To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)
Send any inquiries to http://www.renesas.com/inquiry.
Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.

2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.

4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.

5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.

6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.

7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depend on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
   - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.

9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.

12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.
H8SX Family

Boundary Scan: Usage

Introduction

This Application Note describes the use of the boundary scan function. See the Boundary Scan: Introduction and Boundary Scan: Application documents (Renesas Application Notes) for an overview of the boundary scan function. This document presents a simple application program that uses the boundary scan function. The boundary scan functions described in this document all apply to the H8SX/1648 microcontroller.

Contents

1. System Overview ................................................................................................................ 2
2. Application Conditions ..................................................................................................... 4
3. Specifications .................................................................................................................. 6
4. Operation ........................................................................................................................ 15
5. Software ........................................................................................................................ 24
6. Reference Documents ..................................................................................................... 37
1. System Overview

This document describes an application program that performs a boundary scan by controlling microcontrollers from a personal computer.

The example presented here uses two H8SX/1648 microcontrollers: one is used for control and the other is used for the boundary scan. HyperTerminal is used on the PC for serial communication to manipulate the control microcontroller. A boundary scan is then performed by controlling the boundary scan microcontroller's TAP from the control microcontroller. The control microcontroller's port A (PA0 to PA4) is used for input to and output from the boundary scan microcontroller's TAP. Note that the boundary scan microcontroller must be held fixed in operating mode 3 (boundary scan enabled single-chip mode).

The board for this example includes a DIP switch for verifying the SAMPLE/PRELOAD instruction. The DIP switch is connected to any boundary scan pins. Whether or not the pin states are acquired by the boundary scan cells by the SAMPLE/PRELOAD instruction can be verified by modifying the pin states using this DIP switch. The board also includes LEDs to check the EXTEST and CLAMP instructions. Whether or not the EXTEST and CLAMP instructions are functioning can be verified by modifying the LED output states with the EXTEST and CLAMP instructions.

Figure 1 shows an overview of the whole system, and table 1 lists the pin connections.

![Figure 1 System Overview](image-url)
### Table 1  Pin Connections

<table>
<thead>
<tr>
<th>Role</th>
<th>Control Microcontroller Pin</th>
<th>Boundary Scan Microcontroller Pin</th>
<th>External Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAP manipulation</td>
<td>PA0  TDI</td>
<td>PA1  TDO</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>PA2  TMS</td>
<td>PA3  TCK</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>PA4  TRST</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td>SCI communication</td>
<td>TXD4</td>
<td>RXD4</td>
<td>Connected to the PC COM port over a SCI transceiver and connector.</td>
</tr>
<tr>
<td>LED verification for the</td>
<td>—</td>
<td>P30</td>
<td>Connected to a LED.</td>
</tr>
<tr>
<td>EXTEST and CLAMP instructions</td>
<td>—</td>
<td>P31</td>
<td>Connected to a LED.</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>P32</td>
<td>Connected to a LED.</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>P33</td>
<td>Connected to a LED.</td>
</tr>
</tbody>
</table>
2. Application Conditions

Tables 2-1 and 2-2 list the application conditions for this sample system. Table 3 lists the HyperTerminal settings used in this system example.

