Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Freescale Semiconductor
Data Sheet: Technical Data

DSP56371
Rev. 4.1, 1/2007

DSP56371 Data Sheet

1 Introduction

The DSP56371 is a high density CMOS device with 5.0-V compatible inputs and outputs.

NOTE

> This document contains information on a new product. Specifications and information herein are subject to change without notice.

Finalized specifications may be published after further characterization and device qualifications are completed.

For software or simulation models (for example, IBIS files), contact sales or go to www.freescale.com.

2 DSP56371 Overview

2.1 Introduction

This manual describes the DSP56371 24-bit digital signal processor (DSP), its memory, operating modes and peripheral modules. The DSP56371 is a member of

Table of Contents

the DSP56300 family of programmable CMOS DSPs. The DSP56371 is targeted to applications that require digital audio compression/decompression, sound field processing, acoustic equalization and other digital audio algorithms. Changes in core functionality specific to the DSP56371 are also described in this manual. See Figure 1. for the block diagram of the DSP56371.

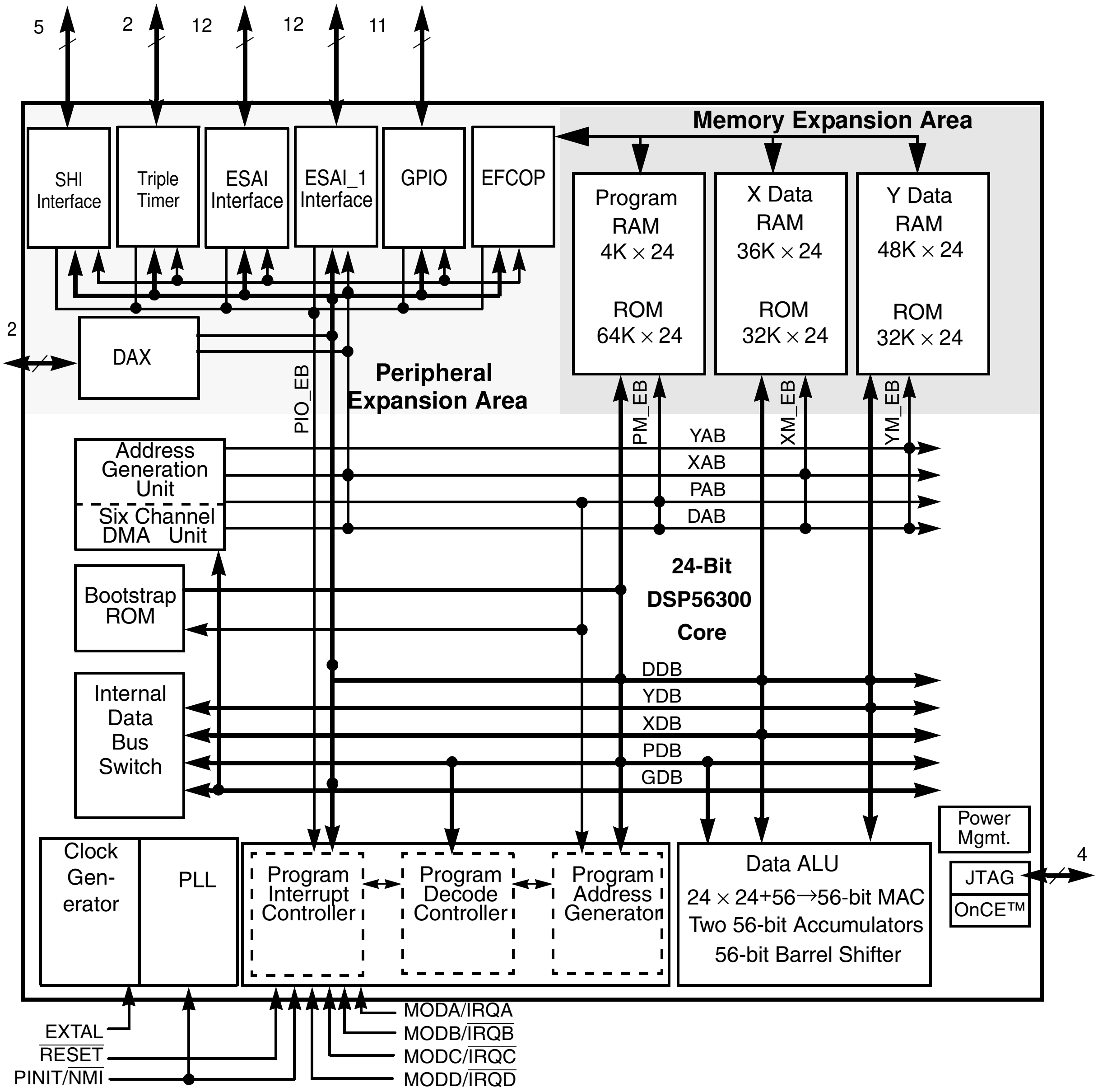

Figure 1. DSP56371 Block Diagram

2.2 DSP56300 Core Description

The DSP56371 uses the DSP56300 core, a high-performance, single clock cycle per instruction engine that provides up to twice the performance of Motorola's popular DSP56000 core family while retaining code compatibility with it.

The DSP56300 core family offers a new level of performance in speed and power, provided by its rich instruction set and low power dissipation, thus enabling a new generation of wireless, telecommunications and multimedia products. For a description of the DSP56300 core, see *Section 2.4 DSP56300 Core Functional Blocks*. Significant architectural enhancements to the DSP56300 core family include a barrel shifter, 24-bit addressing, an instruction patch module and direct memory access (DMA).

The DSP56300 core family members contain the DSP56300 core and additional modules. The modules are chosen from a library of standard pre-designed elements such as memories and peripherals. New modules may be added to the library to meet customer specifications. A standard interface between the DSP56300 core and the on-chip memory and peripherals supports a wide variety of memory and peripheral configurations. Refer to DSP56371 User's Manual, *Memory Configuration* section.

Core features are described fully in the *DSP56300 Family Manual*. Pinout, memory and peripheral features are described in this manual.

