Impact of Multicycled Scheduling on Power-Area Tradeoffs in Behavioural Synthesis

M. A. Ochoa-Montiel
ESD, School of ECS
University of Southampton
Southampton, UK
mao02r@ecs.soton.ac.uk

B. M. Al-Hashimi
ESD, School of ECS
University of Southampton
Southampton, UK
bmah@ecs.soton.ac.uk

P. Kollig
BL Home Entertainment Solutions
Philips Semiconductors
Southampton, UK
peter.kollig@philips.com

***Abstract*— Multicycling is a widely investigated technique for performance optimisation in behavioural synthesis. It allows an operation to execute over two or more control steps with the aim of increasing the performance and/or minimising the power consumption. This paper presents a new time-constrained scheduling (TCS) algorithm that takes into account the combined influence of clock period and the multicycled functional units execution time on the quality of the schedules in terms of power and area. It is shown that it is possible to produce a set of solutions that have similar power consumptions, however differ in terms of resource requirements, yet meeting the imposed real-time constraint. Furthermore, extensive experiments on behavioural benchmarks show that the proposed approach is capable of obtaining schedules with single supply domain that have identical resource requirements and comparable power consumption to schedules obtained using multiple supply voltages, further reducing the design complexity.**

I. INTRODUCTION

Multicycled functional units (FUs) [2] have been employed to reduce the power consumption of digital designs generated using behavioural synthesis approach. This has led to the development of single-supply voltage (SSV) [3], [4] and multiple-supply voltages (MSV) [5]-[9] dynamic power minimization techniques. The SSV technique [3], [4] reduces power by operating all the FUs with the same, but lower, supply voltage than that used for the original design. For example, it was shown that as much as 56% power consumption reduction can be obtained if the original 5V design is translated to 3.3V using the behavioural synthesis system described in [4]. Power reduction using the MSV technique [5]-[9] is achieved by operating the FUs that execute operations on the critical path with the maximum possible supply voltage, to meet the imposed timing constraint, whilst operating FUs that execute operations which are not on the critical path with lower supply voltages. In general SSV techniques achieve moderate power saving when compared to the MSV techniques. However a SSV technique is easier to apply in practice since it does not require the provision of multiple supply voltages and level shifters, which are needed for MSV [8]. All the above techniques perform effective power reduction, however they do not efficiently explore the power-area solution space since they do not consider the feasible multicycled FUs execution times. In addition, while the research in [1] and [3] has shown how clock period and voltage choice may impact the power consumption, it did not illustrate explicitly the influence of the clock period, the multicycled FUs execution time and the voltage, on the power and area of the design. This paper presents a new algorithm that explores efficiently the power-area tradeoffs for a real time constraint by combining adequate selection of clock period and multicycled FU execution times. It is shown that the proposed approach is able to produce a SSV set of solutions with comparable power consumption having different functional resource requirements. Moreover, the power figures obtained by SSV are comparable to those obtained using MSV under the same resource constraints, without the practical implementation overhead of MSV.

II. MOTIVATIONAL EXAMPLE

To demonstrate how the selection of clock period and multicycled FU execution time affects the power consumption and functional resource requirements in behavioural synthesis, an implementation of the differential equation (DIFFEQ) benchmark [9] in 0.18μm technology is considered. The data flow graph (DFG) of DIFFEQ consists of 6 multiplications (*), 3 subtractions (-), 1 addition (+) and 1 comparison (>). Various circuit implementations combined with a dynamic power analysis over a set of input vectors has resulted in the FU power and delay values shown in Table I. Throughout this section we consider a real time constraint of 149ns (10 times the delay of an adder at 1.8V). The clock period (Tclk) may be chosen such that it is determined by the delay of the fastest functional unit when operating at maximum supply voltage [9], the length of the schedule in control steps (c-s) is obtained by dividing the time constraint with the chosen clock period as in [3]. This results in Tclk = 14.9ns and a schedule length (Ls) of 10 c-s. Reducing the voltage down to 0.9V means that the adder requires 2 c-s for execution. The multiplier from Table I requires 2 and 4 c-s at 1.8V and 0.9V, respectively. Operating the FUs at maximum supply voltage requires 3* with execution time (dM) of 3 c-s and 2+ with execution time (dA) of 2 c-s. The power consumption is 0.519mW and the critical path length is 89.6ns, leaving a slack of 60.4ns. It is this slack that allows a reduction in supply voltage, from 1.8V to 1.35V, leading to a power consumption reduction from 0.519mW to 0.438mW. The characteristics of this solution are summarized in the first row of Table II, denoted as S = 1. In the table header, S

