# UG0576
# User Guide
# RTG4 FPGA System Controller

Microsemi

a Microchip company

**Microsemi Headquarters**
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

## About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

# Contents

# Figures

# Tables

# 1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

## 1.1 Revision 7.0

Information about System Controller Suspend Mode, page 19 was updated.

## 1.2 Revision 6.0

The following is a summary of the changes in revision 6.0 of this document.

- Information about System Controller Suspend Mode, page 19 was updated.
- DEVRST_N to Functional Timing Diagram, page 15 was added.
- Updated Figure 3, page 4.

## 1.3 Revision 5.0

Updated Power-Up to Functional Time Data, page 11 and DEVRST_N Power-Up to Functional Time, page 15.

## 1.4 Revision 4.0

The following is a summary of the changes in revision 4.0 of this document.

- Information about DEVRST_N was updated, see Power-On Reset Generator, page 8.
- Information about system controller suspend mode was updated, see System Controller Suspend Mode, page 19.

## 1.5 Revision 3.0

The section Power-Up to Functional Time Data, page 11 and all the associated subsections, figures, and tables are added and edited in revision 3.0 of this document.

## 1.6 Revision 2.0

The following is a summary of the changes in revision 2.0 of this document.

- Updated the document for SERDESIF to SERDES through out the document
- Updated the Power-On Reset Generator, page 8 for PO_RESET_N change
- Added the Power-Up to Functional Time Sequence, page 10
- Updated the Functions, page 2
- Deleted the System IP Interface (SII) and Oscillator Control sections
- Added Using System Controller, page 18

## 1.7 Revision 1.0

Revision 1.0 was the first publication of this document.

# 2 System Controller

The user guide describes the system controller in RTG4™ field programmable gate array (FPGA) devices. The system controller manages the programming, initialization, and configuration of the RTG4 devices.This user guide describes the subsystems and interfaces that are present in the RTG4 system controller.

## 2.1 Functions

The following subsystems in the system controller perform various functions:

- JTAG Programming: helps erase, program, and verify functions.
- Power-On Reset Generator: resets the FPGA functional blocks after power-on or system reset.
- System Controller Suspend Mode: keeps the system controller in Suspend mode after device initialization.

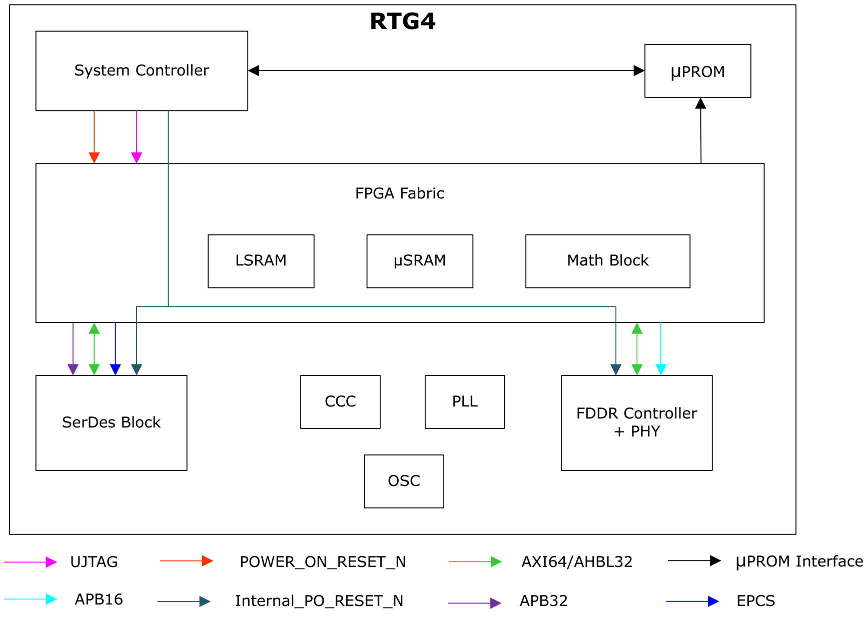
The following figure shows the system controller top-level architecture. The system controller top-level architecture has the following devices:

- FPGA Fabric (LSRAM, uSRAM, and Math Block)
- SerDes Block
- FDDR Controller

The system controller interfaces with the FPGA fabric using user JTAG (UJTAG) and power-on reset (POR) interconnections.

- UJTAG: The UJTAG interface is an extension of the external JTAG port in to the RTG4 devices, controlled by the TAP controller when it is not performing the JTAG programming.
- Power-On Reset Generator: The system controller generates the POR signal (Internal_PO_RESET_N), and sends to all subsystems after power-on. The POWER_ON_RESET_N signal is generated from the Internal_PO_RESET_N signal and can be used in the user design as a reset for the FPGA fabric.

***Figure 1 •*** **System Controller Top - Level Architecture**

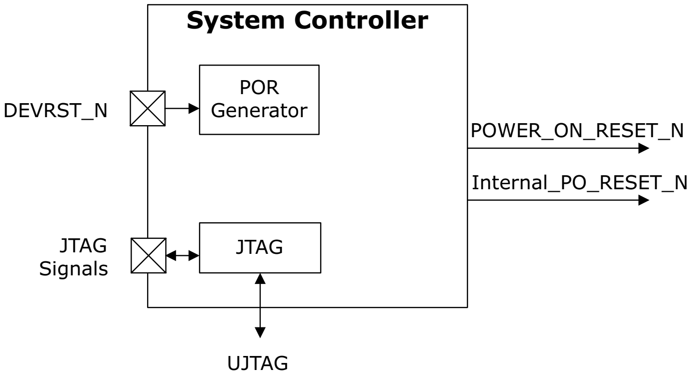


The color codes in the preceding figure differentiate the interconnections in the system controller top-level architecture.

## 2.2 Functional Description

The following figure shows the system controller subsystems and interfaces.

***Figure 2 •*** **External and Internal Interfacing of the System Controller**



The system controller has the following subsystems and interfaces.

### 2.2.1 Subsystems

- JTAG
- Power-On Reset Generator

### 2.2.2 Interfaces

- DEVRST_N
- JTAG signals
- UJTAG
- POWER_ON_RESET_N
- Internal_PO_RESET_N

The following sections provide short descriptions of the subsystems and interfaces:

#### 2.2.2.1 JTAG

The system controller implements the functionality of a JTAG slave, complaint to the IEEE 1532 and IEEE 1149.1 standards. The JTAG interface communicates with the system controller using a command register that conveys the JTAG instruction to be executed, and a 128-bit data buffer that transfers any associated data. The JTAG port from the factory is used for monitoring JTAG accesses, and so on.

JTAG has the following functions:

