Exploiting Power-Area Tradeoffs in Behavioural Synthesis through clock and operations throughput selection

M. A. Ochoa-Montiel, B. M. Al-Hashimi

ESD, School of ECS
University of Southampton
Southampton, UK
mao02r@ecs.soton.ac.uk, bmah@ecs.soton.ac.uk

P. Kollig

Home Innovation Centre Southampton
NXP Semiconductors
Southampton, UK
peter.kollig@nxp.com

Abstract— **This paper describes a new dynamic-power aware High Level Synthesis (HLS) data path approach that considers the close interrelation between clock choice and operations throughput selection whilst attempting to minimize area, power, or a combination thereof. It is shown that the proposed approach with its compound cost function and its novel clock and operations throughput selection algorithm, obtains solutions with lower power and area than using previous relevant work [11]. Moreover, different power-area tradeoffs can be explored due to the appropriate choice of clock period and operations throughput using our novel approach.**

I. INTRODUCTION

High Level Synthesis is becoming a mature research area that has been investigated for two decades now. However, a renewed interest in HLS has been motivated due to the increasing complexity of digital systems and developments such as the advent of systems-on-a-chip (SoC) [4]. For the successful realisation of SoC for wireless communications and portable computing applications, it is fundamental to continue developing low power circuits, algorithms and CAD tools. For this reason, researchers are still investigating and developing improved methods for low power HLS, as recently demonstrated in [2], [13] and [14].

A key decision during HLS is choosing the clock period to schedule the data flow graph (DFG) operations into control steps. The clock selection problem is not new in the context of HLS and has been addressed in previous research, where it has been shown the significant effect of clock choice on the design in terms of area [1], [3], performance [9], [12], and power [8], [10], [11]. From the previous work on clock selection, the most relevant to our work is [11]. In [11], the authors developed a behavioural system that performs the three HLS tasks (scheduling, allocation and binding) and include supply voltage and clock period pruning techniques to eliminate inferior design points during the search for the minimum power solution.

Although [11] has performed effective power reduction, the methodology used to obtain the minimum power solution presents some shortcomings. For example, the optimum power solution is chosen after synthesising a data path for each combination of supply voltage and clock period that could not be pruned, leading to high computational times. Moreover, the optimization cost function ignores the area cost, which may lead to unnecessary big area implementations. The approach described here overcomes those disadvantages and makes the following contributions:

- The proposed clock and operations throughput selection algorithm obtain longer clock periods and lower applied voltages than the clock and supply voltage pruning techniques from [11], resulting in further power reduction.
- Use of an optimisation cost function compounded by area and power, which allows constraining area without compromising the power savings.
- The compound cost function allows the exploration of different power-area tradeoffs that may be of interest to the designer. This is unlike [11], which obtains only a solution with the lowest power consumption and does not investigate explicitly the power-area tradeoffs.
- Reduction of the computational time necessary to obtain a solution by including the clock and operations throughput selection algorithm into the synthesis phase.

This paper is organized as follows. Section II motivates the proposed approach and highlights the differences with existing techniques. Section III presents the proposed algorithm. Section IV shows the efficiency of the power-aware data path synthesis system. Section V concludes the paper.

II. MOTIVATIONAL EXAMPLE

Consider the computation of the DOT PRODUCT of two vectors in Cartesian form [11] as the benchmark used during this section. This benchmark consists of 6 multiplications and 5 additions. The 16-bit library used during our experiments is composed of: Wallace multiplier, CLA adder, register and diverse multiplexers. The area count and delay of the library components were obtained after synthesis using Synplify ASIC from Synplicity and STL 0.12μm technology. Power values at 90MHz were obtained using PrimePower from Synopsys and experimentally averaged over a number of input vectors. Table I shows the library used, and for the sake of space, multiplexers are omitted. Power consumption reported later in this motivational example is based on equation (3) and Table I. Linear regression and Lagrange polynomial were used to obtain component power values at different frequencies and voltages.

