Title

Lower bounds on the switching activity in scheduled data flow graphs with resource constraints

Authors
Kruse, Lars
Schmidt, Eike
Jochens, Gerd
et al.

Publication Date
1999

Peer reviewed

Lower Bounds on the Switching Activity in Scheduled Data Flow Graphs with Resource Constraints

Technical Report ICS-99-33
August 1999

Lars Kruse
Eike Schmidt
Gerd Jochens
Wolfgang Nebel

OFFIS Research Institute
26121 Oldenburg, Germany

kruse@OFFIS.Uni-Oldenburg.DE

Lower Bounds on the Switching Activity in Scheduled Data Flow Graphs with Resource Constraints

Lars Kruse, Eike Schmidt, Gerd Jochens, Wolfgang Nebel

OFFIS Research Institute
26121 Oldenburg, Germany
Kruse@OFFIS.Uni-Oldenburg.DE

Abstract

This paper addresses the problem of estimating lower bounds on the switching activity in scheduled data flow graphs with a fixed number of allocated resources prior to binding. The estimated bound takes into account the effects of resource sharing. It is shown that by introducing Lagrangian multipliers and relaxing the low power binding problem to the Assignment Problem, which can be solved in $O(n^3)$, a tight and fast computable bound is achievable. Experimental results show the quality of the bound. In most cases, deviations smaller than 5% from the optimal binding were observed. The proposed technique can be applied in branch and bound high-level synthesis algorithms for efficiently pruning the design space.

1. Introduction

For most problems in high-level synthesis (HLS) no polynomial time algorithms are known [1]. In order to find optimal or near optimal solutions for this class of problems strategies like branch and bound are applied. A branch and bound algorithm traces a *decision tree* whose leaves represent all possible solutions. Design decisions are made at each internal node while the leaves of the subtree rooted at an internal node are the solutions due to that decision. Given a best solution found during execution of the branch and bound algorithm, a subtree can be pruned if a lower bound estimate of the cost function of all solutions of the subtree is higher than the cost of the current best solution. Tight and fast computable lower bounds therefore improve the run time requirements of such algorithms.

This paper addresses the problem of lower bound estimates for low power HLS and related applications. In particular, a lower bound estimation procedure for the switching activity at the inputs of datapath resources, i.e. registers and functional units (FUs) like adders and multipliers, in scheduled data flow graphs (DFGs) with resource constraints for a given input data stream is given. In the assumed design flow the binding of operations and variables to functional units and registers respectively follows allocation and scheduling. This is a typical flow if resource constrained scheduling is performed. Conditional branches and loops within a DFG are not considered here. Different bindings produce most probably different datapath activities due to the varying data multiplexing schemes if resources are shared.

With switching activity we mean the average Hamming distance of consecutive input vectors. Most HLS for low power algorithms use the switching activity at the inputs of datapath resources or simple functions thereof as a cost function of the power consumption of the design [2,3,4]. The switching activity is a good indicator of the power requirements [5] and often the only power indicating information available at the higher levels of abstraction as considered here. However, the lower bound estimation procedure can also be applied with more accurate information. For example, switched capacitance estimates could be used in case the resource types (e.g. a CLA scheme for adders etc.) are fixed [6,7].

The remainder of the paper is organized as follows: section 2 describes the relation of our approach to previous work. In section 3 the representation and calculation of the switching activity information is introduced. The new lower bound estimation procedure is presented in section 4. Section 5 shows experimental results and conclusions are drawn in section 6.

2. Previous Work

Lower bound estimation (LBE) techniques are often applied to guide HLS. As examples, the authors of [8] present procedures to estimate lower bounds on the resource requirements from a given DFG with a performance goal. In [9] a technique is described that estimates a lower bound on the performance of schedules from a DFG with resource constraints. To the best of our knowledge, LBE techniques for low power at the higher levels of abstraction are first addressed in [10]. Lower and upper bound estimation procedures are given for scheduled DFGs without resource constraints. This paper extends the work of [10] by improving the bounds if the number of resources is constrained.

