

Mitigating System Design Risk With VHDL

Michael K. Lindsey
Motorola Inc., Government and System Technology Group
8201 E. McDowell Road, MD H1102
Scottsdale, Arizona 85252
Michael_Lindsey-P25715@email.mot.com

Abstract

This paper discusses the use of the VHSIC Hardware Description Language (VHDL) to design three separate Application Specific Integrated Circuits (ASICs) for a single Multi-Chip Module (MCM). VHDL was selected in order to meet a customer requirement for deliverable VHDL models of ASIC devices. Synthesis was used to obtain gate level implementations from VHDL models. Originally, the ASIC designs were targeted for a field programmable gate array (FPGA) technology. Due to changes in the system performance requirements of the MCM, the FPGA technology became impractical. The technology independence of VHDL allowed the ASIC technology to be switched mid-course in the ASIC design process without significant impact on the development program schedule. One ASIC is being produced using a standard cell technology and two are being produced using a masked gate array technology. Three FPGAs of limited functionality were produced for rapid system prototyping.

Section 1. Introduction

Over the past several years, Motorola GSTG has been using VHDL on a number of projects. The majority of these projects involve developing complex embedded electronic systems. VHDL is currently being used at GSTG for system modeling,

synthesis, mixed-level simulation, and as executable specifications that are exchanged with design teams from other companies. Although Motorola GSTG is employing state of the art VHDL synthesis and simulation technology, the most compelling reason to do so is simply to fulfill customer requirements that specify the use of VHDL.

For practical reasons, program management and many engineers are reluctant to utilize VHDL for purposes other than meeting customer requirements. System modeling with VHDL has proven to be beneficial, but it is time consuming and of limited use in developing embedded systems that contain components in addition to digital hardware. VHDL synthesis has yet to demonstrate that it is the breakthrough technology that will cause ASIC designers to completely abandon schematic design capture. It is easy to understand these reasons if one assumes the pragmatic viewpoint of an engineer whose job is to develop an embedded system, not to experiment with VHDL.

In the course of developing an embedded system, ASIC development many times represents only a fraction of the development effort. A typical embedded system usually includes software components, analog components, standard logic components, such as microprocessors, and of course ASICs. In today's systems, software components generally represent the majority of the design effort. For example, the system which is the basis of this paper has approximately 75% of its

functionality implemented by software components. Despite the fact that VHDL has found widespread application in ASIC development, the use of VHDL in software development and analog electronics has not progressed much beyond the experimental stage to date. VHDL may truly provide an advantage in ASIC development, but it is not always clear to a pragmatic engineer how big that advantage is and whether it can be leveraged in a system where ASICs represent less than quarter of the total solution.

To compound a pragmatist's reluctance to use VHDL, past experiences have shown that VHDL is not yet a panacea for the afflictions of the established ASIC design environment. In fact, using VHDL on prior ASIC designs at GSTG has had some less than satisfactory results. The tools were not tightly integrated and sometimes didn't perform as advertised. New design methodologies had to be created by project engineers, which detracted from their design tasks. And synthesis was not as foolproof as first anticipated, or was not feasible for some designs. The bottom line is that committing a project to VHDL based ASIC design could present more problems than it could solve.

System level modeling is often cited as a useful application of VHDL. Indeed, system level modeling using VHDL has been performed on many projects at GSTG. However, from a pragmatic view point, graphical based system design tools are a better choice for system level modeling. Generally, it takes considerably less time and effort to create a graphical system model that is equivalent to a textual VHDL model.

Some may attribute the reluctance to adopt VHDL to short sighted program management or lack of vision by engineering staff. This simply is not the reason why VHDL has not found mainstream acceptance in the embedded system design environment. Rather, it has been the inability of VHDL marketers to

identify a must-have, compelling application of VHDL that has been responsible for the failure of VHDL and its associated tools to displace competing technologies from the established design environment.

