Hardware Development Guide for i.MX 6SoloLite Applications Processors

IMX6SLHDG
Rev 1
06/2013

Contents

Paragraph Number	Title	Page Number

Chapter 1
Design Checklist

1.1 Design checklist overview 1-1
1.2 Design checklist tables 1-1
1.3 Bus isolation circuit 1-10
1.4 DDR reference circuit 1-11
1.5 I2C clock speed and division factors (IFDR) 1-11
1.6 JTAG signal termination 1-14
1.7 Oscillator tolerance 1-14
1.8 Unused analog interfaces 1-14

Chapter 2
i.MX 6 Series Layout Recommendations

2.1 Basic design recommendations 2-1
2.1.1 Fanout illustrations 2-3
2.1.2 Placing decoupling capacitors 2-4
2.2 Stackup recommendations 2-4
2.3 DDR connection information 2-6
2.4 DDR routing rules 2-7
2.5 Routing considerations 2-9
2.5.1 Swapping data lines 2-9
2.5.2 DDR3 (32 bits) T topology considerations 2-9
2.5.3 DDR3 (32 bits) Fly-by topology considerations 2-9
2.5.4 2-Gigabyte recommendations 2-10
2.5.5 1-Gigabyte recommendations 2-10
2.5.6 Four chips T topology routing examples 2-12
2.5.7 LPDDR2 FBGA 168 routing example 2-16
2.5.8 High speed signal routing recommendations 2-19
2.5.9 Ground plane recommendations 2-19
2.6 DDR power recommendations (for DDR3 only) 2-22
2.7 USB recommendations 2-23
2.8 Impedance signal recommendations 2-23
2.9 Reference resistors 2-25
2.10 ESD and radiated emissions recommendations 2-25
2.11 Component placement recommendations 2-25
2.12 Reducing skew and phase problems in deferential pairs traces 2-26

Contents

Paragraph Number	Title	Page Number

Chapter 3
Requirements for Power Management

3.1 Power management requirements overview ... 3-1
3.1.1 Voltage domains overview ... 3-1
3.1.2 PF0100 overview ... 3-1
3.2 Requirements for a generic interface between chip and PF0100 ... 3-1
3.3 i.MX6 internal regulators ... 3-3
3.4 Connection diagrams ... 3-4
3.5 Video power recommendations ... 3-5

Chapter 4
Using the Clock Connectivity Table

4.1 Root clocks ... 4-1
4.2 Waking the core up from stop mode ... 4-2

Chapter 5
Using the IOMUX Design Aid

5.1 Application requirements ... 5-2
5.2 IOMUX tool version ... 5-2
5.3 IOMUX tool location ... 5-2
5.4 Learning to use the IOMUX tool ... 5-2

Chapter 6
Configuring JTAG Tools

6.1 JTAG tool requirements ... 6-1
6.2 Extra JTAG functionality ... 6-1
6.3 Updating your RealView ICE ... 6-2
6.4 Defining the JTAG chain ... 6-3
6.5 Reading a register with RealView Debugger v4.1 ... 6-6
6.6 CoreSight Base address references ... 6-8

Chapter 7
Avoiding Board Bring-up Problems

7.1 Power system overview ... 7-1
7.1.1 Block diagram ... 7-1
7.1.2 Maximum supply requirements ... 7-2

Contents

Paragraph Number	Title	Page Number

7.1.3 Internal LDO description ... 7-3
7.2 Power sequencing ... 7-4
7.2.1 Detailed power-up sequence ... 7-5
7.2.2 VDD_SNVS_IN power-up ramp rate ... 7-6
7.2.3 Power-down sequence ... 7-6
7.2.4 Power-up sequence in low voltage system design ... 7-6
7.3 External 32 kHz clock requirements ... 7-7
7.4 PMIC Handshaking Signals ... 7-9
7.5 Watchdog Reset Behavior ... 7-10
7.6 Processor Rail Capacitance ... 7-10
7.7 Power Modes ... 7-12
7.8 Regulator Requirements ... 7-14
7.9 Checking for clock pitfalls ... 7-16
7.10 Avoiding reset pitfalls ... 7-17
7.11 Sample board bring-up checklist ... 7-18

Chapter 8
Understanding the IBIS Model

8.1 IBIS structure and content ... 8-1
8.2 Header Information ... 8-2
8.3 Component and pin information ... 8-2
8.4 Model information ... 8-4
8.4.1 IV information ... 8-5
8.4.2 VT information ... 8-5
8.4.3 Golden Model VT information ... 8-7
8.5 Freescale naming conventions for model names and usage in i.MX6 IBIS file ... 8-8
8.5.1 [Model Selector] ddr ... 8-8
8.5.1.1 DDR [Model Selector] ... 8-8
8.5.2 [Model Selector] gpio ... 8-10
8.5.3 [Model Selector] USB ... 8-10
8.5.4 List of pins not modeled in the i.MX6 IBIS file ... 8-11
8.6 Quality assurance for the IBIS models ... 8-11
8.7 IBIS usage ... 8-12
8.8 References ... 8-12

Chapter 9
Using the Manufacturing Tool

9.1 Overview ... 9-1
9.2 Feature summary ... 9-1

Contents

Paragraph Number	Title	Page Number
9.3	Other references	9-2

Chapter 10
Using BSDL for Board-level Testing

10.1	BSDL overview	10-1
10.2	How BSDL functions	10-1
10.3	Downloading the BSDL file	10-1
10.4	Pin coverage of BSDL	10-1
10.5	Boundary scan operation	10-2
10.6	I/O pin power considerations	10-2

Chapter 11
Using the FEC Interface

11.1	Overview	11-1
11.2	Configuring the FEC signal connections	11-1
11.3	Generating the reference clock	11-3
11.4	Generating the reference clock on chip	11-3
11.5	Using an external clock	11-3

Appendix A
Development Platforms

Appendix B
Revision History

Chapter 1
Design Checklist

1.1 Design checklist overview

This chapter provides a design checklist for the i.MX 6SoloLite. For information on the i.MX 6Quad, 6Dual, 6DualLite, and 6Solo processors, see the *Hardware Development Guide for i.MX 6Quad, 6Dual, 6DualLite, 6Solo Families of Applications Processors (IMX6DQ6SDLHDG)*.

The design checklist tables (Table 1-1–Table 1-10) contain recommendations for optimal design. Where appropriate, the checklist tables also provide an explanation of the recommendation so that users have a greater understanding of why certain techniques are recommended. All supplemental tables referenced by the checklist appear in sections following the design checklist tables.

See also the application note *Common Hardware Design for i.MX 6Dual/6Quad and i.MX 6Solo/6DualLite* (AN4397).

1.2 Design checklist tables

Table 1-1. DDR recommendations

Checkbox	Recommendation	Explanation/supplemental recommendation
	1. Connect ZQPAD to an external 240 Ω 1% resistor to GND.	This is a reference used during DRAM output buffer driver calibration.
	2. Connect DRAM_VREF to a source that is 50% of the voltage value of NVCC_DRAM.	• The user may tie DDR_VREF to a precision external resistor divider. Shunt each resistor with a closely-mounted 0.1 µF capacitor. See Table 1-11 for resistor values. Using resistors with recommended tolerances ensures the ±2% DDR_VREF tolerance per the LPDDR2 and DDR3 specifications. • The user can use a PMIC’s tracking regulator as used on Freescale reference designs. A tracking regulator is recommended as a reference for memory configurations of more than four devices.

Table 1-1. DDR recommendations (continued)

Checkbox	Recommendation	Explanation/supplemental recommendation
	3. Connect DRAM_RESET to a 10 kΩ 5% pulldown resistor to GND.	• DDR3: DRAM_RESET should be pulled down to meet the JEDEC sequence until the controller is configured and starts driving. DRAM_RESET should be kept high when DDR3 enters self-refresh mode. • LPDDR2: DRAM_RESET should be left unconnected. Some Freescale reference designs use a 1% resistor simply to consolidate the BOM. DRAM_RESET is an active-low signal.
	4. DRAM_SDCKE0 and DRAM_SDCKE1 should be connected to individual 10 kΩ 5% resistors to GND.	To minimize current drain, the Freescale BSP (board-support package) disables the EMI I/O during DSM (deep-sleep mode). A pull-down resistor ensures that the DRAM is in the proper state during DSM. • For LPDDR2: SDCKE[1:0] must be pulled down to meet the JEDEC sequence until the controller is configured and starts driving. • For DDR3: SDCKE[1:0] pull-down is not required to meet JEDEC.

Table 1-2. EIM recommendations for developer's boot modes

Checkbox	Recommendation	Explanation/supplemental recommendation
	1. Use isolation buffers and series resistors on the development board design if the boot configuration bus signals (muxed with the EIM bus) are used as the system's EIM signals, GPIO, or other functions after boot.	Isolation buffers are required because 6SoloLite pads can be set to 1.8V or 3V logic levels while boot signals require 3V logic level to latch the boot settings. Any pull-ups for boot configuration line must be properly isolated because there is a chance that the system may use 1.8V logic levels. See Figure 1-1 for an example circuit.
	2. To reduce incorrect boot-up mode selections, do one of the following: • Use EIM boot interface lines only as processor outputs. Ensure EIM boot interface lines are not loaded down such that the level is interpreted as low during power-up, when the intent is to be a high level, or vice versa. • If an EIM boot signal must be configured as an input, isolate the EIM signal from the target driving source with one analog switch and apply the logic value with a second analog switch. Alternately, peripheral devices with three-state outputs may be used; ensure the output is high-impedance during the boot up interval.	Using EIM boot interface lines as inputs may result in a wrong boot up due to the source overcoming the pull resistor value. A peripheral device may require the EIM signal to have an external or on-chip resistor to minimize signal floating. If the usage of the EIM boot signal affects the peripheral device, then an analog switch, open collector buffer, or equivalent should isolate the path. A pullup or pulldown resistor at the peripheral device may be required to maintain the desired logic level. Review the switch or device data sheet for operating specifications.
	3. The BOOT_CFG signals are required for proper functionality and operation and should not be left floating.	See the "System Boot" chapter in your chip reference manual for the correct boot configuration. Note that an incorrect setting may result from an improper booting sequence.

Table 1-3. Boot mode input recommendations

Checkbox	Recommendation	Explanation/supplemental recommendation
	1. For BOOT_MODE1 and BOOT_MODE0, use one of the following options to achieve logic 0: • Tie to GND through any size external resistor • Tie directly to GND • Float For logic 1, use one of the following: • Tie directly to the VDD_SNVS_IN rail • Tie to the VDD_SNVS_IN rail through an external resistor 10 kΩ. A value of 4.7 kΩ is preferred in high-noise environments. If switch control is desired, no external pulldown resistors are necessary. Simply connect SPST switches directly to the VDD_SNVS_IN rail. If desired, a 4.7 kΩ to 10 kΩ series resistor can be used when current drain is critical.	Boot inputs BOOT_MODE1 and BOOT_MODE0 each have on-chip pulldown devices with a nominal value of 100 kΩ, a projected minimum of 60 kΩ, and a projected maximum of 140 kΩ. Be aware that when these are logic high, current is drawn from the VDD_SNVS supply. In production, when on-chip fuses determine the boot configuration, both boot mode inputs can be no connects.

Table 1-4. I^{2}C recommendations

Checkbox	Recommendation	Explanation/supplemental recommendation
	1. Verify the target I^{2}C interface clock rates.	The bus can only operate as fast as the slowest peripheral on the bus. If faster operation is required, move the slow devices to another I^{2}C port.
	2. Verify that the target I^{2}C address range is supported and does no conflict with other peripherals. If there is an unavoidable address conflict, move the offending device to another I^{2}C port.	These chips support up to: • Three I^{2}C ports for the i.MX 6SoloLite If it is undesirable to move a conflicting device to another I^{2}C port, review the peripheral operation to see if it supports remapping the address.
	3. Do not place more than one set of pullup resistors on the I^{2}C lines.	This can result in excessive loading. Good design practice is to place one pair of pullups only.

Table 1-5. JTAG recommendations

Checkbox	Recommendation	Explanation/supplemental recommendation
	1. Do not use external pullup or pulldown resistors on JTAG_TDO.	JTAG_TDO is configured with an on-chip keeper circuit such that the floating condition is actively eliminated if an external pull resistor is not present. An external pull resistor on JTAG_TDO is detrimental. See Table 1-13 for a summary of the JTAG interface.

Table 1-5. JTAG recommendations (continued)

Checkbox	Recommendation	Explanation/supplemental recommendation
	2. Ensure that the on-chip pullup/pulldown configuration is followed. If external resistors are used with JTAG signals, with the exception of JTAG_TDO. For example, do not use an external pulldown on an input that has an on-chip pullup.	External resistors can be used with all JTAG signals except JTAG_TDO, but they are not required. See Table 1-13 for a summary of the JTAG interface.
	3. JTAG_MOD may be referred to as SJC_MOD in some documents. Both names refer to the same signal. JTAG_MOD should be externally connected to GND for normal operation in a system. Termination to GND through an external pulldown resistor is allowed. Use $\leq$ 4.7 kΩ.	When JTAG_MOD is low, the JTAG interface is configured for common software debug, adding all the system taps to the chain. When JTAG_MOD is high, the JTAG interface is configured to a mode compliant with the IEEE 1149.1 standard.

Table 1-6. Power and decouple recommendations

Checkbox	Recommendation	Explanation/supplemental recommendation
	1. Comply with the power-up sequence guidelines as described in the data sheet to guarantee reliable operation of the device.	Any deviation from these sequences may result in the following situations: • Excessive current during power-up phase • Prevention of the device from booting • Irreversible damage to the processor (worst-case scenario)
	2. Do not overload coin cell backup power rail VDD_SNVS_IN. Note that the following I/Os are associated with VDD_SNVS_IN; most inputs have on-chip pull resistors and do not require external resistors: • POR_B – on-chip pullup; see Table 1-8 #1 • ONOFF – on-chip pullup; see Table 1-8 #2 • BOOT_MODE0 – on-chip pulldown; see Table 1-3 #1 • BOOT_MODE1 – on-chip pulldown; see Table 1-3 #1 • TAMPER – on-chip pulldown • PMIC_STBY_REQ – push-pull output • PMIC_ON_REQ – push-pull output • TEST_MODE – on-chip pulldown; see Table 1-10 #1	Freescale PMIC PMPF0100 VSNVS regulator is rated to supply 400 μA output current under worst-case operating conditions. The VDD_SNVS_IN regulator can supply larger current in transient situations without damaging the regulator. Concerning i.MX6: • When VDD_SNVS_IN = VDD_HIGH_IN, SNVS domain current is drawn from both equally. • When VDD_HIGH_IN > VDD_SNVS_IN, VDD_HIGH_IN supplies all SNVS domain current and current flows into VDD_SNVS_IN to charge a coin cell battery. • When VDD_SNVS_IN > VDD_HIGH_IN, VDD_SNVS_IN supplies current to SNVS, and some current flows into VDD_HIGH_IN. **Note:**VDD_HIGH_IN must be valid (above the internal detector threshold) for the current flow to occur. Thus, current flow only happens when VDD_HIGH_IN is powered to a level below VDD_SNVS_IN. If VDD_HIGH_IN is off or low, no extra current is drawn from VDD_SNVS_IN. The whole circuit assumes it is charging a coin cell and starts charging when VDD_HIGH_IN is valid. If you are driving VDD_SNVS_IN with a non-battery power source, it must be at the same level as VDD_HIGH_IN or current will flow between them. • When VDD_SNVS_IN is not powered by a battery, it is recommended that VDD_SNVS_IN = VDD_HIGH_IN. If VDD_SNVS_IN is tied to a battery, the battery eventually discharges to a value equal to that of VDD_HIGH_IN and never subsequently charges above VDD_HIGH_IN. The battery chemistry may add restrictions to VDD_HIGH_IN's voltage range. External charging components should be based on the battery manufacturer's specifications.

Table 1-6. Power and decouple recommendations (continued)

Checkbox	Recommendation	Explanation/supplemental recommendation
	3. Only one 22 μF bulk capacitor should be connected to each of these on-chip LDO regulator outputs: • VDD_ARM_CAP • VDD_SOC_CAP • VDD_PU_CAP A 22 μF bulk capacitor must be placed as near as possible with pins/vias. The distance should be less than 50mil between bulk cap and VDD_xx_CAP pins. Decoupling capacitors such as 0.1 μF or 0.22 μF should also be used.	If the nominal capacitance value is larger than recommended, power-up ramp time is excessive and operation cannot be guaranteed. Note that the ramp up time is constant. Larger capacitors mean more inrush current. Select small capacitors with low ESR (equivalent series resistance). The 22 μF bulk capacitors should be placed as close as possible to the associated VDD_xx_CAP ball, with trace widths and via sizes appropriate to the expected current draw. A trace length of less than 50 mil is recommended. Do not connect any loads to these LDO outputs: VDDARM_CAP, VDDARM23_CAP, or VDDPU_CAP. VDDSOC_CAP is restricted to MX6 loads.
	4. Only one 10 μF bulk capacitor should be connected to each of these on-chip LDO regulator outputs: • VDD_HIGH_CAP • NVCC_PLL • VDD_USB_CAP. Decoupling capacitors such as 0.1 μF or 0.22 μF should also be used.	If the nominal capacitance value is larger than recommended, power-up ramp time is excessive and operation cannot be guaranteed. Select small capacitors with low ESR. These LDOs should only be used to power the loads as described in the reference manual or data sheet. Do not connect any loads to these LDO outputs: NVCC_PLL_OUT or VDDUSB_CAP. VDDHIGH_CAP is restricted to MX6 loads.
	5. One .22 μF decoupling capacitor should be connected to VDD_SNVS_CAP, an on-chip LDO regulator output. A bulk capacitor is not necessary.	If the nominal value is larger than recommended, power-up/down ramp time is excessive and suspend/resume operation cannot be guaranteed. Select a small capacitor with low ESR. Do not connect an external load to this LDO output. **Note:** Larger cap values on VDD_SNVS_CAP slow the ramp rate. If the VDD_SNVS_IN's ramp rate is significantly faster than VDD_SNVS_CAP's ramp rate, excess current consumption may result in SNVS_IN.
	6. Maximum ripple voltage requirements.	Common requirement for ripple noise should be less than 5% Vp-p of supply voltage average value. Related power rails affected: all VDD_xxx_IN and VDD_xxx_CAP.
	7. If VDD_SNVS_IN is directly supplied by a coin cell, a schottky diode is required between VDD_HIGH_IN and VDD_SNVS_IN. The cathode is connected to VDD_SNVS_IN. Alternately, VDD_HIGH_IN and VDD_SNVS_IN can be tied together if the real-time clock function is not needed during system power-down.	When no power is supplied to VDD_VSNVS_IN, the diode limits the voltage difference between the two on-chip SNVS power domains to approximately 0.3 V. The processor is designed to allow current flow between the two SNVS power domains proportional to the voltage difference.

Table 1-7. Oscillator and clock recommendations

Checkbox	Recommendation	Explanation/supplemental recommendation
	1. Precision 32.768 kHz oscillator Connect a crystal between RTC_XTALI and RTC_XTALO. Choose a crystal with a maximum of 100 kΩ ESR (equivalent series resistance) and follow the manufacturer's recommendation for loading capacitance. Do not use an external biasing resistor because the bias circuit is on-chip.	The capacitors implemented on either side of the crystal are about twice the crystal load capacitance. To hit the target oscillation frequency, board capacitors need to be reduced to compensate for board and chip parasitic capacitance; typically 15–16 pF is employed. The integrated oscillation amplifier has an on-chip self-biasing scheme, but is high-impedance (relatively weak) to minimize power consumption. Care must be taken to limit parasitic leakage from RTC_XTALI and RTC_XTALO to either power or ground (> 100 MΩ) as this negatively affects the amplifier bias and causes a reduction of startup margin. Use short traces between the crystal and the processor, with a ground plane under the crystal, load capacitors, and associated traces.
	2. External kilohertz source If feeding an external clock into the device, RTC_XTALI can be driven DC-coupled with RTC_XTALO floated or driven with a complimentary signal.	The voltage level of this driving clock should not exceed the voltage of VDD_SNVS_CAP and the frequency should be <100 kHz under typical conditions. Do not exceed VDD_SNVS_CAP or damage/malfunction may occur. The RTC_XTALI signal should not be driven if the VDD_SNVS_CAP supply is off. This can lead to damage or malfunction. For RTC_XTALI VIL and VIH voltage levels, see the latest i.MX 6 series datasheet. Note that if this external clock is stopped, the internal ring oscillator starts automatically.
	3. Loose-tolerance 40 kHz oscillator An on-chip loose-tolerance ring oscillator is available of approximately 40 kHz. If RTC_XTALI is tied to GND and RTC_XTALO is floating, the on-chip oscillator is automatically engaged.	When a high-accuracy real-time clock is not required, the system may use the on-chip 40 kHz oscillator. The tolerance is ± 50%. The ring oscillator starts faster than an external crystal and is used until the external crystal reaches stable oscillation. The ring oscillator also starts automatically if no clock is detected at RTC_XTALI at any time.
	4. Precision 24 MHz oscillator Connect a fundamental-mode crystal between XTALI and XTALO. An 80 Ω typical ESR crystal rated for a maximum drive level of 250 µW is acceptable. Alternately, a 50 Ω typical ESR crystal rated for a maximum drive level of 200 µW may be used. See the engineering bulletin EB830 for additional options.	Freescale BSP software requires 24 MHz on this clock. This clock is used as a reference for USB, so there are strict frequency tolerance and jitter requirements. See Table 1-14 for guidelines. See the crystal oscillator (XTALOSC) reference manual chapter and relevant interface specification chapters for details. To access a calculator for the 24 MHz crystal drive level, see EB830 on the i.MX Community.
	5. External megahertz source If feeding an external clock into the device, XTALI can be driven DC-coupled with XTALO floated.	For XTALI VIL and VIH voltage levels, see the latest i.MX 6 series datasheet. This clock is used as a reference for USB, PCIe, and SATA, so there are strict frequency tolerance and jitter requirements. See Table 1-14 for guidelines. See the crystal oscillator (XTALOSC) reference manual chapter and relevant interface specification chapters for details.

Table 1-7. Oscillator and clock recommendations (continued)

Checkbox	Recommendation	Explanation/supplemental recommendation
	6. CLK1_P/CLK1_N is LVDS input/output differential pair compatible with TIA/EIA-644 standard. The frequency range is 0 to 600 MHz. Alternatively, a single-ended signal can be used to drive a CLK1_P input. In this case, the corresponding CLK1_N input should be tied to a constant voltage level equal to 50% of VDD_HIGH_CAP. Termination should be provided with high-frequency signals. See the LVDS pad electrical specification in the data sheet for further details. After initialization, the CLK1 inputs/outputs can be disabled (if not used) by the PMU_MISC1 register. If unused, any or both of the CLK1_N/P pairs may be left floating.	The clock inputs/outputs are general-purpose differential high-speed clock Input/outputs. Any or both of them can be configured: • As inputs to feed external reference clocks to the on-chip PLLs and/or modules, for example as alternate reference clock for PCIe or/and SATA or video/audio interfaces. • As outputs to be used as either a reference clock or as a functional clock for peripherals, for example an output of the PCIe master clock (root complex use). See the chip reference manual for details on the respective clock trees.
	7. Bias XTALI with a 2.2 MΩ resistor to GND. Mount the resistor close to the XTALI ball.	The XTALI bias must be adjusted externally to ensure reasonable start-up time. Without the resistor, start-up time may be 200 ms or more.

Table 1-8. Reset and ONOFF recommendations

Checkbox	Recommendation	Explanation/supplemental recommendation
	1. The POR_B input must be asserted at power-up and remain asserted until after the last power rail for devices required for system boot are at their working voltage.	A reset switch may be wired to the chip's POR_B, which is a cold-reset negative-logic input that resets all modules and logic in the IC. POR_B may be used in addition to internally generated power-on reset signal (logical AND, both internal and external signals are considered active low).
	2. For portable applications, the ONOFF input may be connected to an ON/OFF SPST push-button switch. On-chip debouncing is provided, and this input has an on-chip pullup. If not used, ONOFF should be a no connect.	A brief connection to GND in OFF mode causes the internal power management state machine to change state to ON. In ON mode, a brief connection to GND generates an interrupt (intended to be a software-controllable power-down). An approximate 5 second or more connection to GND causes a forced OFF.

Table 1-9. USB recommendations

Checkbox	Recommendation	Explanation/supplemental recommendation
	1. USB OTG To comply with the USB OTG specification, the VBUS supply on the OTG connector should default to *off* when the boards power up.	The processor should turn VBUS *on* as required.
	2. USB Host USB_H1_VBUS should be directly connected to a 5 V supply.	Tie USB_H1_VBUS to an unswitched 5 V supply for the typical use case. However, if the your system is a USB device, then USB_H1_VBUS may be a no connect.

Table 1-10. Miscellaneous recommendations

Checkbox	Recommendation	Explanation/supplemental recommendation
	1. The TEST_MODE input is internally connected to an on-chip pulldown device. The user can either float this signal or tie it to GND.	This input is reserved for Freescale manufacturing use.
	2. For termination of unused analog interfaces, see Table 1-15.	—
	3. GPANAIO must be a no connect.	This output is reserved for Freescale manufacturing use.
	4. NC contacts are no connect and should be floated.	Depending on the feature set, some versions of the IC may have NC contacts connected inside the BGA.

1.3 Bus isolation circuit

The following figure provides supporting information for Table 1-2, recommendation #1

Boot Strap Power Control

Bus isolation

Figure 1-1. Boot control for development mode

1.4 DDR reference circuit

The following table is a resistor chart (see Table 1-1 recommendation #2). The recommendations are appropriate for designs with DDR memory chips with a maximum Vref input current of 2µA each.

Table 1-11. DDR Vref resistor sizing guideline

Number of DRAM with 2 µA Vref input current	Resistor divider value (2 resistors)
2	≤1.21 kΩ 1%
2	≤1.54 kΩ 0.5%
2	≤2.32 kΩ 0.1%
4	≤768 Ω 1%
4	≤1 kΩ 0.5%
4	≤1.5 kΩ 0.1%

1.5 I^2C clock speed and division factors (IFDR)

The I^2C clock is sourced from PERCLK_CLK_ROOT which is routed from IPG_CLK_ROOT. The I^2C clock frequency can be easily obtained using the following formula:

I^2C clock Frequency = (PERCLK_ROOT frequency)/(division factor corresponding to IFDR)

By default, the IPG_CLK_ROOT and PERCLK_CLK_ROOT frequencies are set to 49.5MHz, where the root clock is sourced from PLL2's PFD2. Obtaining the frequencies can be accomplished using the following:

PLL2 = 528MHz

PLL2_PFD2 = 528MHz * 18 / 24 = 396MHz

IPG_CLK_ROOT = (PLL2_PFD2 / ahb_podf)/ ipg_podf = (396MHz/4)/2 = 49.5MHz

PER_CLK_ROOT = IPG_CLK_ROOT/perclk_podf = 49.5MHz/1 = 49.5MHz

NOTE

The above calculation assumes that the default CCM register settings, routing, and division factors are used. If different routing, PFD values, and/or division factors are used, the user must adjust the parameters accordingly to calculate the correct clock frequency.

IFDR, division factor and resulting I^2C CLK frequencies are indicated in the table below. Resulting frequencies will vary according to the PERCLK_CLK_ROOT frequencies selected.

Table 1-12 assumes PERCLK_CLK_ROOT = 49.5MHz.

Table 1-12. IFDR[1]

IFDR	Division factor	Frequency (kHz)
0	30	1650
1	32	1546.875
2	36	1375
3	42	1178.571
4	48	1031.25
5	52	951.9231
6	60	825
7	72	687.5
8	80	618.75
9	88	562.5
A	104	475.9615
B	128	386.7188
C	144	343.75
D	160	309.375
E	192	257.8125
F	240	206.25
10	288	171.875
11	320	154.6875
12	384	128.9063
13	480	103.125
14	576	85.9375
15	640	77.34375
16	768	64.45313
17	960	51.5625
18	1152	42.96875
19	1280	38.67188
1A	1536	32.22656
1B	1920	25.78125
1C	2304	21.48438
1D	2560	19.33594
1E	3072	16.11328
1F	3840	12.89063

Table 1-12. IFDR[1] (continued)

20	22	2250
21	24	2062.5
22	26	1903.846
23	28	1767.857
24	32	1546.875
25	36	1375
26	40	1237.5
27	44	1125
28	48	1031.25
29	56	883.9286
2A	64	773.4375
2B	72	687.5
2C	80	618.75
2D	96	515.625
2E	112	441.9643
2F	128	386.7188
30	160	309.375
31	192	257.8125
32	224	220.9821
33	256	193.3594
34	320	154.6875
35	384	128.9063
36	448	110.4911
37	512	96.67969
38	640	77.34375
39	768	64.45313
3A	896	55.24554
3B	1024	48.33984
3C	1280	38.67188
3D	1536	32.22656
3E	1792	27.62277
3F	2048	24.16992

[1] Shaded cells indicate frequency is outside of the range that guarantees operation.

1.6 JTAG signal termination

The following table is a JTAG termination chart (see recommendations in Table 1-5).