Some researchers have addressed HLS for low power problems that are closely related to our work. In [2] the problem of binding the n variables of a DFG to m registers under the constraint of minimum switching activity at the register inputs is formulated as a max-cost network flow problem. The problem can be solved in $O(mn^2)$. The drawback of this approach, however, is that inter-iteration switching activity cannot be considered. Inter-iteration activity is defined as the switching activity resulting from successive executions of the DFG. For instance, let:

- x^i, $x \in \{a, b, c, d, e\}$, be the value of the binary variable x at iteration i of the DFG,
- $x|y$ be the concatenation of the binary variables x and y,
- $hd(x, y)$ be the Hamming distance between the values of variables x and y.

Suppose that operations $+_1$ and $+_3$ of the DFG depicted in Fig. 1 are bound to one adder. One part of the switching activity at the inputs of the adder is:

$$\sum_{i=1}^{T} hd(a^i|b^i, d^i|c^i), \text{ (intra-iteration activity)},$$

where T is the length of the input stream. The inter-iteration part is defined as:

$$\sum_{i=1}^{T-1} hd(d^i|c^i, a^{i+1}|b^{i+1}),$$

e.g. the switching from the values of iteration i to the new ones of iteration $i+1$.

The same authors investigate the problem of binding operations to a fixed number of resources in a functionally pipelined DFG taking inter-iteration effects into account [3]. Due to the inter-iteration constraint the problem can only be transformed to a max-cost multi-commodity network flow problem which is in general not solvable in polynomial time. An integer linear program (ILP) for the problem of binding n operations/variables to m functional units/registers is formulated in [6]. The inter-iteration effects are considered but no polynomial time algorithm for solving the ILP is given.

3. Switching Activity

The switching activity representation follows the approach described in [10] that originates from the work presented in [2,3,6]. A square *switching activity matrix* (*SAM*) for the variables and for each operation type present in the DFG (e.g. addition, subtraction, multiplication, etc.) is defined. The lower triangular entries including the main diagonal contain the inter-iteration activity between all pairs of

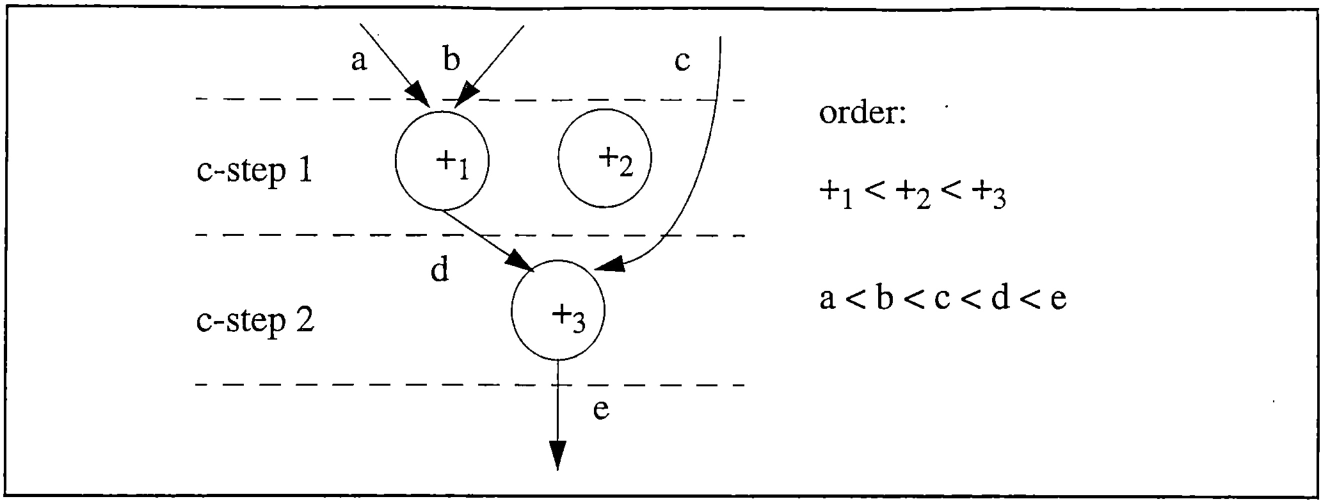

Figure 1: DFG with operation and variable order.

