A Basic Mathematical Formalism for Representation and Analysis of RF/Microwave EDA and Design Flows

Michael Heimlich and Carl Svensson
Macquarie University
Sydney, Australia
michael.heimlich@mq.edu.au

Abstract — **A mathematical formalism is developed for RF/microwave design flows using category theory as a means for creating a formal foundation for their analytical and empirical, rather than anecdotal, study. A simplified model of design flow is developed using categories to capture the essential structure and dynamics of the schematic, layout, and behavior of both in terms of electrical phenomena. Specifically, schematics and layouts are defined as separate monoidal cateogries and the behaviors is defined as sets of name-value pairs. Functions, or morphisms, within and between these categories are explored for interconnect design and analysis in particular, and show typical EDA features such as forward- and back-annotation, portioning, simulation, and analysis. Traditional microwave and analog design are separately investigated using this formalism and found to be in agreement with expected outcomes. A new design flow and supporting EDA tool are identified based on the mathematical implications of the category theory models and associated design flow constraints of top-down design and design closure.**

Index Terms — **EDA, design automation, circuit design, design flow, category theory**

I. Introduction

Electronic Design Automation (EDA) has been one of the great enabling technologies for modern electronics, including the class of analog circuits classified by their operating frequencies: RF/wireless, microwave, millimeter-wave, etc. Initially distinct and discrete software tools were developed for (logical) circuit simulation and (physical) layout, and these were later augmented by physical verification (DRC & LVS), system simulation, and electromagnetic analysis (EM). Later still, all of these tools came together under unifying environments providing a common database and standardized graphical (schematic) entry.

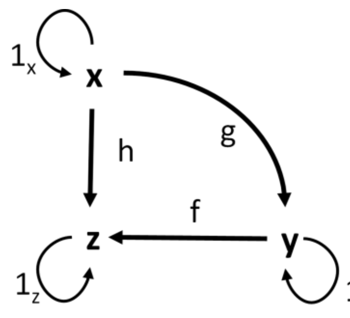
Design flows, then, have come to be supported, enabled, defined, extended, or constrained by the manner in which these individual tools are linked together and utilized by the engineering team. In some cases, the flow utilized reflects the talents, or lack of talent, of the engineers using it. Nevertheless, flows define a repeatable process, a cornerstone of all engineering disciplines. This begs the question: How can the quality of a design flow be determined?

A design flow, at its most abstract level, must transform a nearly infinite number of design possibilities into one acceptable design solution. The speed with which a flow actualizes this and the degree to which the optimal solution is approached would seem to determine a flow's quality. A survey, however, of recent design flow papers [1-3] shows that only anecdotal evidence is given for one design, one product, or one company. An objective, or formal, basis is needed. And while much work has been done already in other EDA domains outside of RF/microwave in the area of Formal Methods in CAD, or FMCAD, there is almost no contribution for high frequency analog.

Here for the first time, a mathematical foundation is developed for the representation and investigation of RF/microwave design flows at a rudimentary level using category theory, an abstract branch of mathematics. Beginning with a derivation of design flow, a basic algebra for schematic capture, layout, and their relationship to simulated behavior is developed. A brief overview of category theory and a descriptive definition of design flow are first presented. Using two requirements for the design flow—design closure and parametric design—several flows for RF/microwave design are investigated, including existing flows as a validation of the technique, by defining categories for fundamental EDA structures of simulation behavior, schematic, layout and the processes of going among them and exploring the relationships among and within them. . By constraining from within the layout a mapping between interconnects and schematic distributed models, requirements for a heretofore unimplemented EDA/CAD tool are derived from this purely mathematical representation of design.

II. Category Theory

Category theory is an abstract branch of mathematics which



$$\begin{tikzcd}
\mathbf{x} \arrow[loop left, "1_x"] \arrow[d, "h"] \arrow[rd, bend left, "g"] & \\
\mathbf{z} \arrow[loop left, "1_z"] & \mathbf{y} \arrow[l, "f"'] \arrow[loop right, "1_y"]
\end{tikzcd}$$

Figure 1 – Example of a category with objects (bold) and arrows.

$$\begin{tikzcd}
& F \arrow[d, "f"] \arrow[ddl, bend right, "f_1"'] \arrow[ddr, bend left, "f_2"] & \\
& H \times P \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] & \\
H & & P
\end{tikzcd}$$

(a)

$$\begin{tikzcd}
& D & \\
& G + P \arrow[u, "d"] & \\
G \arrow[ur, "j_1"] \arrow[uur, bend left, "d_1"] & & P \arrow[ul, "j_2"'] \arrow[uul, bend right, "d_2"']
\end{tikzcd}$$

(b)

Figure 2 – Product and co-product UMPs. The 'd' and 'f' arrows represent the UMP for the product and co-product, respectively

generalizes numbers, or objects, and the functions, or morphisms, that operate on them, with the goal of studying the structures and relationships of the morphisms in a given system, or category. While category theory has received little attention directly in EDA, it has proven to be a powerful tool in related areas, such as computer science [5]. Category theory has been applied to studying simulation of asynchronous circuits [6] but this avenue of investigation has not been applied to EDA nor has it been broadened into circuit design in general.

