MIDAS: Model for IP-inclusive DFM Assessment of System Manufacturability

Kasyab P. Subramaniyan and Per Larsson-Edefors
VLSI Research Group, Department of Computer Science and Engineering
Chalmers University of Technology, SE-412 96 Gothenburg, Sweden
E-mail: {kasyab,perla}@chalmers.se

***Abstract*—Complex system implementations combined with the latest technology nodes allow us to implement hardware for versatile applications. The ever increasing demand for quick time-to-market has led to the widespread use of Intellectual Property (IP) in ASIC design methodologies. These developments, in addition to manufacturing limitations, make early prediction of manufacturability for complete systems challenging. We present MIDAS: a scalable, IP-inclusive model to predict system manufacturability. Results from applying MIDAS to an embedded processor system reveals that several useful insights can be gained towards realizing yield budgets for complex systems allowing quicker co-optimization of all implementation goals.**

***Index Terms*— Manufacturability, ASIC, IP, DFM Metrics.**

I. Introduction

System implementations with a robust cost-effort tradeoff use standard-cells as a distinct level of abstraction in the design of digital circuits. Due to the growing complexity of design management, macros of sub-systems have become indispensable [1]. These macros may be memories or other hard Intellectual Property (IP) functions needed in the system. Typically, the macros are provided for use to the customer as a black box, with verified functionality guarantees from the vendor. Thus, integrating such blocks into a system eases the functional and performance verification effort on the part of the system designers. However, the macros, when considered for place and route, have constraints such as routing blockages which the layout engineer must account for during the place and route stage. Considering the widespread use of standard-cell methodologies and the ever increasing use of IP in complex yield-limited environments, it is important to consider the implications of integrating big macros alongside a collection of small standard-cells on manufacturability [1].

Manufacturability analysis of standard-cells has been carried out from the perspective of yield [2], gate length distribution [3], [4], sensitivity analysis [5], and considerations such as reliability and routing [6]. Regular cell layouts have also been proposed as a means to enhance manufacturability [7]. While qualitative Design For Manufacturability (DFM) guidelines have been the main focus of existing literature, Gomez et al. [6] explicitly propose a quantitative manufacturability metric for standard-cells. Other attempts to introduce a metric for DFM have been carried out in [7]. From the perspective of IP, Aitken [8] examines existing DFM metrics and practises. He does not propose any quantitative metric specific to IP but concludes that careful attention to DFM practices is required in the face of challenges imposed by explicitly incorporating variability into testing.

In this work, we propose MIDAS (**M**odel for **I**P inclusive **D**FM **A**ssessment of **S**ystem manufacturability): an additive model to compute a simple DFM metric to enable early assessment of DFM for System-on-Chips (SoCs). "Early" in this context refers to the earliest stage where realistic physical data become available. We hypothesize that if DFM costs for the standard-cells and IP blocks can be established, then system-level routing determines the overall manufacturability of the SoC. We can view standard-cells, IP blocks and system-level routing as discrete contributors towards the manufacturability. Critical Feature Analysis (CFA) is used to motivate the hypothesis in the next section. We subsequently demonstrate the applicability of the proposed model in early analysis of DFM using an embedded processor system. The MIDAS model builds on existing techniques and extends the ability to coarsely predict manufacturability early in the design flow.

II. Motivation

We quantitatively motivate the MIDAS model through traditional DFM assessment of benchmark circuits from the ISCAS'89 [9] and IWLS'05 [10] suites, and also an embedded processor system (see Section III-A for details).

After place and route, the implementations were imported into the full-custom design environment for DFM assessment, which is enabled through Calibre Critical Feature Analysis (CFA) [11], using foundry-provided rule sets. This tool is a part of the suite of full-custom tools enabling Design Rule Checking (DRC) and Layout Versus Schematic (LVS) checks. CFA relies on detailed rule- or model-based checks to provide metrics on resilience to modeling accuracy, particle defects and process margins[1]. Scores from individual (categorized) rules are summed to form the Weighted DFM Metric (WDM) and the result is normalized to a number based on the number of transistors in the design. A bound is established using the negative exponentiation of the normalized value to give the Normalized DFM Score (NDS). The WDM can have any value from 0 to infinity, while the negative exponentiation restricts the value of the NDS between 0 and 1. Being cumulative, a lower WDM is desirable for manufacturability or, conversely, a design with a NDS approaching 1 has greater resilience to process defects.

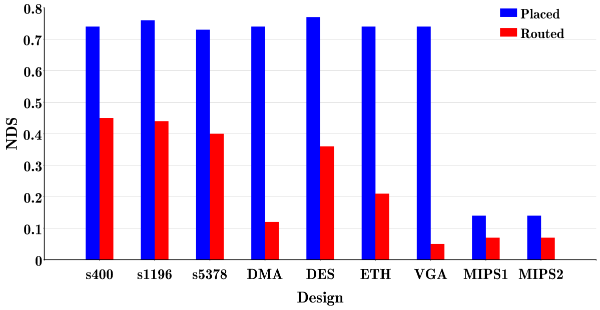


Fig. 1. CFA for placed and routed designs.

