A Framework for Energy and Transient Power Reduction during Behavioral Synthesis

Saraju P. Mohanty and N. Ranganathan
Department of Computer Science and Engineering
Nanomaterial and Nanomanufacturing Research Center
University of South Florida Tampa, FL 33620
smohanty@csee.usf.edu and ranganat@csee.usf.edu

Abstract

In deep submicron and nanometer designs for battery driven portable applications, the minimization of total energy, average power, peak power, and peak power differential are equally important. In this paper, we propose a framework for simultaneous reduction of these energy and transient power components during behavioral synthesis. A new parameter called "Cycle Power Profile Function" (CPF) is defined which captures the transient power characteristics as a weighted sum of mean cycle power and mean cycle differential power. Minimizing this parameter using multiple voltages and dynamic clocking results in reduction of both energy and transient power. Based on the above, a datapath scheduling algorithm called "CPF-Scheduler" is developed which attempts to minimize the CPF. Experimental results show that for two voltage levels, three operating frequencies, switching activity of 0.5 and power profiling factor of 0.5, the scheduler achieves (i) total energy reductions in the range of $27 - 53\%$, (ii) average power reductions in the range of $40 - 73\%$ (iii) peak power reductions in the range of $58 - 78\%$ and (iv) peak power differential reductions in the range of $60 - 97\%$. Further, the impact of switching, profiling factor and resource constraints on the power profile is studied in detail.

1 Introduction

The low power circuit design is a three dimensional problem involving area, performance and power trade-offs. Because of decreasing feature size and increasing packing density, it may be possible to trade area against power. With the increasing clock frequency, this trend has made reliability a big challenge for the designers, mainly bacause of high on-chip electric fields [16, 17]. Several factors such as, demand of portable systems, thermal considerations, environmental concerns and reliability issues have resulted in the need for low power design. In deep submicron and nanometer designs for low power, the total energy, average power, peak power and peak power differential are all equally important considerations. Both peak power and peak power differential drive the transient characterstics of the CMOS circuit. The life time and efficiency of battery is affected by all of the above parameters [8], since higher the current (power) lesser the electrochemical conversion efficiency. Reduction of average current (power) is essential to enhance noise margin (to decrease functional failures) and to increase electromagnetic reliability. The peak power affects packaging and cooling costs, functional failures, hot-electron effects (leading to runaway current failure) and electrostatic discharge failure. Reduction of current (power) fluctuation is necessary to reduce power supply noise (reducing di/dt), cross-talk and other electromagnetic noise. From the above discussion, it is observed that simultaneous minimization of all the four power and energy factors is important.

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. 2 below [10] :

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

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 [17], the authors indicate that there is an increase in both dynamic and static power in nanometer technology domain. The dynamic power component is significant due to increased switching activity in large circuits. In this work, we focus on the dynamic power aspect of the datapath circuits. It is well known that [3, 10], (i) by reducing supply voltage both power and energy can be saved

compromising delay, (ii) slowing down the CPU by reducing the clock frequency will save power but not energy, and (iii) varying frequency as well as voltage in a coordinated manner will save both energy and power while maintaining performance. In this work, we use the concepts of multiple voltages and dynamic clocking [2, 14] to achieve simultaneous minimization of energy and transient power.

2 Related Work

Few works have appeared in the literature for minimization of peak power and peak power differential which drive the transient power characteristics of the system. In [7], a method is developed for saving peak power in the range of $40-60\%$ which comes at the cost of average power penalty of $0.3-2.7\%$. ILP based scheduling schemes have been discussed in [15] that minimize peak power. Through scheduling and pipelining, the authors have achieved peak power reduction in the range of $0-75\%$, but there is no report about average power. A high-level synthesis scheme for simultaneous minimization of peak power and peak power differential is discussed in [13]. In this scheme, the peak power reduction is in the range $17-32\%$ (with an average of 25%) and the peak power differential is in the range $25-58\%$ (with an average of 42%). However, the above works do not address energy minimization.

Numerous low power datapath scheduling techniques have been reported in the literature a few examples of which can be found in [4, 6, 12]. These scheduling techniques are based on a single clock frequency and consider multiple supply voltages, voltage scaling, capacitance reduction, and switching activity reduction for minimization of either total energy or average power, but not both. Also, in these works the tranient power minimization is not considered.

