Energy Minimization of Portable Multimedia Systems through Rate Selection and Dynamic Voltage Scaling

by

Lama Hewage V. P. Chandrasena

BSc(Eng) (Hons)

A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

at the

School of Electrical and Electronic Engineering

The University of Adelaide

Australia

June, 2004

Contents

List of Figures

List of Tables

Abstract

Dynamic supply voltage and frequency scaling involves variation of supply voltage and clock frequency at run-time to minimize the total energy consumption of electronic systems. This method is very effective in reducing energy consumption because of the quadratic relationship between energy and supply voltage. However, use of dynamic voltage and clock scaling requires careful design of systems to function across a range of supply voltages (and frequencies), and additional circuitry such as DC-DC converters.

Dynamic voltage scaling can provide energy savings and extend battery life of portable devices. This thesis focuses on portable multimedia applications operating in fixed throughput mode. These applications have a constant processing requirement, and less than maximum workload levels are characterized by idling, and consequently idle losses. By using dynamic voltage scaling, the processing speed of data samples can be altered at run-time to eliminate idle losses. For optimum energy savings, dynamic voltage scaling requires an infinite number of voltage levels. However, supporting such continuous voltage levels involves introduction of complexities such as closed loop feedback into the DC-DC converter. Moreover, output capacitance of the DC-DC converter must be significantly reduced to achieve fast voltage transitions. This however results in increased output voltage ripple, reduced converter efficiency at low voltages, and decreased system stability. An alternative approach is to use a small number of discrete voltage levels or voltage quantizations and provide open-loop voltage switching. This approach is called the voltage quantization model. Since this approach involves the use of a small number of voltage levels, most workloads can no longer be translated to unique supply voltage levels for voltage scaling, and this leads to selection of higher than ideal voltage quantizations for voltage scaling. This causes idle periods and consequently increased idle losses.

Prior research shows two approaches for reducing idle losses in the voltage quantization model. They are clock gating and voltage dithering. The clock gating technique turns off the clock to minimize energy loss, and this requires special circuitry and special hardware design. Moreover, this technique is only useful when used at a low frequency due to the overheads associated with clock gating. Since fixed throughput mode involves continuous data processing, idle loss reduction through clock gating is ineffective for this class of applications. Voltage dithering on the other hand eliminates idle loss by processing a data sample at two voltage quantizations. However, this increases the number of voltage transitions and consequently increases the transition energy cost and switching noise. Moreover, voltage dithering becomes infeasible when the sample period of the computation becomes comparable to voltage transition time.

In this thesis we propose an alternative algorithmic approach to reducing idle losses in the voltage quantization model. The proposed rate selection approach uses the novel concept of transforming the workload distribution of data sequences to eliminate idle losses. The thesis also presents a number of enhancements to the rate selection approach that improve the energy efficiency and minimize the computational overhead. Our experimental results indicate that the rate selection approach is more energy efficient compared to the best existing approaches, while also significantly reducing the total number of voltage transitions.

Statement of Originality

I hereby declare that this work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying.

Lama H. V. P. Chandrasena

June, 2004

Acknowledgments

I am very grateful to a number of people who were instrumental in successful completion of the research work presented in this thesis.

I would like to thank my supervisor Mr. Michael Liebelt for all his support and guidance. Mike's help started even before I arrived in Adelaide, and has continued unabated ever since. I thank him very much for taking time from his busy schedule to read my writings and provide valuable comments. Thank you also for providing funding to attend a number of overseas conferences.

I wish to thank my fellow postgraduate students Alex Lin, Nasser Asgari, and Nariman Habili for their friendship and constant encouragement. I am also very indebted to Dr. Said Al-Sarawi for always having time for my questions, particularly regarding the DC-DC converter hardware issues.

I am also very grateful to the general staff members in the department. In particular, I appreciate all the computer support given by David Bowler, Nick Kerr, Stephen Guest, and late Norman Blockley. I also want to thank the administrative staff members in the departmental office, Colleen, Yadi, Ivana, and Rose-Marie for their kind support.

I wish to thank my family members for their constant encouragement and support. This thesis is dedicated to my parents.

Finally, I would like to acknowledge the University of Adelaide Scholarship and support of the Department of Electrical and Electronic Engineering, without which this research wouldn't have been possible.

Publications

1. **Lama H. Chandrasena**, Priyadarshana Chandrasena and Michael J. Liebelt, "An Energy Efficient Rate Selection Algorithm for Voltage Quantized Dynamic Voltage Scaling", 14th International Symposium on Systems Synthesis, Montreal, Canada, September 2001.

2. **Lama H. Chandrasena** and Michael J. Liebelt, "A Rate Selection Algorithm for Quantized Undithered Dynamic Supply Voltage Scaling", International Symposium on Low Power Electronics and Design *(ISLPED)*, Rapallo, Italy, July 2000.

3. **Lama H. Chandrasena** and Michael J. Liebelt, "A Comprehensive Analysis of Energy Savings in Dynamic Supply Voltage Scaling Systems using Data Dependent Voltage Level Selection", IEEE International Symposium on Multimedia and Expo (II), New York, USA, July 2000.

4. **Lama H. Chandrasena** and Michael J. Liebelt, "Energy Minimization in Dynamic Supply Voltage Scaling Systems using Data Dependent Voltage Level Selection", IEEE International Symposium on Circuits and Systems, Geneva, Switzerland, May 2000.

Symbols

a	Circuit Activity
α	Device and Circuit Layout Parameters
β	Transistor Gain
C	Capacitance
C_L	Total Node Capacitance
C_{ox}	Oxide Capacitance
D	Duty Cycle
η_{max}	Maximum Conversion Efficiency
E	Energy
E_g	Gate Energy
E_{pr}	Data Processing Energy
E_{tr}	Transition Energy
η	DC-DC Converter Efficiency
f	Clock Frequency
f_{in}	Input Clock Frequency
f_{ext}	External Clock Frequency
Gnd	Ground
I	Current
I_C	Transient Current
I_{short}	Short-Circuit Current
I_{mean}	Average Current
k	History Weight
L	Inductance, Transistor Length

μ	Carrier Mobility
n	No. of Non-zero Coefficients, Reciprocal of Voltage Quantizations
N	Delay Ratio
p	Proportion of Sample Period Spent at a Voltage Quantization
P_{avg}	Average Power
P_k	Probability at rate k
$P_{short-circuit}$	Short-Circuit Power
P_{switch}	Switching Power
$P_{leakage}$	Leakage Power
P_w	Power Loss Due to Conduction
r	Processing Rate
R	Resistance
r_{qn}	Nth Rate Quantization
τ	Rise Time of the Input Signal
t	Fraction of the Sample Time Processed at a Voltage
T_{clk}	Period of the Clock Signal
T_d	Time Delay
T_s	Sample Time
V_{dd},V_{DD}	Supply Voltage
V_{diode}	Diode Voltage
V_{in}	Input Voltage
V_{max}	Maximum Scalable Voltage
V_{min}	Minimum Scalable Voltage
V_{out}	Output Voltage
V_{qn}	Nth Voltage Quantization
V_S	Corner Voltage
V_t	Threshold Voltage
V_{tn}	Threshold Voltage of NMOS Transistor
V_{tp}	Threshold Voltage of PMOS Transistor
w	Workload
W	Transistor Width

$wMax$	Maximum Workload
w_t	Total Workload

Abbreviations

AMD	Advanced Micro Devices Corporation
CMOS	Complementary Metal Oxide Semiconductor
DCT	Discrete Cosine Transform
DC	Direct Current
DCC	Digital Compact Cassette
DC-DC	Direct Current to Direct Current Voltage Converter
DCVSL	Differential Cascode Voltage Switch Logic
DEC	Digital Equipment Corporation (now Compaq)
DSP	Digital Signal Processor
DVD	Digital Versatile Disk
DTFT	Discrete Time Fourier Transform
DVS	Dynamic Voltage (and Frequency) Scaling
EDCL	Enabled/disabled CMOS Differential Logic
FDCT	Forward Discrete Cosine Transform
FET	Field Effect Transistor
FIFO	First in First Out
FMIDCT	Forward Mapped Inverse Discrete Cosine Transform
FT	Fourier Transform
HDTV	High Definition Television
IDCT	Inverse Discrete Cosine Transform
JPEG	Joint Picture Experts Group
LDD	Lightly Doped Drain
Li-Ion	Lithium-Ion

LpARM	Low Power Advanced RISC Machines Processor
MP3	Motion Picture Experts Group Layer 3
MPEG	Motion Picture Experts Group
NiCd	Nickel-Cadmium
NiMH	Nickel-Metal Hydride
QCIF	Quarter common intermediate format
PDA	Portable Digital Assistant
PFM	Pulse Frequency Modulation
PID	Proportional, Integral, Derivative Controller
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
RSA	Rate Selection Approach
RISC	Reduced Instruction Set Computer
SSDL	Sample-set Differential Logic
VCO	Voltage Controlled Oscillator
VQ	Vector Quantization, Voltage Quantization

Chapter 1

Introduction

This chapter introduces portable multimedia systems and provides a brief discussion of the importance of minimizing energy consumption. The chapter also provides an overview of the research undertaken and an outline of the thesis.

1.1 Energy Minimization in Portable Systems

Portable systems can be defined as a class of electronic products that can be carried about without being constrained by location. The key enabling requirements of portability are smaller size (and weight) and portable power supplies typically in the form of batteries. A decade ago the portable systems such as calculators included very simple functionality, whereas today's systems such as personal digital assistants (PDAs) are very complex and include most of the functionality of desktop computers. Due to the growing demand for very compute-intensive applications such as audio and video, high performance processors are increasingly being used in portable systems. Since higher throughput inevitably requires increased clock frequencies, the undesirable consequence is increased power consumption. As the amount of energy stored in batteries is limited, the use of power hungry processors in portable systems results in reduced system operation time. In order to prolong the operation time of portable battery-operated systems, two possible avenues are available.

The first approach for enhancing the operation time of portable systems is through increased battery storage capacity. Though processor clock speeds have approximately doubled every one and half years as predicted by Moore's law [Moo] and the power consump-

tion of the processors have correspondingly increased from under 1 Watt to over 50 Watts in the last two decades, the improvements in the battery technology have been less impressive. More specifically, the improvement in the storage capacity of batteries has only been less than four fold in the last three decades [Eag92]. Currently, the most common battery technologies are Nickel-Metal Hydride (NiMH) (and some Nickel Cadmium (NiCd)), and Lithium (Li-ion), with increasing market penetration of lithium polymer battery technologies. Even though new battery technologies provide much hope, they are bound by the fundamental tradeoffs of batteries, involving storage capacity, re-charge time and re-charge cycles, weight, and cost. For example, the NiMH batteries have 30-40% more capacity compared to NiCd batteries, but they cost 50% more and have longer re-charge times than NiCd batteries with the same specifications [Rec]. Similarly Li-ion batteries weigh about half of an equivalent NiCd battery and have higher number of charge/discharge cycles, but the cost is higher. Though currently available Li-polymer batteries are becoming more popular and hold much promise for future advancement, the bottom line is that only incremental advancements can be expected from the battery technology, and some even speculate that the storage limits set by chemistry are fast approaching [Rec].

The second and more effective way to extend operation time of portable systems is by minimizing the overall energy consumption of the system. This involves energy efficient system design or static approaches and also dynamic approaches of minimizing energy consumption at run-time. The static approaches involve minimizing switched capacitance and operational supply voltage, and can be done at algorithmic, architectural, logic, and circuit design levels. The dynamic approaches involve switching to low-power modes such as SLEEP mode provided by the processors, dynamic voltage and clock scaling, and clock gating. Though static optimizations are important for system energy minimization, dynamic approaches are more effective because they use the system's operational patterns to exploit opportunities for energy minimization.

Unlike computer workstations that require operation at highest throughput level at a majority of time, portable systems often have the complete opposite operational patterns that dominate idling and only sporadic operation at the highest throughput. Due to this variation in processing patterns, dynamic power reduction techniques are extremely important to portable systems.

This thesis focuses on the dynamic voltage and frequency scaling (DVS) technique for energy minimization of portable systems. For the purpose of this research the type of applications considered are digital multimedia processing computations, in particular video and audio decoding used in portable multimedia players.

The next section provides an overview of the research work undertaken, and presented in this thesis.

1.2 Research Overview

Energy consumption in CMOS digital systems is proportional to the square of the supply voltage [CSB92]. Consequently, any reduction of supply voltage provides a quadratic energy saving. The dynamic voltage and frequency scaling technique involves scaling the supply voltage (and clock frequency) at run-time to reduce energy consumption. However, scaling down supply voltage brings about an increase in circuit delays and this results in slowing down of the circuits. For systems such as high performance computer workstations that require the processor throughput to be maximum at all times, the reduction of throughput due to supply voltage scaling is not desirable.

Portable systems on the other hand operate in either burst mode or fixed throughput mode of operation. The former is typical to most portable systems that wait in idle mode for majority of the time and process at the highest throughput level for a small fraction of the time. Application examples for such operation patterns include notebook computers, hand-held organizers, and mobile phones. If idle time in burst mode systems can be operated at a scaled supply voltage, energy consumption can be significantly reduced without sacrificing any throughput. Fixed throughput mode of operation is common in multimedia based portable systems such as audio and video playback systems. Examples of such systems include portable MPEG layer 2 audio (MP3) and digital versatile disk (DVD) players. In these systems, data rates (input, output, or both) determine the maximum processing time for a discrete data sample, and variation of processing time due to changes in processing complexity or workload of data samples result in idling. However, using dynamic voltage scaling to slow down the processing speed, idle time in fixed throughput mode of operation can be reduced, without having any adverse effect on the system throughput. For the purpose

of our research, we use dynamic voltage and clock scaling in the context of fixed throughput mode of operation.

Implementation of the dynamic voltage and frequency scaling (DVS) technique involves the design of a system that functions properly across a range of supply voltage levels. Moreover, voltage and clock scaling is enabled through a DC-DC converter and a specially designed ring oscillator, respectively. In a typical system, a DVS-capable processor determines the necessary workload and commands the DC-DC converter and clock scaling circuitry to switch the supply voltage and clock frequency to required levels. In order to achieve energy savings from dynamic voltage and clock scaling, the voltage conversion of the DC-DC converter must be very efficient and also produce fast voltage transitions for real-time control. In order to determine the optimum scaled voltage for each data sample, the processing complexity or workload of each data sample must be known *a priori*.

The existing research on workload determination demonstrates that there are a number of ways to determine the workload of some multimedia computations, and due to the explosive use of dynamic voltage and frequency scaling in multimedia computations, we anticipate that that future multimedia standards may include the exact workload information as part of the header information during the encoding process such that the workload will be available for the decoding process. Since the intent of our research is *not* to develop a new approach for workload determination, but the use of workload information for minimizing energy consumption, we carefully select our computation such that *a priori* calculation of workload can be achieved through the selection of an alternative algorithm implementation of the computation.

The two key parameters of DC-DC converters that are important for dynamic voltage and frequency scaling are high conversion efficiency and short voltage transition times. To achieve very high DC-DC conversion efficiencies, the switching regulator topology is preferred [SSB94]. In order to scale the supply voltage to any output voltage level, the DC-DC converter must use a feedback control loop. Prior art demonstrates a number of such feedback control systems, and some even involving the sophistication of Proportional, Integral, and Derivative (PID) controllers [WH99]. In order to shorten the transient response time of the DC-DC converter for real-time voltage scaling, the output capacitance of the converter needs to be significantly reduced (by about 20 fold) [Bur01]. However, reducing the output

capacitance increases the supply ripple, decreases the stability of the feedback system, and reduces the DC-DC conversion efficiency at low output voltage levels. Since portable systems that operate in fixed throughput mode idle for a large proportion of the processing time, reduction of low voltage conversion efficiency reduces the energy efficiency of the dynamic voltage scaling technique. Moreover, an increase in supply ripple also increases the processor energy consumption. Therefore, faster voltage transitions achieved through reductions in output capacitance can be detrimental to portable DVS systems operating at fixed throughput mode.

An alternative approach to shorten the voltage transition time in DC-DC converters is to eliminate feedback control and use an open loop approach with a small number of predetermined voltage levels (quantizations) [Gut96]. This approach uses a lookup table of predetermined voltages for supply voltage scaling. Since output capacitance is not reduced in this open loop approach, the conversion efficiency is very high at low output voltage levels (and hence for the full range of supply voltages). Moreover, in order to reduce the overhead of the lookup table in the open loop voltage controller, a small number of voltage levels are preferred. However, use of a small number of predetermined voltage levels means that most workload values of data samples can no longer be directly translated to a unique scaled voltage that eliminates idle loss. This leads to selection of voltage quantizations that are higher than ideal, and this results in the computation finishing ahead of the maximum available processing time and idling part of the sample period. Thus, the main limitation of voltage quantization model is the increase in idle losses.

There are two existing methods for minimizing the idle losses associated with voltage quantization model. The first is clock gating, and the second is voltage dithering [Gut96]. Clock gating is a hardware-based approach that has been extensively used in super-scaler microprocessors to reduce the clock power by turning off the clock to unused functional units, and this technique can be applied to minimize the idle energy loss. However, since the majority of data samples in a fixed throughput mode system will contribute to idle losses, clock gating will have to be performed at a very high frequency comparable to data rates. This is undesirable because clock gating is only effective for low frequency use, due to the associated limitations such as increased switching losses caused by large variations of current. Moreover, clock gated systems require a special design and equally complex test

efforts [Xan99].

Voltage dithering involves processing a data sample at two voltage quantizations within a sample period such that the idle time is eliminated. The key requirement for this technique is that the sample period must be long enough to allow the additional voltage transition (due to finite transition time). Moreover, having only a small number of voltage quantizations implies that a vast majority of samples in a typical data sequence will have to be processed at higher than ideal voltage quantizations and this results in increased idle losses. Moreover, the additional voltage transitions incurred in voltage dithering can significantly increases the total number of voltage transitions. Since output capacitance is high in the voltage quantization model, increasing the total number of voltage transitions due to voltage dithering increases the transition energy loss and can dominate the total energy consumption. Additionally, increased voltage transitions can also contribute to increased noise in the system, and this limits the use of this technique in noise sensitive applications such as portable wireless multimedia systems.

There are also other shortcomings associated with the voltage quantization model using open loop approach. These are: 1) predetermined voltage levels do not scale with process and temperature variations, and 2) higher output capacitance results in higher energy loss per voltage transition. However, by using a hybrid feedback control system that utilizes a lookup table and a feedback controller can provide compensation for process and temperature variations [Gut96]. Since the rate of dynamic voltage scaling must be greater than or equal to the data processing rate [Gut96], transition energy loss can be minimized by performing one transition per data sample.

Our research aims to develop a novel idle loss minimization technique for the voltage quantization model utilizing an open-loop voltage scaling method. To this end we propose an algorithmic approach that uses buffering to transform the workload distribution patterns of data sequences, in eliminating idle loss. Additionally, we aim to minimize the total number of voltage transitions and the associated transition energy loss.

The outcome of this research is the development of an approach called rate selection that minimizes the idle loss, transition energy loss, and total number of voltage transitions in the voltage quantization model. To develop and analyze the performance of this approach, we use the IDCT computation of Motion Pictures Experts Group MPEG-2 video decoding com-

putation as the fixed throughput mode computation. IDCT computation was specially chosen because of the availability of a FMIDCT implementation that enables *a priori* determination of sample workload values based on the number of non-zero coefficients. As for the test data, we use 8 color video sequences in QCIF format. These sequences were specifically selected to represent a wide variety of workload distribution patterns. As for the processor specific parameters, we used the specification of lpARM [Bur01], with a number of modifications assumed to suit our single application environment.

1.3 Thesis Outline

The thesis organization is as follows:

Chapter 2 presents the background information for this research. This chapter includes a brief review of the state of the art in power minimization techniques for systems based on digital CMOS circuits. The techniques presented in this chapter achieve power reductions through minimization of capacitance and supply voltage, and are applicable to all levels of systems design. Finally, this chapter presents a brief introduction to multimedia computations and MPEG coding standard relevant for our research.

Chapter 3 introduces the concept of dynamic voltage scaling and reviews the key requirements, application domain, basic system architecture, and the fundamental tradeoffs involved. Moreover, the chapter presents the state of the art in workload determination techniques relevant to multimedia computing, dynamic voltage and frequency scaling circuitry and energy models, the voltage scaling models, and the effect of sample buffering on total energy consumption.

Chapter 4 discusses the state of the art in idle loss reduction in the voltage quantization model, and presents the research problem and directions for this research. Finally a review of the experimental framework is presented.

Chapter 5 presents the key concepts of the proposed rate selection approach. This chapter also presents a simple rate selection algorithm and provides experimental results on its

performance.

Chapter 6 presents a number of enhancements for improving the rate selection approach. The enhancements discussed include techniques to improve energy efficiency and reduce computational overhead.

Chapter 7 presents the performance analysis of the rate selection approach and comparison of results to existing approaches.

Chapter 8 presents conclusions on our research, the key contributions made, and possible future research directions.

Chapter 2

Background

This chapter presents the background information related to our research, and is organized under two main sections: 1) A discussion of the sources of power consumption in Complementary Metal Oxide Semiconductor (CMOS) technology-based circuits, and a brief review of techniques used in minimizing power consumption. 2) An introduction to multimedia signal processing and coding using the Motion Pictures Experts Group (MPEG) standard.

2.1 State of the Art in Low Power Design

This section reviews the two main approaches for power minimization, namely minimization of effective capacitance and scaling of supply voltage. The techniques presented below are static optimizations that occur at design time and applicable to all stages of design. A more complete review of low power techniques can be found in [CB95a],[RP96],[SRP+95].

2.1.1 Sources of Power Consumption

The vast majority of modern circuits are implemented in CMOS technology and the power consumption of digital circuits designed with CMOS can be attributed to three power components, namely switching power, short-circuit power, and leakage power, as shown by Equation 2.1 [CSB92]:

$$P = P_{switch} + P_{short-circuit} + P_{leakage} \tag{2.1}$$

The following subsections provide a detailed discussion on each of these power components.

Switching Power Component (P_{switch}) The switching power component is caused by the charging of internal nodes of a CMOS circuit from zero volts to the full signal, which is typically the supply voltage, V_{dd}. Figure 2.1 shows a circuit model used for calculating the switching power component.

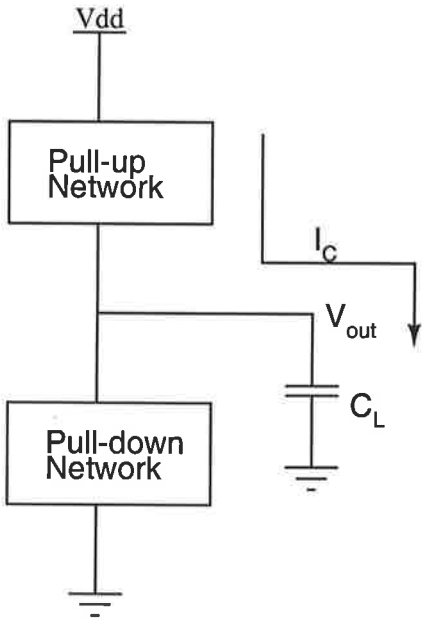


Figure 2.1: Circuit model for calculating the switching power component

During the $0 \rightarrow V_{dd}$ transition, a current I_c flows from the power supply through the network of pull-up transistors into the load capacitor C_L. This transition current, I_c, is given by Equation 2.2:

$$I_c = C_L \frac{dV_{out}}{dt} \tag{2.2}$$

Since power is equal to the product of current and voltage, the energy delivered by the power supply is given by Equation 2.3:

$$E_{0 \rightarrow V_{dd}} = V_{dd} \int_0^{V_{dd}} C_L dV_{out} = C_L V_{dd}^2 \tag{2.3}$$

Since part of this energy is dissipated in the conducting pull-up transistors, the energy stored in the load capacitor, E_C, is calculated from Equation 2.4:

$$E_C = \int_0^{V_{dd}} C_L V_{out} dV_{out} = \frac{1}{2} C_L V_{dd}^2 \tag{2.4}$$

From this equation it is evident that one-half of the energy delivered by the power supply is dissipated in the pull-up network, while the other half is stored in the capacitor.

During the second half of the clock cycle ($V_{dd} \rightarrow 0$ transition), the load capacitor releases the stored energy, and consequently the released energy is dissipated in the pull-down network. Thus, the calculation of total power dissipation of a given circuit can be approximated by calculating the total number of $0 \rightarrow V_{dd}$ transitions in the circuit. Since this calculation is based on a single node in a given design, the total power dissipation of a design can be estimated by use of a coefficient a known as circuit activity. The circuit activity coefficient is defined in synchronous circuits as the fraction of circuit nodes that make the $0 \rightarrow V_{dd}$ transition in each clock cycle. Using activity a, and total capacitance of all nodes C_L, the total switching power component of a design is determined by Equation 2.5 [CSB92]:

$$P_{switch} = a C_L f V_{dd}^2 \tag{2.5}$$

where, a is the circuit activity, C_L is the total node capacitance, f is the clock frequency, and V_{dd} is the supply voltage. The product of a and C_L is also known as the effective capacitance of the circuit.

Short-circuit Power Component ($P_{short-circuit}$) The short-circuit power component is caused by the direct-path short-circuit current which flows from the power supply to ground when both NMOS and PMOS networks are simultaneously active. This current flows for a very short period of time when the input signal of a circuit makes a transition in either direction, from ground Gnd to supply voltage V_{dd} and vice versa. As Figure 2.2 shows, when the input signal makes the transition from Gnd to V_{dd}, the P-transistor is initially on, and as the voltage increases, it eventually reaches the threshold voltage of the N-transistor (V_{tn}) and turns it on. At this point both N and P transistors are on and a direct path exists between the power supply and ground. As the input signal voltage continues to increase, it eventually reaches the threshold voltage of the P-transistor and turns it off. At this point no direct path

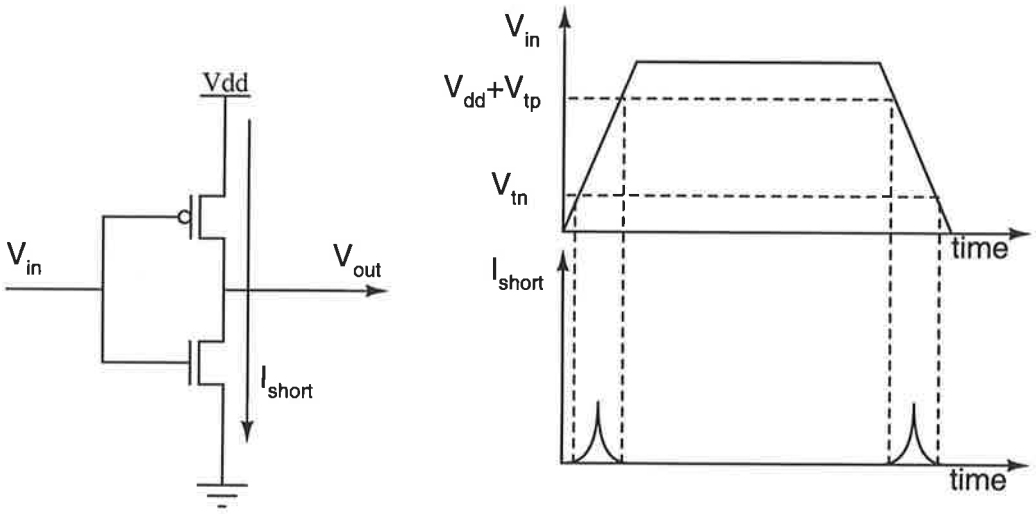


Figure 2.2: Circuit model for calculating the short-circuit power component

exists between the power supply and ground. As sub-plots of the input signal V_{in}, and the short-circuit current I_{short}, in Figure 2.2 show, the direct path between V_{dd} and ground can never exist when $V_{tn} + \mid V_{tp} \mid > V_{dd}$. In this figure, V_{tn} and V_{tp} are the threshold voltages of the N-type and P-type transistors, respectively.

Using the analytic model of [Vee84], the short-circuit component of power can be computed in terms of the mean current (I_{mean}) and the supply voltage (V_{dd}) as follows [Nie97]:

$$P_{short-circuit} = V_{dd} I_{mean} = \frac{\beta}{12}(V_{dd} - 2V_t)^3 \frac{\tau}{T_{clk}} \tag{2.6}$$

where, β is the transistor gain, V_t is the device threshold voltage, τ is the rise time of the input signal, and T_{clk} is the period of the input signal. This result follows from worst-case analysis of the short-circuit current when there is no capacitive load, and assumes that both P and N transistors have the same β and V_t.

From Equation 2.6 it can be deduced that the short-circuit power component is directly proportional to the rise time (τ) and effective transistor gain β. Consequently, shortening the rise time of the input signal when the direct path to ground exists minimizes the short-circuit power component.

Leakage Power Component ($P_{leakage}$) There are two sources of leakage currents contributing to the leakage power component, namely the reverse-bias diode leakage at the drain of the transistor, and the sub-threshold leakage current flowing through the transistor in the *OFF* state. Though in theory this current is zero, the actual currents are in the order of nanoamperes [Nie97].

2.1.2 Power Minimization Techniques

From the three power components discussed above, the switching component contributes to about 90% of the total power, and by properly selecting the transistor sizes, the short-circuit component can be kept to less than 10% [CB95a]. Compared to the switching and short-circuit components, the contribution of leakage component is negligible. Thus, power minimization in CMOS circuits becomes a task of minimizing the dominant switching power component. As a result, power minimization is achieved by optimization of effective capacitance, aC_L, clock frequency f, and supply voltage V_{dd} of the circuit.

This section presents a number of power minimization techniques, involving reduction of switching capacitance and supply voltage, and performed at all levels of system design. For most applications clock frequency is fixed at the system level depending on the required throughput level of the system. However, for a given throughput level, clock frequency can be reduced by introducing parallelism. However, parallelism comes at the cost of increased area, control overhead, and load capacitance [CSB92]. Thus frequency reduction or the associated tradeoffs are not further discussed in this section.

Minimizing Effective Capacitance (aC_L) Power minimization through the reduction of effective capacitance can be accomplished by minimizing the switching activity (a), or simply, the number of switching transitions for a computation. The techniques that accomplish such reductions in switching transitions may involve a simple solution such as powering down a circuit or part of it, to complex clock gating techniques.

A number of algorithm level optimizations can minimize the switching activity. These optimizations include operation minimization and use of encoding schemes.

From the available alternative algorithm implementations for a given computation, selecting the algorithm that reduces the number of operations minimizes the switching activ-

ity. To demonstrate this, consider the Fourier Transform and Discrete Cosine Transform (DCT) computations. Two alternative algorithms available for Fourier Transform are the Discrete Time Fourier Transform (DTFT) and the Fast Fourier Transform (FFT). Considering a transform of N elements, DTFT takes N^2 operations, whereas the FFT takes only $Nlog_2N$ operations. Using $N = 512$, the choice of FFT algorithm over DTFT algorithm produces a 50% reduction in the number of operations [CB95b]. Similarly, blind application of 2-Dimensional forward DCT (FDCT) using an 8×8 block of image samples results in 1024 multiplications and 896 additions per block. However, using a fast algorithm implementation that utilizes the symmetry properties of the cosine basis functions such as Chen's DCT algorithm implementation [CSF77] results in 256 multiplications and 416 additions per block. This is a significant reduction in the number of operations and switching activity. Thus, selection of the algorithm with the least number of operations is crucial for power minimization.

Encoding schemes that alter the data representation also play a vital role in minimizing the switching activity at the algorithm level. Examples of such encoding schemes include the *one-hot coding*, *Gray coding*, and *sign-magnitude representation*. One-hot coding is an encoding scheme used in state assignment of finite state machines and in inter-chip communications [CB95b]. When used in inter-chip communications, this coding scheme has the overhead of an encoder on the sender chip and a decoder on the receiver chip. Moreover, to transmit a n-bit word requires 2^n wires, and involves setting each wire to the corresponding logic level of each bit. Since this approach requires a large number of wires, the application domain of this encoding scheme is very limited. An example of use of this encoding scheme in a filter bank memory design is documented in [Nie97].

Gray code encoding scheme has the important advantage of changing only one bit for consecutive number representations. This property is particularly useful for minimizing the switching activity in applications where consecutive sequences of numbers are processed. Su *et al.* [STD94] use Gray code encoding for memory access operations in a processor environment, where instructions are generally fetched from memory using a sequence of consecutive addresses. The experimental work performed by these authors on a reduced instruction set computer (RISC) processor shows that 30 to 50% reduction in switching activity can be achieved using Gray code, when compared to the binary encoded addressing scheme.

The use of sign-magnitude number representation as opposed to two's complement representation can also reduce switching activity and achieve significant power savings [CB95b]. This is because sign-magnitude representation has a dedicated bit for the sign, and that bit changes only when the sign of the number changes. The main limitations of this number representation scheme are the reduced number of bits available for magnitude representation, and the lack of ease in performing arithmetic operations. Two's complement number representation on the other hand is more common due to its apparent simplicity for performing arithmetic operations such as addition and subtraction. However, changing two's complement values from positive to negative or vice versa will result in several bits switching, increasing the switching activity. Thus, sign-magnitude representation can reduce switching activity in applications that involve a large number of sign changes.

A number of architectural optimizations can also minimize the switching activity. These optimizations include number representation, ordering of operations, and glitching activity reduction.

As discussed above, the number representation at the algorithm level for arithmetic computations has a major impact on the switching activity. If however, the algorithmic level encoding schemes are supported through efficient implementations at the architectural level, significant switching activity reductions can be achieved. For example, if encoding schemes such as sign-magnitude are supported through custom designed circuitry for arithmetic operations, the reductions in switching activity can be translated to power savings. The tradeoffs that need to be considered here are the increased circuit area, and special design and test effort required for custom implementations.

Ordering of operations at the architectural level can also lead to reduced switching activity. To demonstrate this, consider a multiplication operation of a number by a constant coefficient. The optimization performed here is the replacement of the multiplication operation with a shift-add operation. The effectiveness of this optimization on switching activity becomes apparent when the value of the constant is zero. This transformation is particularly useful in signal processing algorithms where multiplications with constant coefficients are very common.

Glitching activity can also have a significant impact on the switching activity. Glitching activity refers to the spurious transitions that occur in circuits implemented using static logic

where multiple transitions can occur during a clock cycle, before settling to the correct logic value. The number of glitching transitions is a function of logic depth, signal skew caused by different arrival times of the input, and signal patterns [CB95a]. The number of worst-case glitching transitions grows as $O(N^2)$, where N is the logic depth [CB95a]. To minimize the glitching transitions, all signal paths need to be balanced and the logic depth reduced. Alternatively, applying one of several asynchronous logic design techniques also achieves the same effect.

The choice of logic structure that implements a function can also have a significant impact on minimizing the switching activity. For example, implementation of the logic function of an adder can be done in many different topologies including ripple carry, carry lookahead, conditional sum, carry skip, and carry select. However, depending on the topology chosen for a given implementation, the number of transitions varies significantly [CJ92].

Circuit level optimizations can also have a significant impact on the overall switching activity. The use of static or dynamic logic, pass gate or conventional circuits, and asynchronous or synchronous circuits determine how the switching activity is affected.

Dynamic logic has been proven to be better suited to low power operation than static logic. This is mainly because dynamic logic eliminates short-circuit loss, has reduced parasitic node capacitances, and reduced switching transitions due to hazards [CB95a]. Static logic on the other hand have no precharge operation and charge sharing. Even though dynamic logic has a clear advantage for low power, the choice in real designs also involve issues such as ease of design and testability, speed, and area.

The physical capacitance of a circuit is proportional to the number of transistors required to implement the function. Thus use of a logic style that reduces the number of transistors reduces the capacitance. Hence using pass gate logic in place of conventional CMOS circuits can reduce the transistor count, and consequently the capacitance.

All logic circuits based on synchronous implementations are continuously being switched during each clock cycle, regardless of whether those circuits are performing any useful operations or not. Since each switching event consumes power, it is necessary to selectively power down any unused circuitry when not in use. Such power down in synchronous circuits can be achieved through clock gating. Clock gating enables and disables the clock signal to a portion of the circuit depending on whether that portion of the circuit is performing

any useful work or not. Clock gating is widely used to reduce clock related losses in high performance microprocessor architectures such as the DEC Alpha [GBJ98], Intel Pentium [TSR+98], and AMD Athlon [Dev01], where approximately 32% of the power is consumed in the global clock circuitry network [GBJ98]. However, this technique has a number of associated disadvantages such as not being able to power up a turned-off block in time for the next clock cycle, and increased risk of glitches. There is also a slight overhead in circuitry associated with clock-gating. Asynchronous circuits on the other hand use no global clock signals to synchronize data transfer between logic blocks. Instead, they use handshaking signals, *request* and *acknowledge*, to communicate between logic blocks. However, handshaking signals increase the circuit area of the asynchronous designs. The asynchronous style of circuit design is extremely effective for power minimization. This is because only the circuit blocks that perform useful work in asynchronous systems consume power, and all other idling blocks consume almost no power. However, selection of logic family can make the design of asynchronous circuits easy or difficult. For example, the use of dual-rail encoding technique which involves the use of two wires per bit to encode both timing information and data values, can be implemented using clocked Differential Cascode Voltage Switch Logic (DCVSL) [JB90],[CP87]. This is a differential precharged logic family similar to domino logic. A simple extension of DCVSL with an *OR* gate provides the completion (or handshake) signals necessary for the asynchronous circuit implementation. Although DCVSL logic family is ideal for implementing self-timed designs, they have been found to consume at least twice the energy per input transition compared to the conventional static logic family [CB95b]. Therefore selection of logic family is crucial for exploiting power savings in asynchronous systems. This is particularly true for designs in which a constant throughput is necessary. A number of alternative logic families such as sample-set differential logic (SSDL) [GH86], enabled/disabled CMOS differential logic (EDCL) [Liu88] and latched domino CMOS logic [PSS86] also provide differential output signals, and can be used in self-timed design.

Optimizations at the physical design level can also result in significant power reductions. Place and route optimizations such as shortening of the wires associated with high switching activity such as the clock signals can result in power savings. Details of investigations on switching activity-based place and route optimizations can be found in [HP93], [CW94].

The layout optimizations to minimize parasitic capacitance in the physical design level can also produce reductions in overall physical capacitance.

Minimizing Voltage (V_{dd}) Apart from the minimization of switched capacitance, power minimization can also be achieved through reduction of supply voltage. Since power consumption is proportional to the square of supply voltage, even a small reduction in supply voltage can produce a quadratic power saving. However, the reduction of supply voltage increases circuit delays, resulting in slower circuits. The relationship between the circuit delays and supply voltage in a CMOS inverter is modeled by Equation 2.7 [CSB92]:

$$T_d = \frac{1}{f} = \frac{C_L \cdot V_{dd}}{\mu C_{ox}(W/L)(V_{dd} - V_t)^2} = \frac{V_{dd}}{\alpha(V_{dd} - V_t)^2} \tag{2.7}$$

where, f is the clock frequency, T_d is the time delay, V_{dd} is the supply voltage, C_L is the total node capacitance in the circuit, μ is the carrier mobility, C_{ox} is the oxide capacitance, V_t is the threshold voltage, and W/L is the width to length ratio of transistors. For convenience, the device and circuit layout parameters are absorbed into the parameter α.

Figure 2.3 shows the normalized time delay and supply voltage relationship of Equation 2.7 based on a threshold voltage value V_t of 0.7V. As this figure illustrates, the delay increases as V_{dd} is decreased, and at higher voltages the change in delay is insignificant compared to voltages approaching the threshold voltage. As supply voltage approaches the threshold voltage, the time delay becomes increasingly large and displays an asymptotic behavior. The reason for this behavior is that Equation 2.7 only applies to transistors in strong inversion, and hence will not be valid in modeling operation at supply voltage values close to the threshold voltage.

The validity of the supply voltage and time delay relationship of Equation 2.7 has been verified on a number of circuits sizes ranging from 56 to 44802 transistors, implementing a variety of functions [CSB92]. The circuits modeled in this work include a microcoded DSP chip, a multiplier, an adder, a ring oscillator, and a clock generator using 2μm technology. The results of [CSB92] show that Equation 2.7 provides a very accurate model even for complex circuits.

This section discusses three techniques that enable supply voltage to be reduced to achieve power savings. The first technique is optimization of transistor sizing. Transistor size opti-

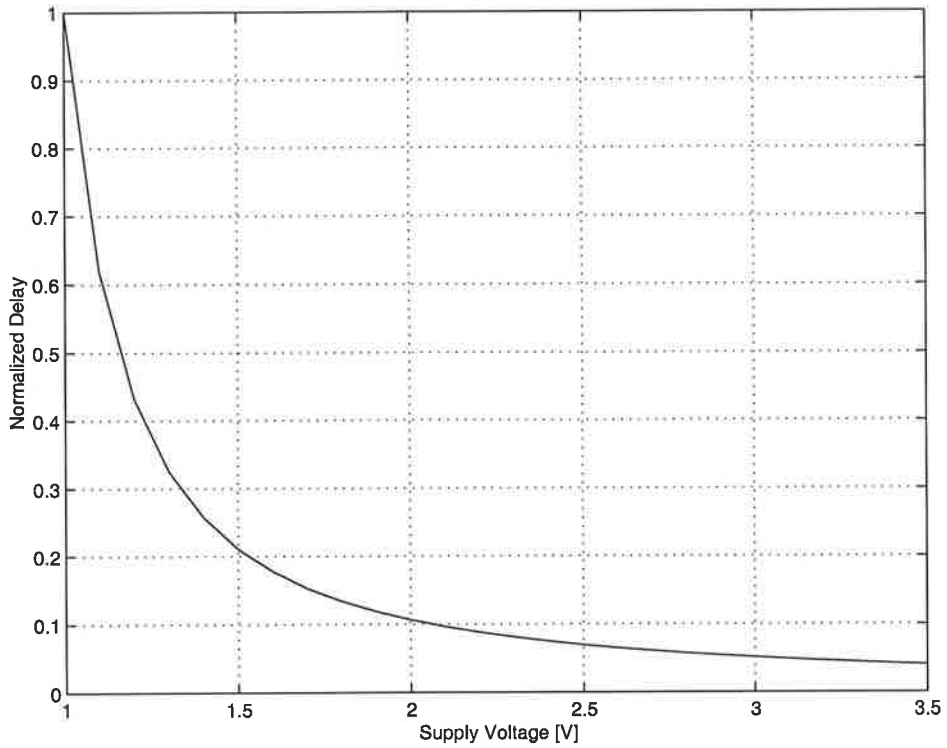


Figure 2.3: Normalized circuit delay vs. supply voltage relationship for digital CMOS circuits

mization is very important for achieving power reductions, regardless of what logic family or topology is used for implementation. For higher power efficiency, all delay paths must be equalized so that performance of a single critical path does not limit the overall performance of the circuit. However, there is the issue of how far the (W/L) ratios can be uniformly raised for all devices, resulting in a uniform decrease in gate delay while producing a corresponding reduction in voltage and power. As [CSB92] shows, the optimization of transistor size for high-speed design does not result in the same transistor size when optimized for low power, so a special effort is necessary for optimizing transistor size for power minimization. The second technique is threshold voltage reduction. As discussed above, scaling down supply voltage while keeping threshold voltage constant slows down circuits. However, scaling down *both* supply voltage and threshold voltage reduces switching power without any loss of speed. For example, it can be shown using Equation 2.7 that a circuit operating at V_{dd}=1.5V with V_t=1V, and V_{dd}=0.9V with V_t=0.5V, have the same performance [CB95b]. However,

since the latter circuit operates at a lower voltage, it consumes less power than the former circuit. The work of [LS93] quantifies the power savings achievable from threshold voltage reduction, and estimates the savings to vary from 3.2 to 8.2 times compared to circuits without threshold voltage scaling. The power savings can also be further improved from the use of low-threshold MOS devices. However, the reduction of threshold voltage is limited by the noise margins and increased sub-threshold currents. The third technique is the optimal operating voltage selection, and this is dependent on the speed requirements and long-term reliability tradeoffs of deep-submicron technologies [DCW+88]. If circuit speed is increased by increasing the electric fields, the devices degrade with time, resulting in changes in threshold voltage, reduced transconductance, and increased sub-threshold currents, resulting in eventual breakdown. The changing of the physical structure of the devices by techniques such as the lightly doped drain (LDD) can minimize this degradation by reducing the number of hot carriers. Using this technique, an optimum operating voltage of 2.5V has been derived for 0.25μm technology [DCW+88]. An optimal operating voltage can also be selected based on operating speed and performance tradeoffs of submicron technologies [KK90]. This idea exploits the independence of circuit delay on supply voltage at higher supply voltages, and establishes the concept of a *critical voltage* as the lower bound for supply voltage. As a result, there is no speed advantage achievable through operating the circuits above the critical voltage. This idea was fundamental to the movement of standard operating voltage from 5V to 3.3V without having a major degradation in circuit speed [Bel91],[Dah91], while achieving a 60% power reduction [Dah91].

