Automating from Tapeout to Factory for a High-mix Fab

Christopher Frias

Qorvo, Inc., 500 W Renner Rd, Richardson, Texas 75080-1324
e-mail: chris.frias@qorvo.com Phone: (972) 994-3938

Tapeout, High-mix, Automation, Business Rules, Data-Driven

Abstract

This paper will address the work done to create a structured, flexible, and sustainable system that allows for the automation of new product introduction to a factory as well as the automation of factory technology configuration management.

INTRODUCTION

High-mix semiconductor fabrication facilities, those with a large number of supported technology processes and a large number of custom products, have many challenges. One of these is the need for the fabrication facility (fab) to support multiple products that are unique in many ways. Supporting products with unique attributes, requires the fab to create factory lines that are diligent in the core processing, as well as flexible enough to allow the processing variations required to meet the desired performance of each product.

Another challenge for a high-mix fab is the ability to support multiple technology lines (TL). For example, a fab may produce both GaAs and GaN wafers. It may also produce wafers of varying sizes such as 100mm and 150mm. Furthermore, each core TL can have different sub-lines and features. For example, a 100mm GaAs line may be able to produce transistors of 0.25um, 0.15um, and 0.50um in gate length. However, the 150mm GaAs line may only be able to produce transistors of 0.25um and 0.15um in length.

BUSINESS REQUIREMENTS

To support these requirements, we looked to create a system that would be used to automate the configuration of each new product's (NP) factory deliverables. These include, but are not limited to Photomasks, the Manufacturing Execution System (MES) flow, and factory recipe specifications.

These deliverables are governed by the business rules (BR) of each TL and the BRs of the fab overall. As it is common for the list of supported TLs and the BRs that govern them to vary frequently, we also required, to the extent possible, that the system should be maintainable by the subject matter experts (SME) and not solely an IT like group. One of the first steps taken was to transform our real world experiences into a model.

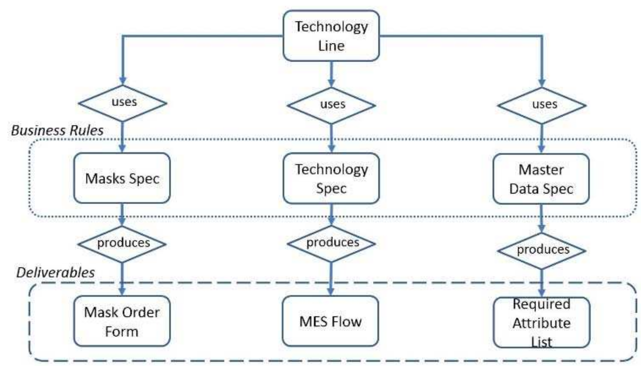

Figure 1. Technology Line Model

The model in Figure 1 is simple and straightforward, yet it makes it easy to see the entities involved, their relationships. A TL is governed by Specs. Specs provide guidance to produce Deliverables. The three Specs in Figure 1 represent BRs which need to be managed in a system that can change easily and rapidly. In addition, the Deliverables need to be in a format that can easily integrate with and feed other systems. Our answer is data-driven.

DATA-DRIVEN

Here, the term data-driven is used to describe that the outputs which are Deliverables, and inputs which are BRs, are both structured data stored in a relational database. This in contrast to each BR being its own custom program code, or manually executed by people. This approach is quite powerful because it allows us to add a new TL and its BRs simply by creating new records in a database and not by writing more code.

Data-driven is also used to highlight the fact that the Deliverables, the objects provided for each NP, are also structured data stored in a relational database and not static documents.

6b

The figure below describes the processing and variants for a fictitious semiconductor TL. It represents the BRs of a TL that would be detailed in the Technology Spec from Figure 1. A wafer being processed on this flow would proceed through the fab following the process blocks from left to right.