Table 1-13. JTAG interface summary

JTAG signal	I/O type	On-chip termination	External termination
JTAG_TCK	Input	47 kΩ pullup	Not required; can use 10 kΩ pullup
JTAG_TMS	Input	47 kΩ pullup	Not required; can use 10 kΩ pullup
JTAG_TDI	Input	47 kΩ pullup	Not required; can use 10 kΩ pullup
JTAG_TDO	3-state output	Keeper	Do not use pullup or pulldown
JTAG_TRSTB	Input	47 kΩ pullup	Not required; can use 10 kΩ pullup
JTAG_MOD	Input	100 kΩ pullup	Use 1 kΩ pulldown or tie to GND

1.7 Oscillator tolerance

The following table provides 24 MHz oscillator tolerance guidelines (see Table 1-7, recommendations #4 and #5). Because these are guidelines, the designer must verify all tolerances per the official specifications.

Table 1-14. 24 MHz crystal tolerance guidelines

Interface	Tolerance (± ppm)
Ethernet	50
USB2.0	500

1.8 Unused analog interfaces

Table 1-15 shows the recommended connections for unused analog interfaces (see Table 1-10, recommendation #2).

Table 1-15. Recommended connections for unused analog interfaces

Module	Contact name	Recommendations if unused
CCM	CLK1_N, CLK1_P	Float
USB	USB_H1_DN, USB_H1_DP, USB_H1_VBUS, USB_OTG_CHD_B, USB_OTG_DN, USB_OTG_DP, USB_OTG_VBUS	Float

Chapter 2
i.MX 6 Series Layout Recommendations

This chapter provides recommendations to assist design engineers with the correct layout of their i.MX 6 series-based system. The majority of the chapter discusses the implementation of the DDR interface, but it also provides recommendation for power, the USB, reference resistors, ESD and related emissions.

This chapter uses the IMX6SLEVK board as its reference for illustrating the key concepts. See the IMX6SLEVK board layout files as a companion to this chapter.

2.1 Basic design recommendations

The i.MX 6SoloLite processor comes in a 13 × 13 mm package with 0.5 mm ball pitch. The ball-grid array contains 24 rows and 24 columns, making it a 576 ball BGA package. For detailed information about the package, see the i.MX 6SL series Consumer datasheets.

The following figure shows the ball-grid array. Figure 2-2 shows additional package information.

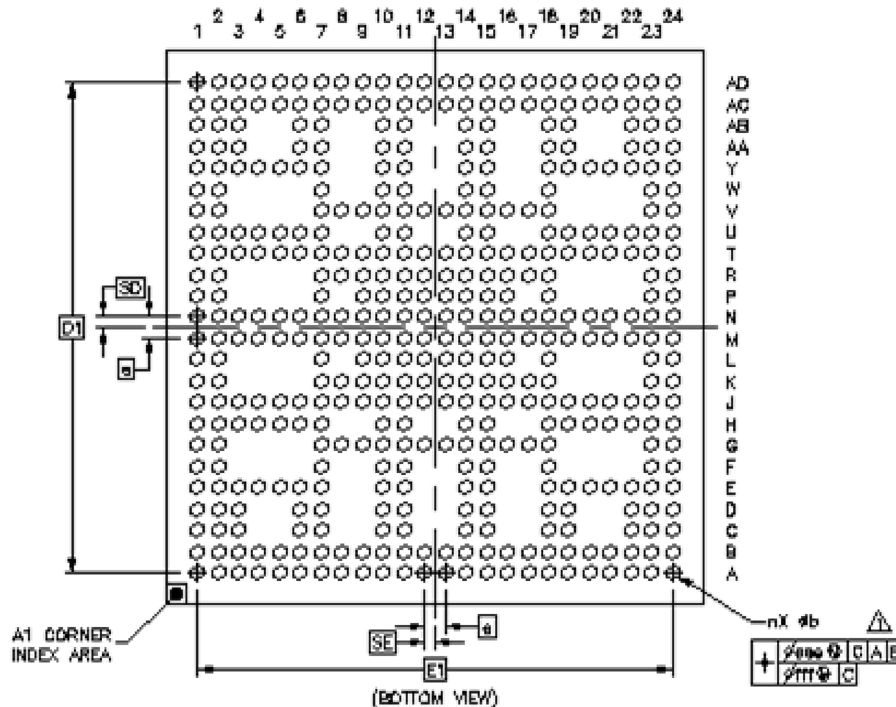


Figure 2-1. i.MX 6SL ball-grid array

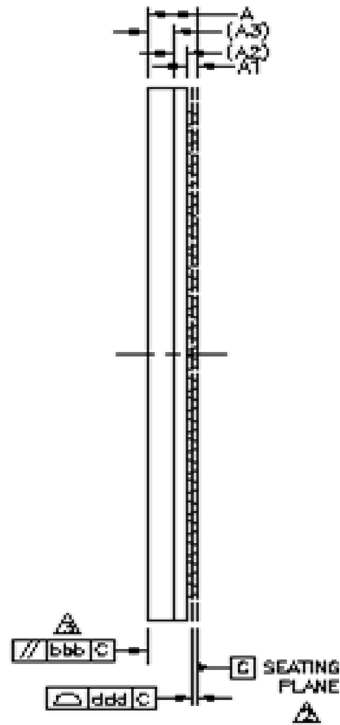


Figure 2-2. i.MX 6SoloLite package information

It is critical to maintain the recommended footprint of a 10-mil pad with a 14-mil open solder mask for ease of fanout. In this case, the solder paste is the same as the pad with 10 mils.

When using the Allegro tool, optimal practice is to use the footprint as created by Freescale. When not using the Allegro tool, use the Allegro footprint export feature (supported by many tools). If export is not possible, create the footprint as per the package mechanical dimensions outlined in the product data sheet.

2.1.1 Fanout illustrations

The following figures show the top and bottom layer fanouts for the i.MX SL chip.

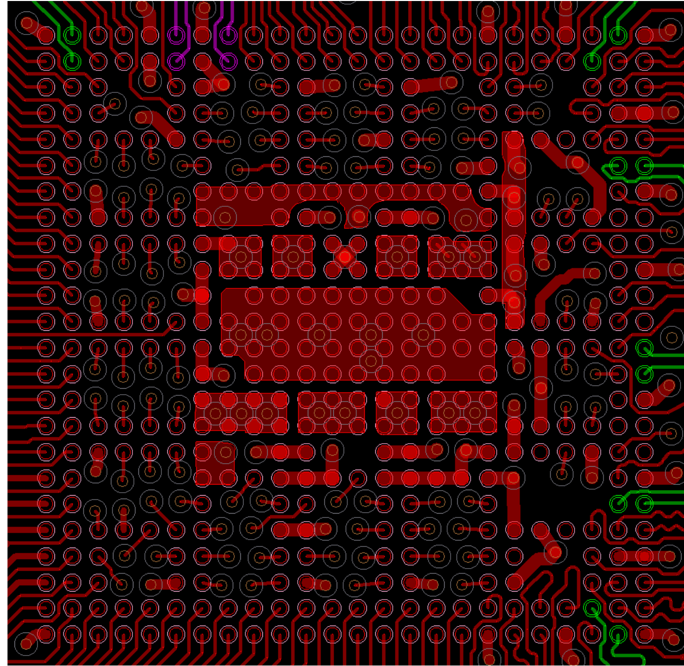


Figure 2-3. i.MX6SL fanout example, top layer view

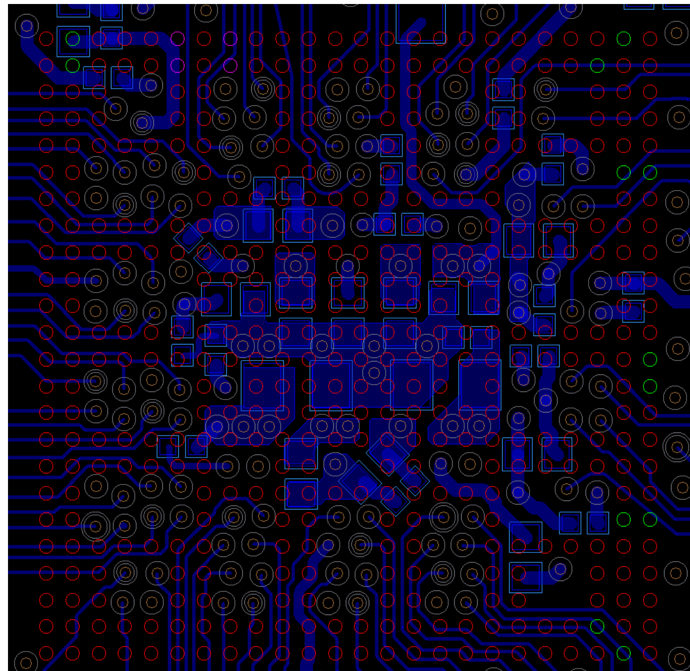


Figure 2-4. i.MX6SL fanout example, bottom view

The colors signify the following:

- Top layer
  - Red = etch
  - Yellow = pad
  - Gray = vias
- Bottom layer
  - Blue = etch

2.1.2 Placing decoupling capacitors

The fanout scheme creates a four quadrant structure that facilitates the placement of decoupling bulk capacitors on the bottom side of the PCB.

The 0201 decoupling and 0603 bulk capacitors should be mounted as close as possible to the power vias. The distance should be less than 50 mils. Additional bulk capacitors can be placed near the edge of the BGA via array. Placing the decoupling capacitors close to the power balls is critical to minimize inductance and ensure high-speed transient current demand by the processor.

A correct via size is critical for preserving adequate routing space. The recommended geometry for the via pads is: pad size 18 mils and drill 8 mils.

The following list provides the main recommendations for choosing the correct decoupling scheme for the i.MX6 family boards.

- Place the largest capacitance in the smallest package that budget and manufacturing can support.
- For high speed bypassing, select the required capacitance with the smallest package (for example, 0.22 µF and package 0201).
- Minimize trace length (inductance) to small caps.
- Series inductance cancels out capacitance.
- Tie caps to GND plane directly with a via.
- Place capacitors close to the power contact of the associate package designed from the schematic.

The MCIMX6SLEVK CPU uses the preferred BGA power decoupling design. Note that the layout is available through www.freescale.com. Customers should use the reference design strategy for power and decoupling.

2.2 Stackup recommendations

High-speed design requires a good stackup in order have the right impedance for the critical traces. The constraints for the trace width may depend on a number of factors, such as the board stackup and associated dielectric and copper thickness, required impedance, and required current (for power traces). The Freescale reference design uses a minimum trace width of 3 mils for the DDR routing. The stackup also determines the constraints for routing and spacing.

Consider the following when designing the stackup and selecting the material for your board.

- Board stack-up is critical for high-speed signal quality.
- You must preplan impedance of critical traces
- High-speed signals must have reference planes on adjacent layers to minimize cross-talk.
- FSL reference design equals Isola 370HR.
- FSL validation boards equals Isola FR408.

The recommended stackup is 6-layers, with the layer stack as shown in the following figure. The lefthand image shows the detail provided by Freescale inside the fabrication detail as a part of the Gerber files. The righthand side shows the solution suggested by the PCB fabrication company for our requirements. Additional power planes to support i.MX 6Dual/6Quad and i.MX 6Solo power options only.

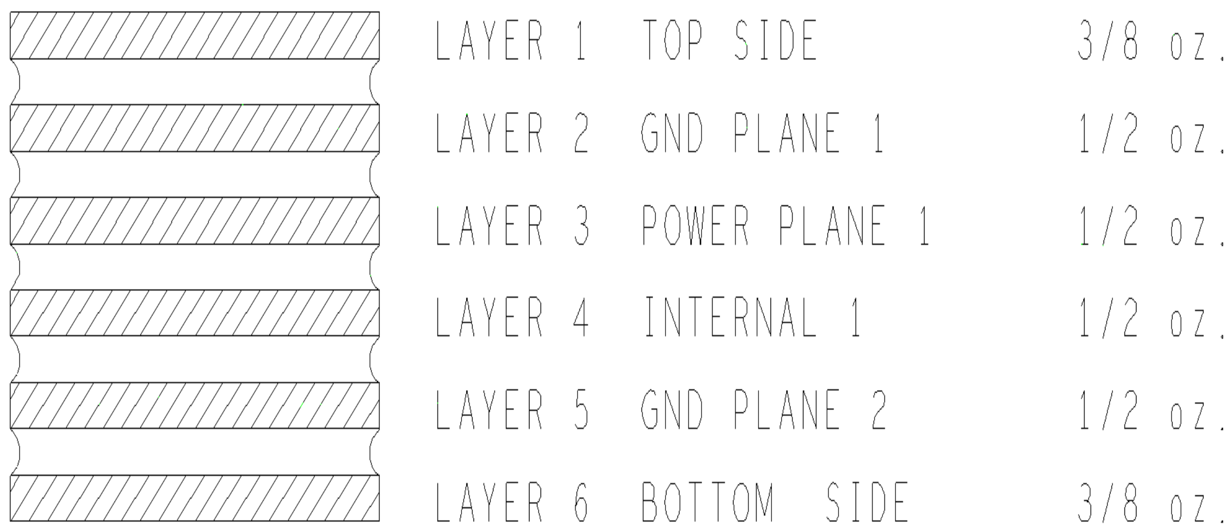


Figure 2-5. Layer stack EVK board

The following table shows a working stack-up implementation:

Table 2-1. Stackup implementation

Layers	Single ended		Differential					
	Trace width (Mils)	Impedance (Ωs)	Trace width (Mils)	Trace spacing 'Airgap' (Mils)	Impedance (Ωs)	Trace width (Mils)	Trace spacing 'Airgap' (Mils)	Impedance (Ωs)
TOP	4.75	50	4.2	4.8	90	3.5	5.5	100
INT1	5.5	50						
BOT	4.75	50	4.2	4.8	90	3.5	5.5	100

2.3 DDR connection information

The following figures show the block diagrams from the reference design boards for the DDR3 interface and the LPDDR2 interface (respectively) with the i.MX6SL.

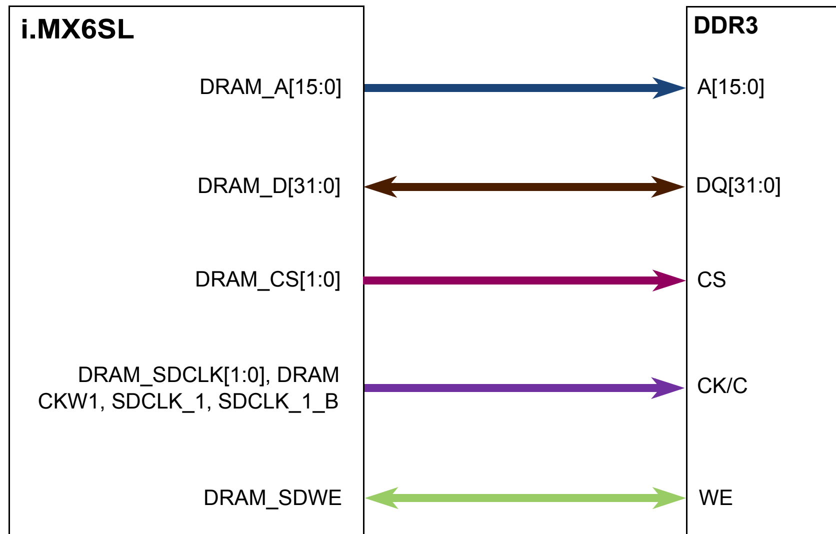


Figure 2-6. Connection between i.MX 6SoloLite and DDR3

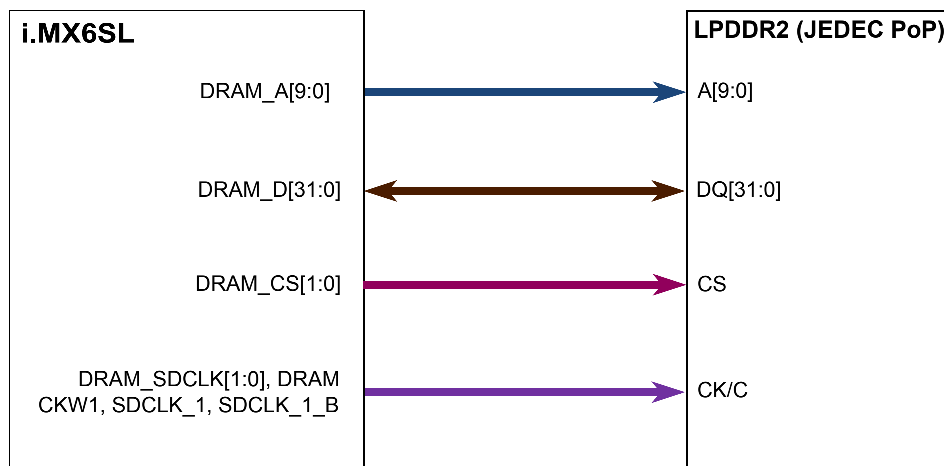


Figure 2-7. i.MX 6SoloLite and LPDDR2

The LPDDR2 interface is one of the most critical interfaces for chip routing. It must have the controlled impedance for the single ended traces be equal to 50 Ω and for the differential pairs be equal to 100 Ω.

The following figure shows the physical connection scheme for both top and bottom placement of the DDR chips, showing the final placement of the LPDDR2 memory and the decoupling capacitors. The red elements show the top layer and the blue elements show the bottom layer. It is very important to place the

memory as close to the processor as possible to reduce trace capacitance and keep the propagation delay to the minimum. Follow the reference board layout as a guideline for memory placement and routing.

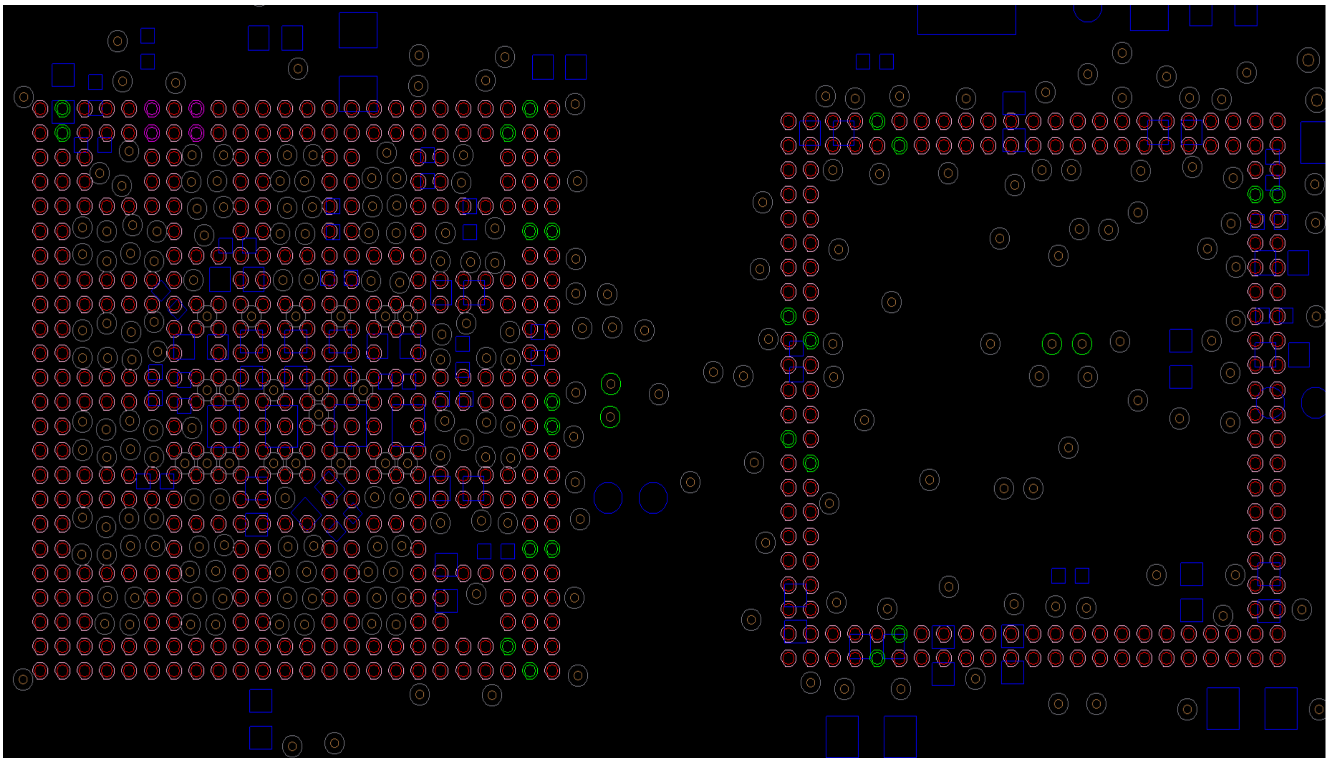


Figure 2-8. Final placement of memories and decoupling capacitors

2.4 DDR routing rules

DDR3/LPDDR2 routing can be accomplished in two different ways: routing all signals at the same length or routing by byte group.

Ideally, we could route all the signals at the same length, However, it could be difficult because of the large number of connections in the tight space between the DDR and the processor. The following table explains the rules for routing the signals by the same length.

Table 2-2. DDR3/LPDDR2 routing by the same length

Signals	Total length	Recommendations
Address and Bank	Clock length	Match the signals ±25 mils of the value specified in the length column
Data	Clock length	
Control signals	Clock length	

Table 2-2. DDR3/LPDDR2 routing by the same length (continued)

Signals	Total length	Recommendations
Clock DRAM_SDCLK[1:0]	Longest trace ≤ 3 inches	Match the signals of clocks signals ±5 mils. Each differential clock pair
DRAM_SDQS[3:0] and DRAM_SDQS[3:0]_B	Clock length	Match the signals of DQS signals ±10 mils of the value specified in the length column.

Routing by byte group requires better control of the signals of each group. It is also more difficult for analysis and constraint settings. However, its advantage is that the constraint to match lengths can be applied to a smaller group of signals. This is often more achievable once the constraints are properly set. The following table explains the rules for routing the signals by byte group.

Table 2-3. DDR3/LPDDR2 routing by byte group

Chip signals	Group	Length		Recommendations
		Min	Max	
DRAM_SDCLK[1:0] DRAM_SDCLK_B[1:0]	Clock	Short as possible	2.25 inches	Match the signals ± 5 mils. 2.25 inches is recommended.
DRAM_A[15:0] DRAM_SDBA[2:0] DRAM_RAS DRAM_CAS DRAM_SDWE	Address and Command	Clock (min) – 200	Clock (min)[1]	Match the signals ± 25 mils.
DRAM_D[7:0] DRAM_DQM0 DRAM_SDQS0 DRAM_SDQS0_B	Byte Group 1	—	Clock (min)	Match the signals of each byte group ± 25 mils. Match the differential signals of DQS ± 10 mils.
DRAM_D[15:8] DRAM_DQM1 DRAM_SDQS1 DRAM_SDQS1_B	Byte Group 2	—	Clock (min)	
DRAM_D[23:16] DRAM_DQM2 DRAM_SDQS2 DRAM_SDQS2_B	Byte Group 3	—	Clock (min)	
DRAM_D[31:24] DRAM_DQM3 DRAM_SDQS[3:0] DRAM_SDQS[3:0]_B	Byte Group 4	—	Clock (min)	
DRAM_CS[1:0] DRAM_SDCKE[1:0] DRAM_SDODT[1:0]	Control signals	Clock (min) – 200	Clock (min)	Match the signals ± 50 mils.

1. Clock (min)—The shortest length of the clock group signals because this group has a ± 5 mil matching tolerance.

Finally, the impedance for the signals should be 50 Ω for single ended and 100 Ω for differential pairs.

2.5 Routing considerations

The chip can handle up to 2 Gbytes of DRAM memory. i.MX6SL DDR routing needs to be separated into three groups: data, address, and control. Each group has its own method of routing from an i.MX 6SL chip to DDR memory. The DDR layout has 2 Gbytes.

2.5.1 Swapping data lines

The DDR3 pin swapping technique for the data bus lines within bytes makes it easier to:

- Route direct lines
- Avoid changes between layers

The rules are as follows:

- Hardware write leveling – lowest order bit within byte lane must remain on lowest order bit of lane by JEDEC compliance (see the “Write Leveling” section in JESD79-3E)
  — D0, D8, D16, D24, D32, D40, D48, and D56 are fixed
  — Other data lines free to swap within byte lane
- JEDEC DDR3 memory restrictions are:
  - No restrictions for complete byte lane swapping
  - DQS and DQM must follow lanes

NOTE

If byte lane swapping was done, target DDR IC register read value must be transposed according to the data line swapping.

2.5.2 DDR3 (32 bits) T topology considerations

Be sure to take into account the following when designing a T-topology system.

- Follow the routing rules described in Table 2-3.
- Termination resistors not required.
- Short routing lengths and on-chip drive strength control.
- Your design is limited to 4 DDR chips.
- DDR3, 2 GBytes using latest memories (4 GBytes coming).

2.5.3 DDR3 (32 bits) Fly-by topology considerations

Pay attention to the following recommendations when the Fly-by topology and routing technique.

- DDR controller provides address mirroring when using two chip selects, which aids address line routing for memories on both sides of board.
- Bus termination resistors are required.

2.5.4 2-Gigabyte recommendations

The 512 Mbyte option has four memories. You should follow these recommendations for best practice:

- Have a balanced routing for the T connection.
- Avoid having many layer transitions.
- Do not cross split reference planes during the routing.

The following figure shows the topology for the ADDR/CMD/CTRL signals. It has a tree topology. Note the balanced T routing.

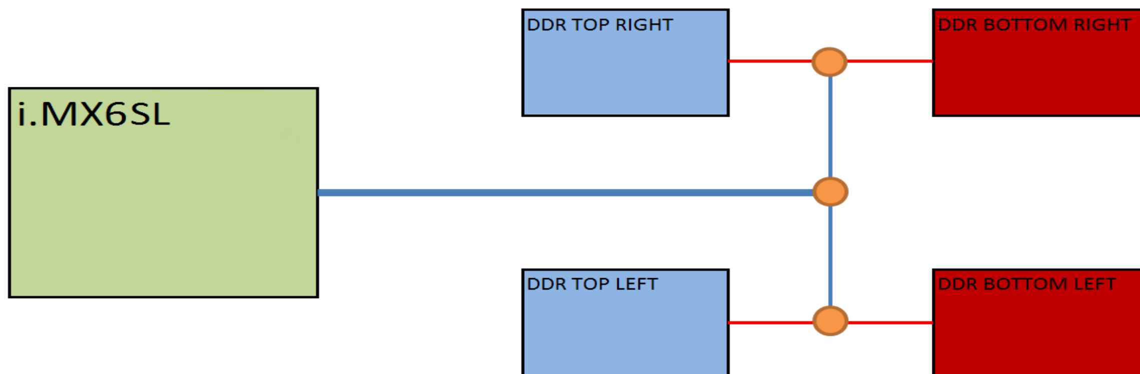


Figure 2-9. ADDR/CMD/CTRL signal topology

The routing for the data groups depends on the bus size. The following figure shows the point-to-point data bus connection, with routing by byte group.

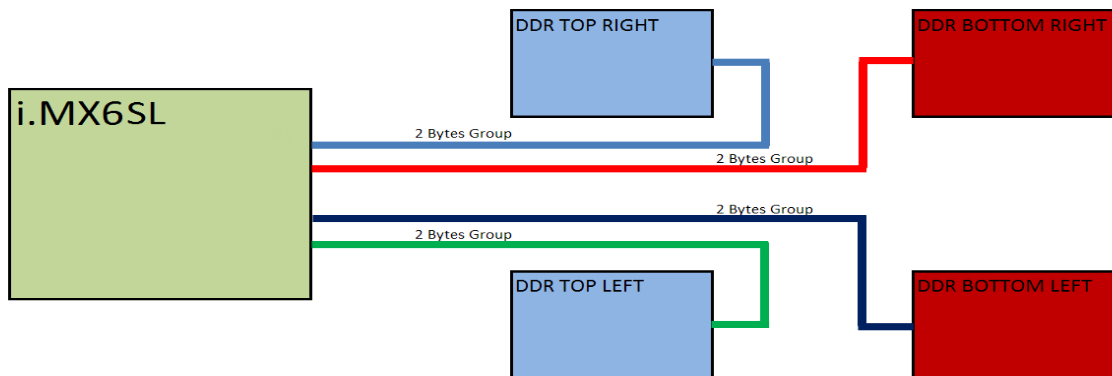


Figure 2-10. Point-to-point data bus connection (routing by byte group)

NOTE

i.MX 6Solo only uses the first two pairs of the 2 Bytes groups. All others are disabled.

2.5.5 1-Gigabyte recommendations

The following diagrams show the 1 Gbyte recommendations using both chip selects (CS[1:0]) and loading 512 Mbytes to each one. This option has eight memories and requires the addition of a termination resistor.

Route the ADDR/CMD signals as shown in the following figure.

