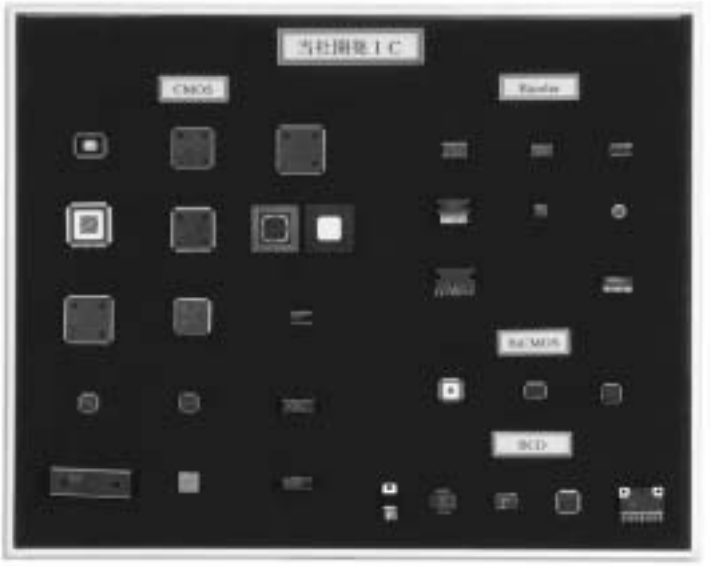
Creation of an LSI Design Environment that Accommodates Short-Range Development

Wataru Ihara
Fumitaka Fushimi
Takashi Itoh
Keiichi Maeda



Abstract

As process miniaturization advances, the large-scale integration (LSI) of circuits continues to proceed. Moreover, market demand is increasing the importance of early introduction, and further shortening of the design period is being sought.

Designing large-scale integrated circuits efficiently and with high quality requires the use of many tools. It is also important to raise design productivity without increasing costs. In order to quickly commercialize and introduce LSI circuits to the market, more efficient LSI evaluation is also indispensable.

The main intent of this report is to introduce the features of design environments created by our company to realize short-range development.

1 Introduction

Fujitsu Ten's IC development began in 1973 with motoronics for seat belt attachment-ensuring systems. At the time, these were small custom ICs, with not even a hundred transistors. Afterward, development proceeded, centering on ICs for emission control, which was the core product. In 1988 a section specializing in IC design was created, and a design environment for bipolar LSI circuit development was established.

Meanwhile, development in the field of audio-visual communication (AVC) began in 1980 with tuner microchips. With the DSP chip in 1988, wire width became 1.2 μm; and even with an external D/A converter, the chip size was a large 10×12 mm.

Later, the motoronics and AVC IC development divisions became integrated and the present-day LSI Development Department was formed.

During that time, miniaturization in process technology advanced by a factor of four in three years according to Moore's law; and now design at the 0.1 μm level is right before us. Of course, circuit scale has increased. In fact, development of chips exceeding ten million gates is no longer a rarity in the world.

However, despite the increased scale of LSI circuits, consumer-driven developments are important in the market; thus, it has become more and more important to shorten the development period.

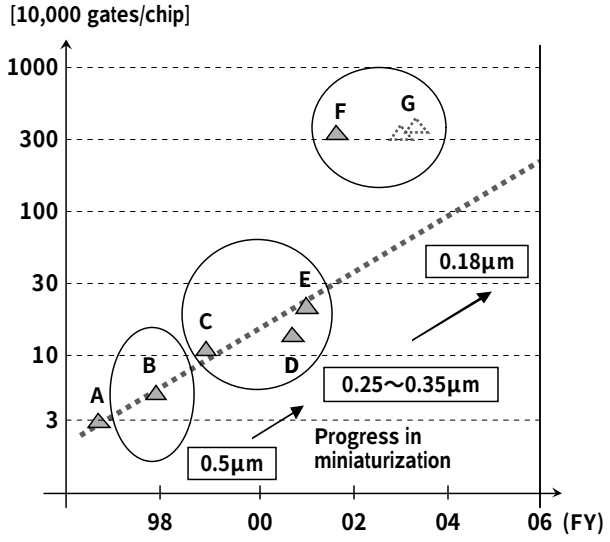


Fig.1 Advances in Fujitsu Ten's LSI (digital)circuit developments

This report will discuss the effect that changes in digital design, analog design, and digital-analog mixed design have had on the speed-up of development, looking from the perspective of the LSI circuit design environment. The report will also introduce examples of actions taken to improve design efficiency, including benchmarks with other companies.