Transistor	Passivation	Thin Film	Frontside Interconnect	Backside Interconnect	Die Sort
Transistor Variant 1	Passivation Variant 1	Thin Film Resistor Variant 1	Frontside Variant 1	Backside Variant1	Die Sort Variant 1
Transistor Variant 2	Passivation Variant 2	Thin Film Resistor Variant 2	Frontside Variant 2		Die Sort Variant 2
Transistor Variant 3		Thin Film Resistor Variant 3	Frontside Variant 3		
Transistor Variant 4			Frontside Variant 4		

Figure 2. Process Technology Matrix

The rows represent a variant of each processing block. For example, the ‘Transistor Variant 1’ cell may represent the fab work instructions used to build a product that requires gate lengths of 0.50um. ‘Transistor Variant 2’ may represent the fab instructions used to build a product that requires a gate length of 0.15um.

Each new product (NP) introduced to the fab requires an analysis of each column comparing the fab’s BRs against one or more of the NP’s attributes. The result of this analysis includes the MES routing which is specific to the NP.

The analysis is done by combining each processing block with a logic expression. Figure 3 below shows the logic expression used for the Transistor processing block. To pick the correct variant, each processing block expression executes a series of ordered tests that operate against attributes of the NP.

If New Product Attribute 1 (NPA1) = “x” THEN use **Transistor Variant 1** processing block
Else If NPA2 = “y” and NPA3 < 0.5um THEN use **Transistor Variant 2** processing block
Else If NPA4 = “blue” and NPA3 >= 0.5um THEN use **Transistor Variant 3** processing block
Else Use **Transistor Variant 4** processing block

Figure 3. Process Block Logic Expression

This logic expression looks similar to a switch or case statement that would be used in a programming language, but to prevent programming each and every new BR, Figures 2 and 3 were modeled so that they could be turned into structured data.

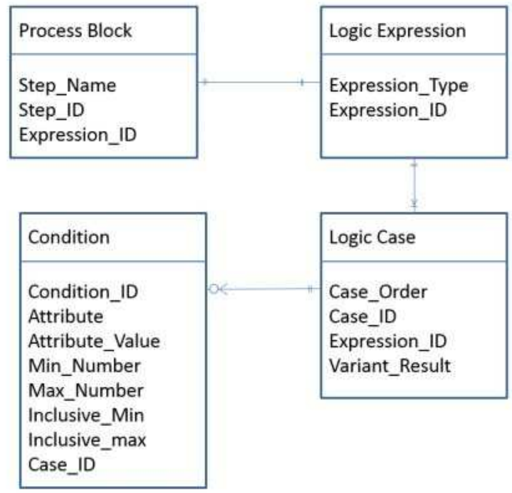

Figure 4. Database Entity Relationship Diagram

The image in Figure 4 is an Entity Relationship Diagram that represents the structured database tables used to store the BRs described in Figures 2 and 3. Each entity in Figure 4 maps to a database table shown in Figure 6.

The six columns in Figure 2 become six rows of data stored in the ‘Process Block’ table in Figure 5. In Figure 5, the ‘Logic Case’ and ‘Condition’ tables contain data representing the scenario in Figure 3. Because the Transistor processing block has four possible variants determined by the four cases in Figure 3, there are also four rows in the ‘Logic Case’ table in Figure 5. These four rows are associated to the first row in the ‘Process Block’ table through Foreign Key relationships. The ‘Condition’ table is used to determine which result to choose. Each case from the ‘Logic Case’ table has zero or more conditions. If the conditions of a case are met, then its variant will be returned.

In other words, the data tables of Figure 5 are set up for the ‘Transistor’ processing block to contain the four possible results of Transistor Variant 1, Transistor Variant 2, Transistor Variant 3, and Transistor Variant 4. The structured data in Figure 5 produces the exact same results as the expression in Figure 3. Its power is that it serves as a framework to create and store as many expressions as required. All without having to create or maintain a library of code for these expressions.