operations and variables respectively. The upper triangular part stores intra-iteration activity. Each operation respectively variable is associated with one column and one row having the same index. The column and row ordering equals a total operation (variable) ordering which can be induced from the given schedule as depicted in Fig. 1. Let $csb(op_i)$ denote the first c-step of operation op_i in the schedule. Then $op_i < op_j$ if $csb(op_i) < csb(op_j)$, e.g. the execution of op_j follows that of op_i within one execution of the DFG. In case p operations $op_1, \ldots, op_p$ are not *compatible*, e.g. start in the same c-step $(csb(op_1) = \ldots = csb(op_p))$, then they can be put in arbitrary order, for example $op_1 < \ldots < op_p$. The same ordering can be defined on the set of variables by replacing $csb(op_i)$ with the birth time of that variable [1]. In the sequel, we only deal with operations and index all n operations of type r according to their ordering:

$$op_1^{(r)} < \ldots < op_n^{(r)}$$

Switching activity information about operation op_i is stored in column and row i of the SAM of the corresponding operation type. An entry $SAM(i, j)$, $i, j \in \{1, \ldots, n\}$ is set to infinity $(+\infty)$ if operations op_i and op_j are not compatible and therefore cannot share a resource. Otherwise if $i < j$ (remember that from $i < j$ follows $op_i < op_j$), the entry stores the average Hamming distance between the input vectors of operations op_i and op_j from the same iteration. If $i > j$, $SAM(i, j)$ is set to the average Hamming distance between the inputs of op_i from iteration t and the inputs of op_j from iteration $t + 1$ (inter-iteration activity). The elements on the main diagonal $SAM(i, i)$ store the activity at the inputs of operation op_i, e.g. the activity at the inputs of a FU if only op_i is bound to it. Formally:

$$SAM(i, j) = \begin{cases} \infty \text{ , } op_i, op_j \text{ are not compatible} \\ \frac{1}{T}\sum_{t=1}^{T} hd(op_i(t), op_j(t)) \text{ , } i < j \\ \frac{1}{T}\sum_{t=1}^{T} hd(op_i(t), op_j(t+1)) \text{ , } i \geq j \end{cases}$$

with:

- $op_i(t)$ the concatenation of the input vectors of operation op_i in iteration t of the DFG, and
- T the total number of vectors in the input data stream, e.g. the number of iterations of the DFG.

For example, the average switching activity per DFG iteration at the inputs of a resource with operations op_1, op_2, op_3 mapped onto it in that order can now be computed by $SAM(1, 2) + SAM(2, 3) + SAM(3, 1)$ [6]. The Hamming distances can be computed by simulating the entire DFG with the input stream or by using statistical techniques as for example proposed in [2,3]. Note that two dimensions suffice to store all necessary information. If $op_i < op_j$ holds than the data $op_j(t + 1)$ can never directly follow $op_i(t)$ at the inputs of a FU.

4. Binding for Low Power with Resource Constraints

4.1. Problem formulation

For a given operation type we define the *low power binding problem with resource constraints* as follows:

Given a switching activity matrix $(SAM(i, j))_{i,\, j \in \{1, \ldots, n\}}$ *of n operations* $op_1, \ldots, op_n$. *Which is the minimum sum of switching activity at the resources inputs if these n operations are bound to m resources.*

An equivalent problem can be stated for binding n variables to m registers. There are

$$\frac{1}{m!} \cdot \left(m^n - \left((m-1)^n \binom{m}{m-1} - \left(\ldots -1^n \binom{m}{1}\right)\ldots\right)\right)$$

possibilities to map n operations onto m resources if all n operations are compatible. The proof is omitted due to space limits. For example, there are more than $4 \cdot 10^{10}$ combinations to map 20 operations onto 4 resources.

