Integrated High-Level Synthesis and Power-Net Routing for Digital Design under Switching Noise Constraints*

Alex Doboli
Department of ECE
State University of New York at Stony Brook
Stony Brook, NY, 11794-2350
adoboli@ece.sunysb.edu

Ranga Vemuri
Department of ECECS
University of Cincinnati
Cincinnati, OH, 45221-0030
Ranga.Vemuri@uc.edu

ABSTRACT

This paper presents a CAD methodology and a tool for high-level synthesis (HLS) of digital hardware for mixed analog-digital chips. In contrast to HLS for digital applications, HLS for mixed-signal systems is mainly challenged by constraints, such as digital switching noise (DSN), that are due to the analog circuits. This paper discusses an integrated approach to HLS and power net routing for effectively reducing DSN. Motivation for this research is that HLS has a high impact on DSN reduction, however, DSN evaluation is very difficult at a high level. Integrated approach also employs an original method for fast evaluation of DSN and an algorithm for power net routing and sizing. Experiments showed that our combined binding and scheduling method produces better results than traditional HLS techniques. Finally, DSN evaluation using the proposed algorithm can be significantly faster than SPICE simulation.

1. INTRODUCTION

Single chip integration of analog and digital hardware offers critical advantages such as cost reduction and yield enhancement with an improvement in high-frequency performance. Design of mixed-signal chips with existing analog and digital synthesis tools is not satisfactory in most of the cases. There is a very high chance that performances of the separately synthesized digital hardware do not match design requirements of analog circuits. For example, encouraging operation concurrency for satisfying throughput constraints might result in a high digital switching noise (DSN), that perturbs a correct functioning of analog blocks. Hence, for mixed-signal applications, digital synthesis must contemplate not only traditional performances (such as area, speed, power) but also new constraints imposed by analog circuits (such as low DSN). This constitutes a completely new research problem and is the topic of this paper.

Digital switching noise (DSN) represents a major design concern for mixed-signal applications [1] [13]. DSN is generated by the switching of digital components. Its propagation path to noise-sensitive analog blocks is mainly through substrate and power net. The amount of noise that can be accommodated at the power pins of an analog block is characterized by its *Power Supply Rejection Ratio* (PSRR) [1]. A high DSN can compromise the correct functioning of analog circuits. Moreover, analog circuits with a high PSRR are difficult to design and require more silicon area. Thus, digital hardware has to be designed so that its DSN is less than the limits acceptable to analog circuits.

DSN depends on a large number of design aspects such as operation scheduling and binding, hardware resource placement and power net routing [1]. Hence, DSN has to be addressed by design activities at various levels of abstraction, including high level synthesis (HLS) and physical synthesis. The difficulty, however, of HLS under noise constraints is that DSN can be effectively evaluated (estimated) only at the physical level, after power net routing is accomplished [1] [13]. An abstract modeling of DSN through metrics such as number of concurrently executed operations is not acceptable. Small variations in DSN can dramatically perturb the functioning of analog blocks. This motivates the need for an integrated approach to HLS and power net routing so that design decisions can be contemplated at a high level without loosing the possibility of precise DSN evaluation.

This paper discusses the problem of digital HLS for mixed analog-digital applications. We consider that HLS accommodates an *analog-domain centered* design approach in which the analog part is fully designed (until the layout level) before digital HLS starts. Hence, DSN constraints demanded by analog circuits are exactly known. Digital domain needs to be synthesized so that its DSN is acceptable for analog circuits and its throughput is minimized. To the best of our knowledge, no previous work discusses the issue of coping with DSN during digital HLS. Following are the principal contributions of the paper:

- **Tight integration of HLS and power net routing**: DSN of each RTL solution produced by HLS is rapidly evaluated by performing power-net routing and sizing. The proposed algorithm routes power pins of digital circuits depending on the amount of DSN originating at a pin. Tight integration is required for accurately calculating DSN during HLS. This secures a correct functioning of the final analog-digital design.

*This work was sponsored by the USAF, Air Force Research Laboratories, Wright Patterson Air Force Base under contract number F33615-96-C-1911

DAC 2001, June 18-22, Las Vegas, Nevada, USA.

