NXP Semiconductors
User's Guide

Document Number: IMX8MNBSDLUG
Rev. 0, 12/2019

i.MX 8M Nano Boundary Scan Entry User's Guide

Contents

1. Overview

This document focuses on the procedure of entering boundary scan mode for board-level test. It provides the setup sequence and script examples to ensure first-pass success.

Engineers should understand the standard for test access port and boundary scan architecture from IEEE 1149.1.

1.1. Boundary Scan

Boundary scan is a method for testing interconnects on PCBs and internal IC sub-blocks. It is defined in the IEEE 1149.1 standard.

In boundary scan test, each primary input and output signal on a device is supplemented with a multi-purpose memory element called a boundary scan cell. These cells are connected to a shift register, which is referred to as the boundary scan register. This register can be used to read and write port states.

In normal mode, these cells are transparent and the core is connected to that ports. In boundary scan mode, the core is isolated from the ports and the port signals are controlled by the JTAG interface.

Figure 1 shows the principle of boundary scan chain.

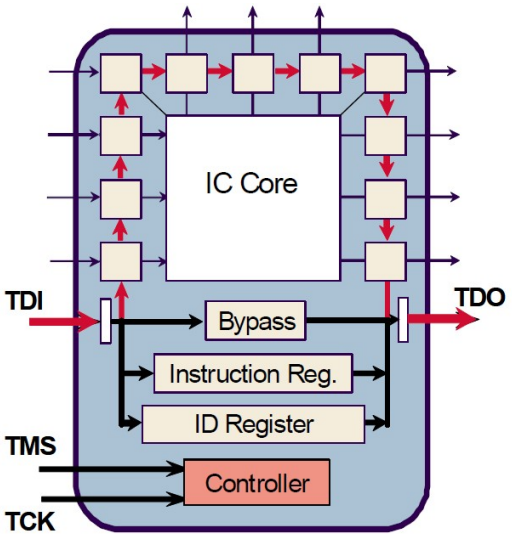

Figure 1. Boundary Scan Principle

1.2. Test access port(TAP)

The TAP is a general-purpose port that can provide access to many test support functions built into the component, it has four or five signals as given in Table 1.

Table 1. PCB design recommendations

Signal Name	I/O Type	Description
TCK	Input	The test clock input provides the clock for the test logic
TMS	Input	The value of the signal present at TMS at the time of a rising edge at TCK determines the next state of the TAP controller
TDI	Input	Serial test instructions and data are received by the test logic.
TDO	Output	Serial output for test instructions and data from the test logic
TRST_N[1]	Input	Optional active low signal to reset the TAP controller.

[1] i.MX 8M Nano does not support this signal.

1.3. TAP controller state diagram

The state diagram for the TAP controller shall be as shown in Figure 2. The value shown adjacent to each state transition in this figure represents the signal present at TMS at the time of a rising edge at TCK. Each state of the TAP controller can be reached by a sequence of bits transmitted via the TMS depending on the current state.

