

# Energy Efficient Scheduling for Datapath 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 this paper, we describe two new algorithms for datapath scheduling which aim at energy reduction while maintaining performance. The proposed algorithms, time constrained and resource constrained, utilize the concepts of multiple supply voltage and dynamic clocking for energy minimization. In dynamic clocking, the functional units can be operated at different frequencies depending on the computations occurring within the datapath during a given clock cycle. The strategy is to schedule high energy units, such as the multipliers at lower frequencies such that they can be operated at lower voltages to reduce energy consumption and the low energy units, such as adders at higher frequencies, to compensate for speed. The algorithms have been applied to various high level synthesis benchmark circuits under different time and resource constraints. The experimental results show that for the time constrained algorithm, energy savings in the range of 33–75% are obtained. Similarly, for resource constrained algorithm, under various resource constraints using two supply voltage levels (5.0V, 3.3V), energy savings in the range of 24 – 53% can be obtained.*

## 1 Introduction

With the increase in demand for personal computing devices and wireless communications equipment, the demand for synthesizing low power consuming circuits has increased. The need for low power synthesis is driven by several factors such as : demand of portable systems (battery life), thermal considerations (cooling and packaging costs), environmental concerns (use of natural resources) and reliability issues. While the first three factors relate to average power dissipation, the last factor is affected by peak power. The power-delay-product has to be minimized to increase battery life, whereas, to increase battery life and reduce delay, the energy-delay-product is minimized.

Let us assume that  $C_{eff}$ ,  $V_{dd}$ ,  $f$  and  $V_T$  denote the effective switched capacitance, supply voltage, frequency and threshold voltage, respectively. For a CMOS circuit, the energy ( $E$ ) and power ( $P$ ) dissipation per operation and the critical delay ( $t_d$ ) can be described by the following equations :  $E = C_{eff}V_{dd}^2$ ,  $P = C_{eff}V_{dd}^2f$  and  $t_d = k \frac{V_{dd}}{(V_{dd}-V_T)^\gamma}$ ; where,  $\gamma$  is a technology dependent factor and  $k$  is a constant. From the above three equations, the following can be deduced [2, 11, 10] : by reducing only  $V_{dd}$ , both energy and power can be saved at the cost of performance (speed / time), slowing down CPU by reducing only  $f$  will save power but not energy and by scaling frequency and voltage in a coordinated manner, both energy and power can be saved while maintaining performance. This forms the major motivations for our approach. We generate a datapath schedule that will be operated with resources operating at multiple voltages and clocked with dynamic clock.

Several approaches have been investigated towards reducing power/energy consumption in both general purpose and special purpose processors. A dynamic voltage scaled microprocessor system is presented in [2], in which the frequency and the voltage levels to be supplied to the processor core, are determined by the help of operating system. A power efficient compiler in [5] determines the voltage level and clock frequency at compilation time for high level code. The authors in [11], describe a system for low-power microprocessor using dynamic voltage scaling. In [4], voltage and clock scheduling algorithms are incorporated in operating system. From the above works, it is quite evident that simultaneous voltage and frequency scaling is becoming important in low power processors. In this work, we aim at incorporating voltage and frequency scaling within the processor itself and develop a scheduling algorithm that can be incorporated into a datapath synthesis tool.

Various low power datapath scheduling techniques have been reported in the literature. A profile driven synthesis system is introduced in [8] which minimizes switching capacitance. A scheduling algorithm called Multiple Operating Voltage Energy Reduction is presented in [6],

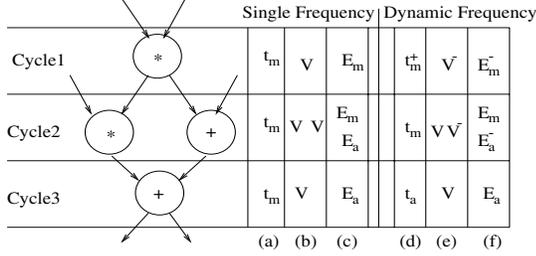


Figure 1. Example data flow graph (DFG)

which uses ILP formulations. A dynamic programming technique for multiple supply voltage scheduling is discussed in [3]. A time constrained multiple voltage scheduling technique is proposed in [13]. A resource constrained scheduling algorithm with multiple supply voltages is given in [7] which helps in reducing power using multiple supply voltages. In [14], a resource and a latency constrained list-based scheduling algorithms with multiple supply voltages are discussed. A resource and a time constrained scheduling algorithms are described [9] which uses the Lagrange multiplier method.