Although operating above the *critical voltage* provides no additional speed advantages, operating below this voltage can significantly reduce the power dissipation, but these savings come at the cost of reduced speed. However, architectural enhancements such as hardware duplication or parallelization, and pipelining can compensate for the lost throughput caused by the reduction in supply voltage. There are also other associated tradeoffs such as increased circuit area and control circuitry. Therefore, this duplication process can only be effective up to a certain point, beyond which the power savings achieved from further reductions in supply voltage become less attractive due to increased overhead associated with control circuitry and increased power dissipation of duplicated circuits.

2.1.3 Summary

This section presented the sources of power consumption and described a number of static design techniques for minimizing the power consumption. The techniques presented reduce power consumption through switching activity and operational voltage reduction, and are applicable to all levels of system design. Apart from static techniques, dynamic power minimization techniques identify low throughput periods of operation in systems and goes into low power operation modes. Such low power modes (i.e. SLEEP and HALT modes) are very common in microprocessors and micro-controllers where the system goes into a low power mode at the execution of a single instruction. These processors also support a WAKEUP instructions, to return to the normal operational mode. These low power modes involve turning portions of the system off, and in some cases the clock is also turned off. The shortcoming in such an approach is the latency associated with restoring the operation to full operation mode (including starting the clock). An alternative strategy is scaling the supply voltage in lock-step with frequency to minimize power consumption. Details of this approach forms the basis of this research and is presented in the next chapter.

2.2 Multimedia Computations

In the recent years, processing of image, video, and audio signal data, collectively known as multimedia has become very popular due to the advancements in algorithms and computer architectures. Since the digital representation of multimedia signals results in large amounts of information, efficient encoding or compression algorithms have made multimedia storage and transmission very effective. Similarly, architectural advancements in computing have also made the cost of the compression chips relatively low, and also the general purpose processors capable of handling the throughput necessary for compression. These advances have resulted in rapid growth in multimedia products, particularly portable consumer electronics.

Since multimedia processing involves either compression or decompression computations, or both, standards that specify such computations are vital to the effective use of multimedia based systems. For example if some multimedia data is compressed for transmission, the receiving decoder must be able to identify the compression standard used such

that the compressed multimedia can be decoded. Currently there are a number of multimedia standards available for use and these include Joint Pictures Experts Group (JPEG) for image compression, Motion Picture Experts Group (MPEG), Windows Media from Microsoft, Quicktime from Apple Computer, RealVideo and RealAudio from RealNetworks for image, video and audio compression. Some of these standards such as JPEG and MPEG are open standards, while most of the others are proprietary. For the research work presented in this thesis, open multimedia standards defined by MPEG for video are used because of the availability of MPEG codecs for experimentation, the evolving nature of the standards (currently on MPEG-21), and the growing popularity of MPEG for many portable multimedia applications.

2.2.1 A Review of Multimedia Compression/Decompression

The main intention of compression is to achieve a compact digital representation of the multimedia signals such that transmission requires less bandwidth and storage requires less space. For example, voice signals at 8 ksamples/s (at 8-bit/sample) requires 64kbps uncompressed, whereas the compressed data rate can be between 2 and 4 kbps [BK95]. Another example is high definition television (HDTV) where the raw data rate of 1.33Gbps at 59.94 frames-per-second (fps), 1280×720 frame size, and 8 bits/pixel is reduced to 20Mbps, or a compression of 68 times. Such examples demonstrate the importance of compression, without which bandwidth limitations make most multimedia applications impractical to implement.

All compression techniques take advantage of statistical redundancy in multimedia data. This involves removal of replicated data such that the size of data representation can be reduced to achieve compression. The compression techniques can be categorized into two main groups: lossy and lossless. In lossy compression, the original data and the reconstructed data are not identical. In other words some information loss occurs during compression such that higher compression ratio can be achieved, and hence this process is irreversible. The majority of multimedia data compression schemes use lossy compression techniques because the loss is not noticeable due to the limitations of the human perception system. Lossless compression produces an identical set of reconstructed data from a compressed set, and hence this process is reversible. However, the achievable compression ratios are not as

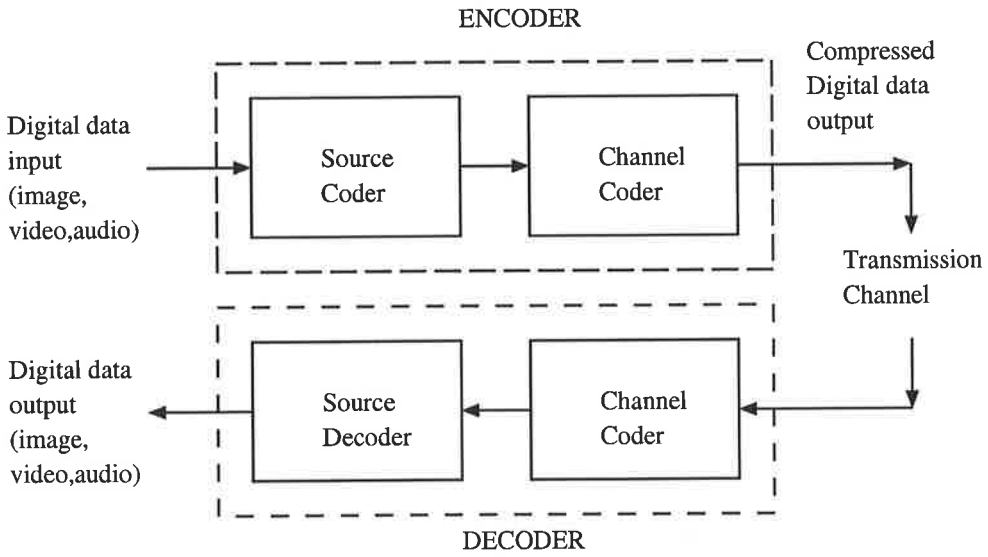


Figure 2.4: Block diagram for generic compression and decompression process

high as the lossy compression techniques.

The block diagram for the typical multimedia compression/decompression process is shown in Figure 2.4 [BK95]. As this figure shows, the source digital data (image, video, and audio) is fed to the encoder for compression. The encoder includes the source coder which compresses the input data, followed by the channel coder which translates the compressed bit stream into a signal that is appropriate for transmission or storage. The decoder receives the compressed bit stream and performs the opposite function of the encoder to retrieve the digital data. If lossy compression is used the digital data output of the decoder will not be identical to the source data input to the encoder. However, if lossless compression is used, input and output data of this compression/decompression process will be identical, provided there is no loss during the transmission. The main tradeoffs of the compression process for minimizing the bitrate are the desired signal level quality, acceptable communication delay of the channel, and implementation complexity in terms of memory, power etc.

2.2.2 MPEG standard

The multimedia standards can be divided into two main groups:

1. The International Telecommunication Union Telecommunication Standardization Sec-

tor (ITU-T, known as CCITT before 1993) which is now part of the United Nations that has produced a number of international standards for real-time digital multimedia communication, including video and data conferencing. Some examples of these standards include the ITU-H-series of standards (H.320 through H.324, and H.310) that cover real-time conversational two-way video and audio, and the ITU-T T.120 series of standards that cover data and graphics conferencing and conference control.

2. The Moving Picture Experts Group (MPEG), an international committee that has produced a sophisticated and commercially important set of standards that are primarily focused on storage, playback, and broadcast applications.

This section provides an overview of MPEG, the standard used in our research work.

In 1988 International Organization for Standardization (ISO) established the MPEG with the goal of developing standards to represent coded moving pictures and audio for digital storage. As a result the MPEG-1 standard (ISO standard 11172) for coding moving pictures and audio for storage at data rates up to 1.5Mbit/s was released in 1991. This lossy compression standard uses limitations of human visual and hearing system by removing details that are not perceptible to achieve higher levels of data compression. The work on the MPEG-2 standard started in 1990, even before MPEG-1 was complete, to provide higher datarates needed for high definition television, and better input-format flexibility and error resilience. The MPEG-2 standard (ISO standard 13818 or ITU-T recommendation H.262) was approved in 1994. The MPEG-4 was born out of the success of MPEG-2 and MPEG-1 and focuses on standards for audio-visual representation such that a scene can be modeled as a set of objects with specific characteristics and behavior. MPEG-4 allows for more flexibility than MPEG-1 and MPEG-2 and supports the growing needs in interactive multimedia applications. MPEG-4 standard (ISO 14496) was finalized in October 1998 and became an international standard in early 1999. Version 2 of the standard was accepted in early 2000. MPEG evolution continues to advance with the changing needs of the multimedia world. The MPEG-7 standard provides a rich set of standards for representing multimedia content for human and machine processing. Using this standard multimedia content can be described such that efficient access (such as search, filtering, and browsing) can be enabled. Currently, the MPEG-21 standard is being created and this standard defines the multimedia distribu-

tion standards and includes features such as intellectual property management and terms and conditions of use [MSS],[MPFL96],[GBL+98].

The generic MPEG video encoding and decoding process is shown in Figure 2.5. The encoding process first preprocesses the incoming video data and then uses motion estimation to predict the current picture from previous picture frames. The motion vectors are included in the final encoded output to enable the decoder to reverse the motion estimation process. Then the predictor for each block is subtracted, and the remaining residual is sent through the discrete cosine transformation (DCT). Finally the DCT coefficients are quantized, and variable-length-coded to produce the output for transmission. The encoder also reconstructs reference frames for future motion estimation and prediction by performing inverse quantization and inverse DCT (IDCT) of coefficients by utilizing the predictor. The decoder starts off with variable length decoding, inverse quantization, and IDCT. Then the predictor is formed from the previously reconstructed pictures and performs a summing operation with the output of the IDCT step to form the reconstructed picture. Finally, the postprocessing block converts the picture for display.

The MPEG audio encoding and decoding process is shown in Figure 2.6. The first step of the encoding process is the sub-band decomposition which filters the audio data into separate frequency bands, which are then scaled and quantized. The second step involves frequency domain analysis that performs additional analysis to select the quantizer step. Finally the data is coded depending on the MPEG audio compliance points called *layers*. For example, coding at layers 1 and 2 uses fixed-rate coding, whereas for layer 3, Huffman coding is used. Finally, the coded samples are formatted with side information for transmission. The decoder first unpacks the data and decodes the side information. Then, inverse quantization is performed to produce reconstructed data values, and finally scaling and sub-band composition is done to produce the audio signal.

For the research work presented in this thesis, the MPEG-2 video coding standard is exclusively used. The selection of this standard for our experimental work was due to its versatility in terms of applications and scope for use in portable systems, and the readily available open source codecs and test data sequences.

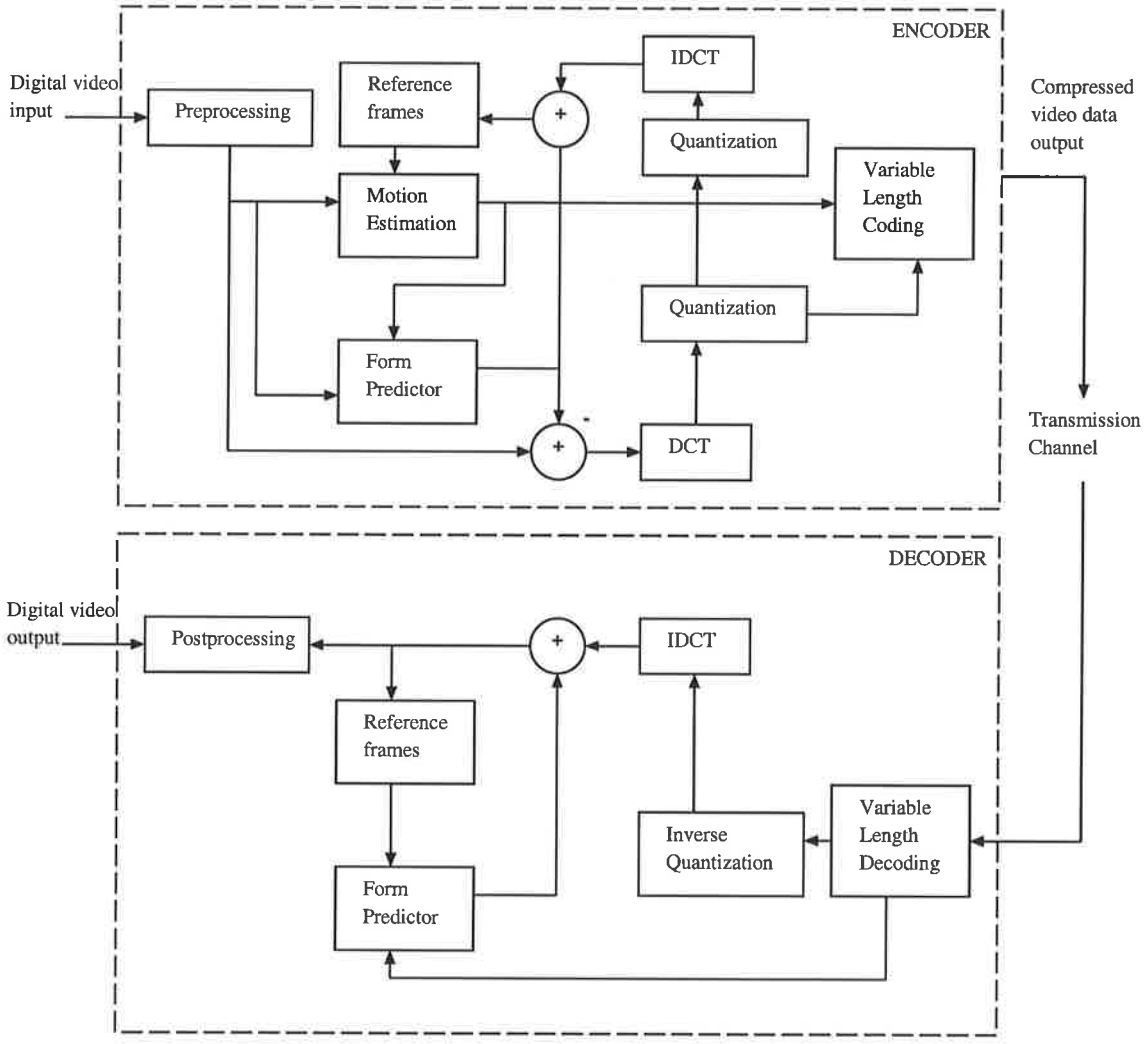


Figure 2.5: Block diagram for MPEG video encoding and decoding process

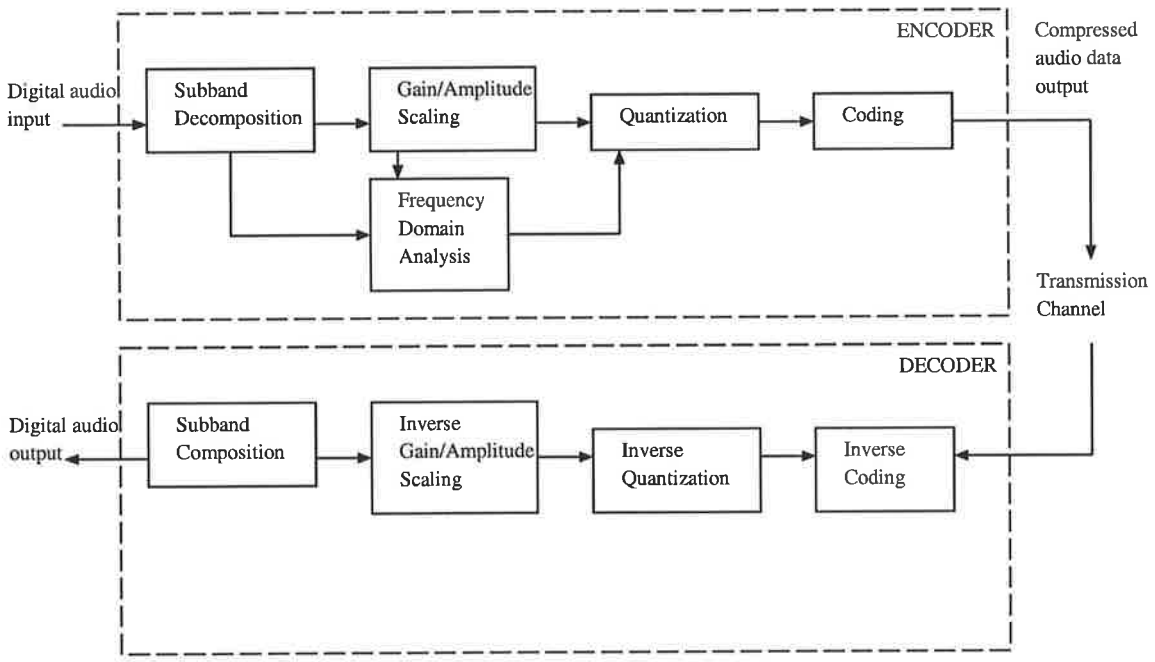


Figure 2.6: Block diagram for MPEG audio encoding and decoding process

2.2.3 MPEG-2 Video Standard

MPEG-2 video is a generic lossy compression standard that is backward compatible with MPEG-1 and uses the encoding steps of discrete cosine transform (DCT) coding, block-based motion compensation, predictive and interpolative interframe coding, and Huffman coding. The standard is intended to support compressed datarates of around 5Mbits/s for NTSC/PAL quality television signals or about twice the bitrate for studio quality video. However, the standard now supports higher data rates (i.e. 80-100Mbits/s for HDTV) and support for different feature sets of the standard are provided using *profiles* and *levels*. A profile defines the subset of the bit stream and a level defines the constraints on the bitstream such as maximum bitrate. The four profiles are Simple, Main, Main+, and Next. The levels are Low, Main, High, and High-1440. Thus, MPEG-2 standard enables the selection of appropriate profile and level for the needs of a particular application. For example, a portable video player using quarter common intermediate format (QCIF) image size (176X144 pixels) and a frame rate of 30 frames per second, can use the Main profile with Low level, or MP@LL for short. As for the video data representation, the standard supports three types of video frames, namely intra frames or *I* frames, predicted pictures or *P* frames, and bidirectionally

predicted frames or *B* frames. Intra frames are coded without reference to any other frame and provide the least compression. The P-frames are predicted using motion compensation of past I-frames and these frames provide points of reference for future motion compensation. P-frames provide better compression levels compared to I-frames. B-frames provide the highest compression and are coded using motion compensation predictions of past and future I or P frames.

Further details on MPEG can be found in [MPE], [MPFL96], [GBL+98].

2.2.4 Summary

This section introduced multimedia computations involving compression and decompression of image, audio and video data, and briefly reviewed the popular MPEG standard. This section also provided a very brief introduction to the MPEG-2 video standard, which is exclusively used in our research work.

2.3 Summary

The aim of this chapter was to provide the background information related to our research. To this end, the first section provided a detailed review of causes of power consumption and some of the power minimization techniques used in digital CMOS circuit based systems. The second section provided a review of multimedia computations such as video and audio compression/decompression, and the MPEG standards, which are relevant to the intended area of application for our research, portable multimedia systems.

Chapter 3

Dynamic Voltage and Frequency Scaling

This chapter introduces the concept of dynamic voltage and frequency scaling (DVS), and demonstrates how this approach utilizes the run-time variation of performance characteristics of a system to achieve energy savings. The chapter also presents the type of applications where DVS can be used, the system architecture, basic requirements, possible energy savings, and fundamental tradeoffs involved. The chapter then presents a review of the state of the art in workload determination, a discussion on voltage scaling models, and the effects of sample buffering and workload averaging.

3.1 Introduction

3.1.1 Background

The primary goal of dynamic energy reduction involves identification of periods when a system is inactive and not doing any useful work, and minimization of the energy consumption during such periods.

The task of identifying idle periods has been approached by Burd *et. al* through characterization of processor utilization models of systems [BB96]. This classification divides all applications into three groups based on the processor throughput desired by the application: 1) maximum throughput mode, 2) fixed throughput mode, and 3) burst throughput mode.

Maximum throughput mode of operation is typical of networked servers where the processor is continuously running and actively being used all the time by a set of users. In this

mode of operation, the objective is to operate at the maximum throughput level all the time to complete all necessary tasks as soon as possible, such that another process can be allocated to the processor. An example of such a system may be a computer at a meteorological station running weather forecast simulations for a number of cities. Since the demand for processor throughput is high in this mode of operation, the processor never idles.

Fixed throughput mode of operation implies that the throughput of a system is fixed, and these systems are typically found in real-time digital signal processing such as MP3 playback devices. In these applications, the throughput is fixed by the data rates, input or output, or both. In these systems, two factors contribute to less than maximum throughput utilization causing idle times and consequently idle losses. The first factor is due to the variation of throughput requirement (workload) of data samples. In other words, if some data samples are less complex to process, they will require less than maximum throughput levels to process them. For such data samples, the remaining processor throughput is unused, resulting in idle losses. The second factor is due to the system implementation designed for worst-case operation. The system design for worst-case operation implies that the system operation is guaranteed across process variations and operating conditions such as temperature. This allows for a safety margin in throughput to be sacrificed, and Nielsen [Nie97] estimates that safety margins of greater than 2.5 times are typical, and this translates to a maximum usable throughput of only 28.6% for a typical system designed for worst-case operation. This implies that a significant throughput level (71.4%) is not utilized, and this contributes to idle losses.

Burst mode of operation involves systems that stay on standby mode for the majority of time and perform useful work only a fraction of the time. This mode of operation is common to portable electronic systems such as mobile phones and personal digital assistants (PDAs), where the system stays in standby mode waiting for an external event to activate the system. Hence, this mode of operation produces the least use of processor throughput, resulting in highest idle times among the three modes of operation.

From the processor usage models above, fixed and burst throughput modes of operation incur significant idle times and can benefit from dynamic energy reduction techniques for energy minimization. Though low power modes such as "sleep" mode are available in most processors, they only support limited functionality such as clock frequency reduc-

tion. The work of [BB96] demonstrates that the energy efficiency of clock frequency reduction depends on the relative proportion of operation time the processor spends idling and at maximum throughput. Moreover, [BB96] shows that if compute energy dominates idle energy, clock frequency reduction *increases* the overall energy consumption. At best, clock frequency scaling is effective in trading off energy and throughput relationship when idle energy consumption dominates compute energy consumption.

3.1.2 Concept

Dynamic voltage and clock scaling involves variation of the supply voltage and clock frequency of a system at runtime. By varying the operational voltage, the processor speed can be varied at runtime, and this is used to minimize idle time and consequently the energy consumption.

This approach is particularly effective for systems where idle loss dominates the total energy consumption. As shown above, such systems operate in burst mode. When such systems are about to undergo a burst of activity, the maximum available throughput is provided by scaling up the supply voltage and clock frequency to the maximum possible levels. Conversely, when the active period is finished and the system enters idle period, the supply voltage and clock frequency are scaled down such that the system enters the lowest throughput level. Figure 3.1 shows in principle how the supply voltage and clock scaling can track processor utilization or throughput level. In this figure, f_{max},and f_{min} refer to maximum and minimum scaled clock frequencies, while V_{max} and V_{min} refer to corresponding maximum and minimum supply voltage levels. By dynamically scaling the supply voltage and clock frequency, the desired highest throughput is provided when activity is detected, while the system remains at the lowest supply voltage for the idle periods consuming only very little energy.

Since systems operating at fixed throughput mode also incur idle losses, DVS can also be applied to fixed throughput mode systems to achieve energy savings. However, voltage and clock scaling in this case does not track the throughput utilization pattern (as in burst throughput mode). Instead, the procedure involves the determination of the workload requirement of a data sample first and then scaling down the supply voltage (and clock) such

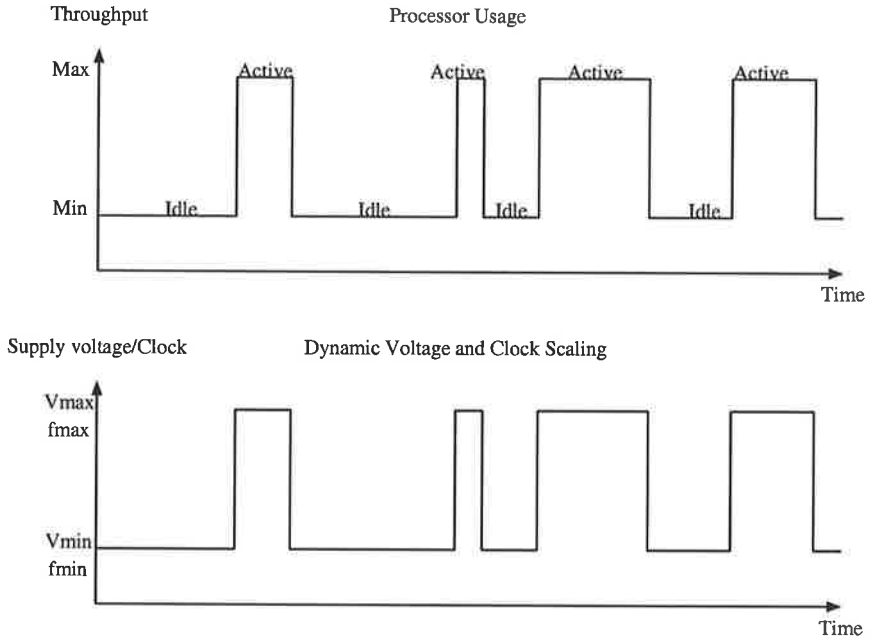


Figure 3.1: Dynamic voltage scaling in burst mode operation

that the full available processing time is used up and no idling occurs. This approach enables data samples with less than maximum workloads to be processed at lower supply voltages and this results in energy savings. This technique is demonstrated in Figures 3.2 and 3.3. Figure 3.2 shows how a data sample with less than maximum workload is processed in a fixed voltage system. Figure 3.3 shows the same task using dynamic voltage scaling. In a fixed voltage system, the processing finishes early and the system idles consuming energy. In the dynamic voltage and frequency scaled system, the system operates at a voltage level between V_{max} and V_{min} such that there is no idling. By operating at a lower supply voltage, the dynamic voltage scaled system uses up less energy compared to fixed voltage system, as explained in the next section. In this figure, T_{sample} is the maximum available processing time for the data sample.

A large body of research exists in using DVS in burst-mode and fixed-throughput mode computations, and the vast majority of work falls under the latter mode of operation. This thesis focuses on DVS for multimedia applications, and hence the discussions in the remainder of the thesis assumes fixed throughput mode of operation.

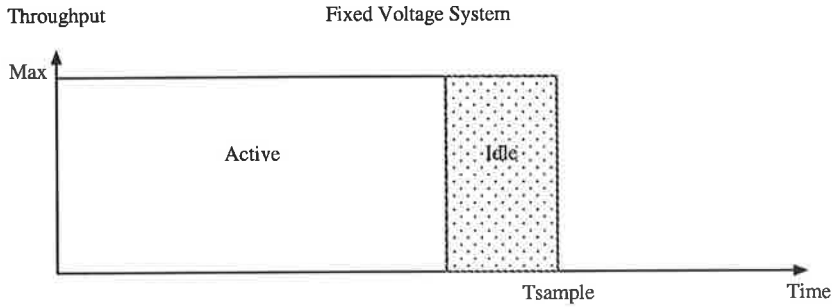


Figure 3.2: Fixed throughput mode of operation for a fixed voltage system

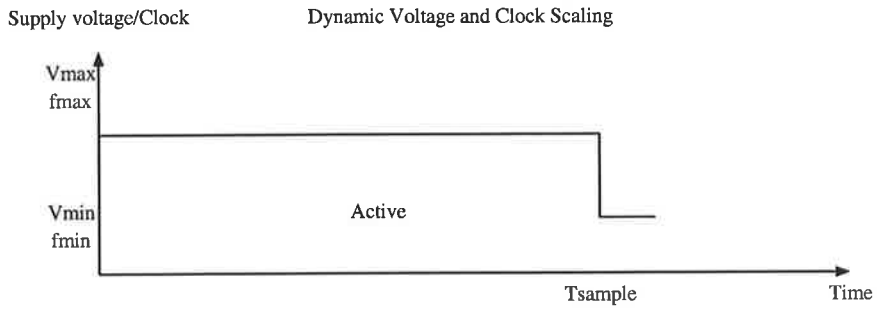


Figure 3.3: Fixed throughput mode of operation for a DVS system

3.1.3 Energy Model

Considering fixed-throughput mode computations and discrete data samples, the energy consumption per data sample E is given by the following Equation [Gut96]:

$$E = nCV_{dd}^2 \tag{3.1}$$

where, n is the number of clock cycles per sample period, C is the average switched capacitance per clock cycle, and V_{dd} is the supply voltage.

Combining Equations 2.7 and 3.1, the relationship between energy per data sample as a function of normalized processing rate r is given by the following Equation [Gut96]:

$$E(r) = E_0 r \left[\frac{V_t}{V_0} + \frac{r}{2} + \sqrt{r\frac{V_t}{V_0} + (\frac{r}{2})^2} \right]^2 \tag{3.2}$$

where, $E_0 = CV_0^2 f T_s$ is a constant with units of energy, C is the average switched capacitance per clock cycle, V_0 equals to $(V_{max} - V_t)^2/V_{max}$ where V_{max} is the maximum

supply voltage, T_s is the maximum sample processing time, r is normalized processing rate ($0 \leq r \leq 1$) or computational workload of the data sample and is equal to f/f_{max}, f_{max} is the maximum frequency, f is the clock frequency corresponding to computational workload of the data sample, V_t is the threshold voltage.

Furthermore, the relationship between rate and supply voltage can also be derived from Equation 2.7 as follows:

$$r = f/f_{max} = \frac{(V_{dd} - V_t)^2/V_{dd}}{(V_{max} - V_t)^2/V_{max}} \tag{3.3}$$

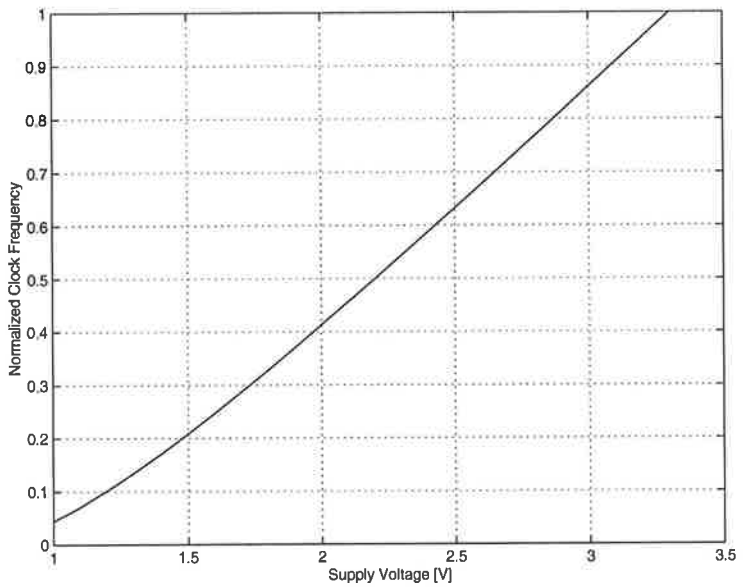
Based on Equations 3.2 and 3.3, the relationships between rate and supply voltage, and energy and supply voltage when supply voltage is scaled in lock-step with clock frequency are shown in Figure 3.4. For this analysis, V_t is 0.7V, V_{max} is 3.3V and V_{min} is 1V.

As the Figure 3.4(a) shows, the rate and supply voltage relationship is nearly linear and hence a full voltage scaling from V_{max} to V_{min} results in a linear scaling of clock frequency (from f_{max} to f_{min}). This implies that a linear reduction in throughput can be brought about by a linear reduction in supply voltage. Similarly, the Figure 3.4(b) demonstrates the corresponding reduction in energy achievable with dynamic voltage and frequency scaling.

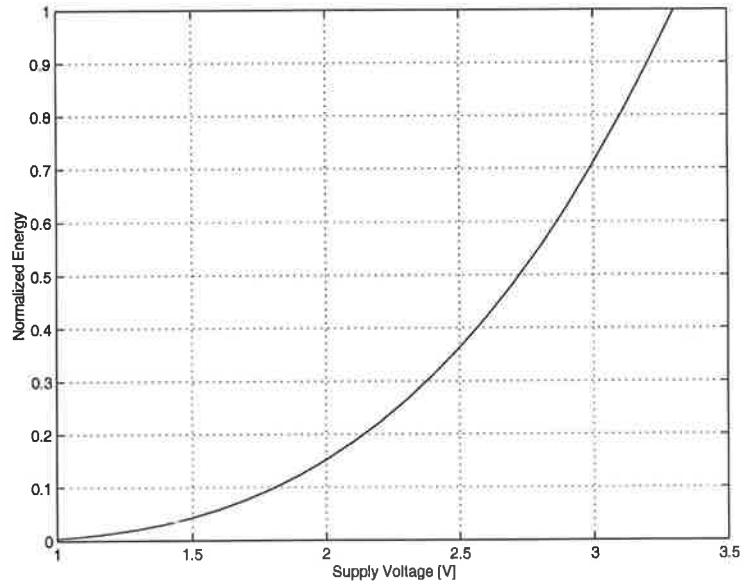
Finally, the important relationship between energy and processing rate as depicted by Equation 3.2 is shown in Figure 3.5. As this figure shows, data samples with low workloads (rate) result in highest energy savings from dynamic voltage scaling, and a full scale of supply voltage results in more than 10 fold reduction in energy compared to fixed voltage operation. This is the desired outcome for fixed throughput computations from dynamic voltage and frequency scaling. For comparison, the flat energy curve for a fixed voltage system is also provided.

3.1.4 Key Requirements

To utilize dynamic voltage and frequency scaling and achieve energy savings for fixed throughput mode systems in the manner discussed above, such systems need to fulfill three key requirements: 1) they should be specially designed to operate over a wide range of voltage levels, 2) the workload per data sample for the computation needs to be known *a priori* for scaling voltage and clock frequency prior to beginning the data processing, and 3) voltage



(a)



(b)

Figure 3.4: First-order relationships. (a) clock frequency vs. supply voltage, and (b) energy consumption vs. supply voltage

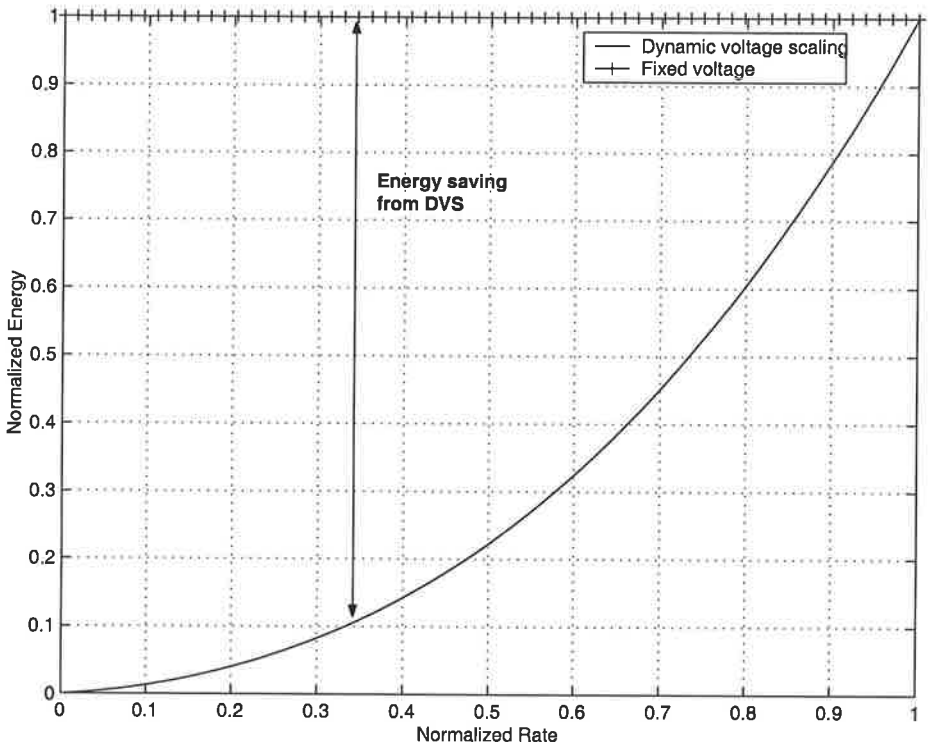


Figure 3.5: Energy vs. rate relationship

and frequency scaling hardware must be capable of switching at a fast rate to provide real-time scaling and this must be achieved at a negligible energy cost and overhead.

Currently, the vast majority of circuits and systems are designed with digital CMOS, and this technology is amenable to dynamic voltage scaling. As improved process technologies bring about reduced operating voltage levels, the range of voltages available for scaling is continuously being reduced. However, device threshold voltages are also being reduced as a result of process improvements, and this helps the continued use of dynamic supply voltage scaling in future processes as well.

Workload determination is crucial for dynamic voltage and frequency scaling in fixed throughput mode computations because it allows the processing speed to be reduced such that idle losses are eliminated. Moreover, workload needs to be determined *a priori* such that voltage and clock can be scaled before processing of data samples begins. However, lack of techniques that accurately determine workload values *a priori* prevents the energy consumption of fixed throughput mode computations being effectively minimized using dy-

namic voltage scaling.

The availability of energy efficient, fast switching hardware that facilitates the on-demand supply voltage and frequency scaling forms the other integral part of dynamic voltage scaling. Typically the voltage scaling is provided by a dynamic DC-DC converter and the frequency scaling is provided by a ring oscillator. The key criteria that govern the effectiveness of the whole supply voltage scaling approach is based on high efficiency DC-DC conversion and ability to provide fast voltage transitions for real-time use.

3.1.5 Basic System Architecture

The basic system architecture of a voltage and frequency scaling system is shown in Figure 3.6. The main components include a processor designed to operate over the full supported range of voltages, a DC-DC converter that takes a request for voltage scaling ($ScaleReq$) from the processor and converts the supply voltage (V_{dd}) to the requested voltage ($V_{dd'}$), and a ring oscillator that produces the clock frequency scaling.

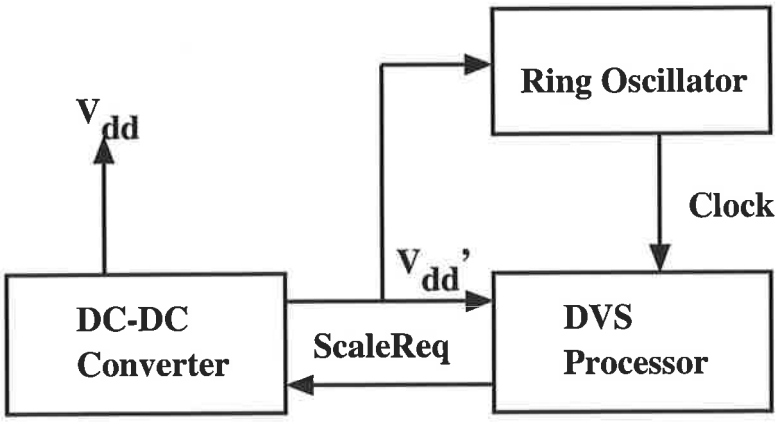


Figure 3.6: System architecture for dynamic voltage and frequency scaling

3.1.6 Fundamental Tradeoffs

The fundamental tradeoffs in dynamic voltage and clock scaling can be categorized into two groups: 1) DC-DC converter, and 2) other system tradeoffs.

The efficiency of the DVS approach is highly dependent on the voltage conversion efficiency of the DC-DC converter. Moreover, fast voltage transitions are essential for real-time operation. To achieve high conversion efficiency, switching voltage converter techniques are the most desirable of all the alternatives. To achieve faster voltage transitions, the output filter capacitance in the converter must be minimized. However, reducing the output capacitance adversely affects converter efficiency at low voltages, increases output supply ripple, and makes the system less stable. Thus, tradeoffs in conversion efficiency, voltage transition time, stability,and output ripple must be made according to the requirements of the fixed throughput mode computation.

The other hardware tradeoffs include design inefficiencies introduced to the system due to the need to support a wide operating voltage range and the reduced throughput caused by slowing down of circuits due to voltage scaling. Depending on the implementation, [Per00] and [Bur01] estimate that 10% to 30% additional energy losses can be attributed to a design that operates over a range of voltages compared to a system designed for a fixed voltage that only accepts a supply voltage variation of about 10%. However, due to the overwhelming availability of idle time in fixed throughput mode computations, the energy savings achievable with DVS far outweigh the increases in static losses. The other hardware tradeoff is reduced throughput at low voltages. However, for the purpose of fixed throughput and burst mode operation, voltage scaling is only done when there is idling. Thus, loss of throughput from DVS has no impact on fixed throughput mode of operation.

3.2 Workload Determination

In order to accurately scale down the supply voltage and clock frequency such that idle time is eliminated in fixed throughput mode computations, the workload of data samples must be accurately determined for each data sample prior to it being processed. Because of this importance, workload determination has been a very active area of research. This section presents a number of techniques that have been used to determine workload for dynamic voltage scaling.

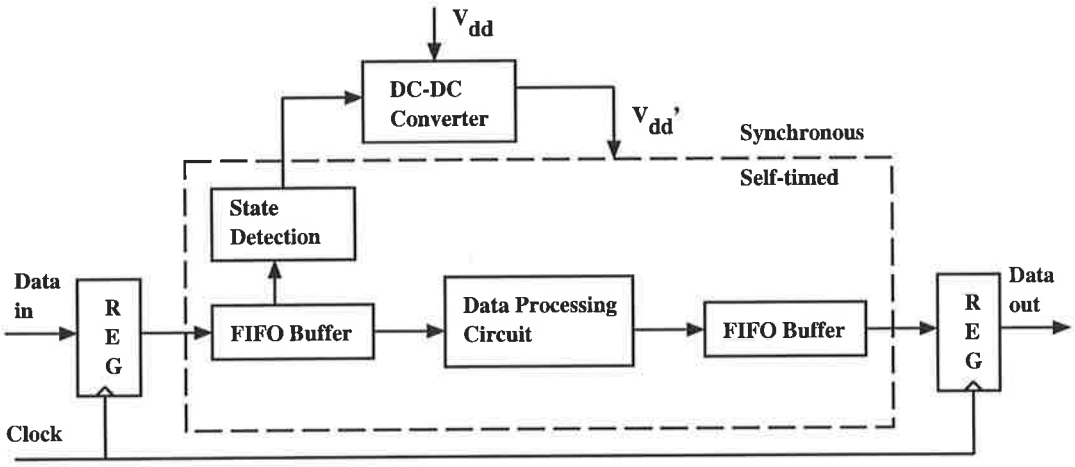


Figure 3.7: Adaptive supply voltage scaling in asynchronous Demonstrator chip [NNSvB94]

3.2.1 Buffer Fullness

The use of dynamic voltage scaling for energy minimization in signal processing applications was first proposed by Nielsen *et. al* [NNSvB94] and the implementation was a chip called Demonstrator. The chip was used in an asynchronous implementation of an error corrector for the digital compact cassette (DCC) [BBK+94]. In the error correction algorithm, the computational complexity is strongly data dependent, and hence the workload of code words without errors is much less compared to that of the code words with errors. Since the majority of code words (over 95%) in a typical data sequence are without any errors, the processor uses very little throughput for the majority of the time. This also implies that the processor idles a large proportion of the time waiting for the next code word to be read and processed. By scaling the supply voltage down when processing code words without errors, the Demonstrator chip achieves 80% power saving compared to the synchronous implementation of the same chip with a fixed supply voltage. Figure 3.7 shows the block diagram for the Demonstrator chip.

The workload of data samples in this system is measured using buffers. As the figure shows, there are two 10-word deep first-in-first-out (FIFO) buffers in this system and as data samples arrive they are buffered in the input FIFO buffer. The state detection is a block that monitors how full the input buffer is. If it is full, the system is operating too slowly, and

the DC-DC converter is instructed to increase the supply voltage to increase the processing speed. Conversely, if the FIFO is running empty, the system is running too fast and the supply voltage is reduced to slow down the processing speed. Though this approach of using buffer fullness to derive the workload is very simple, the shortcoming is the loss of data when the buffer is full. When this occurs, the loss of data samples will continue until the voltage scales up and some samples in the buffer are processed to make room for new data samples.

The same approach for workload calculation using buffer fullness was used in a synchronous digital signal processor (DSP) by Gutnik *et. al* [Gut96]. In this work, the authors use only a single FIFO buffer and clock scaling circuitry to reduce clock frequency along with the supply voltage.

The main limitation of the buffer fullness approach for determining workload is that if the buffer is full when a new data sample is received, it can not be buffered and has to be discarded. If the data loss does not result in degradation of quality of service (QoS) and the DC-DC converter provides rapid voltage transitions, this technique provides a simple way to calculate workload of data samples.