### Table 2-1 Application Conditions (Control Microcontroller)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller</td>
<td>H8SX1648 (R5F61648FPV)</td>
</tr>
<tr>
<td>Operating frequency</td>
<td>System clock ($f_1$) = 12 MHz (input clock x 1)</td>
</tr>
<tr>
<td></td>
<td>External bus clock ($f_2$) = 12 MHz (input clock x 1)</td>
</tr>
<tr>
<td></td>
<td>Peripheral module clock ($f_3$) = 12 MHz (input clock x 1)</td>
</tr>
<tr>
<td>Operating mode</td>
<td>Single-chip mode (mode 7)</td>
</tr>
<tr>
<td>Development tools</td>
<td>Renesas Technology Corp. High-Performance Embedded Workshop, version 4.04.01.001 (integrated development environment)</td>
</tr>
<tr>
<td>C/C++ compiler</td>
<td>Renesas Technology Corp. H8S,H8/300 Standard Toolchain (V.6.2.0.0)</td>
</tr>
<tr>
<td></td>
<td>H8S,H8/300 C/C++ Compiler (V.6.2.0.0)</td>
</tr>
<tr>
<td>Option settings</td>
<td>-cpu=H8SXA:24 -object=&quot;$(CONFIGDIR)$(FILELEAF).obj&quot; -debug -nolist</td>
</tr>
<tr>
<td></td>
<td>-chgincpath -nologo</td>
</tr>
<tr>
<td>Optimizing linkage editor</td>
<td>Renesas Technology Corp. Optimizing Linkage Editor Ver.9.03.00</td>
</tr>
<tr>
<td>Option settings</td>
<td>-noprelink -nodebug -rom=D=R -nomessage</td>
</tr>
<tr>
<td></td>
<td>-list=&quot;$(CONFIGDIR)$(PROJECTNAME).map&quot;</td>
</tr>
<tr>
<td></td>
<td>-nooptimize</td>
</tr>
<tr>
<td></td>
<td>-start=PRestPRG,PInPRG/0400,P,C,C$DSEC,C$BSEC,D/0800,B,R/0FF2000, S/0FFBE00</td>
</tr>
<tr>
<td></td>
<td>-nologo -output=&quot;$$(CONFIGDIR)$(PROJECTNAME).abs&quot; -end</td>
</tr>
<tr>
<td></td>
<td>-input=&quot;$$(CONFIGDIR)$(PROJECTNAME).abs&quot; -form=stype</td>
</tr>
<tr>
<td></td>
<td>-output=&quot;$$(CONFIGDIR)$(PROJECTNAME).mot&quot; -exit</td>
</tr>
</tbody>
</table>
### Table 2-2  Application Conditions (Boundary Scan Microcontroller)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller</td>
<td>H8SX1648 (R5F61648FPV)</td>
</tr>
<tr>
<td>Operating frequency</td>
<td>System clock ((I\phi) = 12\ MHz (input clock \times 1))</td>
</tr>
<tr>
<td></td>
<td>External bus clock ((B\phi) = 12\ MHz (input clock \times 1))</td>
</tr>
<tr>
<td></td>
<td>Peripheral module clock ((P\phi) = 12\ MHz (input clock \times 1))</td>
</tr>
<tr>
<td>Operating mode</td>
<td>Boundary scan enabled single-chip mode (mode 3)</td>
</tr>
<tr>
<td>Development tools</td>
<td>None (This is because no program is written to the boundary scan microcontroller.)</td>
</tr>
<tr>
<td>C/C++ compiler</td>
<td>None (This is because no program is written to the boundary scan microcontroller.)</td>
</tr>
<tr>
<td></td>
<td>Option settings</td>
</tr>
<tr>
<td></td>
<td>None</td>
</tr>
<tr>
<td>Optimizing linkage editor</td>
<td>None (This is because no program is written to the boundary scan microcontroller.)</td>
</tr>
<tr>
<td></td>
<td>Option settings</td>
</tr>
<tr>
<td></td>
<td>None</td>
</tr>
</tbody>
</table>

### Table 3  HyperTerminal Settings

<table>
<thead>
<tr>
<th>Item</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud rate</td>
<td>9600bps</td>
</tr>
<tr>
<td>Data length</td>
<td>8 bits</td>
</tr>
<tr>
<td>Parity</td>
<td>None</td>
</tr>
<tr>
<td>Stop bits</td>
<td>1bit</td>
</tr>
<tr>
<td>Flow control</td>
<td>Hardware</td>
</tr>
</tbody>
</table>
3. Specifications

This sample system uses six boundary scan instructions: SAMPLE/PRELOAD, IDCODE, EXTEST, BYPASS, CLAMP, and HIGHZ instructions. It also acquires the IR status word.

When specified characters are input to HyperTerminal, instructions corresponding to the input characters are issued by the boundary scan microcontroller, and the results are displayed on the HyperTerminal screen. Outputs from boundary scan cells to pins are also performed when an EXTEST or CLAMP instruction is executed. Furthermore, the HIGHZ instruction sets all pins other than the TAP pins to the high-impedance state. Table 4 lists the correspondence between HyperTerminal input commands and the instructions used.

