A Framework for Energy and Transient Power Reduction during Behavioral Synthesis

Saraju P. Mohanty, Student Member of IEEE and Nagarajan Ranganathan, Fellow of IEEE

***Abstract*— In battery driven portable applications, the minimization of energy, average power, peak power, and peak power differential are equally important to improve reliability and efficiency. The peak power and the peak power differential drive the transient characteristics of a CMOS circuit. In this paper, we propose a framework for simultaneous reduction of the energy and transient power during behavioral synthesis. A new metric called "Cycle Power Function" (CPF) is defined which captures the transient power characteristics as an equally weighted sum of the normalized mean cycle power and the normalized mean cycle differential power. Minimizing CPF using multiple supply voltages and dynamic frequency clocking under resource constraints results in the reduction of both energy and transient power. The cycle differential power can be modeled either as the absolute deviation from the average power or as the cycle-to-cycle power gradient. Based on the above, we develop a new datapath scheduling algorithm called CPF-scheduler which attempts at power and energy minimization by minimizing the CPF parameter by the scheduling process. The type and number of functional units available becomes the set of resource constraints for the scheduler. Experimental results indicate that the proposed scheduler achieves significant reductions in terms of power and energy.**

***Index Terms*— Peak power, peak power differential, average power, power fluctuation, multiple supply voltages, dynamic frequency clocking, low-power datapath scheduling.**

I. INTRODUCTION

Low power circuit design is a three dimensional problem involving area, performance and power trade-offs. Because of the decreasing feature size and increasing packing density, it may be possible to trade area versus power [1]. The decreasing feature size and the increasing clock frequency together result in high on-chip electric fields which has made reliability a big challenge for the designers [1], [2], [3]. The proliferation of the use of portable systems combined with other issues such as reliability, thermal considerations, and environmental concerns have driven the need for low power designs [2]. In low power designs for battery driven portable applications, the total energy, the average power, the peak power and the peak power differential are all equally important considerations. Both peak power and peak power differential drive the transient characteristics of the CMOS circuit. The life time and efficiency of battery is affected by all of the above parameters [4], [5], since higher the current (power) lesser the electrochemical conversion efficiency. The minimization of all the four power and energy parameters is crucial in designing efficient and reliable integrated circuits.

The three sources of power dissipation in a CMOS digital circuit are dynamic power (P_d), short-circuit power (P_{sc}) and static power (P_s) as summerized in Eqn. 1 below [1], [6] :

$$\begin{aligned} P_{total} &= P_d + P_{sc} + P_s \\ &= \alpha C V^2 f_{clk} + \tau \alpha V I_{sc} f_{clk} + V I_{leak} \end{aligned} \qquad (1)$$

where, α is the switching activity, C is the total capacitance seen at the gate output, V is the supply voltage, f_{clk} is the operating frequency, τ is the time for which short-circuit occurs, I_{sc} is the short-circuit current and I_{leak} is the leakage current. In [3], it is pointed out that there is an increase in both dynamic and static power in deep submicron and nanometer domains. It is well known that (i) by reducing supply voltage, both power and energy can be saved compromising delay, (ii) slowing down the circuit by reducing the clock frequency will save power but not energy, and finally, (iii) varying frequency as well as voltage in a coordinated manner could save both energy and power while maintaining performance at acceptable levels [7], [6], [8].

In this work, we use the concepts of multiple voltages and dynamic clocking [9], [10], [11] to achieve simultaneous minimization of energy and transient power during behavioral synthesis. We develop a new framework for simultaneous minimization of total energy and transient power during the synthesis of datapath circuits. The rest of the paper is organized as follows. The background and the various related works are discussed in the following section. The derivation of the CPF function based on the two models as well as the scheduling algorithm are presented next, followed by the experimental results and conclusions.

II. RELATED WORK

Several researchers have explored the use of multiple supply voltages for energy reduction during datapath synthesis. In [12], an ILP formulation and a heuristic for variable voltage scheduling is presented. An ILP-based scheduling algorithm called MOVER (Multiple Operating Voltage Energy Reduction) is discussed in [13] and another ILP-based algorithm called MESVS (Minimum Energy Schedule with Voltage Selection) is presented in [14]. The difference between the two algorithms is that MESVS can select a discrete set of voltages, whereas MOVER can select a continuous range of voltages. A dynamic programming technique for multiple supply voltage scheduling is discussed in [15]. A low power datapath synthesis system called SCALP is discussed in [16], which uses both voltage scaling and capacitance reduction.

S. P. Mohanty graduated with Ph. D. from Department of Computer Science and Engineering University of South Florida, Tampa, FL 33620. N. Ranganathan is with Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620. E-mail: smohanty@csee.usf.edu and ranganat@csee.usf.edu

A time constrained multiple voltage scheduling technique is proposed in [17]. A resource constrained scheduling algorithm with multiple supply voltages is given in [18] which helps in reducing power using multiple supply voltages. The authors in [19] propose a resource-constrained and a time-constrained instruction scheduling algorithms for low power pipelined functional units. In [20], resource and latency constrained list-based scheduling algorithms with multiple supply voltages are discussed. Scheduling algorithms with resource and time constraints based on the Lagrange multiplier technique are investigated in [21]. The above scheduling techniques consider various concepts such as single clock frequency, multiple supply voltages, voltage scaling, capacitance reduction, and switching activity reduction for minimizing either total energy or average power, but not both at the same time. Further, these works have not considered dynamic frequency clocking or transient power reduction.

Both the peak power and the peak power differential drive the transient power characteristics of a system. The earliest work on peak power reduction during simultaneous scheduling and assignment is reported in [22], in which power minimization achieved in one level (with SPICE) is used to optimize at behavioral level using genetic algorithms. In [23], ILP and force directed scheduling methods are explored for minimizing peak power under latency constraints. The formulations consider multicycling, pipelining and single supply voltage. ILP based models to minimize peak power and peak area have been proposed in [24] for latency constraint scheduling. The authors also introduce resource binding to minimize the amount of switching at the input of functional units. In [25], the authors describe a time constrained ILP scheduling algorithm for real time systems that minimizes both peak power and number of resources. In [26], the use of data monitor operations for simultaneous peak power and peak power differential reduction is addressed. The above works address only peak power issues and do not include energy minimization and only attempt to minimize any one of the four parameters. In [27], the authors introduce the use of "telescopic" units to improve the throughput. The telescopic units allow variation in the number of clock cycles for execution depending on the input data.

A SIMD linear array image processor design is discussed in [11] in which the modules of a circuit can be operated in different frequencies to improve the system performance. The concept of dynamic frequency clocking is introduced in [11]. A low power design using multiple clocking scheme is presented in [28]. If the overall effective frequency is f, then the circuit is partitioned into n different disjoint modules with each module operating at the frequency $(\frac{f}{n})$ indicating power savings of up to 50% compared to using a single frequency. The use of frequency scaling in an MPEG2 decoder design is described in [10]. In this system, the clock speed is increased if the load is high and the clock frequency is decreased if the load is small. A time constrained heuristic scheduling algorithm is discussed in [29] that uses both frequency and voltage scaling. Energy savings in the range of $33-75\%$ is reported, but power savings is not mentioned. Several system-level approaches [7], [8] have been investigated towards reducing power consumption in both general purpose and special purpose processors with the help of simultaneous voltage and frequency scaling.

