AS3524
Advanced Audio Processor System

1 Description

The AS3524 implements a highly flexible and fully integrated digital audio processor system combining strong calculating power and high performance interfaces commonly used within audio player systems.

Using advanced 0.13µm process technology and large on chip RAM leads to outstanding low power consumption of 0.3 mW/MHz for the ARM922T microcontroller core and 0.6 mW/MHz for the overall system measured with a typical MP3 player SW application.

Based on a powerful ARM9TDMI capable of performing up to 200MIPS it is suited to run MP3, AAC, WMA, OGG… decoders and encoders and, in addition, it can perform extensive user interfaces, motion graphics support, video playback and much more.

The AS3524 SOC (system-on-a-Chip) features dedicated high speed interfaces for ATA IDE, USB2.0 HS-OTG and SDRAM ensuring maximum performance for download, upload, and playback.

Furthermore interfaces for NAND flashes, MMC/SD cards and Memory Stick ensure most flexible system design possibilities. Hardware support for parallel interfaces lower the CPU load serving complex and/or colour user interfaces.

Additional serial high-speed data and control interfaces guarantee the connection to other peripherals and or processors in the system.

Two independently programmable PLLs generate the required frequencies for audio playback/recording, for the processor core and for the USB interface at the same time.

Key Features

1.1 Digital Core

Embedded 32-Bit RISC Controller

- ARM922TDMI RISC CPU
- 2.5Mbit on-chip RAM
- 1Mbit on chip ROM
- Clock speed max. 250MHz (200MIPS)
- Standard JTAG interface

USB 2.0 HS & OTG Interface

- Up to 480Mbit/s transfer speed
- USB 2.0 HS/FS physical inlcuding OTG support
- USB 2.0 HS/FS digital core including OTG host
- Dedicated dual port buffer RAM
- DMA bus master functionality

IDE Host Controller

- Supporting Ultra ATA 33/66/100/133 modes
- Programmable IO and Multi-word DMA capability
- Dedicated dual port buffer RAM
- DMA bus master functionality

External Memory Controller

- Dynamic memory interface
- Asynchronous static memory
- DMA bus master functionality

DMA Controller

- Single Master DMA controller
- 2 DMA channels possible at the same time
- 16 DMA requests supported

Interrupt Controller

- Support for 32 standard interrupts
- Support for 16 vectored IRQ interrupts

Audio Subsystem Interface

- Dedicated 2 wire serial control master
- I2S input and output with dual port buffer RAM

Nand Flash Interface

- 8 and 16bit flash support
- 3, 4 & 5 byte address support
- hardware ECC

MMC/SD Interface

- MMC/SD Card host for multiple card support
- 4 data line support for SD cards

MS / MS Pro Interface

- Dedicated dual port buffer RAM

Display Interface

- Serial and parallel controller supported
- On chip hardware acceleration

Synchronous Serial Interface

- Master and slave operation
- 8 and 16 bit support
- Several protocol standards supported

I2S Interface

- Input multiplexed with audio subsystem
- selectable SPDIF input conversion
- Dedicated dual port buffer RAM

2 Wire Serial Control Interface

- Master and slave operation
- Standard and fast mode support

General Purpose IO Interface

- 4x 8-bit ports

Multiple Boot Options

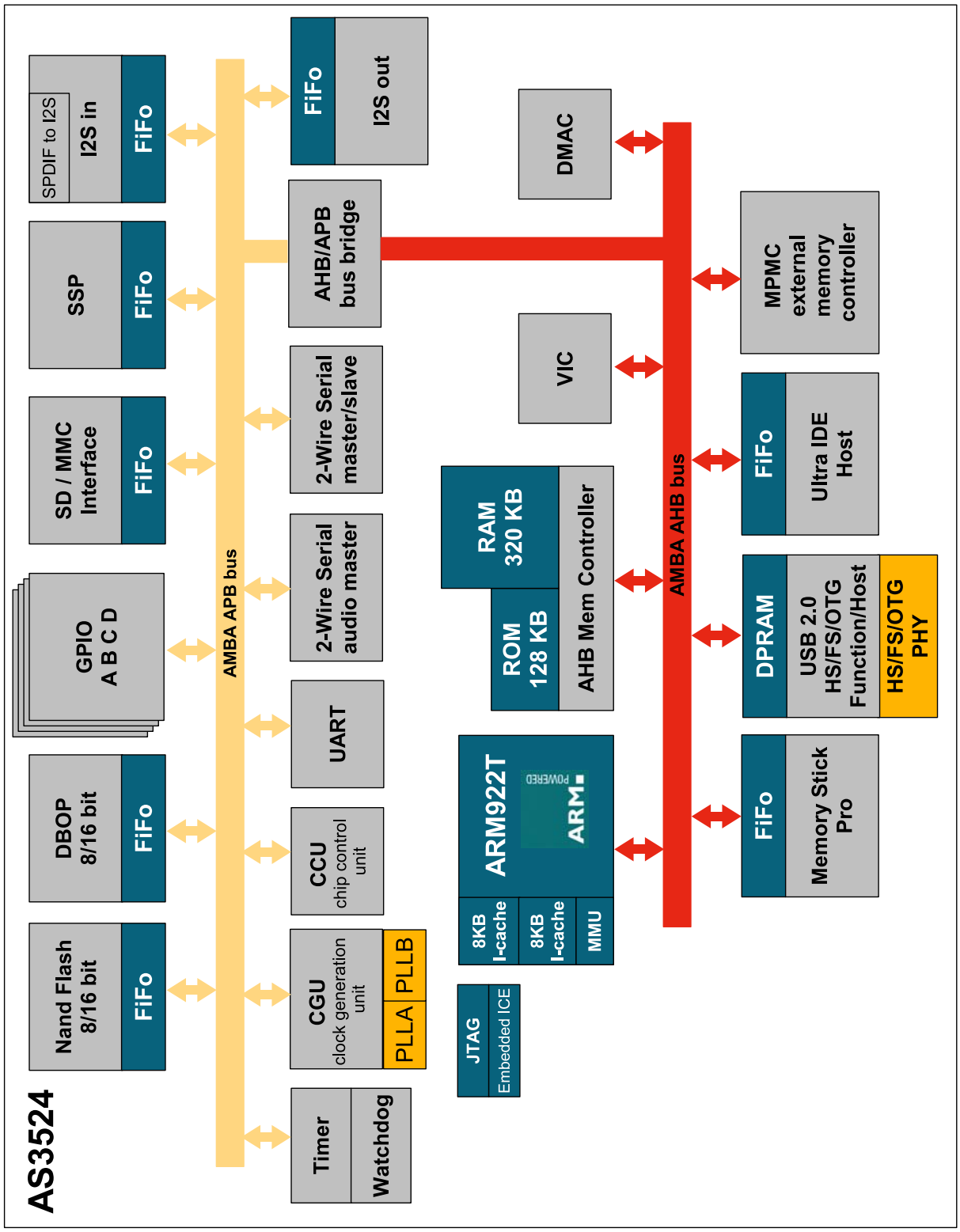
- Selection of internal ROM or external boot device
- Internal boot loader supporting boot from external NorFlash, NandFlash, IDE, SPI host
- Internal USB boot loader with USB promer supporting initial factory programming and firmware update

2 Application

- Portable Digital Audio Player and Recorder
- Portable Digital Media Player
- PDA
- Smartphone

3 Block Diagram

Figure 1 AS3524 Block Diagram



Document Revisions

Revision	Chapter	Date	Owner	Description
0.1	all	9.3.2005	MMA	first preliminary version
0.2		31.3.2005	MMA	package drawing and pinout added
0.3		14.9.2005	PKM	marking description and top view added
1.0	all	8.5.2006	WSG	first release of document generated
1.1	5.1.6.2, 8 5.3.13.1	25.9.2006	WSG	added description for modified C22 bootloader added description for UART Baud rate settings
1.11	5.1.6.2	9.11.2006	WSG	corrected table headers for boot device selection and USB boot frequency settings
1.12	4.2.4	24.3.2010	WSG	Added chapter “Startup Sequence for Supply Voltages”
	4.1	24.3.2010	WSG	Added table for soldering conditions according to Pb-Free process

Related Documents

ARM922T Technical Reference Manual	DDI0184B_922T_TRM.pdf	http://www.arm.com
ARM9TDMI Technical Reference Manual	DDI0180A_9tdmi_trm.pdf	http://www.arm.com
PrimeCell™ MultiPort Memory Controller; PL172 Technical Reference Manual		http://www.arm.com
AMBA Specification (Rev 2.0)	IHI0011A_AMBA_SPEC.pdf	http://www.arm.com
PrimeCell™ Synchronous Serial Port; PL022 Technical Reference Manual		http://www.arm.com
PrimeCell™ General Purpose Input/Output; PL061 Technical Reference Manual		http://www.arm.com
PrimeCell™ Single Master DMA Controller; PL081 Technical Reference Manual		http://www.arm.com
PrimeCell™ Multimedia Card Interface; PL180 Technical Reference Manual		http://www.arm.com
PrimeCell™ Vectored Interrupt Controller; PL190 Technical Reference Manual		http://www.arm.com
CWda03 - SPDIF-AES/EBU TO I2S CONVERTER		http://www.coreworks.pt
TSMC TPZ013G3 Standard I/O Library Databook		http://www.tsmc.com
DesignWare USB 2.0 HI-SPEED ON-THE-GO Controller Subsystem		http://www.synopsys.com
DesignWare USB 2 PHY Hardmacro		http://www.synopsys.com
SMS2IP mem stick host controller		http://www.sony.com
ICON mem stick host con interface		http://www.sony.com
IDE host controller BK3710S		http://www.palmchip.com
AS352x USB MSC Boot Promer specification document	AS352X_USB_MSC_Boot_Promer_V07.doc	

4 Electrical Specifications

4.1 Absolute Maximum Ratings

Stresses beyond those listed under “Absolute Maximum Ratings“ may cause permanent damage to the device. These are stress ratings only. Functional operation of the device at these or any other conditions beyond those indicated under “Operating Conditions” is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. The device should be operated under recommended operating conditions.

Table 1 Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit	Note
VDD_{peri}		-0.5	3.7	V	digital periphery supply voltage
VDD_{mem}		-0.5	3.7	V	digital IO supply for MPMC PADs
$USBVDDA33_T$		-0.5	3.7	V	USB analog supply transmit block to be connected to UVDD
$USBVDDA33_C$		-0.5	3.7	V	USB analog supply common block to be connected to UVDD
VDD_{core}		-0.5	1.68	V	digital core supply voltage
$VDD_{coreana}$		-0.5	1.68	V	core supply for critical blocks (1-TRAM)
$VDDA_{PLL}$		-0.5	1.68	V	core supply forPLLA, PLLB
V_{IN_5V}	5V pins	-0.5	7.0	V	Applicable for pins VBUS
V_{IN_VSS}	Voltage difference at VSS terminals	-0.5	0.5	V	Applicable for pins vss_core, vss_peri, vss_mem, usb_vssa33t, usb_vssa33c
I_{scr}	Input Current (latchup immunity)	-100	100	mA	Norm: JEDEC 17
ESD	Electrostatic Discharge HBM		+/-1	kV	Norm: MIL 883 E method 3015
ESD_USB	Electrostatic Discharge HBM for USB Pins		+/-2	kV	Norm: MIL 883 E method 3015 (Pins: usb_dp, usb_dm, usb_vbus)
P_t	Total Power Dissipation (all supplies and outputs)		1000	mW	for CTBGA180 package
T_{strg}	Storage Temperature	-55	125	°C	
H	Humidity non-condensing	5	85	%	

Table 2 Soldering Conditions

Symbol	Parameter	Min	Max	Unit	Note
T_c	Pb-Free Process – Classification Temperature		260	°C	According to IPC/JEDEC J-STD-020 (2), reflects moisture sensitivity level only
T_p	Pb-Free Process – Peak Temperature	255	260	°C	Peak temperature according to IPC/JEDEC J-STD-020 (2)
t_p	Time t_p		30	s	Time t_p within 5 ^{O}C of the specified classification temperature T_c according to IPC/JEDEC J-STD-020 (2)
D_{well}		30	45	s	Time above 217 ^{O}C

Note: (1) austriamicrosystems AG strongly recommends to use underfill.
(2) IPC/JEDEC J-STD-020: Moisture/Reflow Sensitivity Classification for Nonhermetic Solid State Surface Mount Devices.

4.2 Operating Conditions

4.2.1 Supply Voltages

Following supply voltages for the digital system are generated by internal LDOs.

Table 3 Operating conditions for internal generated supply voltages

Symbol	Parameter	Min	Max	Unit	Note
VDD_{peri}		3.0	3.6	V	digital periphery supply voltage
VDD_{mem}		1.75	3.4	V	digital IO supply for MPMC PADs
VDD_{core}		1.08	1.25	V	digital core supply voltage see Note (1)
$VDD_{coreana}$		1.08	1.25	V	core supply for critical blocks (1-TRAM)
$VDDA_{PLL}$		1.08	1.25	V	core supply forPLLA, PLLB
$USBVDDA33_{T}$		3.15	3.45	V	USB analog supply transmit block to be connected to UVDD
$USBVDDA33_{C}$		3.15	3.45	V	USB analog supply common block to be connected to UVDD
	Difference of Negative Supplies vss_peri, vss_core, vss_core_ana, vss_mem, vssa_pll, usb_vssa33c, usb_vssa33t,	-0.1	0.1	V	To achieve good performance, the negative supply terminals should be connected to low impedance ground plane.

Note(s) (1) For the VDD_CORE supply, voltage scaling should be applied to optimize power consumption and CPU speed performance. For normal operation with fclk (CPU ARM-922T clock) frequencies below 200 MHz, CVDD (supply of VDD_CORE) can be set to a lower value of 1.10 V. Only for setting fclk of the CPU to clock frequencies above 200 MHz, the VDD_CORE supply voltage must be set to 1.20 V typical conditions.

4.2.2 Operating Currents

Table 4 Supply currents

Symbol	Parameter	Typ	Max	Unit	Note
IDD_PERI_OP	Peripheral current	2	20	mA	
IDD_MEM_OP	External memory interface current	-	20	mA	(2)
IDD_CORE_OP	Digital core current	20	145	mA	(1), (2)
IDD_USBA33T_OP	USB transmitter current		30	mA	
IDD_USBA33C_OP	USB common blocks current		30	mA	

Notes (1) Typical condition for playback of MP3 music with 44.1 KHz / 128 kbit with 32Ω headphones. No external SDRAM connected. USB2.0 in standby.
(2) Maximum condition for ARM running at 250 MHz, AHB/APB bus and memory at 64 MHz, USB 2.0 in HS operation.

In the case of standby mode or in the case of configuring the device to stopped clock, following current consumption is measured.

Table 5 Leakage currents

Symbol	Parameter	Typ	Max	Unit	Note
IDD_PERI_LEAK			4	mA	Including USBA33T, USBA33C
IDD_MEM_LEAK			800	µA	
IDD_CORE_LEAK			3	mA	@ $T_{ambient}$=25 ^{O} C
IDD_LEAK(VDDAPLL+COREANA)			1.5	mA	@ $T_{ambient}$=25 ^{O} C

4.2.3 Temperature Range

Table 6 Temperature Range

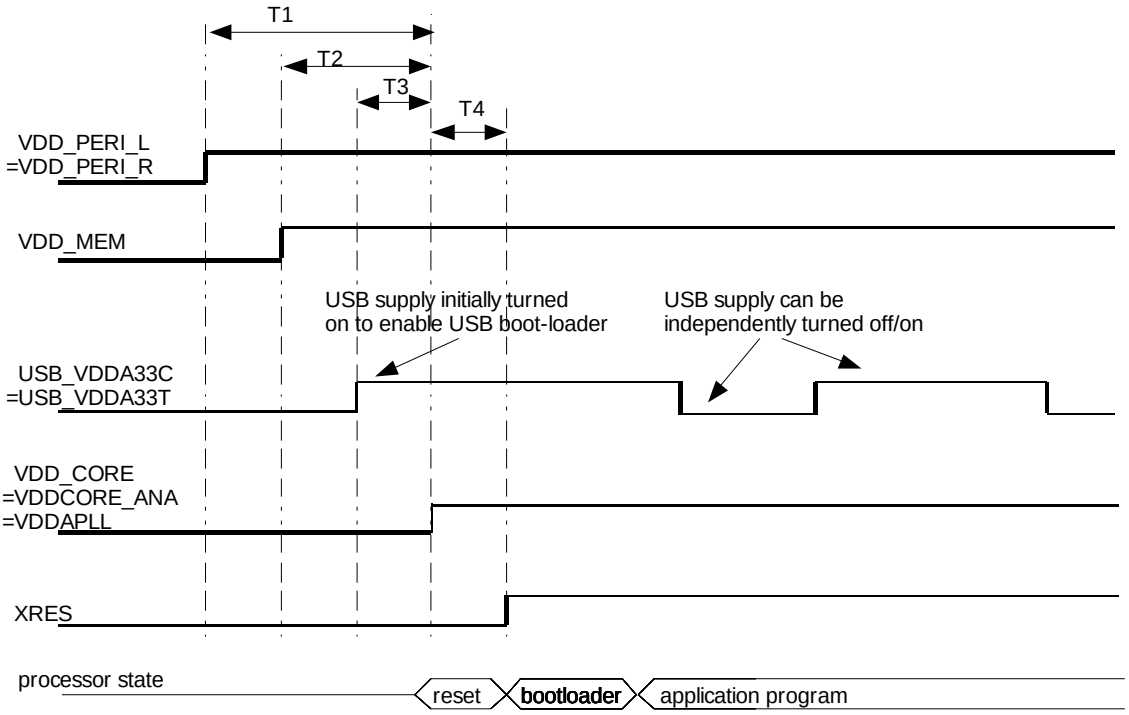
Symbol	Parameter	Min	Typ	Max	Unit	Note
T_{op}	Operating temperature range	0	25	85	°C	
T_j	Junction temperature range	0		110	°C	
R_{th}	Thermal Resistance		29		°C/W	

4.2.4 Start-up Sequence for Supply Voltages

To ensure correct start-up special timing constraints are given for circuit power-on. Also to ensure correct functionality of the USB Bootloader certain restrictions must be met.

For start-up it has to be guaranteed that VDD_CORE is not applied before all other peripheral supply voltages are switched on. The only exception are the USB suppies (USB_VDDA33C, USB_VDDA33T), which can be switched on and off independently of all other supplies.

Figure 2: AS3524 startup



Peripheral voltages can be started simultaneously, but because of transient currents during start-up flowing into decoupling caps it is suggested that these voltages are turned on in a serial sequence.

The USB supply voltage should be turned on at start-up to enable start of USB loader without any needed configuration for the AFE chip.

Symbol	Parameter	Min	Typ	Max	Unit
T1	delay between VDD_PERI and VDD_CORE startup	0.2	3		ms
T2	delay between VDD_MEM and VDD_CORE startup	0.2	2		ms
T3	delay between USB_VDD and VDD_CORE startup	0.2	1		ms
T4	delay between VDD_CORE startup and XRES release	0.01	1		ms

5 Detailed Functional Descriptions

5.1 ARM922-T Processor Core

5.1.1 General

The ARM922T macrocell is a high-performance 32-bit RISC integer processor combining an ARM9TDMI™ processor core with:

- 8KB instruction cache and 8 KB data cache
- Instruction and data Memory Management Unit (MMU)
- Write buffer with 16 data words and 4 addresses
- Advanced Microprocessor Bus Architecture (AMBA™) AHB interface

The ARM922T provides a high-performance processor solution for open systems requiring full virtual memory management and sophisticated memory protection. The ARM922T processor core is capable of running at 250 MHz. The ARM922T hard macrocell has a very low power consumption. The integrated cache helps to significantly reduce memory bandwith demands, improving performance and minimizing power consumption.

At 250 MHz the ARM922T comsumes as little as 65 mW, making it ideal for high-performance battery operated audio or video applications.

The ARM core and associated bus structures are configured for little endian byte order (compatible with Windows CE™ and Symbian™ OS).

Table 7 ARM 922T characteristics

Cache (I/D)	MMU	AHB	Thumb	mW/MHz	MHz
8KB / 8KB	yes	yes	yes	0.25 @ 1.2 V	250

Features

- 32-bit RISC architecture (ARMv4T)
- Harvard architecture with separated instruction (I) and data (D) caches with 8 KB each and 8-word line length
- Five stage pipeline (fetch, decode, execute, memory, write back) enabling high master clock speeds
- 32-bit ARM instruction set for maximum performance and flexibility
- 16-bit Thumb instruction set for increased code density
- Enhanced ARM architecture V4 MMU to provide translation and access permission checks for instruction and data addresses. With this MMU different operating systems (Windows CE, Symbian ...) can be implemented.
- Industry standard AMBA bus interface (AHB and APB)
- Hard-macro implementation
- The processor core clock frequency (FCLK) is programmable up to 250MHz and the ARM922 power consumption is directly proportional to this clock frequency FCLK

5.1.2 Block Diagrams

Figure 3 ARM 922T Functional Block Diagram

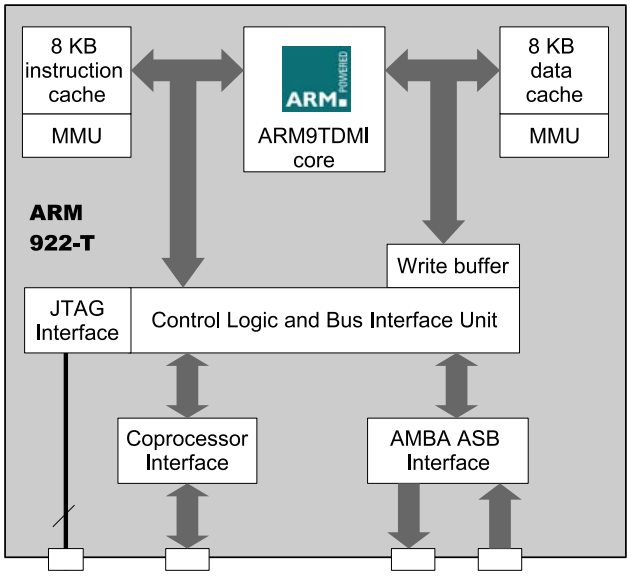
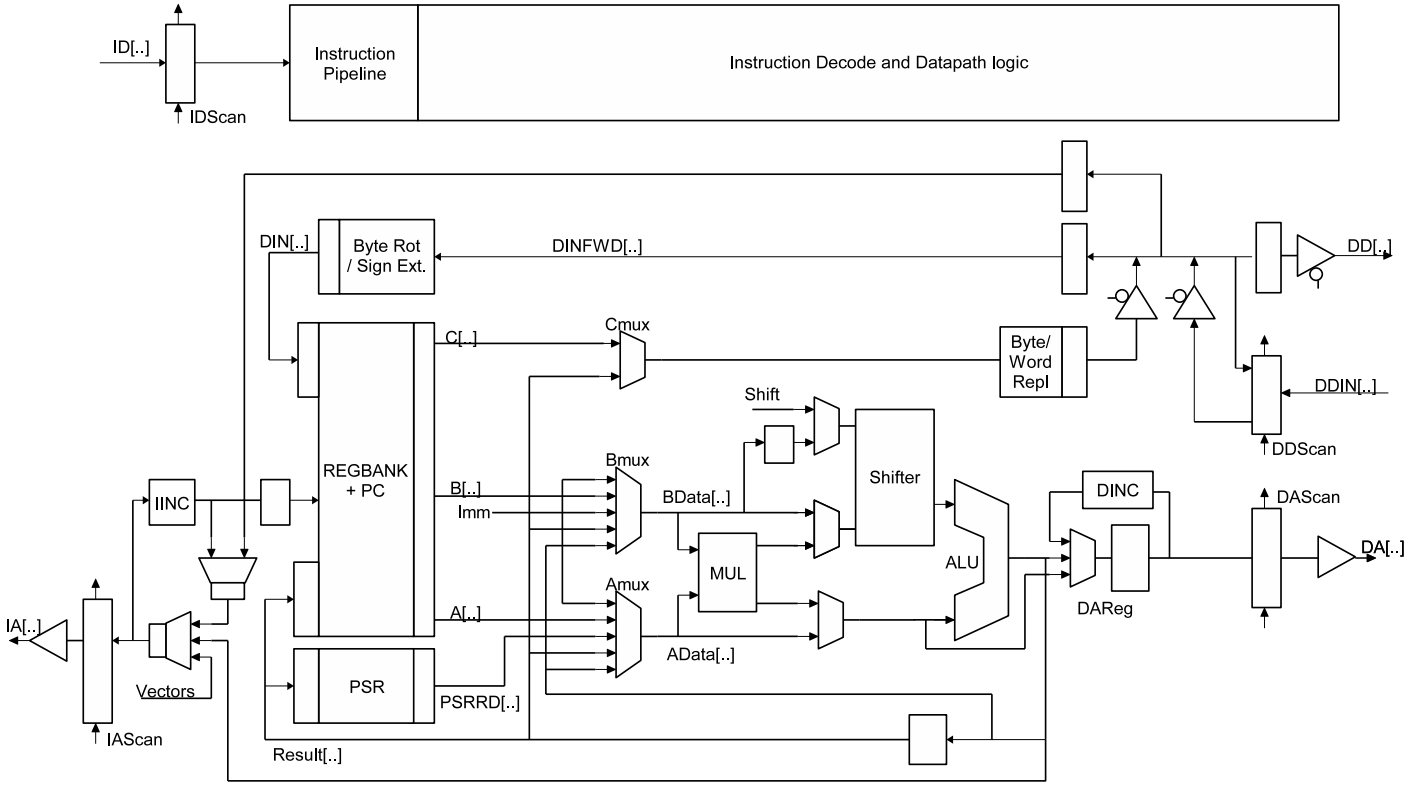


Figure 4 ARM9TDMI Functional Block Diagram



5.1.3 ARM922T Details

The ARM922T macrocell is based on the ARM9TDMI Harvard architecture processor core with an efficient five-stage pipeline. To reduce the effect of memory bandwidth and latency on performance, the ARM922T macrocell includes separate cachs and MMUs for both instructions and data. It also has a write buffer and physical address TAG RAM.

Caches

Two 8KB caches are implemented, one for instructions, the other for data, both with an 8-word line size. Separate buses connect each cache to the ARM9TDMI core permitting a 32 bit instruction to be fetched and fed into the Decode stage of the pipeline at the same time as a 32 bit data access for the memory stage of the pipeline.

Cache lock-down is provided to permit critical code sequences to be locked into the cache to ensure predictability for real-time code. The cache replacement algorithm can be selected by the operating system as either pseudo-random or round-robin. Both caches are 64-way set-associative. Lock-down operates on a per-way basis.

Write Buffer

The ARM922T macrocell also incorporates a 16-data, 4-address write buffer to avoid stalling the processor when writes to external memory are performed.

PA TAG RAM

The ARM922T macrocell implements a physical address TAG RAM (PA TAG RAM) to perform write-backs from the data cache.

The physical addresses of all the lines held in the data cache are stored by the PA TAG memory, removing the requirement for address translation when evicting a line from the cache.

MMU

The ARM922T macrocell implements an enhanced ARMv4 MMU to provide translation and access permission checks for the instruction and data address ports of the ARM9TDMI core.

The MMU features are:

- Standard ARMv4 MMU mapping sizes, domains, and access protection scheme
- Mapping sizes are 1 MB sections, 64 KB large pages, 4 KB small pages, and new 1KB tiny pages
- Access permissions for sections
- Access permissions for large pages and small pages can be specified separately for each quarter of the page (subpages)
- Access permissions for tiny pages
- 16 domains implemented in hardware
- 64-entry instruction Translation-Lookaside-Buffer (TLB) and 64-entry data TLB
- Hardware page table walks
- Round-robin replacement algorithm (also called cyclic)

Control Coprocessor (CP15)

The control coprocessor is provided for configuration of the caches, the write buffer, and other ARM922T options.

Eleven registers are available for program control:

- Register 1 controls system operation parameters including endianness, cache, and MMU enable
- Register 2 and 3 configure and control MMU functions
- Register 5 and 6 provide MMU status information
- Register 7 and 9 are used for cache maintenance operations
- Register 8 and 10 are used for MMU maintenance operations
- Register 13 is used for fast context switching
- Register 15 is used for test.

Debug Features

The ARM9TDMI processor core incorporates an EmbeddedICE unit and EmbeddedICE-RT logic permitting both software tasks and external debug hardware to

- Set hardware and software breakpoints
- Perform single-stepping
- Enable access to registers and memory

This functionality is implemented as a coprocessor and is accessible from hardware through the JTAG port.

Full-speed, real-time execution of the processor is maintained until a breakpoint is hit.

At this point control is passed either to a software handler or to JTAG control.

5.1.4 ARM V4T Architecture

The ARM9TDMI processor core implements the ARMv4T Instruction Set Architecture (ISA). The ARMv4T ISA is a superset of the ARMv4 ISA with additional support for the Thumb 16-bit compressed instruction set.

Performance and Code Density

The ARM9TDMI core executes two instruction sets

- 32-bit ARM instruction set
- 16-bit Thumb instruction set

The ARM instruction set is designed so that a program can achieve maximum performance with the minimum number of instructions. Most ARM9TDMI instructions are executed in a single cycle.

The simpler Thumb instruction set offers much increased code density deducing code size and memory requirement.

Code can switch between the ARM and Thumb instruction sets on any procedure call.

ARM9TDMI Integer Pipeline Stages

The integer pipeline consists of five stages to maximize instruction throughput in the ARM9TDMI core:

- Fetch
- Decode and register read
- Execute shift and ALU operation, or address calculate, or multiply
- Memory access and multiply
- Write register

By using a five-stage pipeline, the ARM922T delivers a throughput approaching one instruction per cycle.

Registers

The ARM9TDMI processor core consists of a 32-bit datapath and associated conrol logic. This datapath contains 31 general-purpose registers, coupled to a full shifter, Arithmetic Logic Unit, and a multiplier. At any one time 16 registers are visible to the user. The remainder are mode-specific replacement registers (banked registers) used to speed up execution processing, and make nested exceptions possible.

Register 15 is the Program Counter (PC) that can be used in all instructions to reference data relative to the current instruction. R14 holds the return address after a subroutine call. R13 is used (by software convention) as a stack pointer.

Exeption Types/Modes

The ARM9TDMI core supports five types of exception, and a privileged processing mode for each type. The types of exceptions are:

- Fast interrupt (FIQ)
- Normal interrupt (IRQ)
- Memory aborts (used to implement memory protection or virtual memory)
- Attempted execution of an undefined instruction
- Software interrupts (SWIs)

All exceptions have banked registers for R14 and R13. After an exception, R14 holds the return address for exception processing. This address is used both to return after the exception is processed and to address the instruction that caused the exception.

R13 is banked across exception modes to provide each exception handler with a private stack pointer. The fast interrupt mode also banks registers 8 to 12 so that interrupt processing can begin without the need to save or restore these registers.

A seventh processing mode, System mode, uses the User mode registers. System mode runs tasks that require a privileged processor mode and enables them to invoke all classes of exceptions.

Status Registers

All other processor states are held in status registers. The current operating processor status is in the Current Program Status Register (CPSR). The CPSR holds:

- Four ALU flags (Negative, Zero, Carry, Overflow)
- An interrupt disable bit for each of the IRQ and FIQ interrupts
- A bit to indicate ARM or Thumb execution state
- Five bits to encode the current processor mode

All five exception modes also have a Saved Program Status Register (SPSR) that holds the CPSR of the task immediately before the exception occurred.

Conditional Execution

All ARM instructions can be executed conditionally and can optionally update the four condition code flags (Negative, Zero, Carry, and Overflow) according to their result. Fifteen conditions are implemented.

Classes of Instructions

The ARM and Thumb instruction sets can be divided into four broad classes of instruction:

- Data processing instructions
- Load and store instructions
- Branch instructions
- Coprocessor instructions

Data Processing Instructions

The data processing instructions operate on data held in general-purpose registers. Of the two source operands, one is always a register. The other has two basic forms:

- An immediate value
- A register value optionally shifted

If the operand is a shifted register, the shift can be an immediate value or the value of another register. Four types of shift can be specified. Most data processing instructions can perform a shift followed by a logical or arithmetic operation.

There are two classes of multiply instructions:

- Normal, 32 bit result
- Long, 64 bi resut variants.

Both types of multiply instruction can optionally perform an accumulate operation

Load and Store Instructions

There are two main types of laod and store instructions:

- Load or store the value of a single register
- Load or store multiple register values

Load and store single register instructions can transfer a 32-bit word, a 16-bit halfword, or an 8-bit byte between memory and a register. Byte and halfword loads can be automatically zero extended or sign extended as they are loaded. These instructions have three primary addressing modes:

- Offset
- Pre-indexed
- Post-indexed

The address is formed by adding an immediate, or register-based, positive, or negative offset to a base register. Register-based offsets can also be scaled with shift operations. Pre-indexed and post-indexed addressing modes update the base registers with the base plus offset calculation.

As the PC is a general-purpose register, a 32-bit balue can be loaded directly into the PC to perform a jump to any address in the 4GB memory space.

Load and store multiple instructions perform a block transfer of any number of the general purpose registers to, or from, memory. Four addressing modes are provided:

- Pre-increment addressing
- Post-increment addressing
- Pre-decrement addressing
- Post-decrement addressing

The base address is specified by a register value (that can be optionally updated after the transfer). As the subroutine return address and the PC values are in general-purpose registers, very efficient subroutine calls can be constructed.

Branch Instructions

As well as letting data processing or load instructions change control flow (by writing the PC) a standard branch instruction is provided with 24-bit signed offset, providing for forward and backward branches of up to 32 MB.

A branch with link (BL) instruction enables efficient subroutine calls. BL preserves the address of the instruction after the branch in R14 (Link register or LR). This lets a move instruction put the LR in to the PC and return to the instruction after the branch.

The branch and exchange (BX) instruction switches between ARM and Thumb instruction sets with the return address optionally preserving the operating mode of the calling subroutine.

Coprocessor Instructions

There are three types of coprocessor instructions:

- Coprocessor data processing instructions
- Coprocessor register transfer instructions
- Coprocessor data transfer instructions

5.1.5 JTAG Interface

The ARM933T debug interface is based on IEEE Std. 1149.1- 1990, standard test access port. The ARM922T contains hardware extensions for advanced debugging features. These are intended to ease the development of application software.

