Abstract

The CHIP validation suite, a set of programs for confirming whether a CHIP implementation conforms to the TRON specifications or not, was completed. The validation suite applies to specification level <<LIR>>, and tests functions of the instruction set, addressing modes, flags and exceptions, etc. These programs were made with a specially developed tool that automatically generates assembler source codes. The size of the validation suite is nearly three million steps. We also developed a validation enviroment system such as a monitor program allowing the validation suite to be run on a tested chip.

1. Introduction

A little over seven years since it was started, the TRON-specification VLSI microprocessor ("CHIP") subproject is now moving from the development stage to a period of actual product marketing. In line with this progress, a system has been established for testing and verifying the compatibility of CHIP implementations, enabling validation services to be offered starting this fiscal year. At the heart of this effort is a "validation suite", a set of software programs for objectively testing the compatibility of products. This validation suite is outlined below, and the validation system that is being put into operation is explained briefly.

2. Organization of the CHIP validation effort

The CHIP Technical Committee in the TRON Association carries out activities in a number of areas aimed at promoting the spread of TRON-specification CHIP products. The establishment of compatibility verification methods is one of these activities. In 1988 a Compatibility Validation and Performance Evaluation Working Group was set up in the committee in order to explore the feasibility of compatibility validation and to study methods. Since then we have examined the requirements for and means of maintaining and confirming compatibility. The TRON-specification CHIP validation suite was devised based on the results of these studies.

Our working group, while retaining its place in the CHIP Technical Committee, at the same time is under the umbrella of the TRON Association Validation Committee. The Validation Committee is responsible for coordinating the overall conformance testing activities in the TRON Project. To this end it deliberates a variety of issues related to the validation efforts in each subproject, including validation system development and conformance determination. Our working group plays a central role in performing validation testing of TRON-specification CHIP implementations and adjudicating the results, as well as setting rules on how this validation is to be carried out. Figure 1 shows the organization.

3. Aims of compatibility validation

TRON-specification microprocessors are designed according to a set of specifications, called "Specification of the Chip Based on the TRON Architecture." For this reason, theoretically all the chips implementing the specification should be functionally compatible with each other. Compatibility holds obvious advantages for software houses and others who program systems around the CHIP architecture. There is always the possibility, however, that a given implementor might have understood the specifications inaccurately or overlooked some point, leading to certain functional incompatibilities among some of the products appearing on the market. The resulting uncertainty is hardly desirable to either makers or users.
Compatibility validation has the purpose of preventing such a situation from arising. The method of testing and evaluating compatibility must be technically objective and clear. The validation suite was developed as a means of achieving these objectives.

CHIP implementations that pass the compatibility validation are certified as conforming to the TRON specification. As a result, users are able to select the product that best suits their needs from among those available, purely on the basis of functions, performance and cost, without having to worry about software compatibility. Compatibility validation thus gives users a wide range of choices while erasing fears that a given chip might not function properly in their system. As for chip manufacturers, those who opt to develop products on the TRON standards are assured of excellent opportunities for acquiring a share of the market, even if they bring out their products later than the original developers. In these and other ways, compatibility validation provides important advantages to both users and manufacturers. (See Figure 2.)

Standard setting and validation have always been closely interrelated, with the standards organization bearing the responsibility for providing conformance testing. Given the increasing importance afforded computer standardization in recent years, there is great significance to establishing a validation method and putting it into practice.
4. Validation specifications, and programs in the suite

The CHIP validation suite is a set of programs for objectively confirming whether a CHIP implementation conforms to the TRON specifications or not. As illustrated in Figure 3, the validation suite is run on the chip being tested, which is certified as a TRON-specification microprocessor if it performs all the required operations correctly.