In order to accurately capture the effects of all the system-level constraints, stream data was saved for the placed design as well as the routed design so that the results of CFA could be compared. Figure 1 shows the results of the CFA analysis. All designs except the last two are benchmark circuits from the ISCAS'89 [9] and IWLS'05 [10]

[1] "Process margin" refers to tolerances that layout features exhibit to defects induced as a result of process steps like lithography, Optical Proximity Correction (OPC), and Chemical Mechanical Polishing (CMP).

suites. The MIPS1 and MIPS2 designs are variants of the embedded processor system, details of which are outlined in Section III-A.

It is clear from Figure 1 that, irrespective of the design size[2], system-level routing affects the NDS; by as much as 70% in some cases. It must be noted here that the generally low NDS values for the MIPS designs occur as a result of the memory macros (the IP components) present in the design. The hard macros used in the implementations are geometrically accurate, but have the active device layers abstracted out. This results in inaccuracies in the NDS computation, additionally so due to the area impact of the macros on the overall area. Excluding the macros from consideration during assessment increases the NDS value to match the NDS for the benchmark circuits proving that complete geometry data is necessary for accurate computation.

It can also be seen that the NDS for the placed designs is almost constant throughout (about 0.75 for the benchmark circuits and 0.14 for the MIPS designs), leading to the conclusion that the system building blocks present a base cost towards manufacturability. The fact that this value degrades to the NDS of the routed designs means that the system-level wiring alone contributes to this degradation. Thus, for a coarse estimate, the main contributions towards assessing DFM can be viewed discretely as the building blocks of the circuit and the system-level routing.

In addition to quantitatively motivating the contributors towards system manufacturability, we use this traditional DFM flow to generate base costs for the standard-cells used in this study. The cost so computed is applied in the early DFM assessment model.

Section III outlines the background, presenting the infrastructure involved at various levels of abstraction. Section IV outlines the various components of the proposed model and the overall DFM metric. Validation results from the MIDAS model are presented in Section V followed by a demonstration of IP inclusion into the model in Section VI. Finally, the conclusions of this study are presented.

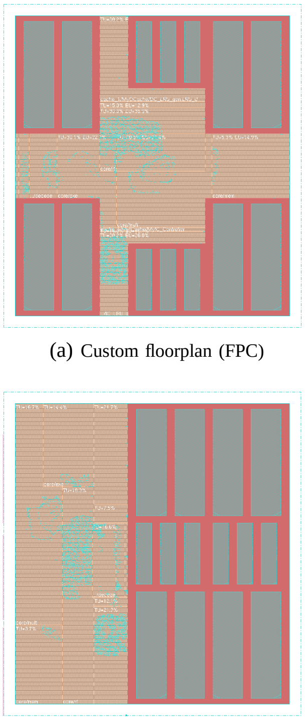
III. Environment and Tools

In order to be able to show the applicability of MIDAS, it is important to target a system that is complex enough to require different blocks (cells vs macros). The test vehicle used to achieve this is an embedded processor system. Additionally, given that the MIDAS model is based on component costs, details of the building blocks (cell or IP) are required. The following headings outline the details at various levels of abstraction. Note that this work uses EDA tools from Cadence Design Systems [12] for full-custom (Virtuoso) and semi-custom (EDI) implementation environments.

A. System-Level Implementation

We use a MIPS processor with a five-stage pipeline [13] and a level-one (L1) cache as the test vehicle in this work. The CPU consists of the standard pipeline units of fetch, decode, register file, ALU, and memory write-back and is augmented with a 32-bit integer multiplier. Each of the 16kB L1 data and instruction caches is implemented with four SRAM memory macros of size 1024x32-bit and three 128x32-bit SRAM blocks for tags [14]. The processor datapath has about 10K logic cells.

Additionally, we implement the processor system using two different floorplans, which utilize the memory macros in different positions in order to explore the sensitivity of the model to different system-level considerations. The floorplan, in combination with the routing blockages presented by the macros, determines the routing solution for the system. This, in combination with settings varying the row density (resulting in larger or smaller dies) and the different libraries available (see Section III-B) for implementation, enables a viable number of test points to be generated. The memory macros used to implement the cache and tags are the same in all implementations and enforce routing blockages for metal layers up to M5. The macros are placed such that they lie in close proximity to the control blocks.



(a) Custom floorplan (FPC)

(b) 'Industrial' floorplan (FPI)

Fig. 2. Implemented processor system floorplans.