3.2.2 Choice of Algorithm

When alternative algorithms for a given task are available, selection of the algorithm can make workload determination easier or difficult. For example, the inverse discrete cosine transform (IDCT) used in multimedia computations has been implemented as a number of different algorithms [LV86],[CSF77],[FW92],[CL92], [AAN88]. All of these implementations utilize the symmetry properties of cosine basis functions and clever optimizations to produce different totals for the number of operations necessary for processing a data sample. However, one common trait in these implementations is that the number of operations required per data sample is fixed for all data samples. If we consider these computations in terms sample workloads, the workload remains at a normalized value of 1 for all data samples. This implies that no matter how different two data samples are in terms of their processing complexity, they both take the same amount of time to process. Thus, using these algorithm implementations in dynamic voltage scaling provides no opportunity for achieving energy savings.

On the other hand, in the forward mapped implementation of inverse discrete cosine transform (FMIDCT) algorithm [MW92], which is based on the forward mapping technique of [Wol90], the processing need for data samples or workload is proportional to the number of non-zero coefficients in the data sample. Thus choosing this algorithm in place of the previously discussed algorithms will provide an opportunity to determine the workload of a data sample exactly and *a priori*, making this algorithm ideal for dynamic voltage scaling.

Thus, selection of an algorithm based on the ease of *a priori* workload determination as opposed to reduced computational complexity is crucial for achieving energy savings from dynamic voltage and frequency scaling. However, this type of alternative algorithms may not always be available, and the workload prediction techniques discussed in the next section may have to be used.

3.2.3 Prediction Schemes

Prediction schemes involve estimation of future workload either from historical workload values or from attributes of the data sample, or both.

Prediction of future workload from historical workload values requires the maintenance of a history of past workload values and a prediction algorithm. The prediction algorithm evaluates the historical workload values and predicts a future workload value. This approach can be very effective for systems where the workload variation is slow.

Prediction of future workload has largely been used in processor environments due to the difficulty associated with determining the future processing needs. In order to solve this problem, processor utilization is used as workload, and future processor utilization is predicted from past processor utilization patterns. There are two main approaches to determining the future processor utilization: 1) interval based, and 2) thread based [Per00] .

The interval based approach monitors the global processor utilization across fixed time intervals and predicts a future workload for the next time interval. Since this approach takes a global view of the system, it has no knowledge of individual application threads. Consequently, there is no way to satisfy applications that require processor time to satisfy various time constraints such as start time, deadlines etc. The earliest use of the interval based approach was in [WWDS94] where the authors collected workload traces from UNIX worksta-

tion and processed off-line using a number of processor utilization techniques. The prediction algorithms used are OPT (unbounded-delay-perfect-future), FUTURE (bounded-delay-perfect future), and PAST (bounded-delay-limited-past). OPT takes the full trace and extends all the run-times such that no idle time is available. FUTURE as the name implies looks into future window of time trying to eliminate idle time. PAST is similar to FUTURE, in that it looks into a fixed window of time in the past to predict the next workload. Among the three algorithms, PAST is the only practical algorithm [WWDS94]. Govil *et. al* [GCW95] extend the PAST algorithm of [WWDS94], and propose 6 new algorithms: 1) FLAT, 2) LONG_SHORT, 3) AGED_AVERAGES, 4) CYCLE, 5) PATTERN, and 6) PEAK. FLAT algorithm predicts the future workload to be as *flat* as possible, thereby operating around the average speed of the processor. LONG_SHORT considers local changes in workload as well as the average processor speed in predicting the future workload. AGED_AVERAGES is a better version of the LONG_SHORT algorithm and this uses a weighted average of the past to predict the future. CYCLE uses cyclical behavior patterns of the past in predicting a future workload. PATTERN uses CYCLE with priority given to more recent variations in predicting future workload. Finally PEAK uses the PATTERN algorithm with some heuristics in predicting the future workload. Further comparisons of these algorithms and the exponential average based workload prediction used in lpARM processor at Berkeley can be found in [PBB98]. Sinha [Sin01] also presents a number of other prediction schemes for determining the future workload from the historical workloads, and these include moving average, weighted exponential average, least mean square and a pure probabilistic scheme known as expected workload state.

Thread based processor utilization prediction is based on detailed knowledge of applications and their behavior patterns. In other words, thread based prediction uses application specific time constraints such as start time, and deadline parameters in determining processor utilization. In thread based prediction, applications are divided into *frames* and these frames have an amount of *work* to be processed and deadlines which defines the *start time* and *completion deadline*. Thus, for a single threaded system, the future processor speed is determined by the ratio of work/(completion deadline - start time) [Per00]. Two online prediction techniques are discussed in [Per00]: ZERO algorithm is based on [YDS95], and assumes all tasks are sporadic, while RATE algorithm assumes all tasks are periodic. Consequently, for

tasks where the assumption fails (about periodicity), the performance suffers.

Comparison of interval based and thread based prediction shows that the latter provides better performance. However, this advantage comes at the cost of increased computational complexity.

Historical workload based workload prediction has also been used in fixed throughput multimedia computations. Son *et. al* [SYK01] demonstrates workload prediction of MPEG decoding computation where the workload histories for different video frames are maintained for prediction of future workload. Since the workload variation in MPEG decoding is very significant partly due to the different frame types and their associated processing overheads, the authors maintain separate historical workload values for each frame type to improve prediction accuracy. This work uses a weighted averaging technique to predict future workload from the historical workloads.

Use of various attributes or parameters of the data samples to predict the workload has been widely used in fixed throughput mode multimedia computations (MPEG and H.263). [BMP97] utilizes MPEG frame-based parameters such as frame type, length of frame, macroblock type, and packet size and length to predict the workload of MPEG frames. This work shows that various combinations of the parameters can provide workload prediction accuracies within 25% of actual values. Similarly, [PLLS01] demonstrates another ad-hoc approach for determining workload in H.263 video compression, a low-bit-rate video coding standard. In this work, the authors use frame type and frame size information in producing an accurate estimate of the workload of frames. Choi *et. al* [CDCP01] also presents another approach for MPEG where the workload or frame decoding time is determined in two parts: 1) a frame independent portion, and 2) a frame dependent portion. The frame independent portion involves parsing, inverse discrete cosine transform and reconstruction steps, and frame dependent part involves the dithering step. The latter is constant for a video sequence based on frame pixel size. Using this approach, the prediction errors can be reduced to within 20% of actual values for all frame types.

The main limitation of the prediction schemes is the prediction error. When the prediction of the workload for a given sample period deviates from the actual workload, a misprediction error occurs. This error implies that more or less work than the actual was predicted, and hence the future workload predictions need to factor in this error for compensation. This

means that subsequent workload values are altered and since the maximum workload per sample period is bounded, compensation of prediction errors may not be resolved for many sample periods. This impacts heavily on time constraints of the multimedia computations.

3.2.4 Future Directions

As the previous section shows, the determination of exact workload *a priori* can be difficult. Even though prediction techniques are inevitable in burst mode processor operation, exact workload determination can be accomplished in fixed throughput mode computation through techniques such as algorithm selection and buffer fullness. Moreover, for multimedia decoding applications such as MPEG decompression that are the reverse computations of forward computations such as MPEG compression, storing the workload values during the encoding computation in the data stream can make exact workload values available during decoding stage. Even though current multimedia standards do not support such workload information in headers, we envisage that future standards *may* support them due to the growing use of DVS in multimedia systems. Thus, this research will utilize any existing techniques for *a priori* workload determination, and focus mainly on developing a new approach for minimizing the energy consumption and transition count for dynamic voltage and frequency scaling.

3.3 Voltage and Frequency Scaling

To provide run-time variation of supply voltage and clock frequency necessary for DVS, a DC-DC converter and a ring oscillator are required as part of the system. This section presents details of these circuits and the tradeoffs involved in maximizing the efficiency of voltage conversion.

3.3.1 DC-DC Converter

Static Converter The primary function of a DC-DC converter is to convert an input DC voltage to an output DC voltage and provide regulation. Such converters are called *static* DC-DC converters because they are optimized for providing a fixed output voltage. Figure 3.8 shows the block diagram of a static DC-DC converter. As this diagram shows, the target

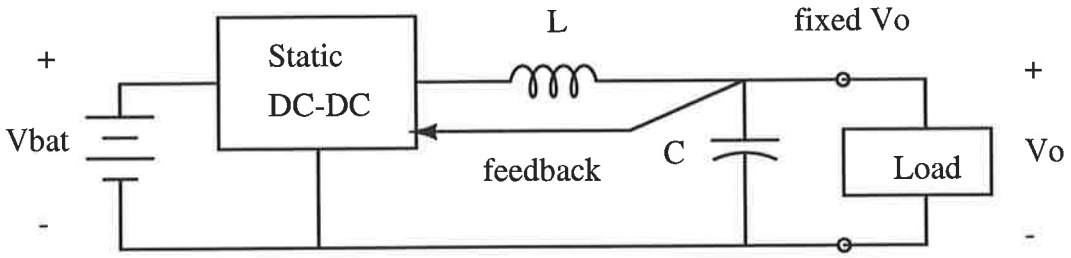


Figure 3.8: Voltage scaling using a static DC-DC converter [Str98]

system or load does not communicate with the DC-DC converter, and the output voltage regulation is provided by comparing the output voltage with a reference voltage. In order to adjust the output voltage, the DC-DC converter uses pulse-width modulation or pulse-frequency modulation [Str98]. The application domain of this type of low output voltage static DC-DC converter is in battery-operated systems where the battery voltage is higher than system operational voltage. For example, a system designed for operation at 3.3V which runs off 3 AA batteries (3*1.5V = 4.5V), can use a static DC-DC converter with input voltage of 4.5V and output voltage of 3.3V. The main advantage of using the static DC-DC converter in this type of application is the reduced energy consumption of the target system due to operation at the lower voltage.

DC-DC converters can be categorized into three groups based on their topologies. These are linear regulators, switched capacitor converters and switching regulators or true DC-DC converters [CB95a].

Figure 3.9 shows the block diagram of a linear regulator. Linear regulators are very attractive for use as DC-DC converters in portable systems because of their small size and simple circuitry requiring few or no reactive components. The main limitation of linear regulators is that the output voltage must *always* be less than the input voltage. Since the conversion efficiency is determined by the ratio of output voltage to input voltage, higher efficiency levels can only be achieved by making the output voltage be slightly less than the input voltage. Consequently, this requirement may limit the practical use of linear regulators. In applications where output voltage is slightly below input voltage and linear regulators are usable, the achievable conversion efficiency is dependent on two parameters: 1) quiescent current and 2) the dropout voltage. The quiescent current controls the power dissipation of

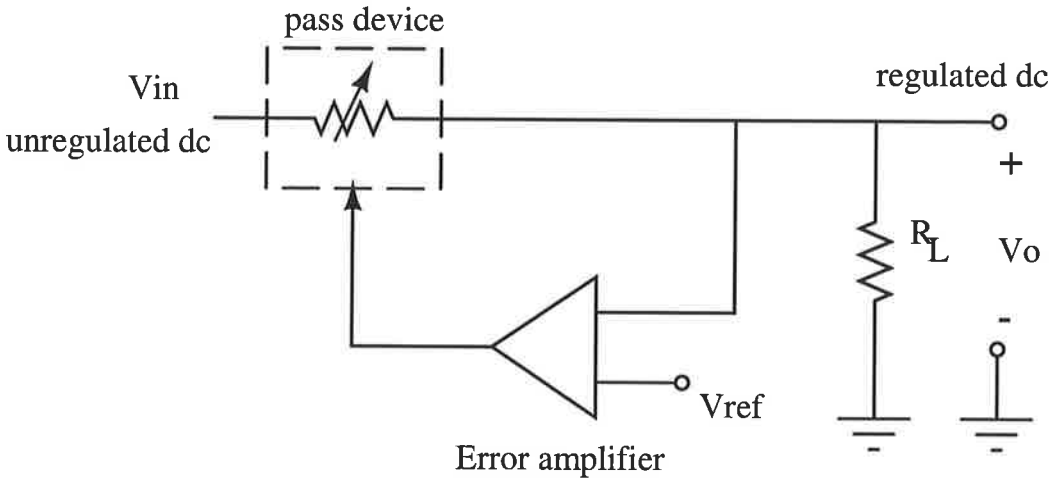


Figure 3.9: Linear (series-pass) regulator [Str98]

the linear regulator when the load current is zero. Hence, circuits with low quiescent current are desired. The dropout voltage is the minimum voltage difference between input and output voltages necessary to maintain regulation. To maintain regulation and high conversion efficiency, a small dropout voltage is desired.

Switched capacitor converters are a group of converters that are designed without magnetic components. They are generally known as charge pumps, and widely used in integrated circuits for step up conversion or polarity inversion. Figure 3.10 shows a switched capacitor converter used as a voltage doubler. As this figure shows, the entire circuit is made out of switches ($S1$,$S2$) and capacitors. By switching the input and output terminals, this circuit could also be used as a step down circuit producing an output that is half the input voltage. The losses in the converter are caused by the parasitic resistances in the capacitances and the switches. Though the switched capacitor converters can efficiently convert voltages, they are unable to regulate the converted voltage any more efficiently than linear regulators. This limits the applicability of switched capacitor converters.

Switching regulators or true DC-DC converters convert an unregulated input voltage to a regulated output voltage. Figure 3.11 shows the basic components of the switching regulator circuitry (based on a buck converter topology) [Str98]. The functionality of this circuit is as follows: The input voltage first gets chopped by the single-throw, double-pole switch producing an intermediary pulse width modulated (PWM) rectangular output voltage wave-

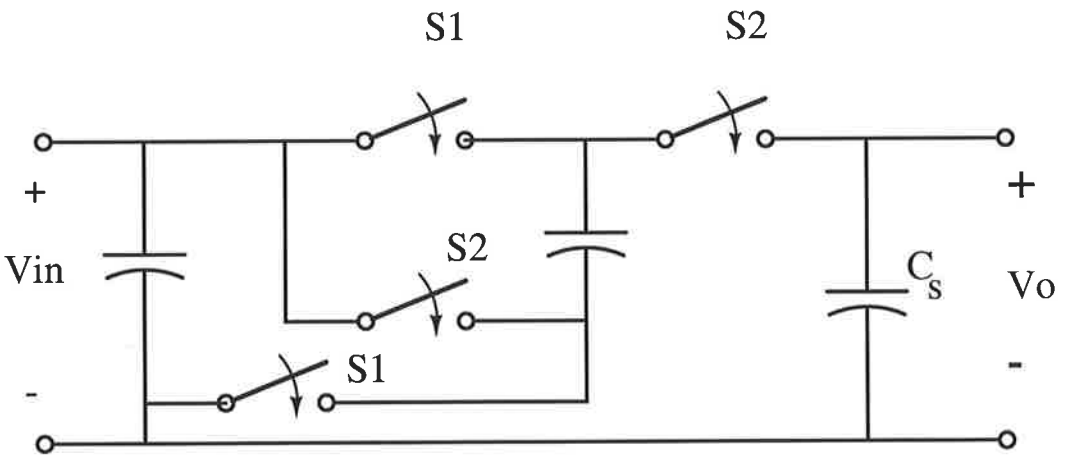


Figure 3.10: Switched capacitor converter - a voltage doubler [Str98]

form that gets filtered by the low pass filter. The resulting output voltage has the desired output voltage level with an attenuated level of AC ripple. The output voltage regulation is maintained by comparing the output voltage with a reference voltage and adjustments to the output voltage is achieved through varying the ON time of the switch. Depending on the need, switching regulator circuitry and filter component selection can be altered to achieve output voltages that are larger or smaller than input voltages, and same or inverted polarity. More importantly, switching regulators provide very high conversion efficiencies approaching 100% with ideal components, with typical and achievable efficiencies of 75% and over 90%, respectively [CB95a].

The switching regulators have a number of alternative topologies: step-down (buck) converter, boost converter, and buck-boost converter. The buck converter is used when the output voltage is less than the input voltage (step-down). The boost converter on the other hand produces an output voltage greater than the input voltage. The buck-boost converter can produce output voltages higher or lower than the input voltage. Since the most common converter used in dynamic voltage scaling is the buck converter ([Gut96], [SMF+97], [BPSB00]), it will be discussed here.

The main components of the buck converter circuit shown in Figure 3.11 include two power transistors ($M1$ and $M2$) and the second-order low pass filter components (L_f and C_f). The two transistors chop the input voltage V_{in}, to produce a rectangular waveform of

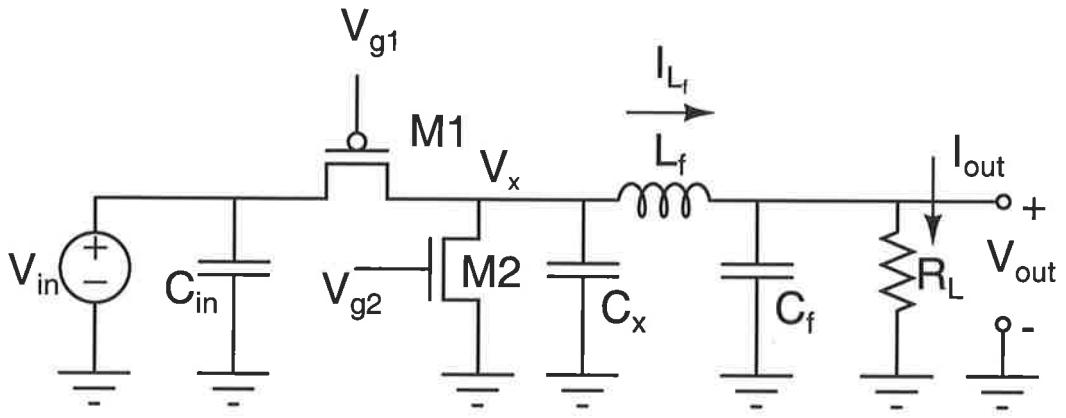


Figure 3.11: Low output voltage Buck converter

duty cycle D, at the inverter node V_x. The low pass filter filters the square-wave signal to produce an output voltage V_{out} with an acceptable ripple. C_x in this figure represents the parasitic capacitance. The ideal relationship between the input voltage and output voltage is given by Equation 3.4 [CB95a]:

$$V_{out} = V_{in}D \tag{3.4}$$

As this equation shows, any output voltage ($V_{out} \leq V_{in}$) can be derived by varying the duty cycle. Additionally, the conversion efficiency of the low output buck converter can be evaluated by considering a more conventional buck converter circuit implemented with one controlled switch (S1) and one uncontrolled switch (a diode), as shown in Figure 3.12. In this circuit, the maximum conversion efficiency is dependent on the forward bias diode voltage (V_{diode}), even if all other losses are considered to be negligible. Considering that the diode only conducts a fraction $(1 - D)$ of the switching period, the maximum conversion efficiency of the buck converter circuit is given by [CB95a]:

$$\eta_{max} = \frac{V_{out}}{V_{out} + (1 - D) \cdot V_{diode}} \tag{3.5}$$

where, η_{max} is the maximum conversion efficiency, V_{out} is the output DC voltage, D is the duty cycle. Figure 3.13 shows the relationship between the maximum theoretical conversion efficiency and output voltage using values of $V_{in} = 3.3V$ and $V_{diode} = 0.7V$ and $0.3V$.

As this figure shows, the buck converter efficiency approaches 100% as the output voltage

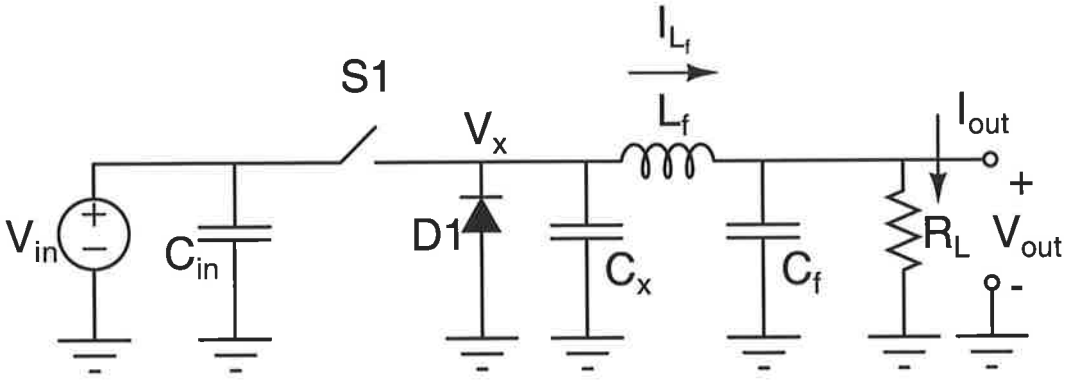


Figure 3.12: Buck converter circuit with a pass device and a diode

approaches the input voltage. Moreover, as the output voltage moves away from the input voltage, the conversion efficiency drops significantly. For example, at output voltage of 1V, the conversion efficiency is about 68% and 84% for $V_{diode} = 0.7V$ and $0.3V$, respectively.

However, there are a number of losses that further reduce the theoretical maximum conversion efficiency of the buck converter. These include: 1) conduction losses caused by the current flow in non-ideal power transistors, filter components and interconnects, 2) gate drive loss caused by switching of the gate of the power transistors, 3) capacitive switching loss in the parasitic capacitor C_{in}, 4) short-circuit loss incurred when both transistors are conducting for a short duration during transitions, and 5) quiescent operating power loss associated with running pulse width modulation and control circuitry of the DC-DC converter. For a more detailed analysis, please refer to [CB95a], [Str98].

Even though the PWM-based DC-DC converters are very efficient at full load, their efficiency at light load levels can be significantly less. This is because the losses associated with the converter are independent of the load current, and as [Str98] shows, a PWM buck converter that is 94% efficient at full load is only 3% efficient at one-thousandth of the full load. Therefore, if the system is operating at light loads most of the time, the losses in the DC-DC converter will dominate any energy savings achievable by operating the target system at a lower voltage.

To maintain high conversion efficiency across a wide range of load, pulse-frequency modulation (PFM) or *burst mode* can be used. In this mode of operation, the converter activates the PWM mode in short bursts separated by idle periods. Thus, at full load, the

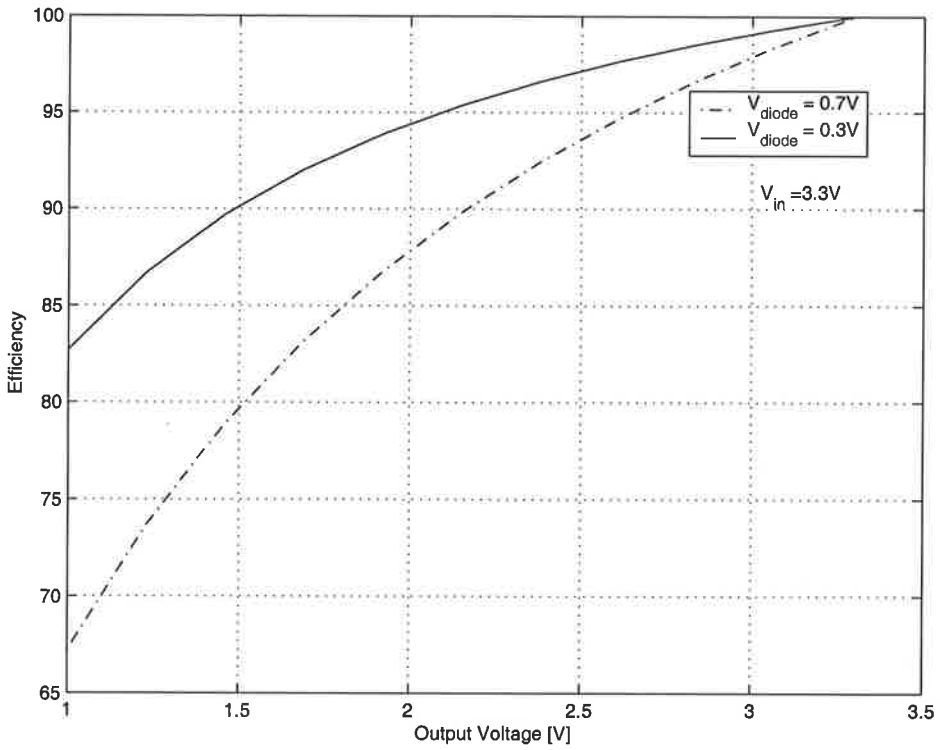


Figure 3.13: DC-DC converter efficiency vs. output voltage

converter produces bursts without any idle periods in between, and as the load is reduced, the bursts become less frequent and idle periods become more frequent. During idle periods, the FETs in the DC-DC converter are turned off, reducing the load independent losses. Even though PFM improves the conversion efficiency, its main limitation is the switching noise which varies as a function of the time between bursts when the load changes. Consequently, the noise issue of the PFM method may make it unsuitable for wireless applications.

Dynamic Converter Unlike static converters that convert a fixed input voltage to a fixed output voltage, *dynamic* converters convert a fixed input voltage to a number of output voltage levels. This type of converter is essential for dynamic voltage scaling because it enables the voltage to be varied at run-time. Figure 3.14 shows the block diagram for the dynamic DC-DC converter (based on a buck converter circuit). As this figure shows, the target system or load communicates with the DC-DC converter in requesting the output voltage to be set to a desired voltage level.

Similar to static converters, conversion efficiency and output voltage ripple are also crucial to dynamic converters. Additionally, dynamic converters have two other performance metrics: 1) transition time, and 2) transition energy [Bur01]. These metrics refer to the time delay overhead and the energy cost of voltage transitions for the DC-DC converter. According to the model developed by Burd [BB00], transition time and energy for a large voltage transition from $V_{dd1} \rightarrow V_{dd2}$ are given by Equations 3.6, 3.7, respectively.

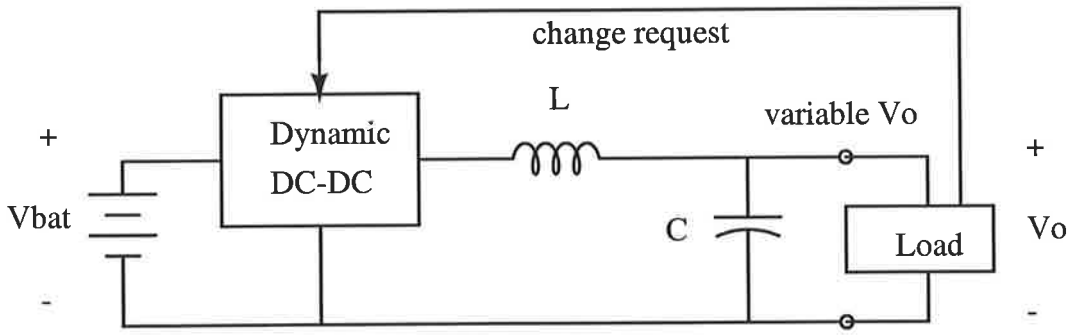


Figure 3.14: Voltage scaling using a dynamic DC-DC converter [Str98]

$$t_{tran} \approx \frac{2 \cdot C}{I_{max}} \cdot \mid V_{dd2} - V_{dd1} \mid \tag{3.6}$$

$$E_{tran} = (1 - \eta) \cdot C \cdot \mid V_{dd1}^2 - V_{dd2}^2 \mid \tag{3.7}$$

where, C is the output filter capacitance of the DC-DC converter, η is the converter efficiency, and I_{max} is the maximum output current of the converter.

To maximize the energy efficiency of the dynamic voltage scaling technique, it is essential to maximize the converter efficiency, and minimize transition time, transition energy and output ripple. To further maximize the conversion efficiency, techniques such as zero voltage switching (ZVS) [SSB94] can be used. This technique enables the FETs in the buck converter to be switched at zero drain potential, virtually eliminating the switching loss. In order to minimize the voltage transition time and transition energy, output filter capacitance must be minimized. However, reducing the capacitance increases supply ripple, decreases the stability of the feedback system, and reduces the low output voltage conversion efficiency. Thus, tradeoffs have to be made in choosing these parameters.

3.3.2 Ring Oscillator

To effectively minimize the energy consumption through dynamic voltage scaling, the clock frequency must also be scaled in lock-step with the voltage scaling. This is accomplished by means of a ring oscillator comprising a string of inverters [Gut96], [Bur01]. The ring oscillator is a replica of the critical path of the system, and this is used to model the delay versus supply voltage relationship of CMOS circuit technology. By comparing the output frequency of the ring oscillator with the desired clock frequency, the supply voltage can be varied using a feedback control system. As far as the power overhead goes, ring oscillators are superior to clock generation by phase-locked loops (PLL), contributing to less than one-tenth of the power consumption of conventional approaches. In lpARM processor, the power overhead of the ring oscillator is about 10 μW, or almost negligible [Bur01].

3.4 Voltage Scaling Models

Section 3.3.1 presented the functionality and tradeoffs of DC-DC converters for use in the context of dynamic voltage scaling. This section presents two voltage scaling models that can be used for dynamic voltage scaling.

3.4.1 Continuous Voltage Levels

The use of continuous voltage levels implies that *any* sample workload can be translated to a unique voltage level for voltage scaling. This also means that the DC-DC converter must be able to produce an *infinite* number of output voltage levels. In order to produce such continuous output voltage levels, a feedback control mechanism is necessary. The feedback control also enables the DC-DC converter to produce accurate output voltage levels across process and temperature variations.

Since continuous voltage levels provides voltage scaling to all possible workloads values, this enables the lowest energy consumption to be achieved from dynamic voltage and frequency scaling. In other words, this model provides the ideal voltage level and minimum energy consumption for a given workload. Figure 3.15 shows this energy relationship for DVS in fixed throughput mode of operation. In this figure, rate is directly proportional to

voltage level.

Even though continuous voltage levels provides the optimum energy saving, there are several disadvantages associated with supporting a large number of voltages. These disadvantages include increased system complexity, area, slow voltage transition times, and power overheads introduced by the feedback control system. A detailed discussion of the tradeoffs involved in the design of a complex DC-DC converter based on a proportional, integral, derivative (PID) controller for feedback control is documented in [WH99].

A more practical implementation of continuous voltage level model assumes a large number of discrete voltage levels as an approximation of the infinite voltage levels. An example of such a system is presented by Kuroda *et. al* [KSM+98]. This system uses 64 discrete voltage levels, with a minimum resolution of 50mV.

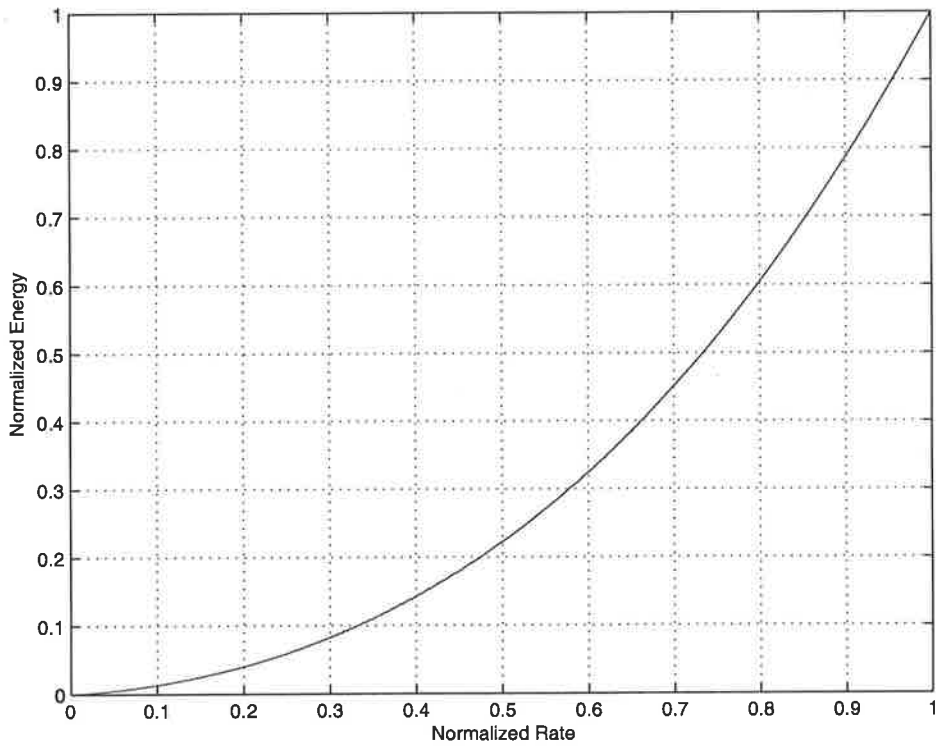


Figure 3.15: Energy and workload relationship for continuous voltage level model

3.4.2 Voltage Quantizations

As opposed to the continuous voltage level model that uses an infinite or large number of discrete voltage levels, the voltage quantization model uses only a small number of discrete voltage levels to cover the full range of voltage levels [Gut96]. By using only a handful of predetermined output voltage levels, the need for the feedback control system in the DC-DC converter can be eliminated. Thus, the voltage quantization model uses a simple table lookup and an open-loop approach for setting output voltage levels.

Practical implementation of this model involves selection of a small number of discrete voltage levels, and the determination of the appropriate pulse-width modulation waveforms for the two FETs of the Buck converter.

A popular method used for determining the voltage quantizations involves two steps [Gut96]: 1) selection of the rate quantizations that are equally spaced in a normalized rate scale of 0 to 1. For example a 4-level voltage quantization would select normalized rate quantizations of 1/4 (0.25), 2/4 (0.5), 3/4 (0.75), 4/4 (1.0). 2) conversion of the rate quantizations to supply voltage levels by substituting each workload quantization value for r and calculating the corresponding voltage quantizations using Equation 3.3. Some examples of systems that use this voltage quantization selection approach are [XCSD96], [LYyTC99], [Gut96].

An alternative approach for selection of voltage quantizations is given in [CL00],[CCL01]. In this work, we proposed a voltage quantization selection scheme that is based on characteristic workload distributions of the data sequences. Though this approach is more energy efficient, it is more complex and requires the workload distribution models of data sequences to be derived.

The main advantage of open-loop DC-DC converter operation is the fast voltage transient response. Gutnik *et. al* [Gut96] report transition times of around 10μs using a hybrid converter that utilizes the open loop and voltage look up approach. The secondary advantages of the open loop approach are low supply voltage ripple and very high DC-DC conversion efficiency for the full range of scaled voltages due to high capacitance in the output filter. Using a typical output capacitance of 100μF, Burd [Bur01] estimates output supply ripple to be less than 1% and conversion efficiency of the DC-DC converter for the full voltage range

(1.2V-3.8V) to be over 95%.

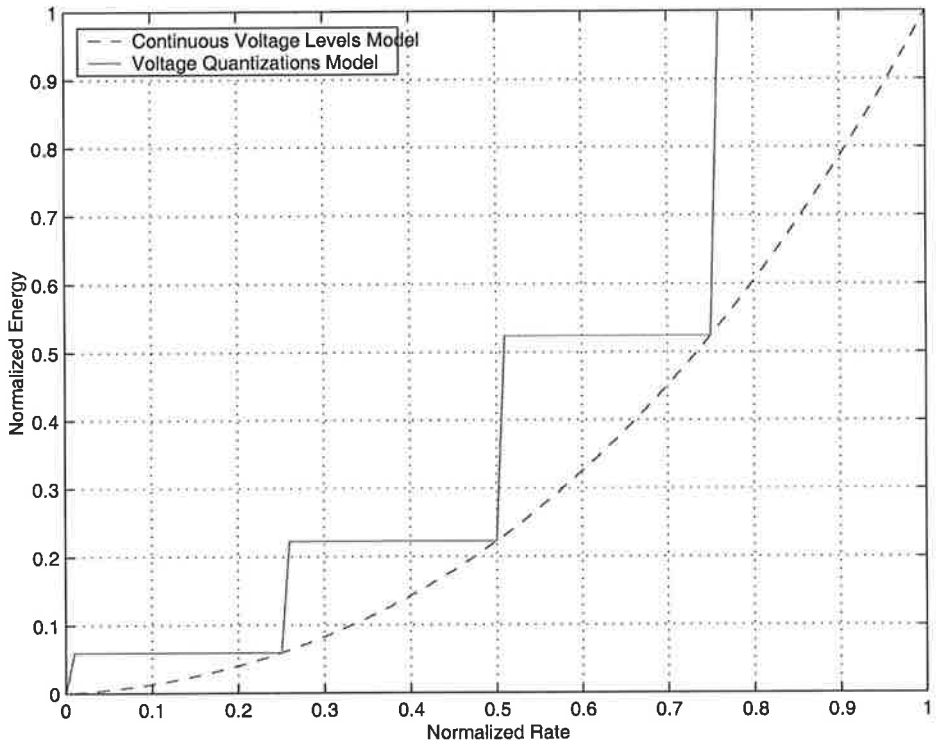


Figure 3.16: Energy and rate relationship for voltage quantization model

Based on a 4-level voltage quantization corresponding to rates of 0.25, 0.5, 0.75, and 1.0, the energy and rate relationship is shown in Figure 3.16. For comparison, the energy and rate relationship for the continuous voltage level model is also provided. As this figure shows, the energy curve of the voltage quantization model is inefficient compared to the ideal continuous voltage level model. This is mainly because the availability of only a handful of voltage levels implies that the vast majority of the workload values will have no matching voltage levels to be scaled to. This causes the vast majority of data samples to be scaled to a voltage quantization that is higher than would have been ideal under the continuous voltage level model. For example, a data sample with workload of 0.6 will have to be processed at rate quantizations of 0.75 to satisfy processing time constraints. (Use of a rate quantization of 0.5 violates the time constraint). However, processing at a higher voltage level than ideal causes the processing to finish before the maximum available time for processing (in fixed throughput mode of operation), and results in the system idling for part of the sample period.

This is illustrated in Figure 3.17. In this figure, voltage quantizations are represented by V_{q1} and V_{q2}, and the ideal operating voltage if continuous voltage levels are available is represented by V_{ideal}. Since a vast majority of data samples in a data sequence have an ideal voltage level not equal to one of the quantized levels, the system will incur increased idle energy losses. Thus, energy efficiency of voltage quantization model depends on efficient reduction of idle losses. The next chapter presents two idle loss reduction techniques.

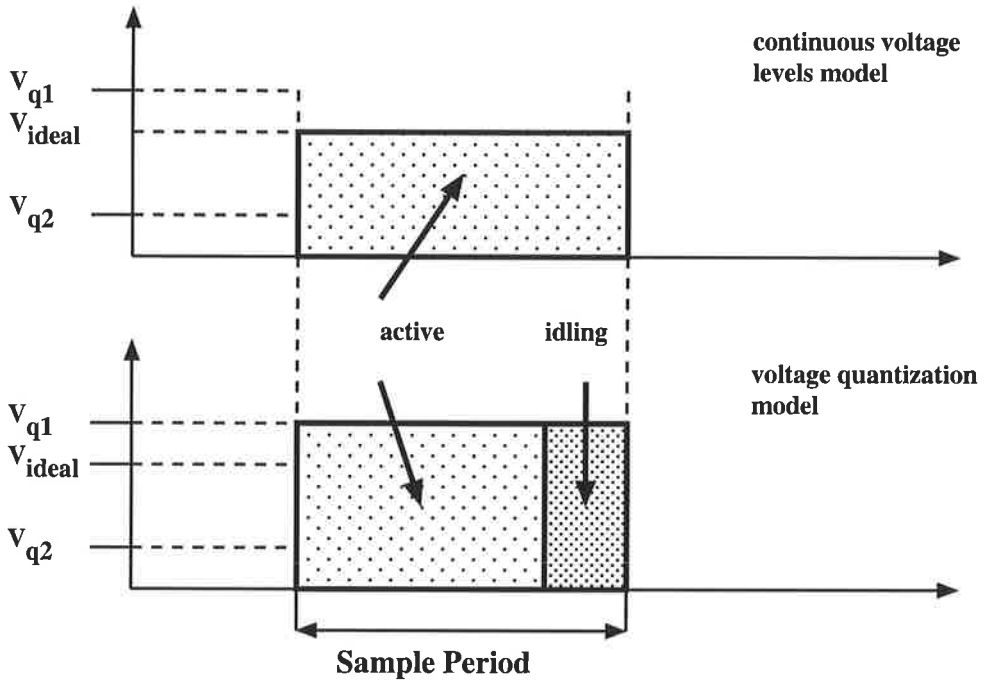


Figure 3.17: Idling in voltage quantization model

Apart from the idle losses, the open loop approach in the voltage quantization model has another limitation. Since the voltage levels are predetermined in terms of ON /OFF pulse width modulation times of the FETS (in the buck DC-DC converter), the actual voltage levels can vary according to the process and temperature variations. In order to allow for process and temperature variations, a hybrid controller comprising the lookup table concept of open-loop and a phased locked loop has been used in [Gut96]. In this implementation, the voltage quantizations are stored in random access memory (RAM) as opposed to read only memory (ROM) such that the adjustments to the voltage quantizations can be done at run-time and at a slower frequency (of 10kHz in this example).

3.5 Buffering and Workload Averaging

The energy model presented in Section 3.1.3 defines the static relationship between energy consumption for all possible sample workload values. In addition to this static energy model, the total energy consumption of a data sequence will also depend on the dynamic behavior or the workload distribution of data samples in the data sequence [Gut96]. If each data sample needs to be processed during each sample period (or no additional delay can be tolerated due to time constraints), the processing rate during a sample period must be greater than or equal to the sample workload. However, if some delay can be tolerated, storing data samples in a first-in-first-out (FIFO) buffer first and then processing them at a processing rate that is equal to the average buffered workload can further minimize the total energy consumption [Gut96]. This section demonstrates this concept.

Buffering data samples and processing them at the rate of the average buffered workload enables the processing rate to be decoupled from the individual sample workload values. This enables partial processing of data samples to occur during a sample period. In other words, a sample may be processed over a number of sample periods, or multiple samples may be processed over a single sample period depending on the sample workload values. However, since the maximum throughput (of fixed throughput mode) has a normalized upper bound of 1.0, all rate values during all sample periods will be less than or equal to 1.0.

The primary objective of buffering and workload averaging is minimization of total energy consumption [Gut96]. The buffering and workload averaging technique produces additional energy savings because of the convex or *concave up* nature of the energy and rate relationship as shown in Figure 3.5. To demonstrate how convex relationship of energy and rate can produce additional energy savings from buffering and workload averaging, consider the processing of two data samples with normalized workloads of 0.1 and 0.9 as shown in Figure 3.18. If the two data samples are processed without buffering, the average energy dissipation lies half-way along the chord joining the two points [0.1,$E(0.1)$] and [0.9,$E(0.9)$] or $0.5[E(0.1) + E(0.9)]$ as shown by point A on the curve. If the same two samples are buffered and processed at a rate that is equal to the average of the two workload values, then the energy dissipation equals to $E(0.5)$, and the corresponding point is shown by point B in Figure 3.18. Thus, buffering the two samples and processing them at average workload

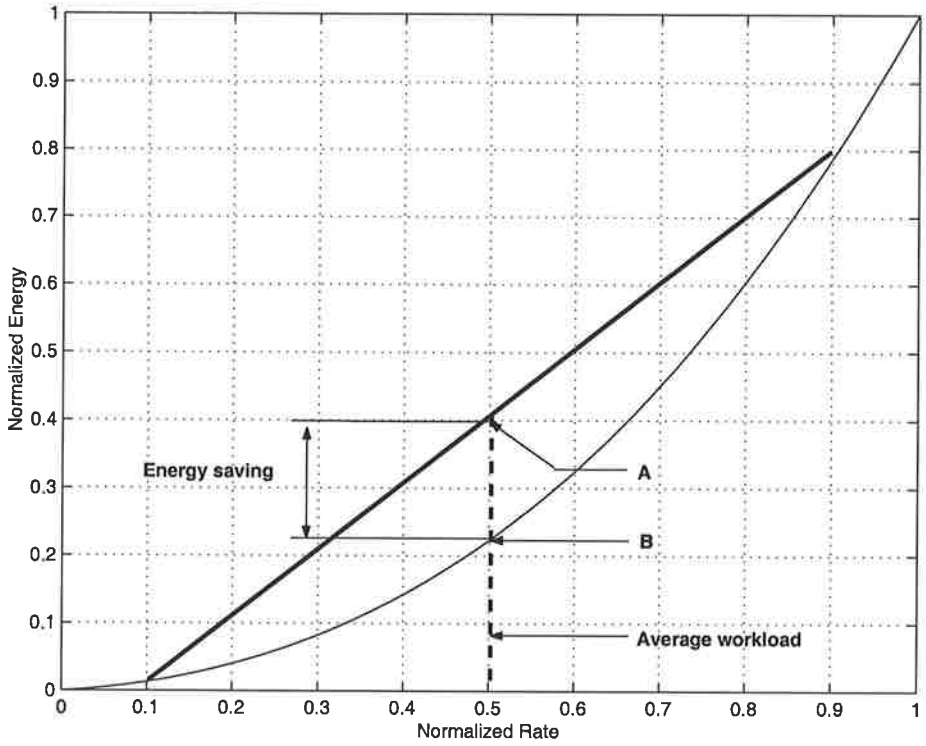


Figure 3.18: Energy savings from buffering and workload averaging

achieves an additional energy saving of $E(A) - E(B)$.