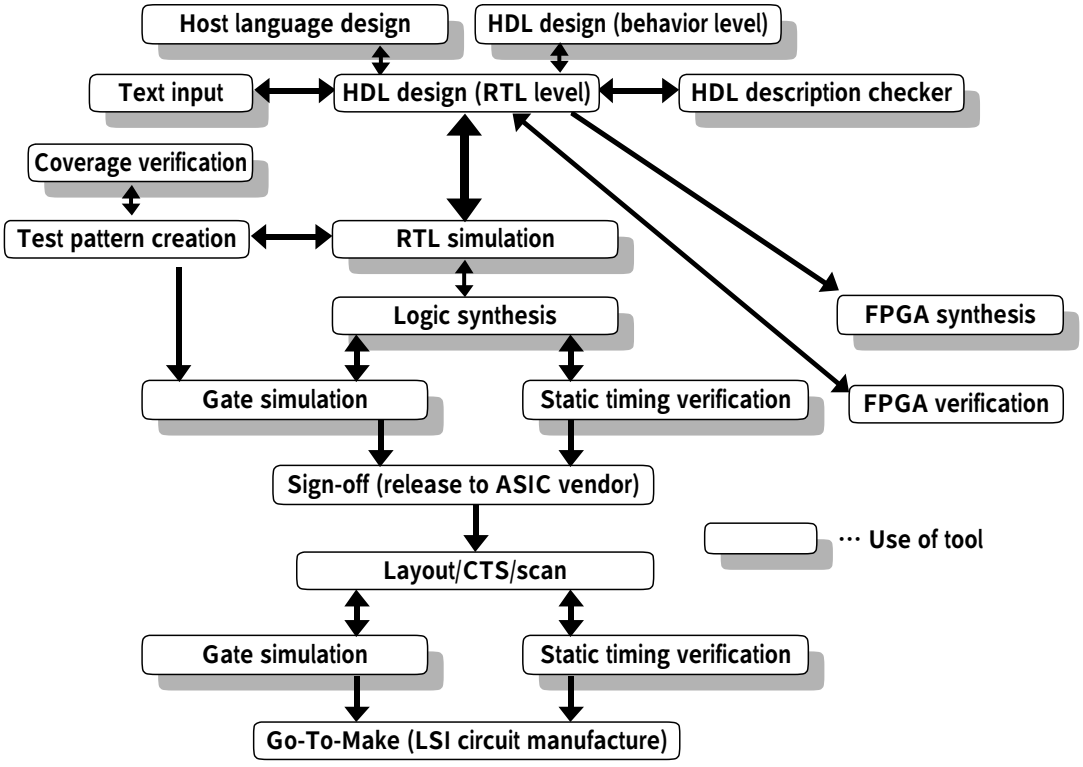


Fig.2 Digital LSI design process

2.1 Digital LSI circuit design environment

Rapid progress has been made in digital large-scale integration. And though the design scale has increased accordingly, there is demand for a design period that is the same or shorter than in the past.

To meet such demand, progress is being made in design techniques and tools that enable short-range development.

At the present time, Fujitsu Ten is creating an environment such as that shown in Figure 2.

1) Design using hardware description language

Digital design makes use of a design technique that employs Hardware Description Language (HDL).

With Register Transfer Level (RTL) HDL, circuit operations are expressed by a description that includes a clock concept. Compared to a classical circuit diagram design (gate circuit), the degree of abstractness is much higher and simulation is faster. Thus, design productivity can be improved by a factor of ten or more.

We use Verilog-HDL, which has been standardized (IEEE 1364) since 1995.

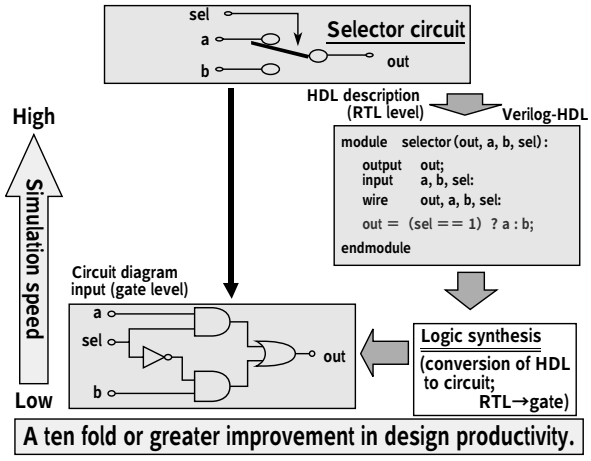


Fig.3 Circuit design using HDL

2) Improvement in quality via description checker

By reflecting our design know-how and applying a description checker for the designed HDL, we are able to check grammar, reduce description errors, and detect descriptions that are inappropriate for subsequent processes, including logic synthesis. This makes it possible to detect and correct bugs at an early stage, and improve design efficiency and circuit quality.