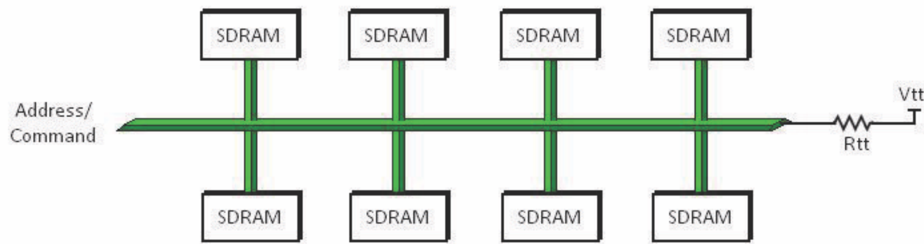


Figure 2-11. ADDR/CMD signal topology

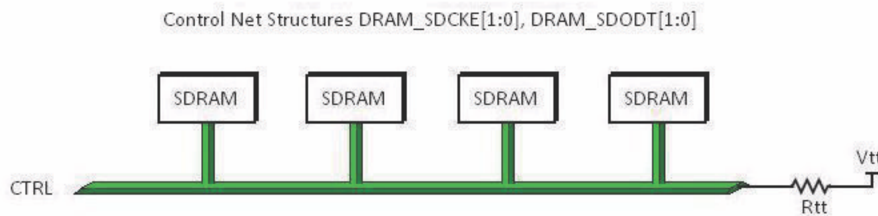


Figure 2-12. CTRL signal topology

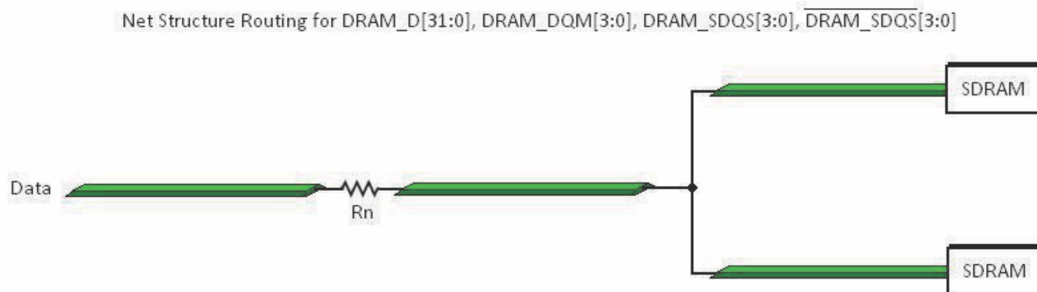


Figure 2-13. Data bus routing topology

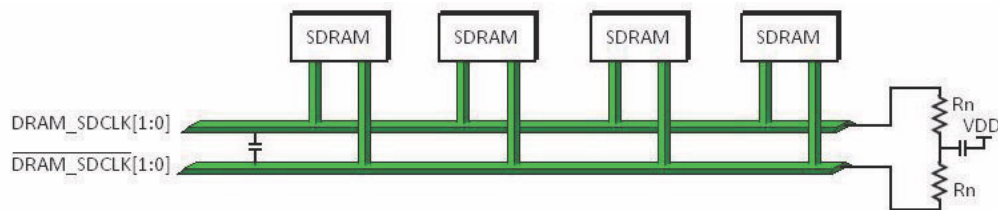


Figure 2-14. Clock routing topology

2.5.6 Four chips T topology routing examples

The figures in this section show examples for the routing of the 2 GByte DDR3 memories. Figure 2-15 through Figure 2-20 are guidelines of the T configuration routing with 12 layers PCB. Table 2-4 shows the color coding used in the figures.

Table 2-4. Color code

Color	Meaning
Soft Green	ADD & CMD Signals
Yellow	Clocks
Soft Pink	Data Byte Group 0
Purple	Data Byte Group 1
Blue	Data Byte Group 2
Brown	Data Byte Group 3
Gray	DDR_1V5 & DDR_VREF
Soft Red	Control Signals

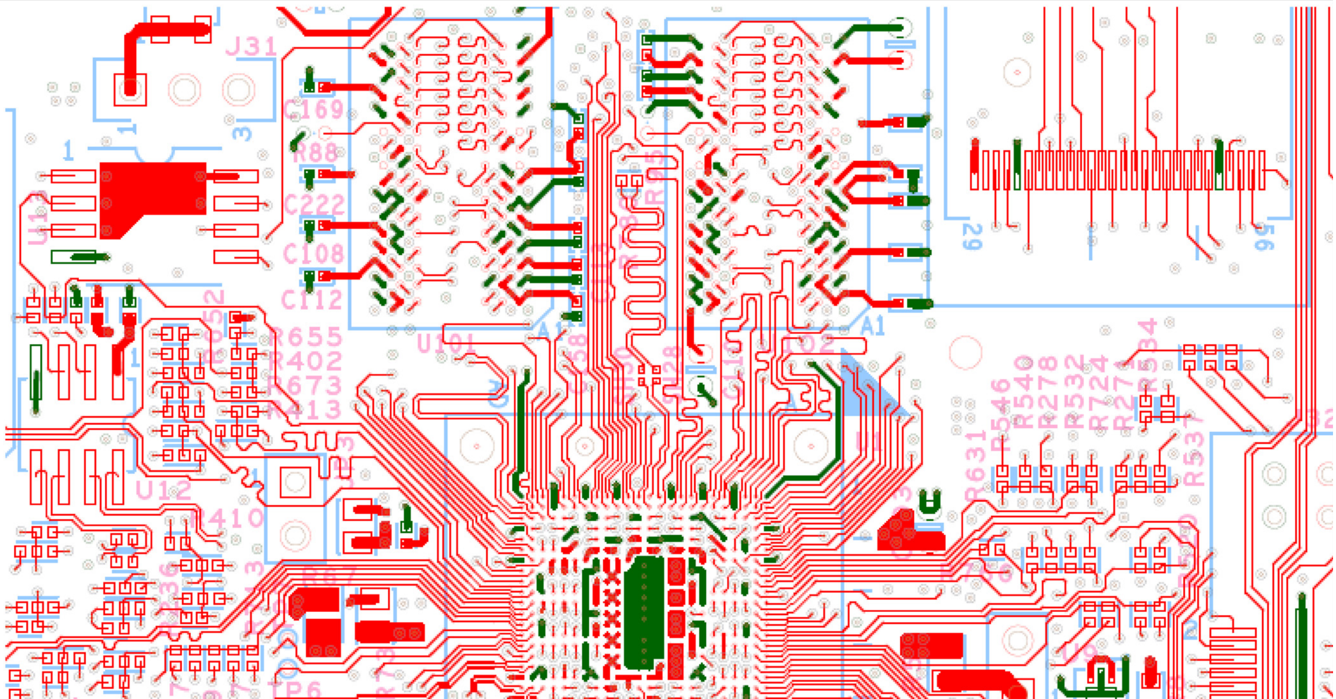


Figure 2-15. Top layer DDR3 routing

Figure 2-16. Internal L3 DDR3 routing

Figure 2-17. Internal L4 DDR3 routing

Figure 2-18. Internal L9 DDR3 routing

Figure 2-19. Internal L10 DDR3 routing

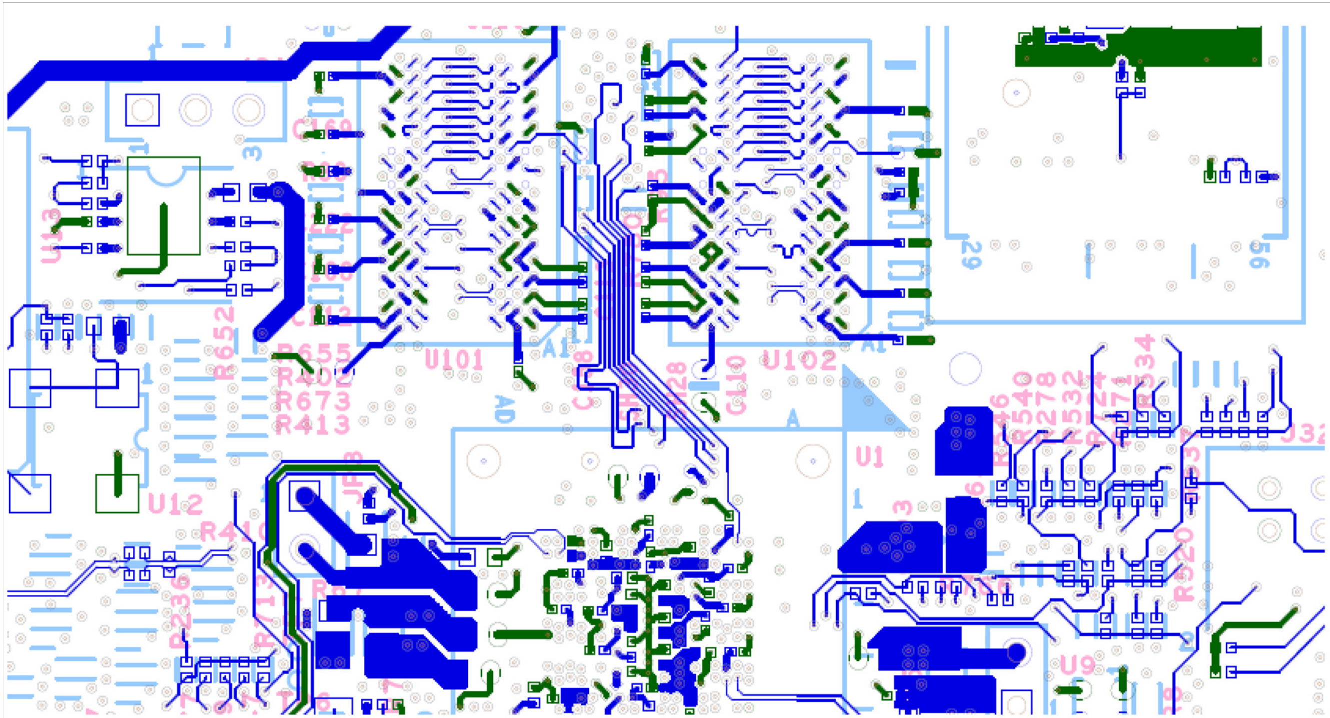


Figure 2-20. Bottom layer DDR3 routing

The following table shows the total etch of the signals for the byte 0 and byte 1 groups. The layout is an example, using 2000 mils for the clock.

Table 2-5. Total signal etch (DDR3)

Signals	Length (Mils)
DRAM_D0	990.27
DRAM_D1	983.71
DRAM_D2	984.54
DRAM_D3	985.65
DRAM_D4	989.39
DRAM_D5	983.84
DRAM_D6	981.65
DRAM_D7	980.52
DRAM_DQM0	1028.62
DRAM_SDQS0	1042.41
DRAM_SDQS0_B	1045.87
DRAM_D8	971.98

Table 2-5. Total signal etch (DDR3) (continued)

Signals	Length (Mils)
DRAM_D9	963.04
DRAM_D10	963.54
DRAM_D11	962.21
DRAM_D12	963.61
DRAM_D13	947.47
DRAM_D14	962.43
DRAM_D15	963.43
DRAM_DQM1	1034.17
DRAM_SDQS1	1026.07
DRAM_SDQS1_B	1010.68
DRAM_SDCLK0	1908.54
DRAM_SDCLK0_B	1903.28

2.5.7 LPDDR2 FBGA 168 routing example

The figures in this section show examples for the routing of 1-Gbyte LPDDR2 memory. These figures are a guideline of the routing by layer used on i.MX6SLEVK. They use the same color code shown in Table 2-4.

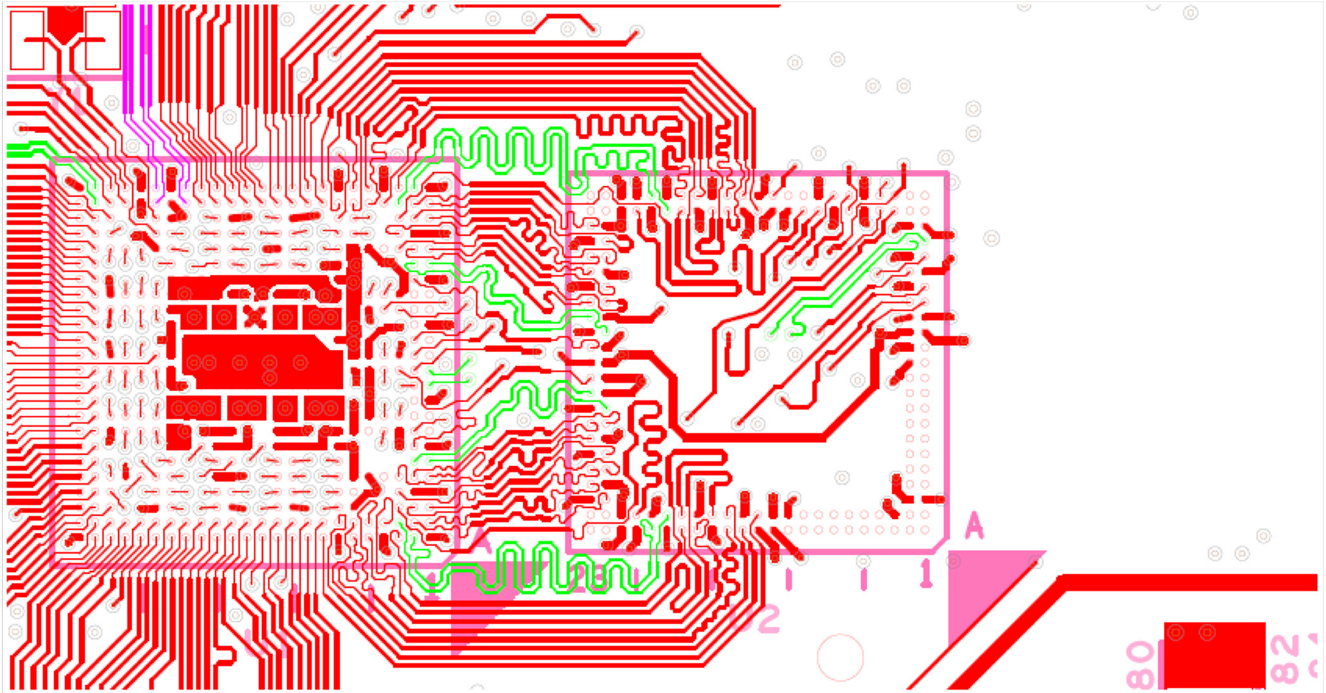


Figure 2-21. Top LPDDR2 routing

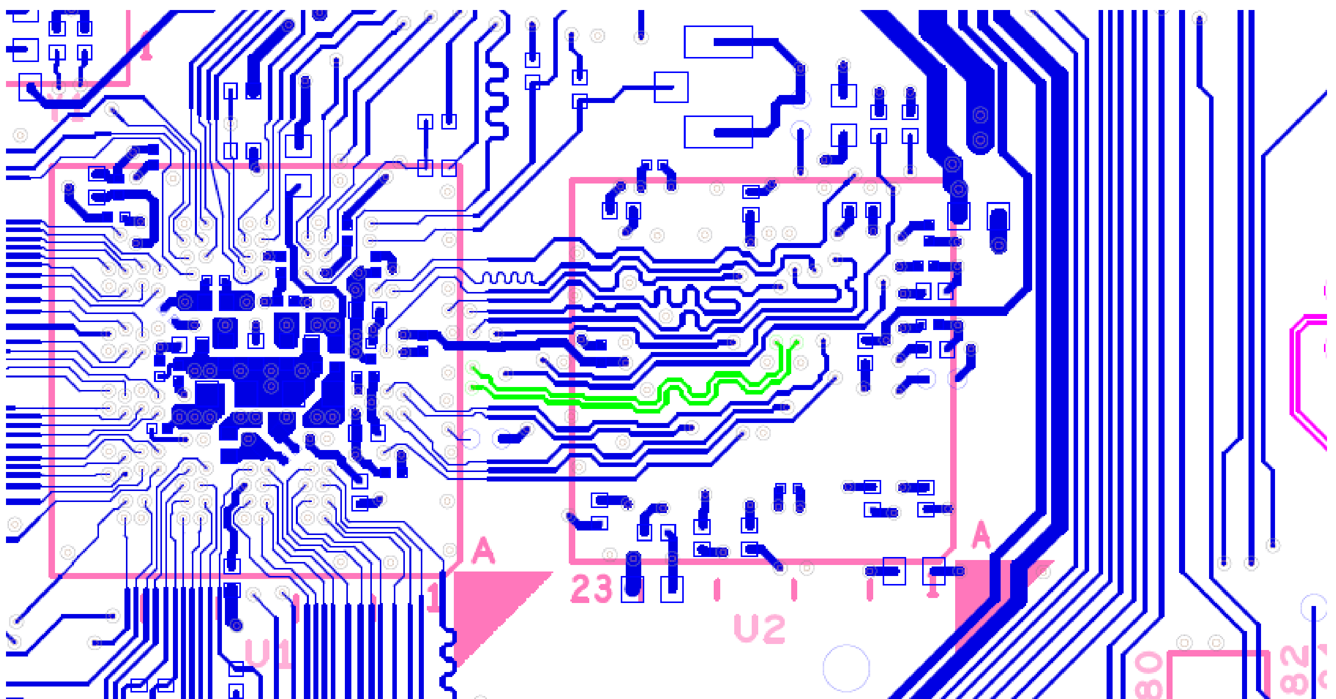


Figure 2-22. Bottom LPDDR2 routing

The following table shows the total etch of the signals for the byte 0 and byte 1 groups.

Table 2-6. Total signal etch (LPDDR2)

Signals	Length (Mils)
DRAM_D0	342.03
DRAM_D1	347.02
DRAM_D2	348.47
DRAM_D3	346.29
DRAM_D4	341.84
DRAM_D5	349.40
DRAM_D6	342.90
DRAM_D7	346.38
DRAM_DQM0	344.57
DRAM_SDQS0	349.30
DRAM_SDQS0_B	350.30
DRAM_D8	342.18
DRAM_D9	343.85
DRAM_D10	349.73
DRAM_D11	350.18
DRAM_D12	349.90
DRAM_D13	346.66
DRAM_D14	345.09
DRAM_D15	343.37
DRAM_DQM1	345.93
DRAM_SDQS1	350.18
DRAM_SDQS1_B	350.45
DRAM_SDCLK0	842.84
DRAM_SDCLK0_B	842.12

2.5.8 High speed signal routing recommendations

The following list provides recommendations for routing traces for high speed signals. Note that the propagation delay and the impedance control should match in order to have the correct communication with the devices.

- High-speed signals (DDR, FEC, display) must not cross gaps in the reference plane.
- Avoid creating slots, voids, and splits in reference planes. Review via voids to ensure they do not create splits (space out vias).
- A solid GND plane must be directly under crystal, associated components, and traces.
- Clocks or strobes that are on the same layer need at least 2.5× spacing from an adjacent trace (2.5× height from reference plane) to reduce cross-talk.
- All synchronous modules should have bus length matching and relative clock length control.
  - For SD module interfaces:
    - Match data and CMD trace lengths (length delta depends on bus rates)
    - CLK should be longer than the longest signal in the Data/CMD group (+5 mils)
  - Similar DDR rules must be followed for data, address and control as for SD module interfaces.

2.5.9 Ground plane recommendations

This section provides examples of good practices and how to avoid common user mistakes when flowing the ground planes layers.

The following two figures show common examples of poor GND planes. The copper plane is represented by the color gray in Figure 2-23 and by the horizontal green lines in Figure 2-24.

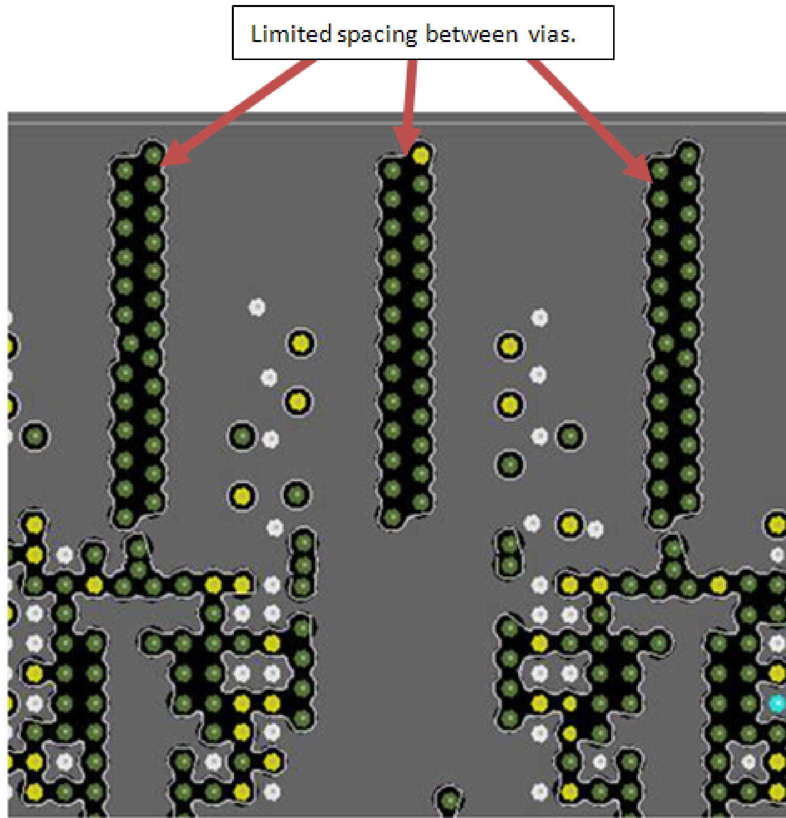


Figure 2-23. Poor GND plane 1

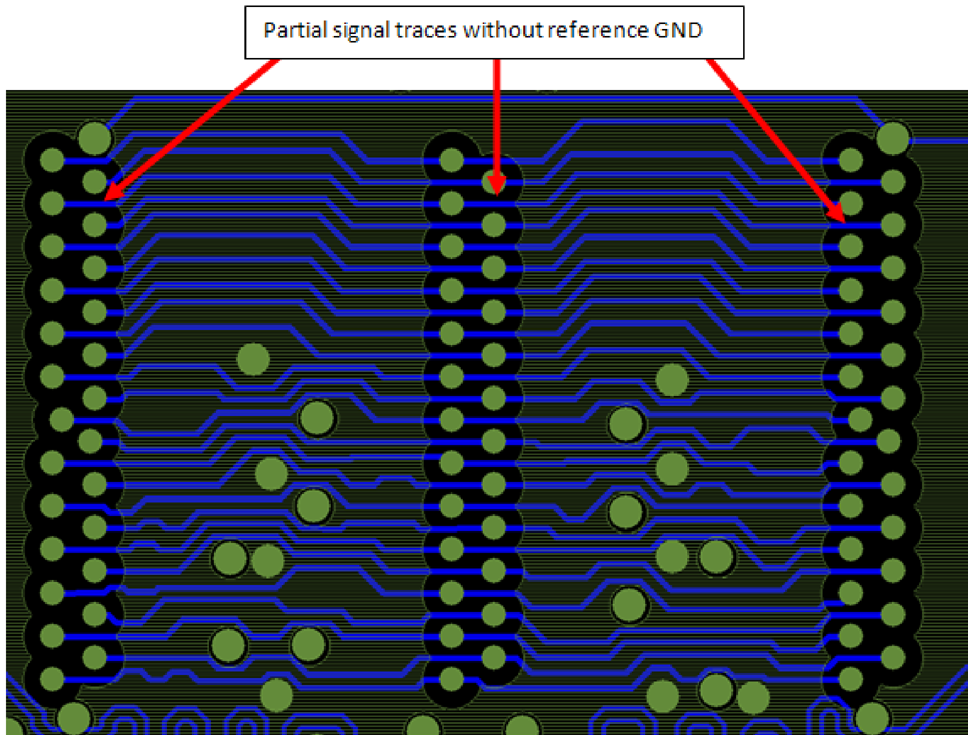


Figure 2-24. Poor GND plane 2

Spacing the vias some mils apart facilitates the GND copper flowing in the plane. The following figures show good practices of ground planes.

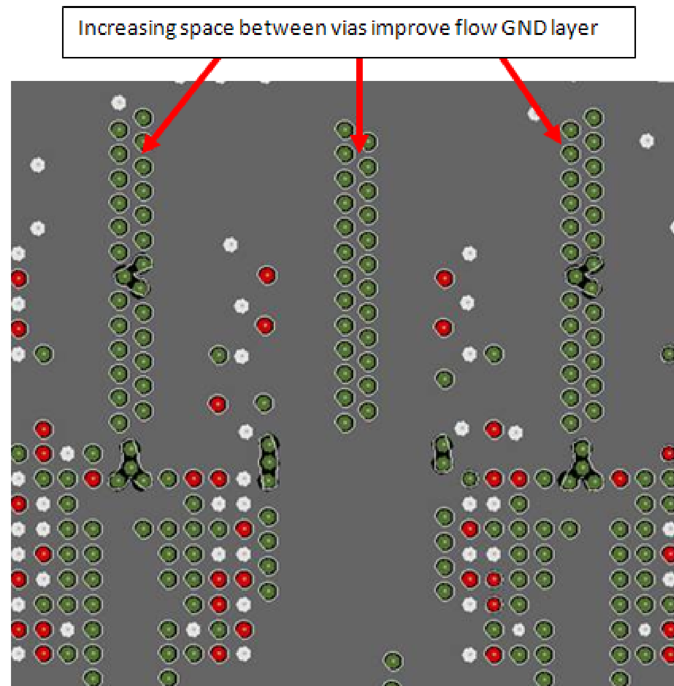


Figure 2-25. Good layout GND plane detail

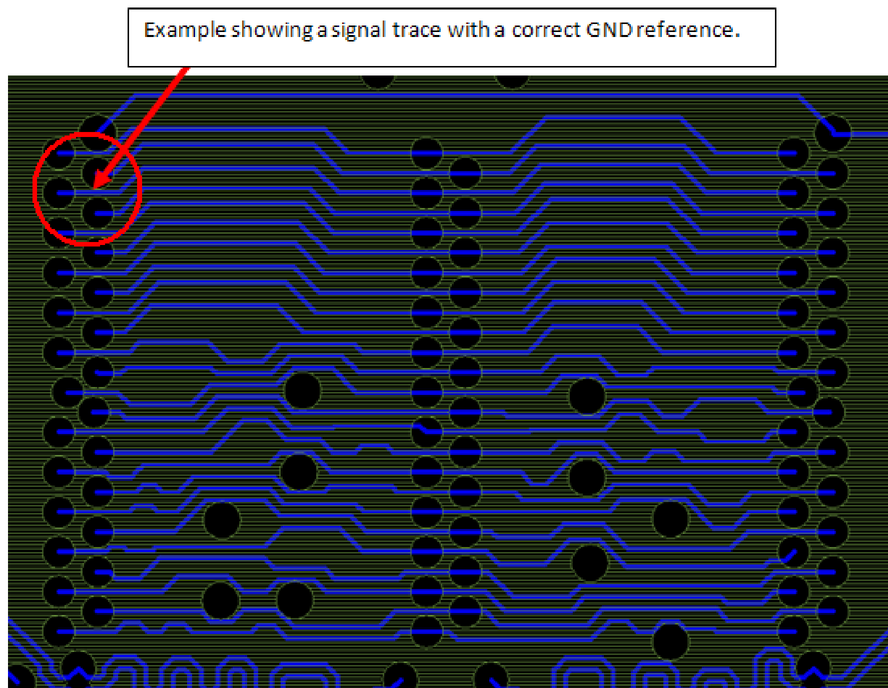


Figure 2-26. Good layout GND plane detail

2.6 DDR power recommendations (for DDR3 only)

The following recommendations apply to the VREF (P0V75_REFDDR) voltage reference plane.

- Use 30 mils trace between decoupling cap and destination.
- Maintain a 25 mils clearance from other nets.
- Isolate VREF and/or shield with ground.
- Decouple using distributed 0.22 µF capacitors by the regulator, controller, and devices.
- Place one 1.0 µF near the source of VREF: one near the VREF pin on the controller and two between the controller and the devices.

The following recommendations apply to the VTT (DDR_VTT) voltage reference plane. The figures are examples from the evaluation board for the VTT reference schematic.

- Place the VTT island on the component side layer at the end of the bus behind the DRAM devices.
- Use a wide-island trace for current capacity.
- Place the VTT generator as close to termination resistors as possible to minimize impedance (inductance).
- Place one or two 0.1 µF decoupling capacitors by each termination RPACK on the VTT island to minimize the noise on VTT. Other bulk (10–22 pF) decoupling is also recommended to be placed on the VTT island.