By definition, a category has four essential features [7-9] as shown in Fig. 1. First, objects are the domain and co-domain upon which morphisms operate. They could be real numbers or elements in a schematic. Second, arrows, from object(s) to object(s) define morphisms. Real functions or a design decision to change a resistor value or add a filter would be examples following respectively from the example objects. Third, an identity morphism, **1**, must be defined for each object such as dividing by 1 or combining 1 schematic with an open circuit "schematic" , both yielding the original object, respectively. Finally, associativity of morphisms must be shown so that if 3 numbers are added, you can sum the second and third number and then add the first, or if you are creating a three-element schematic, it doesn't matter what order the three elements are added (although the connections are important, the order in which the connections are made is not).

Universal Mapping Properties, or UMPs [8], form common structures among arrows in categories and define key relationships of interest for further study or insight [8-9]. The product of two objects, A x B (Fig. 2), requires two projection morphisms to the respective elements comprising the product and abstracts different concepts in individual fields of mathematics, like Cartesian closed coordinates in set theory. The UMP of the product states that for any given object with morphisms to the two elements in the product, there exists a unique morphisms from the given object to the product. The co-product, A + B, is the dual of the product and requires to injection morphisms from the respective elements. The co-product (Fig. 2) is the category theory equivalent of the disjoint union in set theory. A similar UMP exists for co-products. An initial object has a UMP that there exists an arrow from it to all objects in the category and a terminal object has an arrow from all objects. Many other UMPs can exist if a category supports them. This will be revisited in section III.

Two additional concepts in Category Theory will be used here. First, the concept of commuting diagrams establishes that applying one sequence of morphisms arrives at the same object as a separate sequence of morphisms. For example, in Fig. 1, if y=g(x), z=f(y), and z=h(x) for all real numbers, then h(g(x))=f(x) is said to commute. In category theory, this is written as h o g = f. Second, isomorphisms, meaning "same shapes", represent a weakening of the concept of equality. Instead of requiring two things to have the same value, isomorphic relationships require that a correspondence be established so that each element in one category has a corresponding element in the other [9]. For example, if every component on a PCB has a part number then an isomorphism exists between components and part numbers. If a pair of morphisms is isomorphic, then by applying one and then the other the original object is arrived at. In other words, composing the two morphisms gives the identify for that category, or the isomorphic arrows commute to give the identity arrow.

Finally, functors are morphisms that go from one category to another. Category theory is focused on identifying structure in one category and exploring the functors that preserve this structure. So, not only are structure-preserving functors of interest but also so-called natural transformations which map one functor to another. Design processes which are similar and related with simple ways to go between them are one example of natural transformations.

II. GENERALIZED HIGH FREQUENCY DESIGN FLOW

Analog Mixed Signal (AMS) design flows [4] proceed in a schematic-driven top-down fashion from a set of requirements (Fig. 3). A schematic effectively starts with a single block whose parameters describe the design goals (e.g. gain, power dissipation, PAE, BER, etc.) that is then iteratively partitioned into smaller blocks, each of which terminate in blocks that are implemented as manufacturable elements represented in a physical layout. The physical layout is analyzed in a bottom-up process to include phenomena not captured by circuit simulation, such as electromagnetic coupling or thermal heating, to ensure that secondary effects do not diminish achieving the design goals. Final manufacturing verification precedes releasing the design for fabrication.

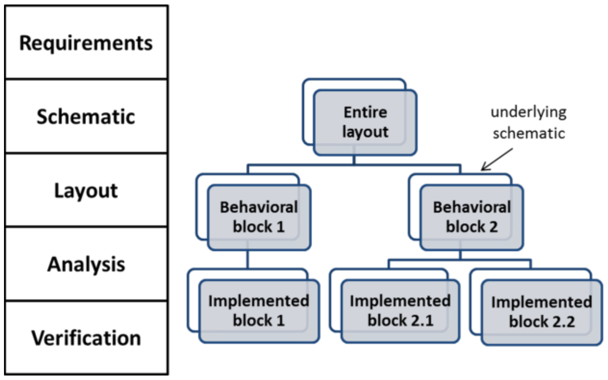


Figure 3 – generalized AMS design flow and supporting hierarchical design data for layout (foreground) and schematic (background).

From this description of design, several categories, and related morphisms, can be identified. The first is a behavior category which captures the performance specifications and the relationships among them. Perhaps most importantly, schematics and layouts as separate but related categories are clearly apparent, with schematic-driven layout or forward annotation going from schematic to layout and back-annotation in the opposite direction. Morphisms from layout and schematic to behaviors represent analysis (EM, thermal, etc.) and simulation, respectively, and a morphism from behaviors to schematics can be identified with synthesis (be it manual design or automated tools). This work will not consider manufacturing verification.