This paper asserts that the compelling reason to use VHDL in today's embedded system development is that the technology independence of VHDL allows ASIC technologies to be selected later in the design cycle. The flexibility to defer technology choices provides significant risk reduction to project schedules and budgets. The advantage of this flexibility was recently demonstrated on the a project at Motorola GSTG.

Although technology independence was one of the original tenets of the DOD effort that conceived VHDL in the early 80's, it has not been practical for system designers to gain advantage from this feature until recently. Over the last couple of years, the availability of synthesis libraries has grown and VHDL simulators and synthesizers have matured.

The remainder of this paper contrasts the potential impact on schedule of changing technologies in a traditional embedded system design, one that relies on technology dependent schematic capture for ASIC design, and the impact of changing technologies during a VHDL based design. The basis for this discussion is the experience that Motorola GSTG had during the development of an MCM, referred to as the WJ System, in which traditional ASIC schematic capture was abandon in favor of using VHDL due to required changes in ASIC technology.

Section 2. The Schedule Impact

Figure 1 shows an ideal schedule for a module being developed in which an ASIC is in the critical path of the design schedule.

Typically, an embedded system contains several modules and the figures shown in this section would represent only a small portion of the entire system development schedule.

In the ideal case, module level design begins when the system level design is entirely complete, resulting in a set of module specifications that are error free, implementable, and not subject to change. At some point during module design an ASIC specification is completed, which is also error free, implementable, and not subject to change. The ASIC technology and vendor are also selected before starting the ASIC design. What is not shown in the diagram is that other module design activities, such as software development, occur parallel to the ASIC development. However, in the ideal case these parallel activities do not affect the specification of the ASIC. Upon completion of the ASIC,

module level integration and test begins, usually with the hardware components being integrated first, and then the hardware and software being integrated together. Upon completion of the module, system integration and test begins, which concludes when the entire system is fully tested and ready for delivery to the customer. Unfortunately, most human endeavors are less than ideal in actual execution.

The reality of the design environment is that specifications change, they have errors, and sometimes they simply specify the impossible. Add to this the fact that design activities that impact others, such as system design, are often executed in parallel, it is possible to achieve a worst case schedule such as the one shown in Figure 2. The problem is that this schedule is almost never what the original project planners had anticipated.