The debug extensions allow the core to be stopped by one of the following:

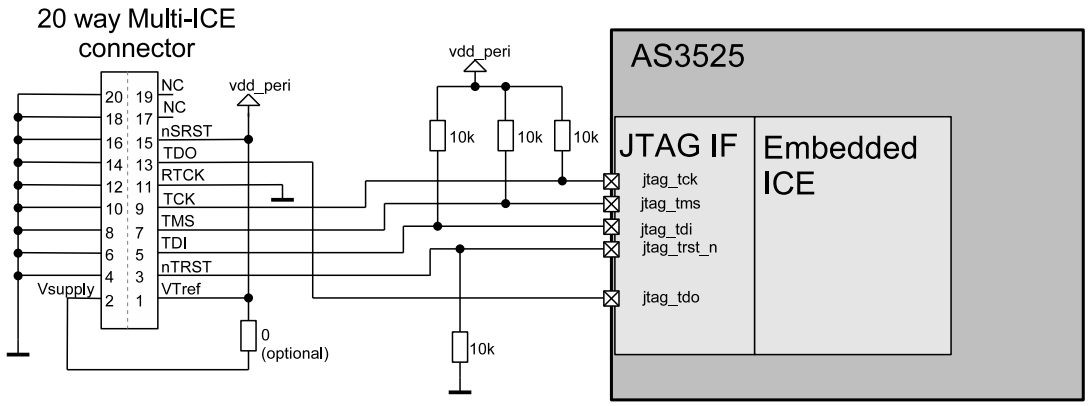
- A given instruction fetch (breakpoint)
- A data access (watchpoint)
- Asynchronously by a debug request

When this happens, the ARM922T is said to be in debug state. At this point, you can examine the internal state of the core and the external state of the system. When examination is complete, you can restore the core and system state and resume program execution.

Normally, all control for debugging is done by running a debugger software (ARM AXD or ARM Realview Debugger) on a debug host PC. Connection to the chip is done by an ARM Multi-ICE interface, which connects either to the parallel port or the USB port of the debug host PC.

The connection to the multi-ICE interface is done via a 20 way connector and ribbon cable. Following diagram shows the signals connections to the ICE connector.

Figure 5 Interface connector to multi-ICE



5.1.6 Boot Concept

It can be selected if the system should boot either using the internal ROM (internal boot loader) or an external ROM/Flash (connected to the MPMC interface). XPC[0] is read within global chip reset to do the selection of either internal or external boot.

Table 8 Boot definitions for internal/external boot selection

XPC[0]	Booting Option
1	Internal ROM
0	External ROM/Flash

For the internal bootloader, two chip versions are available: C21 and C22. Version C22 has additional features and is fully backward compatible to C21.

5.1.6.1 Internal Bootloader Version C21

Within the internal ROM boot loader several options for booting can be selected:

- SSP IF - SPI master for ST serial flash types
- SSP IF - SPI slave
- NandFlash
- Debug UART diagnostics

All boot loader options of the internal bootloader are configured by XPC[3:1] pins. External pull-up or pull-down resistors should be used to configure the boot options.

Table 9 Boot definitions Chip version C21

XPC[3:1]		Boot Device
0	000	SPI master ST M25Pxx serial Nor Flash
1	001	reserved
2	010	SPI slave
3	011	NandFlash (SB/BB - autodetect)
4	100	NandFlash (SB/BB - autodetect)
5	101	UART / Command Line Interface without diagnostics
6	110	UART / Command Line Interface without diagnostics
7	111	UART / Command Line Interface with diagnostics

5.1.6.2 Internal Bootloader Version C22

For chip version C22 the boot loader is extended with two additional features

- IDE boot: direct boot from harddisk
- USB boot promer. In the case that a USB connection is present and either an update button is pressed or there is no bootable device, the USB promer is started (see Figure 6 Boot decision between normal boot and USB boot promer" for details). The USB boot promer allows update of the firmware by using an USB mass storage class device. This update can be used either for initial programming (factory programming) or as mechanism for an in-field firmware update.

The C22 boot loader is fully compatible to the C21 boot loader except for mode 4, where the previous NAF boot mode is replaced by IDE boot. For version C22, NAF boot is only available in mode 3. Refer to Table 10 Boot definitions Chip version C22" for details.

The update button is located between xpa[4] and xpa[0]. Within the key scan routine, xpa[4] is driven shortly to each logic level "0" and "1" and the value of xpa[0] is read back to sense a keypress of the update button.

For the USB promer, it is necessary that frequency settings defining the quarz crystal frequency are defined by the pins xpc[3:1]. For details refer to "Table 11 USB promer frequency settings". These settings are read at the beginning in the initialisation routine of the bootloader.

Figure 6 Boot decision between normal boot and USB boot promer

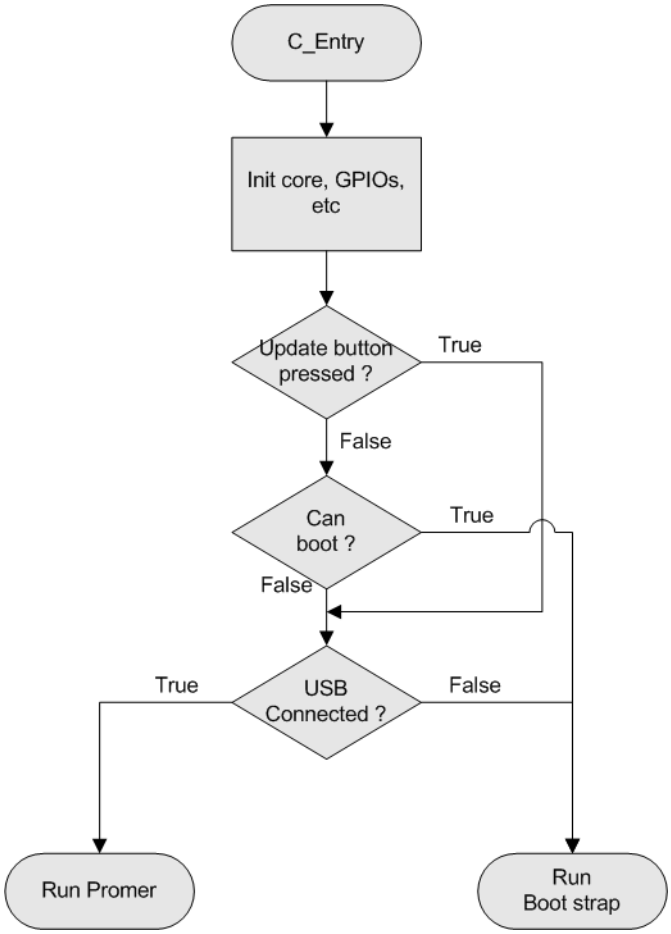


Table 10 Boot definitions Chip version C22

XPC[3:1]		Boot Device
0	000	SPI master ST M25Pxx serial Nor Flash
1	001	SPI master Atmel AT45DB011B serial Nor Flash
2	010	SPI slave
3	011	NandFlash
4	100	IDE
5	101	reserved for developers mode
6	110	UART / Command Line Interface without diagnostics
7	111	UART / Command Line Interface with diagnostics

Table 11 USB promer frequency settings

XPA[6:4]	USB promer frequency settings
000	24 MHz
001	20 MHz
010	13 MHz
011	12 MHz
100	10 MHz
others	reserved / defaults to 24 MHz

5.2 AHB Peripheral Blocks

ARM AHB ("advanced high-performance bus") is the new generation of AMBA bus, which is intended to address the requirements of high-performance synthesizable designs. AMBA AHB implements the features required for high performance, high clock frequency systems including:

- burst transfers
- split transactions
- single cycle bus master handover
- non-tristate implementation
- 32 bit bus width
- the clock frequency of the AHB can set by software up to 65MHz

5.2.1 2.5 MBIT RAM Main Memory

The memory subsystem consists of a RAM part and a ROM part.

Within the RAM memory subsystem, following functions are included:

- 1-TRAM controller with AHB bus slave interface
- 1-TRAM memory macros

5.2.1.1 1-TRAM Controller

The 1T RAM Controller is a slave interface connected to the AMBA AHB bus.

- slave AHB interface
- supports byte(8 bit), half-word(16 bit) and word(32 bit) read/write accesses
- 128-bit Line Buffer as temporary storage to reduce the number of memory accesses and optimise power consumption
- controls 5TSMC 1T-RAM instances

5.2.1.2 On-Chip 1T-RAM macro blocks

TSMC Emb1tRAM™ technology is a special kind of DRAM, which is implemented in a logic CMOS process. This innovative concept and design guarantees lowest power, high density, high performance and high yield advantages.

ECC (Error Correction Code) technique is applied in the macro to dynamically correct errors caused by hard defects or soft errors. No fuses are needed because the conventional redundancy scheme is replaced with ECC design in the macro.

The macro can be operated at clock rate from 20 MHz up to maximum AHB bus clock frequency in flow through random access mode. In the product, one idle cycle for refresh is needed in every 32 clock cycles.

Total 5 macros with organisation of 4Kx128 = 64 KByte each are implemented. For the refresh, one master macro is generating the refresh clock (T1F4Kx128_PIFE) and four macros are connected serially in slave mode to the refresh clock (T1F4Kx128PIFES).

Features

- 20 Mhz to 65 Mhz operation speed
- Flow through random access
- Built-in error correction (ECC)
- 128-bit wide data bus
- Separated data in/out bus
- SRAM-style interface operation
- Built-in refresh controller with refresh clock generator

5.2.2 On-Chip ROM

5.2.2.1 ROM Controller

The ROM controller implements the AHB slave interface for accessing the ROM.

The ROM controller generates OK response for all reads and error response for all writes.

Access width is always 32 bits.

5.2.2.2 1MBIT ROM

128 KByte of on-chip mask-programmable ROM are included.

The ROM is metal mask programmable by a single mask change (VIA2).

The ROM contains the following firmware package

- Boot loader

5.2.2.3 ROM versions and chip versions

There are two versions of the chip with changed Bootloader functionality available

- Version C21: Bootloader supports basic function for boot from external Nor Flash (ST or ATMEL).
- Version C22: Bootloader supports extended boot functions

These two chip versions differ only in the ROM content.

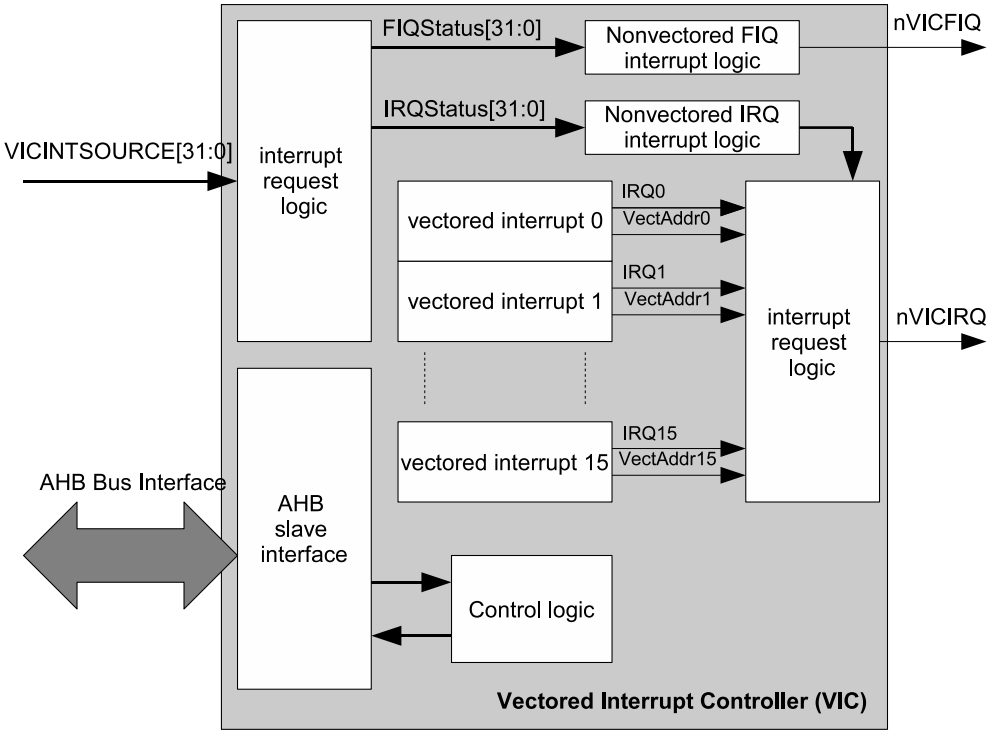
5.2.3 VIC – Vectored Interrupt Controller

The ARM PrimeCell™ PL190 “vectored interrupt controller” is included in the AHB system.

5.2.3.1 Features

- AMBA specification Rev 2.0 compliant
- support for 32 standard interrupts
- support for 16 vectored interrupts
- hardware interrupt priority
- IRQ and FIQ generation
- AHB mapped for fast interrupt response
- software interrupt generation
- test registers
- raw interrupt status
- interrupt request status
- interrupt masking
- privileged mode supportBlock Diagram

Figure 7 VIC Block Diagram



5.2.3.2 VIC Interrupt Sources

Table 12 VIC Interrupt Sources

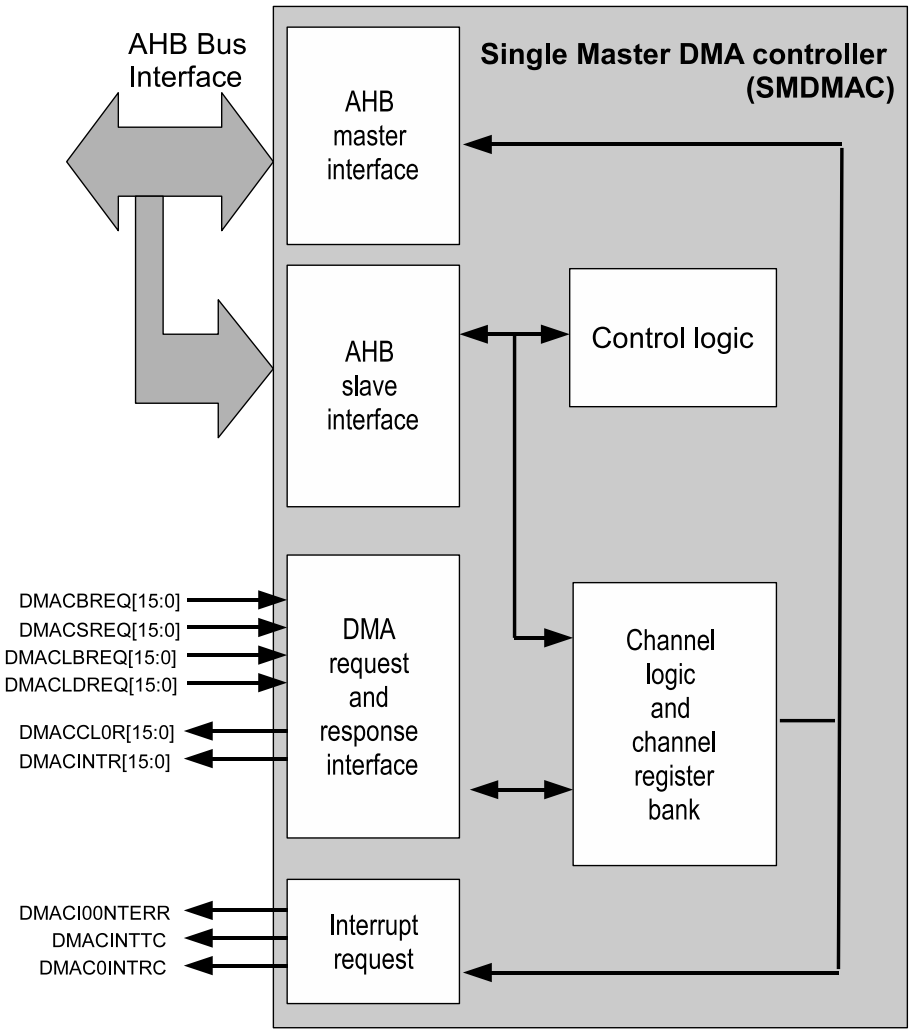
IRQ Source	Module	IRQ Source	Module
0	Watchdog	16	GPIO4 (XPD)
1	Timer 1	17	-
2	Timer 2	18	CGU
3	USB	19	Memory Stick
4	DMAC	20	DBOP
5	Nand Flash	21	-
6	IDE	22	-
7	MCI INTR0	23	-
8	MCI INTR1	24	-
9	AUDIO IRQ	25	-
10	SSP	26	-
11	I2C MS	27	-
12	I2C Audio	28	-
13	I2SIN	29	GPIO1 (XPA)
14	I2SOUT	30	GPIO2 (XPB)
15	UART	31	GPIO3 (XPC)

5.2.4 SMDMAC - Single master DMAC

The ARM PrimeCell™ PL081 “SMDMAC single master DMA controller” is included in the AHB system.

- AMBA specification Rev 2.0 compliant
- two DMA channels. Each channel can support a unidirectional transfer
- provides 16 peripheral DMA request lines
- single DMA and burst DMA request signals. Each peripheral connected to the PrimeCell™ SMDMAC can assert either a burst DMA request or a single DMA request. The DMA burst size is set by programming the PrimeCell™ SMDMAC
- Memory-to-Memory, memory-to-peripheral, peripheral-to-memory and peripheral-to-peripheral transfers.
- Scatter or gather DMA is supported through the use of linked lists. This means that the source and destination areas do not need to occupy contiguous areas of memory
- Hardware DMA channel priority. Each DMA channel has a specific hardware priority. DMA channel 0 has the highest priority and channel 1 has the lowest priority. If requests from two channels become active at the same time the channel with the highest priority is serviced first.
- AHB slave DMA programming interface. The PrimeCell™ SMDMAC is programmed by writing to the DMA control registers over the AHB slave interface
- One AHB bus master for transferring data. This interface is used to transfer data when a DMA request goes active.

Figure 8 SMDMAC Block Diagram



5.2.4.1 DMAC Registers

Table 13 DMAC Registers

Register Name	Base Address	Offset	Note
DMAC_IntStatus	AS3525_DMAC_BASE	0x000	Interrupt status register
DMAC_IntTCStatus	AS3525_DMAC_BASE	0x004	Interrupt terminal count status register
DMAC_IntTCClear	AS3525_DMAC_BASE	0x008	Interrupt terminal count clear register
DMAC_IntErrorStatus	AS3525_DMAC_BASE	0x00C	Interrupt error status register
DMAC_IntErrorClear	AS3525_DMAC_BASE	0x010	Interrupt error clear register
DMAC_RawIntTCStatus	AS3525_DMAC_BASE	0x014	Raw interrupt terminal count status register
DMAC_RawIntErrorStatus	AS3525_DMAC_BASE	0x018	Raw interrupt error status register
DMAC_SoftBReq	AS3525_DMAC_BASE	0x020	Software burst request register
DMAC_SoftSReq	AS3525_DMAC_BASE	0x024	Software single request register
DMAC_SoftLBReq	AS3525_DMAC_BASE	0x028	Software last burst request register
DMAC_SoftSBReq	AS3525_DMAC_BASE	0x02C	Software last single request register
DMAC_Configuration	AS3525_DMAC_BASE	0x030	Configuration register
DMAC_Sync	AS3525_DMAC_BASE	0x034	Synchronisation register
DMAC_C0SrcAddr	AS3525_DMAC_BASE	0x100	Channel 0 source address
DMAC_C0DestAddr	AS3525_DMAC_BASE	0x104	Channel 0 destination address
DMAC_C0LLI	AS3525_DMAC_BASE	0x108	Channel 0 linked list item register
DMAC_C0Control	AS3525_DMAC_BASE	0x10C	Channel 0 control register
DMAC_C0Configuration	AS3525_DMAC_BASE	0x110	Channel 0 configuration register
DMAC_C1SrcAddr	AS3525_DMAC_BASE	0x120	Channel 1 source address
DMAC_C1DestAddr	AS3525_DMAC_BASE	0x124	Channel 1 destination address
DMAC_C1LLI	AS3525_DMAC_BASE	0x128	Channel 1 linked list item register
DMAC_C1Control	AS3525_DMAC_BASE	0x12C	Channel 1 control register
DMAC_C1Configuration	AS3525_DMAC_BASE	0x130	Channel 1 configuration register
DMAC_PeripheralId0	AS3525_DMAC_BASE	0xFE0	peripheral ID0 register
DMAC_PeripheralId1	AS3525_DMAC_BASE	0xFE4	peripheral ID1 register
DMAC_PeripheralId2	AS3525_DMAC_BASE	0xFE8	peripheral ID2 register
DMAC_PeripheralId3	AS3525_DMAC_BASE	0xFEC	peripheral ID3 register
DMAC_CellId0	AS3525_DMAC_BASE	0xFF0	peripheral cell ID0 register
DMAC_CellId1	AS3525_DMAC_BASE	0xFF4	peripheral cell ID1 register
DMAC_CellId2	AS3525_DMAC_BASE	0xFF8	peripheral cell ID2 register
DMAC_CellId3	AS3525_DMAC_BASE	0xFFC	peripheral cell ID3 register

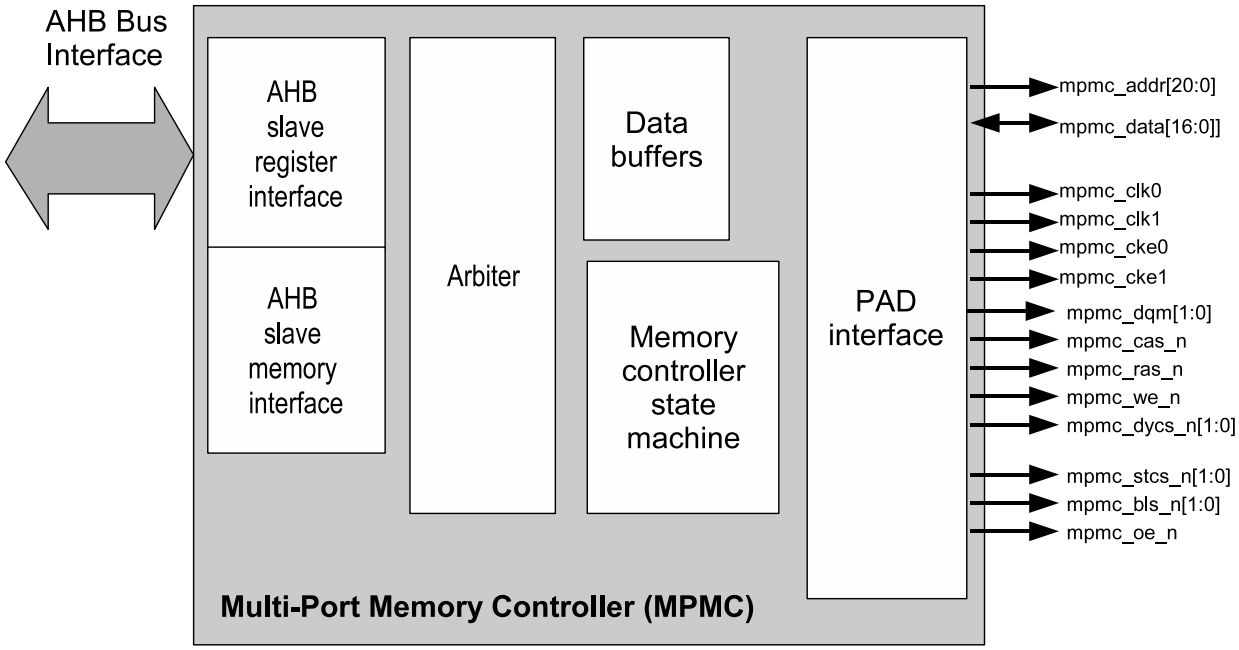
5.2.5 Multi Port Memory Controller (MPMC)

The MPMC block is integrated into the AMBA system through AHB slave port.

The PrimeCell™ MPMC offers:

- AMBA 32-bit AHB compliance.
- Dynamic memory interface support including SDRAM and JEDEC low-power SDRAM
- Asynchronous static memory device support including RAM, ROM, and Flash, with or without asynchronous page mode.
- Low transaction latency.
- Read and write buffers to reduce latency and to improve performance.
- Single AHB interface for accessing external memory.
- 8-bit and 16-bit wide static memory support.
- 16-bit wide chip select SDRAM memory support.
- Static memory features include:
  - asynchronous page mode read
  - programmable wait states
  - bus turnaround delay
  - output enable, and write enable delays
  - extended wait
- Two chip selects for synchronous memory and two chip selects for static memory devices.
- Software controllable HCLK to MPMCCLKOUT ratio.
- Power-saving modes dynamically control SDRAM MPMCCKEOUT and MPMCCLKOUT.
- Dynamic memory self-refresh mode supported by software.
- Controller supports 2K, 4K, and 8K row address synchronous memory parts. That is typical 512MB, 256MB, 128MB, and 16Mb parts, with 8, 16 bits per device.
- Two reset domains enable dynamic memory contents to be preserved over a soft reset.
- A separate AHB interface to program the MPMC. This enables the PrimeCell™ MPMC registers to be situated in memory with other system peripheral registers.
- Locked AHB transactions supported.
- Support for all AHB burst types.

Figure 9 Multi Port Memory Controller Block Diagram



5.2.6 IDE Interface

The IDE host interface core provides an efficient and easy-to-use interface to IDE and ATAPI devices. The core implements programmable I/O, Multi-word DMA, and Ultra ATA-33, -66, -100 and -133 modes of operation and supports up to two devices. The core interface to the system-on-chip provides PIO access and DMA capability to optimise data transfers to and from the IDE devices. For ease of integration, this interface includes a register set compatible with the Intel chip set, including a descriptor-based scatter-gather DMA core. This core is compatible with ATA-4 with Ultra ATA-33, -66, -100 and -133 extensions. Single-word DMA is not supported.
The licensed SpeedSelect™ technology allows the core to be reconfigured to support any timing mode for PIO, Multi-Word DMA, and Ultra ATA transfers (-33, -66, -100 or -133) while running at any clock frequency. Interface to the host processor is the AMBA AHB bus architecture.

There are two AHB interfaces on the core: an AHB master and an AHB slave.

5.2.6.1 AHB Master Interface

The AHB Master implements a subset of the AHB protocol. The following features are supported:

- Single transfer, unspecified-length, 4-beat incrementing and optionally 8-beat incrementing bursts (HBURST will be ‘000’, ‘001’, ‘011’, or optionally ‘101’)
- Accesses that cross a 1kB boundary will be unspecified-length incrementing (HBURST will be ‘001’)
- 16-bit and 32-bit transfers only (HSIZE will only be ‘001’ or ‘010’)
- BUSY cycles are not issued (HTRANS will not be ‘01’)
- HPROT is not implemented
- OKAY, SPLIT and RETRY responses accepted (HRESP may be ‘00’, ‘10’ or ’11’)
- HLOCK asserted during fixed-length bursts
- The AHB master may be granted by default

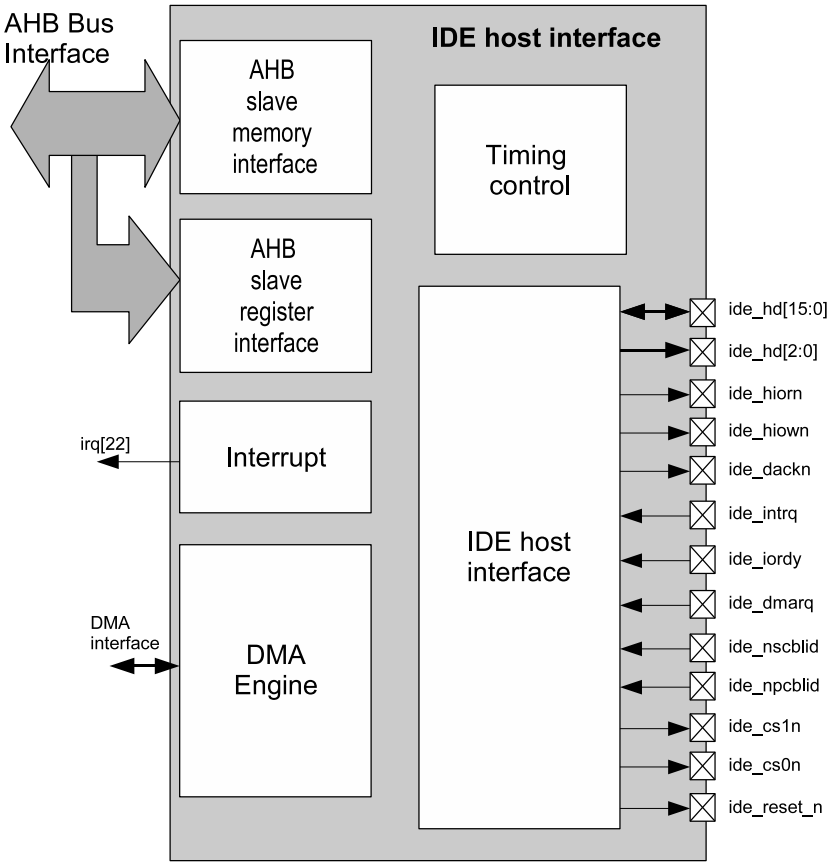
5.2.6.2 AHB Slave Interface

The AHB Slave implements a subset of the AHB protocol. The following features are supported:

- Non-burst only (HBURST must be ‘000’)
- 8-, 16-, or 32-bit transfers only (HSIZE must be ‘000’, ‘001’ or ‘010’)
- No advantage is gained by issuing a SEQ cycle over a NONSEQ cycle (HTRANS values of ‘10’ and ‘11’ are interpreted identically)
- HPROT is ignored
- HRESP is ‘00’ (OKAY)
- HREADY is issued no sooner than 2 clock cycles after a valid SEQ or NONSEQ cycle
- The AHB slave may be selected by default

5.2.6.3 IDE Block diagram

Figure 10 IDE Block Diagram



5.2.6.4 IDE Interface Registers

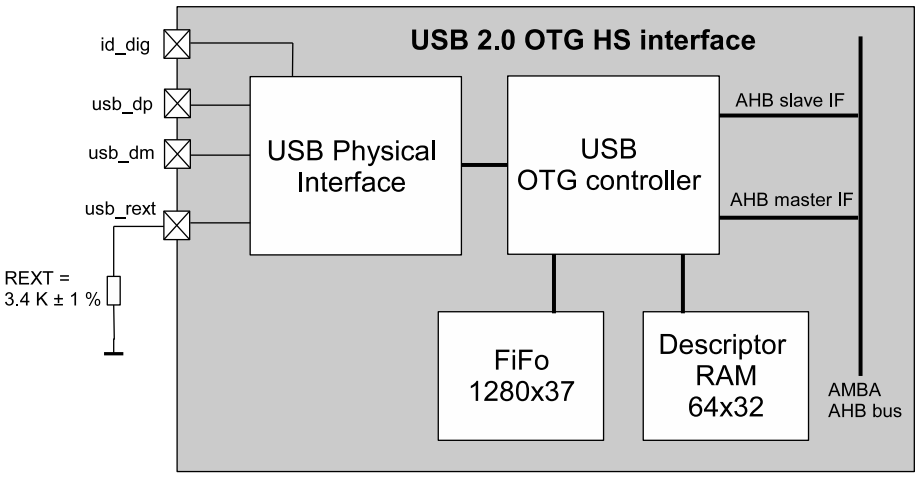
Table 14 IDE Interface Registers

Register Name	Base Address	Offset	Note
IdeReg_BMICP	AS3525_CF_IDE_BASE	0x00	primary channel bus master command
IdeReg_BMISP	AS3525_CF_IDE_BASE	0x02	primary channel bus master status
IdeReg_BMIDTPP_LO	AS3525_CF_IDE_BASE	0x04	primary channel bus master table pointer
IdeReg_BMIDTPP_HI	AS3525_CF_IDE_BASE	0x06	
IdeReg_IDETIMP_LO	AS3525_CF_IDE_BASE	0x40	primary channel timing register
IdeReg_IDETIMP_HI	AS3525_CF_IDE_BASE	0x41	
IdeReg_IDETIMS_LO	AS3525_CF_IDE_BASE	0x42	secondary channel timing register
IdeReg_IDETIMS_HI	AS3525_CF_IDE_BASE	0x43	
IdeReg_SIDETIM	AS3525_CF_IDE_BASE	0x44	slave IDE timing register
IdeReg_SLEWCTL_LO	AS3525_CF_IDE_BASE	0x45	slew rate control register
IdeReg_SLEWCTL_HI	AS3525_CF_IDE_BASE	0x46	
IdeReg_IDESTAT	AS3525_CF_IDE_BASE	0x47	IDE status register
IdeReg_UDMACTL	AS3525_CF_IDE_BASE	0x48	ultra DMA control register
IdeReg_UDMATIM_LO	AS3525_CF_IDE_BASE	0x4A	ultra DMA timing register
IdeReg_UDMATIM_HI	AS3525_CF_IDE_BASE	0x4B	
IdeReg_MISCCTL	AS3525_CF_IDE_BASE	0x50	miscellaneous control register
IdeReg_REGSTB	AS3525_CF_IDE_BASE	0x54	task file register strobe timing register
IdeReg_REGRCVR	AS3525_CF_IDE_BASE	0x58	task file register recovery timing register
IdeReg_DATSTB	AS3525_CF_IDE_BASE	0x5C	data register PIO strobe timing register
IdeReg_DATRCVR	AS3525_CF_IDE_BASE	0x60	data register PIO recovery timing register
IdeReg_DMASTB	AS3525_CF_IDE_BASE	0x64	DMA strobe timing register
IdeReg_DMARCVR	AS3525_CF_IDE_BASE	0x68	DMA recovery timing register
IdeReg_UDMASTB	AS3525_CF_IDE_BASE	0x6C	ultra DMA strobe timing register
IdeReg_UDMATRP	AS3525_CF_IDE_BASE	0x70	ultra DMA ready-to-stop timing register
IdeReg_UDMATENV	AS3525_CF_IDE_BASE	0x74	ultra DMA timing envelope register
IdeReg_IORDYTMP	AS3525_CF_IDE_BASE	0x78	primary IO ready timer configuration reg
IdeReg_IORDYTMS	AS3525_CF_IDE_BASE	0x7C	secondary IO ready timer configuration reg
IdeTaskF_DATA	AS3525_CF_IDE_BASE	0x1F0	
IdeTaskF_ERR_FEAT	AS3525_CF_IDE_BASE	0x1F1	
IdeTaskF_SECT_CNT	AS3525_CF_IDE_BASE	0x1F2	
IdeTaskF_SECT_NUM	AS3525_CF_IDE_BASE	0x1F3	
IdeTaskF_CYL_LO	AS3525_CF_IDE_BASE	0x1F4	
IdeTaskF_CYL_HI	AS3525_CF_IDE_BASE	0x1F5	
IdeTaskF_DEV_HEAD	AS3525_CF_IDE_BASE	0x1F6	
IdeTaskF_STAT_CMD	AS3525_CF_IDE_BASE	0x1F7	
IdeTaskF_ALT_STAT_DEV_CTRL	AS3525_CF_IDE_BASE	0x3F6	
IdeTaskF_DEV_ADDR	AS3525_CF_IDE_BASE	0x3F7	

5.2.7 USB 2.0 HS OTG interface

The USB 2.0 on-chip interface includes the USB 2.0 On-The-Go Physical Interface and the HS OTG controller.