A design flow, on the other hand, manipulates these objects. Two features of design flows will be emphasized here: parametric design and design closure. Parametric design simply means that engineers strive to relate variables (degrees of freedom) defining components of a design solution to the required behavior of that component and, ultimately, the design as a whole. In a very real sense, design is all about finding these key parameters and constraining their numbers and values to achieve an optimal design. Design closure requires that regardless of whether the schematic is simulated or the layout analyzed, top-down or bottom-up through hierarchy, the same behavior is found. The design process goes from nearly infinite design possibilities to the one design implementation by winnowing and refining the parameters describing a design. In other words, a design flow is a composition of morphisms for deciding which of a design's parameters (and their values) best achieve the design goals from all those available. Here we will structure the design process such that a simulated schematic approximately meets the design goals at the end of each "step", where a step is the composition of several morphisms.

III. Categories for Behaviors and Schematics

Three categories—Behavior (B), Schematic (S), and Layout (L)—emerge from the preceding discussion with morphisms among them as labeled in Figure 4. Not shown are the identity morphisms. Morphisms with S as both domain and co-domain (" autmorphisms") represent typical operations on schematics, such as ***pa***rtitioning into sub-blocks, ***i***mplementing behavioral elements with real circuitry, and ***c***onstraining parameters values. Similarly for L, ***pl***acement of cells, ***r***outing of interconnects, and ***c***onstraining parameters and their values. Other morphisms are possible as well.

Definition of B and S as categories are summarized here based on previous work [10]. B is similar to a category in set theory, and has as objects name-value pairs corresponding to behavioral parametrics, such as (gain, 20 dB). Morphisms in B create associations among behaviors, so that we can combine as co-products individual behaviors into larger behaviors, or "requirements", similar to adding elements to a subset. The identity morphism simply returns the original object. Associativity follows from set theory.

S could be represented simply as sets, but this would belie its complexity and richness. A better representation of S as a structural description of schematics and netlists would be a monoid or group (if removal of elements is desired), using a triplet <E, *, e_o >, where E is an elemental basis of schematic elements used to make the schematic, * is a connectivity operator or morphism allowing elements to be added to a schematic, and e_o (assigned to the "open circuit" element) is the unit element connection to any schematic giving the same schematic (i.e. connecting an open circuit, or "no connection" to the terminals of any existing schematic simply returns the initial schematic diagrammatically.)

The ability to represent S as a monoid means that S has products and co-products, along with other structures in category theory that are represented by features of schematics. More structure can be added to S as needed to develop a model for design flow in any number of ways, two of which are worth mentioning. First, in Figure 3, during the design process, the basis changes from behavioral, $E_{B,}$ to elemental, or implementable, in some manufacturing technology, E_E. Second, to capture the notion of parametric design, E should be elaborated as not just a type of element (resistor, capacitor,

$$\begin{tikzcd}
S \arrow[loop above, "{pa, i, c}"] \arrow[d, shift left, "sim"] \arrow[drr, shift left, "f"] & & \\
B \arrow[u, shift left, "synth"] & & L \arrow[ll, "a"] \arrow[ull, shift left, "b"] \arrow[loop below, "{pl, r, c}"]
\end{tikzcd}$$

Figure 4 – category theory diagram for schematic (S), layout (L), and behavior (B) categories

diode for E_E, or amplifier, mixer, filter for E_B) but as the type on a parameter set, P, of name-value pairs. More exactly, by applying the notion of ***c***, the constraint morphism, and using a product of morphisms <1,**c**>,

$$E_n \xrightarrow[<1.c>]{} E_n \; x \; P_n, \tag{1}$$

where the n implies a specific collection, such as behavioral or elemental, elements or parameters, respectively. Substituting into the monoid definition gives

$$S =< E_n \; x \; P_n, *, e_o >, \tag{2}$$

or that S is made of a single object, the monoid, from which all schematics on a basis can be constructed. Change of basis between E_E and E_B as well as automorphisms ***pl***,***i***, and ***c*** are supported [10] by this category.

Functors ***sim*** and ***synth*** operate between B and S. The design process starts by taking the behaviors defined by a parametric set requirements, P_R, as the co-product of individual behaviors represented as parameters,

$$P_R = \; P_{R_1} + P_{R_2+\cdots} \tag{3}$$

as for now there structure is co-product. Additional structure can arise from knowing specifically what some of them are (e.g. DC power requirement would be found by algebraically summing the DC power consumed by all subcircuits using a product). From this, a single behavioral element/schematic from E_B can be instantiated where the P_B , the properties for the E_B, equals P_R,

$$S = synth(P_R) = e_{top} \; x \; P_B \tag{4}$$

and $e_{top} \in E_B$. Automorphisms ***pa***, ***i***, and ***c*** are then applied within the design flow to implement the rest of the schematic portion of the design. At this point the layout can begin.