In order to calculate the rate value for a sequence of data samples, Gutnik [Gut96] uses the finite impulse response (FIR) filter function on the data sequence. Since the FIR filter is a moving average of the workload, it prevents any buffer overflows or underflows. Moreover, the filter is causal and linear time-invariant, and the non-negativity of rate values are guaranteed by choosing the variables according to Equations 3.8 and 3.9 [Gut96]:

$$r = \sum_{k=0}^{B-1} a_k w[k] \tag{3.8}$$

$$0 \leq a_k \leq 1 \qquad \sum_{k=0}^{B-1} a_k = 1 \tag{3.9}$$

where, a_k represents constant coefficients, $w[k]$ is the workload value at the $k+1$th buffer location, and B is the buffer size.

An example that shows how FIR filter is used for calculating the rate for a sequence of data samples is shown in Figure 3.19 [Gut96]. As this figure shows, workload averaging

requires two buffers, one to store the actual data samples (data buffer), and the other to store the workload values (workload buffer). When a new data sample arrives, it is buffered in d[2] of the data buffer. Then the buffered data sample is evaluated, workload value determined and stored in w[2] of the workload buffer. The processing rate r is then calculated during each sample period by taking the weighted average of the last B workload values in the workload buffer, whether or not the data samples have actually been processed. In this figure the buffer size is $B = 3$, and coefficients are $a_0 = a_1 = a_2 = 1/3$. Since the data samples are processed at varying speeds depending on the value of r, the contents of the workload buffer and data buffer will be different during operation. This means that in some sample periods, more than one sample is processed, while in others part of a sample or no sample is processed. This situation is evident in a number of sample periods in Figure 3.19. For example, in sample period 0, the calculated rate is 2, and this implies that only part of the data sample in d[2] (workload of 2 out of 6) will be processed. This leaves behind a workload of 4 out of 6 in d[2] to be processed at the end of sample period 0. Before the beginning of the sample period 1, data in the buffers will be shifted to the left making way for a new data sample to be buffered in location d[2] in the data buffer and its corresponding workload to be stored in location w[2] in the workload buffer. The rate calculation for sample period 1 results in a value of 3 for the rate, and this results in part of d[1] (3 out of 4) to be processed during sample period 1. Similarly, sample period 2 results in a rate value of 10, which processes d[0], d[1], and 6 out of 21 in d[2] to be processed in sample period 2. The review of the first three sample periods show that the original data sample in d[2] in the zeroth sample period (of workload 6) was partially processed over 3 sample periods, and that multiple samples can be processed over a single sample period (sample period 2). The sample period 3 also demonstrates the importance of using a workload buffer separate from the data buffer; if only one buffer (the data buffer) was used without a workload buffer and the rate is calculated using the data buffer, the resulting rate values will be smaller than that predicted by Equation 3.8, and the buffer will eventually overflow.

Extending the buffering idea to include all the data samples and processing them at the average workload minimizes the total energy consumption. Such an approach is however limited due to the following tradeoffs: 1) the increased memory and area overhead for maintenance of two buffers to store all the data and workload values, 2) the latency associated

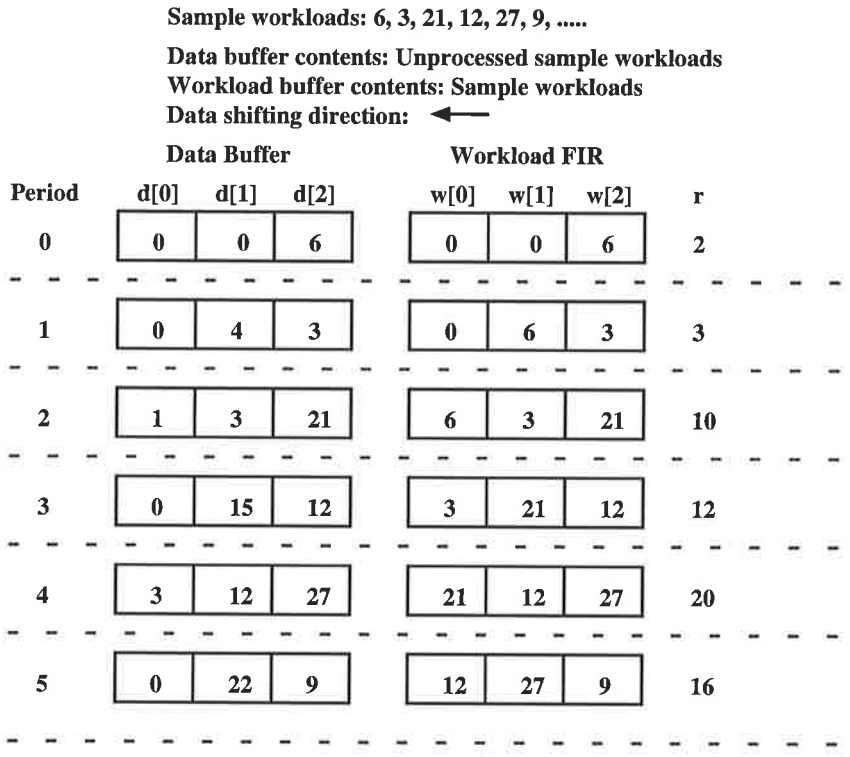


Figure 3.19: Sample rate calculations for buffering and workload averaging [Gut96]

with storing all the samples before they can be processed adds a propagation delay or latency equal to B sample period to the computation, and this may preclude any use of large buffers in real-time applications, and 3) the overhead necessary for tracking partially processed samples across sample period boundaries increases with buffer size. Thus, depending on available memory, area, and latency requirements for real-time processing, a suitable buffer size for a particular application must be determined.

Buffering and workload averaging was originally proposed for minimizing the energy consumption of the continuous voltage level model. This is because, the energy curve (Figure 3.15) for continuous voltage levels is concave up and buffering and processing at average workload produces the energy savings demonstrated above. However, as Figure 3.16 shows, the energy curve for voltage quantization model is no longer concave up, so buffering and workload averaging is no longer effective as a means for minimizing energy consumption with voltage quantization model.

3.6 Summary

This chapter provided an introduction to the concept of dynamic voltage and frequency scaling as applicable to fixed throughput mode multimedia applications. The chapter also presented the relevant architecture, energy models, and tradeoffs involved with the dynamic voltage and frequency scaling technique. Next, the chapter presented a number of workload determination techniques, and the details of the type of hardware circuitry necessary for DVS and the associated tradeoffs. Finally, a presentation of two voltage scaling models and how sample buffering and workload averaging can further minimize energy consumption was discussed.

Chapter 4

Improving Energy Efficiency of Voltage Quantization Model

The voltage quantization model is effective in achieving fast voltage transitions, low supply ripple, reduced complexity of DC-DC converter, and a very high the DC-DC conversion efficiency across the full range of scaled voltages. However, the main limitations of the voltage quantization model are increased idle losses due to the availability of a small number of voltage quantizations, and high energy cost of voltage transitions due to high output filter capacitance of the converter. Thus, improving the energy efficiency of the voltage quantization model involves the minimization of idle losses and voltage transitions.

This chapter presents the prior approaches for improving the energy efficiency of the voltage quantization model, and discusses the weaknesses of these approaches. This chapter then details the research work undertaken, and the experimental framework used for developing and testing the technique presented in this thesis.

4.1 State of the Art in Improving Energy Efficiency

The prior research is primarily focused on idle loss reduction as the *only* method for improving energy efficiency of the voltage quantization model. As a result, the energy cost of voltage transitions in the voltage quantization model remains completely unexplored.

This section presents the two idle loss reduction techniques that have been reported in literature [Gut96]. These techniques are 1) Clock gating, and 2) Voltage dithering.

4.1.1 Clock Gating

One of the biggest challenges high performance microprocessor designers face today is keeping the power dissipation to a minimum. Even though operational supply voltage and feature sizes are reduced in each new generation of microprocessor design, additional complexity of the new designs and increased clock speeds contribute to increased power dissipation. One popular technique for minimizing the power dissipation in high performance microprocessors is through a technique called clock gating. Clock gating involves partitioning the system clock into a clock distribution network and controlling it by turning ON only the portions of the system required for a given clock cycle while the other portions are effectively turned OFF. Turning off portions of the system at run-time enables the total switched capacitance of the system to be reduced, and this reduces the total power dissipation. This technique is effective for high performance microprocessors because clock distribution is the biggest power consuming component in the system [TSR+98],[Dev01],[GBJ98].

Figure 4.1 shows a typical circuit-level implementation of the clock gating technique [TSR+98]. As this figure shows, the system is divided into subsystems (A, B, C, and D), and the clock signal is *qualified* by special enable signals (Enable_A, Enable_B, and Enable_C). In this implementation, the block D is always ON (functioning), while the other three blocks are enabled during run-time. By replacing the regular clock buffers with qualifying gates as shown in this figure also has the secondary benefit of reduced area and performance overheads [TSR+98].

Gutnik *et. al* [Gut96] shows how clock gating can be applied to minimize idle loss in the voltage quantization model. This approach involves turning off the clock when idle time is detected. This method is demonstrated in Figure 4.2. In this figure $Vq1$ and $Vq2$ represent two voltage quantizations. Based on this approach and assuming no overhead associated with clock gating, a 4-level voltage quantization, and zero energy loss during clock gated period, the energy and rate/workload relationship for the voltage quantization model is shown by the "stair-step" curve in Figure 4.3 [Gut96]. For reference, the energy curves for continuous voltage level model and the voltage quantization model (without clock gating) are also shown. As this figure shows, the idle loss reduction due to clock gating results in a more efficient energy model for voltage quantization model, even though the

resulting energy model is not as good as the continuous voltage level model.

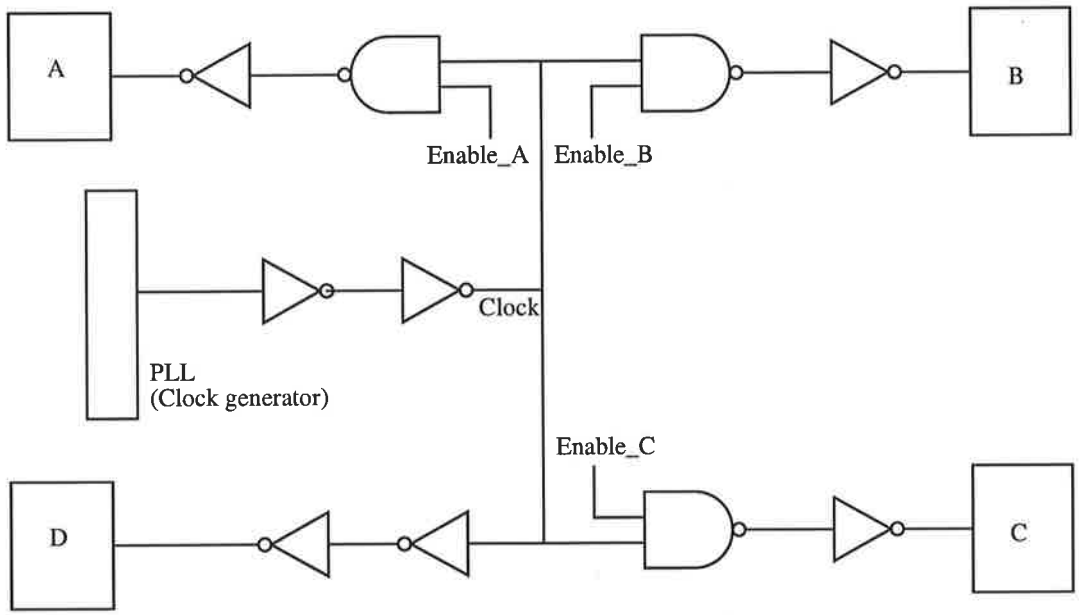


Figure 4.1: Clock gating of clock networks [TSR+98]

4.1.2 Voltage Dithering

Voltage dithering involves processing a data sample at *two* voltage quantizations to eliminate the idle loss [Gut96]. This implies that part of the sample period is processed at one voltage quantization and the remainder of the sample period is processed at another voltage quantization. Figure 4.4 demonstrates the voltage dithering approach. As the figure shows, the approach involves partial processing of the data sample at each of the two voltage quantizations such that the full sample period is used up for computation. This eliminates idle losses. As before, V_{q1} and V_{q2} are two voltage quantizations.

To demonstrate how voltage dithering is performed, consider a data sample with normalized workload of 0.6 being processed using a DVS system with a 4-level voltage quantization corresponding to normalized rate quantizations of 0.25, 0.5, 0.75, and 1.0.

The first step involves the identification of the two voltage quantizations used for voltage dithering (V_{q1} and V_{q2} in the figure) of this data sample. The voltage quantizations selected for a data sample are the ones that are neighboring the selected sample workload. For ex-

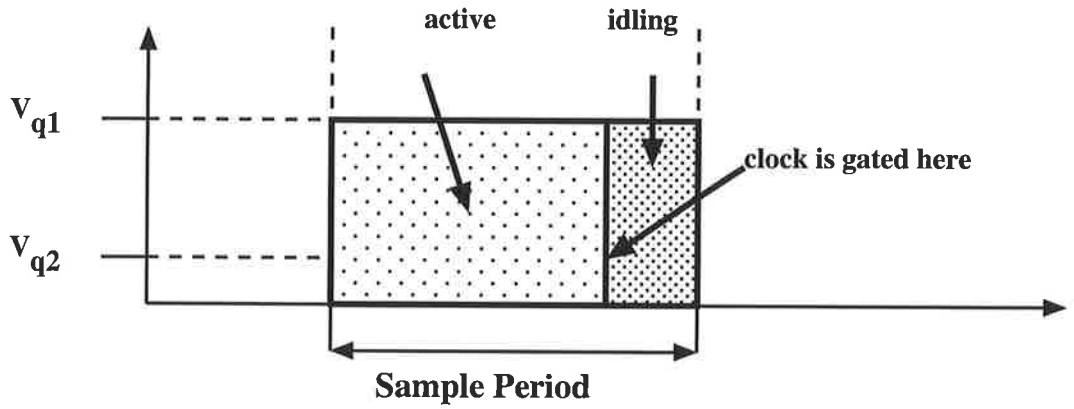


Figure 4.2: Idle loss reduction in voltage quantization model through clock gating

ample, our sample with workload of 0.6 will have the neighboring rate quantizations of 0.5 and 0.75 around it. The second step is to calculate when to dither the voltage such that the idle loss is eliminated. In order to calculate the point of dithering, the available work must be processed over two voltage levels (rate quantizations or clock frequencies) and since total workload processed remains the same with or without dithering, the condition in Equation 4.1 must be fulfilled. In this equation, $workload_{Vq1}$ and $workload_{Vq2}$ represent the workload processed at each of the voltage quantizations.

$$workload_{available} = workload_{Vq1} + workload_{Vq2} \tag{4.1}$$

Assuming that p (normalized) represents the proportion of the sample period processed at the low voltage quantization (V_{q2}), the proportion of time processed at higher voltage quantization is (1-p). Since the workload equals to processing time multiplied by processing rate, Equation 4.1 can be transformed to Equation 4.2.

$$w_{sample} = (1 - p) * r_{q1} + p * r_{q2} \tag{4.2}$$

where w_{sample} is the available workload of the data sample, r_{q1} is the rate quantization corresponding to voltage quantization of V_{q1}, and r_{q2} is the rate quantization corresponding to voltage quantization of V_{q2}. Substituting the values of our example data sample with workload of 0.6 and rate quantizations of 0.5 and 0.75 shown above, the value of p comes

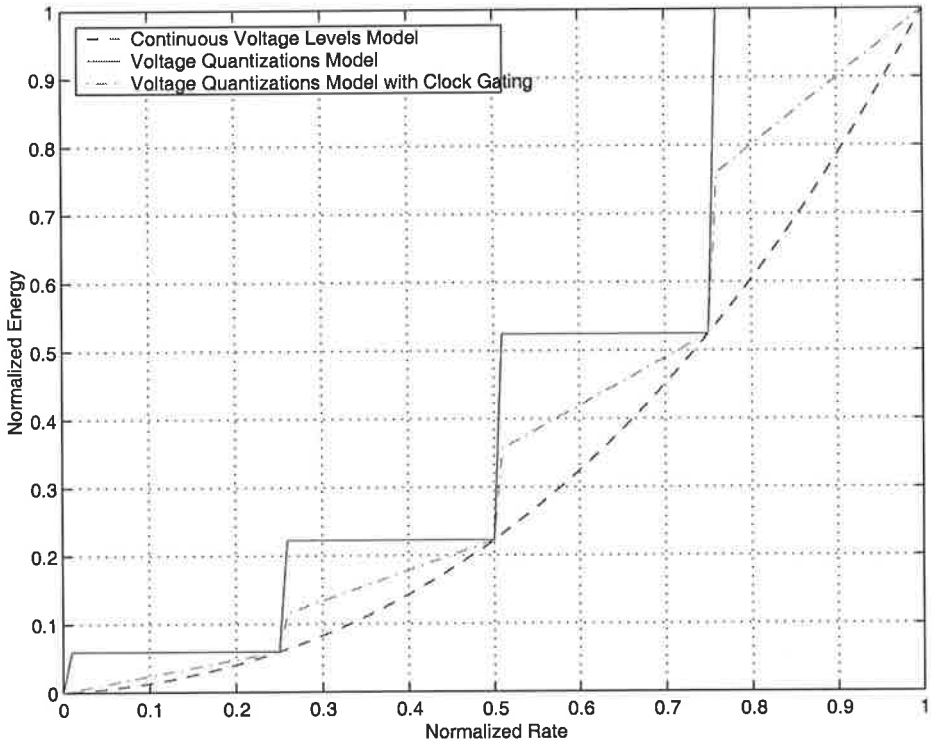


Figure 4.3: Energy vs. rate relationship for voltage quantization model with clock gating [Gut96]

out to be 0.6. In other words, the dithering point occurs at 40% of the sample period when the sample is first processed at the high voltage quantization (V_{q1}). After the dithering transition from $V_{q1} \rightarrow V_{q2}$, processing of the sample continues for the remaining 60% of the sample period at the low voltage quantization (V_{q1}). Thus, voltage dithering enables 100% use of the sample period for a sample that otherwise would have idled for part of the sample period with the voltage quantization model.

In order to present the energy model for voltage quantization model with voltage dithering, the following assumptions are made: 1) the computational overhead involved with calculating the voltage dithering point per data sample is negligible compared to the data processing computation, 2) the voltage transition time is short enough to allow two voltage transitions within a sample period (the initial transition to high voltage quantization and the dithering transition between voltage quantizations), and 3) the energy cost of voltage transitions is negligible. Based on these assumptions, the energy curve for a 4-level voltage

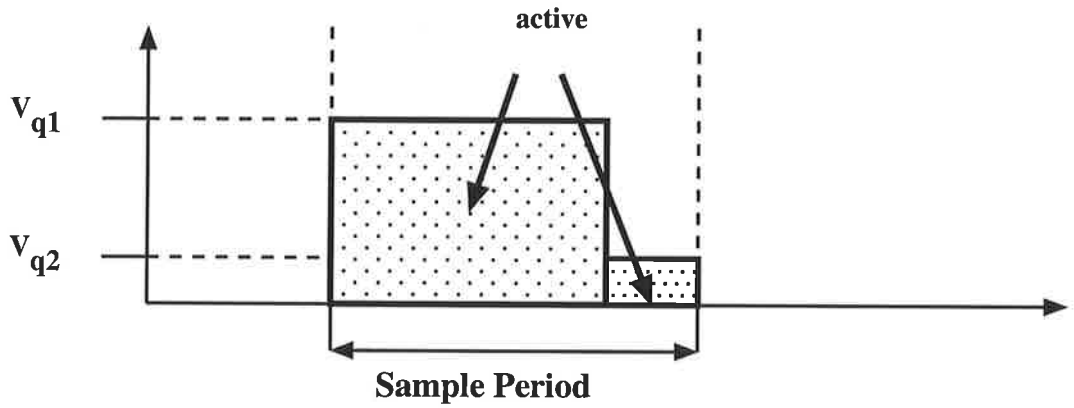


Figure 4.4: Idle loss elimination in voltage quantization model through voltage dithering

quantization is shown in Figure 4.5 [Gut96]. As this figure shows, a 4-level voltage quantization provides a very close approximation to the continuous voltage level model.

4.2 Limitations of Prior Art

This section discusses limitations of the two idle loss reduction techniques presented above.

4.2.1 Clock Gating

As Section 4.1.1 showed, clock gating is a useful technique for power minimization of very high performance microprocessor architectures where the power dissipation due to the clock dominates the total power of the microprocessor. However, use of the clock gating technique has a number of limitations. One of the main concerns of clock gating is that the gated circuit may not power up in time for the next clock cycle [TSR+98]. This implies that if clock gating is done very frequently, the gated circuitry may not be ready to function when required. The other issue is the introduction of spurious transitions or glitches caused by the modified clock signals [TSR+98]. Because of this, the clock gated designs require careful timing verification and functional validation. For higher speed designs (over 100MHz), the additional gate used for qualifying the clock signal may also contribute towards timing critical clock skews. Moreover, turning the clock signal ON/OFF also causes large variations to current, and this

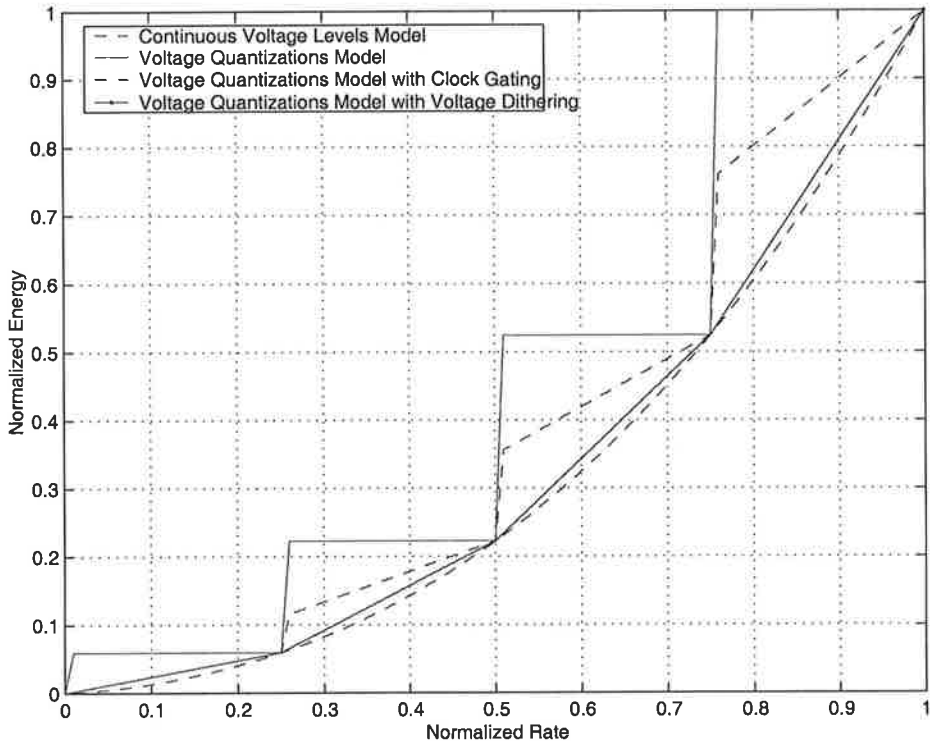


Figure 4.5: Energy vs. rate relationship for voltage quantization model with voltage dithering [Gut96]

leads to *transient power* losses [TSR+98].

In addition to the complex design and test effort involved with clock gated system design, clock gating is not suited to reducing idle energy loss in the voltage quantization model for multimedia applications. This is because using a small number of voltage quantizations with the voltage quantization model implies that a large proportion of data samples in a data sequence would be processed at a higher voltage quantization (than would have been ideal with continuous voltage level model). Consequently, this implies that a very large proportion of the data samples would produce idle losses, and if clock gating is used for reducing idle losses, this would require clock gating to be activated during a vast majority of sample periods. Since clock gating is ideally performed very infrequently due to the above discussed limitations, it will be ineffective as an idle loss reduction technique in the voltage quantization model for multimedia applications. A simpler and more effective approach would thus be to use voltage dithering.

4.2.2 Voltage Dithering

Section 4.1.2 showed how a 4-level voltage quantization can produce an energy curve that approximates the energy curve of the continuous voltage level model. However, this energy model is derived from ideal assumptions about voltage transitions, that they incur negligible energy cost and take short transition time. In reality, the voltage dithering technique has a number of limitations.

Firstly, the voltage dithering approach produces an additional voltage transition per data sample. Since a majority of data samples in a data sequence will produce idling and hence require dithering, a large proportion of data samples will produce the additional voltage transitions. Since the voltage quantization model has the shortcoming of high output capacitance (to maintain high DC-DC converter efficiency and low output voltage ripple), each voltage transition will contribute to increased transition energy losses. Since a vast majority of samples produce the additional voltage transition, the total transition loss can be significant depending on the data processing computation.

Secondly, since voltage transitions incur finite transition times, voltage dithering may be infeasible if the sample period is not long enough to allow an additional voltage transition. Thus, depending on the DC-DC converter parameters and the multimedia computation where DVS is used, voltage dithering may not be feasible.

Thirdly, each dithering transition involves the overhead of calculating the dithering point. The determination of dithering point involves calculation of p of Equation 4.2 as shown in Equation 4.3. This equation shows that during each sample period, determination of dithering point involves a computational overhead of two *subtract* operations and a single *division* operation.

$$p = \frac{r_{q1} - w_{sample}}{r_{q1} - r_{q2}} \tag{4.3}$$

Apart from this overhead, voltage dithering will incur other computational overheads when dithering is done in computations where the sample period and the voltage transition time are of the same order of magnitude. In such cases, the dithering transition is only made if energy can be saved, compared to incurring idle losses. For example, based on Equation 3.7 the energy cost of a dithering transition from $V_{q1} \rightarrow V_{q2}$ can be calculated as shown in

Equation 4.4:

$$E_{tran(q1 \rightarrow q2)} = (1 - \eta) \cdot C \cdot \mid V_{q1}^2 - V_{q2}^2 \mid \tag{4.4}$$

where, C is the output filter capacitance of the DC-DC converter, and η is the converter efficiency.

Similarly, using Equation 3.2 the energy cost of operation at V_{q2} voltage quantization ($E(r_{q2})$) can be calculated by first calculating the normalized energy at the voltage quantization. (Here, r_{q2} refers to the rate quantization at voltage quantization of V_{q2}).

Thus, effectiveness of a dithering transition can be evaluated by comparing $p * E(r_{q2})$ and $E_{tran(q1 \rightarrow q2)}$.

The overhead involved with these comparisons can be reduced by performing the calculations once and storing the results in memory. However, the underlying problem with this approach is the necessity to have a good hardware knowledge of the system to obtain parameters such as threshold voltage (in Equation 3.2) and DC-DC converter efficiency (in Equation 3.7) such that the system can be programmed to make energy conscious dithering transitions.

The final limitation is the increased noise due to the additional voltage transitions of voltage dithering. Since DC-DC converters are noisy components, additional transitions introduced by dithering can cause noise interference to the sensitive components in a system. This is particularly true in portable wireless systems where multimedia and dynamic voltage scaling is being increasingly exploited.

4.2.3 Summary

Considering the limitations of prior art and the lack of research on transition energy loss, we believe that new research work that takes transition losses into consideration and develops a novel technique for reducing *both* idle loss and transition loss is necessary for comprehensively improving the energy efficiency of the voltage quantization model.

4.3 Research Undertaken

The main objective of our research is to take a fresh look at the voltage quantization model and develop a new algorithmic approach for improving its energy efficiency such that a small number of voltage quantization provides a very good approximation to continuous voltage level model. Unlike prior art, we envisage to develop a new technique that not only reduces the idle loss, but also the transition energy cost of fixed throughput mode computations using the voltage quantization model.

In order to formulate the research problem addressed in this research, it is important to define the term *total energy of computation*. The definition of this term is given in the section below.

4.3.1 Total Energy of Computation

The term *computation* in this thesis refers to a fixed throughput mode computation that uses dynamic voltage and frequency scaling to reduce idle loss and minimize energy consumption. Thus the term *total energy of computation* defines the total energy cost of processing all the samples of a data sequence. Since the computation occurs in a dynamic voltage and frequency scaling environment, total energy of computation will include the energy cost of processing the data sample and the energy cost of voltage transitions. This relationship is shown in Equation 4.5:

$$E_{total} = E_{pr} + E_{tr} \tag{4.5}$$

where, E_{total} is the total energy of computation, E_{pr} is the total energy for data processing, and E_{tr} is the total energy cost of voltage transitions.

The total energy of data processing, E_{pr}, can be defined in terms of the rate and energy model as shown in Equation 4.6:

$$E_{pr} = \sum_{r=0}^{1} E(r) \cdot P(r) \tag{4.6}$$

where, r is the normalized rate, $E(r)$ is the energy vs. rate relationship for the voltage quantization model (as shown in Figures 4.3, and 4.5), and $P(r)$ is the sample workload/rate

distribution of processed data samples expressed as a probability mass function.

Similarly, total transition energy, E_{tr}, can be defined in terms of transition energy cost as defined by Equation 3.7 and the total number of transitions expressed as a probability mass function, as shown by Equation 4.7:

$$E_{tr} = \sum_{\tau=1}^{T} E_{tr}(\tau) \cdot P_{tr}(\tau) \tag{4.7}$$

where, τ is an integer variable representing different voltage transitions possible for the voltage quantization model, T is the maximum number of different types of voltage transitions possible with the voltage quantization, $E_{tr}(\tau)$ is the transition energy for the τ type of transition as defined by Equation 3.7, and $P_{tr}(\tau)$ is the probability mass function value of transition type τ.

Thus, minimization of total energy of computation involves minimization of data processing energy and transition energy. According to Equation 4.6, data processing energy can be minimized by reducing the idle loss through optimization of the energy model $E(r)$ or workload distribution expressed as a probability mass function $P(r)$, or both. Similarly, minimization of transition loss can be accomplished through optimization of energy efficiency of transitions taken E_{tr}, and transition distribution expressed as a probability mass function P_{tr}, or both.

However, the contribution of the E_{tr} component towards E_{total} (in Equation 4.5) depends on the E_{pr} component. Since the E_{pr} component is a function of the sample period (T_s according to Equation 3.2), this implies that the contribution of E_{tr} towards total energy of computation depends on the computation itself. For example, if T_s is much larger than the voltage transition time, the contribution of E_{tr} can be ignored. Conversely, as T_s approaches the voltage transition time, the contribution of E_{tr} becomes very important. However, when T_s approaches the voltage transition time, processing time will not be long enough to accomodate the voltage transition, and this makes dynamic voltage scaling completely ineffective.

In order to develop a more general approach to minimizing total energy of computation without considering specifics of the computation itself, we focus on reducing the magnitude of both E_{pr} and E_{tr} separately, without considering their relative magnitudes. This way,

our proposed approach can be used in *any* dynamic voltage scaling application that uses the voltage quantization model.

4.3.2 Research Problem Statement

The key problem this research addresses is reduction of total energy and voltage transition count of fixed throughput mode multimedia computations using dynamic voltage and frequency scaling with the voltage quantization model. The proposed research aims to develop a novel approach that reduces both the data processing and transition energy components such that the energy efficiency of the voltage quantization model is improved. Additionally, we aim to reduce the total number of voltage transitions incurred for the computation.

4.3.3 Research Directions

As discussed above, prior research does not document any comprehensive approach that minimizes both idle loss and voltage transition count. From the existing idle loss reduction techniques of clock gating and voltage dithering, superior energy efficiency is achieved by voltage dithering at the expense of increased voltage transitions.

Considering Equation 4.6, the voltage dithering technique achieves improved energy efficiency from an improved energy model $E(r)$. This is also shown in Figure 4.5. Our research takes a fundamentally different approach to minimizing idle losses by optimizing the second component of Equation 4.6, the workload distribution of the data sequence expressed as a probability mass function or $P(r)$. By transforming the $P(r)$ term, we envisage that idle loss can be minimized. Moreover, we anticipate that transformation of $P(r)$ will also result in significant reductions in transition count, and consequently the E_{tr} as well.

Since the envisaged approach aims to transform $P(r)$, this enables the use of a very simple energy model with few assumptions for E_{pr}. This makes the proposed approach less dependent on hardware issues, which in turn improves its versatility. The energy model used for this research is the model shown in Figure 3.16 for the voltage quantization model and will be discussed in more detail in the next section.

This desired transformation of $P(r)$ that minimizes the idle losses is shown in Figure 4.6. In this figure, the top figure shows the $P(r)$ distribution of a typical data sequence. Using

a 4-level voltage quantization corresponding to normalized rate quantizations of 0.25, 0.5, 0.75, and 1.0, it is clear that the workloads of the majority of data samples do not exactly match the available rate quantizations. This is what causes idle times and consequently idle losses in voltage quantization model. However, if the $P(r)$ distribution of the top figure can be transformed to the $P(r)$ distribution of the bottom figure, the idle time and losses are virtually eliminated. This research explores this possibility by utilizing sample buffering and dynamic behavior of data sequences.

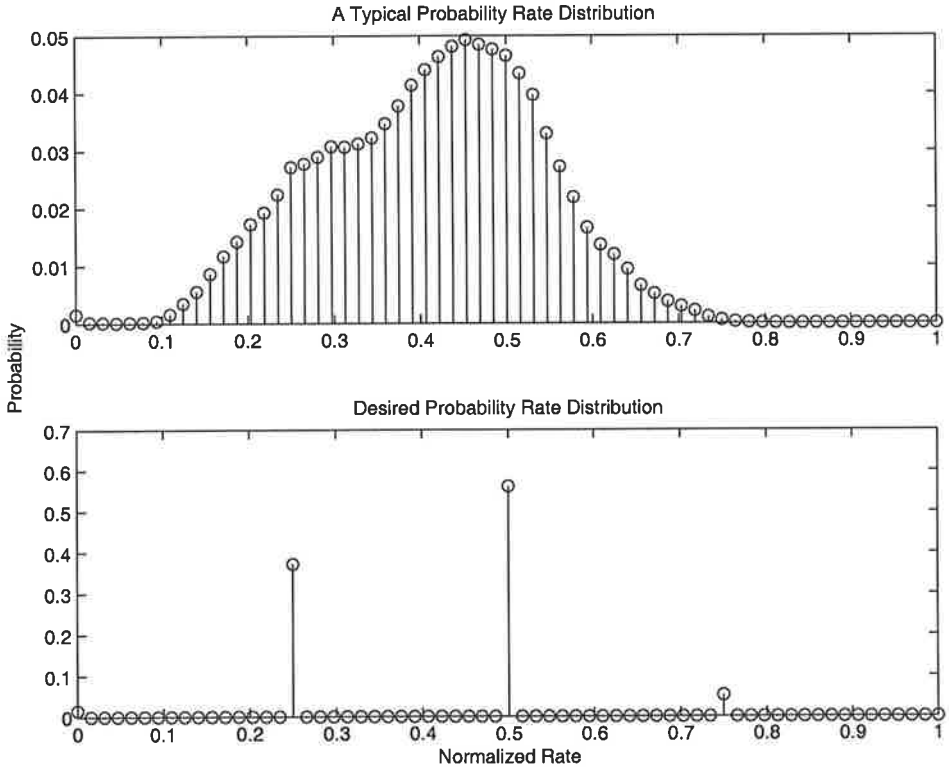


Figure 4.6: Desired optimization of rate distribution for idle loss elimination

4.4 Platform for Experimentation

This section presents the experimental framework used in our research work.

4.4.1 Energy Model

The energy model used in our experimental work is shown in Figure 4.7. This is the same energy model that was discussed in Section 3.4.2 an shown in Figure 3.16. This model is based on the first order energy model of Section 3.1.3. This model assumes that one voltage transition per sample period is used during voltage scaling and if processing finishes early in a sample period, idle time occurs which consumes the *same* energy as data processing. Though this model is very simple, and can be used in a variety of voltage scaling applications without knowledge about specific system details, it contributes to very high idle energy losses. Thus, unless our proposed approach is extremely effective in optimizing $P(r)$, the total energy of computation will be significantly deteriorated.

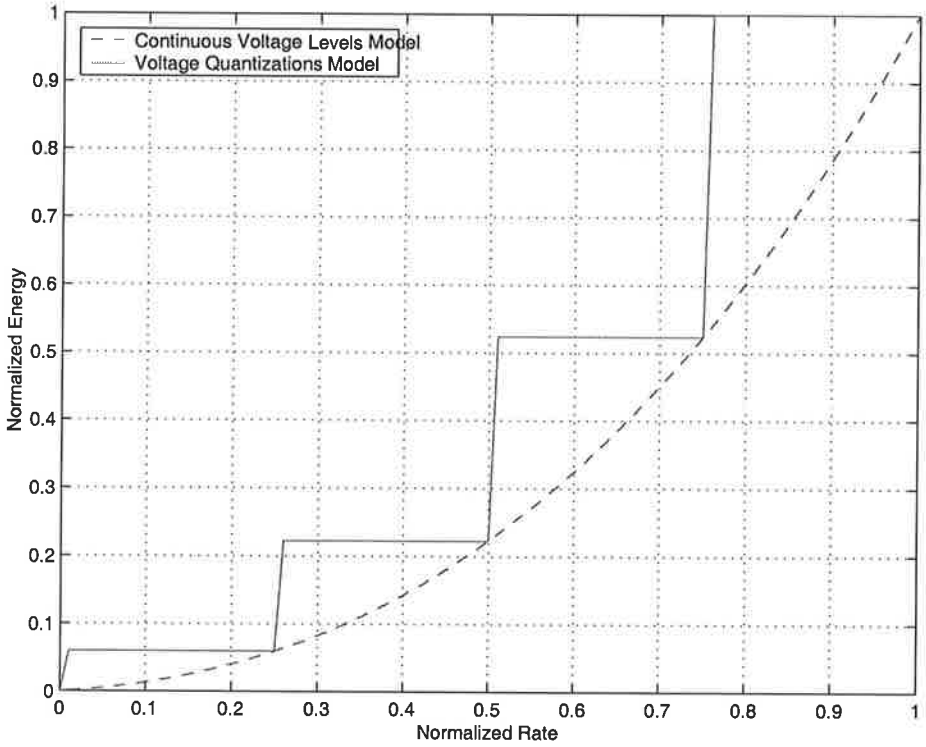


Figure 4.7: Energy model used for this research

4.4.2 Voltage Quantizations

For the development and testing of our proposed approach, we use a voltage quantization of 4. This is because most prior research involving voltage quantization model uses either 4 or 5 discrete voltage levels [Gut96],[CGX96],[LYyTC99]. However, in Chapter 7 we perform analysis at voltage quantizations of 2, 3, 4, and 5.

4.4.3 Transition Energy Computation

We use the recently proposed transition energy model from lpARM project at Berkeley [BPSB00] for evaluating the transition energy component.

4.4.4 Multimedia Computation

The selection of the multimedia computation for our experimental work involves two key criteria. The selected computation 1) must be data-dependent and must be amenable for calculating the sample workload *a priori*. 2) must be open standard based and the source code of implementations must be available for experimentation. Based on the first criteria, we chose the Inverse Discrete Cosine Transform (IDCT) as our computation. IDCT computation is one of the most compute-intensive [CDCP01] and is the core computation in a number of multimedia codecs. For example in MPEG codecs, IDCT computation contributes to about 22% of the total decoding time [BK95]. Additionally, this computation is data-dependent, and availability of the alternative forward mapped implementation or FMIDCT [MW92] makes the *a priori* calculation of workload possible. This algorithm has been used in a number of low power research works [CGX96],[Xan99] due to its ease in calculating the sample workload exactly and *a priori* based on the number of non-zero coefficients in a data sample (8×8 block). Based on the second criterion, we chose MPEG video decoding as our application domain for the IDCT computation. We chose MPEG standard because of its versatility as a set of rapidly evolving standards for a wide range of multimedia applications. From the array of possible MPEG multimedia applications, we focus on video decoding primarily because of its use in a large number of portable multimedia products such as portable video players, video phones, etc. Moreover, MPEG video decoding is known to exhibit large variations in

throughput (workload) [CDCP01] and we use this property of MPEG video to illustrate our proposed approach.

For completeness, a brief overview of the Inverse Discrete Cosine Transform (IDCT) and the FMIDCT implementation are given below.

Inverse Discrete Cosine Transform (IDCT) is the reverse of the Discrete Cosine Transform (DCT), and involves the conversion of scaled cosine basis functions into a set of data samples. For a 2-D array of samples, $f(x, y)$, the 2-D IDCT computation is given by Equation 4.8:

$$f[x,y] = \sum_{\mu=0}^{7}\sum_{\nu=0}^{7} \frac{C(\mu)}{2}\frac{C(\nu)}{2} F(\mu,\nu) cos\left(\frac{(2x+1)\mu\pi}{16}\right) cos\left(\frac{(2y+1)\nu\pi}{16}\right) \tag{4.8}$$

where $F(\mu, \nu)$ is the array of 2-D DCT coefficients and $C(\mu)$ and $C(\nu)$ are given by:

$$C(\mu) = 1/\sqrt{2} \quad if \ \ \mu = 0$$
$$C(\nu) = 1 \qquad if \ \ \nu > 0$$

The formulation of the Forward Mapped Inverse Discrete Cosine Transform (FMIDCT) is given by Equation 4.9 [MW92].

$$\begin{bmatrix} x_{0,0} \ldots x_{0,7} \\ x_{0,1} \ldots x_{1,7} \\ \vdots \\ x_{7,0} \ldots x_{7,7} \end{bmatrix} = X_{0,0} \begin{bmatrix} c_0^{0,0} \\ c_1^{0,0} \\ \vdots \\ c_{63}^{0,0} \end{bmatrix} + X_{0,1} \begin{bmatrix} c_0^{0,1} \\ c_1^{0,1} \\ \vdots \\ c_{63}^{0,1} \end{bmatrix} + \ldots + X_{7,7} \begin{bmatrix} c_0^{7,7} \\ c_1^{7,7} \\ \vdots \\ c_{63}^{7,7} \end{bmatrix} \tag{4.9}$$

where $x_{i,j}$ are the reconstructed image data, $X_{i,j}$ are the input DCT coefficients, and $c_k^{i,j}$ are the reconstruction kernels of the input coefficients.

Based on Equation 4.9 , the normalized workload (w) for a data sample in FMIDCT algorithm can be determined *a priori* using Equation 4.10. For this formulation a block size of 8×8 is used.

$$w = \frac{n}{64} \tag{4.10}$$

where, n ($0 \leq n \leq 64$) is the number of non-zero coefficients in the 8×8 image block.

4.4.5 Test Data

For the analysis of the research problem, and development and testing of the proposed approach, a collection of video sequences encoded in MPEG-2 were chosen as test data. These video sequences are composed of 300 color frames in quarter common intermediate format (QCIF) and represent a wide variety of content types, levels of motion, and camera panning. Tables 4.1 and 4.2 show the MPEG-2 compression parameters and the properties of the test video sequences, respectively. Some sample frame sets from test video sequences are given in Appendix A.

Table 4.1: Summary of MPEG-2 compression parameters

Parameter	Value
Number of Frames in a Group Of Pictures (GOP)	15
I/P Frame Distance	3
Horizontal Picture Size (Pixels)	176
Vertical Picture Size (Pixels)	144
Frame Rate	30 frames/sec
Bit Rate	4 Mbits/sec
Profile	Main
Level	Low
Sub-sampling Format	4:2:0

4.4.6 MPEG Codecs

The MPEG-2 codec software (and source code in C) used for video compression and decompression was acquired from the MPEG Software Simulation Group [MSS].

4.4.7 Simulation Environment

The experimental work for the research presented in this chapter is based on simulations performed using MATLAB (version 5) software package from MathWorks Inc [Mat]. In this

Table 4.2: Properties of the MPEG-2 test video sequences

Video Sequence	Content	Motion	Camera Pan/Direction
Akiyo	woman's head and shoulders	low	no
Carphone	man talking on the phone	low	no
Coastguard	boat	medium	constant/horizontal
Container	ship	medium	no
Hall	surveillance	low	no
Foreman	construction site	medium	constant/various
Mother	mother and young daughter	low	no
Silent	woman using sign language	medium	no

work, no special Matlab toolkits were used. All the source code used in the experimental work is included in the CD included in Appendix C.

4.4.8 Hardware Platform

All the matlab simulations in this research were performed on a dual-CPU Sun Enterprise 250 Server with 2GB of RAM and running Solaris 9.

4.5 Summary

This chapter discussed the limitations of prior approaches for minimizing the idle losses in the voltage quantization model. This chapter also provided an overview of the research work undertaken and the experimental platform used for our research. The next chapter presents a novel approach for enhancing the energy efficiency of the voltage quantization model.