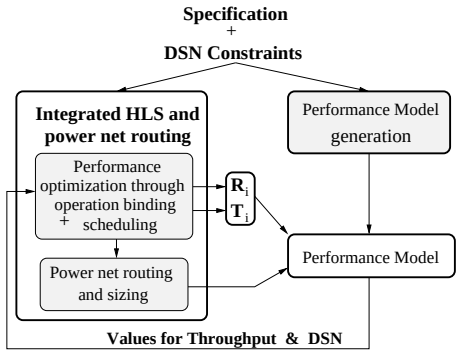

Figure 1: Integrated HLS and power net routing

- **Original technique for DSN evaluation**: *Symbolic* models are automatically derived for noise propagation paths. If current variations due to digital switchings are known then transient voltage variations at the power pins of analog blocks can be computed using these symbolic models. We are not aware of any other DSN evaluation technique for HLS. Experiments showed that symbolic models permit DSN evaluation with the same accuracy as SPICE simulation in a significantly shorter time.
- **New method for combined operation binding and scheduling**: Figure 1 depicts the proposed HLS methodology. In a first step, **performance models** (PM) are generated for an application. PMs are graphs that relate performances i.e. throughput to design variables specific to operation binding (variable R_i in Figure 1 shows the bound resource) and scheduling (variable T_i in the figure indicates the starting time). The second step is **performance optimization** through *Simulated Annealing* algorithm (SA) [11] guided by DSN and throughput PM evaluation. This step is the actual HLS as different binding and scheduling alternatives are produced. Experiments show that superior quality solutions resulted for our method than for traditional techniques. Also, the methodology eliminated the tedious task of defining new performance metrics for considering DSN, also.

We adopted following two assumptions for the proposed HLS methodology. (1) Set of digital hardware resources used for implementation is known. Each digital resource is characterized by a *noise signature* that indicates variations of the current at its power lines due to output switches [4]. Noise signatures are required for calculating DSN at power pins of analog blocks. (2) Analog and digital resources are pre-placed before HLS starts so that a simpler, hence, less time demanding physical synthesis results inside the HLS loop. This assumption is valid as present mixed-signal designs have a small number of resources that can be manually pre-placed by a designer [13].

The paper is organized as six sections. Related work is discussed next. Section 3 explains DSN evaluation. HLS, power net routing, and performance models for throughput are introduced in Section 4. Section 5 presents experimental results. Finally, we provide our conclusions.

2. RELATED WORK

Research on mixed-signal CAD exclusively focuses on physical-level topics such as placement, signal and power-net routing [13]. Obviously, next step for research on mixed-signal CAD

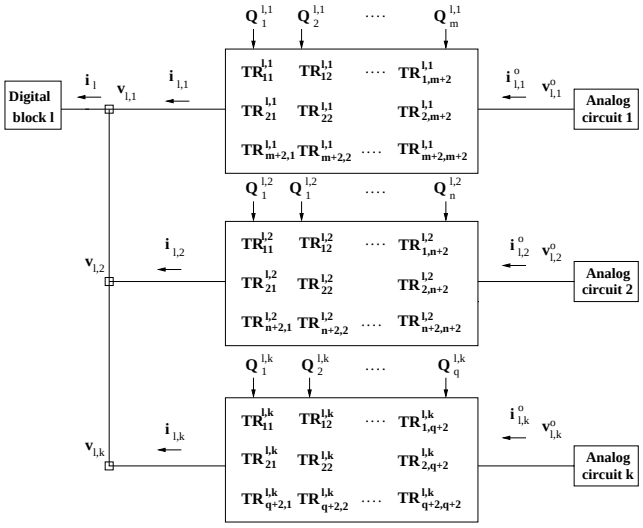

Figure 2: Model for single DSN source

is to consider higher-levels of abstraction. This implies that analog and digital domains must be tackled together even at higher abstraction levels. Benefit is that performances that affect both domains can be approached in a unitary, hence, more effective way. As motivated by Stanisic *et al* [13], difficulty of digital HLS for mixed-signal applications originates in accommodating physical-level aspects i.e. DSN at higher levels of abstraction. We believe that a tight integration of HLS and power net routing permits efficient high-level synthesis under DSN constraints.

Most of the existing digital HLS methods [5] [7] [9] [10] are heuristics guided by specific priority/cost functions i.e. mobility, urgency, force etc, acting as approximate performance models. Addressing new performances i.e. DSN, however, involves a very tedious task of defining and validating characteristic priority functions. Moreover, heuristic techniques are less effective for certain types of applications or if design factors have a positive impact on some performances and a negative influence on other [12]. Finally, HLS and physical synthesis are *successive* activities. This prevents HLS from addressing any physical-level aspects such as interconnection, DSN minimization etc, and results in poor quality designs [2] [7].

To the best of our knowledge, the research presented in this paper constitutes the first attempt to develop a digital HLS methodology for mixed-signal applications. Our solution tightly integrates HLS and power net routing. Critical to the methodology is the original DSN evaluation algorithm that we developed. The algorithm is accurate and fast as compared to SPICE simulation. Finally, the combined binding and scheduling method joins advantages of heuristic and exact methods: it uses exact performance models (similar to exact methods) and heuristic techniques for model optimization (similar to heuristic methods). Better quality solutions resulted than for existing heuristic methods i.e. SALSA HLS tool [9] or FDLS algorithm [10].