It is important to note that schematic hierarchy is represented as a co-product just as long as elements at the same level do not interact to change their nature. In other words this requirement in B can be stated as: for a schematic, S_0, made of two elements or sub-circuits in hierarchy, S_1 and S_2,

$$S_0 = \; S_1 + S_2 \tag{5}$$

$$sim(S_0) = sim(S_1 + S_2) = sim(S_1) + \; sim(S_2)$$

$$= B_1 + B_2 = B_0 \tag{6}$$

When (6) holds, the structure of S is preserved in B, but (6) does not necessarily hold in all cases. For example if S_1 represents an oscillator and S_2 a filter, then under certain values of P_B for the oscillator and/or filter, S_2 will cause S_1 to stop oscillating. In this more general case, the product defines the total behavior as it is a function of the interaction between the subcircuits

$$B_0 = sim(S_0) = sim(S_1 + S_2) = sim(S_1) x \; sim(S_2)$$

$$= B_1 x B_2 \tag{7}$$

For what remains here, simulation results on S will be assumed to be constrained by (6) rather than the more general (7). In other words, each element in a schematic simulates independently from the others. This holds, for example, at the elemental level where a resistor is still resistor regardless of what it is connected to.

IV. Category for Layout

The layout category, L, is similar in structure to S. For each element in connectivity, there is an element set, E_L, and a corresponding parameter set, P_L. By similarity to S, L can be defined as a monoid or group with one very critical difference depending on the design flow: interconnects. Three different possibilities are explored shortly in terms of how to treat the interconnects within L and with regard to the other categories, each of which defines a different design flow.

Morphisms within L allow for the development of the layout, Placement, ***pl***, and routing, ***r***, accomplish the standard tasks of layout with ***pl*** focused on representing schematic element in the layout and ***r*** on implementing the schematic "wires" or interconnects. Functor ***c*** allows for the manipulation of element and interconnect parameters without change in placement or routing. Other arrows within L are possible.

Functors ***f*** and ***b*** exchange unplaced element and unrouted connectivity information with S, such that layout and schematic elements map "piecewise" to their corresponding member between layout and schematic. An amplifier or resistor in E_L map to the corresponding element in the schematic hierarchy on basis E_B or E_E, respetively. Functor ***a*** represents any analysis type that yields a behavior, here limited to EM analysis from a practical perspective, but mathematically we expand this to effectively include an LVS netlist so that in the case of no interconnects, S and L are isomorphic

$$S \cong L,$$

and therefore,

$$a(L) = sim(S), \text{ or} \tag{8}$$

$$a \; o \; f = sim(s) \tag{9}$$

Where (9) represents the commuting diagram formed by taking a schematic to layout by forward annotation and then analyzing it—with no impact from wires, there should be agreement with circuit simulation.

For the more general case of interconnects effecting simulated/analyzed behavior, the relationship of ***f*** and ***b*** is not of a true inverse and the isomorphism breaks down. While it is true that one can apply ***f*** to a given schematic, s, to yield a placed and routed layout, l, and then get back to the same schematic via ***b***, the same is not true in reverse because (independent of a merged database and in a purely mathematical sense) applying ***b*** to any layout strips away the physical placement information which cannot be recovered. Thus, for any given layout there is only one schematic (given a basis E), but given a schematic there are many layouts that could correspond to it. Mathematically,

$$b \; o \; f = 1_S, \text{ and} \tag{10}$$

$$f \; o \; b \neq 1_L \tag{11}$$

To represent this within the structure of L, the monoid defining L should be expanded to including not only the layout elements corresponding to elements in (schematic) connectivity, but also to the interconnects. Redefining the monoid as E_{Li} with

$$E_{Li} = E_L + E_i, \tag{12}$$

To consider design closure, (8) gets redefined so that

$$B_L = a(E_L + E_i) = a(E_L) + a(E_i) = sim(E_E) = B_S, \tag{13}$$

is only true if a(E_i) contributes some behavioral value other than null, zero, etc. then B_L and B_S will not be the same and there is no closure

Considering parametric design, we just apply (1) to E_i in (12) to get an interconnect parameter set, P_i. P_i would be values such as width, layer, miter type, via type, or a host of others. Therefore, this expanded definition of L using E_{Li} allows the consideration of both closure and parametric design.

In what follows, the degree to which interconnects are considered and when they are considered is varied to arrive at different design flows: different approaches to handling interconnects. In each case, the flow will be evaluated against design closure and parametric design requirements for the flow itself. The placement morphism is ignored as we will assume that placement has no effect on behavior.