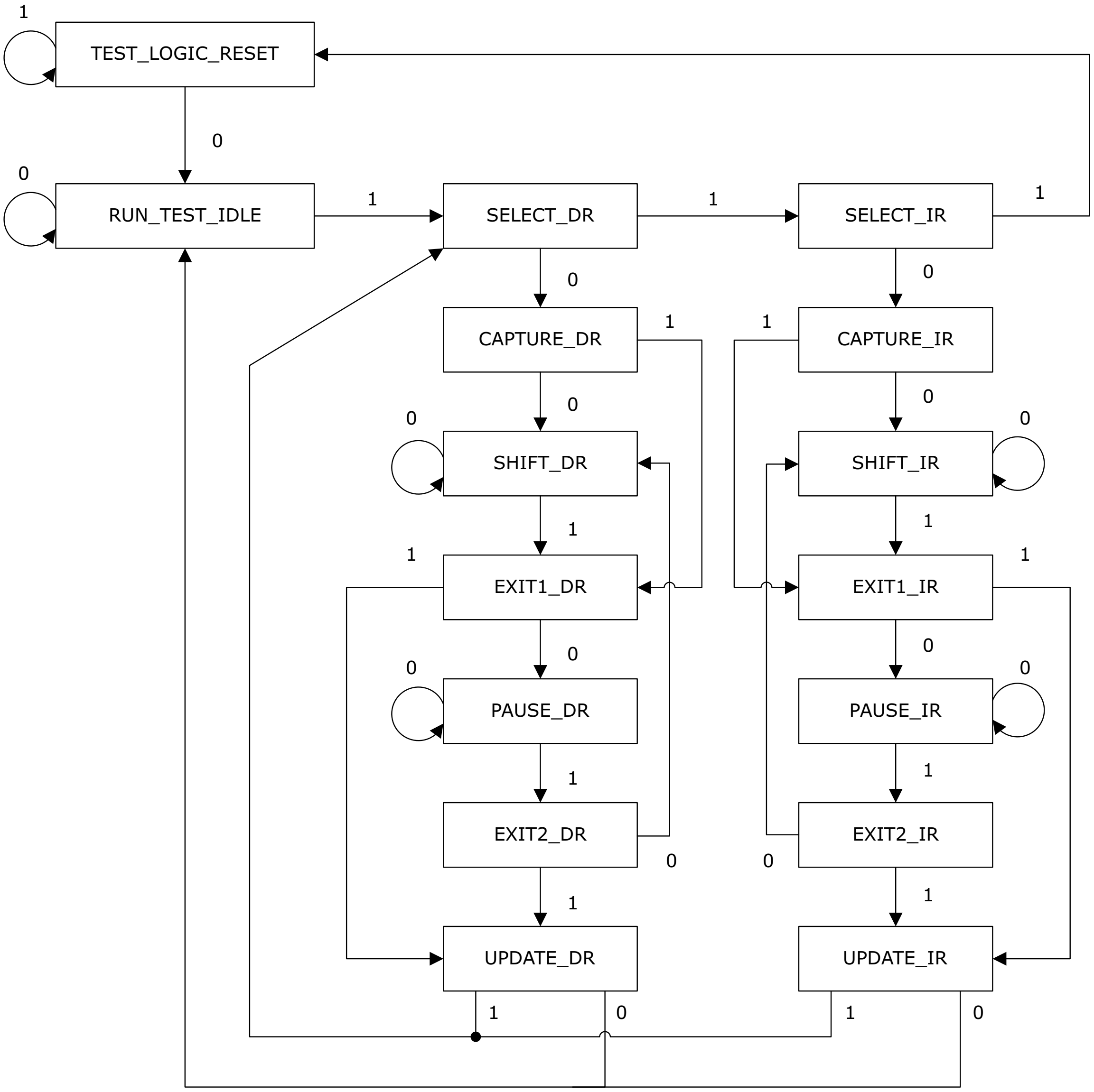
- TAP Controller State Machine
- Boundary Scan Opcodes

#### 2.2.2.1.1 TAP Controller State Machine

The TAP controller is a 4-bit state machine (16 states) that operates as shown in the following figure. The 1s and 0s represent the values that must be present on TMS at a rising edge of TCK for the given state transition to occur. IR and DR indicate that the instruction register or the data register operates in that state.The TAP controller receives two control inputs (TMS and TCK) and generates control and clock signals for the rest of the test logic architecture. On power-up, the TAP controller enters into the TEST_LOGIC_RESET state.

To guarantee that the controller resets from any of the possible states, TMS must remain High for five TCK cycles. The TRST pin can also be used to asynchronously place the TAP controller in the TEST_LOGIC_RESET state.

***Figure 3 •*** **TAP Controller State Machine**



### 2.2.2.1.2 Boundary Scan Opcodes

The RTG4 devices support all mandatory IEEE 1149.1 instructions (EXTEST, SAMPLE/PRELOAD, and BYPASS) and the optional IDCODE instruction (the following table).

***Table 1 •*** **Boundary Scan Opcodes**

| Instruction | Hex Opcode |
|---|---|
| EXTEST | 00 |
| HIGHZ | 07 |
| USERCODE | 0E |
| SAMPLE/PRELOAD | 01 |
| IDCODE | 0F |
| CLAMP | 05 |
| BYPASS | FF |
| UJTAG | 10-7F |

See *UG0574: RTG4 FPGA Fabric User Guide* for pin descriptions and board-level recommendations.

The JTAG interface is used for the following applications:

- Boundary Scan
- UJTAG
  See *UG0602: RTG4 FPGA Programming User Guide* for more information.

### 2.2.2.1.3 Boundary Scan

The IEEE 1149.1 standard defines a hardware architecture and a set of mechanisms for boundary scan testing. The JTAG operations are used during boundary scan testing. The basic boundary scan logic circuit has a test access port (TAP) controller, test data registers, and an instruction register.

The RTG4 devices support three types of test data registers: bypass, device identification, and boundary scan. The bypass register is selected when no other register is accessed in a device and this speeds up test data transfer to other devices in a test data path. The 32-bit device identification register is a shift register with the four fields (LSB, ID Number, Part Number, and Version). The boundary scan register observes and controls the state of each I/O pin, except the SerDes Block I/Os. Each I/O cell has four boundary scan register cells, each with Serial-in pin, Serial-out pin, Parallel-in pin, and Parallel-out pin.

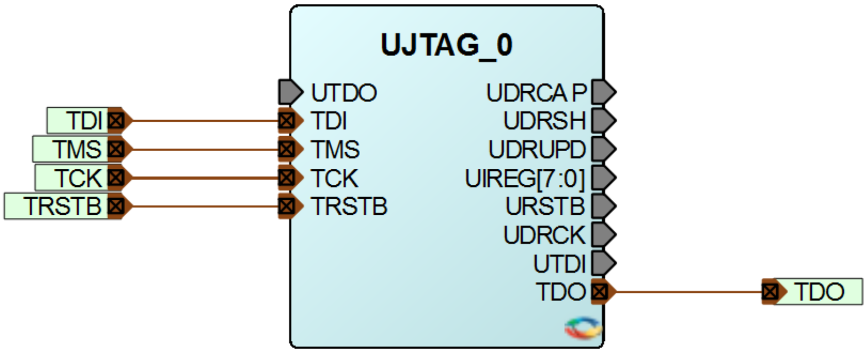
### 2.2.2.1.4 UJTAG

The UJTAG interface is an extension of the external JTAG port in to the RTG4 devices, controlled by the TAP controller when it is not performing the JTAG programming. It can be used to shift data or OPCODEs to and from the internal fabric logic. The UJTAG functionality is made available by instantiating the UJTAG macro from the Libero® System-on-Chip (SoC) IP catalog into a SmartDesign or by instantiating it directly inside the HDL file. The real-time updating and monitoring of the internal behavior of the FPGA fabric are enabled using the UJTAG macro. See *AC227: Flash UJTAG Application Note* for more information about using the UJTAG module.

### UJTAG Macro

The following figure shows a block symbol of the UJTAG macro. The TDI, TMS, TCK, TRSTB, and TDO ports of the UJTAG macro are directly connected to the JTAG TAP controller, and all the other ports are accessible from the FPGA fabric.

***Figure 4 •*** **UJTAG Macro**



The following table lists the UJTAG ports accessible to the FPGA fabric.

***Table 2 •*** **UJTAG Ports Accessible to the FPGA Fabric**

| Port | Direction | Polarity | Description |
|---|---|---|---|
| UIREG [7:0] | Output | – | This 8-bit bus carries the contents of the JTAG instruction register of each device. Instruction register values 16 to 127 are not reserved and can be employed as user-defined instructions. |
| URSTB | Output | Low | URSTB is an Active Low signal and is asserted when the TAP controller is in Test-Logic-Reset mode. URSTB is asserted at power-up, and a power-on reset signal resets the TAP controller. URSTB is asserted until an external TAP access changes the TAP controller state. |
| UTDI | Output | – | This port is directly connected to the TAP's TDI signal. |
| UTDO | Input | – | This port is the user TDO output. Inputs to the UTDO port are sent to the TAP TDO output MUX when the IR address is in user range. |
| UDRSH | Output | High | Active High signal enabled in the Shift_DR TAP state. |
| UDRCAP | Output | High | Active High signal enabled in the Capture_DR TAP state. |
| UDRCK | Output | – | This port is directly connected to the TAP's TCK signal. |
| UDRUPD | Output | High | Active High signal enabled in the Update_DR TAP state. |

### UJTAG Operation

The fundamental concept of the UJTAG design is its connection with the TAP controller state machine. Understanding the basic functions of the UJTAG macro is a must before designing with it. See the TAP Controller State Machine, page 4 for more information.

UIREG [7:0] holds the contents of the JTAG instruction register. The UIREG vector value is updated when the TAP controller state machine enters the Update_IR state. Instructions 16 to 127 are user-defined and can be employed to encode multiple applications and commands within an application. Loading new instructions to the UIREG vector requires sending an appropriate logic pattern to TMS to place the TAP controller in a full IR cycle starting from the Select IR_Scan state and ending with the Update_IR state.

UTDI, UTDO, and UDRCK are directly connected to the JTAG TDI, TDO, and TCK ports, respectively. The TDI input can be used to provide either data (TAP controller in the Shift_DR state) or the new contents of the instruction register (TAP controller in the Shift_IR state).