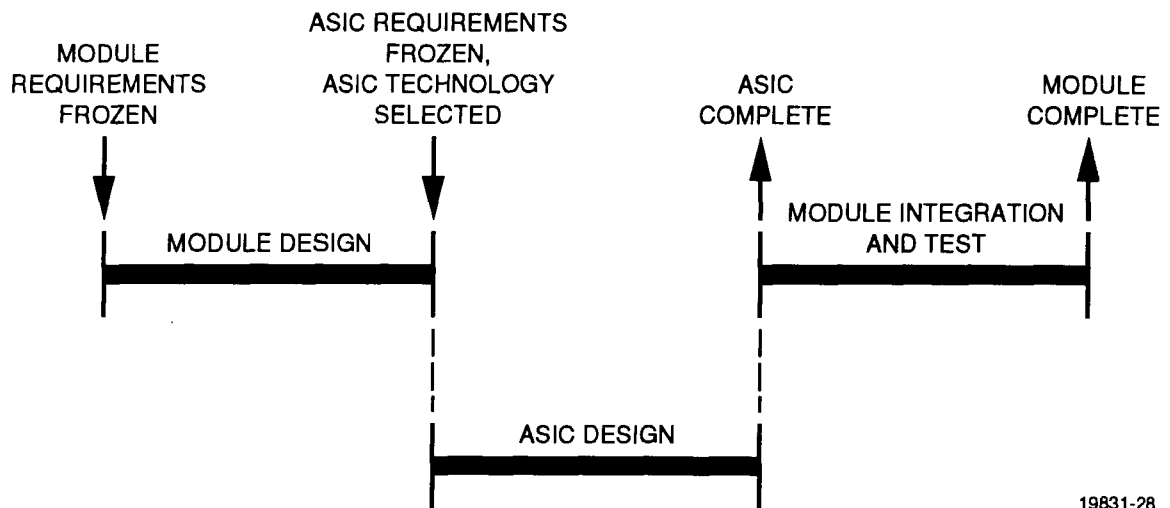


Figure 1. Ideal Module Development Schedule

Figure 2 shows a traditional non-VHDL project schedule in which module and ASIC designs begin before their specifications are solidified. Furthermore, the system requirements have not settled before either the module or ASIC designs start. ASIC

design begins when the first revision of its specification is complete, and usually the project has selected a technology before this point.

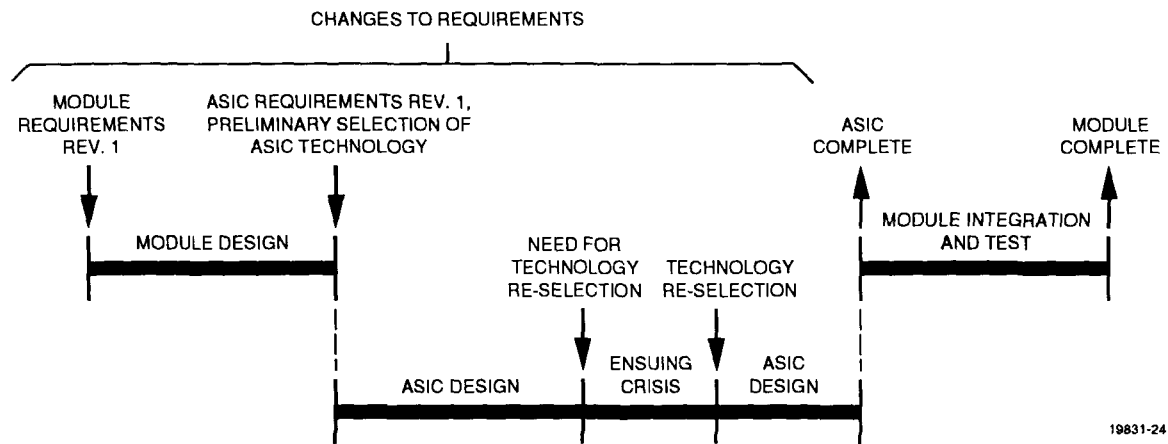


Figure 2. Traditional Module Development Schedule

In a non-VHDL ASIC design, it is necessary to have resources committed to the selected technology. This entails having the cell libraries, simulators, and other technology specific tools available for the designers. The process of selecting a technology and setting up the design tools generally takes weeks to complete. Technology selection involves not only technical issues, such as speed and gate count, but also vendor screening. Vendors are normally evaluated on non technical issues, such as cost and past performance.

Figure 2 also shows what can happen to a project schedule when changes to the ASIC specification exceed the capabilities of the selected technology during the ASIC design. The changes could cause the gate count to grow too much or the need for a faster technology. The "ensuing crisis" of Figure 2 is the period of time during which the project scrambles to pick a new technology. During this time, the selection process repeats and the ASIC design halts. If the ASIC design is in the critical path of the project schedule, the time lost during the "ensuing crisis" will negatively impact the customer delivery date.

During the WJ Project at GSTG, the schedule depicted by Figure 2 was avoided by using VHDL, rather than traditional schematic capture, to perform ASIC design. The effect of using VHDL is shown in Figure 3. Although system requirements were in flux and ASIC specifications were not finalized, ASIC design proceeded using VHDL.

The original technology selected for the ASICs was a Field Programmable Gate Array (FPGA). However, early on in the ASIC designs, engineers determined that the FPGA technology was not suitable for the delays through the FPGA were too great to meet system throughput requirements. Furthermore, the designs were large and each one required multiple FPGAs; the FPGAs took up too much area on the MCM.