Most works as discussed above address either average power or energy or peak power, but do not address all of the power parameters (average power, energy, peak power, and peak power differential) together. In this work, we describe a framework for simultaneous minimization of total energy, average power, peak power, and peak power differential. A new parameter called *Cycle Power Function* (CPF) is defined which is an equally weighted sum of normalized mean cycle power and normalized mean cycle differential power. Minimizing this parameter using multiple supply voltages (MV) and dynamic frequency clocking (DFC) results in the reduction of both energy and transient power. We investigate two different models for defining CPF. The cycle differential power is defined as the absolute deviation of the cycle power from the average power for any given cycle in the first model whereas it is defined as the cycle-to-cycle power gradient in the second. Further, the CPF models take into consideration the switching activity of the different functional units. A datapath scheduling algorithm (called, CPF-Scheduler) is proposed which attempts to minimize the CPF while keeping the time penalty at a minimum and using the concepts of dynamic frequency clocking and multiple supply voltages. The algorithm assumes different types and numbers of resources (such as, multipliers and ALUs) operating at different voltages and frequencies as resource constraints. The CPF-scheduling algorithm generates a parameter called *Cycle Frequency Index*, cfi for each control step which serves as the clock dividing factor for the *Dynamic Clocking Unit* (DCU) generates the different clock frequencies on the fly.

III. Cycle Power Function (CPF)

In this section, we introduce the different notations and terminology required for defining the cycle power function (CPF). The notations and terminology needed for the proposed models are given in Table I. The datapath is represented as a sequencing data flow graph (DFG).

The CPF is defined to consist of two main components: the normalized mean cycle power and the normalized mean cycle difference power. The normalized mean cycle power (P_{norm}) is the mean cycle power (P) normalized with respect to the peak power consumption (P_{peak}) of the DFG. The normalized mean cycle difference power (DP_{norm}) is the mean cycle difference power (DP) normalized with respect to the peak power differential of the DFG. The second component varies between the two models. The mean difference power is the mean of the cycle difference power DP_c over the control steps. In model 1, the cycle difference power DP_c is defined as the absolute deviation of the cycle power from the mean cycle power. Then, the mean cycle difference power DP is the mean deviation of the cycle power from the mean cycle power. On other hand, in model 2, the cycle difference power DP_c of a current cycle is modeled as the cycle-to-cycle power gradient. In other words, the cycle difference power DP_c of a current control step c is the difference (or gradient) of the current cycle

TABLE I
LIST OF NOTATIONS AND TERMINOLOGY USED FOR MODELING CPF

N	: total number of control steps in the DFG
O	: total number of operations in the DFG
c	: a control step or a clock cycle in the DFG
o_i	: any operation i, where $1 \leq i \leq O$,
P_c	: the total power consumption of all functional units active in control step c (cycle power consumption)
P_{peak}	: peak power consumption for the DFG
P	: mean or average power consumption of the DFG
P_{norm}	: normalised mean power consumption of the DFG
DP_c	: difference power for cycle c
DP_{peak}	: peak differential power consumption for the DFG
DP	: mean of the cycle difference powers $\forall c$ in DFG
DP_{norm}	: normalised mean of the mean cycle difference power
CPF	: cycle power function
$FU_{k,v}$	: any functional unit of type k at voltage level v
FU_i	: any functional unit $FU_{k,v}$ needed by o_i for its execution ($o_i \in FU_{k,v}$)
$FU_{i,c}$	: any functional unit FU_i active in control step c
R_c	: total number of functional units active in step c (same as the number of operations scheduled in c)
$\alpha_{i,c}$	: switching activity of resource $FU_{i,c}$
$V_{i,c}$	: operating voltage of resource $FU_{i,c}$
$C_{i,c}$	: load capacitance of resource $FU_{i,c}$
f_c	: frequency of control step c

power and the previous cycle power. This can be expressed mathematically as, $DP_c = P_c - P_{c-1}$ or $DP_{c+1} = P_{c+1} - P_c$. In this case, the mean cycle difference power DP is the mean difference (or the gradient).

A. Model 1 : CPF using absolute deviation

For a set of n observations, $x_1, x_2, ..., x_n$ from a given distribution, the sample mean (which is an unbiased estimator for the population mean, μ) is $m = \frac{1}{n}\sum_{i=1}^{n} x_i$. The absolute deviation of these observations is defined as $\Delta x_i = |x_i - m|$. The mean deviation of the observations is given by $MD = \frac{1}{n}\sum_{i=1}^{n} |x_i - m|$. In this case, we model the cycle difference power DP_c as the absolute deviation of cycle power P_c from the mean cycle power P. Similarly, the mean difference power DP is modeled as mean deviation of the cycle power P_c. The mean cycle power P is an unbiased estimate of the average power consumption of the DFG.

The power consumption for any control step c is given by Eqn. 2. This is the total power consumption of all functional units active in control step c. This also includes the power consumption of the level converters where the level converters are considered as resources operating in a cycle c, if the current resource is driven by a resource operating at lower voltage.

$$P_c = \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c \tag{2}$$

The peak power consumption of the DFG is the maximum power consumption over all the N control steps which can be expressed as below.

$$\begin{aligned} P_{peak} &= max\left(P_c\right)_{\forall c=1,2,....N} \\ &= max\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right)_{\forall c=1,2,....N} \end{aligned} \tag{3}$$

The mean cycle power consumption of the DFG is defined as,

$$P = \frac{1}{N}\sum_{c=1}^{N} P_c = \frac{1}{N}\sum_{c=1}^{N}\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right) \tag{4}$$

The mean cycle power P is an unbiased estimate of the average power consumption of the DFG. The true average power consumption of the DFG is the total energy consumption of the DFG per clock cycle or per second. The normalised mean cycle power (P_{norm}) is obtained by dividing P by maximum cycle power (P_{peak}).

$$P_{norm} = \frac{P}{P_{peak}} = \frac{\frac{1}{N}\sum_{c=1}^{N}\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c}{max\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right)_{\forall c=1,2,....N}} \tag{5}$$

Thus, the normalised mean cycle power (P_{norm}) is an unitless quantitity in the range [0,1].

The cycle difference power (DP_c) for any control step can be defined as follows. This is the absolute deviation of the cycle power from the mean cycle power consumption of the DFG. This is a measure of the cycle power fluctuation of the DFG.

$$\begin{aligned} DP_c &= |P - P_c| \\ &= \left|\frac{1}{N}\sum_{c=1}^{N}\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right) \right. \\ &\quad \left. - \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right| \end{aligned} \tag{6}$$

The peak differential power which characterizes the maximum power fluctuation of the DFG is given by (DP_{peak}). This characterizes the maximum power fluctuation or the transient of the DFG over the entire set of control steps.

$$\begin{aligned} DP_{peak} &= max\left(|P - P_c|\right)_{\forall c=1,2,....N} \\ &= max\left(\left|\frac{1}{N}\sum_{c=1}^{N}\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right) \right.\right. \\ &\quad \left.\left. - \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right|\right)_{\forall c=1,2,...N} \end{aligned} \tag{7}$$

The mean cycle difference power (DP) is calculated as the sample mean of DP_c. This is a measure of the power spread or distribution of the cycle power over all control steps of the DFG.

$$\begin{aligned} DP &= \frac{1}{N}\sum_{c=1}^{N} DP_c = \frac{1}{N}\sum_{c=1}^{N} |P - P_c| \\ &= \frac{1}{N}\sum_{c=1}^{N}\left(\left|\frac{1}{N}\sum_{c=1}^{N}\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right) \right.\right. \\ &\quad \left.\left. - \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right|\right) \end{aligned} \tag{8}$$

The normalised mean cycle difference power (DP_{norm}) can be written as given below.

$$DP_{norm} = \frac{DP}{DP_{peak}} = \frac{\frac{1}{N}\sum_{c=1}^{N}\left(\left|\frac{1}{N}\sum_{c=1}^{N}\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right) - \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right|\right)}{max\left(\left|\frac{1}{N}\sum_{c=1}^{N}\left(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right) - \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\right|\right)_{\forall c}} \tag{9}$$

The above normalised mean cycle difference power DP_{norm} is a unitless quantity in the range [0,1].