3. DSN EVALUATION

This section presents modeling of noise propagation paths for transient (time) analysis of DSN. In [6], we discuss how the same concept can be used for AC (frequency) analysis, also.

The two major noise-propagation paths in a mixed-signal IC are (1) coupling through substrate and (2) coupling through power supply lines [13]. Coupling between interconnections (crosstalk) is small as compared to the other two coupling types, if analog and digital hardware are separated at the floorplan level [1]. If digital domain contains p digital circuits then total noise that appears at an analog block can be calculated by the superposition principle [1]. A current spike i_l that originates at the power supply of digital block l determines a voltage variation $v^o_{l,t}$ at power supply of analog block t. Voltage variation due to all digital circuits at power supply of analog block t is

$$v^o_t = \sum_{i=1}^{p} v^o_{i,t}$$

Thus, DSN that is due to each digital circuit and propagates to the power pins of an analog block has to be found.

Figure 2 presents the model for transient analysis. The model includes a digital noise source (digital block l) and noise distribution paths to each of the k analog blocks. Analog and digital power buses are separated in this model. This is a valid assumption for most cases as current design practice separates analog and digital power lines [1] [13]. If power lines are shared (for reducing number of I/O pins) then same reasoning can be applied to evaluate noise. Current $i_l(t)$ is due to the switching of block l and is characterized by the signature of digital block l. Currents $i^o_{l,1}(t), i^o_{l,2}(t), ..., i^o_{l,k}(t)$ are also known and depend on the impedances of the analog blocks (for our case we assumed these currents to be zero). Voltages $v^o_{l,1}(t), v^o_{l,2}(t), ..., v^o_{l,k}(t)$ represent DSN as they are voltage variations due to the current i_l. Symbols $TR^{l,m}_{i,j}$ are matrix coefficients of the two-port representations of DSN propagation paths. Influence of state variables i.e. the charges $Q^{l,i}_p(t-1)$ stored by capacitors has to be captured by the equation set. Calculating charge values is presented in more detail when discussing modeling of power supply buses.

Using Kirchhoff's laws and definitions of matrices $TR^{l,m}_{i,j}$, $m = 1$ to k, following equation was set up:

$$\begin{aligned}
i_l(t) &= i_{l,1}(t) + i_{l,2}(t) + i_{l,3}(t) + \dots + i_{l,k}(t)\\
v_{l,1}(t) &= v_{l,2}(t) = \dots = v_{l,k}(t)\\
v_{l,1}(t) &= TR^{l,1}_{11} i_{l,1}(t) + TR^{l,1}_{12} i^o_{l,1}(t) + \textstyle\sum_{p=1}^{m} TR^{l,1}_{1,p+2} Q^{l,1}_p(t-1)\\
v^o_{l,1}(t) &= TR^{l,1}_{21} i_{l,1}(t) + TR^{l,1}_{22} i^o_{l,1}(t) + \textstyle\sum_{p=1}^{m} TR^{l,1}_{2,p+2} Q^{l,1}_p(t-1)\\
v_{l,2}(t) &= TR^{l,2}_{11} i_{l,2}(t) + TR^{l,2}_{12} i^o_{l,2}(t) + \textstyle\sum_{p=1}^{n} TR^{l,2}_{1,p+2} Q^{l,2}_p(t-1)\\
v^o_{l,2}(t) &= TR^{l,2}_{21} i_{l,2}(t) + TR^{l,2}_{22} i^o_{l,2}(t) + \textstyle\sum_{p=1}^{n} TR^{l,2}_{2,p+2} Q^{l,2}_p(t-1)\\
&\dots\\
v_{l,k}(t) &= TR^{l,k}_{11} i_{l,k}(t) + TR^{l,k}_{12} i^o_{l,k}(t) + \textstyle\sum_{p=1}^{q} TR^{l,k}_{1,p+2} Q^{l,k}_p(t-1)\\
v^o_{l,k}(t) &= TR^{l,k}_{21} i_{l,k}(t) + TR^{l,k}_{22} i^o_{l,k}(t) + \textstyle\sum_{p=1}^{q} TR^{l,k}_{2,p+2} Q^{l,k}_p(t-1)
\end{aligned}$$

Formulas for unknown voltages $v^o_{l,1}(t), v^o_{l,2}(t), ..., v^o_{l,k}(t)$ result by solving this equation set. For analog block i, voltage variation at its power supply line is given by

$$v^o_{l,i}(t) = \frac{TR^{l,i}_{21}}{TR^{l,i}_{11}} \left(\frac{i_l(t) + \sum_{j=1}^{k} \frac{\sum_p TR^{l,j}_{1,p+2} Q^{l,j}_p(t-1)}{TR^{l,j}_{11}}}{\sum_{j=1}^{k} \frac{1}{TR^{l,j}_{11}}} - \sum_p TR^{l,i}_{1,p+2} Q^{l,i}_p(t-1) \right) + \sum_p TR^{l,i}_{2,p+2} Q^{l,i}_p(t-1)$$

Finally, overall voltage variation at power supply line of analog block i is computed by summing voltages $v^o_{l,i}$ due to all digital blocks.