3) Improvement in circuit quality via improvement in test pattern accuracy

During simulation, the important thing is not only to speed up the simulation itself but to be able to verify all functions in a short time.

By using a coverage tool, We can find out the degree to which descriptions and functions are verifiable by test patterns, and can efficiently verify using the minimum required test patterns.

Furthermore, by implementing failure simulations for test patterns used to detect failures at the time of LSI circuit shipments, we can perform checks at a high failure detection rate and prevent defective products from making their way to the market.

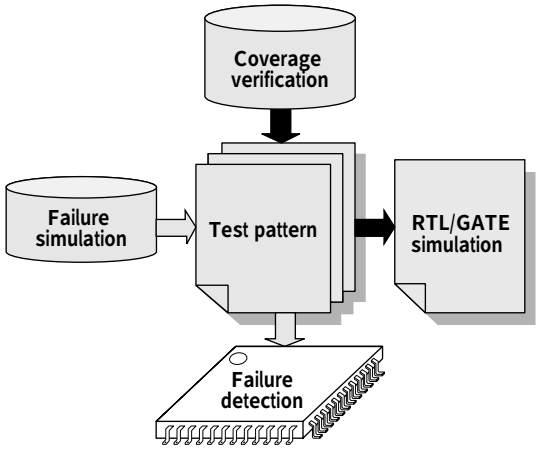


Fig.4 Improvement of test pattern accuracy

4) Improvement in design efficiency via static verification

Following logic synthesis, in addition to logic verification, a gate circuit needs to be checked to see if it operates at the required operating frequency even when there are temperature changes and/or power supply voltage fluctuations (timing verification).

Compared to RTL simulation, however, gate simulation is much slower, from two to several dozen times slower. Furthermore, the best case (low temperature and high voltage) and worst case (high temperature and low voltage) must be verified for the same test pattern, resulting in extremely poor efficiency.

The process therefore does not utilize dynamic verification, whereby the circuit is operated and verified with a test pattern. Rather, a static verification tool is used to calculate the delay between each flip-flop in the circuit and determine whether it meets the standard time (setup/hold time).

Static verification is at least a thousand times faster than simulation. Since all circuit delays are measured mechanically without using test patterns, 100% of the circuits can be measured.

5) Parallel verification via FPGA

Verification that only utilizes simulation requires a vast amount of time. And since a large volume of data is involved, there is concern that some malfunctions may not be detected. For this reason, verification is implemented using an actual board that employs a Field Programmable Gate Array (FPGA), which makes it easy to replace a circuit.

Using an FPGA makes it possible to perform logic/function checks at speeds that are a thousand to several hundred thousand times faster than simulation. Moreover, since it is possible to check the operation of the entire system, malfunctions are easy to detect.

But because the operating frequency of the FPGA is slow and there are limitations to the circuit scale, FPGA design often takes time.

Thus, by using a logic synthesis tool that considers a wiring layout especially for the FPGA, we are able to design quickly through optimal operating frequency and circuit scale.

In this way, through digital LSI circuit design, we have created an environment that can develop high-quality LSI circuits in a short time by using a number of tools and various verification techniques.

2.2 Analog LSI circuit design environment

Although advances have also been made in analog large-scale integration, there have not been large changes in the design techniques themselves. A classical technique is generally employed; that is, circuit diagrams are created and the circuits are checked by a Simulation Program with Integrated Circuit Emphasis (SPICE)-type simulator.

We first introduced a SPICE in 1987.

Since that time, we have strived to shorten design periods by introducing high-speed simulators and simulators that operate on a personal computer.

In order to prevent mistakes, line widths and spacing are automatically checked by Design Rule Check (DRC) with the mask layout, and circuit diagrams and layouts are automatically compared by employing Layout vs. Schematic (LVS).