The first exploratory floorplan is a custom-made one referred to by the acronym "FPC" from here on. The other, a floorplan similar to those seen in industrial processor designs, is referred to by the acronym "FPI" for the rest of this work. The floorplans are laid out as shown in Fig. 2 for implementation with the different library sets.

B. Standard-Cell Libraries

One of the most important aspects involved in MIDAS is to be able to assign base costs to standard-cells. To this end, we develop the standard-cells that are used in the implementations. This allows us to have complete control over the data generation process. Additionally, cell libraries with distinct characteristics and for which accurate costs can be established are available for use with MIDAS.

The shapes and geometries of the devices in the first of the custom libraries match those available in commercial standard-cells. In addition, routing is completed using poly wherever possible. We will refer to this library using the tag "PoR" from here on. The second library contains cells with device widths which are uniform and, additionally, unidirectional poly routing is adopted. In the case of this library, routing is completed using M2 in the vertical direction only. In order to keep the amount of M2 in the cells to a minimum, it was decided to allow small M1 jogs. We will refer to this library using the tag "M2R" from here on. A variant of the M2R library, using only M1 routing, is also available and is termed "M1R".

Each of the library variants consist of all cells required for logical completeness, non-inverting buffers, half- and full-adders, complex gates (like And-Or-Invert), XOR gates and flip-flops. The drive strengths were restricted to minimum (X2) and twice the minimum (X4), owing to the effort involved in creating a large number of cells.

[2] After synthesis, s400 has about 100 cells while VGA has about 40K cells. The MIPS implementations contain about 10K cells and 14 memory macros each.

All the libraries were developed using an industrial 65-nm full-custom flow and industry standard EDA tools.

IV. MIDAS: Model for IP-inclusive DFM Assessment of System manufacturability

The computation of any DFM metric requires details of the physical implementation and the sections immediately preceding this have provided the background for the implementation of the designs considered in this work. From the motivational data presented in Section II, we can identify two main components in a system-level implementation:

- The device components comprising standard-cells and IP blocks.
- The interconnect components comprising wires and vias.

The cost of standard-cells is computed using CFA in this work as indicated earlier, while IP cost can either be a pre-computed CFA metric or coarsely estimated by other means. Predicting the manufacturability of a particular routing solution requires some knowledge of the manufacturing process, but is nonetheless simple once the basis for computation is established. Thus, complexity introduced by way of estimation of standard-cell and macro costs, is abstracted away in the computation of the system-level metric.

The MIDAS model, being additive, does not require extensive flowchart representation. Once the costs for the various components are available, a simple script embedded in the implementation tool of choice should provide results. This has the added advantage of a high degree of customizability for the design under consideration. The hardest part of using this model is establishing the various costs, and we demonstrate the process of arriving at those costs in the following sub-sections.

A. Placement Cost

The device components comprise the standard-cells and the IP, which are interconnected in some fashion to form an SoC. The cost for such blocks can be modeled using techniques such as CFA in order to obtain as accurate a value as possible. However, IPs are typically available as macros for which detailed implementation details are scarce. In such a scenario, alternate means must be employed to assess a cost for such blocks.

In an IP-inclusive scenario, the total **Placement Cost (PC)** is simply the sum of the placement costs for standard-cells and IP blocks. This is expressed as:

$$PC = PC_c + PC_m \tag{1}$$

1) Standard-Cell Cost: We begin by considering the WDM for the custom cells as a measure of placement cost for the standard-cells. The PC for standard-cells (PC_c) can then be modeled as a product of the number of instances of a given cell and its WDM:

$$PC_c = \sum_{i=c1}^{cK} N_i \times WDM_i \tag{2}$$

Here *c1* and *cK* refer to the distinct types of cells in the design, N_i refers to the number of instances of a particular cell, and WDM_i is the cost associated with a single instance of the cell.

In this work, since the size of the cells in the custom libraries is limited, the spread of the NDS is also limited (Figure 3). The inverters in the libraries display the lowest values of NDS and represent the lower bounds of the spread. We use the product of the average WDM value of the library and the number of cells as the PC in order to ease the computational effort. A typical commercial library contains a much larger spread of drive strengths that will make it necessary to utilize accurate cost values in order to accurately assess the standard-cell cost. However, with full automation of the process a much more accurate computation can be carried out in order to increase the accuracy.

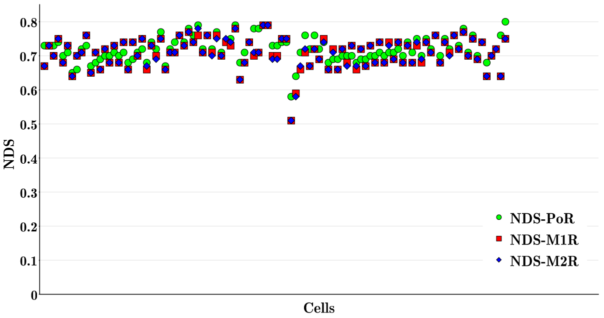


Fig. 3. Scatter plot of NDS values for cells in the custom libraries.