represents the solution, V the voltage, P the power and #FUs the number of functional units. Although the slack time has been reduced to 7.5ns, it is still possible to take better advantage of the slack time and decrease power further by not constraining the clock period to the functional unit delay, as in the case of S = 1, but through appropriate choice of clock period and FU execution time. For example, S=2 divides the time constraint into 6 c-s (Tclk = 24.8ns). The change in execution time dM from 3 to 2, and dA from 2 to 1 reduces the slack time from 7.5ns to 0.001ns. Power consumption reduces when compared to S=1 from 0.438mW to 0.385mW. However, the number of used FUs remains the same. The schedule for this solution is shown in Fig. 1a. S=2 provides a schedule where power consumption has been effectively reduced. However, designers often need guidelines about the possible power-area tradeoffs in the design space. This can be achieved by carefully choosing the execution time of FUs and clock periods. For example, consider the time constraint divided into 12 c-s (Tclk=12.4ns), and execution times of 3 c-s and 2 c-s for the multiplier and adder, respectively (see Table II, S = 3). For the above settings, the minimum voltage required to meet the time constraint is 1.57V. The schedule is given in Fig. 1b. As it can be noted, there is an increase in power consumption (from 0.385mW to 0.497mW), however we reduce the number of functional units by one adder, hence, reducing the total area. For the same time constraint, further reduction in the number of required FUs can be obtained by assigning different values to the execution times of the FUs and the clock period, and by determining the corresponding voltage; as illustrated in Table II, for S = 4 and S = 5. Although S = 5 presents the same power consumption (0.497mW) as S = 3, the functional resources requirement is reduced by one multiplier. Less power consumption is obtained for S = 2, however it requires more functional resources than S = 5 (two extra FUs, one multiplier and one adder). Notice that the time constraint has remained the same (149ns) and that only the schedule length has changed according to the selected clock period. An additional advantage, of efficiently exploiting the power-area tradeoffs can be noted when we compare our results (S =5) with the MSV approach from [5].

To provide a fair comparison, we constrained the resources to 2* and 1+ (#FUs obtained in S = 5) and used the same time constraint. The results for using MSV with two supplies, 1.8V and 0.9V, are given in S = 6. As can be noted, we obtain comparable power consumption (0.431mW for MSV and 0.497mW for SSV) with the same number of functional resources, however using only one supply voltage. Large designs however, have a large design space and ad-hoc clock selection and FUs execution time assignment becomes quickly very time consuming. To overcome this problem, we describe a new time-constrained scheduling algorithm that is capable of exploring the design space efficiently in order to find a set of best power-area tradeoffs.

III. PROPOSED ALGORITHM

For a given time constraint, the proposed algorithm identifies the appropriate clock period, execution times for the different FUs and the minimum supply voltage. Three main parts compose the algorithm: **1)** calculation of the clock period; **2)** identification and validation of the possible execution times for the FUs, and voltage computation; and **3)** scheduling and computation of the total power and area. Unlike relevant previous work [3], which produces an optimised solution for power consumption through voltage scaling and clock selection, our algorithm obtains a set of solutions by identifying both the appropriate choice of FUs execution time and the clock period, such that the usage of low voltage can be applied. The inputs of the algorithm are: a DFG, a maximum clock frequency and a real time constraint. The outputs of the