<table>
<thead>
<tr>
<th>Command input from HyperTerminal</th>
<th>Instruction used (instruction code)</th>
<th>Operation in this sample system</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;ID&quot;</td>
<td>IDCODE instruction (B'0001)</td>
<td>The IDCODE is acquired, converted to hexadecimal, and displayed on HyperTerminal.</td>
</tr>
<tr>
<td>&quot;SP&quot;</td>
<td>SAMPLE/PRELOAD instruction (B'0100)</td>
<td>The pin states are acquired by the boundary scan cells and displayed on HyperTerminal.</td>
</tr>
<tr>
<td>&quot;EX&quot;</td>
<td>EXTEST instruction (B'0000)</td>
<td>The P30 to P33 pin output states are manipulated thus manipulating the LED on/off states.</td>
</tr>
<tr>
<td>&quot;BY&quot;</td>
<td>BYPASS instruction (B'1111)</td>
<td>The input data is passed through the bypass register, and the resulting data is displayed on HyperTerminal.</td>
</tr>
<tr>
<td>&quot;CP&quot;</td>
<td>CLAMP instruction (B'0010)</td>
<td>In the state where the P30 to P33 pin output states are manipulated thus manipulating the LED on/off states, the input data is passed through the bypass register, and the resulting data is displayed on HyperTerminal.</td>
</tr>
<tr>
<td>&quot;HZ&quot;</td>
<td>HIGHZ instruction (B'0011)</td>
<td>In the state where all pins other than the TAP pins are set to the high-impedance state, the input data is passed through the bypass register, and the resulting data is displayed on HyperTerminal.</td>
</tr>
<tr>
<td>&quot;IR&quot;</td>
<td>SAMPLE/PRELOAD instruction (B'0100)</td>
<td>The IR status word is acquired and displayed on HyperTerminal.</td>
</tr>
</tbody>
</table>
3.1 IDCODE Instruction Specifications

The IDCODE instruction is performed as follows.

Type "ID" at HyperTerminal and press the Enter key. The boundary scan microcontroller will then acquire the IDCODE. Then, the acquired IDCODE is converted to hexadecimal and displayed on HyperTerminal. If the IDCODE instruction fails, the system will display "IDCODE INSRUCTION FAILURE" on HyperTerminal. Figure 2 shows an overview of this operation.

Type "ID" and press the Enter key.

The acquired IDCODE (0000 1000 0000 0011 1010 0100 0100 0111) is converted to hexadecimal (0x0803A447).

Message indicating that the IDCODE instruction failed.

IDCODE = 0x0803A447

Figure 2  IDCODE Instruction Specifications
3.2 SAMPLE/PRELOAD Instruction Specifications

The SAMPLE/PRELOAD instruction is performed as follows.

Type "SP" at HyperTerminal and press the Enter key. The boundary scan microcontroller pin states will be acquired by the boundary scan cells. Then, the boundary scan cell states will be displayed as a sequence of zeros and ones on hold time. The data is displayed with a newline after every 100 characters.

If the SAMPLE/PRELOAD instruction fails, "SAMPLE INSTRUCTION FAILURE" will be displayed. Figure 3 presents an overview of this operation.

Note that the acquired boundary scan cell values will change if the DIP switch connected to arbitrary boundary scan pins is manipulated. This can be used to verify that the pin states are actually being acquired by the boundary scan cells by the SAMPLE/PRELOAD instruction.

![Figure 3 SAMPLE/PRELOAD Instruction Specifications](image_url)

The pin states are displayed in order from the TDO pin side. (A newline is inserted every 100 characters.)
3.3 EXTEST Instruction Specifications

The IDCODE instruction is performed as follows.

Type "EX" at HyperTerminal and press the Enter key. The system will display "Please input the output setting of P3x. (H or L):" on HyperTerminal. Next, enter H for a high-level output or L for a low-level output for the specified pin (P30 to P33). If you do not want to change the pin state, enter any other character. When the output levels have been set for all four pins (P30 to P33), the on/off state of the LED corresponding to each pin will change. If the instruction succeeded, "EXTEST INSTRUCTION SUCCESS" will be displayed on HyperTerminal. If the instruction failed, "EXTEST INSTRUCTION FAILURE" will be displayed. Figure 4 shows an overview of this operation.

