NAS 782x PCB Design Guidelines

Version 2.00
August 02 2010
GS-0100

Website	www.plxtech.com
Technical Support	www.plxtech.com/support

NDA—Confidential & Proprietary Information

Document number: GS-0100

Contents

Preface v

Typographic Conventions v

Revision Information vi

Glossary vii

Chapter 1 Component Selection

NAND Flash 1

DDR2 1

Chapter 2 Crystal Selection

Crystal Circuit Design 3

Crystal Specification 4

Load Capacitance 4

Chapter 3 Design Considerations

Power Supply 7

Power Supply Sequencing 7

Level Setting Resistors 8

PCI Express 8

Ethernet RGMII Interface 9

Thermal Considerations 10

Reset Guidelines 11

Multifunction I/O 12

Chapter 4 PCB Layout

Stack Up 13

REXT and Crystal I/Os 14

Crystal Layout 14

Differential Pairs 14

SATA Pairs 14

USB Pairs 16

PCIe and REF_CLK Pairs 16

Routing SATA, USB, PCIe and REF_CLK Tracks 16

Power Supply Layout and Decoupling 17

DDR2 Layout Requirements 19

DDR2 VREF 26

Ethernet RGMII Interface Layout 26
When Tracks Change Reference Planes 26

Chapter 5 Fan Control

Using a Thermistor 29
Components Required 29
How the Sensor Works 29
Circuit Description 30
Component Selection 31
Sample Period 31
Software Interface 32
Calculating Temperature From the Reading 32
Using the Fan Tachometer 32
Components Required 32
How the Tachometer Works 33
Software Interface 33
Calculating Fan Speed From the Reading 33

Appendix A Configuration for SATA and USB PHY Compliance

SATA PHY Testing and Configuration 35
USB PHY Testing and Configuration 36

Appendix B Multi-Function Pin Assignments

Standard MF Pin Assignments 39
MF Pin Options 41

This document gives guidelines to get you started with designing the PCB for your product using the following PLX Technology devices:

- NAS 7820
- NAS 7821
- NAS 7825

In the text the devices are jointly referred to as NAS 782x. Information applies to all devices, unless specifically stated otherwise.

For further information about the device, see the appropriate data sheet.

This document assumes that you have a working knowledge of PCB design and you are familiar with the NAS 782x.You must have access to the relevant reference schematics for the device; to obtain the latest version, contact your PLX Technology representative. In addition, for general guidelines on good layout procedures and digital design, we recommend *High Speed Digital Design* (Howard Johnson and Martin Graham [ISBN 0-13-395724-1]).

The information in this document applies whether you are making a dual-SATA or single-SATA board.

Typographic Conventions

In this document, the following conventions apply.

Convention	Meaning
Italic Letters With Initial Capital Letters	A cross-reference to another publication
Title	A cross-reference to another section within the document
1, 2, 3	A numbered list where the order of list items is significant
■	A list where the order of items is not significant
	Significant additional information
`Courier`	Software code
Bold	Significant names, for example of files or directories Text you type

Revision Information

The following table shows the revisions for this document.

Revision	Date	Modification
2.00	August 02 2010	Updated PCI Express
1.00	June 14 2010	New document

Term	Meaning
AC	Alternating Current
APB	ARM Peripheral Bus
DDR2	Double Data Rate
GMII	Gigabit Media Independent Interface
GND	Ground
HCSL	High-speed Current Steering Logic. An interface used for PCIe clock signals
I/O	Input/Output
NAND	A flash memory type using Not AND logic
PCIe	Peripheral Component Interconnect Express
PHY	PHYsical access interface
PLL	Phase-Lock Loop
RDK	Rapid Development Kit
RGMII	Reduced Gigabit Media Independent Interface
SATA	Serial Advanced Technology Attachment

This page is intentionally blank

Chapter 1

Component Selection

This chapter gives guidelines on components to use when designing and building your Printed Circuit Board (PCB).

NAND Flash

The following NAND flash devices are supported:

- Micron MT29F1G08AACWP
- Hynix H27U1G8F2BTR

Other NAND flash devices may work but have not been verified by PLX Technology.

DDR2

The following configurations are supported:

Memory Size MBytes	Memory Organisation	No. of devices	Preference	Notes
64	One rank of 512Mb, x16	1	1	
128	One rank of 1Gb, x16	1	1	
256	One rank of 2Gb, x16	1	1	
	2 x one rank of 1Gb, x8	2	2	Needs series and parallel termination with termination regulator
	Two ranks of 1Gb, x16	2	3	Needs series and parallel termination with termination regulator. Requires modified software: contact PLX Technology Sales Support
512	Two ranks of 2Gb, x16	2	1	Needs series and parallel termination with termination regulator. Requires modified software: contact PLX Technology Sales Support

This page is intentionally blank

Crystal Selection

This chapter gives guidelines for choosing the crystal and designing the circuits around it.

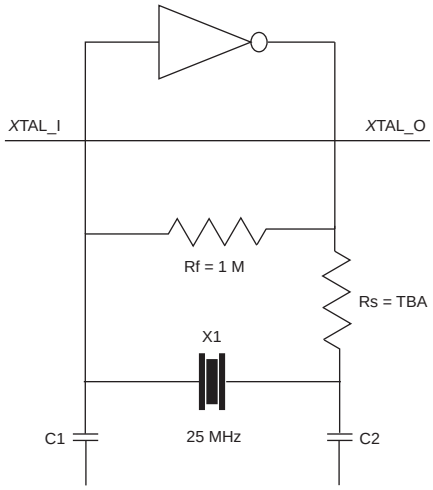
For information on crystal layout, see Crystal Layout on page 14.

The crystal oscillator and the circuit surrounding it are critical factors in the performance of the NAS 782x. The performance of the crystal oscillator circuits affects the integrity of all the system interfaces.

Crystal Circuit Design

Figure 1 shows the circuit that we recommend for the NAS 782x.

Figure 1 Crystal Circuit



where X is a 25-MHz crystal site

The crystal operates in parallel resonant mode.

For reliable crystal start-up for the oscillator, you must have an external feedback resistor (Rf).

Crystal Specification

The following table lists specifications of typical crystals to be used with the NAS 782x.

Parameter	Symbol	Value
Nominal frequency	Fo	25.000000MHz
Oscillation mode		Fundamental
Frequency tolerance (@ 25C)		± 30ppm
Load capacitance	Cl	8pF (max)
Shunt capacitance	Cs	7pF (max)
Effective Series Resistance (ESR)	Resr	≤ 50Ω
Maximum drive level		500µW (min)
Stability over temperature		± 50ppm
Operation temperature range		-10° to +70°
Aging		+/-5ppm/year

Overall frequency tolerance should be < 100ppm.

The oscillator circuit must have the following characteristics:

- Feedback resistor Rf = 1MΩ
- Series resistor Rs =220Ω
- Circuit board ≤ 8pF (= Cl)

Load Capacitance

The load capacitors, shown as C1 and C2 in Figure 1, combine with other circuit capacitance to form the load ‘seen’ by the crystal. The capacitance affects the frequency at which the crystal oscillates. The crystal manufacturer trims the crystal to oscillate at its nominal frequency for the specified load capacitance. Typically a variance of 1pF in the load capacitance adjusts the frequency of the oscillation by 10ppm.

The magnitude of the load capacitance significantly affects the power dissipated in the crystal. For this reason we recommend crystals with Cl ≤ 8pF.

The equation to calculate the values of C1 and C2 is:

$$CL = \frac{C1 \times C2}{C1 + C2} + CS$$

Where:

- $CS = \frac{Cxo \times Cxi}{Cxo + Cxi} + CT$
- Cxo is the device output capacitance, including package parasitics
- Cxi is the device input capacitance, including package parasitics
- CT is the intertrace capacitance between the crystal and device

Where C1 is made the same as C2, this simplifies to:

$C1 = C2 = (CL – CS) \times 2$

Typically, Cxo = Cxi = 4pF; and CT = 1pF, therefore:

$C1 = C2 = (12 – 3) \times 2 = 10pF$

We recommend that you use 5% C0G capacitors for C1 and C2.

As part of the board verification process, we recommend that you measure the frequency of oscillation using a frequency counter with an accuracy of 6 digits or better. If the frequency is too high, add more load capacitance; if too low, decrease the load capacitance.

This page is intentionally blank

Design Considerations

Power Supply

The NAS 782x has a number of different power domains. It is recommended that you filter the following power supplies, as shown in the K149H10 and K149H12 reference schematics:

- XTAL_VDD10
- XTAL_VDD33
- USB_VDD25
- USB_VDD33
- PLLA_VDDA10
- PLLB_VDDA10

The 1.8V, 2.5V and 3.3V supplies are specified with a +/-5% tolerance.

The 1.0V supply is specified as -5% +10%. The -5% is an absolute limit and includes noise and transients on the supply. You must confirm this voltage by measuring the supply voltage with an oscilloscope across the decoupling capacitor that decouples pin M14.

Depending on the regulators used, you may need to use 0.1% tolerance resistors for setting the voltage and to set a mean voltage of 1.025V. To help reduce voltage transients, 5 off 10uF ceramic bulk decoupling capacitors are shown on the reference schematics on the VDD1V0 rail. Place these near the NAS 782x.

Power Supply Sequencing

The only constraint on power supply sequencing is that all the 1.0V supplies must come up together and they must come up last.

There is no power down sequence requirement.