The above scheduling techniques are based on a single clock frequency and consider multiple supply voltages, voltage scaling, capacitance reduction, and switching activity reduction. In this work, we consider the use of dynamic frequency clocking (DFC) alongwith multiple supply voltage in developing resource / time constrained low power datapath synthesis schedulers. We propose two new datapath scheduling algorithms, one is called RC-DFC (resource constrained) and other is called TC-DFC (time constrained), which aim at reducing energy consumption.

## 2 Dynamic Clocking and Energy Savings

In dynamic frequency clocking [1, 12], the clock frequency is varied on-the-fly based on the functional units active in that cycle. In this clocking scheme, all the units are clocked by a single clock line which switches at run-time. The dynamic clocking unit (DCU) generates the required clock frequency uses a clock divider strategy to generate frequency which are submultiples of the base frequency. Base frequency  $f_{base}$  is the maximum frequency (or multiple of maximum) of any functional unit (FU) at the maximum supply voltage.

As discussed in the previous section, frequency scaling helps in reducing power, but not energy [2]. The frequency reduction creates an opportunity to operate the different functional units at different voltages, which in turn, helps in energy reduction. Let us consider the DFG shown in Fig. 1 scheduled to three control steps. Let  $t_a$  and  $t_m$  be the delays of the adder and the multiplier respectively at the maximum

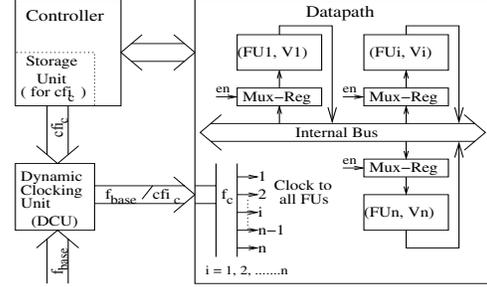


Figure 2. Processor Model

supply voltage  $V$ . Let us consider, operation of this DFG in three possible modes as follows.

(i) Single supply voltage and single frequency : Each cycle has clock width dictated by the slowest operator delay  $t_m$ . The total energy consumption is given by  $E_S = 2E_m + 2E_a$  and the total delay is  $T_S = 3t_m$ .

(ii) Multiple supply voltage and single frequency : Let,  $E_m^-$  and  $E_a^-$  are some energy values less than  $E_m$  and  $E_a$  respectively and  $t_m^+$  be the delay of multiplier at lower voltage  $V^-$ . The energy consumption of the DFG is given by  $E_M = E_m + E_a + E_m^- + E_a^-$ . In this case, the total delay is,  $T_M = 3t_m^+$ . Since,  $E_M < E_S$ , but  $T_M > T_S$ , it means that the energy savings comes at the cost of time penalty. In this case, energy consumption of the level converters is to be taken into account.

(iii) Multiple supply voltage and dynamic frequency : Say, we fix the clock cycle width for the 3rd cycle at  $t_a$  which is smaller than  $t_m$ ; this allows us to increase the clock width of some other cycles from  $t_m$  to some  $t_m^+$  without violating the time constraints. In this case, the total delay is  $T_D = t_m^+ + t_m + t_a$  and the energy consumption is  $E_M$  (same as, case (ii)). Since,  $T_D \leq T_M$ ,  $T_D \approx T_S$  and  $E_M < E_S$ , energy reduction is achieved without degrading performance. In this case, energy overhead of level converters and DCU is to be considered.

The processor model consists of a datapath, a controller and a DCU as shown in Fig. 2. 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 and those that are not active are disabled using the Mux. The controller has a storage unit to store the parameters  $cf_i$  obtained from the scheduling. 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. Level converters are used when a low-voltage FU is driving a high-voltage FU.