A 1.0 micron standard cell technology and a 1.0 micron masked gate array technology were selected by the project to replace the FPGAs. Although technology re-selection did occur during the ASIC design, the impact to the schedule was small due to the fact that ASIC functional design continued during the re-selection process.

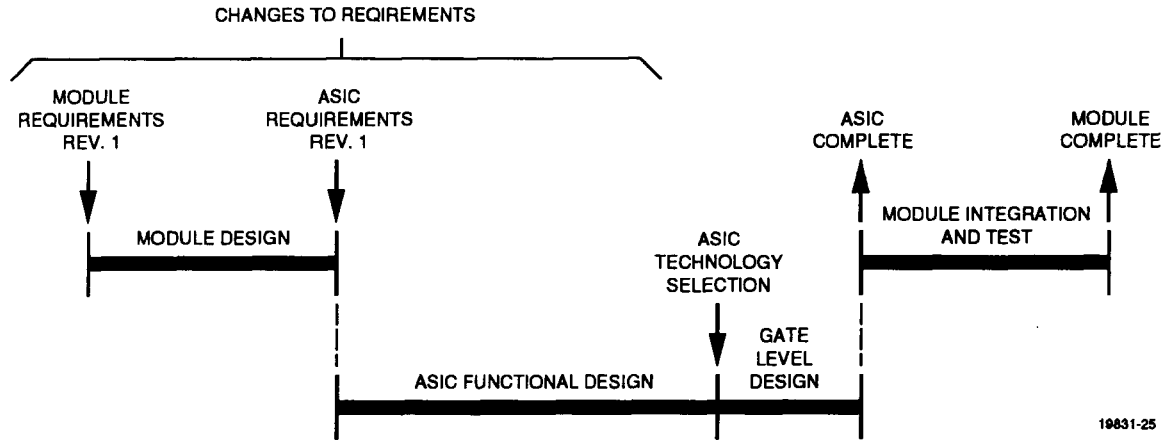


Figure 3. Module Development Schedule Using VHDL

Section 3. The WJ System

The WJ System is shown in Figure 4. It is typical of the embedded systems developed at Motorola GSTG. It is a multiple processor system used for complex mathematical transforms in a real time communication systems. The system

contains three different ASIC designs, with one being used in both the Blanco and Rojo Processors. The entire system operates at 20 MHz and the ASIC sizes vary from 35,000 to 102,000 gates. The Babel and Llave Processor ASICs use the 1.0 micron gate array technology, while the Blanco and Rojo ASICs use the 1.0 micron standard cell technology.