The low power binding problem with resource constraints can be expressed as a graph problem by defining an arc labeled directed graph $G(V, A)$ with $V = \{op_1, \ldots, op_n\}$ the set of nodes (one node for each operation) and $A = V \times V$ the set of arcs. Each arc $(op_i, op_j) \in A$ is labeled with a weight $w_{ij} = SAM(i, j)$. The optimization problem is then to cover all nodes with exactly m (node disjoint) cycles with minimum total cost under the constraint that each cycle contains exactly one *backward arc*, e.g. an arc (op_i, op_j) with $i \geq j$. The total cost is the sum of the arc weights of all cycles. Each cycle of a solution to this problem represents one resource while the nodes of a cycle are the operations bound to it. A possible solution of the optimization problem with four operations and two resources is depicted in Fig. 2. $(+_4, +_1)$ and $(+_3, +_3)$ are the two backward arcs.

The constraint that each cycle $(op_{k_1}, \ldots, op_{k_p}, op_{k_1})$, $k_1 < \ldots < k_p$ has to have exactly one backward arc reflects the precedence constraints of the operations within the schedule of the DFG. Inter-iteration switching activity ($SAM(k_p, k_1)$ in this case) can only occur after all operations $op_{k_1}, \ldots, op_{k_p}$ are executed in one iteration of the DFG on one resource. Loops (op_i, op_i) represent resources with exactly one operation op_i bound to it.

4.2. Bounding the solution space

We first repeat Theorem 4.1 without proof from [10] which defines a lower bound on the switching activity of the low power allocation and binding problem, i.e. the binding problem without resource constraints:

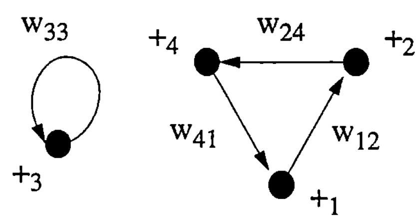

Figure 2: Possible solution of a low power binding problem with resource constraints by covering operation nodes with cycles.

Theorem 4.1 *A solution of the integer linear program*

$$z = \min \sum_{i,j=1}^{n} w_{ij} x_{ij}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n \qquad (4.1.A)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1, \ldots, n \qquad (4.1.B)$$

$$x_{ij} \geq 0 \text{ integer variables}$$

provides a lower bound of the low power allocation and binding problem with switching activity matrix $SAM(i, j) = w_{ij}$.

In this ILP, the (binary) variable x_{ij} is associated with arc (op_i, op_j). The solutions to the ILP describe node disjoint cycles covering all nodes. x_{ij} equals 1 in a solution if and only if the corresponding arc belongs to a cycle, otherwise the variable is zero. However, it is not guaranteed that no cycle has more than one backward arc. Hence a solution to the ILP delivers only a lower bound z on the switching activity and not necessarily the minimum.

As stated in [10] the ILP of Theorem 4.1 can be efficiently solved by the *Hungarian Method* in $O(n^3)$ because it describes the *Assignment Problem* [11].

The following ILP improves the lower bound of Theorem 4.1 with additional constraints on the number of resources:

Theorem 4.2 *A solution of the integer linear program*

$$z = \min \sum_{i,\, j = 1}^{n} w_{ij} x_{ij}$$

subject to

$$\sum_{j = 1}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n \qquad (4.2.A)$$

$$\sum_{i = 1}^{n} x_{ij} = 1 \qquad j = 1, \ldots, n \qquad (4.2.B)$$

$$\sum_{i \geq j} x_{ij} = m \qquad (4.2.C)$$

$$x_{ij} \geq 0 \text{ integer variables}$$

provides a lower bound of the low power binding problem with m resources and switching activity matrix $SAM(i, j) = w_{ij}$.

Proof: Constraints 4.2.A and 4.2.B are identical to the constraints 4.1.A and 4.1.B respectively and guarantee that all nodes are covered by node-disjoint cycles. 4.2.C insures that exactly m backward arcs are included in a solution of the ILP. However, no constraints exist that force each cycle to have exactly one backward arc which is a necessary condition of the low power binding problem with resource constraints. The ILP is therefore a relaxation of the original problem and a solution provides a lower bound q.e.d.