The delay of a control step is dependent on the delays of the FU, Mux and Reg pairs. The worst case delay of a control step can be written as :  $d_c = Regdelay + Muxdelay + FUDelay + levelconverterdelay$ , where,  $d_c$  is the delay

**Table 1. Delay values for 16-bit components**

Library Component	Delay Type	Delay (5V)	Delay (3.3V)	Delay (2.4V)
Register	set-up	3.3ns	5.9ns	10.1ns
Register	propagation	3.3ns	5.9ns	10.1ns
Multiplexor	propagation	6.0ns	10.8ns	18.4ns
add/sub	propagation	15.4ns	27.8ns	47.3ns
Multiplier	propagation	43.7ns	78.9ns	134.2ns

**Table 2. Energy dissipation of functional units**

Library Component	Energy (5V)	Energy (3.3V)	Energy (2.4V)
Mux	9pJ	4pJ	2pJ
add/sub	57pJ	25pJ	13pJ
Multiplier	2202pJ	960pJ	507pJ

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  $c$ . The estimated worst case delays of the library components are shown in Table 1. The worst case delay of a level converter is less than  $1ns$  as seen from HSPICE simulations. Assuming a switching activity of 0.5 at the inputs, the average energy dissipation for the components such as, the adder/sub, multiplier, etc. are estimated for different voltage levels as shown in Table 2 and 3.

Table 4 shows the clock frequencies corresponding to the use of each FU at different voltages. For a given base frequency ( $f_{base}$ ), maximum frequencies of each FU is scaled down to operating frequencies given by ( $\frac{f_{base}}{cfi_c}$ ), where,  $cfi_c = 1, 2, \dots, anynaturalnumber$ . The value of  $cfi_c$  is bounded by the product of the total number of resource types and number of voltage levels. The possible frequencies are,  $ALU_{High}(cfi_c = 1)$ ,  $ALU_{Med}(cfi_c = 2)$ ,  $ALU_{Low}(cfi_c = 4)$ ,  $MULT_{High}(cfi_c = 2)$ ,  $MULT_{Med}(cfi_c = 4)$  and  $MULT_{Low}(cfi_c = 8)$ .

### 3 TC-DFC Scheduler

The datapath is represented in the form of a DFG constructed as a sequencing graph. Fig. 3 shows such a graph for the HAL benchmark. The inputs to the algorithm are an unscheduled DFG, the scaled down operating frequencies, and the execution time constraint  $T_c$  for the whole schedule. To get more energy savings and at the same time maintain performance, the multipliers are to be operated at as low frequencies as possible and the adders at as high frequencies as possible. This objective can be achieved if adders / subtractors are not operated alongwith multipliers in the same duty cycle. In cases, when they are to be operated during the same cycle to meet the time constraint, energy savings will come from the multipliers only. Initially, TC-DFC generates a schedule such that the low frequency operators are

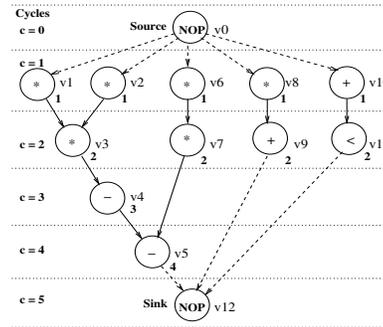
**Table 3. Energy dissipation in level converters**

V1-V2	2.4V	3.3V	5.0V
2.4V	-	53.04	139.4
3.3V	21.53	-	178.1
5.0V	22.5	61.4	-

**Table 4. Delays and operating frequencies**

Library Components	Delays and Frequencies		
	5.0V	3.3V	2.4V
add/sub/comp. (ALU)	25.7ns	45.5ns	76.8ns
Maximum Frequencies	38.91MHz	21.97MHz	13.02MHz
Operating Frequencies	36MHz	18MHz	9MHz
Multiplier (MULT)	54.0ns	96.6ns	163.7ns
Maximum Frequencies	18.51MHz	10.35MHz	6.1MHz
Operating Frequencies	18MHz	9MHz	4.5MHz

scheduled at earlier steps and the high frequency operators are scheduled at later steps. Later on, the TC-DFC modifies the schedule by moving operations from one step to another with the objective of meeting the time constraint.



