

TI ICEPick TAP Router Module Type C

Reference Guide



Literature Number: SPRUH35
February 2011

Preface	7
1 Introduction	9
1.1 Debugging an Embedded Processor	10
1.1.1 Host	10
1.1.2 Target System	10
1.1.3 Debug Bridge	10
1.2 TI Target Interface	11
1.3 TAPs in the Device Scan Chain	12
1.4 Components of the ICEPick Module	13
1.4.1 Debug Connect Logic	13
1.4.2 System Control Logic	14
1.4.3 TAP Routing Logic	14
1.4.4 TAP Control Logic	14
1.4.5 Trigger Control Logic	14
2 Using the ICEPick TAP	17
2.1 Instructions Accepted by the ICEPick TAP	18
2.2 BYPASS Instruction: Bypassing the ICEPick TAP	18
2.3 ROUTER Instruction: Accessing Mapped Registers	18
2.3.1 Reading a Mapped Register	20
2.3.2 Writing to a Mapped Register	21
2.3.3 Mixing Reads and Writes	21
2.4 IDCODE Instruction: Identifying the TI Device	21
2.5 ICEPICKCODE Instruction: Identifying the ICEPick Module	21
2.6 CONNECT Instruction: Accessing the Debug Connect Register	22
2.6.1 Reading the Debug Connect Register	23
2.6.2 Writing to the Debug Connect Register	23
3 System Control	25
3.1 System Control Features	26
3.2 System Control Register	26
4 TAP Routing and Control	29
4.1 Selecting and Deselecting Secondary TAPs	30
4.2 Delay After Selecting a New TAP	30
4.3 Power and Clock Options for Secondary TAPs	30
4.4 Wait-in-Reset Mode for Secondary TAPs	31
4.5 Operation of TAP Signals Throughout the Scan Chain	31
4.5.1 nTRST	31
4.5.2 TMS	32
4.5.3 TDI and TDO	32
4.6 TAP Routing and Control Registers	32
4.6.1 Secondary Debug TAP Registers (SDTAP Registers)	32
4.6.2 Unimplemented TAP Register (TAP_UNIMP)	37
5 Reset Operations	39
5.1 Power-On Reset of the Target Device	40
5.2 Test-Logic Reset Initiated With the Master nTRST Signal	40

5.3	Warm Functional Reset of the Target Device	40
5.4	Warm Functional Reset of a Module in the Target Device	40

List of Figures

1-1.	Debugging an Embedded Processor	10
1-2.	Maximum Scan Chain	12
1-3.	Reduced Scan Chain	13
1-4.	Trigger Control Logic in the ICEPick Module	15
2-1.	Fields of the 32-Bit Scan Value for Accessing Mapped Registers	19
2-2.	Device Identification Register	21
2-3.	ICEPick Identification Register.....	22
2-4.	Debug Connect Register	22
3-1.	System Control Register.....	26
4-1.	Example of Selecting and Deselecting TAPs With the TAP Router	30
4-2.	Format of a Secondary Debug TAP Register (SDTAPx).....	33
4-3.	Unimplemented TAP Register (TAP_UNIMP)	37

List of Tables

1-1.	JTAG Port Pins	11
1-2.	14-Pin JTAG Header.....	11
2-1.	Instructions Accepted by the ICEPick TAP	18
2-2.	Mapped Register Selection	19
2-3.	Scan Values for Accessing Mapped Registers	19
2-4.	Device Identification Register Field Descriptions.....	21
2-5.	ICEPick Identification Register Field Descriptions	22
2-6.	Debug Connect Register Field Descriptions	22
3-1.	System Control Register Field Descriptions	26
4-1.	Field Descriptions for a Secondary Debug TAP Register (SDTAPx)	33
4-2.	Unimplemented TAP Register (TAP_UNIMP) Field Descriptions.....	37
5-1.	Boot (Power-On) TAP Configuration.....	40

Read This First

About This Manual

This document describes the system-level configuration features provided in TI's type-C ICEPick™ module. Among these features is the ability to pick which test access ports (TAPs) appear in the device scan chain.

Notational Conventions

This document uses the following conventions.

- In most cases, hexadecimal numbers are shown with the suffix h. For example, 40h is a hexadecimal 40 (decimal 64). Similarly, binary numbers often are shown with the suffix b. For example, 0100b is the decimal number 4 shown in binary form.
- JTAG® stands for Joint Test Access Group, the committee of the Institute of Electrical and Electronics Engineers (IEEE) that produced IEEE Standard 1149.1, *IEEE Standard Test Access Port and Boundary-Scan Architecture*. The following chapters use JTAG to refer to the TAP signals and functionality defined by this committee.

Introduction

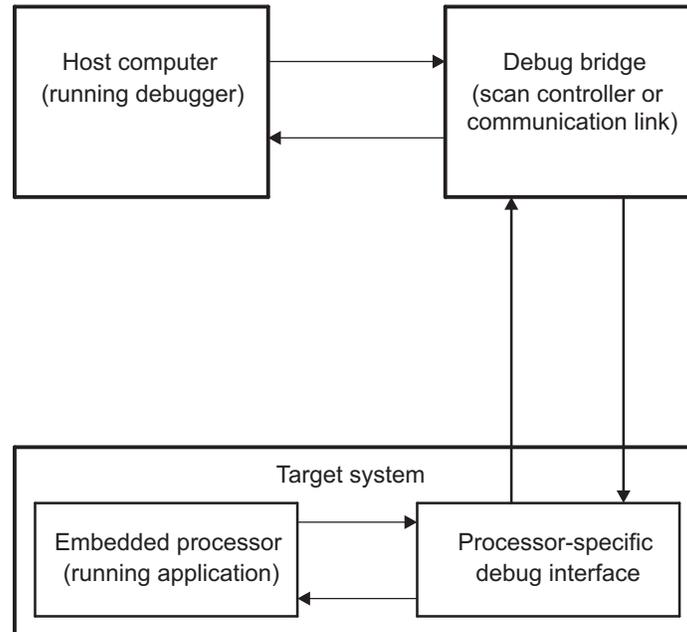
This chapter introduces embedded debug terminology used in this document and gives an overview of the standard TI target interface. It also shows the type of TI device scan chain controlled by the ICEPick module and introduces the functions performed by the ICEPick module.

Topic	Page
1.1 Debugging an Embedded Processor	10
1.2 TI Target Interface	11
1.3 TAPs in the Device Scan Chain	12
1.4 Components of the ICEPick Module	13

1.1 Debugging an Embedded Processor

An embedded processor is debugged in an environment that connects high-level debugging software, executing on a host computer (*host*), to a processor-specific debug interface in the *target system*. A *debug bridge* (a scan controller or communication link) facilitates communication between the host debugger and the target debug interface. The main components of such a system are shown in Figure 1-1 and described in the following subsections.

Figure 1-1. Debugging an Embedded Processor



1.1.1 Host

The host is a computer (for example, a PC) running a processor-specific software debugger as one of its applications. The host lets the user to issue high-level commands such as: set a breakpoint at location 6789h, step one instruction, or display the contents of memory from 1000h to 1048h.

1.1.2 Target System

The target system is defined as one or more devices to be debugged. Each device has its JTAG interface linked in a serial fashion, and each device can have one processor core or multiple cores. The core or cores contain in-circuit emulation (ICE) hardware designed explicitly to ease the task of debugging.

1.1.3 Debug Bridge

The debug bridge consists of a combination of hardware and software that connects the host to the target system. It uses one or more hardware interfaces and/or protocols to convert actions dictated by the debugger user to low-level commands and scans that exercise the ICE hardware. Typically, the debug bridge connects to the device through a TAP that is compatible with IEEE 1149.1.

1.2 TI Target Interface

In the case of TI target devices, the target interface includes:

- Six JTAG port pins. The JTAG port has the five standard IEEE 1149.1 interface pins (nTRST, TCK, TMS, TDI, and TDO) plus a return test clock pin (RTCK). These pins are described in [Table 1-1](#).
- Two JTAG port extension pins: EMU0 and EMU1. These multipurpose pins can be programmed for a number of debug functions, including emulation triggering.
- Up to 18 debug port pins. The debug port enables parallel exporting of ARM® trace data.

Some systems may have additional EMU signals for emulation, profiling, or tracing. The number of these instrumentation pins varies from device to device and depends on the debug features offered on the device.

[Table 1-1](#) lists the JTAP port pins.

Table 1-1. JTAG Port Pins

Pin	Type	Description
nTRST	Input	Test logic reset pin. When asserted (driven low), the nTRST signal causes all test and debug logic in the device to be reset along with the IEEE 1149.1 interface.
TCK	Input	Test clock pin. The TCK pin receives the test clock from the emulator or other debug bridge that connects the host to the target system.
RTCK	Output	Return test clock pin. Inside the target system, the TCK signal is synchronized and (if necessary) delayed before being used to time all JTAG scan activity. The resulting TCK signal can be monitored by the debug bridge through the RTCK pin.
TMS	Input	Test mode select pin. The TMS signal directs the next state of the TAP state-machine.
TDI	Input	Test data input pin. This is the input through which serial data enters the device.
TDO	Output	Test data output pin. This is the output through which serial data exits the device.

If a TI emulator is used as the debug bridge, the emulator is connected to the target system through a 14-pin header. This header is standard for all TI target devices and has the format shown in [Table 1-2](#). In addition to the JTAG port pins and the EMU0 and EMU1 pins, the header requires the target I/O voltage level (Vdd) and ground.

