Behavioral Partitioning in the Synthesis of Mixed Analog-Digital Systems *

Sree Ganesan and Ranga Vemuri
Department of ECECS, ML 0030
University of Cincinnati
Cincinnati, OH 45221-0030
{sganesan,ranga}@ececs.uc.edu

ABSTRACT

Synthesis of mixed-signal designs from behavioral specifications must address analog-digital partitioning. In this paper, we investigate the issues in mixed-signal behavioral partitioning and design space exploration for signal-processing systems. We begin with the system behavior specified in an intermediate format called the Mixed Signal Flow Graph, based on the time-amplitude characterization of signals. We present techniques for analog-digital behavioral partitioning of the MSFG, and performance estimation of the technology-mapped analog and digital circuits. The partitioned solution must satisfy constraints on imposed by the target field programmable mixed-signal architecture on available configurable resources, available data converters, their resolution and speed, and IO pins. The quality of the solution is evaluated based on two metrics, namely feasibility and performance. The former is a measure of the validity of the solution with respect to the architectural constraints. The latter measures the performance of the system based on bandwidth/speed and noise.

1. INTRODUCTION

The growing trend towards integration of analog and digital designs has led to the need for mixed signal design automation tools. The VHDL-AMS Synthesis Environment,*VASE*[1] being developed at the University of Cincinnati performs synthesis of analog and mixed-signal designs for ASIC [1] and field-programmable technologies [2, 3]. In this paper, we address the problem of analog-digital partitioning for synthesis of mixed-signal systems from behavior-level specifications.

Our intermediate format for behavioral representation called Mixed Signal Flow Graph (MSFG) is based on the time-amplitude characterization of signals. Typically a suggestive partitioning can be inferred as directed by the constructs in the input specification. But this might not necessarily be the partitioning that best satisfies the system constraints. Implementing a design in the analog and digital domains has its corresponding advantages. By moving the boundary between the analog and digital domains, the design space can be searched for better solutions. Donnay et al [4] present a methodology that automates the analog/digital partitioning in low-power signal processing applications. They compute the optimal trade-off between analog filtering and digital filtering in the the baseband signal processing circuits of a direct conversion receiver.

A successful mixed-signal synthesis system must have efficient techniques to explore the design space. To aid the trade-off decisions during exploration, estimation techniques are required. On the digital side, estimation utilizes information from functional modules characterized for area, delay and switching noise. The effect of a large number of interacting parameters across different levels of hierarchy (from the macro-cell level to the transistor level) makes it impossible to pre-characterize analog functional modules. Hence analog estimation techniques either employ performance macro models or perform approximate transistor sizing. In the case of field programmable analog macro-cells such a performance characterization is possible because the components are already fabricated and device technology is known. In this paper, we present a behavioral partitioner to perform design space exploration for reconfigurable mixed-signal systems.

The rest of the paper is organized as follows. Section 2 describes the behavioral specification and representation for mixed-signal systems. Section 3 first describes the target mixed signal system architecture and the constraints that must be satisfied, and then presents our strategy for analog-digital partitioning. The techniques for performance estimation are presented in Section 4. In Section 5, an experimental study of the partitioning behavior is presented. Concluding remarks and areas of future work are outlined in Section 6.

2. SYSTEM REPRESENTATION

In this section, we present the behavior representation for mixed signal systems that are restricted to signal processing applications (i.e. control flow is absent or is outside the signal processing system being considered).

Grimm and Waldschmidt [5] employ a homogeneous representation, called KIR, to describe behavior of hybrid systems. Its primary emphasis is on interfacing different time models (continuous time, discrete time and event-driven) of specification. In ARCHGEN, Antao and Brodersen [6] use transfer functions to describe analog (linear time-invariant) system behavior. State space models of the same are generated, and then converted into block diagrams. Doboli and Vemuri [7] build a block-level analog signal flow graph representation, ABLOX from VHDL-AMS specifications.

Our intermediate representation is based on the time-amplitude characterization of mixed signals. Figure 1 shows the four possible do-

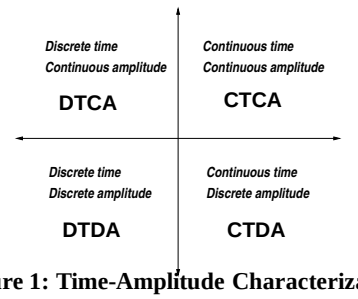

Figure 1: Time-Amplitude Characterization

*This work is sponsored by AFRL, Wright Patterson Air Force Base under contract number F33615-96-C-1911

[1]University of Cincinnati's VASE Project URL: http://www.ececs.uc.edu/~ddel/projects/vase/vase.html

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

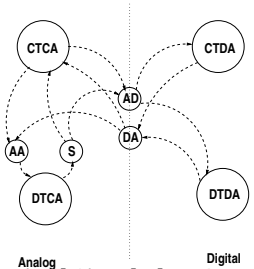

Figure 2: Mixed Signal Flow Graph (MSFG)

mains that result from this classification: (1) continuous time, continuous amplitude(CTCA) - the traditional analog domain (2) continuous time, discrete amplitude (CTDA) - the digital asynchronous and combinational systems eg. memories, (3) discrete-time, continuous amplitude (DTCA) - these are the sampled-data systems eg. switched-capacitor, and (4) discrete time, discrete amplitude (DTDA) - these are the DSP systems.