**Figure 3. HAL differential equation solver**

Fig. 4 shows the flow of the TC-DFC scheduling algorithm. In step 1, ASAP schedule of the DFG is found out. In step 2, the scheduler creates a priority list of vertices such that all multiplications are grouped with higher priority than all ALU operations. Among the multiplications operations higher priority is given to the operations with smaller ASAP time stamp, so also the case among all ALU operations. In step 3, the vertices are time stamped such that multiplication and ALU operations not scheduled concurrently, precedence is satisfied and higher priority vertex scheduled at earlier time stamp. In step 4, for the current schedule, the cycles are categorised as, cycles having only ALU operations, only multiplication and both ALU operations and multiplication (mixed operations). In step 5, priority list of clock cycles is created such that cycles with only ALU operations get higher priority than cycles with only multiplications or mixed operations and cycles with only multiplications get higher priority than the cycles with mixed operations. Among the cycles with only ALU (or multipli-

- Step 1:** Find ASAP schedule for the sequencing UDFG  
**Step 2:** Create a priority list of vertices.  
**Step 3:** Assign control steps to the operations such that the higher priority vertex scheduled at earlier time stamp.  
**Step 4:** Find the cycles having only ALU operations, only multiplications and both ALU operations and multiplications (mixed) for the schedule obtained.  
**Step 5:** Create a priority list of clock cycles.  
**Step 6:** Initialise cycle frequency to minimum operating frequency.  
**Step 7:** If time constraint is not satisfied, the highest priority cycle is assigned next higher frequency.  
**Step 8:** If any cycle having multiplication operation operating at highest frequency then eliminate the cycle having minimum number of ALU operations, adjust the schedule and go to Step 4.  
**Step 9:** Do voltage assignment and estimate energy and delays.  
**Step 10:** Find the cycle frequency index for each cycle.

**Figure 4. TC-DFC Scheduling Algorithm**

**Table 5. Vertex Priority List**

v0	v1	v2	v6	v8	v3	v7	v10	v9	v11	v4	v5	v12
0	1	2	3	4	5	6	7	8	9	10	11	12

operations higher priority is given to the cycle having lesser number of ALU (or multiplication) operations. In step 6, initial cycle frequency is taken as minimum operating frequency. In step 7, in order to fulfil time constraint, the highest priority cycle frequency is increased. If needed the process is repeated for the next higher priority cycle. In step 8, if it is found that a cycle with multiplication has highest voltage then the cycle having minimum number of ALU operations is eliminated and the schedule is adjusted. In step 9, voltage assignment is done and energy details for whole DFG is found out. In step 10, cycle frequency index for each cycle is found out.

#### 4 RC-DFC Scheduler

The objective of RC-DFC is to minimize energy-delay-product of the DFG. For a resource  $i$  operating in clock step  $c$ , let, (i)  $\alpha_{i,c}$  be the switching, (ii)  $C_{i,c}$  be the load capacitance and (iii)  $V_{i,c}$  be the operating voltage. Level converter is being considered as a resource operating in the clock cycle in which it needs to step up the signal. If  $N$  is the total number of clock cycles of the DFG,  $NR_c$  is the number of resources active in cycle  $c$ , and  $f_c$  is the cycle frequency, then the total energy consumption and energy delay product (EDP) of the DFG can be expressed as,  $E_M = \sum_{c=1}^N \sum_{i=1}^{NR_c} \alpha_{i,c} C_{i,c} V_{i,c}^2$  and  $EDP_D = \left( \sum_{c=1}^N \sum_{i=1}^{NR_c} \alpha_{i,c} C_{i,c} V_{i,c}^2 \right) * \sum_{c=1}^N \frac{1}{f_c}$ , respectively.

In order to minimize the EDP, the RC-DFC attempts to operate the multipliers at as low frequency as possible, the resulting decrease in performance is compensated by oper-

**Table 6. Frequency selection for RC-DFC**

FUs in a cycle	Frequency priority order		
MULT	-	$MULT_{Low}, MULT_{Med}, MULT_{High}$	
MULT	ALU	$MULT_{Low}, ALU_{Low}, MULT_{High}$	
-	ALU	$ALU_{High}, ALU_{Med}, ALU_{Low}$	

**Table 7. Resource Look-Up : from left to right**

Clock Cycle	MULT			ALU		
	2.4 V	3.3 V	5.0 V	5.0 V	3.3 V	2.4 V
c	1	2	1	1	1	0

ating the ALUs at as high frequency as possible. Depending on which functional units are active in a given cycle, the algorithm determines the frequency using a lookup table (LUT), called "frequency selection LUT", such as the one shown in Table 6 scanning it left to right. Another lookup table called "resource assignment LUT" constructed using the resource constraints is used to match the selected frequency with a corresponding voltage level. The resources are assigned in the order, from left to right from the LUT. The scheduling while using these tables also uses heuristics to ensure that the number of times level conversions needed for execution of the whole DFG is minimum. An example resource assignment LUT for a clock cycle, is shown in Table 7. The dimension of this LUT depends on the total number of clock cycles of the schedule and number of resource types.