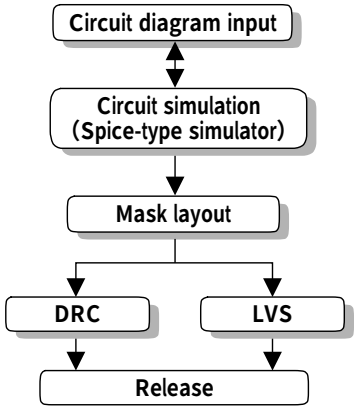


Fig.5 Analog LSI design process

2.2.1 Effective use of licenses

Because of techniques such as that previously described, it is necessary with analog design to increase the number of design personnel and licenses as the design scale increases. But since the cost is high, such steps are not easy to implement.

For this reason, attention has been given to the necessity of licenses in design work. Licenses used during design can be broadly divided into two types: licenses for circuit diagram editing and licenses for the simulators themselves.

A simulator license is required for as long as a simulation is being performed; thus, the only possible action is to increase the number of licenses.

A circuit diagram editing license, however, is required only during the work period. Currently, once a license is obtained, it can be held until a circuit diagram is closed.

Thus, steps have been taken to control costs through the creation of a program that releases licenses when editing is not being performed, enabling multiple designers to simultaneously use licenses through time-sharing.

2.2.2 Use of AHDL

Similar to that of digital HDL design, an analog design technique that employs Analog Hardware Description Language (AHDL) has been proposed.

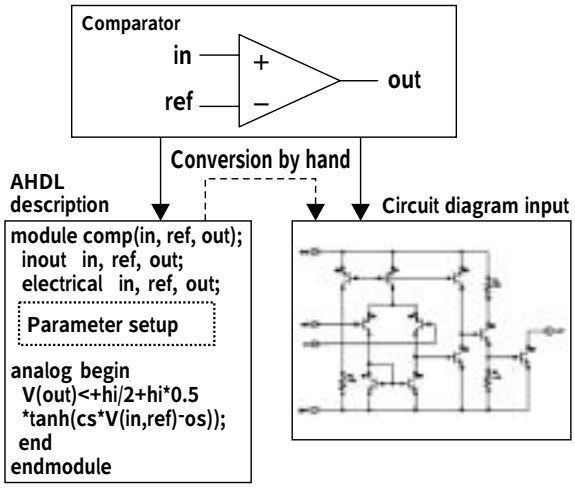


Fig.6 Circuit design using AHDL

Circuits described by AHDL can be simulated at speeds of ten to one hundred or more times faster than circuits consisting of transistors that are simulated by SPICE. (Also depends on the model level.)

Moreover, it is suitable for top-down design in which everything is verified starting from the initial design stage. Great results can be expected, including optimization of design and early detection of interface and other nonconformities.

At the present time, we are exploring the possibility of creating ICs by converting ICs to AHDL in the planning stage and checking their operation, and using this information as decision-supporting material for product development.

Currently, a tool that automatically converts from AHDL to the transistor level has not yet reached the practical application level. This problem is expected to be resolved in the future as market trends are ascertained.

2.3 Creation of digital-analog mixed design environment

Adoption of digital-analog mixed ICs began in earnest in 1996 with linear solenoid ICs. The design technique at the time consisted of the digital part and analog part being designed separately. After each was completed, they were connected.

There are a couple of ways to check the connected circuit: Replace all digital circuits with transistors and then check together with the analog circuits, using a SPICE simulator; or using separate simulators, connect both simulators using a communications protocol and check.

However, these methods often require a vast amount of time for simulation, and operability is poor. Consequently, they were not utilized during actual design at the time.

As a result, simple connection errors in the digital-analog connections occurred, as well as malfunctions in the interfaces; thus, revisions were inevitable.

For this reason a mixed simulator was introduced, enabling simulation under conditions in which digital and analog are mixed.

Since, with a single simulator, this simulator makes it possible to simulate circuits in which various levels (such as transistors, analog language descriptions, digital language descriptions, and system language descriptions) are mixed, it can perform checks faster than conventional systems.

With our digital-analog mixed ICs, the analog part often makes up the majority of the circuit. In such cases the simulation of the transistor level makes up the major portion; thus, even a mixed simulator cannot be expected to produce a large reduction in time.