Several works considering the use of variable latency and multiple frequencies have been reported. In [1], the authors introduce the use of "telescopic" units to improve throughput or performance of digital systems. The telescopic units complete execution in a variable number of clock cycles depending on the input data. They increase the number of cycles required for completion of a computation based on the input data. At the same time to match the critical-path delay, the clock rate is increased. A SIMD linear array image processor design is discussed in [14] that improves the system performance. The chip is operated at different frequencies depending on the type of instruction. A low power design using multiple clocking scheme is presented in [11]. If the overall effective frequency is f, then the circuit is partitioned to n different disjoint modules with each module operating at $(\frac{f}{n})$ frequency. Power savings up to 50% is obtained compared to single frequency. A time constrained heuristic scheduling algorithm is discussed in [9] that uses both frequency and voltage scaling. Energy savings in the range of $33-75\%$ is obtained, but power savings is not mentioned. Several system-level approaches [3, 5] have been investigated towards reducing power consumption in both general purpose and special purpose processors with the help of simultaneous voltage and frequency scaling.

This paper describes a framework for simultaneous minimization of total energy, average power, peak power, and peak power differential. A new parameter called *Cycle power Profile Function* (CPF) is defined which is a weighted sum of mean cycle power and mean cycle differential power. A datapath scheduling algorithm (called, CPF-Scheduler) is proposed to minimize the CPF using dynamic frequency clocking (DFC) and multiple voltages. The algorithm assumes different types and numbers of resources (such as, multipliers and ALUs) at different operating voltages and number of allowable operating frequencies as resource constraints and attempts to minimize CPF while keeping the time penalty at minimum. The scheduling algorithm generates a parameter called *Cycle Frequency Index*, denoted as cfi_c for control step c to be stored in the controller. This parameter serves as the clock dividing factor for the *Dynamic Clocking Unit* (DCU) used to generate different frequencies on the fly.

3 Cycle Power Profile Function

The cycle power profile function (CPF) needs to be defined such that it captures average power, peak power and peak power differential of the datapath. Since the peak power and peak power differential determine the transient power characteristics of the CMOS circuit, the CPF is a measure of the transient power characteristics of the datapath circuit. The datapath is represented as a sequencing data flow graph (DFG). The following definitions and notations are used in the description.

c : a control step or a clock cycle in DFG

N : total number of control steps in the DFG

P_c : the total power consumption of all resources operating during control step c (including overheads due to level conversions and dynamic clocking)

P_p : peak power consumption for the DFG equal to $max(P_c)_{\forall c}$

P : mean or average power consumption of the DFG

P_{norm} : normalised mean power consumption of the DFG

DP_c : mean difference power for cycle c (measure of cycle power fluctuation)

DP : mean of the mean difference powers for all control steps in DFG

DP_{norm} : normalised mean of the mean difference powers for all control steps in DFG

CPF_{norm} : normalised value of CPF

R_c : total number of resources active in step c

PF : power profiling factor

$\alpha_{i,c}$: switching activity of resource i, active in step c
$V_{i,c}$: operating voltage of resource i, active in step c
$C_{i,c}$: load capacitance of resource i, active in step c
f_c : frequency of control step c

The mean cycle power (P) that captures the average power consumption of the datapath can be defined as,

$$P = \frac{1}{N} \sum_{c=1}^{N} P_c \qquad (3)$$

The cycle power fluctuation (DP_c) is the difference of the cycle power and mean cycle power as given in Eqn. 4. This factor characterises the transience or the fluctuation in power consumption and hence, that of power supply.

$$DP_c = |P - P_c| \qquad (4)$$

The mean of the the above cycle difference power (DP) is a measure of power transience of the whole DFG (and hence that of the datapath) over all control steps as described below.

$$DP = \frac{1}{N} \sum_{c=1}^{N} |P - P_c| \qquad (5)$$

The cycle power profile function (CPF) is defined as the weighted sum of the mean cycle power and the mean cycle difference power (Eqn. 6). This function describes both average power and transient power characteristics of the circuit.

$$CPF(P, DP) = PF * P + (1 - PF) * DP \qquad (6)$$

The profile factor (PF) is used to tune the profile function for average power dominating or difference power dominating.