A. Analog design– ignore interconnects in schematic

The analog design style essentially ignores the interconnects as part of the design for low frequencies and are assumed to be electrically benign to simulation. The schematic and layout monoids are built on element sets which are purely component based and do not have the ability to model interconnects at the circuit level. Morphism ***r*** in L simply routes wires between components. Eq (9) clearly holds so there is design closure and since both of the bases for S and L have parameter sets, the design is parametric. More specifically, and with regard to electrical performance only, placement information (parameters) from the layout have no impact on B, so only the schematic parameter set need be considered for the parametric design requirement.

For higher frequencies, the interconnects are analyzed in a post-layout step using parasitic extraction. This will be considered in part C.

B. Microwave design– capture interconnects in schematic

The traditional microwave design flow spaces printed components, such as microstrip and stripline, sufficiently far apart as to minimize coupling. The result of this approach is the ability to fully capture in the schematic, a priori, the interconnects using schematic elements on E_E. The result is parametric design of layout-based components in the schematic. So long as the assumption on minimal coupling is maintained (along with any limitations on model validity), the approach provides design closure.

To accomplish this, E_E is augmented to create a new E_E which contains the original E_E as well as E_W, a family of microstrip, stripline, etc. models which parametrically define layout components. By requiring that there is no routed interconnects in the layout (i.e. ***r*** is eliminated)--so that the E_i--layout equivalents of $E_W$are placed pin-to-pin--this can be satisfied.

In terms of the category theory, an isomorphism is formed between E_W and E_i, namely by applying ***f*** and ***b***

$$E_W \cong E_i, \tag{14}$$

$$P_W = P_i, \tag{15}$$

Were the P are identical parameter sets for these elements in the schematic and the layout. The behavior for these elements in the schematic become

$$(E_W \, x P_W) \xrightarrow{sim} B_W, \tag{16}$$

And by definition, as long as there is minimal coupling

$$(E_i \, x P_i) \xrightarrow{a} B_i = B_W, \tag{17}$$

Without loss of generality, this can be run through a simple top-down design flow in the category theory by taking S_0 containing two subcircuits, S_1 and S_2, are components connected by a subcircuit solely containing the wires, S_W, and seeing if design closure is achieved. By using the (14)

$$S_0 = (S_1 + S_2 + S_w) \xrightarrow{f} f(S_1 + S_2 + S_w)$$

$$= f(S_1) + f(S_2) + f(S_w)$$

$$=(L_1 + L_2 + L_i) = L_0 \quad (18)$$

Simulating the schematic and using (16) gives

$$S_0 = (S_1 + S_2 + S_w) \xrightarrow{sim} sim(S_1 + S_2 + S_w)$$

$$= sim(S_1) + sim(S_2) + sim(S_w)$$

$$=(B_1 + B_2 + B_w) = B_0. \quad (19)$$

And analyzing the layout with (17) gives the identical result gives the identical result by using

$$L_0 = (L_1 + L_2 + L_i) \xrightarrow{a} a(L_1 + L_2 + L_i)$$

$$= a(L_1) + a(L_2) + a(L_i)$$

$$= (B_1 + B_2 + B_w) = B_0. \quad (20)$$

Both design closure and parametric design requirements for the flow are achieved. However, this benefit is attained only if there is a sufficient model set for the interconnects and if enough layout area is available to keep the interconnects widely spaced. In practice, the designer choosing this style of design does it with the model set in mind, so rather than having the layout subflow have the freedom to draw any interconnect, it is presumed that only those interconnects found in E_W will be used.

C. Iterative analysis – analyze interconnects in layout

Perhaps the most widely used high frequency and AMS flow in terms of interconnect design and analysis is the flow whereby interconnects are routed, analyzed and inserted back into the schematic (or netlist) as S-parameters or RLC equivalent circuits and then simulated. Deviations from required behavior, B_0, then starts the whole cycle over again. The analysis step can be performed, in the case of RF/microwave, with a solution to Maxwell's Equations using an EM solver or by parasitic extraction techniques.

To represent this flow with category theory, the EM solver or parasitic extraction tool takes a layout and from it, creates the equivalent of single schematic element, S_i, where

$$S_i = e_s x \, p_s, \quad (21)$$

e_s is a member of E_W and is the equivalent n-port S-parameter block (or equivalent) inserted into the schematic representing the interconnects. p_s is a single parameter indicating whether the analysis is to be included or excluded in any subsequent simulation.

It is necessary now to include the notion of a dynamical design flow. To do this, iterations of a schematic or layout resulting from an automorphism will have a 'prime' (e.g. `) after it. So,

$$L_0 = L_1 + L_2 \xrightarrow{r} L_0 + L_i = \grave{L_0}. \quad (22)$$

Where $\grave{L_0}$ is the placed and routed layout. The goal of the dynamical flow is to have, at each iteration, the behavior of the design meet the requirements, namely B_0.