Instead of solving the ILP of Theorem 4.2 directly a polynomial time bounded approach is proposed which approximates the ILP from below based on Lagrangian Relaxation, i.e. the original problem is relaxed two times. Lagrangian Relaxation explores the fact that for a given ILP

$$z = \min w^T x$$
$$\text{subject to } Ax = b, Bx = d$$
$$x \geq 0, x \text{ integer}$$

a solution of

$$L(y) = \min(w^T x + (Bx - d)^T y)$$
$$\text{s.t. } Ax = b, x \geq 0, x \text{ integer}$$

provides a lower bound for z for all values of y. y is called vector of *Lagrangian multipliers*. This property follows because all feasible solutions of the first ILP are also feasible for the last one with the same objective function value. The best lower bound is found by maximizing $L(y)$, i.e. solving:

$$v = \max_y L(y)$$

The maximization can be performed with *subgradient maximization* [12]. k is *subgradient* of L at x iff $k^T(y-x) \geq L(y) - L(x)$ for all y. It can be shown that if $\hat{x}$ minimizes $L(y)$ for some y than $B\hat{x} - d$ is a subgradient of $L(y)$. The *subgradient method* iteratively solves L at points:

$$y_{p+1} = y_p + t_p q_p$$

where q_p is a subgradient of L at y_p and t_p is a suitable step width in that direction ($p \geq 0$, $y_0 = 0$).

Applying the Lagrangian method to the ILP of Theorem 4.2 by relaxing 4.2.C delivers:

$$\begin{aligned} L(y) &= \min(\textstyle\sum_{i,j} w_{ij} x_{ij} - (\sum_{i \geq j} x_{ij} - m) y) \\ &= \min(\textstyle\sum_{i,j} w_{ij} x_{ij} - \sum_{i \geq j} y x_{ij}) - my \\ &= \min \textstyle\sum_{i,j} w_{ij}' x_{ij} - my \end{aligned}$$

$$\textit{with} \qquad w_{ij}' = \begin{cases} w_{ij} & \text{if } i < j \\ w_{ij} - y & \text{if } i \geq j \end{cases}$$

subject to 4.2.A and 4.2.B. $L(y)$ can again be solved by the *Hungarian Method* in $O(n^3)$ with the objective function $\min \sum_{i,j} w_{ij}' x_{ij}$ and subtracting the constant my afterwards. Note the similarity to the ILP of Theorem 4.1. The subgradient at iteration p is $q_p = \sum_{i \geq j} \hat{x}_{ij} - m$ if $\hat{x}_{ij}$ is the minimizer of $L(y_p)$, i.e. the number of backward arcs of the solution minus the number of resources.

5. Experimental Results

The conditions $\lim_{p \to \infty} t_p = 0$ and $\sum_{p=0}^{\infty} t_p = \infty$ for the sequence of step widths t_p guarantee that $L(y)$ converges to its maximum [12]. But fulfilling them might result in very slow convergence rates. An approximation is used throughout the experiments instead by dropping the condition $\sum_{p=0}^{\infty} t_p = \infty$. The sequence $t_p = 0.95\, t_{p-1}$ with $t_0 = 0.8$ does not guarantee convergence but delivers good approximations of the ILP solution of Theorem 4.2 in a reasonable number of iterations. Because $L(y_p)$ is not monotonely increasing with increasing p the maximum of $L(y_p)$ for $p = 1, \ldots, p_{max}$ is reported in the results as the best lower bound if convergence is not reached.

The experiments were performed on 2 benchmarks investigating the binding of additions and multiplications: a one dimensional FDCT [13] as a part of a 2D-FDCT transforming an image of buildings (13 additions and 16 multiplications) and the Elliptic Wave Filter (EWF) HLS benchmark as specified in [14] with modified coefficient set and a speech signal as input (10 additions and 12 multiplications). Table 1 shows the results for the switching activity at the inputs of adders for a sequential schedule, i.e. only one addition per c-step, depending on the number of allocated resources. The trivial cases of $m = 1$ and $m = n$ are not considered (the first is trivial due to the precedence constraints of the operations). Column 2 shows the number of iterations used in the subgradient method. Experiments that did not converge to the maximum of $L(y_p)$ within 10 iterations were interrupted. The third column shows the switching activity per DFG iteration obtained by the subgradient method while column 4 depicts the results by solving the ILP of Theorem 4.2 directly. The deviation of the subgradient method from the ILP solutions are given in column 5. The activity of the best possible binding, found by exhaustive search, is presented in column 6. The deviation of the subgradient method from the best solution is shown in the last column. Solving the ILP delivers in all cases the best possible binding while the subgradient method efficiently approximates them very well within a few iterations. No deviation is larger

