



ConnectCore 93

System-on-Module

Design Guidelines and Checklist

Revision history—90002624

Revision	Date	Description
A	September 2025	Initial release.

Trademarks and copyright

Digi, Digi International, and the Digi logo are trademarks or registered trademarks in the United States and other countries worldwide. All other trademarks mentioned in this document are the property of their respective owners.

© 2025 Digi International Inc. All rights reserved.

Disclaimers

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International. Digi provides this document “as is,” without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

Warranty

To view product warranty information, go to the following website:

www.digi.com/howtobuy/terms

Customer support

Gather support information: Before contacting Digi technical support for help, gather the following information:

- Product name and model
- Product serial number (s)
- Firmware version
- Operating system/browser (if applicable)
- Logs (from time of reported issue)
- Trace (if possible)
- Description of issue
- Steps to reproduce

Contact Digi technical support: Digi offers multiple technical support plans and service packages. Contact us at +1 952.912.3444 or visit us at www.digi.com/support.

Feedback

To provide feedback on this document, email your comments to

techcomm@digi.com

Include the document title and part number (ConnectCore 93 Design Guidelines and Checklist, 90002624 A) in the subject line of your email.

Contents

About this document

Overview	5
Naming and signal conventions	5

Schematics design checklist

Bring-up guidelines

Power-up	10
Power rails	10
Verify system boot	10

About this document

Overview

The purpose of this document is to ease and drive the hardware design process of ConnectCore 93 SOM-based products.

Schematics milestones keep bring-up of your designs on track for success, and layout recommendations help achieve optimal electrical behavior.

This document, together with the [System-on-module Hardware Reference Manual](#), provide information key to ensuring a functional, quality design featuring the ConnectCore 93 module.

Naming and signal conventions

This document is written assuming the reader has a good working knowledge of common electronics terminology. You may also encounter the following specific conventions:

Convention	Definition
signal_name#	A hash sign at the end of a signal name denotes that it is an active low signal.
_B	An alternate way to indicate a signal is active low.
x	Indicates an alphanumeric value.
Register/signal [a:b]	Brackets after a name refer to the bits or signals contained in a register, or in a set of related signals.

Schematics design checklist

Item # name	Description	Check
Power supply		
1. Input power of the SOM	The one and only input supply of the SOM is VSYS/VSYS2, which powers directly the PMIC of the module. Recommendation is to include some overvoltage and overcurrent protection upstream these inputs of the SOM.	
2. 3V3	3V3 power supply is a high output current buck regulator available for powering external circuitry on the carrier board.	
3. 1V8	1V8 power supply is a high output current buck regulator available for powering external circuitry on the carrier board.	
4. MUX_3V3_1V8	MUX_3V3_1V8 is a regulator intended to be used as the power supply for an SDIO interface capable of reaching higher speed modes (208 MHz). In order to operate at this speed, the interface has to work at 1.8 V. The selection of the operating voltage of this regulator is done through SD_VSELECT pin (AH13): ▪ If SD_VSELECT is at low level, MUX_3V3_1V8 operates at 3.3 V. ▪ If SD_VSELECT is at high level, MUX_3V3_1V8 operates at 1.8 V. In any case, this regulator can be use for any other purpose on the carrier board.	
BOOT		
1. BOOT_MODE [3:0]	These lines select the boot mode of the SOM:	