The intermediate representation called *Mixed Signal Flow Graph (MSFG)* is shown in Figure 2. It is constituted by one block for each domain, where each block is an SFG. Nodes have operation/function (eg. add, multiply, integrate, delay) and block domain (CTCA, CTDA, DTCA, DTDA) attributes. Edges are defined by source and destination nodes, and denote signal flow. The source and destination nodes must be *compatible*. And incompatibility may be resolved by inserting appropriate interface attributes on the edges crossing the domain boundaries. The interface between the domains is based on signal flow as follows: Signal flow from the analog to the digital domain requires analog-to-digital (AD) conversion, and digital-to-analog (DA) conversion from the digital to the analog domain. In addition, sampled-data systems require anti-aliasing (AA) and smoothing (S) functions. So, edges that cross the domain boundaries have an interface attribute (AD, DA, AA, S). The DTCA and DTDA blocks also have a clock attribute.

3. BEHAVIORAL PARTITIONING

3.1 Target Architecture

In this section, we discuss how the target architecture imposes constraints based on the hardware resources. We target reconfigurable mixed-signal systems composed of field-programmable analog and digital arrays (FPAAs and FPGAs). These field-programmable systems are invaluable for rapid hardware prototyping and evaluation. Typically reconfigurable mixed-signal integrated circuit designs have been based on the union of previously designed analog and digital arrays, along with some provision for the exchange of signals between the two domains [8, 9].

We consider a mixed signal system composed of a set of field programmable devices that implement designs of the four block domains (CTCA, CTDA, DTCA, DTDA). The analog subsystem (continuous amplitude) consists of continuous-time and discrete-time FPAAs such as Zetex TRAC [10] and Motorola MPAA [11] respectively. These FPAAs correspond to the CTCA and DTCA blocks respectively. The digital subsystem (discrete amplitude) consists of Xilinx FPGAs [12] that implement the CTDA and DTDA blocks. Data-conversion is performed by a bank of 8-bit A-to-D and D-to-A converters. It is clear that this prototype is constrained by: (1) configurable resources on the FPAAs and FPGAs, (2) number of available data-converters, their resolution and speed, and (3) available I/O pin resources on the devices.

For a single-chip implementation with both the analog and digital portions on the same silicon, the constraints are modified as follows: (1) The combined areas of the analog, digital and data-converter interface should be less than the chip's die-area, (2) Total number of pins must be minimized, (3) In addition, when the analog and digital parts are implemented on a common substrate and share power pins, the switching noise coupled via the substrate and power/ground lines also need to be considered in addition to the fundamental noise mechanisms such as white noise and flicker noise.

Although we target reconfigurable analog and digital systems, the algorithms we propose for behavioral partitioning may be applied to custom/semi-custom mixed-signal systems. The main difference is in the estimation techniques employed for field-programmable and custom designs (especially for analog performance estimation).

3.2 Constraints

For a given reconfigurable mixed-signal hardware, the configurable analog and digital resources are fixed. The SFG and DFG must be mapped onto these fixed resources, also ensuring that the mapped netlist may be successfully placed and routed to produce a working design. Thus we have the following constraints:

- $A_{digital} < p\% * A_{fpga}$, where $A_{digital}$ is the estimated area of the digital part and $p\% * A_{fpga}$ is the logic area available on the FPGA after allowing $(100-p)\%$ as the estimated area for routing.
- $A_{analog} < q\% * A_{fpaa}$, where A_{analog} is the estimated analog area and $q\% * A_{fpaa}$ is the area available on the FPAA, with $(100-q)\%$ being the estimated routing area.

Depending on the interface attribute, edges are mapped onto the data-converters in the system. The data-converters play the most important role in determining signal quality. The signal-to-noise ratio (SNR) of an analog signal is the ratio of the largest signal to the noise when no signal is present. In the digital realm, the SNR is the ratio of the largest representable number to the quantization error. Increasing the number of bits increases the SNR and reduces quantization noise.

To maintain the SNR established by the AD, the noise floor of digital signals should not be larger than the AD conversion noise floor. Similarly it must be ensured that the DA converters are not affected by the noise introduced due to truncation/roundoff errors. Therefore a well-designed system must employ a precision greater than the input and output sample wordsize so that the additional bits allow the system to maintain the data-converters' SNR. As for the analog part, the signals sampled by the AD must have a noise floor lesser than the AD converter's noise floor so that the code is not erroneous. For the target architecture with 8-bit converters, we choose 16-bit resolution for the digital portion. Figure 3 illustrates the relation between computation errors and converter noise floors.