# res.	p_{max}	*Lag.*	*ILP*	*Dev.*	*Best*	*Dev.*
2	*9*	*102.3*	*102.3*	*0%*	*102.3*	*0%*
3	*5*	*98.53*	*98.53*	*0%*	*98.53*	*0%*
4	*10*	*95.16*	*95.69*	*0.6%*	*95.69*	*0.6%*
5	*10*	*91.81*	*92.27*	*0.5%*	*92.27*	*0.5%*
6	*10*	*89.48*	*89.52*	*0.1%*	*89.52*	*0.1%*
7	*10*	*86.73*	*87.10*	*0.4%*	*87.10*	*0.4%*
8	*10*	*84.17*	*85.52*	*1.6%*	*85.52*	*1.6%*
9	*4*	*81.90*	*81.90*	*0%*	*81.90*	*0%*
10	*9*	*81.26*	*81.26*	*0%*	*81.26*	*0%*
11	*10*	*80.98*	*81.20*	*0.3%*	*81.20*	*0.3%*
12	*10*	*80.45*	*81.02*	*0.7%*	*81.02*	*0.7%*

Table 1: Sequential schedule of adders (FDCT)

than 2%. If the maximum number of iterations is restricted to 2, 3, 5, and 7 respectively, the deviations shown in table 2 are obtained. Solving the Assignment Problem only 3 times gives approximations that are at most 6% off the best binding.

Table 3 shows the results for binding the multiplications of the FDCT. The number of iterations is restricted to 5. The deviations are larger because the ILP has minimum solutions that represent cycles having more than one backward arc. The error thus does not stem from the subgradient method but from the ILP description of the problem.

Table 4 presents the deviations from the optimal binding for 3 different schedules of the multipliers (columns 2 to 4) and for sequential schedules of the additions and multiplications of the EWF benchmark. The results reflect the robustness of the presented approach.

In order to measure the cpu time requirements, we generated larger SAMs with random contents. For $n = 60, m = 5$ and $n = 200, m = 20$ with 10 iterations each the proposed technique required 0.2 s and 8.1 s, respectively (Ultra-Sparc 10, 300 MHz). For comparison: exhaustive search was only feasible upto $n = 16$. It required more than 8 hours to find the optimal binding for $n = 16, m = 8$.

# res.	$p_{max} = 2$	$p_{max} = 3$	$p_{max} = 5$	$p_{max} = 7$
2	4.4%	3.6%	2.2%	1.0%
3	8.0%	4.9%	0%	0%
4	11.1%	4.0%	0.6%	0.6%
5	12.2%	0.5%	0.5%	0.5%
6	10.1%	2.5%	2.5%	2.5%
7	7.6%	6.0%	5.1%	0.4%
8	5.9%	5.9%	2.8%	1.6%
9	1.8%	1.8%	0%	0%
10	1%	1%	0.1%	0.1%
11	0.9%	0.7%	0.6%	0.4%
12	0.7%	0.7%	0.7%	0.7%

Table 2: Sequential schedule of adders (FDCT). The number of iterations in subgradient method is bounded.

6. Conclusion

This paper presented a fast estimation technique that provides tight lower bounds on the switching activity at resource inputs for a given schedule with resource constraints. Other cost functions can be applied by simply changing the definition of the switching activity matrix. The low power binding problem under resource constraints was formulated and relaxed to the Assignment Problem with Lagrangian multipliers. A few number of iterations suffice to get estimates that are very close to the best possible solution.