Using the dynamic power model from Eqn. 2, the cycle power (P_c) can be written as given below.

$$P_c = \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c \qquad (7)$$

Using Eqn. 7, we can rewrite Eqn. 3 as follows.

$$P = \frac{1}{N} \sum_{c=1}^{N} \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c \qquad (8)$$

The normalised mean cycle power (P_{norm}) is found out by dividing P by maximum cycle power, ($max(P_1, P_2, ..., P_N)$).

$$P_{norm} = \frac{\frac{1}{N} \sum_{c=1}^{N} \sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c}{P_p} \qquad (9)$$

where, the maximum power consumption for any cycle (P_p) defined below, captures the peak power consumption for the DFG.

$$P_p = max\Big(\sum_{i=1}^{R_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 f_c\Big)\Big|_{\forall c:1,2,...,N} \qquad (10)$$

Following similar steps, using Equations 7 and 8, the normalised mean cycle difference power (DP_{norm}) can be written as,

$$DP_{norm} = \frac{\frac{1}{N} \sum_{c=1}^{N} |P - P_c|}{max(|P - P_c|)|_{\forall c:1,2,...,N}} \qquad (11)$$

Using Eqn. 9 and Eqn. 11, the normalised CPF can be defined as follows :

$$CPF_{norm} = PF * P_{norm} + (1 - PF) * DP_{norm} \qquad (12)$$

We develop a scheduling algorithm that tries to minimize the above function (Eqn. 12) with the help of multiple voltages and dynamic clocking to reduce energy and the powers. In case of multiple voltage operations different resources can operate at different supply voltages. In dynamic frequency clocking, dynamic clocking or frequency scaling, all the units are clocked by a single clock line which can switch frequencies at run-time. The generation of such clocks have been studied extensively in [2, 14]. 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. At the same time the frequency reduction creates opportunity to operate the different functional units at different voltages, which in turn helps in energy reduction.

The processor model consists of a datapath, a controller and a dynamic clocking unit (DCU). The datapath consists of n functional units (FUs) with registers (Reg) and multiplexors (Mux). A controller decides which FUs are active in each control step. The controller has a storage unit to store the parameters, "cycle frequency index" (cfi_c) obtained from the scheduler, which serves as the clock dividing factor for the DCU. The cycle frequency f_c is generated dynamically and a FU operating at one of the supply voltages ($5.0V, 3.3V$ or $2.4V$) is activated. A level converter is used whenever a low-voltage FU is driving a high-voltage FU.

The delay for a control step is dependent on the delays of the functional units, multiplexer, register and level converters as expressed in following equation.

$$d_c = d_{FU} + d_{Mux} + d_{Reg} + d_{Conv} \qquad (13)$$

where, d_c is the delay of control step c, the register delays include the set-up and propagation delays, and FU delay is the delay of the slowest FU in the control step c. Using the

CPF-SchedulerAlgorithmFlow

Input : UDFG, resource constraints, V_i, L, d_{FU}, C_i, α
Output : f_{base}, N, cfi_c, scheduled DFG, power and delay

Step 1: Find ASAP and ALAP schedules of the UDFG.
Step 2: Determine the number of multipliers and ALUs
at different operating voltages.
Step 3: Modify both ASAP and ALAP schedules obtained
in Step 1 using the number of resources
found in Step 2 as initial resource constraint.
Step 4: Calculate the total number of control steps which is the
maximum of ASAP and ALAP schedules in Step 3.
Step 5: Find the vertices having non-zero mobility
and vertices with zero mobility.
Step 6: Use the CPF-Scheduler-Heuristics to assign time stamp,
operating voltage for the vertices and cycle frequencies
such that CPF_{norm} (Eqn. 12) is minimum.
Step 7: Find f_{base} and cfi_c using Eqn. 16.
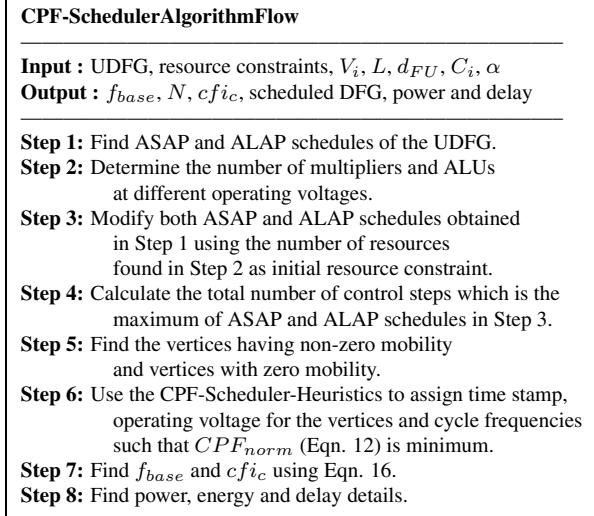**Step 8:** Find power, energy and delay details.