Using the calculation similar to what was done in the previous sections, for the design after routing, the schematic becomes

$$b \circ r \circ f(S_0) = \grave{S_0} = S_1 + S_2 + b(L_i)$$

$$\xrightarrow{sim} B_1 + B_2 + sim \circ b(L_i)$$

$$= B_0 + B_i \neq B_0, \quad (23)$$

where the back-annotated and simulated behavior of L_i is taken as B_i. The final line is true when the p_s indicates that the interconnects should be included in the simulation. Following a similar approach to (20) and (23) for the routed layout gives a similar result—since the pre-routed schematic and the layout without interconnects is designed to give B_0, adding interconnects of any significance will require a design iteration. Thus, there is no explicit design closure. The argument could be made that the initial design (no "prime") is done based on S such that (23) ends with an equality, but this implies that the design team "anticipated" B_i which for a design of any significance implies either that the microwave design flow was used or that team has achieved an exceptional degree of serendipity. The composition $\boldsymbol{b} \circ \boldsymbol{r} \circ \boldsymbol{f}$ followed by *sim* is easily identified as the iterative aspect of the flow.

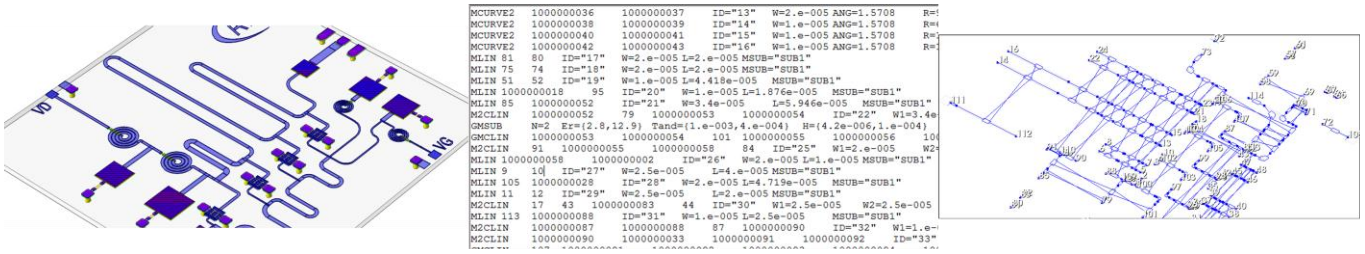
Furthermore, since the only parameter that comes into the schematic with the routing and back-annotation of the interconnects, p_s, is essentially a Boolean value indicating, there is no parametric design of the interconnects when routed in this manner.

Thus, this flow exhibits neither closure (in a guaranteed sense) or parametric design.

D. Hybrid analysis – layout-generated parametric design

The microwave and iterative analysis flows have both benefits and limitations. The microwave design flow provides parametric design at the cost of having to anticipate, on the initial schematic, the interconnects that will be routed. The iterative analysis flow provides freedom to route, but at the cost of limited parametric design and design closure.

The category theory for the iterative and, to a lesser extent, the microwave flows have the design flow driven from the schematic. The layout gets generated from the schematic after the schematic is simulated and shown to agree with the design requirements. The layout then is augmented and checked against the simulation of the original schematic. Both flows do this in different ways. However both flows have one key feature in common which drives the design process: both rely on keeping $sim(S_0) = B_0$ at the end of each iteration.



```
MCURVE2   1000000036    1000000037    ID="13"   W=2.e-005 ANG=1.5708     R=
MCURVE2   1000000038    1000000039    ID="14"   W=1.e-005 ANG=1.5708     R=
MCURVE2   1000000040    1000000041    ID="15"   W=1.e-005 ANG=1.5708     R=
MCURVE2   1000000042    1000000043    ID="16"   W=1.e-005 ANG=1.5708     R=
MLIN 81   80   ID="17"   W=2.e-005 L=2.e-005 MSUB="SUB1"
MLIN 75   74   ID="18"   W=2.e-005 L=2.e-005 MSUB="SUB1"
MLIN 51   52   ID="19"   W=1.e-005 L=4.418e-005   MSUB="SUB1"
MLIN 1000000018     95   ID="20"   W=1.e-005 L=1.876e-005   MSUB="SUB1"
MLIN 85   1000000052    ID="21"   W=3.4e-005     L=5.946e-005   MSUB="SUB1"
M2CLIN    1000000052    79   1000000053    1000000054    ID="22"   W1=3.4e
GMSUB     N=2  Er=(2.8,12.9) Tand=(1.e-003,4.e-004)  H=(4.2e-006,1.e-004)
GMCLIN    1000000053    1000000054    101  1000000055    1000000056    10
M2CLIN    91   1000000055    1000000058    84   ID="25"   W1=2.e-005     W2
MLIN 1000000058     1000000002    ID="26"   W=2.e-005 L=1.e-005 MSUB="SUB1"
MLIN 9    10   ID="27"   W=2.5e-005     L=4.e-005 MSUB="SUB1"
MLIN 105  1000000028    ID="28"   W=2.e-005 L=4.719e-005   MSUB="SUB1"
MLIN 11   12   ID="29"   W=2.5e-005     L=2.e-005 MSUB="SUB1"
M2CLIN    17   43   1000000083    44   ID="30"   W1=2.5e-005    W2=2.5e-005
MLIN 113  1000000088    ID="31"   W=1.e-005 L=2.5e-005     MSUB="SUB1"
M2CLIN    1000000087    1000000088    87   1000000090    ID="32"   W1=1.e-
M2CLIN    1000000090    1000000033    1000000091    1000000092    ID="33"
```