Since the EXTEST instruction performs a forcible output to the pins, the user must assure that no problems will occur in the system when the instruction is executed, even if there are outputs from the set pins. Also, since the EXTEST instruction is a test mode instruction, normal microcontroller operation will be stopped until the microcontroller is set to normal mode after EXTEST instruction execution.

![Figure 4 EXTEST Instruction Specifications](image-url)
3.4 **BYPASS Instruction Specifications**

The BYPASS instruction is performed as follows.

Type "BY" at HyperTerminal and press the Enter key. The system will display "Please input the Bypass Data (H or L):" on HyperTerminal. Next, enter the bypass data using the letters H and L and press the Enter key. You can enter up to 100 characters. Then, the system will pass the bypass data through the bypass register and display the result on HyperTerminal. Since this sample system performs a TAP controller reset each time an instruction is entered with HyperTerminal, the first character must be the bypass register's initial value, that is, 0. Also, since the last bit in the bypass data remains in the bypass register, it will not be output. If the BYPASS instruction fails, "BYPASS INSTRUCTION FAILURE" will be displayed on HyperTerminal. Figure 5 shows an overview of this operation.

---

**Figure 5**  **BYPASS Instruction Specifications**
3.5 CLAMP Instruction Specifications

The CLAMP instruction is performed as follows.

Type "CP" at HyperTerminal and press the Enter key. The system will display "Please input the output setting of P3x. (H or L):" on HyperTerminal. Next, enter H for a high-level output or L for a low-level output for the specified pin (P30 to P33). If you do not want to change the pin state, enter any other character. When the output levels have been set for all four pins (P30 to P33), the system will then display "Please input the Bypass Data (H or L):" on HyperTerminal. Enter the bypass data (up to 100 characters) and press the Enter key. The on/off state of the LED corresponding to each pin will change according to the previously entered data. Then, the system will pass the bypass data through the bypass register and display the result on HyperTerminal. Since this sample system performs a TAP controller reset each time an instruction is entered with HyperTerminal, the first character must be the bypass register's initial value, that is, 0. Also, since the last bit in the bypass data remains in the bypass register, it will not be output. Note that if the CLAMP instruction fails, the bypass result will not be displayed on HyperTerminal, but rather the system will display "CLAMP INSTRUCTION FAILURE". Figure 6 shows an overview of this operation.

Since the CLAMP instruction performs a forcible output to the pins, the user must assure that no problems will occur in the system when the instruction is executed, even if there are outputs from the set pins. Also, since the CLAMP instruction is a test mode instruction, normal microcontroller operation will be stopped until the microcontroller is set to normal mode after CLAMP instruction execution.
Type "CP" and press the Enter key.

Please input the output setting of P30.(H or L) : H
Please input the output setting of P31.(H or L) : L
Please input the output setting of P32.(H or L) : H
Please input the output setting of P33.(H or L) : L

Text prompting for specification of the output states for the pins P30 to P33 is displayed.
Enter "H" or "L". Enter any other character if you do not want to change the pin state.

Please input the output setting of P30.(H or L) : H
Please input the output setting of P31.(H or L) : L
Please input the output setting of P32.(H or L) : H
Please input the output setting of P33.(H or L) : L
Please input the Bypass Data(H or L) : HLHL

Display of a prompt for input of the bypass data.
Enter the data as "H" and "L" and press the Enter key.

A message indicating that the CLAMP instruction failed is displayed.

The result of passing the bypass data through the bypass register is displayed.
(The first data output will always be 0, and the last bit in the bypass data is not output.)

Figure 6  CLAMP Instruction Specifications
3.6 HIGHZ Instruction Specifications

The HIGHZ instruction is performed as follows.

Type "HZ" at HyperTerminal and press the Enter key. The system will then display "Please input the Bypass Data (H or L):" on HyperTerminal. Enter the bypass data (up to 100 characters) and press the Enter key. The system will pass the bypass data through the bypass register while holding all pins other than the TAP pins in the high-impedance state and display the result on HyperTerminal. Since this sample system performs a TAP controller reset each time an instruction is entered with HyperTerminal, the first character must be the bypass register's initial value, that is, 0. Also, since the last bit in the bypass data remains in the bypass register, it will not be output. If the HIGHZ instruction fails, "BYPASS INSTRUCTION FAILURE" will be displayed on HyperTerminal. Figure 7 shows an overview of this operation.