For this reason, we tried to reduce simulation time by converting unrelated blocks to AHDL and raising the degree of abstractness, based on the contents of the verification. As a result, time was reduced to approximately one third of that required to verify at all transistor levels. Moreover, the absence of connection errors and interface malfunctions can be confirmed, making it possible to shift to IC manufacture without cause for concern.

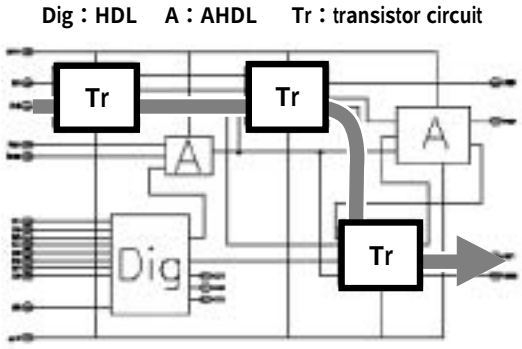


Fig.7 Digital-analog mixed circuit

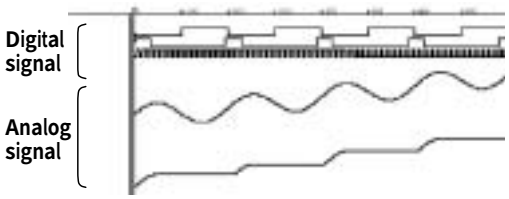


Fig.8 Simulation results

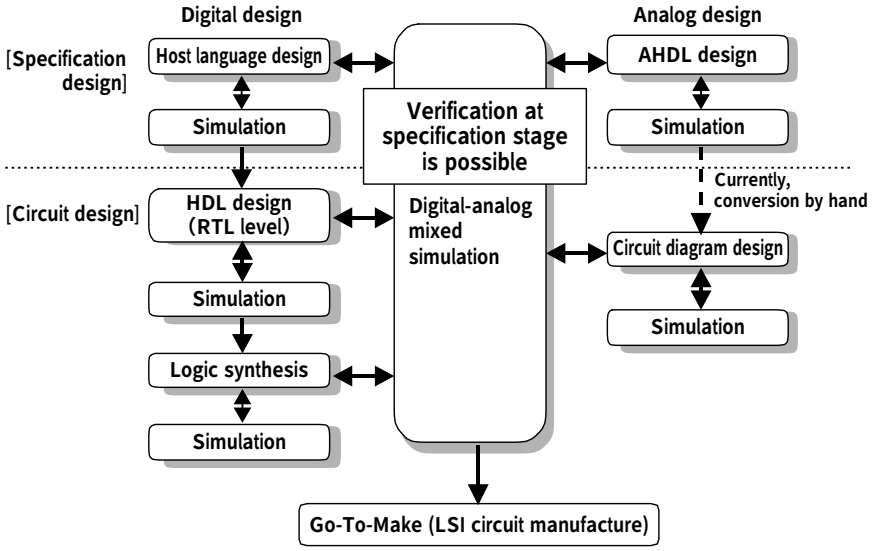


Fig.9 Top-down design process with digital-analog mixed IC

3 Actions for promoting improved design efficiency

As mentioned in the preceding chapter, we created an environment in which effective tools can be utilized to reduce the design period. However, increases in design scale are continuing to advance, making further measures necessary. Thus, in order to create an effective design environment while controlling costs, we are promoting the following three activities:

①Effective use of LSI circuit design equipment
②Reuse of intellectual property (design assets)
③Promotion of automatic evaluations

3.1 Improved design efficiency via LSF

Load Sharing Facility (LSF) refers to software that provides centralized control of the computers on a network, operates tools with optimal computers (load sharing), and controls the operating conditions. We have been using LSF since introducing it in 1999.

3.1.1 Advantages of LSF

1) Effective use of hardware

The performance of computers used for design (engineering workstations) improves each year. Thus, there is variation in the performance of our company's Engineering Workstations (EWSs), based on their time of introduction.

When designers perform simulations using an EWS that they have personally selected, they tend to use a high-performance EWS. This creates a problem, however, in that the processing capacity is reduced to half or less and verification can no longer be completed in a short time, even though a high-speed EWS has been used.

Therefore, using an LSF makes it possible to execute automatically "with the lightest possible load and with an EWS that performs verifications most efficiently." This has made it possible to minimize execution time and maximize the use of EWS assets.