Level Setting Resistors

The USB, SATA, and PCIe interfaces as well as the HCSL buffers have an external current-setting resistor that connects to the appropriate _REXT pin and ground. These resistors must be 1% tolerance and have the following values:

- USBA_REXT—43.2Ω
- USBB_REXT—43.2Ω
- SATA_REXT—191Ω
- PCIE_REXT—191Ω
- HCSL_REXT—475Ω

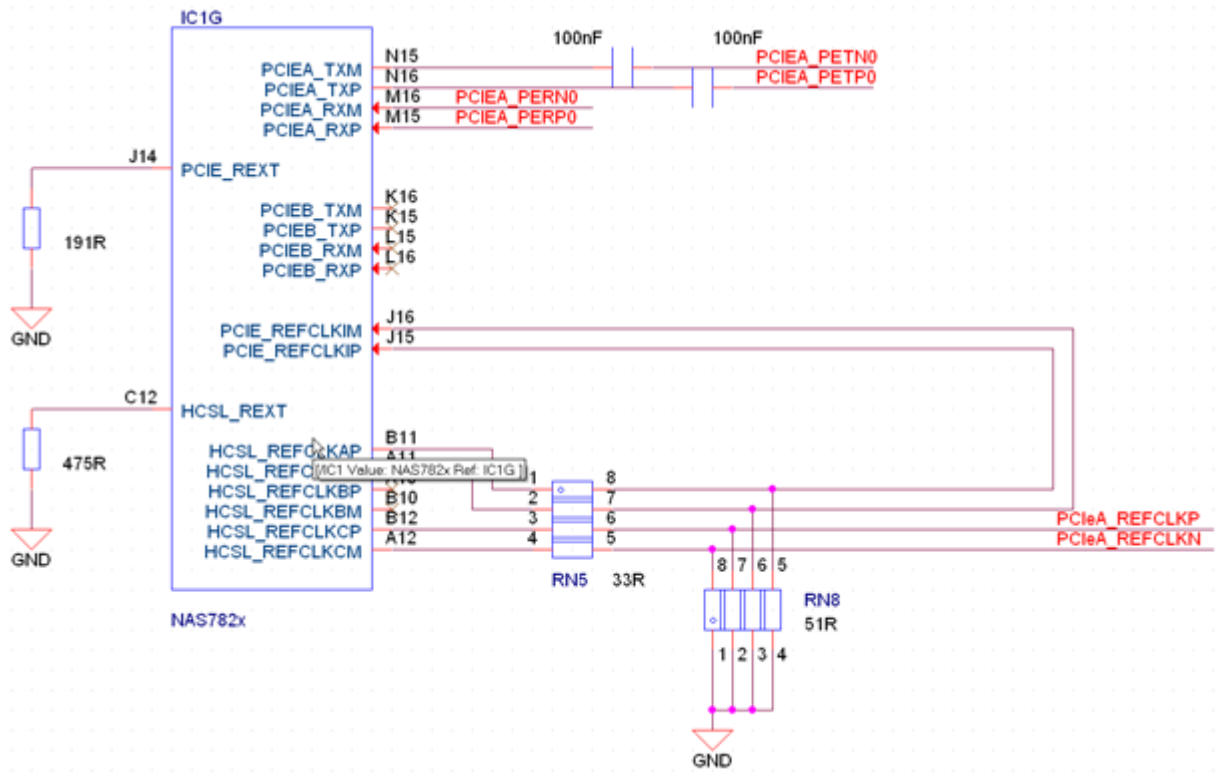
PCI Express

The NAS 7820 and NAS 7821 support a single PCI Express (PCIe) interface. The NAS 7825 supports two PCIe interfaces.

The NAS 782x has built in HCSL buffers to supply the reference clock to the PCIe slot and to the NAS 782x. To be fully compliant for PCIe clocking, you must use an external clock buffer. For details of the internal HCSL buffer, contact your PLX Technology sales representative.

You can use either HCSL_REFCLKA or HCSL_REFCLKC for each reference clock. The HCSL buffers require 33Ω series termination resistors near the NAS 782x, with a 51Ω load immediately adjacent, as shown in Figure 2.

Figure 2 HCSL Buffer Termination Resistors



Do not use the HCSL_REFCLKB differential pair.

To use two PCIe interfaces with a NAS 7825, you must use an external HCSL buffer designed for generating PCIe reference clocks. In this case, you must still fit the HCSL_REXT current setting resistor but you can leave the HCSL_RECLK outputs unconnected.

Each PCIe interface uses three out of band signals: nWAKE, nCLKREQ and nPERST. The nPERST signal is common to both interfaces and is driven from one of the multi function pins as a generic GPIO. The nWAKE, and nCLKREQ signals are monitored by Multi Function pins specific to each of the interfaces. For more information, see Appendix B Standard MF Pin Assignments. These signals must have 5K1 pull up resistors.

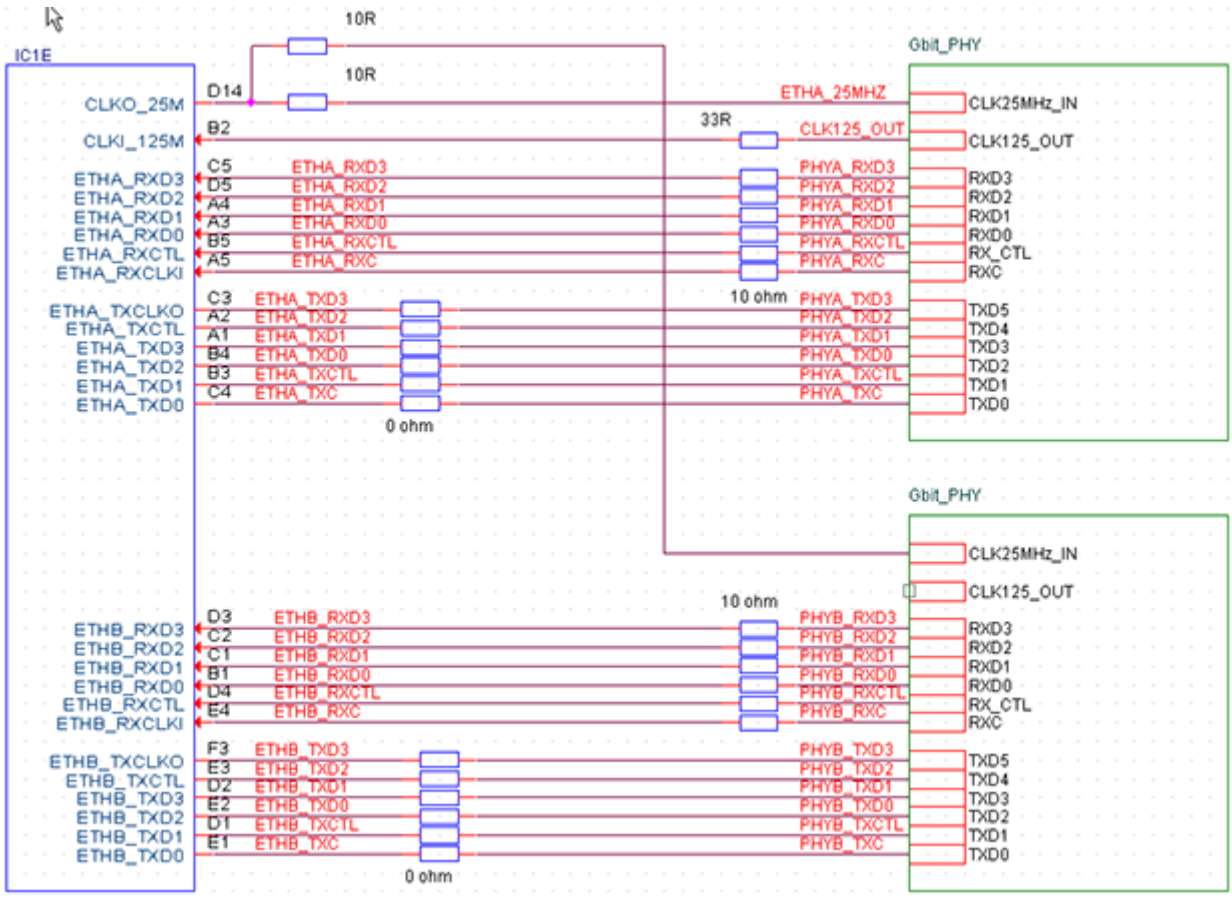
Ethernet RGMII Interface

The RGMII is implemented with 3.3V I/O normally associated with a GMII. Devices that you connect to this interface must be capable of operating their RGMII at this voltage. The ICplus IP1001 has been verified for operation in this mode.

Other Ethernet PHYs may work but have not been verified by PLX Technology.

Figure 3 shows how a NAS 7825 connects to two generic Gbit PHYs. Note that only one of the PHYs provides the 125MHz clock input to the NAS 7825.

Figure 3 NAS 7825 Connected to Generic Gbit PHYs



Refer to the PHY vendor's documentation for information on layout and component selection of magnetics and connectors for the media side of the interface.

For more information on RGMII layout, see Ethernet RGMII Interface Layout on page 26.

Thermal Considerations

There are no special thermal requirements for the NAS 782x. However, we recommend that you take into consideration heat generated by the gigabit PHY and any linear regulators used.

Reset Guidelines

To ensure that all initial conditions are met after power up, the NAS 782x requires a reliable timed reset. If the device comes out of reset too early after power up, it may result in unreliable operation.

A conservatively designed RC circuit should provide a reliable timed reset on power up. The NAS 782x uses a Schmitt input on the reset pin, to assist with generating a reliable reset.