Table 1-2. 14-Pin JTAG Header

Name	Pin	Pin	Name
TMS	1	2	nTRST
TDI	3	4	GND ⁽¹⁾
PD (Vdd) ⁽²⁾	5	6	No pin (key)
TDO	7	8	GND ⁽¹⁾
RTCK	9	10	GND ⁽¹⁾
TCK	11	12	GND ⁽¹⁾
EMU0	13	14	EMU1

⁽¹⁾ Ground input/output

⁽²⁾ Target power supply input, used as power detect.

The system-level port must have JTAG-compatible signal timing and meet the following criteria if the standard TI debug products are to work with the target with no start-up problems:

- A logic zero (by a passive or active pulldown) applied to the nTRST header signal
- 10 kΩ pullups on EMU0 and EMU1
- A continuous scan path between TDI and TDO of the header
- A system power supply connection to PD that matches the I/O voltage of the JTAG signals and exceeds 0.5 V when system power is on and good. The system should supply no voltage when the system supply is off.

Regarding the TCK signal supplied to the header:

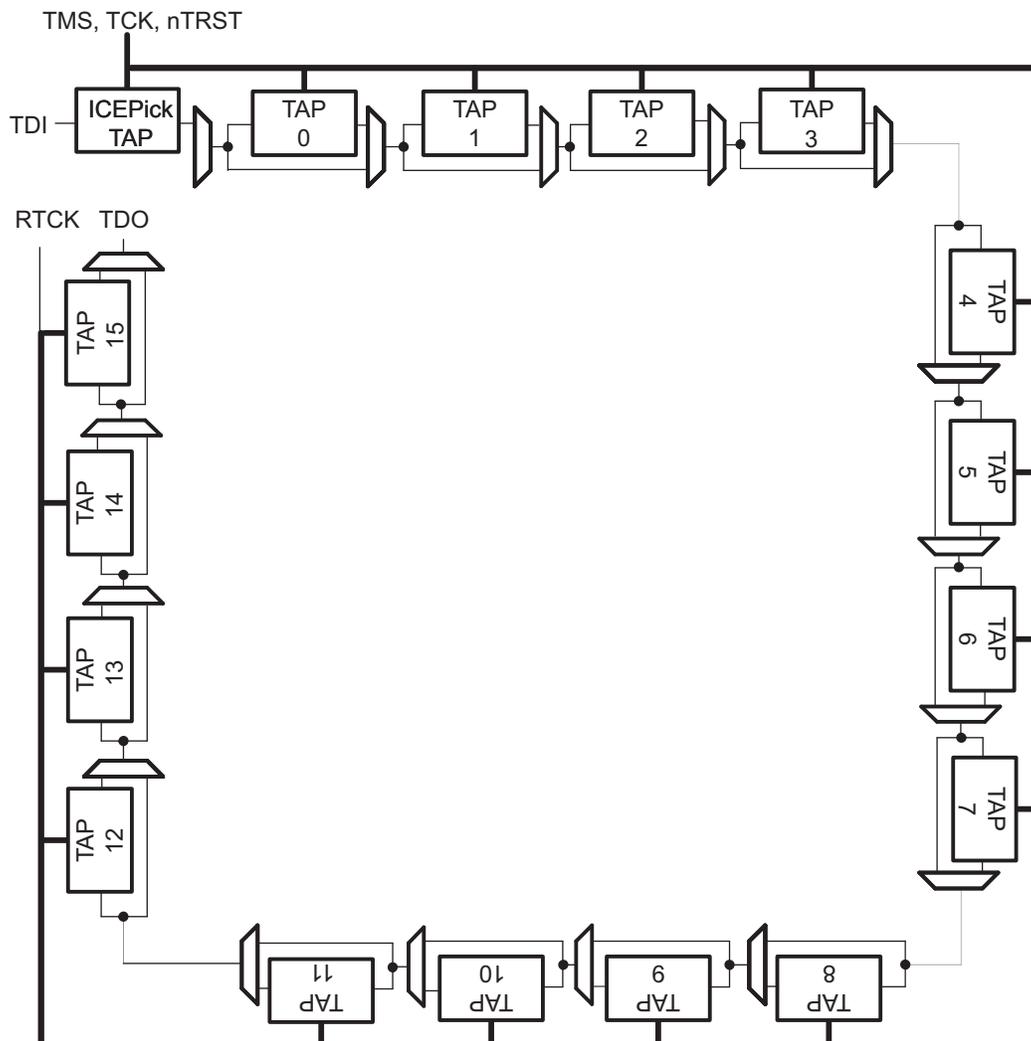
- Based upon the debug bridge, TCK can have any frequency from 0.5 to 50 MHz.
- The TCK frequency can be higher than the frequency of a functional clock in the target system. For example, an ARM core can be operating at 2 MHz while TCK is set to 40 MHz.
- A functional clock in the target system can have a frequency equal to or higher than the frequency of TCK.
- In cases when TCK is synchronized to an ARM functional clock, any decrease in the functional clock frequency also decreases the TCK frequency used throughout the device scan chain.

1.3 TAPs in the Device Scan Chain

Each debug-accessible module in the TI target device, including the ICEPick module, has a dedicated TAP in the device scan chain. The ICEPick TAP acts as the primary TAP for controlling other, secondary TAPs. A device can have up to 16 secondary TAPs; the device design determines how many are needed.

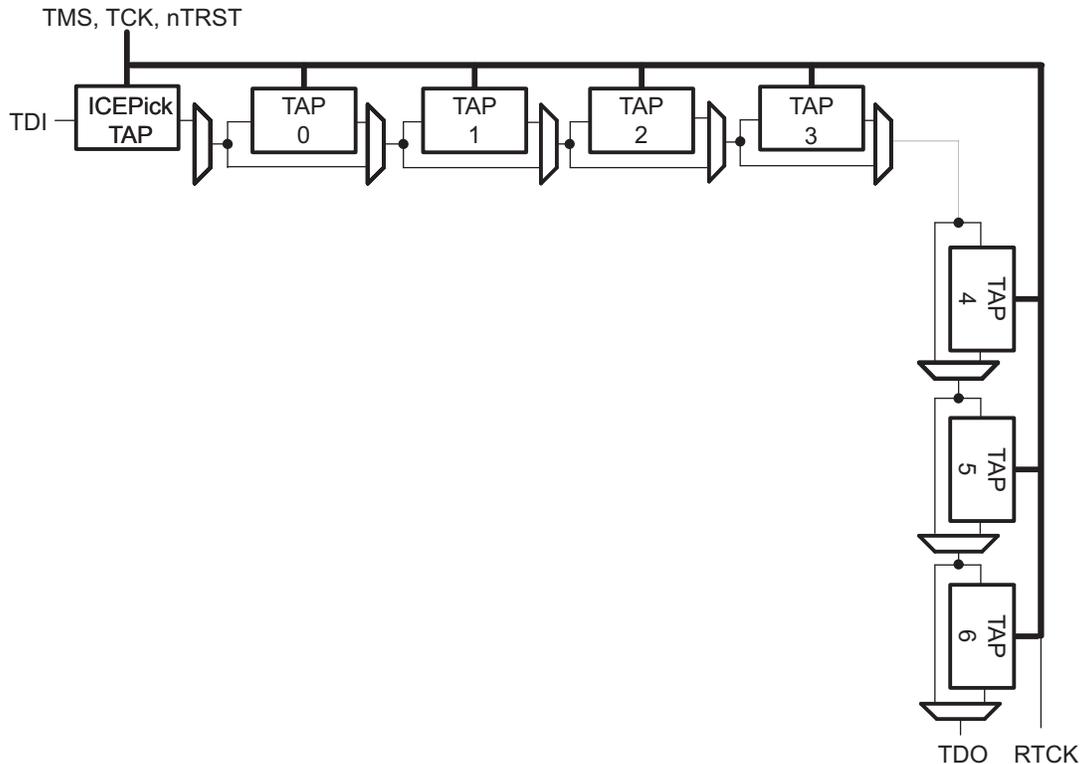
Figure 1-2 shows the maximum number of user-accessible TAPs in a TI device scan chain. The ICEPick TAP and 16 secondary TAPs are connected in series between the TDI and TDO pins. The secondary TAP closest to the ICEPick TAP is named 0 and the other TAPs are named sequentially as the scan chain continues toward the TDO pin. The TAP router in the ICEPick module controls the logic (shown as multiplexers) that includes or excludes each TAP.

Figure 1-2. Maximum Scan Chain



When fewer than 16 TAPs are needed for debugging and testing modules, the modules are assigned to a group of TAPs with the lowest numbers. Figure 1-3 shows an example in which only seven secondary TAPs were needed. TAPs 0 through 6 are implemented, and TAPs 7 through 15 are not.

Figure 1-3. Reduced Scan Chain



In any TAP configuration, the ICEPick TAP has a scan instruction length of 6 bits. For the other TAPs, the length of a scan instruction depends on the module associated with each TAP.

The RTCK signal from ICEPick may or may not be operating at the same frequency or phase as the TCK input signal, depending on which TAPs are selected, the input TCK, and the functional clock rate of each TAP. Some TAPs, such as those in ARM cores, synchronize TCK using their functional clocks. This may cause phase and/or frequency differences between the input TCK and the output RTCK.