- Step 1:** Find ASAP and ALAP schedule of the UDFG.  
**Step 2:** Find the number of Multipliers and ALUs at different operating voltages.  
**Step 3:** Modify the schedules obtained in Step 1 using the number of resources found in Step 2.  
**Step 4:** Find the total number of control steps which is the maximum of ASAP and ALAP schedules in Step 3.  
**Step 5:** Construct the "resource assignment LUT" and "frequency selection LUT".  
**Step 6:** Find the vertices having non-zero mobility and vertices with zero mobility and assume ASAP schedule in Step 3 as the current schedule.  
**Step 7:** Do voltage and frequency assignment using the current schedule and the LUTs.  
**Step 8:** Taking a vertex with non-zero mobility find proper time stamp for it with the help of LUTs such that number of level conversions needed for the execution of whole DFG is minimum.  
**Step 9:** Adjust current schedule, predecessor and successor time stamps, LUTs, and repeat Step 7 and 8 to time stamp other vertices with non-zero mobility.  
**Step 10:** Find cycle frequency index for each cycle.

**Figure 5. RC-DFC Scheduling Algorithm**

Fig. 5 shows the flow of the algorithm. The inputs to the algorithm are an unscheduled sequencing DFG, the re-

**Table 8. Resource constraints**

Multipliers		ALUs		Serial No. (RCN)
3.3 V	5.0 V	3.3 V	5.0 V	
2	1	1	1	1
3	0	1	1	2
2	0	0	2	3
1	1	0	2	4
1	1	0	1	5
2	0	0	1	6

source constraints which include the number of resources, their corresponding operating voltages and the scaled down operating frequencies. In step 1, the scheduler determines the ASAP and the ALAP schedules for the UDFG. In step 2, the total number of resources is found out as the sum of each resource at different voltage levels. In step 3, the ASAP and ALAP schedules of step 1 are modified using the number of resources found in step 2. In step 4, the total number of control steps for both ASAP and ALAP schedule are found out and the number of control steps for the final steps is assumed to be the maximum of the two. In step 5, the "resource assignment LUT" and "frequency selection LUT" are constructed. In step 6, the vertices having non-zero mobility and the vertices with zero mobility are found out and the current schedule is initialized as the ASAP schedule obtained in step 3. In step 7, voltage and frequency assignments are made for the current schedule using the LUTs. In step 8, the scheduler finds a proper step for each vertex having non-zero mobility such that the number of level conversion needed for the execution of the whole DFG is minimum. In step 9, current schedule, LUTs are adjusted to satisfy the precedence. In step 10, clock frequency indices are found for all cycles which would be stored in the controller and would be fed to the DCU for dynamic frequency generation.

## 5 Experimental Results and Conclusions

Both RC-DFC and TC-DFC schedulers were implemented in C and tested with selected benchmark circuits. The benchmarks used are : (1) auto regressive filter, (2) band pass filter, (3) elliptic wave filter, (4) FIR filter and (5) HAL. The following notations are used to express results : (i)  $E_S$ ,  $E_M$  are the total energy consumption (in  $pJ$ ) for single supply voltage and multiple supply voltage operations, respectively, (ii)  $EDP_S$ ,  $EDP_M$  and  $EDP_D$  are the energy-delay-products (in  $10^{-18} Js$ ) for single supply voltage and single frequency, for multiple supply voltage and single frequency and for multiple supply voltage and dynamic clocking operations, respectively, (iii)  $T_S$ ,  $T_M$  and  $T_D$  are the corresponding delays (in  $ns$ ) for the three modes of operations, (iv)  $N_S$  denotes the number of clock steps of the schedule for single supply voltage and and sin-

gle frequency operations, (v)  $N_D$  is the equivalent clock steps of  $T_D$  found out taking the delay of slowest functional unit as the base clock width in case of multiple voltage operation, (vi)  $NOC$  is the number of times level conversion needed for the execution of all control steps of the DFG, (vii) The percentage energy savings is calculated as,  $S_E = \frac{(E_S - E_M)}{E_S} * 100$ , (viii)  $S_{EDP}^M = \frac{(EDP_M - EDP_D)}{EDP_M} * 100$  and (ix)  $S_{EDP}^D = \frac{(EDP_S - EDP_D)}{EDP_S} * 100$ .