Just as the converter resolution affects the noise performance, its speed determines the analog signal bandwidth. In order to ensure that the design will operate correctly, the digital read frequency and the AD conversion time (or latency) are used to determine the AD sampling rate. Similarly, the digital write frequency and the DA setting time are used determine the DA sampling rate. Following are the interface constraints:

- The AD sampling frequency, f^{ad}_{sample} cannot be smaller than the digital read frequency, f_{read}, i.e. $f^{ad}_{sample} > f_{read}$. The AD sampling period must be at least equal to the AD conversion time, $T^{ad}_{convert} < 1/f^{ad}_{sample}$.
- The DA sampling frequency, f^{da}_{sample} cannot be smaller than the output write frequency, f_{write} of the digital part, and hence $f^{da}_{sample} > f_{write}$. The DA sampling period must be at least equal to the DA settling time, $T^{da}_{settle} < 1/f^{da}_{sample}$.
- The resolution of the AD and DA converters imposes a limit on the dynamic range of the signals. We constrain the analog signal's noise floor, $noise\_floor$ based on the number of bits n on the data-converter interface: $noise\_floor < Vdd/2^n$.
- Interface edges from the analog to the digital partition, num_{ad} and vice versa num_{da} must be respectively lesser than the number of available AD (N_{ad}) and DA (N_{da}) converters. So, $num_{ad} < N_{ad}$ and $num_{da} < N_{da}$.

Finally, the partitioned design must meet the pin constraints:

- The cutset or number of connections between the partitions (num_{io}) must be lesser than the number of pins (N_{io}) available on the device(s) i.e. $num_{io} < N_{io}$.

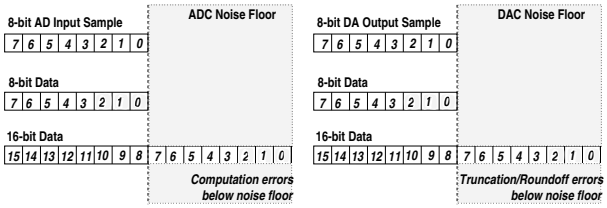

Figure 3: Compute Error and Converter Noise Floor

```
procedure beh_partition()
    current ← initial solution;
    evaluate_fitness(current);
    temp ← initial temperature;
    while temp > final temperature
        for iter = 1 to Maxiter
            neighbor ← perturb(current);
            map_onto_target_arch(neighbor);
            evaluate_fitness(neighbor);
            if accept(neighbor,current,temp)
                current ← neighbor;
            end if
        end for
        temp ← α * temp;
    end while
end
```

Figure 4: Simulated Annealing Algorithm

3.3 Partitioning Strategy

The behavior of the mixed-signal system is represented as a mixed-signal flow graph (MSFG), where nodes represent the operations on the signals and edges represent the signal flow, in each of the four domains. The objective of the problem is to assign operator nodes of the behavioral MSFG to the analog and digital portions such that the architectural constraints are satisfied and the area, bandwidth/speed and noise performance are optimized. Nodes cannot be arbitrarily assigned to any domain since the function/operation performed by the node must be available among the library of operations in that domain. Following are the allowed moves for nodes:

We employ a simulated annealing formulation to perform behavioral partitioning. Figure 4 illustrates the algorithm. The specified system behavior is represented in the behavior MSFG format, and randomly partitioned into analog and digital. For edges crossing the interface, the attributes are set so as to resolve incompatibility. The algorithm begins the search with this *initial solution*. After the operator nodes are partitioned and interface attributes updated, the system-level behavior is reduced to behavior SFG and DFG in the analog and digital domains. The partitioned design is mapped onto the target mixed-signal hardware by procedure *map_onto_target_arch()*.

The algorithm explores the design space by incrementally moving from the current solution, *current* to a new solution, *neighbor* in the same neighborhood. This is done by picking an operator node from one of the domains and moving it to the other. The move cannot be random, but may be made only if the node's function can be implemented using the functions in the target domain's library. Procedure *perturb()* chooses a random node, and determines if the node's function is implementable in the target domain. If so the node is moved, otherwise another node is chosen for the move.

Solution quality is evaluated by the procedure *evaluate_fitness()* (described in section 4). In procedure *accept()*, the new solution is accepted if its fitness value is higher than the current. Otherwise, it is accepted with a probability depending on the current *temperature*. This results in the "hill-climbing" performance of the simulated annealing algorithm employed, thereby enabling it to jump out of local minima during the search. Initially the temperature is set at a high value, and is then gradually decreased to the based on a *cooling schedule*. The algorithm terminates upon reaching the final temperature.

4. PERFORMANCE ESTIMATION

After the operator nodes have been partitioned, the appropriate interface nodes must be inserted. The partitioned design must then be mapped onto the target FPAA and FPGA technology. While mapping the analog SFG and digital DFG, various possible structural FPAA and FPGA implementations are contemplated and the mappings with best performance respectively chosen. In this section, we present our approach for performance estimation of the analog and digital circuits, as well as the fitness evaluation of the partitioned mixed-signal implementation.