Since the HIGHZ instruction is a test mode instruction, normal microcontroller operation will be stopped until the microcontroller is set to normal mode after HIGHZ instruction execution.

---

**Figure 7  HIGHZ Instruction Specifications**
3.7 IR Status Word Acquisition Specifications

The IR status word is acquired as follows.

Type "IR" at HyperTerminal and press the Enter key. The system will acquire the IR status word from the boundary scan microcontroller and display it on HyperTerminal in the order it is output from TDO. Figure 8 shows an overview of this operation.

```
HyperTerminal
IR  
Type "IR" and press the Enter key.

HyperTerminal
IR  
IR STATUS WORD:
(TDO)
1000
(TDI)

The IR status word is acquired and displayed.
```

Figure 8  IR Status Word Acquisition Specifications
4. Operation

This section describes the operation of the boundary scan function as used in this sample system.

4.1 IDCODE Instruction

The IDCODE instruction operation is performed as follows in this sample system.

The IDCODE instruction code is input to the boundary scan microcontroller TDI pin, and the IDCODE instruction is executed. In the following descriptions, the procedure for issuing instructions is the same. The device IDCODE is acquired from the TDO pin by iterating the DR-Shift state the IDCODE length (32 times). Figure 9 shows the data flow for this operation.

![Diagram showing the IDCODE instruction process]

Figure 9 Data Flow for the IDCODE Instruction
4.2 SAMPLE/PRELOAD Instruction

The SAMPLE/PRELOAD instruction operation is performed as follows in this sample system.

The SAMPLE/PRELOAD instruction is issued in the boundary scan microcontroller to acquire the pin states to the boundary scan cells in the capture-DR state. Then the boundary scan cell states are acquired from the TDO pin by iterating the DR-Shift state for the number of boundary scan cells (296 times). Figure 10 shows the data flow. Note that in this figure one boundary scan cell is shown for one pin's I/O state for simplicity. In the actual device, however, there may be up to three cells for a single pin. The following figures in this section are similar in this respect.

---

**Figure 10** Data Flow for the SAMPLE/PRELOAD Instruction
4.3 EXTEST Instruction

The EXTEST instruction operation is performed as follows in this sample system.

First, the pin states are acquired in the boundary scan cells by performing a SAMPLE/PRELOAD instruction on the boundary scan microcontroller. Then, only the values whose output states are to be changed from the acquired boundary scan cell values are changed (for a high-level output, the output cell and output enable cell are changed to 1 and for a low-level output, the output cell is changed to 0 and the output enable cell is changed to 1), and the boundary scan cells are set again. An EXTEST instruction is executed to forcibly output the pin states. Figure 11 shows the data flow. Table 5 lists the boundary scan cell setting values set in the operation shown in figure 11 and pin state examples (LED on/off state manipulations only).

![Diagram of EXTEST Instruction Operation](image-url)

**Figure 11 Data Flow for the EXTEST Instruction**
<table>
<thead>
<tr>
<th>Pin (pin number)</th>
<th>TDI</th>
<th>P30(56)</th>
<th>P31(57)</th>
<th>P32(58)</th>
<th>P33(62)</th>
<th>TDO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boundary scan</td>
<td>165</td>
<td>164</td>
<td>163</td>
<td>162</td>
<td>161</td>
<td>160</td>
</tr>
<tr>
<td>cell bit</td>
<td></td>
<td>159</td>
<td>158</td>
<td>157</td>
<td>149</td>
<td>148</td>
</tr>
<tr>
<td>I/O</td>
<td></td>
<td>147</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cell setting value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pin State</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:
- I: Input
- E: Output enable
- O: Output
4.4 BYPASS Instruction

The BYPASS instruction operation is performed as follows in this sample system.