Process Block

STEP_NAME	STEP_ID	EXPRESSION_ID
Transistor	1	100
Passivation	2	101
Thin Film	3	102
Frontside Interconnect	4	103
Backside Interconnect	5	104
Die Sort	6	105

Logic Expression

EXPRESSION_ID	EXPRESSION_TYPE
100	VariantSelection
101	VariantSelection
102	VariantSelection
103	VariantSelection
104	VariantSelection
105	VariantSelection

Logic Case

CASE_ORDER	CASE_ID	EXPRESSION_ID	VARIANT_RESULT
1	1000	100	Transistor Variant 1
2	1001	100	Transistor Variant 2
3	1002	100	Transistor Variant 3
4	1003	100	Transistor Variant 4

Condition

CONDITION_ID	ATTRIBUTE	VALUE	CASE_ID	MIN	MAX	INCL_MIN	INCL_MAX
100	NPA1	x	1000	(null)	(null)	(null)	(null)
101	NPA2	y	1001	(null)	(null)	(null)	(null)
102	NPA3	(null)	1001	0.5	(null)	0	(null)
103	NPA4	blue	1002	(null)	(null)	(null)	(null)
104	NPA3	(null)	1002	(null)	0.5	(null)	1

Figure 5. Structured Database Tables

CODE ONCE

The first and third column in Figure 1, also contain their own set of entities and database tables similar to those in Figures 2 through 5. With the BRs stored as data, this allows for a set of core programs to be written and maintained. Because this system is working on data to produce data, the programming language used is PL/SQL.

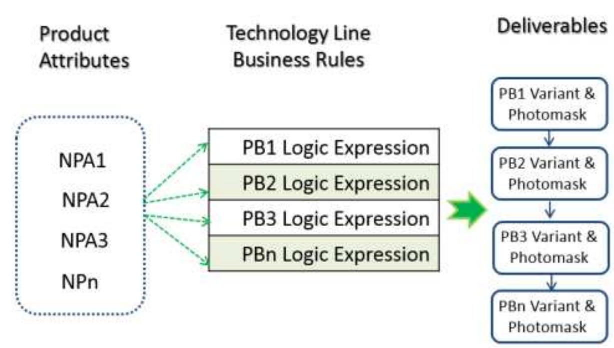

Figure 6. Abstracted Core Analysis Program

The image in Figure 6 depicts the PL/SQL programs that analyze a product's set of attributes against the BRs and then create the fab deliverables. Like Figure 6 depicts, the program is abstracted to work on n number of attributes, n number of BRs, and provide the desired deliverables over and over. The thing that changes from job to job, are the BRs provided to it. To create Deliverables for a GaAs product, the core program operates on GaAs BRs. To create Deliverables for a GaN product, the core program operates on GaN BRs.

TEMPLATE CENTER

It can be common for high-mix factories to have BRs change often. When a rule changes, it requires a change to the database. Ultimately, records are either inserted, updated, or deleted in a controlled manner. Just like in the case of the Core Analysis Program, these operations are executed by more PL/SQL code. This code allows for things like adding new Process Blocks, Removing Logic Cases, Modifying Conditions, etc. Basically, all the change verbs for BRs have a corresponding PL/SQL program.

Figure 7. Template Center Web Application

Most SMEs will not have a deep enough understanding of databases to manually execute PL/SQL in an effort to manage BRs. In contrast, they are able to use web applications. Therefore, accompanying the work presented so far is an application named Template Center. It is an interface between users and the underlying PL/SQL programs that operate on the database. Through the GUIs presented in this web application, SMEs can intuitively add or change BRs as needed. The example in Figure 7 shows how the app was used to replace the fourth case with a result of Transistor Variant 5 instead of Transistor Variant 4. It also shows how the cases were modified to operate on a different set conditions.

Changes similar to the one shown here are typical BR changes that a TL may experience. And with a tool like Template Center, the changes can deployed in minutes or hours as opposed to hours, days, or even weeks.

WORKFLOW

Another key component to this system is workflow automation. A Business Process Management Suite (BPMS) is used for this. The software allowed us to turn the workflow from Figure 8 into a business application. In the software, an instance of the Figure 8 workflow is generated for each NP. The software then manages and tracks each instance until it is completed. Our NP introduction process follows a structured and ordered flow, making it an ideal match for a BPMS.