Figure 1. The CPF-Scheduler algorithm flow

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 :

$$f_{base} = \left\lfloor \frac{\lfloor 1/d_c^{min} \rfloor}{2^L} \right\rfloor 2^L \qquad (14)$$

$$cfi_c = \left\lceil \frac{\lfloor d_c/d_c^{min} \rfloor}{2^n} \right\rceil 2^n \qquad (15)$$

$$f_c = \frac{f_{base}}{cfi_c} \qquad (16)$$

where, d_c^{min} is the minimum delay of the control steps in which fastest resource is operating, L is 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 $\lceil d_c/d_c^{min} \rceil$.

4 CPF-Scheduler Algorithm

The inputs to the algorithm are an unscheduled data flow graph (UDFG), the resource constraints, the number of allowable voltage levels, the number of allowable frequencies (L), switching activity α, load capacitance of each resource (C_i), delay (d_{FU}) of each resource at different voltage levels. The resource constraint includes the number of ALUs and multipliers at voltage level V_i. The scheduling algorithm determines the proper time stamp for each operation, f_{base}, cfi_c and voltage level such that the function in Eqn. 12 is minimum. The algorithm also attempts to achieve this with minimum time penalty. The energy saving is achieved by utilizing the energy hungry resources operating at reduced voltages to maximum extent. The loss in
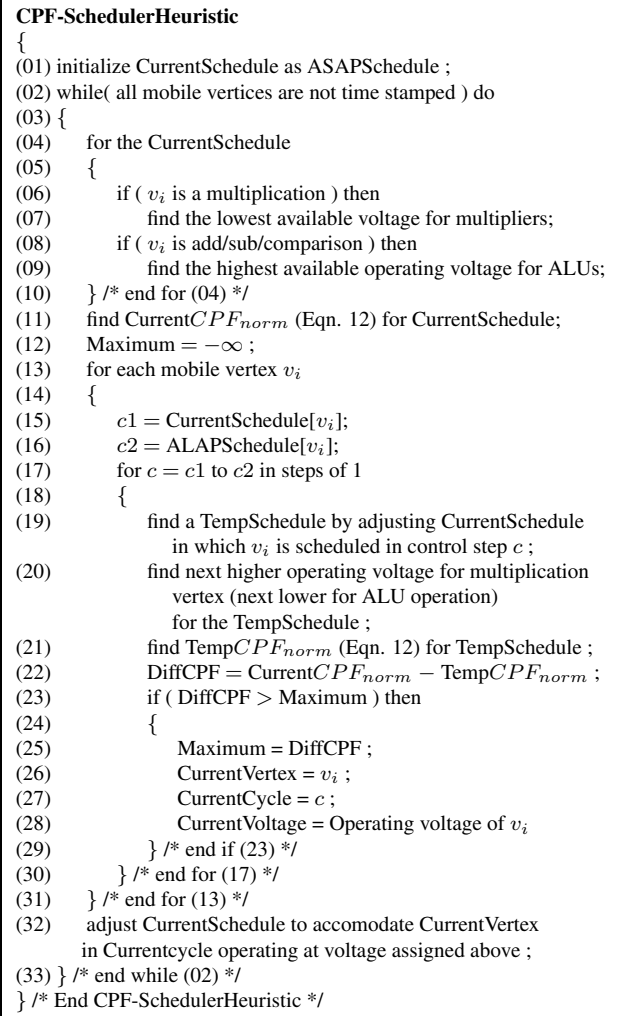
CPF-SchedulerHeuristic
{
(01) initialize CurrentSchedule as ASAPSchedule ;
(02) while(all mobile vertices are not time stamped) do
(03) {
(04)   for the CurrentSchedule
(05)   {
(06)   if (v_i is a multiplication) then
(07)   find the lowest available voltage for multipliers;
(08)   if (v_i is add/sub/comparison) then
(09)   find the highest available operating voltage for ALUs;
(10)   } /* end for (04) */
(11)   find CurrentCPF_{norm} (Eqn. 12) for CurrentSchedule;
(12)   Maximum $= -\infty$;
(13)   for each mobile vertex v_i
(14)   {
(15) $c1$ = CurrentSchedule[v_i];
(16) $c2$ = ALAPSchedule[v_i];
(17)   for $c = c1$ to $c2$ in steps of 1
(18)   {
(19)   find a TempSchedule by adjusting CurrentSchedule
                    in which v_i is scheduled in control step c ;
(20)   find next higher operating voltage for multiplication
                    vertex (next lower for ALU operation)
                    for the TempSchedule ;
(21)   find TempCPF_{norm} (Eqn. 12) for TempSchedule ;
(22)   DiffCPF = CurrentCPF_{norm} − TempCPF_{norm} ;
(23)   if (DiffCPF > Maximum) then
(24)   {
(25)   Maximum = DiffCPF ;
(26)   CurrentVertex = v_i ;
(27)   CurrentCycle = c ;
(28)   CurrentVoltage = Operating voltage of v_i
(29)   } /* end if (23) */
(30)   } /* end for (17) */
(31)   } /* end for (13) */
(32)   adjust CurrentSchedule to accomodate CurrentVertex
            in Currentcycle operating at voltage assigned above ;
(33) } /* end while (02) */
} /* End CPF-SchedulerHeuristic */