2) IP Cost: Hard macros or IP blocks incur a placement cost in the system-wide context depending on the floorplan and the routing obstructions that the block enforces. The floorplan, influenced by the macros, also affects the core area of the SoC as well as the routing. The obstructions presented by the IPs mainly affect the routing. In the context of placement, the placement cost of incorporating IPs can be described as:

$$PC_m = \sum_{i=m1}^{mK} N_i \times C_i \tag{3}$$

Here *m1* and *mK* refer to the distinct types of macros present in the design, N_i refers to the number of instances of each type of macro, and C_i refers to the weight of the IP block in question, be it the WDM or any other measure used.

Availability of an accurate cost certainly increases the accuracy of MIDAS and assumes great importance when the paradigm of IP-dominated designs is taken into account. However, for a coarse estimate the cost of an IP block can be approximated using known WDM values. Consider a memory macro of size 1024x32b, which is a hard macro with abstract active layers in the test implementations. It is known that each cell in the SRAM memory core consists of six devices, so if we consider the cost per cell using the WDM of a 6-device logic gate, then the cost per memory cell can be approximated to 1.5. The total cost of the memory core[3] can then be computed as 1,024x32x1.5 = 49,152. If the number of logic cells in the memory macro is assumed to be the same as the number of core memory cells, then this cost can be doubled to give a value of 98,304. Accounting for the dense, regular nature of the macro, a conservative cost of 90,000 is used for computations in subsequent sections. Similarly the 128x32b macro is assigned a cost of 9,000.

B. Interconnect Cost

Interconnect cost can be split into two distinct components, vias and wiring, each requiring individual treatment. The total **Interconnect Cost (IC)** is simply the sum of interconnect cost of vias and wiring:

$$IC = IC_v + IC_w \tag{4}$$

The following headings detail each of the components. *Weight*, as used in this context, can be considered to be a product of the criticality of a component or geometric feature and the risk in a given geometric context. Indeed, computations of this type are applied in various risk assessment schemes such as Failure Mode and Effect Analysis (FMEA) [15], [16]. As such, both the criticality and risk values are empirically determined and assigned by the foundry. However, with some experience, for coarse estimates realistic values can be assumed. The considerations for weighting are explained for each case in the following subsections.

[3] The WDM for DMA (~25K cells, WDM of 50940) and ETH (~28K cells, WDM of 62310) benchmarks have comparable values. Note, however, that in these cases there are a number of diverse standard-cells in the design.

1) Layer Change Cost: Manufacturing limitations create risks when vias are introduced while changing layers. Long recognized as one of the yield-limiting features [17], this forms one component of the interconnect cost. A general equation to represent the via cost is:

$$IC_v = \sum_{i=vsc}^{vmc} N_i \times R_i \times C_i \tag{5}$$

The bounds of summation, *vsc* and *vmc*, refer to the types of vias used in the implementation. These, in order of decreasing risk, are single-cut vias and multi-cut vias. The R_i term refers to the risk for a particular type of via, while C_i refers to the criticality. The risk and criticality associated with a particular type of via is typically dependent on empirical values that the foundry determines. Thus, knowing the number of instances of each type of via enables us to weight it reasonably to compute the cost of vias of a design.

As a matter concerning accuracy, it must be noted here that further granularity can be obtained by using instances for layer pairs with more accurate weights to ascertain this cost. The expression for the cost of vias is then modified to:

$$IC_v = \sum_{i=vsc}^{vmc} \left[\sum_{j=lp1}^{lpK} N_{ij} \times R_{ij} \times C_{ij} \right] \tag{6}$$

Equation 5 is used exclusively in this work. Here we assign a via risk of 0.08 for single-cut vias and 0.02 for multi-cut vias. Assuming criticality of 5 and 3 for single-cut and multi-cut vias, respectively, the weight can be computed as a product of the risk and criticality. Statistics of the numbers of each type of via are obtained through the EDI command `pdi report_design`.

2) Wire Spacing Cost: In typical semi-custom design flows, the wire layers are directionally constrained to either be horizontal or vertical in order for heuristic routing to work. Thus, the weight due to a certain layer is limited, since the criticality for wire segments running in the same direction becomes a function of the space between them alone. Additionally, the different layers can be categorized into bins depending on the similarity of their geometries. Typically, lower layers display smaller geometries and pitches, and thus warrant a higher criticality. Risk is assigned based on the pair-wise spacing in a layer, in multiples of minimum spacing as required of DRC. A pair separated by the minimum space is more prone to defects than one with a pair with larger spacing. However, it is not critical to consider wire widths. While this is an important parameter that should be exploited to gain increased resilience to electromigration and noise immunity, the measure of wire-widening is never applied at the cost of area. Hence from an early estimation perspective, it is more critical to include meaningful spacing statistics. Thus, as alluded to earlier, layer-wise data on spacing is sufficient to compute a coarse cost of routing in order to establish a DFM metric.