TABLE I. FUNCTIONAL UNITS LIBRARY

Operator	*		+, -, <	
voltage (V)	0.9	1.8	0.9	1.8
Power (mW)	0.105	0.42	0.008	0.031
delay (ns)	59.6	29.8	29.8	14.9

TABLE II. SCHEDULE CHARACTERISTICS USING DIFFERENT POWER REDUCTION TECHNIQUES

S	Ls (c-s)	Tclk (ns)	slack (ns)	dM (c-s)	dA (c-s)	V (V)	P (mW)	#FUs
1	10	14.9	7.5	3	2	1.35	0.438	3*,2+
2	6	24.8	≈0	2	1	1.20	0.385	3*,2+
3	12	12.4	6.2	3	2	1.57	0.497	3*,1+
4	7	21.3	≈0	2	1	1.45	0.458	2*,2+
5	8	18.6	≈0	2	1	1.57	0.497	2*,1+
6	10	20	-	2/4	1/ 2	1.8/0.9	0.431	2*,1+

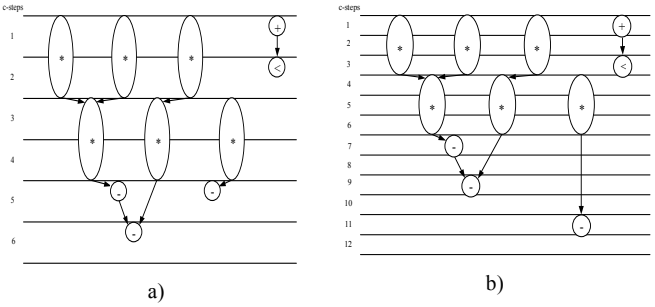

Figure 1. DIFFEQ schedules with different power-area tradeoffs.

algorithm are a set of solutions that consist of schedules with different power-area tradeoffs and their respective clock period, FU execution time and voltage. This information can be taken by the designer to generate RTL VHDL for the solution that best suits his needs. In the following we assume that: *C* is the real-time constraint in ns, *max_clk_freq* is the maximum clock frequency, *Tclk* is the clock period in ns, *DFUtype* is the functional unit delay in ns corresponding to the type of FU, *EFUtype* is the execution time in terms of c-s depending on the FU type and *maxDFUtype_library* is the maximum functional unit delay in ns corresponding to the type of FU (taken from the library in Table I). The algorithm starts to calculate the total power of an initial schedule at maximum supply, for example 1.8V for a 0.18μm process. The FUs are ordered in terms of maximum power consumption in *lstfu*. And the type of FU with maximum power *power_op* is identified. Now, *c_steps* is initialised with a schedule length equal to the critical path at 1.8V. According to the value of *c-steps*, the clock period *Tclk* is computed using the time constraint *C*. The execution time in terms of c-s *EFUpower_op* is evaluated for operations of type *power_op*. Later, its delay *DFUpower_op* is calculated and used to compute the voltage *volt* of the design. To calculate the values of delay at different voltages a 1st order Lagrange interpolating polynomial is used. The delay of the multicycled operator of type *power_op* is given with the product of the execution time and the clock period. Operators that are not of type *power_op* are included in the type *remain_op*. Their delay *DFUremain_op* is evaluated for the voltage *volt*, and then the execution time *EFUremain_op* computed. With *EFUpower_op* and *EFUremain_op*, the critical path can be computed and the feasibility of chosen delays without violating the time constraint is verified. If the time constraint is not violated, a modified version *mod_sched* of the low computational time TCS

