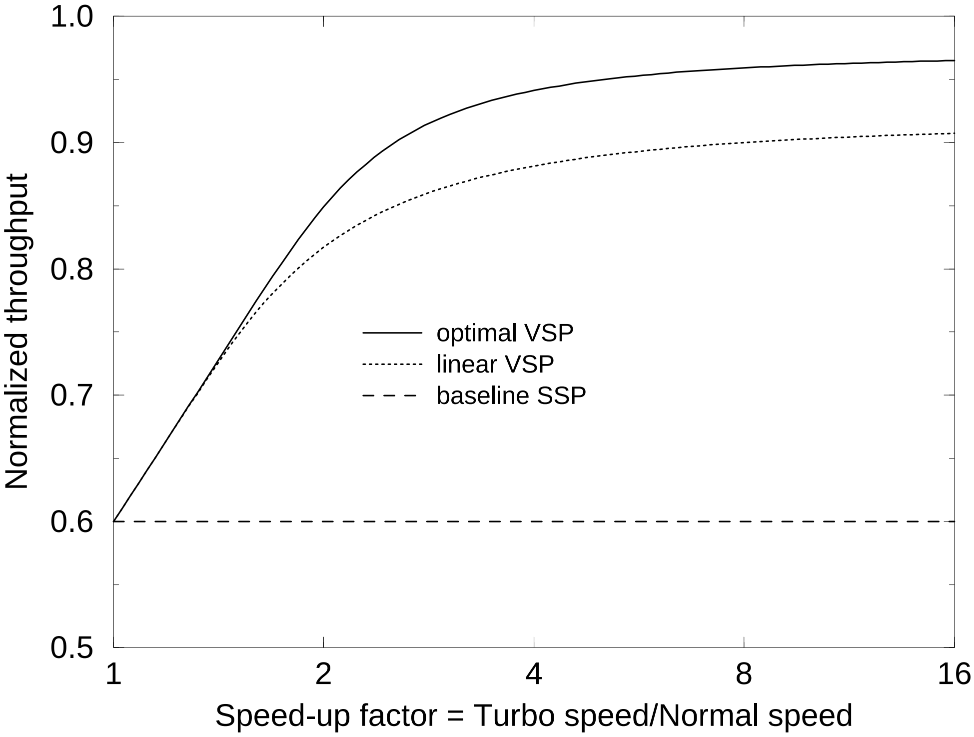

Figure 4: Effects of changing the turbo clock rate on the normalized throughput. (T=95^oC, Workburst=10^6 cycles, Idle period = 20ms)

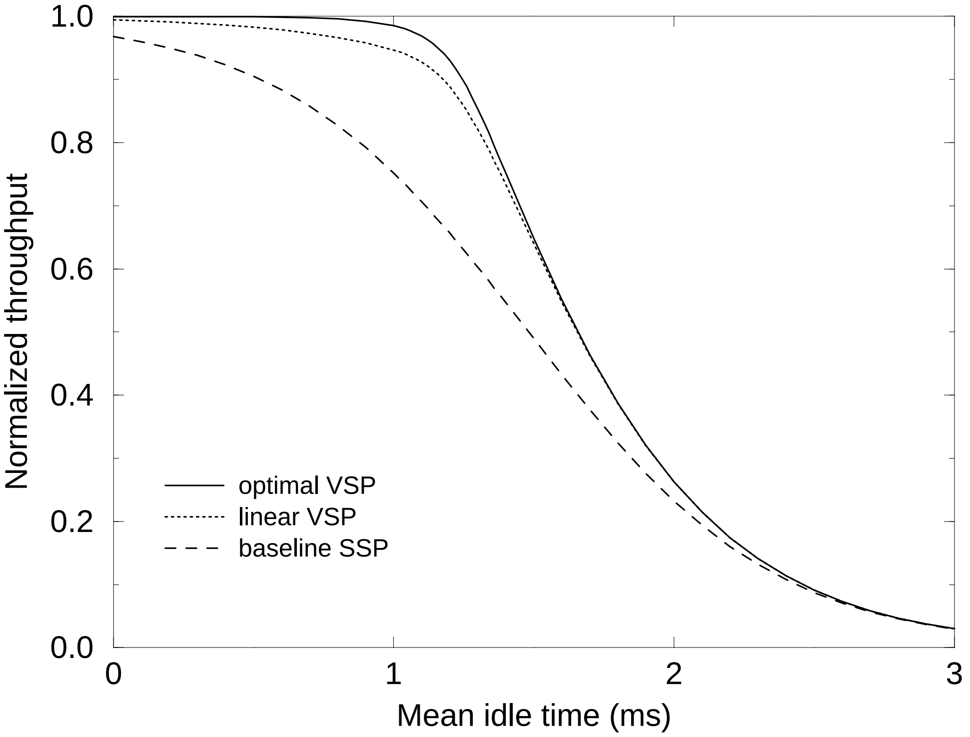

Figure 5: Effects of changing the mean idle period on the normalized throughput. (T=95^oC, Workburst=10^6 cycles, turbo clock rate=66MHz)