Table II shows four possible solutions for a time constraint $T = 47$ ns. In the table header, *sol.* represents the solution, Ls the schedule length, $Tclk$ the clock period, TP_m and TP_a the throughput of the multiplication and addition respectively in terms of control steps or csteps (cs). The slack of the critical path is represented by sk_{cp}, V represents the voltage, A the

TABLE I.   0.12μM COMPONENT LIBRARY

	MULTIPLIER		ADDER		REGISTER	
V (V)	P (mW)	D (ns)	P (mW)	D (ns)	P (mW)	D (ns)
1.08	1.237	11.048	0.040	4.292	0.047	0.547
1.2	1.870	7.143	0.049	2.861	0.055	0.344
1.32	2.556	4.650	0.066	1.976	0.077	0.233
A (μm)	10660.9		1107.4		548.7	

TABLE II.   SOLUTIONS FOR DOT PRODUCT WITH DIFFERENT OPTIMISATION GOALS

sol.	Ls (cs)	$Tclk$ (ns)	TP_m (cs)	TP_a (cs)	sk_{cp} (ns)	V (V)	A (μm)	P (mW)	Ct (s)
1	9	5.2	3	1	21.6	1.13	81251	1.304	134
2	5	9.4	2	1	16.5	1.08	78971	1.063	3
3	8	5.9	1	1	32.4	1.29	20192	2.976	57
4	19	2.5	6	3	16.5	1.08	35478	1.929	34

area, P the power and Ct the elapsed computational time to obtain the solution.

To calculate the clock period $Tclk$, the time constraint of the design T is divided into a number of control steps (schedule length Ls) equal to $\lfloor T/Tclk \rfloor$. The number of csteps necessary to perform an operation is equal to $\lceil delay_{op}/Tclk \rceil$. In this paper, we refer to the number of csteps over which a certain operation can be executed as the operation throughput TP_{op}. The operations delay $delay_{op}$ is considered as a typical register-to-register transfer that includes reading the operands from the registers, performing an operation on the operands and storing the results in another register [9]. Then, the delay of an operation can be approximated as:

$$delay_{op} = delay_{FU} + delay_{reg} + 2delay_{mux} \qquad (1)$$

where $delay_{FU}$ is the functional unit delay, $delay_{reg}$ is the register delay and $delay_{mux}$ is the multiplexer delay.

In Table II, the first solution denoted by $sol. = 1$, was obtained using an approach that includes the clock period and supply voltage pruning techniques from [11] together with data path synthesis system developed in [6] and using power as the cost function and the metric for evaluating moves. Low power consumption was achieved by choosing a clock period of 5.2ns and scaling the voltage to 1.13, leaving a critical path slack sk_{cp} of 21.6ns. We define the critical path slack sk_{cp} as the difference between the time constraint T, e.g. 47ns, and the time necessary to execute the critical path at certain voltage, e.g. for DOT PRODUCT, the critical path has one multiplication and three additions, and takes 25.4ns to be executed at 1.13V. Although $sol. = 1$ presents a reduced power solution, i.e. 1.304mW it also has the largest area implementation with 81251μm.

Using the proposed approach which involves appropriate clock and operations throughput selection, it is possible to achieve lower power solutions as shown in $sol. = 2$. This is accomplished through the careful choice of clock period, e.g. $Tclk = 9.4$ns, and operations throughput, e.g. $TP_m = 2$, $TP_a = 1$, which enables a greater voltage scaling, i.e. 1.08V,

decreasing the slack sk_{cp} from 21.6ns to 16.5ns, and reducing the power consumption from 1.304mW to 1.063mW.

It is noticeable that the power reduction is due not only to a voltage decrease, i.e. from 1.13V to 1.08V, but also to a frequency reduction, i.e. from 192.3MHz (1/5.2ns) to 106.4MHz (1/9.4ns). It is noticeable that a lower area implementation, i.e. 78971μm, was possible due to the area-power cost function used by the proposed approach, which allows constraining the area while searching for the minimum power solution. The proposed approach also obtains a solution in less time than $sol. = 1$, i.e. the computational time has been reduced from 134s to 3s. This is due to the inclusion of the clock and operations throughput selection into the datapath synthesis system, as shown in section III.