The cycle power function CPF which is modeled as the equally weighted sum of the normalized mean cycle power (P_{norm}) and the normalized mean cycle difference power (DP_{norm}) is given below.

$$CPF(P_{norm}, DP_{norm}) = P_{norm} + DP_{norm} \tag{10}$$

Thus, the CPF will have a value in the range [0,2]. The CPF can be impacted by various constraints, including the resource constraints. In terms of peak cycle power (P_{peak}) and peak cycle difference power (DP_{peak}), the CPF can be expressed as :

$$CPF = \frac{P}{P_{peak}} + \frac{DP}{DP_{peak}} = \frac{\frac{1}{N}\sum_{c=1}^{N} P_c}{P_{peak}} + \frac{\frac{1}{N}\sum_{c=1}^{N} |P - P_c|}{DP_{peak}} \quad (11)$$

Using Eqn. 5 and 9, the cycle power function (CPF) can be written as follows.

$$CPF = \frac{\frac{1}{N}\sum_{c=1}^{N}\sum_{i=1}^{R_c} \alpha_{i,c}C_{i,c}V_{i,c}^2 f_c}{max\left(\sum_{i=1}^{R_c} \alpha_{i,c}C_{i,c}V_{i,c}^2 f_c\right)_{\forall c}} + \frac{\frac{1}{N}\sum_{c=1}^{N}\left(\left|\frac{1}{N}\sum_{c=1}^{N}\left(\sum_{i=1}^{R_c} \alpha_{i,c}C_{i,c}V_{i,c}^2 f_c\right) - \sum_{i=1}^{R_c} \alpha_{i,c}C_{i,c}V_{i,c}^2 f_c\right|\right)}{max\left(\left|\frac{1}{N}\sum_{c=1}^{N}\left(\sum_{i=1}^{R_c} \alpha_{i,c}C_{i,c}V_{i,c}^2 f_c\right) - \sum_{i=1}^{R_c} \alpha_{i,c}C_{i,c}V_{i,c}^2 f_c\right|\right)_{\forall c}} \quad (12)$$

B. Model 2 : CPF using cycle-to-cycle gradient

For a set $x_1, x_2, ..., x_n$ of n observations from a given distribution, the observation-to-observation gradient can be defined as, $|x_{i+1} - x_i|$, where $1 \leq i \leq n-1$. The mean gradient is given by $\frac{1}{n-1}\sum_{i=1}^{n-1} |x_{i+1} - x_i|$. It should be noted that there are $n-1$ gradients for n observations. In this case, we model the cycle difference power DP_c as the cycle-to-cycle power gradient and the mean difference power DP as the mean gradient. The models for the mean cycle power or the average power (Eqn. 2 - 4) remains the same as before.

The cycle difference power (DP_c) for any control step is defined as the difference in the power consumption of the current to the previous control step, as given below.

$$\begin{aligned} DP_{c+1} &= |P_{c+1} - P_c| \\ &= \left|\sum_{i=1}^{R_{c+1}} \alpha_{i,c+1}C_{i,c+1}V_{i,c+1}^2 f_{c+1} \right. \\ &\quad \left. - \sum_{i=1}^{R_c} \alpha_{i,c}C_{i,c}V_{i,c}^2 f_c\right| \end{aligned} \quad (13)$$

The peak differential power is characterized by (DP_{peak}) :

$$\begin{aligned} DP_{peak} &= max\left(|P_{c+1} - P_c|\right)_{\forall c=1,2,....N-1} \\ &= max\left(\left|\sum_{i=1}^{R_{c+1}} \alpha_{i,c+1}C_{i,c+1}V_{i,c+1}^2 f_{c+1} \right.\right. \\ &\quad \left.\left. - \sum_{i=1}^{R_c} \alpha_{i,c}C_{i,c}V_{i,c}^2 f_c\right|\right)_{\forall c=1,2,...N-1} \end{aligned} \quad (14)$$

The mean cycle difference power (DP) is calculated as,

$$\begin{aligned} DP &= \frac{1}{N-1}\sum_{c=1}^{N-1} DP_{c+1} \\ &= \frac{1}{N-1}\sum_{c=1}^{N-1} |P_{c+1} - P_c| \\ &= \frac{1}{N-1}\sum_{c=1}^{N-1}\left(\left|\sum_{i=1}^{R_{c+1}} \alpha_{i,c+1}C_{i,c+1}V_{i,c+1}^2 f_{c+1} \right.\right. \\ &\quad \left.\left. - \sum_{i=1}^{R_c} \alpha_{i,c}C_{i,c}V_{i,c}^2 f_c\right|\right) \end{aligned} \quad (15)$$

Using Eqn. 5, 14 and 15, the cycle power function (CPF) can be written as follows.

$$\begin{aligned} CPF &= P_{norm} + DP_{norm} \\ &= \frac{P}{P_{peak}} + \frac{DP}{DP_{peak}} \\ &= \frac{\frac{1}{N}\sum_{c=1}^{N} P_c}{P_{peak}} + \frac{\frac{1}{N-1}\sum_{c=1}^{N-1} |P_{c+1} - P_c|}{DP_{peak}} \end{aligned} \quad (16)$$

The power models expressed in equations 16 and 12 use generic parameters, such as $\alpha_{i,c}, C_{i,c}, V_{i,c}$ and f_c to keep the CPF definition independent of any specific energy or power models. It can accomodate both the look-up table based energy (power) models and energy (power) macro-models. The generic model can also help in easy integration of the CPF model in a behavioral synthesis tool that uses both behavioral power estimator and datapath scheduler. Using the dynamic energy model proposed in [15], we can express the effective switching capacitance of our proposed model as,

$$\alpha_i C_i = C_{sw\,i}(\alpha_i{}^1, \alpha_i{}^2) \quad (17)$$

Here, the α_i and C_i are the parameters corresponding to the functional unit FU_i. The $C_{sw\,i}$ is a measure of the effective switching capacitance of resource (functional unit) FU_i, which is a function of $\alpha_i{}^1$ and $\alpha_i{}^2$; where $\alpha_i{}^1$ and $\alpha_i{}^2$ are the average switching activity values on the first and second input operands of resource FU_i. It should be noted that the above switching model (in Eqn. 17) handles input pattern dependencies. The CPF model can be easily modified for different modes of operation of the datapath circuit: (i) single supply voltage and single frequency, (ii) multiple supply voltages and single frequency, (iii) multiple supply voltages and dynamic frequency and (iv) multiple supply voltage and multicycling. For example, for single supply voltage and single frequency scheme, $V_{i,c}$ and f_c are same for all c, for multiple supply voltage and multicycling f_c is same for all c. Using Eqn. 17 we rewrite Eqn. 12 as,

$$CPF = \frac{\frac{1}{N}\sum_{c=1}^{N}\sum_{i=1}^{R_c} C_{sw\,i,c}V_{i,c}^2 f_c}{max\left(\sum_{i=1}^{R_c} C_{sw\,i,c}V_{i,c}^2 f_c\right)_{\forall c}} + \frac{\frac{1}{N}\sum_{c=1}^{N}\left(\left|\frac{1}{N}\sum_{c=1}^{N}\left(\sum_{i=1}^{R_c} C_{sw\,i,c}V_{i,c}^2 f_c\right) - \sum_{i=1}^{R_c} C_{sw\,i,c}V_{i,c}^2 f_c\right|\right)}{max\left(\left|\frac{1}{N}\sum_{c=1}^{N}\left(\sum_{i=1}^{R_c} C_{sw\,i,c}V_{i,c}^2 f_c\right) - \sum_{i=1}^{R_c} C_{sw\,i,c}V_{i,c}^2 f_c\right|\right)_{\forall c}} \quad (18)$$

Using Eqn. 17 we can derive a similar expression for Eqn. 16. The notation $C_{sw i,c}$ represents $C_{sw\,i}$ for the functional unit FU_i active in control step c. A look-up table constructed to store the $C_{sw\,i}$ values for different combinations of ($\alpha_i{}^1$ and $\alpha_i{}^2$) for different types of functional units, such as multipliers and ALUs. We use interpolation technique to determine the $C_{sw\,i}$ values for the ($\alpha_i{}^1$ and $\alpha_i{}^2$) combinations that are not available in the look-up table. The size of the look-up table impact the accuracy of the results; larger the size better is the accuracy.