6.1 Heat bank using a counter

A simple up-down counter circuit, can accumulate the gain and loss in terms of heat production units, while operating at a certain clock rate. An upper bound is imposed on the maximum heat savings possible to gain (i.e., even if the circuit was idle for one hour it does not mean it can be operated at twice the normal clock rate for the next hour), based on the given thermal characteristics of the particular chip package. A special circuit has to detect when the circuit can be operated in a lower clock rate (e.g. when it is idling or waiting for external events to occur). This circuit monitors the activity of the various units in the chip and change their corresponding clock speed, while accounting for the heat saved this way. The particular activity that is monitored, depends on the circuit design. For a CPU chip, this activity can be the specific instructions executed at any given time. An instruction decoder can be used to detect instructions that take a long time to complete using an external reference (e.g. a floating point numerical coprocessor), while the main CPU chip is idle, waiting for the instruction to be finished. For the time these instructions are executed, the main CPU chip is switched to the minimal clock rate and while accounting for the heat generation saved. Another type of monitoring is observing bus activity. A circuit detects the idle periods on the bus (i.e. a DMA transfer, forcing the CPU into a “hold” state; or a “not ready” signal).

If there is room to operate with higher clock rates (e.g. useful work is being done) and there are enough ”heat savings in the bank“ (the value of the up-down heat counter is high enough), a synchronous shift to a higher clock rate can take place.

A schematic drawing of the variable clock implementation mechanism is presented in Figure 6.

The components of the possible implementation as pictured in Figure 6 are as following:

Activity Monitor: The activity monitor can be simply a control line (e.g., a “wait” line

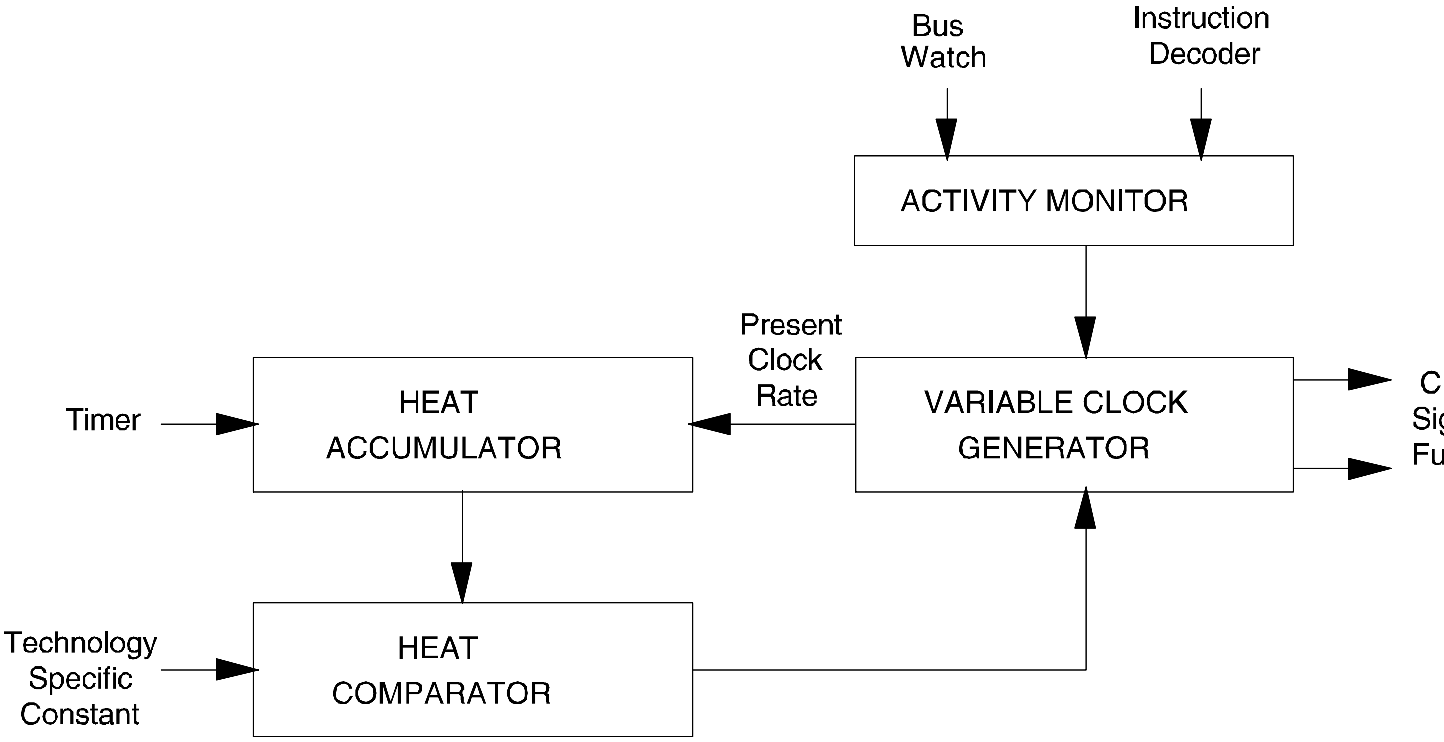

Figure 6: The VSP implementation

in a cpu microprocessor, by which external world signals the cpu to wait because of external delays). Or it can be a logical “and” function (implemented with an “and” gate) of several such lines (e.g., “wait”, “hold”, as well as other synchronization control lines). In a cpu, some instructions have different execution requirements so that parts of the cpu chip are not needed (for example, when performing an integer operation, there is no need for the floating point circuits to be operational). The Activity Monitor, can also include a table for such instructions, so that only the needed circuits are operational, while other parts of the chip do not generate heat.