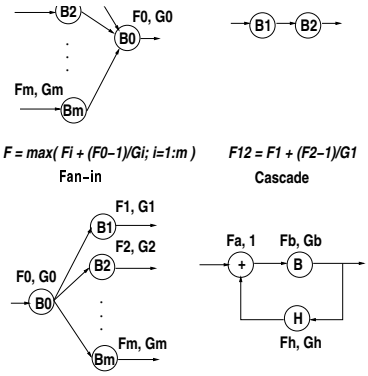

Figure 5: Noise Figure Estimation

4.1 Analog Performance

In the design of analog systems, the performance measures include silicon area, power dissipation, bandwidth and dynamic range. When analog systems are interfaced with digital systems, bandwidth is limited by the data-conversion speeds. Also analog designs tend to have lower power dissipation compared to digital. Noise and nonlinearity are two factors that affect dynamic range. And silicon area is always a premium. In order to evaluate the analog partition, area and noise estimates are required.

The technology-independent analog SFG is transformed into a netlist of FPAA library components. The FPAA macro-library is a set of analog building blocks, such as adder/subtracters, amplifiers, integrators, filters and so on. Each building block is characterized by its behavior, as well as functional and performance parameters. In [13], we have presented our techniques for analog library binding. Mapping the analog partition to the library begins with architecture generation, where the behavior SFG is transformed into various possible architecture SFGs. The next step is to obtain the optimal FPAA netlists by binding the generated architecture SFGs to the target library, and then determining the binding that optimizes the circuit performance. We perform gain redistribution and explore various possible circuit architectures for the best area and noise performance.

4.1.1 Area Estimation

FPAAs comprise of configurable analog blocks (CABs) and programmable interconnect, that can be configured to implement analog signal processing circuits. Hence the number of CABs provide an estimate of the area occupied by the component. If we target an ASIC, then the library component provides the circuit topology, but the netlist is *unsized* at this stage of synthesis. With opamp-based topologies where the largest percentage of the component area is contributed by the opamps, the number of opamps can be used to provide an estimate of the component area.

Each library component's area is estimated a priori. The area of the analog partition is given by the sum of the areas of the nodes of the behavioral SFG. The nonlinear nodes are directly mapped to the library, and their area is determined. For transfer function nodes, the architecture SFG corresponding to the minimal realization is bound to the target library by performing hierarchical pattern matching and covering. The area at each architecture SFG node is the sum of the areas of the bound components.

4.1.2 Noise Figure Estimation

Noise figure, F is a figure-of-merit for a device or circuit with respect to noise [14]. It is defined as the ratio of the signal-to-noise (power) ratios between the input and output.

In order to estimate the noise performance of the bound behavioral SFG, we employ the composite noise figure to compare the degradation in SNR at the outputs of the system. As detailed in [13], we identify four types of component configurations, namely *cascade* (series connection of blocks), *fan-out* (output of one block feeding inputs of several others), *fan-in* (outputs of multiple blocks feeding inputs of one block) and *feedback* (output of one block fed back into the signal path, thereby creating a cycle in the signal flow). Figure 5 shows the four configurations and their noise figure estimates. The composite noise figure is obtained by repeatedly applying the expressions for the noise figure of four component configurations.

4.1.3 Bandwidth Estimation

The analog signal bandwidth is obtained based on the Nyquist criterion. It is the determined by the sample rates of the AD and DA. $BW_{analog} < 1 \,/\, 2 \cdot min(\, f^{ad}_{sample}, f^{da}_{sample})$.

4.2 Digital Performance

The performance of the digital design depends on the implementation choices. Power dissipation affects thermal characteristics of the circuit, thereby affecting the circuit's performance. Silicon area adds to manufacturing cost. Throughput determines data-conversion speeds and hence signal bandwidth. Simultaneously switching nodes can couple noise to the power lines and substrate due to parasitic impedance. Hence the performance parameters considered are the design throughput, switching noise, clock power and area.

Given the behavioral DFG and the RTL library of resources characterized for area and execution delays, the estimator performs scheduling to determine the start times for the execution of the nodes and their binding to the resources. Resource allocation, scheduling and binding may be performed in various ways depending on the desired circuit implementation style. Since we target signal processing systems in this work, we employ a macro-cell based design style (typical of DSP circuits). Module generators synthesize the required resources from primitive functional and storage resources, and their area and delay is estimated.

The amount of hardware resources allocated during synthesis affects the overall area of the design. Constraints on resource usage therefore arise if the available silicon area for the circuit is limited. We assume that silicon area is unconstrained, and nodes of the behavioral DFG are bound to dedicated resources.

4.2.1 Area-Delay Estimation of Resources

Nonlinear nodes are directly mapped to RTL library components and their area determined from the area-delay characterization table. Module generators translate the z-domain transfer functions into digital filters as illustrated in Figure 6. Its area is estimated using the area estimates of registers, adders and multipliers from the RTL library. For the Xilinx XC4000 technology, where each CLB has two 4-input lookup-tables (LUT) and two flip-flops (FF). The ZTF node's area estimate $area_{ztf}$ is given by the maximum of the CLBs required for the LUT-based logic, CLB_{LUT} and flip-flops, CLB_{FF}.

$$area_{ztf} = max(CLB_{LUT}, CLB_{FF})$$

The delay of the ZTF node, $delay_{ztf}$ is the sum of the delays of the primitive resources in the critical path. Here, we assume that the FPGA local interconnections are utilized and therefore delay due to wiring is negligible.

$$delay_{ztf} = t_{d\_mul} + t_{d\_add} * max(\lceil log(num\_add) \rceil, \lceil log(den\_add) \rceil)$$

where t_{d_mul} and t_{d_add} are the delays of the multiplier and adder respectively. The number of levels in the adder tree of the numerator is $\lceil log(num\_add) \rceil$ and that of the denominator is $\lceil log(den\_add) \rceil$.