Chapter 5

Rate Selection: A New Approach

Using the eight MPEG-2 test video sequences, this chapter presents the analysis of probability rate/workload distributions in identifying the proportion of data samples that contribute to idling in the voltage quantization model. Then, the concept of the proposed rate selection approach for minimizing idle losses is presented.

5.1 Problem Analysis

As discussed in Section 3.4.2, the voltage quantization model produces idle losses because the vast majority of data samples in a data sequence will have no matching voltage quantizations for voltage scaling. Using the actual workload values from MPEG-2 test video data, this section provides a quantification of this fact.

Figures 5.1 and 5.2 show the actual workload/rate distributions for the 8 test data sequences. Based on this data, the proportion of data samples that have matching rate quantizations and cause no idling are plotted in Figure 5.3.

From Figures 5.1,5.2, and 5.3, the following conclusions can be made:

- As Figures 5.1, and 5.2 show, workload distribution patterns are unique for the data sequences.
- As Figure 5.3 shows, the proportion of data samples that exactly match the workload to available rate/voltage quantizations is very small. For the 8 test video data sequences used in this analysis, the maximum proportion of workloads that had matching

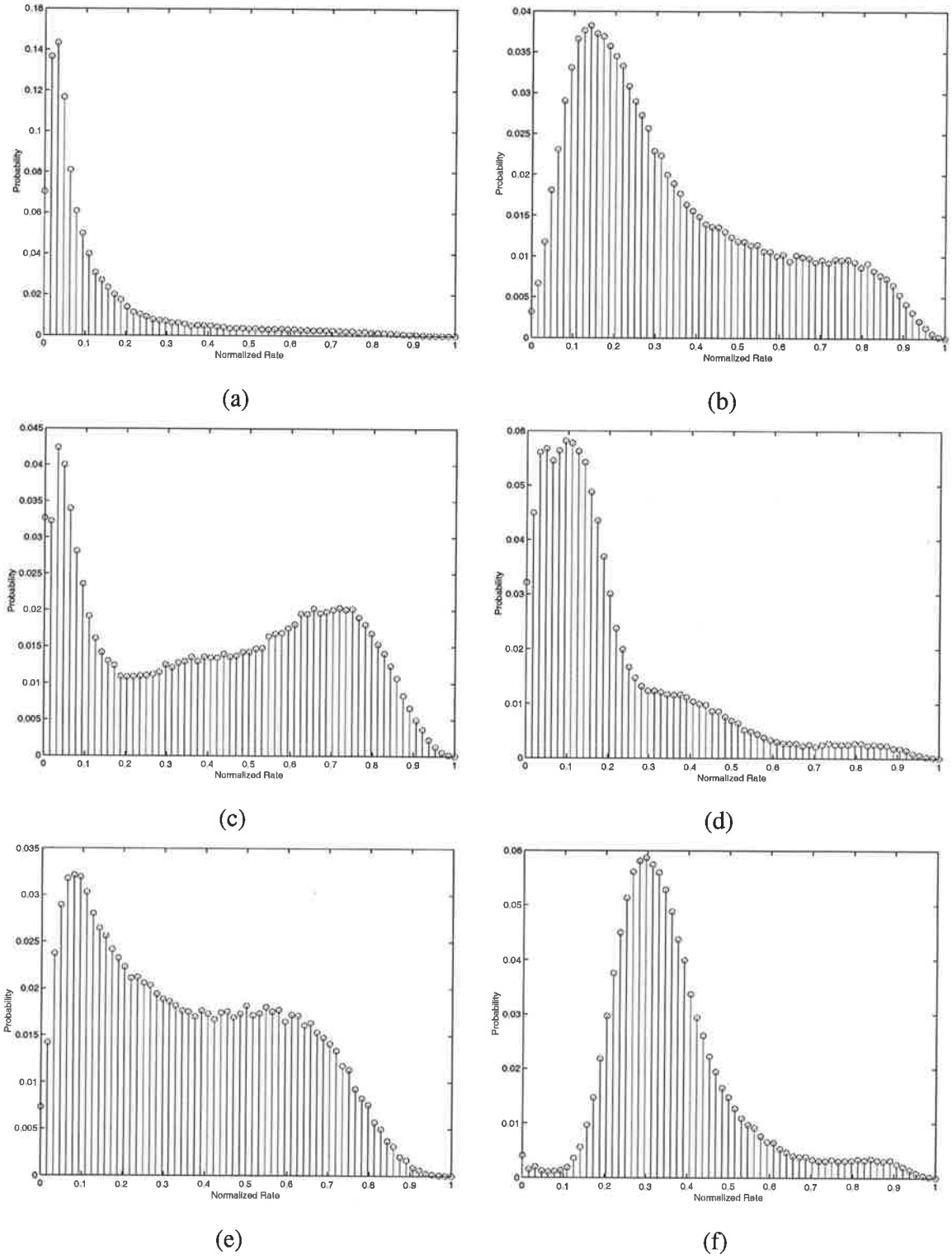
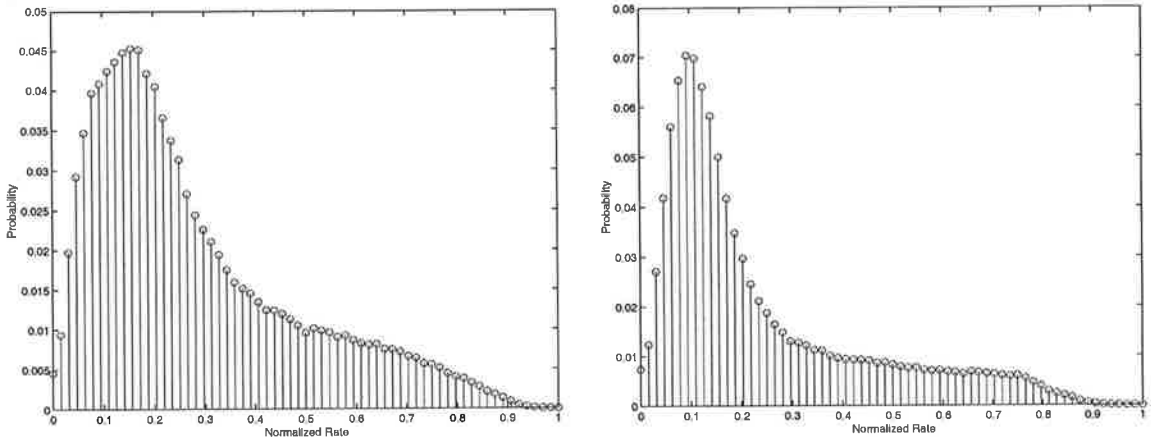


Figure 5.1: Workload/rate distribution for MPEG-2 test video data. (a) Akiyo, (b) Carphone, (c) Coastguard, (d) Container (e) Foreman, (f) Hall



(g) (h)

Figure 5.2: Workload/rate distribution for MPEG-2 test video data. (g) Mother, and (h) Silent

rate/voltage quantizations was less than 7%.

Consequently, when the voltage quantization model is used for dynamic voltage scaling, over 93% of all data samples in these data sets will have to be processed at higher voltage quantizations and this contributes to idle losses.

The next section presents a novel algorithmic approach that attempts to transform the workload distributions to minimize the idle loss.

5.2 Rate Selection

5.2.1 Introduction

The intent of this research is to transform the raw workload distributions of data sequences such that the resulting distributions will be focused at rate/voltage quantizations. Thus, the function of the proposed rate selection approach is shown in 5.4.

However, such a transformation is impossible if data samples need to be processed as they are received, without any delay. In such a case, the processing rate (r) will be equal to workload (r=w) and no transformation in workload distribution can be achieved. If on the other hand, a fixed processing delay can be tolerated, data samples can be buffered and the

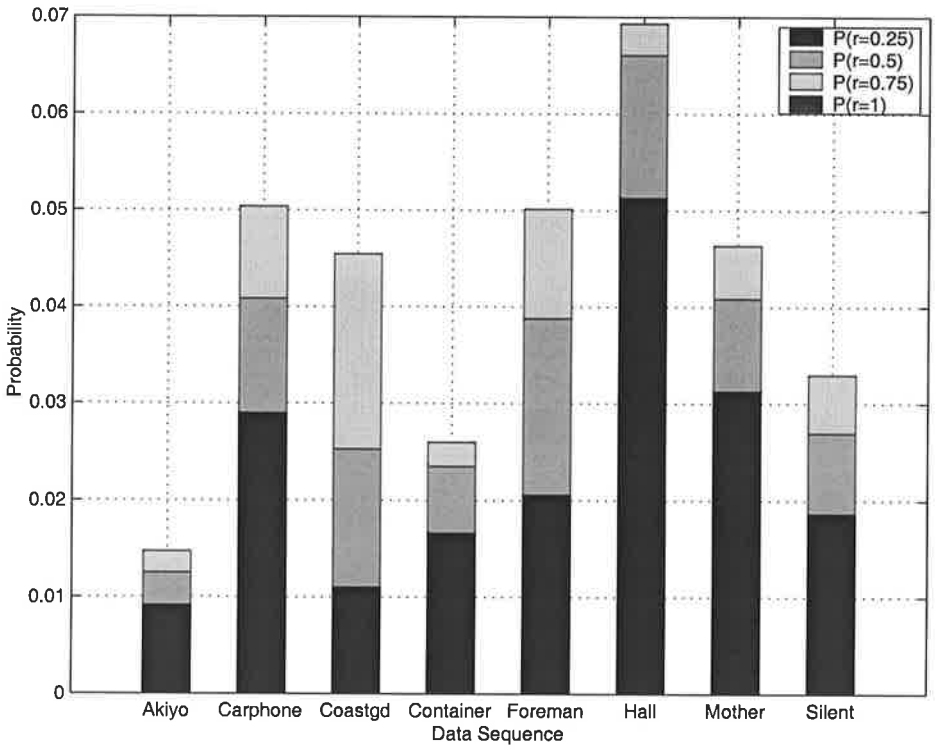


Figure 5.3: Workload distributions at rate quantizations of 0.25, 0.5, 0.75, and 1

processing rate can be altered as shown in the buffering and averaging technique [Gut96] discussed in Section 3.5. Since multimedia computations inevitably involve some form of buffering, we draw motivation from buffering and workload averaging technique by using buffering as a means of decoupling rate from workload such that the desired transformation of workload distribution can be achieved.

Figure 5.5 shows where rate selection approach fits into the overall DVS architecture. As this figure shows, the rate selection block is placed between the input data buffer and the voltage/clock scaling block. In other words, it uses sample workload values in the buffer as an input and outputs a rate value to the voltage/clock scaling block. An alternative way of looking at this is to imagine that the workload averaging function which was used to minimize energy in the continuous voltage model has been replaced by the rate selection function for energy minimization in the voltage quantization model.

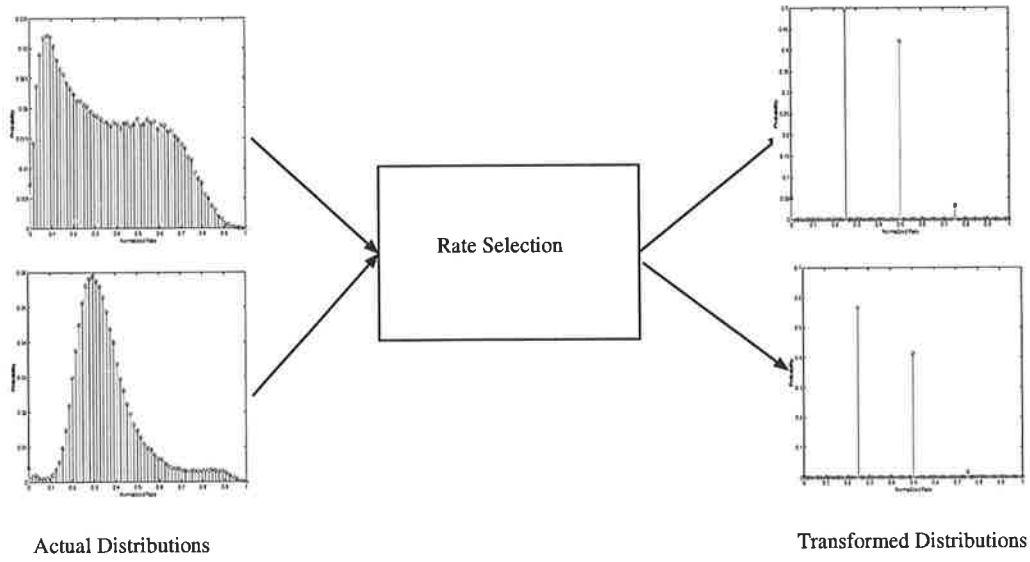


Figure 5.4: Objective of rate selection approach

5.2.2 Concept

The Rate Selection approach involves storage of data samples in a FIFO buffer and determination of a processing rate for each sample period based on the contents of the buffer. Since idle loss is eliminated by selecting rate values that are equal to rate quantizations, the rate selection approach uses total workload in the buffer to where possible select rate values that are equal to rate quantizations. Since the rate selection approach utilizes total buffered workload to select rate values, more than a single sample or part of a single sample can be

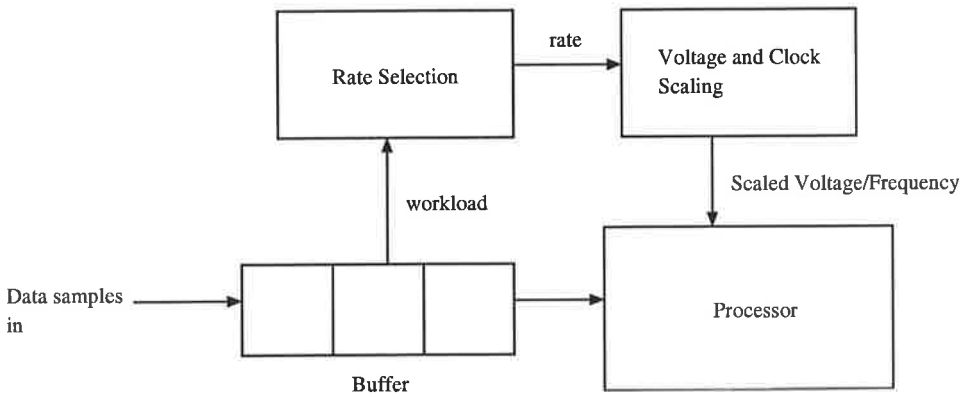


Figure 5.5: Rate selection approach in dynamic voltage scaling

processed during a sample period.

5.2.3 A Simple Rate Selection Algorithm

To demonstrate the concept of the proposed approach, a simple rate selection algorithm is shown below. As in previous sections, we continue to use a 4-level voltage quantization corresponding to normalized rate quantizations of 0.25, 0.5, 0.75, and 1.0 to demonstrate this algorithm. The notation here uses B for buffer length, $w[0], w[1] \ldots w[B-1]$ for sample workloads in buffer locations 1, 2, ..., and B, respectively, wt for total workload in the buffer, r' for an intermediary predicted rate, and r for the selected rate for a sample period.

The algorithm comprises three distinct steps and the operation is described as follows:

Step 1 of the algorithm calculates the total workload in the buffer for a given sample period. This is done by sequentially adding the workload of each buffer location. Thus wt represents the total buffered workload for a given sample period.

Step 2 represents the core of the rate selection algorithm. This step involves prediction of a rate that is equal to a rate quantization depending on the value of total workload calculated in step 1. For example, if the total workload is greater than 1, a predicted rate of 1 is chosen. Similarly for $1 > wt \geq 0.75$, $0.75 > wt \geq 0.5$, and $0.5 > wt \geq 0.25$, the step 2 of the algorithm predicts rate quantizations of 0.75, 0.5, and 0.25 as rate. If wt is less than 0.25, a predicted rate of 0 is assigned.

Step 3 of the algorithm performs the most important buffer overflow protection check against the predicted rate. This step compares the predicted rate with the workload in first buffer location ($w[0]$) to make sure that the predicted rate can at least process the workload $w[0]$. If this is possible, processing rate is set equal to the predicted rate. Consequently, the selected processing rate guarantees that at the very least $w[0]$ is processed, and when the shifting of the data samples occurs at the beginning of the next sample period, no unprocessed data in $w[0]$ is lost due to buffer overflow. If the comparison in step 3 fails, it is implied that the predicted rate is not large enough to process the workload in the first buffer location, and since processing $w[0]$ is important to prevent data loss from buffer overflow, the processing rate is set equal to the workload in the first buffer location.

Based on this algorithm, step 2 always predicts a rate quantization as a predicted rate.

```
Step 1: calculate total workload in the buffer
wt = 0
for (i=0;i < B-1;i++) {
  wt = wt + w[i]
}
Step 2: select an intermediary predicted rate
if (wt >= 1.0) {
  r' = 1
}
else if (wt >= 0.75) {
  r' = 0.75
}
else if (wt >= 0.5) {
  r' = 0.5
}
else if (wt >= 0.25) {
  r' = 0.25
}
else {
  r' = 0
}
Step 3: buffer overflow prevention
if (w[0] <= r') {
  r = r'
}
else {
  r = w[0]
}
```

Figure 5.6: Simple rate selection algorithm

However, step 3 can over-ride the prediction in selecting the processing rate, and this is where rates that are not equal to rate quantizations can be chosen as processing rate.

5.2.4 An Example

This section presents an example that demonstrates the workings of the proposed rate selection algorithm. This example is based on the FMIDCT computation and the same 4-level voltage quantization used in the previous section. For convenience, the workloads of data samples are *not* normalized, hence workloads can range from 0 to 64, corresponding to the non-zero coefficients in a 8×8 block. Consequently, the non-normalized rate quantizations are 16, 32, 48, and 64, corresponding to normalized rate quantizations of 0.25, 0.5, 0.75, and 1. For this example, a buffer size of $B = 3$ is selected. The example is shown in Figure 5.7.

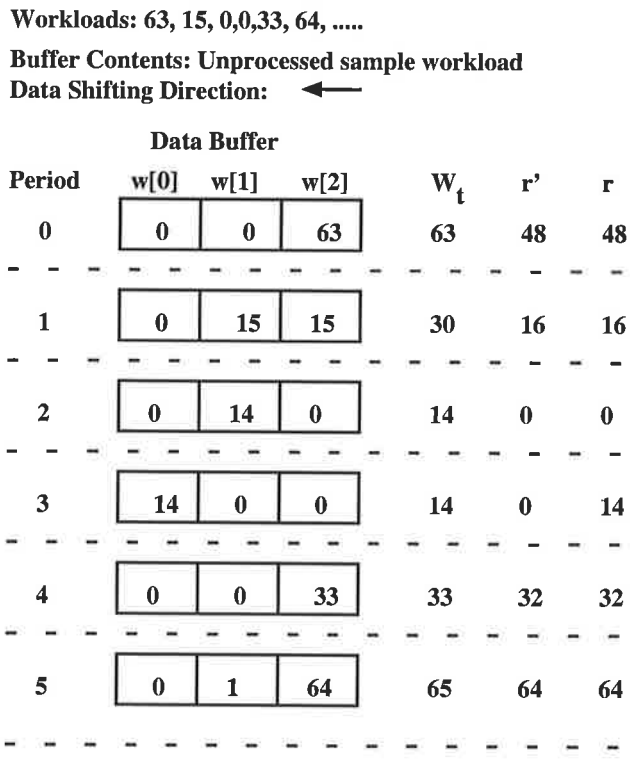


Figure 5.7: An example of simple rate selection algorithm

The operation of the algorithm in this example is as follows:

Before the zeroth sample period in Figure 5.7, the buffer contains no sample workloads. In the zeroth sample period a sample with a workload of 63 is received and buffered at

location $w[2]$. As step 1 of the algorithm shows, the total workload value in the buffer is calculated to be $wt = 63$. Since wt is not ≥ 64, but ≥ 48, the step 2 of the algorithm suggests a predicted rate of $r' = 48$ for the zeroth sample period. Finally step 3 of the algorithm checks for the buffer overflow condition, and in this sample period $w[0] = 0$ and $w[0] < r'$, so the rate $r = r' = 48$ is selected.

In the first sample period the $w[1]$ buffer location contains the unprocessed workload of $w[2]$ location (63-48=15) from the zeroth sample period shifted one buffer location left. The new data sample with workload of 15 received during the first sample period is buffered into location $w[2]$. As in the previous sample period, the remaining of steps of the rate selection algorithm produce $wt = 30$ and the predicted rate of $r' = 16$, and overflow prevention selects r' as the rate or $r = 16$.

The second, fourth, and fifth sample periods operate in the same way as the zeroth and first sample periods. The only difference is the selected rate values.

The third sample period shows the buffer overflow prevention step in action. In this sample period the total workload in the buffer is $wt = 14$, but the predicted rate is $r' = 0$. This implies that the predicted rate is not large enough to process the workload in $w[0]$, and hence the step 3 of the algorithm sets $r = w[0]$. As shown in the previous section, this sample period thus produces a rate that is not a rate quantization.

5.2.5 Operational Details

Similar to the workload averaging approach [Gut96], the rate selection algorithm requires two buffers to maintain the data samples and their corresponding workload values. Considering the FMIDCT computation, Figure 5.8 shows the anatomy of the two buffers. As this figure shows, data samples are stored in the data buffer, while the actual workload of the samples in data buffer are stored in workload buffer. Since in the FMIDCT computation a data sample consists of 64 coefficients, there must be 64 storage locations per buffer location in the data buffer. The workload for a data sample is the number of non zero coefficients and this information is stored in the workload buffer. As Figure 5.8 shows, the workload value per data sample requires only one buffer location per data sample. Since the rate selection algorithm processes data at varying rates (more or less than a complete sample in some sam-

ple periods), it is necessary to track the data sample boundaries. This is achieved using the buffer pointers. As the figure shows, buffer pointers always point to the starting point for processing a data sample. As a new sample gets buffered in, the buffer pointer points to the first value in the data sample (i.e. coefficient 1). However, if a sample is processed over a number of sample periods, buffer pointers are incremented accordingly such that processing will begin only from where a buffer pointer points to in a sample. The weakness of this approach is the need to shift through the entire data sample through the buffer, even though the computation only involves part of a data sample. For example, even though FMIDCT is only performed on non zero coefficients, the full data sample (including zero coefficients) are shifted through the buffer.

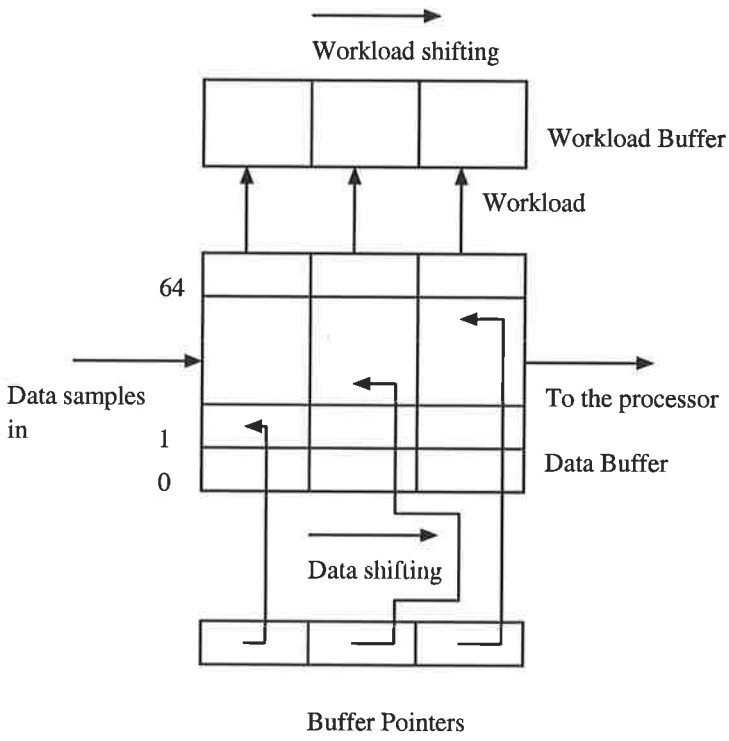


Figure 5.8: Operation of two buffers [Gut96]

An alternative approach to buffer management has been proposed in [XCSD96]. This approach uses a single buffer and stores only the part of a data sample necessary for processing. This technique is called run length coding and is shown in Figure 5.9. Using this technique, the data samples for FMIDCT computation will only store non-zero coefficients in the data

buffer. However, each coefficient is annotated with their position in the data sample (a value between 0 to 63 using an index that starts from 0). The added benefit of this technique is that it requires no workload buffer. This is because, the workload value can be simply determined based on the number of coefficients present in each buffer location. As coefficients are processed, they are removed from the buffer, so that the buffer always contains the unprocessed data.

In our work we assume the run length coding technique for buffer management.

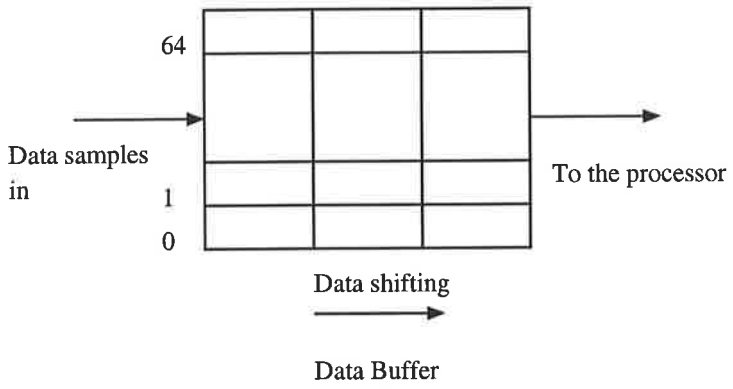


Figure 5.9: Operation of one buffer [XCSD96]

5.2.6 Computational Overhead

Apart from the buffer overhead, the rate selection approach also introduces a computational overhead. The evaluation of rate selection algorithm shows that it requires a constant number $(B-1)$ of *add* operations and a variable number of *compare* operations (n_c) per sample period. n_c is such that $2 \leq n_c \leq 5$. Assuming that the overhead of all these types of operations are equal, the total number of operations (n_t) for a sample period is $B+1 \leq n_t \leq B+4$. Consequently, the computational overhead can be reduced by decreasing the buffer size.

5.2.7 Results

Figures 5.10, and 5.11 show the transformation of rate distributions of MPEG-2 test video sequences (in Figures 5.1, and 5.2) when the rate selection algorithm is used. The data

for this figure is based on a buffer size of 6. A buffer size of 6 is chosen because our data samples being 8×8 blocks of MPEG-2 video at 4:2:0 sub-sampling format requires a macroblock to contain 6 8×8 data samples. In other words, our buffer size is based on the MPEG-2 macroblock size. As this figure shows, the algorithm produces a significant change in selection of rate quantizations. To more effectively evaluate rate quantization selection, Figure 5.12 shows the distribution at *all* rate quantizations, and 5.13 shows the actual change in distribution at all rate quantizations due to the use of the rate selection algorithm. As these figures show (particularly the latter figure), the rate selection algorithm has effectively increased the selection of rate quantizations as rate. For data sequences such as Hall, the change is over 85% and for sequences such as Akiyo, the change is just over 15%.

Even though the rate selection algorithm is very efficient in selecting rate quantizations, the total energy of computation is the more important metric which ultimately decides whether the rate selection algorithm is successful or unsuccessful for minimizing energy consumption. Thus, using the energy model discussed in Section 4.4.1, the comparisons of data processing energy, transition energy, and transition count for the MPEG-2 test video sequences, with and without rate selection algorithm are shown in Figure 5.14. As this figure shows, the rate selection approach has significantly reduced the data processing energy consumption, and the magnitude of energy savings range from 33% for Carphone and Foreman, to 41% for Akiyo video sequence. As for the transition energy, there has been a reduction in seven out of 8 sequences, and the magnitude of savings ranged from 2% (Silent) to 27% (Hall). For Foreman sequence however, the transition energy has increased by 11%. The transition counts have also been reduced for all the test video sequences. More specifically, the reductions range from 31% in Coastguard, Foreman, and Silent, to 65% in Akiyo and Mother. This analysis also used a buffer size of 6.

5.3 Summary

Using MPEG-2 test video sequences, this chapter demonstrated that the vast majority of data samples in typical video data sequences contribute to idle losses when the voltage quantization model is used. In order to minimize the idle losses and minimize the total energy of computation, a novel approach called rate selection is then introduced. This approach uses

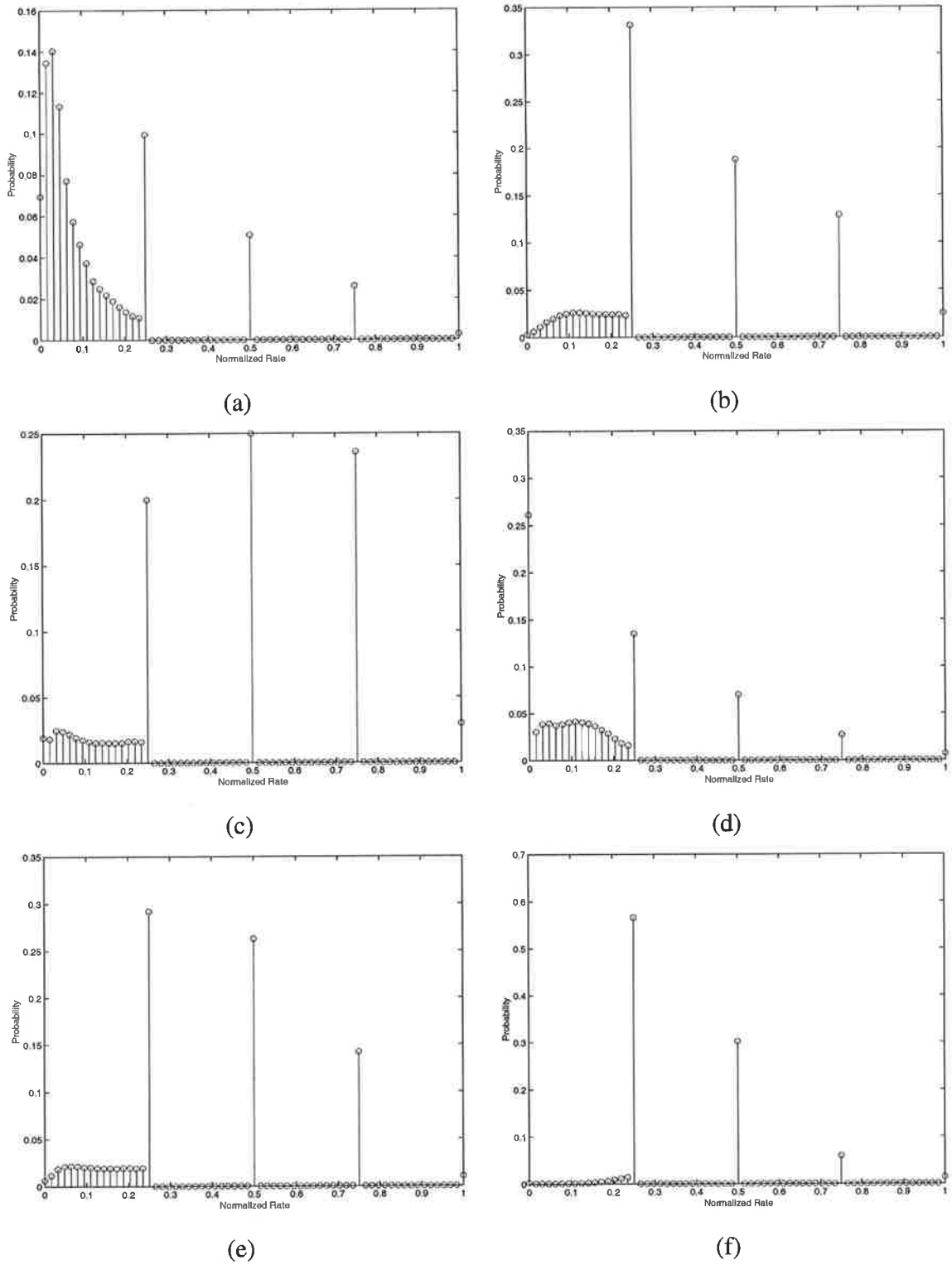
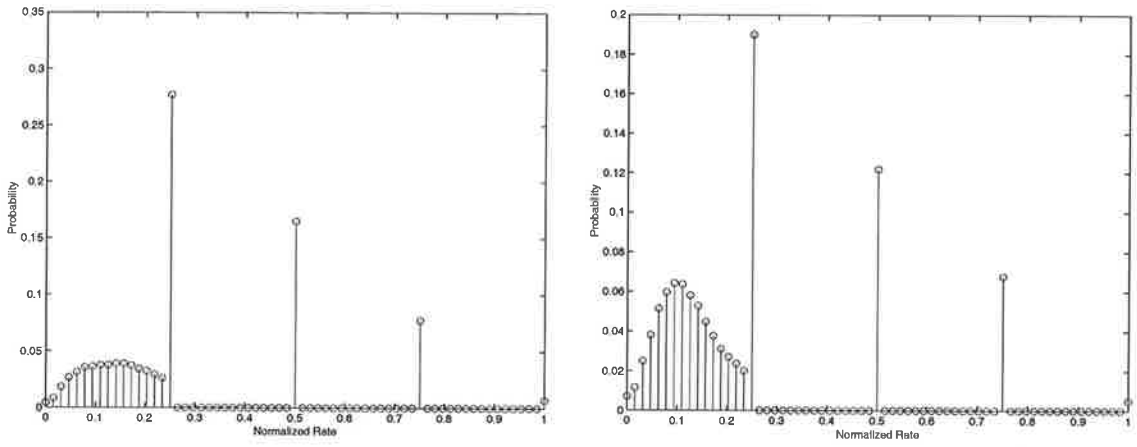


Figure 5.10: Rate distributions of MPEG-2 test video sequences for rate selection algorithm. (a) Akiyo, (b) Carphone, (c) Coastguard, (d) Container (e) Foreman, (f) Hall



(g) (h)

Figure 5.11: Rate distributions of MPEG-2 test video sequences for rate selection algorithm. (g) Mother, and (h) Silent

buffering to decouple rate from sample workloads and where possible selects rate quantizations as the rate. This chapter then presented a simple rate selection algorithm and its effect on energy minimization. The results demonstrate that the rate selection algorithm is very efficient in selecting rate quantizations and also in achieving significant energy savings. However, there are still significant numbers of samples processed at non-quantized rates. The next chapter presents a number of enhancements to further improve energy efficiency and reduce the computational overhead of the rate selection approach.

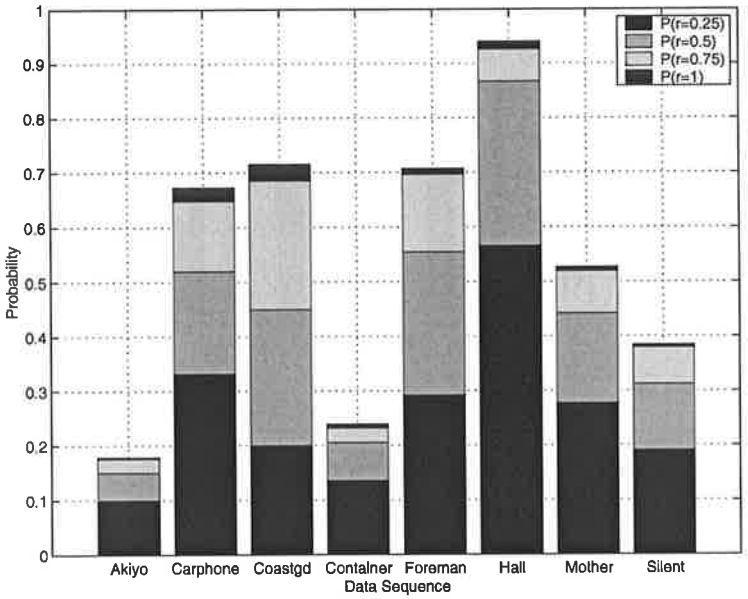


Figure 5.12: Rate quantization distributions for rate selection algorithm.

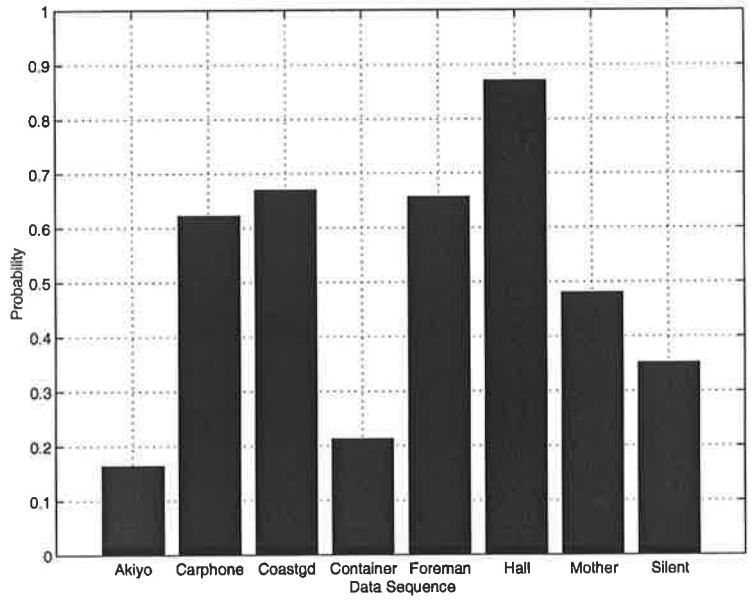
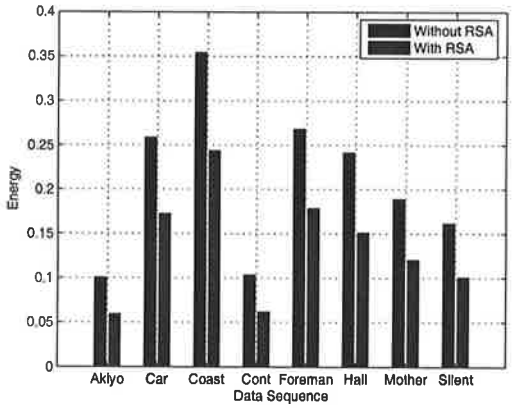
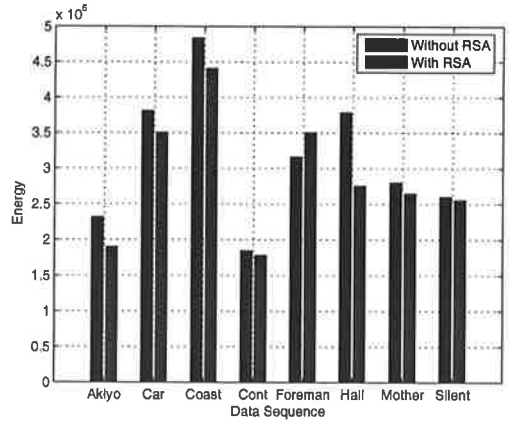


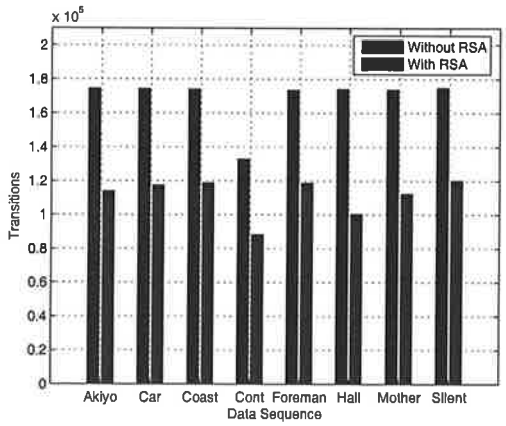
Figure 5.13: Rate quantization distribution change due to rate selection algorithm.



(a)



(b)



(c)

Figure 5.14: Energy costs and transition count for rate selection approach. (a) Data processing energy cost E_{pr}, (b) Transition energy cost E_{tr}, (c) transition count

Chapter 6

Enhancements to Rate Selection Approach

The primary objectives of this chapter are to improve the energy efficiency of the simple rate selection algorithm presented in the preceding chapter and reduce the computational overhead. In order to improve the energy efficiency of the algorithm, the rate selection policy is improved to recognize the relative energy costs of rate quantizations. Moreover, since the idle condition (when $r = 0$) provides the best energy efficiency for dynamic voltage and frequency scaling in fixed throughput mode of computation, the rate selection policy is also improved to include idle rate selection. As for computational overhead reduction, the repetitive addition in step 1 of the algorithm is replaced with a low overhead operation. Finally, the overall effectiveness of the rate selection approach is evaluated.

6.1 Introduction

6.1.1 Relative Energy Costs of Rate Quantizations

Figure 6.1 shows the normalized energy costs of processing a data sample at different rate quantizations. Based on this figure, the relative energy cost of processing a single data sample at the highest rate quantization and the number of data samples that consume the same energy at lower rate quantizations are shown in Table 6.1. As this table shows, processing a single sample at the highest rate quantization ($r = 1$) is the least energy efficient. This is

because processing a data sample at the highest rate quantization implies that the processing occurs at the highest V_{dd} or V_{max}, and since there is no opportunity for voltage scaling, the energy consumption is at a maximum. More importantly, Table 6.1 shows that one sample processed at the highest voltage quantization consumes the same energy of approximately 17 samples processed at the lowest voltage quantization. Based on these results, a rate selection strategy that gives higher priority to smaller rate quantizations can improve the efficiency of the rate selection approach.

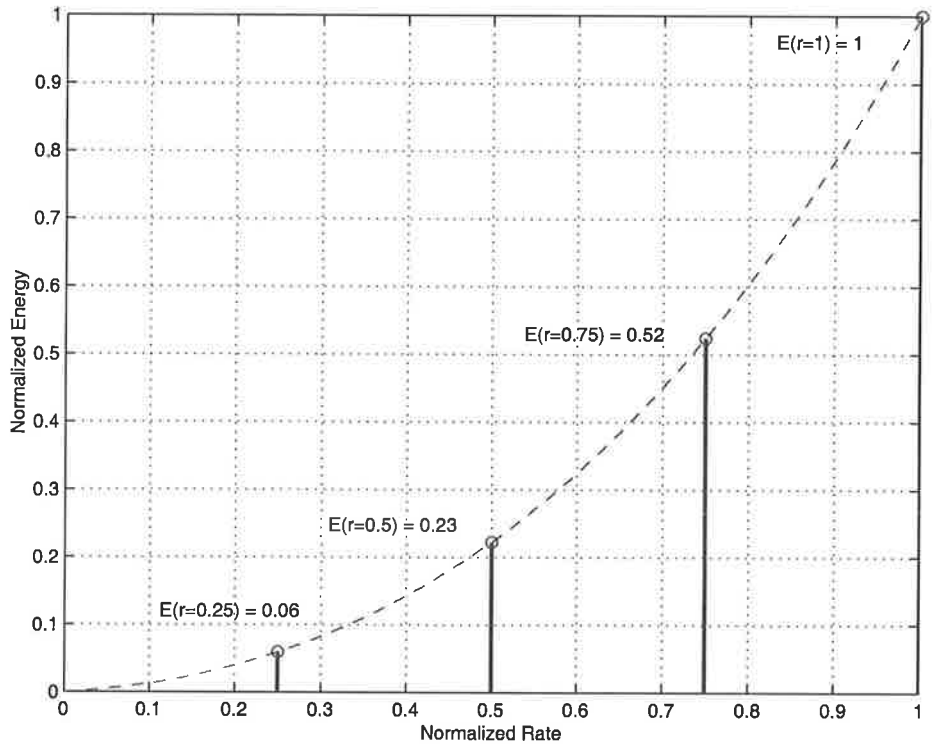


Figure 6.1: Relative energy costs at different voltage quantizations

6.1.2 Possible Enhancements

Even though the simple rate selection algorithm presented in the previous chapter is very simple, it has a very serious limitation: it gives higher priority to more energy costly higher rate quantizations during rate selection. Thus, further energy savings from the rate selection algorithm can be achieved if rate selection policy is changed. In order to achieve this objective, three methods are explored: 1) increasing selection of smaller rate quantizations, 2) use

Table 6.1: Number of data samples consuming same energy at lower quantizations as one data sample at highest voltage quantization

Voltage Quantization	Number of Data Samples
$r = 0.25$	17
$r = 0.5$	5
$r = 0.75$	2
$r = 1$	1

of an average function to minimize higher rate quantizations, and 3) increasing selection of the idle rate ($r = 0$). Increasing idle rate implies idling at V_{min}, and this in turn results in the lowest energy consuming scenario.

Apart from investigating enhancements for minimizing energy consumption, alternative formulations to reduce the computational complexity of the rate selection algorithm will also be explored.

6.2 Energy Efficient Enhancements

6.2.1 Enhancement 1: Prioritized Selection of Lower Rate Quantizations

In order to give higher priority to smaller rate quantizations during rate selection, it is necessary to reverse the selection policy used in the simple rate selection algorithm. Moreover, it is also very important to make sure that the selected rate quantization is also large enough to process the workload of the first buffer location, so that buffer overflow is prevented. Thus, the key constraints of the prioritization will be to select the smallest rate quantization from the total workload in the buffer that also prevents buffer overflow.