For RC-DFC scheduler, the experimental set-up is as follows. The processor model was simulated using the different sets of resource constraints listed in Table 8. The experimental results for various benchmark circuits are reported in Tables 9. The energy estimation includes the energy consumption of the overheads. It is assumed that each resource has equal switching activity ( $\alpha_{i,c}$ ).

The TC-DFC scheduler is tested for three different time constraints: 1.5, 1.75 and 2.0 times critical delay. It is assumed that the processor has two ALUs and two MULTs. But voltage constraint is relaxed unlike the RC-DFC. The results for various benchmark circuits are reported in Table 10. Our observation is that circuits which require equal number of ALUs related operations and multiplier operations save more energy.

**Table 10. Energy savings for TC-DFC**

Bench. Circuits	Time Constraints	Energy consumption and savings		
		$E_S(pJ)$	$E_M(pJ)$	$S_E(\%)$
ARF (1)	$1.5T_c$	36186	21491	41
	$1.75T_c$	36186	18139	47
	$2.0T_c$	36186	15274	58
EWF (3)	$1.5T_c$	19422	12335	36
	$1.75T_c$	19422	8814	55
	$2.0T_c$	19422	5341	73
FIR (4)	$1.5T_c$	18696	4910	74
	$1.75T_c$	18696	4877	74
	$2.0T_c$	18696	4820	74
HAL (5)	$1.50T_c$	13614	7808	43
	$1.75T_c$	13614	6821	50
	$2.0T_c$	13614	4449	67

The energy savings for the proposed RC-DFC scheduling algorithm is listed alongwith other resource constrained multiple voltage scheduling algorithms in Table 11. The minimum and maximum range of energy savings shown in the table. The energy savings obtained using different existing multiple voltage based time-constraints scheduling algorithm is shown in Table 12.

**Table 11. Resource constrained schedulings**

Bench marks	Percentage Energy savings ( $S_E$ )			
	RC-DFC	Shiue[14]	Sarratzadeh[13]	Johnson[6]
ARF(1)	24-58	11-14	16-20	16-59
EWF(3)	38-61	14-14	13-32	11-50
FIR(4)	20-67	-	16-29	28-73

Our aim is to use frequency scaling concepts for

**Table 9. Energy and delay estimates for different benchmarks (for  $\alpha = 0.5$ ) using RC-DFC scheduler**

Bench mark Ckts	R C N	Energy Estimates			Energy-Delay-Product Estimates				Time Estimates				N O C		
		$E_S$	$E_M$	$S_E$	$EDP_S$	$EDP_M$	$EDP_D$	$S_{EDP}$	$N_S$	$T_S$	$T_M$	$T_D$		$N_D$	
		(pJ)	(pJ)	(%)	( $10^{-18}$ Js)	( $10^{-18}$ Js)	( $10^{-18}$ Js)	(%)		(ns)	(ns)	(ns)			
ARF (1)	1	36168	21768	40	20093	24186	19954	17	1	10	556	1111	917	9	11
	2	36168	18205	50	20093	20227	16688	17	17	10	556	1111	917	9	12
	3	36168	19065	47	20093	21183	18006	15	10	10	556	1111	944	9	16
	4	36168	27617	24	26121	44877	31452	29	-20	13	722	1625	1139	10	8
	5	36168	27617	24	26121	39890	28384	28	-9	13	722	1444	1028	10	8
	6	36168	19066	47	26121	27539	19595	28	25	13	722	1444	1028	10	16
BPF (2)	1	27654	16491	40	13827	16490	14659	11	-6	9	500	1000	889	8	9
	2	27654	14175	49	13827	14174	12600	11	9	9	500	1000	889	8	10
	3	27654	14827	46	13827	14827	12356	16	11	9	500	1000	833	8	12
	4	27654	20172	27	26118	42864	23253	45	11	17	944	2125	1153	10	7
	5	27654	20172	27	26118	38102	21292	44	18	17	944	1888	1056	10	7
	6	27654	14827	46	26118	28006	15651	44	40	17	944	1888	1056	10	12
EWF (3)	1	19404	10802	44	17248	19203	12902	32	25	16	889	1777	1194	11	10
	2	19404	10802	44	17248	19203	12902	32	25	16	889	1777	1194	11	10
	3	19404	10853	44	17248	19293	11154	42	35	16	889	1777	1028	10	8
	4	19404	11922	39	29106	40235	17055	57	41	27	1500	3375	1431	12	7
	5	19404	11922	39	29106	35765	15896	55	45	27	1500	3000	1333	13	7
	6	19404	10853	44	29106	32558	14470	55	50	27	1500	3000	1333	13	8
FIR (4)	1	18678	9979	47	11414	12196	6653	45	42	11	611	1222	667	7	8
	2	18678	9979	47	11414	12196	6653	45	42	11	611	1222	667	7	8
	3	18678	10126	45	11414	12377	6470	47	43	11	611	1222	639	6	8
	4	18678	10127	46	15565	18987	12096	36	22	15	833	1875	1194	10	8
	5	18678	10127	46	15565	16877	10971	34	30	15	833	1666	1083	10	8
	6	18678	10127	46	15565	16877	10971	34	30	15	833	1666	1083	10	8