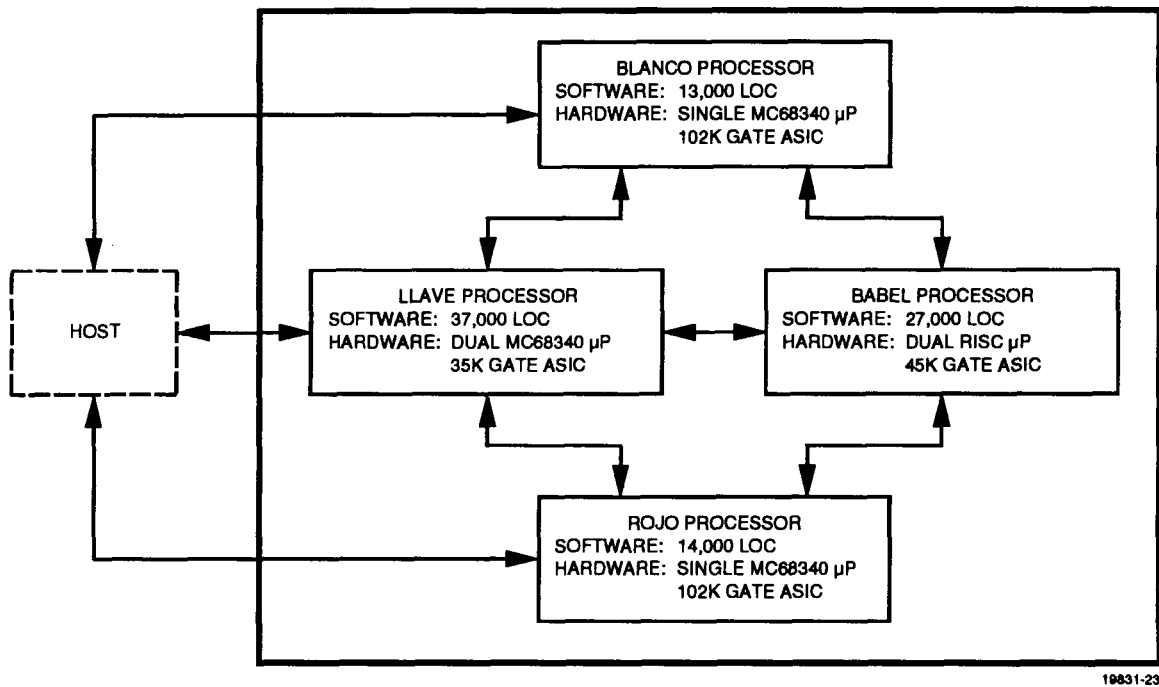


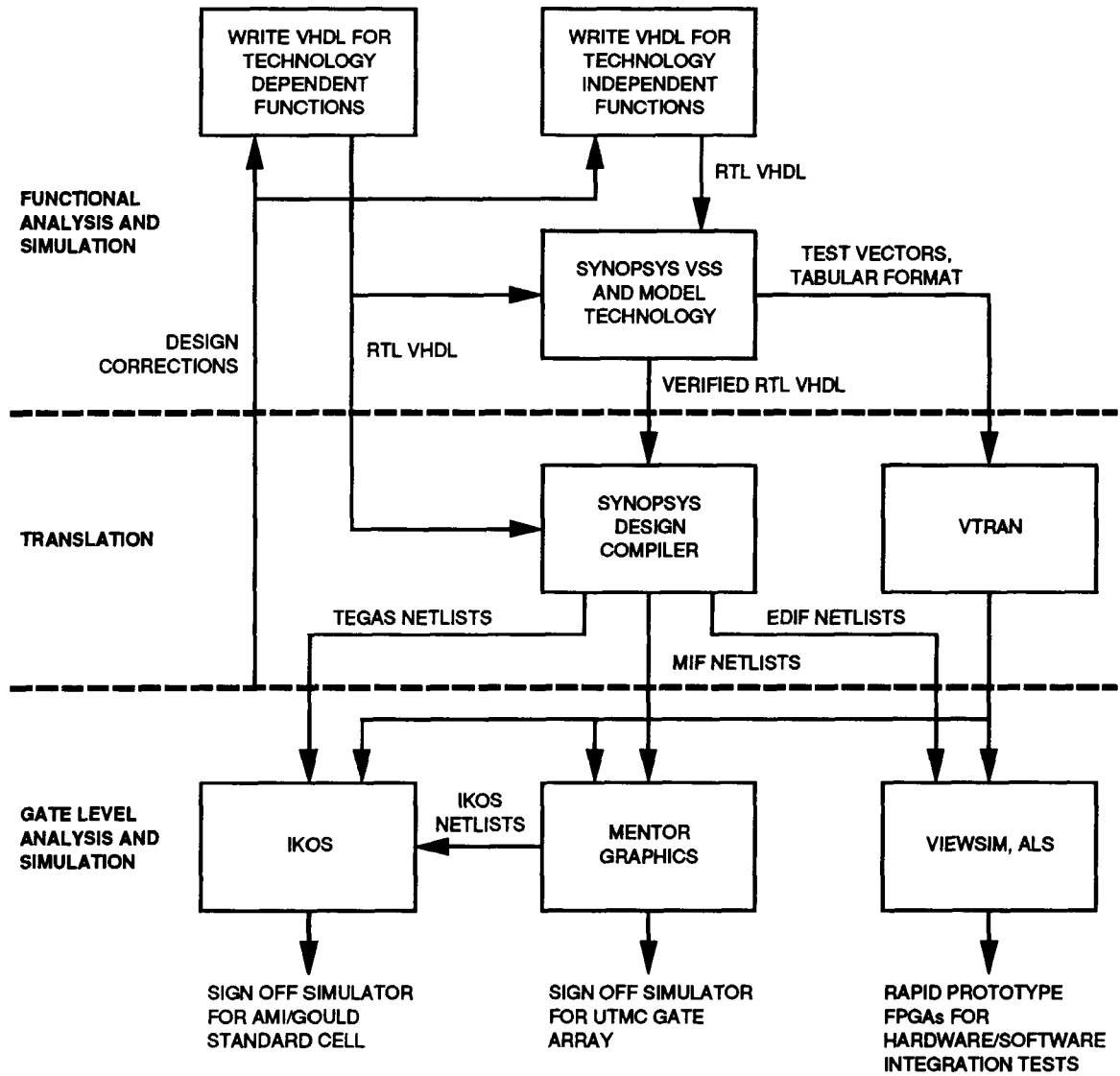
Figure 4. The WJ System

Figure 5 shows the methodology used for the ASIC designs. VHDL is used to capture both technology dependent and independent functions of the designs. The technology dependent functions are things such as logic used for manufacturing tests, nand trees, and I/O pad instantiation. At least 70% of the designers' time was spent performing functional VHDL design.

Figure 6 shows the format of the VHDL test benches used to test functional VHDL. The test vectors generated during functional simulation were translated using VTRAN to the targeted gate-level simulator. This was accomplished by using test bench monitors that generated ASCII tabular stimulus files.