Although, creating this application is non-trivial, and requires an intimate knowledge of the actual workflow process, sub-processes, roles involved, and systems. It also requires all exceptions, or at least all points in the process where an exception can happen, be understood. Without this, in production, an instance of the flow will reach an unhandled exception and will be not able to continue, or will continue with an error.

However, once understood, and with a capable BPMS, a custom business application that orchestrates human tasks, email communication, database integration, approvals, and integration to corporate systems such as an ERP can be developed.

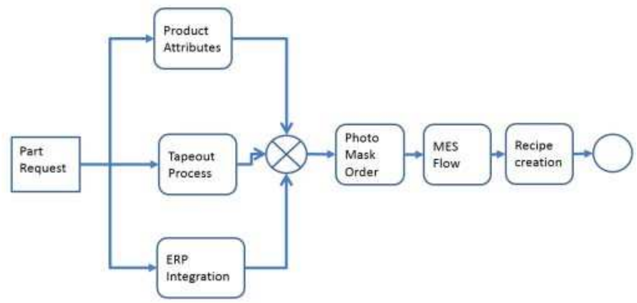

Figure 8. Custom Business Workflow

AUTOMATED ATTRIBUTES

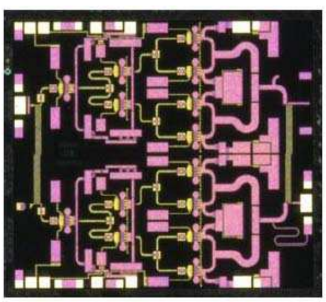

Figure 9. Example Integrated Circuit [1]

With the BR and Workflow systems serving as a foundation, it is possible for even more automation to be enabled. For example, integrated circuits like the one in Figure 9 can require minor customized fab processing based on attributes extracted from the Computer-aided design (CAD) file. These CAD files contain the drawings that are used to produce Photomasks, which in turn are used by the fab to produce wafers of integrated circuits.

For fab processes such as depositions, etches, and plating that vary with affected Photomask area, instead of having engineers or operators determine the exact process recipe, this can be integrated into the BR and Workflow system already described.

For example, each product, like the one in Figure 9 requires a custom Plating Parameter to properly form the plated transmission lines. In the Recipe creation sub-process from Figure 8, programs are executed that analyze the CAD file, calculate the Plating Parameter, and store the result as a Product Attribute as structured data.

CONCLUSIONS

The paper presented here represents work that has progressed from the desire to minimize human errors and reduce cycle time of new product introduction to the factory. The journey was not and is not without its challenges.

The first generation of this system was developed almost 100% within a BPMS tool. This includes a workflow similar to what is in Figure 8, but more consequently the BRs are also managed using the BPMS. The result of this approach is that any update to the BRs can only be done by a trained developer of the BPMS tool and not a SME. The lessons learned in this first generation were highly influential in the design of the work presented here.

Factories that do not have a high-mix of technologies and custom products, may not require such a system presented here. It is possible for their technology management and product configuration process to be much narrower in scope that they can be successfully managed simply using spreadsheets, email, documents, and other methods.

However, for fabs that do support a high-mix of lines or produce a high number of customized products, a system like the one presented here, is almost essential. Furthermore, if a company contains many fabs, once this system is deployed at one fab, it is scalable to others.

ACKNOWLEDGEMENTS

The author would like to thank the people for their help with this paper: Howard Witham, Donna Mortensen, Scott Hawkins, Maimoona Iqbal, Dan Madoni, John McConnell, Jeffrey John, and Omar Sulaiman.

REFERENCES

[1] C. Campbell, Y Liu, M. Kao and S. Nayak, *High Efficiency Ka-Band Gallium Nitride Power Amplifier MMICs,* 2013 IEE COMCAS

ACRONYMS

fab: Fabrication Facility
NP: New Product
TL: Technology Line
BR: Business Rule
SME: Subject Matter Expert
MES: Manufacturing Execution System
ERP: Enterprise Resource Planning
BPMS: Business Process Management Suite