1.4 Components of the ICEPick Module

The ICEPick module includes the following blocks of logic:

- Debug connect
- System control
- TAP routing
- TAP control
- Trigger control

1.4.1 Debug Connect Logic

The debug connect logic can be programmed to enable or disable debug accessibility to the registers that are accessed with the ROUTER scan instruction. The debug connect logic is programmed solely through the CONNECT scan instruction (see Section 2.6, *CONNECT Instruction: Accessing the Debug Connect Register*).

1.4.2 System Control Logic

The debugger can program the system control logic to use various system-level features. Two examples are keeping the ICEPick module powered at all times and initiating a system reset. For details about this block of logic, see [Chapter 3, System Control](#).

1.4.3 TAP Routing Logic

A TAP router in the ICEPick module can be programmed to deselect one or more of the secondary TAPs; that is, one or more of the secondary TAPs may be logically disconnected from the scan chain. When a TAP is deselected, the scan chain operates as if that TAP does not exist.

The default after a power-on reset (POR) is for the ICEPick TAP to be the only TAP logically connected between the TDI and TDO pins. As described in [Section 5.1, Power-On Reset of the Target Device](#), the EMU1 and EMU0 pins can be driven during a POR to choose an alternate TAP configuration.

The details about TAP routing are in [Chapter 4, TAP Routing and Control](#).

1.4.4 TAP Control Logic

The TAP control logic is used to manage and check clock, power, and reset states of the secondary TAPs. As with the TAP routing logic, the TAP control logic is described in [Chapter 4, TAP Routing and Control](#).

1.4.5 Trigger Control Logic

The trigger control logic acts without user intervention and is not user-programmable. Conceptually, the trigger control logic can be viewed as shown in [Figure 1-4](#). It manages two emulation triggers: trigger 0 and trigger 1. Each of these trigger can be used independently to generate:

- Pulses between an internal processor and an EMU pin (external triggering).
- Pulses between processors inside the device (internal triggering).

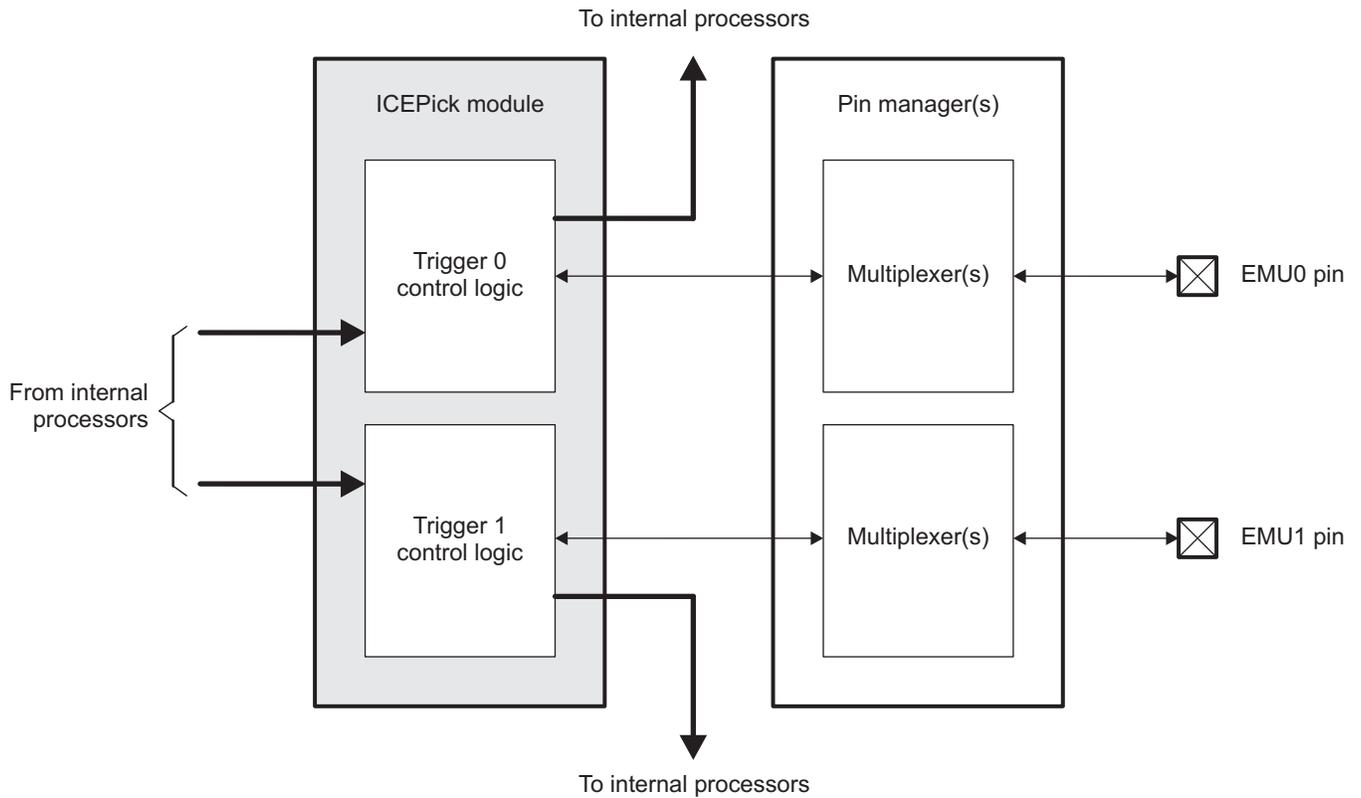
Each secondary TAP can support two trigger lines. If they are supported, one line is connected to trigger 0 and the other line to trigger 1. If the module associated with the TAP does not support triggering, or if the module is powered down, its trigger connections are kept inactive. Triggering within a secondary TAP occurs regardless of whether the TAP is selected in the device scan chain.

Before an EMU pin can be used for external triggering, the pin must be connected to the associated trigger by one or more pin managers outside the ICEPick module. Trigger 0 is associated with the EMU0 pin; trigger 1 is associated with the EMU1 pin. By default, EMU0 and EMU1 are configured to be trigger inputs.

When an internal processor generates a trigger, the trigger control logic distributes it to the other internal processor(s). In addition, if the EMU pin is configured for output triggering, the trigger control logic distributes the trigger to the EMU pin as an active-low output. When a trigger is received through the EMU pin as an input, the trigger is distributed to the internal processors.

The EMU device pins are bidirectional pins. Even when an output signal is driven to the pin, the level on the pin is sensed as an input.

Figure 1-4. Trigger Control Logic in the ICEPick Module



Using the ICEPick TAP

The ICEPick TAP accepts scanned instructions from the debugger and is used by the debugger to read and program all of the registers in the ICEPick module. This chapter describes the available instructions and explains how to access the registers.

Topic	Page
2.1 Instructions Accepted by the ICEPick TAP	18
2.2 BYPASS Instruction: Bypassing the ICEPick TAP	18
2.3 ROUTER Instruction: Accessing Mapped Registers	18
2.4 IDCODE Instruction: Identifying the TI Device	21
2.5 ICEPICKCODE Instruction: Identifying the ICEPick Module	21
2.6 CONNECT Instruction: Accessing the Debug Connect Register	22

2.1 Instructions Accepted by the ICEPick TAP

The ICEPick TAP has a 6-bit instruction register (IR) to hold the current instruction. This TAP can execute the instructions shown in [Table 2-1](#). Before using the ROUTER instruction, the debugger must use the CONNECT instruction to put the ICEPick module in the *connected state*. Otherwise, the ICEPick module is in the *disconnected state*, and the ROUTER instruction is executed as a BYPASS instruction. Instructions other than the ROUTER instruction operate normally regardless of whether the ICEPick module is in the connected state. All reserved instructions act as bypass but should not be used.

Table 2-1. Instructions Accepted by the ICEPick TAP

Opcode in IR	Instruction	Debug Connection Required For Execution?
000000b	Reserved	No
000001b	Reserved (acts as BYPASS)	–
000010b	ROUTER	Yes
000011b	Reserved (acts as BYPASS)	–
000100b	IDCODE	No
000101b	ICEPICKCODE	No
000110b	Reserved (acts as BYPASS)	–
000111b	CONNECT	No
001000b–111110b	Reserved (do not use)	–
111111b	BYPASS	No

In the shift state, the 6 least-significant bits (LSBs) of the ICEPick shift register are selected as the serial path between the TDI and TDO signals. At the CAPTURE IR state, 000001b is loaded into the shift register. In the SHIFT IR state, the contents are shifted out with the LSB first, while a new instruction is shifted in, also with the LSB first. In the UPDATE IR state, the new value in the shift register is loaded into IR so that it becomes the current instruction. When the ICEPick TAP is placed in the TEST LOGIC RESET (TLR) state (after a POR, after the assertion of the nTRST signal, or after five test-clock cycles with the TMS signal high), the IDCODE instruction becomes the current instruction.

2.2 BYPASS Instruction: Bypassing the ICEPick TAP

The BYPASS instruction is used when the debugger wants to shift in data or commands to another TAP on the scan chain without disturbing the ICEPick TAP. When the TAP executes the BYPASS instruction, the LSB of the shift register becomes a 1-bit bypass register between TDI and TDO in the SELECT DR state. In the first CAPTURE DR state, the bypass register is automatically loaded with a 0. As a result, the bit exported in the SHIFT DR state is a 0. Each bit shifted in through TDI is transferred out through TDO one test-clock cycle later. The contents of the shift register are ignored in the UPDATE DR state.