**Table 12. Time constrained schedulings**

Bench- marks	% Energy savings ( $S_E$ )			
	TC-DFC	Chang[3]	Shiue[14]	Manzak[9]
ARF(1)	41-58	40-63	38-76	25-61
EWF(3)	36-73	44-69	13-76	10-55
HAL(5)	43-67	41-61	22-77	19-62

energy-efficient high-performance special purpose processor (ASIC) design. The energy reduction being achieved by voltage reduction and high-performance by using DFC alongwith multiple voltage. This paper introduced a resource-constrained and a time-constrained datapath scheduling algorithm based on dynamic frequency clocking. The dynamic frequency clocking could generate enough slack to apply reduced voltages which in turn saves energy.

## References

- [1] I. Brynjolfson and Z. Zilic. Dynamic clock management for low power applications in fpgas. In *Proc. of IEEE Custom Integrated Circuits Conference*, pages 139–142, 2000.
- [2] T. Burd and R. W. Brodersen. Energy efficient cmos microprocessor design. In *Proc. of the 28th Hawaii International Conference on System Sciences*, pages 288–297, 1995.
- [3] J. M. Chang and M. Pedram. Energy minimization using multiple supply voltages. *IEEE Trans. on VLSI Systems*, 5(4):436–443, Dec 1997.
- [4] D. Grunwald, P. Levis, and K. I. Farkas. Policies for dynamic clock scheduling. In *Proc. of 2000 Operating Systems Design and Implementation (OSDI)*, 2000.
- [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] A. Kumar and M. Bayoumi. Multiple voltage-based scheduling methodology for low power in the high level synthesis. In *Proc. of International Symposium on Circuits and Systems, 1999, ISCAS'99*, pages 371–379, 1999.
- [8] N. Kumar, S. Katkoori, L. Rader, and R. Vemuri. Profile-driven behavioral synthesis for low power vlsi system. *IEEE Design and Test of Computers*, 12(3):70–84, Fall 1995.
- [9] A. Manzak and C. Chakrabarti. A low power scheduling scheme with resources operating at multiple voltages. *IEEE Trans. on VLSI Systems*, 10(1):6–14, Feb 2002.
- [10] T. L. Martin and D. P. Siewiorek. Nonideal battery and main memory effects on cpu speed-setting for low power. *IEEE Trans. VLSI Systems*, 9(1):29–34, Feb 2001.
- [11] J. Pouwelse, K. Langendoen, and H. Sips. Dynamic voltage scaling on a low-power microprocessor. In *Proc. of 7th International Conference on Mobile Computing Network (MOBICOM)*, July 2001.
- [12] N. Ranganathan, N. Vijaykrishnan, and N. Bhavanishankar. A linear array processor with dynamic frequency clocking for image processing applications. *IEEE Trans. on CSVT*, 8(4):435–445, August 1998.
- [13] M. Sarrafzadeh and S. Raje. Scheduling with multiple voltages under resource constraints. In *Proc. of ISCAS'99*, pages 350–353, 1999.
- [14] 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):536–543, June 2000.