UDRSH, UDRUPD, and UDRCAP are High, when the TAP controller state machine is in the Shift_DR, Update_DR, and Capture_DR states. Hence, they act as flags to indicate the stages of the data shift process. These flags are useful for applications in which blocks of data are shifted to the design from JTAG pins. For example, an active UDRSH can indicate that UTDI contains the data bitstream, and UDRUPD is a candidate for the end-of-data-stream flag.

### Typical UJTAG Applications

Bi-directional access to the JTAG port from the FPGA fabric creates flexibility for implementing a variety of applications. This section describes one of these applications. The bi-directional access is done by importing or exporting the data through the UJTAG macro. However, the applications are not limited to what is presented in this section. UJTAG can serve different purposes in many designs as an elementary or auxiliary part of the design.

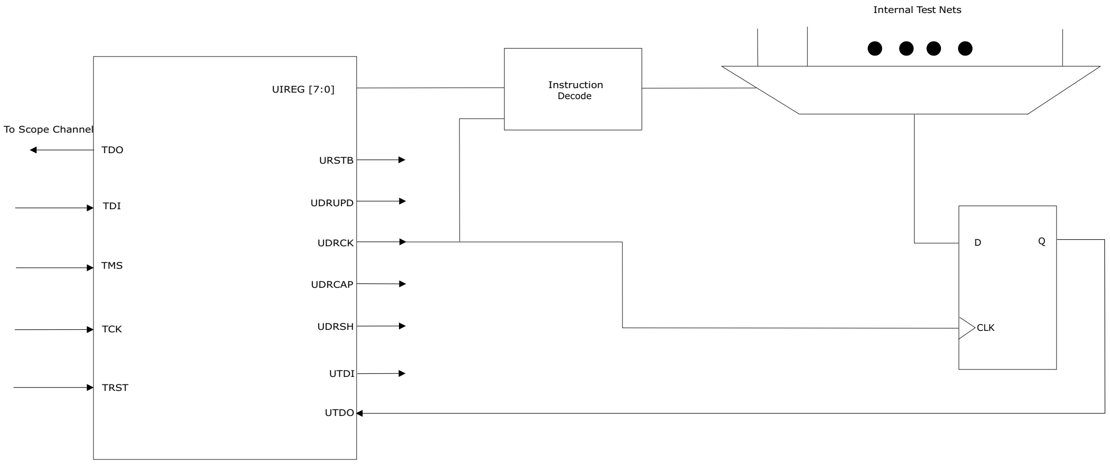
### UJTAG Design Testing and Debugging

The design must be tested, debugged, and verified on silicon or in the final embedded application. To debug and test the functionality of designs, it is required to monitor some internal logic (or nets) during device operation. The approach of adding design test pins to monitor the critical internal signals has many disadvantages, such as limiting the number of user I/Os. Furthermore, adding external I/Os for test purposes requires an additional or dedicated board area for testing and debugging.

The UJTAG macro provides a flexible and cost-effective solution for silicon test and debug applications. In this solution, the signals under test are shifted out to the TDO pin of the TAP controller.

The main advantage is that all the test signals are monitored from the TDO pin, and it does not require additional pins or board-level resources. The following figure shows the UJTAG usage example in test and debug applications.

***Figure 5 •*** **UJTAG Usage Example in Test and Debug Applications**



Multiple test nets are used in the internal MUX architecture. MUX is selected using the contents of the TAP controller instruction register, where an individual instruction (values from 16 to 127) corresponding to different signals is under test. The selected test signal can be synchronized with the rising or falling edge of TCK (optional) and sent out to UTDO to drive the TDO output of JTAG.

#### 2.2.2.2 Power-On Reset Generator

The following figure shows the conceptual block diagram of power-on reset generation. The POR generator block in system controller generates a power-on reset signal, Internal_PO_RESET_N.

***Figure 6 •*** **Conceptual Block Diagram of Power-On Reset Generation**

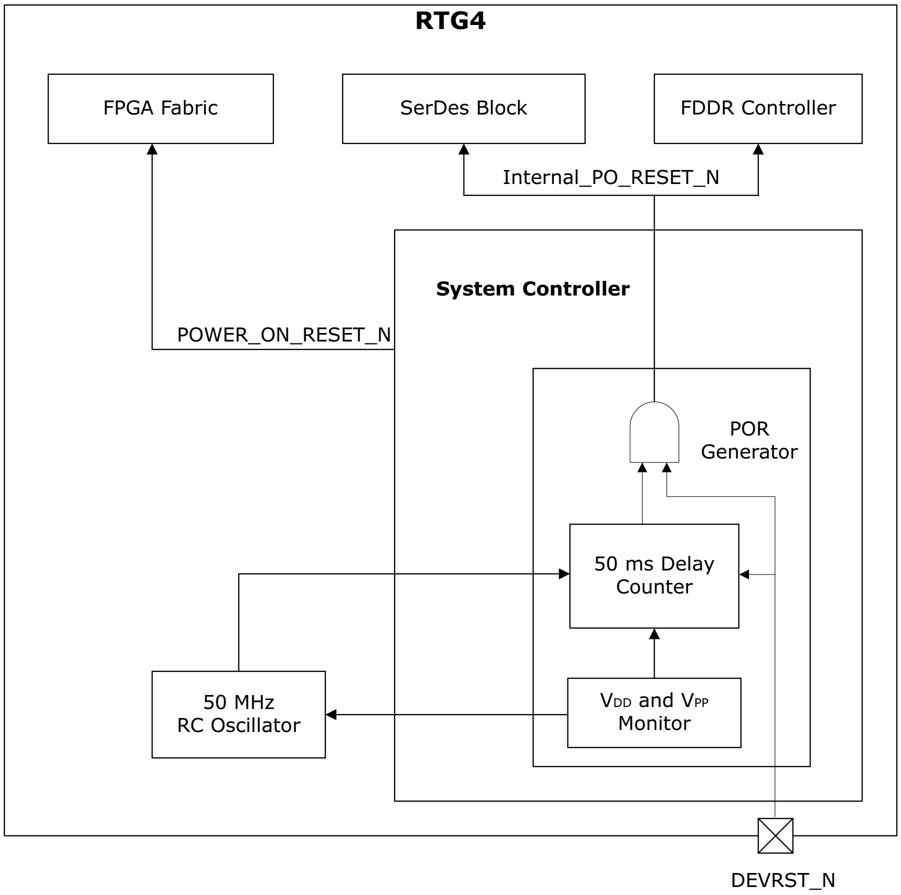


Figure 7, page 10 shows the power up to functional time sequence diagram. On power-up, the VDD and VPP monitor blocks in the POR generator block assert a power-on reset signal, Internal_PO_RESET_N. If the VDD and VPP supplies reach their threshold point (VDD ~ 0.55 V, VPP ~ 2.2 V), the 50 MHz RC Oscillator is turned-on. The Oscillator provides the clock to the 50 ms delay counter. The 50 ms Delay Counter starts counting once DEVRST_N is released. The delay counter is used to allow for the power supply rise time. All power supplies must be stable within 50 ms. When the counter reaches its maximum value (50 ms), the Internal_PO_RESET_N signal is de-asserted. Upon de-assertion of the Internal_PO_RESET_N signal, the system controller starts the initialization sequence of I/O banks, and FPGA Fabric Subsystem.

The POR delay counter can add a maximum of 50 ms of delay during the power-up to functional sequence. In the power-up scenario where DEVRST_N is not asserted externally, allowing DEVRST_N to be pulled-up to VPP during the supply ramp-up, the counter delay will be closer to 50 ms. In this case, the delay counter starts counting when the power supplies have not reached nominal voltage levels, and thus it initially counts slower. In contrast, for the power-up scenario where DEVRST_N is externally asserted until the power supplies have reached normal operating levels, the counter duration will be closer to 40 ms. Microsemi specifies the 50 ms maximum time as an upper bound for this delay across process, voltage and temperature conditions.

A dedicated input-only reset pad (DEVRST_N) is present on all the RTG4 devices with an internal pull-up resistor. Holding DEVREST_N in low will cause assertion of the Internal_PO_RESET_N signal. If an external reset circuit is connected to the DEVRST_N pin, it increases the power up to functional time due to the delays that the external reset device does add.

DEVRST_N is an asynchronous reset pin and must be asserted only when the device is unresponsive due to some unforeseen circumstances. It is not recommended to assert the DEVRST_N pin during programming operation, which might cause severe consequences including corrupting the device configuration. For more details on DEVRST_N timing information, see *DS0131: RTG4 FPGA Datasheet*.