Figure 11 USB 2.0 Interface



5.2.7.1 HS OTG controller subsystem

The Synopsys HS OTG subsystem is a configurable design. The HS OTG subsystem is fully compliant with the On-The-Go supplement to the USB 2.0 specification, Revision 1.0a. The subsystem supports high speed (480-Mbps) and full-speed transfers. It is designed to interface to the AMBA AHB bus, shielding the application from the complexities of the HS OTG subsystem-native protocols and simplifying the system interface.

The OTG subsystem can be configured using application software as follows:

- OTG dual-role device (DRD)
- OTG device only
- OTG mini host only
- USB High-Speed (HS) device
- USB HS mini host
- USB Full-Speed (FS) device

The HS OTG subsystem has the following interfaces

- the UTMI+, which connect the on-chip PHY to the HS OTG core
- the AHB slave interface, which provides the microcontroller with read and write access to the core's control and status register (CSRs)
- the AHB master interface, which enables the core to act as a master on the AHB to transfer data to and from the core's DMA controller
- the descriptor prefetch buffer RAM interface, which connects to an single-port RAM for DMA descriptor prefetch buffer storage
- the data RAM interface, which connects to and dual-port RAM (FIFO memory) for transaction data storage

General features

- handles all clock synchronisation within the core
- uses a descriptor prefetch buffer for optimal AHB use in host mode
- supports adaptive buffering for dynamic FIFO memory allocation, avoiding gaps in RAM utilisation
- SOFs are supported in high/full speed modes
- includes built-in DMA
- includes hardware transaction scheduling for enhanced performance
- supports memory mapped address space for the CSRs

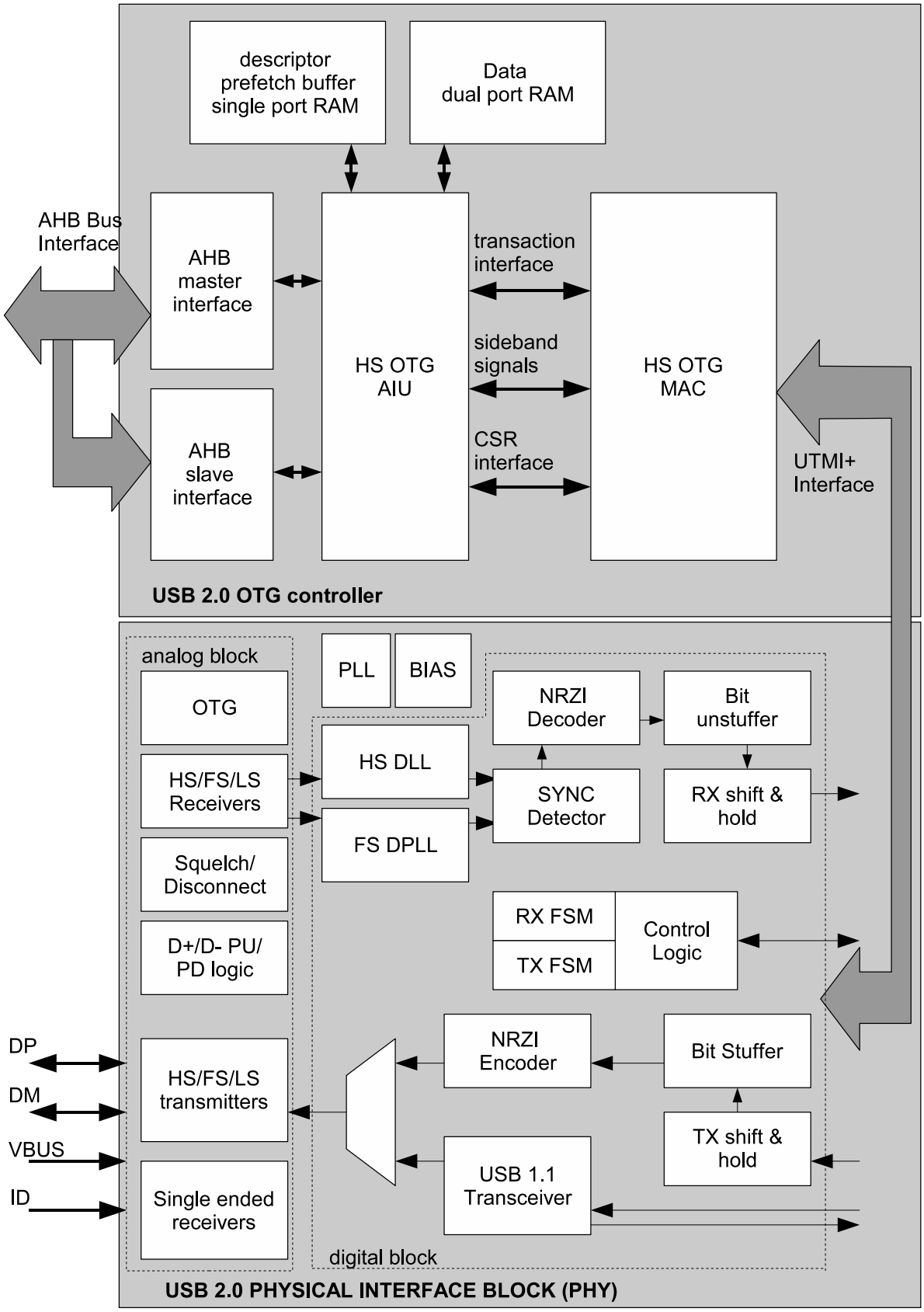
USB 2.0 supported features

- supports up to 15 configurations in Device mode
  - each configuration supports 15 interfaces
  - each interface handles up to 15 alternate settings
- supports session request protocol (SRP)
- supports session request protocol (SRP)
- supports Host Negotiation Protocol (HNP)
- recovers clock and data from the USB
- supports a generic root hub
- includes auto ping/split completion capabilities
- complies with UTMI+ level 3 interface

Implemented Controller configurations are:

- configured with 4 host channels and 3 bidirectional- plus 1 in-endpoints in device mode
- dynamic alternate configuration selection (for different bandwidths of isochronous endpoints)

Figure 12 USB 2.0 OTG Controller Block Diagram



5.2.7.2 USB 2.0 OTG PHY

- Complete PHY for USB2.0 On-The-Go
- USB 2.0 UTMI+ specification compliant
- Supports high speed (480 Mbit/s), full speed (12 Mbit/s) and low speed (1.5 Mbit/s) data transmission
- Supports OT supplement features: VBUS state detecting SRP request by “data-line pulsing” method
- Low jitter clock from either on-chip PLL (48MHz) or optional additional crystal (12MHz, 24MHz or 48MHz) which is available with the 224 pin package, only
- 16 bit parallel datain/out interface
- Typical current consumption on vdda33c and vdd33t:
  - 12 mA in FS RX mode
  - 30 mA in FS TX mode
  - 30 mA in HS RX mode
  - 40 mA in HS TX mode
  - < 100 uA in suspend mode
- Rext = 3.4kOhm (+/- 1%) must be connected between pads “rext” and “vssa33c” to set the bias current.

5.2.7.3 USB 2.0 OTG Interface Registers

Table 15 USB Interface Registers

Register Name	Base Address	Offset	Note
USB_IEP0_CTRL	AS3525_USB_BASE	0x00000	Control Register
USB_IEP0_STS	AS3525_USB_BASE	0x00004	Status Register
USB_IEP0_TXFSIZE	AS3525_USB_BASE	0x00008	TxFIFO Size
USB_IEP0_MPS	AS3525_USB_BASE	0x0000c	Maximum Packet Size
USB_IEP0_DESC_PTR	AS3525_USB_BASE	0x00014	Data Descriptor Pointer
USB_IEP0_STS_MASK	AS3525_USB_BASE	0x00018	Status Mask Register
USB_IEP1_CTRL	AS3525_USB_BASE	0x00020	Control Register
USB_IEP1_STS	AS3525_USB_BASE	0x00024	Status Register
USB_IEP1_TXFSIZE	AS3525_USB_BASE	0x00028	TxFIFO Size
USB_IEP1_MPS	AS3525_USB_BASE	0x0002c	Maximum Packet Size
USB_IEP1_DESC_PTR	AS3525_USB_BASE	0x00034	Data Descriptor Pointer
USB_IEP1_STS_MASK	AS3525_USB_BASE	0x00038	Status Mask Register
USB_IEP2_CTRL	AS3525_USB_BASE	0x00040	Control Register
USB_IEP2_STS	AS3525_USB_BASE	0x00044	Status Register
USB_IEP2_TXFSIZE	AS3525_USB_BASE	0x00048	TxFIFO Size
USB_IEP2_MPS	AS3525_USB_BASE	0x0004c	Maximum Packet Size
USB_IEP2_DESC_PTR	AS3525_USB_BASE	0x00054	Data Descriptor Pointer
USB_IEP2_STS_MASK	AS3525_USB_BASE	0x00058	Status Mask Register
USB_IEP3_CTRL	AS3525_USB_BASE	0x00060	Control Register
USB_IEP3_STS	AS3525_USB_BASE	0x00064	Status Register
USB_IEP3_TXFSIZE	AS3525_USB_BASE	0x00068	TxFIFO Size
USB_IEP3_MPS	AS3525_USB_BASE	0x0006c	Maximum Packet Size
USB_IEP3_DESC_PTR	AS3525_USB_BASE	0x00074	Data Descriptor Pointer
USB_IEP3_STS_MASK	AS3525_USB_BASE	0x00078	Status Mask Register
USB_OEP0_CTRL	AS3525_USB_BASE	0x00200	Control
USB_OEP0_STS	AS3525_USB_BASE	0x00204	Status Register
USB_OEP0_RXFR	AS3525_USB_BASE	0x00208	Rx Packet Frame Number Register
USB_OEP0_MPS	AS3525_USB_BASE	0x0020c	RxFIFO Size/Maximum Packet Size
USB_OEP0_SUP_PTR	AS3525_USB_BASE	0x00210	Setup buffer Pointer Register
USB_OEP0_DESC_PTR	AS3525_USB_BASE	0x00214	Data Descriptor Pointer
USB_OEP0_STS_MASK	AS3525_USB_BASE	0x00218	Status Mask Register
USB_OEP1_CTRL	AS3525_USB_BASE	0x00220	Control Register
USB_OEP1_STS	AS3525_USB_BASE	0x00224	Status Register
USB_OEP1_RXFR	AS3525_USB_BASE	0x00228	Rx Packet Frame Number Register

Register Name	Base Address	Offset	Note
USB_OEP1_MPS	AS3525_USB_BASE	0x0022c	RxFIFO Size/Maximum Packet Size
USB_OEP1_SUP_PTR	AS3525_USB_BASE	0x00230	Setup buffer Pointer Register
USB_OEP1_DESC_PTR	AS3525_USB_BASE	0x00234	Data Descriptor Pointer
USB_OEP1_STS_MASK	AS3525_USB_BASE	0x00238	Status Mask Register
USB_OEP2_CTRL	AS3525_USB_BASE	0x00240	Control Register
USB_OEP2_STS	AS3525_USB_BASE	0x00244	Status Register
USB_OEP2_RXFR	AS3525_USB_BASE	0x00248	Rx Packet Frame Number Register
USB_OEP2_MPS	AS3525_USB_BASE	0x0024c	RxFIFO Size/Maximum Packet Size
USB_OEP2_SUP_PTR	AS3525_USB_BASE	0x00250	Setup buffer Pointer Register
USB_OEP2_DESC_PTR	AS3525_USB_BASE	0x00254	Data Descriptor Pointer
USB_OEP2_STS_MASK	AS3525_USB_BASE	0x00258	Status Mask Register
USB_OEP3_CTRL	AS3525_USB_BASE	0x00260	Control Register
USB_OEP3_STS	AS3525_USB_BASE	0x00264	Status Register
USB_OEP3_RXFR	AS3525_USB_BASE	0x00268	Rx Packet Frame Number Register
USB_OEP3_MPS	AS3525_USB_BASE	0x0026c	RxFIFO Size/Maximum Packet Size
USB_OEP3_SUP_PTR	AS3525_USB_BASE	0x00270	Setup buffer Pointer Register
USB_OEP3_DESC_PTR	AS3525_USB_BASE	0x00274	Data Descriptor Pointer
USB_OEP3_STS_MASK	AS3525_USB_BASE	0x00278	Status Mask Register
USB_DEV_CFG	AS3525_USB_BASE	0x00400	Device Configuration Register
USB_DEV_CTRL	AS3525_USB_BASE	0x00404	Device Control Register
USB_DEV_STS	AS3525_USB_BASE	0x00408	Device Status Register
USB_DEV_INTR	AS3525_USB_BASE	0x0040c	Device Interrupt Register
USB_DEV_INTR_MASK	AS3525_USB_BASE	0x00410	Device Interrupt Mask Register
USB_DEV_EP_INTR	AS3525_USB_BASE	0x00414	Device Endpoint Interrupt
USB_DEV_EP_INTR_MASK	AS3525_USB_BASE	0x00418	Device Endpoint Interrupt Mask
USB_PHY_EP0_INFO	AS3525_USB_BASE	0x00504	Information Register
USB_PHY_EP1_INFO	AS3525_USB_BASE	0x00508	Information Register
USB_PHY_EP2_INFO	AS3525_USB_BASE	0x0050c	Information Register
USB_PHY_EP3_INFO	AS3525_USB_BASE	0x00510	Information Register
USB_PHY_EP4_INFO	AS3525_USB_BASE	0x00514	Information Register
USB_PHY_EP5_INFO	AS3525_USB_BASE	0x00518	Information Register
USB_HOST_CH0_SPLT	AS3525_USB_BASE	0x01000	Split Information Register
USB_HOST_CH0_STS	AS3525_USB_BASE	0x01004	Status Register
USB_HOST_CH0_TXFSIZE	AS3525_USB_BASE	0x01008	TxFIFO Register
USB_HOST_CH0_REQ	AS3525_USB_BASE	0x0100c	Request Register
USB_HOST_CH0_PER_INFO	AS3525_USB_BASE	0x01010	Periodic/Split Transaction Information Register
USB_HOST_CH0_DESC_PTR	AS3525_USB_BASE	0x01014	Data Descriptor Pointer
USB_HOST_CH0_STS_MASK	AS3525_USB_BASE	0x01018	Status Mask Register
USB_HOST_CH1_SPLT	AS3525_USB_BASE	0x01020	Split Information Register
USB_HOST_CH1_STS	AS3525_USB_BASE	0x01024	Status Register
USB_HOST_CH1_TXFSIZE	AS3525_USB_BASE	0x01028	TxFIFO Register
USB_HOST_CH1_REQ	AS3525_USB_BASE	0x0102c	Request Register
USB_HOST_CH1_PER_INFO	AS3525_USB_BASE	0x01030	Periodic/Split Transaction Information Register
USB_HOST_CH1_DESC_PTR	AS3525_USB_BASE	0x01034	Data Descriptor Pointer
USB_HOST_CH1_STS_MASK	AS3525_USB_BASE	0x01038	Status Mask Register
USB_HOST_CH2_SPLT	AS3525_USB_BASE	0x01040	Split Information Register
USB_HOST_CH2_STS	AS3525_USB_BASE	0x01044	Status Register
USB_HOST_CH2_TXFSIZE	AS3525_USB_BASE	0x01048	TxFIFO Register
USB_HOST_CH2_REQ	AS3525_USB_BASE	0x0104c	Request Register
USB_HOST_CH2_PER_INFO	AS3525_USB_BASE	0x01050	Periodic/Split Transaction Information

Register Name	Base Address	Offset	Note
			Register
USB_HOST_CH2_DESC_PTR	AS3525_USB_BASE	0x01054	Data Descriptor Pointer
USB_HOST_CH2_STS_MASK	AS3525_USB_BASE	0x01058	Status Mask Register
USB_HOST_CH3_SPLT	AS3525_USB_BASE	0x01060	Split Information Register
USB_HOST_CH3_STS	AS3525_USB_BASE	0x01064	Status Register
USB_HOST_CH3_TXFSIZE	AS3525_USB_BASE	0x01068	TxFIFO Register
USB_HOST_CH3_REQ	AS3525_USB_BASE	0x0106c	Request Register
USB_HOST_CH3_PER_INFO	AS3525_USB_BASE	0x01070	Periodic/Split Transaction Information Register
USB_HOST_CH3_DESC_PTR	AS3525_USB_BASE	0x01074	Data Descriptor Pointer
USB_HOST_CH3_STS_MASK	AS3525_USB_BASE	0x01078	Status Mask Register
USB_HOST_CFG	AS3525_USB_BASE	0x01400	Host Configuration Register
USB_HOST_CTRL	AS3525_USB_BASE	0x01404	Host Control Register
USB_HOST_INTR	AS3525_USB_BASE	0x0140c	Host Interrupt Register
USB_HOST_INTR_MASK	AS3525_USB_BASE	0x01410	Host Interrupt Mask Register
USB_HOST_CH_INTR	AS3525_USB_BASE	0x01414	Host Channel Interrupt Register
USB_HOST_CH_INTR_MASK	AS3525_USB_BASE	0x01418	Host Channel Interrupt Mask Register
USB_HOST_FRAME_INT	AS3525_USB_BASE	0x0141c	Host Frame Interval Register
USB_HOST_FRAME_REM	AS3525_USB_BASE	0x01420	Host Frame Remaining Register
USB_HOST_FRAME_NUM	AS3525_USB_BASE	0x01424	Host Frame Number Register
USB_HOST_PORT0_CTRL_STS	AS3525_USB_BASE	0x01500	Host Port and Status Register
USB_OTG_CSR	AS3525_USB_BASE	0x02000	OTG Control and Status Register
USB_I2C_CSR	AS3525_USB_BASE	0x02004	I2C Access Register
USB_GPIO_CSR	AS3525_USB_BASE	0x02008	General Purpose Input/Output Register
USB_SNPSID_CSR	AS3525_USB_BASE	0x0200c	Synopsys ID Register
USB_USERID_CSR	AS3525_USB_BASE	0x02010	User ID Register
USB_USER_CONF1	AS3525_USB_BASE	0x02014	User Config1 Register
USB_USER_CONF2	AS3525_USB_BASE	0x02018	User Config2 Register
USB_USER_CONF3	AS3525_USB_BASE	0x0201c	User Config3 Register
USB_USER_CONF4	AS3525_USB_BASE	0x02020	User Config4 Register
USB_USER_CONF5	AS3525_USB_BASE	0x02024	User Config5 Register

5.2.8 Memory Stick / Memory Stick Pro Interface

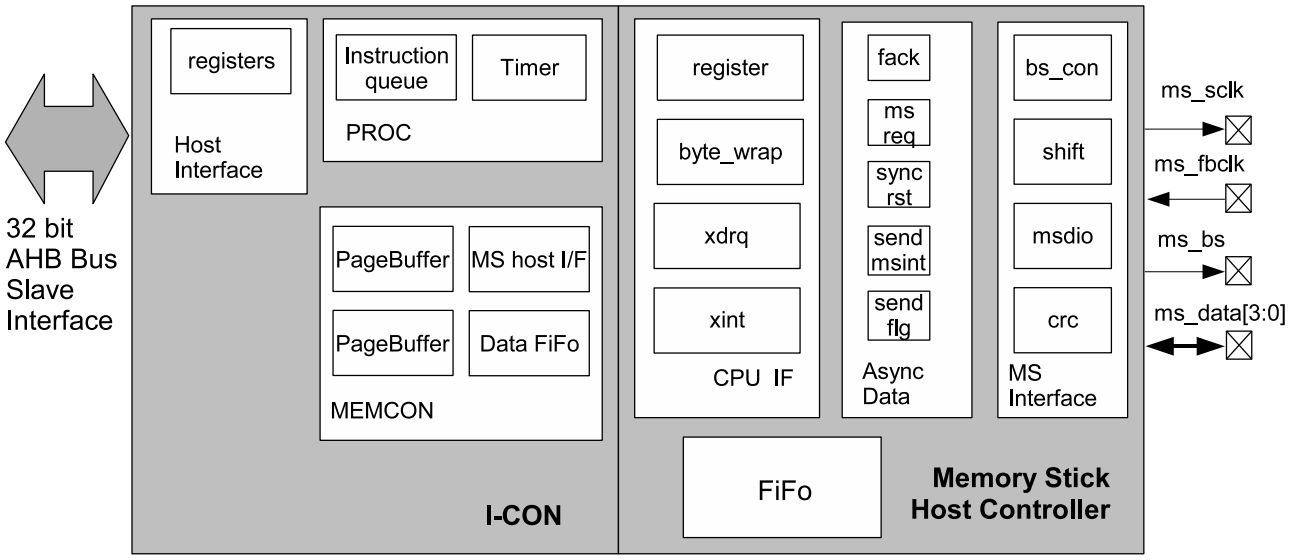
The Sony memory stick interface is an AHB bus slave device. This interface conforms to following standards:

- Memory Stick Standard Format Specifications version 1.4-00
- Memory Stick PRO Format Specifications version 1.00-01

5.2.8.1 Block Diagram

The memory stick interface contains two main blocks, the ICON and the host controller.

Figure 13 SONY memory stick interface block diagram



5.2.8.2 I-CON

This IP is Memory Stick / Memory Stick PRO Host Controller automatic control IP with a 32-bit CPU interface. This IP automatically controls the series of TPC-based communication with the Memory Stick in place of the CPU, and aims to reduce the burden on the Host CPU.

The contents of communication with the Memory Stick are designated in this IP by micro codes.

Features

- 32-bit CPU interface
- Inside controller specified by microcodes
- Buffer for two-way data transmission loaded (256 byte x 2)
- 32/16 bit access available
- DMA support
- General-purpose data transmit/receive FIFO (12 Bytes)

5.2.8.3 Host Controller

Features

- Memory Stick and Memory Stick PRO support
- FiFo memory (64 bits × 4) for two-way data transmission
- Built-in CRC circuit
- Memory Stick serial clock (Serial: 20 MHz (max.), Parallel: 40 MHz (max.))
- DMA support
- 16/32/64-bit access possible

5.2.8.4 Functional description

Communication with the Memory Stick

The communication protocol with the Memory Stick is started by write from the CPU to the command register. When the protocol finishes, the CPU is notified that the protocol has ended by an interrupt request.

Data transfer request

When the protocol is started and enters the data transfer state, data is requested by issuing a DMA transfer request or an interrupt request to the CPU. Data can also be requested to an external memory.

Memory Stick communication time out

The RDY time out time when the handshake state (read protocol: BS2, write protocol: BS3) is established in communication with the Memory Stick can be designated as the number of Memory Stick transfer clocks. When a time out occurs, the CPU is notified that the protocol has ended due to a time out error by an interrupt request.

CRC off

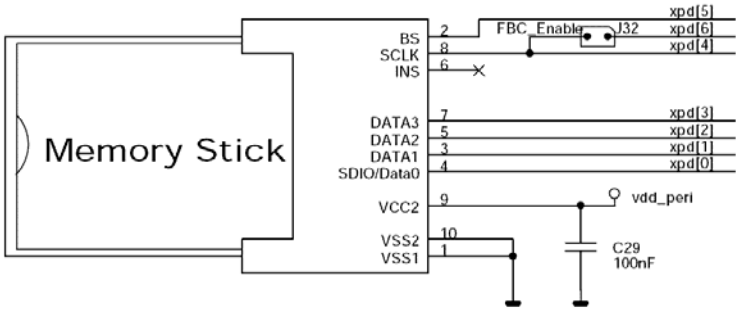
CRC off can be set as a test mode.

When CRC off is set, CRC is not added to the data transmitted to the Memory Stick.

PAD cells

The connections to the MemoryStick Interface are shared with the General Purpose I/O port-D (GPIO xpd[0:7]).

Figure 14 external memory stick connection



5.3 APB Peripheral Block

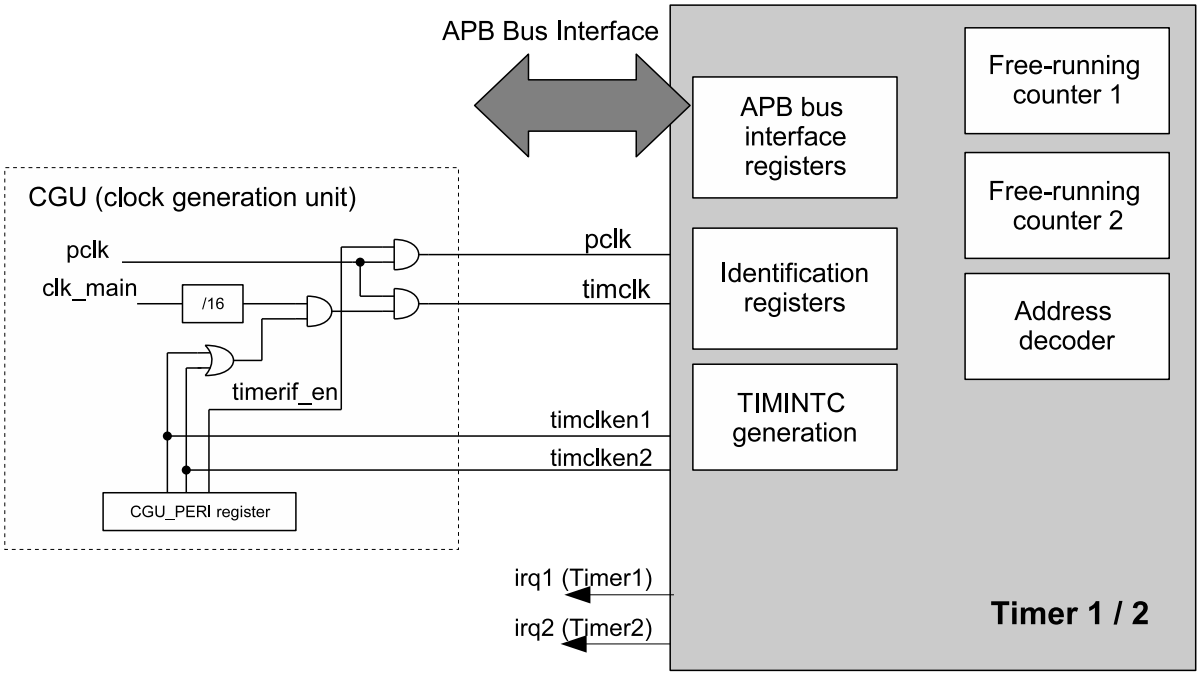
5.3.1 Timers

The Dual Input Timers module is an APB slave that provides access to two interrupt-generating, programmable 32-bit free-running decrementing counters (FRCs). The system clock (PCLK) is used to control the programmable registers, and the second clock input is used to drive the counter, enabling the counters to run from a much slower clock than the system clock. This input clock of the counters (TIMCLK) is connected to a clock derived (divided by 16) from the main clock (clk_main) signal. That clock clk_main is always running and is coming from the internal or external oscillator (set by clk_sel pad).

5.3.1.1 Timer modes

- Free-running mode: the counter wraps after zero and continues at the maximum value. This is the default mode.
- Periodic mode: reload of original value after wrapping past zero.
- One-shot mode - interrupt is generated once, counter halts after reaching zero

Figure 15 Timer Block Diagram



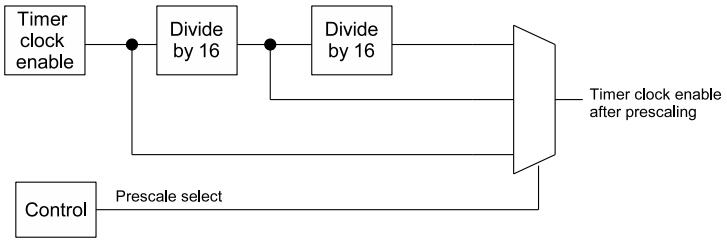
Each timer has an identical set of registers shown in table Table 16. The operation of each timer is identical. The timer is loaded by writing to the load register and, if enabled, counts down to zero. When a counter is already running, writing to the load register will cause the counter to immediately restart at the new value. Writing to the background load value has no effect on the current count. The counter continues to decrement to zero, and then recommences from the new load value (if in periodic mode, and one shot mode is not selected).

When zero is reached, an interrupt is generated. The interrupt can be cleared by writing to the clear register. If One Shot Mode is selected, the counter halts on reaching zero One Shot Mode is deselected, or a new load value is written. Otherwise, after reaching a zero count, if the timer is operating in free-running mode it continues to decrement from its maximum value. If periodic timer mode is selected, the timer reloads the count value from the load register and continues to decrement. In this mode the counter effectively generates a periodic interrupt. The mode is selected by a bit in the timer control register. At any point, the current counter value can be read from the value register. The counter is enabled by a bit in the control register. At reset, the counter is disabled, the interrupt is cleared, and the load register is set to zero. The mode and prescale values are set to free-running, and clock divide of 1 respectively.

The timer clock enable is generated by a prescale unit. The enable is then used by the counter to create a clock with a timing of one of the following.

- The system clock
- The system clock divided by 16, generated by 4 bits of prescale
- The system clock divided by 256, generated by a total of 8 bits of prescale

Figure 16 - timer prescaler



5.3.1.2 Interrupt generation

An interrupt is generated when the full 32-bit counter reaches zero, and is only cleared when the TimerXClear location is written to. A register holds the value until the interrupt is cleared. The most significant carry bit of the counter detects the counter reaching zero.

Interrupts can be masked by writing 0 to the interrupt enable bit in the control register. Both the raw interrupt satus (prior to masking) and the final interrupt status (after masking) can be read from status registers.

Timer 1 interrupt output is connected to interrupt input line irq1 (VIC input) and Timer 2 interrupt output is connected to interrupt line irq2.

5.3.1.3 Timer Register Descriptions

Table 16 – Timer 1 and 2 registers

Register Name	Base Address	Offset	Note
Timer1Load	AS3525_TIMER_BASE	0x00	load value for Timer 1
Timer1Value	AS3525_TIMER_BASE	0x04	current value for Timer 1
Timer1Control	AS3525_TIMER_BASE	0x08	Timer 1 control register
Timer1IntClr	AS3525_TIMER_BASE	0x0C	Timer 1 interrupt clear
Timer1RIS	AS3525_TIMER_BASE	0x10	Timer 1 raw interrupt status
Timer1MIS	AS3525_TIMER_BASE	0x14	Timer 1 masked interrupt status
Timer1BGLoad	AS3525_TIMER_BASE	0x18	Timer 1 background load value
Timer2Load	AS3525_TIMER_BASE	0x20	load value for Timer 2
Timer2Value	AS3525_TIMER_BASE	0x24	current value for Timer 2
Timer2Control	AS3525_TIMER_BASE	0x28	Timer 2 control register
Timer2IntClr	AS3525_TIMER_BASE	0x2C	Timer 2 interrupt clear
Timer2RIS	AS3525_TIMER_BASE	0x30	Timer 2 raw interrupt status
Timer2MIS	AS3525_TIMER_BASE	0x34	Timer 2 masked interrupt status
Timer2BGLoad	AS3525_TIMER_BASE	0x38	Timer 2 background load value
Periheral ID register bits 7:0	AS3525_TIMER_BASE	0xFE0	Peripheral ID register bits 7:0
Periheral ID register bits 15:8	AS3525_TIMER_BASE	0xFE4	Peripheral ID register bits 15:8
Periheral ID register bits 23:16	AS3525_TIMER_BASE	0xFE8	Peripheral ID register bits 23:16
Periheral ID register bits 31:24	AS3525_TIMER_BASE	0xFEC	Peripheral ID register bits 31:24
Primecell ID register bits 7:0	AS3525_TIMER_BASE	0xFF0	Primecell ID register bits 7:0
Primecell ID register bits 15:8	AS3525_TIMER_BASE	0xFF4	Primecell ID register bits 15:8
Primecell ID register bits 23:16	AS3525_TIMER_BASE	0xFF8	Primecell ID register bits 23:16
Primecell ID register bits 31:24	AS3525_TIMER_BASE	0xFFC	Primecell ID register bits 31:24

Load register, Timer1Load, Timer2Load

This is a 32-bit register containing the value from which the counter is to decrement. This is the value used to reload the counter when periodic mode is enabled, and the current count reaches zero.

When this register is written to directly, the current count is immediately reset to the new value at the next rising edge of TIMCLK which is enabled by TIMCLKEN.

The value in this register is also overwritten if the TimerXBGLoad register is written to, but the current count is not immediately affected.

If values are written to both the timerXLoad and TimerXBGLoad registers before an enabled rising edge on TIMCLK, the following occurs:

- On the next enabled TIMCLK edge the value written to the TimerXLoad value replaces the current count value
- Following this, eacht time the counter reaches zero, the current count value is reset to the value written to TimerXBGLoad.

Reading from the TimerXLoad register at any time after the two writes have occurred will retrieve the value written to TimerXBGLoad. That is, the value read from TimerXLoad is always the value which will take effect for periodic mode after the next time the counter reaches zero.

Current value register, Timer1Value, Timer2Value

This register gives the current value of the decrementing counter.

Timer control register

Table 17 Timer control register

Name		Base		Default
Timer1Control, Timer2Control		**AS3525_TIMER_BASE**		**0x20**
Offset: 0x08, 0x28		**Timer Control Register**		
		Contains control bits of the PLLA register.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	Timer Enable	0	R/W	Enable bit: 0: timer disabled (default) 1: timer enabled
6	Timer Mode	0	R/W	Mode bit 0: timer is in free-running mode (default) 1: timer is in periodic mode
5	Interrupt Enable	1	R/W	Interrupt enable bit 0: timer interrupt disabled 1: timer interrupt enabled (default)
4	RESERVED			Reserved bit, do not modify, and ignore on read
3:2	TimerPre	00	R/W	Prescale bits: 00: no prescale, clock is divided by 1 (default) 01: 4 stages of prescale, clock is divided by 16 10: 8 stages of prescale, clock is divided by 256 11: undefined, do not use
1	Timer Size	0	R/W	Selects 16/32 bit counter operation 0: 16 bit counter (default) 1: 32 bit counter
0	OneShotCount	0	R/W	Selects one-shot or wrapping counter mode 0: wrapping mode (default) 1: one-shot mode

Interrupt clear register, Timer1IntClr, Timer2IntClr

Any write to this register will clear the interrupt output from the counter

Raw Interrupt status register, Timer1RIS, Timer2RIS

This register indicates the raw interrupt status from the counter. This value is ANDed with the timer interrupt enable bit from the control register to create the masked interrupt, which is passed to the interrupt output pin.

Table 18 raw interrupt status register

Name		Base		Default
Timer1RIS, Timer2RIS		**AS3525_TIMER_BASE**		
Offset: 0x10, 0x30		**Timer raw interrupt status register**		
		Contains control bits of the PLLA register.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
0	Raw Timer Interrupt		R	Raw interrupt status from the counter

Interrupt status register, TIMERXMIS

This register indicates the masked interrupt status from the counter. This value is the logical AND of the raw interrupt status with the timer interrupt enable bit from the control register, and is the same value which is passed to the interrupt output pin.