2.3 ROUTER Instruction: Accessing Mapped Registers

When the ICEPick module is in the connected state, the ROUTER instruction lets the debugger access specially mapped, 24-bit registers. The features provided by these registers are explained in [Chapter 3, System Control](#), and [Chapter 4, TAP Routing and Control](#). If the ICEPick module is in the disconnected state, the ROUTER opcode is interpreted as a BYPASS opcode.

For the ROUTER instruction, all 32 bits of the ICEPick data shift register are placed between TDI and TDO in the SELECT DR state. In the SHIFT DR state, the 32 bits shifted in must have the format shown in [Figure 2-1](#). When a value is scanned in, bit 31 indicates whether the register is to be read or written, bits 30–24 indicate which register is to be accessed, and bits 23–0 contain the data (if any). When a value is scanned out, bit 31 indicates whether the previous write succeeded (0) or failed (1), bits 30–24 indicate which register was read, and bits 23–0 contain the data.

A write to a secondary debug TAP register can fail for a number of reasons, including the following:

- The ICEPick module is in the disconnected state.
- The write was made to a reserved address.
- A previously programmed change to reset-control bits has not been processed yet.

Table 2-2 lists the registers that correspond to the possible REG values in the input scan. Table 2-3 lists the 32-bit values that must be shifted in for reading from and writing to each of the available registers. Of course, shifting in the correct 32-bit value is only part of the register-access sequence. See Section 2.3.1, *Reading a Mapped Register*, for the full procedure for reading a register, and Section 2.3.2, *Writing to a Mapped Register*, for the procedure for writing to a register.

Figure 2-1. Fields of the 32-Bit Scan Value for Accessing Mapped Registers

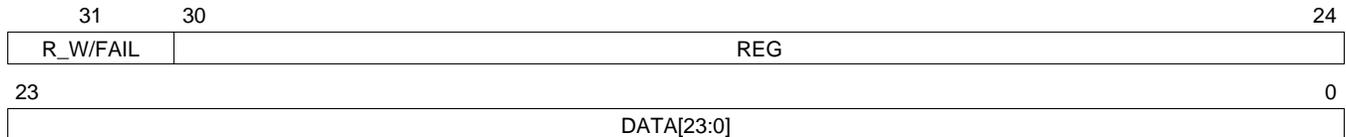


Table 2-2. Mapped Register Selection

REG	Register
000 0000b	All zeros
000 0001b	System control register (SYS_CNTL)
000 0010b–001 1111b	Reserved
010 0000b	Secondary debug TAP 0 register (SDTAP0)
010 0001b	Secondary debug TAP 1 register (SDTAP1)
010 0010b	Secondary debug TAP 2 register (SDTAP2)
010 0011b	Secondary debug TAP 3 register (SDTAP3)
010 0100b	Secondary debug TAP 4 register (SDTAP4)
010 0101b	Secondary debug TAP 5 register (SDTAP5)
010 0110b	Secondary debug TAP 6 register (SDTAP6)
010 0111b	Secondary debug TAP 7 register (SDTAP7)
010 1000b	Secondary debug TAP 8 register (SDTAP8)
010 1001b	Secondary debug TAP 9 register (SDTAP9)
010 1010b	Secondary debug TAP 10 register (SDTAP10)
010 1011b	Secondary debug TAP 11 register (SDTAP11)
010 1100b	Secondary debug TAP 12 register (SDTAP12)
010 1101b	Secondary debug TAP 13 register (SDTAP13)
010 1110b	Secondary debug TAP 14 register (SDTAP14)
010 1111b	Secondary debug TAP 15 register (SDTAP15)
011 0000b–111 1111b	Reserved

Table 2-3. Scan Values for Accessing Mapped Registers

To Perform This Access ...	Scan In This 32-Bit Value ...	
	R_W:REG	DATA
SYS_CNTL read	01h	24 don't care bits
SYS_CNTL write	81h	24-bit write value
SDTAP0 read	20h	24 don't care bits
SDTAP0 write	A0h	24-bit write value
SDTAP1 read	21h	24 don't care bits
SDTAP1 write	A1h	24-bit write value
SDTAP2 read	22h	24 don't care bits
SDTAP2 write	A2h	24-bit write value

Table 2-3. Scan Values for Accessing Mapped Registers (continued)

To Perform This Access ...	Scan In This 32-Bit Value ...	
	R_W:REG	DATA
SDTAP3 read	23h	24 don't care bits
SDTAP3 write	A3h	24-bit write value
SDTAP4 read	24h	24 don't care bits
SDTAP4 write	A4h	24-bit write value
SDTAP5 read	25h	24 don't care bits
SDTAP5 write	A5h	24-bit write value
SDTAP6 read	26h	24 don't care bits
SDTAP6 write	A6h	24-bit write value
SDTAP7 read	27h	24 don't care bits
SDTAP7 write	A7h	24-bit write value
SDTAP8 read	28h	24 don't care bits
SDTAP8 write	A8h	24-bit write value
SDTAP9 read	29h	24 don't care bits
SDTAP9 write	A9h	24-bit write value
SDTAP10 read	2Ah	24 don't care bits
SDTAP10 write	AAh	24-bit write value
SDTAP11 read	2Bh	24 don't care bits
SDTAP11 write	ABh	24-bit write value
SDTAP12 read	2Ch	24 don't care bits
SDTAP12 write	ACH	24-bit write value
SDTAP13 read	2Dh	24 don't care bits
SDTAP13 write	ADh	24-bit write value
SDTAP14 read	2Eh	24 don't care bits
SDTAP14 write	A Eh	24-bit write value
SDTAP15 read	2Fh	24 don't care bits
SDTAP15 write	AFh	24-bit write value

2.3.1 Reading a Mapped Register

To read a mapped register through the ICEPick TAP:

1. Shift in the ROUTER instruction.
2. While in the SHIFT DR state, shift in the 32 bits of register-access information. Bit 31 must be 0 for a read, bits 30–24 select a register, and bits 23–0 are don't care bits.
3. When the TAP advances to the UPDATE DR state, the register-access information is decoded.
4. When the TAP advances to the CAPTURE DR state, the result of a prior scan operation is loaded into the shift register.
5. In the SHIFT DR state, the content of the shift register is shifted out. While the value is being shifted out, new register-access information (for a read or a write) can be shifted in.
6. Perform another DR scan to capture and shift out the content of the register.

Multiple registers can be read in sequence without the need to scan in another ROUTER instruction.

2.3.2 Writing to a Mapped Register

To load a mapped register through the ICEPick TAP:

1. Scan in the ROUTER instruction.
2. In the SHIFT DR state, scan in the 32 bits of register-access information. Bit 31 must be 1, bits 30–24 select a register, and bits 23–0 must contain the data to be written.
3. When the TAP advances to the UPDATE DR state, the register-access information is decoded and the data is written to the register.

Multiple registers can be loaded in sequence without the need to scan in another ROUTER instruction. In this case, after each value is written, the output of the subsequent scan shows the value written.

2.3.3 Mixing Reads and Writes

Reads and writes can be mixed in any order while the ROUTER opcode is in IR. During each DR scan, bit 31 of the register-access information determines whether the next operation is a read or a write.

2.4 IDCODE Instruction: Identifying the TI Device

The IDCODE instruction commands the ICEPick TAP to read and export the content of the device identification register. This 32-bit register value is available through scans only. [Figure 2-2](#) and [Table 2-4](#) summarize the the format and fields of the register.

For the IDCODE instruction, all 32 bits of the ICEPick data shift register are placed between TDI and TDO in the SELECT DR state. The 32-bit device ID code is loaded into the shift register in the CAPTURE DR state. This code is shifted out in the SHIFT DR state with the LSB, while a don't-care value is shifted in. The shifted-in data is ignored in the UPDATE DR state. The IDCODE instruction is the default instruction selected by a test-logic reset.

Figure 2-2. Device Identification Register

31	28 27	12	11	1	0
VERSION	PARTNUMBER	MANUFACTURER		RESERVED	
R-ds	R-ds	R-017h			

LEGEND: R = Read; -n = Value after reset; -ds = Value after reset is device specific

Table 2-4. Device Identification Register Field Descriptions

Bit	Field	Value	Description
31:28	VERSION	0h–Fh	4-bit number for the version of the device. The value 0h would indicate the first version, 1h the second version, 2h the third version, and so on.
27:12	PARTNUMBER	0000h–FFFFh	16-bit part number for the device
11:1	MANUFACTURER	017h	11-bit company code for the device. For TI this number is fixed at 017h.
0	RESERVED	1	This bit is always 1.

2.5 ICEPICKCODE Instruction: Identifying the ICEPick Module

The debugger uses the ICEPICKCODE instruction to read the content of the ICEPick identification register. This register indicates the version and features of the ICEPick module. Its 32 bits are divided into the fields shown in [Figure 2-3](#) and described in [Table 2-5](#). A JTAG scan is the only way to access this register.

For the ICEPICKCODE instruction, all 32 bits of the ICEPick data shift register are placed between TDI and TDO in the SELECT DR state. The 32-bit ICEPick ID code is loaded into the shift register in the CAPTURE DR state. This code is shifted out in the SHIFT DR state with the LSB first, while a don't-care value is shifted in. The shifted-in data is ignored in the UPDATE DR state.