Minimization of CPF : CPF is used as the objective function for low power datapath scheduling. From the above equations, we make the following observations about the cycle power function (CPF). The CPF is a *non-linear* function. It is a function of four parameters, such as average power (P), peak power (P_{peak}), average difference power (DP) and peak difference power (DP_{peak}). Each of the above power parameters are dependent on switching activity, capacitance, operating voltage and operating frequency. The absolute function (abs or $|\ |$) in the numerator (of Eqn. 12 or 16) contributes to the nonlinearity. The complex behavior of the function is also contributed by the denominator parameters, P_{peak} and DP_{peak}. A fractional function can be minimized by decreasing the numerator or by increasing the denominator. But, we are aiming at minimizing both the numerator and denominator of the above fractional form objective function. So, we need to use specific approach to minimize CPF, such that both the numerator and denominator are minimized. We have to put constraints on the denominators of the function and minimize the overall function. The constraints on the denominators can be imposed through the use of resource constraints. *Thus, we conclude that the minimization of CPF using multiple supply voltages, dynamic frequency clocking and multicycling under resource constraints will lead to the reduction of energy and power parameters.*

IV. CPF-Scheduler Algorithm

In this section, we develop a scheduling algorithm that minimizes the objective functions using multiple voltages and dynamic clocking to reduce energy and the power. We assume the availability of different functional units operating at different supply voltages. In dynamic frequency clocking or frequency scaling, all the units are clocked by a single clock line which can switch frequencies at run-time [11], [9], [10]. In such systems, a dynamic clocking unit (DCU) generates different clocks using a clock dividing strategy. It should be noted that frequency scaling helps in reducing power, but not energy. Moreover, the frequency reduction facilitates the operations of the different functional units at different voltages, which in turn helps in energy reduction.

The target architecture model assumed for the scheduling is from [13]. Each functional unit is associated with a register and a multiplexor. The register and the multiplexor will operate at the same voltage level as that of the functional units. Level converters are used when a low-voltage functional unit is driving a high-voltage functional unit [13], [20]. A controller decides which of the functional units are active in each control step and those that are not active are disabled using the multiplexors. The controller will have a storage unit to store the cycle frequency index (cfi_c) values obtained from the scheduling, used as the clock dividing factor for the dynamic clocking unit. The cycle frequency f_c is generated dynamically and a corresponding functional unit is activated.

The delay for a control step is dependent on the delays of the functional units (d_{FU}), multiplexor (d_{Mux}), register (d_{Reg}) and level converters (d_{Conv}) as expressed in following equation.

$$d_c = d_{FU} + d_{Mux} + d_{Reg} + d_{Conv} \tag{19}$$

where, d_c is the delay of control step c, d_{FU} is the delay of the slowest FU in the control step c and the register delays include the set-up and propagation delays. Using the above delay model, the worst case delays of the library components are estimated. For a given base frequency (f_{base}), maximum frequencies of each FU are scaled down to operating frequencies (f_c). These parameters are determined as follows :

$$\begin{aligned} f_{base} &= \left\lfloor \frac{\lfloor 1/d_c^{min} \rfloor}{2^{L_f}} \right\rfloor 2^{L_f} \\ cfi_c &= \left\lceil \frac{\lfloor d_c/d_c^{min} \rfloor}{2^n} \right\rceil 2^n \\ f_c &= \frac{f_{base}}{cfi_c} \end{aligned} \tag{20}$$

where, d_c^{min} is the minimum of the control step delays and L_f is the number of allowable frequencies. The value of n is chosen in such a way that cfi_c is closest value greater than or equal to $\lceil \frac{d_c}{d_c^{min}} \rceil$.

The inputs to the algorithm are an unscheduled data flow graph (UDFG), the resource constraints, the number of allowable voltage levels (L_V), the number of allowable frequencies (L_f), delay of each resource (d_{FU}), multiplexor (d_{Mux}), register (d_{Reg}) at different voltage levels. The delays of level converters (d_{Conv}) are represented in the form of a matrix that shows the delay for converting one voltage level V_i to another voltage level V_j (where, both $V_i, V_j \in V_{L_V}$). The resource constraint includes the number of ALUs and multipliers at different voltage levels V_i (where, $V_i \in V_{L_V}$). The scheduling algorithm determines the proper time stamp for each operation, f_{base}, cfi_c and the voltage level such that CPF as well as the time penalty is minimum. To reduce the time penalty, the lesser energy consuming resources are used at as maximum frequency as possible.

The CPF-Scheduler : The flow of the proposed algorithm is outlined in Fig. 1. In step 1, the switching activities at the inputs of each node of the DFG are determined. For this purpose, different sets of application specific input vectors (having different correlations) are given at the primary inputs of the DFG and the average switching activity at each node is calculated. In step 2, the scheduler constructs a look-up table with effective switching capacitance and the average switching activity pair as described in Eqn. 17. If the look-up table is large enough to contain the switching capacitance for all estimated average switching activities is step 1, then the power model accuracy is the highest. The algorithm determines the as-soon-as-possible (ASAP) and the as-late-as-possible (ALAP) schedules for the UDFG in step 3. The ASAP schedule is unconstrained and the ALAP schedule uses the number of clock steps found in the ASAP schedule as the latency constraint. In step 4, the number of resources of each type and voltage levels is determined. For example, if the resource constraint is 1 multiplier at $2.4V$, 2 multipliers at $3.3V$, 2 ALUs at $2.4V$ and 3 ALUs at $3.3V$, then the relaxed voltage initial resource constraint is found out to be 3 multipliers and 5 ALUs. In step 5, the scheduler uses the above relaxed voltage resource constraints and modifies the ASAP and ALAP schedules to take into account the resource constraints. This helps in restricting the mobility of vertices to a great extent and reducing the solution search space for the heuristic. Due to the resource constraints the number of control steps of modified ASAP and modified ALAP may be different from that of the ASAP and ALAP schedule in step 3. In step 6, the scheduler fixes the total number of control steps of the schedule which is the maximum of the control steps of the modified ASAP or modified ALAP in step 5. In step 7, the vertices are marked as having zero mobility or non-zero mobility. The zero mobility vertices are those having same modified ASAP time stamp and modified ALAP time stamp, and non-zero mobility vertices are those having different modified ASAP and modified ALAP time stamp. On determining the vertices having zero mobility and vertices having non-zero mobility, proper time stamp and operating voltage for mobile vertices, and operating voltages for non-mobile vertices are found out. Further, operating clock

Input	: UDFG, resource constraints, L_V, L_f, all $V_i \in V_{L_V}$, d_{FU}, d_{Mux}, d_{Reg}, d_{Conv}
Output	: scheduled DFG, f_{base}, N, cfi_c, power, energy and delay estimates
Step 1	: Calculate the switching activity at the inputs of each node of the DFG.
Step 2	: Construct a look-up table of effective switching capacitance, switching activity pairs.
Step 3	: Find ASAP and ALAP schedules of the UDFG.
Step 4	: Determine the number of multipliers and ALUs at different operating voltages.
Step 5	: Modify both ASAP and ALAP schedules obtained in Step 1 using the number of resources found in Step 2 as initial resource constraint.
Step 6	: Calculate the total number of control steps as the maximum of ASAP and ALAP schedules from Step 5.
Step 7	: Find the vertices having non-zero mobility and vertices with zero mobility.
Step 8	: Use the CPF-Scheduler-Heuristics to assign the time stamp and operating voltage for the vertices, and the cycle frequencies such that CPF and time penalty are minimum.
Step 9	: Find base frequency f_{base} and cycle frequency index cfi_c.
Step 10	: Calculate power, energy and delay details.