Asserting DEVRST_N does not enable the delay counter (Ramp Delay) in the POR circuitry. The delay counter is operational only at power-up.

**WARNING:** When DEVRST_N is asserted (low), all user I/Os are fully tri-stated, When DEVRST_N is deasserted (high), all user I/Os are fully tri-stated with weak pull up. See the *DS0131: RTG4 FPGA Datasheet* for more information about the resistance values. Although, the JTAG I/Os are still enabled, they cannot be used as the TAP controller is in reset.

The system controller initiates a reset, causing a full-chip reset in the following conditions:

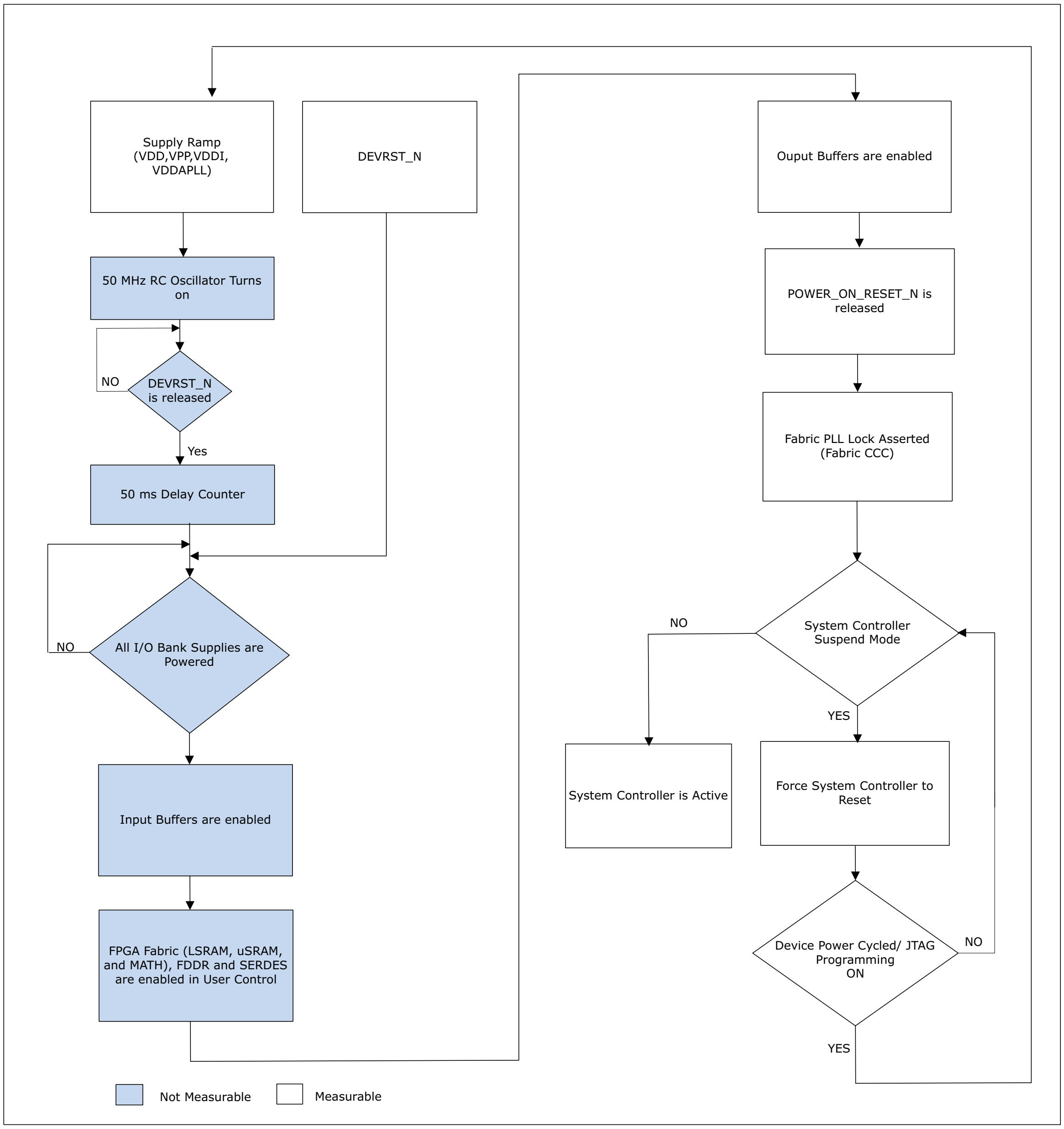
- Power-up
- Assertion of DEVRST_N input
- Completion of programming operation

**Note:** In RTG4, power-on-reset to all Flip-Flops in the chip is “hardware embedded” and does not require user routed power-on-reset signals from the SYSRESET macro to be routed in the FPGA fabric. If asynchronous set/reset are needed during user mode/normal operation, then they can be sourced from GRESET & RGRESET blocks.

### 2.2.3 Power-Up to Functional Time Sequence

The following figure shows the power up to functional time sequence diagram.

***Figure 7 •*** **Power - Up to Functional Time Sequence Diagram**


Supply Ramp (VDD,VPP,VDDI, VDDAPLL)
DEVRST_N
50 MHz RC Oscillator Turns on
NO
DEVRST_N is released
Yes
50 ms Delay Counter
NO
All I/O Bank Supplies are Powered
Input Buffers are enabled
FPGA Fabric (LSRAM, uSRAM, and MATH), FDDR and SERDES are enabled in User Control
Ouput Buffers are enabled
POWER_ON_RESET_N is released
Fabric PLL Lock Asserted (Fabric CCC)
NO
System Controller Suspend Mode
YES
System Controller is Active
Force System Controller to Reset
Device Power Cycled/ JTAG Programming ON
NO
YES
Not Measurable
Measurable

The power up to functional sequence is as follows:

- **Supply Ramp (VDD, VPP, VDDI, and VDDA phase-locked loops (PLL))** - There is no specific power up or power down sequencing requirement for RTG4 devices. The I/O banks can be brought-up in any order, before or after the core voltage. However, the device is only functional if all I/O bank supplies are powered-up. See *DS0131: RTG4 FPGA Datasheet*, for more information.
- The 50 MHz RC Oscillator is turned on, which provides the clock to the 50 ms delay counter. The counter will start running once DEVRST_N is released. When the counter reaches its maximum value (50 ms), the Internal_PO_RESET_N signal is de-asserted.
- All I/O bank supplies and core voltage (VDD) must be powered-up. If all I/O bank supplies reach 0.7 V, the device moves to a next sequence.
- Input buffers are enabled.
- FPGA fabric (LSRAM, uSRAM, and MATH), FDDR, and SerDes are enabled in User Control.
- Output buffers are enabled.
- POWER_ON_RESET_N is released. This can be accessed by user using the Sysreset macro.
- Fabric PLL (Fabric clock conditioning circuitry (CCC)) Lock is asserted.

For configuring System Controller Suspend Mode, see System Controller Suspend Mode, page 19.

**Note:** See the following section for characterized timing diagrams.Timing diagrams also show, when the flipflops inside the fabric “without GRESET, RGRESET" and "with GRESET" become operational during the power-up to functional sequence.

## 2.3 Power-Up to Functional Time Data

This section describes power-up to functional time sequence and provides timing numbers based on DEVRST_N assertion and VDD ramp up.

### 2.3.1 VDD Power-Up to Functional Time Data

The following sections describe the VDD power-up to functional time data design set up, measurement methodology and the parameters used to obtain the VDD power-up to functional time data.