Figure 2-3. ICEPick Identification Register

31	28 27	24 23	20 19	16
MAJORVERSION		MINORVERSION		EMUTAPS
R-ds		R-ds		R-ds
15				4 3
ICEPICKTYPE				CAPABILITIES
R-1CCh				R-ds

LEGEND: R = Read; -n = Value after reset; -ds = Value after reset is device specific

Table 2-5. ICEPick Identification Register Field Descriptions

Bit	Field	Value	Description
31:28	MAJORVERSION	0h–Fh	Major version of the ICEPick module. This field value is updated each time ICEPick undergoes a major architectural change.
27:24	MINORVERSION	0h–Fh	Minor version of the ICEPick module. This field value is updated each time ICEPick undergoes a minor architectural change.
23:20	TESTTAPS	0	Number of secondary test-only TAPs. The value 0 indicates that the ICEPick module can support up to 16 of these TAPs. Test-only TAPs are not user-accessible.
19:16	EMUTAPS	0	Number of secondary emulation/debug TAPs. The value 0 in this field indicates that the ICEPick module can support up to 16 of these TAPs. Some or all of the implemented TAPs of this type let the user debug modules in the TI device.
15:4	ICEPICKTYPE	1CCh	ICEPick type. The value 1CCh indicates a type-C ICEPick module.
3:0	CAPABILITIES	3h	ICEPick capabilities. Bits 3 and 2 are reserved and return 0 when read. Bit 1 contains 1 to indicate that the clock-voting capability is instantiated to support adaptive clocking. Bit 0 contains 1 to indicate that the reduced TCK mode is instantiated.

2.6 CONNECT Instruction: Accessing the Debug Connect Register

The CONNECT instruction is used to read or modify the 4 LSBs of the debug connect register (see [Figure 2-4](#) and [Table 2-6](#)). The four bits must be changed together, not individually. When the 4 LSBs are 1001b, the ICEPick module is in the connected state. If the 4 LSBs are any other value, the ICEPick module is in the disconnected state, and the ICEPick TAP interprets the ROUTER opcode as a BYPASS opcode. A power-on reset or a test-logic reset forces the 4 LSBs to 0110b, which selects the disconnected state.

Figure 2-4. Debug Connect Register

7	6	4 3	0
WRITEENABLE		CONNECTKEY	
R/W		R-0110b	

LEGEND: R = Read; -n = Value after power-on reset or test-logic reset

Table 2-6. Debug Connect Register Field Descriptions

Bit	Field	Value	Description
7	WRITEENABLE		
6:4	RESERVED	0	These reserved bits return 0s when read.
3:0	CONNECTKEY	0000b–1111b	Debug connect key. Write 1001b to put the ICEPick module in the connected state (ROUTER opcode enabled). Any other value written to this register puts the ICEPick module in the disconnected state (ROUTER opcode treated as BYPASS opcode).

Writing XXX1001b to the debug connect register asserts a debug connect signal, and writing any other value deasserts the debug connect signal. Other points about DCON are:

- DCON does not affect the execution flow of the application.
- When the ICEPick TAP enters the TEST LOGIC RESET state, DCON is asynchronously deasserted.
- When the disconnected state is caused by cable removal, a test-logic reset must be initiated with the nTRST pin to deassert DCON.

2.6.1 Reading the Debug Connect Register

To read the debug connect register:

1. Shift in the CONNECT instruction.
2. While in the SHIFT DR state, shift in the 8 bits of register-access information. Because this is a read, bit 7 must be 0, and bits 6–0 are don't care bits.
3. When the TAP advances to the UPDATE DR state, the register-access information is decoded.
4. When the TAP advances to the CAPTURE DR state, the result of a prior scan operation is loaded into the shift register.
5. In the SHIFT DR state, the content of the shift register is shifted out.
6. Perform another DR scan to capture and shift out the content of the debug connect register.

2.6.2 Writing to the Debug Connect Register

To write the key to the debug connect register:

1. Scan in the CONNECT instruction.
2. In the SHIFT DR state, shift in the 8 bits of register-access information LSB first. Bit 7 must be 1, bits 6–4 are don't care bits, and bits 3–0 must contain the key value to be written.
3. When the TAP advances to the UPDATE DR state, the register-access information is decoded and bits 6–0 are written to the register.

System Control

The ICEPick system control register enables a host processor to perform the system debug actions described in this chapter.

Topic	Page
3.1 System Control Features	26
3.2 System Control Register	26

3.1 System Control Features

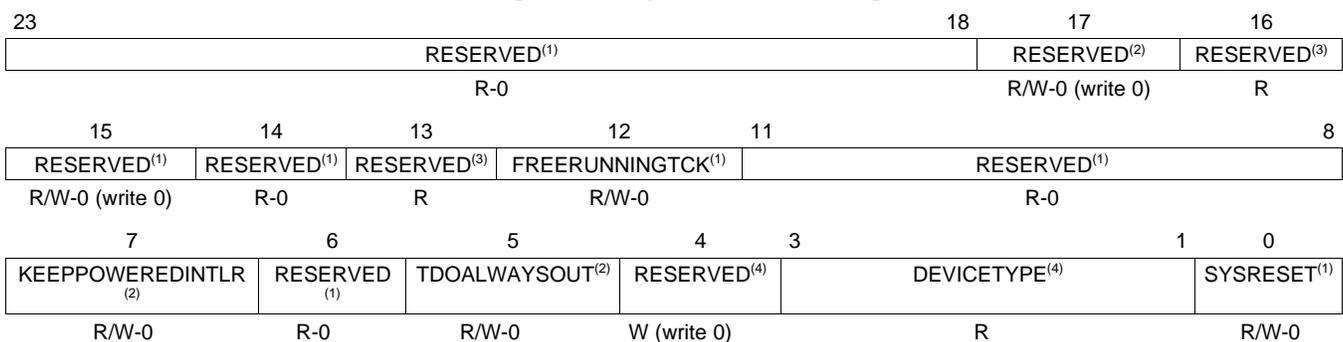
By programming the system control register, the debugger can:

- When adaptive clocking is needed for an ARM core, indicate whether the TCK signal from the debug bridge will be free-running or controlled.
- Keep the ICEPick logic powered even when the ICEPick TAP enters the TEST LOGIC RESET state.
- Force the TDO pin to be in the output state regardless of the TAP state.
- Read the factory-programmed device type.
- Initiate a system reset.

3.2 System Control Register

The fields of the system control register are shown in [Figure 3-1](#) and described in [Table 3-1](#).

Figure 3-1. System Control Register



LEGEND: R = Read; W = Write only; -n = Value after reset

⁽¹⁾ These bits are reset by a power-on reset or by a test-logic reset.

⁽²⁾ These bits are reset by a power-on reset only.

⁽³⁾ These bits are not affected by either of the resets, and their read values are undefined.

⁽⁴⁾ These bits are not affected by either of the resets, and their read values are fixed.

Table 3-1. System Control Register Field Descriptions

Bit	Field	Value	Description
23:17	RESERVED	0	These are read-only bits that return 0s when read.
17	RESERVED	0	Always write 0 to this reserved bit.
16	RESERVED	x	This is a read-only bit with an undefined read value.
15	RESERVED	0	Always write 0 to this reserved bit. Writing 1 to this bit selects a reduced TCLK mode on devices that support this feature. Normally the bit is left at 0.
14	RESERVED	0	This is a read-only bit that returns 0 when read.
13	RESERVED	x	This is a read-only bit with an undefined read value.
12	FREERUNNINGTCK		Set this bit if the emulator provides a free-running TCK. Clear this bit if an adapter clocking emulator is being used.
11:8	RESERVED	0	These are read-only bits that return 0s when read.
7	KEEPPOWEREDINTLR	0 1	Keep powered in the TEST LOGIC RESET (TLR) state. The ICEPick logic may be powered down in the TLR state. However, when the TLR state is a normal part of the debug flow, the KEEPPOWEREDINTLR bit can be set to prevent the system from removing power. Allow the ICEPick module to be powered down in the TEST LOGIC RESET state. Do not allow the ICEPick module to be powered down in the TEST LOGIC RESET state.
6	RESERVED	0	This is a read-only bit that returns 0 when read.
5	TDOALWAYSOUT		
4	RESERVED	0	Writing 1 forces the device to reset in secure mode if the device supports security features and the device-type is emulator or test.

Table 3-1. System Control Register Field Descriptions (continued)

Bit	Field	Value	Description
3:1	DEVICETYPE	000b 001b 010b 011b 1XXb	Device type. This field indicates the device type preset at the factory. Test Emulator: A device made for emulation. Secure: A device with security features. GP (general purpose). A device made for production. Reserved
0	SYSRESET	0 1	System reset. This bit gives the debugger control of the system functional reset. No effect Behaves as if the external chip reset has been momentarily asserted. This bit does not reset any emulation logic. SYSRESET is a self-clearing bit.

TAP Routing and Control

This chapter describes the block of TAP routing and control logic in the ICEPick module. Specifically, this block is responsible for selecting, controlling, and monitoring the TAPs in the device scan chain.