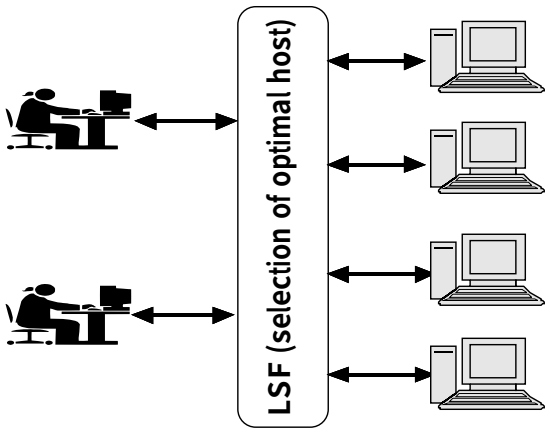


Fig.10 Effective utilization of LSF-utilizing EWS

2) Effective use of design tools

Many expensive tools are available for use in LSI circuit design. Thus, it is important to find a tool that meets the minimum requirements for design and produces the greatest effect.

Because an LSF can easily monitor the frequency of use of each design tool, it can be used as an indicator of license increases and decreases.

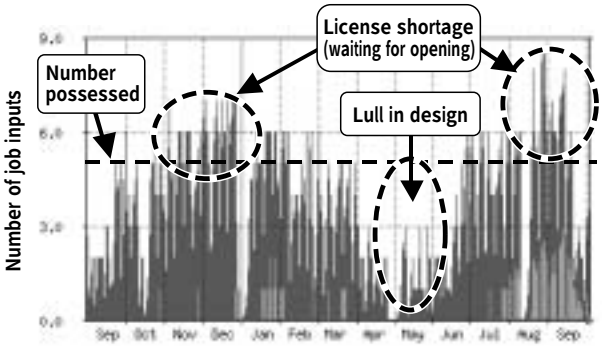


Fig.11 Monitoring of tool use frequency

We use such LSFs and have incorporated features such as the following:

①For tools with CPU loading, only one job will run per CPU.
②For tools that do not require a constant CPU load and whose main function is to provide displays such as circuit displays and waveform displays, execution centers on terminals having a somewhat inferior CPU capacity.
③Since the directory structure is made exactly the same for different operating systems, jobs can be executed without concern about OS differences.

As a result, even though there are slight differences according to the tool, execution time can be reduced by approximately 20% to 40%.

3.2 Reuse of Intellectual Property

3.2.1 Registration as IP

Because large-scale circuits were designed from scratch, it was difficult to meet short delivery deadlines.

As a means to resolve this problem, there is a growingly popular technique that reuses past Intellectual Property (IP, i.e. design assets) during circuit design.

Using intellectual property makes it possible to reduce the number of blocks that must be newly designed. Furthermore, if the process is the same, the operation of the reused portions is guaranteed, making verification unnecessary. As a result, it is possible to develop even large-scale LSI circuits in a short period.

During digital design, our company designates communications and processing modules as intellectual property and operational modules. During analog design, cir-

cuits such as op-amps and converters are made into intellectual property.

3.2.2 Actions for improving IP utilization

Because such intellectual property was arranged by individual project, all of this intellectual property could not be utilized efficiently.

Therefore, a database for intellectual property was created so that all intellectual property could be checked on the World Wide Web. Rules for registered intellectual property were established and levels were standardized. Also, steps were taken to register specifications, characteristics, and usage history information so that designers could use intellectual property without reservation.

Currently, the rate of intellectual property usage is approximately 20% for digital. It is approximately 30% on average for analog, which varies by process from 10% to 60%.

The rate of usage is expected to rise further in the future because of system improvements and the promotion of intellectual property registration. Design periods are also expected to shorten.

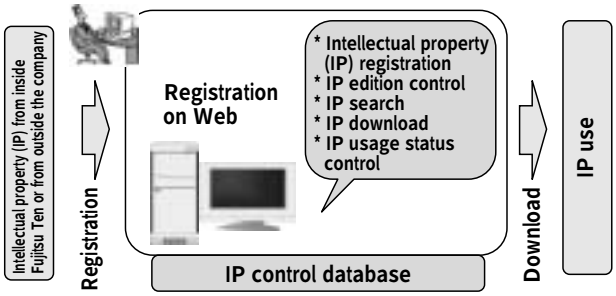


Fig.12 IP control database

3.3 Promotion of Improved design efficiency for LSI circuits evaluation environment

In order to realize LSI short-term development, emphasis is placed simultaneously on both design efficiency and highly efficient implementation of these tests and evaluations.