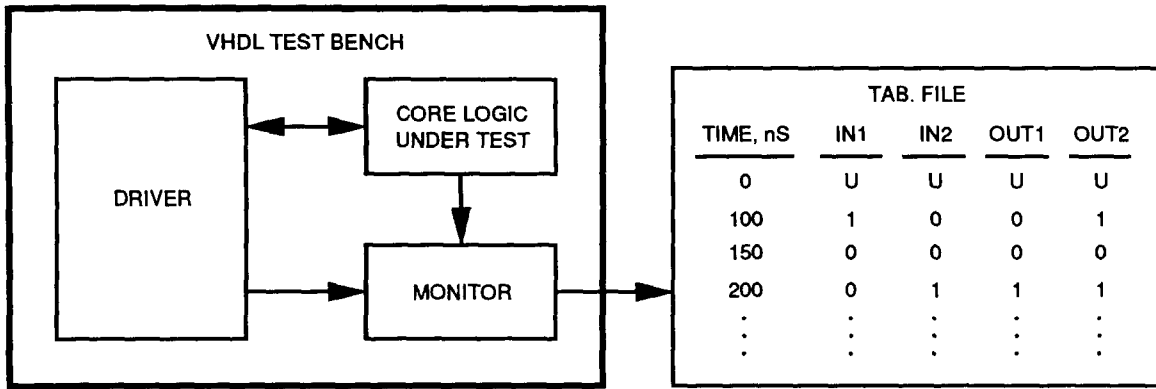
Section 4. Summary

VHDL has many promising applications; however, the compelling reason to use VHDL in embedded system development is to provide flexibility in choosing ASIC technologies. Development projects at Motorola GSTG are using VHDL in a number of different applications. Most of these applications are customer specified and have not demonstrated a compelling, non-contractual reason to use VHDL. The experience of the WJ Project at Motorola GSTG has demonstrated how invaluable VHDL's technology independence is during the design of a complex embedded system.



10831-27

Figure 5. ASIC Design Methodology



19831-26

Figure 6. VHDL Test Benches