- DSP56300 modular chassis
  - 181 Million Instructions Per Second (MIPS) with a 181 MHz clock at an internal logic supply (QVDDL) of 1.25 V
  - Object Code Compatible with the 56K core
  - Data ALU with a 24 x 24 bit multiplier-accumulator and a 56-bit barrel shifter. 16-bit arithmetic support
  - Program Control with position independent code support and instruction patch support
  - EFCOP running concurrently with the core, capable of executing 181 million filter taps per second at peak performance
  - Six-channel DMA controller
  - Low jitter, PLL based clocking with a wide range of frequency multiplications (1 to 255), predivider factors (1 to 31) and power saving clock divider (2^i: i=0 to 7). Reduces clock noise.
  - Internal address tracing support and OnCE for Hardware/Software debugging
  - JTAG port
  - Very low-power CMOS design, fully static design with operating frequencies down to DC
  - STOP and WAIT low-power standby modes
- On-chip Memory Configuration
  - 48Kx24 Bit Y-Data RAM and 32Kx24 Bit Y-Data ROM
  - 36Kx24 Bit X-Data RAM and 32Kx24 Bit X-Data ROM
  - 64Kx24 Bit Program and Bootstrap ROM
  - 4Kx24 Bit Program RAM.
  - PROM patching mechanism
  - Up to 32Kx24 Bit from Y Data RAM and 8Kx24 Bit from X Data RAM can be switched to Program RAM resulting in up to 44Kx24 Bit of Program RAM.
- Peripheral modules
  - Enhanced Serial Audio Interface (ESAI): up to 4 receivers and up to 6 transmitters, master or

slave. I^2S, left justified, right justified, Sony, AC97, network and other programmable protocols

- Enhanced Serial Audio Interface I (ESAI_1): up to 4 receivers and up to 6 transmitters, master or slave. I^2S, left justified, right justified, Sony, AC97, network and other programmable protocols
- Serial Host Interface (SHI): SPI and I^2C protocols, multi master capability in I^2C mode, 10-word receive FIFO, support for 8, 16 and 24-bit words
- Triple Timer module (TEC).
- 11 dedicated GPIO pins
- Digital Audio Transmitter (DAX): 1 serial transmitter capable of supporting the SPDIF, IEC958, CP-340 and AES/EBU digital audio formats
- Pins of unused peripherals (except SHI) may be programmed as GPIO lines

2.3 DSP56371 Audio Processor Architecture

This section defines the DSP56371 audio processor architecture. The audio processor is composed of the following units:

- The DSP56300 core is composed of the Data ALU, Address Generation Unit, Program Controller, DMA Controller, Memory Module Interface, Peripheral Module Interface and the On-Chip Emulator (OnCE). The DSP56300 core is described in the document *<st-blue>DSP56300 24-Bit Digital Signal Processor Family Manual, Motorola publication DSP56300FM/AD*.
- Phased Lock Loop and Clock Generator
- Memory modules
- Peripheral modules. The peripheral modules are defined in the following sections.

Memory sizes in the block diagram are defaults. Memory may be differently partitioned, according to the memory mode of the chip. See Section 2.4.7 *On-Chip Memory* for more details about memory size.

2.4 DSP56300 Core Functional Blocks

The DSP56300 core provides the following functional blocks:

- Data arithmetic logic unit (Data ALU)
- Address generation unit (AGU)
- Program control unit (PCU)
- DMA controller (with six channels)
- Instruction patch controller
- PLL-based clock oscillator
- OnCE module
- Memory

In addition, the DSP56371 provides a set of on-chip peripherals, described in Section 2.5 *Peripheral Overview*.

2.4.1 Data ALU

The Data ALU performs all the arithmetic and logical operations on data operands in the DSP56300 core. The components of the Data ALU are as follows:

- Fully pipelined 24-bit × 24-bit parallel multiplier-accumulator (MAC)
- Bit field unit, comprising a 56-bit parallel barrel shifter (fast shift and normalization; bit stream generation and parsing)
- Conditional ALU instructions
- 24-bit or 16-bit arithmetic support under software control
- Four 24-bit input general purpose registers: X1, X0, Y1, and Y0
- Six Data ALU registers (A2, A1, A0, B2, B1 and B0) that are concatenated into two general purpose, 56-bit accumulators (A and B), accumulator shifters
- Two data bus shifter/limiter circuits

2.4.1.1 Data ALU Registers

The Data ALU registers can be read or written over the X memory data bus (XDB) and the Y memory data bus (YDB) as 24- or 48-bit operands (or as 16- or 32-bit operands in 16-bit arithmetic mode). The source operands for the Data ALU, which can be 24, 48, or 56 bits (16, 32, or 40 bits in 16-bit arithmetic mode), always originate from Data ALU registers. The results of all Data ALU operations are stored in an accumulator.

All the Data ALU operations are performed in two clock cycles in pipeline fashion so that a new instruction can be initiated in every clock, yielding an effective execution rate of one instruction per clock cycle. The destination of every arithmetic operation can be used as a source operand for the immediately following arithmetic operation without a time penalty (for example, without a pipeline stall).

2.4.1.2 Multiplier-Accumulator (MAC)

The MAC unit comprises the main arithmetic processing unit of the DSP56300 core and performs all of the calculations on data operands. In the case of arithmetic instructions, the unit accepts as many as three input operands and outputs one 56-bit result of the following form- Extension:Most Significant Product:Least Significant Product (EXT:MSP:LSP).

The multiplier executes 24-bit × 24-bit, parallel, fractional multiplies, between two's-complement signed, unsigned, or mixed operands. The 48-bit product is right-justified and added to the 56-bit contents of either the A or B accumulator. A 56-bit result can be stored as a 24-bit operand. The LSP can either be truncated or rounded into the MSP. Rounding is performed if specified.

2.4.2 Address Generation Unit (AGU)

The AGU performs the effective address calculations using integer arithmetic necessary to address data operands in memory and contains the registers used to generate the addresses. It implements four types of arithmetic: linear, modulo, multiple wrap-around modulo and reverse-carry. The AGU operates in parallel with other chip resources to minimize address-generation overhead.