The contents of this test will also become more diverse and complex, particularly in response to the large-scale integration of systems in recent years. It has also become important to create an environment in which efficient evaluations can be performed.

In order to make evaluations more efficient, we are promoting the automation of evaluations by means of LSI testers.

3.3.1 Evaluation environment for digital LSI circuits

In order to use LSI testers to perform function tests of digital LSI circuits, a test pattern is needed.

Two methods are implemented to create this test pattern:

①Use of a tool that can generate a test pattern directly from waveform data taken from simulation results

②Use of a conversion tool developed by our company in the independent language of the LSI tester

Use of these methods should reduce the number of labor hours required for evaluation preparation.

3.3.2 Evaluation environment for analog LSI circuits

Analog LSI has no equivalent to digital waveform data; thus, such data cannot be reutilized. In order to evaluate, it is necessary to create a program based on the test specifications.

We developed a special analog tool that automatically generates a test program from written test specifications.

This frees the designer from having to do complicated programming and creates an environment in which the LSI tester can be used with ease.

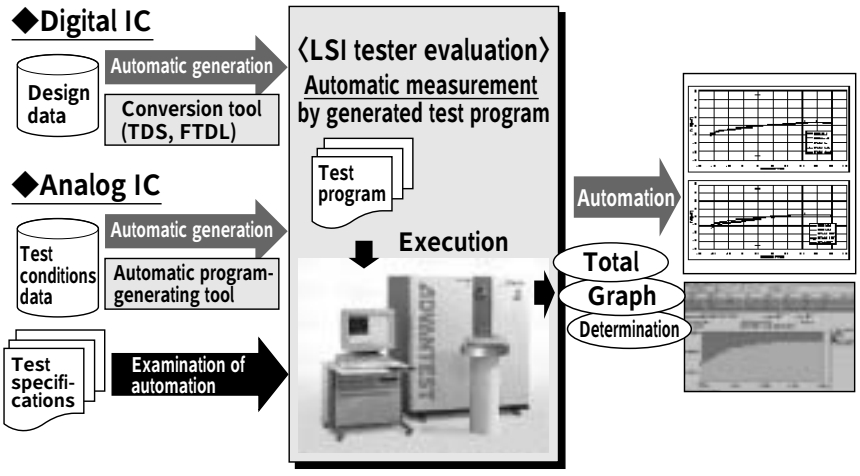


Fig.13 Outline of LSI evaluation

4 Comparison with design environments of other companies

While making the aforementioned advances in design efficiency, we compared our design environment with those of other companies in the industry and confirmed its relative level.

Survey items included ①design productivity, ② design know-how, ③number of tools, ④number of EWSs, and ⑤design quality.

①Design productivity

Comparison of design capacity; namely, the number of gates and components that can be designed per month per person

②Design know-how

Comparison of the rate of use of intellectual property from ICs that are developed

③Number of tools

Comparison of the ratio of designers per tool

④Number of EWSs

Comparison of the ratio of designers per EWS

⑤Design quality

Comparison of "completions without revisions due to design errors, bugs, etc." for ICs that are developed

Table 1 Comparison of digital design environments

	A	B
①Design productivity	—	—
②Design know-how	○	○
③EWS	—	—
④CAD tool	—	×
⑤Design quality	—	—

Table 2 Comparison of analog design environments

	C	D
①Design productivity	—	—
②Design know-how	△	×
③EWS	○	—
④CAD tool	—	—
⑤Design quality	—	○

○ : Slightly superior to our company
— : Same level as our company
△ : Slightly inferior to our company
× : Inferior to our company

The results of these comparisons confirmed the superiority of our company in the utilization of analog intellectual property, but indicated a slight lag in the use of digital ones. Other items confirmed that our environment and capacity were at levels that compared favorably to those of other companies (digital environment: two companies; analog environment: two companies).

5 Future challenges

While continuing to make progress in areas we have been engaging in up to now, we plan to reduce design periods and improve design quality from the perspectives described below.

5.1 Support for C language design

Along with increases in design scale, design periods have lengthened and there have been difficult times, even in HDL design.

Thus, in recent years, the focus of attention has been on design by C language, whose degree of abstractness is higher than that of HDL.

With C language design, advantages such as the following can be expected:

①Higher simulation speed

Making descriptions in C language further increases the degree of abstractness. As a result, the length of the description can be expected to shorten to approximately 1/25 of the conventional length, and the simulation speed can be expected to increase by a factor of approximately 50.