The rate selection algorithm with the prioritization of small rate quantizations selection is shown in Figure 6.2. As in previous sections, we use a 4-level voltage quantization corresponding to normalized rate quantizations of 0.25, 0.5, 0.75, and 1.0. The notation here uses B for buffer length, $w[0], w[1] \ldots w[B-1]$ for sample workloads in buffer locations 1,

2, . . ., and B, respectively, wt for total workload in the buffer, and r for the selected rate of a sample period.

```
Step 1: calculate total workload in the buffer
        wt = 0
        for (i=0;i < B-1;i++) {
          wt = wt + w[i]
        }
Step 2: select the rate quantization
        if (wt >= 0.25 & w[0] <= 0.25) {
          r = 0.25
        }
        else if (wt >= 0.5 & w[0] <= 0.5) {
          r = 0.5
        }
        else if (wt >= 0.75 & w[0] <= 0.75) {
          r = 0.75
        }
        else if (wt >= 1) {
          r = 1
        }
        else {
          r = wt
        }
```

Figure 6.2: Rate selection algorithm with enhancement 1

Since the proposed prioritization of rate selection occurs in the second step of the algorithm, the first step that calculates the total workload in the buffer is the same as in the simple rate selection algorithm. In the second step, the comparison with rate quantizations are done in ascending order (0.25, 0.5, 0.75, and 1.0), and this gives the possibility of selecting a smaller rate quantization as rate. Moreover, buffer overflow protection is now merged into

the rate selection step (2), making it possible to select the smallest rate quantization that also processes the workload in the first buffer location. If no rate quantization is selected, the total workload is selected as rate. In this algorithm buffer overflow protection is guaranteed since $r \geq w[0]$ is maintained for all comparisons, because $wt \geq r \geq w[0]$.

An example that demonstrates the workings of the proposed prioritization of rate selection algorithm is given in Figure 6.3. As before, this example is based on the FMIDCT computation and the same 4-level voltage quantization. For convenience, the workloads of data samples are *not* normalized, hence workloads can range from 0 to 64, corresponding to the non-zero coefficients in a 8×8 block. Consequently, the non-normalized rate quantizations are 16, 32, 48, and 64, corresponding to normalized rate quantizations of 0.25, 0.5, 0.75, and 1. For this example, a buffer size of $B = 3$ is selected. The example is shown in Figure 6.3.

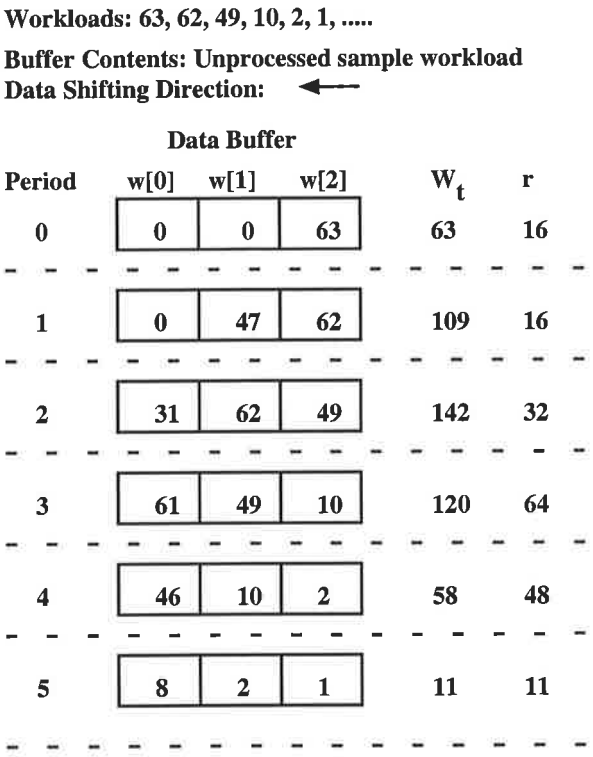


Figure 6.3: An example of rate selection algorithm with enhancement 1

The operation of the algorithm in this example is as follows:

Before the zeroth sample period in Figure 6.3 the buffer contains no sample workloads.

In the zeroth sample period a sample with a workload of 63 arrives and is buffered at $w[2]$. During this sample period the total buffered workload is $w_t = 63$. Since $w_t \geq 16$ and $w[0] \leq 16$ the selected rate for the zeroth sample period is $r = 16$.

In the first sample period $w[1]$ contains the unprocessed workload from $w[2]$ of (63-16=47) from the zeroth sample period shifted one buffer location left. The new sample buffered during this sample period has a workload of 62 and is found in $w[2]$. The w_t calculation results in $w_t = 109$, and hence a rate value of $r = 16$ is selected for the first sample period.

The second, third, and fourth sample periods operate in the same way as the zeroth and first sample periods. The only difference is the selected rate value in each sample period.

The fifth sample period shows that the *else* part of the algorithm becomes *TRUE*. In this period the total workload in the buffer $w_t = 11$ is less than the first rate quantization of 16. In this case the entire buffered workload is processed during this sample period.

The change in computational complexity due to the prioritization enhancement is insignificant. The first step of the algorithm is unchanged; however, merging of step two and three of the rate selection algorithm has added slightly to the complexity of the algorithm. Quantitatively, the prioritized rate selection algorithm requires a total of $B+2 \leq n_t \leq B+9$ operations. This is based on a constant number $(B-1)$ of *add* operations and a variable number of *compare* operations (n_c) and *and* operations (n_a) per sample period, where n_c and n_a are such that $2 \leq n_c \leq 7$ and $1 \leq n_a \leq 3$. As before, the computational overhead can be reduced by decreasing the buffer size.

Figures 6.4 and 6.5 show the rate distributions of MPEG-2 test video sequences when rate selection algorithm with enhancement 1 (prioritization) is used. As before, this data is also based on a buffer size of 6. Based on this data, Figure 6.6 shows rate quantization distributions. Furthermore, the change in rate distributions due to enhancement 1 is shown in Figure 6.7. As this figure shows, the prioritization of the rate selection has indeed improved the rate quantizations selection. For example, the Hall and Coastguard sequences have achieved over 6% and 25% improvements in selection efficiency, respectively. Finally the more important comparisons of energy and transition count are shown in Figure 6.8. As this figure shows, prioritization reduces data processing energy in 5 out of the 8 video sequences, and the savings range from 3% for Hall and Mother, to 15% for Coastguard.

However, for Akiyo, Container, and Silent sequences, the enhancement 1 has increased the data processing energy. As for the transition energy, all sequences have achieved a energy reduction due to prioritization, and the magnitude of savings range from 14% for Akiyo to 42% for Silent. Finally, prioritization has resulted in 6 out of 8 sequences to experience a reduction in voltage transitions, and the magnitude of reduction ranges from 10% for Silent to 36% for Foreman. As for Akiyo and Container, the transition count has increased by 22% and 2%, respectively.

In summary, the results of our analysis show that prioritization improves rate quantization selection and this in turn reduces total energy consumption. The next section discusses another enhancement that further helps to minimize energy efficiency of the rate selection approach by reducing the more energy costly higher rate quantizations.

6.2.2 Enhancement 2: Reduced Selection of Higher Rate Quantizations

In addition to using prioritization in rate selection, this section explores the use of an averaging function as a method of reducing the more energy costly higher rate quantizations in the rate selection algorithm. The characteristic property of the averaging function is its ability to minimize extreme values in a data distribution. We use this property for improving the selection policy of the rate selection algorithm such that the more energy costly rate quantizations are further reduced.

The rate selection algorithm with the averaging function is shown in Figure 6.9. As in previous sections, we continue to use a 4-level voltage quantization corresponding to normalized rate quantizations of 0.25, 0.5, 0.75, and 1.0. The notation here uses B for buffer length, $w[0], w[1] \ldots w[B-1]$ for sample workloads in buffer locations 1, 2, ..., and B, respectively, wt for total workload in the buffer, ra for the average workload, and r for the selected rate for a sample period.

Compared to the algorithm discussed in the previous section, this algorithm includes two main changes: 1) step 2 has become a new step that calculates the average workload in the buffer, and 2) the selection policy in step 3 now includes average rate as part of the selection conditions. This leaves the step 1 unchanged from the previous version. As before, the rate selection in this algorithm involves choosing the smallest rate quantization that not only

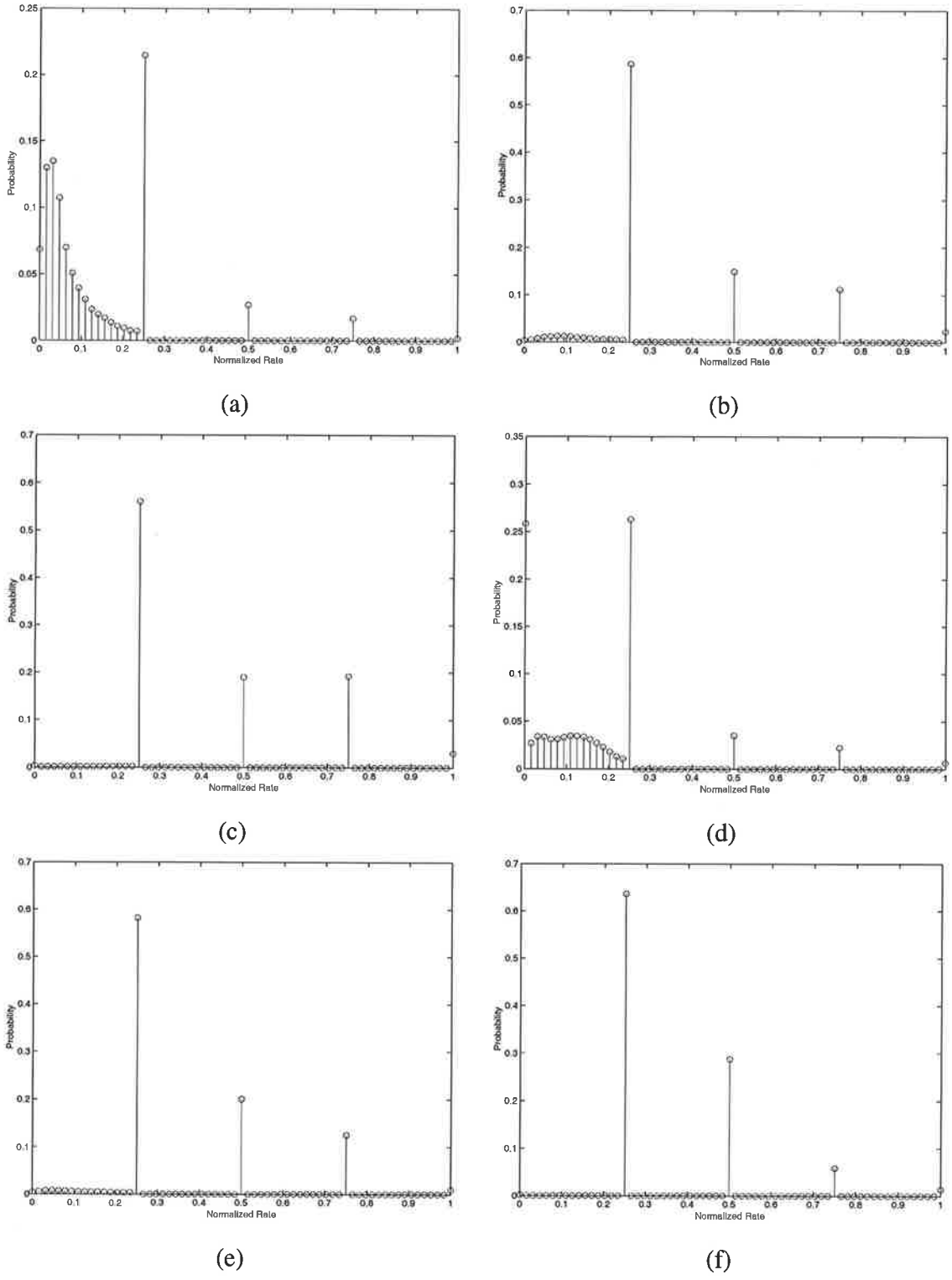
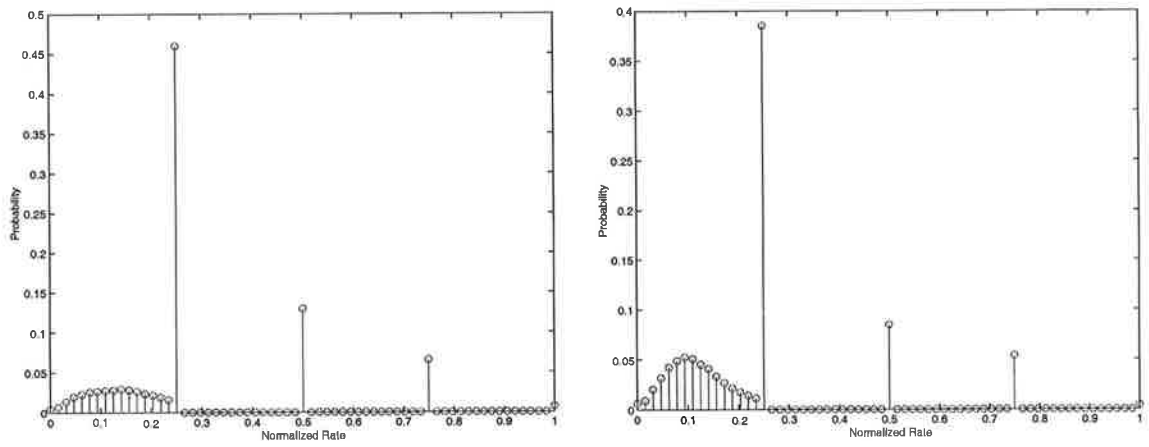


Figure 6.4: Rate distributions of MPEG-2 test video sequences for the rate selection algorithm with enhancement 1. (a) Akiyo, (b) Carphone, (c) Coastguard, (d) Container (e) Foreman, (f) Hall



(g) (h)

Figure 6.5: Rate distributions of MPEG-2 test video sequences for the rate selection algorithm with enhancement 1. (g) Mother, and (h) Silent

processes the workload in the first buffer location, but also uses average workload to make sure that the rate selection minimizes selection of higher rate quantization when the buffer is overwhelmed by a sudden queuing of large workloads. If no rate quantization is selected, the total workload is selected as the rate. In this algorithm buffer overflow protection is guaranteed since $wt \geq r \geq w[0]$ is maintained for all comparisons.

An example that demonstrates the workings of the proposed rate selection algorithm with averaging function is given in Figure 6.10. As before, this example is based on the FMIDCT computation and the same 4-level voltage quantization. For convenience, the workloads of data samples are *not* normalized, hence workloads can range from 0 to 64, corresponding to the non-zero coefficients in a 8×8 block. Consequently, the non-normalized rate quantizations are 16, 32, 48, and 64, corresponding to normalized rate quantizations of 0.25, 0.5, 0.75, and 1. For this example, a buffer size of $B = 3$ is selected. The example is shown in Figure 6.10.

The operation of the algorithm in this example is as follows:

Before the zeroth sample period in Figure 6.10 the buffer contains no sample workloads. In the zeroth sample period a sample with a workload of 15 arrives and is buffered at $w[2]$. During this sample period the total buffered workload is $w_t = 15$ and the average buffered workload is $ra = 5$. Since w_t is less than the smallest quantized rate (16), all the *if* conditions

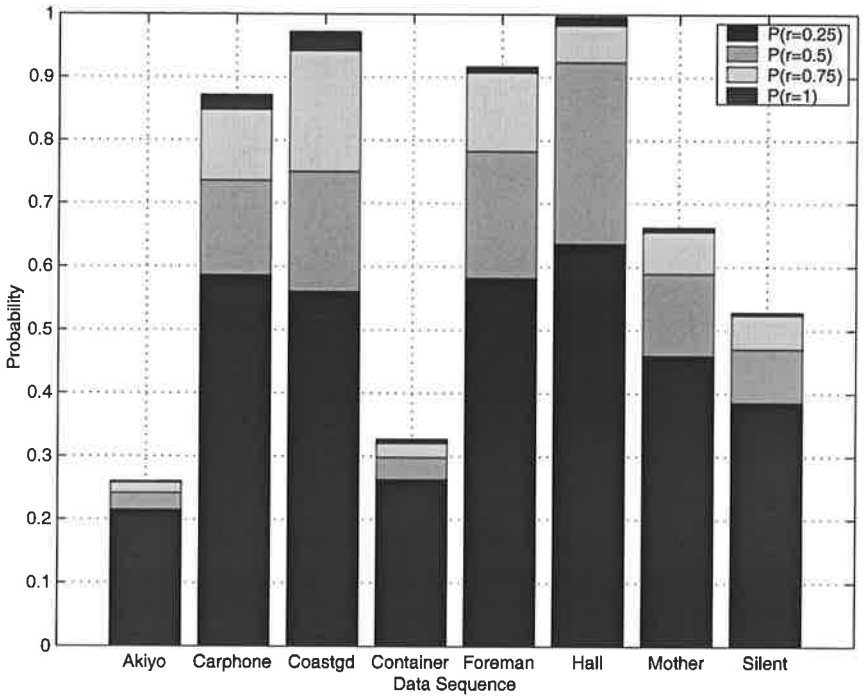


Figure 6.6: Rate quantization distributions for the rate selection algorithm with enhancement 1

in the algorithm are evaluated to *FALSE* and the *else* part of the algorithm becomes *TRUE*, setting the rate $r = w_t = 15$ for the zeroth sample period.

In the first sample period the new workload 16 gets buffered into the location $w[2]$, and the total buffered workload $wt = 16$. The integer portion of the average rate is $ra = 5$, and the first *if* condition in the algorithm becomes *TRUE*, selecting the rate of $r = 16$ for the first sample period. In the second sample period the $wt = 63$ and $ra = 21$ and hence a rate of $r = 32$ is selected. Similarly the third, fourth, and fifth periods produce rate values of 32, 48, and 48, respectively.

The additional computational complexity involved with adding the averaging function to the rate selection algorithm is very small. This is because the first step of the algorithm is unchanged, and only the average workload calculation and additional comparisons contribute to increase in overhead. More specifically, the introduction of the averaging function incurs a total of $B + 5 \leq n_t \leq B + 16$ operations. This is based on a constant number $(B - 1)$ of *add* operations, a single *divide* operation, and variable numbers of *compare* operations (n_c)

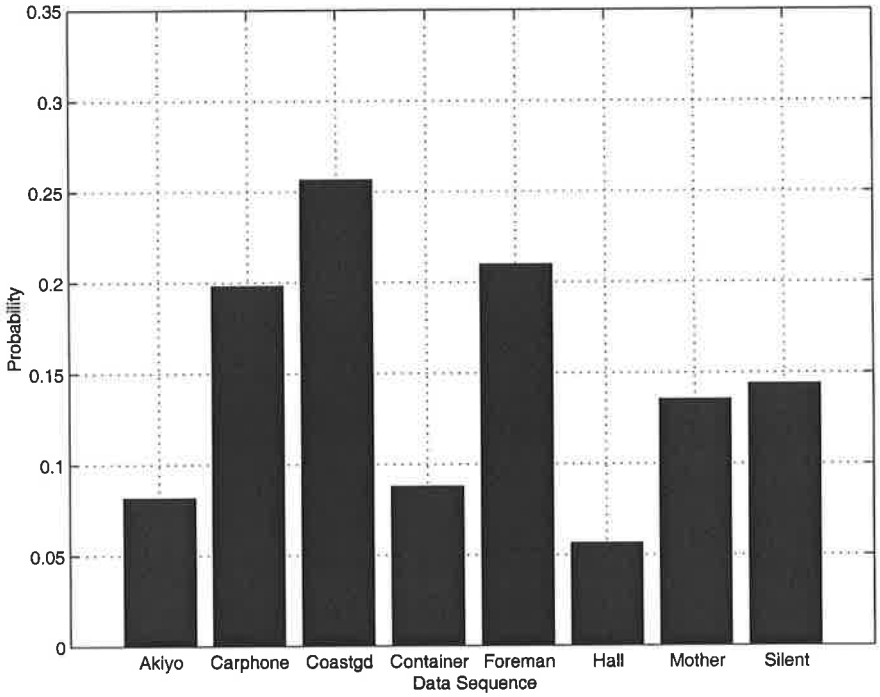
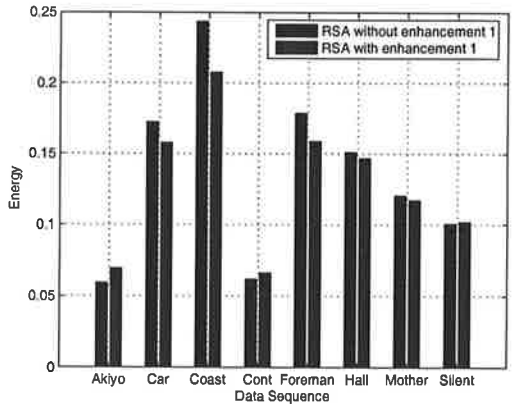


Figure 6.7: Rate quantization distribution change due to enhancement 1 in the rate selection algorithm

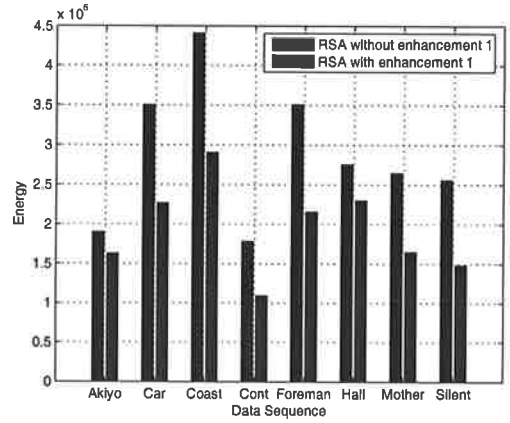
and *and* operations (n_a) per sample period, where n_c and n_a are such that $3 \leq n_c \leq 10$ and $2 \leq n_a \leq 6$. As before, the computational overhead can be reduced by decreasing the buffer size.

As in the previous section, Figures 6.11,6.12,6.13,6.14, and 6.15 show the rate distributions of MPEG2 test video sequences when the rate selection algorithm with enhancements 1 and 2 is used, rate quantization distributions, change in rate quantization distributions, and the comparison of energy and transition count. As these figures (in particular Figure 6.14) show, the averaging function has not significantly changed the total rate quantization selection but has reduced higher rate quantizations to achieve energy savings and reduced transition counts for all test data sequences. More specifically, the data processing energy savings ranged from 4% for Akiyo to 16% for Coastguard sequence. As for transition energy, the savings ranged from 11% for Akiyo to 64% for Coastguard sequence. Finally, the transition count reduction ranged from 2% for Akiyo to 44% for Coastguard sequence.

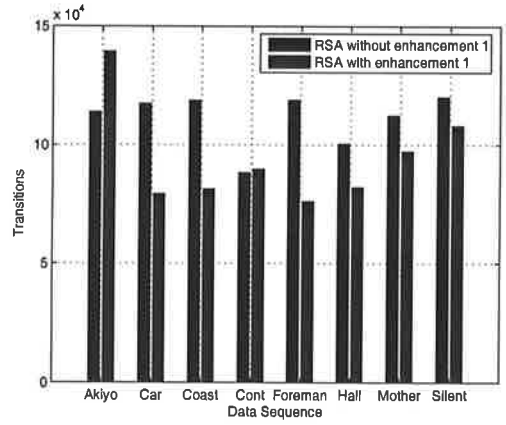
In summary, the results of our analysis shows that the averaging function in fact reduces



(a)



(b)



(c)

Figure 6.8: Energy costs and transition count for rate selection approach with enhancement 1. (a) Data processing energy cost E_{pr}, (b) Transition energy cost E_{tr}, (c) transition count

```
Step 1: calculate total workload in the buffer
        wt = 0
        for (i=0;i < B-1;i++) {
          wt = wt + w[i]
        }

Step 2: calculate the average workload in the buffer
        ra = wt/B

Step 3: select the rate quantization
        if (wt >= 0.25 & w[0] <= 0.25 & ra <= 0.25) {
          r = 0.25
        }
        else if (wt >= 0.5 & w[0] <= 0.5 & ra <= 0.5)) {
          r = 0.5
        }
        else if (wt >= 0.75 & w[0] <= 0.75 & ra <= 0.75)) {
          r = 0.75
        }
        else if (wt >= 1) {
          r = 1
        }
        else {
          r = wt
        }
```

Figure 6.9: Rate selection algorithm with enhancements 1 and 2

Workloads: 15, 16, 63, 62, 62, 64, ...
Buffer Contents: Unprocessed Work
Data Shifting Direction:←

	Data Buffer						
Period	w[0]	w[1]	w[2]	Wt	ra	w[0]	r
0	0	0	15	15	5	0	15
1	0	0	16	16	5	0	16
2	0	0	63	63	21	0	32
3	0	31	62	93	31	0	32
4	0	61	62	123	41	0	48
5	13	62	64	139	46	13	48

Figure 6.10: An example of the rate selection algorithm with enhancements 1 and 2

the more energy costly higher rate quantizations during rate selection. Consequently, such selection results in reduced energy of computation for all the test video sequences.

6.2.3 Enhancement 3: Prioritized Selection of Idle Rate

The two enhancements discussed above involved minimization of total energy of computation through reduction of the more energy costly higher rate quantizations during rate selection. Though these enhancements reduce high rate quantizations, the algorithm selects some non rate quantizations as the rate. This is evident particularly in the $0 < r < 0.25$ ranges of a number of test video sequences. This section presents an enhancement to the rate selection algorithm that gives higher priority to the *idle rate* during rate selection and aims to minimize the non rate quantization selection in the $0 < r < 0.25$ range.

Idle rate implies that the processing rate for a given sample period is zero ($r = 0$). This condition is very important to dynamic voltage scaling because whenever a $r = 0$ condition

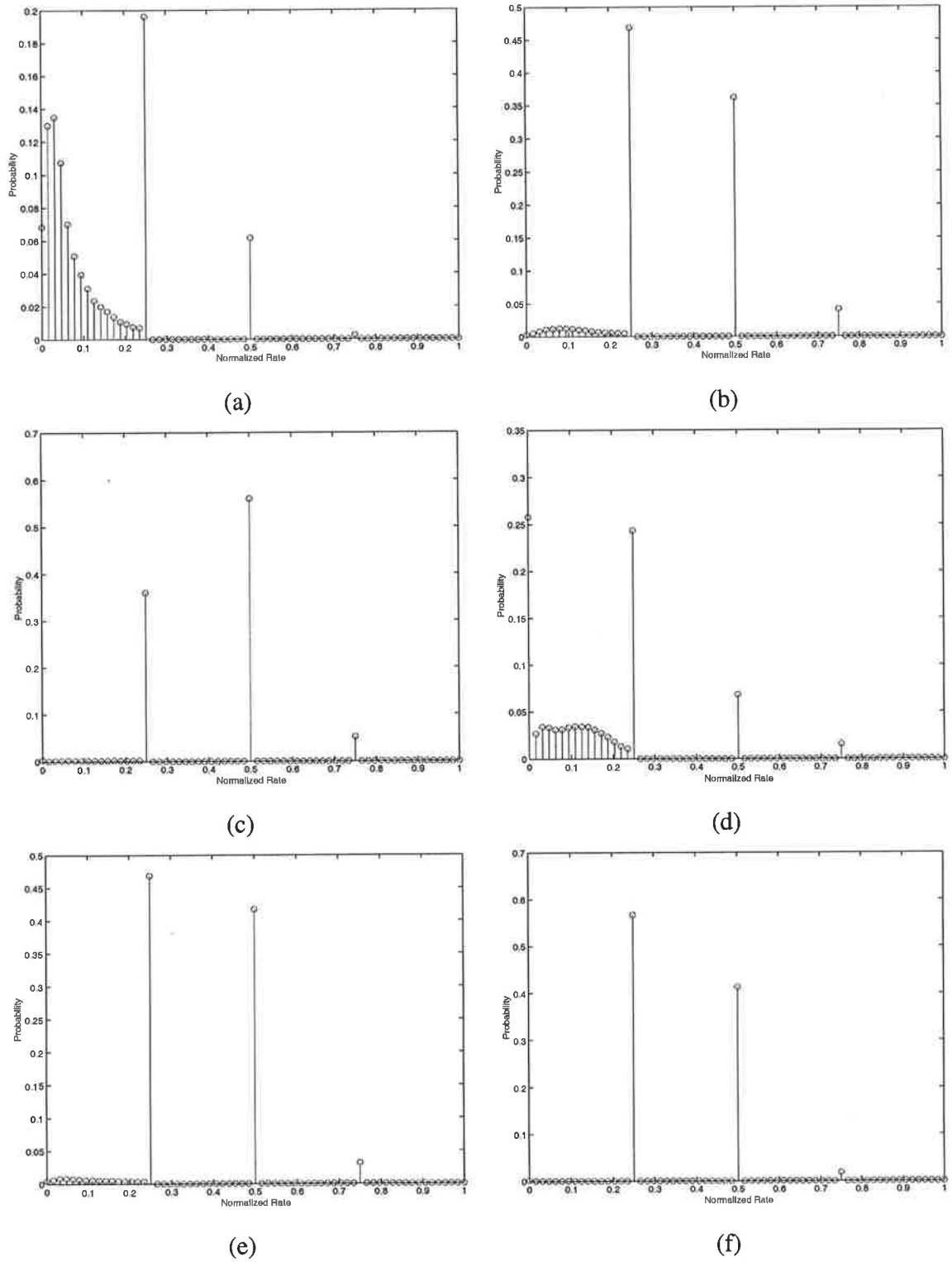
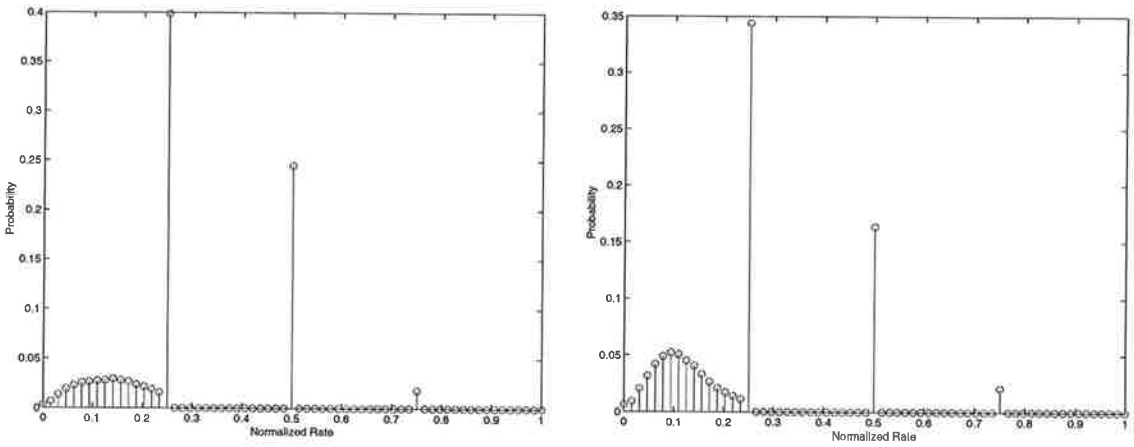


Figure 6.11: Rate distributions of MPEG-2 test video sequences for the rate selection algorithm with enhancements 1 and 2. (a) Akiyo, (b) Carphone, (c) Coastguard, (d) Container (e) Foreman, (f) Hall



(g) (h)

Figure 6.12: Rate distributions of MPEG-2 test video sequences for the rate selection algorithm with enhancements 1 and 2. (g) Mother, and (h) Silent

occurs, the supply voltage is scaled down to the lowest possible voltage (V_{min}), and the processor idles. In terms of energy consumption, the idle rate represents the highest energy saving achievable for a sample period with dynamic voltage scaling.

The rate selection algorithm incorporating the idle rate selection step is shown in Figure 6.16. As in previous sections, we continue to use a 4-level voltage quantization corresponding to normalized rate quantizations of 0.25, 0.5, 0.75, and 1.0. The notation here uses B for buffer length, $w[0], w[1] \ldots w[B-1]$ for sample workloads in buffer locations 1, 2, ..., and B, respectively, wt for total workload in the buffer, ra for the average workload, and r for the selected rate for a sample period.

In this algorithm, the portion corresponding to idle rate selection can be found above the last *ELSE* condition in step 3. As this condition shows, idle rate ($r = 0$) is only assigned when the $w[0]$ has a workload of zero and when the total workload in the buffer is less than the first rate quantization. In other words, idle rate is only chosen when the first buffer location has a workload of 0 and the total workload is very small. Moreover, leaving idle selection to the end of the algorithm gives rate quantization selection higher priority than idle rate selection. This priority order is important to minimize the more energy costly higher rate quantizations. Since the addition of idle rate selection into the rate selection algorithm continues to guarantee $wt \geq r \geq w[0]$, buffer overflow protection is preserved.

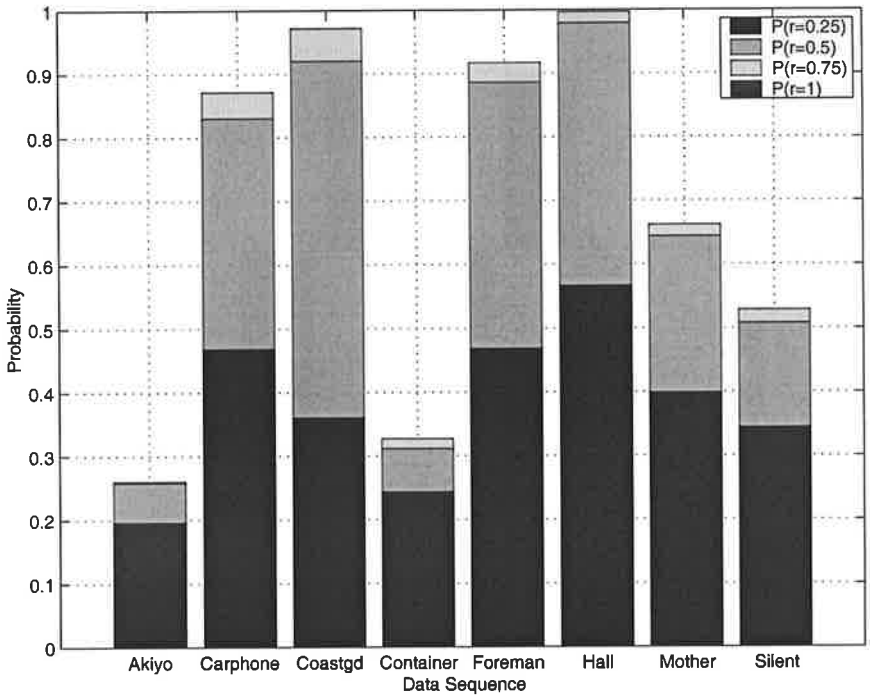


Figure 6.13: Rate quantization distributions for the rate selection algorithm with enhancements 1 and 2

An example that demonstrates the workings of the rate selection algorithm with idle rate selection is shown in Figure 6.17. As before, this example is based on the FMIDCT computation and the same 4-level voltage quantization. For convenience, the workloads of data samples are *not* normalized, hence workloads can range from 0 to 64, corresponding to the non-zero coefficients in a 8×8 block. Consequently, the non-normalized rate quantizations are 16, 32, 48, and 64, corresponding to normalized rate quantizations of 0.25, 0.5, 0.75, and 1. For this example, a buffer size of $B = 3$ is selected. The example is shown in Figure 6.17.

The operation of the algorithm in this example is as follows:

Before the zeroth sample period in Figure 6.17 the buffer contains no sample workloads. In the zeroth sample period a sample with a workload of 15 arrives and is buffered at $w[2]$. During this sample period the total buffered workload is $w_t = 15$ and the average buffered workload is $ra = 5$. Since w_t is less than the smallest quantized rate (16), all the *if* conditions in the algorithm are evaluated to *FALSE* except for the last one that checks for idle rate. Since $w[0] = 0$ and $wt < 16$, the rate selected for this sample period is $r = 0$.

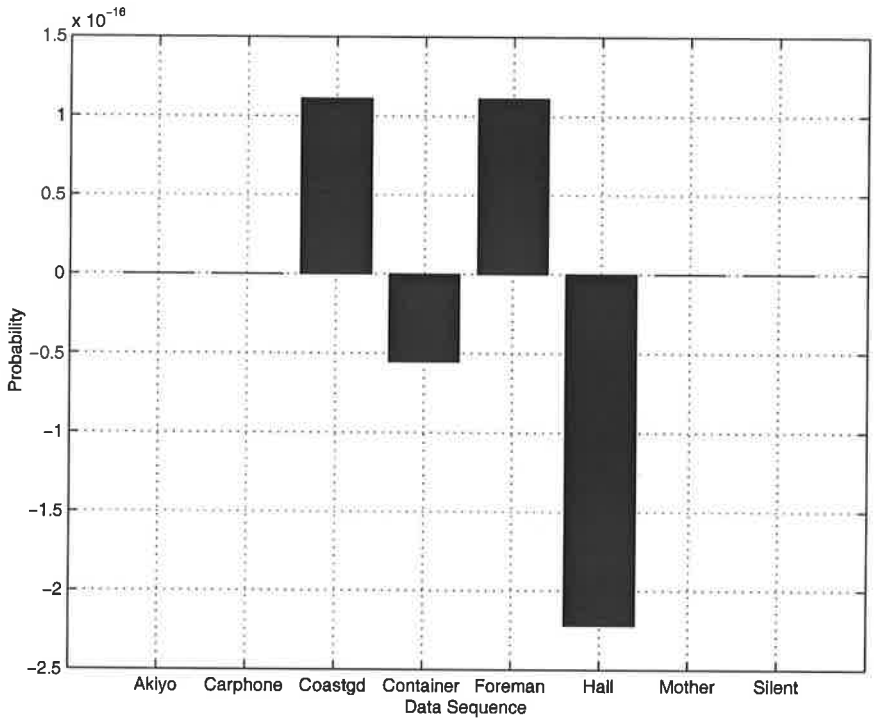
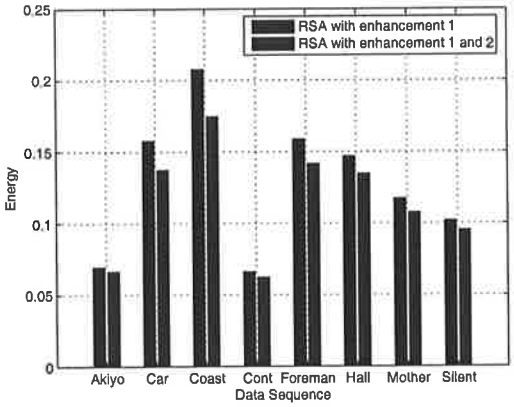


Figure 6.14: Rate quantization distribution change due to enhancements 1 and 2 in the rate selection algorithm

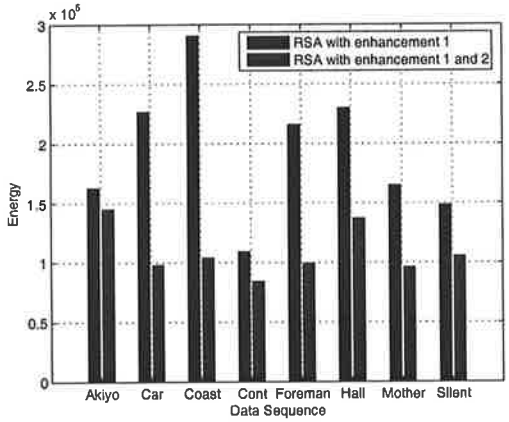
In the first sample period the new workload 16 gets buffered into the location $w[2]$, and the total buffered workload $wt = 31$. The integer portion of the average workload is $ra = 10$, and the first *if* condition in the algorithm becomes *TRUE*, selecting the rate of $r = 16$ for the first sample period. In the second sample period the $wt = 78$ and $ra - 26$ and hence a rate of $r = 32$ is selected. Similarly the third, fourth, and fifth periods produce rate values of 48.

The additional computational complexity involved with the idle rate selection is very small. In terms of the number of operations, the algorithm modification only added two extra operations (one *compare* and one *and*) to the worst case overhead. Since the first two steps are virtually unchanged, the total number of operations involved with the algorithm is $B + 5 \leq n_t \leq B + 18$, where B is the buffer size.

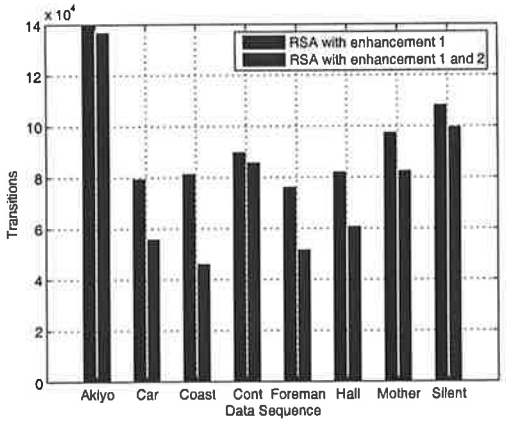
Figures 6.18 and 6.19 show the rate distributions of MPEG-2 test video sequences when the rate selection algorithm with idle rate selection is used. Based on this data, the rate quantizations and idle rate selection is shown in Figure 6.20. As this figure shows, the algorithm is very effective in selecting a rate quantization or idle rate as the rate. Moreover,



(a)



(b)



(c)

Figure 6.15: Energy costs and transition count for rate selection approach with enhancement 2. (a) Data processing energy cost E_{pr}, (b) Transition energy cost E_{tr}, (c) transition count

```
Step 1: calculate total workload in the buffer
        wt = 0
        for (i=0;i < B-1;i++) {
          wt = wt + w[i]
        }

Step 2: calculate the average workload in the buffer
        ra = wt/B

Step 3: select the smallest rate quantization
        if (wt >= 0.25 & w[0] <= 0.25 & ra <= 0.25) {
          r = 0.25
        }
        else if (wt >= 0.5 & w[0] <= 0.5 & ra <= 0.5)) {
          r = 0.5
        }
        else if (wt >= 0.75 & w[0] <= 0.75 & ra <= 0.75)) {
          r = 0.75
        }
        else if (wt >= 1) {
          r = 1
        }
        else if (w[0] == 0 & wt < 0.25) {
          r = 0
        }
        else {
          r = wt
        }
```

Figure 6.16: Rate selection algorithm with enhancements 1, 2, and 3

Workloads: 63, 62, 49, 10, 2, 1, ...

Buffer Contents: Unprocessed sample workload

Data Shifting Direction: ←

Period	Data Buffer w[0]	w[1]	w[2]	W_t	r_a	w[0]	r
0	0	0	15	15	5	0	0
1	0	15	16	31	10	0	16
2	0	15	63	78	26	0	32
3	0	46	62	108	36	0	48
4	0	60	62	122	40	0	48
5	12	62	64	138	46	12	48

Figure 6.17: An example of the rate selection algorithm with enhancements 1, 2, and 3

the non rate quantization has significantly improved in the $0 < r < 0.25$ range. For all the test video sequences, the selection efficiency is over 95% and in 7 out of the 8 sequences, the efficiency is approximately 100%. The energy results and transition count for the data sequences for this enhancement are shown in Figure 6.21. This figure shows that idle rate in fact reduces the data processing energy for all data sequences, and the savings range from 0% for Hall to 42% for Akiyo. As for transition energy, the idle rate selection has only reduced energy in Akiyo, and for all other sequences the energy consumption has gone up to 17% for Mother sequence. Finally, the idle rate selection has reduced voltage transitions in all but one data sequence (Hall). More specifically, the magnitude of reduction ranges from 2% for Coastguard to 47% for Akiyo.

In summary, the experimental results demonstrates that idle rate selection enhances the rate selection algorithm in minimizing the total energy of computation.