#### 2.3.1.1 Measurement Methodology

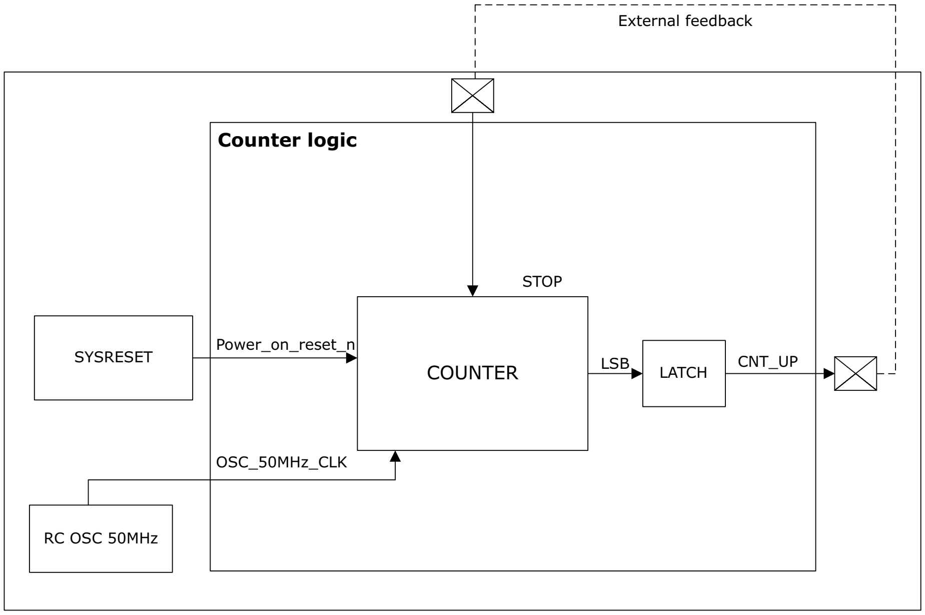
The core supply voltage VDD is connected to the appropriate source and VDD is monitored by the power-on reset circuitry to check if it reaches the minimum threshold value and initiates the system controller to release the device from reset. This scenario provides power-up to functional time data when only the FPGA fabric and the FPGA I/O are used with all supplies ramped up except VDD, which is ramped up at the last.

This method allows measuring timing of internal signal delays.The design uses a fabric counter that starts to operate when fabric flip flops are available (with no GRESET or RGRESET). The LSB of the counter output is connected to a latch and given to an output buffer, which is then connected to an input buffer of the fabric using external loopback. This input is used for stopping the counter from incrementing. The counter stops as soon as the LSB bit transitions to logic HIGH. Weak pull down of 1k is used so that counter does not stop because of weak pull up on CNT_UP output pad. The counter has overflow protection, once it reaches h'FFFF, it would stop counting.The power-up to functional time is measured from the VDD supply ramp to transition of the fabric buffer output.

GRESET generates a global asynchronous reset signal during power-up or programming, and allows the user to apply an asynchronous reset globally on the fabric flip flops, if required. RGRESET has a hardwired connection from GRESET resource and it spans for half a row. For more information on GRESET and RGRESET and it's software use models, see the dedicated global I/Os section in *UG0741: RTG4 FPGA I/Os User Guide.*

The following figure shows the characterization test design setup used for obtaining the VDD power-up to functional timing values.

*Figure 8 •* **VDD Power-Up to Functional Time Design Setup**



## 2.3.2 Parameters used for Obtaining VDD Power-Up to Functional Time Data

This section describes the parameters used for obtaining power-up to functional time data. Following are the test conditions:

- Ramp time
  - Power-on reset delay: 50 ms
  - VDD ramp time: 5 µs
- Testing conditions
  - Temperature: -55 °C, 25 °C, 125 °C
  - Voltage: Min, Typ, Max

Power-on reset delay indicates how long VDD takes to ramp up. The following table lists how to measure VDD power - up to functional time of RT4G150-LG1657 packaged devices.

*Table 3 •* **VDD Power-Up to Functional Time Test Cases**

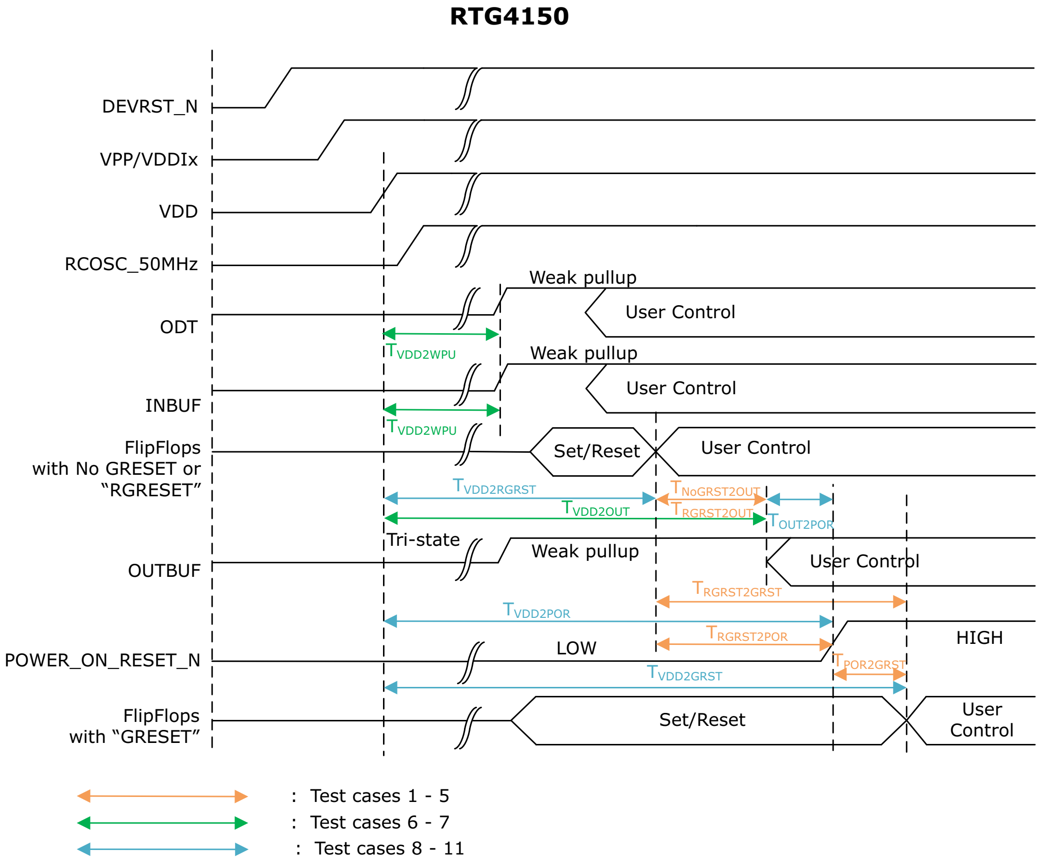
| Test Case | From | To | Description | How to Measure |
|---|---|---|---|---|
| 1 | Flipflops with No GRESET active | Output available | Fabric to output | fabric counter |
| 2 | Flipflops with "RGRESET" active | Output available | Fabric to output | fabric counter |
| 3 | Flipflops with "RGRESET" active | Flipflops with "GRESET" active | Fabric to output | fabric counter |
| 4 | Flipflops with "RGRESET" active | POWER_ON_RESET_N | Fabric to output | fabric counter |
| 5 | POWER_ON_RESET_N | Flipflops with "GRESET" active | Fabric to output | fabric counter |
| 6 | VDD | Output available | VDD to output | scoped |
| 7 | VDD | WPU (weak pull-up) | VDD to output | scoped |

*Table 3 •* **VDD Power-Up to Functional Time Test Cases *(continued)***

| Test Case | From | To | Description | How to Measure |
|---|---|---|---|---|
| 8 | VDD | Flip Flops without "GRESET" or “RGRESET" | VDD to fabric | Calculated |
| | | Flip Flops with "RGRESET" active | | |
| 9 | VDD | Flipflops with "GRESET" active | VDD to fabric | Calculated |
| 10 | VDD | POWER_ON_RESET_N | VDD to fabric | Calculated |
| 11 | Output available | POWER_ON_RESET_N | Output available to fabric | Calculated |

The following figure shows the behavior of different signals when device gets powered up with parameter names. Typical conditions: VDD = 1.2 V, VDDI = 2.5 V, $T_J$ = 25 °C.