Table 19 interrupt status register

Name		Base		Default
Timer1MIS, Timer2MIS		**AS3525_TIMER_BASE**		
Offset: 0x10, 0x30		**Timer raw interrupt status register**		
		Contains control bits of the PLLA register.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
0	Raw Timer Interrupt		R	Raw interrupt status from the counter

Background load register, TimerXBGLoad

This is a 32 bit register containing the value from which the counter is to decrement. This is the value used to reload the counter when periodic mode is enabled, and the current count reaches zero.

This register privides an alternative method of accessing the TimerXLoad register. The difference is that writes to TimerXBGLoad will not cause the counter immediately to restart from the new value.

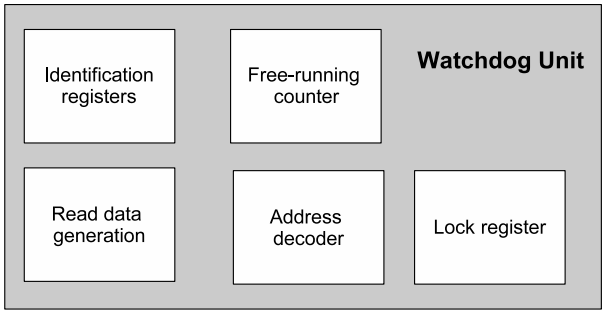
Reading from this register returns the same value returned from TimerXLoad.

5.3.2 Watchdog Unit

The watchdog unit provides a way of recovering from software crashes. The watchdog clock is used to generate a regular interrupt (WDOGINT), depending on a programmed value. The watchdog monitors the interrupt and asserts a reset signal (WDOGRES) if the interrupt remains unserviced for the entire programmed period. You can enable or disable the watchdog unit as required.

Clock reference for the watchdog is PCLK divided by 256.

Figure 17 watchdog unit



5.3.2.1 Watchdog register descriptions

Table 20 Watchdog Registers

Register Name	Base Address	Offset	Note
WDT_LOAD	AS3525_WDT_BASE	0x00	load register
WDT_VALUE	AS3525_WDT_BASE	0x04	counter current value
WDT_CONTROL	AS3525_WDT_BASE	0x08	control register
WDT_INTCLR	AS3525_WDT_BASE	0x0C	Interrupt clear register
WDT_RIS	AS3525_WDT_BASE	0x10	Raw interrupt status register
WDT_MIS	AS3525_WDT_BASE	0x14	Masked interrupt status register
WDT_LOCK	AS3525_WDT_BASE	0xC00	Lock register
WDT_PERIPHID0	AS3525_WDT_BASE	0xFE0	Watchdog peripheral ID 0 register
WDT_PERIPHID1	AS3525_WDT_BASE	0xFE4	Watchdog peripheral ID 1 register
WDT_PERIPHID2	AS3525_WDT_BASE	0xFE8	Watchdog peripheral ID 2 register
WDT_PERIPHID3	AS3525_WDT_BASE	0xFEC	Watchdog peripheral ID 3 register
WDT_PCELLID0	AS3525_WDT_BASE	0xFF0	Watchdog primecell ID 0 register
WDT_PCELLID1	AS3525_WDT_BASE	0xFF4	Watchdog primecell ID 1 register
WDT_PCELLID2	AS3525_WDT_BASE	0xFF8	Watchdog primecell ID 2 register
WDT_PCELLID3	AS3525_WDT_BASE	0xFFC	Watchdog primecell ID 3 register

Watchdog load register, WdogLoad

This is a 32-bit register containing the value from which the counter is to decrement. When this register is written to, the count is immediately restarted from the new value. The minimum valid value for WdogLoad is one.

Watchdog control register, WdogControl

This is a read/write register that enables the software to control the watchdog unit.

Table 21 watchdog control register

Name			Base		Default
WdogControl			**AS3525_WDT_BASE**		**0x04**
Offset: 0x08		**Watchdog Control Register**			
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**	
1	RESEN	0	R/W	Enable Watchdog reset output (WDOGRES). Acts as a mask for the reset output. 0: disable the reset 1: enable the reset	
0	INTEN	0	R/W	Enable the interrupt event (WDOGINT). 0: disable the counter and interrupt 1: enable the counter and interrupt	

Watchdog clear interrupt register, WdogIntClr

A write of any value to this location clears the watchdog interrupt, and reloads the counter from the value in WdogLoad.

Raw interrupt status register, WdogRIS

This register indicates the raw interrupt status from the counter. This value is ANDed with the inerrupt enable bit from the control register to create the masked interrupt, which is passed to the interrupt output pin.

Table 22 watchdog raw interrupt status register

Name			Base		Default
WdogRIS			**AS3525_WDT_BASE**		
Offset: 0x10		**Watchdog interrupt status register**			
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**	
0	Watchdog Interrupt		R	Enabled interrupt status from the counter	

Interrupt status register, WdogMIS

This register indicates the masked interrupt status from the counter. This value is the logical AND of the raw interrupt status with the INTEN bit from the control register, and is the same value which is passed to the interrupt output pin.

Name			Base		Default
WdogMIS			**AS3525_WDT_BASE**		
Offset: 0x14		**Watchdog raw interrupt status register**			
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**	
0	Raw Watchdog Interrupt		R	Raw interrupt status from the counter	

Table 23 watchdog interrupt status register

Watchdog lock register, WdogLock

Use of this register allows write-access to all other registers to be disabled. This is to prenent rogue software from disabling the watchdog functionality. Writing a value of 0x1ACCE551 will enable write access to all other registers. Writing any other value will disable write accesses. A read from this register will return only the bottom bit:

- 0 indicates that write access is enabled (not locked)
- 1 indicates that write access is disabled (locked)

Table 24 watchdog lock register

Name			Base		Default
WdogLock			**AS3525_WDT_BASE**		**0x00**
Offset: 0xC00		**Watchdog raw interrupt status register**			
Bit	**Bit Name**	**Default**	**Access**		**Bit Description**
31:0	Enable register writes		W		Enable write access to all other registers by writing 0x1ACCE551. Disable write access by writing any other value.
0	Register write enable status	0	R		0: write access to all other registers is enabled (default) 1: write access to all other registers is disabled

5.3.3 SSP – Synchronous Serial Port

The SSP is a master or slave interface that enables synchronous serial communication with slave or master peripherals having one of the following:

- a Motorola SPI-compatible interface
- a TI synchronous serial interface
- a National Semiconductor MicroWire interface
- In both master and slave configurations the SSP performs
- parallel-to-serial conversion on data written to an internal 16-bit wide, 8-location deep transmit FIFO
- serial-to-parallel conversion on received data, buffering it in a similar 16-bit wide, 8 location-deep receive FIFO

Interrupts are generated to:

- request servicing of the transmit and receive FIFO
- inform the system that a receive FIFO overrun has occurred
- inform the system that data is present in the receive FIFO after an idle period has expired

SSP Features:

- compliant to AMBA Rev 2.0
- master or slave operation
- programmable clock bit rate and prescale
- separate receive and transmit memory buffers each 16 bits wide and 8 bits deep
- programmable data frame size from 4 to 16 bit
- independent masking of receive FIFO, transmit FIFO and receive overrun interrupts
- internal loopback testmode available
- support for DMA
- identification register uniquely identifying the PrimeCell™ itself (support for OS)

SPI features:

- full-duplex, four wire synchronous transfer
- programmable clock polarity and phase

MicroWire features:

- half duplex transfer using 8 bit control message

Texas Instruments SSI features:

- full-duplex, four wire synchronous transfer
- transmit data PIN tristateable when not transmitting

Programmable parameters:

- master or slave mode
- enabling of operation
- frame format
- communication baud rate
- clock phase and polarity
- data width from 4 to 16 bit
- interrupt masking

Figure 18 Serial Synchronous Port Block Diagram

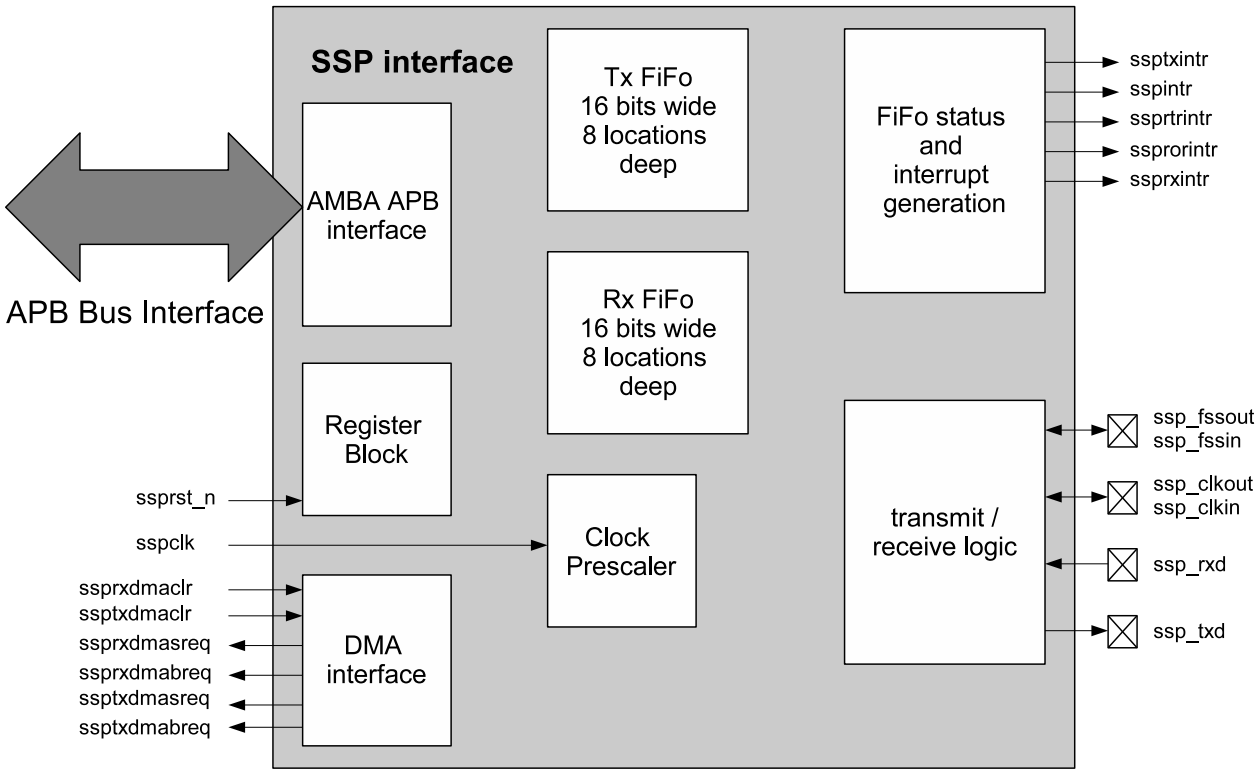


Table 25 SSP Registers

Register Name	Base Address	Offset	Note
SPI_SSPCR0	SSP_BASE	0x00	CR0 control register
SPI_SSPCR1	SSP_BASE	0x04	CR1 control register
SPI_SSPRXD	SSP_BASE	0x08	Read Data Register
SPI_SSPTXD	SSP_BASE	0x08	Write Data register
SPI_SSPSR	SSP_BASE	0x0C	SSP status register
SPI_SSPCPSR	SSP_BASE	0x10	SSP Pre-scaler register
SPI_SSPIMSC	SSP_BASE	0x14	SSP Interrupt Mask and clear register
SPI_SSPIRS	SSP_BASE	0x18	SSP Raw interrupt status register
SPI_SSPMIS	SSP_BASE	0x1C	SSP Masked interrupt status register
SPI_SSPICR	SSP_BASE	0x20	SSP interrupt clear register
SPI_SSPDMACR	SSP_BASE	0x24	SSP DMA control register

5.3.4 GPIO - General purpose input/output ports

The ARM PrimeCell™ PL061 "General Purpose Input/Output" is included in the APB system.

- compliant to AMBA Rev 2.0
- each port has eight individually programmable input/output pins, default to input at reset
- four ports A, B, C, D are included
- programmable interrupt generation capability, from a transition or level condition, on any number of PINs
- hardware control capability of GPIO's for different system configurations.
- bit masking in both read and write operations through address lines

Figure 19 GPIO Block Diagram

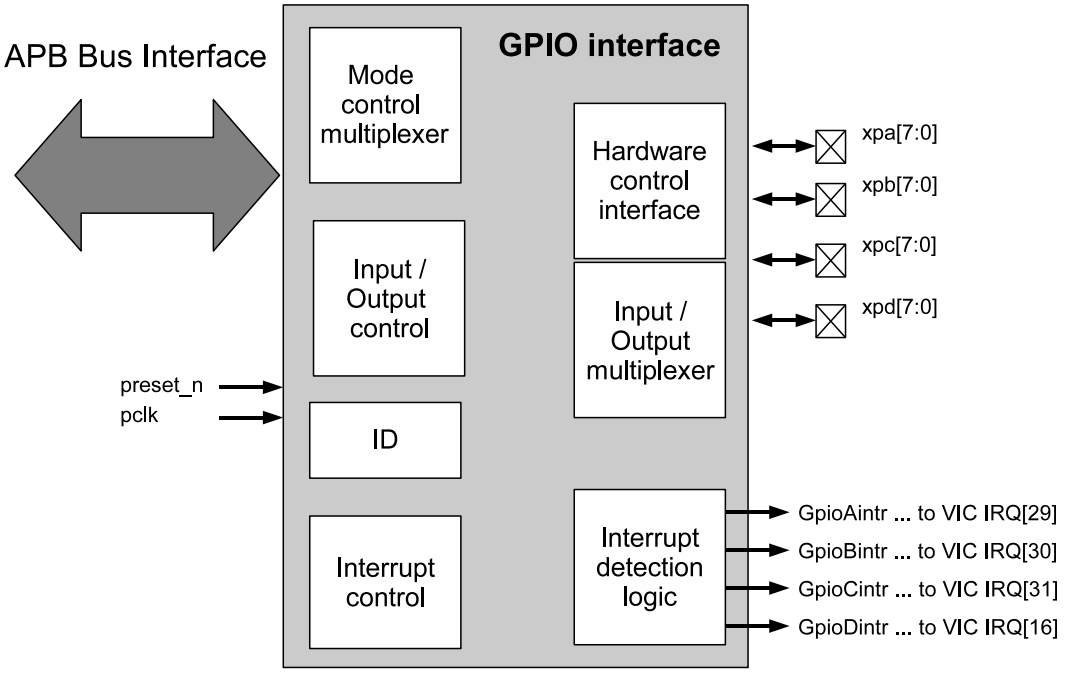


Table 26: GPIO Registers

Register Name	Base Address	Offset	Note
GPIO1_DATA	AS3525_GPIO1_BASE	0x000	GPIO data register
GPIO1_DIR	AS3525_GPIO1_BASE	0x400	GPIO data direction register
GPIO1_IS	AS3525_GPIO1_BASE	0x404	GPIO interrupt sense register
GPIO1_IBE	AS3525_GPIO1_BASE	0x408	GPIO interrupt both edges register
GPIO1_IEV	AS3525_GPIO1_BASE	0x40C	GPIO interrupt event register
GPIO1_IE	AS3525_GPIO1_BASE	0x410	GPIO interrupt mask register
GPIO1_RIS	AS3525_GPIO1_BASE	0x414	GPIO raw interrupt status
GPIO1_MIS	AS3525_GPIO1_BASE	0x418	GPIO masked interrupt status
GPIO1_IC	AS3525_GPIO1_BASE	0x41C	GPIO interrupt clear
GPIO1_AFSEL	AS3525_GPIO1_BASE	0x420	GPIO mode control select
GPIO2_DATA	AS3525_GPIO2_BASE	0x000	GPIO data register
GPIO2_DIR	AS3525_GPIO2_BASE	0x400	GPIO data direction register
GPIO2_IS	AS3525_GPIO2_BASE	0x404	GPIO interrupt sense register
GPIO2_IBE	AS3525_GPIO2_BASE	0x408	GPIO interrupt both edges register
GPIO2_IEV	AS3525_GPIO2_BASE	0x40C	GPIO interrupt event register
GPIO2_IE	AS3525_GPIO2_BASE	0x410	GPIO interrupt mask register
GPIO2_RIS	AS3525_GPIO2_BASE	0x414	GPIO raw interrupt status
GPIO2_MIS	AS3525_GPIO2_BASE	0x418	GPIO masked interrupt status
GPIO2_IC	AS3525_GPIO2_BASE	0x41C	GPIO interrupt clear
GPIO2_AFSEL	AS3525_GPIO2_BASE	0x420	GPIO mode control select

Register Name	Base Address	Offset	Note
GPIO3_DATA	AS3525_GPIO3_BASE	0x000	GPIO data register
GPIO3_DIR	AS3525_GPIO3_BASE	0x400	GPIO data direction register
GPIO3_IS	AS3525_GPIO3_BASE	0x404	GPIO interrupt sense register
GPIO3_IBE	AS3525_GPIO3_BASE	0x408	GPIO interrupt both edges register
GPIO3_IEV	AS3525_GPIO3_BASE	0x40C	GPIO interrupt event register
GPIO3_IE	AS3525_GPIO3_BASE	0x410	GPIO interrupt mask register
GPIO3_RIS	AS3525_GPIO3_BASE	0x414	GPIO raw interrupt status
GPIO3_MIS	AS3525_GPIO3_BASE	0x418	GPIO masked interrupt status
GPIO3_IC	AS3525_GPIO3_BASE	0x41C	GPIO interrupt clear
GPIO3_AFSEL	AS3525_GPIO3_BASE	0x420	GPIO mode control select
GPIO4_DATA	AS3525_GPIO4_BASE	0x000	GPIO data register
GPIO4_DIR	AS3525_GPIO4_BASE	0x400	GPIO data direction register
GPIO4_IS	AS3525_GPIO4_BASE	0x404	GPIO interrupt sense register
GPIO4_IBE	AS3525_GPIO4_BASE	0x408	GPIO interrupt both edges register
GPIO4_IEV	AS3525_GPIO4_BASE	0x40C	GPIO interrupt event register
GPIO4_IE	AS3525_GPIO4_BASE	0x410	GPIO interrupt mask register
GPIO4_RIS	AS3525_GPIO4_BASE	0x414	GPIO raw interrupt status
GPIO4_MIS	AS3525_GPIO4_BASE	0x418	GPIO masked interrupt status
GPIO4_IC	AS3525_GPIO4_BASE	0x41C	GPIO interrupt clear
GPIO4_AFSEL	AS3525_GPIO4_BASE	0x420	GPIO mode control select

5.3.5 MCI – SD / MMC Card Interface

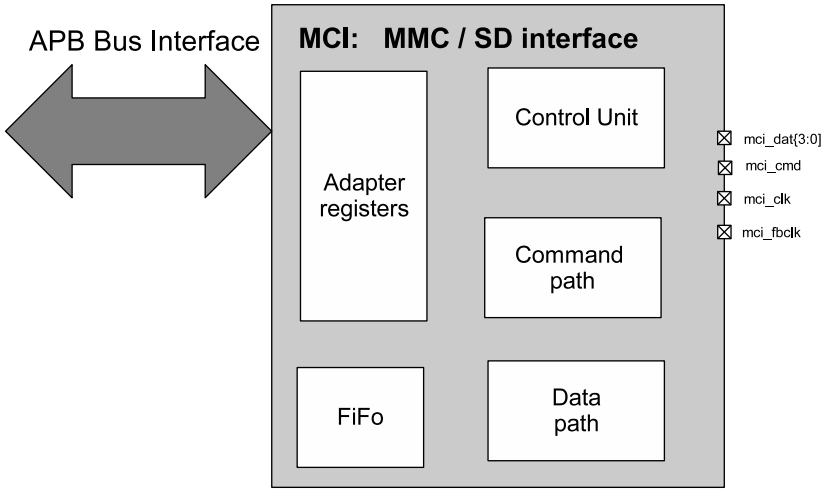
Features

- Conformance to Multimedia Card Specification v2.11
- Conformance to Secure Digital Memory Card Physical Layer Specification, v0.96
- uses multimedia card bus or SD card bus.

The PrimeCell™ MCI provides an interface between the APB system bus and multimedia and/or secure digital memory cards. It consists of two parts:

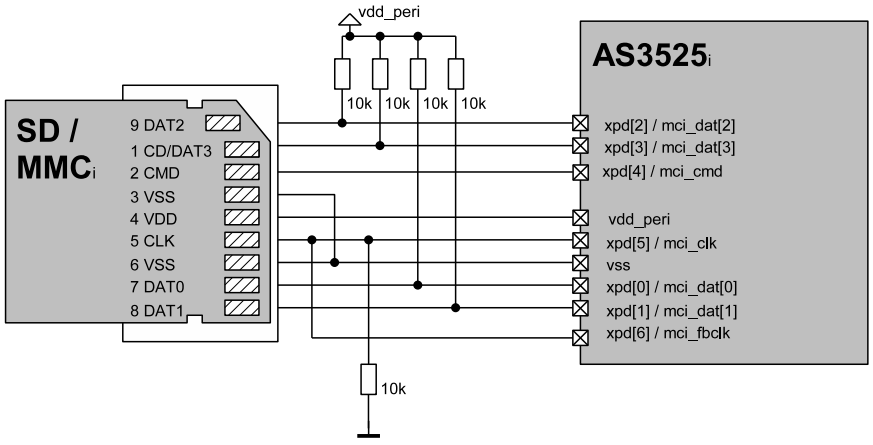
- The PrimeCell™ MCI adapter block includes the clock generation unit, the power management control, command and data transfer
- the APB interface provides access to the MCI adapter registers, and generates interrupt and DMA request signals.

Figure 20 Multimedia Card Interface Block Diagram



The connections to the Multimedia Card Interface are shared with the General Purpose I/O port-D (GPIO xpd[0:7]). Following diagram shows the external circuit elements for connection to a SD card adapter. Note that a feedback clock must be routed back to xpd[6]/mci_fbclk.

Figure 21 Connecting SD / MC to GPIO-D



5.3.6 I2cAudMas - I2C audio master interface

This is the control interface between the digital and the audio-part. The corresponding signal lines are connected inside of the MCM on the BGA substrate. For test purposes of the audio chip only, the signals are available at dedicated balls.

- The key features of this interface block are:
- serial 2-wire I2C bus master
- supports standard (100 kbps) and fast speed (400kbps)
- 7-bit addressing
- sub-addressing
- programmable clock divider
- programmable transfer count
- soft reset bit
- interrupt generation (on RX Full, TX Empty, RX Overrun, no acknowledge received)
- status register
- test register

Figure 22 I2C Audio Master Interface Block Diagram

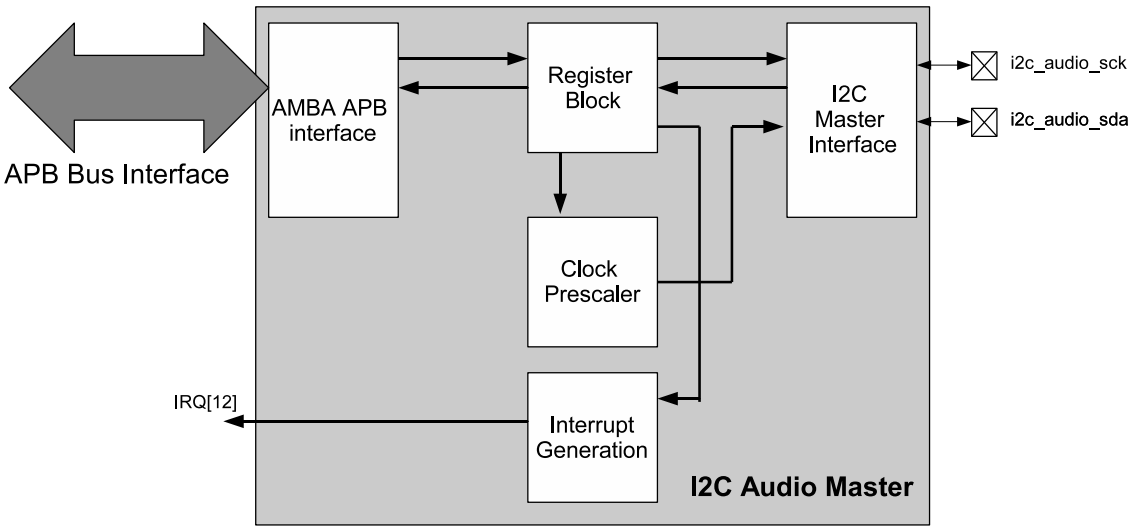


Table 27 I2C Audio Master Registers

Register Name	Base Address	Offset	Note
I2C2_DATA	AS3525_I2C_AUDIO_BASE	0x00	transmit/receive FIFO data register
I2C2_SLAD0	AS3525_I2C_AUDIO_BASE	0x04	slave ID register
I2C2_CNTRL	AS3525_I2C_AUDIO_BASE	0x0C	control register
I2C2_DACNT	AS3525_I2C_AUDIO_BASE	0x10	master data count register
I2C2_CPSR0	AS3525_I2C_AUDIO_BASE	0x1C	clock prescale register 0
I2C2_CPSR1	AS3525_I2C_AUDIO_BASE	0x20	clock prescale register 1
I2C2_IMR	AS3525_I2C_AUDIO_BASE	0x24	interrupt mask register
I2C2_RIS	AS3525_I2C_AUDIO_BASE	0x28	raw interrupt status register
I2C2_MIS	AS3525_I2C_AUDIO_BASE	0x2C	masked interrupt status register
I2C2_SR	AS3525_I2C_AUDIO_BASE	0x30	I2C status register
I2C2_INT_CLR	AS3525_I2C_AUDIO_BASE	0x40	interrupt clear register
I2C2_SADDR	AS3525_I2C_AUDIO_BASE	0x44	sub-address register
I2C2_TESTIN	AS3525_I2C_AUDIO_BASE	0x50	test register (monitors state of SCL and SDA)
I2C2_TESTOUT1	AS3525_I2C_AUDIO_BASE	0x54	test mode register for driving output interrupt
I2C2_TESTOUT2	AS3525_I2C_AUDIO_BASE	0x58	test mode register for driving SCLout, SCLOEn, SDAOUT and SDAOEN signals

5.3.7 I2CMSI - I2C master/slave interface

This is a general control interface for chip-to-chip communication. The corresponding IOs are either used by the general purpose port C (xpc[6:7]) or by this I2C interface.

The features of this interface block are:

- serial 2-wire I2C bus master
- supports standard (100 kbps) and fast speed (400kbps)
- supports multi-master system architecture
- programmable clock divider
- programmable transfer count
- programmable slave wait enable (for slave mode of operation, insertion of wait on the bus)
- soft reset bit
- interrupt generation (on RX Full, TX Empty, RX Overrun, no acknowledge received)
- status register
- test register

Figure 23: I2C Interface

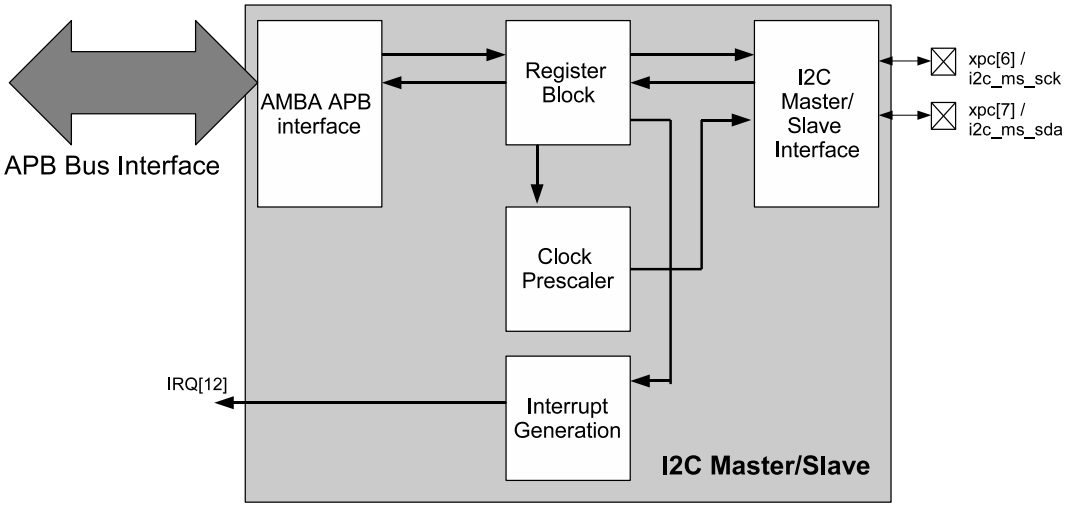


Table 28 I2C Interface Registers

Register Name	Base Address	Offset	Note
I2C1_DATA	AS3525_I2C_MS_BASE	0x00	transmit/receive FIFO data register
I2C1_SLAD0	AS3525_I2C_MS_BASE	0x04	slave ID register 0
I2C1_SLAD1	AS3525_I2C_MS_BASE	0x08	slave ID register 1
I2C1_CNTRL	AS3525_I2C_MS_BASE	0x0C	control register
I2C1_DACNT	AS3525_I2C_MS_BASE	0x10	master data count register
I2C1_SEAD0	AS3525_I2C_MS_BASE	0x14	self ID of slave 0
I2C1_SEAD1	AS3525_I2C_MS_BASE	0x18	self ID of slave 1
I2C1_CPSR0	AS3525_I2C_MS_BASE	0x1C	clock prescale register 0
I2C1_CPSR1	AS3525_I2C_MS_BASE	0x20	clock prescale register 1
I2C1_IMR	AS3525_I2C_MS_BASE	0x24	interrupt mask register
I2C1_RIS	AS3525_I2C_MS_BASE	0x28	raw interrupt status register
I2C1_MIS	AS3525_I2C_MS_BASE	0x2C	masked interrupt status register
I2C1_SR	AS3525_I2C_MS_BASE	0x30	I2C status register
I2C1_TXCNT	AS3525_I2C_MS_BASE	0x34	transmit Fifo data count register
I2C1_RXCNT	AS3525_I2C_MS_BASE	0x38	receive Fifo data count register
I2C1_TX_FLUSH	AS3525_I2C_MS_BASE	0x3C	TX Fifo flush register
I2C1_INT_CLR	AS3525_I2C_MS_BASE	0x40	interrupt clear register
I2C1_TESTIN	AS3525_I2C_MS_BASE	0x50	test register (monitors state of SCL and SDA)
I2C1_TESTOUT1	AS3525_I2C_MS_BASE	0x54	test mode register for driving output interrupt
I2C1_TESTOUT2	AS3525_I2C_MS_BASE	0x58	test mode register for driving SCLout, SCLOEn, SDAOUT and SDAOEN signals

5.3.8 I2SIN - I2S input interface

The I2S input interface module (called I2SINIF module hereafter) is used to connect an external audio source to the processor system. The communication is based on the standardized I2S interface. The interface module connects to the processor system using the AMBA APB bus.