Generally, the greatest timing variance that needs to be accounted for is the start up of the supply voltages. This is completely dependent on the power supply components you use.

Before you negate the reset, you must ensure the following conditions are met:

- All power supplies must be within specification
- The 25MHz oscillator must be stable. This typically occurs within 5ms of the power supplies being stable
- 1ms has elapsed after both conditions above are met

The reference schematics for the NAS 782x show an RC circuit with a time constant of 15 - 20ms. This value assumes that the 3V3 rail has a rise time of <4ms. The reset is measure from when the 3V3 rail crosses 3V, to when the nRESET signal crosses 1.5V.

Note that, because MF_25 is used to allow software to generate a reset, you must take account of the internal 26K-63K pullup.

This is more than sufficient in most applications. However, we strongly recommend that you test the reset on any new design, with realistic power supplies and loads, before committing to production.

Multifunction I/O

The MF_A* and MF_B* pins can be configured for many different input or output functions.

Some I/O functions may not be needed for a particular board design. In order to maximize the usefulness of the limited number of I/O pins, such functions are multiplexed with others onto the MF_A* and MF_B* multifunction pins.

For example, the ‘etha_mdio’ pin function may not be needed if the Ethernet PHY does not need to be configured or if the PHY presents an I^2C interface rather than an MDIO interface. This frees up the MF_A[4] pin to be used for something else.

All MF_A* and MF_B* multifunction pins use 3.3V signal levels.

For a brief summary of which functions are available on particular multifunction pins, see Appendix B Multi-Function Pin Assignments. Note that these pins are not all the same:

- Some multifunction pins have Schmitt input buffers
- Some multifunction pins do not have a programmable internal pull-up
- Some multifunction pins have 8mA output buffers

Some functions can be routed to more than one multifunction pin but some are only available on one multifunction pin. In the latter case, if a function is wanted, other functions must be allocated to other multifunction pins.

Any multifunction pin can be used for General Purpose I/O for buttons, switches and LEDs. Because of this, allocate these functions to particular chip pins after all other required functions are assigned.

Due to the restrictions mentioned here, we recommend that you map particular functions to particular multifunction pins early in the design phase.

This page is intentionally blank

PCB Layout

Printed Circuit board (PCB) layout can affect the performance of the NAS 782x. This chapter outlines the principles for optimizing device performance and gives guidelines for avoiding potential problems.

The following considerations are central to obtaining the best performance from the device:

- Crystal oscillator layout
- Good layout of the high-frequency pairs for the SATA, USB and PCIe ports
- Good layout of the DDR2 interface
- Adequate decoupling
- Keep the tracking for USBA_REXT, USBB_REXT, PCIe_REXT, HCSL_REXT and SATA_REXT pins as short as possible

We recommend that you lay out the high-frequency pairs and their associated connectors first, together with the decoupling capacitors, followed by the DDR2 interface.

Stack Up

We recommend that you use a six layer board for all NAS 782x designs.

For a six layer board, use the following stack:

1 Signal

2 GND

3 Signal

4 Signal

5 Power

6 Signal

For a four layer board, use the following stack:

1 Signal

2 GND

3 Power

4 Signal

REXT and Crystal I/Os

Keep the traces from the various REXT resistors (connected to pins D7, D9, F13, J14 and C12) as short as possible. We recommend that you put relevant resistors on the opposite side of the board to the device and as close to the via associated with the pin as possible. Noise picked up on these nets affects the USB, SATA, PCIe, PCIe_REF and system clocks.

The oscillator pins are sensitive to noise; keep the tracking from them to the crystal and associated components away from all other signals.

Crystal Layout

Consider the crystal and the load capacitors as a unit when laying out the board. Place them as close together as possible so that the loop area of the three components is a minimum. In addition, place the crystal components as close as possible to the XTAL_I and XTAL_O pins to minimize trace lengths.

Differential Pairs

Layout of the USB and SATA differential pairs can have a substantial effect on these interfaces.

SATA Pairs

Lay out SATA pairs as a differential pair, with a 100Ω +/- 15% differential impedance, and a minimum common mode impedance of 40Ω.

There is compensation inside the NAS 782x package for the discrepancy introduced by one half of the differential pair being one row inside the package. Avoid the use of kinks on the tracks.

The SATA TX and RX pairs on the SATAA interface are reversed compared to a standard SATA connector. This means that either one of the pairs must approach the connector from the front, or it must change layers. Changing layer is acceptable for six layer boards where the nets track on layers 1 and 3, either side of a common ground plane. Make the length of trace on layer 3 as small as possible. For a four layer board, tracking to the front is preferred.

Figure 4 illustrates changing layers on a six layer board. Figure 5 illustrates tracking to the front for a four or six layer board.

Figure 4 Tracking SATA Pairs on a Six Layer Board

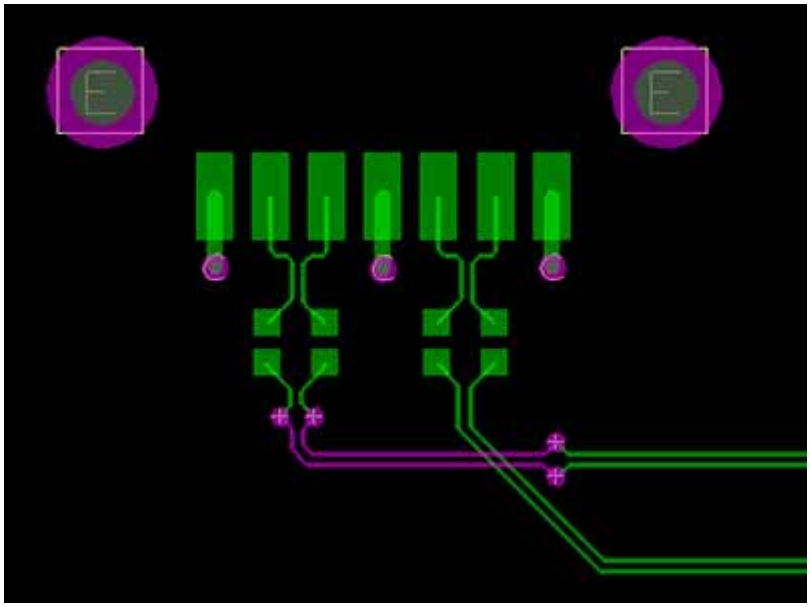
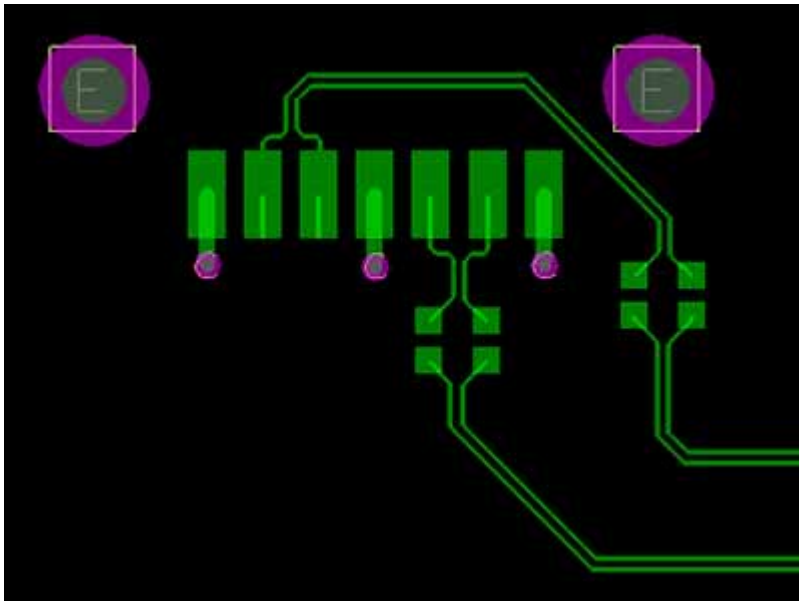


Figure 5 Tracking SATA Pairs on a Four or Six Layer Board



Do not add ‘wiggles’ to compensate for length variations between the two traces in the pair for the differential pair. It is more important to maintain impedance continuity.

USB Pairs

USB pairs require a differential impedance of 90Ω +/- 15% and a common mode impedance of 30Ω +/- 30%.

There is compensation inside the NAS 782x package for the discrepancy introduced by one half of the differential pair being one row inside the package. Avoid using kinks on the tracks.

PCIe and REF_CLK Pairs

Track these signals with a differential impedance of 100Ω +/-15% and with a single-ended impedance of 60Ω +/-15%.

Routing SATA, USB, PCIe and REF_CLK Tracks

The guidelines below, in descending level of importance, summarize best practice routing SATA, USB, PCIe and REF_CLK tracks:

1 Do not route the differential pairs over splits in the reference plane.

2 Avoid vias in the traces, but if vias must be used, you must consider the AC current return path. Failure to do this results in an impedance discontinuity and degraded signal integrity.

3 Keep the traces away from other traces, components and vias, and keep the pairs away from each other—spacing >= 3* the spacing between the two pairs is best, when possible.