Such a wire spacing cost can be represented as:

$$IC_w = \sum_{i=b1}^{bn} \left[\sum_{j=l1}^{lK} N_j \times C_j \right] \times R_i \tag{7}$$

As before, according to this notation, C_j represents criticality of layer *j* while R_i is the risk associated with bin *i*. In this work, we use layer-wise spacing statistics produced using the EDI command `pdi report_dfm_metric` [4]. Layers M1 through M3, in the eight layer process used for the implementations, comprise the first criticality bin and are assigned a criticality of 5. Similarly, layers M4 through M6 are assigned a criticality of 3 and the top two layers are assigned a criticality of 1. The risk for computing IC_w is assigned based on the spacing bins: instances with minimum spacing are assigned a risk of 0.9; those with twice the minimum spacing are assigned a risk of 0.2 and instances at three times the minimum spacing are assigned a risk of 0.05. Instances having a spacing greater than this are judged to be more or less immune to the vagaries of the manufacturing process.

C. Total DFM Cost and Normalization

Sections IV-A and IV-B cover the components of the early DFM assessment model. The placement components are governed by Equations 1, 2 and 3, while the routing components are governed by Equations 4, 5 and 7.

The total Design Manufacturability Cost (DMC) of the design can now be expressed as:

$$\mathbf{DMC} = \mathbf{PC} + \mathbf{IC} \tag{8}$$

This represents the overall cost of manufacturability of the design, while each of the individual components represents a measure for the manufacturability arising out the more abstract design decisions of the respective components. In order for the DMC to be useful it must be normalized. The normalization in this work is carried out against a value representing worst-case cost. This normalization cost holds little meaning in terms of a product, but is a theoretical representation of the worst-case risk indicative of a non-functional design. This value can be computed by assuming the highest criticality and worst bins for all components of the MIDAS model. For standard-cells, this is simply the product of the total number of cells and the worst WDM among them. The macro cost, if applicable, is the product of the number of macros and the cost of the macros. This cost is typically constant across the calculations, since implementation details for IP are typically unavailable. For worst-case routing cost, we consider all vias to be single cut and all the wire instances reported by `pdi report_dfm_metric` to be in the M1 layer with minimum spacing. Equations 9, 10, 11 and 12 show all of the component expressions.

$$PC_{cwc} = N_{sc} \times WDM_{worst}, \tag{9}$$

$$PC_{mwc} = \sum_{i=m1}^{mK} N_i \times C_i, \tag{10}$$

$$IC_{vwc} = N_v \times R_{sc} \times C_{sc}, \tag{11}$$

$$IC_{wwc} = N_{wi} \times R_{MinSpace} \times C_{M1} \tag{12}$$

and finally, the normalizer can be expressed as:

$$\mathbf{Norm} = \mathbf{PC_{cwc}} + \mathbf{PC_{mwc}} + \mathbf{IC_{vwc}} + \mathbf{IC_{wwc}} \tag{13}$$

The DMC computed in Equation 8 can now be normalized to this value to express the fraction of the design cost to the total worst-case cost. The Design Manufacturability cost Normalized (DMN) is expressed as:

$$\mathbf{DMN} = \frac{\mathbf{DMC}}{\mathbf{Norm}} \tag{14}$$

A figure-of-merit (FoM) for manufacturability can then be expressed as:

$$\mathbf{FoM} = (\mathbf{1} - \mathbf{DMN}) \tag{15}$$

This value is indicative of the total risk that can be *avoided* as a result of the design decisions related to floorplanning, choice of standard-cells and IP selection.

V. Model Calibration

In order to test the sensitivity of the MIDAS model to various DFM considerations, we implemented the datapath portion of the MIPS system described in Section III-A. Among the various considerations tested at this level were:

1) *Sensitivity to cell architecture*: Different logic libraries, described in Section III-B, were employed in the implementation of the MIPS datapath to test the sensitivity of MIDAS to standard-cell architecture.

[4] The NanoRoute router actually provides both `pdi report_design` and `pdi report_dfm_metric`.

TABLE I

COMPUTATION OF AN EARLY DFM METRIC FOR THE MIPS DATAPATH.