Symbols $TR^{l,j}_{m,n}$ describing each noise propagation path j were calculated for completing DSN modeling. Figure 3 presents the noise propagation path between digital circuit l and analog block j. The figure was used to motivate how coefficients $TR^{l,j}_{m,n}$ relate to the coefficients that characterize the substrate, contacts and analog and digital power buses.

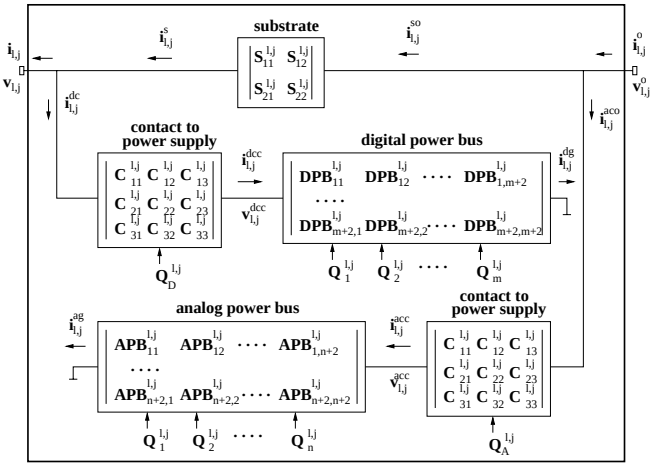

Figure 3: Noise path modeling

Coefficients assume that currents $i_{l,j}$ and $i^o_{l,j}$ are knowns and voltages $v_{l,j}$ and $v^o_{l,j}$ are unknowns. Let's assume that substrate is described by matrix $S^{l,j}$, contact to power supply by matrix $C^{l,j}$, digital power bus by matrix $DPB^{l,j}$ and analog power bus by matrix $APB^{l,j}$. First, we applied Kirchhoff's laws and definitions of matrices of the two-port blocks. Formulas to relate parameters $TR^{l,j}_{m,n}$ to the parameters of substrate, power nets and contacts were obtained by solving this equation set. Due to space limitations, we do not provide the resulting formulas. Following subsections detail how matrices $S^{l,j}$, $C^{l,j}$, $DPB^{l,j}$ and $APB^{l,j}$ relate to the physical and geometrical characteristics of substrates, contacts and power buses.

Coupling through substrate is due to the finite resistivity of substrate and junction capacitances. As application bandwidth is less than 1 GHz, in this paper we adopted the resistive substrate modeling proposed by Joardar [8]. Expressions for resistances are derived depending on their geometrical dimensions [8]. Hence, as the HLS methodology assumes that hardware resources are already placed on the layout, substrate resistances can be accurately extracted. If resistive substrate model is not sufficient then mesh model [13] for substrate can be used.

Stanisic *et al* [13] indicate that power buses can be modeled as a set of RC segments. Relationships that express resistance and capacitance of a segment depending on segment length, height and width are presented in [13]. For transient analysis, influence of state variables $Q^{l,j}_k(t)$ (charges stored by segment capacitances) needs to be captured. To find coefficients $DPB^{l,i}_{m,n}$ of the power bus of digital block i and coefficients $APB^{l,j}_{m,n}$ of the power bus of analog block j, relationships between known currents at the two ends of a segment, charges stored by capacitances and unknown voltages need to be established. We refer to Figure 4(a) for this task.

Transient analysis of an RC segment results by solving the following set of differential and algebraic equations (DAEs) (equations represent Kirchhoff's and Ohm's laws and definition of capacitance):

$$\dot{q}(t) = ic(t) \qquad i_o(t) = i_c(t) + i_i(t) \qquad i_i(t)\,R = v_o(t) - v_i(t)$$
$$v_o(t) = \frac{q(t)}{C}$$

Charge value *q(t-1)* stored by capacitor C at previous time moment is known. Besides the two voltages, charge *q(t)* at the current time moment is also unknown.

