# Emmitsburg Platform Controller Hub (PCH)

## Boundary Scan Description Language (BSDL)

***June 2021***

**Revision 0.6**

***Intel Confidential***

**Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information.**

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document. The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel is a sponsor and member of the Benchmark XPRT Development Community and was the major developer of the XPRT family of benchmarks. Principled Technologies is the publisher of the XPRT family of benchmarks. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others

# Contents



## Figures



## Tables

# Revision History

| Revision Number | Description | Date |
|---|---|---|
| 0.6 | • Updated 2001 and 2013 BSDL file to correct various errors (misspelled names, incorrect number of power pins in definitions, pin names too long, and others) | June 2021 |
| 0.5 | • Initial Release | May 2020 |

# 1 Introduction

## 1.1 About This Manual

This document will be used as a support document for the Emmitsburg Platform Controller Hub (PCH) Boundary Scan Description Language (BSDL).

This document is intended for the development of IEEE* 1149.6 boundary scan tests for the Emmitsburg PCH. This manual assumes a working knowledge of IEEE* 1149.6 methodologies and the In-Circuit Test (ICT) manufacturing test methods.

Chapter 1 provides information on the organization of this document.

Chapter 2 provides a detailed discussion of the implementation and use of the boundary scan test mode via the JTAG* interface in the PCH.

**Table 1-1. Reference Documents**

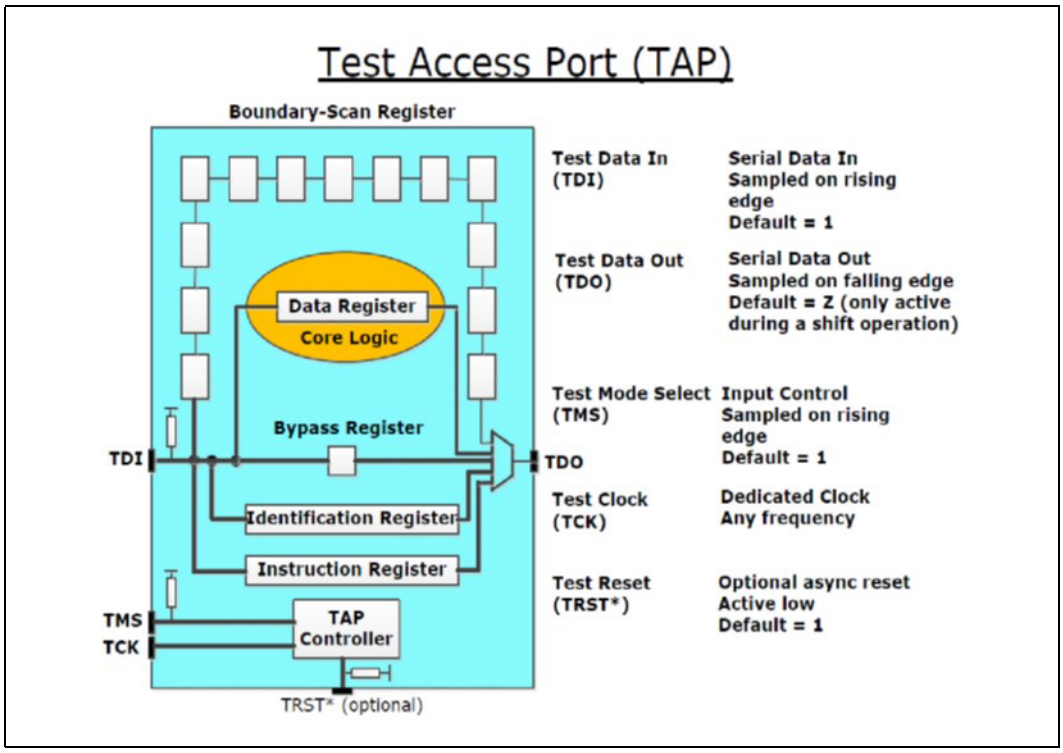
| Document Name | Document ID |
|---|---|
| *Emmitsburg Platform Controller Hub External Design Specification* | 606161 |

# 2 PCH JTAG* and Boundary Scan Test Modes

## 2.1 Test Access Port (TAP) Controller

The Emmitsburg PCH has dedicated JTAG* pins and Test Access Port (TAP) controller as shown in Figure 2-1. For detailed information on functionality of the interface, see the following specifications: *IEEE Standard Test Access Port and Boundary Scan Architecture 1149.1-2001 Specification* and *1149.6-IEEE Standard for Boundary-Scan Testing of Advanced Digital Networks – Specification*.