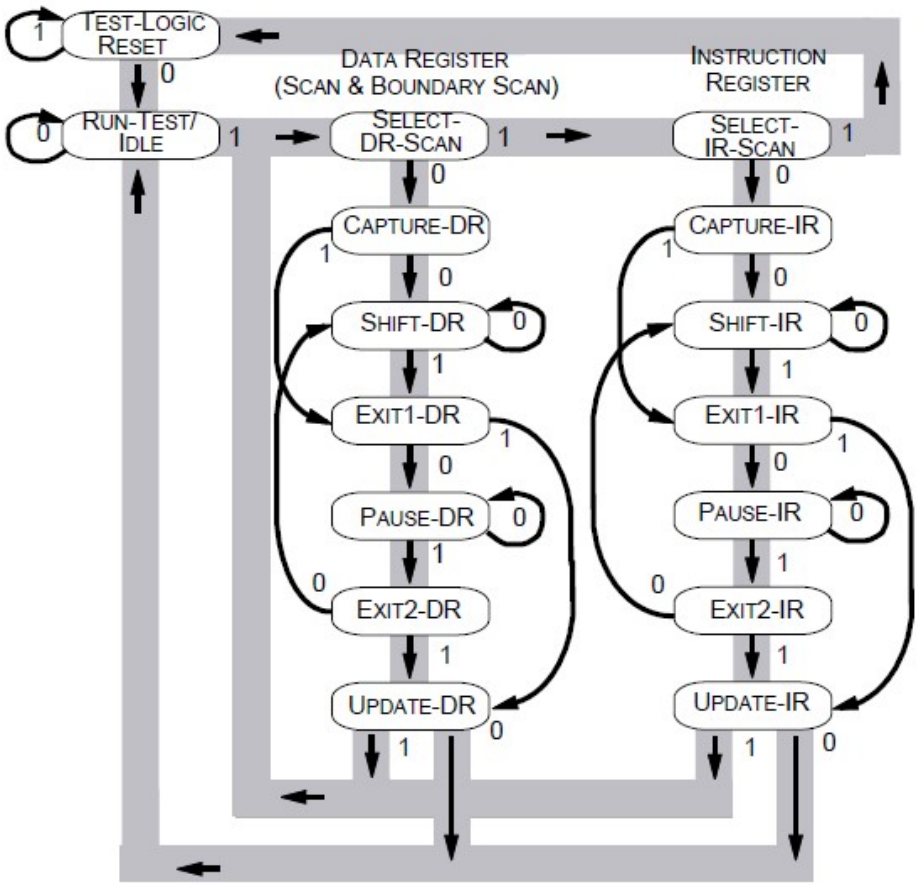

Figure 2. TAP controller state diagram

The following states of the TAP controller are importance:

- Test-Logic Reset: This is the reset state. Regardless of the TAP controller's current state, it will enter Test-Logic Reset when TMS is held high for at least five rising edges of TCK.
- Run-Test/Idle: Once entered, the controller will remain in the Run-Test/Idle state as long as TMS is held low. When TMS is high, the controller moves to the Select-DR-Scan.
- Select-DR-Scan: This is a temporary controller state, the next state will be decided by TMS. If TMS is held low when the controller is in this state, the controller moves to the Capture-DR state. If TMS is held high, the controller moves to the Select-IR-Scan state.
- Select-IR-Scan: This is a temporary controller state, the next state will be decided by TMS. If TMS is held low when the controller is in this state, the controller moves to the Capture-IR state. If TMS is held high, the controller moves to the Test-Logic Reset state.
- Shift-IR: In this state, the Instruction Register is connected between TDI and TDO and shifts the captured bits on stage towards its serial output. It also shifts the new instruction bits into the Instruction Register from TDI. If TMS is held high when the controller is in this state, the controller moves to the Exit1-IR state. If TMS is held low, the controller remains in this state.
- Update-IR: The instruction previously shifted into the Instruction Register is latched in this

controller state. When the controller is in this state, it enters the Select-DR-Scan state if TMS is high or the Run-Test/idle state if the TMS is low.

- Shift-DR: In this state, the Data Register is connected between TDI and TDO and shifts the test data on stage towards its serial output. It can also shift data into the data register from TDI. If TMS is held high when the controller is in this state, the controller moves to the Exit1-DR state. If TMS is held low, the controller remains in this state.
- Update-DR: The data is latched onto the parallel outputs of these test data registers from the shift-register path. When the controller is in this state, it enters the Select-DR-Scan state if TMS is high or the Run-Test/idle state if the TMS is low.

2. Boundary scan Entry

The 8M Nano doesn't have the COMPIANCE_PATTERNS, it is not completely compliant with IEEE 1149.1, which needs a special setup sequence to enter boundary scan mode as below.

1. BOOT_MODE0&BOOT_MODE1&BOOT_MODE2&BOOT_MODE3 should be high, then power up the board.
2. Load the instruction codes(b10000) and data codes(b00111000) to the chip TAP controller by JTAG port to enter boundary scan mode.
   a) Set TMS to 11111, soft reset the JTAG TAP controller, go to Test-Logic Reset state.
   b) Set TMS to 01100, go to Shift-IR state.
   c) Load IR Instruction = b10000 in Shift-IR state.
   d) Set TMS to 110, go to RUN-TEST-IDLE sate.
   e) Set TMS to 100, go to Shift-DR state.
   f) Load DR data = b00111000 in Shift-DR state.
   g) Set TMS to 110, go to RUN-TEST-IDLE state.
   h) End task.