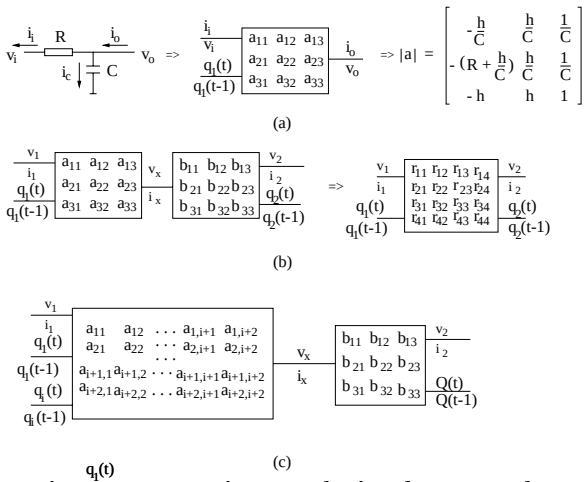

Figure 4: Transient analysis of a power bus

Previous set of DAEs cannot be directly solved with an analytical method because of the derivative of charge *q(t)*. Instead, derivate $\dot{q}(t)$ is replaced by the Backward Euler integration formula [13] $\dot{q}(t) = \frac{q(t) - q(t-1)}{h}$. Constant h is the integration step. After solving the resulting equation set, following formulas resulted for unknowns *q(t)*, $v_i(t)$ and $v_o(t)$:

$$v_o(t) = \frac{1}{C}(q(t-1) + h\,(i_o(t) - i_i(t)))$$

$$v_i(t) = \frac{1}{C}(q(t-1) + h\,i_o(t) - (RC + h)i_i(t)))$$

$$q(t) = q(t-1) + h\,(i_o(t) - i_i(t))$$

Hence, for transient analysis, the RC stage can be modeled as a three-port block, where two ports describe the physical ports of the stage (these ports are described by a pair current-voltage) and a virtual port that has the charge value $q(t-1)$ as a known and the charge value $q(t)$ as an unknown (Figure 4(a)). Figure 4(a) also depicts the symbolic matrix for the three-port representation of an RC stage.

To complete power net modeling, a composition rule is needed to replace a set of two connected blocks with a composed block. Composing blocks can be either RC stages or other previously composed blocks. Figure 4(b) shows the composition of two RC stages into a single composed block. Note that this composition generates a four-port block: two physical ports (described by current-voltage pair) and two virtual ports for the two state-variables of the composed block. Hence, each composition step (that involves an RC stage and another RC stage or a composed block) increases by one the number of ports of the composed block. As Figure 4(c) shows, after i composition steps, the composed block has $i + 2$ ports (two physical ports and i virtual ports). To characterize a new composition step, the matrix parameters of the overall block with $i+3$ ports must be related to the parameters of its two composing blocks (one with $i + 2$ ports and the second with three ports). Parameters $DPB^{l,j}$ of a digital power net composed of n RC segments (or parameters $APB^{l,j}$ of an analog power net formed of k RC segments) can be calculated by applying recursively the composition rule n (k) times.

4. INTEGRATED HLS AND POWER NET ROUTING

4.1 Combined Binding and Scheduling

4.1.1 Performance Models for Throughput

This subsection presents our method for representing and creating Performance Models (PMs) that describe throughput of all possible implementations of a CDFG. Figure 5(a) presents a CDFG, where operations 2, 3 and 5 share the same resource. Its PM for throughput is shown in Figure 5(b). PM is a directed acyclic graphs (DAGs) that defines start and end times of all operations and relates throughput to these time points. PM includes following elements:

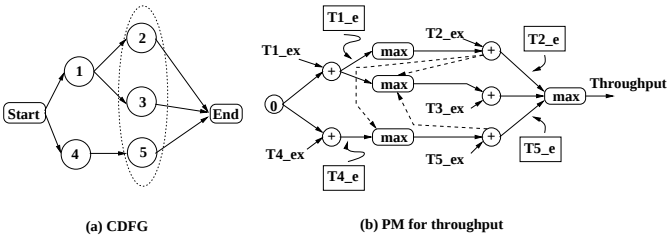

Figure 5: Throughput Performance Model for CDFG

- *Starting node* labeled as *0*: This node describes that all observed performances (throughput in our case) are set to value 0.
- *Constant part* represented as nodes and solid arcs in the figure: It reflects invariant CDFG characteristics i.e. data and control dependencies and their influence on observed performances. *Max* and *addition* nodes are used to calculate operation execution times. Max nodes denote that an operation can not start earlier than the moment when all its predecessors were executed. Outputs of max nodes describe starting times of their corresponding CDFG operations. Addition nodes express that end time Ti_e of operation op_i is the sum of its start time and its execution time Ti_ex.
- *Variable part* shown as dotted arcs in the figure: It indicates the influence of HLS decisions on modeled performances. For our case, variable PM part shows the connection between scheduling decisions for nodes 2, 3 and 5 and throughput value. For example, scheduling order 2, 5, 3 is represented in the figure between addition nodes that calculate end times for operation executions and max nodes that characterize starting times. Other scheduling orders are easily captured by PMs by accordingly changing orientation of dotted arcs.