Power Supply Layout and Decoupling

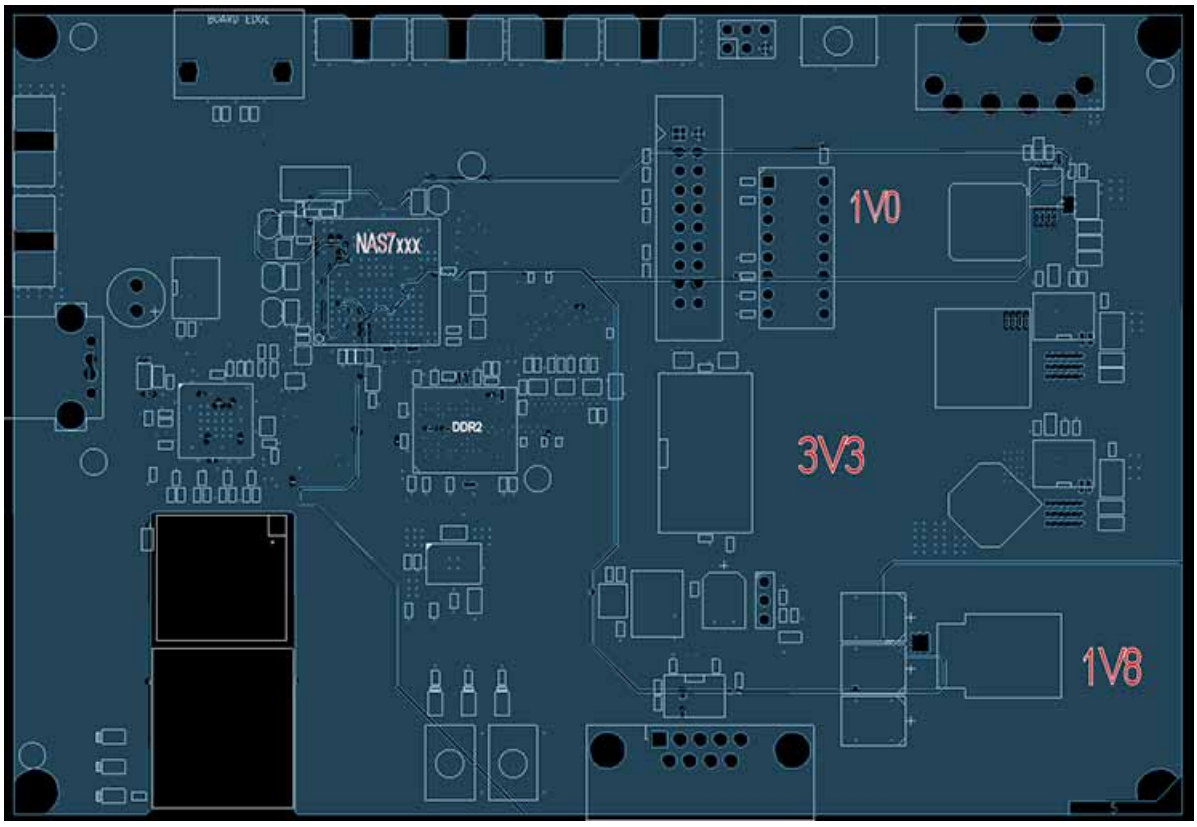
Due to the number of voltage levels needed by a NAS 782x design, you must split the power plane. It is important that no high speed nets cross these splits on the adjacent layers This is especially true for:

- DDR2 interface signals
- All differential pairs
- The ethernet RGMII interfaces

It is important that the 1V0 section of the power plane is of a significant area, to minimise voltage drops across the board.

Figure 6 shows an example power plane partition used on a simple NAS RDK board.

Figure 6 Power Plane



The area of 1V8 plane means that DDR2 signals can be routed on layers 4 and 6 without crossing a split.

The following power supplies have filter components shown in the reference schematics:

- XTAL_VDD10
- XTAL_VDD33
- USB_VDD25
- USB_VDD33
- PLLA_VDDA10
- PLLB_VDDA10

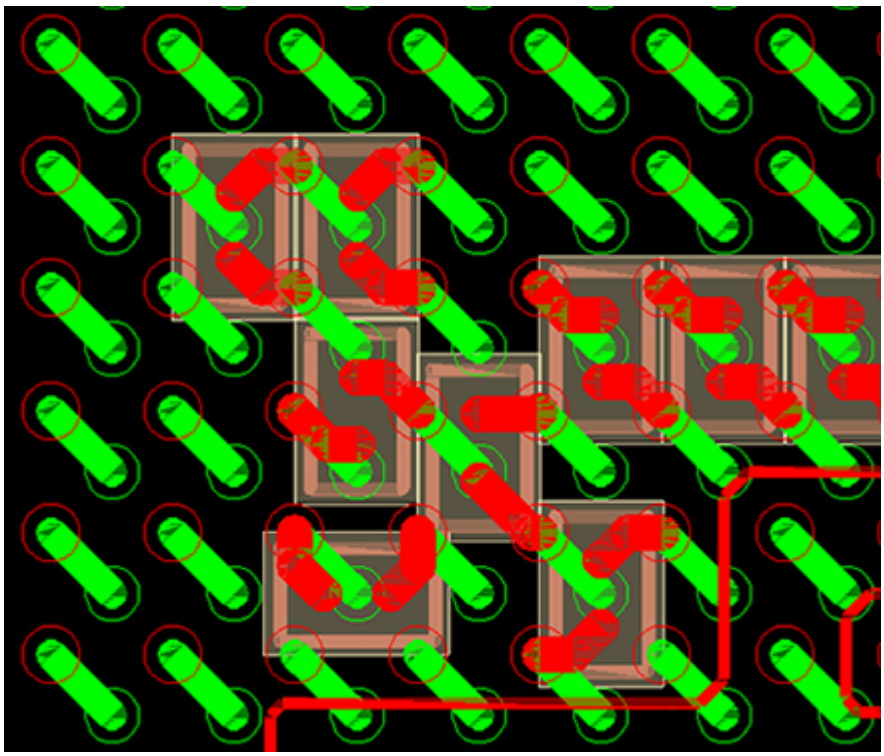
The high frequency decoupling for these pins must be as close to the pins as possible. The bulk decoupling must also be close to the NAS 782x.

It is recommended that you associate a high frequency decoupling capacitor with each power pin on the NAS 782x. This capacitor must be at least 100nF of X5R or X7R dielectric. The package must be 0201 size, as this allows the capacitors to sit within the pin via array.

Place the high frequency decoupling capacitors on the solder side beneath the device and adjacent to the target power pin. This minimises the inductance associated with the capacitor by minimising the loop between the power and ground connections. Wherever possible, avoid sharing power and ground vias between capacitors.

Figure 7 shows some examples of good capacitor placement and tracking.

Figure 7 Capacitor Placement and Tracking



DDR2 Layout Requirements

The NAS 782x can run at full speed with a single DDR2 device fitted. Dual-DDR2 systems can be implemented, but the maximum clock speed may be reduced. For details of supported DRAM configurations, see Chapter 1 Component Selection.

For details of single and dual DDR2 implementations, see reference schematics K149H10 and K149H14.

Implement dual-DDR2 system on a 6 layer board, as described in Stack Up on page 13.

In this description, the DDR signal name follows the name used on the NAS 782x schematic symbol, with the “DDRAM_” removed. The following wording conventions are used:

- **DQ** refers to any of DQ[15:0]
- **DM** refers to either of DM[1:0]
- **DQSP/DQSN** refers to either of the pairs DQSP1/DQSN1 or DQSP1/DQSN1
- **CSN** refers to either of CSN[1:0]
- **ODT** refers to either of ODT[1:0]
- **A** refers to any of A[13:0]
- **BA** refers to any of BA[2:0].

Unless otherwise specified, references to these nets include both sides of any series terminations that may be between the NAS 782x and the DDRAM.

Address nets are the set of signals {A, BA}.

Command nets are the set of signals {CKE, CSN, RASN, CASN, WEN, ODT}.

DDR2 design considerations include:

- A dual-DDR2 system requires six layers to track.
- In dual-DDR2 systems, the two DDR2 parts mount immediately above one another on opposite sides of the board.
- Track all DDR signals over unbroken reference planes. For signals on layers 1 (and 3 for six layer board) this is the GND plane. The plane must be free of slots, intentional or otherwise, where the DDR is tracked. For signals tracked on layers 4 (and 6 for six layer board), the power plane is the reference plane. It is likely that this plane is partitioned to deliver various voltages to different parts of the board. Where layers 4 (and 6 for six layer board) are used to track DDR signals, this power plane must be the 1V8 plane used to power the DDR system. Do not track DDR signals on layers 4 (and 6) closer than 0.3mm to the edge of the 1V8 power island in this region.
- When a DDR signal changes reference plane, ensure a 100nF 0201 decoupling capacitor is placed close to the transition and decouples the 1V8 and GND. When this transition occurs at the BGA escape via of the NAS 782x, this decoupling function is provided by the standard chip 1V8 decoupling capacitors on the 1V8 balls.

- All the signals from the NAS 782x to the DDR2 are 60Ω single-ended impedance and differential signals (CK, CKN, DQSP/DQSN) have a differential impedance of 100Ω.
- All DQ, DM, DQSP/DQSN and CK/CKN signals track on layers 1 (and 3).
- All other signals, command and address, may also track on layers 4 (and 6).
- All address and command nets have a minimum spacing requirement of 0.3mm.
- All DQ and DM signals must be at least 0.5mm away from any other net.
- All differential signals, (CK,CKN, DQSP/DQSN) must be at least 0.5mm away from any other net.
- Net length requirements are defined in Table 1 and Table 2. The net segments TL1 to TLn are defined in Figure 8 to Figure 11 for single-DDR2 systems and Figure 12 to Figure 16 for dual-DDR2 systems.
- The net lengths are used to compensate for variations in package delay. To accomplish this, the parameter TLcomp is defined for every data path signal. This length is subtracted from the net lengths that define TLDu and TLDl. Table 3 provides a list of delays that need compensating for. To arrive at the length that needs to be subtracted, this time should be divided by the delay factor for the individual PCB taking into account the layer on which the traces are tracked and the specific materials from which the PCB is constructed (refer to the PCB vendor for details). The example lengths given are for a PCB with a propagation delay of 6.3pS/mm (160 pS/inch).
- The net DLL_TRACK is TLD_DLL long and is defined in Single-DDR2 System and Dual-DDR2 System.