algorithm from [11] is called. To allow consideration of power consumption the *lstfu* priority list has been added to the scheduling algorithm from [11]. Due to the iterative nature of the proposed approach, a large number of solutions may have to be investigated. Here we benefit from the low computational complexity (*O(nl)* where *n* is the number of nodes of the benchmark that can be assigned to a schedule length of *l*) of the algorithm in [11]. In addition to further reduce the computational time, the algorithm searches the solution only in the lower and upper bounds of *EFUpower_op*, instead of analysing the whole range of values for every *Tclk*. This restriction has been introduced based on experimental observations, which show that the "best power-area tradeoffs" tend to be located in these areas. The low computational complexity of the scheduler, together with the applied heuristic, allows the new algorithm to efficiently search the solution space in a reasonable time, considering the large solution space. Now, the total power consumption *total_power* of the design is evaluated and the area of the design *total_area* is estimated. To compute the values of power for the found voltage, a 2nd order Lagrange interpolating polynomial was used. The average power of a design is calculated in the same way as in [10]. The previous steps are repeated while *DFUpower_op* is less than, or equal to, *maxDFUpower_op_library*. The next step is to increase the value of *c-s* for the design, and repeat the external loop until the clock period *Tclk* exceeds the inverse of the maximum clock frequency *max_clk_freq*.

IV. EXPERIMENTAL RESULTS

Three experiments have been conducted using the DIFFEQ, elliptical wave filter (EWF) and discrete cosine transform (DCT) benchmarks using different time constraints.

A. *Exp. 1 - Power-area tradeoffs analysis*

Power-area tradeoffs are shown in Fig. 2 for the example of the DCT assuming a number of time constraints. We observe three interesting results. The first result is that for the same functional resources requirement, longer time constraints result in lower power consumption, as expected. For example, for 6* 5+, the power consumption is 1.076mW when the time constraint is 183ns, however it reduces to 0.447mW when the time constraint increases to 261ns. The second result is that there exist some solutions that have similar power consumption but different functional resources requirements. For example, for a time constraint of 156ns, a power consumption of 1.321mW can be obtained with an area of 6* 6+. Increasing the area to 7* 6+ reduces the power consumption to 1.283mW, which is not a significant saving at the expense of one extra multiplier. Similar results can be seen for other time constraints such as 130ns and 209ns. The third result extracted from Fig. 2 is that for a given time constraint, a greater number of FUs does not necessarily lead to lower power consumption. This is illustrated for time constraints of 209ns and 261ns, when the resources change from 4* 4+ to 4* 5+, producing power consumption of 1.022 and 0.746mW, respectively. This reinforces the conclusion recently reached in [7]. Hence, it is important to provide a designer with a set of *best power-area tradeoffs*, as the ones obtained by our approach. The set of best power-area tradeoffs of the DCT is shown in Table III. It can be seen that increasing the resource usage, is not always leading to power reduction. For example, comparing rows 3 and 4 of Table III shows no power saving by adding *one adder* to the design. However, adding one *multiplier* (row 5 and 8) can lead to a power reduction of 11%.

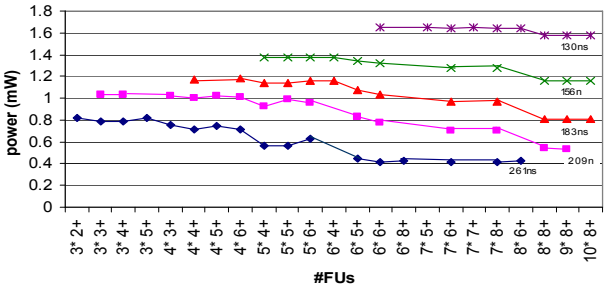

Figure 2. Power-area tradeoffs for DCT for different real time constraints

TABLE III. CHARACTERISTICS OF POWER-AREA TRADEOFFS FOR DCT WITH A TIME CONSTRAINT OF 183NS

Ls (c-s)	Tclk (ns)	DM (c-s)	DA (c-s)	V (V)	P (mW)	#FUs
33	5.5	6	3	1.70	1.167	4* 4+
18	10.1	3	2	1.78	1.177	4* 6+
10	18.2	2	1	1.59	1.136	5* 4+
20	9.1	4	2	1.59	1.136	5* 5+
16	11.4	3	2	1.66	1.159	5* 6+
43	4.2	8	4	1.67	1.161	6* 4+
9	20.3	2	1	1.47	1.076	6* 5+
17	10.7	4	2	1.40	1.031	6* 6+
8	22.8	2	1	1.32	0.972	7* 6+
40	4.5	10	5	1.32	0.972	7* 8+
7	26.1	2	1	1.12	0.805	8* 8+
21	8.7	6	3	1.12	0.805	9* 8+
56	3.2	16	8	1.12	0.805	10* 8+