Fig. 1. The CPF-Scheduler algorithm flow

frequencies are established such that the CPF as well as the time penalty is minimum. The CPF-Scheduler uses an heuristic algorithm for the same. In step 9, the scheduler determines the base frequency (f_{base}) and cycle frequency index (cfi_c) using Eqn. 20. In step 10, the scheduler calculates the peak power, average power, peak power differential, energy estimates of the scheuled DFG and also the critical path delay.

The CPF-Scheduler Heuristic : Fig. 2 shows the heuristic algorithm used by the CPF-Scheduler. The inputs to the CPF-Scheduler heuristic are modified ASAP time stamp of each vertex (S_i), the modified ALAP time stamp of each vertex (E_i), the resource constraints, the number of allowable voltage levels (L_V), the number of allowable frequencies (L_f). Delay of each functional unit (d_{FU}), multiplexor (d_{Mux}), register (d_{Reg}) at different voltage levels are also given as inputs. Delays of level converters (d_{Conv}) is represented in the form of a matrix. The heuristic has to find time stamp c (in the range $[S_i, E_i]$) and operating voltage $V_{i,c}$ for each vertex v_i with operation o_i. The aim of the heuristic is to minimize CPF while keeping time penalty at a minimum. The heuristic minimized time ratio R_T alongwith CPF to minimize the time penalty. The time ratio (R_T) is defined as the ratio between the critical path delay when the vertices of the DFG are operating at multiple voltage (T_D) and when each of the vertices of the DFG is operated at the highest voltage. Expressing mathematically, $R_T = \frac{T_D}{T_S}$. These two objectives, minimization of CPF (minimization of energy and power) and minimization of time penalty are mutually conflicting. This is due to the fact that if operating voltage is reduced to minimize energy or power consumption this results in increase of critical path delay and hence increase of time penalty. The heuristic operates the energy hungry functional units at the highest possible voltage (frequency) and the less energy consuming functional units at lowest voltage (frequency) to achieve the simultaneous minimization of the mutually conflicting objectives. The heuristic fixes the operating voltages of the non-mobile vertices as per this order depending on the types of resource they need. The heuristic attempts to find suitable time stamp and operating voltage for the mobile vertices using exhaustive search. The mobile-vertices are attempted to be placed in each of the time stamps within their mobile range ($[S_i, E_i]$), when each placement and voltage assignment is done, the CPF and R_T value is calculated. The predecessor and successor time stamps are adjusted accordingly to maintain the precedence. For this purpose the heuristic maintains a matrix of dimension ($N * |k| V_{L_V}$) having number of resources of different types (k) as entries rowwise over all control steps. The $|k|$ is the type of resources available, for example, if only multiplier and ALUs are the available resources then the $|k| = 2$. If a voltage is assigned for a vertex, then the matrix entry of the corresponding type and operating voltage is decremented. A particular vertex is placed in a cycle for which the sum of CPF and R_T is minimum. The heuristic, initially assumes the modified ASAP schedule (with relaxed voltage resource constrained) as the current schedule (line 01). In case a vertex is a multiplication operation, then the initial voltage assignment is the minimum available operating depending on the number of multipliers, whereas, for ALU operations vertex, it is the maximum available operating voltage (line 04-08). Then the CPF and R_T value for the current schedule is calculated (line 09 and line 10). The heuristic finds CPF (and R_T) values for each allowable control step of each mobile vertices and for each available operating voltages denoted as TempCPF (and TempR_T) (line 17-20). The statement in line 17 adjusts the current schedule by adjusting the time stamps of successor vertices while maintaining the resource constraint (using the matrix) and guaranting that the precedence is satisfied. In line 12, the vertices are visited in ASAP manner. Another possible way of visiting the mobile vertices is to prioritise them in some manner, say vertex with lower mobility is visited first. The heuristic fixes the time step and operating voltage for a vertex and hence cycle frequency for which $CPF + R_T$ is minimum (line 22-26). For CPF computation the heuristics uses $\frac{1}{d_c}$ as a temporary measure for f_c. The above steps are repeated until all mobile vertices are time stamped.

Time complexity of CPF-Scheduler Heuristic : Let there be $|V|$ number of vertices in the DFG, out of which $|V_m|$ number of vertices have mobility and the maximum mobility of any mobile vertex is t_m. It should be noted that the total number

CPF-Scheduler-Heuristic

```
(01) Initialize CurrentSchedule as modified ASAPSchedule ;
(02) while( all mobile vertices are not time stamped ) do
(03) {
(04)     for the CurrentSchedule
(05)     {
(06)         if ( v_i is a multiplication ) then find the lowest available voltage for multipliers;
(07)         if ( v_i is add/sub/comparison ) then find the highest available operating voltage for ALUs;
(08)     } /* end for (04) */
(09)     Find CPF for CurrentSchedule and denote is as CurrentCPF ;
(10)     Find R_T for CurrentSchedule and denote is as CurrentR_T;
(11)     Maximum = −∞ ;
(12)     for each mobile vertex v_i
(13)     {
(14)         c1 = CurrentSchedule[v_i]; c2 = ALAPSchedule[v_i];
(15)         for c = c1 to c2 in steps of 1
(16)         {
(17)             Find a TempSchedule by adjusting CurrentSchedule in which v_i is scheduled in step c ;
(18)             Find next higher operating voltage for multipliers (next lower for ALUs) vertex for the TempSchedule;
(19)             Find CPF for TempSchedule, denoted by TempCPF ;
(20)             Find R_T for TempSchedule, denoted TempR_T;
(21)             Difference = (CurrentCPF + CurrentR_T) − (TempCPF + TempR_T) ;
(22)             if ( Difference > Maximum ) then
(23)             {
(24)                 Maximum = Difference ; CurrentVertex = v_i ; CurrentCycle = c ;
(25)                 CurrentVoltage = Operating voltage of vertex v_i;
(26)             } /* end if (22) */
(27)         } /* end for (15) */
(28)     } /* end for (12) */
(29)     Adjust CurrentSchedule to accomodate CurrentVertex in CurrentCycle operating at voltage assigned above ;
(30) } /* end while (02) */
```