# res.	p_{max}	Lag.	ILP	Dev.	Best	Dev.
2	2	90.59	90.59	0%	90.59	0%
3	5	84.88	84.88	0%	84.88	0%
4	5	79.96	79.96	0%	81.50	1.9%
5	3	75.06	75.06	0%	78.65	4.6%
6	5	71.72	72.30	0.8%	76.68	6.5%
7	5	68.29	68.61	0.5%	75.28	9.3%
8	2	65.09	65.09	0%	72.44	10.2%
9	4	63.75	63.75	0%	70.46	9.5%
10	5	62.69	62.92	0.4%	68.49	8.5%
11	5	62.46	63.21	1.2%	67.15	7.0%
12	1	62.39	62.39	0%	66.33	5.9%
13	5	62.46	63.74	2.0%	65.73	5.0%
14	5	62.69	63.15	0.7%	65.15	3.8%
15	5	63.25	64.65	2.2%	64.65	2.2%

Table 3: Sequential schedule of multipliers (FDCT)

# res.	*FDCTMUL 1*	*FDCT MUL2*	*FDCT MUL3*	*EWF ADD*	*EWF MUL*
2	-	-	-	0.4%	0.6%
3	-	-	-	0.7%	0.4%
4	-	0.4%	0.9%	0.7%	2.1%
5	-	0%	1.3%	0.7%	2.6%
6	-	0.3%	1.2%	1.3%	4.5%
7	-	0.2%	3.6%	0.7%	4.1%
8	0%	0.6%	6.6%	0.4%	3.3%
9	0.2%	3.6%	10.7%	0.4%	0.8%
10	0.4%	3.0%	10.5%	-	0%
11	2.1%	3.1%	8.6%	-	0%
12	0.9%	2.9%	5.9%	-	-
13	0.2%	3.2%	5.1%	-	-
14	0%	0%	3.8%	-	-
15	0.3%	1.9%	2.2%	-	-

Table 4: Three different schedules of multiplications (FDCT) and sequential schedules of additions and multiplications (EWF).

7. References

[1] G. De Micheli, "Synthesis and Optimization of Digital Circuits", McGraw-Hill, 1994

[2] J.-M. Chang, M. Pedram, "Low Power Register Allocation and Binding", Proc. of the DAC 1995

[3] J.-M. Chang, M. Pedram, "Module Assignment for Low Power", Proc. of the Euro-DAC 1996

[4] E. Musoll, J. Cortadella, "Scheduling and Binding for Low Power", Proc. of the ISSS, 1995

[5] G. Jochens, L. Kruse, E. Schmidt, W. Nebel, "A New Parameterizable Power Macro-Model for Datapath Components", Proc. of DATE 1999

[6] A. Raghunathan, N.K. Jha, "An ILP formulation for low power based on minimizing switched capacitance during datapath allocation", Proc. of the Int. Symposium on Circuits & Systems, 1995

[7] P.E. Landman, J.M. Rabaey, "Activity-Sensitive Architectural Power Analysis", IEEE Transactions on CAD of Integrated Circuits and Systems, 15(6), June 1996

[8] S.Y. Ohm, F.J. Kurdahi, N.D. Dutt, "A Unified Lower Bound Estimation Technique for High-Level Synthesis", IEEE Transactions on CAD, 16(5), May 1997

[9] M. Langevin, E. Cerny, "A Recursive Technique for Computing Lower-Bound performance of Schedules", ACM Transactions on Design Automation of Electronic Systems, 1(4), October 1996

[10] L. Kruse, E. Schmidt, G. Jochens, W. Nebel, "Lower and Upper Bounds on the Switching Activity in Scheduled Data Flow Graphs", submitted to ISLPED 1999

[11] C.H. Papadimitriou, K. Steiglitz, "Combinatorial Optimization, Algorithms and Complexity", Prentice Hall, Inc., 1982

[12] B.T. Polyak, "Subgradient Methods: A Survey of Soviet Research" in C. Lemarechal, R. Mifflin (eds.) "Nonsmooth Optimization", pp. 5 - 29, Pergamon Press, Oxford, 1978

[13] K.R. Rao, P. Yip, "Discrete Cosine Transform - Algorithms, Advantages, Applications", pp. 378 - 380, Academic Press, Inc., 1990

[14] A. Raghunathan, N.K. Jha, S. Dey, "High-Level Power Analysis and Optimization", p. 25, Kluwer Academic Publishers, 1998