A BYPASS instruction is issued in the boundary scan microcontroller and the values to bypass are input to the TDI pin while iterating the shift-DR state for the amount of bypass data. The bypass data passes through the bypass register and is acquired by the TDO pin. Since this sample system performs a TAP controller reset each time an instruction is entered with HyperTerminal, the first character must be the bypass register's initial value, that is, 0. Also, since the last bit in the bypass data remains in the bypass register, it will not be output. Figure 12 shows the data flow.

Figure 12 Data Flow for the BYPASS Instruction
4.5 CLAMP Instruction

The CLAMP instruction operation is performed as follows in this sample system.

First, the pin states are acquired in the boundary scan cells by performing a SAMPLE/PRELOAD instruction on the boundary scan microcontroller. Then, only the values whose output states are to be changed from the acquired boundary scan cell values are changed (for a high-level output, the output cell and output enable cell are changed to 1 and for a low-level output, the output cell is changed to 0 and the output enable cell is changed to 1), and the boundary scan cells are set again. Also, bypass data (up to 100 characters) is input. A CLAMP instruction is issued to force pin output and the values to bypass are input to the TDI pin while iterating the shift-DR state for the amount of bypass data. The bypass data passes through the bypass register and is acquired by the TDO pin. Since this sample system performs a TAP controller reset each time an instruction is entered with HyperTerminal, the first character must be the bypass register's initial value, that is, 0. Also, since the last bit in the bypass data remains in the bypass register, it will not be output. The data flow diagram is shown in figures 13-1 and 13-2. Note that the boundary scan cell data and pin states set when the operation of figures 13-1 and 13-2 are performed are the same as those in table 5.

![Data Flow for the Clamp Instruction](image)

**Figure 13-1 Data Flow for the Clamp Instruction**
Data that was bypassed (The first bit will be 0 (the initial value of the bypass register) and the last data bit bypassed is not output.)

Figure 13-2 Data Flow for the Clamp Instruction
4.6 HIGHZ Instruction

The HIGHZ instruction operation is performed as follows in this sample system.

First, a HIGHZ instruction is issued in the boundary scan microcontroller to set the pins to the high-impedance state. Then the values to bypass are input to the TDI pin while iterating the shift-DR state for the amount of bypass data. The bypass data passes through the bypass register and is acquired by the TDO pin. Since this sample system performs a TAP controller reset each time an instruction is entered with HyperTerminal, the first character must be the bypass register’s initial value, that is, 0. Also, since the last bit in the bypass data remains in the bypass register, it will not be output. Figure 14 shows the data flow.

![Figure 14 Data Flow for the HIGHX Instruction](image-url)
4.7 IR Status Word Acquisition

In this sample system, the IR status word is acquired when a SAMPLE/PRELOAD instruction is issued.

The IR status word acquisition operation in this sample system is performed as follows.

The SAMPLE/PRELOAD instruction code is input to the TDI pin while iterating the IR-Shift state for the instruction code length (4 times) in the boundary scan microcontroller. As a result, the IR status word is acquired from the TDO pin. Figure 15 shows the data flow.

Note that while this sample system uses the SAMPLE/PRELOAD instruction to acquire the IR status word, the IR status word can also be acquired in the same manner using other instructions.

![Diagram showing data flow for IR status word acquisition](image-url)

**Figure 15  Data Flow for IR Status Word Acquisition**
5. Software

5.1 Functions
Table 6 lists the main functions used in this sample system.

<table>
<thead>
<tr>
<th>Function</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>main()</td>
<td>Identifies the received data and calls the instructions.</td>
</tr>
<tr>
<td>tap_reset()</td>
<td>Resets the TAP controller</td>
</tr>
<tr>
<td>Idcode_Instruction()</td>
<td>IDCODE instruction processing</td>
</tr>
<tr>
<td>Sample_Instruction()</td>
<td>SAMPLE/PRELOAD instruction processing</td>
</tr>
<tr>
<td>Extest_Instruction()</td>
<td>EXTEST instruction processing</td>
</tr>
<tr>
<td>Bypass_Instruction()</td>
<td>BYPASS instruction processing</td>
</tr>
<tr>
<td>Clamp_Instruction()</td>
<td>CLAMP instruction processing</td>
</tr>
<tr>
<td>Highz_Instruction()</td>
<td>HIGHZ instruction processing</td>
</tr>
<tr>
<td>IRWord_Display()</td>
<td>Displays the acquired IR status word on the PC</td>
</tr>
<tr>
<td>execute_Instruction()</td>
<td>Controls the TAP controller</td>
</tr>
<tr>
<td>Get_IRword()</td>
<td>Acquires the IR status word</td>
</tr>
</tbody>
</table>