4.2.2 Total Area Estimation

In the case of DSP-based systems, the circuits are typically resource-dominated, and the area and delay of the resources dominate those of the steering and control logic. Hence area of the bound resources gives a fair estimate of the overall area.

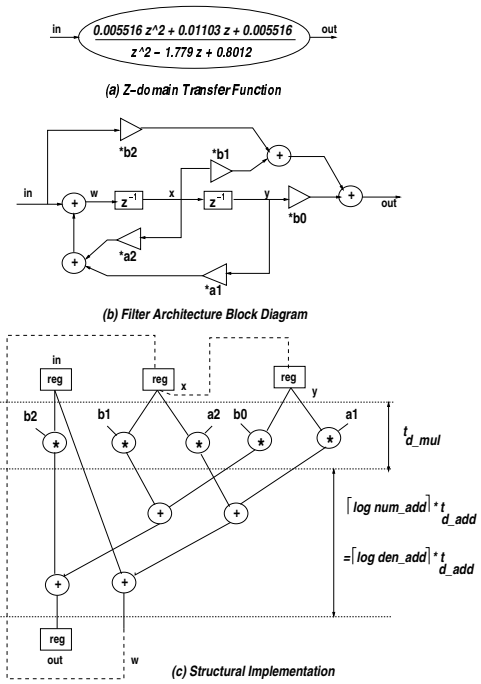

Figure 6: Z-domain Transfer Function Implementation

4.2.3 Throughput Estimation

The throughput of the design is the rate at which the inputs are consumed and outputs produced. The estimated throughput is given by

$$T_{est} = \mathcal{L} * \{max(exec\_delay(v) + route\_delay(v,w)) + reg\_setup + reg\_hold\}$$

where $\mathcal{L}$ is the schedule latency, reg_setup, reg_hold are the setup and hold times of registers, $exec\_delay(v)$ is the execution delay of node v and $route\_delay(v,w)$ is the interconnect delay between node v and its successor node w.

In order to compute the routing delay, fast placement of the resources in the scheduled DFG is performed. After the positions of the resources have been determined, we lookup delays from an FPGA wiring delay characterization table. This table contains the average interconnect delay associated with the target FPGA architecture, obtained experimentally by directly interconnecting CLBs at various distances and measuring the pin-pin delay. (Note: here we assume the FPGA accomodates the entire digital part, and hence no off-chip wiring delay estimate is required).

4.2.4 Clock Power Estimation

The total power dissipation in digital circuits is the sum of the static, dynamic and short-circuit power dissipation. Typically the dynamic power dissipated is used to estimate the total power. It is proportional to the total capacitance of the registers and the interconnect capacitance switched by the clock. Since all the registers in the circuit are clocked, the number of registers in the design is used as the metric for estimating the clock power.

4.2.5 Switching Noise Estimation

Switching noise affects both the analog and the digital circuits. When inputs and outputs of several gates switch, a large cumulative current spike flows through parasitic resistances and inductances, creating power supply spikes known as Vdd bounce or Gnd bounce. Some fraction of this noise is inevitably injected into the substrate. For the analog part, switching noise injected into the substrate tends to degrade circuit performance. For the digital part, it may result in incorrect evaluation and can cause functional failures.

Switching noise is characterized by the simultaneous switching transitions in the digital circuit. Nagata et al [15] present a switching

noise macro-model based on this. In our work, simultaneous switching activity is attributed to operation concurrency. The maximum number of concurrent operations in any control step is used as the metric to estimate the switching noise. We assume that each operation contributes one unit of switching noise (pessimistic) and that they act additively (optimistic).

4.2.6 Estimation Technique

The estimator performs scheduling for the DFG to obtain the desired performance estimates. Since we assume that silicon area is unconstrained, no constraints on resource usage are imposed. The nodes of the behavioral DFG are bound to dedicated resources. Resources sharing is not performed, but registers are shared.