**Figure 2-1. Tap Controller**




To support boundary scan for AC coupled, high-speed I/Os, the master TAP supports IEEE* 1149.6. This includes the extra instructions **extest_pulse** and **extest_train**. There is also an additional instruction that is not in the IEEE* 1149.6 specification, defined by Intel - **extest_toggle**. For details on supported boundary scan instructions see Table 2-2. The Emmitsburg network is accessed serially through four dedicated component pins as shown in Table 2-1.

***Note:*** The 1149.1 specification also defines an optional TRST# TAP input pin, to asynchronously reset the TAP controllers; however, the Emmitsburg platform does not implement this pin. Instead, each TAP controller is asynchronously reset by an internal power OK signal corresponding to the power well that it is in.

**Table 2-1. JTAG* Signals**

| Name | Type | Description |
|---|---|---|
| JTAG_TCK | I/O | **Test Clock Input (TCK):** The test clock input provides the clock for the JTAG* test logic. |
| JTAG_TMS | I/O | **Test Mode Select (TMS):** This signal is decoded by the TAP controller to control test operation. |
| JTAG_TDI | I/O | **Test Data Input (TDI):** Serial test data and instructions are received by the test logic at TDI. |
| JTAG_TDO | I/O | **Test Data Output (TDO):** TDO is the serial output of test instruction and data from the test logic defined in this standard. |

During the boundary scan test mode, a boundary scan unit will take over the IO/Analog Front End (AFE) control for direct control to execute all the different IEEE* 1149.1 and 1149.6 instructions.

## 2.2 JTAG* Test Mode Entry

For functionality of the JTAG* boundary scan interface, see the specification mentioned previously. To enter the boundary scan test mode, the PCH should follow the regular power up sequencing to boot to S0 first. See the *Emmitsburg Platform Controller Hub External Design Specification*, document number 606161, for details on desired power sequencing requirement.

## 2.3 TAP Timing

**Table 2-2. JTAG* Signal Timings**

| Parameter | Minimum | Maximum | Unit | Notes |
|---|---|---|---|---|
| TCK Frequency | | 50 | MHz | |
| TCK Duty Cycle | 45% | 55% | | TYP = 50% |
| TRST_N Assert Time | 3 | | TCK | |
| Setup Time of TMS and TDI | 2.5 | | ns | |
| Hold Time of TMS and TDI | 2.5 | | ns | |
| Slew Rate | 0.05 | 0.3 | V/ns | 20%/80% Threshold |

## 2.4 Supported Instructions

The JTAG* instruction register has predefined decodes for EXTEST (all “0”s) and BYPASS (all “1”s). SAMPLE and PRELOAD are required instructions, but the decode values are user defined. The Emmitsburg platform implements the commands listed in Table 2-3 in the TAP controller.

**Table 2-3. JTAG* Instruction Register Description**

| IR Opcode (9 bits) | R/W Opcode (Bit[8]) | Unique Opcode (Bit[7:0]) | Command/Register (Length) |
|---|---|---|---|
| 001h | 0b | 00h | SAMPLE/PRELOAD |
| 002h | 0b | 02h | IDCODE(32) |
| 004h | 0b | 04h | CLAMP |
| 008h | 0b | 08h | HIGHZ |
| 009h | 0b | 09h | EXTEST |
| 00Dh | 0b | 0Dh | EXTEST_TOGGLE |
| 00Eh | 0b | 0Eh | EXTEST_PULSE |
| 00Fh | 0b | 0Fh | EXTEST_TRAIN |
| 0FFh | 1b | FFh | BYPASS |

## 2.4.1 JTAG* Instruction Registers

The following sections describe the registers shown in Table 2-3 in greater detail. Bit eight is used as Read/Write (R/W) bit for most registers, the corresponding aliasing address are implied unless specifically mentioned.