Single-DDR2 System

Table 1 gives net length requirements for a single-DDR2 system.

Table 1 Single-DDR2 System Constraints			
Condition	Requirement	Length match tolerance(ie delta between track lengths)	Comment
TLcomp			Compensation for internal delays. See Table 3
TL1a	<12mm	5mm	Across all address, command and CK/CKN
TLA = TL1a + TL1b + TR3 (if applicable)	<70mm	0.2mm	Across all address, command and CK/CKN
TL3	<2mm		
TL8	<2mm		
TLDu = TL6 + Tlcomp	<60mm	0.2mm	Across DQ[15:8], DM1 DQSP1/DQSN1
TLDl = TL6 + Tlcomp	<60mm	0.2mm	Across DQ[7:0], DM0 DQSP0/DQSN0
|TLDu - TLDl |	<10mm		The net lengths in upper and lower byte lanes must be within 10mm of each other.
|TLA – TLDu |	<10mm		The length of the address/command bus is within 10mm of both byte lanes.
|TLA – TLDl |			
TLD_DLL = ((TLD(DQS1) +TLD(DQS0))/2 +TLA(CK/CKB)			The average length of the DQS + the length of CK/ CKB

The following figures illustrate the net segments TL1 to TLn for single-DDR2 systems.

Figure 8 Single-DDR2 Address and Command

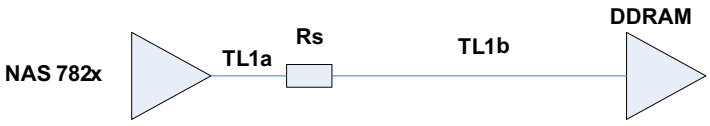


Figure 9 Single-DDR2 CK, CKN Diff Pair

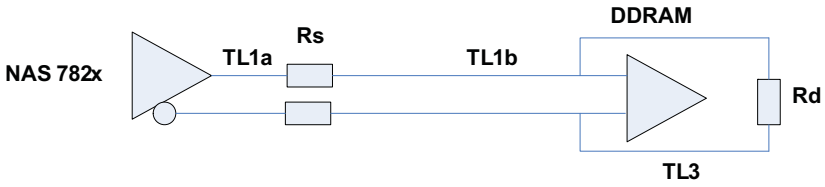
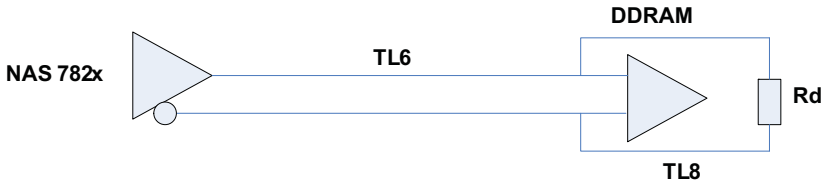


Figure 10 Single-DDR2 DQ and DM

Figure 11 Single-DDR2 DQSP and DQSN



Dual-DDR2 System

Table 2 gives net length requirements for a dual-DDR2 system.

Table 2 Dual-DDR2 System Constraints (Sheet 1 of 2)

Condition	Requirement	Length Match Tolerance	Comment
TLcomp			Compensation for internal delays. See Table 3
TL1a	<10mm	5mm	Across all address, command and CK/CKN
TL1=TL1a + TL1b	TL1		
TL2	<10mm		
TL5	<2mm		
TL3 = TL4	< 15mm	2mm	
TLA = TL3 + TL1	TLA <60mm	0.2mm	Across all address, command and CK/CKN
Alternative values if Figure 16 Topology is used			
TL2	<15mm		
TL5	<3mm		
TL3 = TL4	< 9mm	1mm	
TLA = TL3 + TL1 + TL2	TLA <60mm	0.2mm	Across all address, command and CK/CKN
TL6b	<15mm		
TL6 = TL6a + TL6b			
TL8 = TL7	<3mm	2mm	
TL9 = TL10	<5mm	1mm	
TLDu = TL8 + (TL6 – Tlcomp)	<50mm	0.2mm	Across DQ[15:8], DM1 DQSP1/DQSN1
TLDu = TL9 + (TL6 – Tlcomp)			

Table 2 Dual-DDR2 System Constraints (Sheet 2 of 2)			
Condition	Requirement	Length Match Tolerance	Comment
TLDl = TL8 + (TL6 + Tlcomp)	<50mm	0.2mm	Across DQ[7:0], DM0 DQSP0/DQSN0
TLDl = TL10 + (TL6 + Tlcomp)			
|TLDu = TLDl |	<10mm	10mm	The net lengths in upper and lower byte lanes should be within 10mm of each other
|TLA – TLDu |	<10mm		The length of the address/command bus is within 10mm of both byte lanes
|TLA – TLDl |			
TLD_DLL = ((TLD(DQS1) +TLD(DQS0))/2 +TLA(CK/CKB)			The average length of the DQS + the length of CK/ CKB

The following figures illustrate the net segments TL1 to TLn for dual-DDR2 systems.