Scheduling assigns time-steps for each operation, and thereby determines the lifetimes of the carriers. Thus scheduling affects register allocation and sharing. Hence minimizing the number of register bits in the design can reduce the clock power. Scheduling operators concurrently tends to reduce the latency, thereby improving throughput. But simultaneous switching activity increases with the operation concurrency. Hence switching noise may be reduced by minimizing concurrent operations. Therefore we contemplate various schedules and determine the best one encountered based on the following cost function:

$$sched\_cost = w_1 \cdot \frac{\mathcal{L}_{min}}{\mathcal{L}_{est}} + w_2 \cdot \frac{SN_{est}}{SN_{max}} + w_3 \cdot \frac{PD_{est}}{PD_{max}}$$

$\mathcal{L}_{est}$ is the latency of the schedule while $\mathcal{L}_{min}$ is the latency of the shortest possible schedule. SN_{est} is the estimated switching noise i.e. maximum number of concurrent operations in a control step. SN_{max} is the maximum switching noise, that occurs when all operations are concurrent. PD_{est} is the estimated clock power given by the number of registers required. PD_{max} corresponds to the clock power for maximum number of registers, and is determined by assuming no registers are shared and every variable requires a register.

4.3 Fitness Evaluation

In order to determine the quality of the design, we employ two metrics: *feasibility* and *performance*. The former determines the validity of the solution with respect to the architectural constraints. The latter measures the performance of the design such as noise and bandwidth/speed.

The feasibility cost, $\mathcal{F}$ indicates whether the design is mappable onto the target hardware. $\mathcal{F}$ is assigned 0 or 1 value depending on whether the constraints were violated or satisfied.

if $(A_a < p\% \cdot A_{fpaa})$ **and** $(A_d < q\% \cdot A_{fpga})$ **and**
$(f_{read} < 1/T^{ad}_{convert})$ **and** $(f_{write} < 1/T^{da}_{settle})$ **and**
$(num_{ad} < N_{ad})$ **and** $(num_{da} < N_{da})$ **and**
$(num_{io} < N_{io})$ **and** $(noise\_floor < Vdd/2^n)$
then $\mathcal{F} \leftarrow 1$;
else $\mathcal{F} \leftarrow 0$;

The performance cost, $\mathcal{P}$ is given by

$$\mathcal{P} = w_n \cdot \frac{NF_{desired}(f_i) - NF_{est}(f_i)}{NF_{desired}(f_i)} + w_b \cdot \frac{BW_{est} - BW_{desired}}{BW_{desired}} + w_p \cdot \frac{PD_{desired} - PD_{est}}{PD_{desired}}$$

where w_n, w_b, w_p are the associated non-negative weights in the range [0:1], and $w_n + w_b + w_p = 1.0$. The first term favors better noise performance. The next term attempts to minimize the deviation between the estimated and desired bandwidth measures. Designs with power dissipation worse than desired are penalized by the last term.

5. EXPERIMENTS

In this section, we study the partitioning behavior experimentally. It is evident that architectural constraints may prefer a particular partitioning compared to pure analog or digital implementations. Here, we investigate whether partitioned solutions are preferred in the absence of architectural constraints. Hence, partitions are evaluated based on the performance cost function only.

$$Cost = w_n \cdot NF_{cost} + w_b \cdot BW_{cost} + w_p \cdot PD_{cost}$$

where $NF_{cost}, BW_{cost}, PD_{cost}$ are set to 0 if the desired noise figure, bandwidth, power constraints are satisfied, else -1. Therefore all solutions that have a zero *cost* are *constraint-satisfying*.

Experiment 1

In the first experiment, the partitions are evaluated using only two performance metrics, namely the noise figure and bandwidth. In this study, we observe the effect of various factors on the number of fully analog, fully digital, and mixed analog-digital solutions. The following experiments were performed: (1) For a given NF and BW constraint, the weight w_n was varied from 0.0 to 1.0. (2) The same was repeated for a given NF but now varying BW constraint also. (3) The experiment was performed again with a given BW and varying NF constraints.

Figures 7 and 8 show the effect of $BW_{desired}$ and $NF_{desired}$ respectively, for w_n ranging from 0.0 to 1.0. Along the y-axis is plotted the number of mixed analog-digital solutions produced. Figures 10 and 11 show the number of fully analog solutions versus w_n, for varying $BW_{desired}$ and $NF_{desired}$ respectively. Finally, Figures 13 and 14 show the number of fully digital solutions versus w_n, for varying $BW_{desired}$ and $NF_{desired}$ respectively.

Based on these plots, we make the following observations. From Figure 7, we observe that as w_n increases, the number of mixed analog-digital solutions tends to decrease. For higher BW constraints, mixed solutions begin to disappear for lower w_n values itself. This is because when part of the design is digital, this tends to degrade the bandwidth.

In Figure 8, we observe that as w_n increases, the number of mixed analog-digital solutions tends to decrease. For higher $NF_{desired}$ values more mixed solutions are seen owing to the more relaxed NF constraint, and hence the noise performance of the analog part is tolerable.

For high w_n values, the fully analog solutions are absent indicating that they failed to satisfy NF constraints. As the desired NF is increased, w_n above which the fully analog solution is feasible increases (as seen in Figure 11) because the NF constraint is now relaxed. As the desired BW increases, w_n below which the fully analog solution is feasible decreases(Figure 10).

Correspondingly, for low w_n values, the fully digital solutions are absent indicating that they failed to satisfy constraints. This is justified by the fact that the bandwidth is limited in the digital solution, therefore making it a poor choice when higher weight is associated with the BW cost. In Figure 13, as the desired BW is increased, w_n above which the fully digital solution is feasible decreases.