Lib.	PC	IC	DMC	Normalizer	DMN	FoM	% Full	% Mod	Comment
PoR	13554.57	402756.51	416311.08	1879996.34	0.22144	0.77856	-	-	**Full datapath.** FoM calculated using the WDM.
M1R	14456.64	313253.98	327710.62	1659795.88	0.19744	0.80256	-	-	
M2R	14580.96	329828.95	344409.91	1738066.48	0.19816	0.80184	-	-	
PoR	48699.45	339559.49	388258.94	1661614.58	0.23366	0.76634	-1.57	-	**ALU as a macro;** using model for FoM computation.
M1R	46063.67	294778.73	340842.40	1604774.85	0.21239	0.78761	-1.86	-	
M2R	47626.64	308431.13	356057.77	1716607.05	0.20742	0.79258	-1.16	-	
PoR	14610.22	339559.49	354169.71	1627525.35	0.21761	0.78239	0.49	2.09	**ALU as a macro;** using WDM for FoM computation.
M1R	16088.33	294778.73	310867.06	1574799.51	0.19740	0.80260	0.005	1.90	
M2R	16819.87	308431.13	325251.00	1685800.28	0.19294	0.80706	0.65	1.83	
PoR	97138.78	291808.35	388947.13	1487075.54	0.26155	0.73845	-5.15	-	**Multiplier as a macro;** using model for FoM computation.
M1R	83123.29	267881.74	351005.03	1485928.15	0.23622	0.76378	-4.83	-	
M2R	80331.74	259595.83	339927.57	1478428.54	0.22992	0.77008	-3.96	-	
PoR	18322.87	291808.35	310131.22	1408259.63	0.22022	0.77978	0.16	5.60	**Multiplier as a macro;** using WDM for FoM computation.
M1R	19797.03	267881.74	287678.77	1422601.89	0.20222	0.79778	-0.60	4.45	
M2R	19464.91	259595.83	279060.74	1417561.71	0.19686	0.80314	0.16	4.29	
PoR	132095.02	242235.87	374330.89	1422552.10	0.26314	0.73686	-5.36	-	**ALU and multiplier as macros;** using model for FoM computation.
M1R	114435.80	230060.05	344495.85	1385292.93	0.24868	0.75132	-6.38	-	
M2R	112717.34	222045.30	334762.64	1368710.47	0.24458	0.75542	-5.79	-	
PoR	19189.88	242235.87	261425.75	1309646.96	0.19962	0.80038	2.80	8.62	**ALU and multiplier as macros;** using WDM for FoM computation.
M1R	21134.20	230060.05	251194.25	1291991.33	0.19442	0.80558	0.38	7.22	
M2R	21043.74	222045.30	243089.04	1277036.87	0.19035	0.80965	0.97	7.18	
PoR	132262.70	296639.77	428902.47	1575027.36	0.27231	0.72769	-6.53	-	**ALU and multiplier as macros;** with routing blockages; using model for FoM computation.
M1R	114574.92	264961.14	379536.06	1525187.57	0.24885	0.75115	-6.41	-	
M2R	112803.18	265044.35	377847.53	1518056.29	0.24890	0.75110	-6.33	-	
PoR	19357.56	296639.77	315997.33	1462122.22	0.21612	0.78388	0.68	7.72	**ALU and multiplier as macros;** with routing blockages; using WDM for FoM computation.
M1R	21273.32	264961.14	286234.46	1431885.97	0.19990	0.80010	-0.31	6.52	
M2R	21129.58	265044.35	286173.93	1426382.69	0.20063	0.79937	-0.31	6.43	

2) *Sensitivity to IP inclusion*: The ALU and multiplier which are employed in the MIPS datapath were constructed as macros to test the behavior of MIDAS in the presence of macros of different sizes.
3) *Sensitivity to IP cost*: The sensitivity to the cost of including IPs was tested using the MIPS datapath. The overall metric was computed using the WDM and again, using the cost occurring as a result of the MIDAS model.
4) *Sensitivity to routing blockages*: In order to test the MIDAS model for effects introduced by routing blockages in IP blocks, the ALU and multiplier were implemented as macros with routing blockages.

Data required for MIDAS were collected from the different implementations. The results of the FoM computation are presented in Table I. Here, for each of the MIPS datapath implementations, the first column shows the logic library used in the implementation, while the last column describes the constraints of the implementation. Columns two through seven indicate the PC, the IC, the DMC, the normalizer, the DMN, and the FoM. In the two columns following the FoM, the percentage change of the FoM is displayed for two cases: The FoM for a particular implementation compared to the "Full datapath" implementation (titled % Full) and the FoM calculated using the WDM as compared to the FoM calculated using MIDAS (titled % Mod). Note that the "Full datapath" implementation serves as a reference since, consisting entirely of standard-cells, the most accurate costs are available for this implementation.

