

# Low-Power meets High-Performance – Dynamic Voltage Scaling on the CARMEL Signal Processor

**Arnold Maier**

maier.external@infineon.com  
Infineon Technologies  
Design Center Villach  
AUSTRIA

**Paul Fugger**

paul.fugger@infineon.com  
Infineon Technologies  
Design Center Villach  
AUSTRIA

**Bernhard Rinner**

rinner@iti.tu-graz.ac.at  
Institute for Technical Informatics  
Graz University of Technology  
AUSTRIA

**Abstract** – Reducing the power consumption is one of the primary design goals for many digital circuits, systems and applications such as mobile devices and high-performance computing systems. Power efficiency is also a major concern for digital signal processors (DSP) since they often combine the requirements of mobile and high-performance computing devices.

In this paper we demonstrate the effect of low-power design strategies on the CARMEL DSP core from Infineon Technologies. We focus on reducing the energy consumption by dynamic voltage scaling (DVS), i.e., the supply voltage and clock frequency of the DSP core is adjusted at runtime. Simulation results show that the energy consumption can be reduced to a quarter compared to the standard implementation.

**Keywords:** Low-power; CMOS design; dynamic voltage scaling; digital signal processor; CARMEL

## I. INTRODUCTION

Reducing the power consumption is an emerging trend in the design of many digital circuits, systems and applications. There are various examples where power efficiency is essential. In portable systems, efficient low-power design strategies need to be implemented to prolong the runtime due to the limited energy capacity of batteries. In complex high-performance systems, limiting power consumption is paramount because the power density per chip is approaching the physical limit for heat dissipation. Fig. 1 summarizes the different aspects for low-power design.

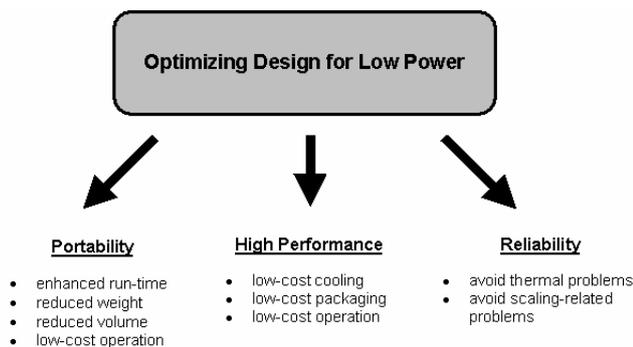


Fig. 1. Motivation for Low-Power Design [11].

An important area for low-power design is digital signal processing. Nowadays signal processing is ubiquitous and the application areas for digital signal

processors (DSP) span from handheld devices to dedicated high-performance computing systems. A DSP combines often the requirements for portable and high-performance computing. Power efficiency is, therefore, very important for DSPs.

Increasing speed, minimizing silicon area and power consumption are essential challenges in designing digital CMOS circuits. In a typical CMOS circuit, the dynamic charging and discharging of the capacitance caused by switching activities dominate the overall power dissipation. Thus, a majority of low-power design methods is dedicated to minimize this predominant factor of power dissipation [9].

The dynamic switching power  $P_{switching}$  depends on the switching activity  $\alpha$ , the node capacitance  $C_L$ , the clock frequency  $f_{clk}$  and the supply voltage  $V_{DD}$  (Equation (1)).

$$P_{switching} = \alpha \cdot C_L \cdot f_{clk} \cdot V_{DD}^2 \quad (1)$$

Lowering the supply voltage  $V_{DD}$  is an effective way for power reduction. However, if the supply voltage is lowered, the performance of a circuit also degrades due to the increased switching delay time  $t_D$ . Thus, CMOS designs usually require a trade-off between power and timing costs.

Equation (2) expresses the delay  $t_D$  as a function of the supply voltage  $V_{DD}$  and the transistor threshold voltage  $V_T$  for a typical CMOS device:

$$t_D \propto \frac{V_{DD}}{(V_{DD} - V_T)^2} \quad (2)$$

Note that power and energy are not the same. For example, even if the *power* consumption of one processor is as twice as high than the one from a competitor, its *energy* consumption could actually be less if the same program can be executed much faster [1].