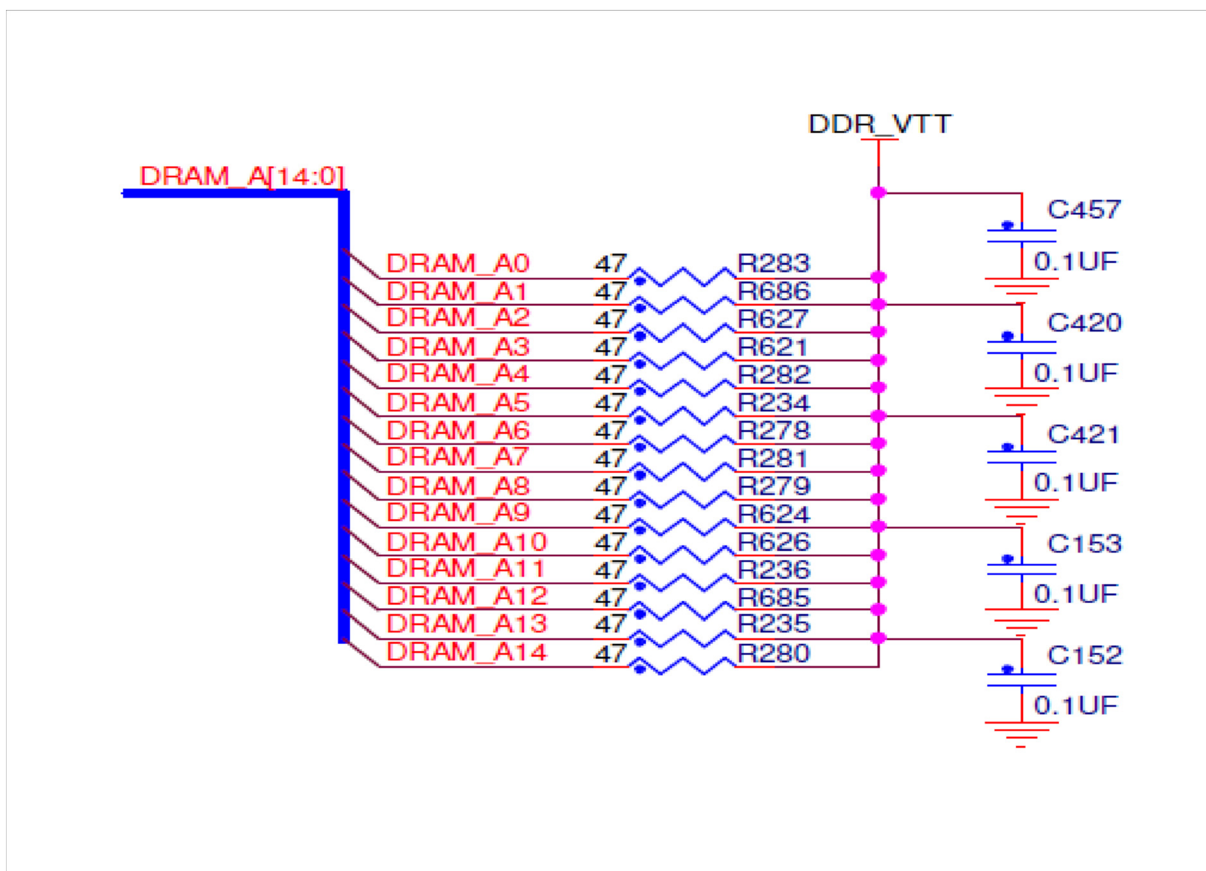


Figure 2-27. DDR_VTT evaluation board example

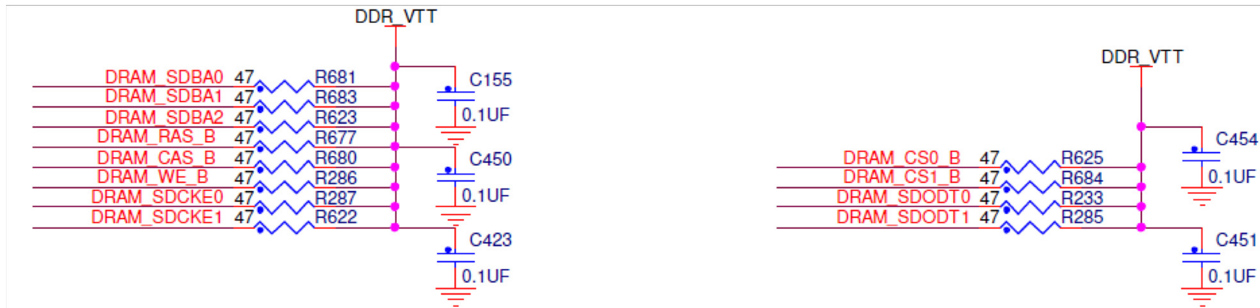


Figure 2-28. DDR_VTT evaluation board examples

2.7 USB recommendations

Use the following recommendations for the USB.

- Route the DP and DM differential pair first.
- Route DP and DM signals on the top or bottom layer of the board
- The trace width and spacing of the DP and DM signals should be such that the differential impedance is 90 Ω.
- Route traces over continuous planes (power and ground).
  — They should not pass over any power/GND plane slots or anti-etch.
  — When placing connectors, make sure the ground plane clearouts around each pin have ground continuity between all pins.
- Maintain the parallelism (skew matched) between DP and DM; these traces should be the same overall length.
- Do not route DP and DM traces under oscillators or parallel to clock traces and/or data buses.
- Minimize the lengths of high speed signals that run parallel to the DP and DM pair.
- Keep DP and DM traces as short as possible.
- Route DP and DM signals with a minimum amount of corners. Use 45-degree turns instead of 90-degree turns.
- Avoid layer changes (vias) on DP and DM signals. Do not create stubs or branches.

2.8 Impedance signal recommendations

Use the following table as a reference when you are updating or creating constraints in your software PCB tool to set up the impedance and the correct trace width.

Table 2-7. Impedance signal recommendations

Signal Group	Impedance	Layout Tolerance (±)
All signals, unless specified	50 Ω SE	10%

Table 2-7. Impedance signal recommendations (continued)

Signal Group	Impedance	Layout Tolerance (±)
USB Diff signals	90 Ω Diff	10%
Diff signals: DDR, Phy IC to Ethernet Connector	100 Ω Diff	10%

The following figure shows the dimensions of a stripline and microstrip pair. Figure 2-30 shows the differential pair routing.

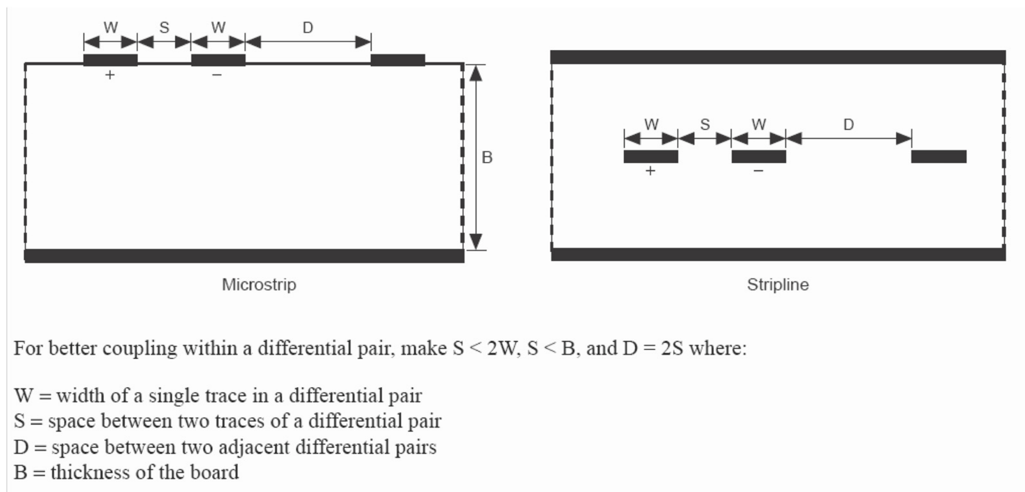


Figure 2-29. Microstrip and stripline differential pair dimensions

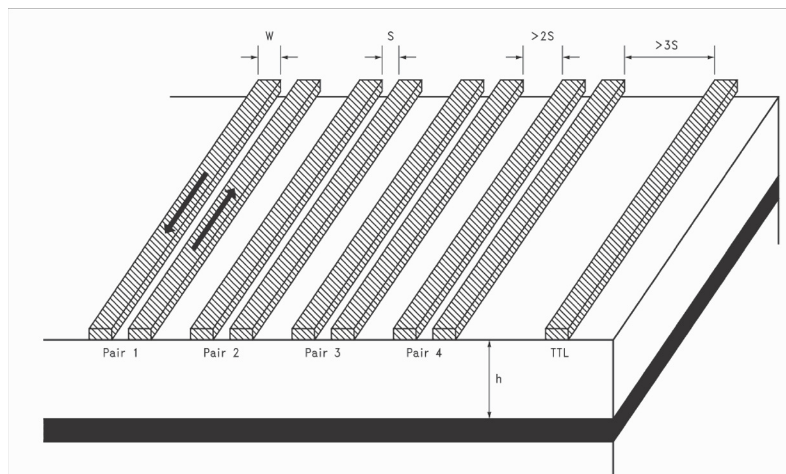


Figure 2-30. Differential pair routing

- The space between two adjacent differential pairs should be greater than or equal to twice the space between the two individual conductors.

2.9 Reference resistors

NOTE

The reference resistor and the connection should be placed away from noisy regions. Noise induced on it may impact the internal circuit and degrade the interface signals.

2.10 ESD and radiated emissions recommendations

The PCB design should use six or more layers, with solid power and ground planes. The recommendations for ESD immunity and radiated emissions performance are as follows:

- All components with ground chassis shields (USB jack, buttons, and so forth) should connect the shield to the PCB chassis ground ring.
- Ferrite beads should be placed on each signal line connecting to an external cable. These ferrite beads must be placed as close to the PCB jack as possible.

NOTE

Ferrite beads should have a minimum impedance of 500 Ω at 100 MHz with the exception of the ferrite on USB_5V.

- Ferrite beads should NOT be placed on the USB D+/D– signal lines as this can cause USB signal integrity problems. For radiated emissions problems due to USB, a common mode choke may be placed on the D+/D– signal lines. However, in most cases, it should not be required if the PCB layout is satisfactory. Ideally, the common mode choke should be approved for high speed USB use or tested thoroughly to verify there are no signal integrity issues created.
- It is highly recommended that ESD protection devices be used on ports connecting to external connectors. See the IMX6SLEVK (available on the Freescale website) for detailed information about ESD protection implementation on the USB interfaces.
- If possible, stitch all around the board with vias with 100 mils spacing between them connected to GND planes with exposed solder mask to improve EMI.

2.11 Component placement recommendations

Adhere to the following recommendations when placing components.

- Place components such that short and/or critical routes can be easily laid out.
  - Critical routes determine component location.
  - Orient devices to facilitate routes (minimize length and crossovers).
- Consider placing the following pairings adjacent.
  - i.MX and DDR
  - PHY and associated jack
  - Jack and CODEC input
  - Bluetooth® (or other RF) and antenna

2.12 Reducing skew and phase problems in deferential pairs traces

Differential pair technology has evolved to require more stringent checking in the area of phase control. This is evident on the higher data rates associated with parallel buses such as DDR, or Ethernet. In the simplest of terms, Diff Pair technology sends opposite and equal signals down a pair of traces. Keeping these opposite signals in phase is essential to assuring that they function as intended.

Figure 2-31 and Figure 2-32 show two examples of static routing where a match is achieved without needing to tune one element of the differential pair.

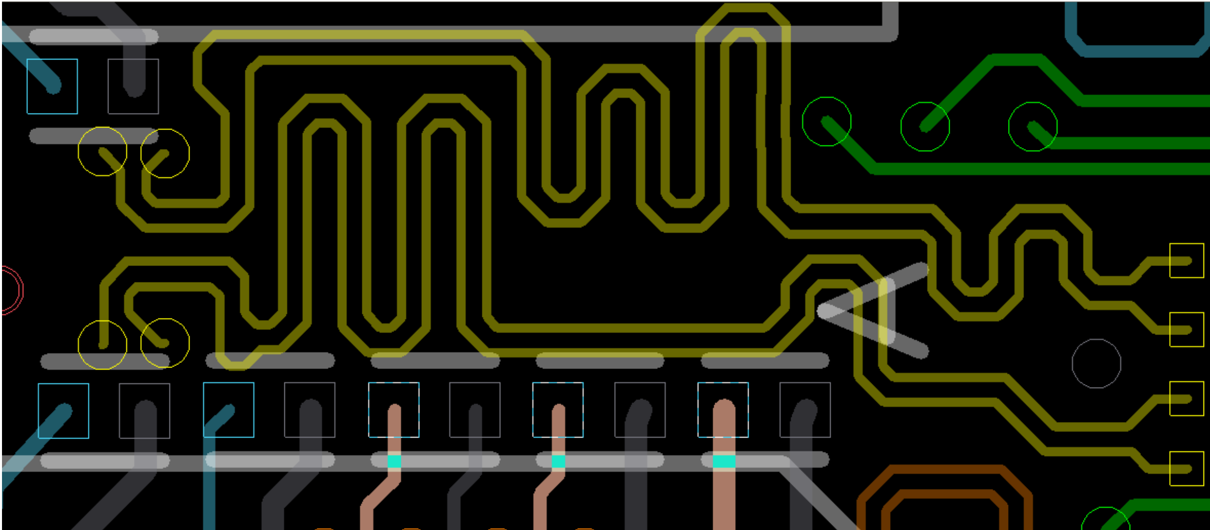


Figure 2-31. Yellow traces diff pairs 1

The following figure shows the addition of a delay trace to one element of the differential pair to avoid length mismatch (which reduces skew and phase problems). The green box marks the detail.

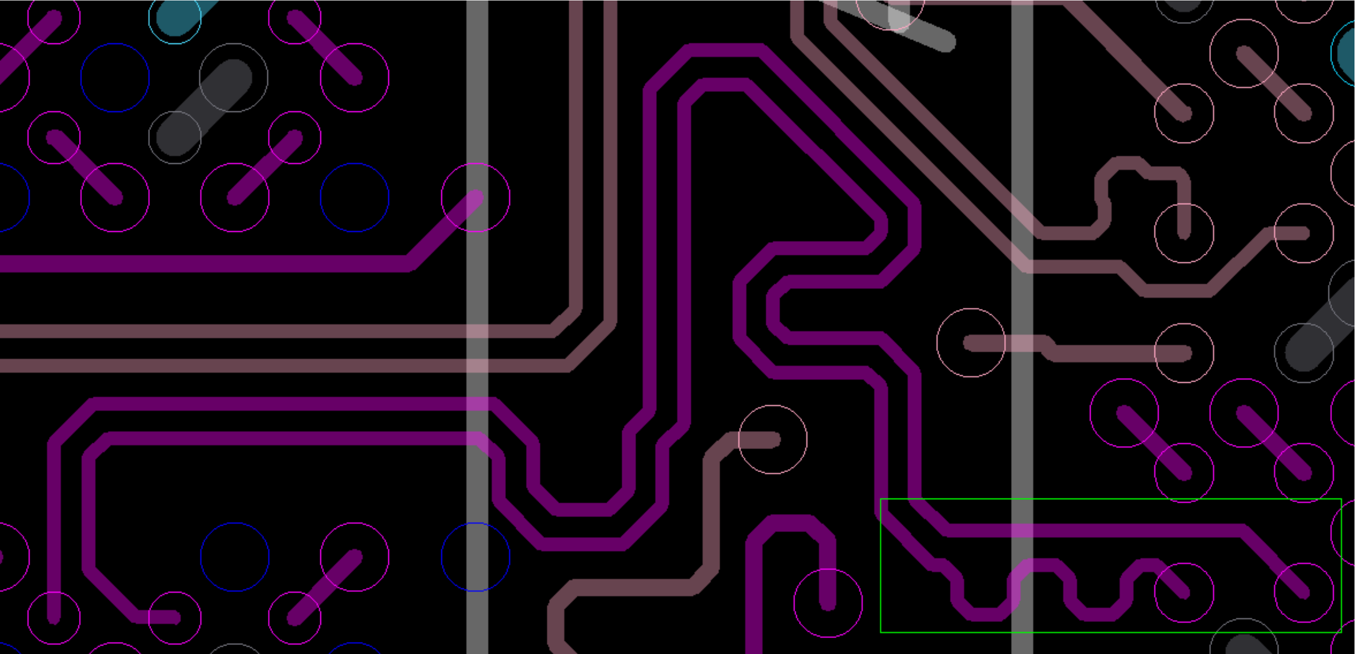


Figure 2-32. Small bumps added to the shorter differential pair

Having this delay reduces skew and phase problems.

Chapter 3
Requirements for Power Management

3.1 Power management requirements overview

- This chapter provides the power requirements for the following i.MX 6SoloLite.

3.1.1 Voltage domains overview

The chip have several voltage domains that may need to be supplied with different voltages depending on system needs. The chip internal regulators and its complementary PMIC PF0100 provide a complete and simple way to supply each voltage domain with different voltages when needed. Section 3.4, "Connection diagrams," shows the internal regulators and the connections to PF0100.

3.1.2 PF0100 overview

PF0100 consists of the following components used to supply the i.MX6 voltage domains as well as other blocks on the system:

- 4 buck regulators
- 1 boost regulator
- 8 LDOs

The default PF0100 power-up sequence is programmed to fit the requirements of the i.MX 6 series families of processors. However, the PF0100 can be adjusted to meet the specific requirements for system applications by using the one time programmable (OTP) feature.

3.2 Requirements for a generic interface between chip and PF0100

Table 3-1 shows the generic interface between the chip and PF0100, using a suitable power-up sequence.

For more info about PF0100 functionality and the i.MX 6SL processors' power requirements, see the product data sheet.

The following table shows the i.MX6SL chip-specific interface.

Table 3-1. Interface between i.MX6SL and PF0100

Voltage rail	Supply reg	Voltage (V)	Current (A)	Generated by	Power-up sequence	Notes
VDD_ARM_IN	SW1A/B	1.375	2.5	PF0100	1	—
VDD_ARM_CAP	Note 1	—	—	i.MX	—	—

Table 3-1. Interface between i.MX6SL and PF0100 (continued)

Voltage rail	Supply reg	Voltage (V)	Current (A)	Generated by	Power-up sequence	Notes
VDD_SOC_IN	SW1C	1.375	1.75	PF0100	1	—
VDD_PU_IN						—
VDD_SOC_CAP	Note 1	—	—	i.MX	—	—
VDD_PU_CAP	Note 1	—	—	i.MX	—	—
VDD_HIGH_IN	SW2	3.15	2	PF0100	2	—
VDD_HIGH_CAP	Note 1	—	—	i.MX	—	—
VDD_SNVS_IN	VSNVS	3	400 μA	PF0100	0	—
VDD_SNVS_CAP	Note 1	1.1	—	i.MX	—	—
NVCC_PLL	Note 1	—	—	i.MX	—	—
USB_OTG1_VBUS	—	—	—	—	—	Connect to VBUS pin of USB plug
USB_OTG2_VBUS	—	—	—	—	—	Connect to VBUS pin of USB plug
VDD_USB_CAP	Note 1	—	—	i.MX	—	—
NVCC_DRAM	SW3A/B	1.2	2.5	PF0100	4	SW3 can be configured from 0.4 to 3.3V so the right voltage is chosen for the respective DDR technology.
NVCC_DRAM_2P5	VDDHIGH_ CAP or VGEN5	2.5	0.1	i.MX or PF0100	5	—
DRAM_VREF	VREFDDR	0.6	0.01	PF0100	4	—
NVCC33_IO	SW2	3.15	2	PF0100	2	—
NVCC18_IO	VGEN4	1.8	0.35	PF0100	3	—
NVCC_1P2V	VGEN1	1.2	0.1	PF0100	4	—

1 These voltage domains are supplied by i.MX6 internal regulators.

The following table shows the PF0100 regulators that are available to supply the rest of the system circuitry for an i.MX6SL interface.

Table 3-2. PF0100 regulators for other system circuitry (i.MX6SL)

Supply	Output voltage(V)	Step size (mV)	Maximum load current (mA)
SW4	0.5 × SW3A_OUT, 0.4 – 3.3	25/50	1000
VGEN2	0.8 – 1.55	50	250
VGEN3	1.8 – 3.3	100	100

Table 3-2. PF0100 regulators for other system circuitry (i.MX6SL) (continued)

Supply	Output voltage(V)	Step size (mV)	Maximum load current (mA)
VGEN5	1.8 – 3.3	100	100
VGEN6	1.8 – 3.3	100	200

3.3 i.MX6 internal regulators

These chips have been equipped with 7 internal regulators that simplify the power supply scheme of the system. The following table shows the regulators' power requirements. See Section 3.4, "Connection diagrams," for the distribution and connections of these LDOs.

Table 3-3. Internal regulator power requirements[1]

LDO	Output voltage (V)
LDO_ARM	1.1
LDO_SOC	1.2
LDO_PU	1.1
LDO_2P5	2.5
LDO_1P1	1.1
LDO_SNVS	1.1
LDO_USB	3.0

[1] Each LDO voltage rail can be adjusted through LDO registers. See the *i.MX 6SoloLite Applications Processors for Consumer Products* data sheet (IMX6SLCEC) and the "Power Management Unit (PMU)" chapter of the *i.MX 6SoloLite Applications Processor Reference Manual* (IMX6SLRM) for more details.

3.4 Connection diagrams

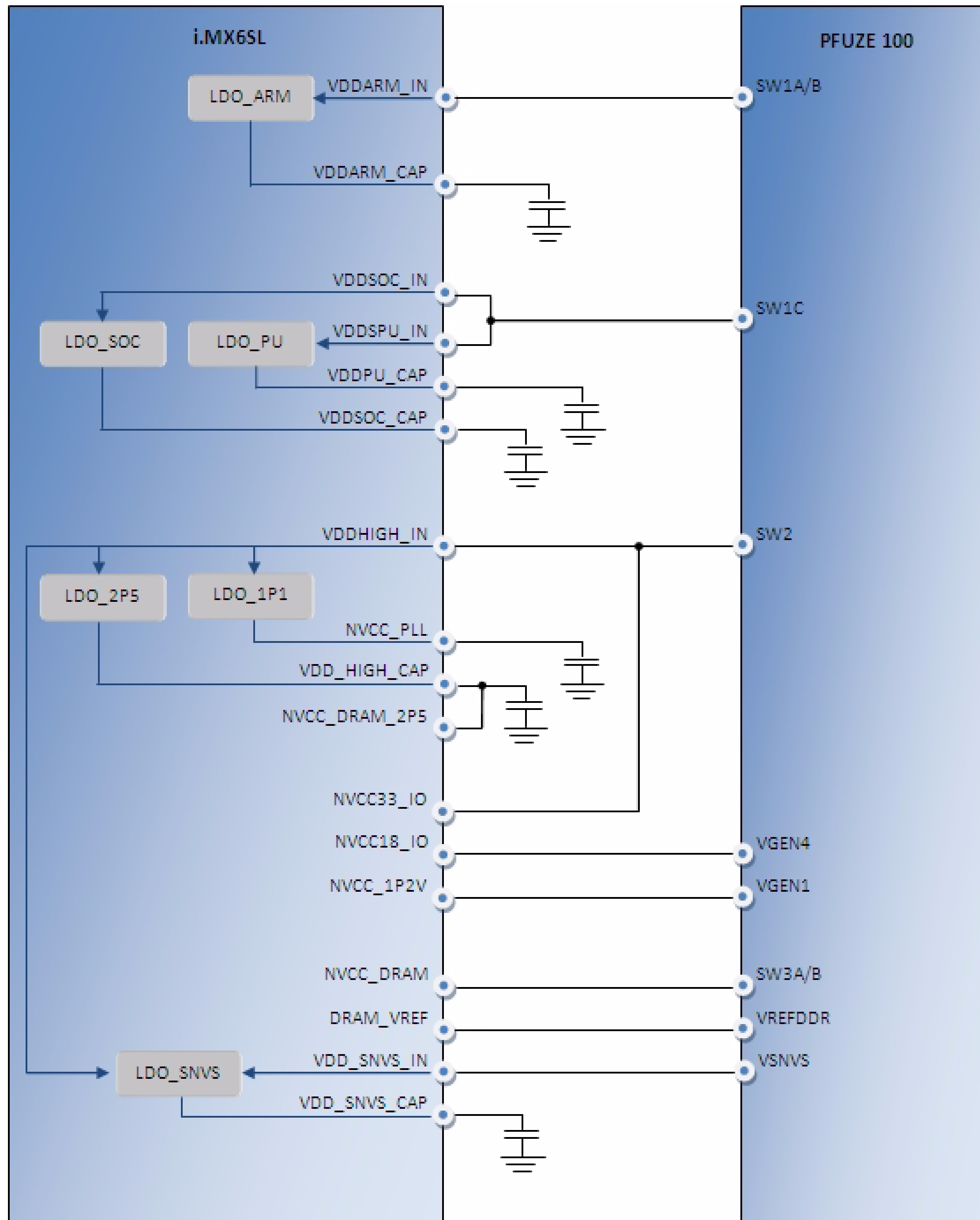
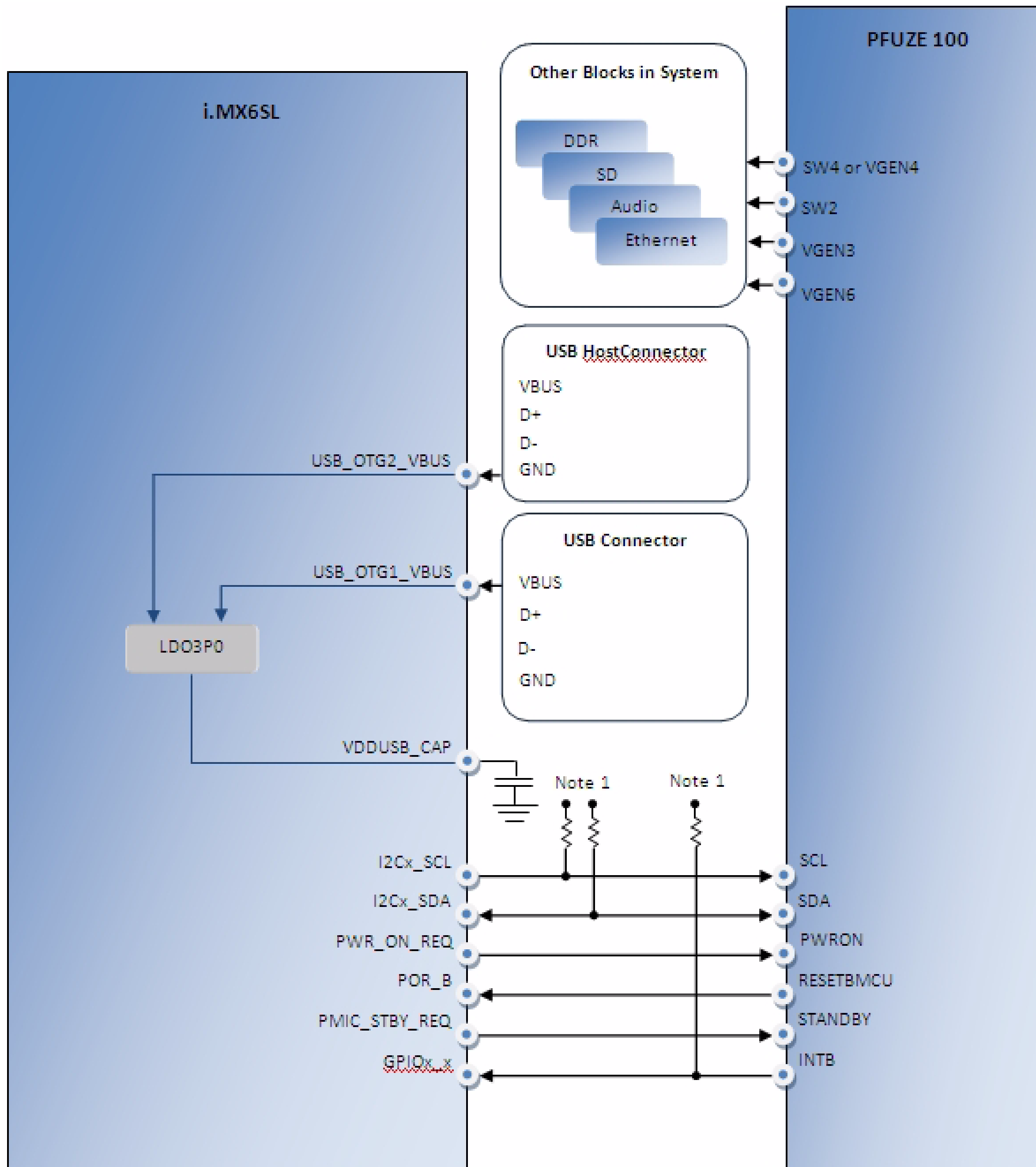


Figure 3-1. i.MX6SL / PF0100 Connections Diagram, 1 of 2



[1] GPIO I/O logic level is configurable to either 3.3 V or 1.8 V after power-up. Pull-up voltage has to be chosen accordingly.

Figure 3-2. i.MX6SL / PF0100 Connections Diagram, 2 of 2

3.5 Video power recommendations

VDD_PU_CAP is the supply for the internal graphic processing units (GPU). For graphic intensive operations, the GPU requires a lot of power and may undergo large swings of instantaneous current requirements. Therefore, the power supply to the GPU must be designed to handle relatively large surges

of current at high frequencies from the original source to the processor input for power (VDD_SOC_IN) and at the output of the internal regulator for GPU operations (VDD_PU_CAP). The following list provides recommendations for each specific point along the current supply path. It may be necessary to implement all of these recommendations to ensure that one particular point along the supply path does not become a current choke point.