Figure 2. The CPF-Scheduler heuristic

performance is compensated by maximizing utilization of lesser energy consuming resources operating at higher voltages so that they can be operated at higher frequencies.

Fig. 1 outlines the flow of the proposed algorithm. The heuristic algorithm used to minimize the proposed objective function is shown in Fig. 2. Initially, the algorithm determines the ASAP and the ALAP schedules for the UDFG. The ASAP schedule is unconstrained and ALAP schedule uses the number of clock steps found in ASAP schedule as the latency constraint. Then, the algorithm finds the total number of each type of resources operating at all allowable voltage levels. The ASAP and ALAP schedules are modified with the help of the number of each type of resources found above. This helps in restricting the mobility of vertices in great extent and reducing the solution search space for the heuristic. The vertices are then marked as having zero mobility or non-zero mobility.

Once the vertices having zero mobility and vertices having non-zero mobility are determined, the next thing to be done is to find proper time stamp, operating voltage for mobile vertices and operating voltages for non-mobile vertices and operating clock frequencies such that the CPF_{norm} is minimum. The heuristic, initially assumes the modified ASAP schedule (with relaxed voltage resource constrained) as the current schedule. 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. Then the current CPF_{norm} value for the current schedule is calculated. The heuristic finds CPF_{norm} values (TempCPF_{norm}, in Fig. 2) for each allowable control step of each mobile vertices and for each available operating voltages. The heuristic fixes the time step, operating voltage and hence cycle frequency for which CPF_{norm} is minimum in a greedy manner as described in the heuristic.

5 Experimental Results

The CPF-Scheduler has been implemented in C and tested with selected benchmark circuits. The FUs used in the processor model are ALUs and multipliers. The benchmarks used are [9]: (1) Auto-Regressive filter, (2) Band-Pass filter, (3) Elliptic-Wave filter, (4) DCT filter, (5) FIR filter, (6) HAL differential equation solver. The processor model was simulated using the five sets of resource constraints as follows (RC1, RC2, RC3, RC4, RC5) : (1) multipliers : 1 at $3.3V$; ALUs : 1 at $5.0V$, (2) multipliers : 2 at $3.3V$; ALUs : 1 at $5.0V$, (3) multipliers : 2 at $3.3V$; ALUs : 2 at $5.0V$, (4) multipliers : 2 at $3.3V$; ALUs : 1 at $5.0V$ and 1 at $3.3V$, and (5) multipliers : 1 at $3.3V$ and 1 at $5.0V$; ALUs : 1 at $3.3V$ and 1 at $5.0V$. The number of allowable voltage levels being two and maximum number of allowable frequencies being three. The following parameters are used to express our experimental results.
E_S : total energy consumption assuming single frequency and single supply voltage
E_D : total energy consumption for dynamic 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 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 clocking and multiple supply voltage
T_S : execution time assuming single frequency
T_D : execution time assuming dynamic frequency
$\Delta E = \frac{E_S - E_D}{E_S}$: total energy reduction

$\Delta P = \frac{(E_S/T_S) - (E_D/T_D)}{(E_S/T_S)}$: average power reduction
$\Delta P_p = \frac{P_{pS} - P_{pD}}{P_{pS}}$: peak power reduction
$\Delta DP = \frac{(P_{pS} - P_{mS}) - (P_{pD} - P_{mD})}{(P_{pS} - P_{mS})}$: differential power reduction
$r_T = \frac{T_S}{T_D}$: time ratio