The AGU is divided into two halves, each with its own Address ALU. Each Address ALU has four sets of register triplets, and each register triplet is composed of an address register, an offset register and a modifier register. The two Address ALUs are identical. Each contains a 24-bit full adder (called an offset adder).

A second full adder (called a modulo adder) adds the summed result of the first full adder to a modulo value that is stored in its respective modifier register. A third full adder (called a reverse-carry adder) is also provided.

The offset adder and the reverse-carry adder are in parallel and share common inputs. The only difference between them is that the carry propagates in opposite directions. Test logic determines which of the three summed results of the full adders is output.

Each Address ALU can update one address register from its respective address register file during one instruction cycle. The contents of the associated modifier register specifies the type of arithmetic to be used in the address register update calculation. The modifier value is decoded in the Address ALU.

2.4.3 Program Control Unit (PCU)

The PCU performs instruction prefetch, instruction decoding, hardware DO loop control and exception processing. The PCU implements a seven-stage pipeline and controls the different processing states of the DSP56300 core. The PCU consists of the following three hardware blocks:

- Program decode controller (PDC)
- Program address generator (PAG)
- Program interrupt controller

The PDC decodes the 24-bit instruction loaded into the instruction latch and generates all signals necessary for pipeline control. The PAG contains all the hardware needed for program address generation, system stack and loop control. The Program interrupt controller arbitrates among all interrupt requests (internal interrupts, as well as the five external requests: $\overline{\text{IRQA}}$, $\overline{\text{IRQB}}$, $\overline{\text{IRQC}}$, $\overline{\text{IRQD}}$ and $\overline{\text{NMI}}$) and generates the appropriate interrupt vector address.

PCU features include the following:

- Position independent code support
- Addressing modes optimized for DSP applications (including immediate offsets)
- On-chip instruction cache controller
- On-chip memory-expandable hardware stack
- Nested hardware DO loops
- Fast auto-return interrupts

The PCU implements its functions using the following registers:

- PC—program counter register
- SR—Status register
- LA—loop address register
- LC—loop counter register
- VBA—vector base address register
- SZ—stack size register
- SP—stack pointer
- OMR—operating mode register
- SC—stack counter register

The PCU also includes a hardware system stack (SS).

2.4.4 Internal Buses

To provide data exchange between blocks, the following buses are implemented:

- Peripheral input/output expansion bus (PIO_EB) to peripherals
- Program memory expansion bus (PM_EB) to program memory
- X memory expansion bus (XM_EB) to X memory
- Y memory expansion bus (YM_EB) to Y memory
- Global data bus (GDB) between registers in the DMA, AGU, OnCE, PLL, BIU and PCU, as well as the memory-mapped registers in the peripherals
- DMA data bus (DDB) for carrying DMA data between memories and/or peripherals
- DMA address bus (DAB) for carrying DMA addresses to memories and peripherals
- Program Data Bus (PDB) for carrying program data throughout the core
- X memory Data Bus (XDB) for carrying X data throughout the core
- Y memory Data Bus (YDB) for carrying Y data throughout the core
- Program address bus (PAB) for carrying program memory addresses throughout the core
- X memory address bus (XAB) for carrying X memory addresses throughout the core
- Y memory address bus (YAB) for carrying Y memory addresses throughout the core

All internal buses on the DSP56300 family members are 24-bit buses. See Figure 1.

2.4.5 Direct Memory Access (DMA)

The DMA block has the following features:

- Six DMA channels supporting internal and external accesses
- One-, two- and three-dimensional transfers (including circular buffering)
- End-of-block-transfer interrupts

- Triggering from interrupt lines and all peripherals

2.4.6 PLL-based Clock Oscillator

The clock generator in the DSP56300 core is composed of two main blocks: the PLL, which performs clock input division, frequency multiplication, skew elimination and the clock generator (CLKGEN), which performs low-power division and clock pulse generation. PLL-based clocking:

- Allows change of low-power divide factor (DF) without loss of lock
- Provides output clock with skew elimination
- Provides a wide range of frequency multiplications (1 to 255), predivider factors (1 to 31), PLL feedback multiplier (2 or 4), output divide factor (1, 2 or 4), and a power-saving clock divider (2^i: i = 0 to 7) to reduce clock noise

The PLL allows the processor to operate at a high internal clock frequency using a low frequency clock input. This feature offers two immediate benefits:

- A lower frequency clock input reduces the overall electromagnetic interference generated by a system.
- The ability to oscillate at different frequencies reduces costs by eliminating the need to add additional oscillators to a system.

NOTE

The PLL will momentarily overshoot the target frequency when the PLL is first enabled or when the VCO frequency is modified. It is important that when modifying the PLL frequency or enabling the PLL that the two-step procedure defined in Section 3, *DSP56371 Overview* be followed.

2.4.7 On-Chip Memory

The memory space of the DSP56300 core is partitioned into program memory space, X data memory space and Y data memory space. The data memory space is divided into X and Y data memory in order to work with the two Address ALUs and to feed two operands simultaneously to the Data ALU. Memory space includes internal RAM and ROM and can not be expanded off-chip.

There is an instruction patch module. The patch module is used to patch program ROM. The memory switch mode is used to increase the size of program RAM as needed (switch from X data RAM and/or Y data RAM).

There are on-chip ROMs for program and bootstrap memory (64K x 24-bit), X ROM (32K x 24-bit) and Y ROM (32K x 24-bit).

More information on the internal memory is provided in the DSP56371 User’s Manual, *Memory* section.

2.4.8 Off-Chip Memory Expansion

Memory cannot be expanded off-chip. There is no external memory bus.