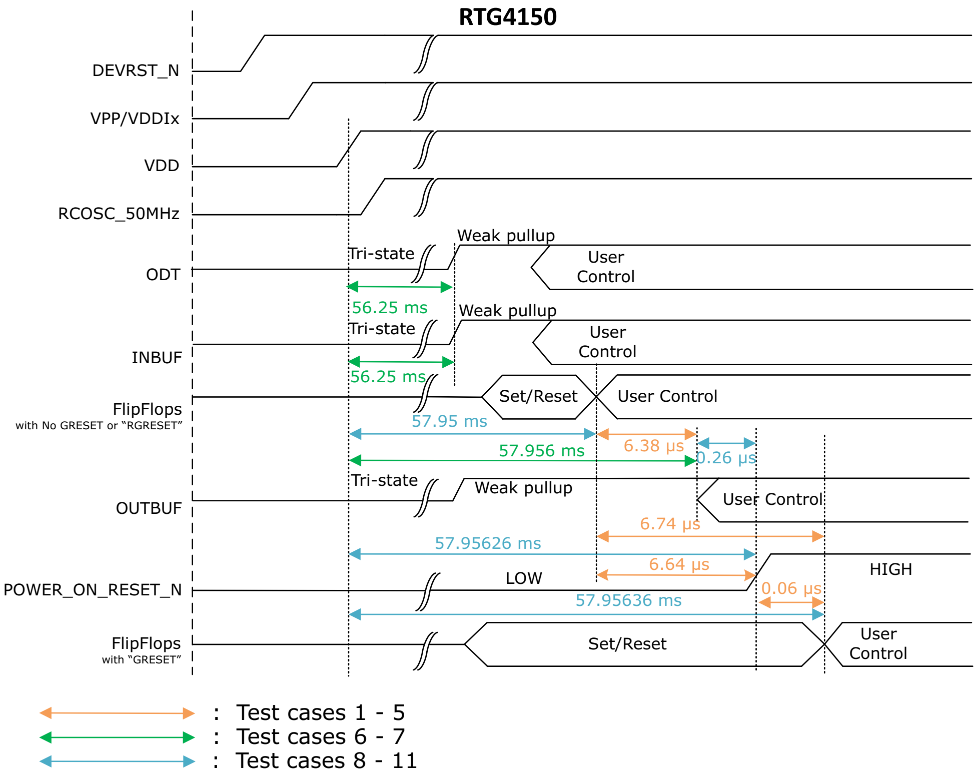
*Figure 9 •* **VDD Power-Up to Functional Timing With Parameter Names**



**Note:** RCOSC_50MHZ represents the transition of the signal.

Here all supplies are powered up except VDD, which is ramped last. Figure 10, page 14 shows the behavior of different signals when device gets powered up with timing numbers.
Typical conditions: VDD = 1.2 V, VDDI = 2.5 V, $T_J$ = 25 °C.

*Figure 10 •* **VDD Power-Up to Functional Timing with Timing Numbers**



**Note:** RCOSC_50MHZ represents the transition of the signal.

The following table lists the VDD power-up to functional time data of RT4G150 (internal signals and I/O).

*Table 4 •* **VDD Power-Up to Functional Time_1**

| Test Case | Parameter | From | To | Delay in (µs) |
|---|---|---|---|---|
| 1 | $T_{NoGRST2OUT}$ | Flipflops without "GRESET" or "RGRESET" | Output Available | 6.38 |
| 2 | $T_{RGRST2OUT}$ | Flipflops with "RGRESET" active | Output Available | 6.38 |
| 3 | $T_{RGRST2GRST}$ | Flipflops with "RGRESET" active | Flipflops with "GRESET" active | 6.74 |
| 4 | $T_{RGRST2POR}$ | Flipflops with "RGRESET" active | POWER_ON_RESET_N | 6.64 |
| 5 | $T_{POR2GRST}$ | POWER_ON_RESET_N | Flipflops with "GRESET" | 0.06 |

*Table 5 •* **VDD Power-Up to Functional Time_2**

| Test Case | Parameter | From | To | Delay in ms |
|---|---|---|---|---|
| 6 | $T_{VDD2OUT}$ | VDD | Output Available | 57.95 |
| 7 | $T_{VDD2WPU}$ | VDD | WPU(weak pull-up) | 56.25 |

The following table lists power-up to functional time data of RT4G150 (specification for VDD to internal signals).

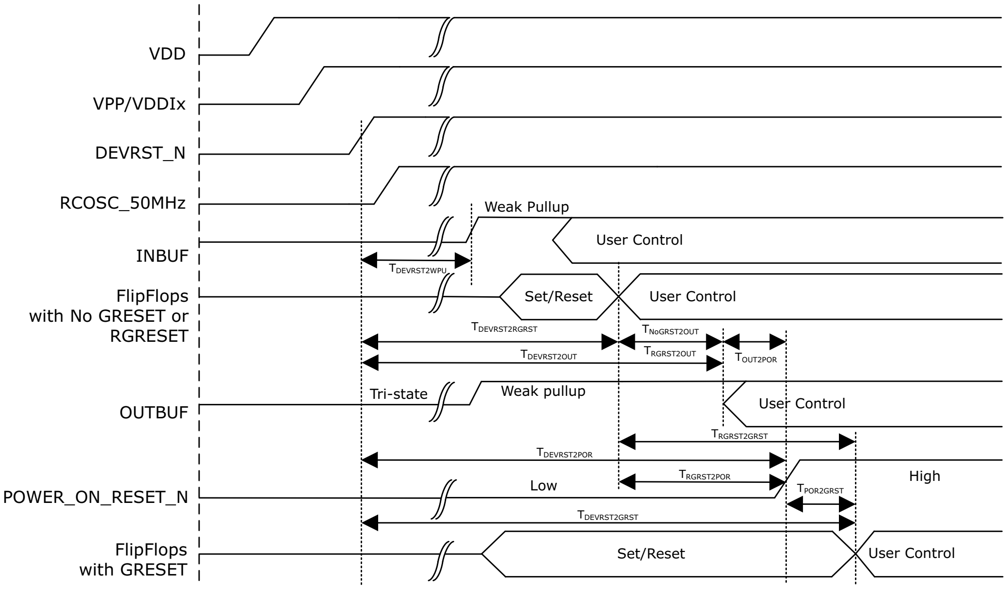
*Table 6 •* **VDD Power-Up to Functional Time_3**

| Test Case | Parameter | From | To | Delay in ms |
|---|---|---|---|---|
| 8 | $T_{VDD2RGRST}$ | VDD | No GRESET or "RGRESET" | 57.94962 |
| 9 | $T_{VDD2GRST}$ | VDD | ""GRESET" | 57.95636 |
| 10 | $T_{VDD2POR}$ | VDD | POWER_ON_RESET_N | 57.95626 |
| 11 | $T_{OUT2POR}$ | Output available | POWER_ON_RESET_N | 0.26 (µs) |

## 2.4 DEVRST_N Power-Up to Functional Time

This scenario provides DEVRST_N power-up to functional time data when the FPGA fabric, the FPGA I/O, and external oscillator are used. The design setup is same as the VDD power-up to functional time.

*Figure 11 •* **DEVRST_N to Functional Timing Diagram**



**Note:** RCOSC_50MHZ represents the transition of the signal.

**Note:** It is not recommended to assert DEVRST_N pin during programming (including eNVM), as it corrupts the device configuration. For more information on proper usage of the DEVRST_N pin, see *AC439: Board Design Guidelines for RTG4 FPGA Application Note.*

### 2.4.1 Parameters Used for Obtaining DEVRST_N Power-Up to Functional Time

This section describes the parameters used for obtaining DEVRST_N power-up to functional time data. Following are the test conditions and the experiment was done on RT4G150-LG1657 packaged devices.

- Ramp time
  - Power-on reset delay: 50 ms
  - DEVRST_N ramp time: 5 µs
- Testing conditions
  - Temperature: -55 °C, 25 °C, 125 °C
  - Voltage: Min, Typ, Max

**Note:** The DEVRST_N ramp time 5 µs is used for power-up to functional timing characterization and it is not the actual limit. See *RTG4 FPGA Datasheet* for the actual limit.

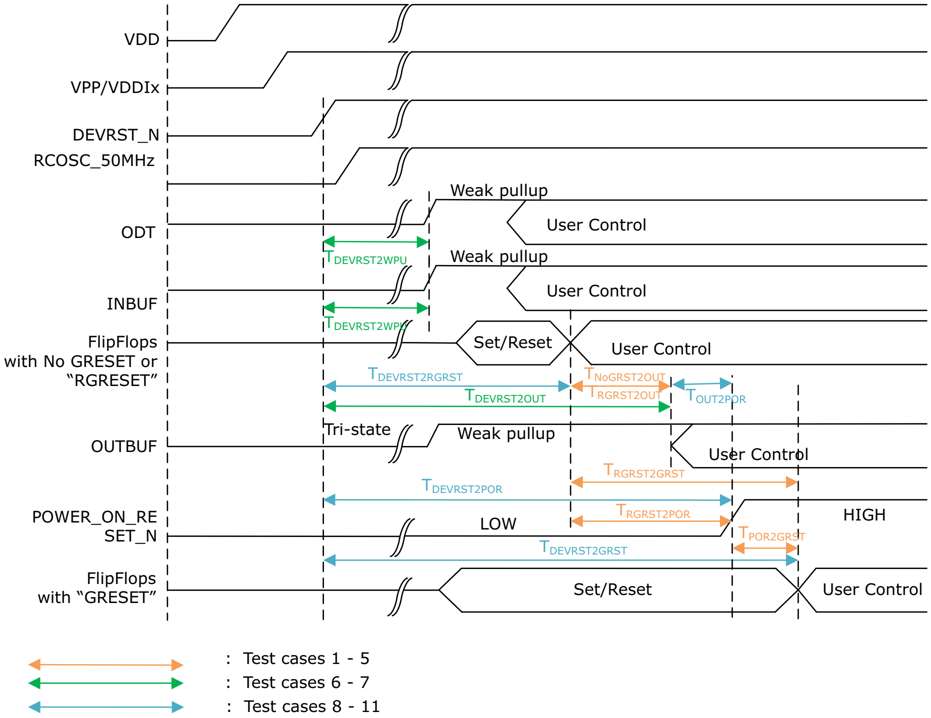
The following table lists how to measure DEVRST_N power-up to functional time data of RT4G150-LG1657 packaged devices.