To show the influence of clock and operations throughput selection on the quality of the datapath in terms of power and area, consider now $sol. = 3$. As it can be seen, the adequate combination of clock period and operations throughput, i.e. $Tclk = 5.9$ns, $TP_m = 1$cs, $Tp_a = 1$cs, has allowed scaling the voltage to 1.29V without violating the time constraint and obtaining a minimum area implementation, i.e. 20192μm, at the expense of a power increase when compared to $sol. = 2$.

The proposed approach is able not only of optimising power ($sol. = 2$) or area ($sol. = 3$), but also of obtaining power-area tradeoffs that might be of interest to the designer, as the one shown in $sol. = 4$. An appropriate choice of clock period ($Tclk = 2.5$ns), and operations throughput ($TP_m = 6$cs, $TP_a = 3$cs) allows scaling the voltage to 1.08V, reducing the power consumption from 2.976mW to 1.929mW when compared to an area optimised solution ($sol. = 3$), and reducing the area implementation to 35478μm when compared to a power optimised solution ($sol. = 2$). From $sol. = 2$, $sol. = 3$ and $sol. = 4$, it can also be seen that the clock period and operations throughputs are chosen accordingly to the optimisation goal, e.g. for power, $Tclk = 9.4$ns, $TP_m = 2$cs and $TP_a = 1$cs; for area, $Tclk = 5.9$ns, $TP_m = 1$cs and $TP_a = 1$cs; and for the power-area tradeoff, $Tclk = 2.5$ns, $TP_m = 6$cs and $TP_a = 3$cs.

III.   PROPOSED APPROACH

For a given time constraint, the proposed approach considers concurrently the interrelation between the three HLS tasks combined with clock period and operations throughput selection while searching for solutions that meet the optimisation objective. Two main parts compose the approach: 1) generation of the list of clock candidates, 2) a simulated annealing optimisation process that considers simultaneously the three HLS tasks as well as clock and operations throughput selection. The inputs to the algorithm are: a DFG, the user constraints (time T, maximum frequency f) and the optimisation ratios for power and area (W_P and W_Q respectively). The output consists of a data path structure and timing information required for the synthesis of the control path.

The proposed approach starts generating a list of clock candidates using a modified version of [10], where it was shown how an appropriate choice of a clock period and functional units execution time allows the application of lower voltages, minimising the slack and leading to power savings. Three novel improvements were made to [10]. The

first modification consisted of finding the operations throughput not only for the type of operation that has maximum power consumption in the design, but also for the type of operation that don't have maximum power consumption in the design. This will allow exploring solutions that were not explored by the algorithm described in [10]. The second modification consists of considering only clock periods whose operations throughputs are different from those of the last clock period included in the list of possible candidates. The third modification consists of considering clock period and its respective operations throughputs that are not multiple of any clock period and its respective operations throughputs already included in the list. The second and third modifications decrease the number of clock periods and operations throughput to be analysed, hence reducing the computational time of the algorithm.

The algorithm for clock and operations throughput selection is shown in Listing I. The algorithm starts initializing the length of the schedule Ls (line 1) with the number of $csteps$ of the critical path of the design assuming that the operations are single cycled. It is also necessary to compute the maximum length allowed of the schedule max_Ls (line 2) since this will indicate the algorithm when to stop. This can be determined as $max\_Ls = T*f$, where T is the time constraint and f is the maximum frequency allowed for the design, both specified by the user. Then, the clock period $Tclk$ is computed (line 3) as shown in section II. After checking that the schedule length does not exceed the maximum schedule length allowed (line 4), the delay of an operation at maximum voltage, i.e. 1.32V, is increased until it fits exactly into the number of csteps of its throughput.

The algorithm aims firstly to increase the delay (line 5) of the type of operation that consumes more power. An increase of the delay allows a voltage reduction that has a higher impact on the power consumption of operations that are more power hungry. This could lead to a decrease in the total power consumption of the design. However, in some cases it is not possible to lower the voltage in such operations due to the violation either of the critical path or the maximum delay allowed by the library. Then, it is convenient to increase the delay of other operations (line 23). If the increased delay is larger than the maximum delay allowed by the library (line 5) it is necessary to increase Ls (line 25) and calculate a new clock period (line 26). When a correct increased delay has been obtained, the operations throughputs of the reminding operations are computed (line 6) as shown in section II.