Topic	Page
4.1 Selecting and Deselecting Secondary TAPs	30
4.2 Delay After Selecting a New TAP	30
4.3 Power and Clock Options for Secondary TAPs	30
4.4 Wait-in-Reset Mode for Secondary TAPs	31
4.5 Operation of TAP Signals Throughout the Scan Chain	31
4.6 TAP Routing and Control Registers	32

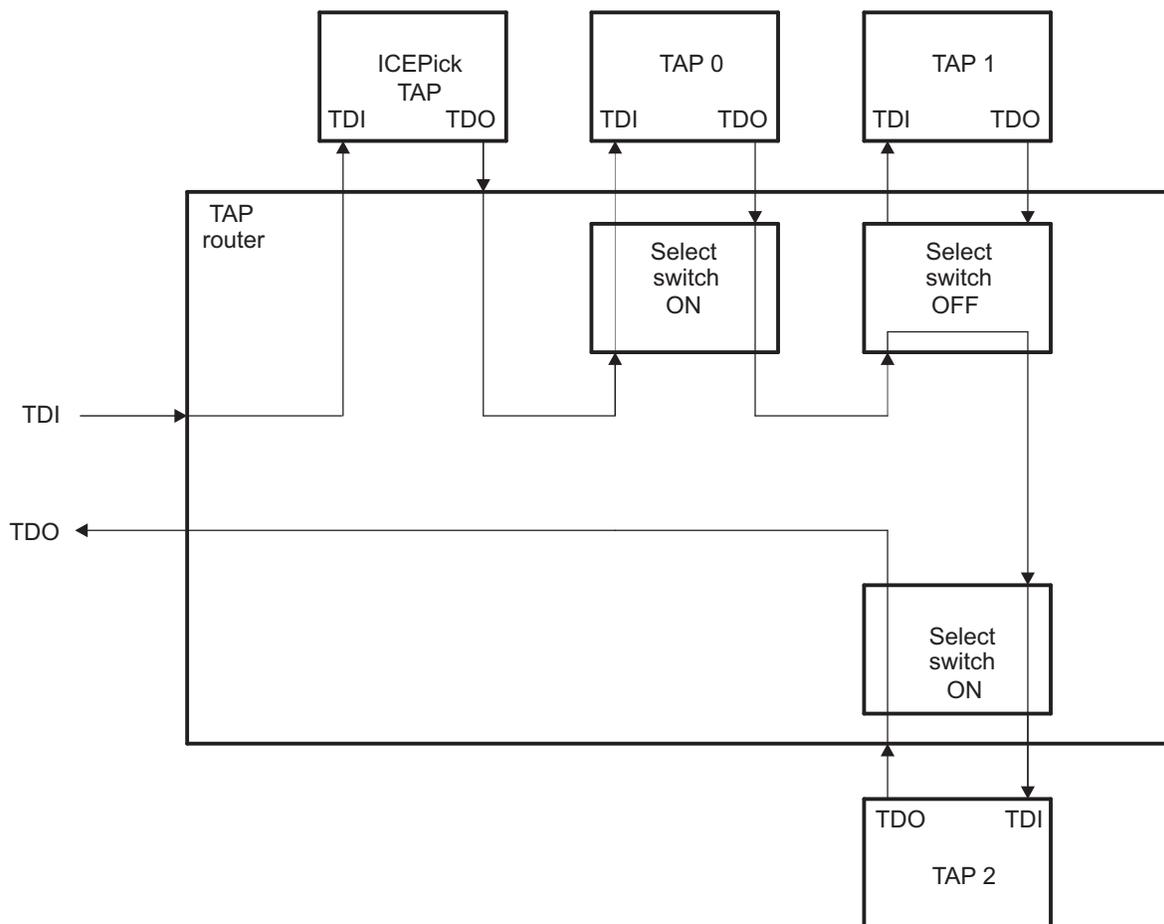
4.1 Selecting and Deselecting Secondary TAPs

Figure 1-2 shows an example of TAPs in a device scan chain. The ICEPick TAP and the external TAP signals comprise the master scan path. The ICEPick TAP is always present in the scan chain and is always closest to the TDI input of the device. The secondary TAPs are connected in series with the ICEPick TAP.

Through the ICEPick TAP, the TAP router is programmed to select or deselect each secondary TAP by writing to the TAPSELECT bit in the associated secondary debug TAP register. Deselecting unneeded TAPs reduces the number of bits in the scan chain and, therefore, reduces the normal time needed for scans. From the perspective of the external TAP signals, secondary TAPs that are not selected appear not to exist.

Figure 4-1 provides an example of TAP selection in a device with three secondary TAPs. The router in the figure is programmed to include TAPs 0 and 2, and to exclude TAP 1.

Figure 4-1. Example of Selecting and Deselecting TAPs With the TAP Router



4.2 Delay After Selecting a New TAP

When a new TAP is selected for inclusion in the scan chain, the debug bridge must hold the RUN TEST IDLE state for at least three test clock cycles before proceeding to any other state. This allows the previously selected TAPs and the newly selected TAP to become synchronized and for all to be in the RUN TEST IDLE state.

4.3 Power and Clock Options for Secondary TAPs

Each secondary debug TAP register (SDTAP_x) includes fields for controlling and monitoring power and clock states of the secondary TAP and its associated module:

- In the case when the TAP and its associated module are in different power domains, the TAPPOWER bit can be read to determine whether the power to the TAP is on or off.
- The POWERDOWNDESIRED bit indicates whether the module has been requested to turn its power off.
- The POWER bit indicates whether the module power is on or off.
- The POWERLOSSDETECTED bit is set and latched when the module power goes off.
- The INHIBITSLEEP bit controls whether the module clock and power can be turned off.
- The FORCEPOWER bit can be used to force the module power to turn on or stay on.
- The CLOCKDOWNDESIRED bit indicates whether the module has been requested to turn its clock off.
- The CLOCK bit indicates whether the module clock is on or off.
- The FORCEACTIVE bit can be used to force the module clock and power to turn on or stay on.

4.4 Wait-in-Reset Mode for Secondary TAPs

A wait-in-reset mode is implemented independently for each secondary TAP (x) through the associated TAP register (SDTAPx). In the wait-in-reset mode, if the module associated with the TAP enters its reset state, the module is held in the reset state, even after the source of the reset is deasserted. This extension of the reset state allows the debugger to:

- Gain emulation control of any processor at POR for the power domain of the processor.
- Capture and extend system-generated functional resets while the target device is running under control of the application or the debugger.
- Hold an entire power domain in reset until emulation control of the subsystem can be established.
- Stall the entire system while reset is extended to a power domain.
- Make the reset extension visible external to the power domain.
- Provide debug execution of code from the first cycle of execution.
- Prevent processor execution of random instructions in uninitialized program memory at power up.
- Download code before any code execution occurs.
- Coordinate debug initialization across multiple cores before code execution begins.

To start the wait-in-reset mode, the debugger must write 001b to the RESETCONTROL bit of SDTAPx. This action does not assert reset itself, but when reset is asserted by normal means, the module is held in the reset state until the debugger releases it. To bring the module out of the reset state, the debugger must set the RELEASEFROMWIR bit to 1. If the wait-in-reset mode is not the reason for the extended reset, setting this bit to 1 does not end the reset state.

The write to the RELEASEFROMWIR bit does not disable the wait-in-reset mode. If RESETCONTROL remains 001b, the next reset will trigger another reset extension. The wait-in-reset mode is disabled only when the debugger writes 00b to the RESETCONTROL bit.

4.5 Operation of TAP Signals Throughout the Scan Chain

The TAP pins at the boundary of the TI device include the five signals defined in IEEE 1149.1: TMS, TDI, TDO, nTRST, and TCK. Inside the device, these master signals are connected to the corresponding signals at the ICEPick TAP and the secondary TAPs. This section describes how the master TMS, TDI, TDO, and nTRST signals affect the signals at the internal TAPs.

4.5.1 nTRST

The master nTRST signal is used to asynchronously assert a test-logic reset.

The ICEPick TAP state always changes with respect to the master nTRST signal. When TRST is driven low, the ICEPick TAP is immediately changed to the TEST LOGIC RESET state, even if the test clock is off. This has the same effect as entering the TEST LOGIC RESET state through a sequence of scans (five test-clock cycles with TMS high). The ICEPick register bits that are affected by nTRST are returned to their default values. The ICEPick TAP state-machine is reset. When the rising edge of nTRST is detected, some TAPs can be automatically selected for inclusion in the scan chain based upon the signal levels on the EMU0 and EMU1 pins (see [Section 5.1](#), POWER-ON RESET OF THE TARGET DEVICE, for details).

All secondary TAPs that are selected respond to the master nTRST signal under all conditions. This is needed to ensure that all selected TAPs advance through states in unison.

4.5.2 TMS

The master TMS signal is used to drive all selected TAPs through sequences of states in unison. The TMS signal for each selected TAP follows the master TMS signal. A POR forces all TMS signals high, moving all TAPs to the TEST LOGIC RESET state and holding them there.

When the master nTRST signal is asserted low, the nTRST signal for each selected TAP is also driven low. It is expected that the external debug bridge will hold the master TMS signal (and by extension, the internal TMS signals) high when the master nTRST signal returns high. This is a recommendation of the IEEE 1149.1 specification to avoid metastability issues. It ensures that the test logic remains in the TEST LOGIC RESET state for at least one cycle after nTRST returns high.

4.5.3 TDI and TDO

The ICEPick TAP and the selected secondary TAPs are linked in series. The master TDI signal is buffered onto the TDI signal of the ICEPick TAP. The TDO signal of the ICEPick TAP is buffered onto the TDI signal of the next secondary TAP that is selected. In a similar fashion, the TDO signal of this TAP is connected to the TDI signal of the next selected TAP. The TDO signal of the last selected TAP is buffered onto the master TDO signal.