Heat Accumulator: This is an up-down binary counter. The counter accumulates heat savings (counts up) when there is no activity in the circuit (and its clock rate was reduced to save heat generation). The counter counts down, when some activity is present and the clock rate was increase to enhance performance.

Heat Comparator: The maximum savings of the Heat Accumulator, is limited to a certain level. This level can be specific to the packaging technology used for the circuit. When enough heat was saved, the Heat Comparator indicates that the clock rate can be increased (pending on the availability of circuit activity).

Variable Clock Generator: Several possible operational frequencies are selected by this unit according to the activity and the heat savings. If a functional unit in the chip is to be operational, and enough heat savings were made in the past (as the Heat Comparator indicates), this unit will operate at a higher clock rate. If a functional unit is idle or waiting (as indicated by the Activity Monitor), this unit will operate at the lowest clock rate possible. The state of clock rates is transmitted to the Heat accumulator, so that up or down counting (accumulation) can take place.

6.2 Additional applications

The improvement on overall chip throughput depends on the characteristics of the particular application. For example, consider a CISC type processor made of one CPU VLSI chip and a second external chip working as a math co-processor. In a particular program, the main CPU may have to wait for the math co-processor to finish a relatively long floating point operation before it can continue. During this period, we can save heat generation by lowering the main CPU clock rate to the minimum imposed by the technology. After the floating point operation is finished, the following main CPU instructions can be executed with a higher clock rate, speeding up the overall performance. The ratio of the idling period to the active period, and the interleaving instruction ratio (i.e. the mix between idling and active periods, without having a long active period followed by a long idling one), will determine the overall gain in performance.

The VSP principle can be used in a more wider aspect. Consider a parallel computer architecture made of many "one chip" CPUs. The interconnection network can be (and is in most cases) the performance bottleneck due to long latency times present for global information interchange. The cause for this time latency might be due to the interconnection network performance or because of concurrently accessing common global data needed by a certain parallel program. Using the principle of varying the clock rates of each CPU, it is possible to make up for the lost waiting time, resulting in an overall performance as if there was no time latency accessing the global network.

Transaction processing is another good example. The nature of the application is characterized by short burst type operations, with idling periods in between.Increasing the clock rate while processing the transactions, is possible if the circuits can rest (i.e. producing no heat) while they are idling.

The VSP principle can be applied for any electronic circuit that is performance bounded by the heat dissipation ability of its package, as long as the nature of the application using this circuit, displays idling periods.

7 Conclusions

We have examined the idea of using a variable speed clock to ameliorate losses in performance, due to cache misses and other delays, in processors where the speed is limited by the ability of its packaging to remove heat. An alternative viewpoint is that a variable speed clock makes it possible to improve performance by taking advantage of "naturally occurring" idle periods in the workload. This technique can be applied to any microprocessor which is limited by the heat-removal capacity of its packaging. We defined a workload model consisting of workbursts alternating with idle periods, and proved the optimality of the following bang-bang control policy: if the processor has not reached the maximum temperature, then run it at the maximum possible speed; otherwise use the clock speed which will maintain this maximum temperature. At a more practical level, we examined a policy in which an upper-bound estimate of the junction temperature is used instead of the actual temperature, since this may be simpler to implement. Numerical studies of both policies found that the performance of a VSP is significantly better than that of a comparable baseline SSP. Furthermore, the study suggests that a VSP with a maximum clock rate of 2–4 times that of the SSP would suffice to obtain the bulk of the performance improvement. In many cases, the average throughput gain is on the order of 40–60%, without exceeding thermal limits.

References

[1] Michael Fitzgerald, *"Everex set to show* 50 *MHz PC"*, Computerworld, Vol. XXIV, No. 41, pp. 4, October 8, 1990.

[2] P. Horowitz and W. Hill. *The Art of Electronics.* Cambridge University Press, 1989.

[3] Intel Corp. *Intel486 DX Microprocessor Data Book*, October 1992. Order no. 240440-005.

[4] S. Karlin and H. M. Taylor. *A First Course in Stochastic Processes.* Academic Press Inc., 1975.

[5] Nico Krohn, *Intel to Introduce* 20-*MHz Chip Set.* InfoWorld, Vol. 12, Issue 42, pp. 5, October 15, 1990.

[6] C. Georgiou, T. Larsen, and E. Schenfeld. *Variable Chip-Clocking Mechanism*, U.S. Patent number 5,189,314, February 23, 1993.