Having such a set of solutions, the designer is no longer forced to accept *one* solution, which gives either the best power or the best area, but he can *choose* the one that suits best the application requirements. A general observation from Fig. 2 and other examples is that the relation between power and area is non-linear and varies depending on the benchmark, the time constraint and the functional resources used. It can be also observed that the shape of the curves changes according to the time constraint. This makes the design space exploration more complex and hence time consuming, and an efficient algorithm is needed. The proposed approach has reasonably low computational times as shown in Table IV. For example, the algorithm needs 206s to obtain the set of solutions shown in Table IV (DCT with a time constraint of 183ns). The computational time increases as the time constraint increases. This is due to the fact that the number of clock periods to be analysed by the algorithm increases as the time constraint increases.

B. *Exp. 2 - Comparison with MSV*

This experiment demonstrates further the importance of trading area for power using the proposed algorithm. The obtained solutions using SSV provide comparable power consumption values with the solutions obtained by a MSV approach [5] that uses two voltages, 1.8V and 0.9V. Comparable power values have been achieved without increasing the functional resource usage whilst meeting the imposed real time constraint. Besides the comparable power consumption, the proposed approach avoids the problems associated with MSV, such as the use of additional costly metal layers for the power supply grid and/or the use of level shifters. As can be seen from Fig. 3, the proposed algorithm produces solutions of comparable quality in terms of power than those generated using

TABLE IV. RUN TIMES FOR DCT

DCT	
Time Constraint (ns)	***Computational time (s)***
130	57
156	123
183	206
209	318
261	686

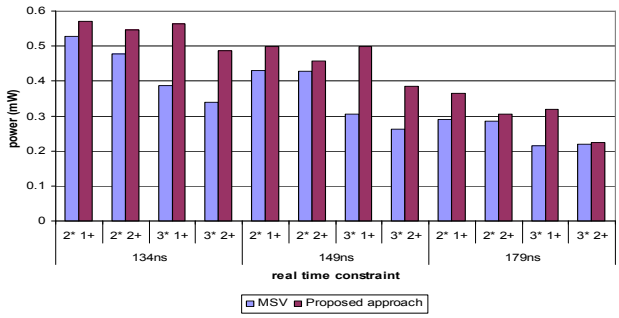

Figure 3. Power consumption of DIFFEQ using MSV[5] and the proposed approach

MSV for DIFFEQ. For example, the power consumption for 2* 1+ for a real time constraint of 134ns is 0.57mW with the presented algorithm, whilst it is 0.53mW with MSV. For EWF, the proposed approach obtains comparable or slightly higher power consumption values (25% average). It can be concluded from the performed experiments that the proposed approach obtains comparable power values with the time and resource constrained MSV technique.

C. Exp. 3 - Comparison with TCS

The aim of this section is to demonstrate the benefits of using the proposed approach compared with TCS algorithms [11] that target area optimisation but that are not power- aware. Applying the proposed algorithm to the benchmarks DIFFEQ and EWF has shown that the power savings increase with increasing time-constraint. For example, in the case of the EWF with 1* 2+, the power savings are approximately 5.5% when the time constraint is 403ns, but increase to 22.8% when the time constraint is 507ns. The voltages applied are 1.54V and 1.24V respectively. Note that the described power savings were obtained without adding new resources. The achieved power reduction is due to an appropriate selection of clock period and execution times of the multicycled FUs. Fig. 4 shows an even higher power reduction but at the expense of additional functional resources when compared to [11].