Once the operations throughputs are obtained; the critical path needs to be calculated (line 7). If there has not been a time violation the operations are scheduled into Ls and the delay, voltage and power of the operators are computed (line 8). The operations were scheduled with the low run time LMSE scheduler used in [10]. The clock period, schedule information, operations throughput, and the values of delay, voltage and power for the operators are saved in a structure. This structure will be analysed to determine if it should be added to the *list of clock candidates* or not (line 9) according to the second and third modification previously explained.

So far, only the lower bound of the operations throughput for a certain clock period has been found. This lower bound allows obtaining schedules with reduced number of resources, i.e. multipliers and adders, leading to low area implementations. However, the proposed approach not only targets area optimization, but also power optimization. For this reason, it is also necessary to compute the upper bound of the operations throughput for a certain clock period. This upper bound allows the utilization of lower voltages than the lower bound, leading to power reductions. To calculate the upper bound, it is necessary to increase successively the operation throughput (line 11). The loop finishes when either the critical path (line 10) or maximum delay allowed by the library (line 12) is exceeded. Then, the last valid execution times of the operations are calculated (line 17 and line 18), followed by the scheduling and the calculation of delay, voltage and power of the operators (line 19). Later, a new structure that contains the clock period, schedule information, new operations throughput, and the values of delay, voltage and power for the operators is created. This structure will be also analysed to determine if it should be added to the list of clock candidates or not (line 20). Finally, all the process is repeated from the beginning until the maximum clock frequency is reached.

The *list of clock candidates* generated will be checked every time the move "clock and operations throughput selection" is executed, as it will be explained later in this section. Once the list of clock candidates has been created, an initial solution composed by the module and register binding is generated for the first element of the clock candidates list using the left edge algorithm [5].

Since our approach considers simultaneously the selection of clock period and operations throughput, as well as the three HLS tasks, a modified version of the data path synthesis algorithm developed in [6] is used to achieve the optimisation goal. The necessary modification involved the addition of the move "clock and operations throughput selection" to the original algorithm. Starting with the initial solution, the proposed approach generates new solutions that can be either

LISTING I. CLOCK AND OPERATIONS THROUGHPUT SELECTION ALGORITHM

```
1  initialise_Ls ( );
2  determine_max_Ls
3  calculate_Tclk
4  while Ls <= max_Ls do
5    if increase_delay(Op_power) <= max_d_lib do
6        calculate_throughputs( );
7        if critical_path( ) <= T do
8            schedule_calc( );
9            clock_list( );
10           while critical_path( ) <= Ls do
11               increase_throughput(Op_power) ;
12               if delay(Op_power) <= max_d_lib do
13                   break;
14               end if
15               calculate_throughputs(Op_reminding);
16           end while
17           decrease_throughput(Op_power);
18           calculate_throughputs(Op_remainding);
19           schedule_calc( );
20           clock_list( );
21       end if
22    else do
23        substitute Op_power  by Op_power_next and repeat lines 5 to 21
24    end if
25    increase_Ls( );
26    calculate_Tclk( );
27  end while
```

accepted or rejected depending on the acceptance criterion defined in the simulated annealing algorithm. The probability of accepting solutions with increasing cost depends on a control parameter, which is gradually lowered while the annealing process proceeds [7].

New synthesis solutions are generated using the strategy presented in the following. An operation is chosen randomly and one of the following moves is applied: 1) Schedule selected operation into a new control step, 2) Bind the selected operation result to a new register, 3) Bind selected operation to a new functional module, 4) Swap the module inputs of selected operation and 5) Clock and operations throughput selection. The fifth move consists on taking a new clock period, operations throughputs and applied voltage from the list of clock candidates. The application of this move results in a complete new solution that includes a new scheduling, allocation and binding. All the moves are applied in turns generating new solutions that can be either accepted or rejected according to the acceptance criterion of the simulated annealing. The proposed data path synthesis for low power is presented in Listing II.