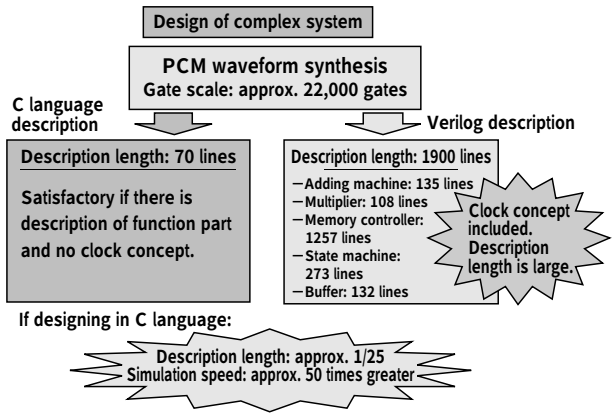


Fig.14 Circuit description in C language

②Verification of hardware/software coordination

Normally, software debugging cannot be executed if the hardware is not complete. Using virtual hardware, however, makes it possible to execute software debugging before the hardware has been completed.

In recent years, designing hardware in C language has made it possible to perform debugging in even faster environments. As a result, the verification of hardware and software coordination has gathered attention.

Some large ASIC(Application Specific IC) vendors have also made moves to switch to C language as their standard in-house design language.

As a practical matter, however, there still has not been a complete shift from HDL to C language.

Primary reasons include the following:

- A standard language has not been established.
- C language to HDL conversion tool does not exist at the practical application level.

Furthermore, there are some frustrating aspects for HDL designers. First, this requires the learning of a new language; moreover, some portions must currently be converted from C language to HDL by hand.

5.2 Support for UML

The Unified Modeling Language (UML) is a specification description language.

Expressing specifications in UML eliminates specification ambiguity, thereby reducing problems caused by specification errors that occur with conventional design techniques.

Up to now, UML has been known as a major language in the field of software development. Currently, it is also attracting attention in hardware design along with C language as interest rises in the verification of hardware/software coordination via system LSI.

Given these circumstances, our company will support design that utilizes C and UML, while monitoring trends around the globe.

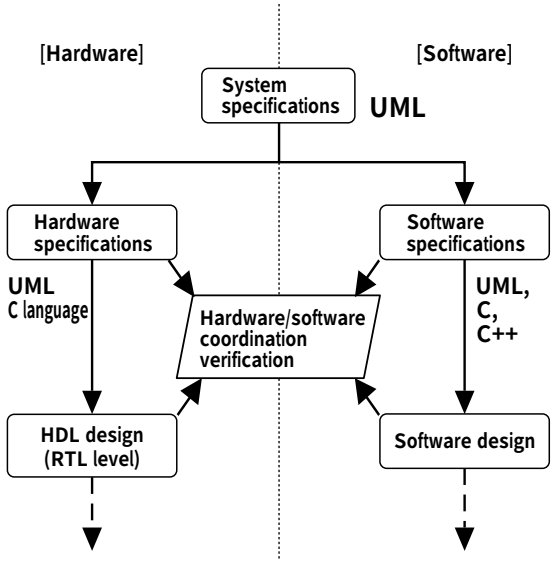


Fig.15 Design process using UML and C language

6 Conclusion

In the future there will be increasing demand for improvements in development speed and design quality in IC development. To meet such demand, it will be important to resolve the issues mentioned in this report. We will continue to take on these challenges while placing emphasis on improving the speed of C language design and digital-analog mixed design and on using intellectual property more effectively.

Profiles of Writers

Wataru Ihara

Entered the company in 1993. Since then, has been involved in the development of LSI design environments and LSI for automotive electronic equipment. Currently in the LSI Research & Development Department.

Fumitaka Fushimi

Entered the company in 1993. Since then, has been involved in the development of LSI design environments and LSI for automotive electronic equipment. Currently in the LSI Research & Development Department.

Takashi Itoh

Entered the company in 1992. Since then, has been involved in the development of evaluation technical development and inspection of LSI for automotive electronic equipment. Currently in the LSI Research & Development Department.

Keiichi Maeda

Entered the company in 1974. Since then, has been involved in the development of modbile communication related equipment, and has been working in IC development since 1995. Currently the Department General Manager of the LSI Research & Development Department.