All the input left & right channel data are mapped to either 14 or 24 bit format, selectable within the control register. If the data word length is less than 24 bit, the unused lower bits are set to zero. To reduce the interrupt frequency for the processor, a FIFO buffer is provided. The buffer can hold up to 32 words of 48 bit length (left plus right channel). Generation of interrupt request signal with several maskable interrupt sources (Pop Full, Pop Empty, Pop Error, Push Error, …etc)

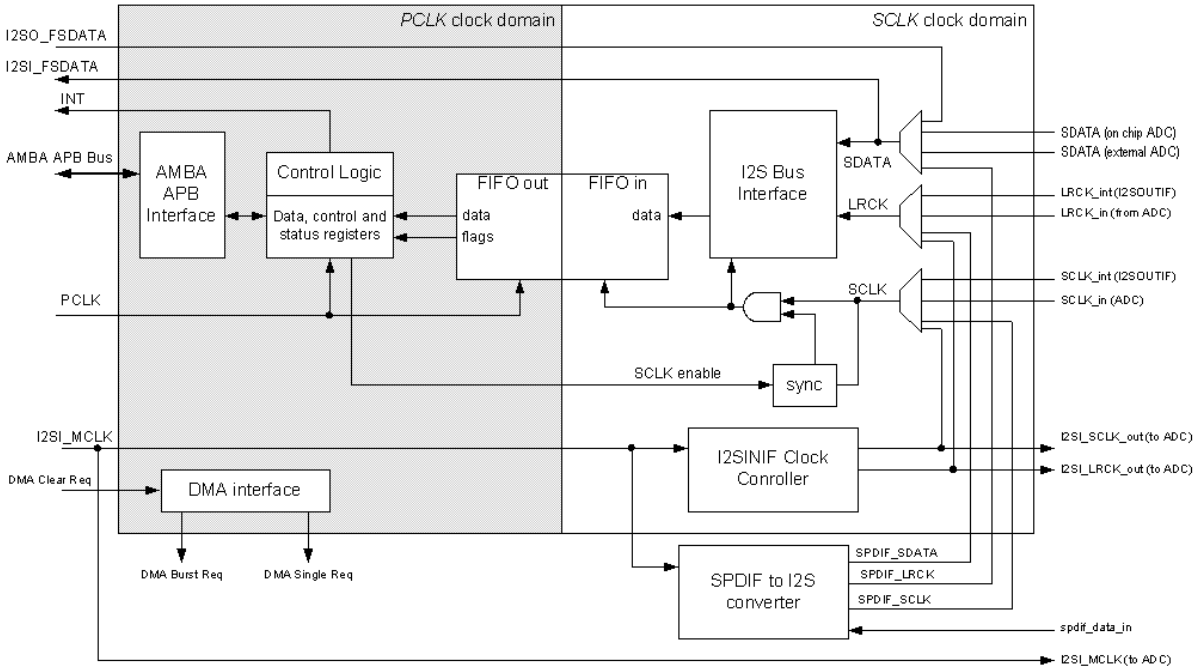
The I2SINIF provides the following features:

- two independent clock domains: AMBA APB clock PCLK, I2S input clock i2si_sclk
- FIFO (32 words/48 bit) separating clock domains
- support of several oversampling rates: 128x, 256x, 512x
- interrupt support for FIFO data read
- DMA support for FIFO data transfer

The I2SINIF provides five different modes:

- input from on-chip audio ADC
- input from external audio ADC in master mode (SCLK, LRCK generated by external ADC)
- input from external audio ADC in slave mode (SCLK, LRCK, MCLK generated internally and fed to external ADC)
- input from SPDIF (SPDIF to I2S converter)
- feedback mode with input from I2S output interface: used for test purposes

Figure 24 I2S Input Interface



5.3.8.1 I2S Input Register Mapping

I2S Input Interface Registers

Table 29 I2S Input Interface Registers

Register Name	Base Address	Offset	Note
I2SIN_CONTROL	AS3525_I2SIN_BASE	0x0000	Control register
I2SIN_MASK	AS3525_I2SIN_BASE	0x0004	Interrupt mask register
I2SIN_RAW_STATUS	AS3525_I2SIN_BASE	0x0008	Raw status register
I2SIN_STATUS	AS3525_I2SIN_BASE	0x000C	Status register
I2SIN_CLEAR	AS3525_I2SIN_BASE	0x0010	Interrupt clear register
I2SIN_DATA	AS3525_I2SIN_BASE	0x0014	Audio data register
I2SIN_SPDIF_STATUS	AS3525_I2SIN_BASE	0x0018	SPDIF status signals register

Table 30 I2S Input Control Register

Name		Base		Default
I2SIN_CONTROL		**AS3525_I2SIN_BASE**		**0x04**
Offset: 0x0000		**Control register**		
		12 bit wide read/write register containing the control bits of the I2SINIF.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
11	DMA_req_en	0	R/W	DMA request enable 0: disable 1: enable
10	mclk_invert	0	R/W	Invert MCLK 0: disable (SCLK changes at MCLK's falling edge) 1: enable (SCLK changes at MCLK's rising edge)
9,8	i2s_clk_source	00	R/W	Define the source of SCLK and LRCK for I2SINIF 00: SCLK and LRCK from I2SOUTIF (used if AFE sends data) 01: SCLK and LRCK from external ADC device (outside AS3525) 10: SCLK and LRCK from SPDIF converter 11: SCLK and LRCK from I2SINIF's clock controller
7,6	sdata_source	00	R/W	Define the source of SDATA for I2SINIF 00: SDATA from AFE 01: SDATA from external ADC device (outside AS3525) 10: SDATA from SPDIF converter 11: loopback SDATA from I2SOUTIF (test purpose)
5	14bit_mode	0	R/W	0: ADC data from FIFO transferred in two 32-bit words to I2SIN_DATA (first left and then right data as indicated by the stereo24_status bit) 1: ADC data from FIFO transferred in one 32-bit word to I2SIN_DATA
4	sclk_idle	0	R/W	Enable/disable SCLK for I2SINIF 0: SCLK enabled 1: SCLK disabled
3	SDATA_valid	0	R/W	0: SDATA ignored at first SCLK edge (I2S standard) 1: valid SDATA at first SCLK edge
2	sclk_edge	1	R/W	0: data valid at negative edge of SCLK 1: data valid at positive edge of SCLK
1,0	osr	00	R/W	Oversampling rate (needed for generating sclk and lrck) 00: 128x 01: 256x 10: 512x 11: 128x

The following table shows the valid combinations for sdata_source (bit 7 and 6) and i2s_clk_source (bit 9 and 8) of the I2SIN_CONTROL register.

sdata_source	i2s_clk_source	Description
00	00	default mode (AFE with AS3525)
01	00	external data, external clock
01	11	external data, internal clock
10	10	data and clock from SPDIF converter
11	00	loopback, internal data and clock

Table 31 I2S Input mask register

Name		Base		Default
I2SIN_MASK		**AS3525_I2SIN_BASE**		**0x00**
Offset: 0x0004		**Interrupt mask register**		
		The interrupt mask register determines which status flags generate an interrupt by setting the corresponding bit to 1.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	reserved	0	R/W	stereo24_status cannot assert interrupt request
6	I2SIN_MASK_PUER	0	R/W	1 enables the FIFO PUSH error interrupt
5	I2SIN_MASK_POE	0	R/W	1 enables the FIFO POP is empty interrupt
4	I2SIN_MASK_POAE	0	R/W	1 enables the FIFO POP is almost empty interrupt
3	I2SIN_MASK_POHF	0	R/W	1 enables the FIFO POP is half full interrupt
2	I2SIN_MASK_POAF	0	R/W	1 enables the FIFO POP is almost full interrupt
1	I2SIN_MASK_POF	0	R/W	1 enables the FIFO POP is full interrupt
0	I2SIN_MASK_POER	0	R/W	1 enables the FIFO POP error interrupt

Table 32 I2S Input raw status register

Name		Base		Default
I2SIN_RAW_STATUS		**AS3525_I2SIN_BASE**		**0x00**
Offset: 0x0008		**Raw status register**		
		The read-only raw status register contains the actual bit values as reflected by the FIFO controller status signals. I2SIN_PUER and I2SIN_POER are static bits, since FIFO controller gives the PUSH/POP error bit only for one clock. This means that these two bits remain asserted until they are cleared in the I2SIN_CLEAR register. All other bits change state depending on the underlying logic, i.e. state of FIFO controller.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	stereo24_status	0	R	Status of write interface for 24 bit stereo mode 0: left audio sample will be transferred next 1: right audio sample will be transferred next
6	I2SIN_PUER	0	R	1 if FIFO PUSH error
5	I2SIN_POE	0	R	1 if FIFO POP is empty
4	I2SIN_POAE	0	R	1 if FIFO POP is almost empty
3	I2SIN_POHF	0	R	1 if FIFO POP is half full
2	I2SIN_POAF	0	R	1 if FIFO POP is almost full
1	I2SIN_POF	0	R	1 if FIFO POP is full
0	I2SIN_POER	0	R	1 if FIFO POP error

Table 33 I2S input status register

Name		Base		Default
I2SIN_STATUS		**AS3525_I2SIN_BASE**		**0x00**
Offset: 0x000C		**Status register**		
		The status register is a read-only register. A read to this register returns the value of the raw status bits AND'ed with the corresponding mask of enable bits set in the mask register.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	stereo24_status	0	R	Status of write interface for 24 bit stereo mode 0: left audio sample will be transferred next 1: right audio sample will be transferred next
6	I2SIN_PUER	0	R	1 if FIFO PUSH error
5	I2SIN_POE	0	R	1 if FIFO POP is empty
4	I2SIN_POAE	0	R	1 if FIFO POP is almost empty
3	I2SIN_POHF	0	R	1 if FIFO POP is half full
2	I2SIN_POAF	0	R	1 if FIFO POP is almost full
1	I2SIN_POF	0	R	1 if FIFO POP is full
0	I2SIN_POER	0	R	1 if FIFO POP error

Table 34 I2S Input interrupt clear register

Name		Base		Default
I2SIN_CLEAR		**AS3525_I2SIN_BASE**		**0x00**
Offset: 0x0010		**Interrupt clear register**		
		The interrupt clear register is a write-only register. The corresponding static status bit can be cleared by writing a 1 to the corresponding bit in the clear register. All other interrupt flags are level interrupts depending on the status of the FIFO. The bits are de-asserted depending on the FIFO controller.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	reserved		W	
6	I2SIN_clear_puer		W	Clear PUSH error interrupt flag
5:1	reserved		W	
0	I2SIN_clear_poer		W	Clear POP error interrupt flag

I2SIN_DATA

The I2SINIF provides a single 32 bit wide data register. The register is used to read the audio samples from FIFO. If 14 bit mode is selected, both the left and right data are made available in the same register. Otherwise in the 24 bit mode the left and right data are provided through the same register alternatively. The stereo24_status bit in the I2SIN_STATUS register provides information which channel's data will be provided next. The 14bit_mode bit in the I2SIN_CONTROL register defines how the values are read from the FIFO.

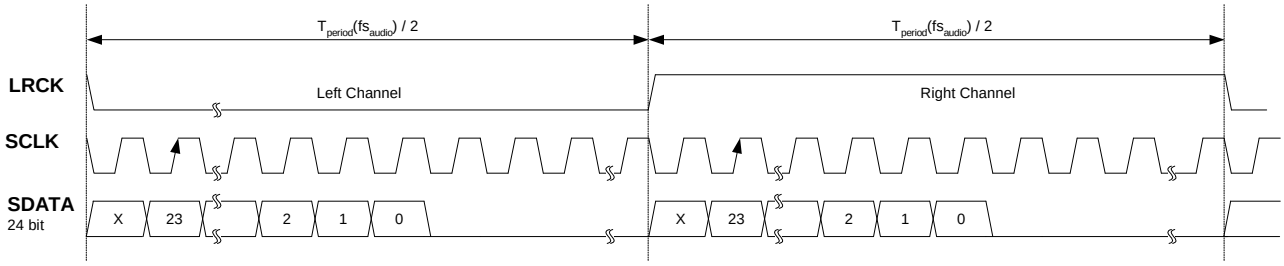
5.3.8.2 I2S Input Signals

The following specifications signals are given:

- Data are valid at rising/falling edge of SCLK (depending on I2SI_CONTROL's setting).
- The left and right channels are indicated by the LRCK signal.

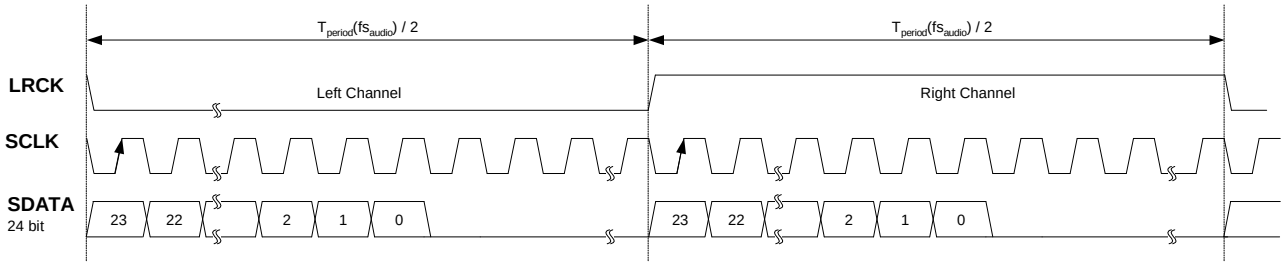
The timing diagram of the standard I2S interface signals from the ADC is shown below (Figure 28).

Figure 25 - I2S standard timing diagram



While the I2S standard states that the LRCK line changes one clock cycle before the MSB is transmitted. If the ADC sends the MSB directly after LRCK line changes, the *SDATA_valid* bit in the I2SI_CONTROL register must be set.

Figure 26 - I2S standard timing diagram with SDATA valid directly after LRC changes



Assumption: The LRCK toggles every 32 clocks of SCLK.

5.3.8.3 Power Modes

The I2SINIF contains two clock domains. The PCLK domain can be turned off in the clock controller. The SCLK clock domain can be turned off locally using the SCLK_idle bit in the I2SIN_CONTROL register. Note that the SCLK's clock gating signal has to be synchronized with the SCLK clock in order to guarantee correct operation.

If PCLK is turned off, no interrupt must be triggered by the I2SINIF module.

The I2SI_MCLK clock can be turned on/off in the clock generation unit.

5.3.8.4 Loopback Feature

On the AS3525 are two I2S interfaces:

- I2SOUTIF is responsible to send values to the DAC of the audio chip via I2SO_SDATA
- I2SINIF is responsible to receive audio values from ADC of the audio chip via I2SI_SDATA

In the AS3525 both SDATA signals are provided as loopback signals (I2SO_FSDATA, I2SI_FSDATA):

- I2SO_SDATA to I2SINIF: This loopback is mainly for testing the transmit and receive paths of both I2S interfaces. The loopback signal is called I2SO_FSDATA.
- I2SI_SDATA to I2SOUTIF: This loopback feature allows the application to echo the input audio samples directly to a loudspeaker. The signal provided by the I2SINIF is called I2SI_FSDATA.

In normal mode the I2SINIF pushes audio values into the FIFO based on the I2SI_SDATA signal. If the loopback feature is enabled, the *sdata_source* bit in the control register must be set to 3. The FIFO content is filled with audio values send by the I2SOUTIF (signal I2SO_FSDATA).

NOTE: This feature will only be available if SCLK is the same for I2S input and output interface. For implementation the I2SO_FSDATA signal is simply routed through a multiplexer to the I2SI_SDATA interface.

5.3.8.5 DMA Interface

The I2SINIF supports DMA transfers. The DMA controller supports incrementing and non-incrementing (single address) addressing for source and destination. For I2SINIF the single-address mode is used. The address of the I2SI_DATA register is used as DMA source address.

5.3.8.6 The 24 bit Stereo DMA Mode

In 24 bit stereo mode, right and left audio samples must be read separately from the FIFO. In single-address DMA-mode both data must be read from the same address. The I2SINIF is responsible to split up the 48 bit FIFO entries into two 24 bit samples. The 24 bit value can then be transferred via the 32 bit wide AMBA bus.

The I2SINIF provides the data in a specific order: first the left value is sent, and afterwards the right value is provided. Then a left value follows, and so on. In the destination memory the words are stored incrementally as shown below.

Address	Value
addr 0	LDATA 0
addr 1	RDATA 0
addr 2	LDATA 1
addr 3	RDATA 1
…	…
addr n*2	LDATA n
addr n*2+1	RDATA n

5.3.9 SPDIF interface

As part of the I2SIN module also a SPDIF receiver interface is included. This SPDIF interface works as converter from SPDIF-AES/EBU to I2S.

The SPDIF-AES/EBU standard is a serial audio interface that conveys 2 time-multiplexed audio channels, the left and right channels, as is the case in audio stereo transmission. The two channels are encoded in a 64-bit frame. Each individual channel is encoded in a sub-frame that consists of a 4-bit preamble, followed by 24 bits of audio data and 4 control bits, in a total of 32 bits per sub-frame. The SPDIF-AES/EBU standard provides for LSB first, up to 24-bit audio samples, Samples of 20 bits or less may be used, in which case the 4 least significant bits may be used for a 12-bit monitoring channel, transmitted at 1/3 of the sample rate. Please refer to the SPDIF-AES/EBU, AES3 or IEC958 standard documentation for more information.

Features

- Feed-forward operation: extracts audio data from the SPDIF-AES/EBU input signal by sampling it with a fast clock signal which not necessarily related to the sample rate frequency
- Purely digital receiver solution, without need of an input PLL for synchronisation.
- The audio samples are output serially in I2S format.
- PLL interface to filter out the jitter and generate a jitter-free I2S output.
- Recognizes all common audio and video related sample frequencies and outputs a nibble code for each.

5.3.9.1 SPDIF register description

Table 35 SPDIF status register

Name		Base		Default
I2SIN_SPDIF_STATUS		**AS3525_I2SIN_BASE**		**0x00**
Offset: 0x0018		**SPDIF status signals register**		
		This read-only register contains status information of the SPDIF interface. The spdif_sample_freq and spdif_sync status bits are directly derived from the SPDIF converter. In order to provide valid status bits, these signals must be synchronized with pclk, i.e. clk_i2sin.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
4:1	spdif_sample_freq		R	Incoming sample frequency
0	spdif_sync		R	Recognition of sub-frame preamble 0: first sub-frame preamble not recognized 1: successful recognition of the first sub-frame preamble

The following table shows the input sample rate in KHz according to the sample_freq_code (bit 5 to 1) in the I2SIN_SPDIF register.

sample_freq_code	Input Sample Rate (KHz)
0001	22.050
0010	24.000
0011	32.000
0100	44.100
0101	48.000
0110	64.000
0111	88.200
1000	96.000
1001	176.400
1010	192.000

5.3.10 I2SOUT - I2S output interface

The I2S output interface module (called I2SOUTIF module hereafter) is used to connect the processor system to an audio DAC. The communication is based on the standardized I2S interface. The audio samples are transferred from the processor to the I2SOUTIF module using the AMBA APB bus. A FIFO for 128 dual-channel audio samples is provided as a data buffer. Furthermore, the module provides a set of data, control and status registers.

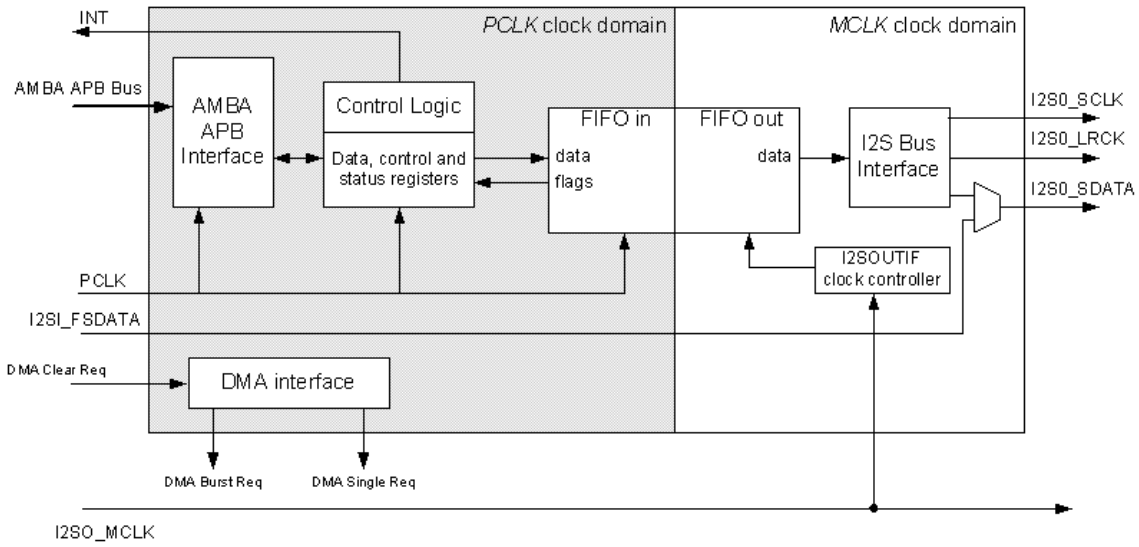
The I2SOUTIF provides the following features:

- two independent clock domains: AMBA APB clock PCLK, I2S output clock i2so_mclk
- FIFO (128 words with 36 bit) separating clock domains
- support of 16 and 18 bit audio samples
- clock generator for I2S clocks (LCLK, I2SO_SCLK)
- support of several oversampling rates: 128x, 256x, 512x
- interrupt support for FIFO data write
- DMA support for FIFO data transfer

For data output, following modes are implemented:

- two 18 bit audio samples, one for each channel (R,L). The values are written to I2SO_DATA.
- two 16 bit audio samples, one for each channel (R,L). Both values are written to the 32-bit wide I2SO_DATA register at the same time. This mode is highly efficient for 32-bit processor architectures.
- one 18 bit mono audio sample; the sample is used for both channels (R and L). The value is written to the I2SO_DATA.
- one 16 bit mono audio sample; the sample is used for both channels (R and L). The value is written to the I2SO_DATA.

Figure 27 I2SO Block Diagram



5.3.10.1 I2S Output Interface Registers

Table 36 I2S Output Interface Registers

Register Name	Base Address	Offset	Note
I2SOUT_CONTROL	AS3525_I2SOUT_BASE	0x0000	Control register
I2SOUT_MASK	AS3525_I2SOUT_BASE	0x0004	Interrupt mask register
I2SOUT_RAW_STATUS	AS3525_I2SOUT_BASE	0x0008	Raw status register
I2SOUT_STATUS	AS3525_I2SOUT_BASE	0x000C	Status register
I2SOUT_CLEAR	AS3525_I2SOUT_BASE	0x0010	Interrupt clear register
I2SOUT_DATA	AS3525_I2SOUT_BASE	0x0014	Audio data register

Table 37 I2SOUT control register

Name		Base		Default
I2SOUT_CONTROL		**AS3525_I2SOUT_BASE**		**0x0C**
Offset: 0x0000		**Control register**		
		7 bit wide read/write register containing the control bits of the I2SOUTIF.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
6	DMA_req_en	0	R/W	DMA request enable 0: disable 1: enable
5	sdata_lb	0	R/W	I2SDATA loopback from I2SINIF 0: I2SOUT_SDATA source is I2SOUTIF's FIFO 1: I2SOUT_SDATA source is loopback value from I2SINIF (signal I2SIN_FDATA)
4	mclk_invert	0	R/W	Invert MCLK 0: disable (SCLK changes at MCLK's falling edge) 1: enable (SCLK changes at MCLK's rising edge)
3	stereo_mode	1	R/W	Audio samples provided by processor 0: mono 1: stereo
2	18bit_mode	1	R/W	Bit width of audio samples provided by processor 0: 16 bit 1: 18 bit
1,0	osr	00	R/W	Oversampling rate 00: 128x 01: 256x 10: 512x 11: 128x

CAUTION: The control bit sdata_lb can only be set, if the I2SIN_FSDATA is synchronous to I2SOUT_SCLK. This is the case if AFE is used together with the AS3525 (in this case the I2SINIF uses also I2SOUT_CLK).

Table 38 I2S Output mask register

Name		Base		Default
I2SOUT_MASK		**AS3525_I2SOUT_BASE**		**0x00**
Offset: 0x0004		**Interrupt mask register**		
		The interrupt mask register determines which status flags generate an interrupt by setting the corresponding bit to 1.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	reserved	0	R/W	stereo18_status cannot assert interrupt request
6	I2SOUT_MASK_POER	0	R/W	1 enables the FIFO POP error interrupt
5	I2SOUT_MASK_PUE	0	R/W	1 enables the FIFO PUSH is empty interrupt
4	I2SOUT_MASK_PUAE	0	R/W	1 enables the FIFO PUSH is almost empty interrupt
3	I2SOUT_MASK_PUHF	0	R/W	1 enables the FIFO PUSH is half full interrupt
2	I2SOUT_MASK_PUAF	0	R/W	1 enables the FIFO PUSH is almost full interrupt
1	I2SOUT_MASK_PUF	0	R/W	1 enables the FIFO PUSH is full interrupt
0	I2SOUT_MASK_PUER	0	R/W	1 enables the FIFO PUSH error interrupt

Table 39 I2S output raw status register

Name		Base		Default
I2SOUT_RAW_STATUS		**AS3525_I2SOUT_BASE**		**0x00**
Offset: 0x0008		**Raw status register** The read-only raw status register contains the actual bit values as reflected by the FIFO controller status signals. I2SOUT_POER and I2SOUT_PUER are static bits, since FIFO controller gives the PUSH/POP error bit only for one clock. This means that these two bits remain asserted until they are cleared in the I2SOUT_CLEAR register. All other bits change state depending on the underlying logic, i.e. state of FIFO controller.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	stereo18_status	0	R	Status of write interface for 18 bit stereo mode 0: left audio sample is expected next 1: right audio sample is expected next
6	I2SOUT_POER	0	R	1 if FIFO POP error
5	I2SOUT_PUE	0	R	1 if FIFO PUSH is empty
4	I2SOUT_PUAE	0	R	1 if FIFO PUSH is almost empty
3	I2SOUT_PUHF	0	R	1 if FIFO PUSH is half full
2	I2SOUT_PUAF	0	R	1 if FIFO PUSH is almost full
1	I2SOUT_PUF	0	R	1 if FIFO PUSH is full
0	I2SOUT_PUER	0	R	1 if FIFO PUSH error

Table 40 I2S output status register

Name		Base		Default
I2SOUT_STATUS		**AS3525_I2SOUT_BASE**		**0x00**
Offset: 0x000C		**Status register** The status register is a read-only register. A read to this register returns the value of the raw status bits AND'ed with the corresponding mask of enable bits set in the mask register.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	stereo18_status	0	R	Status of write interface for 18 bit stereo mode 0: left audio sample is expected next 1: right audio sample is expected next
6	I2SOUT_POER	0	R	1 if FIFO POP error
5	I2SOUT_PUE	0	R	1 if FIFO PUSH is empty
4	I2SOUT_PUAE	0	R	1 if FIFO PUSH is almost empty
3	I2SOUT_PUHF	0	R	1 if FIFO PUSH is half full
2	I2SOUT_PUAF	0	R	1 if FIFO PUSH is almost full
1	I2SOUT_PUF	0	R	1 if FIFO PUSH is full
0	I2SOUT_PUER	0	R	1 if FIFO PUSH error

Table 41 I2S output interrupt clear register

Name		Base		Default
I2SOUT_CLEAR		**AS3525_I2SOUT_BASE**		**0x00**
Offset: 0x0010		**Interrupt clear register**		
		The interrupt clear register is a write-only register. The corresponding static status bit can be cleared by writing a 1 to the corresponding bit in the clear register. All other interrupt flags are level interrupts depending on the status of the FIFO. The bits are de-asserted depending on the FIFO controller.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	reserved		W	
6	I2SOUT_clear_poer		W	Clear POP error interrupt flag
5:1	reserved		W	
0	I2SOUT_clear_puer		W	Clear PUSH error interrupt flag

I2SOUT_DATA

The I2SOUTIF provides two 32 bit wide data registers. The registers are used to store the audio samples before they are written to the FIFO. The registers can be used in different modes depending on the setting of the I2SOUT_CONTROL register.

Basically, there are four ways to fill the FIFO.

The processor can provide

- two 18 bit audio samples, one for each channel (R,L). The values are written to I2SOUT_DATA.
- two 16 bit audio samples, one for each channel (R,L). Both values are written to the 32-bit wide I2SOUT_DATA register at the same time. This mode is highly efficient for 32-bit processor architectures.
- one 18 bit mono audio sample; the sample is used for both channels (R and L). The value is written to the I2SOUT_DATA.
- one 16 bit mono audio sample; the sample is used for both channels (R and L). The value is written to the I2SOUT_DATA.

In 18 bit stereo mode the data in I2SOUT_DATA is interpreted either as left or right audio value. The stereo18_status bit in the I2SOUT_STATUS register provides the information which channel's audio sample is expected next.

The I2S Output Signals

The following specifications signals are given:

- Data are valid at the rising edge of I2SO_SCLK.
- The MSB is left justified to the I2S frame identification (I2SO_LRCK). According to standard I2S definition, a delay of one clock cycle between transition of I2SO_LRCK and the data MSB is used.

The timing diagram of the I2S interface signals for 18bit and 16bit DAC is shown below.

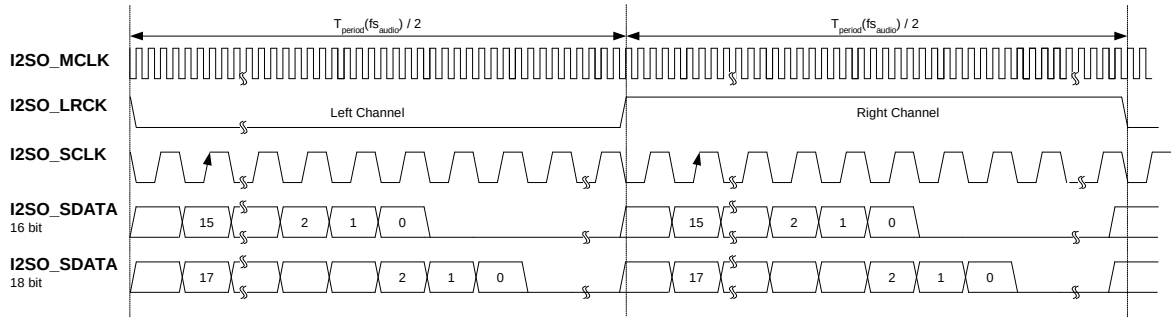
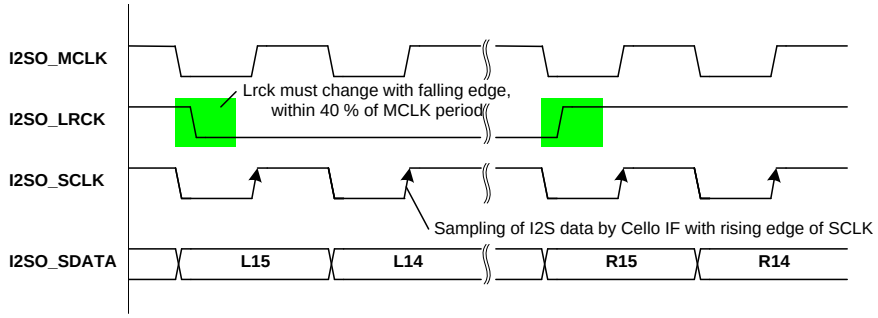


Figure 28 - I2S output timing diagram

For the relationship of the clocks following constraints must be met:

- LRCK must change with the falling edge of MCLK while MCLK is low (constrained should be set to 40 % of the MCLK period, see figure below).
- SDATA must change at the falling edge of SCLK. It will be read with the rising edge of SCLK.

Figure 29 Clock constraints



5.3.10.2 Power Modes

The I2SOUTIF contains two clock domains. Each clock domain can be turned off separately. The I2SO_MCLK must be turned off in the global clock controller register. This is necessary, as the audio chip requires I2SO_MCLK and I2SO_SCLK not only for I2S output, but also I2S input (see I2SINIF).

PCLK Idle Mode

If the PCLK is turned off (by the clock controller) the I2SOUT_STATUS register can hold invalid data. However, no interrupt should be triggered if the I2SOUTIF is in idle mode.

I2SO_MCLK Idle Mode

If I2SO_MCLK is disabled (by the clock controller) no audio samples are read from the FIFO. The output signals remain unchanged until the I2SO_MCLK is enabled again.

5.3.10.3 Loopback Feature

On the AS3525 are two I2S interfaces:

- I2SOUTIF is responsible to send values to the DAC of the audio chip via I2SO_SDATA
- I2SINIF is responsible to receive audio values from ADC of the audio chip via I2SI_SDATA

In the AS3525 both SDATA signals are provided as loopback signals (I2SO_FSDATA, I2SI_FSDATA):

- I2SO_SDATA to I2SINIF: This loopback is mainly for testing the transmission and reception paths of both I2S interfaces. The loopback signal is called I2SO_FSDATA.
- I2SI_SDATA to I2SOUTIF: This loopback feature allows the application to echo the input audio samples directly to a loudspeaker. The signal provided by the I2SINIF is called I2SI_FSDATA.

In normal mode the I2SOUTIF generates the I2SO_SDATA signal based on the contents of the FIFO. If the loop back feature is enabled, the SDATA_LB bit in the I2SOUT_CONTROL register must be set.

NOTE: This feature will only be available if SCLK is the same for I2S input and output interface. For implementation the I2SI_FSDATA signal is simply routed through a multiplexer to the I2SO_SDATA interface.

5.3.10.4 DMA Interface

The I2SOUTIF supports DMA transfers. The DMA controller supports incrementing and non-incrementing (single address) addressing for source and destination. For I2SOUTIF the single-address mode is used. The address of the I2SOUT_DATA register is used as DMA destination address.

Stereo 18 bit DMA Mode

In 18 bit stereo mode, right and left audio samples must be transferred separately to the FIFO. In single-address DMA-mode both data must be written to the same address. The I2SOUTIF is responsible to put the two 18 bit samples together to a 36 bit word. This word is written into the 36 bit wide FIFO.

The I2SOUTIF requires a specific ordering of the samples written to the I2SOUT_DATA register: first the left value must be written, and afterwards the DMA controller must write the right value. Then a left value can follow, a.s.o. The status bit *stereo18_status* shows which audio sample is expected.

In order to set up a correct DMA transfer the values must be placed in the source memory as follows:

Address	Value
addr 0	LDATA 0
addr 1	RDATA 0
addr 2	LDATA 1
addr 3	RDATA 1
…	…
addr n*2	LDATA n
addr n*2+1	RDATA n

5.3.11 NAND Flash Interface

The NAND FLASH interface module enables control of NAND flash devices. The design follows the hardware reference implementation described in SMIL (SmartMedia™ Interface Library), Hardware Edition 1.00, TOSHIBA Corporation, but has extensions to support the latest generation of NAND flash devices.

Programming and Reading can be done either by direct access to/from data register (normal mode) or by using a FIFO (burst mode). NAF supports 8-bit and 16-bit transfers.