Fig. 2. The CPF-Scheduler algorithm heuristic

of vertices in the DFG is total number of operations in DFG and the total number of NO-OPs. The running time of finding an operating voltage from the matrix for particular type of operation is $O(L_V)$. The statements from line 04-08 have running time of $\Theta(|V|L_V)$. The worst case running time of the statement in line 17 (or line 29) that adjusts the current schedule is $O(|V_m|)$. The running time of the code segment between line 17-26 is $O(|V_m|) + O(L_V) + \Theta(|V|) + \Theta(|V|)$, which is $\Theta(|V|)$, since it is always true that $|V_m|, L_V < |V|$. So, the running time of the code segment from line 15-27 is $\Theta(t_m|V|)$. Thus, the running time of the code segment line 12-28 is $\Theta(t_m|V_m||V|)$. The other statements of the pseudocode have constant running time. So, the running time or time complexity of the code segment in line 03-29 is $\Theta(|V||L_V|)+\Theta(t_m|V_m||V|)+O(|V_m|)$. This can be simplified to an weak upper bound on worst case running of the code segment (line 03-29) under the assumption that $|V_m| \approx |V|$, but in practice $|V_m| << |V|$. Under the above assumption we conclude that the worst case upper bound on the running time of the code segement in line 03-29 is $\Theta(t_m|V|^2)$. Considering the *while* loop in line 02 the overall running time of the algorithm can be written as $\Theta(t_m|V|^2|V_m|)$. Again under the assumption that $|V_m| \approx |V|$, we conclude that the worst case upper bound on the running time of the algorithm is $\Theta(t_m|V|^3)$. In other words, *the heuristic runs in time cubic to the number of vertices in the DFG. It can be noted that the time complexity of the algorithm is independent of the number of operating voltage levels.*

V. Experimental Results

The CPF-Scheduler algorithm was implemented in C and tested with selected benchmark circuits. The benchmarks used are :

(1) Auto-Regressive filter (ARF) (total 28 nodes, 16*, 12+, 40 edges).
(2) Band-Pass filter (BPF) (total 29 nodes, 10*, 10+, 9-, 40 edges).
(3) DCT filter (total 42 nodes, 13*, 29+, 68 edges).
(4) Elliptic-Wave filter (EWF) (total 34 nodes, 8*, 26+, 53 edges).
(5) FIR filter (total 23 nodes, 8*, 15+, 32 edges).
(6) HAL differential equation solver (total 11 nodes, 6*, 2+, 2-, 1<, 16 edges).

Our algorithm can handle large DFGs and find solutions in reasonable time. The parameters used to express our experimental results are shown in Table II. The look-up table construction consists of two phases, such as input pattern generation and cell characterization. We generate the primary input signals of different correlations and perform the characterization of the physical implementations of the library modules available in [29].

TABLE II
NOTATIONS USED TO EXPRESS THE RESULTS

E_S	: total energy consumption assuming single frequency and single supply voltage
E_D	: total energy consumption for dynamic frequency clocking and multiple supply voltage
P_{pS}	: peak power consumption for any cycle assuming single frequency and single supply voltage
P_{pD}	: peak power consumption for any cycle for dynamic frequency clocking and multiple supply voltage
P_{mS}	: minimum power consumption for any cycle assuming single frequency and single supply voltage
P_{mD}	: minimum power consumption for any cycle for dynamic frequency clocking and multiple supply voltage
T_S	: execution time assuming single frequency
T_D	: execution time assuming dynamic frequency
ΔE	: total energy reduction $= \frac{E_S - E_D}{E_S}$
ΔP	: average power reduction $= \frac{(E_S/T_S)-(E_D/T_D)}{(E_S/T_S)}$
ΔP_p	: peak power reduction $= \frac{P_{pS} - P_{pD}}{P_{pS}}$
ΔDP	: differential power reduction $= \frac{(P_{pS}-P_{mS})-(P_{pD}-P_{mD})}{(P_{pS}-P_{mS})}$
R_T	: time ratio $= \frac{T_D}{T_S}$

Our first set of experiments were carried out for the CPF model 1 (Eqn. 18) in which the cycle difference power is based on the absolute deviation. We tested the scheduling algorithm using the following sets of resource constraints (RC1, RC2, RC3, RC4) :

(1) Number of multipliers : 1 at $2.4V$; Number of ALUs : 1 at $3.3V$,
(2) Number of multipliers : 2 at $2.4V$; Number of ALUs : 1 at $3.3V$,
(3) Number of multipliers : 2 at $2.4V$; Number of ALUs : 1 at $2.4V$ and 1 at $3.3V$, and
(4) Number of multipliers : 1 at $2.4V$ and 1 at $3.3V$; Number of ALUs : 1 at $2.4V$ and 1 at $3.3V$.

The sets of resource constraints were chosen so as to cover resources at different operating voltages. The number of allowable voltage levels was assumed to be two $(2.4V, 3.3V)$ and maximum number of allowable frequencies are three. The CPF-scheduler determines the frequencies, in this case they are $4.5MHz$, $9.0MHz$, and $18.0MHz$. The experimental results for different benchmarks are shown in Table III for different resource constraints. The results take into account the power or energy consumptions in overheads, such as level converters and dynamic clocking unit. This indicates that the scheduling scheme could achieve significant reductions in peak power, peak power differential, average power and total energy with reasonable time penalties. The time penalty for the ARF and HAL benchmarks circuits were relatively high. *For many cases, CPF-Scheduler could reduce energy and power even without any time penalty or even with gain in time.* This happens when the performance degradation due to multiplications in the critical path are adequately compensated by the number of ALU operations in the critical path. For this to happen, the ALU operations should be larger than or equal to the number of multiplications in the critical path. This is the case for most of the schedules obtained for the EWF and FIR benchmarks indicated by the time ratio (R_T) of less than or equal to one.

For the above experimental set up, we plotted the power consumption per cycle, over all the control steps (clock steps) for different benchmarks in Fig. 3(a) and 3(b) for resource constraints RC1 and RC3, respectively. The curves labeled as "S" correspond to the profile when the schedule is operated at a single frequency (which is the maximum frequency of the slowest operator, the multiplier) and single voltage. The profiles labeled as "D" correspond to the case when dynamic clocking and multiple voltage scheme are used. The effectiveness of the proposed scheduling scheme is obvious from the figures. Since the CPF is a complex function consisting of several parameters, it is difficult to quantify the impact of a specific parameter accurately.

We also performed experiments with three voltage levels $(1.5V, 2.4V, 3.3V)$ and four frequency levels. The results could improve within the range of $5 - 10\%$ in terms of power or energy reductions. However, the time penalty increased by 15%. It is to be noted that the number of allowable frequency levels should be as close to the number of allowable voltages in order to keep the time penalty within a reasonable limit. We performed the same set of experiments for the CPF model 2 in which the cycle difference power is modeled as cycle-to-cycle power gradient. The experimental results indicate that the energy and power reduction were similar with small differences, but there were no changes in terms of time penalty. We conclude that the minor difference is due to the fact that the quantitative difference between the values of $(\frac{1}{N}\sum_{c=1}^{N}|P - P_c|)$ and $(\frac{1}{N-1}\sum_{c=1}^{N-1}|P_{c+1} - P_c|)$ are not significant.