The ICEPick TAP is always connected closest to the device TDI before any secondary TAPs. When a TAP is deselected, its TDI signal is held high, and its test-clock signal is held low. The TDO signal from each deselected TAP is ignored.

4.6 TAP Routing and Control Registers

This section describes the registers of the TAP routing and control block inside the ICEPick module.

4.6.1 Secondary Debug TAP Registers (SDTAP Registers)

Each implemented TAP in the OMAP2430 scan chain has a dedicated register that lets the debugger perform the following tasks:

- Control the reset operation of the TAP.
- Include the TAP in the scan chain or exclude it.
- Control and monitor the power state of the TAP and its associated module.
- Control and monitor the clock state of the TAP and its associated module.
- Verify the presence of the TAP.
- Select whether the debug logic of the module can be turned off.
- Recognized when the module has entered a state that requires the attention of the debugger.
- Specify the debug mode of the module upon POR.

In addition, the register contains 2 status bits, DEBUGCONNECT and DEBUGSTATE. The fields of a TAP control and status register are shown in [Figure 4-2](#) and described in [Table 4-1](#).

Figure 4-2. Format of a Secondary Debug TAP Register (SDTAPx)

23	22	21	20	19	18	17	16	
RESERVED	RESERVED ⁽⁴⁾	POWERLOSS DETECTED ⁽¹⁾	INHIBIT SLEEP ⁽¹⁾	TAP POWER ⁽²⁾	UNNATURAL RESET ⁽²⁾	INRESET/ RELEASEFROMWIR ⁽¹⁾	RESET CONTROL ⁽¹⁾	
R/W-0	R	R/W-0	R/W-0	R	R	R/W-0	R/W-0	
		14	13	12	11	10	9	8
RESET CONTROL ⁽¹⁾		DEBUG CONNECT ⁽¹⁾		DEBUG MODE ⁽¹⁾		DEBUG ATTENTION ⁽²⁾	TAP VISIBLE ⁽²⁾	TAP SELECT ⁽¹⁾
R/W-0		R/W-0		R/W-0		R	R	R/W-boot
7	6	5	4	3	2	1	0	
POWERDOWN DESIRED ⁽²⁾	FORCE POWER ⁽¹⁾	POWER ⁽²⁾	CLOCKDOWN DESIRED ⁽²⁾	FORCE ACTIVE ⁽¹⁾	CLOCK ⁽²⁾	TAP ACCESSABLE	TAP PRESENT ⁽⁴⁾	
R	R/W-0	R	R	R/W-0	R	R	R	

LEGEND: R = Read; W = Write; -n = Value after reset; -boot = Value after reset depends on the selected boot mode

⁽¹⁾ These bits are reset by a POR or by a test-logic reset.

⁽²⁾ These bits are not affected by either of the resets, and their values depend on the state of the TAP or its associated module.

⁽³⁾ This bit is not affected by either of the resets, and its read value is undefined.

⁽⁴⁾ These bits are not affected by either of the resets, and their read values are fixed.

Table 4-1. Field Descriptions for a Secondary Debug TAP Register (SDTAPx)

Bit	Field	Value	Description
23	RESERVED (test only)		
22	RESERVED	0	This read-only bit returns 0 when read.
21	POWERLOSSDETECTED	0 1	<p>Power loss detected. This bit indicates the power state of the module associated with this TAP.</p> <p>When reading:</p> <p>0 The module power is on.</p> <p>1 The module power is off or has been off. The POWERLOSSDETECTED bit remains 1 until the debugger writes 1 to reset it.</p> <p>When writing:</p> <p>0 No effect</p> <p>1 If this bit is latched at 1, reset it to 0.</p>
20	INHIBITSLEEP	0 1	<p>Inhibit sleep. This bit controls whether the module associated with this TAP can be put to sleep. The INHIBITSLEEP bit applies whether the TAP is selected or deselected. The value read does not reflect the value written until the power and clock controller has acted upon a change to the INHIBITSLEEP bit.</p> <p>0 Do not inhibit sleep. The module clock and power settings follow the normal application settings, unless an emulation module other than the ICEPick module is affecting them.</p> <p>1 Inhibit sleep. This bit affects the module clock state and power state individually but in similar ways.</p> <ul style="list-style-type: none"> • Clock state: If the clock is on when the INHIBITSLEEP bit is set, the clock cannot be turned off. If the clock is off when the INHIBITSLEEP bit is set, the clock can be turned on, but then it cannot be turned off. (Although the application cannot turn the clock off, it can change the frequency of the clock.) • Power state: If the power is on when the INHIBITSLEEP bit is set, the power cannot be turned off. If the power is off when the INHIBITSLEEP bit is set, the power can be turned on, but then it cannot be turned off. (The application can change the voltage level if the change would not stop the operation of the module.)

Table 4-1. Field Descriptions for a Secondary Debug TAP Register (SDTAP_x) (continued)

Bit	Field	Value	Description
19	TAPPOWER	0 1	<p>TAP power. In some cases, a TAP and its associated modules can be in separate power domains. In such situations, the TAP can have power even when the module does not, and the TAPPOWER bit must be consulted. To check the power state of the module, read the POWER bit (bit 5).</p> <p>The TAP is either not powered or not operational.</p> <p>The TAP is powered and operational.</p>
18	UNNATURALRESET	0 1	<p>Unnatural reset. This bit detects a functional warm reset inconsistency in the module associated with this TAP. Two examples follow:</p> <ul style="list-style-type: none"> The application is holding the module in reset, but emulation logic is blocking reset. The application has deasserted the reset signal, but emulation logic is extending the assertion of the reset signal. <p>The functional reset state of the module is consistent with the application settings.</p> <p>The functional reset state of the module is not consistent with the application settings due to emulation actions or settings.</p>
17	INRESET/ RELEASEFROMWIR	0 1 0 1	<p>In reset/release from wait-in-reset mode. This bit is used to detect the reset state of the module associated with this TAP and to release the module from an extended reset state.</p> <p>When reading (INRESET):</p> <p>The functional warm reset signal of the module is not asserted.</p> <p>The functional warm reset signal of the module is asserted.</p> <p>When writing (RELEASEFROMWIR):</p> <p>No effect</p> <p>If the module is in its reset state due to a wait-in-reset condition, release the module from reset, and clear the InReset/ReleaseFromWIR bit.</p>
16:14	RESETCONTROL	000b 001b 010b 011b 100b–111b	<p>Reset control. This bit field can be used to override the application controls of the functional warm reset to the module associated with this TAP.</p> <p>Normal reset mode. Reset operates under the normal control of the application or device controls.</p> <p>Wait-in-reset mode. When the module enters its reset state, it remains in the reset state. This setting alone does not cause the reset; it extends a reset once the reset is asserted. Bit 17 of this register can be used to release the module from the reset state.</p> <p>Block reset mode. The module is blocked from undergoing a functional warm reset. In this mode, only emulation reset controls affect the module.</p> <p>Assert and hold reset mode. The functional warm reset signal to the module is asserted and held active.</p> <p>Reserved (do not use)</p>
13	DEBUGCONNECT	0 1	<p>Debug connect. This bit indicates whether debug logic is enabled in the module associated with this TAP.</p> <p>The debug logic of the module is disabled. The debug logic can be turned off, allowing emulation events within the module to be ignored.</p> <p>The debug logic of the module is enabled. Emulation events within the module are acted upon.</p>

Table 4-1. Field Descriptions for a Secondary Debug TAP Register (SDTAP_x) (continued)

Bit	Field	Value	Description
12:11	DEBUGMODE	00b 01b 10b 11b	<p>Debug mode. If the module associated with this TAP is a processor, the DEBUGMODE bit field specifies the debug mode selected for the processor at power-on reset. Some processors may not support all of these modes and must translate the mode into the nearest equivalent.</p> <p>00b Default debug mode (the default mode for the processor)</p> <p>01b Monitor mode. Debug events cause interrupts rather than halt the processor. A portion of the application (the monitor) must handle debug events in response to the interrupts.</p> <p>10b Stop mode. When a debug event occurs, the instruction pipeline of the processor appears to halt. While the processor is stopped, all of its activity is under direct control of the debugger through scans at the JTAG port.</p> <p>11b Real-time mode. When a debug event occurs, the instruction pipeline of the processor appears to stop; however, a designated subset of interrupts can be serviced while the processor is halted.</p>
10	DEBUGATTENTION	0 1	<p>Debugger attention needed. If the module associate with this TAP is a processor, the DEBUGATTENTION bit indicates when the processor has entered a state that requires debugger interaction.</p> <p>0 The processor is in an execution state.</p> <p>1 The processor is in a debug state that requires debugger interaction.</p>
9	TAPVISIBLE	0 1	<p>TAP visible. This read-only bit indicates whether the TAP is now part of the device scan chain. The initial request is made by setting or clearing the TAPSelect bit. Then the TAPVisible bit is updated when the RUN TEST IDLE TAP state is reached.</p> <p>0 The TAP is deselected; it is logically excluded from the device scan chain.</p> <p>1 The TAP is selected; it is visible (logically included) in the scan chain.</p>
8	TAPSELECT	0 1	<p>TAP select. The value written to the SELECT bit determines whether the TAP will be selected or deselected in the next RUN TEST IDLE state. Selection or deselection is latched in the UPDATE DR state, and therefore the TAPSELECT bit shows the updated value even if the TAP has not reached the RUN TEST IDLE state.</p> <p>0 Deselect the TAP. When the deselection is complete, the TAPVISIBLE bit will be 0.</p> <p>1 Select the TAP. When the selection is complete, the TAPVISIBLE bit will be 1.</p>
7	POWERDOWNDESIRED	0 1	<p>Power down desired. This bit indicates whether the module associated with this TAP has been requested to power down by the application or by emulation logic outside the ICEPick module. If the FORCEPOWER bit is 1, the module stays active.</p> <p>0 The module has not received a request to power down.</p> <p>1 The module has received a request to power down.</p>