Features

- interface compliant to AMBA APB bus
- generation of interrupt request signal with several maskable interrupt sources (ready, empty, almost_empty…)
- hardware error detection (2 detect, 1 correct per 256 bytes block) for up to 8 *256 bytes (up to 24 ECC bytes)
- 8-bit and 16-bit transfer Mode fore X8/X16 devices
- big endian / little endian support
- DMA Mode
- Normal Mode
- Data/Mode/Status Register
- write/read on/from data register automatically generates read/write strobes
- Burst Transfer
- 36 x 32 bit FIFO for DMA/burst support
- read- & write controller for automatic data resizing (32bit <=> 8/16bit) and read/write control
- configurable strobe (low and high time) for higher PCLK clocks / lower speed NAND Flash devices
- little endian/ big endian selectable
- load interrupts when FIFO is ‘almost_empty’ & ’almost_full’ to ensure continuous data flow

Figure 30 Block Diagram of NAND Flash Interface

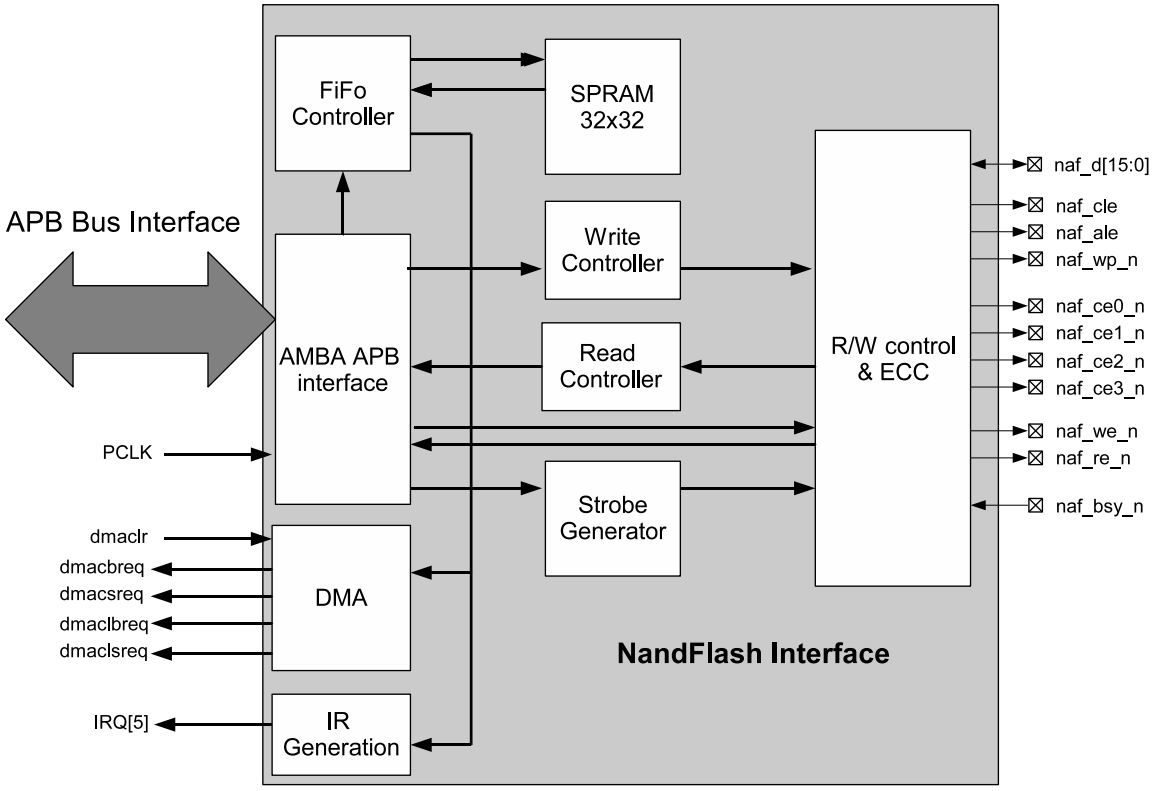
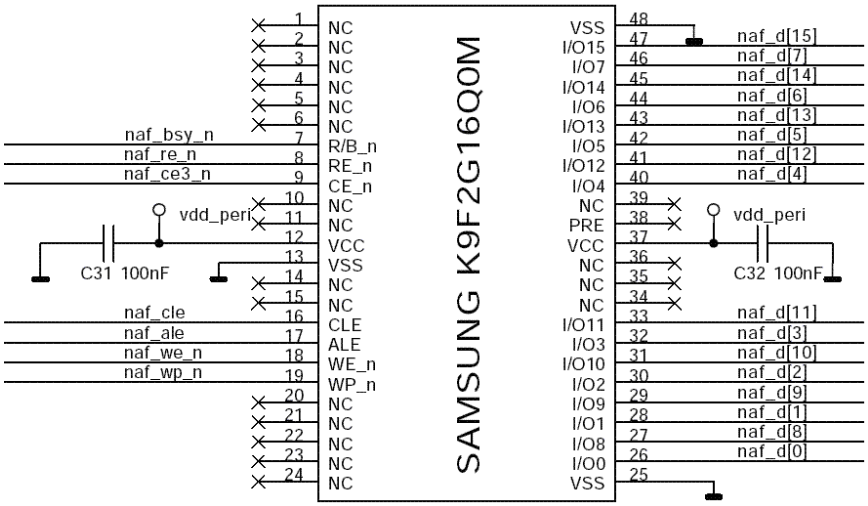


Figure 31 Connecting a NAND Flash



5.3.11.1 NandFlash Interface Registers

Table 42 NAF registers

Register Name	Base Address	Offset	Note
NAFCONFIG	AS3525_NAND_FLASH_BASE	0x00	Configuration register
NAFCONTROL	AS3525_NAND_FLASH_BASE	0x04	Control register
NAFECC	AS3525_NAND_FLASH_BASE	0x08	Error correction code reg
NAFDATA	AS3525_NAND_FLASH_BASE	0x0C	Data register
NAFMODE	AS3525_NAND_FLASH_BASE	0x10	Mode register
NAFSTATUS	AS3525_NAND_FLASH_BASE	0x14	Status register
NAFMASK	AS3525_NAND_FLASH_BASE	0x18	Interrupt mask register
NAFFIFODATA	AS3525_NAND_FLASH_BASE	0x1C	buffered read/write data register
NAFWORDS	AS3525_NAND_FLASH_BASE	0x20	Words register
NAFCLEAR	AS3525_NAND_FLASH_BASE	0x24	Interrupt clear register
NAFTEST	AS3525_NAND_FLASH_BASE	0x28	Test register

Table 43 NAF configuration register

Name		Base		Default
NAFConfig		**AS3525_NAND_FLASH_BASE**		**0x00**
Offset 0x0000		**NAF Configuration Register**		
		The register is used for basic setup. 8 or 16-bit data width, little or big endian can be selected. DMA and FIFO on/off can be controlled as well as duty cycle and duration of read & write signals.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
19:16	write_strobe_low [3:0]	0x00	R/W	low time (# of PCLK cycles + 1) of the output 'naf_we_n' (e.g. a value of 1 will keep naf_we_n at '0' for 3 PCLK cycles during write)
15:12	write_strobe_high [3:0]	0x00	R/W	high time (# of PCLK cycles + 2) of the output 'naf_we_n' (e.g. a value of 0 will keep naf_we_n at '1' for 2 PCLK cycles during write)
11:8	read_strobe_low [3:0]	0x00	R/W	low time (# of PCLK cycles + 1) of the output 'naf_re_n' (e.g. a value of 2 will keep naf_re_n at '0' for 3 PCLK cycles during read)
7:4	read_strobe_high [3:0]	0x00	R/W	high time (# of PCLK cycles + 2) of the output 'naf_re_n' (e.g. a value of 0 will keep naf_re_n at '1' for 2 PCLK cycles during read)
3	dma_on	0x0	R/W	0: DMA is disabled and all DMA request signals are tied to 1: DMA is enabled
2	fifo_staticreset_n	0x0	R/W	0: FIFO is reset 1: FIFO is enabled
1	big_endian	0x0	R/W	0: little endian (FIFO data word will be processed in the order word(7:0), word(15:8), word(23:16) and word(31:24) when x16_device is 0; word(15:0) and word(31:16) when x16_device is 1 1: big endian (FIFO data word will be processed in the order word(31:24), word(23:16), word(15:8) and word(7:0) when x16_device is 0; word(31:16) and word(15:0) when x16_device is 1 Note: big_endian is only supported for r/w access through register NAFFifodata
0	x16_device	0x0	R/W	0: X8 Device (for NAND flash with 8-bit data bus) 1: X16 Device (for NAND flash with 16-bit data bus)

Table 44 NAF control register

Name		Base		Default
NAFControl		**AS3525_NAND_FLASH_BASE**		**0x2**
Offset 0x0004		**NAFControl Register**		
		The NAFControl register controls read access and FIFO dynamic reset.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
1	read_strobe	0x1	W	1: triggers a FIFO reset pulse (when NAFConfig bit 'fifo_staticreset_n' is 1) The bit is cleared automatically in the next PCLK cycle.
0	fifo_reset_strobe	0x1	W	1: triggers one single read cycle on output 'naf_re_n'. The bit is cleared automatically in the next PCLK cycle.

Table 45 NAF error correction register

Name		Base		Default
NAFEcc		**AS3525_NAND_FLASH_BASE**		**0x2**
Offset 0x0008		**NAF Error correction code register**		
		The NAFEcc register offers access to the error correction code registers.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
32:0	Nafecc [32:0]	0x0001	R	This register can be accessed up to 8 times and contains the following data: 1.access => Line Parity Block1 2.access => Column Parity Block1** 3.access => Line Parity Block2 4.access => Column Parity Block2** 5.access => Line Parity Block3 6.access => Column Parity Block3** 7.access => Line Parity Block4 8.access => Column Parity Block4** (9.access => same as 1.access)

Note: * Before access to NAFEcc registers is possible, NAFMode register has to be set to 0xd4 (after page write operation) or to 0x54 (after page read operation). NAFEcc register contents will be cleared if NAFMode register bits 6 and 5 are both '1'.
** Only bits 11 to 0 are relevant for column parity, other bits are '0';

The content of NAFEcc depends on the device type.

X8 (8-bit data bus) devices:

Line Parity Block1	: will contain the line	parity of byte 1 to 512 (after 512 r/w cycles)
Column Parity Block1	: will contain the column	parity of byte 1 to 512 (after 512 r/w cycles)
Line Parity Block2	: will contain the line	parity of byte 513 to 1024 (after 1024 r/w cycles)
Column Parity Block2	: will contain the column	parity of byte 513 to 1024 (after 1024 r/w cycles)
Line Parity Block3	: will contain the line	parity of byte 1025 to 1536 (after 1536 r/w cycles)
Column Parity Block3	: will contain the column	parity of byte 1025 to 1536 (after 1536 r/w cycles)
Line Parity Block4	: will contain the line	parity of byte 1537 to 2048 (after 2048 r/w cycles)
Column Parity Block4	: will contain the column	parity of byte 1537 to 2048 (after 2048 r/w cycles)

X16 (16-bit data bus) devices:

Line Parity Block1	: will contain the line	parity of halfword(7:0) 1 to 512 (after 512 r/w cycles)
Column Parity Block1	: will contain the column	parity of halfword(7:0) 1 to 512 (after 512 r/w cycles)
Line Parity Block2	: will contain the line	parity of halfword(15:8) 1 to 512 (after 512 r/w cycles)
Column Parity Block2	: will contain the column	parity of halfword(15:8) 1 to 512 (after 512 r/w cycles)
Line Parity Block3	: will contain the line	parity of halfword(7:0) 513 to 1024 (after 1024 r/w cycles)
Column Parity Block3	: will contain the column	parity of halfword(7:0) 513 to 1024 (after 1024 r/w cycles)
Line Parity Block4	: will contain the line	parity of halfword(15:8) 513 to 1024 (after 1024 r/w cycles)
Column Parity Block4	: will contain the column	parity of halfword(15:8) 513 to 1024 (after 1024 r/w cycles)

Note: Read ECC is not performed in unbuffered READ mode (this means when CPU accesses the Nand Flash through the NAF_DATA registers)

Table 46 NAF data register

Name		Base		Default
NAFData		**AS3525_NAND_FLASH_BASE**		**0x0000**
Offset 0x000C		**Data Register**		
		The NAFData register offers unbuffered access to the data bus of the NAND flash device.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
15:0	NAFData	0x01	R/W	For X8 devices (8-bit data bus) only bits 7:0 are relevant, other bits are ignored For X16 devices (16-bit data bus) all are relevant

Table 47 NAF mode register

Name		Base		Default
NAFMode		**AS3525_NAND_FLASH_BASE**		**0x00**
Offset 0x0010		**Mode register**		
		The NAFMode register controls NAND flash read/write/erase procedures.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	write protection	0x0	R/W	Used to control 'command latch enable' 0: output 'naf_cle' is set to '0' 1: output 'naf_cle' is set to '1' (Command Latch Cycle)
6:5	Ecc [1:0]	0x0	R/W	controls 'address latch enable' 0: output 'naf_ale' is set to '0' 1: output 'naf_ale' is set to '1' (Address Latch Cycle)
4	ce	0x0	R/W	0: power off (all output enable signals are turned off) 1: power on
3	-	0x0	R/W	always '0'
2	power_on	0x0	R/W	0: power off (all output enable signals are turned off) 1: power on
1	ale	0x0	R/W	controls 'address latch enable' 0: output 'naf_ale' is set to '0' 1: output 'naf_ale' is set to '1' (Address Latch Cycle)
0	Cle	0x0	R/W	controls 'command latch enable' 0: output 'naf_cle' is set to '0' 1: output 'naf_cle' is set to '1' (Command Latch Cycle)

Table 48 NAF status register

Name		Base		Default
NAFStatus		**AS3525_NAND_FLASH_BASE**		**-**
Offset 0x0014		**Status Register**		
		The NAFStatus register contains information on the internal status.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
13	fifo_error	0x0	R	FIFO error signal 0: if FIFO is reset 1: if FIFO contains 36 words and FIFO push(write) has occurred or when FIFO contains 0 words and a FIFO pop(read) has occurred. The FIFO error will lock the FIFO and has to be reset by a reset of the FIFO (by setting NAFControl register bit 1 to '1')
12	fifo_full	0x0	R	FIFO full signal 0: if FIFO contains less than 36 words 1: if FIFO contains 36 words
11	fifo_almost_full	0x0	R	FIFO almost_full signal 0: if FIFO contains less than 32 words 1: if FIFO contains more than or equal 32 words
10	fifo_almost_empty	0x0	R	FIFO almost_empty signal 0: if FIFO contains more than 4 words 1: if FIFO contains less than or equal 4 words
9	fifo_empty	0x0	R	FIFO empty signal 0: if FIFO contains more than 0 words 1: = when FIFO contains 0 words
8	strobe_ready	0x0	R	read/write strobe ready signal 0: if read/write strobe '0' (strobe active) 1: if read/write strobe '1' (strobe inactive)
7	flash_ready	0x0	R	synchronised NAND flash ready signal 0: if synchronised input 'naf_busy_in_n' is '0' (busy) 1: if synchronised input 'naf_busy_in_n' is '1' (ready)
6	got_fifo_error	0x0	R	FIFO error indication (edge triggered) 0: if bit 6 of NAFClear register is set to '1' 1: if FIFO contains 36 words and FIFO push(write) occurs or when FIFO contains 0 words and a FIFO pop(read) occurs.
5	got_fifo_full	0x0	R	FIFO full indication (edge triggered) 0: if bit 5 of NAFClear register is set to '1' 1: if FIFO contains 36.
4	got_fifo_high	0x0	R	FIFO high indication (edge triggered) 0: if bit 4 of NAFClear register is set to '1' 1: if FIFO gets full (36 words) or changes from 31 to 32 words (and when the NAFWords register is greater than 32). Note: When this bit gets '1' during 'Page Read' mode, a new FIFO burst read of up to 32 words is possible.
3	got_fifo_low	0x0	R	FIFO low indication (edge triggered) 0: if bit 3 of NAFClear register is set to '1' 1: if FIFO gets empty or changes from 5 to 4 words (and when the NAND Flash requires more than 32 bytes/halfwords). Note: When this bit gets '1' during 'Page Programming' mode, a new FIFO burst write of up to 32 words is possible

Name		Base		Default
NAFStatus		**AS3525_NAND_FLASH_BASE**		**-**
Offset 0x0014		**Status Register**		
		The NAFStatus register contains information on the internal status.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
2	got_empty_and_rdy	0x0	R	NAFWords empty and Controller ready indication (edge triggered) 0: when bit 2 of NAFClear register is set to '1' 1: when read/write strobe changes from '0' to '1' (end of strobe) and NAFWords register has become empty. Note: This bit is used to detect the end of a multiple read/write burst transaction
1	got_strobe_ready	0x0	R	Read/write strobe ready indication (edge triggered) 0: when bit 1 of NAFClear register is set to '1' 1: when read/write strobe changes from '0' to '1' (end of strobe) Note: read/write strobes can last from 3 to 33 PCLK cycles depending on NAFConfig settings.
0	got_flash_ready	0x0	R	NAFWords empty and Controller ready indication (edge triggered) 0: when bit 2 of NAFClear register is set to '1' 1: when read/write strobe changes from '0' to '1' (end of strobe) and NAFWords register has become empty. Note: This bit is used to detect the end of a multiple read/write burst transaction

Table 49 NAF interrupt mask register

Name		Base		Default
NAFMask		**AS3525_NAND_FLASH_BASE**		**0x0018**
Offset 0x0018		**Interrupt Mask Register**		
		The NAFMask register is used to mask/enable the internal interrupt requests.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
6	mask6	0x1	R/W	Mask 'FIFO error indication' interrupt request 0: enable 1: masked
5	mask5	0x1	R/W	Mask 'FIFO full indication' interrupt request 0: enable 1: masked
4	mask4	0x1	R/W	Mask 'FIFO high indication' interrupt request 0: enable 1: masked
3	mask3	0x1	R/W	Mask 'FIFO low indication' interrupt request 0: enable 1: masked
2	mask2	0x1	R/W	Mask 'NAFWords empty and Controller ready indication' interrupt request 0: enable 1: masked
1	mask1	0x1	R/W	Mask 'Read/write strobe ready indication' interrupt request 0: enable 1: masked
0	mask0	0x1	R/W	Mask 'NAND flash ready indication' interrupt request 0: enable 1: masked

Table 50 NAF FiFo Data register

Name		Base		Default
NAFFifodata		**AS3525_NAND_FLASH_BASE**		**0x0000**
Offset 0x001c		**FIFO Data Register**		
		The NAFFifodata register offers access to the internal FIFO.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
32:0	Fifodata [32:0]	-	R/W	Writing this register will push a word on the FIFO and the write address will be incremented by 1. When the FIFO is full (36 words) then a write access on the register is ignored and the FIFO ERROR status bit is set. Reading on this register will pop a word from the FIFO and the read address will be incremented by 1. When the FIFO is empty then a read access on the register is ignored and the FIFO ERROR status bit is set.

Table 51 NAF interrupt mask register

Name		Base		Default
NAFWords		**AS3525_NAND_FLASH_BASE**		**0x0000**
Offset 0x0020		**Interrupt Mask Register**		
		The NAFWords register informs the controller about the maximum words to be transferred and controls the FIFO transfer both in interrupt and DMA mode.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
32:0	Words [32:0]	0x0000	R/W	0: FIFO based data transfer is disabled not 0: FIFO transfer is in progress

Note: For page transfers (program or read) the initial number of words depends on the NAND flash device. For a page size of 512 bytes, an initial word value of 512/4 = 128 has to be written. For a page size of 2k bytes, an initial word value of 512 has to be used.

Table 52 NAF interrupt clear register

Name		Base		Default
NAFClear		**AS3525_NAND_FLASH_BASE**		**0x0018**
Offset 0x0024		**Clear Register**		
		The NAFClear register clears interrupt status information and re-enables interrupt detection.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
6	clear6	-	W	Reset of 'FIFO error indication' status bit 0: no action 1: bit 6 of NAFStatus is reset and interrupt 6 detection is enabled
5	clear5	-	W	Reset of 'FIFO full indication' status bit 0: no action 1: bit 5 of NAFStatus is reset and interrupt 5 detection is enabled
4	clear4	-	W	Reset of 'FIFO high indication' status bit 0: no action 1: bit 4 of NAFStatus is reset and interrupt 4 detection is enabled
3	clear3	-	W	Reset of 'FIFO low indication' status bit 0:no action 1:bit 3 of NAFStatus is reset and interrupt 3 detection is enabled
2	clear2	-	W	Reset of 'NAFWords empty and Controller ready indication' status bit 0:no action 1:bit 2 of NAFStatus is reset and interrupt 2 detection is enabled
1	clear1	-	W	Reset of 'Read/write strobe ready indication' status bit 0: no action 1: bit 1 of NAFStatus is reset and interrupt 1 detection is enabled
0	clear0	-	W	Reset of 'Read/write strobe ready indication' status bit 0:no action 1: bit 0 of NAFStatus is reset and interrupt 0 detection is enabled

Table 53 NAF test register

Name		Base		Default
NAFTest		**AS3525_NAND_FLASH_BASE**		**0x0000**
Offset 0x0028		**Test Register**		
		The NAFTest register is used for functional tests of the FIFO.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
1	datainvert		W	0: default mode 1: disables FIFO access by the internal controller => FIFO is accessed by APB interface only
0	fifotest	-	W	0: default mode 1: data word both on FIFO input and output is inverted

5.3.12 DBOP - Data Block Output Port

Purpose of this ARM APB peripheral module is a high-speed data output port that can support data transfer to various display controllers based on synchronous control interfaces. Programmability of polarity and timing of the generated control signals makes it possible to support various kinds of displays. Example of a supported display controller is the Hitachi HD77766R LCDE controller.

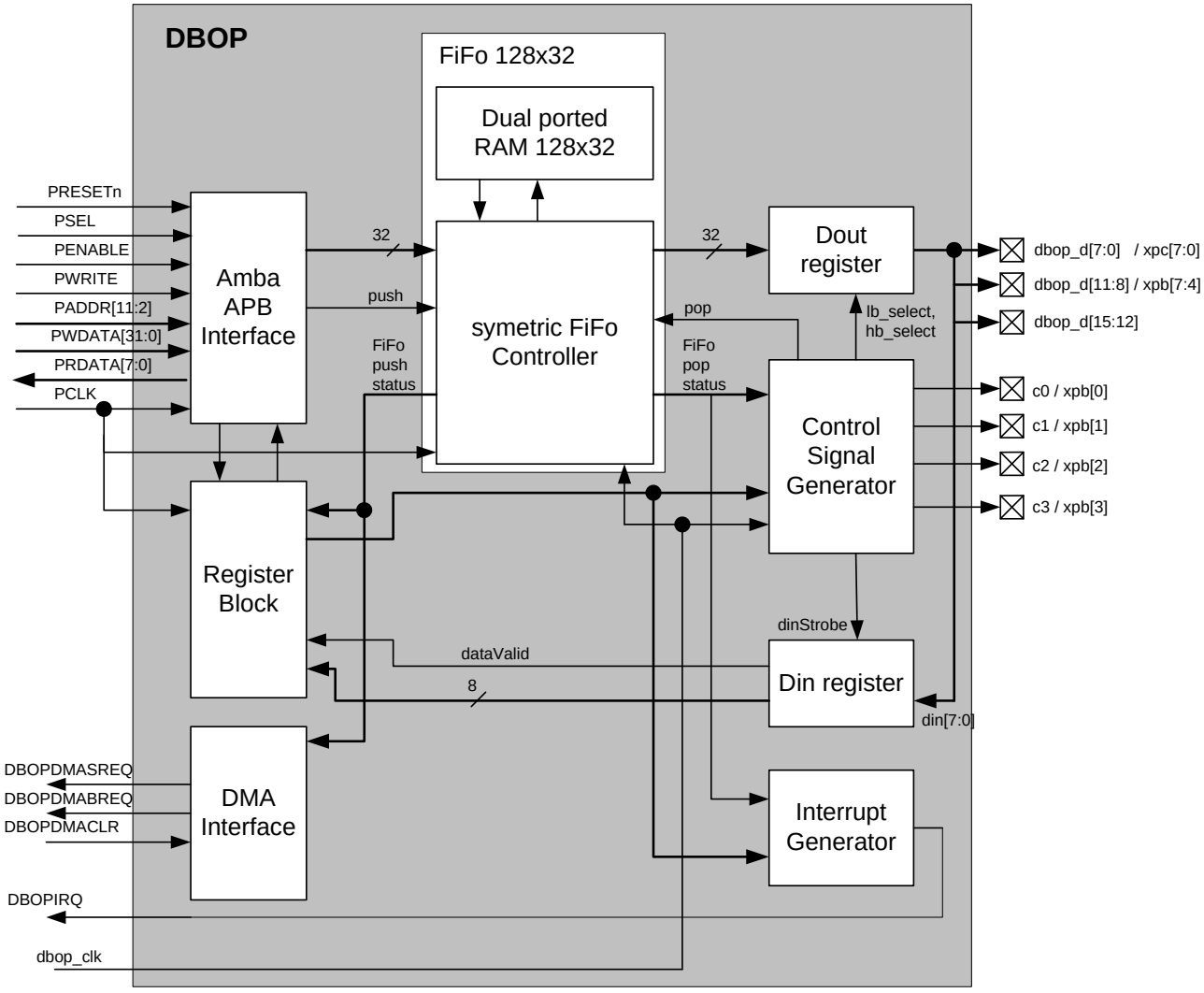
From the programmers point of view the DBOP module can be serviced by DMA accesses. With the large size of the data FIFO and the programmable interrupt request conditions the overhead for SW is minimised. Simple read instructions to read for example a status register of the LCD controller are also supported.

The usage of this cell results in a great performance boost compared to the standard ARM GPIO PrimeCell™ architecture.

Features

- APB bus interface
- support for direct memory access (DMA)
- data output FIFO with 128 words (32 bit wide)
- 8 or 16 bit parallel data output (configurable)
- 4 control outputs - flexible programming of the signal waveforms with respect to polarity and timing
- programmable even/odd control output generation
- 8 or 16 bit parallel data input register with programmable read strobe
- programmable conditions for interrupt generation based on FIFO flags
- usage of FIFO for simple division of APB clock domain and output clock domain
- programmable data output rate in range of 0.05 to 4 MHz
- APB Clock & DBOP Clocks are synchronous.

Figure 32 DBOP Block Diagram



5.3.12.1 DBOP register definitions

Table 54 DBOP Registers

Register Name	Base Address	Offset	Note
DBOP_TIMPOL_01	AS3525_DBOP_BASE	0x00	Timing and polarity for control 0 and 1
DBOP_TIMPOL_23	AS3525_DBOP_BASE	0x04	Timing and polarity for control 1 and 2
DBOP_CTRL_REG	AS3525_DBOP_BASE	0x08	Control Register
DBOP_STAT_REG	AS3525_DBOP_BASE	0x0C	Status Register
DBOP_DOUT_REG	AS3525_DBOP_BASE	0x10	Data output register
DBOP_DIN_REG	AS3525_DBOP_BASE	0x14	Data input register

Timing & Polarity Control register TPC01

This register contains all information necessary for definition of control signals C0 and C1.

Table 55 DBOP control registers C0 and C1

Register bits	Name	type	function	default value	
31	c1_p0	r/w	polarity 1	0	
30	c1_p1	r/w	polarity 2	1	
29	c1_p2	r/w	polarity 3	0	
28:24	c1_t1	r/w	Time 1	0xA	
23:19	c1_t2	r/w	Time 2	0x14	
18	c1_ev	r/w	even enable	1	
17	c1_od	r/w	odd enable	1	
16	c1_qs	r/w	quiescent state	0	
15	c0_p0	r/w	polarity 1	0	
14	c0_p1	r/w	polarity 2	1	
13	c0_p2	r/w	polarity 3	0	
12:8	c0_t1	r/w	Time 1	0xA	
7:3	c0_t2	r/w	Time 2	0x14	
2	c0_ev	r/w	even enable	1	
1	c0_od	r/w	odd enable	1	
0	c0_qs	r/w	quiescent state	0	

Timing & Polarity Control register TPC23

This register contains all information necessary for definition of control signals C2 and C3.

Table 56 DBOP control registers C2 and C3

Register bits	Name	type	function	default value	
31	c3_p0	r/w	polarity 1	0	
30	c3_p1	r/w	polarity 2	1	
29	c3_p2	r/w	polarity 3	0	
28:24	c3_t1	r/w	Time 1	0xA	
23:19	c3_t2	r/w	Time 2	0x14	
18	c3_ev	r/w	even enable	1	
17	c3_od	r/w	odd enable	1	
16	c3_qs	r/w	quiescent state	0	
15	c2_p0	r/w	polarity 1	0	
14	c2_p1	r/w	polarity 2	1	
13	c2_p2	r/w	polarity 3	0	
12:8	c2_t1	r/w	Time 1	0xA	
7:3	c2_t2	r/w	Time 2	0x14	
2	c2_ev	r/w	even enable	1	
1	c2_od	r/w	odd enable	1	
0	c2_qs	r/w	quiescent state	0	

Table 57 DBOP control register

Register bits	Name	type	function	default value	
31:22			*reserved*		
21	clr_pop_err	W	*Interrupt clear signal for pop error interrupt*	0	Writing 1 to this bit will clear the pop error interrupt. Writing 0 has no effect.
20	clr_push_err	W	*Interrupt clear signal for push error interrupt*	0	Writing 1 to this bit will clear the push error interrupt. Writing 0 has no effect .
19	en_data	r/w	*Tri-state enable for dout bus*	0	When set, dout bus is tri-stated when there is no active write on the bus.
18	sdc	r/w	*short count bit*	0	
17	res_even	r/w	reset to even cycle	0	when set, next output cycle is even
16	enw	r/w	enable write	0	0: write disabled 1: write enabled
15	strd	r/w	start read	0	
14:13	osm	r/w	output serial mode	0	0: single word out 1: 2 serial words out 2: 4 serial words out
12	ow	r/w	output data width	0	0: 8 bit data width 1: 16 bit data width
11	ir_enable	r/w	IR enable		0: all IR disabled 1: IR enabled
10	ir_po_err	r/w	IR enable on pop error	0	
9	ir_pu_err	r/w	IR enable on push error	0	
8	ir_e_en	r/w	IR enable set on push empty	0	
7	ir_ae_en	r/w	IR enable set on push almost empty	0	
6	ir_af_en	r/w	IR enabbe set on push almost full	0	
5	ir_f_en	r/w	IR enable set on push full	0	
4:0	rs_t	r/w	read strobe time	0x1F	

Notes:
- If the start read bit is issued by setting the strd bit to 1, a single read cycle is generated. After this read cycle the strd bit is set to 0 again by HW.
- If write is enabled by setting enw=1, no read is possible (strd does not cause any action).
- res_even is a reset bit that defines the start of even/odd generated signals. With res_even bit set, the next output cycle is a even cycle. Within this first even output cycle the res_even bit is set to 0 by the SW.
- sdc selects the counter length for the timing generator. Default is end value of 31. With sdc set to 1, the count end value is 15.
- en_data is used as a tri-state enable for the dout bus . When set as 1, dout is tri-stated if there is no active write on the bus . When this bit is set as 0, dout is bus is tri-stated only during the read cycle.

Table 58: DBOP status register

Register bits	Name	type	function	default value	
31:17			*reserved*		
16	rd_d_valid	r	read data valid		
15:12	Reserved				
11	fi_pu_err	f	push error		
10	fi_pu_e	r	push fifo empty		
9	fi_pu_ae	r	push fifo almost empty		
8	fi_pu_hf	r	push fifo half full		
7	fi_pu_af	r	push fifo almost full		
6	fi_pu_f	r	push fifo full		
5	fi_po_err	r	pop error		
4	fi_po_e	r	pop fifo empty		
3	fi_po_ae	r	pop fifo almost empty		
2	fi_po_hf	r	pop fifo half full		
1	fi_po_af	r	pop fifo almost full		
0	fi_po_f	r	pop fifo full		

The read data valid flag is cleared with every start read and set after read data strobe is issued (at read data valid 1 the data can be readout by SW).

Data Output Register

32 bit register for data output - the data written to this register are directly written to the FiFo. Depending on the serial output mode and the output data width, the effective register width of this register is 8, 16 or 32 bits.
Following table shows the effective data width for this register:

	osm=0	osm=1	osm=2
odw = 0	8 (byte0)	16 (byte0, byte1)	32 (byte0, byte1, byte2, byte3)
odw = 1	16 (HW0)	32 (HW0, HW1)	32 (HW0, HW1)

Depending on odw,

- either one, two or four bytes are transmitted serially for odw=0
- or one or two half words (HW = 16 bits) are transmitted serially for odw=1.

Note that for the 8 or 16 bit width only a part of the FiFo memory is used (to keep HW design simple).

Data Input Register

16 bit data input register that holds the value of the last read cycle. It is only valid if the data valid flag is set in the status register. No interrupt support is given, for data input the read data valid flag must be polled.

Dbop Integration Test Registers

The Dbop module is programmed to integration test mode using test control register. The integration test mode enables the user to access all the input/output pins through the APB bus interface.

Name	Offset	R/W	Reset Value	Description
DBOPITC	0x18	R/W	0x00000000	DBOP integration test control register
DBOPITIP1	0x1C	R/W	0x00	DBOP integration test input register
DBOPITOP1	0x20	R	0x0	DBOP integration test output register

Table 59 DBOPITC test register

Register bits	Name	type	function	default value	
31:1			*reserved*		
0	iten	r/w	Integration test enable		1 will enable the integration test mode

Table 60 DBOPITIP1 test register

Register bits	Name	type	function	default value	
31:5			*reserved*		
4	Testctrloen	r/w	Test value for out_enControl_n	0	The value on this bit will be reflected in out_enControl_n
3	Testdataoen	r/w	Test value for out_enData_n	0	The value on this bit will be reflected in out_enData_n
2	Testdmasreq	r/w	Test value for DMASREQ	0	The value on this bit will be reflected in DBOPDMACSREQ
1	Testdmabreq	r/w	Test value for DMABREQ	0	The value on this bit will be reflected in DBOPDMACBREQ
0	testirq	r/w	Test value for interrupt	0	The value on this bit will be reflected in DBOPIRQ

Table 61 DBOPITOP1 test register

Register bits	Name	type	function	default value	
31:1			*reserved*		
0	Testdmaclr	r	*DBOPDMACCLR test register.*	0	Read of this register will return the value on the DBOPDMACCLR input.

5.3.12.2 DBOP DMA Interface

This block generates all necessary interface signals with the DMAC primecell for DMA transfer. Following table gives a description of these signals.

DBOPDMASREQ	single word request, asserted by DBOP. This signal is asserted when there is at least one empty location in the FiFo
DBOPDMABREQ	burst DMA transfer request, asserted by DBOP. This signal is asserted when there are at least four empty locations in the FiFo
DBOPDMACLR	DMA request clear, asserted by DMA controller to clear the DMA request signals. If DMA burst transfer is requested, the clear signal is asserted during the transfer of the last data in the burst

Symmetric FiFo

The FiFo buffer has two main purposes:

- data buffering: the FiFo contains 128 locations with 32 bits for data storage: with according DMA transfer, the data can be transferred in short time without need for any SW control

- clock domain crossing: the FiFo is at the boarder of clock domain PCLK and DBOPCLK. All necessary synchronisation is done internally. All flags are available as push flags (synchronised to the push clock PCLK) and pop flags (synchronised to the POP clk, which is synchronous to DBOPCLK.

The FiFo controller gives empty, almost empty, half full, almost full and full flags which are available in two fashions: synchronous to the push or the pop side (pop_empty, push_empty, ...).

5.3.12.3 Control Signal Generator

Four independent control signals can be generated: typical application for such signals is a 80xx interface with RS, RD*, WR* and E or a 68xx interface with RS, E, RWN. The idea of this control signal generator is a general-purpose block, which generates any signal timing/waveform that is necessary to transfer the data to any specific display.

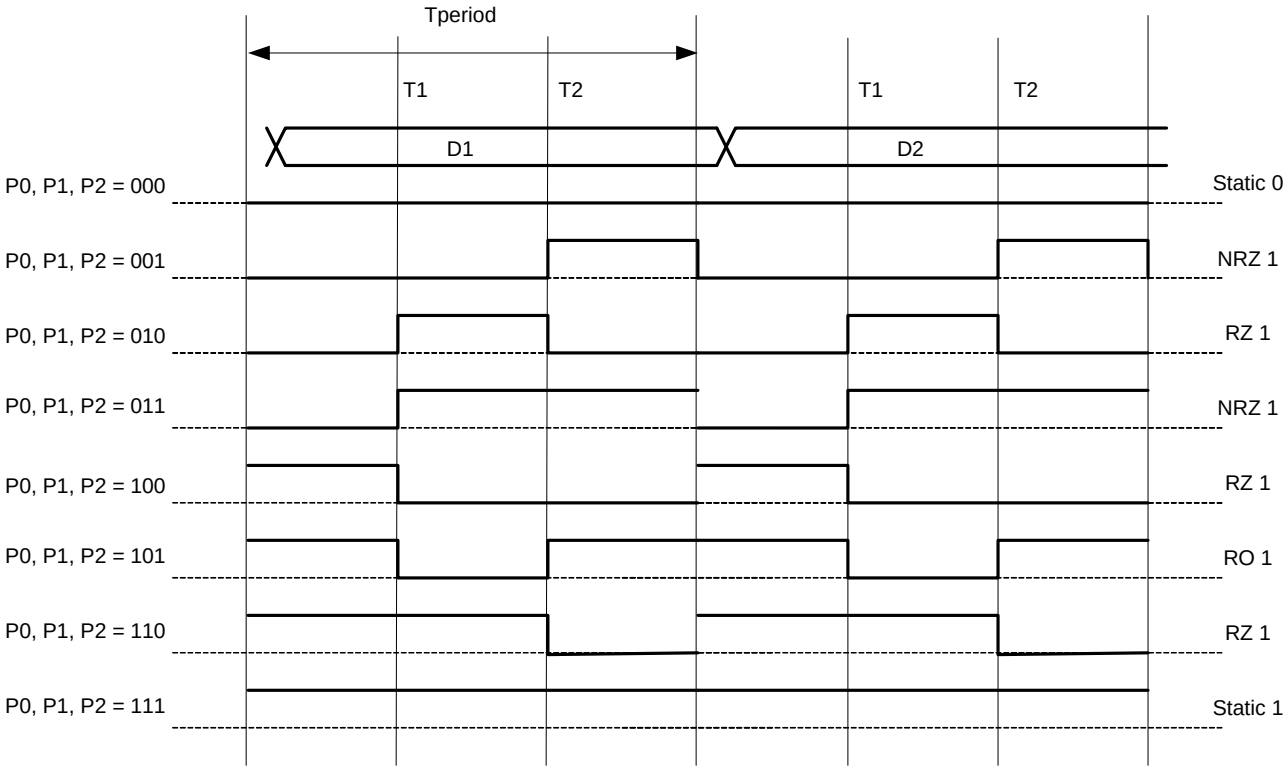
Polarity Parameters

For each of the control signals c0 - c3 following polarity parameters are defined:

- p0 … polarity 0 at start of cycle
- p1 … polarity 1 following polarity 0
- p2 … polarity 2 following polarity 1

Following figure shows an example for timing waveforms defined with these control parameters.