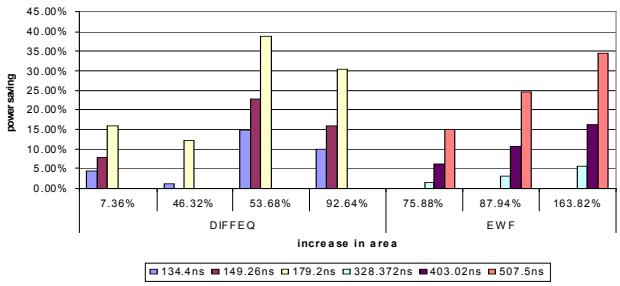

Figure 4. Increase in area and power savings compared with TCS [11]

V. CONCLUSIONS

This paper presented a new TCS algorithm capable of finding trade-offs between power consumption and area under real time constraints. It has been shown that power consumption and area have a non-linear relation, thus resulting in a large and complex search space. The proposed algorithm is capable of exploring this search space in an efficient way and with reasonable computational time. Relevant power-area tradeoffs are possible because of the careful choice of clock period and multicycled functional unit execution time, and the generated single supply voltage. The combination of these three parameters in the proposed approach is essential to obtain low power and area designs. It has been shown that power savings comparable to MSV approaches are achievable whereby the proposed approach leads to lower implementation complexity (single supply voltage versus multiple supply voltages).

ACKNOWLEDGMENT

The authors gratefully acknowledge Dr. Paul T. Gonciari for useful comments and suggestions on this paper.

REFERENCES

[1] R. Mehra and J. Rabaey, "Behavioural level power estimation and exploration", in Proc. International Workshop Low Power Design, Apr. 1994, pp. 197-202

[2] D. D. Gajski, Principles of Digital Design, Prentice Hall, 1997, pp. 363

[3] A. Raghunathan and N. K. Jha, "SCALP: An Iterative-Improvement_Based Low-Power Data Path Synthesis System", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems", Vol. 16, Issue: 11, Nov. 1997, pp.1260-1277

[4] L. Goodby, A. Orailoglu, P. M. Chau, "Microarchitectural Synthesis of Performance-Constrained, Low Power VLSI Designs", ACM Transactions on Design Automation of Electronic Systems, Vol. 7, Issue: 1, Jan. 2002, pp.122-136

[5] A. Manzak and C. Chakrabarti, "A Low Power Scheduling scheme With Resources Operating at Multiple Voltages", IEEE Transactions on Very Large Scale Integration Systems, Vol. 10, Issue 1, Feb. 2002, pp. 6 – 14.

[6] J. M. Chang and M. Pedram, "Energy Minimization Using Multiple Supply Voltages", IEEE Trans. on Very Large Scale Integration Systems, Vol. 5, Issue: 4, Dec. 1997, pp. 436-443.

[7] A. Kumar, M. Bayoumi and M. Elgamel, "A methodology for low power scheduling with resources operating at multiple voltages", INTEGRATION, the VLSI journal 37, 2004, pp. 29-62

[8] M.C. Johnson and K. Roy, "Data path Scheduling with Multiple Supply Voltages and Level Converters", ACM Transactions on Design Automation of Electronics Systems, Vol. 2, No. 3, 1997, pp. 227-248.

[9] W. T. Shiue and C. Chakrabarti, "Low Power Scheduling with Resources Operating at Multiple Voltages", IEEE Transactions on Circuits and Systems II. Analog and Digital Signal Processing, Vol. 47, Issue 6, June 2000, pp. 536-543.

[10] R. San Martin and J. P. Knight, "Optimising Power in ASIC Behavioural Synthesis", IEEE Design & Test of Computers, Vol. 13, Issue: 2, summer 1996, pp. 58-70

[11] P. Kollig, B.M. Al-Hashimi and K.M. Abbott, "Efficient scheduling of behavioural descriptions in high-level synthesis", IEE Proceedings Computer Digital Techniques, Vol. 144, No. 2, March 1997, pp. 75-82