Throughput of a given HLS solution can be exactly calculated by numerically evaluating the PM. In [6], we present detailed rules for automatically seting up PMs for a specification.

4.1.2 Performance Optimization Algorithm

Proposed HLS technique generates binding and scheduling solutions using simulated annealing (SA) algorithm [11], and analyzes their quality by evaluating the already introduced PMs for throughput and DSN. SA algorithm iteratively selects a new solution from neighborhood of the current solution point. For each current solution, neighborhood is defined as the set of solution points that differ from current point by (1) scheduling order of *one* operation pair that shares the same resource or (2) resource binding of *one* operation. Power net of each RTL solution is synthesized using the algorithm presented in the next subsection. PMs are updated for selected solution and then used for numerically evaluating the resulting DSN and throughput of the design.

```
procedure Routing Algorithm is
    inputs: layout and module placement
    inputs: binding and scheduling of CDFG operations
    outputs: routing topology and area for the power-net
(1)     for all digital blocks, in the decreasing order of their criticality do
(2)       i = 1;
(3)       while i <= Max_k do
(4)         route power bus of digital block according to its i-th shortest path;
(6)         find the minimum wire width that guarantees correct electrical
(7)           behavior at the power pins of all analog blocks;
(8)         if widths of the power segments are less than channel capacity then
(9)             calculate area occupied by the power-net;
(10)            break;
            end if
            i = i + 1;
          end while
(11)      if i > Max_k then
(12)        return routing is not possible;
          end if
        end for
(13)    return routing area of the power net;
end procedure
```

Figure 6: DSN-criticality driven global routing

Initial solution point is found by uniformly distributing operations to resources and then scheduling operations with list scheduling algorithm having critical path as priority function [5].

Search space for HLS has a hierarchical structure. Each possible operation binding permits a multitude of possible schedulings for operations that share the same hardware resource. To emulate hierarchical solution space exploration, SA algorithm performs binding and scheduling decisions with different probabilities. A small probability p is used to select a change in operation binding and a greater probability 1 - p is utilized to decide a new scheduling order. Hence, on the average, for each binding solution $\frac{1-p}{p}$ new scheduling solutions are considered. For example, if p = 0.001 then for each binding solution on the average 999 scheduling solutions are produced.

4.2 Power-Net Routing and Sizing

We employ a global routing algorithm for power net routing and sizing. Addressed power-net routing problem can be defined as finding topology and sizes (lengths, widths) of power wires such that overall power net area is minimized and following two constraints are fulfilled: (1) amplitude of DSN transmitted through substrate and power-supply lines to the power pins of analog blocks is below a certain limit and (2) power bus width is below a fixed limit (fixed by the channel width). Minimizing power net area is important if power nets are not routed on a dedicated layer [13].

Because having short routing times was essential for HLS exploration, we developed a greedy power net router based on the critical-sink routing tree method developed by Boese *et al* [3]. The reason for selection Boese's algorithm was its low time complexity of $O(k^2)$ and its ability to provide good quality solutions as compared to optimal routers [3]. Figure 6 presents our DSN criticality-driven power net routing algorithm. Digital power pins are ordered according to their *criticality* which is the amount of *potential* DSN (PDSN) that originates at the power pin. PDSN for a hardware resource is defined as its number of operation switches over time. Critical power pins are routed first so that no additional restrictions exists due to previously routed power pins. Intuition behind this strategy is that noisy power pins will impose more demanding routing requirements (i.e. wider power buses) than the quiet power pins. Routing area reduction is targeted by minimizing the wire length between net pins and wire segment widths. Routing of a new power-net pin is done on the shortest path to an already routed power-pin or to an I/O power cell. In order to meet the imposed transient behavior for DSN at analog power pins, power segments are sized to their minimum feasible widths. If infeasible routing results because of exceeding channel capacity then next shortest path between current power pin and already routed pins is picked. To avoid extremely large routing area, we restrict that the algorithm considers only the first Max_k shortest paths. If at least one power pin cannot be routed then the power-net is unroutable.

5. EXPERIMENTAL RESULTS

DSN evaluation algorithm was implemented in 1000 lines of C++ code. Method for scheduling and resource binding required about 2500 lines of code. Finally, the algorithm for power net routing and sizing consists of 1000 lines of code. All experiments were run on a SUN SPARC 5 workstation.