NOTE

When left the Shift-IR state, the TAP controller can go to RUN-TEST-IDLE if TMS set to 110, or go to Select DR-Scan state directly if TMS set to 111

3. Appendix

3.1. Script A with Verilog Language

This script is created with Verilog language to enter boundary scan mode for 8M Nano.

```
initial begin

reset
load_instr(5'b10000);
```

```
load_data(512'b00111000,10'd8);

end

task reset;                //Test-Logic Reset
begin
   #(tck_clock_period)
   jtag_tms=1'b1;
   #(tck_clock_period);
   #(tck_clock_period);
   #(tck_clock_period);
   #(tck_clock_period);
   #(tck_clock_period);
end
endtask

task load_instr;
input [4:0] instr;
integer i;
reg [4:0] instr_d;
begin
   #(tck_clock_period)
   jtag_tms=1'b1;         //Select-DR-Scan
   #(tck_clock_period)
   jtag_tms=1'b1;         //Select-IR-Scan
   #(tck_clock_period)
   jtag_tms=1'b0;         //Capture-IR
   instr_d = instr;
   #(tck_clock_period)
   jtag_tms=1'b0;         //Shift-IR
   for(i=0;i<5;i=i+1)
     begin
       #(tck_clock_period);
       jtag_tdi=instr_d[i];        //Shift-IR
       //jtag_tdi=instr_d[5];      //Shift-IR
       //instr_d = {instr_d[4:0],1'b0};
     end
   jtag_tms=1'b1;           //Exit1-IR
   #(tck_clock_period)
   jtag_tms=1'b1;           //Update-IR
   #(tck_clock_period)
   jtag_tms=1'b0;           //Run-Test/Idle
   #(tck_clock_period)
   #(tck_clock_period) ;
   #(tck_clock_period) ;
   #(tck_clock_period) ;
   #(tck_clock_period) ;
   #(tck_clock_period) ;
   #(tck_clock_period) ;
end
endtask

task load_data;
input [511:0] data;
input [9:0] length;
reg [511:0] data_d;
```

```
   integer i;
   begin
      #(tck_clock_period)
      jtag_tms=1'b1;           //Select-DR-Scan
      #(tck_clock_period)
      jtag_tms=1'b0;           //Capture-DR
      data_d = data;
      #(tck_clock_period)
      jtag_tms=1'b0;           //Shift-DR
      //for(i=0;i<414;i=i+1)
      for(i=0;i<length;i=i+1)
        begin
          #(tck_clock_period);
          jtag_tdi=data_d[i];          //Shift-DR
          //jtag_tdi=data_d[413];      //Shift-DR
          //data_d = {data_d[412:0],1'b0};
        end
      jtag_tms=1'b1;           //Exit1-DR
      #(tck_clock_period)
      jtag_tms=1'b1;           //Update-DR
      #(tck_clock_period)
      jtag_tms=1'b0;           //Run-Test/Idle
      #(tck_clock_period)
      #(tck_clock_period) ;
      #(tck_clock_period) ;
      #(tck_clock_period) ;
      #(tck_clock_period) ;
      #(tck_clock_period) ;
      #(tck_clock_period) ;
   End
   endtask
```

3.2. Script B with Trace32

This script is used with Trace32 software tool to enter boundary scan mode for 8M Nano.

```
   // Script to enter boundary scan mode over JTAG using TMS, TDI shifts (TDO
ignored)

   AREA
   SYStem.JtagClock 10MHz

   //JTAG.LOCK

   JTAG.SHIFTTMS 1 1 1 1 1
   // soft reset of the JTAG interface, go to Test-Logic Reset state
   PRINT "Reset JTAG TAP"

   // Set TAP IR
   JTAG.SHIFTTMS 0 1 1 0 0
   // go to Shift-IR state
   JTAG.SHIFTREG 0 0 0 0 1
   // IR Instruction = b10000 and go to Exit1-IR state directly
   PRINT "Set TAP IR"
```

```
// Set TAP DR
JTAG.SHIFTTMS 1 1 0 0
// go to Shift-DR state
JTAG.SHIFTREG 0 0 0 1 1 1 0 0
// DR Instruction = b00111000 and go to Exit1-DR state directly
PRINT "Set TAP DR"

JTAG.SHIFTTMS 1 0
// go to RUN-TEST/IDLE state

//JTAG.UNLOCK

ENDDO
```

4. Revision History

Table 2. Revision History

Revision number	Date	Substantive changes
0	10/2019	Initial release.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

Document Number: IMX8MNBSDLUG
Rev. 0
12/2019