- The voltage with which VDD_SOC_IN will be fed must have a maximum tolerance of ± 25 mV. PF0100's SW1C is already designed with this tolerance. Care must be taken if the design uses a different regulator.
- VDD_SOC_CAP and VDD_PU_CAP bulk capacitance must be equal to 22 μF so that start up current through the on board LDOs is reduced.
- These bulk capacitors must be very close to the VDD_SOC_CAP and VDD_PU_CAP pins respectively and the connecting traces must be as thick as the design allows so the ability of being a bulk capacitor for high speed operations is not limited.
- VDD_SOC_IN requires 66 μF of bulk capacitance because it supplies power for both VDD_SOC_CAP and VDD_PU_CAP.

Chapter 4
Using the Clock Connectivity Table

This chapter provides a reference table of the root clock default speed and a list of the i.MX modules available to exit stop mode.

4.1 Root clocks

Clock connectivity is described in the "System Clocks Connectivity" section in the CCM chapter of the chip reference manual. This section contains a series of tables that describe the clock inputs of each module and which clock is connected to it.

NOTE

In some cases, a clock is associated with an external interface and is sourced from a pad (mainly through IOMUX) and not from the CCM. Such clocks do not appear in the clock connectivity table. They are found in the "External Signals and Pin Multiplexing" chapter.

Clock gating is done with the low power clock gating (LPCG) module based on a combination of the clock enable signals. For information about how the clock gating signals are logically combined, see the LPCG section in the CCM chapter of the chip reference manual.

Table 4-1 lists the available clock sources and the default frequencies that are configured by design. In some cases, users need to divide the clock inside the module when the maximum frequency is used in order to meet the protocol requirements. CCM (the clock controller module) generates and drives the clock sources.

For information about how the root clocks are generated, see the clock generation diagrams in the CCM chapter of the chip reference manual.

Table 4-1. Clock roots

Clock Root	Default Frequency (MHz)
ARM_CLK_ROOT	792[1]
PERCLK_CLK_ROOT	66
AHB_CLK_ROOT	132
IPG_CLK_ROOT	66
MMDC_CLK_ROOT	396
USDHCn_CLK_ROOT	198
SSIn_CLK_ROOT	63.5
OCRAM_CLK_ROOT	264

Table 4-1. Clock roots (continued)

Clock Root	Default Frequency (MHz)
CSI_CORE_CLK_ROOT	4.8
LCDIF_AXI_CLK_ROOT	264
ACLK_EIM_SLOW_CLK_ROOT	264
GPU2D_OVG_CORE_CLK_ROOT	528
GPU2D_CORE_CLK_ROOT	528
PXP_AXI_CLK_ROOT	87.3
EPDC_AXI_CLK_ROOT	87.3
LCDIF_PIX_CLK_ROOT	180
EPDC_PIX_CLK_ROOT	56.5
SPDIF0_CLK_ROOT	30
SPDIF1_CLK_ROOT	30
EXTERN_AUDIO_CLK_ROOT	30
ECSPI_CLK_ROOT	4
UART_CLK_ROOT	4

[1] During the power-up sequence and u-boot operation, the core operates at 396 MHz. Freescale's Linux kernel changes the frequency to 996 MHz when the OS becomes operational.

4.2 Waking the core up from stop mode

The following modules can wake the core up from stop mode.

- ECSPI
- EIM
- EPIT
- GPC
- GPIO
- GPT
- I2C
- KPP
- SDMA
- UART
- USB

Chapter 5
Using the IOMUX Design Aid

This chapter provides users with the basic information required to use the IOMUX system design aid (IOMux.exe). The IOMUX design aid facilitates the assignment of internal signals to external device balls/pins by helping users:

- Record signal assignments for the supported i.MX device
- Identify conflicts, allowing them to be resolved in real time
- Add notes or comments for each signal to the list of recorded assignments
- Generate C code to configure the IOMUXC registers according to the user’s design
- Move signals to different modules to order configuration code into logical functions

Users can save design configurations for future use and/or export them for use in schematics or software source code as supplementary documentation of a system.

The following figure shows a screenshot of the IOMUX application window with various areas labeled.

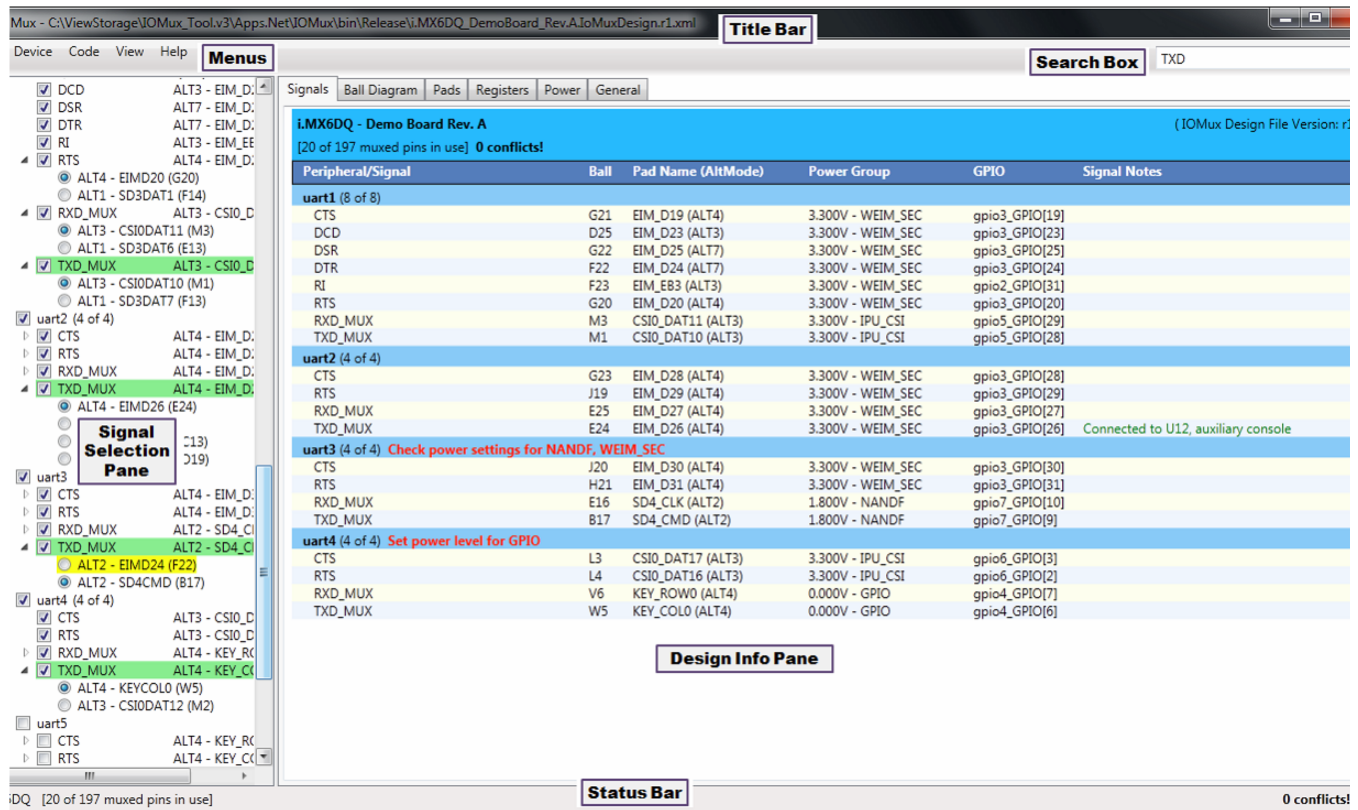


Figure 5-1. IOMUX tool for the i.MX6 families of applications processors

5.1 Application requirements

The IOMUX application requires that the following be installed:

- Microsoft Windows XP or newer
- Microsoft's .NET Framework, .NET Framework to 4.0 or newer.

5.2 IOMUX tool version

The IOMUX application *i.MX 6SoloLite IOMux Tool v3.3* or later supports the i.MX 6SoloLite.

5.3 IOMUX tool location

To obtain the IOMUX tool for the chip, consult your Freescale sales representative or download the IOMUX tool from www.freescale.com.

Note that the IOMUX tool must be version v3.3 or later.

5.4 Learning to use the IOMUX tool

Consult the IOMUX user's manual file inside the package for a detailed walkthrough of how to use the IOMUX tool.

Chapter 6
Configuring JTAG Tools

This chapter explains how to configure JTAG tools for debugging. The JTAG module is a standard JEDEC debug peripheral. It provides debug access to important hardware blocks, such as the ARM® processor and the system bus, which can give users access and control over the entire chip.

To prevent JTAG manipulation while allowing access for manufacturing tests and software debugging, the i.MX 6SL series processor incorporates a secure JTAG controller for regulating JTAG access. The secure JTAG controller provides four different JTAG security modes, which are selected by e-Fuse configuration. For more information about the security modes, see the "Security" section in the "System JTAG Controller (SJC)" chapter of the chip reference manual.

6.1 JTAG tool requirements

To use JTAG tools, your system must have the following:

- Windows based PC
- RVDS v4.1 package or newer
- RealView ICE box connected to your computer

Freescale recommends making the JTAG port accessible during platform initial validation bring-up and for software debugging. It is accessible in all development kits from Freescale.

Multiple tools are available for accessing the JTAG port for tests and software debugging. Freescale recommends use of the ARM JTAG tools for compatibility with the ARM core. However, the JTAG chain as described in the following sections should work with non-ARM JTAG tools. For more information about configuring non-ARM tools, contact the third party tool vendor for support.

6.2 Extra JTAG functionality

Additional CoreSight debug components, such as trace machines using DS-5 debug software and DSTREAMER hardware, can be used for extra JTAG functionality. However, they are not mandatory for a basic configuration and are beyond the scope of this document.

NOTE

There is no option for using RVDS at its version at time of publication (4.1) because it does not support PTM (i.MX 6 series trade module).

6.3 Updating your RealView ICE

Before using the RealView ICE for JTAG debugging, ensure you have the most up-to-date version available. To update your RealView ICE, perform the following steps:

1. Launch the RVI Update utility by using the following path:
   `Start → Programs → ARM → RealView ICE v4.1 → RealView ICE Update`
2. Connect to the ICE by selecting it from the list, as shown in the following figure.

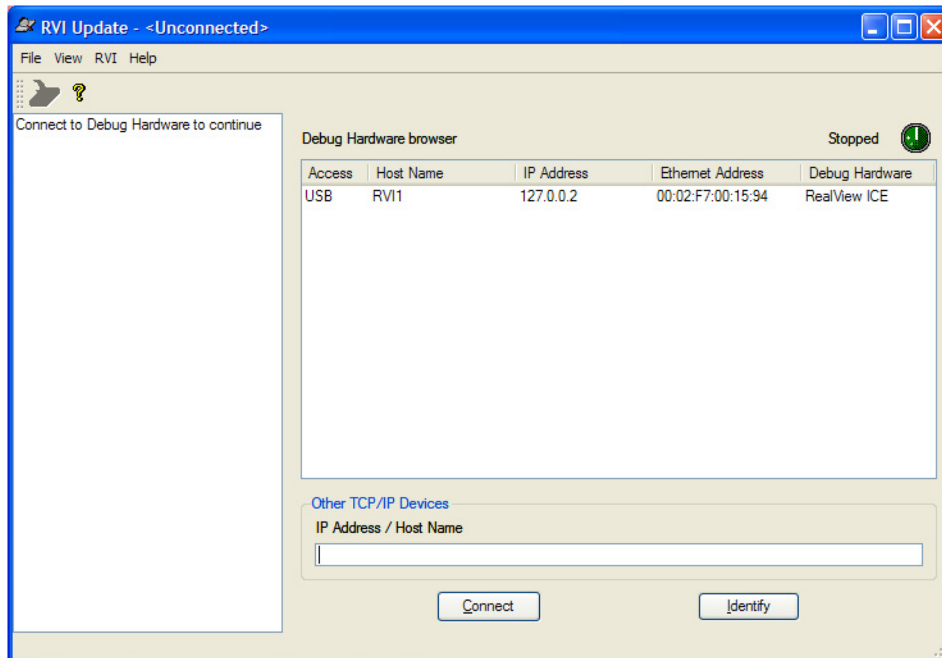


Figure 6-1. Connecting to ICE

NOTE

The ICE must be disconnected from any other target at this step.

3. Select the firmware update from the upside menu: RVI → Install Firmware Update
4. Select the following file (or an equivalent more recent version):
   `C:\Program Files\ARM\RVI\Firmware\4.2\23\ARM-RVI-4.3.0-1-base.rvi`
5. Select “Continue” from the install update window and wait until the update is complete.

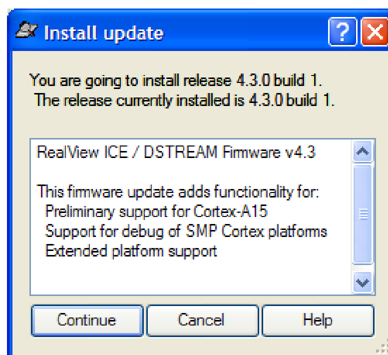


Figure 6-2. Install update window

6. RVI automatically reboots.
7. Upon reconnecting to the RVI, you should see version number 4.3.0 build 1 or a later version number. The exact version name should match with the version number installed in step 4 (see the following figure).

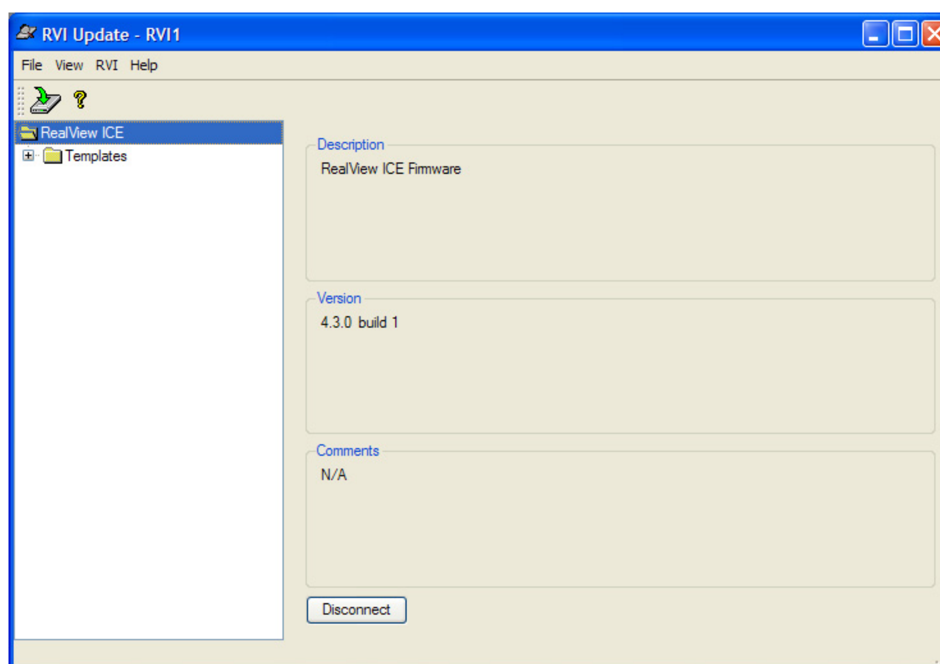


Figure 6-3. RVI window after reconnecting

6.4 Defining the JTAG chain

To define the JTAG chain for an ARM Cortex®-A9 based chip, perform the following steps:

1. Find `Freescale_iMX6 SL.rvs` at the following location: `…/My Documents\ARM\rvconfig\platformFiles`

NOTE

Be sure to use this path exactly, or the tool-chain configuration will not be available from the Debugger-Connect to Target.

Contact your sales representative or go to www.freescale.com to obtain your copy of the `Freescale_iMX6 SL.rvs` file.

2. Run RealView Debugger by using the following path: `Start` → `Programs` → `ARM` → `RealView Development Suite v4.1` → `RealViewDebugger v4.1`
3. Select Connect to Target in the RealView Debugger upside menu: `Target` → `Connect To Target`
4. Press `Add` near `RealView ICE`.

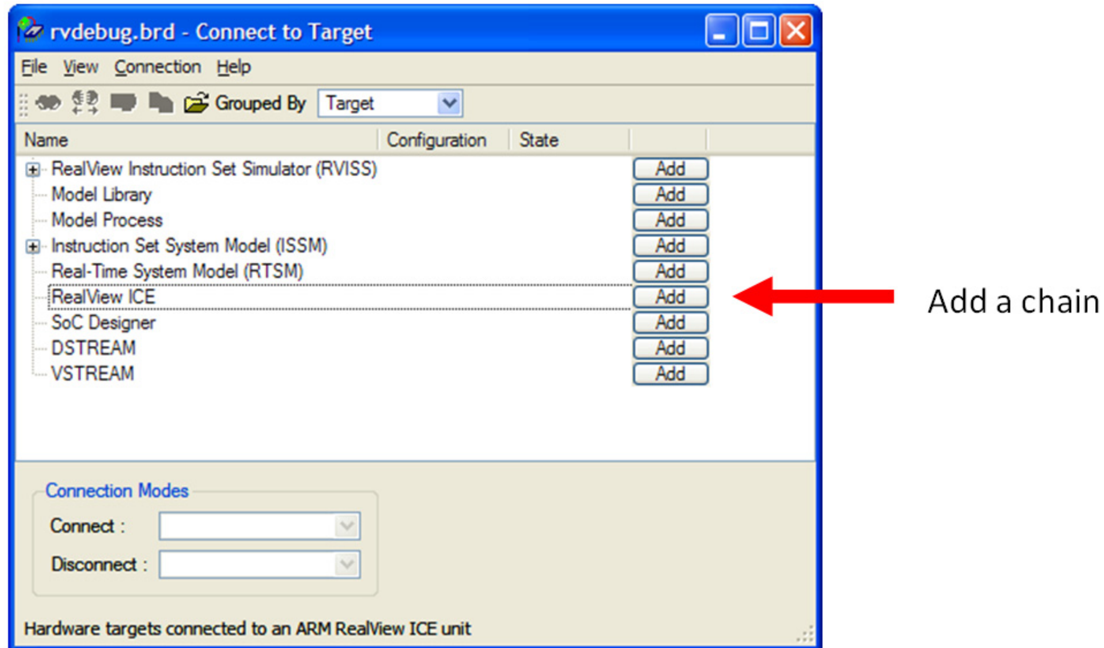


Figure 6-4. Adding your ICE

5. Select your ICE from the list and press `Connect` (see Figure 6-1).
6. In the new window, choose `Select Platform…`
7. Expand the “Freescale” list and select `iMX6 Q`.
8. Save the file (File → Save).
9. Close the window.

After finishing this procedure, you should see the following screen:

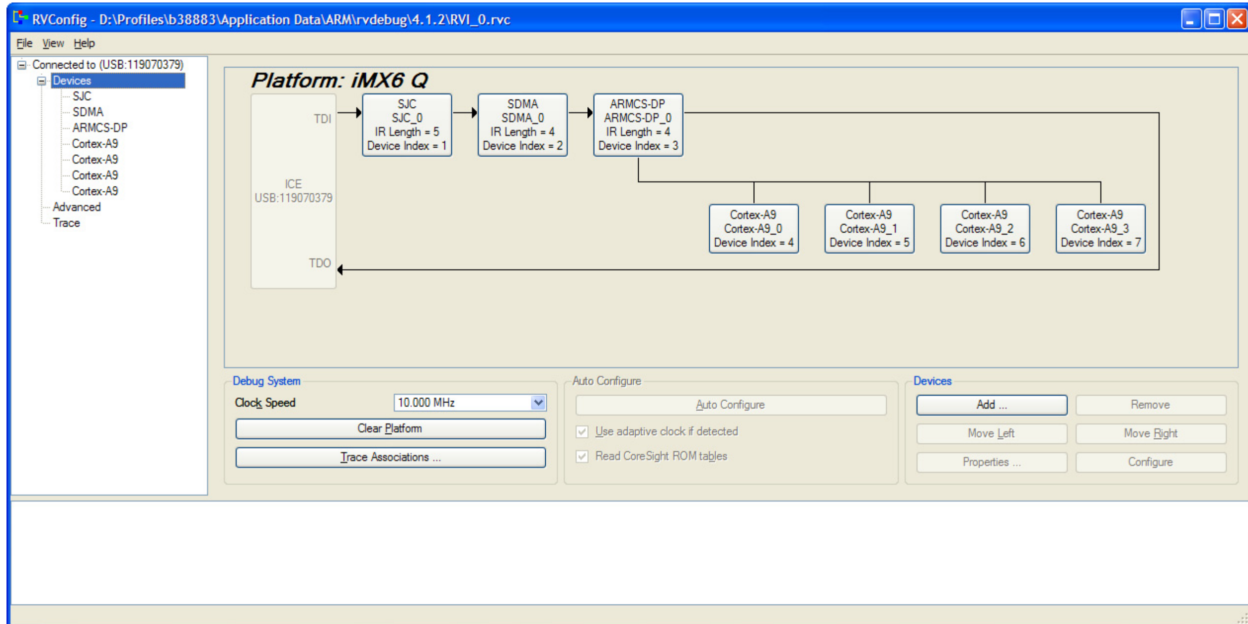


Figure 6-5. RealView debugger screenshot

Add the correct amount of Cortex A-9 cores desired to access your CPUs.

6.5 Reading a register with RealView Debugger v4.1

To read a register, perform the following steps:

1. Open the RealView Debugger 4.1 and connect to the target, as shown in the following figure.

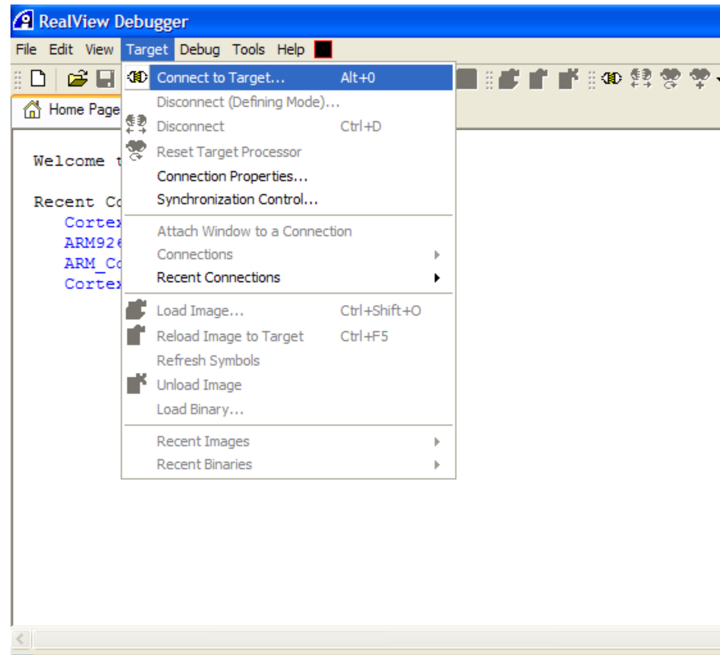


Figure 6-6. Connecting to the target

2. You are now at `rvdebug.brd`; if you have successfully completed your setup, it looks like the following screenshot:

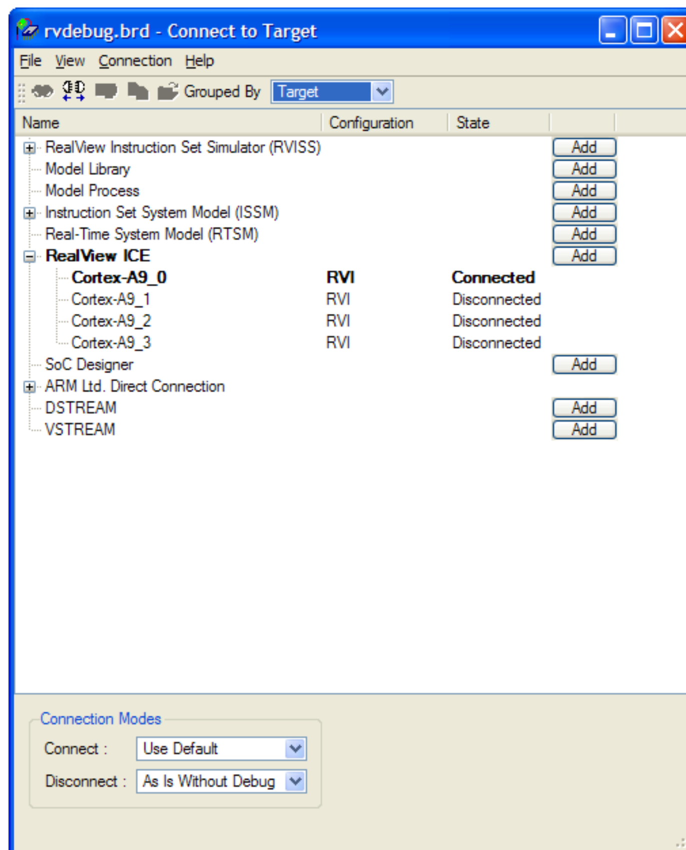


Figure 6-7. Establishing a connection to the core

3. Establish the connection to the core of your choice by using the Connect icon or the shortcut CTRL+N.

You now have a new RVI configuration with four Cortex-A9 targets and the RealView Debugger up and running. You can now use the RealView Debugger window to access a register, as shown in the following figure.

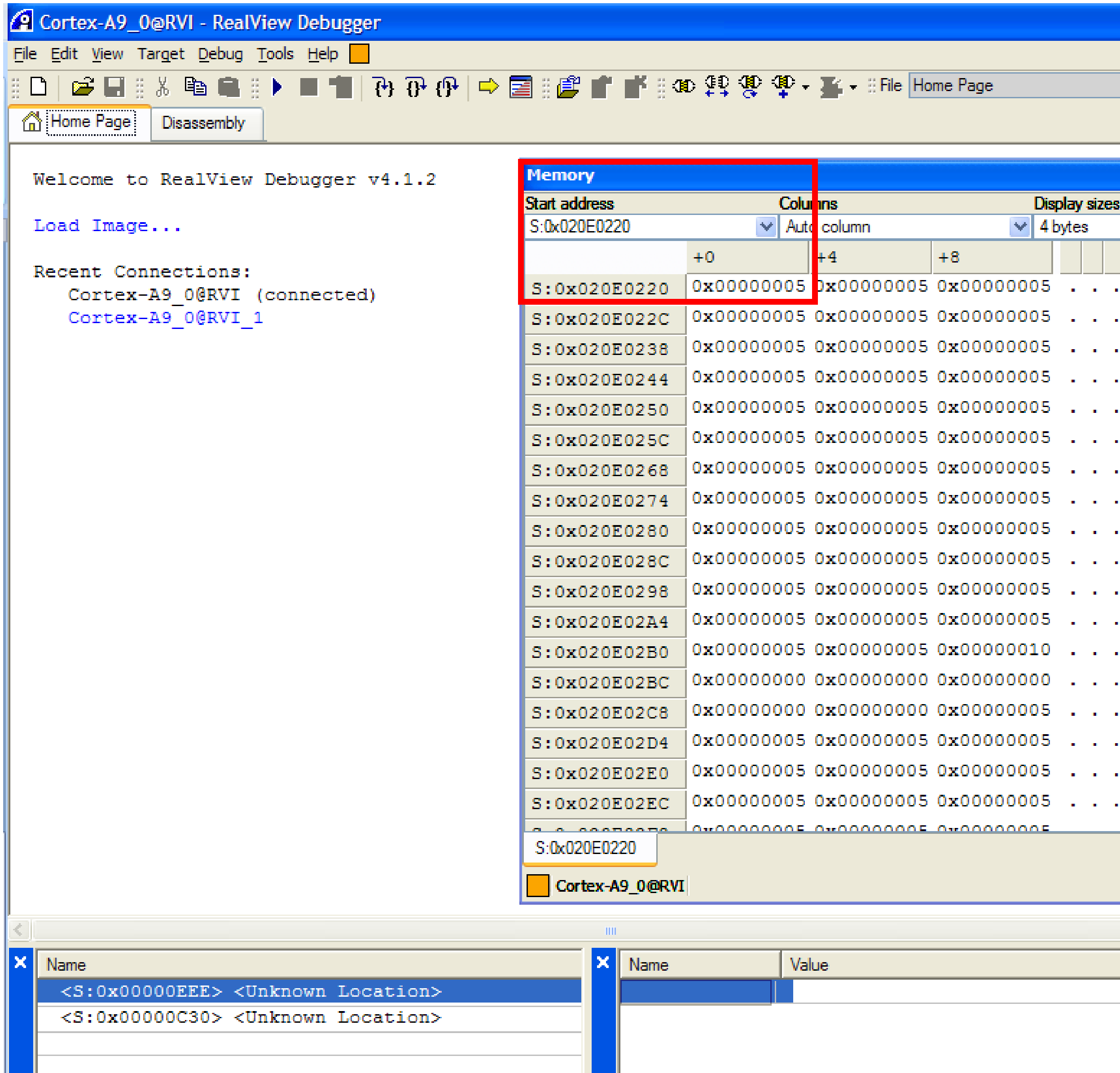


Figure 6-8. Accessing a register

Figure 6-8 shows an example of using the RealView Debugger to access the IOMUX register IOMUXC_SW_MUX_CTL_PAD_GPIO_0, whose address is 0x020E0220 and whose default value after reset is 0x5.