The remainder of this paper is organized as follows. Section II briefly summarizes power management methods already applied in many DSPs. Section III introduces the concept of dynamic voltage scaling (DVS). Section IV presents the low-power design of the CARMEL DSP and demonstrates the effect of DVS. Section V concludes the paper with a discussion on related and future work.

## II. DYNAMIC POWER MANAGEMENT IN DSPs

In many signal processing applications such as voice

recognition, coding and image processing, the processors' performance requirements generally vary over runtime. As a result, the processor does not always need to run at its peak performance and waste energy.

This time-variance on the performance requirement has been exploited by many DSPs by the introduction of different *power modes* [6]. Depending on the actual workload the processor can be set into an active and (several) low-power modes. Low-power modes are typically achieved by switching off the clock of the DSP core or its peripherals. Several DSPs further allow the programmer some degree of freedom over the DSP's master clock frequency. This represents a compromise between full-speed operation and a low-power mode, since the DSP is still executing, albeit at a reduced rate.

The different power modes are usually entered by executing a special instruction, by setting a bit in a control register or by providing a signal on an external pin. For the transition between two power modes, there is some latency which may be a concern for some applications.

As an example, we briefly present the different power modes of the *CARMEL DSP* from Infineon Technologies:

- **ACTIVE:** The processor core is running at its highest performance mode.
- **LOW POWER:** The processor core performs only NOP's and consumes about 20 % of the usual power consumption.
- **STOP:** The processor's clock is stopped and only leakage current is dissipated.
- **IDLE:** The processor core is in the STOP mode until there is a DMA request.

### III. DYNAMIC VOLTAGE SCALING

*Dynamic Voltage Scaling* (DVS) is a method to improve a processor's energy efficiency by setting the performance level to the minimum that is required by an application. In essence, the processors supply voltage  $V_{DD}$  and clock frequency  $f_{clk}$  is adjusted *at runtime* without limiting peak performance. As long as a task completes by its deadline, the processor speed can be slowed down without limiting performance [1]. DVS aims in completing the tasks just on time, thereby minimizing the overall energy consumption.

Note that the task's deadline is usually determined by the environment of the applications. In interactive applications, response times below 50 to 100 ms do not affect user think time [12]. The deadline for handling a user interaction event can, therefore, be assumed as 50 ms. In real-time applications, the deadlines are directly determined by the timing requirements of the environment. In signal processing applications, the deadline is often determined by the sampling rate.

The key question in DVS is how the adjustment of the supply voltage and clock frequency is done at run time. This can be seen as a form of *scheduling* with the objective to minimize the overall energy consumption while satisfying each task's deadline. A schedule in DVS is given as the pair of supply voltage/clock frequency for

each task. The order in which the tasks are executed is not affected by the voltage scheduling strategy.

The effect of voltage scheduling is dependent on the number of available voltage/frequency pairs. With more pairs available a better energy reduction can be achieved. If we can use an arbitrary voltage/frequency setting the energy consumption can be reduced to an optimum.

Basically, DVS combines two important equations of CMOS design that are based on Equations (1) and (2):

$$E_{op} \propto C_L \cdot V_{DD}^2 \propto V_{DD}^2 \quad (3)$$

$$f_{max} \propto \frac{(V_{DD} - V_T)^2}{V_{DD}} \quad (4)$$

Equation (3) expresses the circuit's energy per operation while in Equation (4) the maximum usable clock frequency for a CMOS device is given. There is a limit on the reduction of the supply voltage in CMOS circuits which can only operate down to  $V_{DD} \cong 2 \cdot V_T$ .

### IV. THE CARMEL LOW-POWER IMPLEMENTATION

This section presents the results of a low-power redesign of the CARMEL DSP. We first show the achieved power reduction using "standard" low-power design methods applied to the CARMEL core [2]. We then demonstrate the effect of DVS using a typical DSP benchmark.

#### A. CARMEL Low-Power Redesign

The CARMEL is a licensable high-performance 16-bit DSP core from Infineon Technologies primarily targeted for System-on-Chip solutions. The processor core contains about 1 million transistors [3]. It is available as a VHDL library which was also the starting point for our low-power redesign.