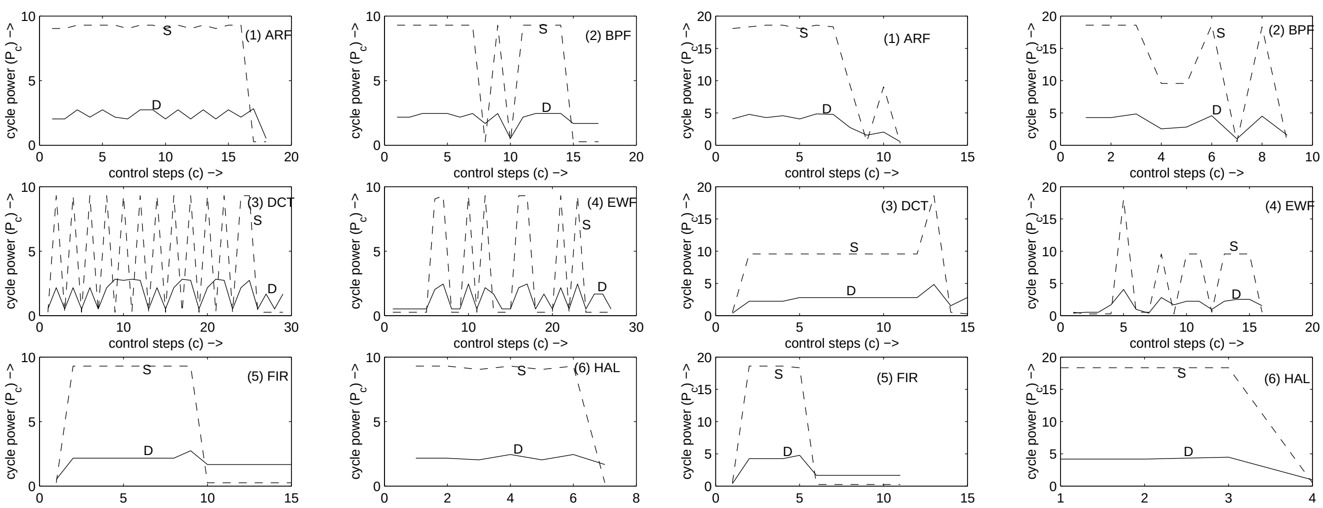
VI. CONCLUSIONS

For deep submicron and nanometer technology designs used in low power battery driven systems, simultaneous minimization of total energy and transient power is beneficial. The CPF parameter defined and used in this work essentially facilitates such simultaneous optimization. The datapath scheduling algorithm described in this paper is particularly useful for synthesizing data intensive application specific integrated circuits. The algorithm attempts to optimize energy and power while keeping the time penalty at a minimum. The CPF-Scheduler algorithm assumes the number of different types of resources at each voltage level and the number of allowable frequencies as resource constraints. The main contribution of this work is a unified framework for simultaneous multicost space metric optimization of different energy and power components in CMOS circuit design. Future work could address leakage reduction and interconnect issues. The effectiveness of the CPF in the context of a pipelined datapath and in control intensive applications needs to be investigated.

TABLE III

POWER ESTIMATES FOR DIFFERENT BENCHMARKS (USING MODEL 1)

Bench-mark Circuits	Power reduction details, Energy savings, Number of clock cycles and Time penalty											
	RC	P_{pS} (mW)	P_{pD} (mW)	ΔP_p (%)	P_{mS} (mW)	P_{mD} (mW)	ΔDP (%)	ΔP (%)	ΔE (%)	N	r_T	CPF
1	2	3	4	5	6	7	8	9	10	11	12	13
ARF (1)	1	9.30	2.83	69.60	0.26	0.52	74.50	71.40	47.57	18	1.6	0.52
	2	18.33	4.77	73.96	0.26	0.52	76.47	68.30	47.57	13	1.4	0.55
	3	18.59	4.84	73.96	0.26	0.52	76.44	71.72	49.87	11	1.5	0.58
	4	18.59	7.26	60.96	0.26	0.52	63.25	59.10	29.49	11	1.5	0.62
	Average values			**69.62**			**72.67**	**67.63**	**43.62**		**1.5**	
BPF (2)	1	9.30	2.45	73.62	0.26	0.52	78.64	65.80	46.69	17	1.3	0.55
	2	18.33	4.20	77.10	0.26	1.67	86.03	58.81	46.69	17	1.2	0.47
	3	18.59	4.84	73.96	0.52	0.97	78.59	71.09	48.61	9	1.4	0.61
	4	18.59	7.33	60.60	0.52	0.97	64.84	64.01	32.02	9	1.4	0.64
	Average values			**71.32**			**77.02**	**64.93**	**43.50**		**1.3**	
DCT (3)	1	9.30	2.83	69.60	0.26	0.52	74.50	50.90	42.44	29	1.1	0.66
	2	9.30	2.83	69.60	0.26	0.52	74.50	50.90	42.44	29	1.1	0.64
	3	18.59	4.84	73.96	0.26	0.40	75.75	67.70	42.93	15	1.4	0.39
	4	18.59	7.61	59.05	0.26	0.40	60.63	65.19	38.49	15	1.4	0.25
	Average values			**68.05**			**71.35**	**58.67**	**43.58**		**1.2**	
EWF (4)	1	9.30	2.45	73.62	0.26	0.52	78.64	41.17	44.43	27	0.9	0.56
	2	18.07	4.07	77.49	0.26	0.52	80.09	37.49	44.43	27	0.9	0.30
	3	18.07	4.07	77.49	0.26	0.40	79.38	57.89	44.73	16	1.2	0.39
	4	18.07	6.55	63.75	0.26	0.40	65.49	53.10	38.45	16	1.2	0.25
	Average values			**73.09**			**75.90**	**47.41**	**43.01**		**1.1**	
FIR (5)	1	9.30	2.74	70.52	0.26	0.52	75.45	58.54	46.11	15	1.3	0.42
	2	9.30	2.74	70.52	0.26	0.52	75.45	58.54	46.11	15	1.3	0.47
	3	18.59	4.77	74.32	0.26	0.40	76.12	51.21	46.77	11	1.0	0.53
	4	18.59	7.04	62.15	0.24	0.40	63.77	40.69	27.21	11	1.2	0.49
	Average values			**69.38**			**72.70**	**52.25**	**41.55**		**1.2**	
HAL (6)	1	9.30	2.45	73.62	0.26	1.67	91.38	72.32	50.58	7	1.6	0.64
	2	18.33	4.49	75.53	0.26	1.67	84.44	64.70	50.58	5	1.4	0.76
	3	18.33	4.49	75.53	0.52	0.97	80.27	72.48	51.84	4	1.5	0.64
	4	18.33	6.97	61.98	0.52	0.97	66.32	57.14	25.00	4	1.5	0.63
	Average values			**71.67**			**80.60**	**66.66**	**44.50**		**1.5**	
Average values				**70.52**			**75.04**	**59.59**	**43.29**		**1.3**	



(a) for resource constraint RC1

(b) for resource constraint RC3

Fig. 3. Cycle power consumption of different benchmarks for various resource constraints

REFERENCES

[1] A. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, 27(4), pp. 473–483, Apr 1992.

[2] D. Singh, J. M. Rabaey, M. Pedram, F. Catthoor, S. Rajgopal, N. Sehgal, and T. J. Mozdzen, "Power Conscious CAD Tools and Methodologies: A Perspective," *Proc. of the IEEE*, 83(4), pp. 570–594, Apr 1995.

[3] D. Sylvester and H. Kaul, "Power-Driven Challenges in Nanometer Design," *IEEE Design and Test of Computers*, 13(6), pp. 12–21, Nov-Dec 2001.

[4] L. Benini, G. Casterlli, A. Macii, and R. Scarsi, "Battery-Driven Dynamic Power Management," *IEEE Design and Test of Computers*, 13(2), pp. 53–60, Mar-Apr 2001.

[5] T. L. Martin and D. P. Siewiorek, "Non-ideal Battery Properties and Low Power Operation in Wearable Computing," in *Proc. the 3rd Intl. Symp. on Wearable Computers*, 1999, pp. 101–106.

[6] Trevor N. Mudge, "Power: A First Class Design Constraint for Future Architecture and Automation," in *Proc. the Intl. Conf. on High Performance Computing*, 2000, pp. 215–224.

[7] T. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, "A Dynamic Voltage Scaled Microprocessor System," *IEEE Journal of Solid-State Circuits*, 35(11), pp. 1571–1580, Nov 2000.

[8] J. Pouwelse, K. Langendoen, and H.Sips, "Energy Priority Scheduling for Variable Voltage Processor," in *Proc. the International Symposium on Low Power Electronics and Design*, Aug 2001, pp. 28–33.

[9] I. Brynjolfson and Z. Zilic, "Dynamic Clock Management for Low Power Applications in FPGAs," in *Proc. the IEEE Custom Integrated Circuits Conference*, 2000, pp. 139–142.