Figure 12 Dual-DDR2 Address and Command

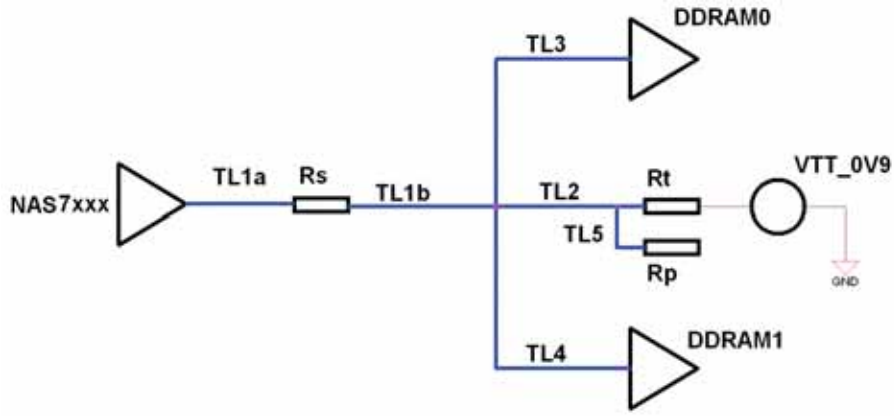


Figure 13 Dual-DDR2 CK and CKN Diff Pair

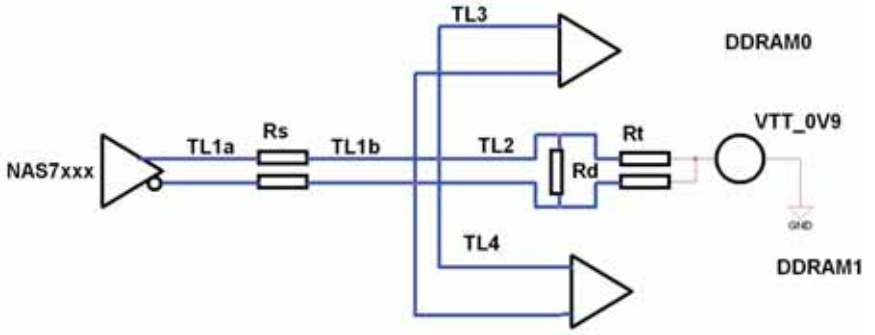


Figure 14 Dual-DDR2 DQ and DM

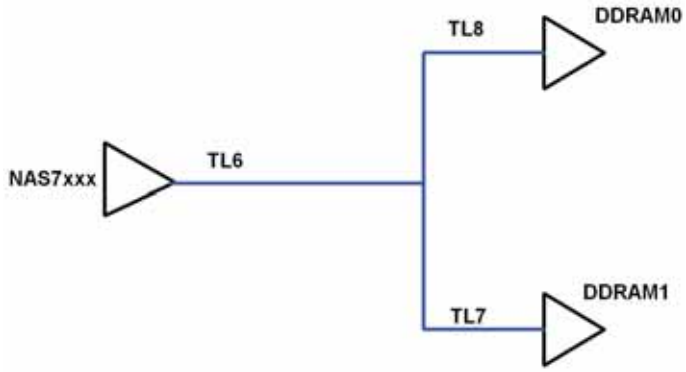


Figure 15 Dual-DDR2 DQSP and DQSN

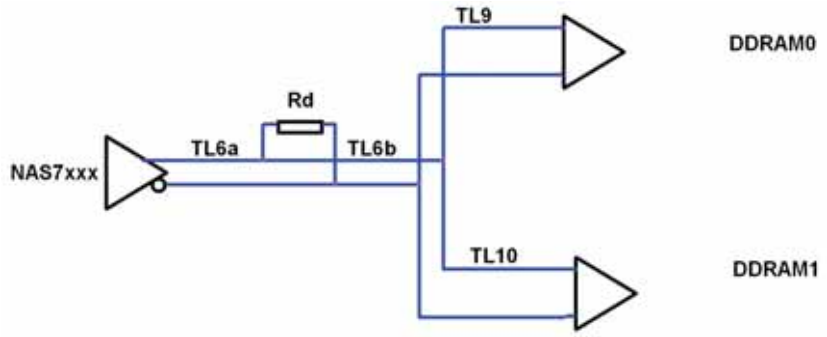


Figure 16 Alternative Dual-DDR2 Address and Command Topology

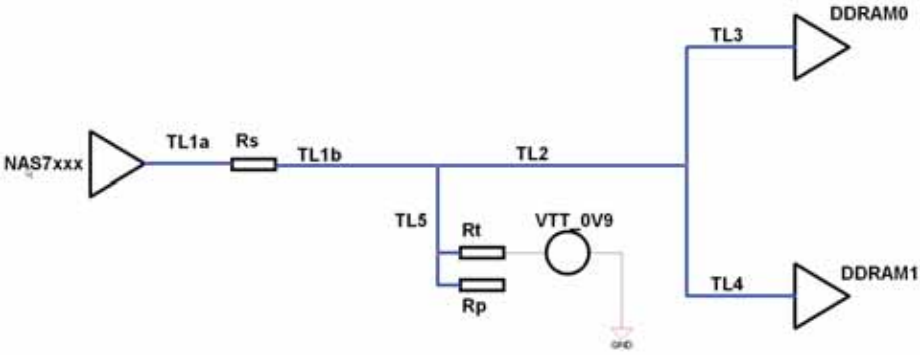


Table 3 shows net lengths used to compensate for variations in package delays.

Table 3 Net Lengths for Variations in Package Delays		
DDRAM Name	Differential compared to shortest delay (pS)	TLcomp Trace length reduction (mm) assuming PCB prop delay = 6.3pS/mm
Lower Data		
DQSN0	17.4	2.8
DQSP0	19.8	3.1
DM0	11.5	1.8
DQ0	4.3	0.7
DQ1	9.7	1.5
DQ2	9.3	1.5
DQ3	17.1	2.7
DQ4	12.4	2.0
DQ5	16.6	2.6
DQ6	2.8	0.4
DQ7	12.7	2.0
Upper Data		
DQSN1	3.0	0.5
DQSP1	1.7	0.3
DM1	15.9	2.5
DQ8	8.2	1.3
DQ9	21.6	3.4
DQ10	4.1	0.7
DQ11	0.0	0.0
DQ12	6.7	1.1
DQ13	13.7	2.2
DQ14	2.8	0.4
DQ15	2.1	0.3

DDR2 VREF

The net DDR2_VREF is a DC net, but is sensitive to noise. Keep it as far away from other nets as possible. For single DDR2 designs the track must be 0.5mm wide (where possible). A decoupling capacitor must be placed close to the DDRAM_VREF pin of the NAS 782x, and another one placed close to the DDRAM_VREF pin of the memory device. A potential divider produces a 0.9V reference voltage from the 1.8V supply.

A bulk decoupling capacitor must be placed close to the resistor potential divider. Use components with a tolerance of 1% for the potential divider.

For dual DDR2 systems a voltage regulator is required to supply the termination resistors. DDR2 specific regulators are available for this purpose and these normally provide a VREF output. All device input VREF pins still require individual decoupling capacitors.

Ethernet RGMII Interface Layout

The RGMII interface is a fast bus operating at 125MHz. For this reason, take care with the routing of the interface. The following constraints apply (where *nn* is any one of D[3:0], CTL or C):

- Track the interface with 50Ω traces
- Make the ETHA_RX*nn* and PHYA_RX*nn* the same length, to within 0.5mm
- Make PHYA_RX*nn* < 10mm
- Make the ETHB_RX*nn* and PHYB_RX*nn* the same length, to within 0.5mm.
- Make PHYB_RX*nn* < 10mm

Net names refer to those used in Figure 12.

Although no maximum length is specified, make these nets as short as possible. Track length in excess of 60mm increases the power dissipation of the interface above the numbers quoted in the data sheet.

Do not allow the nets shown in Figure 12 to cross a split in a power plane. The best way to route these signals is on layers 1 and 3 of a six layer board. If you are using a four layer board and nets have to track on layer 4, ensure the ground and power plane have additional decoupling capacitors near where the signal changes layer. If the transition occurs at a device pin, then the device's own decoupling capacitors are sufficient.

When Tracks Change Reference Planes

When routing high speed digital buses, you may need to use more than one layer. In this case, you must keep the number of layer transitions as low as possible.

When routing the signal, the AC return current path must be considered. Each signal has an associated return current. At higher frequencies, the AC current return path is on the directly adjacent reference plane, because it is the path with the least inductance. If you use a four-layer board, or if you use a power plane as a reference plane, you must consider the case shown in Figure 17.

Figure 17 Current Return Paths

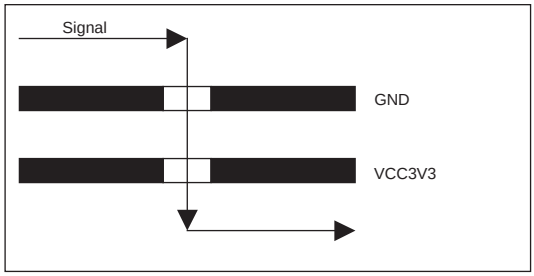


Figure 17 shows the current return path initially on the VCC3V3 plane, but after the via the adjacent reference plane is the GND plane, so a return path must be found between the two reference planes—usually a decoupling capacitor. For this reason, if you must use vias and they result in a change in reference plane, place decoupling capacitors close to the via. Failure to do so increases the loop area of the current return path and introduces an impedance discontinuity, which results in increased emissions and reduced signal quality.

This page is intentionally blank

Fan Control

The NAS 782x can be configured to control fan operation by combining the use of a temperature sensor with a fan tachometer. These features allow an application to:

- Determine the environmental state of an enclosure
- Undertake optimal cooling actions
- Ensure safe device operation

In summary, when the temperature in an enclosure reaches a specified level, an application can vary the fan speed so that an acceptable temperature is maintained.

Using a Thermistor

This section describes using a NAS 782x to control a temperature sensor operating at ±5°C accuracy and free of software timing requirements.

The FAN_TACHO temperature sensor input is multiplexed onto the MF_A0 multifunction pin. You select MF_A0 by enabling SYSCTRL MFIOA_SEC_SEL[0].

Components Required

To implement the sensor device, you need:

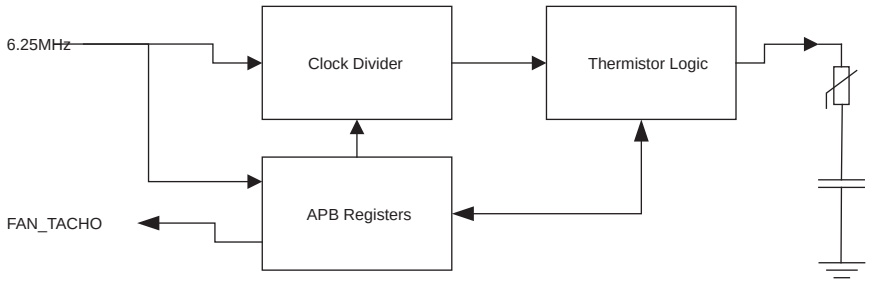
- Thermistor
- Capacitor

How the Sensor Works

The temperature sensor is a basic resistance meter based on an RC time constant of charging a capacitor using a thermistor, as shown in Figure 18 on page 30.

The sensor also reads the junction temperature from the SATA/PCIe PHY.

Figure 18 Sensor Operation



Circuit Description

The circuit used for the sensor is shown in Figure 19.

Figure 19 Sensor Circuit

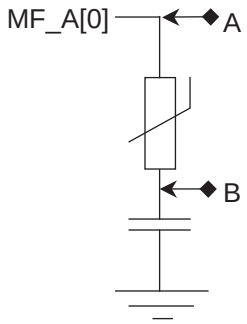
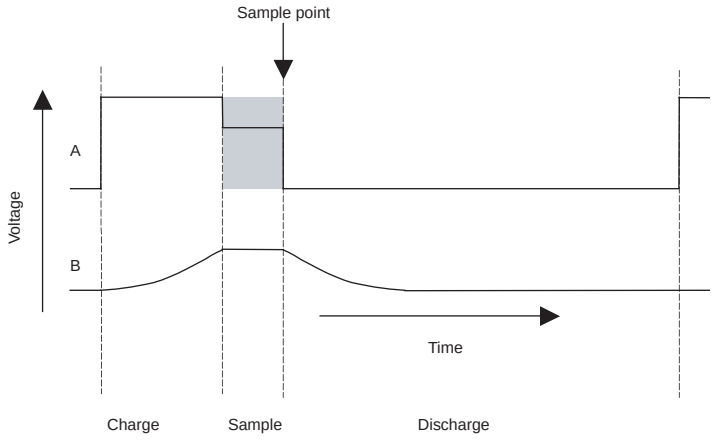


Figure 20 is the voltage diagram associated with Figure 19.

Figure 20 Sensor Voltage Diagram



Operational Cycle

As shown in Figure 19, the NAS 782x is attached using a single GPIO to point A. (Point B is shown to aid understanding of the circuit and is not connected to the NAS 782x).

At time zero the capacitor is fully discharged and voltage B is at zero. The GPIO A is driven as an output, and charges up the capacitor through the thermistor.