The low-power redesign was realized using *Synopsys* synthesis tools. *Clock gating* [2] was applied as the main low-power strategy in this redesign. The layout was generated with the APOLLO tool from Avant. The power consumption was computed at the transistor level using the POWERMILL simulation tool from *Synopsys*. This results in an accurate computation of the power consumption, i.e., the deviation to the actual power consumption lies within 2 % of the computed power consumption.

TABLE I - LOW-POWER CARMEL REDESIGN

Carmel	$V_{DD}$	$f_{clk}$	$P_{total}/TI$	$P_{total}/EFR$
standard implementation	1.8 V	100 MHz	51.94 mW	103.03 mW
with gated clocks	1.8 V	100 MHz	40.89 mW	75.45 mW
Reduction			<b>21.3 %</b>	<b>26.8 %</b>

The results of the low-power redesign using gated clocks are summarized in Table I. This table compares the

computed power consumption of the standard CARMEL implementation with the low-power redesign. Two different test patterns were used for this comparison. TI is the standard test-pattern used by various chip vendors and represents an “average” workload for the DSP. Enhanced full rate (EFR) is a high workload test-pattern. A significant power reduction was achieved with both test patterns.

### B. CARMEL Operating Limits

Given CARMEL’s low-power redesign we evaluated the limits on its operation with respect to the supply voltage and the clock frequency. This evaluation is a prerequisite in determining different active power modes required for dynamic voltage scaling. Table II presents the computed power consumption of the CARMEL low-power redesign with various values of  $V_{DD}$  and  $f_{clk}$ . This evaluation is based on the EFR test-pattern.

TABLE II - CARMEL POWER SIMULATION RESULTS

$V_{DD}$	150 MHz	100 MHz	50 MHz	10 MHz
1.8 V	113.17mW	75.45 mW	n.a.	n.a.
1.7 V	t. v.	65.76 mW	n.a.	n.a.
1.6 V	t. v.	57.77 mW	n.a.	n.a.
1.5 V	t. v.	49.38 mW	n.a.	n.a.
1.4 V	t. v.	41.76 mW	n.a.	n.a.
1.3 V	t. v.	35.68 mW	17.27 mW	n.a.
1.2 V	t. v.	t. v.	14.39 mW	n.a.
1.1 V	t. v.	t. v.	11.87 mW	n.a.
1.0 V	t. v.	t. v.	t. v.	1.99 mW

The operating limit for a given clock frequency is given as the smallest supply voltage which does not result in a timing violation (t.v.). The power consumption was not simulated for all  $V_{DD}/f_{clk}$  pairs (n.a.).

TABLE III - CARMEL POWER MODES USED FOR DVS

Task	$V_{DD}$	$f_{clk}$	MIPS	Norm. Energy
ACTIVE 1	1.8 V	150 MHz	600	0.754 nJ/Cycle
ACTIVE 2	1.4 V	100 MHz	400	0.417 nJ/Cycle
ACTIVE 3	1.2 V	50 MHz	200	0.288 nJ/Cycle
ACTIVE 4	1.0 V	10 MHz	40	0.199 nJ/Cycle

Based on this power simulation four different active power modes have been chosen for our DVS evaluation. Table III presents these power modes referred to as ACTIVE 1 to ACTIVE 4 as well as their supply voltage, clock frequency, computation performance and normalized energy consumption.

### C. DVS Evaluation of the CARMEL DSP

TABLE IV - BENCHMARK CHARACTERISTICS

Task	Description	Cycles
1	1024-point FFT	103400

2	80-tap complex FIR	167936
3	80-tap FIR	40960
4	40-tap complex FIR	83968
5	1024-point FFT	103400
Total		<b>499664</b>

We demonstrate the effect of DVS on the CARMEL DSP based on a typical benchmark. This benchmark consists of five different signal processing tasks. These tasks have to be executed sequentially. Table IV presents these tasks and the number of cycles required for their execution [4].