2.5 Peripheral Overview

The DSP56371 is designed to perform a wide variety of fixed-point digital signal processing functions. In addition to the core features previously discussed, the DSP56371 provides the following peripherals:

- As many as 39 dedicate or user-configurable general purpose input/output (GPIO) signals
- Timer/event counter (TEC) module, containing three independent timers
- Memory switch mode in on-chip memory
- Four external interrupt/mode control lines and one external non-maskable interrupt line
- Enhanced serial audio interface (ESAI) with up to four receivers and up to six transmitters, master or slave, using the I^2S, Sony, AC97, network and other programmable protocols
- A second enhanced serial audio interface (ESAI_1) with up to four receivers and up to six transmitters, master or slave, using the I^2S, Sony, AC97, network and other programmable protocols.
- Serial host interface (SHI) using SPI and I^2C protocols, with multi-master capability, 10-word receive FIFO and support for 8-, 16- and 24-bit words
- A Digital audio transmitter (DAX): a serial transmitter capable of supporting the SPDIF, IEC958, CP-340 and AES/EBU digital audio formats

2.5.1 General Purpose Input/Output (GPIO)

The DSP56371 provides 11 dedicated GPIO and 28 programmable signals that can operate either as GPIO pins or peripheral pins (ESAI, ESAI_1, DAX, and TEC). The signals are configured as GPIO after hardware reset. Register programming techniques for all GPIO functionality among these interfaces are very similar and are described in the following sections.

2.5.2 Triple Timer (TEC)

This section describes a peripheral module composed of a common 21-bit prescaler and three independent and identical general purpose 24-bit timer/event counters, each one having its own register set.

Each timer can use internal or external clocking and can interrupt the DSP after a specified number of events (clocks). Two of the three timers can signal an external device after counting internal events. Each timer can also be used to trigger DMA transfers after a specified number of events (clocks) occurred. Two of the three timers connect to the external world through bidirectional pins (TIO0, TIO1). When a TIO pin is configured as input, the timer functions as an external event counter or can measure external pulse width/signal period. When a TIO pin is used as output the timer is functioning as either a timer, a watchdog or a Pulse Width Modulator. When a TIO pin is not used by the timer it can be used as a General Purpose Input/Output Pin. Refer to DSP56371 User's Manual, *Triple Timer Module* section.

2.5.3 Enhanced Serial Audio Interface (ESAI)

The ESAI provides a full-duplex serial port for serial communication with a variety of serial devices including one or more industry-standard codecs, other DSPs, microprocessors and peripherals that implement the Motorola SPI serial protocol. The ESAI consists of independent transmitter and receiver sections, each with its own clock generator. It is a superset of the DSP56300 family ESSI peripheral and of the DSP56000 family SAI peripheral. For more information on the ESAI, refer to DSP56371 User's Manual, *Enhanced Serial Audio Interface (ESAI)* section.

2.5.4 Enhanced Serial Audio Interface 1 (ESAI_1)

The ESAI_1 is a second ESAI interface. The ESAI_1 is functionally identical to ESAI. For more information on the ESAI_1, refer to DSP56371 User's Manual, *Enhanced Serial Audio Interface (ESAI_1)* section.

2.5.5 Serial Host Interface (SHI)

The SHI is a serial input/output interface providing a path for communication and program/coefficient data transfers between the DSP and an external host processor. The SHI can also communicate with other serial peripheral devices. The SHI can interface directly to either of two well-known and widely used synchronous serial buses: the Motorola serial peripheral interface (SPI) bus and the Philips inter-integrated-circuit control (I^2C) bus. The SHI supports either the SPI or I^2C bus protocol, as required, from a slave or a single-master device. To minimize DSP overhead, the SHI supports single-, double- and triple-byte data transfers. The SHI has a 10-word receive FIFO that permits receiving up to 30 bytes before generating a receive interrupt, reducing the overhead for data reception. For more information on the SHI, refer to DSP56371 User's Manual, *Serial Host Interface* section.

2.5.6 Digital Audio Transmitter (DAX)

The DAX is a serial audio interface module that outputs digital audio data in the AES/EBU, CP-340 and IEC958 formats. For more information on the DAX, refer to DSP56371 User's Manual, *Digital Audio* section.

3 Signal/Connection Descriptions

3.1 Signal Groupings

The input and output signals of the DSP56374 are organized into functional groups, which are listed in Table 1. and illustrated in Figure 2.

The DSP56374 is operated from a 1.25 V and 3.3 V supply; however, some of the inputs can tolerate 5.0 V. A special notice for this feature is added to the signal descriptions of those inputs.

Table 1. DSP56374 Functional Signal Groupings

Functional Group		Number of Signals	Detailed Description
Power (V_{DD})		12	Table 2
Ground (GND)		12	Table 3
Scan Pins		1	Table 4
Clock and PLL		2	Table 5
Interrupt and mode control		5	Table 6
SHI		5	Table 7
ESAI	Port C[1]	12	Table 8
ESAI_1	Port E[2]	12	Table 9
SPDIF Transmitter (DAX)	Port D[3]	2	Table 10
Dedicated GPIO	Port F[4]	11	Table 11
Timer		2	Table 12
JTAG/OnCE Port		4	Table 13

Note:

1. Port C signals are the GPIO port signals which are multiplexed with the ESAI signals.
2. Port E signals are the GPIO port signals which are multiplexed with the ESAI_1 signals.
3. Port D signals are the GPIO port signals which are multiplexed with the DAX signals.
4. Port F signals are the dedicated GPIO port signals.