A number of observations can be made in Table I. The FoM values in the results here are not extremely sensitive to the cell architecture as a result of the fact that average values are used in the estimation. In reality a number of factors other than this affect the value. For example, in order to ensure power efficiency, a number of libraries with different threshold voltages are usually mixed, resulting in different costs for the cells. If instances of cells from the different libraries occur in substantial numbers, which is likely to be the case for a larger design, the effect on the accuracy will be more pronounced. Additionally, if the actual cell costs are incorporated instead of the average, the FoM will be more accurate. From these results, however, it can be said that the libraries with more regular geometries (M1R/M2R) result in a marginally better FoM than the less regular library (PoR). Note that this is the case in spite of the fact that the average WDM is worse for the M1R and M2R libraries when compared to the PoR library (1.48 vs. 1.31).

Table I also shows that when the DFM model is used to create the cost for macros, the estimation tends to be pessimistic. This can be established from the fact that when the FoM for such implementations (rows with "using model for FoM computation") are compared against the FoM predicted for the "Full datapath" implementation (for similar libraries), smaller values are predicted. In these results up to ~7% pessimism is observed. In contrast to this, usage of WDM (rows with "using WDM for FoM computation") for assigning macro costs is more optimistic with predictions up to ~3% higher. The FoM does not change substantially when both the ALU and multiplier are included as macros showing that the sensitivity to IP inclusion is tolerable.

On a related note, using WDM values for macros during computation of the FoM results in more optimistic prediction than using the model itself. Note that the implementation for which the manufacturability is being assessed stays the same; only the method of assigning cost for the macro changes. Up to ~9% higher values are seen in this comparison. The particular case for which this occurs is the implementation using the PoR library with both the ALU and the multiplier as macros and no routing blockages enforced. Note that, when compared against the "Full" implementation, the FoM using MIDAS is ~5% less while the FoM using WDM is ~3% more. This shows that an accurate cost for the IP provides a better estimate for the SoC, confirming the need for accurate DFM metrics for IPs.

That said, the estimation provided by MIDAS shows tolerable error considering that this is early estimation. Considering the last two rows in Table I, we observe that the MIDAS model does not seem to display any sensitivity to routing blockages. This is because there is no penalty assigned to using the upper level metal layers for routing. The only consideration is a legal routing solution that is verified through traditional means.

Blockages affect the wire length and the number of vias. From the design statistics for M1R-based datapath implementations with the ALU and multiplier implemented as macros, as a result of deliberately introduced blockages, there are 94 more cells, 11% more routing and 2% more vias. A similar trend is seen for implementations with the other libraries as well and this in turn will affect parametric yield and timing closure if not accounted for during later design stages.

VI. Metrics in IP-Limited Designs

The results in Table I show that MIDAS provides a reasonable estimate of the manufacturability of a design. However, the effects of floorplan and cell density in an IP-limited scenario remain to be tested. For this purpose we use implementations of the MIPS system described in Section III-A along with the libraries in Section III-B. The initial cell density is specified during the configuration phase of the design and manual floorplanning was used to ensure that placement violations did not occur as a result of the macros. The initial cell densities used were 30%, 50% and 70%. In the last two cases, the die area generated using the default density had to be resized in order to legally accommodate all the memory macros. This shows that in an IP-dominated SoC, the cell density settings are dominated by the IP geometry.

The placement cost varies very little and can as such be considered constant for a given design with a particular library and IP set. The interconnect cost on the other hand varies quite substantially. The largest IC is 28.4% larger than the least, while on average the FPI floorplan yields ~5% less interconnect cost. The trends seen earlier with respect to the effect of the logic library on the FoM continues here with the M1R and M2R libraries displaying better manufacturability than the PoR library.

As a final test, with the same implementations the weights were changed: the IP costs were increased 10% and the via risks reduced by an order of magnitude. It is worth noting that under these conditions the FoM trends remained roughly the same, indicating that the MIDAS model scales mainly according to the design.

In terms of prediction, the FoM provides a scaled measure of the total risk that can be avoided with the current combination of standard-cells, IP and floorplan. The individual components—the PC and the IC—provide a design-specific measure of the risk contributed by each of the components. Splitting this down further enables more specific diagnosis; as a general rule, the greater the granularity, the greater the capability of specific diagnosis. For example, if multiple libraries are involved, then library specific sub-totals can indicate how an optimal mix of cells can be used to achieve overall yield targets. If the cost of a particular IP (ideally, internally created) is high when used in a design specific scenario, it may warrant changes to enable meeting overall goals.

Note, however, that MIDAS does not specifically pin-point DRC violations or parametric violations. These must be dealt with in other ways so as to ensure a clean hand-off to the foundry.