***Note:*** The number of bits "N" depends on the number of boundary scan cells implemented.

### 2.4.1.1 Offset 01h: SAMPLE/PRELOAD - SAMPLE/PRELOAD for Boundary Scan

| Bit | Type | Reset | Description |
|---|---|---|---|
| N | RW | 0 | **SAMPLE:** Command for Boundary Scan. Selects the boundary scan register and puts the boundary scan cell in functional mode. Use it to capture a functional signal to connect to a boundary scan cell input. |

### 2.4.1.2 Offset 02h: IDCODE - Identify Code for Boundary Scan

| Bit | Type | Reset | Description |
|---|---|---|---|
| 31:28 | RO | 0 | **Version:** Used to identify a variant of the component type.<br>Implementation note: Implement it to change the value in any metal layers. |
| 27:12 | RO | A113h | **Part Number:** Unique value to represent the component. The Emmitsburg part number is A114h. |
| 11:1 | | 00000 00100 1b | **Manufacture Identity:** Intel identity is 00000001001b |
| 0 | RO | 1b | **Hard code to "1"** as indication of start of IDCODE. |

### 2.4.1.3 Offset 03h: PRELOAD - PRELOAD for Boundary Scan

| Bit | Type | Reset | Description |
|---|---|---|---|
| N | RW | 0 | **PRELOAD:** Command for Boundary Scan. Selects the boundary scan register and puts a boundary scan cell in functional mode. Use it to load to a boundary scan cell with a known input value shifting in from the TDI ping. |

### 2.4.1.4 Offset 04h: CLAMP - CLAMP for Boundary Scan

| Bit | Type | Reset | Description |
|---|---|---|---|
| N | RW | 0 | **CLAMP:** Command for Boundary Scan. The clamp command drives the boundary scan value to the output pins, but leaves the bypass register as a selected register. It is the same as the EXTEST command, but CLAMP selects the bypass register. |

### 2.4.1.5 Offset 08h: HIGHZ - HIGHZ for Boundary Scan

| Bit | Type | Reset | Description |
|---|---|---|---|
| N | RW | 0 | **HIGHZ:** Command for Boundary Scan. Puts all the output pins supporting a boundary scan in the High-Z state and leaves the bypass register as selected register. It is the same function as INTEST except Bypass register is selected. |

### 2.4.1.6 Offset 09h: EXTEST - EXTEST for Boundary Scan

| Bit | Type | Reset | Description |
|---|---|---|---|
| N | RW | 0 | **EXTEST:** Command for Boundary Scan. Selects a boundary scan register and puts the boundary scan cell in output mode for DUT connectivity testing. |

### 2.4.1.7 Offset 0Dh: EXTEST_TOGGLE - EXTEST TOGGLE Boundary Scan

| Bit | Type | Reset | Description |
|---|---|---|---|
| N | RW | 0 | **EXTEST_TOGGLE:** Special Command for Boundary Scan. Extest toggle command toggles the boundary scan value to the output pins if the value in the boundary scan cell is "1", and drives constant "0" if the value in boundary scan cell is "0".<br>**Exception:** For USB2 and OPI pins: a value of "0" in the boundary scab cell will enable EXTEST toggle, whereas a value of "1" will drive a constant output on to the pins. |

### 2.4.1.8 Offset 0Eh: EXTEST_PULSE - EXTEST PULSE Boundary Scan

| Bit | Type | Reset | Description |
|---|---|---|---|
| N | RW | 0 | **EXTEST_PULSE:** Command for boundary scan 1149.6. The EXTEST pulse command generates a pulse at the output pins. No 1149.6 pin behave like the EXTEST command. |

### 2.4.1.9 Offset 0Fh: EXTEST_TRAIN - EXTEST TRAIN Boundary Scan

| Bit | Type | Reset | Description |
|---|---|---|---|
| N | RW | 0 | **EXTEST_TRAIN:** Command for the boundary scan 1149.6 only. The EXTEST train command toggles the output pins. No 1149.6 pin behave like EXTEST command. |

### 2.4.1.10 Offset FFh: BYPASS - JTAG* Bypass Register

| Bit | Type | Reset | Description |
|---|---|---|---|
| N | RW | 0 | **BYPASS:** Selects the JTAG* bypass register. Connects the TDI to the TDO using a flop. The TDO value follows the TDI after one TCK clock during the data register shift. The data register captures this register and returns value of "0". |

§