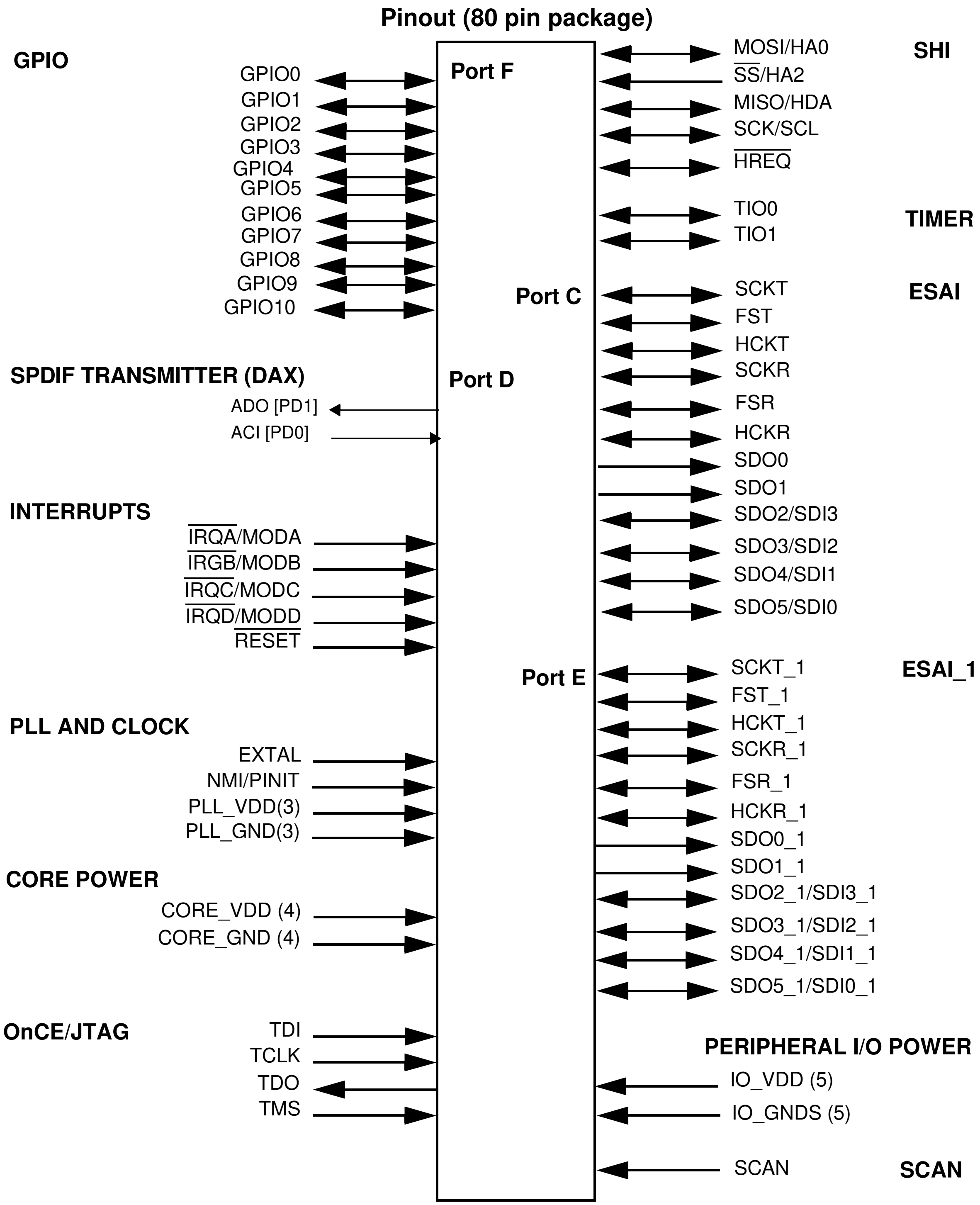

Figure 2. Signals Identified by Functional Group

3.2 Power

Table 2. Power Inputs

Power Name	Description
PLLA_VDD (1) PLLP_VDD(1)	**PLL Power**— The voltage (3.3 V) should be well-regulated and the input should be provided with an extremely low impedance path to the 3.3 V_{DD} power rail. The user must provide adequate external decoupling capacitors.
PLLD_VDD (1)	**PLL Power—** The voltage (1.25 V) should be well-regulated and the input should be provided with an extremely low impedance path to the 1.25 V_{DD} power rail. The user must provide adequate external decoupling capacitors.
CORE_VDD (4)	**Core Power**—The voltage (1.25 V) should be well-regulated and the input should be provided with an extremely low impedance path to the 1.25 V_{DD} power rail. The user must provide adequate decoupling capacitors.
IO_VDD (5)	**SHI, ESAI, ESAI_1, DAX and Timer I/O Power** —The voltage (3.3 V) should be well-regulated and the input should be provided with an extremely low impedance path to the 3.3 V_{DD} power rail. This is an isolated power for the SHI, ESAI, ESAI_1, DAX and Timer I/O. The user must provide adequate external decoupling capacitors.

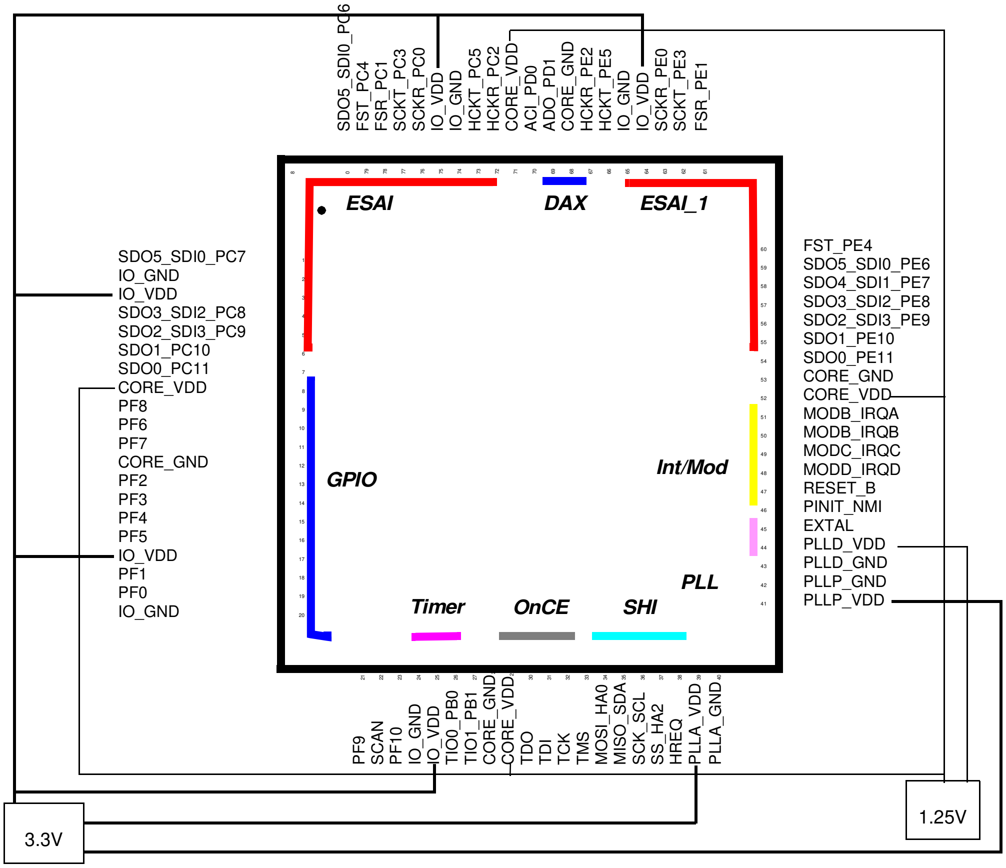

Figure 3. VDD Connections

3.3 Ground

Table 3. Grounds

Ground Name	Description

PLLA_GND(1)
PLLP_GND(1)
	PLL Ground—The PLL ground should be provided with an extremely low-impedance path to ground. The user must provide adequate external decoupling capacitors.