Figure 33 DBOP timing waveform



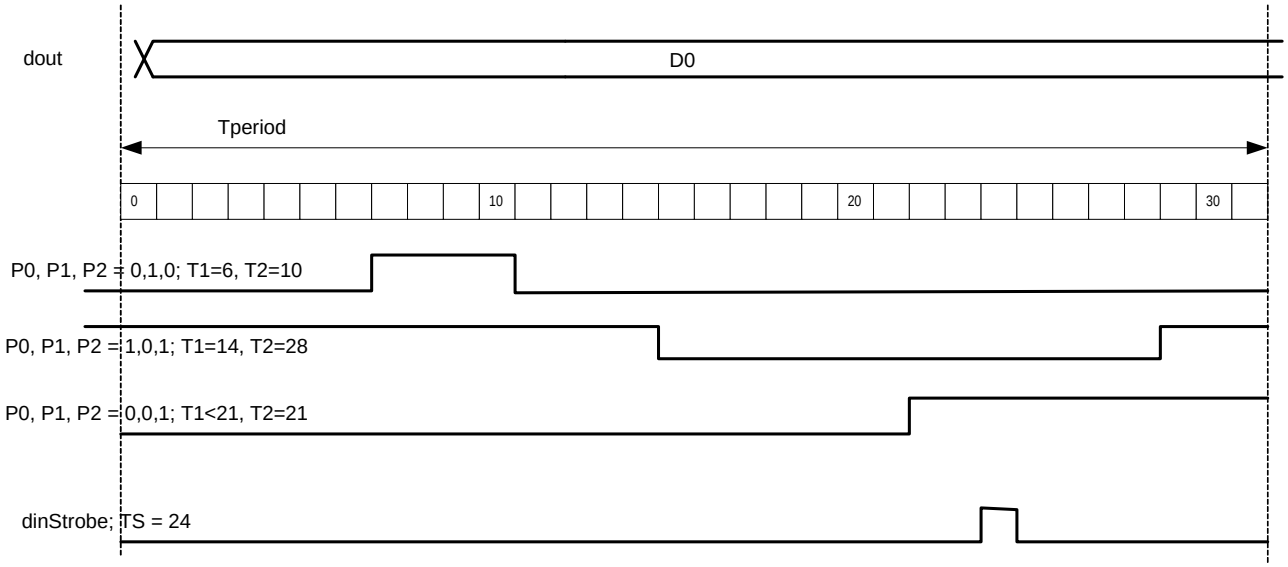
Quiescent State

The control signals are only generated with each data output cycle (data output cycles are generated as long as the FiFo is not empty). With FiFo empty and in the absence of a read cycle, all control signals are set to a quiescent state. For each control signal, this quiescent state can be programmed either to 1 or 0.

Timing Parameters

Also the time points for change from p0-p1 (t1) and p1-p2 (t2) can be programmed. For these programmable timing parameters each data output cycle is divided into 32 steps. Both T1 and T2 can be in the range of 0 to 31. For short count bit set (sdc bit in control register), T1 and T2 must be in the range of 0 to 15.

Figure 34 DBOP timing parameters



Even/odd generated signals

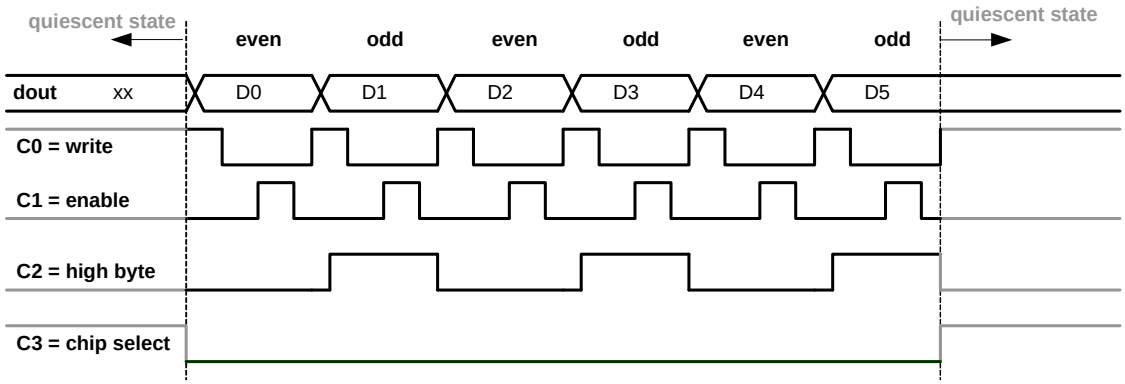
In addition to these timing parameters, signals can also be programmed to go active only during the even (0, 2, 4, …) or the odd cycles (1, 3, 5, …). For example the indication of even/odd bytes for the case that two bytes in serial are transmitted can be used.

Two control bits are used to set this signal behaviour:

evenEnable, oddEnable. For default, both are set to 1 and both cycles will appear. For cases where even or odd should be omitted, set according evenEnable/oddEnable to 0. With both set to 0, no cycles will appear at the output!

Following example illustrates a typical waveform for an output interface where the evenEnable=0 and oddEnable=1 for control signal C2. In this example, C2 is an active high indication of the high byte (D1, D3, D5, …).

Figure 35 DBOP even/odd generated signals waveforms



Normally the even/odd cycles are toggling all the time, also if there are quiescent states in between. To have the possibility of defining a new start, reset of this even/odd counter can be done via the res_even bit inside of the control register. With res_even set, the counter starts with an even cycle. Res_even is then set to 0 again by SW at the new start.

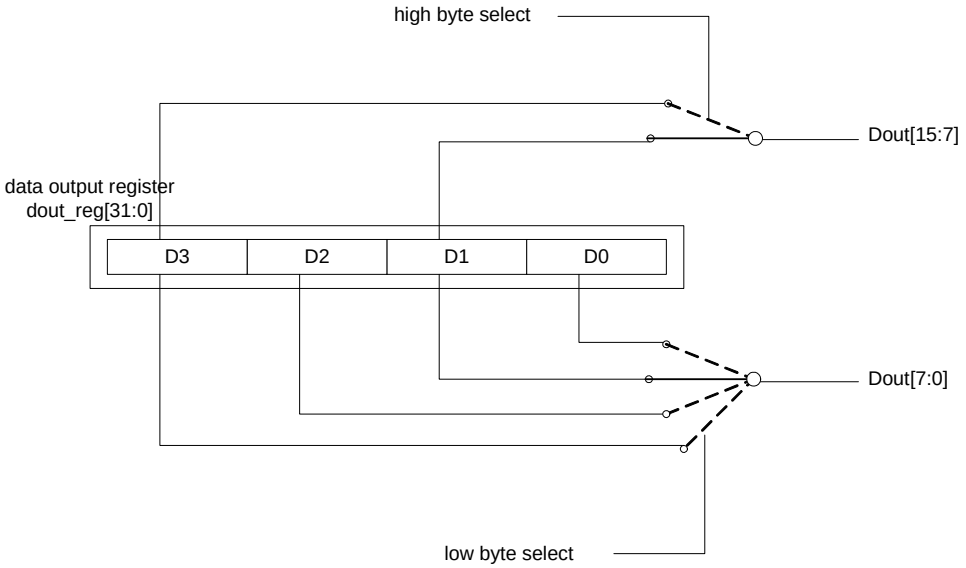
Input strobe generation

In addition to the generation of the control signals, also an input strobe signal dinStrobe is generated within the control signal generator. With active dinStrobe, the input data are strobed with rising clock edge (see DIN register).

5.3.12.4 Data Output Register

The data output register handles different output widths and serial output mode (selected by parameters osm and odw). Following diagram illustrates the function of the data output register.

Table 62 DBOP data output register



The control part generates the according signals for low byte select and high byte select.

5.3.12.5 DIN register

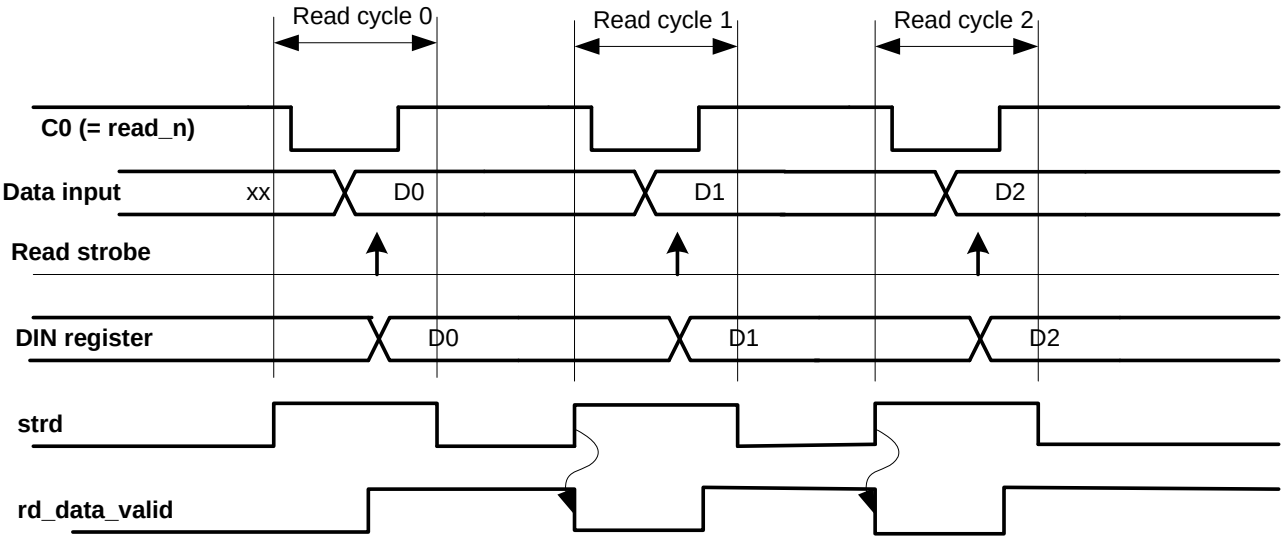
With the dinStrobe, data are written to the DIN register. This gives a simple mechanism, in which for example the status data can be read from a LCD display interface.

To do a data read, first the START READ (strd) bit is programmed into the control register. With START READ, the control signal generator starts to generate one cycle with the according control signals. Data are strobed by the programmed strobe time into the din register. After the cycle is completed the HW resets the strd bit to 0. With set of the strd bit, the rd_data_valid bit is also reseted.

The SW just has to poll the rd_data_valid bit, when the bit gets set the input data can be read from the din register. After read cycle, the control signal generator returns to the quiescent state.

Following timing diagram shows an example of three read cycles.

Figure 36 DBOP read cycle example



Note: Be aware that the read cycle should only be activated when there is no active write cycle (FiFo is empty). Otherwise the results of such action get unpredictable.

For any read cycle, the write enable bit must be set to 0 (write disabled).

start read (strd)	write enable (wen)	FiFo empty Status	DBOP function
0	0	0	quiescent
0	0	1	quiescent
0	1	0	valid write
0	1	1	quiescent
1	0	0	valid read
1	0	1	valid read
1	1	0	valid write
1	1	1	quiescent

5.3.12.6 Interrupt Generator

Depending on the FiFo Status, an interrupt request can be generated. The conditions that cause an interrupt are set within the control register.

The interrupt output DBOPIRQ is active high.

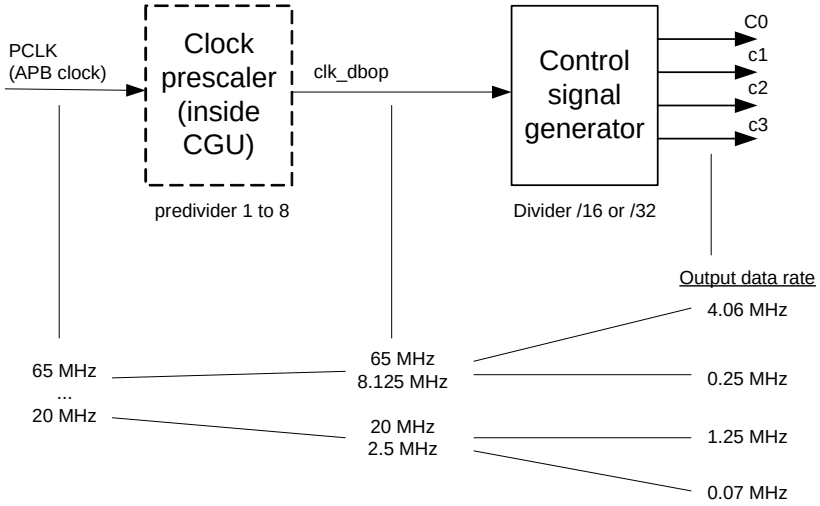
5.3.12.7 Clock frequencies

The input clock is directly taken from the PCLK clock. A programmable prescaler is implemented within the CGU. Input clock for the prescaler is in the range of 20 - 60 MHz.

Programmable division factors for the prescaler in the range of 1 to 8. Input clock to the module is in the range of 2.5 to 60 MHz.

Within the module the control signal generator is doing a division by 16 or 32 (selectable). So the effective output data rates are in the range of 1.25 to 4 MHz for maximum performance and can be scaled down in the range of 0.07 to 0.25 MHz.

Figure 37 DBOP data rate



Time constraining for the module should be done with 65 MHz, if there is a demand the time constraints for the output pads can be reduced.

5.3.12.8 Interface with GPIO PINs / additional PINs

For the SW, the usage of either ARM primecell GPIO ports or DBOP port can be configured with the GPIOAFSEL registers.

Following IO ports are used for the basic 8 bit interface

xpc[7:0] **for dout[7:0] and din[7:0]**
xpb[3:0] **for {C3, C2, C1, C0}**

Following IO ports are used for the optional 16 bit interface

xpb[7:4] **for dout[11:8] and din[11:8]**
dbop_d[15:12] **for dout[15:12] and din[15:12]**

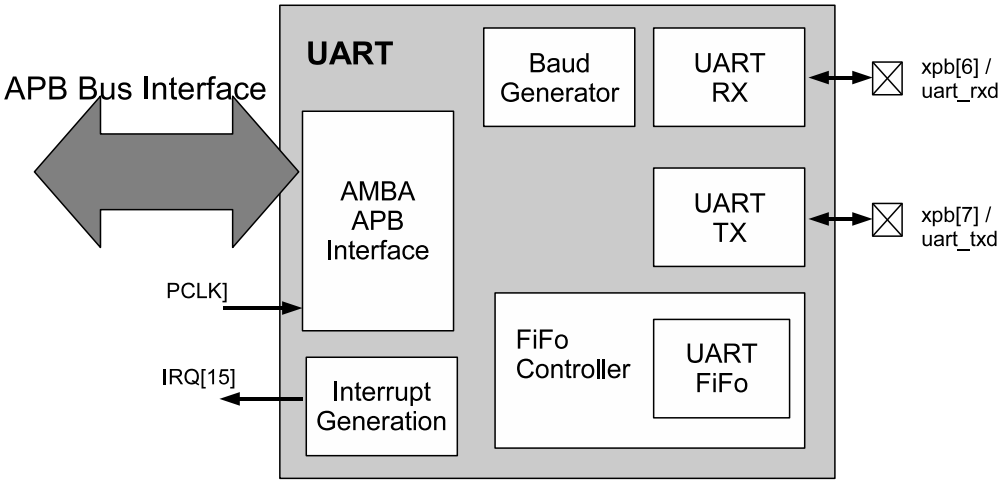
5.3.13 UART – Universal Asynchronous Receiver/Transmitter

The UART is a Universal Asynchronous Receiver Transmitter compatible to industry standard 16550 with APB slave interface. This UART provides FIFO based transmitter-receiver pair with programmable Baud-rate, character widths and parity encoding. Status and error information is also provided by the design. Maximum baud rate supported by this UART is 1Mbps for input clock of 16MHz.

Features

- Compliance to Industry Standard 16550 UART.
- APB slave interface.
- Separate 16x8 Transmit and 16x11 Receive FIFOs.
- Programmable FIFO disabling for 1-byte depth.
- Programmable Baud rate Generator.
- Independent masking for transmit, receive and Error interrupts.
- False Start bit detection.
- Line Break generation and detection.
- Fully programmable serial interface characteristics:
  - Supports 5,6,7 and 8 bits.
  - even, odd, stick and no parity generation and detection.
  - 1, 11/2 and 2 stop bits.

Figure 38 UART Block Diagram



5.3.13.1 UART Baud Generator and Clock Divider Settings

The internal baud generator module generates the required baud clock using the divisor register value. To achieve correct synchronizationincoming bits are over sampled by a factor of 16x. Software should program the divisor value by which the system clock has to be divided to achieve the required baud clock frequency.

The equation to calculate baud divisor is

Baud Divisor = (input frequency) ÷ (baud rate x 16)

Important: the internal clock divider must be set to a value of 2 or higher. Setting the value to 1 (no division) is not allowed!

For example, for 16 MHz PCLK clock following table gives the list of settings for different BAUD rates.

Baud Rate	Required Baud clock frequency	Decimal divisor value
50	800	20000
75	1200	13333
110	1760	9091
134.5	2152	7435
150	2400	6667
300	4800	3333
600	9600	1667
1200	19200	833
1800	28800	556
2000	32000	500
4800	76800	208
7200	115200	139
9600	153600	104
19200	307200	52
38400	614400	26
56000	896000	18
128000	204800	8
250000	4000000	4
300000	4800000	3
500000	8000000	2

5.3.13.2 UART Register Descriptions

All registers are 8 bits wide. Registers are selected based on the address and the value of Divisor Latch Select (DLS) bit in the line control register (UART_LNCTR_REG).

Table 63 UART registers

Register Name	Base Address	Offset	DLS	Note
UART_DATA_REG	AS3525_UART_BASE	0x00	0	Data register (Rx / Tx)
UART_DLO_REG	AS3525_UART_BASE	0x00	1	Clock divider lower byte register
UART_DHI_REG	AS3525_UART_BASE	0x04	1	Clock divider higher byte register
UART_INTEN_REG	AS3525_UART_BASE	0x04	0	Interrupt enable register
UART_INTSTATUS_REG	AS3525_UART_BASE	0x08		Interrupt status register
UART_FCTL_REG	AS3525_UART_BASE	0x0C		FIFO control register
UART_LNCTL_REG	AS3525_UART_BASE	0x10		Line control register
UART_LNSTATUS_REG	AS3525_UART_BASE	0x14		Line status register

Table 64 UART Data Register

Name	Base	Default
UART_DATA_REG	**AS3525_UART_BASE**	**0xC8110000**

Offset: 0x00 **DLS bit set to 0**	**Data register** Holds the data byte received or the data byte to be transmitted respectively. RX: This register holds the received data byte. In FIFO mode, this byte will be the top byte of the 16-byte FIFO. If FIFO mode is disabled, it will be the content of the receive shift register after a byte has been shifted in. A read to the address value 3b000 with Divisor Latch Select (DLS) bit 1'b0 will give the content of this register. If a character less than 8 bits is received, extra zero bits will be padded to this register. TX: This register contains the data to be transmitted. This register will be written by the processor. In FIFO mode, a write to this address will write data into the FIFO. In FIFO mode, top byte of txFIFO is passed on to transmitter shift register. If FIFO is disabled, a write to the address 3'b000 with DLS bit 1'b0 will write into this register. If FIFO is disabled, this register will be overwritten with new data. If FIFO is disabled, data in this register will be passed on to transmitter shift register.

Bit	Bit Name	Default	Access	Bit Description
7:0	UART_DATA_REG	00000000	RW	Holds the data byte received or the data byte to be transmitted respectively.

Table 65 UART Clock divider lower byte register

Name		Base		Default
UART_DLO_REG		**AS3525_UART_BASE**		**0xC8110000**
Offset: 0x00 **DLS set to 1**		**Clock divider lower byte register** This register holds the clock divider value (decimal) which is used to derive the baud clock. To achieve a desired baud rate, the baud clock should be 16-times higher then the baud rate. To derive this clock the ratio of the system clock and the required baud clock should be calculated and the value should be programmed into the clock divider lower byte and higher byte registers (UART_DLO_REG and UART_DHI_REG). Clock divider value = (input frequency) / (baud rate x 16)		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7:0	UART_DLO_REG	00000000	W	This register holds the lower byte of the decimal divisor value to calculate baud clock.

Table 66 UART Clock divider higher byte register

Name		Base		Default
UART_DHI_REG		**AS3525_UART_BASE**		**0xC8110000**
Offset: 0x04 **DLS set to 1**		**Clock divider higher byte register** This register holds the higher byte of the decimal divisor value to calculate baud clock.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7:0	UART_DHI_REG	00000000	W	This register holds the higher byte of the decimal divisor value to calculate baud clock.

Table 67 UART Interrupt enable register

Name		Base		Default
UART_INTEN_REG		**AS3525_UART_BASE**		**0xC8110000**
Offset: 0x04 **DLS set to 0**		**Interrupt enable register** This register will enable the three types of interrupts. Setting the bits of this register to logic 1 enables the selected interrupt.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7:3	Reserved	00000		These bits are reserved for future use.
2	lnStatusEn	0	W	This bit enables the “rxLineStatus” interrupt.
1	txDataEmptyEn	0	W	This bit enables the “txDataEmpty” interrupt.
0	rxDataRdyEn	0	W	This bit enables the “rxDataRdy” interrupt.

Table 68 UART Interrupt status register

Name		Base		Default
UART_INTSTATUS_REG		**AS3525_UART_BASE**		**0xC8110000**
Offset: 0x08		**Interrupt status register**		
		This register will give the status of the interrupt. Depending on the enabled interrupt bits in the interrupt enable register (UART_INTEN_REG) different interrupts will be generated and the status will be updated in this register. On sensing an interrupt the software should read this register to get the status of the interrupt.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7:3	Reserved	00000		These bits are reserved for future use.
2	rxLineStatus	0	RU	This interrupt is set on any error condition on the receive line. There are four types of error possibilities. These error conditions are set in bits 4:1 of the line status register (UART_LNSTATUS_REG). This bit is reset on a read of the line status register (UART_LNSTATUS_REG).
1	txDataEmpty	0	RU	In FIFO mode this bit is set when txFIFO is empty. If FIFO mode is disabled this interrupt is set if the data register (UART_DATA_REG (Tx)) is empty. This bit will be reset on write to the data register (UART_DATA_REG (Tx)).
0	rxDataRdy	0	RU	This is the data ready interrupt. In FIFO mode this bit is set when the number of bytes in the FIFO reaches the trigger level. This bit is also set in FIFO mode when a timeout occurs in the reception, i.e. Rx line idle for more than 4 char times and there is data in the FIFO. If FIFO mode is disabled this bit is set when one full byte is received. This bit is cleared when the FIFO is empty or the data register (UART_DATA_REG (Rx)) is read.

Table 69 UART FIFO control register

Name		Base		Default
UART_FCTL_REG		**AS3525_UART_BASE**		**0xC8110000**
Offset: 0x0C		**FIFO control register**		
		This register holds the control parameters to control receive (rx) and transmit (tx) FIFO. The parameters will enable the FIFOs, set the receiver trigger level, etc.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7:5	Reserved	000		These bits are reserved for future use.
4:3	trigLevel	00	W	These two bits will select the trigger level for the rxFIFO. Once the FIFO pointer reaches this level rxDataRdy interrupt is asserted. 00: 01 byte 01: 04 bytes 10: 08 bytes 11: 14 bytes
2	rxFIFORst	0	W	This bit will reset rxFIFO pointers and clear all the bytes in the rxFIFO. This bit is self clearing, i.e. after resetting FIFO this bit will become zero.
1	txFIFORst	0	W	This bit will reset txFIFO pointers and clear all the bytes in the txFIFO. This bit is self clearing, i.e. after resetting FIFO this bit will become zero.
0	FIFOModeEn	0	W	This bit will enable the FIFO mode. By default this will be reset.

Table 70 UART Line control register

Name		Base		Default
UART_LNCTL_REG		**AS3525_UART_BASE**		**0xC8110000**
Offset: 0x10		**Line control register**		
		This register controls the asynchronous data. Parameters in this register set the transmit and receive character format, the data length, parity bit, stop bit length, etc.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	DLS	0	RW	Divisor Latch Select Bit. This bit is used to select Divisor Latch registers. 1: Divisor Latch registers can be accessed. To access other registers this bit should be zero.
6	breakCntl	0	RW	1: Will cause a break condition to be transmitted, i.e. TX line is pulled low. Normal transmission can be recovered once this bit is cleared. Transmitter logic can be used as break timer.
5	stickParity	0	RW	1: If this bit is set, along with parityEn a fixed parity bit will be transmitted and expected. This fixed parity bit will be the complement of the bit 4.
4	evenParity	0	RW	0: Data byte along with parity bit will be sent and expected to be odd parity. 1: Data byte along with the parity bit will be even parity.
3	parityEn	0	RW	Enable parity bit. 0: Data byte will be transmitted and received without parity bit. 1: Will enable the parity bit at the end of the data byte.
2	stopBits	0	RW	This bit decides how many stop bits should be sent along with a data byte. 0: 1 stop bit transmitted 1: 2 stop bits transmitted if 6, 7 or 8 bit wordLenSel 1: 1.5 stop bits transmitted if 5 bit wordLenSel Receiver will always check for one stop bit.
1:0	wordLenSel	00	RW	These bits will select the number of data bits to be transmitted and received. 00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits

Table 71 UART Line status register

Name		Base		Default
UART_LNSTATUS_REG		**AS3525_UART_BASE**		**0xC8110000**
Offset: 0x14		**Line status register**		
		This register holds the status information of the data transfer. It gives information about the received data.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	FIFODataError	0	RU	1: This bit is set when any data character in the FIFO has parity or framing error or break condition. 0: This bit is reset once the line status register (LINE_STATUS_REG) is read.
6	Reserved	0		This bit is reserved for future use.
5	txHoldRegEmpty	0	RU	This bit is associated with the txDataEmpty interrupt. 1: Indicates that there is no data in txFIFO or the data register (UART_DATA_REG (Tx)). This bit is set once the data is shifted out. 0: This bit is reset once data is written into the data register (UART_DATA_REG (Tx)).
4	breakDetect	0	RU	This bit is associated with the rxLineStatus interrupt. 1: This bit is set if a break condition is detected, i.e. if a zero is detected on receive line for one full character duration. This condition will always cause framingError condition.
3	framingError	0	RU	This bit is associated with the rxLineStatus interrupt. 1: Indicates that the first stop bit of the received data byte is not valid, i.e. a zero is received in place of a one. This error condition causes the receiver to re-synchronize.
2	parityError	0	RU	This bit is associated with rxLineStatus interrupt. 1: Indicates that parity of the received data byte is different from the expected parity as set in the line control register (UART_LNCTL_REG).
1	overrunError	0	RU	This bit is associated with the rxLineStatus interrupt. 1: Indicates an error condition which occurs when one character is fully assembled by the receiver but there is no space to write that byte. In FIFO mode, the content of the FIFO remains unaffected. If FIFO is disabled, the data register (UART_DATA_REG (Rx)) will be overwritten with the new data.
0	dataReady	0	RU	0: There is no data available. 1: There are one or more data bytes ready to be read by the processor.

5.3.14 CGU - Clock generation unit

The clock generation unit generates all clocks for all modules on the chip.

- Hardware programmable selection of clock input either from internal oscillator or external clock input
- Two on-chip PLL circuits for generation of internal clocks
- Programmable divider for generation of ARM922T clock (fclk)
- Programmable divider for generation of AMBA bus clock (pclk)
- Support of ARM922T fastbus, synchronous and asynchronous mode
- Included clock gating registersto optimise power consumption
- Three clock busses at input of all dividers (clk_main, clk_a, clk_b) for utmost flexibility
- Spike-free switches between divider clock inputs (clk_main, clk_a, clk_b)
- Independent clock dividers for peripheral modules

System startup

At startup, the system is configured in a way to run without the need of PLLs. PLLs are disabled and clk_main is used for generation of the clock for the ARM controller (fclk) and ARM AMBA bus (pclk). Within the clock gating register, only the clocks that are really necessary for initial boot are enabled: clock for ARM, for the internal 1-TRAM memory, for the internal ROM and for the external memory. So the boot loader can start either from internal ROM or from the external MPMC.

Clock switching

The system can be reconfigured to run from PLLA or PLLB. Because the 1-TRAM is a dynamic memory that must always get the clock for the internal memory refresh, this switching must be implemented in a way that the PCLK clock is never stopped. The easiest solution to fulfil this requirement is always switching back to clk_main for reconfiguring the PLLs. After reprogramming of the PLLs it must be checked that the PLLs are locked before the system is switched onto the PLL output frequency.

ARM922T and AMBA bus clock

The ARM processor can run in different modes. These modes can be set within the iA, nF bits of the ARM922T CP15 (coprocessor) register 1.

Fastbus mode

This is the default mode after startup. The ARM922T input clock frequency is the same as the AHB/APB bus frequency.

Synchronous mode

Within the synchronous mode, the ARM922T frequency must be higher than the AHB/APB bus frequency and it must be an integer multiple of the AHB/APB bus frequency. Advantage of the synchronous mode is a higher performance because of less synchronisation effort between the ARM922T and the AHB bus.

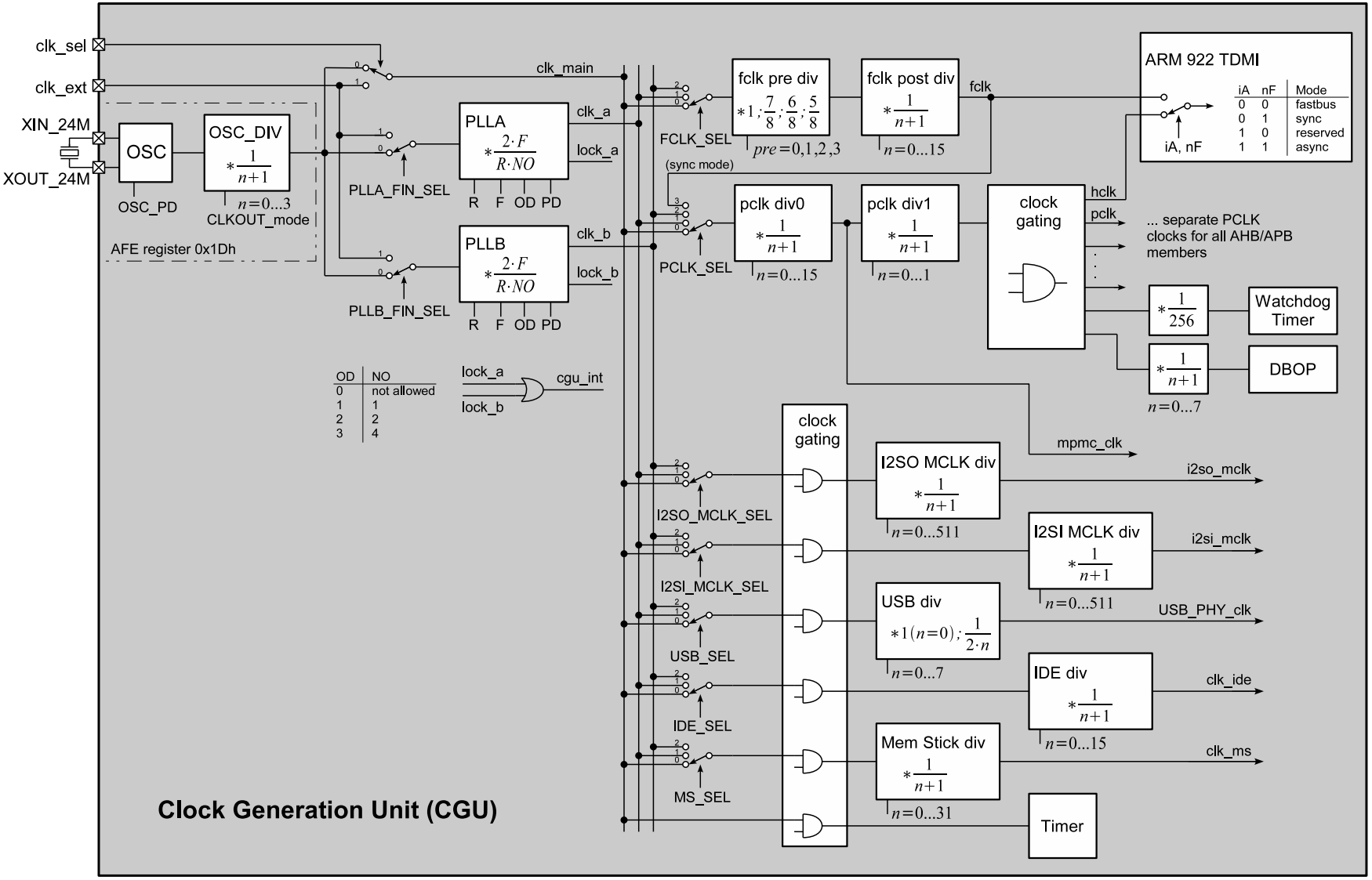
Asynchronous mode

Within asynchronous mode, the ARM922T frequency must be higher than the AHB bus frequency, but it can be completely asynchronous. Disatvantage is a slightly reduced performance of the system because of the higher effort for synchronisation between the ARM922T and AHB clock domains.

Block Diagram

The block diagram on the following page gives a detailed view of the structure of the CGU.

Figure 39 Clock generation unit block diagram



5.3.14.1 Input clock selection

Input clock is either coming directly from the clk_ext pin or from the internal 24MHz crystal oscillator. Usage of external pin or internal oscillator is selected by the dedicated pin clk_sel.

Table 72 Clock Selection

Clk_sel	Description
0	clk_main = clk_int
1	clk_main = clk_ext

Three main internal clocks are generated as source for all clock dividers for all modules.