Under the assumption that each resource has the same amount of switching activity, the detailed results for different benchmarks are shown in Table 1. The results are shown for datapath components from [9]. In order to gain some insight from the experimental results, we analysed various energy or power reduction and time penalties. For each benchmark circuit, the average values of energy, power and time penalties for all 5 resource constraints were determined. We found that the scheduling scheme could achieve significant reductions in peak power, peak power differential, average power and total energy with reasonable time penalties. *For many cases, CPF-Scheduler could reduce energy or power even without any time penalty. This happens when there are equal number of multiplication and ALU operations in the critical path.*

To study the power consumption per cycle, we plotted the power profile for the benchmarks over all the control steps for different resource constraints, switching activity and profiling factors. Some of the power profiles are shown in Fig. 3 and 4. In all the cases, it is assumed that each resource has equal switching activity α. The curves labeled as "S" correspond to the profile when the schedule is operated at a single frequency (which is the maximum frequency of slower operator, 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. To examine the behavior of the CPF_{norm} versus the PF, we plotted them for different benchmarks as shown in Fig. 5. Since the CPF_{norm} is a complex function consisting of several parameters, it is difficult to accurately quantify the impact of a specific parameter. However, the graphs shown in the Fig. 5, point towards the fact that CPF_{norm} is approximately proportional to profiling factor.

A few works which attempt to minimize peak and/or peak power differential are listed in Table 2 alongwith the results of this work. In Table 2, it should be noted that "NA" indicates that the parameter is not addressed at all and "NO" indicates that there is no optimization of the parameter. The objective of including this table is not to provide performance comparison, but to provide a relatively broad idea of the performace of these works. Also, it should be noted that those works do not aim at simultanous reduction of energy and transient power. The main contribution of this work thus differs from those reported in [7, 13, 15].

Table 1. Power Estimates for different benchmarks (for $\alpha = 0.5$ and $PF = 0.5$)

Bench-mark Circuits	Resource Constraints	P_{p_S} (mW)	P_{p_D} (mW)	ΔP_p (%)	P_{m_S} (mW)	P_{m_D} (mW)	ΔDP (%)	ΔP (%)	ΔE (%)	N	r_T	CPF_{norm}
						Power reduction details, Energy savings, No. of clock cycles and Time penalty						
ARF(1)	RC1	40.99	15.20	62.92	1.19	2.38	67.78	71.25	47.29	18	1.8	0.48
	RC2	80.78	21.25	73.82	1.19	2.38	76.41	62.96	47.29	13	1.4	0.55
	RC3	81.97	24.95	69.56	1.19	2.38	72.05	69.48	47.29	11	1.7	0.51
	RC4	81.97	21.41	73.88	1.19	2.38	76.43	68.96	49.56	12	1.6	0.55
	RC5	81.97	32.64	60.19	2.38	8.12	69.20	63.31	29.96	11	1.9	0.72
BPF(2)	RC1	40.99	10.87	73.47	1.19	2.38	78.65	65.60	46.38	17	1.5	0.53
	RC2	80.78	19.55	75.80	1.19	2.38	78.42	58.57	46.38	17	1.2	0.43
	RC3	81.97	21.75	73.47	2.38	11.47	87.09	70.75	46.38	9	1.8	0.70
	RC4	81.97	21.41	73.88	2.38	4.92	79.27	71.17	48.74	9	1.7	0.60
	RC5	81.97	32.64	60.19	2.38	4.92	65.17	64.00	31.99	9	1.8	0.65
DCT(3)	RC1	40.99	15.20	62.92	1.19	2.38	67.78	49.90	41.26	29	1.1	0.56
	RC2	40.99	15.20	62.92	1.19	2.38	67.78	49.90	41.26	29	1.1	0.53
	RC3	42.17	16.28	61.41	1.19	4.75	71.88	67.37	41.26	15	1.7	0.56
	RC4	81.97	21.41	73.88	1.19	1.71	75.61	67.05	41.79	15	1.7	0.41
	RC5	81.97	34.24	58.23	1.19	1.71	59.73	64.42	37.14	15	1.7	0.26
EWF(4)	RC1	40.99	10.87	73.47	1.19	2.38	78.65	40.78	44.07	27	0.9	0.59
	RC2	40.99	10.87	73.47	1.19	2.38	78.65	40.78	44.07	27	0.9	0.56
	RC3	42.17	13.07	69.01	1.19	2.38	73.91	55.26	44.07	16	1.2	0.58
	RC4	79.60	17.35	78.20	1.19	1.71	80.05	57.59	44.33	16	1.3	0.41
	RC5	79.60	28.58	64.10	1.19	1.71	65.74	52.69	37.90	16	1.3	0.26
FIR(5)	RC1	40.99	12.48	69.56	1.19	2.38	74.62	58.29	45.78	15	1.3	0.42
	RC2	40.99	12.48	69.56	1.19	2.38	74.62	58.29	45.78	15	1.3	0.47
	RC3	81.97	18.54	77.38	1.19	4.75	82.93	54.92	45.78	11	1.1	0.60
	RC4	81.97	21.15	74.20	1.19	1.71	75.93	51.03	46.57	11	1.0	0.53
	RC5	81.97	31.03	62.14	1.19	1.71	63.70	40.15	26.55	11	1.2	0.49
HAL(6)	RC1	40.99	10.87	73.47	1.19	8.68	94.48	72.62	51.10	7	1.7	0.72
	RC2	80.78	19.55	75.80	1.19	8.79	86.48	65.07	51.10	5	1.4	0.77
	RC3	80.78	19.55	75.80	2.38	17.58	97.48	69.91	51.10	4	1.6	0.72
	RC4	80.78	19.55	75.80	2.38	4.92	81.34	72.96	52.68	4	1.7	0.64
	RC5	80.78	29.17	63.89	2.38	17.58	85.20	69.90	34.14	4	1.7	0.67