The optimisation goal of the proposed approach considers the minimisation of not only the resource usage but also the power consumption in a given time constraint. Since two parameters, power and area, need to be optimised, the compound cost function is defined as:

$$\cos t = \sqrt{\left(W_P - \frac{P_i}{P_0}\right)^2 + \left(W_Q - \frac{Q_i}{Q_0}\right)^2} \qquad (2)$$

where W_P and W_Q are the target power and area ratios defined by the user, P_i and Q_i are respectively the power and area cost of a new solution, P_0 and Q_0 are respectively the maximum estimated power and maximum estimated area of the design.

The values of W_P and W_Q are set by the user according to the parameter to be optimized, area, power or both. For example, with $W_P = 1$ and $W_Q = 0$, the proposed approach gives good solutions in terms of area, whereas with $W_P = 0$ and $W_Q = 1$, the proposed approach targets to minimize power. For other combinations of W_P and W_Q, the proposed approach obtains solutions with power-area tradeoffs as the ones shown in section IV. The power cost P_i is the same that the estimated power consumption in the data path of the design. To allow a quick comparison among different design alternatives, the power of the data path can be expressed as:

$$P_{DP} = P_{REG} + P_{MUX} + P_{FU} \qquad (3)$$

where P_{DP} is the power consumption of the datapath, P_{REG} is the power dissipated by the registers, P_{MUX} is the power consumption due to the multiplexers and P_{FU} is the power dissipated by the functional units. The values of P_{REG}, P_{MUX} and P_{FU} are calculated assuming that the inputs are static when they are being used and clocks are switched off when they are idle. The area cost is given as in [6]:

$$Q = \sum_{All\ FUs\ used} Fa_F + Ra_R + Xa_X \qquad (4)$$

where Q is the area cost, F is the number of FUs of each type, a_F is the area of FUs of each type, R is the number of registers used, a_R is the area of a register, X is the number of multiplexers and a_X is the area of a multiplexer.

LISTING II. PROPOSED DATA PATH SYNTHESIS FOR LOW POWER

```
1  generate list of clock candidates
2  generate initial solution
3  while system is not frozen do
4     while  valid solutions < number of solutions to generate at this
       control parameter do
5        choose one operation randomly
6        generate a new solution applying one of the following moves:
7           1) schedule selected operation into a new control step
8           2) bind the selected operation result to a new register
9           3) bind selected operation to a new functional module
10          4) swap the module inputs of selected operation
11          5) clock and operations throughput selection
12       evaluate the cost of the new solution
13       accept or reject the new solution
14    end while
15    decrease simulated annealing control parameter
16 end while
```

IV. EXPERIMENTAL RESULTS

We have tested our approach with three benchmarks: ar filter (AR), elliptical wave filter (EWF) and discrete cosine transform (DCT). Two experiments have been conducted.

A. Exp. 1 - Comparison with a power aware base case

This section demonstrates the benefits of the proposed algorithm when compared with an approach that includes the clock period and supply voltage pruning techniques from [11] combined with the datapath synthesis system in [6], targeting power as the cost function and the metric for evaluating moves. We refer to this approach as the base case. To find the minimum power solution, the base case follows the methodology in [11], which enters the synthesis phase for all the values of clock period and supply voltage that could not be pruned.

Table III shows the improvement percentages in terms of area A, power P and computational time Ct for AR, EWF and DCT, with a time constraint T ranging from 1.5 times the critical path (1.5cp) to 3.5 times the critical path (3.5cp). It should be noted that a %saving greater than 0 means a decrease whereas a %saving lower than 0 means an increase with respect to the base case.

From Table III, it can be seen that the proposed approach is able to obtain significant power and area reductions for the three benchmarks. For example, for $T = 3.5$cp, power was decreased by 4.4% with an area decrease of 57.3% for AR, whereas for EWF, there was a power reduction of 6.3% with an area saving of 37.7%. In the case of DCT, an area decrease of 38.6% and power reduction of 19.6% were achieved with the same time constraint.