DSN Evaluation

Experiments for the proposed DSN evaluation method studied its accuracy and execution time. We compared obtained results with SPICE simulation as SPICE is a widely used simulator. We used a circuit with seven substrate models and ten RC stages. This circuit reflects the situation when noise originating at a digital block propagates to the power pins of seven analog blocks. Using the technique presented in Section 3, symbolic models were built for the seven noise propagation paths to the power pins of analog blocks. Then, output voltages where calculated using these symbolic models. Same circuit was described as a SPICE file and then simulated. Accuracy of the proposed method is excellent as its errors are less than 1% as compared to SPICE simulations. This error could be reduced by selecting a smaller integration step. However, this accuracy was sufficient for HLS.

We measured execution time of the proposed symbolic method for DSN evaluation and compared it against SPICE simulation time. Size of analyzed models was modified by changing the number of RC segments. For each experiment, execution time of the symbolic method was significantly smaller than SPICE simulation time. Speed-ups differ from 33 times for shorter power buses to 4 times for longer power buses. As power bus length increases, there is a decrease in speed-up. This happens because of the very simple (but more time expensive) software prototype we developed for the symbolic technique. In our experiments, global speed-up, considering all routed power nets, was about 10 times faster than SPICE simulation. This is a very important advantage of the proposed symbolic analysis method. A reduced DSN evaluation time permits the inspection of more HLS solutions, hence, better implementations.

Combined operation binding and scheduling

Quality of SA algorithm for combined binding and scheduling was studied by comparing its solutions with well-known synthesis techniques. Four well-known examples were considered: Fifth-Order Elliptic Wave Filter (EWF) [9], Discrete Cosine Transform (DCT) [9], Linear System Solver 3x3 (LSS3x3) [12], and Linear System Solver 4x4 (LSS 4x4) [12]. Multiple resource sets were contemplated for each test case. We compared our technique with following three methods: Force-Directed List Scheduling (FDLS) [10] - a very popular HLS method, Cone-based List Scheduling (CS) [12] - a HLS technique that specifically addresses a certain class of CDFGs, and SALSA HLS tool (SALSA) [9]. Comparison with SALSA is particularly interesting as Nestor *et al* [9] motivate that SALSA outperforms other approaches.

Table 1 shows the characteristics of our test examples and ob-

Example	# op.	Resource Set			T-put	Throughput			Time
		*	+	/		FDLS	CS	SALSA	(sec)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
ELF-1	34	1	1	-	27	-	-	28	2472.9
ELF-2	34	1	2	-	21	-	-	21	902.6
ELF-3	34	3	2	-	18	-	-	18	514.0
ELF-4	34	3	3	-	17	-	-	17	241.5
DCT-1	48	1	1	-	34	-	-	35	3241.5
DCT-2	48	1	2	-	34	-	-	34	564.0
DCT-3	48	2	2	-	18	-	-	19	605.0
DCT-4	48	2	3	-	18	-	-	18	490.0
DCT-5	48	3	3	-	14	-	-	14	2241.5
DCT-6	48	4	4	-	10	-	-	10	1241.5
LSS3x3-1	24	4	3	2	7	8	7	-	570.1
LSS3x3-2	24	8	6	4	4	6	4	-	948.8
LSS4x4-1	44	4	3	2	10	13	11	-	510.63
LSS4x4-2	44	4	3	4	7	10	7	-	314.3
LSS4x4-3	44	8	6	4	6	8	7	-	719.0

Table 1: Experimental results for combined operation scheduling and binding

Example	Integrated Approach			Traditional Methodology
	Max DSN 200mV	Max DSN 300mV	Max DSN 500mV	No DSN Constraints
(1)	(2)	(3)	(4)	(5)
ELF-3	20	18	18	18
ELF-4	19	18	17	17
LSS4x4-1	14	14	10	10
LSS4x4-2	11	11	9	7

Table 2: Experimental results for integrated HLS and power net routing

tained HLS results. Column 2 indicates the number of operations in each CDFG. Columns 3, 4 and 5 present the resource sets that were considered. Column 6 shows the throughput values for designs created by our SA method. Columns 7, 8 and 9 introduce throughputs obtained by FDLS, CS and SALSA. Finally, Column 10 describes CPU times for the proposed HLS algorithm. In seven cases we obtained same results as SALSA. For three cases (ELF-1, DCT-1 and DCT-3), our results were better than those of SALSA. We believe that this improvement can be explained by allowing in our approach a more global exploration for scheduling and binding. Exploration of SALSA tends to be localized around the current solution [9]. Our results for examples LSS3x3 and LSS4x4 were compared with those offered by CS and FDLS methods. CS is an extension of FDLS that explicitly targets CDFGs with operations grouped in cones (DSP examples tend to have this characteristics) [12]. The two examples are CDFGs of this kind. Our results are superior to those obtained with CS and FDLS. Note that better results were obtained without using a "customized" priority function as in the CS algorithm. Column 11 indicates that exploration times were reasonably long in all cases.