Table 4-1. Field Descriptions for a Secondary Debug TAP Register (SDTAP_x) (continued)

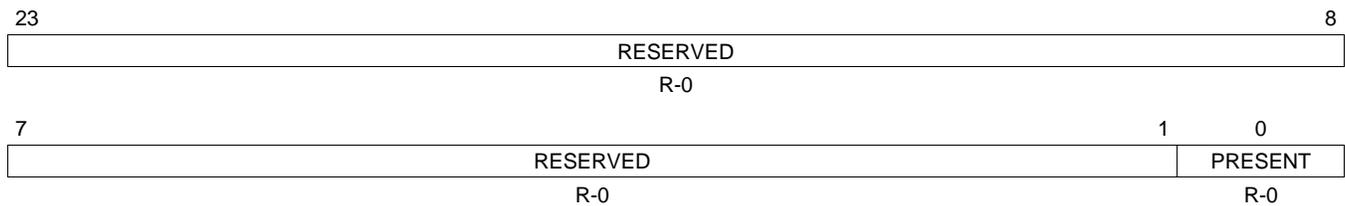
Bit	Field	Value	Description
6	FORCEPOWER	0 1	<p>Force power. This bit can allow the debugger to force power to the module associated with this TAP; it does not interfere with clock settings driven by the application. The FORCEPOWER bit applies whether the TAP is selected or deselected. The value read from the FORCEPOWER bit does not reflect the value written until the power controller has acted upon the value change.</p> <p>When reading:</p> <p>0 The power settings of the module follow the normal application settings, unless another emulation setting is affecting them.</p> <p>1 If the power is on when the FORCEPOWER bit is set, the power cannot be turned off. If the power is off when the FORCEPOWER bit is set, the power is automatically turned on, but then it cannot be turned off. (The application can change the voltage level of the module as long as the module remains operational.)</p> <p>When the power is automatically turned on, the module goes through its normal power-up sequence, including the release of the POR. The clock to the module remains off and the functional warm reset for the module remains active. When power is restored, the application can change the clock or reset conditions as needed.</p> <p>When writing:</p> <p>0 Return control of the power settings to the application and to other emulation logic.</p> <p>1 Force the module power to be active, and block attempts by the application to turn the module off.</p>
5	POWER	0 1	<p>Power status. This indicates the power state of the module associated with this TAP.</p> <p>0 The module is either not powered or not operational.</p> <p>1 The module is powered and operational.</p>
4	CLOCKDOWNDESIRED	0 1	<p>Clock down desired. This bit indicates whether the module associated with this TAP has been requested to stop its clock. The request can come from the application or from emulation logic outside the ICEPick module. If the FORCEACTIVE bit is 1, the clock stays active.</p> <p>0 The module has not received a request to stop its clock.</p> <p>1 The module has received a request to stop its clock.</p>
3	FORCEACTIVE	0 1	<p>Force active power and clock. This bit forces the clock on, and because power is necessary for clocking, this bit also has the same effects as the FORCEPOWER bit. The FORCEACTIVE bit applies whether the TAP is selected or deselected. The value read from the FORCEACTIVE bit does not reflect the value written until the power and clock control logic has acted upon the value change.</p> <p>When reading:</p> <p>0 The clock and power settings of the module follow the normal application settings, unless one of the other emulation settings is affecting them.</p> <p>1 The application cannot turn off the clock or the power, and POR cannot be asserted to the module. (The application can change the voltage level of the module and frequency of the clock. A warm reset is possible.)</p> <p>When writing:</p> <p>0 Return control of the power settings to the application and to other emulation logic.</p> <p>1 Force the module clock and power to be active, and block attempts by the application to turn them off. If the clock is off when the FORCEACTIVE bit is set, restart it to its default frequency, or to the frequency selected when the clock was last stopped.</p>
2	CLOCK	0 1	<p>Clock status. This bit indicates the clock state of the module associated with this TAP.</p> <p>0 The module clock is off.</p> <p>1 The module clock is on.</p>
1	TAPACCESSABLE	x	<p>When 0, the TAP cannot be accessed due to security. When 1, the TAP can be accessed.</p>

Table 4-1. Field Descriptions for a Secondary Debug TAP Register (SDTAP_x) (continued)

Bit	Field	Value	Description
0	TAPPRESENT	1	TAP present. TAPPRESENT = 1 because this TAP has been implemented in the device.

4.6.2 Unimplemented TAP Register (TAP_UNIMP)

On a particular TI device, the number of secondary TAPs implemented varies. Accesses made to an unimplemented TAP are made to a special, non-mapped register (see [Figure 4-3](#) and [Table 4-2](#)). Regardless of how many secondary TAPs are unimplemented, there is only one unimplemented TAP register (TAP_UNIMP). During reads, bit 0 of the register indicates that the TAP is not present.

Figure 4-3. Unimplemented TAP Register (TAP_UNIMP)


LEGEND: R = Read, -n = Value after reset

Table 4-2. Unimplemented TAP Register (TAP_UNIMP) Field Descriptions

Bit	Field	Value	Description
23:1	RESERVED	0	These read-only bits are reserved.
0	PRESENT	0	TAP present bit. PRESENT = 0 because the TAP has not been implemented in this chip.

Reset Operations

The ICEPick module reacts appropriately to four reset operations:

- POR of the target device
- Test-logic reset initiated with the master nTRST signal
- Warm functional reset of the target device (for example, a reset initiated by a watchdog timer)
- Warm functional reset of a particular module in the target device (for example, a processor core being reset)

Topic	Page
5.1 Power-On Reset of the Target Device	40
5.2 Test-Logic Reset Initiated With the Master nTRST Signal	40
5.3 Warm Functional Reset of the Target Device	40
5.4 Warm Functional Reset of a Module in the Target Device	40

5.1 Power-On Reset of the Target Device

A POR is asserted when power is first applied to the target device. This device-wide signal applies to all modules and TAPs. All ICEPick functionality is reset when this reset condition is asserted. A POR configures the EMU1 and EMU0 pins as inputs. On the rising edge of the POR signal, the EMU pins are sampled and select an initial TAP configuration (see [Table 5-1](#)).

Table 5-1. Boot (Power-On) TAP Configuration

EMU1	EMU0	TAP Configuration	Other Effects
0	0	ICEPick TAP and default secondary TAP(s)	
0	1	ICEPick TAP only	Reserved, do not use.
1	0	ICEPick TAP only	Device wait-in-reset mode on
1	1	ICEPick TAP only	

When an ICEPick-TAP-only configuration is used, none of the secondary TAPs are selected at power up. Until more TAPs are selected by the debugger, the ICEPick TAP is the only TAP between the device-level TDI and TDO pins.

EMU1 and EMU0 have internal pullups. TI recommends placing external pullups on these pins, too. When EMU1 and EMU0 are both high, the scan chain powers up with only the ICEPick TAP selected and no special modes on.

To start operation in the device wait-in-reset mode, make EMU1 high and EMU0 low. In this mode, if any part of the target system enters its reset state, it is held in the reset state until released. The device wait-in-reset mode is selected one time at power up; it is not reselected when the master nTRST signal is asserted.

When both EMU pins are driven low, the default TAP selection is the ICEPick TAP plus one or more secondary TAPs that have been preset at the factory. This option is available so that tools which are not aware of the ICEPick TAP can operate on the other TAP(s). If this TAP configuration is selected at power up, it is reselected each time the master nTRST signal is asserted.

Due to synchronization logic, no JTAG operations are possible until three TCK cycles after the rising edge of the POR signal.

5.2 Test-Logic Reset Initiated With the Master nTRST Signal

When the master nTRST signal is driven low, the following actions are taken, regardless of whether the test clock is active:

- The EMU1 and EMU0 pins are automatically changed to inputs. If EMU1 and EMU0 were both sampled low at power up, the default TAP configuration is selected. Otherwise, the EMU pins select one of the other TAP configurations shown in [Table 5-1](#). After POR, a rising edge on nTRST causes the ICEPick module to latch this TAP configuration.
- The ICEPick TAP state-machine is reset.

These are the same actions taken when the TAP is moved to its TEST LOGIC RESET state with the TMS signal.

5.3 Warm Functional Reset of the Target Device

A warm functional reset of the device has no effect on TAP selection. The current set of secondary TAPs that are selected remains the same after the reset.

One possible cause of this reset is the setting of the SYSRESET bit of the ICEPick system control register. The reset clears this bit.

5.4 Warm Functional Reset of a Module in the Target Device

A warm functional reset of a particular module has no effect on TAP selection. A secondary TAP can still be included in the scan chain even though the associated module is held in reset. Bits in the appropriate SDTAP register can change to reflect that the module is in the reset state.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
RF/IF and ZigBee® Solutions	www.ti.com/lprf

Applications

Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Transportation and Automotive	www.ti.com/automotive
Video and Imaging	www.ti.com/video
Wireless	www.ti.com/wireless-apps

TI E2E Community Home Page

e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2011, Texas Instruments Incorporated