PLLD_GND(1)	**PLL Ground**—The PLL ground should be provided with an extremely low-impedance path to ground. The user must provide adequate external decoupling capacitors.
CORE_GND (4)	**Core Ground**—The Core ground should be provided with an extremely low-impedance path to ground. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors.
IO_GND (5)	**SHI, ESAI, ESAI_1, DAX and Timer I/O Ground**—IO_GND is an isolated ground for the SHI, ESAI, ESAI_1, DAX and Timer I/O. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors.

3.4 SCAN

Table 4. SCAN Signals

Signal Name	Type	State During Reset	Signal Description
SCAN	Input	Input	**SCAN**—Manufacturing test pin. This pin should be pulled low. Internal Pull down resistor.

3.5 Clock and PLL

Table 5. Clock and PLL Signals

Signal Name	Type	State during Reset	Signal Description
EXTAL	Input	Input	**External Clock Input**—An external clock source must be connected to EXTAL in order to supply the clock to the internal clock generator and PLL. *This input is 5 V tolerant.*
PINIT/NMI	Input	Input	**PLL Initial/Nonmaskable Interrupt**—During assertion of RESET, the value of PINIT/NMI is written into the PLL Enable (PEN) bit of the PLL control register, determining whether the PLL is enabled or disabled. After RESET de assertion and during normal instruction processing, the PINIT/NMI Schmitt-trigger input is a negative-edge-triggered nonmaskable interrupt (NMI) request internally synchronized to internal system clock. Internal Pull up resistor. This input is 5 V tolerant.

3.6 Interrupt and Mode Control

The interrupt and mode control signals select the chip's operating mode as it comes out of hardware reset. After $\overline{\text{RESET}}$ is deasserted, these inputs are hardware interrupt request lines.

Table 6. Interrupt and Mode Control

Signal Name	Type	State During Reset	Signal Description
MODA/$\overline{\text{IRQA}}$	Input	Input	**Mode Select A/External Interrupt Request A—**MODA/$\overline{\text{IRQA}}$ is an active-low Schmitt-trigger input, internally synchronized to the DSP clock. MODA/$\overline{\text{IRQA}}$ selects the initial chip operating mode during hardware reset and becomes a level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC and MODD select one of 16 initial chip operating modes, latched into the OMR when the $\overline{\text{RESET}}$ signal is deasserted. If the processor is in the stop standby state and the MODA/$\overline{\text{IRQA}}$ pin is pulled to GND, the processor will exit the stop state. Internal Pull up resistor. This input is 5 V tolerant.

Table 6. Interrupt and Mode Control (continued)

Signal Name	Type	State During Reset	Signal Description
MODB/$\overline{\text{IRQB}}$	Input	Input	**Mode Select B/External Interrupt Request B—**MODB/$\overline{\text{IRQB}}$ is an active-low Schmitt-trigger input, internally synchronized to the DSP clock. MODB/$\overline{\text{IRQB}}$ selects the initial chip operating mode during hardware reset and becomes a level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC and MODD select one of 16 initial chip operating modes, latched into OMR when the $\overline{\text{RESET}}$ signal is deasserted. Internal Pull up resistor. This input is 5 V tolerant.
MODC/$\overline{\text{IRQC}}$	Input	Input	**Mode Select C/External Interrupt Request C—**MODC/$\overline{\text{IRQC}}$ is an active-low Schmitt-trigger input, internally synchronized to the DSP clock. MODC/$\overline{\text{IRQC}}$ selects the initial chip operating mode during hardware reset and becomes a level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC and MODD select one of 16 initial chip operating modes, latched into OMR when the $\overline{\text{RESET}}$ signal is deasserted. Internal Pull up resistor. This input is 5 V tolerant.
MODD/$\overline{\text{IRQD}}$	Input	Input	**Mode Select D/External Interrupt Request D**—MODD/$\overline{\text{IRQD}}$ is an active-low Schmitt-trigger input, internally synchronized to the DSP clock. MODD/$\overline{\text{IRQD}}$ selects the initial chip operating mode during hardware reset and becomes a level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC and MODD select one of 16 initial chip operating modes, latched into OMR when the $\overline{\text{RESET}}$ signal is deasserted. Internal Pull up resistor. This input is 5 V tolerant.
$\overline{\text{RESET}}$	Input	Input	**Reset—**$\overline{\text{RESET}}$ is an active-low, Schmitt-trigger input. When asserted, the chip is placed in the Reset state and the internal phase generator is reset. The Schmitt-trigger input allows a slowly rising input (such as a capacitor charging) to reset the chip reliably. When the $\overline{\text{RESET}}$ signal is deasserted, the initial chip operating mode is latched from the MODA, MODB, MODC and MODD inputs. The $\overline{\text{RESET}}$ signal must be asserted during power up. A stable EXTAL signal must be supplied while $\overline{\text{RESET}}$ is being asserted. Internal Pull up resistor. This input is 5 V tolerant.

3.7 Serial Host Interface

The SHI has five I/O signals that can be configured to allow the SHI to operate in either SPI or I^2C mode.