5.2 Section Settings
Table 7 Section Settings

<table>
<thead>
<tr>
<th>Address</th>
<th>Section name</th>
</tr>
</thead>
<tbody>
<tr>
<td>H'000000400</td>
<td>PResetPRG,PIntPRG</td>
</tr>
<tr>
<td>H'000000800</td>
<td>P,C,S$DSEC,C$BSEC,D</td>
</tr>
<tr>
<td>H'00FF2000</td>
<td>B,R</td>
</tr>
<tr>
<td>H'00FFBE00</td>
<td>S</td>
</tr>
</tbody>
</table>
5.3 Flowcharts

5.3.1 Main Routine

- **main()**
  - Port initialization
  - SCI initialization
  - Receive data from PC
  - `tap_reset()`

  - **Received data = "ID"?**
    - Yes: **Idcode_Instruction()**
    - No: **Display error on PC**

  - **Received data = "SP"?**
    - Yes: **Sample_Instruction()**
    - No: **Display()**

  - **Received data = "EX"?**
    - Yes: **Extest_Instruction()**
    - No: **Display()**

  - **Received data = "BY"?**
    - Yes: **Bypass_Instruction()**
    - No: **Display()**

  - **Received data = "CP"?**
    - Yes: **Clamp_Instruction()**
    - No: **Display()**

  - **Received data = "HZ"?**
    - Yes: **Highz_Instruction()**
    - No: **Display()**

  - **Received data = "IR"?**
    - Yes: **IRWord_Display()**
    - No: **Display error on PC**
5.3.2 TAP Controller Reset Routine

```
tap_reset()
```

Switch to Test-Logic-Reset state
Resets TAP controller

END

5.3.3 IDCODE Instruction Routine

```
Idcode Instruction()
```

Execute IDCODE instruction with execute_instruction()
Failure
Success

- Convert received IDCODE to hexadecimal and display on HyperTerminal
- Display IDCODE failure message on HyperTerminal

END
5.3.4 SAMPLE/PRELOAD Instruction Routine

- Sample_Instruction()
- Execute SAMPLE/RELOAD instruction with execute_instruction()
- Display acquired boundary scan values on HyperTerminal
- Display SAMPLE/RELOAD instruction failure message on HyperTerminal
- Success
- Failure
- END
5.3.5 EXTEST Instruction Routine

Extest_instruction()}

Execute SAMPLE/PRELOAD instruction with execute_instruction()}

Enter output settings (H or L) for target pins from HyperTerminal

Success

Failure

Display SAMPLE/PRELOAD instruction failure message on HyperTerminal

Load output states set in boundary scan cells

Execute SAMPLE/PRELOAD instruction with execute_instruction()}

Execute EXTEST instruction with execute_instruction()}

Success

Failure

Display EXTEST instruction success indication on HyperTerminal

Display EXTEST instruction failure message on HyperTerminal

END
5.3.6 BYPASS Instruction Routine

Bypass Instruction()

Enter bypass data (H or L) using HyperTerminal

Execute BYPASS instruction with execute_instruction()

Display bypass data on HyperTerminal

Display BYPASS instruction failure message on HyperTerminal

END
5.3.7 CLAMP Instruction Routine

Clamp_Instruction() 

- Execute SAMPLE/PRELOAD instruction with execute_instruction() 
  - Failure 
  - Success 

  - Enter output settings (H or L) for target pins using HyperTerminal 
  - Display SAMPLE/PRELOAD instruction failure message on HyperTerminal 

  - Enter bypass data (H or L) using HyperTerminal 
  - Load output states set in boundary scan cells 
  - Execute SAMPLE/PRELOAD instruction with execute_instruction() 
  - Execute CLAMP instruction with execute_instruction() 
    - Success 
    - Failure 

    - Display bypass data on HyperTerminal 
    - Display CLAMP instruction failure message on HyperTerminal 

END
5.3.8 HIGHZ Instruction Routine

Highz_Instruction()

Enter bypass data (H or L) using HyperTerminal

Execute HIGHZ instruction with execute_instruction()

Success

Display bypass data on HyperTerminal

Failure

Display HIGHZ instruction failure message on HyperTerminal

END
5.3.9 IR Status Word Display Routine

IRWord_Display()

Get_IRword() Acquire IR status word.