The substantial area savings are obtained because the compound cost function considers power and area, (see equation 2), unlike the base case, which uses a power cost function, ignoring totally the area and leading to big area implementations. While searching for the lowest power solution, the proposed approach constrains the area, thus avoiding excessive area implementations like the ones obtained by the base case. The power savings are due to improved selection of clock period and operations throughputs. This will be explained in more detail later in this section.

TABLE III. AREA, POWER AND COMPUTATIONAL TIME SAVINGS PERCENTAGE FOR DIFFERENT BENCHMARKS

T	savings for **AR**			savings for **EWF**			savings for **DCT**		
	%A	%P	%Ct	%A	%P	%Ct	%A	%P	%Ct
1.5cp	11.7	13.8	-9.7	1.4	20.9	81.7	26.2	14.2	81.8
2cp	50.2	5.2	26.6	26.2	-0.8	93.8	25.4	8.3	92.9
2.5cp	53.8	-0.2	9.7	15.5	4.8	96.9	26.3	11.2	96.0
3cp	47.3	0.0	-28.0	41.0	0.8	97.2	36.6	0.0	96.2
3.5cp	57.3	4.4	21.6	37.7	6.3	97.1	38.6	19.6	96.1

There are also some solutions that present a slight increase in the power consumption whilst at the same time achieving a significant area reduction. For example, AR with 2.5cp shows an increment of 0.2% in power consumption whilst the area decreases by 53.8%. For EWF with 2cp, a 0.8% power increase and 26.2% area decrease was obtained. Notice that all the solutions present lower area than the base case, with savings going from 11.7% to 57.3%, 1.4% to 41%, and 25.4% to 38.6%, for AR, EWF and DCT respectively.

From Table III, it can also be seen that the proposed approach can obtain optimised power solutions in much less time than the base case, with computational time savings going from 21% to 97% in general. The exceptions are for AR with 1.5cp and 3cp, which took 9.7% and 28% longer than the base case. The computational time savings are due to the inclusion of clock period and operations throughput selection algorithm that determines the scaled supply voltage into the data path synthesis, unlike the base case, where the synthesis phase is performed for all the combinations of clock period and voltage that could not be pruned

To give a better insight into the achieved power savings, solutions obtained by the two approaches are shown in Table IV for DCT with different time constraints. It can be seen that for the first three time constraints, 1.5cp, 2cp, and 2.5cp, the proposed approach obtains lower supply voltages and longer clock periods, thus reducing power when compared to the base case. For example, for a time constraint of 1.5cp, the proposed approach selects a clock period $Tclk =$ 4ns and operations throughput $TP_m = 3$ and $TP_a = 1$, which allows a reduction in the supply voltage, i.e. from 1.30V to 1.23V, and a decrease in frequency from 312.5MHz (1/3.2ns) to 250MHz (1/4ns), reducing the power consumption from 9.828mW to 8.435mW.

TABLE IV. MINIMUM POWER SOLUTIONS FOR DCT

T	**Base case**					**Proposed approach**				
	$Tclk$ (ns)	TP_m (cs)	Tp_a (cs)	V (V)	P (mW)	$Tclk$ (ns)	TP_m (cs)	Tp_a (cs)	V (V)	P (mW)
1.5cp	3.2	3	1	1.30	9.828	4.0	3	1	1.23	8.435
2cp	3.6	3	1	1.20	5.603	4.0	3	1	1.15	5.136
2.5cp	3.6	3	1	1.13	3.613	4.0	3	1	1.08	3.208
3cp	3.6	3	1	1.08	2.539	3.6	3	1	1.08	2.539
3.5cp	1.5	5	2	1.08	2.652	4.6	2	1	1.08	2.133

It can also be seen that for the time constraint 3cp, the proposed approach and the base case obtain the same voltage, i.e. 1.08V, and frequency, i.e. 277.8MHz, leading to the same power consumption, i.e. 2.539mW. An interesting observation is that for a time constraint 3.5cp, the supply voltage obtained by the proposed approach and the base case is the same, i.e. 1.08V, however the frequency has decreased from 666.7MHz (1/1.5ns) to 217.4MHz (1/4.6ns), reducing the power consumption from 2.652mW to 2.133mW.