Table 7. Serial Host Interface Signals

Signal Name	Signal Type	State during Reset	Signal Description
SCK	Input or output	Tri-stated	**SPI Serial Clock**—The SCK signal is an output when the SPI is configured as a master and a Schmitt-trigger input when the SPI is configured as a slave. When the SPI is configured as a master, the SCK signal is derived from the internal SHI clock generator. When the SPI is configured as a slave, the SCK signal is an input, and the clock signal from the external master synchronizes the data transfer. The SCK signal is ignored by the SPI if it is defined as a slave and the slave select ($\overline{SS}$) signal is not asserted. In both the master and slave SPI devices, data is shifted on one edge of the SCK signal and is sampled on the opposite edge where data is stable. Edge polarity is determined by the SPI transfer protocol.
SCL	Input or output		**I^{2}C Serial Clock**—SCL carries the clock for I^{2}C bus transactions in the I^{2}C mode. SCL is a Schmitt-trigger input when configured as a slave and an open-drain output when configured as a master. SCL should be connected to V_{DD} through a pull-up resistor. This signal is tri-stated during hardware, software and individual reset. Thus, there is no need for an external pull-up in this state. Internal Pull up resistor. This input is 5 V tolerant.
MISO	Input or output	Tri-stated	**SPI Master-In-Slave-Out**—When the SPI is configured as a master, MISO is the master data input line. The MISO signal is used in conjunction with the MOSI signal for transmitting and receiving serial data. This signal is a Schmitt-trigger input when configured for the SPI Master mode, an output when configured for the SPI Slave mode, and tri-stated if configured for the SPI Slave mode when $\overline{SS}$ is deasserted. An external pull-up resistor is not required for SPI operation.
SDA	Input or open-drain output		**I^{2}C Data and Acknowledge**—In I^{2}C mode, SDA is a Schmitt-trigger input when receiving and an open-drain output when transmitting. SDA should be connected to V_{DD} through a pull-up resistor. SDA carries the data for I^{2}C transactions. The data in SDA must be stable during the high period of SCL. The data in SDA is only allowed to change when SCL is low. When the bus is free, SDA is high. The SDA line is only allowed to change during the time SCL is high in the case of start and stop events. A high-to-low transition of the SDA line while SCL is high is a unique situation, and it is defined as the start event. A low-to-high transition of SDA while SCL is high is a unique situation defined as the stop event. This signal is tri-stated during hardware, software and individual reset. Thus, there is no need for an external pull-up in this state. Internal Pull up resistor. This input is 5 V tolerant.

Table 7. Serial Host Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
MOSI	Input or output	Tri-stated	**SPI Master-Out-Slave-In**—When the SPI is configured as a master, MOSI is the master data output line. The MOSI signal is used in conjunction with the MISO signal for transmitting and receiving serial data. MOSI is the slave data input line when the SPI is configured as a slave. This signal is a Schmitt-trigger input when configured for the SPI Slave mode.
HA0	Input		**I^{2}C Slave Address 0**—This signal uses a Schmitt-trigger input when configured for the I^{2}C mode. When configured for I^{2}C slave mode, the HA0 signal is used to form the slave device address. HA0 is ignored when configured for the I^{2}C master mode. This signal is tri-stated during hardware, software and individual reset. Thus, there is no need for an external pull-up in this state. Internal Pull up resistor. This input is 5 V tolerant.
SS	Input	Tri-stated	**SPI Slave Select**—This signal is an active low Schmitt-trigger input when configured for the SPI mode. When configured for the SPI Slave mode, this signal is used to enable the SPI slave for transfer. When configured for the SPI master mode, this signal should be kept deasserted (pulled high). If it is asserted while configured as SPI master, a bus error condition is flagged. If SS is deasserted, the SHI ignores SCK clocks and keeps the MISO output signal in the high-impedance state.
HA2	Input		**I^{2}C Slave Address 2**—This signal uses a Schmitt-trigger input when configured for the I^{2}C mode. When configured for the I^{2}C Slave mode, the HA2 signal is used to form the slave device address. HA2 is ignored in the I^{2}C master mode. This signal is tri-stated during hardware, software and individual reset. Thus, there is no need for an external pull-up in this state. Internal Pull up resistor. This input is 5 V tolerant.
HREQ	Input or Output	Tri-stated	**Host Request**—This signal is an active low Schmitt-trigger input when configured for the master mode but an active low output when configured for the slave mode. When configured for the slave mode, HREQ is asserted to indicate that the SHI is ready for the next data word transfer and deasserted at the first clock pulse of the new data word transfer. When configured for the master mode, HREQ is an input. When asserted by the external slave device, it will trigger the start of the data word transfer by the master. After finishing the data word transfer, the master will await the next assertion of HREQ to proceed to the next transfer. This signal is tri-stated during hardware, software, personal reset, or when the HREQ1–HREQ0 bits in the HCSR are cleared. There is no need for an external pull-up in this state. Internal Pull up resistor. This input is 5 V tolerant.

3.8 Enhanced Serial Audio Interface

Table 8. Enhanced Serial Audio Interface Signals

Signal Name	Signal Type	State during Reset	Signal Description
HCKR	Input or output	GPIO disconnected	**High Frequency Clock for Receiver**—When programmed as an input, this signal provides a high frequency clock source for the ESAI receiver as an alternate to the DSP core clock. When programmed as an output, this signal can serve as a high-frequency sample clock (for example, for external digital to analog converters [DACs]) or as an additional system clock.
PC2	Input, output, or disconnected		**Port C2**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.
HCKT	Input or output	GPIO disconnected	**High Frequency Clock for Transmitter**—When programmed as an input, this signal provides a high frequency clock source for the ESAI transmitter as an alternate to the DSP core clock. When programmed as an output, this signal can serve as a high frequency sample clock (for example, for external DACs) or as an additional system clock.
PC5	Input, output, or disconnected		**Port C5**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.

Table 8. Enhanced Serial Audio Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
FSR	Input or output	GPIO disconnected	**Frame Sync for Receiver**—This is the receiver frame sync input/output signal. In the asynchronous mode (SYN=0), the FSR pin operates as the frame sync input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as either the serial flag 1 pin (TEBE=0), or as the transmitter external buffer enable control (TEBE=1, RFSD=1). When this pin is configured as serial flag pin, its direction is determined by the RFSD bit in the RCCR register. When configured as the output flag OF1, this pin will reflect the value of the OF1 bit in the SAICR register, and the data in the OF1 bit will show up at the pin synchronized to the frame sync in normal mode or the slot in network mode. When configured as the input flag IF1, the data value at the pin will be stored in the IF1 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.
PC1	Input, output, or disconnected		**Port C1**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.
FST	Input or output	GPIO disconnected	**Frame Sync for Transmitter**—This is the transmitter frame sync input/output signal. For synchronous mode, this signal is the frame sync for both transmitters and receivers. For asynchronous mode, FST is the frame sync for the transmitters only. The direction is determined by the transmitter frame sync direction (TFSD) bit in the ESAI transmit clock control register (TCCR).
PC4	Input, output, or disconnected		**Port C4**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.