Integrated HLS and Power-net routing

Last set of experiments addressed the issue of integrated HLS and power net routing under DSN constraints. We studied the possibility of generating feasible RTL designs if various DSN constraints are fixed. We performed two sets of experiments: first, we used the integrated approach to HLS and power net routing. Next, we considered the traditional synthesis methodology that includes (1) HLS without tackling DSN and (2) a subsequent step for power net routing. We used examples ELF-3, ELF-4, LSS4x4-1 and LSS4x4-2 with same characteristics as in Figure 2. Hardware resources were placed so that noisy blocks (blocks with a high noise signature) were positioned far away from the sensitive analog circuits.

Experimental results are listed in Table 2. Columns 2, 3, and 4 indicate throughput values obtained for designs synthesized with the integrated approach. Maximum DSN constraints were set to 200 mV, 300 mV and 500 mV. Column 5 presents throughputs for traditional methodology when no DSN constraints are contemplated by HLS. As DSN constraints are addressed during exploration, integrated approach generated feasible designs in all situations. Experiments also show that integrated approach prevents a significant throughput degradation even for tight DSN requirements. Traditional methodology was able to produce correct results if DSN constraints were loose (500 mV). In this case, it was possible to fully accommodate DSN constraints by properly routing the power net. Nevertheless, for tight DSN constraints (200 mV), resulting noise was unacceptable to the analog circuits in the traditional approach. Main reason for this limitation was that binding and scheduling were trying to maximize concurrency of operation executions. High concurrency offered a small throughput but resulting DSN was high. One way of solving this problem was to place hardware resources at a bigger distance from analog blocks. However, the result was that a bigger silicon area was needed in the traditional approach.

6. CONCLUSIONS

This paper describes a CAD methodology and a tool for high-level synthesis (HLS) of digital hardware for mixed analog-digital chips. Goal of synthesis is to generate digital hardware so that its DSN is below a certain noise constraint and throughput is minimized. The paper presents an integrated approach to HLS and power net routing for effectively reducing DSN. Motivation for this research is that HLS has a high impact on DSN reduction, however, DSN evaluation is very difficult at a high level. Proposed HLS method includes a new technique for combined binding and scheduling that uses application-specific models for performances (i.e. throughput and DSN) to guide HLS. Integrated approach also employs an original symbolic technique for fast evaluation of DSN and an algorithm for power net routing and sizing based on DSN criticality. Integrated approach eliminates the possibility that produced HLS solutions are infeasible because of their high DSN. This risk always exists in current HLS methodologies, where HLS and power net routing are independent tasks. Experiments also show that our combined binding and scheduling method produces better results than traditional HLS techniques. Finally, noise evaluation using the proposed algorithm can be significantly faster than SPICE simulation.

7. REFERENCES

[1] X. Aragones *et al*, "Analysis and Solutions for Switched Noise Coupling in Mixed-Signal ICs", Kluwer, 1999.

[2] R. Bergamaschi, "Behavioral Network Graph Unifying the Domains of High-Level and Logical Synthesis", *Proc. of DAC*, 1999.

[3] K. Boese, A. Kahng, G. Robins, "High-Performance Routing Trees with Identified Critical Skins", *Proc. of DAC*, 1993.

[4] E. Charbon *et al*, "Modeling Digital Substrate Noise Injection in Mixed-Signal IC's", *IEEE Trans. on CADICS*, Vol.18, No.3, March 1999, pp.301-310.

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

[6] A. Doboli, "Specification and Design-Space Exploration for High-Level Synthesis of Analog and Mixed-Signal Systems", Ph.D. Thesis, University of Cincinnati, 2000.

[7] W. Dougherty, D. Thomas, "Unifying Behavioral Synthesis and Physical Design", *Proc. of DAC*, 2000.

[8] K. Joardar, "A Simple Approach to Modeling Cross-Talk in Integrated Circuits", *IEEE JSSC*, Vol. 29, no. 10, October 1994, pp. 1212-1219.

[9] J. Nestor, G. Krishnamoorthy, "SALSA: A New Approach to Scheduling with Timing Constraints", *IEEE Trans. on CADICS*, May 1993.

[10] P. G. Paulin, J. P. Knight, "Force Directed Scheduling for the Behavior Synthesis of ASICs", *IEEE Trans. on CADICS*, volume 8, June 1989, pp.661-679.

[11] C. Reeves, "Modern Heuristic Techniques for Combinatorial Problems", J. Wiley, 1993.

[12] S. Govindarajan *et al*, "Cone-Based Clustering Heuristic for List-Scheduling Algorithms", *Proc. of ED & TC*,1997.

[13] B. Stanisic *et al*, "Synthesis of Power Distribution to Manage Signal Integrity in Mixed-Signal ICs", Kluwer, 1996.