B. Exp. 2 – Power-area tradeoffs

The proposed approach with its compound cost function and clock and operations throughput selection algorithm is also able to obtain power-area tradeoffs with smaller area and higher power values when compared to a power optimised solution. The comparisons were made using power and area ratios, which were normalized with respect to a power-optimised solution. It should be noted that a power or area ratio greater than 1 means an increase whereas a ratio lower than 1 means a decrease with respect to the base case.

From Figure 1, it can be seen that EWF presents area reductions of 40% with power increases of 40% and 50% for 1.5cp and 3.5cp respectively. It is also noticeable that for 2.5cp, there has been an area reduction of 25% with a power increase of 35%. For DCT, it can be seen that for time constraints of 1.5cp and 2cp, the proposed approach obtained solutions with lower area than a power optimised solution, 35% approximately, at the expense of a power increase of 45%. Further area reductions are possible, i.e. for 3cp and 3.5cp, which present average area savings of 44%, but with power increases of 87% and 125% respectively. For AR, it can be seen that 1.5cp and 2cp present an area reduction of 35% with power increases of 30% and 38% respectively. The remaining time constraints present an average area reduction of 44% with average power increase of 90%.

To give a better insight into the possible power-area tradeoffs for a given time constraint, consider the case of EWF with 2cp. Table V shows four possible power-area tradeoffs for the specified time constraint. It can be seen that different clock periods are obtained by using different schedule lengths, e.g. for $Ls = 37cs$, $Tclk$ is 2.5ns, whereas for $Ls = 20cs$, $Tclk = 4.6ns$. The schedule lengths in combination with the operations throughputs allow obtaining different schedules, hence, different allocation and binding, leading to different area implementations.
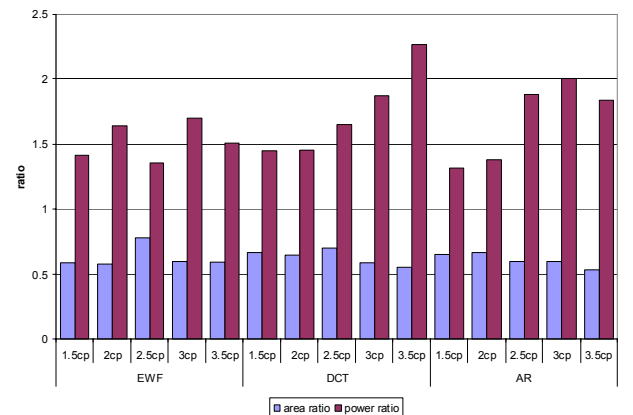

Figure 1. Power and area ratios for different power-area tradeoffs

TABLE V.   POWER-AREA TRADEOFFS FOR EWF WITH 2CP

$Tclk$ (ns)	Ls (cs)	TP_m (cs)	Tp_a (cs)	*	+	r	x	V (V)	A (μm)	P (mW)
2.5	37	3	2	1	5	17	16	1.23	31109.5	1.855
2.6	35	3	2	2	3	12	12	1.21	36650.8	1.787
4.6	20	2	1	2	3	17	11	1.17	37393.1	1.619
5.4	17	2	1	3	3	13	12	1.13	46698.4	1.412

For example, for $Ls = 37$, $TP_m = 3$ and $TP_a = 2$, the datapath requires 1 multiplier (*), 5 adders (+), 17 registers (r) and 16 multiplexers (x), leading to an area of 31109.5μm, while for $Ls = 20$, $TP_m = 2$ and $TP_a = 1$, the datapath needs 2*, 3+, 17r and 11x, leading to an area of 37393.1μm. It can be also noticed that clock and operations throughput selection affect not only the area of the design, but also its power consumption, e.g. for $Ls = 35$, $TP_m = 3$ and $TP_a = 2$, the minimum voltage applied is 1.21V, leading to a power consumption of 1.787mW, while for $Ls = 17$, $TP_m = 2$ and $TP_a = 1$, the minimum voltage applied is 1.13V with a power consumption of 1.412mW.