6.6 CoreSight Base address references

The CoreSight base address is as follows: CPU#0: 0x82150000

Chapter 7
Avoiding Board Bring-up Problems

This chapter provides recommendations for avoiding typical mistakes when bringing up a board for the first time. These recommendations consist of basic techniques that have proven useful in the past for detecting board issues and addressing the three most typical bring-up pitfalls: power, clocks, and reset. A sample bring-up checklist is provided at the end of the chapter.

7.1 Power system overview

7.1.1 Block diagram

The block diagram in Figure 7-1 shows major power systems blocks and internal/external connections for the i.MX 6SoloLite processor.

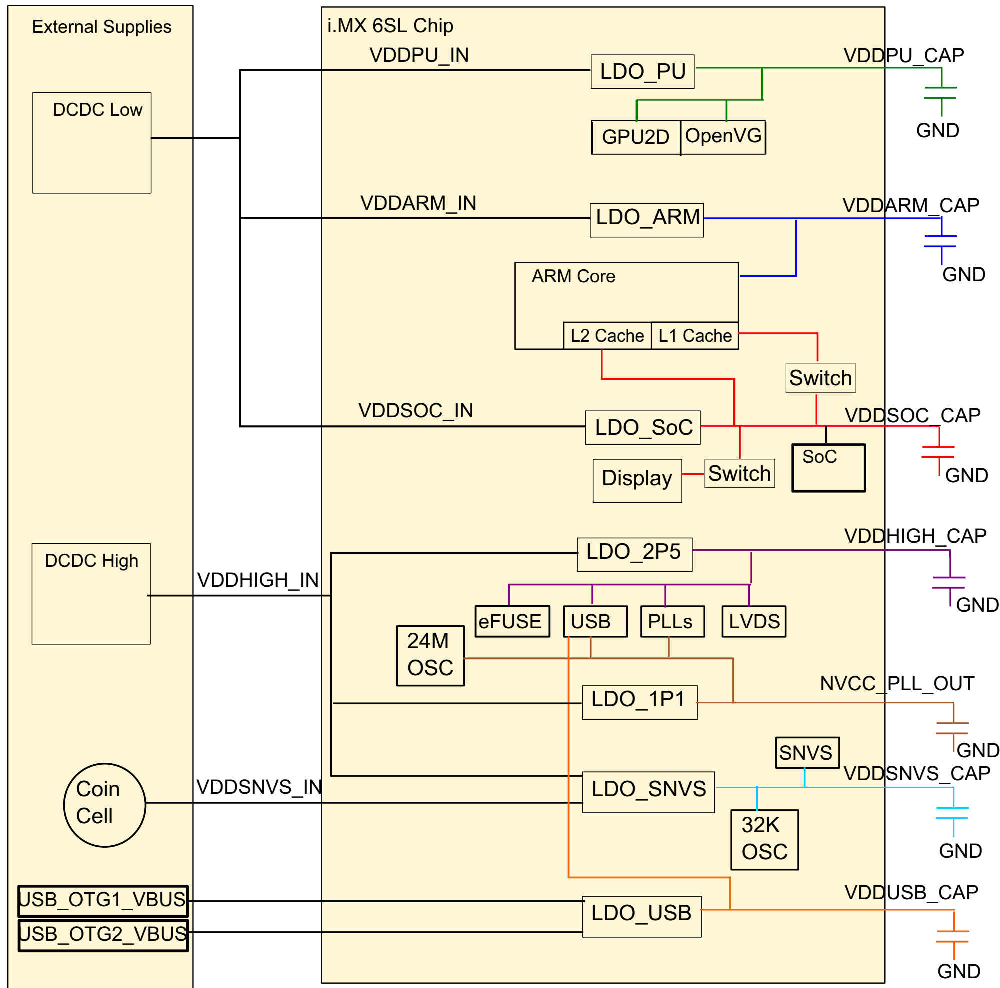


Figure 7-1. i.MX 6SoloLite system-on-chip power block diagram

7.1.2 Maximum supply requirements

Table 7-1 shows the preliminary maximum supply requirements for each power rail from the PMIC into the power supply inputs of the i.MX6Sololite SoC.

Table 7-1. Maximum current supply requirements

Power Supply Rail	Suggested Max Current Requirement	Comments
1.2V rail => VDD_SOC_IN + VDD_PU_IN + VDD_ARM_IN	1.95A	
2.8V - 3.6V rail => NVCC33_IO + VDD_HIGH_IN (Analog)	300mA*	Requirement for MX6Sololite SoC only, does not include external device. *The max current may differ according to the switching speed, capacitive loading and number of GPIOs used for NVCC33_IO
1.8V rail => NVCC18_IO	125mA*	Requirement for MX6Sololite SoC only, does not include external devices. *The max current may differ according to the switching speed, capacitive loading and number of GPIOs used for NVCC18_IO
1.2V rail (LPDDR2) => NVCC_DRAM	200mA	Requirement for MX6Sololite SoC only, does not include DRAM IC
5.0V => USB VBUS	60mA	Requirement for MX6Sololite SoC only, does not include additional external USB loads.

7.1.3 Internal LDO description

The i.MX6Sololite internal power system consists of 7 LDOs. Each LDO is described in Table 7-2.

Table 7-2. Internal LDO descriptions

LDO Name	Description
LDO_ARM	Sources the power to the ARM core.
LDO_PU	Sources the power to the graphics processor blocks.
LDO_SOC	Sources power to the rest of the SoC gates.
LDO_2P5	Provides 2.5V power to the chip serial interfaces, PLLs, and DRAM pre-drivers.
LDO_1P1	Provides power to the PLL digital circuitry.
LDO_SNVS	This is a low power LDO, which supplies the low power 1.1V to the 32kHz RTC and other SNVS circuitry.
LDO_USB	Powers the on-chip USB blocks. Also handles the 5V power muxing when two 5V sources are present.

7.2 Power sequencing

Table 7-3 shows the power-up and power-down sequence orders. Two cases are shown: Using the i.MX6Sololite internal supplies (non-bypass) and bypassing the internal LDO (Low Drop Out linear regulator) supplies.

Table 7-3. Power sequence

Power Rail Name	Using all internal LDOs		Internal LDOs Bypassed		Comments
	Power Up[1]	Power Down	Power Up	Power Down	
VDD_SNVS_IN	1	6	1	9	
VDD_HIGH_IN / NVCC33_IO (NVCC_HIGH)	4	3	2	8	
VDD_HIGH_CAP	Powered internally	Powered internally	3	7	
NVCC18_IO (NVCC_LOW)	5	2	4	6	
NVCC_PLL	Powered internally	Powered internally	5	5	
NVCC_DRAM_IN	6	1	6	4	
VDD_ARM_IN	2	5	7	3	For bypass mode, DC levels must be kept within +/-25mV of the nominal voltage level. Maximum noise level shall be kept within +/-25mVp-p of the supply voltage for corresponding operating frequency.
VDD_PU_IN	3	4	8	2	For bypass mode, DC levels must be kept within +/-25mV of the nominal voltage level. Maximum noise level shall be kept within +/-25mVp-p of the supply voltage for corresponding operating frequency.
VDD_SOC_IN	3	4	9	1	For bypass mode, DC levels must be kept within +/-25mV of the nominal voltage level. Maximum noise level shall be kept within +/-25mVp-p of the supply voltage for corresponding operating frequency.
USB_OTG1_VBUS /USB_OTG2_VBUS	N/A	N/A	N/A	N/A	Off by default. Software must turn it on.

[1] Internally controlled by processor.

NOTE

VDD_ARM_IN, VDD_PU_IN, and VDD_SOC_IN can startup at the same. However, VDD_ARM_IN and VDD_PU_IN must be at their target values within 1ms of VDD_SOC_IN.

There are no special timing requirements for USB_OTG1_VBUS/USB_OTG2_VBUS.

7.2.1 Detailed power-up sequence

The power-up sequence timing diagram is shown in Figure 7-2.

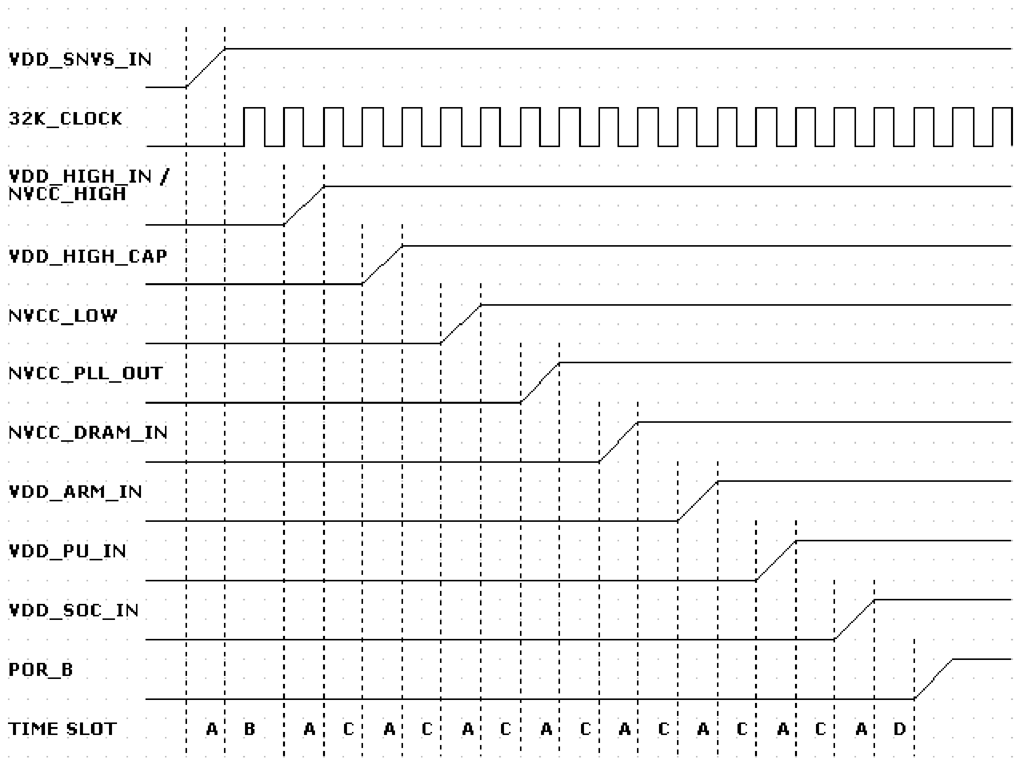


Figure 7-2. Startup Timing Diagram—LDO Bypass

NOTE

VDD_ARM_IN, VDD_PU_IN add VDD_SOC_IN can startup at the same. However, VDD_ARM_IN and VDD_PU_IN must be at their target values within 1ms of VDD_SOC_IN.

Table 7-4. Startup timing diagram parameters

Time Slot	Description	Timing Requirement
A	Ramp time for each rail from 0V to the target voltage.	Max. total ramp time = 1ms.
B	The 32k clock from the PMIC can begin clocking the processor as soon as VDD_SNVS_IN reaches its target voltage. The PMIC should then wait until at least one 32k clock cycle has completed before ramping the remaining rails.	At least one 32k clock cycle.
C	Time between previous rail reaching its target value and next rail beginning to ramp.	The next power rail can begin ramping as soon as the previous rail reaches its target value. No max. time requirement.
D	Time between last rail reaching target value and de-asserting POR_B (power on reset)	POR_B can be de-asserted as soon as the final rail reaches its target value. No max. time requirement.

7.2.2 VDD_SNVS_IN power-up ramp rate

During power-up, if VDD_SNVS_IN's ramp rate is significantly faster than the ramp rate seen at VDD_SNVS_CAP, VDD_SNVS_IN domain may pull higher current. In order to prevent in rush current, slow ramp rate is recommended for VDD_SNVS_IN. Assuming 0.22 µF attached to VDD_SNVS_CAP with low trace capacitance and load capacitance, 0.5 V/ms ramp rate or slower is recommended.

7.2.3 Power-down sequence

Before powering down the power rails, the PMIC should assert POR_B. When powering down the power rails, the power-down sequence order in table 3-4 should be followed. The POR_B signal, power rails, and 32K external clock power-down sequence should be done in the reverse order of figure 2. Each power rail should be turned off and discharged to ground by the PMIC with an impedance of approximately 50-100 ohms. Each power rail should be turned off and discharged to at least 10% of its initial value until before turning off and discharging the next power rail. The PMIC should also ensure the external 32.768kHz clock is turned off before turning off the VDD_SNVS_IN supply.

7.2.4 Power-up sequence in low voltage system design

On i.MX6SL, NVCC33_IO voltage rail is selected as the default logic level for all dual voltage GPIO pads. If the pads are connected to 3V logic level external peripheral, this poses no issue. However, some systems are designed to operate with external ICs and components at 1.8V logic levels. If the majority of the dual voltage I/Os are connected to ICs powers up to 1.8V logic level operation, following power-up sequence should be considered.

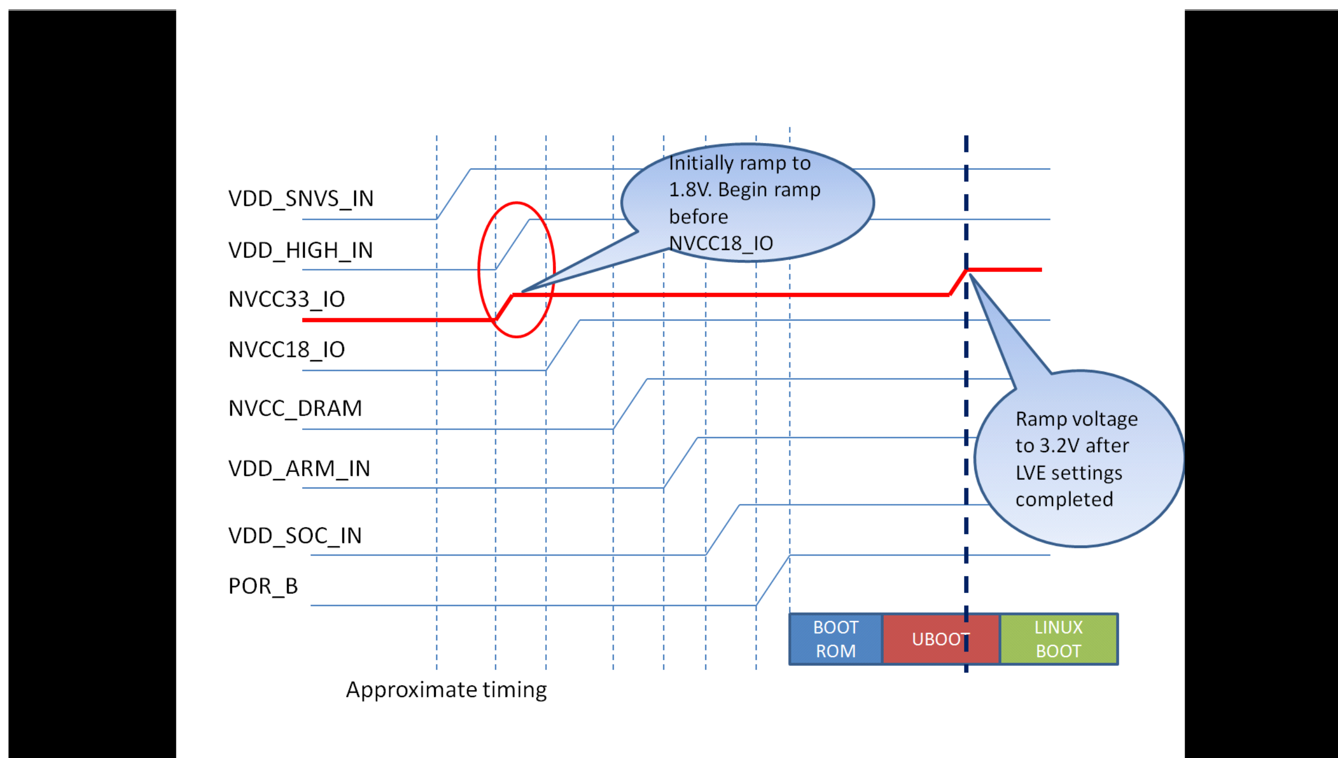


Figure 7-3. Ramp up recommendation for low-voltage design systems (power-up LDO bypass mode)

Step1 – Follow the proper power-up sequence up to NVCC33_IO rail power-up point.

Step2 – Power up NVCC33_IO to 1.8V, instead of 3Vnominal.

Step3 – Complete the rest of the power-up sequence.

Step4 – Set up each of the dual voltage pad's operating voltage to NVCC18_IO by setting LVE bit located in corresponding IOMUX control register.

Step5 – Change bump NVCC33_IO's voltage to nominal 3V.

Above step will require additional steps, but will avoid possible electrical issues to 1.8V logic level lines or having to add external glue logic which could be cost prohibitive.

CAUTION

When ramping NVCC33_IO to 1.8V, make sure to have NVCC33_IO is powered up **before** NVCC18_IO. Also, NVCC33_IO must be ramped up **after** VDD_HIGH_IN is powered up. If these sequences were not followed, it may result in unreliable boot or possibly cause IC failure.

7.3 External 32 kHz clock requirements

If an external 32kHz crystal output is supplied to the processor, it should be connected to the RTC_XTALI pin. The RTC_XTALO pin should remain floating. Figure 7-4 is an example circuit.

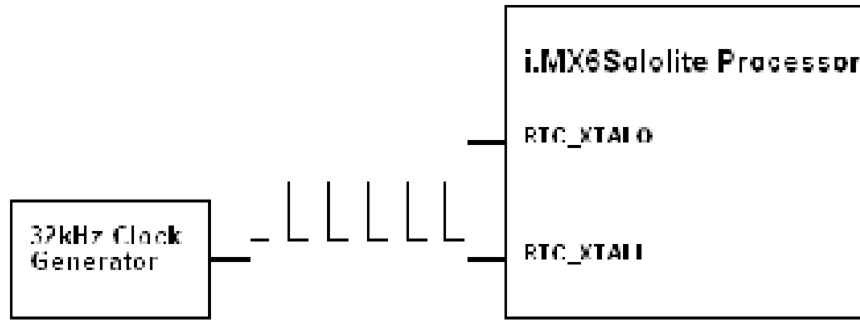


Figure 7-4. External 32k clock connection diagram

Table 7-5 provides a list of specifications for the 32kHz input clock and RTC_XTALI.

Table 7-5. RTC_XTALI and 32kHz clock specifications

Parameter Description	Min	Typ	Max	Units
RTC_XTALI input source/sink current	-	-	1	µA
RTC_XTALI Input Voltage	0	-	VDD_SNVS_CAP(1.1V)	V
RTC_XTALI High-level input voltage ($V_{IH,MIN}$)	0.8 x VDD_SNVS_CAP(1.1V)	-	-	V
RTC_XTALI Low-level input voltage ($V_{IL,MAX}$)	-	-	0.2 V(1.1V)	V
32kHz Clock Nominal Frequency	-	32.768	-	kHz
32kHz Clock Frequency Accuracy[1]	-	10-30	200	ppm

[1] Note: The 32kHz Clock accuracy needed will be dependent upon the overall RTC time keeping accuracy required. A typical 32kHz crystal used for RTC time keeping has an accuracy of around 10-30ppm.

7.4 PMIC Handshaking Signals

The i.MX6Sololite PMIC handshaking signals that will be considered in this section are POR_B, PMIC_STBY_REQ, PMIC_ON_REQ, PMIC_RDY, WDOG_B, and WDOG_RST_B_DEB.

Table 7-6 provides a description of each of the signals:

Table 7-6. PMIC handshaking signal descriptions

Signal Name	Signal Description
POR_B	This signal is an input only pin and is active low. When asserted, it will reset the processor. The processor will remain in reset until the POR_B is de-asserted. The pads for this signal are powered from VDD_SNVS_IN, so VDD_SNVS_IN must be present for this pin to function. **POR_B must be asserted for a least one 32k clock cycle (~30µs) for the processor to qualify it as valid.**
PMIC_STBY_REQ	This output signal is active high. When asserted, the processor is requesting the PMIC to configure the power rails to place the system into a low power standby mode. When de-asserted, the PMIC should exit low power standby mode. The pads for this signal are powered from VDD_SNVS_IN, so VDD_SNVS_IN must be present for this pin to function.
PMIC_ON_REQ	This output signal is active high. When de-asserted, the processor is requesting the PMIC to turn the system off. When asserted, the PMIC should bring the system out of off mode. The pads for this signal are powered from VDD_SNVS_IN, so VDD_SNVS_IN must be enabled for this pin to function.
WDOG_B	This is one of the watchdog output signals from the processor. This signal gets asserted low for either of the following two conditions: 1. Software write to WDA bit of Watchdog Control Register (WDOG_WCR). This signal remains asserted as long as the WDA bit is 0. 2. Watchdog time-out event. WDT bit of Watchdog Control Register (WDOG_WCR) must be set for this scenario. WDOG_B signal remains asserted until a Power-on-Reset (POR) occurs. It gets cleared after the POR occurs and not due to any other system reset.

The POR_B, PMIC_STBY_REQ, and PMIC_ON_REQ pins are powered by the VDD_SNVS_IN domain. The specifications for these signals are shown in Table 7-7.

Table 7-7. VDD_SNVS_IN domain signals

Parameter	POR_B	PMIC_STBY_REQ	PMIC_ON_REQ
Supply Domain Power	VDD_SNVS_IN	VDD_SNVS_IN	VDD_SNVS_IN
I/O Voltage Range (V)	-0.3 to VDD_SNVS_IN + 0.3	-0.3 to VDD_SNVS_IN + 0.3	-0.3 to VDD_SNVS_IN + 0.3
Min. High-level Input Voltage $V_{IH,MIN}$	0.7 x VDD_SNVS_IN	N/A	N/A
Max. Low-level Input Voltage $V_{IL,MAX}$	0.3 x VDD_SNVS_IN	N/A	N/A
High-level Output Driver Impedance	N/A	50-250 ohms (drive strength setting dependent)	50-250 ohms (drive strength setting dependent)
Low-level Output Driver Impedance	N/A	50-250 ohms (drive strength setting dependent)	50-250 ohms (drive strength setting dependent)
Active State	Low	High	High

Table 7-7. VDD_SNVS_IN domain signals (continued)

I/O type	CMOS input	CMOS output	open drain
Internal Pull device	100kohm pull-up	N/A	100kohm pull-up

The PMIC_RDY, WDOG_B, and WDOG_RST_B_DEB pins are programmable and can powered from either the NVCC33_IO or NVCC_LOW domains. They will be powered from the NVCC33_IO domain by default. See Table 7-8.

Table 7-8. NVCC33_IO domain signals NVCC33_IO

Parameter	WDOG_B
Supply Domain Power	NVCC33_IO
I/O Voltage Range (V)	-0.3 to NVCC33_IO + 0.3
Min. High-level Input Voltage $V_{IH,MIN}$	N/A
Max. Low-level Input Voltage $V_{IL,MAX}$	N/A
High-level Output Driver Impedance	50-250 ohms (drive strength setting dependent)
Low-level Output Driver Impedance	50-250 ohms (drive strength setting dependent)
Active State	Low
I/O type	open drain or CMOS output (programmable)
Internal Pull device	Optional 100kohm pull-up

7.5 Watchdog Reset Behavior

When a watchdog time out / reset event occurs and the processor asserts WDOG_B, the PMIC should immediately assert the POR_B signal and execute the power-down sequence. However, it is desirable to keep the VDD_SNVS_IN and 32k external clock powered up and running. This can be useful for the processor to retain its RTC value and also to retain the captured AXI bus traffic in the SNVS domain. When restarting, the POR_B should continue to be asserted until the power-up sequence in figure 2 is completed.

In some cases, it may be desired that only the POR_B be asserted to reset the processor (with all the processor rails remaining powered) when a watchdog reset event occurs. In this case, the POR_B must be asserted for at least one 32k clock cycle (~30µs) for the processor to qualify it as valid.

7.6 Processor Rail Capacitance

Table 7-9 lists the capacitance (high frequency and bulk) required on each processor rail. Ceramic capacitors should be used and should be rated for the maximum voltage specified for each rail.

Table 7-9. Capacitance required for each processor rail

Processor Rail	High Frequency Caps	Recommended Bulk Caps	Comments
VDD_ARM_IN	0.22uF for each pin.	22uF	For normal operation using MX6SoloLite internal LDOs. These caps should be placed close to the processor pins. Values may vary dependent upon the PMIC regulator requirements and PCB impedances.
VDD_ARM_CAP	0.22uF for each pin.	22uF	For normal operation using MX6Sololoite internal LDOs. These caps should be placed close to the processor pins. Values may vary dependent upon the PMIC regulator requirements and PCB impedances. Direct through via to caps.
VDD_SOC_IN	0.22uF for each pin.	66uF	For normal operation using MX6Sololoite internal LDOs. Should be placed close to processor pins. Values may vary dependent upon the PMIC regulator requirements and PCB impedances.
VDD_SOC_CAP	0.22uF for each pin.	22uF	For normal operation using MX6Sololoite internal LDOs. These caps should be placed close to the processor pins. Values may vary dependent upon the PMIC regulator requirements and PCB impedances.
VDD_PU_IN	0.22uF for each pin.	22uF	For normal operation using MX6Sololoite internal LDOs. These caps should be placed close to the processor pins. Values may vary dependent upon the PMIC regulator requirements and PCB impedances.
VDD_PU_CAP	0.22uF for each pin.	22uF	For normal operation using MX6Sololoite internal LDOs. These caps should be placed close to the processor pins. Values may vary dependent upon the PMIC regulator requirements and PCB impedances.
NVCC_DRAM	0.22uF for each pin.	22uF External PMIC and PCB design dependent	The recommended bulk capacitance is shown. However, the recommended value can be reduced dependent upon the external PMIC and PCB design impedances as long as the specification parameters and requirements in table 10-15 are met when measured at the processor pins.
NVCC18_IO	0.22uF for each pin.	10uF || 10uF, External PMIC and PCB design dependent	The recommended bulk capacitance is shown. However, the recommended value can be reduced dependent upon the external PMIC and PCB design impedances as long as the specification parameters and requirements in table 10-15 are met when measured at the processor pins.

Table 7-9. Capacitance required for each processor rail (continued)

NVCC33_IO	0.22uF for each pin.	10uF || 10uF, External PMIC and PCB design dependent	The recommended bulk capacitance is shown. However, the recommended value can be reduced dependent upon the external PMIC and PCB design impedances as long as the specification parameters and requirements in table 10-15 are met when measured at the processor pins
VDD_HIGH_IN	0.22uF for each pin.	10uF, External PMIC and PCB design dependent	The recommended bulk capacitance is shown. However, the recommended value can be reduced dependent upon the external PMIC and PCB design impedances as long as the specification parameters and requirements in table 10-15 are met when measured at the processor pins.
VDD_HIGH_CAP	0.22uF for each pin.	2.2uF(min) - 10uF(max)	Place close to processor pins.
NVCC_PLL	0.22uF (single pin)	2.2uF(min) - 10uF(max)	Place close to processor pins.
VDD_SNVS_IN	0.22uF (single pin)	-	Place close to processor pins.
VDD_SNVS_CAP	0.22uF (single pin)	-	Place close to processor pins.
USB_OTG1_VBUS	0.22uF (single pin)	4.7uF	This pin provides close to the max allowable capacitance according to the USB specification to compensate for any USB cable voltage droop. Place close to processor pins.
USB_OTG2_VBUS	0.22uF (single pin)	4.7uF	This pin provides close to the max allowable capacitance according to the USB specification to compensate for any USB cable voltage droop. Place close to processor pins.
VDD_USB_CAP	0.22uF (single pin)	2.2uF(min) - 10uF(max)	Place close to processor pins.

7.7 Power Modes

Table 7-10 defines the processor power modes and indicates the on/off state of the power supplies and different processor modules in each of the power modes.

Table 7-10. Power Mode Table

Power Supply or Processor Module	Power Mode					Comments
	Active	Idle	Suspend	SNVS[1]	OFF	
POR_B	HIGH	HIGH	HIGH	LOW	LOW	This is the logic state of the POR_B signal.
PMIC_STBY_REQ	LOW	LOW	HIGH	LOW	LOW[2]	
PMIC_ON_REQ	HIGH	HIGH	HIGH	LOW	LOW[3]	
WDOG_B	HIGH	HIGH	HIGH	LOW	LOW[4]	These logic states assume a watchdog event has NOT triggered.