Figure 5 – Prototype results of hybrid flow routing capability that allows routed lines where a schematic modelset supports it. The left view is the layout being routed, the center view is parameterized netlist being generated as the layout is routed, and the right view gives an extracted view.

From this analysis of the category theory, a new flow can be examined consisting of three features. First, the reference for maintaining design closure should be shifted from S to L as routing becomes the critical activity in the design. Therefore, for each "primed" iteration,

$$L_0 = (L_1 + L_2 + L_i) \overset{a}{\rightarrow} a(L_1 + L_2 + L_i) = B_0 \qquad (24)$$

Second, the parametric design of interconnects should be driven by the routing activity to take advantage of the microwave design flow's benefits. One way to accomplish this is to generate S_i from the routed interconnect such that

$$r: L \overset{r}{\rightarrow} L_i |\, L_i \cong S_i \qquad (25)$$

This is clearly not the case for the iterative flow, and for the microwave flow the interconnect elements are introduced to the design via morphism ***p*** on S, not on L. To accomplish (25), the basis for the interconnects in S and L need to be isomorphic (to preserve the structure). Thus,

$$L_i = < E_i x\ P_i, *. e_0 >, \qquad (26)$$

and

$$S_i = < E_w x\ P_i, *. e_0 >, \qquad (27)$$

with

$$E_i \cong\ E_w, \qquad (28)$$

and the P_i are the same for both schematic and layout. This guarantees that we have piecewise design closure as *sim*(Si) and a(L_i) (using EM analysis with the assumption of minimal stray coupling not in the models of *sim*(E_W)) form a commuting diagram with ***b*** and ***f***.

Finally, the freedom of routing in the iterative flow should be maintained as best as possible. This is really about EDA ergonomics, however it does raise issues for tools implementing this flow which will be discussed shortly.

Demonstrating the design flow with category theory, L_0 is generated from S_0 by ***f*** and still maintains agreement with B_0 with an initial design step driven from the schematic side. Continuing with the routing using (25) and (26) with (24)

$$\grave{L}_0 = L_1 + L_2 + L_i \overset{a}{\rightarrow} a(L_1 + L_2 + L_i) = \grave{B}_0 = B_0$$
$$= B_1 + B_2 + B_i = \grave{B}_0 = B_0 \qquad (29)$$

And for the schematic

$$\grave{L}_0 \overset{b}{\rightarrow} \grave{S}_0 = b(L_1 + L_2 + L_i) = S_1 + S_2 + S_i$$
$$\xrightarrow{sim} sim(S_1 + S_2 + S_i) = B_1 + B_2 + B_i$$
$$= \grave{B}_0 = B_0 \qquad (30)$$

By (29) and (30) closure is achieved and by (26) through (28), the flow will be top-down parametric.

V. Implementation of the Hybrid Flow

The previous section identifies a new flow from the category theory representation for the schematic, layout and behavior with additional structure added in the form of (24) through (28). This implements the hybrid flow in the abstract sense only. To truly implement this flow would require having the set of components represented by (28) implemented in a schematic, layout, and simulation tool. Additionally, a procedure for (25) must be established, whether it is manual or automated. Implementing both (28) and (25) will be discussed.

A set of schematic-layout components for most routed interconnects in an IC or PCB process in commercial RF/microwave design software. Microstrip and stripline elements for all "manhattan" routes and discontinuities are fairly accurate, assuming a well-defined current return path. The need for a current return path makes some PCBs and silicon RFICs, in particular, difficult to achieve. Nevertheless, the modelset with corresponding artwork generators that are bidirectional is generally available and includes coupled lines.

To the best of the authors' knowledge, automation for (25) is not currently available. However, AWR's ACE™ Automated Circuit Extractor comes close [11]. ACE is the microwave design flow equivalent of parasitic extraction and converts interconnects in a static (already routed) layout to the equivalent of an S-parameter block, only instead of the block containing S-parameters, it contains a netlist comprised of microstrip, stripline, etc. models: the models implied by (28). Fig. 5 shows a prototype representation, using, ACE of the automatic generation of a subcircuit (here a netlist) while the layout is being routed. Routes are only allowed that match elements with microstrip models. Other routes are disallowed.