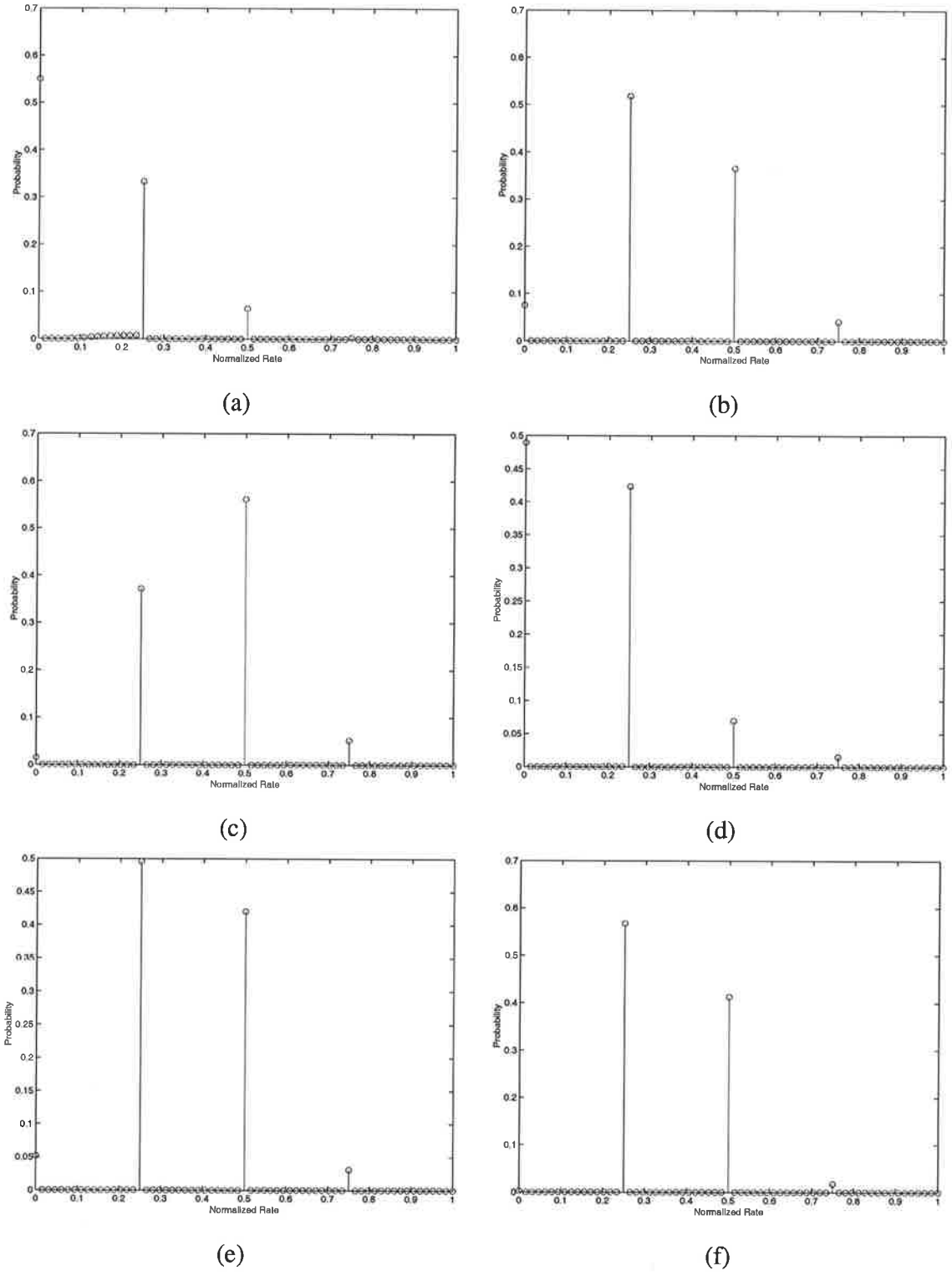
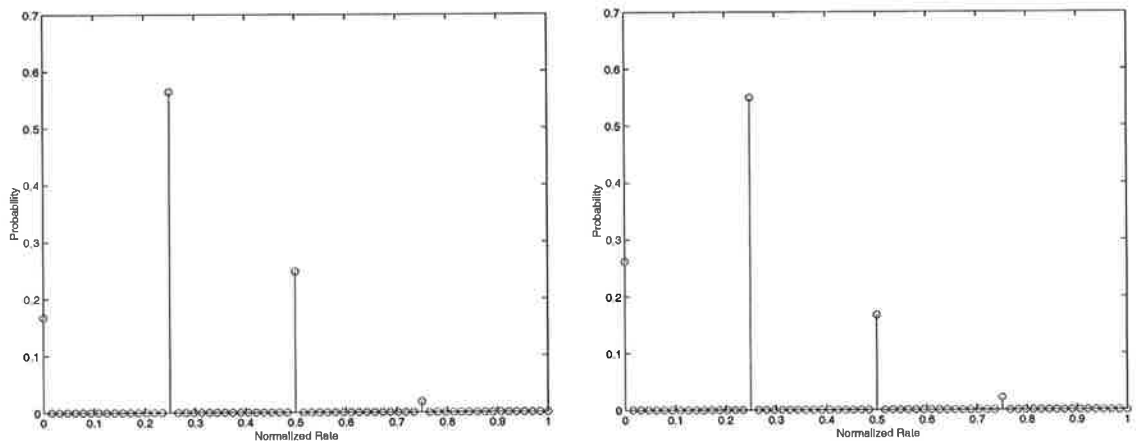


Figure 6.18: Rate distributions of MPEG-2 test video sequences for the rate selection algorithm with enhancements 1, 2, and 3. (a) Akiyo, (b) Carphone, (c) Coastguard, (d) Container (e) Foreman, (f) Hall



(g) (h)

Figure 6.19: Rate distributions of MPEG-2 test video sequences for the rate selection algorithm with enhancements 1, 2, and 3. (g) Mother, and (h) Silent

6.3 Computational Complexity Reduction

Previous enhancements involved improvements made to the selection policy of the rate selection algorithm such that the total energy of computation is minimized. This section aims to minimize the computational overhead associated with the rate selection algorithm.

The final rate selection algorithm presented in the last section comprises three main steps: 1) calculation of total workload in the buffer, 2) calculation of average workload, and 3) the selection of rate quantization. This section shows two methods of reducing the complexity of steps 1 and 2.

From these three steps, step 1 involves a repetitive, fixed-overhead summation operation which provides the best opportunity for optimization. In order to accomplish this goal we propose an alternative implementation to step 1. This implementation is shown in Figure 6.22. As this figure shows, the total buffered workload of a sample period is calculated relative to the total workload in the previous sample period. In other words, the total workload in the buffer is equal to what was there in the previous sample period, plus the workload of the new sample buffered in ($w[B-1]$), minus the amount of workload processed by the rate r in the last sample period. Thus, the algebraic sum provides a simpler computation in calculating the total buffered workload. In order to get this technique to work, the wt and r

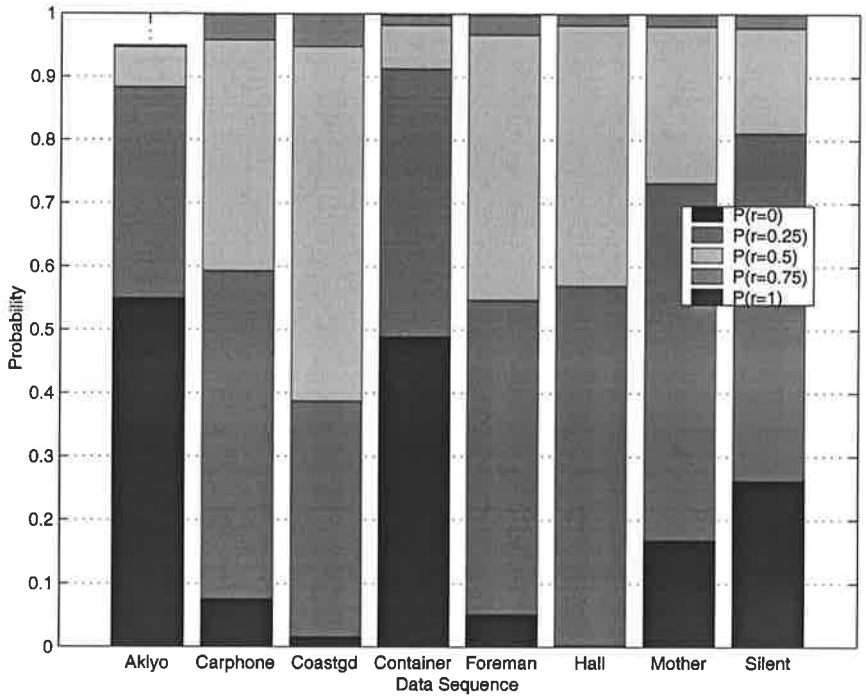


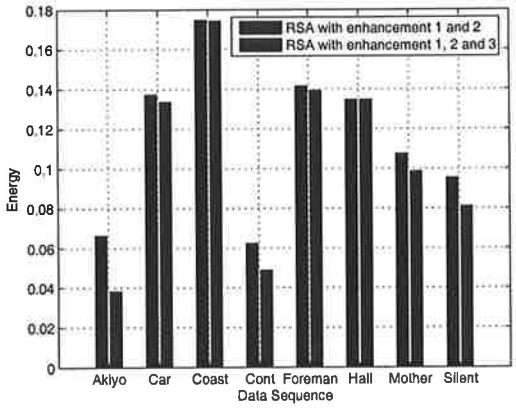
Figure 6.20: Rate quantization distributions for the rate selection algorithm with enhancements 1, 2, and 3

values must be initialized to 0 at the beginning of processing, and these values need to persist throughout the computation.

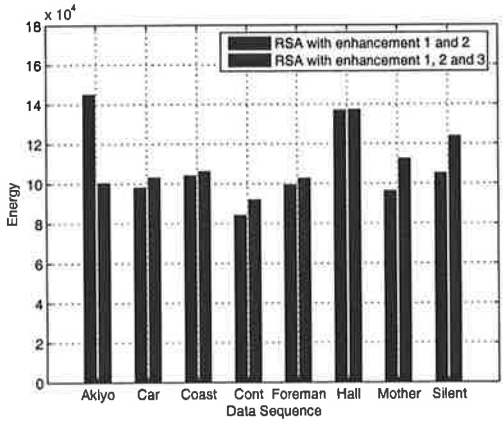
By using this technique, calculation of total buffered workload can be accomplished by 2 operations (one *add* and one *subtract*). Consequently, the total computational overhead of the rate selection algorithm varies between 6 and 19.

In terms of using this implementation in the rate selection algorithm and dynamic voltage scaling, the biggest advantage is its independence of computational complexity from buffer size. Thus, larger buffer sizes can be used with the rate selection approach without affecting the computational complexity.

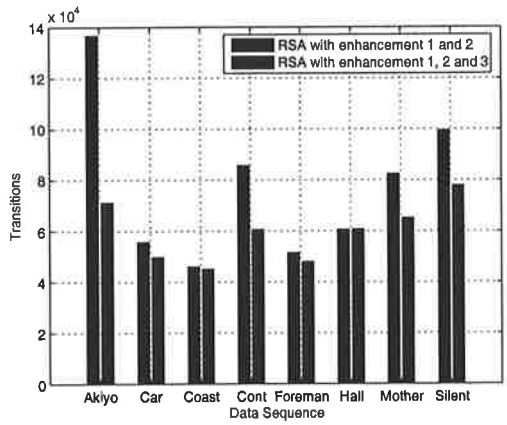
Step 2 of the algorithm involves a more complex division operator. However, the complexity of this step can be reduced by selecting the buffer size such that $B = 2^n$, where n is an integer ($n > 0$), and using a right shift operation for division.



(a)



(b)



(c)

Figure 6.21: Energy costs and transition count for rate selection approach with enhancement 3. (a) Data processing energy cost E_{pr}, (b) Transition energy cost E_{tr}, (c) transition count

```
Step 1: calculate total workload in the buffer
        wt = 0 (initialized before zeroth sample period)
        r = 0  (initialized before zeroth sample period)

        wt = wt + w[B-1] - r
```

Figure 6.22: Complexity reduced implementation of step 1 of the rate selection algorithm

6.4 Summary

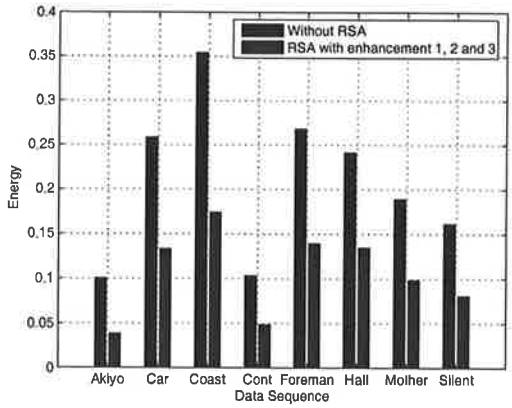
This chapter presented a number of enhancements that improve the energy efficiency and reduce the computational complexity of the rate selection approach. The chapter also presented the results of our experimental analysis performed on these enhancements.

Based on these enhancements, the final rate selection algorithm is presented in Figure 6.23. Using this algorithm, the overall energy and transition reduction is shown in Figure 6.24. Based on this figure, it is clear that use of final rate selection algorithm provides significant data processing and transition energy savings for all test data sequences. Moreover, the algorithm produces significant reductions in voltage transitions for all data sequences as well. Quantitatively, the magnitude of savings achieved range from 44% (Hall) to 62% (Akiyo) for data processing energy, 52% (Silent) to 78% (Coastguard) for transition energy, and 54% (Container) to 74% (Coastguard) for transition count.

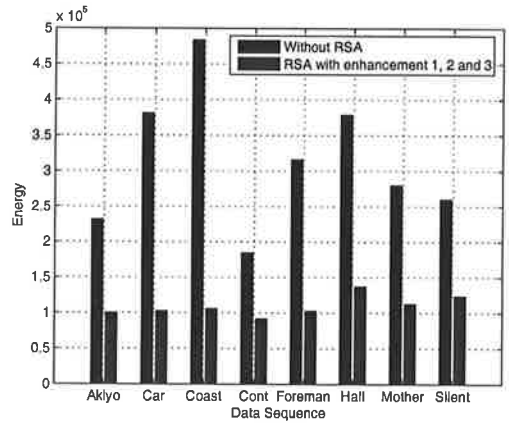
The next chapter uses this algorithm in evaluating its effectiveness across a number of voltage quantizations and buffer sizes.

```
Step 0: Initialize variables
        wt = 0 (initialized before zeroth sample period)
        r = 0  (initialized before zeroth sample period)

Step 1: calculate total workload in the buffer
        wt = wt + w[B-1] - r (Note r <= wt)

Step 2: calculate the average workload in the buffer
        ra = wt/B

Step 3: select the rate quantization
        if (wt >= 0.25 & w[0] <= 0.25 & ra <= 0.25) {
          r = 0.25
        }
        else if (wt >= 0.5 & w[0] <= 0.5 & ra <= 0.5)) {
          r = 0.5
        }
        else if (wt >= 0.75 & w[0] <= 0.75 & ra <= 0.75)) {
          r = 0.75
        }
        else if (wt >= 1) {
          r = 1
        }
        else if (w[0] == 0 & wt < 0.25) {
          r = 0
        }
        else {
          r = wt
        }
```

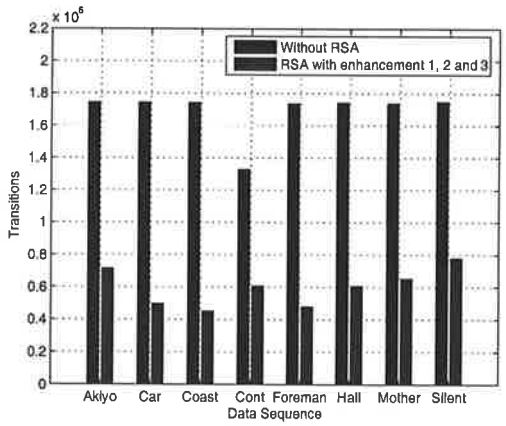
Figure 6.23: Final Rate selection algorithm



(a)



(b)



(c)

Figure 6.24: Energy costs and transition count for final rate selection approach. (a) Data processing energy cost E_{pr}, (b) Transition energy cost E_{tr}, (c) transition count

Chapter 7

Performance Analysis of Rate Selection

The previous two chapters demonstrated the concept of the rate selection approach and a number of enhancements to minimize idle energy loss and computational complexity. This chapter extends the work of the previous two chapters by providing an extended performance analysis of the rate selection approach.

7.1 Introduction

The development of rate selection approach in the last two chapters were driven mainly by the need to minimize idle loss or the E_{pr} term of Equation 4.5. As the results of last chapter shows that the rate selection approach is effective in reducing E_{pr}, this chapter performs a full analysis of the algorithm to show its effectiveness in minimizing total number of voltage transitions and E_{tr} as well. This chapter also presents the performance of the rate selection approach as a function of buffer size and the number of voltage quantizations. Finally, performance comparisons to prior art is also presented. Since E_{pr} and E_{tr} can be significantly different in terms of orders of magnitude (depending on the computation), this chapter evaluates the two variables separately.

7.2 Buffer Size Variation

Since rate selection approach depends on sample buffering, it is important to evaluate the effect of buffer size on energy efficiency, and the total number of voltage transitions. This

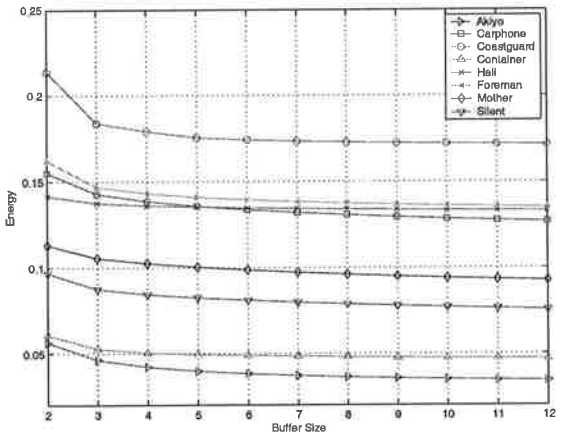
section presents the results of this analysis. As in the previous two chapters, this analysis uses a 4-level voltage quantization corresponding to normalized rate quantizations of 0.25, 0.5, 0.75, and 1.0. As for the buffer size, it is varied from 2 to 12. Buffer size of 1 is not used because such a buffer will not provide any opportunity to decouple the rate from the workload, and rate selection approach requires such decoupling of rate and workload. The results are shown in Figure 7.1. Figures 7.1(a), 7.1(b), and 7.1(c) show the data processing energy, transition energy, and the transition counts, respectively. As these figures show, both the energy costs and transition costs are reduced as the buffer size is increased. Thus, it can be concluded that increasing buffer size is beneficial to the rate selection approach, and for resource limited systems, any buffer sizes higher than $B = 2$ can provide additional energy savings and reductions in transitions.

7.3 Voltage Quantizations Variation

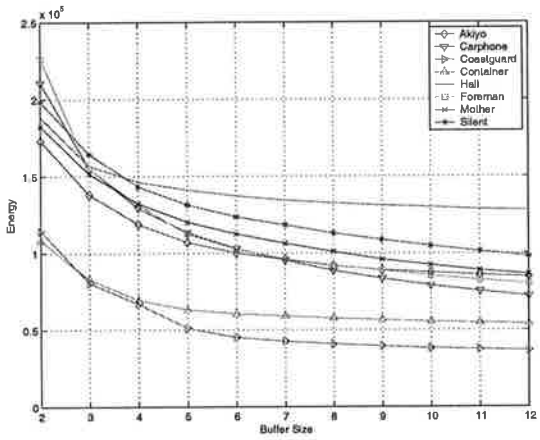
Throughout this thesis, analysis of the voltage quantization model used a 4-level voltage quantization. This section evaluates the performance of the rate selection approach for voltage quantizations of 2, 3, 4, and 5. For these voltage quantizations, the data processing energy, transition energy, and transition counts are evaluated and the results are shown in Figure 7.2. As these curves show, the energy costs are reduced as the number of voltage quantizations is increased. This is consistent with the voltage quantization model where the energy efficiency improves as the number of voltage quantizations is increased. As for the transition counts, the same trend applies to the majority of data sequences. For this analysis, a buffer size of 6 is used.

7.4 Comparison to Prior Work

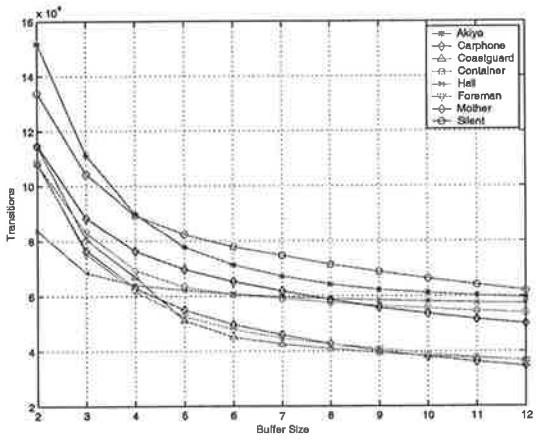
This section compares the rate selection approach with the voltage dithering approach proposed by Gutnik *et. al* [Gut96]. Moreover, comparison to continuous voltage level model with buffering and workload averaging is also performed. The comparisons will be made against voltage quantizations and buffer size. As before, separate comparisons will be made on data processing energy, transition energy, and transition counts.



(a)

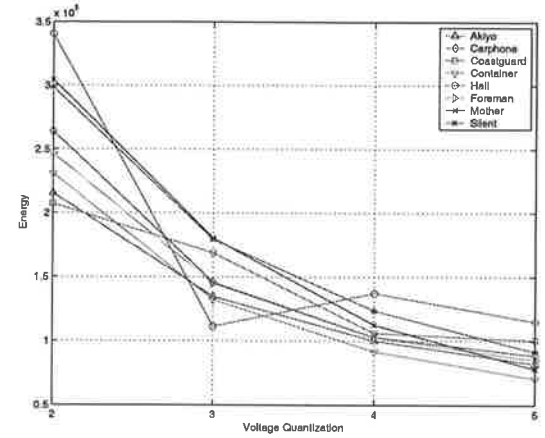


(b)

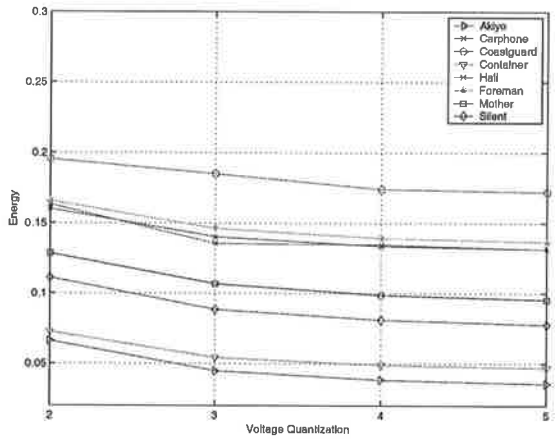


(c)

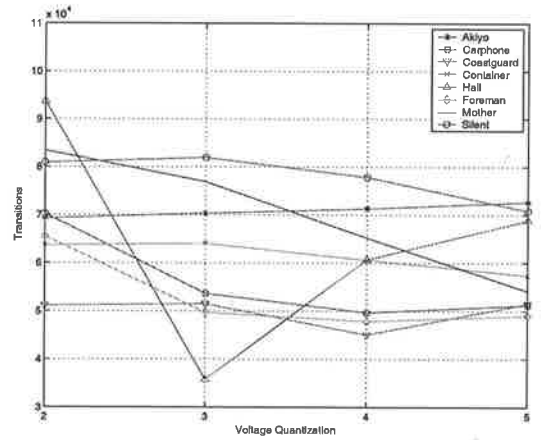
Figure 7.1: Energy costs and transition count versus buffer size for all data sequences. (a) Data processing energy cost E_{pr}, (b) Transition energy cost E_{tr}, (c) transition count



(a)



(b)



(c)

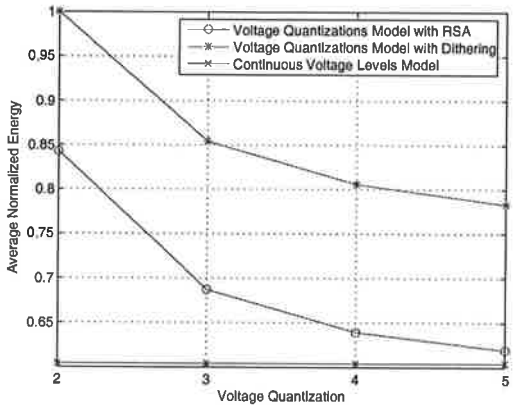
Figure 7.2: Energy costs and transition count versus voltage quantizations for all data sequences. (a) Data processing energy cost E_{pr}, (b) Transition energy cost E_{tr}, (c) transition count

7.4.1 Versus Voltage Quantizations

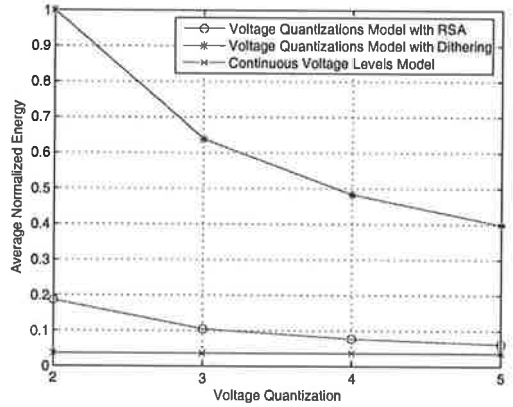
The average normalized energy cost and transition count for all data sequences are shown in Figure 7.3. (The plots for each data sequence are given in Section B.1 of Appendix B). In Figures 7.3(a) and (b), the curves for the continuous voltage levels remain flat across voltage quantizations. This is because the continuous voltage levels uses infinite voltage levels, and has no impact on voltage quantizations. As Figure 7.3(a) shows, rate selection approach provides additional savings of 16% to 21% in data processing energy compared to voltage dithering. However, comparison with continuous voltage levels shows that rate selection is 2% to 40% less energy efficient compared to continuous voltage levels model. As for transition energy, Figure 7.3(b) shows that rate selection achieves additional energy savings of 81% to 84% compared to voltage dithering. However, comparison with continuous voltage levels shows that additional savings range from 42% to about -14%. The results for transition count as shown in Figure 7.3(c) provides the key strength of rate selection approach: it provides the the least number of voltage transitions compared to *both* voltage dithering and continuous voltage levels. Quantitatively the reductions in voltage transitions range from 75% to 84% compared to voltage dithering, and 47% to 52% compared to continuous voltage levels. Based on these results it can be concluded that for all voltage quantizations, the rate selection approach reduces both energy costs and voltage transitions compared to the voltage dithering technique. Moreover, if reduction of voltage transitions is of importance (i.e. to reduce switching noise for application areas such as wireless communication systems), the rate selection approach is superior to both voltage dithering and continuous voltage level models.

7.4.2 Versus Buffer Size

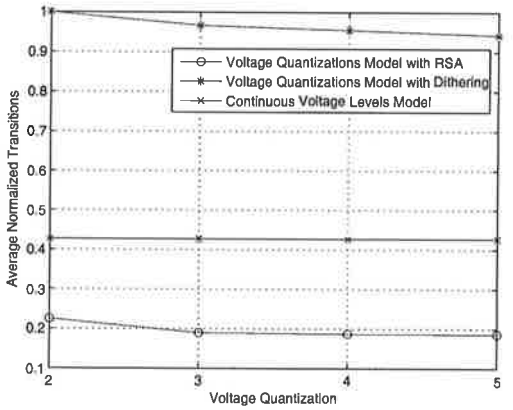
The average energy cost and transition count for all data sequences are shown in Figure 7.4. (The plots for each data sequence are given in Section B.2 of Appendix B). In Figures 7.4(a) and (b), the curves for the continuous voltage levels remain flat across voltage quantizations. This is because the continuous voltage levels uses infinite voltage levels, and has no impact on voltage quantizations. As Figure 7.4(a) shows, rate selection approach provides additional savings of 5% to 32% in data processing energy compared to voltage dithering.



(a)



(b)

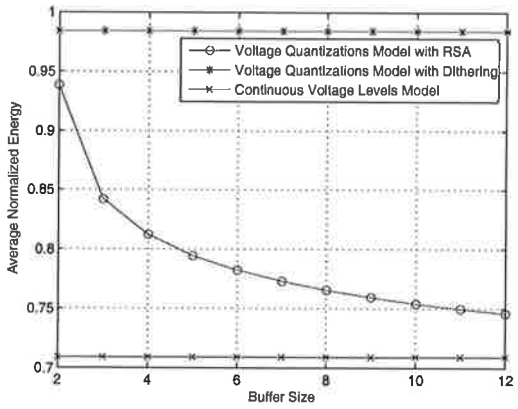


(c)

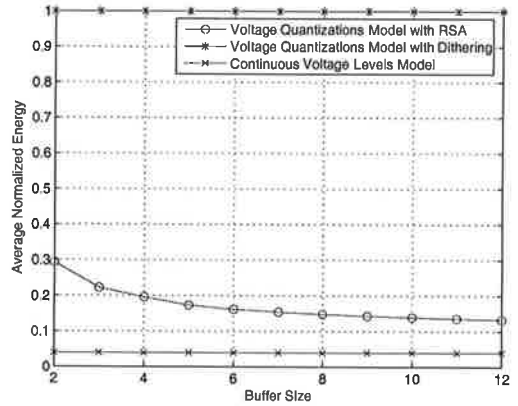
Figure 7.3: Average energy costs and transition count of all data sequences compared to prior work (versus voltage quantizations). (a) Data processing energy cost E_{pr}, (b) Transition energy cost E_{tr}, (c) transition count

However, comparison with continuous voltage levels shows that rate selection is 5% to 24% less energy efficient compared to continuous voltage levels model. As for transition energy, Figure 7.4(b) shows that rate selection achieves additional energy savings of 246% to 642% compared to voltage dithering. However, comparison with continuous voltage levels shows that additional savings range from 62% to about 84%. The results for transition count is shown in Figure 7.4(c). As this figure shows, the transition count is less for rate selection approach compared to both voltage dithering and continuous voltage levels, except at $B = 2$. At $B = 2$, continuous voltage levels provide an additional saving of 12% in transition count. At all other buffer sizes, rate selection approach provides 62% to 84% savings in transition count compared to dithering, and upto 52% saving compared to continuous voltage levels. Based on these results it can be concluded that for all voltage quantizations (except 2), the rate selection approach reduces both energy costs and voltage transitions compared to the voltage dithering technique. Moreover, if reduction of voltage transitions is of importance (i.e. to reduce switching noise for application areas such as wireless communication systems), the rate selection approach is superior to both voltage dithering and continuous voltage level models.

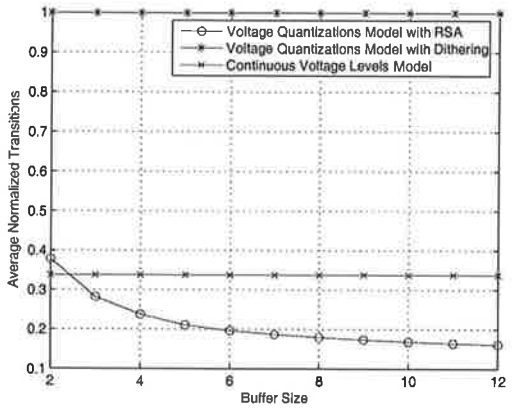
In these figures, curves for voltage dithering technique stay flat across buffer sizes. This is because the voltage dithering technique does not use buffering. As these figures show, the rate selection approach is more energy efficient and reduces total voltage transitions compared to voltage dithering technique for all test data sequences. As buffer size is increased, the E_{pr} of rate selection approach decreases in a fashion similar to continuous voltage levels model, and E_{tr} for RSA is lower than continuous voltage levels model. For the test data sequences used in our experiments, the maximum energy savings achieved by the rate selection approach compared to voltage dithering were 30% and 91% for data processing energy and transition energy, respectively. As for transition counts, the maximum saving was 89%. Comparison to continuous voltage levels shows that energy costs can be as close as 0.8% of the energy cost of continuous voltage levels or as high as 4.4 times the energy cost of continuous voltage levels. As for the transition counts, rate selection provides a maximum saving of 76%.



(a)



(b)



(c)

Figure 7.4: Average energy costs and transition count of all data sequences compared to prior work (versus buffer). (a) Data processing energy cost E_{pr}, (b) Transition energy cost E_{tr}, (c) transition count

7.5 Summary

This chapter presented the performance of the rate selection approach and compared the results with existing approaches. The results indicate that rate selection is more energy efficient than the best existing technique for voltage quantization model (the voltage dithering technique). Moreover, the rate selection approach significantly reduces the number of voltage transitions. Comparison to ideal continuous voltage levels with workload averaging also reveals that the rate selection technique can achieve data processing energy values very close to continuous voltage level model, particularly with higher voltage quantizations and buffer sizes. Moreover, if reduction of total voltage transitions is of utmost importance due to noise issues, rate selection approach provides a better alternative to any existing approach, including continuous voltage levels model.

Chapter 8

Conclusions

8.1 Summary of the Research

This study focused on evaluating the voltage quantization model and developing a novel approach for improving its effectiveness for dynamic voltage scaling. The voltage quantization model uses a small number of discrete voltage levels to represent the full voltage range used for scaling. The advantage of this approach is faster voltage transitions. The main disadvantage of this approach is the inability to scale voltage for all workload values and this results in selection of higher than ideal processing rates. Consequently, processing at higher voltage levels than ideal leads to idle times and idle losses. There are two existing approaches to reducing this idle loss: 1) clock gating, and 2) voltage dithering. Clock gating is a hardware approach that turns the clock off when idle time is identified. However, in the context of fixed throughput mode of computation, clock gating becomes less effective because most samples in a data sequence will need clock gating, and hence clock gating will be performed at a very high frequency. Voltage dithering on the other hand requires additional voltage transition per data sample, and this incurs an energy cost and increased number of voltage transitions. Moreover, if the sample period of the computation is very small, voltage dithering becomes infeasible.

The research work presented in this thesis involved developing an approach that not only improved the energy efficiency, but also minimized the total number of voltage transitions. Our approach, the rate selection algorithm, is a low overhead, algorithm-level technique that

uses sample buffering to decouple processing rate from actual workload values and where possible selects rate quantizations as the processing rate. This thesis describes a simple rate selection algorithm and a number of enhancements such as prioritization, use of average buffered workload, and idle rate selection for improving the energy efficiency of dynamic voltage scaling using voltage quantization model. The thesis also demonstrates a technique that also minimizes the computational overhead of the rate selection approach.

The development and performance analysis of the rate selection approach was done using a MPEG-2 video test data set. Since dynamic voltage scaling in fixed throughput mode computations requires the *a priori* knowledge of workload, we used the IDCT portion of the MPEG-2 decoding computation as our computation because of the availability of the FMIDCT implementation for IDCT computation which enables the *a priori* calculation of workload based on the number of non-zero coefficients in image blocks. In order to test the performance of the rate selection approach across a wide variety of workload distributions, 8 data sequences comprising different levels of motion, content, and camera panning were used.

Based on our data set, our experimental results demonstrate that the rate selection approach is very energy efficient compared to existing approaches for the voltage quantization model. Based on our research, the following conclusions can be made:

1. The rate selection approach improves the energy efficiency of the voltage quantization model by reducing both data processing energy and transition energy, compared to the best existing approach (voltage dithering). The maximum savings achieved were around 30% for data processing energy, and around 90% for transition energy, compared to voltage dithering.

2. The rate selection approach reduced the total number of voltage transitions, compared to voltage dithering. Based on our data sets, the maximum reduction of voltage transitions was around 90%, compared to voltage dithering.

3. The rate selection approach has a very low computational overhead. More specifically, determination of rate values in the rate selection algorithm involves between 6 and 19 arithmetic operations per sample. More importantly, computational overhead is buffer size independent.

4. The increase of voltage quantizations and buffer size causes the rate selection approach to produce increased energy savings and reduced voltage transitions. However, even at smaller voltage quantizations and buffer sizes, the rate selection approach is more effective than the voltage dithering technique. This in turn makes the rate selection approach useful in applications supporting a very small number of voltage quantizations and small buffer sizes.

5. If reduction of voltage transitions is the most important consideration (such as for wireless applications), the rate selection approach is even more effective than the continuous voltage level model. Moreover, if data processing energy dominates transition energy, increasing the buffer size and using a higher number of voltage quantizations enables the rate selection approach to achieve energy efficiencies that approach the continuous voltage level model.

Based on these conclusions, it is clear that our rate selection approach is a very effective replacement for existing voltage dithering technique for voltage quantization model. Moreover, by selecting the number of voltage quantizations and buffer size, the energy efficiency of the rate selection technique can approach the ideal energy efficiency of the continuous voltage level model, while incurring the least number of voltage transitions compared to any existing technique.

8.2 Summary of Research Contributions

The key contributions of this research are given below.

- Successfully demonstrated that total energy of computation for the voltage quantization model can be minimized through transformation of the workload distribution of data sequences. This was a new direction compared to all existing research which only focuses on energy minimization through optimization of the energy model. Consequently, we were able to use a very general energy model, which in turn resulted in making less assumptions about the energy model.

- Developed a novel approach called rate selection that uses sample buffering to where possible select rate values that are equal to rate quantizations or idle rate. This approach transforms the workload distribution of data sequences to minimize the energy cost and total transitions. We also proposed a number of enhancements to the rate selection approach that improve the energy efficiency while minimizing the computational overhead. The performance analysis of the rate selection approach with a wide variety of workload distribution patterns also demonstrated that the approach is very effective in transforming workload distributions and minimizing total energy cost and voltage transitions.

- Successfully analyzed the total number of voltage transitions and their energy cost as part of our experimental work. This was done because transition energy can be significant in computations where sample period is comparable to voltage transition time. In prior research transition energy and transition counts have been ignored. For our analyses, we used the transition energy model recently proposed by Burd [Bur01].

8.3 Limitations and Future Research

This thesis introduced workload distribution transformation as a new direction of research for improving the energy efficiency of the voltage quantization model. We also successfully demonstrated a very effective approach for transforming workload distributions to achieve better energy efficiencies and low transition counts for the voltage quantization model. However, our approach has a number of limitations, and exploring solutions for these limitations can be done in future research.

The design idea for the proposed approach is very highly dependent on buffering as a means of decoupling rate from workload. However, this can be a serious limitation in some applications because buffering requires allocation of some memory to store data samples (and workload values), and this would mean that larger buffers will be required for computations with larger data samples. Moreover, buffering also introduces a processing delay to the computation. Typically a buffer size of B delays the processing of the entire data sequence by B sample periods. Thus, in some applications, limitations on memory and acceptable

processing delay may limit the use of the rate selection approach. Some future research may be carried out to investigate alternative approaches to buffering as methods for decoupling workload from rate.

The proposed approach is also dependent upon the *a priori* knowledge of the workload. This means that exact workload determination must be possible for the computation before the rate selection approach can be used. However, this is infeasible in most real fixed throughput mode computations even though they are data dependent and their workload varies with the data. Moreover, current multimedia standards do not support embedding workload values in header information, and until this occurs, the proposed rate selection approach will only have limited use. Thus, future research for incorporating workload predictions and prediction error handling into the rate selection approach can make the approach more useful to a wide range of fixed throughput mode computations.

Another limitation of the proposed approach is its design for fixed throughput mode of computations. Even though fixed throughput mode of operation is common in many dedicated portable systems, a large proportion of portable systems also operate in burst mode of operation. Thus, future research that improves the rate selection approach to operate in burst mode of operation can also increase the use of the approach in portable devices.

The final limitation of the proposed approach is the assumption of a single computation mode of operation. Since this type of signal processing application is common at the moment, reduction in the cost of faster general purpose processors running operating systems are becoming more and more common. Thus, research into using the rate selection approach in general purpose processors with operating systems would be another area of future research.

8.4 Recent Developments

Since the completion of the research presented in this thesis, feature sizes have continued to shrink towards sub-micron levels with each technology generation, and this has lead to lower threshold voltages. One of the key problems associated with low threshold voltages is the exponential increase in leakage current and consequently the leakage power component [MFBM02]. Chandrakasan [CWB01] predicts that as technologies continue to scale, leakage

power component to become comparable to dynamic power component. This trend is clearly evident in the recently released Intel Pentium4 3.0GHz processor that has comparable dynamic and leakage power components [Int]. Thus, leakage power reduction techniques will be very important for reducing the overall power in submicron technologies.

Current research demonstrates two main leakage power reduction methods. These are adaptive reverse body biasing [MIS+99], [KSN01], [LM02] and forward body biasing [MKC02], [NMH02]. These adaptive body biasing (ABB) methods control the leakage current during standby mode, and have the important advantage of exponentially reducing the leakage current [MFBM02].

In addition to leakage power reduction, DVS would continue to be important for reducing dynamic power component in sub-micron technologies. In addition to dynamic power savings, DVS also provides a linear reduction in leakage current [MFBM02]. Thus, combined approaches that reduce both dynamic and leakage power components (i.e. DVS and ABB) will be very important in future research.

Appendix A

Test Video Sequences

NOTE:
This appendix is included on pages 141-145 of the print copy of the thesis held in the University of Adelaide Library.

Appendix B

Energy Cost and Transition Count Comparison to Prior Work

B.1 Versus Voltage Quantizations

This section shows the energy costs and transition count comparisons with prior approaches for all individual data sequences. These are shown in Figures B.1,B.2,B.3,B.4,B.5, and B.6.

B.2 Versus Buffer Size

This section shows the energy costs and transition count comparisons to prior approaches for all individual data sequences. These are shown in Figures B.7,B.8,B.9,B.10,B.11, and B.12.