The compound cost function used by the proposed approach obtains not only power-optimised solutions or power-area tradeoffs, but it can also provide area-optimised solutions. For example, when compared to power optimised solutions, area reductions go from 55% to 68% for EWF depending of the time constraint. In the case of DCT, the area savings range goes from 45% to 65% and for AR, area reductions from 47% to 58% were obtained for different time constraints. Notice that since the optimization goal targets area, the parameter of power has been relaxed, leading to higher power consumption than the base case, i.e. EWF, DCT and AR present an average power increase of 100%, 115% and 114% respectively. As it can be seen, the area savings are at the expense of higher power consumption

V.   CONCLUSIONS

This paper presented a new dynamic-power aware data path synthesis approach that considers the close interrelation among scheduling, allocation, binding, clock and operations throughput selection. This provides solutions not only optimised for low power or low area, but also facilitates the automatic exploration of power-area tradeoffs. It has been shown that the proposed approach obtains solutions in lower computational time and with lower power and area than a base case approach that uses the clock and supply voltage pruning techniques from [11]. Power reductions were due to the use of larger clock periods (lower frequencies), or a combination of lower voltages and lower frequencies obtained after an appropriate selection of clock period and operations throughput using our novel algorithm. It has also

been shown how the clock and operations throughput selection affects the schedule, allocation, binding and voltage applied leading to solutions with different area and power consumption tradeoffs. Extensive experimental results for typical HLS benchmarks with different time constraints have shown considerable savings in power and area using the proposed approach.

At present we are extending the proposed approach to include the effect that multicycling has in leakage power.

REFERENCES

[1]   Blythe S. A. and Walker R. A., "Toward a practical methodology for completely characterizing the optimal design space", Proceedings of the 9th ISSS, 1996

[2]   Chabini N. and Wolf W., "Unification of scheduling, binding and retiming to reduce power consumption under timing and resources constraints", IEEE Transactions on VLSI, October 2005

[3]   Chauduri S., Blythe S. A. and WalkerR. A., "A solution methodology for exact design space exploration in a three dimensional design space", IEEE Transactions on VLSI systems, 1997

[4]   Gupta S., Gupta R. K., Dutt N. K., Nicolau A., SPARK: A Parallelizing Approach to the High-Level Synthesis of Digital Circuits, Kluwer Academic Publishers, 2004

[5]   Hashimoto A., Stevens J., "Wire routing by optimising channel assignment with large apertures", Proceedings 8th Design Automation Workshop, 1971

[6]   Kollig P. and Al-Hashimi, B.M., "Simultaneous scheduling, allocation and binding in high level synthesis", IEE Electronics Letters, August 1997

[7]   Laarhoven, P.J.M. van. and Aarts E., "Simulated annealing: Theory and applications", Kluwer academic, 1988.

[8]   Mehra R. and Rabaey J., "Behavioral level power estimation and exploration", Proceedings International Synposum on Low Power Design, 1994.

[9]   Narayanan S. and Gajski D. D., "System clock estimation based on clock slack minimization", Proceedings European Design Automation Conference, 1992

[10]   Ochoa-Montiel, M. A., Al-Hashimi, B.M., Kollig P.: "Impact of Multicycled Scheduling on Power-Area Tradeoffs in Behavioural Synthesis", Proceedings of the ISCAS, Japan, 2005

[11]   Raghunathan A. and Jha N. K., "SCALP: an iterative-improvement-based low-power data path synthesis system", IEEE Trans. on CAD of Integrated Circuits and Systems, 1997

[12]   Ramanujam J., Deshpande S., Hong J. and Kandemir M., "A Heuristic for clock selection in high level synthesis", Proceedings of the 15th International Conference on VLSI Design, 2002

[13]   Ranganathan N., Namballa R., Hanchate N., "CHESS: a comprehensive tool for CDFG extraction and synthesis of low power designs from VHDL", Proceedings ISVLSI, March 2006

[14]   Rettberg A., Rammig F. J., "A new design partitioning approach for low power high-level synthesis", Proceedings DELTA, Jan 2006