Table 7-10. Power Mode Table (continued)

Power Supply or Processor Module	Power Mode					Comments
	Active	Idle	Suspend	SNVS[1]	OFF	
ARM Core + L1Cache	ON	Power Gated	Power Gated	OFF	OFF	Power Gated = Power is still supplied to VDDARM_IN, but there is no power to VDD_ARM_CAP because the LDO_ARM is off. OFF = There is no power to either VDDARM_IN or VDD_ARM_CAP.
SoC + L2 Cache	ON	ON	Low Voltage	OFF	OFF	Low Voltage = The SoC and L2 cache are provided a low voltage source (~0.9V) from VDDSOC_IN with the LDO_SOC operating in internal bypass mode. OFF = There is no power to either VDDSOC_IN or VDD_SOC_CAP.
GPU2D + OpenVG	ON	Power Gated	Power Gated	OFF	OFF	Power Gated = Power is still supplied to VDDPU_IN, but there is no power to VDD_PU_CAP because the LDO_PU is off. OFF = There is no power to either VDDPU_IN or VDD_PU_CAP.
24MHz XTAL	ON	OFF	OFF	OFF	OFF	
24MHz RC OSC	OFF	ON	OFF	OFF	OFF	
32kHz OSC/Clock	ON	ON	ON	ON	OFF	This is the 32kHz clock either being generated by the processor or being provided to the processor by the PMIC.
PLL	ON	OFF	OFF	OFF	OFF	
USB PHY	ON	OFF	OFF	OFF	OFF	
DRAM I/O Pre-Driver	ON	ON	ON	OFF	OFF	The DRAM I/O Pre-drivers are powered by the DRAM_2P5 input pin. This pin should be connected to the VDD_HIGH_CAP pin of the processor.
LDO ARM	ON	OFF	OFF	OFF	OFF	OFF = Internal ARM LDO turned off and power to the core is gated (has no power).
LDO SOC	ON	ON	BYPASS MODE	OFF	OFF	BYPASS MODE = Internal SoC LDO is turned off, but placed into internal bypass mode. External PMIC low voltage (~0.9V) power is supplied to the SoC domain via the VDDSOC_IN pins.
LDO PU	ON	OFF	OFF	OFF	OFF	OFF = Internal PU LDO turned off and power to the PU domain is gated (has no power).
LDO 1P1	ON	ON	OFF	OFF	OFF	OFF = 1.1V internal LDO turned off
LDO 2P5	ON	ON	MICRO POWER MODE	OFF	OFF	When the LDO_2P5 is placed into “micro power mode”, the main LDO is turned off and a low current bias/regulation circuit is used to supply 2.5V to the DRAM I/O pre-drivers (require about 10uA) from the VDD_HIGH_CAP pin.

Table 7-10. Power Mode Table (continued)

Power Supply or Processor Module	Power Mode					Comments
	Active	Idle	Suspend	SNVS[1]	OFF	
LDO USB (3.0V)	ON	OFF	OFF	OFF	OFF	OFF = Internal USB LDO is turned off
Bandgap	ON	OFF	OFF	OFF	OFF	
Low Power Bandgap	ON	ON	OFF	OFF	OFF	
Charge Pump	OFF	OFF	ON	OFF	OFF	
NVCC HIGH I/O (3.3V)	ON	ON	ON	OFF	OFF	
NVCC LOW I/O (I.8V)	ON	ON	ON	OFF	OFF	
NVCC DRAM I/O LPDDR2 (1.2V)	ON	ON	ON	OFF	OFF	
VDDSNVS_IN / LDO_SNVS	ON	ON	ON	ON	OFF	Power to VDDSNVS_IN is provided either by the PMIC or external coin cell. When VDDSNVS_IN is present (ON), the LDO_SNVS automatically turns on and will provide power to the VDD_SNVS_CAP pin. When OFF, neither the coin cell or PMIC power into the VDDSNVS_IN pin is present.

1 SVNS = Secure Non-volatile State
2 When in the OFF state, this signal will not have any power applied to its pad since it is powered from the VDDSNVS_IN domain, which will be powered off. As a result, this signal should be at 0V. Care should be taken so that the PMIC does not drive or inject current into this signal when the system is placed in the OFF state.
3 When in the OFF state, this signal will not have any power applied to its pad since it is powered from the VDDSNVS_IN domain, which will be powered off. As a result, this signal should be at 0V. Care should be taken so that the PMIC does not drive or inject current into this signal when the system is placed in the OFF state.
4 When in the SNVS and OFF states, this signal will not have any power applied to its pad since it is powered from the VDDHIGH_IN domain, which will be powered off. As a result, this signal should be at 0V. Care should be taken so that the PMIC does not drive or inject current into this signal when the system is placed in the SVNS and OFF states.

7.8 Regulator Requirements

The tables in this section list the requirements for the step-down DCDC converters, general purpose LDOs, and low noise LDOs, and low noise LDOs. The PMIC should meet the specification parameters and requirements for the applicable regulators in this section. While there are no specific requirements for the transient slew rate of the regulators, a 1uA/s should be acceptable.

Table 7-11 lists the key DCDC converter specifications and requirements.

Table 7-11. Step down DCDC converter requirements

Specification Parameter	Recommended Requirement
Operating Input Voltage (V_{IN}).	System/Application input power supply dependent
Output voltage ripple.	10mVpp maximum
Load regulation, 1mA < I_{LOAD} < $I_{LOADMAX}$, for any operational V_{IN}.	10mV maximum
Line regulation, for any I_{LOAD}, 1mA < I_{LOAD} < $I_{LOADMAX}$, for any operational V_{IN}.	10mV maximum
Transient Load Response. V_{OUT} overshoot and undershoot when V_{IN} steps from 10mA to 0.75 x $I_{LOADMAX}$ and back.	20mV maximum, transient response less than 30µs
Transient Line Response. VOUT overshoot and undershoot for a V_{IN} step of 1V up or down within the operating range, for any V_{OUT} and for any I_{LOAD}, 1mA < I_{LOAD} < $I_{LOADMAX}$.	20mV maximum, transient response less than 30µs
Output resistance to GND when OFF.	50-100ohms
Startup / Voltage ramp up time.	Min. ramp rate = 0.1V / µs. Max. total ramp time = 1ms.

Table 7-12 lists the key general purpose LDO specifications and requirements.

Table 7-12. General purpose LDO requirements

Specification Parameter	Recommended Requirement
Operating Input Voltage (V_{IN}).	System/Application input power supply dependent
Output voltage accuracy (including bandgap variation), 1mA < I_{LOAD} < $I_{LOADMAX}$, for any operational V_{IN}.	V_{NOM} +/- 3%
Load regulation, 1mA < I_{LOAD} < $I_{LOADMAX}$, for any operational V_{IN}.	0.25mV / mA maximum
Line regulation, for any I_{LOAD}, 1mA < I_{LOAD} < $I_{LOADMAX}$, for any operational V_{IN}.	10mV maximum
Transient load regulation, V_{OUT} undershoot and overshoot when I_{LOAD} steps from 1 mA to 0.75 x $I_{LOADMAX}$ and back in 5µs (1µs rise and 1µs fall).	20mV maximum
Transient Line regulation, V_{OUT} overshoot and undershoot when V_{IN} steps from V_{OUT}+0.3V to V_{OUT} + 1.0V and back (10µs rise and 10µs fall), for any V_{OUT} and for any I_{LOAD}, 1mA < I_{LOAD} < $I_{LOADMAX}$.	10mV maximum
Output noise, 10Hz - 10MHz	10mV maximum
Startup / Voltage ramp up time	Min. ramp rate = 0.1V / µs. Max. total ramp time = 1ms.

Table 7-13 lists the key low noise LDO specifications and requirements.

Table 7-13. Low-noise LDO requirements

Specification Parameter	Recommended Requirement
Operating Input Voltage (V_{IN}).	System/Application input power supply dependent
Output voltage accuracy (including bandgap variation), 1mA < I_{LOAD} < $I_{LOADMAX}$, for any operational V_{IN}.	V_{NOM} +/- 3%
Load regulation, 1mA < I_{LOAD} < $I_{LOADMAX}$, for any operational V_{IN}.	0.25mV / mA maximum
Line regulation, for any I_{LOAD}, 1mA < I_{LOAD} < $I_{LOADMAX}$, for any operational V_{IN}.	10mV maximum
Transient load regulation, V_{OUT} undershoot and overshoot when I_{LOAD} steps from 1 mA to 0.75 x $I_{LOADMAX}$ and back in 5μs (1μs rise and 1μs fall).	5mV maximum
Transient Line regulation, V_{OUT} overshoot and undershoot when V_{IN} steps from V_{OUT}+0.3V to V_{OUT} + 1.0V and back (10μs rise and 10μs fall), for any V_{OUT} and for any I_{LOAD}, 1mA < I_{LOAD} < $I_{LOADMAX}$.	10mV maximum
Output noise, 10Hz - 10MHz	10mV maximum
Startup / Voltage ramp up time	Min. ramp rate = 0.1V / μs. Max. total ramp time = 1ms.

7.9 Checking for clock pitfalls

Problems with the external clocks are another common source of board bring-up issues. Ensure that all of your clock sources are running as expected.

The XTALI/XTALO and the RTC_XTALI/RTC_XTALO clocks are the main clock sources for 24 MHz and 32 kHz reference clocks respectively on the i.MX6. Although not required, the use of low jitter external oscillators to feed CLK1_P/N or CLK2_P/N on the i.MX6 can be an advantage if low jitter or special frequency clock sources are required by modules driven by CLK1_P/N or CLK2_P/N. See the CCM chapter in your i.MX6 chip reference manual for details. If a 32.768 kHz crystal is not connected to the i.MX6, an on-chip ring oscillator is automatically used for the low-frequency clock source.

When checking crystal frequencies, use an active probe to avoid excessive loading. A parasitic probe typically inhibits the 32.768 kHz and 24 MHz oscillators from starting up. Use the following guidelines:

- RTC_XTALI clock is running at 32.768 kHz (can be generated internally or applied externally).
- XTALI/XTALO is running at 24 MHz (used for the PLL reference).
- CLK1_P/N can be used as oscillator inputs for low jitter special frequency sources.
- CLK1_P/N is optional.

In addition to probing the external input clocks, you can check internal clocks by outputting them at the debug signals CLKO1 and CLKO2 (iomuxed signals). See the CCM chapter in the chip reference manual for more details about which clock sources can be output to those debug signals. JTAG tools (see Chapter 6, “Configuring JTAG Tools”) can be used to configure the necessary registers to do this.

7.10 Avoiding reset pitfalls

Follow these guidelines to ensure that you are booting using the correct boot mode.

- During initial power on while asserting the POR_B reset signal, ensure that 24 MHz clock is active before releasing POR_B.
- Follow the recommended power-up sequence specified in the i.MX6SL data sheet.
- Ensure the POR_B signal remains asserted (low) until all voltage rails associated with bootup are on.

The GPIOs and internal fuses control how the i.MX6SL boots. For a more detailed description about the different boot modes, see the system boot chapter of the chip reference manual.

The following figures show two examples of the power-up sequence.

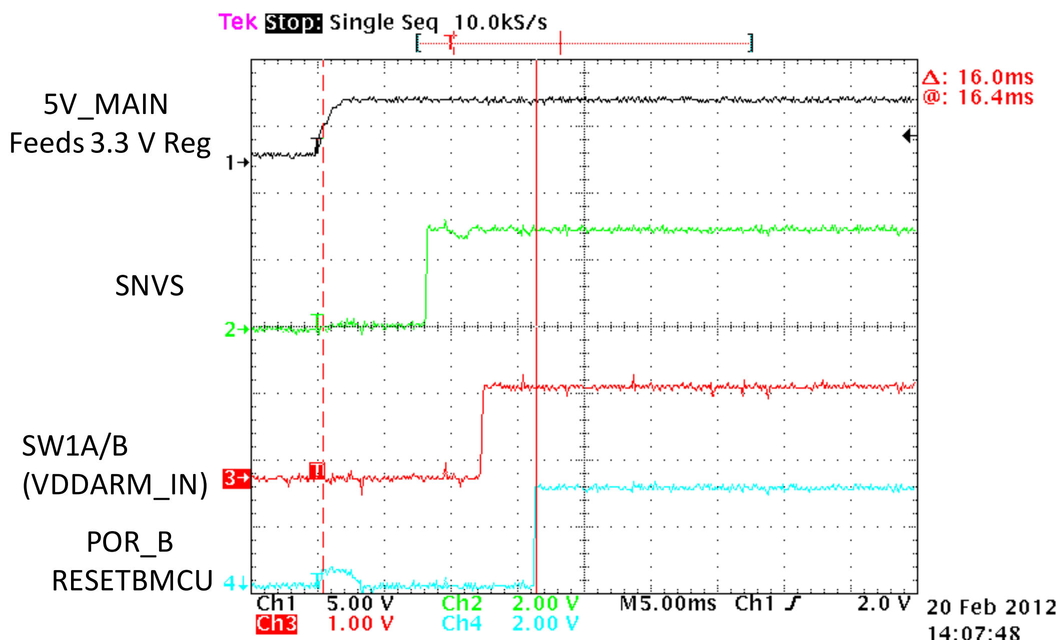


Figure 7-5. Power-up sequence example 1

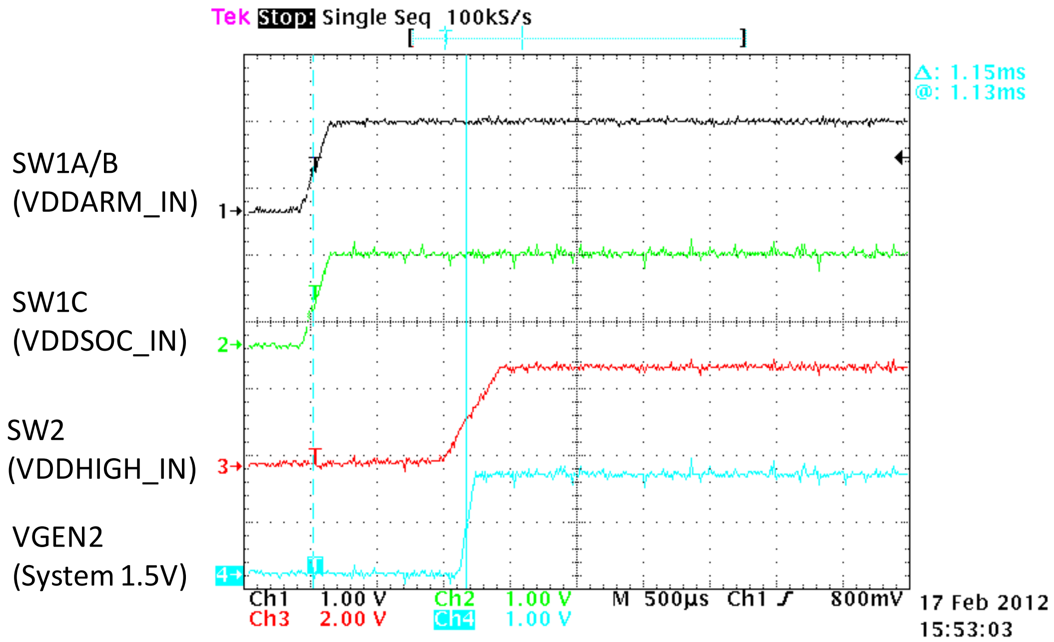


Figure 7-6. Power-up sequence example 2

7.11 Sample board bring-up checklist

Table 7-14 provides a sample board bring-up checklist. Note that the checklist incorporates the recommendations described in the previous sections. Blank cells should be filled in during bring-up as appropriate.

Table 7-14. Board bring-up checklist

Checklist Item	Details	Owner	Findings & status
Note: The following items must be completed serially.			
1. Perform a visual inspection.	Check major components to make sure nothing has been misplaced or rotated before applying power.		
2. Verify all i.MX6 voltage rails.	Confirm that the voltages match the data sheet's requirements. Be sure to check voltages not only at the voltage source, but also as close to the i.MX6 as possible (like on a bypass capacitor). This reveals any IR drops on the board that will cause issues later. **Note:** Ideally all of the i.MX6 voltage rails should be checked, but VDD_ARM_IN and VDD_SOC_IN are particularly important voltages. These are the core logic voltages and must fall within the parameters provided in the i.MX6 data sheet. VDD_SNVS_IN, NVCC_JTAG, and NVCC_DRAM are also critical to the i.MX6 boot up.		

Table 7-14. Board bring-up checklist (continued)

Checklist Item	Details	Owner	Findings & status
3. Verify power-up sequence.	Verify that power on reset (POR_B) is de-asserted (high) after all power rails have come up and are stable. See the i.MX6 data sheet for details about power-up sequencing.		
4. Measure/probe input clocks (32 kHz, 24MHz, others).	Without a properly running clock, the i.MX6 will not function properly. The 24MHz clock must be running before POR_B is released high.		
5. Check JTAG connectivity (RV-ICE).	This is one of the most fundamental and basic access points to the i.MX6 to allow the debug and execution of low level code.		
Note: The following items may be worked on in parallel with other bring up tasks.			
Access internal RAM.	Verify basic operation of the i.MX6 in system. The on-chip internal RAM starts at address 0090_0000h and is 128 Kbytes in density. Perform a basic test by performing a write-read-verify to the internal RAM. No software initialization is necessary to access internal RAM.		
Verify CLKO outputs (measure and verify default clock frequencies for desired clock output options) if the board design supports probing of the CLKO pin. **Note:** Test ports must be provided in order to monitor signals assigned to these pins.	This ensures that the corresponding clock is working and that the PLLs are working. Note that this step requires chip initialization, for example via the JTAG debugger, to properly set up the IOMUX to output CLKO and to set up the clock control module to output the desired clock. See the reference manual for more details.		

Table 7-14. Board bring-up checklist (continued)

Checklist Item	Details	Owner	Findings & status
Measure boot mode frequencies. Set the boot mode switch for each boot mode and measure the following (depending on system availability): • NAND (probe CE to verify boot, measure RE frequency) • SPI-NOR (probe slave select and measure clock frequency) • MMC/SD (measure clock frequency)	This verifies the specified signals' connectivity between the i.MX6 and boot device and that the boot mode signals are properly set. See the "System Boot" chapter in the reference manual for details about configuring the various boot modes.		
Run basic DDR initialization and test memory.	**1.** Assuming the use of a JTAG debugger, run the DDR initialization and open a debugger memory window pointing to the DDR memory map starting address. **2.** Try writing a few words and verify if they can be read correctly. **3.** If not, recheck the DDR initialization sequence and whether the DDR has been correctly soldered onto the board. It is also recommended that users recheck the schematic to ensure that the DDR memory has been connected to the i.MX6 correctly.		

Chapter 8
Understanding the IBIS Model

This chapter explains how to use the IBIS (input output buffer information specification) model, which is an Electronic Industries Alliance standard for the electronic behavioral specifications of integrated circuit input/output analog characteristics. The model is generated in ASCII text format and consists of multiple tables that capture current vs. voltage (IV) and voltage vs. time (VT) characteristics of each buffer. IBIS models are generally used to perform PCB-board-level signal integrity (SI) simulations and timing analyses.

The IBIS model's features are as follows:

- Supports fast chip-package-board simulation, with SPICE-level accuracy and faster than any transistor-level model
- Provides the following for portable model data
  - I/O buffers, series elements, terminators
  - Package RLC parasitics
  - Electrical board description

8.1 IBIS structure and content

An IBIS file contains the data required to model a component's input, output, and I/O buffers behaviorally in ASCII format. The basic IBIS file contains the following data:

- Header information regarding the model file
- Information about the component, the package's electrical characteristics, and the pin-to-buffer model mapping (in other words, which pins are connected to which buffer models)
- The data required to model each unique input, output, and I/O buffer design on the component

IBIS models are component-centric, meaning they allow users to model an entire component rather than only a particular buffer. Therefore, in addition to the electrical characteristics of a component's buffers, an IBIS file includes the component's pin-to-buffer mapping and the electrical parameters of the component's package.

8.2 Header Information

The first section of an IBIS file provides the basic information about the file and its data. The following table explains the header information notation. Example 8-1 shows what header information looks like in an IBIS file.

Table 8-1. Header Information

Keyword	Required	Description
[IBIS Ver]	Yes	Version of IBIS Specification this file uses.
[Comment char]	No	Change the comment character. Defaults to the pipe (|) character
[File Name]	Yes	Name of this file. All file names must be lower case. The file name extension for an IBIS file is .ibs
[File Rev]	Yes	The revision level of this file. The specification contains guidelines for assigning revision levels.
[Date]	No	Date this file was created
[Source]	No	The source of the data in this file. Data is taken from a simulation and validated on the board.
[Notes]	No	Component or file-specific notes.
[Disclaimer]	No	May be legally required
[Copyright]	No	The file’s copyright notice

Example 8-1. Header Information

```
[IBIS Ver]                                              4.2
[Comment Char]                                          |_char
[File Name]                                             mx6sl_416mapbga13x13_consmr.ibs
[File Rev]                                              02
[Date]                                                  Mon Jul 23 12:12:30 2012
[Source]
iolib=cln40lp_io35u50dgo7m5x0y1z_wb_1.2p4+IBIS@1.2p6,
RLC=MB0432_1313_050e_01_16FEB2012
[Notes]                                                 FSL Viper 2012.01.31
```

8.3 Component and pin information

The second section of an IBIS file is where the data book information regarding the component’s pinout, pin-to-buffer mapping, and the package and pin electrical parameters is placed. The following table explains the component and pin information notation, and Example 8-2 shows what it looks like in an IBIS file.

Table 8-2. Component and Pin Information

Keyword	Required	Comment
[Component]	Yes	The name of the component being modeled. Standard practice has been to use the industry standard part designation. Note that IBIS files may contain multiple [Component] descriptions.
[Manufacturer]	Yes	The name of the component manufacturer

Table 8-2. Component and Pin Information (continued)

Keyword	Required	Comment
[Package]	Yes	This keyword contains the range (minimum, typical and maximum values) over which the packages' lead resistance, inductance, and capacitance vary (the R_pkg, L_pkg, and C_pkg parameters).
[Pin]	Yes	This keyword contains the pin-to-buffer mapping information. In addition, the model creator can use this keyword to list the package information: R, L, and C data for each individual pin (R_pin, L_pin, and C_pin parameters).
[Package Model]	No	If the component model includes an external package model (or uses the [Define Package Model] keyword within the IBIS file itself), this keyword indicates the name of that package model.
[Pin Mapping]	No	This keyword is used if the model creator wishes to include information on buffer power and ground connections. This information may be used for simulations involving multiple outputs switching.
[Diff Pin]	No	This keyword is used to associate buffers that should be driven in a complementary fashion as a differential pair.
[Model Selector]		This keyword provides a simple means by which several buffers can be made optionally available for simulation at the same physical pin of the component.

Example 8-2. Component and pin information

```
[Component]          mx6sl_416mapbga13x13
[Manufacturer]       FREESCALE
[Package]
| variable       typ                 min                 max
R_pkg            0.1820888           0.0062144           0.424531
L_pkg            2.33166nH           0.07436nH           4.98212nH
C_pkg            4.85350pF           0.58625pF           52.4327pF
|
[Pin]    signal_name                 model_name             R_pin       L_pin      C_pin
A1       GND                         GND                    NA          NA         NA
A2       DRAM_SDQS3_B                ddr                    0.274507    2.93233nH  0.87287pF
A3       DRAM_D24                    ddr                    0.308072    3.38497nH  0.82353pF
. . .

[Pin Mapping]    pulldown_ref    pullup_ref
A1               GND             NC
A2               GND             NC
A3               GND             NC
. . .
[Diff Pin]      inv_pin        vdiff        tdelay_typ       tdelay_min       tdelay_max
B3              A2             NA           NA               NA               NA
F1              F2             NA           NA               NA               NA
L1              M1             NA           NA               NA               NA
. . .
```

8.4 Model information

The [Model] keyword starts the description of the data for a particular buffer. Table 8-3 shows the main sets of parameters and keywords, composing the model definition.

Table 8-3. Model information

Keyword	Comment
[Model Spec]	General set of parameters for the model simulation.
[Receiver Thresholds]	Threshold information for the different simulation cases.
[Temperature Range]	The temperature range over which the min, typ and max IV and switching data have been gathered.
[Voltage Range]	The range over which Vcc is varied to obtain the min, typ and max pullup and power clamp data.
[Pulldown] [Pullup] [GND_clamp] [POWER_clamp]	IV information. For more details, see Section 8.4.1, " IV information."
[Ramp] [Rising Waveform] [Falling Waveform]	VT information. For more details, see Section 8.4.2, "VT information."
[Test Data] [Rising Waveform Near] [Rising Waveform Far] [Falling Waveform Near] [Falling Waveform Far] [Test Load]	VT golden model information. For more details, see Section 8.4.3, "Golden Model VT information."

8.4.1 IV information

IV information is composed of four Current-over-Voltage tables: [Pullup], [Pulldown], [GND_clamp], and [Power_clamp]. Each look-up table describes a different part of the IO cell model, as shown in Table 8-1.

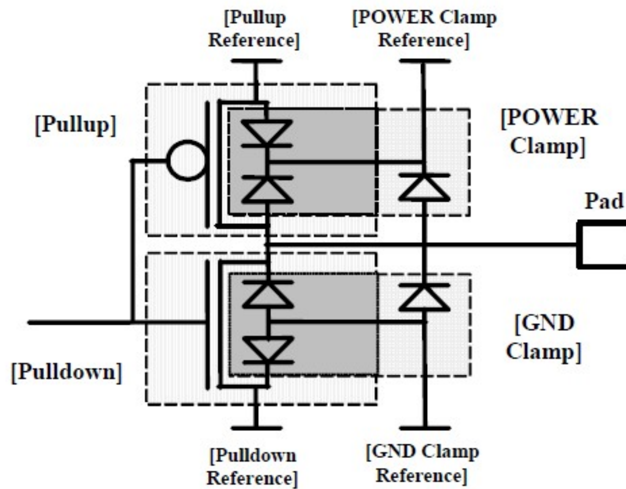


Figure 8-1. Model IV parameters' structure

8.4.2 VT information

The following table defines the keywords that provide the information about an output or I/O buffer, and Example 8-3 shows what they look like in an IBIS file.

Table 8-4. Ramp and waveform keywords

Keyword	Required	Comment
[Ramp]	Yes	Basic ramp rate information, given as a dV/dt_r for rising edges and dV/dt_f for falling edges, see the following equation. $$\frac{dV}{dt} = \frac{20\text{ \% to }80\%\text{ voltage swing}}{\text{time taken to swing above voltage}}$$ **Note:** The dV value is the 20% to 80% voltage swing of the buffer when driving into the specified load, R_load (for [Ramp], this load defaults to 50). For CMOS drivers or I/O buffers, this load is assumed to be connected to the voltages defined by the [Voltage Range] keyword for falling edges and to ground for rising edges.
[Rising Waveform]	No	The actual rising (low to high transition) waveform, provided as a VT table.
[Falling Waveform]	No	The actual falling (high to low transition) waveform, provided as a VT table.

Example 8-3. Ramp and waveform keywords example

```
[Ramp]
| variable       typ                 min                 max
dV/dt_r  0.4627/0.3456n      0.4326/0.4568n      0.4962/0.3030n
dV/dt_f  0.4546/0.3481n      0.4272/0.3918n      0.4774/0.3569n
R_load = 0.2400k
|
[Rising Waveform]
R_fixture= 0.2400k
V_fixture= 0.0
V_fixture_min= 0.0
V_fixture_max= 0.0
|time     V(typ)              V(min)              V(max)
|
|0.0S              0.3369uV          12.4052uV           41.7335nV
|19.7866fS         0.6730uV          12.7375uV           0.3823uV
|20.8863fS         0.6917uV          12.7519uV           0.4013uV
|21.9489fS         0.7058uV          12.7657uV           0.4196uV
...
|
[Falling Waveform]
R_fixture= 0.2400k
V_fixture= 0.0
V_fixture_min= 0.0
V_fixture_max= 0.0
|time     V(typ)              V(min)              V(max)
|
|0.0S     0.7711V            0.7211V             0.8270V
|0.3334nS 0.7711V            0.7211V             0.8270V
|0.3445nS 0.7711V            0.7211V             0.8269V
...
```

The [Ramp] keyword is always required, even if the [Rising Waveform] and [Falling Waveform] keywords are used. However, the VT tables under [Rising Waveform] and [Falling Waveform] are generally preferred to [Ramp] for the following reasons:

- VT data may be provided under a variety of loads and termination voltages
- VT tables may be used to describe transition data for devices as they turn on and turn off.
- [Ramp] effectively averages the transitions of the device, without providing any details on the shapes of the transitions themselves. All detail of the transition ledges would be lost.

The VT data should be included under two [Rising Waveform] and two [Falling Waveform] sections, each containing data tables for a Vcc-connected load and a Ground-connected load (although other loading combinations are permitted).

The most appropriate load is a resistive value corresponding to the impedance of the system transmission lines the buffer will drive (own impedance). For example, a buffer intended for use in a 60 Ω system is best modeled using a 60 Ω load (R_fixture).

The following figure shows how to interpret the model data.

- I_down [GND clamp] + [Power clamp] + [Pulldown]
- I_up [GND clamp] + [Power clamp] + [Pullup]
- I_recvr [GND clamp] + [Power clamp]

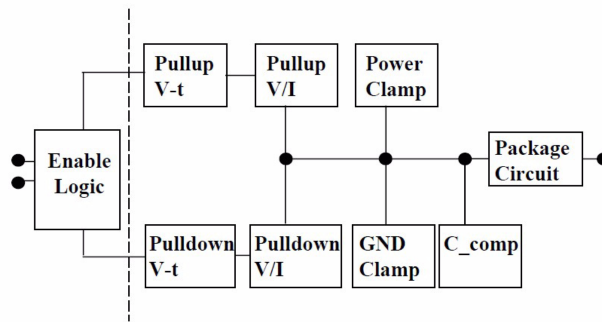


Figure 8-2. Model data interpretation

8.4.3 Golden Model VT information

Golden waveforms are a set of waveforms simulated using known ideal test loads. They are useful for verifying the accuracy of behavioral simulation results against the transistor level circuit model from which the IBIS model parameters originated.