Table 8. Enhanced Serial Audio Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
SCKR	Input or output	GPIO disconnected	**Receiver Serial Clock**—SCKR provides the receiver serial bit clock for the ESAI. The SCKR operates as a clock input or output used by all the enabled receivers in the asynchronous mode (SYN=0), or as serial flag 0 pin in the synchronous mode (SYN=1). When this pin is configured as serial flag pin, its direction is determined by the RCKD bit in the RCCR register. When configured as the output flag OF0, this pin will reflect the value of the OF0 bit in the SAICR register, and the data in the OF0 bit will show up at the pin synchronized to the frame sync in normal mode or the slot in network mode. When configured as the input flag IF0, the data value at the pin will be stored in the IF0 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.
PC0	Input, output, or disconnected		**Port C0**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.
SCKT	Input or output	GPIO disconnected	**Transmitter Serial Clock**—This signal provides the serial bit rate clock for the ESAI. SCKT is a clock input or output used by all enabled transmitters and receivers in synchronous mode, or by all enabled transmitters in asynchronous mode.
PC3	Input, output, or disconnected		**Port C3**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.
SDO5	Output	GPIO disconnected	**Serial Data Output 5**—When programmed as a transmitter, SDO5 is used to transmit data from the TX5 serial transmit shift register.
SDI0	Input		**Serial Data Input 0**—When programmed as a receiver, SDI0 is used to receive serial data into the RX0 serial receive shift register.
PC6	Input, output, or disconnected		**Port C6**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.

Table 8. Enhanced Serial Audio Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
SDO4 SDI1 PC7	Output Input Input, output, or disconnected	GPIO disconnected	**Serial Data Output 4**—When programmed as a transmitter, SDO4 is used to transmit data from the TX4 serial transmit shift register. **Serial Data Input 1**—When programmed as a receiver, SDI1 is used to receive serial data into the RX1 serial receive shift register. **Port C7**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.
SDO3 SDI2 PC8	Output Input Input, output, or disconnected	GPIO disconnected	**Serial Data Output 3**—When programmed as a transmitter, SDO3 is used to transmit data from the TX3 serial transmit shift register. **Serial Data Input 2**—When programmed as a receiver, SDI2 is used to receive serial data into the RX2 serial receive shift register. **Port C8**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.
SDO2 SDI3 PC9	Output Input Input, output, or disconnected	GPIO disconnected	**Serial Data Output 2**—When programmed as a transmitter, SDO2 is used to transmit data from the TX2 serial transmit shift register **Serial Data Input 3**—When programmed as a receiver, SDI3 is used to receive serial data into the RX3 serial receive shift register. **Port C9**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.

Table 8. Enhanced Serial Audio Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
SDO1 PC10	Output Input, output, or disconnected	GPIO disconnected	**Serial Data Output 1**—SDO1 is used to transmit data from the TX1 serial transmit shift register. **Port C10**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.
SDO0 PC11	Output Input, output, or disconnected	GPIO disconnected	**Serial Data Output 0**—SDO0 is used to transmit data from the TX0 serial transmit shift register. **Port C11**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.

3.9 Enhanced Serial Audio Interface_1

Table 9. Enhanced Serial Audio Interface_1 Signals

Signal Name	Signal Type	State during Reset	Signal Description
HCKR_1	Input or output	GPIO disconnected	**High Frequency Clock for Receiver**—When programmed as an input, this signal provides a high frequency clock source for the ESAI_1 receiver as an alternate to the DSP core clock. When programmed as an output, this signal can serve as a high-frequency sample clock (for example, for external digital to analog converters [DACs]) or as an additional system clock.
PE2	Input, output, or disconnected		**Port E2**—When the ESAI_1 is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.
HCKT_1	Input or output	GPIO disconnected	**High Frequency Clock for Transmitter**—When programmed as an input, this signal provides a high frequency clock source for the ESAI_1 transmitter as an alternate to the DSP core clock. When programmed as an output, this signal can serve as a high frequency sample clock (for example, for external DACs) or as an additional system clock.
PE5	Input, output, or disconnected		**Port E5**—When the ESAI_1 is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.

Table 9. Enhanced Serial Audio Interface_1 Signals

Signal Name	Signal Type	State during Reset	Signal Description
FSR_1	Input or output	GPIO disconnected	**Frame Sync for Receiver_1**—This is the receiver frame sync input/output signal. In the asynchronous mode (SYN=0), the FSR_1 pin operates as the frame sync input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as either the serial flag 1 pin (TEBE=0), or as the transmitter external buffer enable control (TEBE=1, RFSD=1). When this pin is configured as serial flag pin, its direction is determined by the RFSD bit in the RCCR_1 register. When configured as the output flag OF1, this pin will reflect the value of the OF1 bit in the SAICR_1 register, and the data in the OF1 bit will show up at the pin synchronized to the frame sync in normal mode or the slot in network mode. When configured as the input flag IF1, the data value at the pin will be stored in the IF1 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.
PE1	Input, output, or disconnected		**Port E1**—When the ESAI_1 is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.
FST_1	Input or output	GPIO disconnected	**Frame Sync for Transmitter_1**—This is the transmitter frame sync input/output signal. For synchronous mode, this signal is the frame sync for both transmitters and receivers. For asynchronous mode, FST_1 is the frame sync for the transmitters only. The direction is determined by the transmitter frame sync direction (TFSD) bit in the ESAI_1 transmit clock control register (TCCR_1).
PE4	Input, output, or disconnected		**Port E4**—When the ESAI_1 is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. Internal Pull down resistor. This input is 5 V tolerant.