Item # name	Description	Check																																																							
	<table><tr><th>BOOT_MODE[3:0]</th><th>BOOT CORE</th><th>BOOT DEVICE</th><th>COMMENT</th></tr><tr><td>0000</td><td>Cortex-A55</td><td>From internal fuses</td><td rowspan="6">USB1/2 with SFDP (JESD-216) discoverable parameters</td></tr><tr><td>0001</td><td>Cortex-A55</td><td>Serial Downloader</td></tr><tr><td>0010</td><td>Cortex-A55</td><td>USDHC1 8-bit eMMC 5.1</td></tr><tr><td>0011</td><td>Cortex-A55</td><td>USDHC2 4-bit SD3.0</td></tr><tr><td>0100</td><td>Cortex-A55</td><td>FlexSPI Serial NOR</td></tr><tr><td>0101</td><td>Cortex-A55</td><td>FlexSPI Serial NAND 2K page</td></tr><tr><td>0110</td><td>Cortex-A55</td><td>Infinite Loop</td><td rowspan="6">USB1 with SFDP (JESD-216) discoverable parameters</td></tr><tr><td>0111</td><td>Cortex-A55</td><td>Test Mode</td></tr><tr><td>1000</td><td>Cortex-M33</td><td>From internal fuses</td></tr><tr><td>1001</td><td>Cortex-M33</td><td>Serial Downloader</td></tr><tr><td>1010</td><td>Cortex-M33</td><td>USDHC1 8-bit eMMC 5.1</td></tr><tr><td>1011</td><td>Cortex-M33</td><td>USDHC2 4-bit SD3.0</td></tr><tr><td>1100</td><td>Cortex-M33</td><td>FlexSPI Serial NOR</td><td rowspan="4">with SFDP (JESD-216) discoverable parameters</td></tr><tr><td>1101</td><td>Cortex-M33</td><td>FlexSPI Serial NAND 2K page</td></tr><tr><td>1110</td><td>Cortex-M33</td><td>Infinite Loop</td></tr><tr><td>1111</td><td>Cortex-M33</td><td>Test Mode</td></tr></table> BOOT_MODE1 has an internal pull-up to configure the SOM to boot by default from the internal eMMC flash. It is recommended not to leave the BOOT_MODE lines floating, so in order to boot from the internal eMMC drive BOOT_MODE0, BOOT_MODE2 and BOOT_MODE3 low in the carrier board. In order to drive the BOOT_MODE lines recommendation is to use: ■ 4.7K pull-up resistors to 3V3 to set them to high level (logic “1”). ■ 100 K pull-down resistors to GND to set them to low level (logic “0”). Because BOOT_MODE1 has an internal 4.7 K resistor, connect this pin directly to ground.	BOOT_MODE[3:0]	BOOT CORE	BOOT DEVICE	COMMENT	0000	Cortex-A55	From internal fuses	USB1/2 with SFDP (JESD-216) discoverable parameters	0001	Cortex-A55	Serial Downloader	0010	Cortex-A55	USDHC1 8-bit eMMC 5.1	0011	Cortex-A55	USDHC2 4-bit SD3.0	0100	Cortex-A55	FlexSPI Serial NOR	0101	Cortex-A55	FlexSPI Serial NAND 2K page	0110	Cortex-A55	Infinite Loop	USB1 with SFDP (JESD-216) discoverable parameters	0111	Cortex-A55	Test Mode	1000	Cortex-M33	From internal fuses	1001	Cortex-M33	Serial Downloader	1010	Cortex-M33	USDHC1 8-bit eMMC 5.1	1011	Cortex-M33	USDHC2 4-bit SD3.0	1100	Cortex-M33	FlexSPI Serial NOR	with SFDP (JESD-216) discoverable parameters	1101	Cortex-M33	FlexSPI Serial NAND 2K page	1110	Cortex-M33	Infinite Loop	1111	Cortex-M33	Test Mode	
BOOT_MODE[3:0]	BOOT CORE	BOOT DEVICE	COMMENT																																																						
0000	Cortex-A55	From internal fuses	USB1/2 with SFDP (JESD-216) discoverable parameters																																																						
0001	Cortex-A55	Serial Downloader																																																							
0010	Cortex-A55	USDHC1 8-bit eMMC 5.1																																																							
0011	Cortex-A55	USDHC2 4-bit SD3.0																																																							
0100	Cortex-A55	FlexSPI Serial NOR																																																							
0101	Cortex-A55	FlexSPI Serial NAND 2K page																																																							
0110	Cortex-A55	Infinite Loop	USB1 with SFDP (JESD-216) discoverable parameters																																																						
0111	Cortex-A55	Test Mode																																																							
1000	Cortex-M33	From internal fuses																																																							
1001	Cortex-M33	Serial Downloader																																																							
1010	Cortex-M33	USDHC1 8-bit eMMC 5.1																																																							
1011	Cortex-M33	USDHC2 4-bit SD3.0																																																							
1100	Cortex-M33	FlexSPI Serial NOR	with SFDP (JESD-216) discoverable parameters																																																						
1101	Cortex-M33	FlexSPI Serial NAND 2K page																																																							
1110	Cortex-M33	Infinite Loop																																																							
1111	Cortex-M33	Test Mode																																																							
2. BOOT_MODE pins	The boot mode can be selected in two ways: ■ Boot mode pins (recommended for development, default configuration) ■ Fuses (recommended for production) When booting from the boot mode pins, the boot mode is determined by the binary value stored in the internal BOOT_MODE register. Boot is initialized by sampling the BOOT_MODE[3:0] inputs on the rising edge of POR_B. After these inputs are sampled, the subsequent states do not affect BOOT_MODE internal register value. BOOT_MODE[3:0] bits are available in the following signals of the SOM:																																																								