As the NF and BW constraints become tighter, the number of mixed signal solutions reduces. Beyond a certain BW and below a certain NF constraint, there are no longer any mixed solutions. The only solutions are either fully analog and fully digital.

Experiment 2

In the second experiment, the partitions are evaluated based on three performance metrics, namely the noise figure, bandwidth and digital clock power. Since all registers in the design are clocked, and the other components are mostly combinational, the number of registers in the design was used as the metric for the clock power dissipation. We observe the effect of various factors on the number of fully analog, fully digital, and mixed analog-digital solutions. In this study, for a given NF, BW and PD constraints (3dB, 2MHz and 200 register-bits/flip-flops respectively), the weights w_n, w_b and w_p were varied from 0.0 to 1.0.

The number of mixed-signal solutions versus variation in w_n, w_b and w_p is shown in Figure 9. We observe that as w_n increases, the number of mixed solutions tends to decrease. For high w_n and low w_b, w_p we see that no mixed solutions are present. Since some part of the mixed solution is analog and it contributes to the noise figure, these solutions are not preferred with increasing emphasis on the noise cost. The number of mixed solutions increases with increasing w_b and decreases with increasing w_p. This is because mixed solutions with large number of digital components contribute

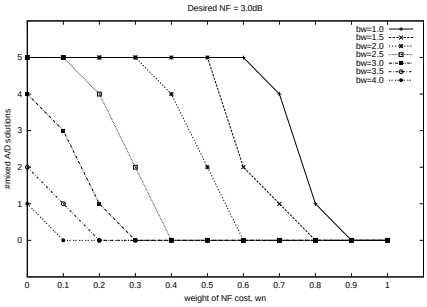

Figure 7: $w_n, BW_{desired}$ **vs #Mixed**

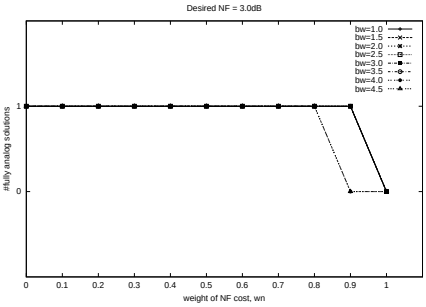

Figure 10: $w_n, BW_{desired}$ **vs Analog**

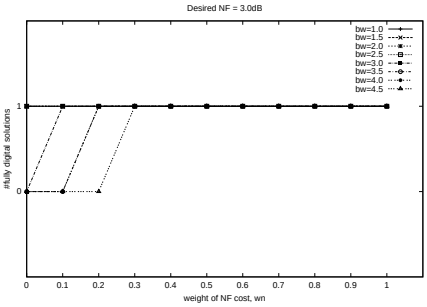

Figure 13: $w_n, BW_{desired}$ **vs Digital**

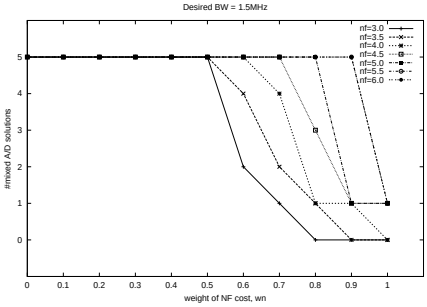

Figure 8: $w_n, NF_{desired}$ **vs #Mixed**

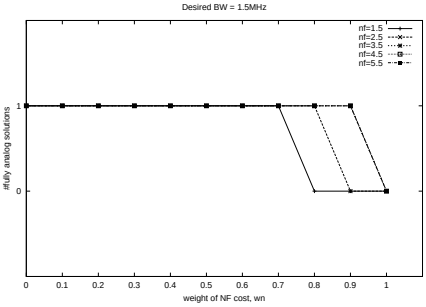

Figure 11: $w_n, NF_{desired}$ **vs Analog**

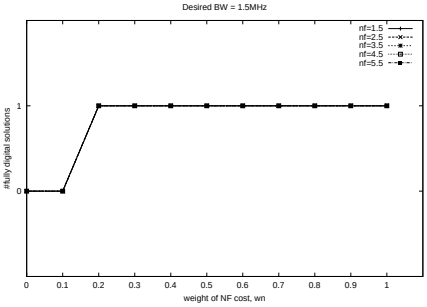

Figure 14: $w_n, NF_{desired}$ **vs Digital**

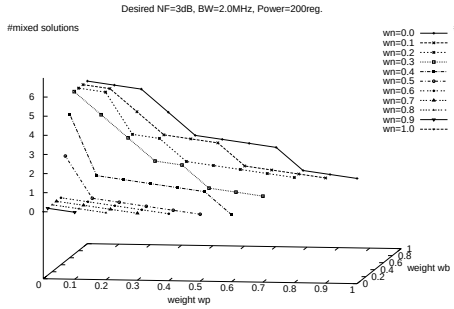

Figure 9: w_n, w_b, w_p **vs #Mixed**

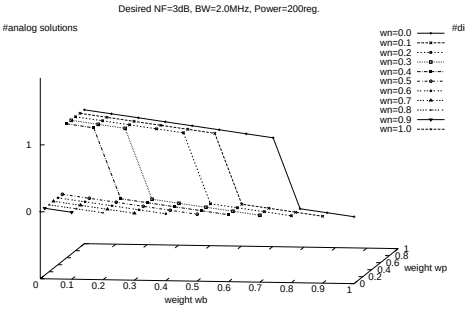

Figure 12: w_n, w_b, w_p **vs Analog**

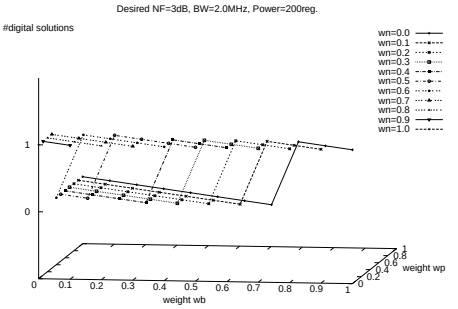

Figure 15: w_n, w_b, w_p **vs Digital**

to more power, and hence they are not preferred with increasing emphasis on power cost.

Figures 12 and 15 show the plots of the number of fully analog and fully digital solutions respectively, for various weight values. As expected, for low w_n, low w_b and high w_p the fully analog solution is alone feasible while the fully digital solution is absent. And vice versa in other regions. In the mid-range values of w_n, w_b and w_p, both fully analog and fully digital solutions are absent.

Thus in the low w_n, low w_b and high w_p region, only the fully analog solution is feasible. In the high w_n, low w_b, low w_p region, only the fully digital solution is feasible. And in the mid-range w_n, w_b and w_p region, where neither the fully analog nor the fully digital solution is feasible, we observe only mixed solutions are feasible.