In this evaluation we assume that this benchmark is used for an audio application with a sampling rate of  $f_{sr}=44.1$  kHz. The deadline for the completion of this benchmark is given by the sampling rate and the block size of 1024. We assume that the benchmark should be completed before the next block is completely sampled. Thus, the overall deadline given as 23.22 ms.

Fig. 2 shows the execution times of the benchmark using the highest performance mode (ACTIVE 1). The benchmark completes after 3.33 ms. Table V presents the energy consumption of each task of this benchmark. The overall energy consumption is simulated as 376.73  $\mu$ J.

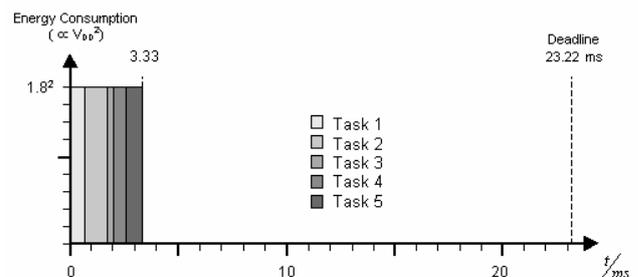


Fig. 2. Benchmark Execution using Mode ACTIVE 1

TABLE V – ENERGY CONSUMPTION OF EACH TASK

Task	Mode	Norm. Energy	Energy
1	ACTIVE 1	0.754 nJ/Cycle	77.96 $\mu$ J
2	ACTIVE 1	0.754 nJ/Cycle	126.62 $\mu$ J
3	ACTIVE 1	0.754 nJ/Cycle	30.88 $\mu$ J
4	ACTIVE 1	0.754 nJ/Cycle	63.31 $\mu$ J
5	ACTIVE 1	0.754 nJ/Cycle	77.96 $\mu$ J
Total Energy			<b>376.73 <math>\mu</math>J</b>

As shown in Fig. 2, peak computing performance for this benchmark is not required. The execution of the individual tasks in more energy efficient modes is possible. The problem is to find the optimal schedule for this benchmark. Running five tasks in four different modes results in a total of 625 ( $5^4$ ) possible schedules. However, only schedules that complete the benchmark within the deadline of 23.22 ms are accepted.

Fig. 3 depicts the optimal schedule which has been derived by an exhaustive search among all possible schedules. By using slower but more energy efficient modes, the overall completion time for the optimal schedule is given as 21.55 ms. The energy consumption is

significantly reduced to about 131.40  $\mu\text{J}$  (Table VI).

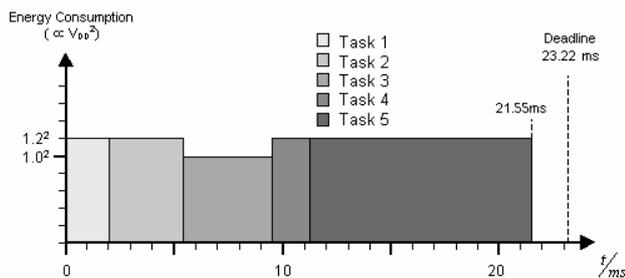


Fig. 3. The Optimal Voltage Schedule for the Benchmark

TABLE VI – ENERGY CONSUMPTION OF EACH TASK

Task	Mode	Norm. Energy	Energy
1	ACTIVE 3	0.288 nJ/Cycle	29.78 $\mu\text{J}$
2	ACTIVE 3	0.288 nJ/Cycle	48.36 $\mu\text{J}$
3	ACTIVE 4	0.199 nJ/Cycle	8.51 $\mu\text{J}$
4	ACTIVE 3	0.288 nJ/Cycle	24.18 $\mu\text{J}$
5	ACTIVE 4	0.199 nJ/Cycle	20.57 $\mu\text{J}$
		Total Energy	<b>131.40 <math>\mu\text{J}</math></b>

This simple example dramatically demonstrates the effect of dynamic voltage scaling on the energy consumption of a DSP. Dynamic voltage scaling reduces the energy consumption of the CARMEL core to 34.8 % compared to the peak performance execution in ACTIVE 1 mode. An even better result is achieved when we compare the energy consumption with the CARMEL standard implementation, i.e. the energy consumption is reduced to a quarter of the standard implementation (25.5 %).