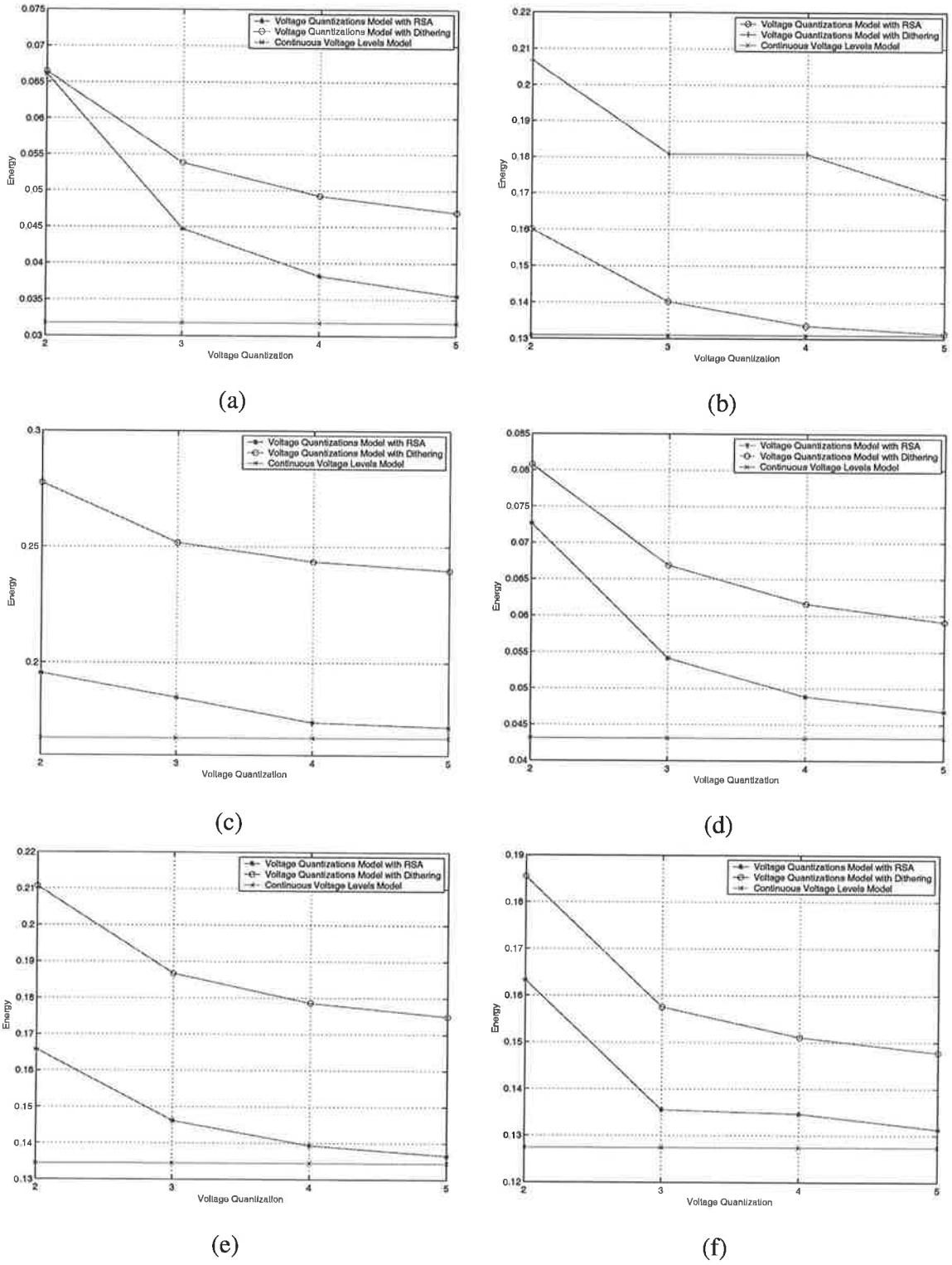
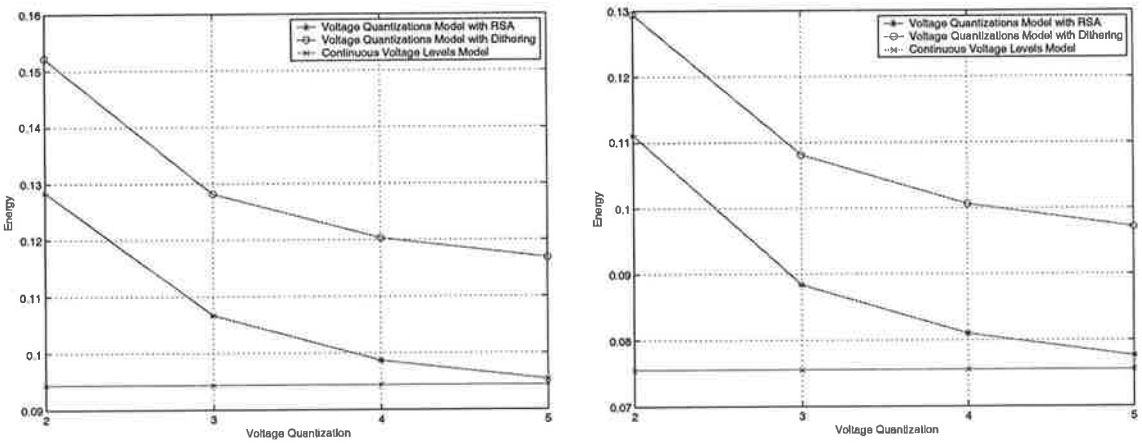


Figure B.1: Data processing energy cost E_{pr} comparison to prior work (versus voltage quantizations). (a) Akiyo, (b) Carphone, (c) Coastguard, (d) Container (e) Foreman, (f) Hall



(g) (h)

Figure B.2: Data processing energy cost E_{pr} comparison to prior work (versus voltage quantizations). (g) Mother, and (h) Silent

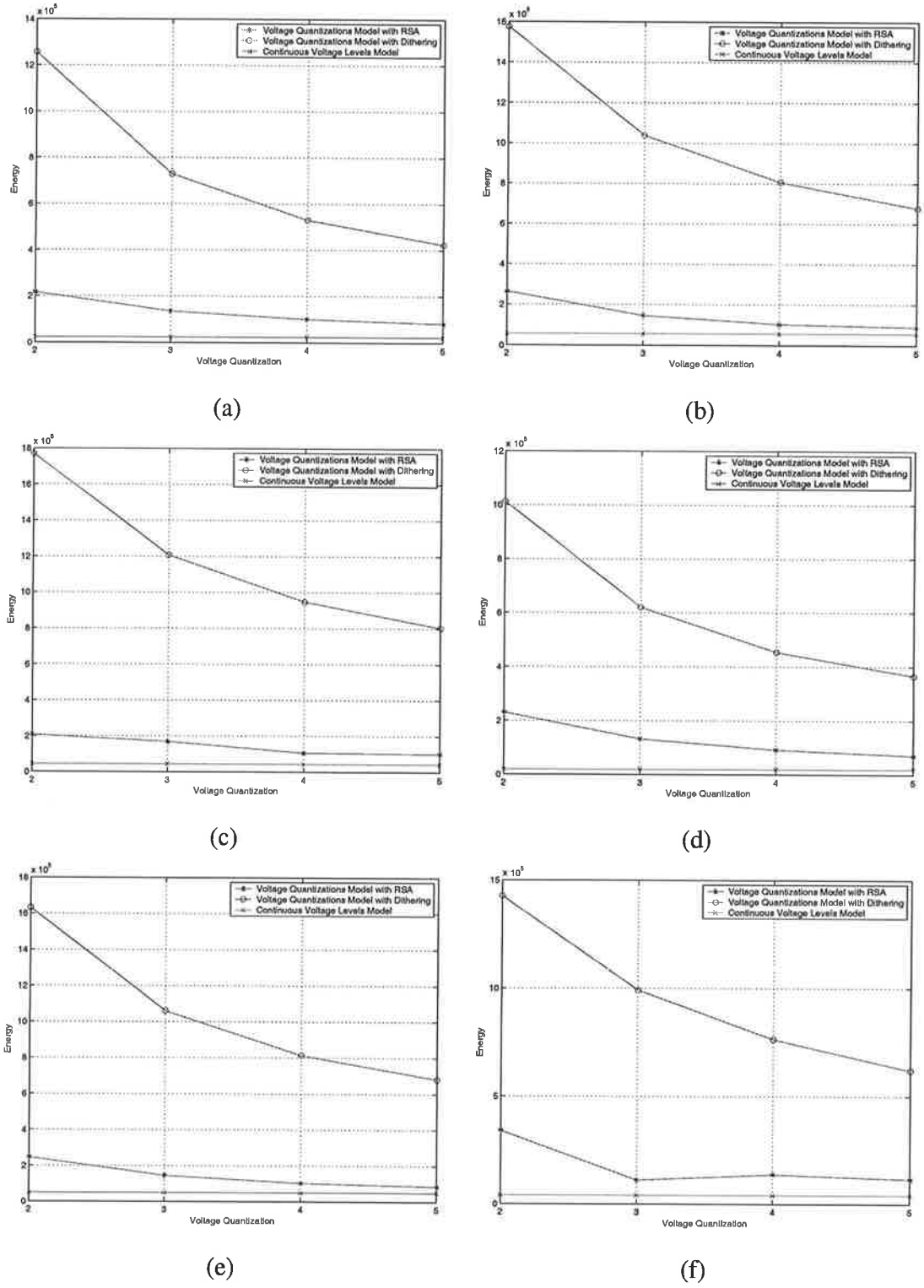
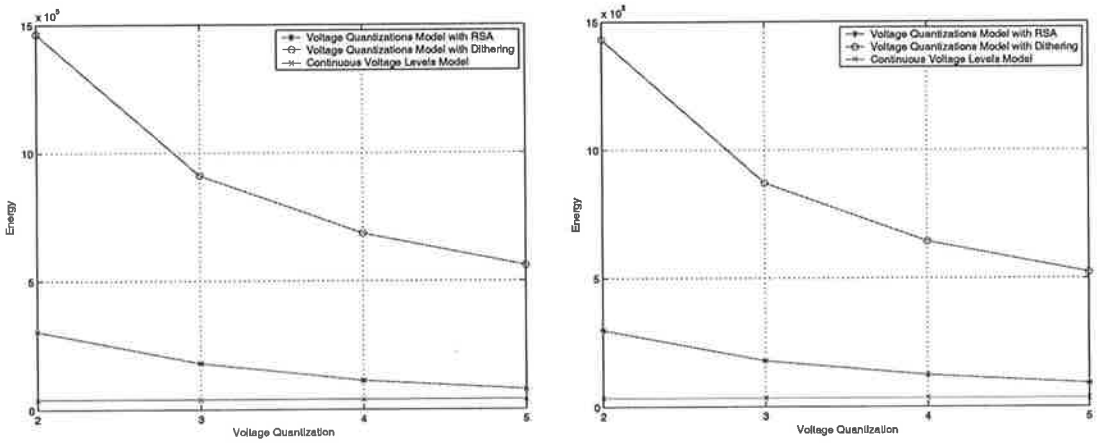


Figure B.3: Transition energy cost E_{tr} comparison to prior work (versus voltage quantizations). (a) Akiyo, (b) Carphone, (c) Coastguard, (d) Container (e) Foreman, (f) Hall



(g) (h)

Figure B.4: Transition energy cost E_{tr} comparison to prior work (versus voltage quantizations). (g) Mother, and (h) Silent

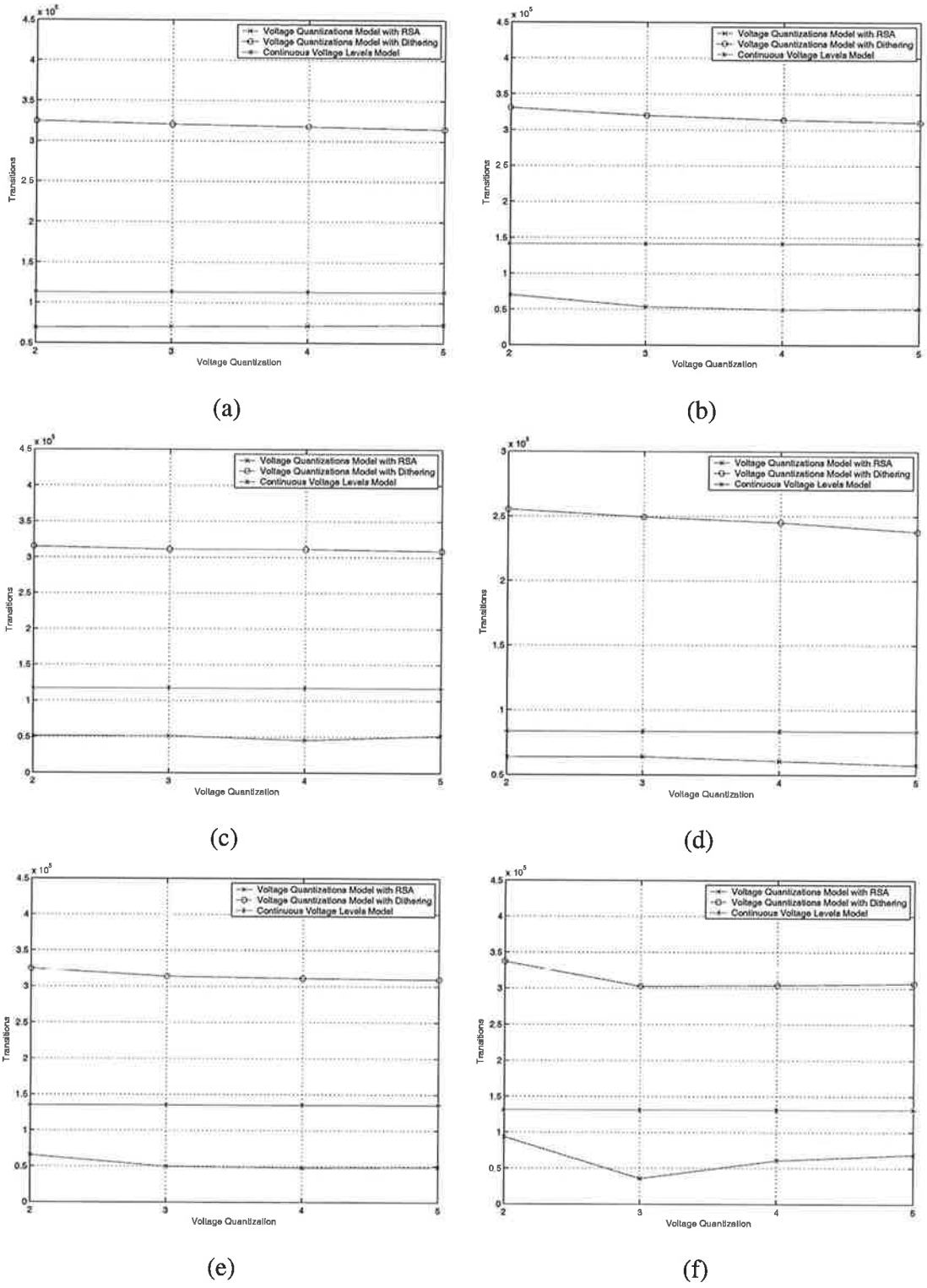
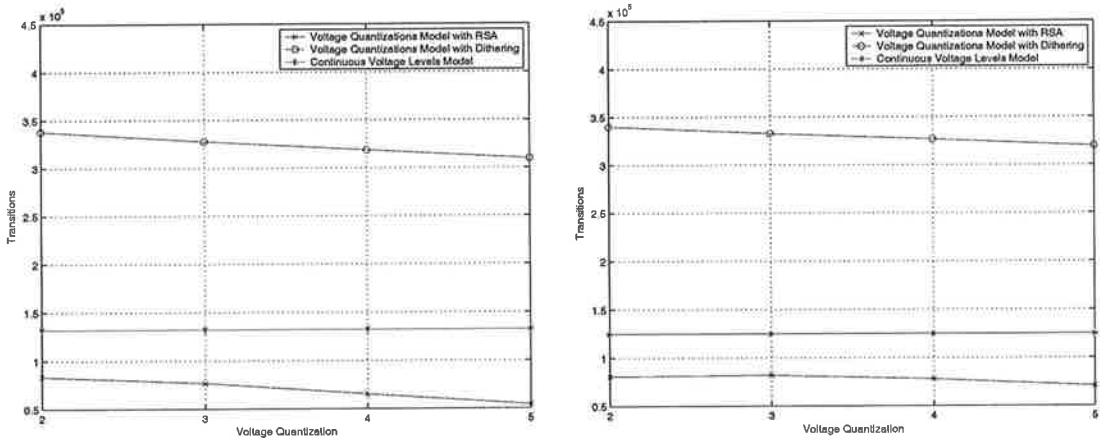


Figure B.5: Transition count comparison to prior work (versus voltage quantizations). (a) Akiyo, (b) Carphone, (c) Coastguard, (d) Container (e) Foreman, (f) Hall



(g) (h)

Figure B.6: Transition count comparison to prior work (versus voltage quantizations). (g) Mother, and (h) Silent

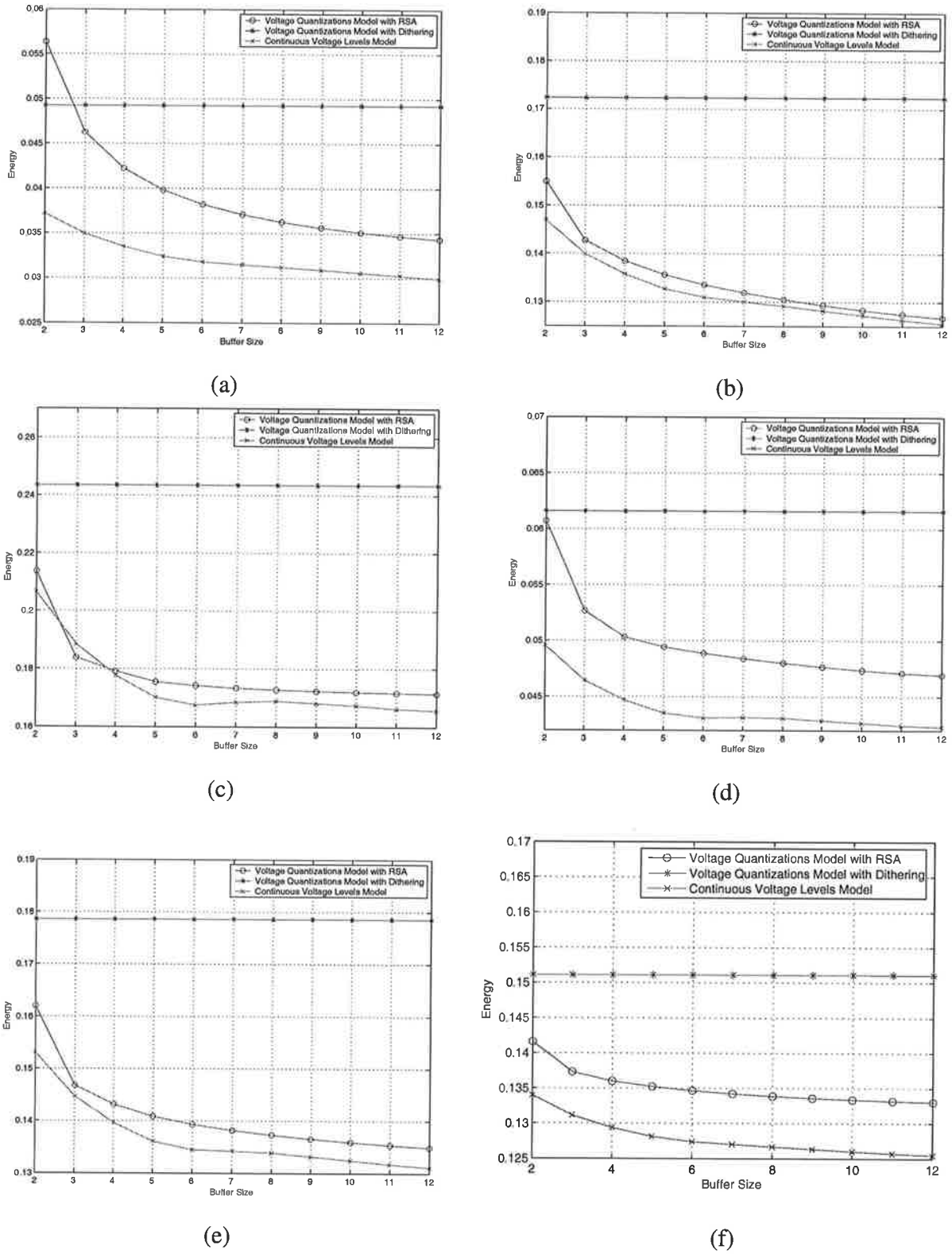
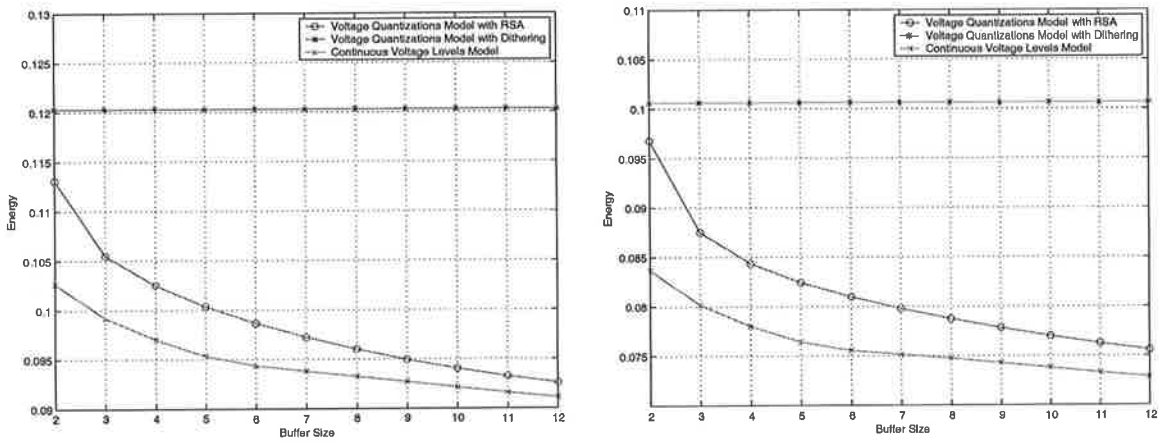


Figure B.7: Data processing energy cost E_{pr} comparison to prior work (versus buffer size). (a) Akiyo, (b) Carphone, (c) Coastguard, (d) Container (e) Foreman, (f) Hall



(g) (h)

Figure B.8: Data processing energy cost E_{pr} comparison to prior work (versus buffer size). (g) Mother, and (h) Silent

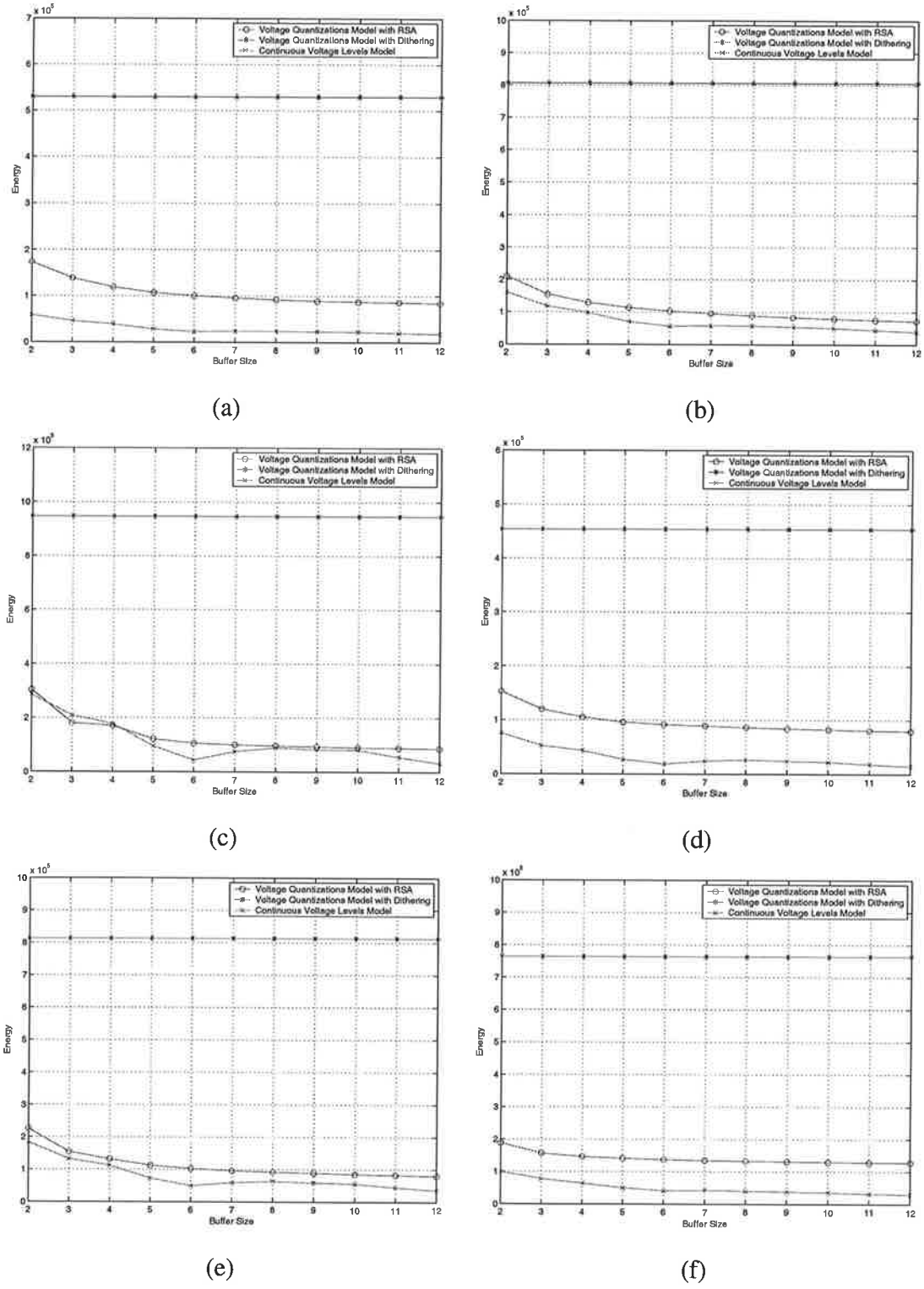
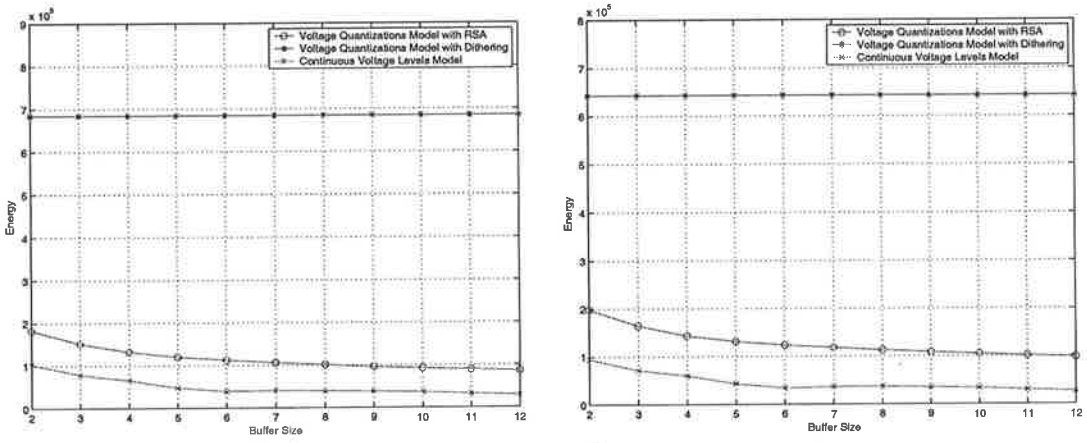


Figure B.9: Transition energy cost E_{tr} comparison to prior work (versus buffer size). (a) Akiyo, (b) Carphone, (c) Coastguard, (d) Container (e) Foreman, (f) Hall



(g) (h)

Figure B.10: Transition energy cost E_{tr} comparison to prior work (versus buffer size). (g) Mother, and (h) Silent

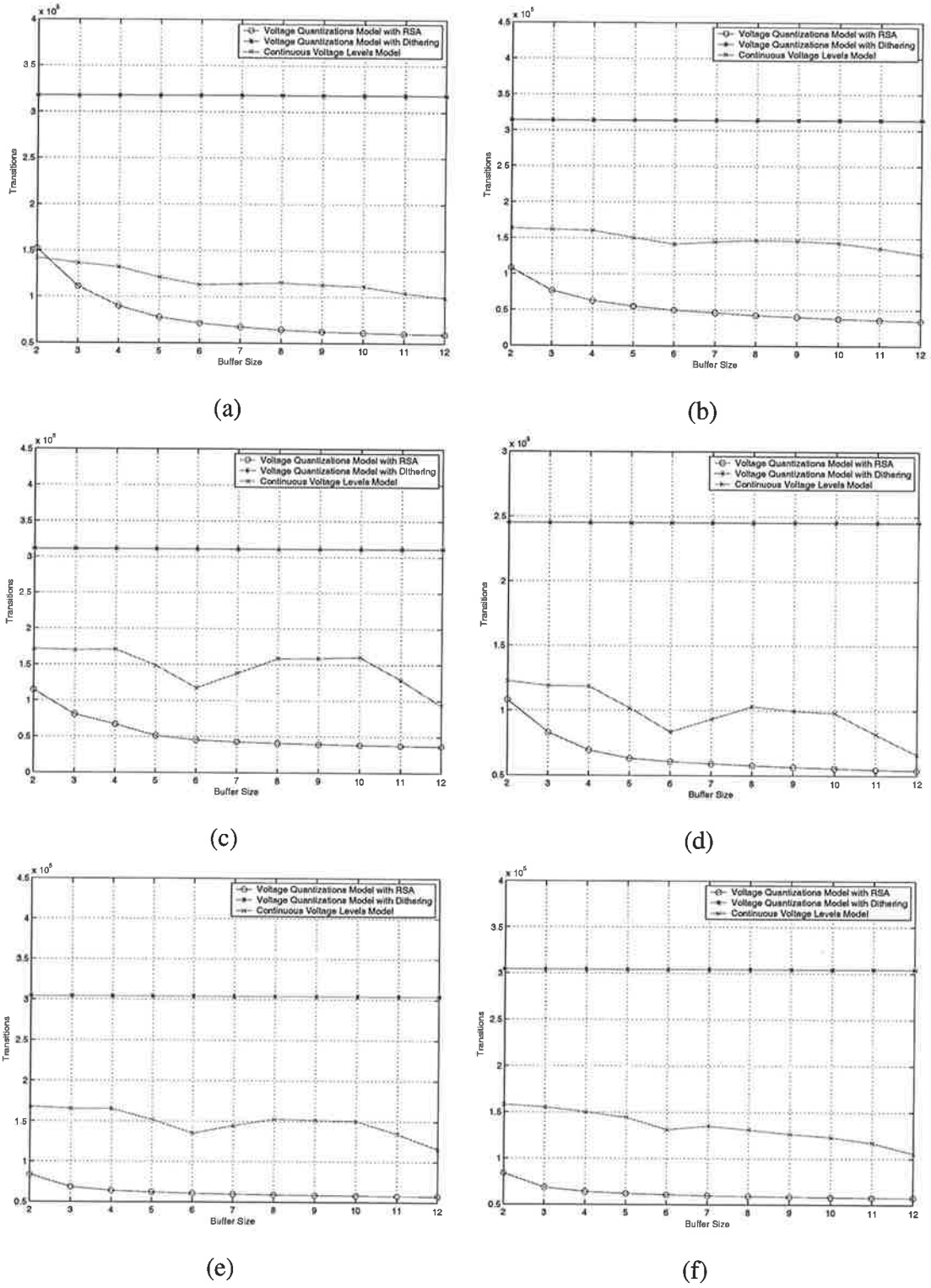
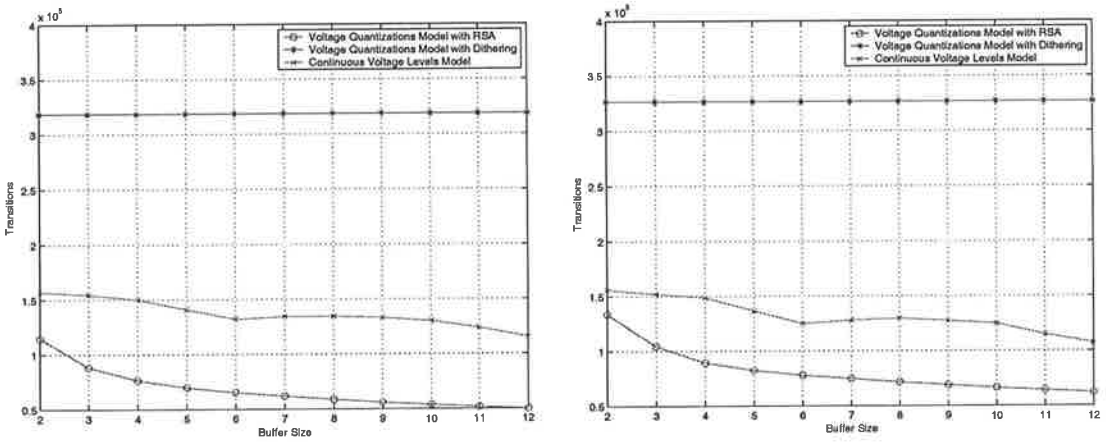


Figure B.11: Transition count comparison to prior work (versus buffer size). (a) Akiyo, (b) Carphone, (c) Coastguard, (d) Container (e) Foreman, (f) Hall



(g) (h)

Figure B.12: Transition count comparison to prior work (versus buffer size). (g) Mother, and (h) Silent

Appendix C

Source Code

The attached CDROM contains the C and Matlab code, video sequences, and MPEG codec used in developing and evaluation of the rate selection algorithm.

Bibliography

[AAN88] Y. Arai, T. Agui, and M. Nakajima. A Fast DCT-SQ Scheme for Images. *Transactions of the IEICE*, E71(11):1095–1097, November 1988.

[BB96] Thomas D. Burd and Robert W. Brodersen. Processor Design for Portable Systems. *Journal of VLSI Signal Processing*, 13(2/3):203–222, August 1996.

[BB00] Thomas D. Burd and Robert W. Brodersen. Design Issues for Dynamic Voltage Scaling. In *International Symposium on Low Power Electronics and Design*, August 2000.

[BBK+94] Kees Van Berkel, Ronan Burgess, Joep Kessels, Marly Roncken, and Frits Schalij. Asynchronous Circuits for Low Power: A DCC Error Corrector. *IEEE Design & Test of Computers*, 11(2):22–32, Summer 1994.

[Bel91] T. Bell. Incredible Shrinking Computers. In *IEEE Spectrum*, pages 37–43, May 1991.

[BK95] Vasudev Bhaskaran and Konstantinos Konstantinides. *Image and Video Compression Standards - Algorithms and Architectures*. Kluwer Academic Publishers, 1995.

[BMP97] Andy Bavier, Brady Montz, and Larry L. Peterson. Predicting MPEG Execution Times. Technical report, The University of Arizona, 1997.

[BPSB00] Thomas D. Burd, Trevor A. Pering, Anthony J. Stratakos, and Robert W. Brodersen. A Dynamic Voltage Scaled Microprocessor System. *IEEE Journal of Solid State Circuits*, 35(11):1571–1579, November 2000.

[Bur01] Thomas D. Burd. *Energy-Efficient Processor System Design*. PhD thesis, University of California, Berkeley, 2001.

[CB95a] Anantha P. Chandrakasan and Robert W. Brodersen. *Low Power Digital CMOS Design*. Kluwer Academic Publishers, 1995.

[CB95b] Anantha P. Chandrakasan and Robert W. Brodersen. Minimizing Power Consumption in Digital CMOS Circuits. *Proceedings of the IEEE*, 83(4):498–523, April 1995.

[CCL01] Lama H. Chandrasena, Priyadarshana Chandrasena, and Michael J. Liebelt. An Energy Efficient Rate Selection Algorithm for Voltage Quantized Dynamic Voltage Scaling. In *14th International Symposium on Systems Synthesis*, September 2001.

[CDCP01] Kihwan Choi, Karthik Dantu, Wei-Chung Chen, and Massoud Pedram. Frame-Based Dynamic Voltage and Frequency Scaling for a MPEG Decoder. In *International Conference of Parallel and Distributed Systems*, 2001.

[CGX96] Anantha Chandrakasan, Vadim Gutnik, and Thucydides Xanthopoulos. Data Driven Signal Processing: An Approach for Energy Efficient Computing. In *International Symposium on Low Power Electronics and Design*, August 1996.

[CJ92] T. Callaway and E. Swartzlander Jr. Optimizing Arithmetic Elements for Signal Processing. In *VLSI Signal Processing*, pages 91–100, IEEE Special Publications 1992.

[CL92] N. Cho and S. Lee. A Fast 4×4 DCT Algorithm for the Recursive 2-D DCT. *IEEE Transactions on Signal Processing*, 40(9):2166–2172, September 1992.

[CL00] Lama H. Chandrasena and Michael J. Liebelt. Energy Minimization in Dynamic Supply Voltage Scaling Systems using Data Dependent Voltage Level Selection. In *IEEE International Symposium on Circuits and Systems*, May 2000.

[CP87] K. Chu and D. Pulfreyn. A Comparison of CMOS Circuit Techniques: Differential Cascode Voltage Switch Logic Versus Conventional Logic. *IEEE Journal of Solid State Circuits*, pages 528–532, August 1987.

[CSB92] Anantha P. Chandrakasan, Samuel Sheng, and Robert W. Brodersen. Low-Power CMOS Digital Design. *IEEE Journal of Solid State Circuits*, 27(4):473–483, April 1992.

[CSF77] W. H. Chen, C. H. Smith, and S. C. Fralick. A Fast Computational Algorithm for the Discrete Cosine Transform. *IEEE Transactions on Communications*, COM-25(9):1004–1009, September 1977.

[CW94] K. Chao and D. Wong. Low Power Considerations in Floorplan Design. In *Int'l Workshop on Low Power Design*, pages 45–50, 1994.

[CWB01] A. Chandrakasan and F. Fox eds. W. Bowhill. *Design of High-Performance Microprocessor Circuits*. IEEE Press, 2001.

[Dah91] D. Dahle. Designing High Performance Systems to Run from 3.3V or Lower Sources. In *Sillicon Valley Personal Computer Conference*, pages 685–691, 1991.

[DCW+88] B. Davari, W.H. Chang, M.R. Wordeman, C.S. Oh, Y. Taur, K.E. Petrillo, D. Moy, J.J. Bucchignano, H.Y. Ng, M.G. Rosenfield, F.J. Hohn, and M.D. Rodriguez. A High Performance 0.25μm CMOS Technology. In *Int'l Electron Devices Meeting*, pages 56–59, 1988.

[Dev01] Advanced Micro Devices. AMD Athlon XP Processor Model 6 Data Sheet, November 2001.

[Eag92] J. Eager. Advances in Rechargeable Batteries Spark Product Innovation. In *Sillicon Valley Computer Conference*, pages 243–253, August 1992.

[FW92] E. Feig and S. Winograd. Fast Algorithms for the Discrete Cosine Transform. *IEEE Transactions on Signal Processing*, 40(9):2174–2193, September 1992.

[GBJ98] Michael K. Gowan, Larry L. Biro, and B. Jackson. Power Considerations in the Design of the Alpha 21264 Microprocessor. In *Design Automation Conference*, pages 726–731, 1998.

[GBL+98] J. D. Gibson, T. Berger, T. Lookabaugh, D. Lindbergh, and R. L. Baker. *Digital Compression for Multimedia.* Morgan Kaufmann, 1998.

[GCW95] K. Govil, E. Chan, and H. Wasserman. Comparing Algorithms for Dynamic Speed-Setting of a Low Power CPU. In *Mobicom*, November 1995.

[GH86] T. A. Grotjohn and B. Hoefflinger. Sample-Set Differential Logic (SSDL) for Complex High-Speed VLSI. *IEEE Journal of Solid State Circuits*, SC-21(2):367–369, April 1986.

[Gut96] V. Gutnik. Variable Supply Voltage for Low Power DSP. Master's thesis, Massachusetts Institute of Technology, 1996.

[HP93] V. Hirendu and M. Pedram. PCUBE: A Performance Driven Placement Algorithm for Low Power Designs. In *Euro-DAC Conference*, pages 72–77, 1993.

[Int] Intel Corporation. http://developer.intel.com/design/mobile/datashts/.

[JB90] Gordon M. Jacobs and Robert W. Brodersen. A Fully Asynchronous Digital Signal Processor Using Self-Timed Circuits. *IEEE Journal of Solid State Circuits*, 25(6):1526–1536, December 1990.

[KK90] M. Kakumu and M. Kinugawa. Power-Supply Voltage Impact on Circuit Performance for Half and Lower Submicrometer CMOS LSI. *IEEE Transactions on Electron Devices*, 37:1902–1908, August 1990.

[KSM+98] T. Kuroda, K. Suzuki, S. Mita, T. Fujita, F. Yamane, F. Sang, A. Chiba, Y. Watanabe, K. Matsuda, T. Maeda, T. Sakurai, and T. Fumyama. Variable Supply-Voltage Scheme for Low-Power High-Speed CMOS Digital Design. *IEEE Journal of Solid State Circuits*, 33(3):454–463, March 1998.

[KSN01] A. Keshavarzi and et. al. S. Narendra. Effectiveness of Reverse Body Bias for Leakage Control in Scaled Dual Vt CMOS ICs. In *ISLPED*, 2001.

[Liu88] S. L. Liu. Implementation of Iterative Networks with CMOS Differential Logic. *IEEE Journal of Solid State Circuits*, 23(4), August 1988.

[LM02] X. Liu and S. Mourad. Performance of Submicron CMOS Devices and Gates with Substrate Biasing. In *IEEE Int'l Symposium on Circuits and Systems*, 2002.

[LS93] Drake Liu and Christer Svensson. Trading Speed for Low Power by Choice of Supply and Threshold Voltages. *IEEE Journal of Solid State Circuits*, 28(1):10–17, January 1993.

[LV86] A. Lightenberg and M. Vitterli. A Discrete Fourier Cosine Transform Chip. *IEEE Journal on Selected Areas in Communications*, SAC-4(1):49–61, January 1986.

[LYyTC99] Oliver Yuk-Hang Leung, Chung-Wai Yue, Chi ying Tsui, and Roger S. K. Cheng. Reducing Power Consumption of Turbo Code Decoder using Adaptive Iteration with Variable Supply Voltage. In *International Symposium on Low Power Electronics and Design*, August 1999.

[Mat] MathWorks Inc. http://www.mathworks.com/.

[MFBM02] S. Martin, K. Flautner, D. Blaauw, and T. Mudge. Combined Dynamic Voltage Scaling and Adaptive Body Biasing for Optimal Power Consumption in Microprocessors under Dynamic Workloads. In *Int'l Conference on Computer Aided Design (ICCAD)*, 2002.

[MIS+99] H. Mizuno, K. Ishibashi, T. Shimura, T. Hattori, S. Narita, K. Shiozawa, S. Ikeda, and K. Uchiyama. A 18μA-Standby Current 1.8V 200MHz Microprocessor with Self Substrate-Biased Data-Retentive Mode. In *IEEE Int'l Solid-State Circuit Conference*, 1999.

[MKC02] M. Miyazaki, J. Kao, and A. Chandrakasan. A 175mV Multiply-Accumulate Unit using an Adaptive Supply Voltage and Body Bias Architecture. In *IEEE Int'l Solid-State Circuits Conference*, 2002.

[Moo] Moore's Law. http://www.intel.com/intel/museum/25anniv/hof/moore.htm.

[MPE] http://www.mpeg.org.

[MPFL96] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D.J. LeGall. *MPEG Video Compression Standard*. Chapman & Hall, 1996.

[MSS] http://www.bok.net/mpeg/MSSG/.

[MW92] L. McMillan and L. Westover. A Forward-Mapping Realization of the Inverse Discrete Cosine Transform. In *Data Compression Conference*, pages 219–228, March 1992.

[Nie97] Lars S. Nielsen. *Low-Power Asynchronous VLSI Design*. PhD thesis, Technical University of Denmark, 1997.

[NMH02] S. Narendra and et. al. M. Haycock. 1.1V 1GHz Communications Router with On-Chip Body Bias in 150nm CMOS. In *IEEE Int'l Solid-State Circuits Conference*, 2002.

[NNSvB94] Lars S. Nielsen, Cees Niessen, Jens Sparsø, and Kees van Berkel. Low-Power Operation Using Self-Timed Circuits and Adaptive Scaling of Supply Voltage. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2(4):391–397, December 1994.

[PBB98] T. Pering, T. Burd, and R. Brodersen. The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms. In *International Symposium on Low Power Electronics and Design*, August 1998.

[Per00] Trevor A. Pering. *Energy-Efficient Operating System Techniques*. PhD thesis, University of California, Berkeley, 2000.

[PLLS01] Johan Pouwelse, Koen Langendoen, Inald Lagendijk, and Henk Sips. Power-Aware Video Decoding. In *22nd Picture Coding Symposium*, 2001.

[PSS86] J. A. Pretorious, A. S. Shubhat, and C. A. T. Salama. Latched Domino CMOS Logic. *IEEE Journal of Solid State Circuits*, SC-21(4):514–522, August 1986.

[Rec] Rechargeable Battery/Systems for Communication/Electronic Applications: An Analysis of Technology Trends, Applications, and Projected Business Climate, NAT-IBO. http://www.dtic.mil/natibo/docs/BatryRpt.pdf.

[RP96] Jan M. Rabaey and Massoud Pedram. *Low Power Design Methodologies*. Kluwer Academic Publishers, 1996.

[Sin01] Amit Sinha. *Energy Efficient Operating Systems and Software*. PhD thesis, Massachusetts Institute of Technology, 2001.

[SMF+97] Kojiro Suzuki, Shinji Mita, Tetsuya Fujita, Fumiyuki Yamane, Fumihiko Sano, Akihiko Chiba, Yoshinori Watanabe, Koji Matsuda, Takeo Maeda, and Tadahiro Kuroda. A 300MIPS/W RISC Core Processor with Variable Supply-Voltage Scheme in Variable Threshold-Voltage CMOS . In *IEEE Custom Integrated Circuits Conference*, pages 587–590, 1997.

[SRP+95] Deo Singh, Jan M. Rabaey, Massoud Pedram, Francky Catthoor, Suresh Rajgopal, Naresh Sehgal, and Thomas J. Mozdzen. Power Conscious CAD Tools and Methodologies: A Perspective. *Proceedings of the IEEE*, 83(4):570–594, April 1995.

[SSB94] A. Stratakos, S. Sanders, and R. Brodersen. A Low-Voltage CMOS DC-DC Converter for a Portable Low-Powered Battery-Operated System. In *Power Electronics Specialists Conference*, 1994.

[STD94] C. Su, C. Tsui, and A. Despain. Low Power Architecture Design and Compilation Techniques for High-Performance Processors. In *COMPCON*, pages 489–498, February 1994.

[Str98] Anthony J. Stratakos. *High-Energy Low-Voltage DC-DC Conversion for Portable Applications*. PhD thesis, University of California, Berkeley, 1998.

[SYK01] Donghwan Son, Chansu Yu, and Heung-Nam Kim. Dynamic Voltage Scaling in MPEG Decoding. In *ICPADS*, pages 633–640, June 2001.

[TSR+98] Vivek Tiwari, Deo Singh, Suresh Rajgopal, Gaurav Mehta, Rakesh Patel, and Franklin Baez. Reducing Power in High-Performance Microprocessors. In *Design Automation Conference*, pages 732–737, 1998.

[Vee84] Harry J. M. Veendrick. Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits. *IEEE Journal of Solid State Circuits*, 19(4):468–473, August 1984.

[WH99] Gu-Yeon Wei and Mark Horowitz. A Fully Digital, Energy-Efficient Adaptive Power-Supply Regulator. *IEEE Journal of Solid State Circuits*, 34(4):520–528, April 1999.

[Wol90] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1990.

[WWDS94] M. Weiser, B. Welch, A. Demers, and S. Shenker. Scheduling for Reduced CPU Energy. In *OSDI*, November 1994.

[Xan99] Thucydides Xanthopoulos. *Low Power Data Dependent Transform Video and Still Image Coding*. PhD thesis, Massachusetts Institute of Technology, 1999.

[XCSD96] Thucydides Xanthopoulos, Anantha P. Chandrakasan, Charles G. Sodini, and William J. Dally. A Data-Driven IDCT Architecture for Low Power Video Applications. In *European Solid-State Circuits Conference*, 1996.

[YDS95] F. Yao, A. Demers, and S. Shenker. A Scheduling Model for Reduced CPU Energy. *IEEE Annual Foundations of Computer Science*, pages 374–382, 1995.