The following figure shows a generic test load network.

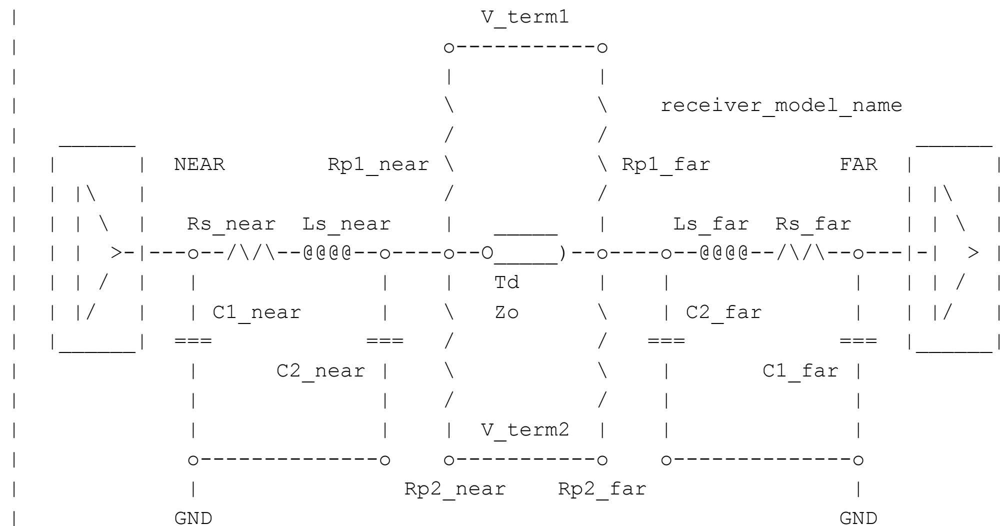


Figure 8-3. Generic test load network

The following table explains the golden waveform keywords.

Table 8-5. Golden waveform keywords

Keyword	Required	Comment
[Test Data]	No	• Provides a set of golden waveforms and references the conditions under which they were derived. • Useful for verifying the accuracy of behavioral simulation results against the transistor level circuit model from which the IBIS model parameters originated.
[Rising Waveform Near] [Rising Waveform Far] [Falling Waveform Near] [Falling Waveform Far]	Yes	Current-Over-Voltage tables, for far and near portions of the golden model as described by Figure 8-3.
[Test Load]	Yes	• Defines a test load network and its associated electrical parameters for reference by golden waveforms under the [Test Data] keyword. • If Test_load_type is Differential, the test load is a pair of the circuits shown in . If the R_diff_near or R_diff_far subparameter is used, a resistor is connected between the near or far nodes of the two circuits. • If Test_load_type is Single_ended, R_diff_near and R_diff_far are ignored.

8.5 Freescale naming conventions for model names and usage in i.MX6 IBIS file

The model names are defined per each [Model selector]. The models may differ from each other by having different parameters—such as voltage, drive strength, mode of operation, and slew rate. The mode of operation, drive strength, and slew rate parameters are programmable by software.

8.5.1 [Model Selector] ddr

The “ddr” model type supports the DDR protocol signals.

8.5.1.1 DDR [Model Selector]

“ddr” models exist for DDR3, DDR3L, DDR3U and LPDDR2 protocols.

This model has the following parameters:

- DDR protocol
- DDR IO type
- Drive strength
- ODT enable/disable

The IBIS model name is composed from the parameters’ values in two ways, as follows:

- Without active ODT circuit:

```
<ddr protocol>_sel<ddr_type>_ds<drive_strength>_mio
```

- With active ODT circuit:

```
<ddr protocol>odt_t<ODT_value>_sel<ddr_type>_mi
```

DDR write models ("_mio" suffix) have no simulated ODT, as ODT is disabled during write. Write models' DS parameter is meaningful and changes to describe the different levels of drive strength.

DDR read models ("_mi" suffix) have no meaningful DS parameter, as no driving happens during read. Read models' ODT parameter is meaningful and changes to describe different levels of ODT impedance.

DDR Protocol	Selected according to the used DDR. DDR IO voltage level is selected accordingly.
DDR IO Type	Controlled by the IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE[19:18] register in IOMUXC (IOMUX controller) DDR_SEL bits, to select between DDR3 & LPDDR2.
Drive strength	Controlled by bits [5:3] (DSE) of the following registers in IOMUXC (IOMUX controller): IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_x (2 registers) IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS IOMUXC_SW_PAD_CTL_PAD_GRP_ADDDS IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKEx (2 registers) IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODTx (2 registers) IOMUXC_SW_PAD_CTL_PAD_GRP_CTLDS IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQSx (8 registers) IOMUXC_SW_PAD_CTL_PAD_DRAM_BxDS (8 registers) IOMUXC_SW_PAD_CTL_PAD_DRAM_DQMx (8 registers)
ODT value	Controlled by bits [18:16], [14:12], [10:8], and [6:4] in MPODTCTRL register of MMDC.

Example 8-4. [Model Selector] DDR in IBIS file

```
ddr3_sel11_ds111_mio                    DDR, 1.5V, ddr3 mode, 34 Ohm driver impedance
...
lpddr2_sel10_ds111_mio                  LPDDR, 1.2V, lpddr2 mode, 34 Ohm driver impedance
lpddr2_sel10_ds110_mio                  LPDDR, 1.2V, lpddr2 mode, 40 Ohm driver impedance
...
```

See the register description in the IOMUXC chapter in the chip reference manual for further details about this model.

8.5.2 [Model Selector] gpio

This model has the following parameters:

- Voltage level
- Drive strength
- Slew rate
- Speed

The IBIS model name is composed from parameters' values as follows:

```
gpio<voltage_level>_ds<drive_strength>_sr<slew_rate(1 bit)><speed(2 bits)>_mio
```

Voltage level — For i.MX 6SoloLite, unlike other i.MX 6 series chips, each dual voltage level GPIO pad can be assigned to high or low voltage via the LVE bit. IBIS users can choose between high and low voltage by selecting a different model at [Model Selector].

Drive strength — Controlled by the DSE bits (bits [5:3]) in the IOMUXC_SW_PAD_CTL_PAD_<pad name>.

Slew rate — Controlled by the SRE bit (bit 0) in the IOMUXC_SW_PAD_CTL_PAD_<pad name>.

Speed — Controlled by the SPEED bits (bits [7:6]) in the IOMUXC_SW_PAD_CTL_PAD_<pad name>.

Example 8-5. [Model Selector] gpio in IBIS file

```
[Model Selector] gpio
gpiohv_ds111_sr111_mio                  GPIO, 1.8V,extra drive,fast sr,max fsel
gpiohv_ds111_sr110_mio                  GPIO, 1.8V,extra drive,fast sr,fast fsel
gpiohv_ds111_sr101_mio                  GPIO, 1.8V,extra drive,fast sr,medium fsel
...
GPIO, 3.3V, extra drive, fast sr,maxfsel
gpiohv_ds111_sr110_mio                  GPIO, 3.3V,extra drive,fast sr,fast fsel
gpiohv_ds111_sr101_mio                  GPIO, 3.3V,extra drive,fast sr,medium fsel
gpiohv_ds111_sr100_mio                  GPIO, 3.3V,extra drive,fast sr,slow fsel
gpiohv_ds111_sr0                        GPIO, 3.3V,extra drive,slow sr,max fsel
...
```

See the register description in the IOMUXC chapter in the chip reference manual for further details about this model.

8.5.3 [Model Selector] USB

At the time of publication, i.MX6 IBIS rev 3 does not contain the USB model. It is expected to be published in a future revision.

8.5.4 List of pins not modeled in the i.MX6 IBIS file

The following table provides a list of analog or special interface pins that are not modeled in the i.MX6 IBIS file.

Table 8-6. i.MX6 pins not supported by IBIS

Domain	Pins
Analog	RTC_XTALI
	RTC_XTALO
	XTALI
	XTALO
	ZQPAD
Analog USB	USB_OTG1_VBUS
	USB_OTG1_DP
	USB_OTG1_DN
	USB_OTG1_CHD_B
	USB_OTG2_VBUS
	USB_OTG2_DP
	USB_OTG2_DN

NOTE

In rev3 of the i.MX6 IBIS, some of the above unsupported pins are described as “GPIO” cells. These are no more than placeholders and cannot be used for signal modeling.

8.6 Quality assurance for the IBIS models

The IBIS models are validated against the IBIS specification, which provides a way to objectively measure the correlation of model simulation results with reference transistor-level spice simulation or measurements.

Correlation — The process of making a quantitative comparison between two sets of I/O buffer characterization data, such as lab measurement vs. structural simulation or behavioral simulation vs. structural simulation.

Correlation Level — A means for categorizing I/O buffer characterization data based on how much the modeling engineer knows about the processing conditions of a sample component and which correlation metric he or she used.

All models (GPIO, DDR, MLB) have passed the following checks:

- IBISCHK without errors or unexplained warnings
- Data for basic simulation checked

- Data for timing analysis checked
- Data for power analysis checked
- Correlated against Spice simulations

Validation reports can be provided upon demand.

8.7 IBIS usage

Freescale board designers used the i.MX6SL IBIS model with the Hyperlynx tool by Mentor Graphics. The HyperLynx version used was HyperLynx v8.1.1 + Update 3.

Effective board design results achieved after loading:

- i.MX6SL IBIS model.
- Companion IC IBIS models.
- Board model in HyperLynx format.

Board simulations for various GPIO and DDR signals were then run.

8.8 References

Consult the following references for more information about the IBIS model.

- IBIS Open Forum (http://www.eda.org/ibis/)

  The IBIS Open Forum consists of EDA vendors, computer manufacturers, semiconductor vendors, universities, and end-users. It proposes updates and reviews, revises standards, and organizes summits. It promotes IBIS models and provides useful documentation and tools.
- IBIS specification (http://eda.org/pub/ibis/ver5.0/)

Chapter 9
Using the Manufacturing Tool

9.1 Overview

The i.MX manufacturing tool is designed to program firmware onto storage devices such as NAND or eMMC through the EVK and preload the data area with media files in an efficient and convenient manner. It is intended for Freescale Semiconductor customers or their OEMs who plan to mass manufacture i.MX-based products.

The application is not designed to test the devices or to diagnose manufacturing problems. Devices initialized with this application still need to be functionally verified.

9.2 Feature summary

The tool includes the following features:

- Continuous operation—operations automatically begin with the connection of a new device, and multiple operations such as update and copy can be linked together seamlessly.
- Enumeration—static-ID firmware loaded into RAM in recovery-mode prevents Windows® from enumerating every device.
- AutoPlay—various Windows® 'pop-up' application and status messages, such as Explorer in Windows® XP and Windows 7.

In addition, the following characteristics improve the tool's ease of use:

- An independent process bar is set up for each physical USB port.
- The tool begins processing with the connection of the first device detected and allows users to replace each device after completion instead of needing to wait for all devices to complete.
- The tool uses color-based indicators to indicate the work status on each of the ports.
  - Blue indicates the device is being processed.
  - Green indicates the device was successfully processed and that the programmed device can be replaced with a new one independent of the of the device's progress.
  - Red indicates the device failed to process.

9.3 Other references

For more detailed information about the manufacturing tool, see the following documents included in the manufacturing tool release package. Contact your local Freescale sales office for assistance obtaining documents if needed:

- For detailed information about how to use the manufacturing tool, see *Manufacturing Tool V2 Quick Start Guide*.
- For detailed information about how to script the processing operations of the manufacturing tool, see the *Manufacturing Tool V2 UCL User Manual*.
- For information about how to generate the manufacturing tool firmware for Linux and Android, see *Manufacturing Tool V2 Linux or Android Firmware Development Guide*.
- For the change list and known issues, see *Manufacturing Tool V2 Release Notes*.

Chapter 10
Using BSDL for Board-level Testing

10.1 BSDL overview

Boundary scan description language (BSDL) is used for board-level testing after components have been assembled. The interface for this test uses the JTAG pins. The definition is contained within IEEE Std 1149.1.

10.2 How BSDL functions

A BSDL file defines the internal scan chain, which is the serial linkage of the IO cells, within a particular device. The scan chain looks like a large shift register, which provides a means to read the logic level applied to a pin or to output a logic state on that pin. Using JTAG commands, a test tool uses the BSDL file to control the scan chain so that device-board connectivity can be tested.

For example, when using an external ROM test interface, the test tool would do the following:

1. Output a specific set of addresses and controls to pins connected to the ROM
2. Perform a read command and scan out the values of the ROM data pins.
3. Compare the values read with the known golden values.

Based on this procedure, the tool can determine whether the interface between the two parts is connected properly and does not contain shorts or opens.

10.3 Downloading the BSDL file

The BSDL file for each i.MX processor is stored on the Freescale website upon product release. Contact your local sales office or fields applications engineer to check the availability of information prior to product releases.

10.4 Pin coverage of BSDL

Each pin is defined as a port within the BSDL file. You can open the file with a text editor (like Wordpad) to review how each pin will function. The BSDL file defines these functions as shown:

```
-- PORT DESCRIPTION TERMS
-- in      = input only
-- out     = three-state output (0, Z, 1)
-- buffer  = two-state output (0, 1)
-- inout   = bidirectional
-- linkage = OTHER (vdd, vss, analog)
```

The appearance of ”linkage” in a pin’s file implies that the pin cannot be used with boundary scan. These are usually power pins or analog pins that cannot be defined with a digital logic state.

10.5 Boundary scan operation

The boundary scan operation is controlled by:

- BOOT_MODE0, BOOT_MODE1, and JTAG_MOD pins
- On-chip Fuse bits

The JTAG_MOD pin state controls the selection of JTAG to the core logic or boundary scan operation. See the following references for further information:

- The “System JTAG Controller (SJC)” chapter in the chip reference manual for the definitions of the JTAG interface operations.
- The “JTAG Security Modes” section in the same chapter for an explanation of the operation of the e-Fuse bit definitions in the following table.
- The “Fusemap” chapter in the chip reference manual the fusemap tables.

Table 10-1. System considerations for BSDL

Pin name	Logic state	Description
JTAG_MOD	1	IEEE 1149.1 JTAG compliant mode
BOOT_MODE[1:0]	[0:0] [0:1] [1:0]	Boot From Fuses Serial Downloader Internal Boot (Development)
POR_B	1	Power On Reset for the device
e-Fuse bits		
JTAG_SMODE[1:0]	[0:0] [0:1]	JTAG enable mode Secure JTAG mode
SJC_DISABLE	0	Secure JTAG Controller is enabled

10.6 I/O pin power considerations

The boundary scan operation uses each of the available device pins to drive or read values within a given system. Therefore, the power supply pin for each specific module must be powered in order for the IO buffers to operate. This is straightforward for the digital pins within the system.

NOTE

BSDL was only tested at 1.8 V.

Chapter 11
Using the FEC Interface

11.1 Overview

This chapter provides supporting instructions for the use of the i.MX 6 series SLFast Ethernet Controller (FEC) interface.

NOTE

This chapter only covers the required hardware and register settings. Modifications to the Ethernet driver or its initialization code are beyond its scope. For this information, see your BSP documentation.

11.2 Configuring the FEC signal connections

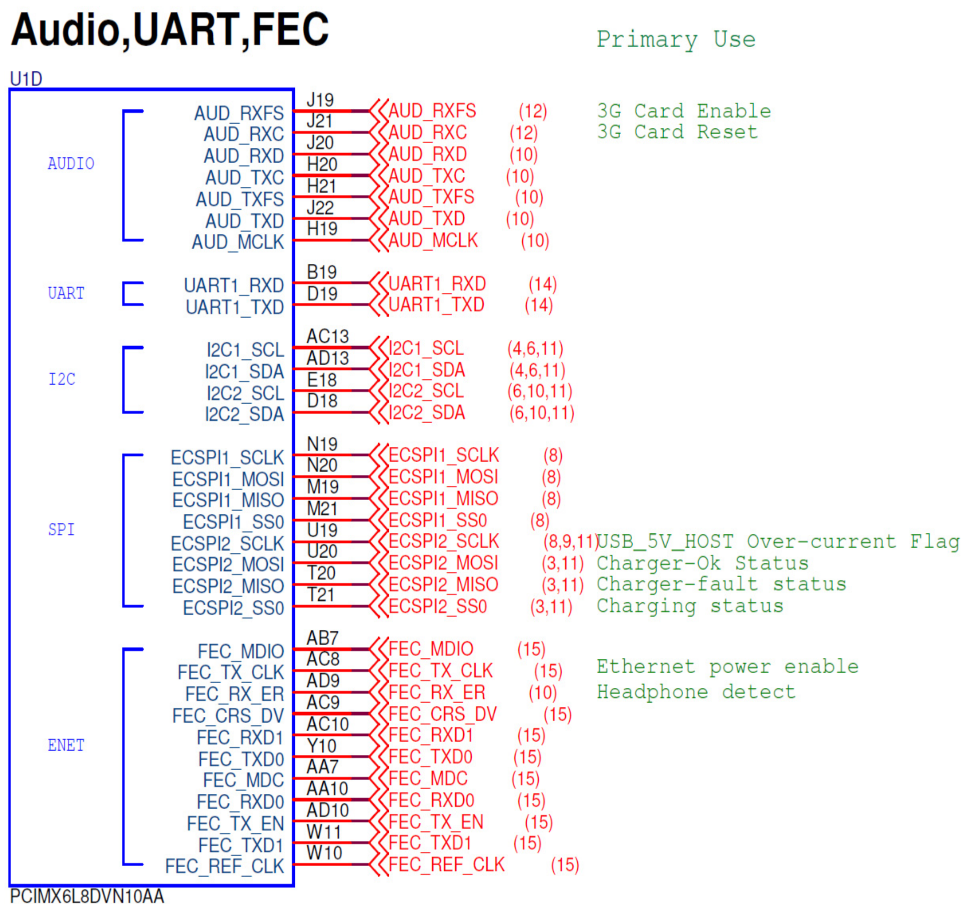


Figure 11-1. Reference schematic, part 1 of 2

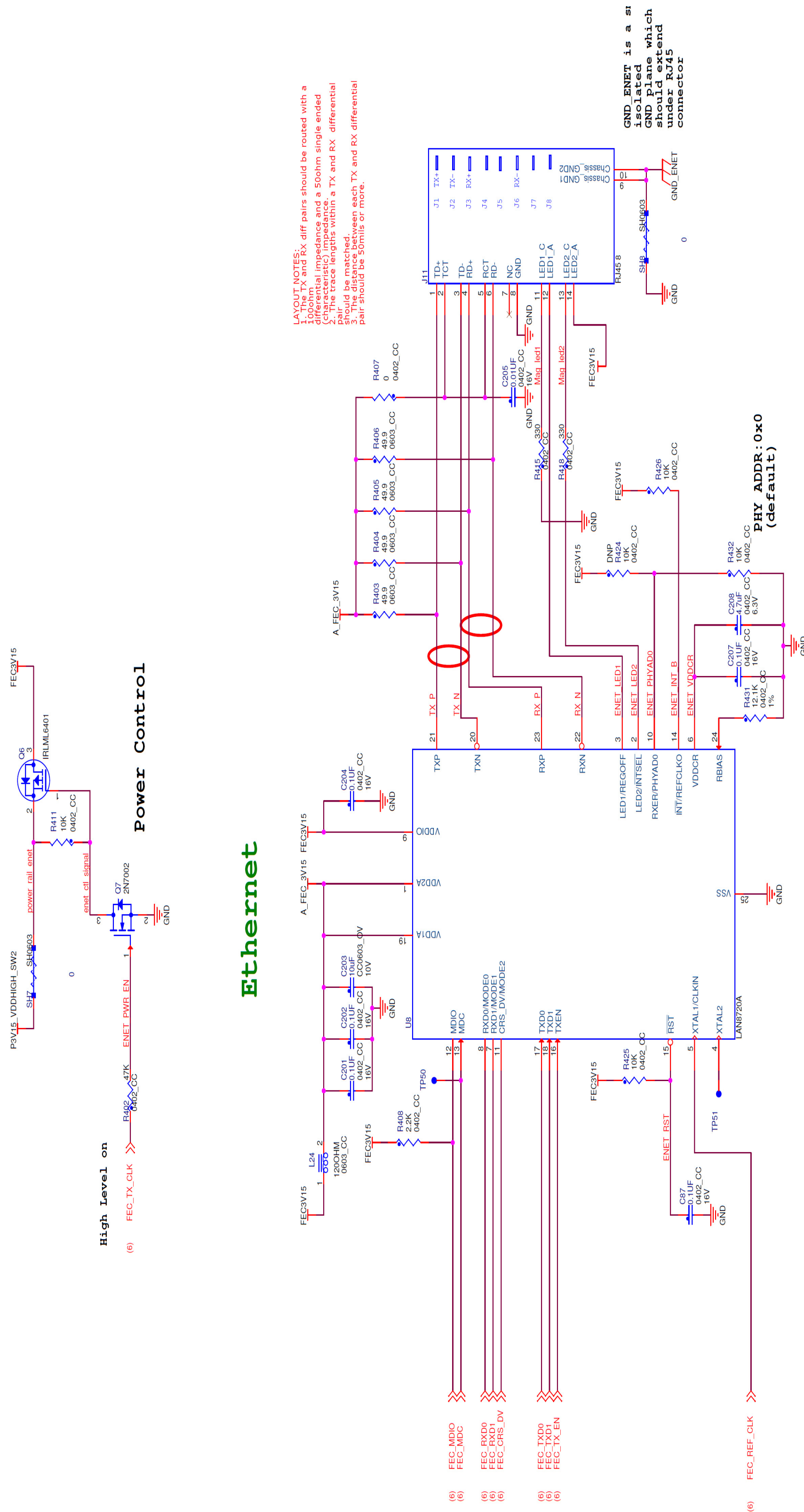


Figure 11-2. Reference schematic, part 2 of 2

Figure 11-2 provides a reference schematic, which shows the connections required to use the FEC interface. These signal connections are generally self-explanatory or explained in the chip reference manual. However, there are some required modifications.

11.3 Generating the reference clock

The Ethernet MAC needs to have a reference clock, which can be generated in one of the following three ways:

- On chip clock generator
- By an external oscillator
- By the FEC PHY

11.4 Generating the reference clock on chip

The reference clock can be generated internally and output to the PHY on the FEC_REF_CLK pin. In this case, it can also be fed back to the FEC_TX_CLK by clearing the GPR1[14] (ENET_CLK_SEL) bit.

Figure 11-3. Internal reference clock (ENET_CLK_SEL = 0)

11.5 Using an external clock

The reference clock can be generated externally by the Ethernet PHY or an external oscillator. It can be input to the i.MX6SL through the FEC_TX_CLK pin. The following figures show the possible configurations.

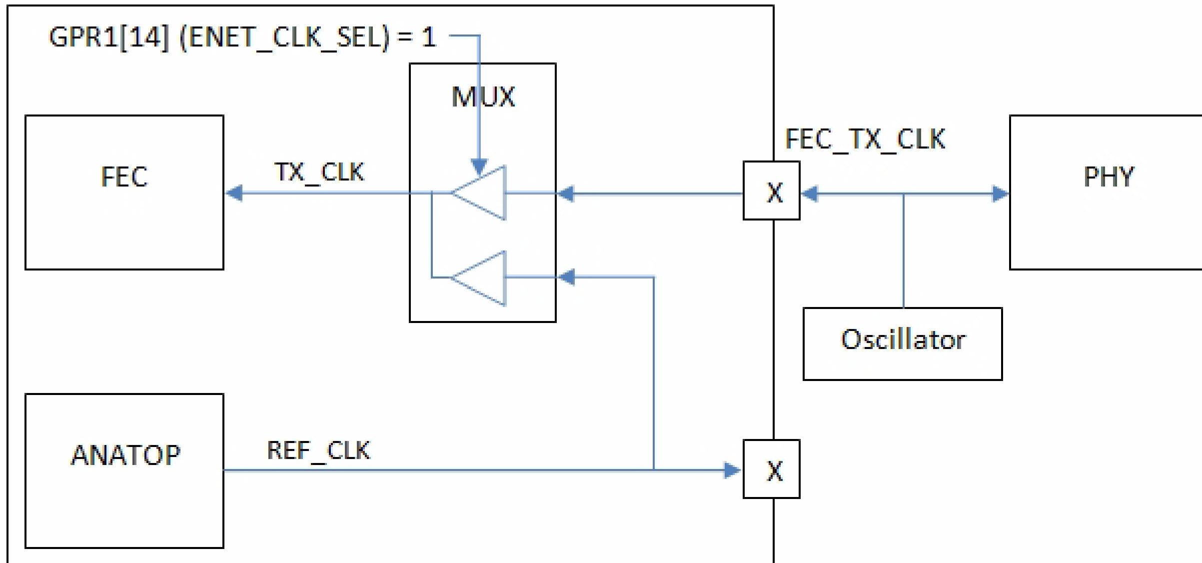


Figure 11-4. External oscillator. Input at FEC_TX_CLK pin

Figure 11-4 shows how to use an external clock. This configuration is almost identical when using an external oscillator or the Ethernet PHY to supply a clock.

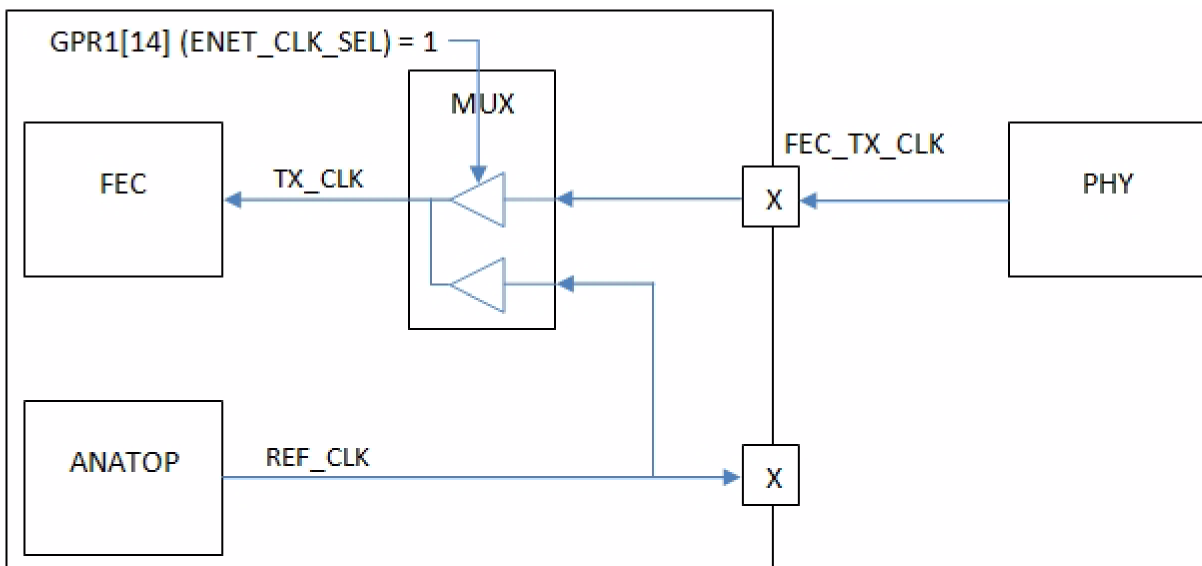


Figure 11-5. Clock supplied by Ethernet PHY. Input at FEC_TX_CLK pin

Appendix A
Development Platforms

This appendix provides a complete list of the development platforms that are available from Freescale to support the i.MX 6SoloLite processor.

You can use these tables as a quick guide for finding the best development platform for your needs. Note that although these development platforms are based on a specific product family, they will work with any of the i.MX product families listed above.

Table A-1. EVK Board for Smart Devices

Version i.MX used	i.MX 6SoloLite
Schematic PN and Rev.	170-27452
Features	• 1 Gbyte LPDDR2 • SPI Nor • eMMC Socket • SD Card Socket • Port of CSI CMOS Sensor (camera) • Parallel Display Port • TouchScreen • Audio CODEC • Ethernet • 3 Axis Accelerometer • Aux SDIO Socket • Mini PCIe (only USB port connection supported) • EPDC support via EPD daughter board • Dual display support via EPD daughter board and LCD daughter board
Quick Start Guide	Available at www.freescale.com/6SLEVK on the Freescale website.
Schematic	Available at www.freescale.com/6SLEVK on the Freescale website.
Layout	Available at www.freescale.com/6SLEVK on the Freescale website.

Appendix B
Revision History

Table B provides a revision history for this document.

Table B-1. Document Revision History

Rev. Number	Date	Substantive Change(s)
1	6/2013	Update to comments in Table 7-3 regarding bypass mode.
0	4/2013	Initial release.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale 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.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 Freescale 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. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. ARM is the registered trademark of ARM Limited.
© 2013 Freescale Semiconductor, Inc.

Document Number: IMX6SLHDG
Rev. 1
06/2013