## V. DISCUSSION

In this paper we have presented a low-power redesign of the CARMEL DSP and have demonstrated the effect of dynamic voltage scaling on the overall energy consumption. The evaluation of this design is based on the simulation of the power consumption at the transistor level. When DVS is applied the overall energy consumption can be significantly reduced as demonstrated on a DSP benchmark.

There are two major areas for future work. First, in order to apply dynamic voltage scaling the power management unit of the CARMEL DSP must be enhanced such that the supply voltage can be altered at runtime. Second, a voltage scheduling algorithm needs to be implemented such that the performance modes can be adjusted at runtime.

Finding a voltage scheduling algorithm that results in an optimal energy efficiency may be quite difficult. There exist several different approaches in the literature and some of them have already been implemented. In general, the scheduling algorithm can be divided into two parts called *prediction* and *speed-setting*. Whenever a new task begins, the prediction part predicts the processors workload for this task. This information is handed on to the speed-setting part to set the processor speed for the current interval [5].

To implement an optimal voltage scheduling algorithm comprising an exact setting of voltage and frequency, individual knowledge of deadlines and current processor utilization is required. The key difficulty with such an algorithm is that it always requires exact knowledge about future workload to set the right speed. This problem makes this method quite impractical. Because of that, improving a scheduling algorithm mainly consists of developing better prediction methods such as “*Past*”, “*Aged- $\alpha$* ”, “*LongShort*” and “*Flat-U*” [10].

A recent and interesting approach of improving DVS algorithms is PACE (“Processor Acceleration to Conserve Energy”) [8]. This method achieves improvements that can be applied to any scaling algorithm.

DVS has already been implemented in several processor designs. Some of the most important processors using DVS are the experimental processor “*lpARM*”, Intel’s “*StrongARM-1100*” and “*Xscale*” CPU as well as the Transmeta “*Crusoe TM5400*” [7]. However, no DSP with dynamic voltage scaling has been described in the literature so far.

## VI. REFERENCES

- [1] T. Burd, *Energy Efficient Processor Design*, Ph.D. Thesis, University of California, Berkeley, 2001
- [2] P. Fugger, *Low-Power Design for Systems on Chip (SoC)*, In Proc. of Telecommunication and Mobile Computing Workshop on Wearable Computing, Oct. 2001, Graz, Austria.
- [3] Infineon Technologies, *CARMEL DSP Webpage*, <http://www.carmeldsp.com>, 2001
- [4] Infineon Technologies, *CARMEL DSP Application Library*, <http://www.carmeldsp.com/pdf/ALibraryV11.pdf>, 2001
- [5] T. Ishihara, H. Yasuura, *Voltage Scheduling Problem for Dynamically Variable Voltage Processors*, Graduate School of Information Science and Electrical Engineering, Kyushu University, 1998
- [6] P. Lapsley et. al., *DSP Processor Fundamentals- Architectures and Features*, Berkeley Design Technology Inc. 1996
- [7] A. Maier, *An Overview of Low-Power Design Methods for Digital Signal Processors*, Technical Report, Institut für Technische Informatik, Technische Universität Graz, 2001
- [8] J. Lorch, A. Smith, *PACE: A New Approach to Dynamic Voltage Scaling*, EECS Computer Science Division, University of California, Berkeley, 2001
- [9] W. Nebel, *Low-Power Design in Deep Submicron Electronics*, Kluwer Academic Publishers, 1996
- [10] T. Pering, T. Burd, R. Brodersen, *Dynamic Voltage Scaling and the Design of a Low-Power Microprocessor System*, Tech. Rep. University of California at Berkeley, 1998.
- [11] I. Ruge et. al., *Low-Power Research Group Web Page*, <http://www.lis.ei.tum.de/research/me/topics/lp.html>, Lehrstuhl für Integrierte Schaltungen, Technische Universität München, 2001
- [12] Ben Shneiderman, *Designing the User Interface: Strategies*

*for Effective Human-Computer Interaction*, Addison-Wesley,  
1998