Item # name	Description		Check										
	<table><tr><th>Signal</th><th>Boot mode</th></tr><tr><td>BT_UART1_TX</td><td>BOOT0</td></tr><tr><td>BT_UART1_RTS</td><td>BOOT1</td></tr><tr><td>SPI1_CS0</td><td>BOOT2</td></tr><tr><td>SPI1_SCK</td><td>BOOT3</td></tr></table>	Signal	Boot mode	BT_UART1_TX	BOOT0	BT_UART1_RTS	BOOT1	SPI1_CS0	BOOT2	SPI1_SCK	BOOT3		
	Signal	Boot mode											
	BT_UART1_TX	BOOT0											
	BT_UART1_RTS	BOOT1											
	SPI1_CS0	BOOT2											
	SPI1_SCK	BOOT3											
Digi recommends adding a buffer to protect these lines during booting. External circuitry and peripherals connected to these lines can modify the expected value when system is booting, so that the process can fail.													
3. Recover/Debug mode (USB)	Recommendation is to give access to USB1 interface of the CPU in order to have a recovery/debug mechanism.												
Console debug port													
1. UART6	The default U-Boot loaded in the modules from manufacturing configures UART6 as the console debug port. This console port can be changed to any other UART in the system. Digi recommends using the default UART as the console port so no additional efforts (such as compiling a new U-Boot, loading it through USB, etc.) are required.												
Wireless and Bluetooth													
1. Wireless USDHC	USDHC3 interface of the ConnectCore 93 module is used internally in the SOM for the wireless connectivity. The following pads must be left unconnected to preserve this wireless functionality: ▪ D12 (WLAN_SD3_CLK) ▪ D11 (WLAN_SD3_CMD) ▪ D9 (WLAN_SD3_D0) ▪ D8 (WLAN_SD3_D1) ▪ D6 (WLAN_SD3_D2) ▪ D5 (WLAN_SD3_D3)												

Item # name	Description	Check
	This applies only to Wireless variants of the SOM. In non-wireless variants, this USDHC bus is available.	
2. Bluetooth UART	UART1 port of the ConnectCore 93 module is used internally in the SOM for Bluetooth connectivity. The following pads must be left unconnected to preserve this Bluetooth functionality: ▪ E13 (BT_UART1_TX) ▪ E12 (BT_UART1_CTS) ▪ F12 (BT_UART1_RX) ▪ F11 (BT_UART1_RTS) This applies only to Wireless variants of the SOM. In non-wireless variants, this UART port is available.	
CPU control lines		
1. SYS_RESET	SYS_RESET is the default reset line of the ConnectCore 93. It is active low and has an internal pull-up .	
2. ON_OFF	ON_OFF is the default power-on/off line of the ConnectCore 93. It is active low and has an internal pull-up .	
3. PWR_ON	PWR_ON is a line internally controlled by the CPU. Recommendation is to leave this signal unconnected and never drive it externally.	

Bring-up guidelines

This section offers guidelines for performing bring-up of a ConnectCore 93-based product.

Power-up

The very first power-up of a product is one of the most critical step of the bring-up. Use a power supply with current limiter so the system is protected in case an unexpected short (or any other high-current incident) occurs. The ConnectCore 93 SOM will have a static power consumption around 200-250 mA from factory mode (except booting peaks), where only U-Boot is loaded into the device. So, together with the expected power consumption of the carrier board, you can estimate an upper limit for the current consumption in the bring-up.

Power rails

Once you have verified that the power consumption of the system is acceptable and no shorts are present, the next step is to verify that all power rails also have their expected voltage value. Both 1V8 and 3V3 must have a stable 1.8 V and 3.3 V values, respectively. If that is the case, it means that the system is completely powered. The PMIC has already initialized and the SOM should boot.

Verify system boot

If everything has gone right and the system has booted, you should be able to communicate with the SOM through the console port of the system. If this is not the case, check the following:

- Make sure the UART debug port lines are not crossed. Crossing the TX and RX lines of the console UART is a common mistake.
- Verify the value of the BOOT_MODE lines.
- Verify that POR_B signal is at high level.