VII. Conclusions

We have presented MIDAS, a model to enable early prediction of DFM, built on the basis of the hypothesis that standard-cells, IP and routing components contribute discretely to manufacturability. The model uses spacing-related routing statistics in addition to costs for standard-cells and IP blocks ascertained using existing DFM techniques, to determine a FoM for the manufacturability of a design. The MIDAS model is calibrated for different considerations on a MIPS datapath design and is then demonstrated on a processor system with an L1 cache. Commercial memory macros were used in the implementation of the cache. Different floorplans and custom logic libraries demonstrate the capabilities of MIDAS. From the results presented in Sections V and VI, it can be concluded that the FoM so established indicates the amount of risk that can be *potentially avoided* in a design implemented with a given set of standard-cell libraries and IP-blocks. By considering cost of constituent cells and blocks as a base cost, the technology-related details can be abstracted out allowing for the computation of the cost of system-level routing. Additionally, by relying on weighting factors provided by the foundry, the FoM can track the yield-ramp of a given technology node.

Acknowledgements

The authors would like to acknowledge the ProViking program and the Swedish Foundation for Strategic Research for funding this work. Thanks also to our foundry partner for making the design kits and DRC/DFM/LVS rule decks available for this work. The authors would also like to acknowledge the efforts of Lars Svensson in reviewing the paper and making valuable suggestions.

References

[1] International Technology Roadmap for Semiconductors 2011 Edition, “System Drivers,” ITRS System Drivers report. [Online].

[2] H. Heineken, J. Khare, and M. d’Abreu, “Manufacturability Analysis of Standard Cell Libraries,” in *Proc. of Custom Integrated Circuits Conf.*, May 1998, pp. 321–324.

[3] H. Muta and H. Onodera, “Manufacturability-Aware Design of Standard Cells,” *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E90-A, no. 12, pp. 2682–2690, Dec. 2007.

[4] H. Sunagawa, H. Terada, A. Tsuchiya, K. Kobayashi, and H. Onodera, “Effect of Regularity-Enhanced Layout on Printability and Circuit Performance of Standard Cells,” in *Proc. of Int. Symp. on Quality of Electronic Design*, Mar. 2009, pp. 195–200.

[5] S. Sundareswaran, R. Maziasz, V. Rozenfeld, M. Sotnikov, and M. Konstantin, “A Sensitivity-Aware Methodology to Improve Cell Layouts for DFM Guidelines,” in *Proc. of Int. Symp. on Quality Electronic Design*, Mar. 2011, pp. 1–6.

[6] S. Gomez and F. Moll, “Evaluation of Layout Design Styles using a Quality Design Metric,” in *Proc. of IEEE Int. SOC Conference*, 2012, pp. 125–130.

[7] T. Jhaveri, V. Rovner, L. Liebmann, L. Pileggi, A. Strojwas, and J. Hibbeler, “Co-Optimization of Circuits, Layout and Lithography for Predictive Technology Scaling Beyond Gratings,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 4, pp. 509–527, Apr. 2010.

[8] R. Aitken, “The Design and Validation of IP for DFM/DFY Assurance,” in *Proc. IEEE Int. Test Conf.*, Oct 2006, pp. 1–7.

[9] ACM/SIGDA benchmarks (NCSU resource), “ISCAS Benchmark Circuits,” ACM/SIGDA benchmarks homepage. [Online].

[10] C. Albrecht, Cadence Research Laboratories at Berkeley, “IWLS 2005 Benchmarks,” IWLS 2005 Benchmarks homepage. [Online].

[11] Mentor Graphics, *YieldAnalyzer and YieldEnhancer Reference Manual*, 2010, Calibre DFM Suite Datasheet.

[12] EDI ver. 10.12, Virtuoso ver 5.1.41, ELC ver. 10.12, Cadence Design Systems Homepage.

[13] D. A. Patterson and J. L. Hennessy, *Computer Organization & Design, The Hardware/Software Interface*, 2nd ed. Morgan Kaufman Publishers Inc., 1998.

[14] V. Saljooghi, A. Bardizbanyan, M. Själander, and P. Larsson-Edefors, “Configurable RTL model for level-1 caches,” in *Proc. IEEE NORCHIP Conf.*, Nov. 2012.

[15] D. H. Stamatis, *Failure Mode and Effect Analysis: FMEA from Theory to Execution*. ASQ Press, 2003, ch. 11,12.

[16] J. Bickford, J. Hibbeler, D. Mueller, S. Peyer, and V. Kumar, “Optimizing Product Yield using Manufacturing Defect Weights,” in *Proc. Adv. Semiconductor Manufacturing Conf.*, May 2012, pp. 16–20.

[17] C. Hess, B. Stine, L. Weiland, T. Mitchell, M. Karnett, and K. Gardner, “Passive Multiplexer Test Structure for Fast and Accurate Contact and Via Fail-Rate Evaluation,” *IEEE Trans. on Semiconductor Manufacturing*, vol. 16, no. 2, pp. 259–265, May 2003.