4.1 Scope of validation testing

The "Specification of the Chip Based on the TRON Architecture" incorporates various levels of specification. Of these, the validation suite applies to level <<L1R>>, which is the standard specification for systems that do not make use of virtual memory management. Functions that are dependent on hardware or on the implementation method, such as bus specifications, external interrupts, and cache memory functions, are outside the scope of validation testing. Figure 4 shows the scope of validation testing. There are some CHIP implementations that also incorporate functions in the <<L1>> specification, which includes MMU instructions relating to virtual memory management, or functions in the <<L2>> extended specification, such as variable-length bit field instructions. The functions of <<L1>> and <<L2>>, however, are not mandatory for all implementations, and thus are not included in the validation testing. Moreover, hardware-dependent functions or those that depend on the implementation method are likewise outside the scope of testing, mainly because it would be technically unfeasible to include them. Nearly all general application programs can be written within the range of the level <<L1R>> specification. Staying within this range is thus a way of ensuring that an application program will be compatible across CHIP implementations.

Figure 3. Role of validation suite

<table>
<thead>
<tr>
<th>Instruction set specification level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware Implementation</td>
</tr>
<tr>
<td>Independent</td>
</tr>
<tr>
<td>Scope of validation</td>
</tr>
<tr>
<td>General application programs</td>
</tr>
<tr>
<td>MMU-related instructions, etc.</td>
</tr>
<tr>
<td>Variable-length bit field instructions, etc.</td>
</tr>
<tr>
<td>dependent</td>
</tr>
<tr>
<td>External interrupts, etc.</td>
</tr>
<tr>
<td>Memory management, etc.</td>
</tr>
<tr>
<td>Some control registers, etc.</td>
</tr>
</tbody>
</table>

Figure 4. Scope of validation
4.2 Test items

The following tests are included in the validation suite.
- Operation tests, which verify functions specific to the instruction set.
- Addressing mode tests.
- Tests for proper flag setting and clearing.
- Tests of EIT (exceptions, instructions and traps) that occur when instructions are issued.

(Operations and flags are tested in various combinations relating to the operand size.)

These tests cover 82 instructions, eight instruction formats, 26 operand addressing modes, three sizes, six flags, and ten EIT. Altogether approximately 30,000 items are tested. The configuration of the validation suite will be discussed later below.

4.3 How the suite was developed

A major feature of the validation suite development was the use of a specially developed tool for automatic generation of assembler source programs. The purpose of this tool is to prevent bugs from arising due to human error, shorten the development time, and assure ease of maintenance when the programs are revised or updated later on.

The development flow is shown in Figure 5. After the validation suite specifications are decided, a functional design is created on this basis. The design is written in a format allowing it to be input into the source generation tool. When the functional design is input into the tool, a source program including macro instructions is automatically output. This source program is then assembled, resulting in an object.

4.4 Validation performing

The validation is performed for each product's formal name which is implementing the TRON specifications. Chips that differ only in operating frequency or package are considered as the same product, with certification applying to all of the variations.

Implementors may apply for validation at any time. Normally the validation suite is run in the presence of staff representing the TRON Association, who take the suite to the validation site and retain possession of it afterward. It is possible, however, to contract with the TRON Association in advance for permission to use the suite, either for validation or for other purposes.

Naturally, a microprocessor cannot run software by itself, but must be incorporated in a system of some kind. The applicant is asked to provide a validation environment consisting of a single-board computer or similar hardware, in which the tested chip is built, and basic software such as a monitor program. The validation suite is loaded to the hardware system, and the results are output by the monitor program. This environment is described further below under "Sample validation system."

When a chip implementation passes the validation testing and is certified as a bona fide TRON-specification microprocessor, the manufacturer obtains the right to display the validation statement and logo mark shown in Figure 6, in manuals and other documents.

5. Outline of the validation suite

5.1 Suite makeup and file configuration

The validation suite consists of 49 modules, each of which is run as a separate unit in the validation process. Each module corresponds to one object file, and consists of several test units called sections, as shown in Figure 7. A section combines all the validation items for one format of one instruction.

An item is the smallest testing unit. Each item is made up of four processing parts (initial processing part, execution part, judgement part, and post-processing part).
This product has passed validation testing within the range of specifications indicated below, and has received qualification from the TRON ASSOCIATION.