To achieve (25) with an ACE-like tool would require at least three additions. First, instead of producing an extracted netlist which is static, an element representing S_i would need to be instantiated and accessible from S. Second the layout and schematic element associated with S_i (and L_i) would need to have parametric values which could be accessed by the engineer for tuning, optimization, sensitivity analysis, etc. Third, given the permissible modelset, it would be highly desirable to have (25) running in real-time while routing lines in the layout, allowing the user to "manual" routes lines only if they can be represented by the modelset. For example, while most modelsets have a microstrip T-junction, MTEE, very few have a model for a Y-junction. Thus, if the designer tried to route a junction with lines at angles not a multiple of 90 degrees, the line could not be routed. Implementing this third feature would satisfy the routing freedom requirement so often desirable and mentioned in section IV D.

Additional features could of course be added. For example, additional models might need to be added for commonly routed structures not currently used in a microwave design flow. The Y-junction might be one of these.

VI. Applications

While the mathematical formalism introduced offers much research opportunity in both depth and breadth, the problems to which it can be applied and their significance are substantial. Here purely through the formalism, a new flow is defined and features for a new EDA tool identified. With a more detailed implementation of this formalism in regard to representing the interconnects and their associated morphisms, it is hoped that new flows and tools could be found to increase microprocessor clock rates beyond the current 2-3 GHz boundary. In analog design, the approach can be extended to look at improving synthesis routines by defining functional structures which tend to be more deterministic and relate parameters to performance. For RF design, as applied here to elements and interconnects, more progress can be made at building co-design platforms spanning simulation and EM, IC and package, or other domains, like thermal.

VI. Conclusion

A simple mathematical formalism for describing and analyzing AMS design flows has been established and applied. Category theory is an ideal branch of mathematics for examining the relationships among dynamic processes executed serially and concurrently. Here, a simple view of design flow comprising behaviors, schematics and layouts was used to capture many of the essential features of top-down hierarchical AMS design at RF/microwave frequencies.

In so much as it pertains to the electrical modeling of interconnects, these flows have been analyzed to see what impact their particular method of incorporating interconnect effects has on design closure and top-down parametric design. With regard to classical microwave design, where interconnects are explicitly modeled on the schematic *a priori* to routing, the category theory showed that both design closure and parametric design are maintained throughout the flow. For the more free form, iterative design approach, the same analysis showed that post-layout analysis as a design tool coupled with back-annotated S-parameters (or equivalent circuit models), does not explicitly guarantee closure, can indeed lead to many iterations, and does not expose these critical layout-based parameters to further design or analysis.

By forcing a set of constraints on the flow, namely layout-driven design closure and interconnect layout-schematic duality, a flow is developed taking the benefits of both the microwave and iterative design approaches. EDA requirements for this tool relative to the current state of the art have been identified for implementation.

Acknowledgement

One author (Svensson) kindly thanks Macquarie University for Macquarie Research Scholarship support.

References

[1] A. Victor and J. Nash, "Design of C-Band Microstrip Voltage Controlled Oscillator Using and Electromagnetic-Harmonic Balance Co-Design Technique,"Asia-Pacific Microwave Conf. 2007 (APMC 2007), pp. 1-4.

[2] D. Wu and S. Boumaiza, "Comprehensive First-Pass Design Methodology for High Efficiency Mode Power Amplifer," IEEE Microwave Magazine, Vol 11. Issue 1, pp 116-121. February 2010.

[3] S. Shin, D. Dawn, D. Yeh, and J. Laskar, "A Model Inaccruacy Aware Design Methodology of Millimeter-wave CMOS Tuned Amplifiers, Wireless and Microwave Tech. Conf 2011 (WAMICON 2011), pp 1-6.

[4] G. Gielen and R. Rutenbar, "Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits," Proc. IEEE, vol. 88, no. 12, pp. 1825–1854, Dec. 2000.

[5] B. Pierce, Basic Category Theory for Computer Scientists, MIT Press, Cambridge, 1991.

[6] N. Sabadini, R.F.C Walters, and H. Weld, "On Categories of Asynchronous Circuits," 1994. Available on http://citeseerx.ist.psu.edu.

[7] S. Mac Lane, Categories for the Working Mathematician, 2nd ed., Springer-Verlag, New York, 2010, pp. 7-10.

[8] S. Awodey, Category Theory, Oxford University Press, Oxford, 2010.

[9] F. W. Lawvere and S.H. Schanuel, Conceptual Mathematics: A First Introduction to Categories, 2nd ed., Cambridge University Press, Cambridge, 2009.

[10] M. Heimlich and C. Svensson, "An Algebra for Analog Mixed-Signal Design Partitioning," unpublished.

[11] ACE Automated Circuit Extraction, AWR Corporation, 2012. http://web.awrcorp.com/content/Downloads/ACE-Datasheet.pdf