Table 2. Power reduction using different schemes

Bench-mark Circuits	CPF-Scheduler (Average values)				Shiue [15]		Martin [7]		Raghunathan [13]	
	ΔP_p	ΔDP	ΔP	ΔE	ΔP_p	ΔP	ΔP_p	ΔP	ΔP_p	ΔDP
ARF(1)	68	72	67	44	50	NA	-	-	-	-
BPF(2)	71	78	66	44	-	-	-	-	-	-
DCT(3)	64	69	60	41	50	NA	71	NO	28	45
EWF(4)	72	75	49	43	0	NA	-	-	-	-
FIR(5)	71	74	53	42	63	NA	45	NO	23	38
HAL(6)	73	89	70	48	28	NA	-	-	-	-

6 Conclusions

The work described in this paper provides a unified framework for simultaneous multicost space metric optimization of different energy and power components in CMOS circuit design. 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 maintaining performance. The CPF-Scheduler assumes several types of resources at each voltage levels and number of allowable frequencies as resource constraints. For two voltage levels, three operating frequencies, switching activity of 0.5 and $PF = 0.5$, experimental results show average reduction of, 44% total energy reduction, 61% average power, 70% peak power reductions and 76% of peak power differential. The average time penalty is 1.4. Future work needs to be done in applying better multicost metric optimization methods and algorithms for further improvement of the results. Also, the effectiveness of the CPF in the context of pipelined datapath and control intensive applications need to be investigated.

References

[1] L. Benini, E. Macii, M. Pnocino, and G. D. Micheli. Telescopic units : A new paradigm for performance optimization of vlsi design. *IEEE Trans. on CAD*, 17(3):220–232, Mar 1998.

[2] I. Brynjolfson and Z. Zilic. Dynamic clock management for low power applications in fpgas. In *Proc. of IEEE Custom*
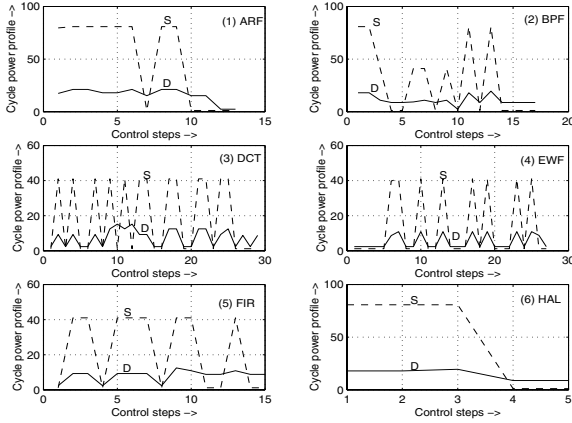
Figure 3. Power profile ($\alpha = 0.5$**,** $PF = 0.4$**, resource constraint : RC2)**
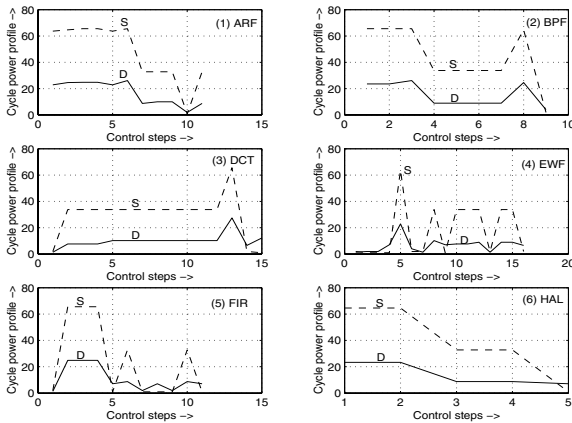