6. CONCLUSIONS

In this paper, we have presented techniques for behavioral partitioning of mixed-signal systems. The system's behavior represented as a network of functional blocks using a mixed-signal flow graph (MSFG). Sub-graphs of the MSFG belong to one of four domains determined based on the time-amplitude characterization of signals. Next analog-digital partitioning is performed followed by mapping of the analog and digital onto reconfigurable hardware. We have described the constraints imposed by the hardware, and the methods for performance estimation and evaluation of solution quality. Finally we presented an experimental study of partitioning in the absence of architectural constraints. Future work will focus on mixed-signal ASIC issues and high-level synthesis techniques to address them.

7. REFERENCES

[1] A. Doboli *et al.*, "Behavioral Synthesis of Analog Systems using Two Layered Design Space Exploration," in *IEEE/ACM DAC*, June 1999.

[2] S. Ganesan and R. Vemuri, "Technology Mapping and Retargeting for Field Programmable Analog Arrays," in *IEEE DATE*, pp. 58–64, 2000.

[3] S. Ganesan and R. Vemuri, "An FPGA/FPAA-Based Rapid Prototyping Environment for Mixed Signal Systems," in *Reconfigurable Technology: FPGAs for Computing and Applications, Proc. of SPIE*, vol. 3844, pp. 49–60, 1999.

[4] S. Donnay *et al.*, "High-level Analog/Digital Partitioning in Low-power Signal Processing Applications," in *PATMOS*, (Louvain-la-Neuve, Belgium), Sep. 1997.

[5] C. Grimm and K. Waldschmidt, "A Graph based Model for Description of Mixed Analog-Digital Systems," in *EURODAC*, (Geneva, Switzerland), 1996.

[6] B. Antao and A. Brodersen, "ARCHGEN: Automated Synthesis of Analog Systems," *IEEE Trans. VLSI Systems*, vol. 3, pp. 231–244, June 1995.

[7] A. Doboli and R. Vemuri, "aBlox - A Hierarchical Representation for Behavioral Synthesis of Analog Systems from VHDL-AMS," Tech. Rep. TR-DDEL, University of Cincinnati, April 1998.

[8] J. Faura *et al.*, "FIPSOC: A Field Programmable System On a Chip," in *DCIS*, 1997.

[9] P. Chow *et al.*, "A Field-Programmable Mixed Analog-Digital Array," in *ACM/SIGDA FPGA*, pp. 104–109, (Monterey, CA), February 1995.

[10] Inc. Zetex, *TRAC User Guide*, July 1997.

[11] Inc. Motorola, *Easy Analog Design Software User's Manual*, October 1997. (Now Anadigm Inc., since 2000).

[12] Inc. Xilinx, *Xilinx Data Book*, May 1999.

[13] S. Ganesan and R. Vemuri, "Library Binding for High-level Synthesis of Analog Systems," in *Intl. Conf. VLSI Design*, January 2001.

[14] C. D. Motchenbacher and J. Connelly, *Low-Noise Electronic System Design*, J. Wiley and Sons, New York, 1993.

[15] M. Nagata and A. Iwata, "Substrate Noise Simulation Techniques for Analog-Digital Mixed LSI Design," in *IEICE Trans. Fundamentals*, vol. E82-A, pp. 271–277, February 1999.