**Specification of the Validation Suite for the Chip Based on the TRON Architecture  Ver 1.00.00.00**

Certification No. 1992 - XXX

![Validation statement](image_url)

**Figure 6. Validation statement**

as well as a data part containing addressing modes, test data and the like. The configuration of a test item is shown in Figure 8. The initial processing part initializes the data, registers, and flags, etc., in memory required for executing the instructions to be tested. Next, the execution part sets up the test environment, executes the tested instructions, and saves the execution results. The judgement part then compares these results with the expected values to determine whether operation was normal, and checks for the unexpected occurrence of an EIT. Finally, the post-processing part of the program restores the values set by the initial processing part to their original values.

The validation suite is available in either S-record format (Motorola) or MUFOM-format object programs. In S format, the file size is around 16 to 445 kbytes per module and totals 12.8 Mbytes. Each module, however, has been designed to allow the program to be made resident in a 252-kbyte memory on the evaluation board (i.e., the validation system described later below).

**5.2 Memory configuration**

Figure 9 shows the memory configuration of the validation suite. A read/write memory area of 252 kbytes, from H'00001000 to H'0003FFFF, is necessary for running the suite. This area is divided into the following sub-areas.

a. Validation suite body

This consists of the programs for executing each test item, routines for handling EITs and for holding and processing error information when an error occurs, and areas for data used in the testing. The head of the validation suite body is the starting address of the validation suite.

b. Error information storage area

When a program aborts due to error, detailed information is stored in this area.

c. EIT vector table area

The EIT vector table used when the validation suite is running is stored here.

**Figure 7. Validation suite configuration**

**Figure 8. Test item configuration**
d. Test results storage area
When the validation is completed, the test results are found here. If an item was executed successfully, the result is H'00000000; an error is registered as H'FFFFFFFF.
e. Next section address area
The starting address of the next section to be executed after completion of a section is stored here.
f. Section name area
This area is used by the validation suite to store the names of test sections.
g. User areas
The monitor program or other basic software may use these areas as necessary for running the validation suite.

5.3 Suite operation and pass/fail judgement

(1) Register initialization
The validation suite uses control registers PSW, EITVB, JRNGVB, CTXBB, CSW, DIR, SPI, SP0, SP1, SP2, and SP3, and general registers R0 to R15. Initialization of these registers is performed by the validation suite and therefore does not have to be done by the monitor program. Registers not covered by the <<LIR>> specification, such as UATB, SATB, LSID, IOADDR, IOMASK, and MJRNGV, are not initialized by the validation suite, so this is the responsibility of the applicant.

(2) Operation
Execution of the validation suite starts from the head of the validation suite body of each module. The execution results are compared with expected values for each section in a module; if they match, normal operation is assumed and the testing goes on to the next section. If different results are obtained, or if an unexpected EIT occurs, execution of the validation suite is stopped. If all the sections in a module operate normally, that module is exited and a value of H'00000000 is written to the test results storage area. In case of abnormal operation, H'FFFFFFFF is written to the same area, and the test number along with detailed error data are stored in the error information storage area. The remaining sections are not executed in this case.

(3) Modules for checking validation environment
In addition to the 49 modules making up the validation suite itself, modules are executed for checking the validation environment. These are modules created by altering normal modules in ways such as the following in order to cause errors to occur.
- Different values are written in operands (including immediate operands and literals), or in the expected results values.
- Register list contents (including the number of registers) are changed.
- An EIT that should occur is made not to occur.
- An EIT that should not occur is made to occur.
The matters of these changes are different each time the validation is performed, and are not made public.

(4) Pass/fail judgement
When the validation suite is run on the chip to be tested, all of the test items must be passed successfully. All 49 modules must be completed normally. Moreover, the execution results of all validation environment check modules must differ from normal completion. Only when these conditions are satisfied can a chip be certified as compatible.
6. Sample validation system

As noted above, the applicant for validation testing is required to provide an execution environment allowing the validation suite to be run on the chip to be tested. This validation system consists of hardware built around the chip, such as a single-board computer, and basic software such as a monitor program. The system must meet the three requirements below.