Figure 4. Power profile ($\alpha = 0.4$**,** $PF = 0.7$**, resource constraint : RC5)**

Figure 5. Normalised CPF **Vs** PF **for different benchmarks for resource constraints : RC4 and RC3 (**$\alpha = 0.5$**)**

Integrated Circuits Conference, pages 139–142, 2000.

[3] 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):1571–1580, Nov 2000.

[4] J. M. Chang and M. Pedram. Energy minimization using multiple supply voltages. *IEEE Trans. on VLSI Systems*, 5(4):436–443, Dec 1997.

[5] C. H. Hsu, U. Kremer, and M. Hsiao. Compiler-directed dynamic frequency and voltage scheduling. In *Proc. of Workshop on Power-Aware Computer Systems (PACS'00)*, pages 65–81, Nov 2000.

[6] M. Johnson and K. Roy. Datapath scheduling with multiple supply voltages and level converters. *ACM Trans. on Design Automation of Electronic Systems*, 2(3):227–248, July 1997.

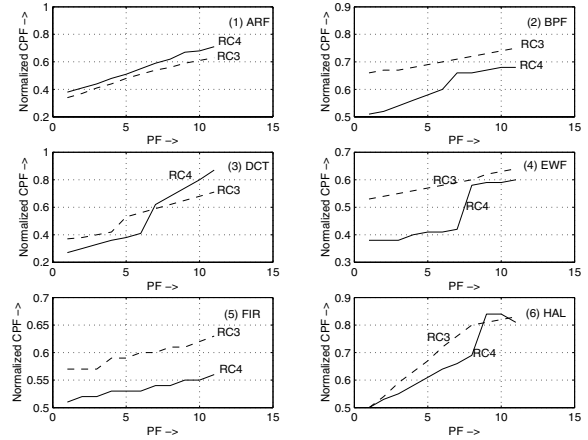[7] R. S. Martin and J. P. Knight. Using spice and behavioral synthesis tools to optimize asics' peak power consmption. In *Proc. of 38th Midwest Symposium on Circuits and Systems*, pages 1209–1212, 1996.

[8] T. L. Martin and D. P. Siewiorek. Non-ideal battery properties and low power operation in wearable computing. In *Proc. of 3rd International Symposium on Wearable Computers*, pages 101–106, 1999.

[9] S. P. Mohanty, N. Ranganathan, and V. Krishna. Datapath scheduling using dynamic frequency clocking. In *Proc. of ISVLSI'2002*, pages 65–70, Apr 2002.

[10] T. N. Mudge. Power: A first class design constraint for future architecture and automation. In *Proc. of HiPC*, pages 215–224, 2000.

[11] C. Papachristou, M. Spining, and M. Nourani. A multiple clocking scheme for low power rtl design. *IEEE Trans. VLSI Systems*, 7(2):266–276, June 1999.

[12] A. Raghunathan and N. K. Jha. Scalp: An iterative-improvement-based low-power data path synthesis system. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 16(11):1260–1277, Nov 1997.

[13] V. Raghunathan, S. Ravi, A. Raghunathan, and G. Lakshminarayana. Transient power management through high level synthesis. In *Proc. of ICCAD*, pages 545–552, 2001.

[14] N. Ranganathan, N. Vijaykrishnan, and N. Bhavanishankar. A vlsi array architecture with dynamic frequency clocking. In *Proc. of ICCD'96*, pages 137–140, 1996.

[15] W. T. Shiue. High level synthesis for peak power minimization using ilp. In *Proc. of IEEE International Conference on Application Specific Systems, Architectures and Processors*, pages 103–112, 2000.

[16] 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. *Proceedings of the IEEE*, 83(4):570–594, Apr 1995.

[17] D. Sylvester and H. Kaul. Power-driven challenges in nanometer design. *IEEE Design & Test of Computers*, 13(6):12–21, Nov-Dec 2001.