*Table 7 •* **DEVRST_N Power-Up to Functional Time data**

| TEST CASE | Parameter | From | To | How to Measure |
|---|---|---|---|---|
| 1 | $T_{NoGRST2OUT}$ | Flipflops with No GRESET active | Output available | Fabric counter |
| 2 | $T_{RGRST2OUT}$ | Flipflops with “RGRESET” active | Output available | Fabric counter |
| 3 | $T_{RGRST2GRST}$ | Flipflops with “RGRESET” active | Flipflops with “GRESET” active | Fabric counter |
| 4 | $T_{RGRST2POR}$ | Flipflops with “RGRESET” active | POWER_ON_RESET_N | Fabric counter |
| 5 | $T_{POR2GRST}$ | POWER_ON_RESET_N | Flipflops with “GRESET” active | Fabric counter |
| 6 | $T_{DEVRST2OUT}$ | DEVRST_N | Output Available | Scoped |
| 7 | $T_{DEVRST2WPU}$ | DEVRST_N | WPU(weak pull up) | Scoped |
| 8 | $T_{DEVRST2RGRST}$ | DEVRST_N | Flipflops with “NO_GRESET” or “RGRESET” active | Calculated |
| 9 | $T_{DEVRST2GRST}$ | DEVRST_N | Flipflops with “GRESET” active | Calculated |
| 10 | $T_{DEVRST2POR}$ | DEVRST_N | POWER_ON_RESET_N | Calculated |
| 11 | $T_{OUT2POR}$ | Output available | POWER_ON_RESET_N | Calculated |

The following figure shows the behavior of different signals when DEVRST_N is asserted with parameter names. Typical conditions: VDD = 1.2 V, VDDI = 2.5 V, $T_J$ = 25 °C.

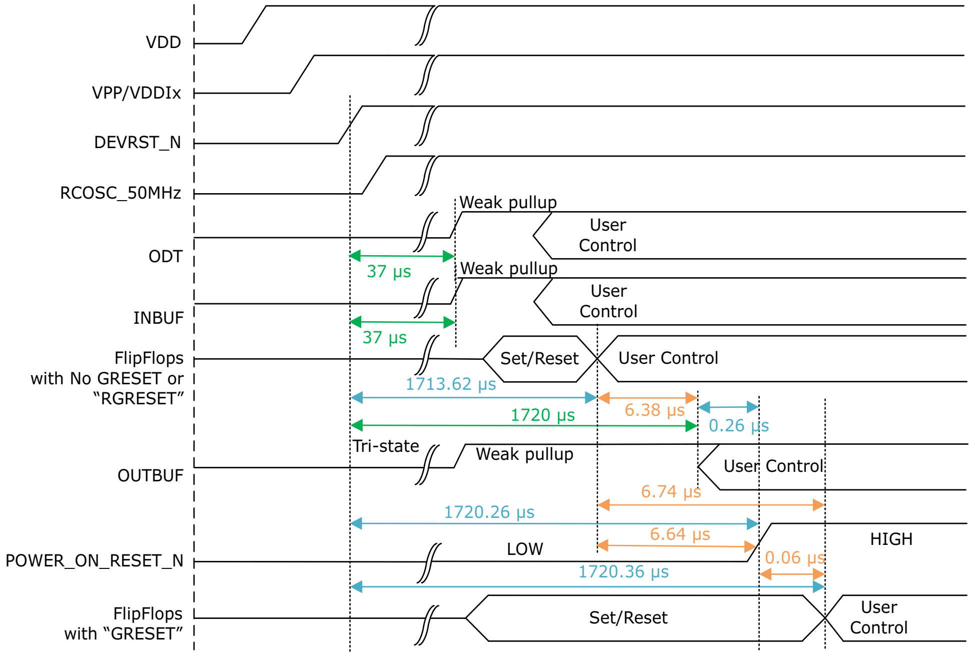
*Figure 12 •* **DEVRST_N Power-Up to Functional Timing**



**Note:** RCOSC_50MHZ represents the transition of the signal.

The following figure shows the behavior of different signals when DEVRST_N is asserted with timing numbers. Typical conditions: VDD = 1.2 V, VDDI = 2.5 V, $T_J$ = 25 °C.

*Figure 13 •* **DEVRST_N Power-Up to Functional Timing with Values**



**Note:** RCOSC_50MHZ represents the transition of the signal.

The following table lists DEVRST_N power-up to functional time data of RT4G150 (specifications for internal signals and I/O).

*Table 8 •* **DEVRST_N Power-Up to Functional Time_1**

| TEST CASE | Parameter | From | To | Delay µs |
|---|---|---|---|---|
| 1 | $T_{NoGRST2OUT}$ | Flipflops with No GRESET active | Output available | 6.38 |
| 2 | $T_{RGRST2OUT}$ | Flipflops with "RGRESET" active | Output available | 6.38 |
| 3 | $T_{RGRST2GRST}$ | Flipflops with "RGRESET" active | Flipflops with "GRESET" active | 6.74 |
| 4 | $T_{RGRST2POR}$ | Flipflops with "RGRESET" active | POWER_ON_RESET_N | 6.64 |
| 5 | $T_{POR2GRST}$ | POWER_ON_RESET_N | Flipflops with "GRESET" | 0.06 |

The following table lists DEVRST_N Power-up to functional time data of RT4G150 (specifications for DEVRST_N to weak pull - up and DEVRST_N to output).

*Table 9 •* **DEVRST_N Power-Up to Functional Time_2**

| TEST CASE | Parameter Name | From | To | Delay in µs |
|---|---|---|---|---|
| 6 | $T_{DEVRST2OUT}$ | DEVRST_N | Output Available | 1720 |
| 7 | $T_{DEVRST2WPU}$ | DEVRST_N | WPU(weak pull up) | 36.88 |

The following table lists the DEVRST_N to functional time data of RT4G150. Specification for DEVRST_N to internal signalizing numbers for VDD and DEVRST_N power-up to functional time are for typical conditions. See *DS0131: RTG4 FPGA Datasheet* for worst case conditions.

***Table 10 •*** **DEVRST_N Power-Up to Functional Time _3**

| TEST CASE | Parameter name | From | To | Delay in µs |
|---|---|---|---|---|
| 8 | $T_{DEVRST2RGRST}$ | DEVRST_N | Flipflops with No GRESET or “RGRESET” active | 1713 |
| 9 | $T_{DEVRST2GRST}$ | DEVRST_N | Flipflops with “GRESET” active | 1720 |
| 10 | $T_{DEVRST2POR}$ | DEVRST_N | POWER_ON_RESET_N | 1720 |
| 11 | $T_{OUT2POR}$ | Output available | POWER_ON_RESET_N | 0.26 |

### 2.4.2 System Controller Clock Requirements

The system controller is clocked by the on-chip 50 MHz RC Oscillator. It is not required to instantiate the Oscillator macro for system controller operations, because it has a dedicated hardwired connection from the 50 MHz RC oscillator.

The 50 MHz RC Oscillator is powered by the VDD power pins and does not require external components for operation. The on-chip Oscillator cannot be disabled if the system controller is not instantiated in a design.

## 2.5 Using System Controller

The following section and sub-sections describe how to use the system controller functions:

- Programming
- UJTAG
- Fabric Reset
- System Controller Suspend Mode

### 2.5.1 Programming

The RTG4 devices can be programmed using JTAG.

#### 2.5.1.1 JTAG Programming

An RTG4 device can be programmed using the dedicated JTAG interface. An external programmer, such as FlashPro4/FlashPro5, is used to program the device. The devices can be programmed in both single and chain modes.