The length of time to charge determines the magnitude of the capacitor charge, in conjunction with the resistance of the thermistor. The GPIO A is then tri-stated and, after a small settling time, is read digitally to determine whether the capacitor voltage B is above or below the digital threshold level. This is used to adjust the timing of the next charge cycle (by successive binary approximation).

The GPIO A is then set to drive low and discharge the capacitor for the next cycle.

After 10 repetitions a resistance measurement is obtained. At the end of the measurement the output result is latched and the software notified.

Accuracy

Using 10% tolerance capacitors, accuracy is approximately ±20%, which corresponds to ±5°C.

Component Selection

We recommend components with the following characteristics:

- Capacitor—100nF with best-possible tolerance, such as X7R ±10%
- Thermistor—10KΩ at 25°C with best-possible tolerance, such as SMD310KJ375J

To limit the output current requirements, use a thermistor with a resistance of 5-10KΩ at room temperature.

Sample Period

Ensure clock ticks are made to occur at a rate close to 128KHz. To do so, set the Clock_Divider register to 48 (6.25MHz/128KHz) Each iteration of the measurement takes 4096 clock ticks and ten iterations are required for a full reading, so each reading takes 320ms.

However, when register **Thermistor_Control** bit 0 (**thermistor enable**) is set, the logic runs continuously and sets register bit 1 (**thermistor valid**) to indicate that a reading has been completed.

The clock divider also provides the sample clock to the Fan Tacho block; ensure that the two software modules are aware of each other.

Software Interface

Apart from initially setting the clock divider (register **Clock_Divider** bits 0–9) and enabling the thermistor (register **Thermistor_Control** bit 0), no software control is required. The measurement can be read at any time after **thermistor valid** is set and readings are continually updated while **thermistor enable** is set.

Calculating Temperature From the Reading

The reading provided through register **Thermistor_RC_Counter** is the ten-bit value representing the time it takes to charge the capacitor to the threshold voltage of the GPIO input through the resistance of the thermistor. This value is in counts of the 128KHz tick.

A look-up table in the software is used to convert the count to the temperature in degrees Kelvin. The specific table depends on the thermistor type and the capacitor used.

Using the Fan Tachometer

This section describes using a NAS 782x to control a fan speed counter. The fan tachometer is an APB peripheral that shares logic with the temperature sensor block. Once set up, the assembly is free of software timing requirements.

You can use the FAN_TACHO PWM value to control the fan motor speed. This is available as a secondary function on MF_A[2].

Components Required

To implement the fan speed counter, you need:

- Fan with tachometer
- Circuit to reduce the fan tachometer pin signal to a level suitable for NAS 782x 5V tolerant I/O

A three-wire fan has a square wave output to indicate speed. You must connect this to the NAS 782x and wire it to provide a safe and valid voltage.

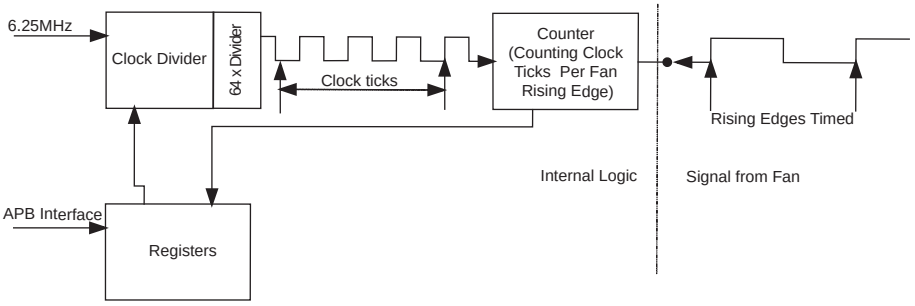
For fans with an open-collector tachometer output, the input pin must be pulled high to 3.3V with a moderate resistance. The capacitance and inductance of the fan cable must be considered, and also the sink capability of the fan output resistor: an 8mA drive can be achieved with a 430Ω resistor.

Some fans output a voltage on the tachometer output. Reduce the voltage to a safe level for the NAS 782x. The GPIOs are 5V tolerant, so any divider that produces that output is acceptable. For example, you can reduce a 12V output to 3.3V with an 8k7 and 3k3 resistor chain.

How the Tachometer Works

The fan speed counter operation is shown in Figure 21.

Figure 21 Fan Speed Counter Operation



The fan outputs a square wave for every revolution. The number of clock ticks per rising edge of the signal are counted and stored in the **Fan_Speed_Counter** register, using the divided-down system clock to provide the ticks. The system clock is divided by a programmable amount in the register **Clock_Divider** using the following algorithm:

$$\text{ClockFreq} = \frac{6.25\text{MHz}}{\text{ClockDivider} + 1}$$

where *ClockDivider* is the value in the register **Clock_Divider**.

For every rising edge coming from the fan tachometer output, the counter is stopped and started, and the value of the counter allows the time between rising edges to be recorded.

The clock divider also provides the sample clock to the temperature sensor block; ensure that the two software modules are aware of each other.

Software Interface

Apart from initially setting the clock divider (register **Clock_Divider** bits 0–9), no software control is required.

Calculating Fan Speed From the Reading

The reading provided through the **Fan_Speed_Counter** register is the ten-bit value representing the time between clock ticks. The following algorithm provides a true fan-speed reading in RPM:

$$\text{Fan Speed (RPM)} = \frac{60 \times \text{PwmClockFreq}}{64 \times (\text{count} + 1)}$$

This page is intentionally blank

Configuration for SATA and USB PHY Compliance

This appendix describes how to test your SATA and USB PHYs for compliance, once you have your PCB designed and available to test.

SATA PHY Testing and Configuration

The test configuration uses an Agilent Digital Signal Analyzer (DSA91204A) which automatically runs a series of conformance tests. The test equipment is approved by the standard compliance tests.

The SATA PHYs are tested in loopback mode.

To configure the device in loopback mode, you need to write to several registers. One way to do this is to use an Abatron bdei JTAG debugger.

The Abatron commands for configuring the SATA port PHYs in loopback mode are:

- For SATA PHY PORT 0:

  mm 0x44e0002c 0x00000010

  mm 0x44e00038 0x00003800

  mm 0x4590070 0x00000301

  mm 0x4590078 0x28

  mm 0x4590070 0x00000300

  mm 0x4590078 0x28

  mm 0x4590070 0x01090000

  mm 0x4590078 0x18

  mm 0x4590070 0x64197720

  mm 0x4590078 0x04

  mm 0x4590070 0x0b04250e

  mm 0x4590078 0x30

- For SATA PHY PORT 1:

  mm 0x44e0002c 0x00000010

  mm 0x44e00038 0x00003800

  mm 0x45910070 0x00000301

  mm 0x45910078 0x28

  mm 0x45910070 0x00000300

  mm 0x45910078 0x28

  mm 0x45910070 0x01090000

  mm 0x45910078 0x18

  mm 0x45910070 0x64197720

  mm 0x45910078 0x04

  mm 0x45910070 0x0b04250e

We recommend you update the following PHY configuration registers for improved compliance results:

- LOS_LVL = 5
- TX_LVL = 2
- TX_ATTEN = 2

This is done by writing the following registers for both PORT 0 and PORT 1:

- mm 0x45900070 0x00002988
- mm 0x45900078 0x60
- mm 0x45900070 0x00055629
- mm 0x45900078 0x70
- mm 0x45910070 0x00055629
- mm 0x45910078 0x70

USB PHY Testing and Configuration

The test configuration uses an Agilent Digital Signal Analyzer (DSA91204A), which automatically runs a series of conformance tests. The test equipment is approved by the standard complianc tests.

The system must output the following test patterns:

- J_state
- K_state
- SE0
- Packet

These test patterns can be configured by writing to registers within the USB core, using the NAS 782x code with a read-modify-write to a single register:

1 Boot the system with the NAS 782x code.

2 Log in into the console.

3 Compile the 'devmem2' program (if not done so previously) using:

gcc –o devmem2 devmem2.c

4 Configure the USB core for the different test configuration using register address 0x40200184 (port A), 0x40200188 (port B) using the command:

./devmem2 0x40200188 w 0x00040000

Use bits [19:16] to set the test mode:

- 0001b J_State
- 0010b K_State
- 0011b SE0
- 0100b Packet (this is used for eye diagrams etc)
- 0101b force_anable_hs
- 0110b force_enable_fs

This page is intentionally blank

Multi-Function Pin Assignments

This appendix describes the standard Multi-Function (MF) pin assignments in the RDK board, and the possible options for MF pin assignments.

Standard MF Pin Assignments

To minimise the changes in software required for various hardware platforms, MF pin usage is defined as shown in the table below. These are not the only usage options, but are a good starting point to maintain compatibility with existing software.