[10] J. M. Kim and S. I. Chae, "New MPEG2 Decoder Architecture using Frequency Scaling," in *Proc. the IEEE Intl. Symp. on Circuits and Systems*, 1996, pp. 253–256.

[11] N. Ranganathan, N. Vijaykrishnan, and N. Bhavanishankar, "A VLSI Array Architecture with Dynamic Frequency Clocking," in *Proc. the Intl. Conf. on Computer Design*, 1996, pp. 137–140.

[12] Y. R. Lin, C. T. Hwang, and A. C. H. Wu, "Scheduling Techniques for Variable Voltage Low Power Design," *ACM Trans. on Design Automation of Electronic Systems*, 2(2), pp. 81–97, Apr 1997.

[13] M. Johnson and K. Roy, "Datapath Scheduling with Multiple Supply Voltages and Level Converters," *ACM Trans. on Design Automation of Electronic Systems*,2(3), pp. 227–248, July 1997.

[14] M. Johnson and K. Roy, "Optimal Selection of Supply Voltages and Level Conversions during Datapath Scheduing under Resource Constraints," in *Proc. of the Intl. Conf. on Computer Design*, Oct 1996, pp. 72–77.

[15] J. M. Chang and M. Pedram, "Energy Minimization using Multiple Supply Voltages," *IEEE Trans. on VLSI Systems*, 5(4), pp. 436–443, Dec 1997.

[16] A. Raghunathan and N. K. Jha, "SCALP: An Iterative-Improvement Based Low-Power Datapath Synthesis System," *IEEE Trans. on CAD of Integrated Circuits and Systems*, 16(11), pp. 1260–1277, Nov 1997.

[17] M. Sarrafzadeh and S. Raje, "Scheduling with Multiple Voltages under Resource Constraints," in *Proc. of the IEEE Symp. on Circuits and Systems (Vol. 1)*, 1999, pp. 350–353.

[18] A. Kumar and M. Bayoumi, "Multiple Voltage-Based Scheduling Methodology for Low Power in the High Level Synthesis," in *Proc. of the Intl. Symp. on Circuits and Systems (Vol. 1)*. July, July 1999, pp. 371–379.

[19] M. M. Mansour, M. M. Mansour, I. Hajj, and N. Shanbhag, "Instruction Scheduling for Low Power on Dynamically Variable Voltage Processors," in *Proc. of the 7th IEEE Intl. Conf. on Electronics, Circuits and Systems*, 2000, pp. 613–618.

[20] W. T. Shiue and C. Chakrabarti, "Low-Power Scheduling with Resources Operating at Multiple Voltages," *IEEE Trans. on Circuits and Systems-II : Analog and Digital Signal Processing*, 47(6), pp. 536–543, June 2000.

[21] A. Manzak and C. Chakrabarti, "A Low Power Scheduling Scheme with Resources Operating at Multiple Voltages," *IEEE Trans. on VLSI Systems*, 10(1), pp. 6–14, Feb 2002.

[22] R. S. Martin and J. P. Knight, "Optimizing Power in ASIC Behavioral Synthesis," *IEEE Design and Test of Computers*, 13(2), pp. 58–70, Summer 1996.

[23] W. T. Shiue, "High Level Synthesis for Peak Power Minimization using ILP," in *Proc. the IEEE Intl. Conf. on Application Specific Systems, Architectures and Processors*, 2000, pp. 103–112.

[24] W. T. Shiue and C. Chakrabarti, "ILP Based Scheme for Low Power Scheduling and Resource Binding," in *Proc. of the IEEE Intl. Sym. on Circuits and Systems (Vol. 3)*, 2000, pp. 279–282.

[25] W. T. Shiue, J. Denison, and A. Horak, "A Novel Scheduler for Low Power Real Time Systems," in *Proc. of the 43rd Midwest Sym. on Circuits and Systems*, Aug 2000, pp. 312–315.

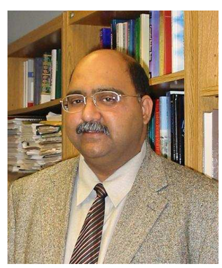
[26] V. Raghunathan, S. Ravi, A. Raghunathan, and G. Lakshminarayana, "Transient Power Management through High Level Synthesis," in *Proc. of the Intl. Conf. on Computer Aided Design*, 2001, pp. 545–552.

[27] L. Benini, E. Macii, M. Pnocino, and G. De Micheli, "Telescopic Units : A New Paradigm for Performance Optimization of VLSI Design," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 17(3), pp. 220–232, Mar 1998.

[28] C. Papachristou, M. Spining, and M. Nourani, "A Multiple Clocking Scheme for Low Power RTL Design," *IEEE Trans. on VLSI Systems*, 7(2), pp. 266–276, June 1999.

[29] S. P. Mohanty and N. Ranganathan, "Energy Efficient Scheduling for Datapath Synthesis," in *Proc. the Intl. Conf. on VLSI Design*, Jan 2003, pp. 446–451.

Saraju P. Mohanty obtained the Bachelor of Technology (First Class Honors) degree in Electrical Engineering from College of Engineering and Technology, Orissa University of Agriculture and Technology, Bhubansewar, India in 1995. He received the Master of Engineering degree in Systems Science and Automation from the Indian Institute of Science, Bangalore, India in 1999. He obtained Ph.D. in Computer Science and Engineering from the University of South Florida in 2003. He has published several research papers in areas of VLSI design automation, VLSI design and Digital watermarking. His paper was nominated for best paper award in international conference on VLSI Design 2003. In the year 2002 and 2003, he recieved certificate of recognition from Provost, University of South Florida for outstanding teaching. His research interests include high level synthesis, low power synthesis, VLSI CAD for DSM regime, dynamic power management, low power ASIC design. He is a member of IEEE-CS and ACM-SIGDA.



Nagarajan Ranganathan (S'81-M'88-SM'92, F'02) received the B.E. (Honors) degree in Electrical and Electronics Engineering from Regional Engineering College, Tiruchirapalli, University of Madras, India in 1983, and the Ph.D. degree in Computer Science from the University of Central Florida, Orlando in 1988. He is currently a professor in the Department of Computer Science and Engineering and the Nanomaterials and Nanomanufacturing Research Center at the University of South Florida, Tampa. His research interests include VLSI system design, design automation, power estimation and optimization, high level synthesis, embedded systems and computer architecture. He has developed many special purpose VLSI chips for computer vision, image processing, pattern recognition, data compression and signal processing applications. He has published over 190 papers in reputed journals and conferences and is a co-owner of five U.S. patents related to application specific integrated circuits and has recently filed a patent application on a new technique for leakage reduction in CMOS circuits. He was elected as Fellow of IEEE in 2002 for his contributions to algorithms and architectures for VLSI systems design.

Dr. Ranganathan is a member of IEEE, IEEE Computer Society, IEEE Circuits and Systems Society and the VLSI Society of India. He served as the chair of the IEEE CS Technical Committee on VLSI during 1997-2000. He has served on the program committees of international conferences such as ICCD, CAMP, ICPP, IPPS, SPDP, VLSI Design and ICHPC. He has served on the editorial boards of *Pattern Recognition*, *Intl. Journal of VLSI Design*, *IEEE Transactions on VLSI Systems*, *IEEE Transactions on Circuits and Systems TCAS-II*, *IEEE Transactions on CAS for Video Technology*. He served as the steering committee chair for the IEEE Transactions on VLSI Systems during 2000-02. He is currently serving as the Editor-In-Chief of the IEEE Transactions on VLSI Systems for the term 2003-04. He received the Theodore-Venette Askounes Ashford Distinguished Scholar Award at the University of South Florida in 2003.