### 2.5.2 UJTAG

If UJTAG is not held in Suspend mode, UJTAG does not perform the JTAG programming. The UJTAG interface is an extension of the external JTAG port in to the RTG4 devices, controlled by the TAP controller. UJTAG can be used to shift data or OPCODEs to and from the internal fabric logic. The UJTAG functionality is made available by instantiating the UJTAG macro from the Libero SoC IP catalog in to a SmartDesign or by instantiating it directly inside the HDL file. The real-time updating and monitoring of the internal behavior of the FPGA fabric are enabled using the UJTAG macro. See *AC227: Flash UJTAG Application Note* for more information on using the UJTAG module.

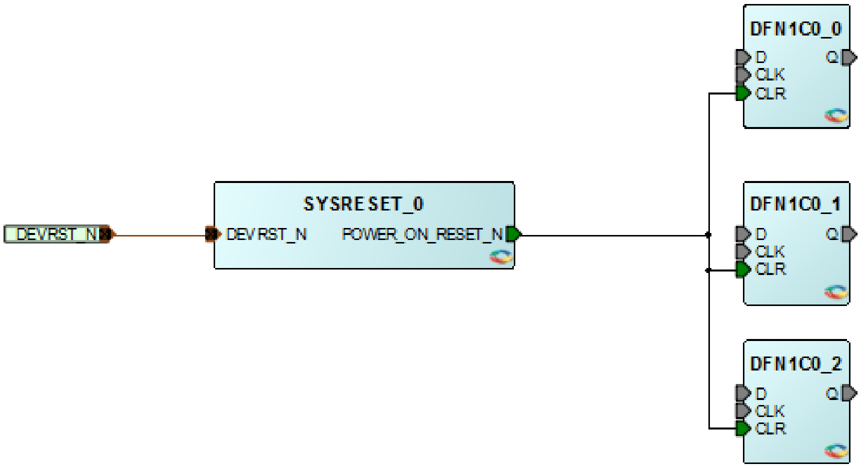
### 2.5.3 Fabric Reset

The POWER_ON_RESET_N signal is generated from the Internal_PO_RESET_N signal and can be used in the user design as a reset for the FPGA fabric. It is an active low output signal. It is made available by instantiating the SYSRESET macro from the Libero SoC IP catalog in SmartDesign or by instantiating the SYSRESET macro directly in the HDL file.

**Note:** The POR and DEVRST_N work regardless of Suspend mode.

The following figure shows the connection of SYSRESET macro with fabric logic.

***Figure 14 •*** **Fabric Logic Connection**



## 2.5.4 System Controller Suspend Mode

To protect the device from unintended behavior due to single event upset (SEUs), the system controller can be held in Suspend mode after device initialization.The system controller is active if the device is power-cycled or if a hard reset is applied, but it returns to Suspend mode once the initialization cycle is completed. A flash bit that is programmed during device programming controls the system controller suspend mode. This flash bit is not accessible from the customer design or by any external pin. The flash bit is only accessible through the programming file loaded into the device.

As the control bit is stored in a flash cell, it is immune to radiation effects due to the following:

- Neutrons or alpha particles in the terrestrial and airborne applications
- Heavy ions in the space applications

In the system controller suspend mode, the device can be reprogrammed or debugged using the JTAG port if the JTAG_TRSTB pin is High. If the JTAG_TRSTB pin is Low, all the other JTAG input signals are blocked from activating the system controller. For prototyping or debugging, the RTG4 device can be forced out of Suspend mode by toggling the JTAG_TRSTB pin to High and power cycling the device.

For programming, run **Scan Chain** using FlashPro software (from the **Menu** bar, click **Programmers > Scan Chain**), which keeps JTAG_TRSTB pin High. While keeping JTAG_TRSTB pin high, power-cycle the device which causes system controller to exit from suspend mode and then program the device.

The JTAG_TRSTB pin is read only during device power-up. When in space, the JTAG_TRSTB pin must be held Low using one of the following methods:

- Hardwired to ground
- Connected to ground through a jumper
- Tied to ground through a pull-down if an active device is included in the circuit to allow on-orbit reprogramming

To restore normal operation, the device must be reprogrammed using the JTAG port with the system controller suspend mode bit turned off, that is, disable the system controller suspend mode in Libero SoC software, regenerate the bitstream, and reprogram the device.

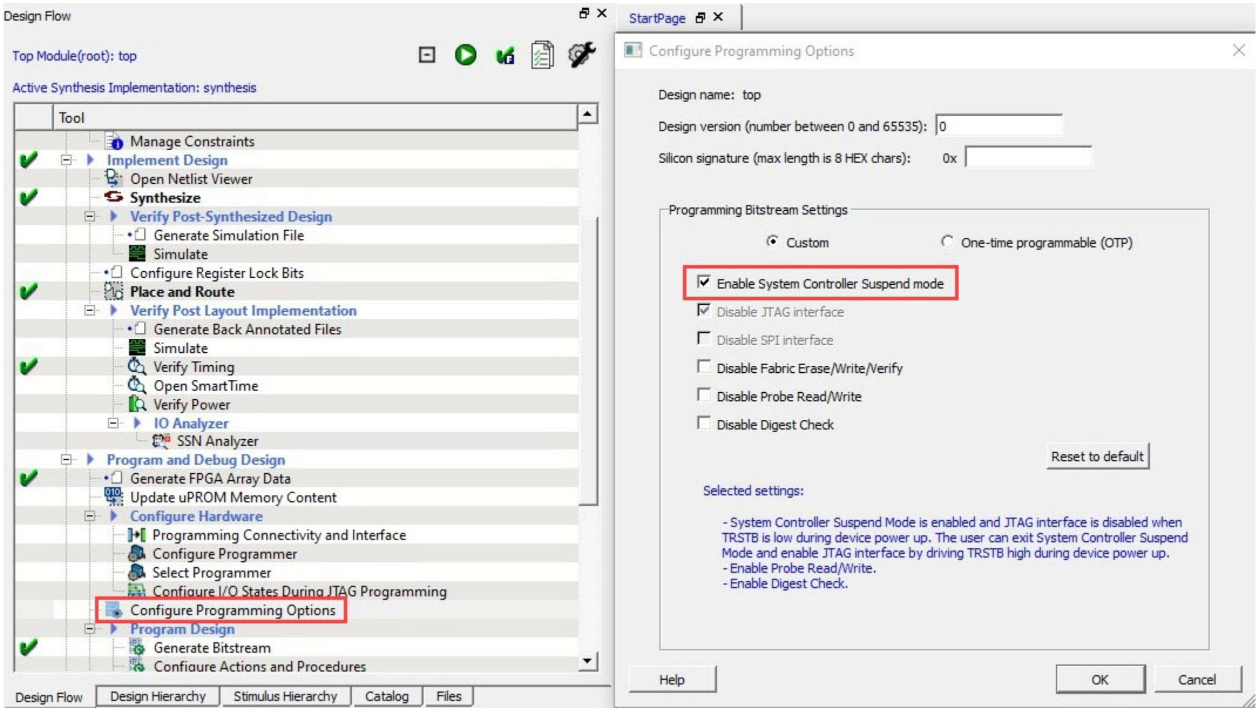
When RTG4 FPGAs are used in System controller suspend mode, device programming is disabled to protect the device from unintended programming because of single event upsets. After device initialization, the system controller is held in reset state and cannot provide system services such as security, IAP, or auto update programming. After the device exits system controller suspend mode, it can be programmed as usual.

If the System controller suspended mode is disabled, it increases vulnerability to radiation single event effects in the System controller.

### 2.5.4.1 Configuring System Controller Suspend Mode

The system controller suspend mode can be configured, that is, enabled or disabled in Libero SoC software as shown in the following figure.

***Figure 15 •*** **System Controller Suspend Mode**



Design name: top

Design version (number between 0 and 65535): 0

Silicon signature (max length is 8 HEX chars): 0x

Programming Bitstream Settings

● Custom ○ One-time programmable (OTP)

☑ Enable System Controller Suspend mode

☑ Disable JTAG interface

☐ Disable SPI interface

☐ Disable Fabric Erase/Write/Verify

☐ Disable Probe Read/Write

☐ Disable Digest Check

Selected settings:

- System Controller Suspend Mode is enabled and JTAG interface is disabled when TRSTB is low during device power up. The user can exit System Controller Suspend Mode and enable JTAG interface by driving TRSTB high during device power up.
- Enable Probe Read/Write.
- Enable Digest Check.