MFIO	NAS 782x RDK Board	NAS 782[0/1] Reference Schematics
MF_A0	Fan thermistor input (fan_temp)	
MF_A1	Fan speed input (fan_tacho)	
MF_A2	Fan PWM output drive (fan_pwm)	
MF_A3	ETHA MDIO clock (etha_mdc)	ETHA MDIO clock (etha_mdc)
MF_A4	ETHA MDIO data (etha_mdio)	ETHA MDIO data (etha_mdio)
MF_A5		
MF_A6		
MF_A7		
MF_A8		
MF_A9	LED2 (gpioa(9))	
MF_A10	USB device on port A VBUS detect (usba_vbus)	USB device on port A VBUS detect (usba_vbus)
MF_A11	BUT2 (gpioa(11))	
MF_A12	NAND FLASH (static_data(0))	NAND FLASH (static_data(0))
MF_A13	NAND FLASH (static_data(1))	NAND FLASH (static_data(1))
MF_A14	NAND FLASH (static_data(2))	NAND FLASH (static_data(2))
MF_A15	NAND FLASH (static_data(3))	NAND FLASH (static_data(3))
MF_A16	NAND FLASH (static_data(4))	NAND FLASH (static_data(4))
MF_A17	NAND FLASH (static_data(5))	NAND FLASH (static_data(5))
MF_A18	NAND FLASH (static_data(6))	NAND FLASH (static_data(6))

MFIO	NAS 782x RDK Board	NAS 782[0/1] Reference Schematics
MF_A19	NAND FLASH (static_data(7))	NAND FLASH (static_data(7))
MF_A20	NAND FLASH (static_we_n)	NAND FLASH (static_we_n)
MF_A21	NAND FLASH (static_oe_n)	NAND FLASH (static_oe_n)
MF_A22	NAND FLASH CS (static_cs_n(0))	NAND FLASH CS (static_cs_n(0))
MF_A23	NAND FLASH (static_addr(18))	NAND FLASH (static_addr(18))
MF_A24	NAND FLASH (static_addr(19))	NAND FLASH (static_addr(19))
MF_A25	GPIO POR (gpioa(25))	GPIO POR (gpioa(25))
MF_A26	Early on LED (gpioa(26))	
MF_A27		
MF_A28		USB port B (host) power enable (usbb_pwro)
MF_A29		USB port B (host) overcurrent input (usbb_overi)
MF_A30	Uart A SIN (uarta_sin)	
MF_A31	Uart A SOUT (uarta_sout)	
MF_B0	JTAG nRST (gpiob(0))	
MF_B1	JTAG TCK (gpiob(1))	
MF_B2	JTAG TMS (gpiob(2))	
MF_B3	JTAG TDI (gpiob(3))	
MF_B4	JTAG TDO (gpiob(4))	
MF_B5	PCIeB nCLKREQ (gpiob(5))	
MF_B6	BUT1 (gpiob(6))	
MF_B7	LED3 (gpiob(7))	
MF_B8	Realtek switch IRQ (gpiob(8))	
MF_B9	I2C DATA (gpiob(9))	
MF_B10	I2C CLK (gpiob(10))	
MF_B11	LED1 (gpiob(11))	
MF_B12	PCIe RST (gpiob(12))	
MF_B13	PCIeB nWAKE (gpiob(13))	
MF_B14	PCIeA nCLKREQ (gpiob(14))	
MF_B15	PCIeA nWAKE (gpiob(15))	
MF_B16		
MF_B17		

MF Pin Options

The following table shows all MF pin options.

A multiplexer multiplexes many functions onto the MF_A[31:0] and MF_B(17:0) pins as determined by register values:

- mf[a|b]_sec_sel
- mfa_ter_sel
- mfa_debug_sel
- mfb_mon_sel

Note the following use rules:

- Secondary function takes precedence over tertiary, debug and monitor functions if they are enabled
- Tertiary function takes precedence over debug and monitor functions if they are enabled
- Debug function takes precedence over monitor function if both are enabled
- Primary (GPIOA/GPIOB) function is selected if no other functions are selected. This is generally the default (defaulting to inputs, hence I/O are not driven)

MFIO	GPIO	Secondary	Tertiary	Quaternary	Debug	Alternative
MF_A0	gpioa(0)	fan_temp	usbb_overi	INVALID	INVALID	INVALID
MF_A1	gpioa(1)	fan_tacho	INVALID	INVALID	INVALID	INVALID
MF_A2	gpioa(2)	fan_pwm	INVALID	INVALID	INVALID	INVALID
MF_A3	gpioa(3)	etha_mdc	sda_dat1	INVALID	INVALID	INVALID
MF_A4	gpioa(4)	etha_mdio	sda_dat2	INVALID	INVALID	INVALID
MF_A5	gpioa(5)	sd_clk	usbb_pwro	vdo_vs_valid	uartb_cts_n	div8_sys
MF_A6	gpioa(6)	sda_cmd	ref300_wck	vdo_blue[0]	uartb_rts_n	div8_pipe
MF_A7	gpioa(7)	sda_dat0	ref300_bck	INVALID	uartb_sin	div4_utmia
MF_A8	gpioa(8)	sda_dat3cd	pllb_wck	INVALID	uartb_sout	div4_utmib
MF_A9	gpioa(9)	pllb_bck	INVALID	vdo_blue[1]	mon_muxa	irrx_in
MF_A10	gpioa(10)	usba_pwro	usba_vbus	vdo_green[1]	mon_muxb	pllb_wck
MF_A11	gpioa(11)	usba_overi	irrx_in	vdo_red[1]	INVALID	pllb_bck
MF_A12	gpioa(12)	static_data(0)	sd_clk	INVALID	mon_muxa	INVALID
MF_A13	gpioa(13)	static_data(1)	sdb_cmd	INVALID	mon_muxb	INVALID
MF_A14	gpioa(14)	static_data(2)	sdb_dat1	INVALID	uarta_sin	INVALID
MF_A15	gpioa(15)	static_data(3)	sdb_dat2	INVALID	uarta_sout	INVALID
MF_A16	gpioa(16)	static_data(4)	sdb_dat0	INVALID	uarta_cts_n	INVALID
MF_A17	gpioa(17)	static_data(5)	sdb_dat3cd	INVALID	uarta_rts_n	INVALID

MFIO	GPIO	Secondary	Tertiary	Quaternary	Debug	Alternative
MF_A18	gpioa(18)	static_data(6)	etha_mdc	INVALID	INVALID	ref300_wck
MF_A19	gpioa(19)	static_data(7)	etha_mdio	INVALID	INVALID	ref300_bck
MF_A20	gpioa(20)	static_we_n	INVALID	INVALID	INVALID	INVALID
MF_A21	gpioa(21)	static_oe_n	INVALID	INVALID	INVALID	INVALID
MF_A22	gpioa(22)	static_cs_n(0)	INVALID	INVALID	INVALID	INVALID
MF_A23	gpioa(23)	static_addr(18)	INVALID	INVALID	INVALID	INVALID
MF_A24	gpioa(24)	static_addr(19)	INVALID	INVALID	uarta_ri_n	INVALID
MF_A25	gpioa(25)	static_cs_n(1)	INVALID	INVALID	uarta_cd_n	INVALID
MF_A26	gpioa(26)	sda_dat1	ethb_mdc	vdo_hs_pix	uarta_dsr_n	INVALID
MF_A27	gpioa(27)	sda_dat2	ethb_mdio	vdo_blue[2]	uarta_dtr_n	INVALID
MF_A28	gpioa(28)	usbb_pwro	INVALID	vdo_blue[3]	INVALID	uarta_cts_n
MF_A29	gpioa(29)	usbb_overi	INVALID	vdo_blue[4]	INVALID	uarta_rts_n
MF_A30	gpioa(30)	static_addr(20)	pciea_wake_n	vdo_green[0]	INVALID	uarta_sin
MF_A31	gpioa(31)	static_addr(21)	pcieb_wake_n	vdo_red[0]	INVALID	uarta_sout
MF_B0	gpiob(0)	static_addr(0)	audio_txlrck	INVALID	INVALID	audio_rxlrck
MF_B1	gpiob(1)	static_addr(1)	audio_tx_bitck	INVALID	INVALID	audio_rx_bitck
MF_B2	gpiob(2)	static_addr(2)	audio_txd(0)	INVALID	INVALID	INVALID
MF_B3	gpiob(3)	static_addr(3)	audio_txd(1)	vdo_blue[5]	INVALID	INVALID
MF_B4	gpiob(4)	static_addr(4)	audio_txd(2)	vdo_blue[6]	INVALID	audio_rxd(0)
MF_B5	gpiob(5)	static_addr(5)	audio_txd(3)	vdo_blue[7]	INVALID	sdb_dat0
MF_B6	gpiob(6)	static_addr(6)	audio_rxlrck	vdo_green[2]	INVALID	sdb_cmd
MF_B7	gpiob(7)	static_addr(7)	audio_rx_bitck	vdo_green[3]	INVALID	sdb_dat3
MF_B8	gpiob(8)	static_addr(8)	audio_rxd(0)	vdo_green[4]	uartb_ri_n	sd_clk
MF_B9	gpiob(9)	static_addr(9)	audio_rxd(1)	vdo_green[5]	uartb_cd_n	sdb_dat1
MF_B10	gpiob(10)	static_addr(10)	audio_rxd(2)	vdo_green[6]	uartb_dsr_n	sdb_dat2
MF_B11	gpiob(11)	static_addr(11)	audio_rxd(3)	vdo_green[7]	uartb_dtr_n	INVALID
MF_B12	gpiob(12)	static_addr(12)	eth_trig	vdo_red[2]	INVALID	uartb_cts_n
MF_B13	gpiob(13)	static_addr(13)	etha_pps	vdo_red[3]	INVALID	uartb_rts_n
MF_B14	gpiob(14)	static_addr(14)	ethb_pps	vdo_red[4]	INVALID	uartb_sin
MF_B15	gpiob(15)	static_addr(15)	INVALID	vdo_red[5]	INVALID	uartb_sout
MF_B16	gpiob(16)	static_addr(16)	usba_pwro	vdo_red[6]	INVALID	sda_dat1
MF_B17	gpiob(17)	static_addr(17)	usba_overi	vdo_red[7]	INVALID	sda_dat2