Display acquired IR status word on HyperTerminal

END
5.3.10 TAP Controller Control Routine

execute_instruction()

What instruction is to be executed?

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDCODE</td>
<td>Set transition count to DR-Shift state to 32 (IDCODE bit length)</td>
</tr>
<tr>
<td>SAMPLE/PRELOAD</td>
<td>Set transition count to DR-Shift state to 296 (boundary scan cell count)</td>
</tr>
<tr>
<td>EXTEST</td>
<td>Set transition count to DR-Shift state to 296 (boundary scan cell count)</td>
</tr>
<tr>
<td>BYPASS</td>
<td>Set transition count to DR-Shift state to bypass data count</td>
</tr>
<tr>
<td>CLAMP</td>
<td>Set transition count to DR-Shift state to bypass data count</td>
</tr>
<tr>
<td>HIGHZ</td>
<td>Set transition count to DR-Shift state to bypass data count</td>
</tr>
<tr>
<td>Other</td>
<td></td>
</tr>
</tbody>
</table>

Failure

Switch to Run-Test/Idle state

Switch to Select-DR state

1

return FAILURE
1. Switch to Select-IR state
2. Switch to Capture-IR state
3. Switch to Shift-IR state

4. Loop 4 times (instruction code length)
   - Switch to Shift-IR state

5. Acquire TDO pin state
6. Apply SAMPLE/PRELOAD instruction code data to TDI pin

7. Has loop count reached 4?
   - Yes
     - Switch to Exit1-IR state
   - No
     - Acquire IR status word.

8. Is value acquired from TDO = 0001?
   - Yes
     - Determined if correct IR status word was acquired.
   - No
     - Failure

9. Return FAILURE
Switch to Update-IR state

Switch to Select-DR state

Switch to Capture-DR state

Switch to Shift-DR state

Acquire TDO pin state

Apply instruction code value to TDI pin

Has loop until number of transitions minus 1 due to each instruction in the Shift-DR state completed?

Yes

Switch to Exit1-DR state

Switch to Update-DR state

Switch to Run-Test/Idle state

return SUCCESS

Success

Data is acquired from the specified register (boundary scan cell or bypass register) according to the instruction issued.

Data is stored to the specified register (boundary scan cell or bypass register) according to the instruction issued.

Issue instruction

No

Output data is updated for the EXTEST and CLAMP instructions.
5.3.11 IR Status Word Acquire Rutine

- Get_IRword()
- Switch to Run-Test/Idle state
- Switch to Select-DR state
- Switch to Select-IR state
- Switch to Capture-IR state
- Switch to Shift-IR state
- Loop 4 times (instruction code length)
- Acquire TDO pin state
- Apply SAMPLE/PRELOAD instruction code data to TDI pin
- Has loop count reached 4?
  - No
  - Yes
- Switch to Exit1-IR state
- Switch to Update-IR state
- Switch to Run-Test/Idle state
- END
6. Reference Documents

- Hardware Manuals
  H8SX/1648 Group Hardware Manual
  The latest versions of this manual can be ordered from the Renesas Technology Corp. web site.

- Technical Manuals and Technical Updates
  The latest information is available on the Renesas Technology Corp. web site.
Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Revision Record

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Jan.22.09</td>
<td>— First edition issued</td>
</tr>
</tbody>
</table>

All trademarks and registered trademarks are the property of their respective owners.
Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.

2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.

3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.

4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)

5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.

6. When using or otherwise relying on the information in this document, you should evaluate the information in the light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guarantees regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.

7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.

8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages or infringement of any intellectual property or other rights arising out of such applications.

9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.

10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.

11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.

12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.

13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

© 2009. Renesas Technology Corp., All rights reserved.