- Memory connected to the chip must contain at least 252 kbytes of RAM (from addresses H'00001000 to H'0003FFFF).
- It must be possible to download the validation suite.
- It must be possible to output the execution results to a console.

An example is given below of a typical validation system for the Gmicro/200 chip.

6.1 Hardware environment

The hardware configuration of this sample system is as shown in Figure 10.

A SUN3 workstation is used as the host machine. The Gmicro/200 processor is mounted on a target board having a VME bus interface, namely, the single-board computer HS232SBC01H. An Ethernet interface board (EP-10M/VME) is used to download the validation suite from the host computer to the target board.

The SUN3 connects to the EB-10M/VME board via an Ethernet interface; two RS-232C interface lines run between the SUN3 and HS232SBC01H, and a VME bus connects the EB-10M/VME with the HS232SBC01H.

![Diagram of hardware configuration of sample validation system](image-url)
6.2 Software configuration

The software for the validation system consists of three programs running on the target board (a. through c. below), and three programs that run on the host machine (d. through f.).

a. Monitor program for validation suite
   The monitor program controls and executes modules in the validation suite, and requests output of the results to the console when a module ends its processing. This program is independent of the target board specifications.

b. Validation suite driver routines
   There are five driver routines, namely, a driver open routine, one-module load routine, one-character console input routine, one-character console output routine, and monitor exit routine. These routines handle interfacing between the monitor program and the target board and interface board, and are thus dependent on the target board.

c. H32/200 EMS
   This program is a monitor program provided as standard on the HS232BC01H target board. It handles console I/O processing and target board initialization. Console I/O of the validation suite drivers is linked to this EMS. The program is dependent on the target board.

d. RS-232C processing process
   This program receives a request from the EMS on the target board, via RS-232C interface, and loads the monitor program used for the validation suite. It also handles target board console I/O via RS-232C. The program is dependent on the target board.

e. LAN processing process
   This program causes the host machine to operate as a client, receives a validation suite load request from the target board, and transfers the suite via LAN. The program is dependent on the target board.

f. Transfer data format conversion program
   The validation suite is stored on magnetic tape in S-record format or ASCII code format. This program converts the data to binary format (see Figure 11) for efficient data transfer. It is independent of the target board.

Programs a. and f., which are independent of the target board, are available for use by the applicant on a special contract basis along with the validation suite, so as to reduce the load on the applicant for validation system development. The applicant then is responsible only for development of the programs that are dependent on the target board specifications.

7. In conclusion

The VLSI microprocessors based on the TRON architecture have been developed by six firms, and a number of products are now available on the market. During the current fiscal year, compatibility validation testing will be carried out for the six product lines listed in Table 1.

The establishment of a validation system in the TRON-specified CHIP subproject has the direct benefit of giving both makers and users the assurance of application program portability. At the same time, it is expected to contribute indirectly to the widespread acceptance of this architecture by helping eliminate any unclear aspects of the specification and making it more complete.

A consistent policy throughout the TRON Project is that of open specifications, so that when a specification becomes a standard it may be implemented by multiple vendors. The application of this policy to a VLSI processor, which is a key aspect of hardware, is an accomplishment of great significance.

<table>
<thead>
<tr>
<th>Table 1. TRON-specified CHIP validation testing schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
</tr>
<tr>
<td>Fujitsu Limited</td>
</tr>
<tr>
<td>Hitachi, Ltd.</td>
</tr>
<tr>
<td>Matsushita Electric Industrial Co., Ltd.</td>
</tr>
<tr>
<td>Mitsubishi Electric Corp.</td>
</tr>
<tr>
<td>Oki Electric Industry Co., Ltd.</td>
</tr>
<tr>
<td>Toshiba Corporation</td>
</tr>
</tbody>
</table>
Acknowledgments

The authors wish to express their sincere gratitude for the assistance offered by Zero Software House Co., Ltd. and by Personal Media Corporation, which made possible the enormous tasks of developing the CHIP validation suite and monitor program, totaling nearly 3 million steps and requiring over 40 man-months of effort.

References