- clk_a, clk_b: the outputs of two independently configurable PLLs.
- clk_main: this clock is always available without the need of configuring any internal PLL

An important constraint of the system is the memory type of the RAM: the internal 1-TRAM needs refresh cycles, with the following important restrictions:

- the free running AHB/APB clock (PCLK) for the 1-TRAM must always be present: also for changing frequency settings, this must be taken into account (e.g. switch from clk_main to PLL output only after PLL is settled (start-up time).
- the minimum frequency for the free running AHB/APB clock of the 1-TRAM is 20 MHz.

Important note: Switching between the different frequencies must be done in a pre-defined order using the CGU-driver software.

5.3.14.2 Clock Gating

For all peripheral clock domains clock gating is possible. Clock gating can be enabled/disabled by the corresponding bits within the clock control register CGU_PERI. After start-up, only the modules, which are necessary for booting the device, are enabled. These enabled peripherals are

- 1-TRAM controller and 1-TRAM macros
- external memory interface MPMC
- internal ROM
- vectored IR controller (VIC)

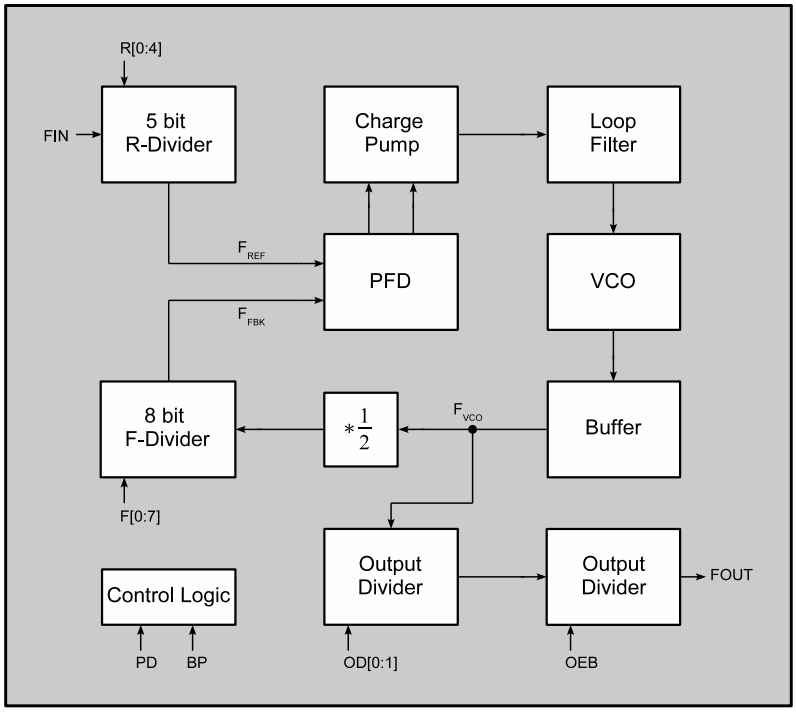
5.3.14.3 Interrupt generation

An interrupt can be generated after the PLL is locked.

5.3.14.4 PLL description

- runs on single power supply at 1.2 V (special power PADs are used within the chip layout to guarantee lowest jitter: vddapll, vssapll which are connected to vdd_core, vss_core within the BGA substrate)
- fully integrated with internal loop filter
- VCO operating frequency from 200 - 400 MHz
- phase comparator input frequency from 2 - 8 MHz
- low power dissipation of typical 2.5 mW

Figure 40 PLL block diagram



Programming and calculation of the PLL output frequency

The output frequency is controlled by three programmable dividers within the PLL. These dividers are: the input divider NR, the feedback divider NF and the output divider NO. The divider settings are programmed by bits within CGU_PLLA, CGU_PLLB registers. The table on the following page gives the detailed formulas for setting the PLL output frequency.

Table 73 Setting the PLL output frequency

Input divider NR.

NR = 16 * R4 + 8*R3 + 4*R2 + 2*R1 + R0

Feedback divider NF:

NF = 2 * (128*F7 + 64*F6 + 32*F5 + 16*F4 + 8*F3 + 4*F2 + 2*F1 + F0

Output divider NO:

Output divider setting	NO (output divider value)
OD0=0, OD1=0	Not allowed
OD0=1, OD1=0	1
OD0=0, OD1=1	2
OD0=1, OD1=1	4

The PLL output frequency is calculated with following formula

Output frequency $$f_{out} = \frac{NF}{NR \cdot NO} \cdot f_{in}$$

Comparison frequency $$f_{ref} = \frac{f_{in}}{NR}$$

VCO frequency $$f_{vco} = \frac{NF}{NR} \cdot f_{in}$$

Following constraints must be followed for the comparison and output frequency:

$$2\,MHz \leq fref \leq 8\,MHz$$

$$200\,MHz \leq fVCO \leq 400\,MHz$$

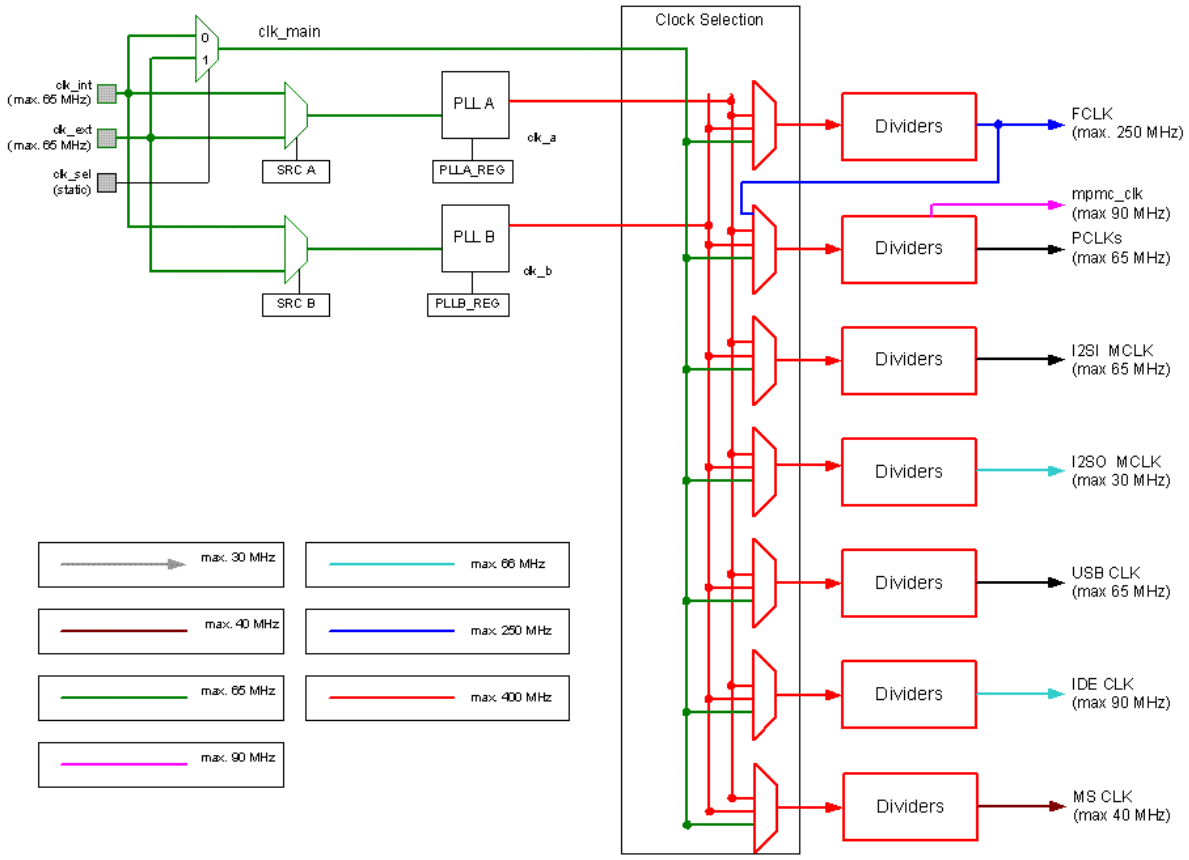
Clock Constraining

Different clocks are constraint to different maximum clock speeds. As the clock frequencies can be set by software, care must be taken not to exceed these maximum clock frequencies.

Table 749 Clock Constraining

Clock Domain	Max. Freq. [MHz]	Description
FCLK	250	Processor Clock
PCLK	65	AHB/APB bus clock
MPMC_CLK	90	MPMC (external memory interface) clock
I2SI MCLK	65	I2S input interface master clock
I2SO MCLK	30	I2S output interface master clock
USB CLK	48	USB interface clock
IDE CLK	90	IDE interface clock
MS CLK	40	Memory Stick Interface clock

Figure 41 Clock Generation Unit Block Diagram



5.3.14.5 Clock Generation Unit Registers

Table 20 CGU Registers

Register Name	Base Address	Offset	Note
CGU_PLLA	AS3525_CGU_BASE	0x00	PLLA configuration register
CGU_PLLB	AS3525_CGU_BASE	0x04	PLLB configuration register
CGU_PLLASUP	AS3525_CGU_BASE	0x08	PLLA supervisor register
CGU_PLLBSUP	AS3525_CGU_BASE	0x0C	PLLB supervisor register
CGU_PROC	AS3525_CGU_BASE	0x10	processor clock control register
CGU_PERI	AS3525_CGU_BASE	0x14	peripheral clock control register
CGU_AUDIO	AS3525_CGU_BASE	0x18	audio clock control register
CGU_USB	AS3525_CGU_BASE	0x1C	USB clock control register
CGU_INTCTRL	AS3525_CGU_BASE	0x20	CGU interrupt mask and enable register
CGU_IRQ	AS3525_CGU_BASE	0x24	interrupt clear and lock status register
CGU_COUNTA	AS3525_CGU_BASE	0x28	PLLA lock counter register
CGU_COUNTB	AS3525_CGU_BASE	0x2C	PLLB lock counter register
CGU_IDE	AS3525_CGU_BASE	0x30	IDE clock control register
CGU_MS	AS3525_CGU_BASE	0x34	Memory Stick clock control register
CGU_DBOP	AS3525_CGU_BASE	0x38	DBOP clock controller register

Table 21 CGU_PLLA Register

Name		Base		Default
CGU_PLLA		AS3525_CGU_BASE		0x00
Offset0x00		PLLA Configuration Register		
		The CGU_PLLA register is used to configure the PLL A		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
14:13	PLLA_OD [1:0]	0x00	R/W	PLLA output divider control, 2 bit
12:8	PLLA_R [4:0]	0x00	R/W	PLLA input divider control, 5-bit
7:0	PLLA_F [7:0]	0x00	R/W	PLLA feedback divider control, 8 bit

Table 22 CGU_PLLB Register

Name		Base		Default
CGU_PLLB		AS3525_CGU_BASE		0x00
Offset0x04		PLLB Configuration Register		
		The CGU_PLLB register is used to configure the PLL B		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
14:13	PLLB_OD [1:0]	0x00	R/W	PLLB output divider control, 2 bit
12:8	PLLB_R [4:0]	0x00	R/W	PLLB input divider control, 5-bit
7:0	PLLB_F [7:0]	0x00	R/W	PLLB feedback divider control, 8 bit

Table 23 PLLA Supervisor Register

Name		Base		Default
CGU_PLLASUP		**AS3525_CGU_BASE**		**0x08**
Offset0x08		**PLLA Supervisor Register**		
		This register contains control bits of the PLLA which are used very rarely, but have major impact on the functionality of the system.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
3	PLLA_PD	0x00	R/W	PLLA power down if SET
2	PLLA_OEB	0x00	R/W	PLLA output enable, active low
1	PLLA_BP	0x00	R/W	PLLA bypass if SET
0	PLLA_FIN_SEL	0x00	R/W	PLLA clock source select 0: clk_int [PAD] 1: clk_ext [PAD]

Table 24 PLBB Supervisor Register

Name		Base		Default
CGU_PLLBSUP		**AS3525_CGU_BASE**		**0x08**
Offset0x0c		**PLLB Supervisor Register**		
		This register contains control bits of the PLLB which are used very rarely, but have major impact on the functionality of the system.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
3	PLLB_PD	0x00	R/W	PLLB power down if SET
2	PLLB_OEB	0x00	R/W	PLLB output enable, active low
1	PLLB_BP	0x00	R/W	PLLB bypass if SET
0	PLLB_FIN_SEL	0x00	R/W	PLLB clock source select 0: clk_int [PAD] 1: clk_ext [PAD]

Table 25 Processor Clock Controller Register

Name		Base		Default
CGU_PROC		**AS3525_CGU_BASE**		**0x00**
Offset0x10		**Processor Clock Controller Register**		
		This register contains control bits for ARM processor clock generation => FCLK.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7:4	FCLK_POSTDIV_SEL [3:0]	0x00	R/W	post divider division ratio => post_div = 1/(fclk_postdiv_sel + 1)
3:2	FCLK_PREDIV_SEL [1:0]	0x00	R/W	pre divider (fractional) division ratio 00: pre_div = 1/1 01: pre_div = 7/8 10: pre_div = 6/8 11: pre_div = 5/8
1:0	FCLK_SEL[1:0]	0x00	R/W	clkin select 00: clk_main 01: plla_fout 10: pllb_fout 11: reserved (clk_main)

NOTE: f(fclk) := f(clkin) * pred_div * post_div;

Table 26 Peripheral Clock Controller Register

Name		Base		Default
CGU_PERI		AS3525_CGU_BASE		0x0F800000
Offset0x14		Peripheral clock controller register		
		This register allows setting the peripheral clocks.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
28	MBIST_EN	0	R/W	memory bist manager clock enable
27	EXTMEM_EN	1	R/W	external memory clock enable
26	EXTMEMIF_EN	1	R/W	external memory AHB IF clock enable
25	1TRAM_EN	1	R/W	1TRAM controller AHB IF clock enable
24	ROM_EN	1	R/W	ROM AHB IF clock enable
23	VIC_EN	1	R/W	vectored interrupt controller AHB IF clock enable
22	DMAC_EN	0	R/W	DMA controller AHB IF clock enable
21	USB_EN	0	R/W	USB controller AHB IF clock enable
20	I2SO_APB_EN	0	R/W	I2Sout APB IF clock enable
19	I2SI_APB_EN	0	R/W	I2Sin APB IF clock enable
18	I2C_EN	0	R/W	I2C master/slave APB IF clock enable
17	I2C_AUDIO_EN	0	R/W	I2C audio APB IF clock enable
16	GPIO_EN	0	R/W	general purpose IO APB IF clock enable
15	SDMCI_EN	0	R/W	secure digital/multimedia APB IF clock enable
14	NANDFLASH_EN	0	R/W	NAND flash/Smart Media APB IF clock enable
13	UART_EN	0	R/W	UART APB IF clock enable
12	WDOCNT_EN	0	R/W	watchdog counter clock enable
11	WDOIF_EN	0	R/W	watchdog timer module APB IF clock enable
10	SSP_EN	0	R/W	synchronous serial port APB IF clock enable
9	TIMER1_EN	0	R/W	timer module timer1 clock enable
8	TIMER2_EN	0	R/W	timer module timer2 clock enable
7	TIMERIF_EN	0	R/W	timer module APB IF clock enable
6	PCLK_DIV1_SEL	0	R/W	division ratio div1 (AHB/APB clock) => div1 = 1/(pclk_div1_sel + 1)
5:2	PCLK_DIV0_SEL [3:0]	0x0	R/W	division ratio div0 (ext. memory clock) => div0 = 1/(pclk_div0_sel + 1)
1:0	PCLK_SEL[1:0]	0x0	R/W	clkin select b'00: clk_main b'01: plla_fout b'10: pllb_fout b'11: fclk

CAUTION: Clock gating takes effect immediately! Software must assure that all transactions to/from the module are finished before the clock is disabled.

CAUTION: The peripheral clock must not exceed 65 MHz. The software must assure that requirement.
Note:
f(clk_extmem) := f(clkin) * div0;
f(pclk) := f(clkin) * div0 * div1;

Table 27 Audio Clock Controller Register

Name		Base		Default
CGU_AUDIO		**AS3525_CGU_BASE**		**0x00**
Offset0x18		**Audio Clock Controller Register**		
		This register allows setting the audio clock to I2S input and output interface.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
24	I2SI_MCLK2PAD_EN	0	R/W	I2S audio input clock (I2SI_MCLK) to PAD connection enable
23	I2SI_MCLK_EN	0	R/W	I2S audio input clock (I2SI_MCLK) enable
22:14	I2SI_MCLK_DIV_SEL [8:0]	0x0	R/W	I2Sin audio IF clock division ratio => div_i = 1/(i2si_mclk_div_sel + 1)
13:12	ISI_MCLK_SEL[1:0]	0x0	R/W	I2SI_MCLK clkin select 00: clk_main 01: plla_fout 10: pllb_fout 11: reserved (clk_main)
11	I2SO_MCLK_EN	0	R/W	I2S audio output clock (I2SO_MCLK) enable
10:2	I2SO_MCLK_DIV_SEL [8:0]	0x0	R/W	I2Sout audio IF clock division ratio => div_o = 1/(i2so_mclk_div_sel + 1)
1:0	ISO_MCLK_SEL[1:0]	0x0	R/W	I2SO_MCLK clkin select 00: clk_main 01: plla_fout 10: pllb_fout 11: reserved (clk_main)

Note:
The clock gating bits in this register apply only to the audio clocks. To enable/disable the APB parts of the corresponding I2S IF CGU_PERI has to be configured.

f(i2si_mclk) := f(I2SI_mclk clkin) * div_i;
f(i2so_mclk) := f(I2SO_mclk clkin) * div_o;

Table 28 Processor USB Clock Controller Register

Name		Base		Default
CGU_USB		**AS3525_CGU_BASE**		**0x00**
Offset0x1c		**USB Clock ControllerRegister**		
		This register allows setting the USB PHY interface clock.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
5	USB_CLK_EN	0x00	R/W	USB PHY clock enable => clk_usb
4:2	USB_DIV_SEL [2:0]	0x00	R/W	division ratio 0: div = 1/1 > 0: div = 1/(2*n); (even division factors only)
1:0	USB_SEL[1:0]	0x00	R/W	clkin select 00: clk_main 01: plla_fout 10: pllb_fout 11: reserved (clk_main)

Note:
The clock gating bit applies only to the USB PHY clock. To enable/disable the clock to the AHB part (USB CORE) CGU_PERI has to be configured.

f(clk_usb) = f(clk_core_48m) = f(clkin) * div;

Table 29 Interrupt Mask and PLL Lock Status Register

Name		Base		Default
CGU_INTCTRL		**AS3525_CGU_BASE**		**0x00**
Offset: 0x20		**Interrupt Mask and PLL Lock Status Register**		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
3	INT_EN_PLLB_LOCK	0x00	R/W	interrupt on PLLB lock enable (R/W)
2	INT_EN_PLLA_LOCK	0x00	R/W	interrupt on PLLA lock enable (R/W)
1	PLLB_LOCK	0x00	R	PLLB lock status, locked if SET (not cleared on read)
0	PLLA_LOCK		R	PLLA lock status, locked if SET (not cleared on read)

Table 30 Interrupt Clear Register

Name		Base		Default
CGU_IRQ		**AS3525_CGU_BASE**		**0x00**
Offset: 0x24		**Interrupt Clear Register**		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
1	PLLB_LOCK	0x00	R	PLLB lock status, locked if SET (not cleared on read)
0	PLLA_LOCK	0x00	R	PLLA lock status, locked if SET (not cleared on read)

Table 31 PLL A Lock Counter Register

Name		Base		Default
CGU_COUNTA		**AS3525_CGU_BASE**		**0x20**
Offset: 0x28		**PLL A Lock Counter Register**		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7:0	COUNTA[7:0]	0x00	R/W	number of PLL A's fout-clock cycles until the LOCKA bit is set

Table 32 PLL B Lock Counter Register

Name		Base		Default
CGU_COUNTB		**AS3525_CGU_BASE**		**0x20**
Offset: 0x2c		**PLL B Lock Counter Register**		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7:0	COUNTB[7:0]	0x00	R/W	number of PLL B's fout-clock cycles until the LOCKB bit is set

Table 33 IDE Clock Controller Register

Name		Base		Default
CGU_IDE		**AS3525_CGU_BASE**		**0x20**
Offset: 0x30		**IDE Clock Controller Register**		
		This register allows setting the IDE interface clocks.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
7	IDEIF_CLK_EN	0	R/W	IDE AHB IF clock enable
6	IDE_CLK_EN	0	R/W	IDE IF clock enable (90MHz domain) => clk_ide
5:2	IDE_DIV_SEL [2:0]	0x0	R/W	division ratio => div = 1/(ide_div_sel + 1)
1:0	IDE_SEL[1:0]	0x0	R/W	clkin select (clk_ide) 00: clk_main 01: plla_fout 10: pllb_fout 11: reserved (clk_main)

Note: f(clk_ide) := f(clkin) * div;

Table 34 Memory Stick (MS) Clock Controller Register

Name		Base		Default
CGU_MS		**AS3525_CGU_BASE**		**0x00**
Offset: 0x34		**MS Clock Controller Register**		
		This register allows setting the MS interface clocks.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
8	MSIF_CLK_EN	0	R/W	MS APB IF clock enable
7	MS_CLK_EN	0	R/W	MS IF clock enable (20/40MHz domain) => clk_ms
6:2	MS_DIV_SEL [2:0]	0x0	R/W	division ratio => div = 1/(ms_div_sel + 1)
1:0	MS_SEL[1:0]	0x0	R/W	clkin select (clk_ms) 00: clk_main 01: plla_fout 10: pllb_fout 11: reserved (clk_main)

Note: f(clk_ms) = f(clkin) * div;

Table 35 Data Block Output Port (DBOP) Clock Controller Register

Name		Base		Default
CGU_DBOP		**AS3525_CGU_BASE**		**0x00**
Offset: 0x38		**DBOP Clock Controller Register**		
		This register allows setting the DBOP interface clocks.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
3	DBOP_EN	0	R/W	DBOP APB IF clock enable
2:0	DBOP_PREDIV_SEL [2:0]	0x0	R/W	division ratio => div = 1/(dbop_prediv_sel + 1)

Note: Setting DBOP_EN will enable both clocks (push/APB and pop) immediately.
clk_dbop clock (pop clock) generation uses DBOP APB IF clock as input clock.

f(clk_dbop) = f(PCLKDBOP) * div;

Figure 42 Table with verified CGU frequency settings for Audio and USB applications with 24MHz crystal

nf	nr	no	Fref [MHz]	fvco [MHz]	plla_fout [MHz]	fclk_ pre	fclk_ post	fclk [MHz]	pclk_ div0	pclk_ div1	pclk [MHz]	mclk_ div	mclk [Hz]	fsaudio [Hz]	faudio error [%]	usb_ div	fusbphy [Hz]	fusb error [%]	fsaudio target [Hz]	CPU clock mode
Target: **48.000 Hz**																				
48	**6**	**1**	4,000	384,000	384,000	**0,00**	**5,00**	64,000	**0**	**0**	64,000	**61**	6.193.548	48.387	0,806	**3**	48.000.000	0,000	48000	Fastbus
24	**3**	**3**	8,000	384,000	96,000	**0,00**	**1,00**	48,000	**0**	**0**	48,000	**15**	6.000.000	46.875	-2,344	**1**	48.000.000	0,000	48000	Fastbus
41	**8**	**2**	3,000	246,000	123,000	**0,00**	**1,00**	61,500	**0**	**0**	61,500	**19**	6.150.000	48.047	0,098				48000	fastbus
23	**5**	**3**	4,800	220,800	55,200	**0,00**	**0,00**	55,200	**0**	**0**	55,200	**8**	6.133.333	47.917	-0,174				48000	fastbus
Target: 44.100 Hz																				
48	**6**	**1**	4,000	384,000	384,000	**0**	**5**	64,000	**0**	**0**	64,000	**67**	5.647.059	44.118	0,040	**3**	48.000.000	0,000	44100	fastbus
24	**3**	**3**	8,000	384,000	96,000	**0**	**1**	48,000	**0**	**0**	48,000	**16**	5.647.059	44.118	0,040	**1**	48.000.000	0,000	44100	fastbus
47	**10**	**2**	2,400	225,600	112,800	**0**	**1**	56,400	**0**	**0**	56,400	**19**	5.640.000	44.063	-0,085				44100	fastbus
79	**12**	**3**	2,000	316,000	79,000	**0**	**1**	39,500	**0**	**0**	39,500	**13**	5.642.857	44.085	-0,034				44100	fastbus
47	**10**	**3**	2,400	225,600	56,400	**0**	**1**	28,200	**0**	**0**	28,200	**9**	5.640.000	44.063	-0,085				44100	fastbus
Target: 32.000 Hz																				
48	**6**	**1**	4,000	384,000	384,000	**0**	**5**	64,000	**0**	**0**	64,000	**93**	4.085.106	31.915	-0,266	**3**	48.000.000	0,000	32000	fastbus
24	**3**	**3**	8,000	384,000	96,000	**0**	**1**	48,000	**0**	**0**	48,000	**22**	4.173.913	32.609	1,902	**1**	48.000.000	0,000	32000	fastbus
41	**6**	**3**	4,000	328,000	82,000	**0**	**1**	41,000	**0**	**0**	41,000	**19**	4.100.000	32.031	0,098				32000	fastbus
31	**7**	**3**	3,429	212,571	53,143	**0**	**1**	26,571	**0**	**0**	26,571	**12**	4.087.912	31.937	-0,197				32000	fastbus
Target: 24.000 Hz																				
48	**6**	**1**	4,000	384,000	384,000	**0**	**5**	64,000	**0**	**0**	64,000	**124**	3.072.000	24.000	0,000	**3**	48.000.000	0,000	24000	fastbus
24	**3**	**3**	8,000	384,000	96,000	**0**	**1**	48,000	**0**	**0**	48,000	**30**	3.096.774	24.194	0,806	**1**	48.000.000	0,000	24000	fastbus
41	**8**	**2**	3,000	246,000	123,000	**0**	**1**	61,500	**0**	**0**	61,500	**39**	3.075.000	24.023	0,098				24000	fastbus
Target: 22.050 Hz																				
48	**6**	**1**	4,000	384,000	384,000	**0**	**5**	64,000	**0**	**0**	64,000	**135**	2.823.529	22.059	0,040	**3**	48.000.000	0,000	22050	fastbus
24	**3**	**3**	8,000	384,000	96,000	**0**	**1**	48,000	**0**	**0**	48,000	**33**	2.823.529	22.059	0,040	**1**	48.000.000	0,000	22050	fastbus
47	**10**	**2**	2,400	225,600	112,800	**0**	**1**	56,400	**0**	**0**	56,400	**39**	2.820.000	22.031	-0,085				22050	fastbus
Target: 16.000 Hz																				
48	**6**	**1**	4,000	384,000	384,000	**0**	**5**	64,000	**0**	**0**	64,000	**187**	2.042.553	15.957	-0,266	**3**	48.000.000	0,000	16000	fastbus
24	**3**	**3**	8,000	384,000	96,000	**0**	**1**	48,000	**0**	**0**	48,000	**46**	2.042.553	15.957	-0,266	**1**	48.000.000	0,000	16000	fastbus
41	**6**	**3**	4,000	328,000	82,000	**0**	**1**	41,000	**0**	**0**	41,000	**39**	2.050.000	16.016	0,098				16000	fastbus
Target: 12.000 Hz																				
48	**6**	**1**	4,000	384,000	384,000	**0**	**5**	64,000	**0**	**0**	64,000	**249**	1.536.000	12.000	0,000	**3**	48.000.000	0,000	12000	fastbus
24	**3**	**3**	8,000	384,000	96,000	**0**	**1**	48,000	**0**	**0**	48,000	**62**	1.523.810	11.905	-0,794	**1**	48.000.000	0,000	12000	fastbus
41	**6**	**3**	4,000	328,000	82,000	**0**	**1**	41,000	**0**	**0**	41,000	**52**	1.547.170	12.087	0,727				12000	fastbus
Target: 11.025 Hz																				
48	**6**	**1**	4,000	384,000	384,000	**0**	**5**	64,000	**0**	**0**	64,000	**271**	1.411.765	11.029	0,040	**3**	48.000.000	0,000	11025	fastbus
24	**3**	**3**	8,000	384,000	96,000	**0**	**1**	48,000	**0**	**0**	48,000	**67**	1.411.765	11.029	0,040	**1**	48.000.000	0,000	11025	fastbus
41	**6**	**3**	4,000	328,000	82,000	**0**	**1**	41,000	**0**	**0**	41,000	**57**	1.413.793	11.045	0,184				11025	fastbus
Target: 8.000 Hz																				
48	**6**	**1**	4,000	384,000	384,000	**0**	**5**	64,000	**0**	**0**	64,000	**374**	1.024.000	8.000	0,000	**3**	48.000.000	0,000	8000	fastbus
24	**3**	**3**	8,000	384,000	96,000	**0**	**1**	48,000	**0**	**0**	48,000	**93**	1.021.277	7.979	-0,266	**1**	48.000.000	0,000	8000	fastbus
41	**6**	**3**	4,000	328,000	82,000	**0**	**1**	41,000	**0**	**0**	41,000	**79**	1.025.000	8.008	0,098				8000	fastbus

5.3.15 CCU - Chip Control Unit

Following chapters describe the functions of the CCU.

Table 75 CCU Registers

Register Name	Base Address	Offset	Note
CCU_SRC	AS3525_CCU_BASE	0x0000	Software reset control register
CCU_SRL	AS3525_CCU_BASE	0x0004	Software reset lock register
CCU_MEMMAP	AS3525_CCU_BASE	0x0008	Memory map register
CCU_IO	AS3525_CCU_BASE	0x000C	IO configuration register
CCU_SCON	AS3525_CCU_BASE	0x0010	System configuration register
CCU_VERS	AS3525_CCU_BASE	0x0014	Chip version register
CCU_SPARE1	AS3525_CCU_BASE	0x0018	spare register 1 (for future use)
CCU_SPARE2	AS3525_CCU_BASE	0x001C	spare register 2 (for future use)

5.3.15.1 Reset Controller

- Generation of the internal reset: the external reset pin XRES is used to generate the internal global reset. This internal reset is synchronised to clk_main and the active reset time is enlarged. This is necessary to wait for the startup of the DC/DC converter and LDO's that are generating the supplies of the digital chip. The time assumed for this startup is 10 ms, therefore 2^18 cycles of clk_main are counted before the internal reset is released. This mechanism is also used for the WATCHDOG reset.
- Softreset: for each module, the reset can also be generated by SW control. For this purpose, the SW can write to the software reset control register (CCU_SRC). To avoid unintended SW resets, the access to this control register is locked by the SW reset lock register (CCU_SRL). So the correct usage is:
  - write CCU_SRC
  - write CCU_SRL (magic number 0x1A720212) to CCU_LOCK to activate resets
  - write CCU_SRL (0x00000000) to deactivate resets

Table 76 Software Reset Control Register

Name		Base		Default
CCU_SRC		AS3525_CCU_BASE		0x00
Offset: 0x0000h		Software Reset Control Register		
		Writing a logic 1 to the single bits in the read/write register enables resets to each module.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
24	DBOP_EN	0	R/W	1: enable DBOP reset 0: disable DBOP reset
23	MBIST_EN	0	R/W	1: enable MBIST manager reset 0: disable MBIST manager reset
22	SPDIF_EN	0	R/W	1: enable SPDIF reset 0: disable SPDIF reset
21	TIMER_EN	0	R/W	1: enable timer module reset 0: disable timer module reset
20	SSP_EN	0	R/W	1: enable synchronous serial port reset 0: disable synchronous serial port reset
19	WDO_EN	0	R/W	1: enable watchdog timer module reset 0: disable watchdog timer module reset
18	IDE_EN	0	R/W	1: enable compact flash/IDE reset (except AHB part) 0: disable compact flash/IDE reset (except AHB part)
17	IDE_AHB_EN	0	R/W	1: enable compact flash/IDE's AHB interface reset 0: disable compact flash/IDE's AHB interface reset

Name		Base		Default
CCU_SRC		**AS3525_CCU_BASE**		**0x00**
Offset: 0x0000h		**Software Reset Control Register**		
		Writing a logic 1 to the single bits in the read/write register enables resets to each module.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
16	UART_EN	0	R/W	1: enable UART interface reset 0: disable UART interface reset
15	NAF_EN	0	R/W	1: enable NAND flash/Smart Media interface reset 0: disable NAND flash/Smart Media interface reset
14	SDMCI_EN	0	R/W	1: enable secure digital/multimedia interface reset 0: disable secure digital/multimedia interface reset
13	GPIO_EN	0	R/W	1: enable general purpose IO reset 0: disable general purpose IO reset
12	I2C_AUDIO_EN	0	R/W	1: enable audio I2C interface reset 0: disable audio I2C interface reset
11	I2C_EN	0	R/W	1: enable master/slave I2C interface reset 0: disable master/slave I2C interface reset
10	MMS_EN	0	R/W	1: enable memory stick interface reset 0: disable memory stick interface reset
9	I2SI_APB_EN	0	R/W	1: enable I2S input interface reset for APB part 0: disable I2S input interface reset for APB part
8	I2SO_APB_EN	0	R/W	1: enable I2S output interface reset for APB part 0: disable I2S output interface reset for APB part
7	USB_AHB_EN	0	R/W	1: enable USB AHB reset 0: disable USB AHB reset
6	USB_PHY_EN	0	R/W	1: enable USB PHY reset 0: disable USB PHY reset
5	DMAC_EN	0	R/W	1: enable DMA controller reset 0: disable DMA controller reset
4	VIC_EN	0	R/W	1: enable vectored interrupt cell reset 0: disable vectored interrupt cell reset
3	RAMC_EN	0	R/W	1: enable RAMC reset 0: disable RAMC reset
2	1TRAM_EN	0	R/W	1: enable 1TRAM reset 0: disable 1TRAM reset
1	MPMC_EN	0	R/W	1: enable external memory AHB reset 0: disable external memory AHB reset
0	BRIDGE_EN	0	R/W	1: enable bridge reset 0: disable bridge reset

Table 77 Software Reset Lock Register

Name		Base		Default
CCU_SRL		**AS3525_CCU_BASE**		**0x00**
Offset: 0x0004h		**Software Reset Lock Register**		
		Use of this register enables the software reset selected with Software Reset Control Register. Writing a value of 0x1A720212 will enable the selected reset; writing any other value will not enable software reset.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
0:31	software_reset_lock	0	R/W	0x1A720212: enables selected reset Other values: no effect

5.3.15.2 IO_PADRING functions

Within the IO_PADRING module all multiplexing for selecting alternative functions is implemented. The selection of active functions is chosen within the IO_configuration_register. Following table gives a description of the IO configurations:

Table 78 IO_PADRING Configurations

Name		Base		Default
CCU_IO		**AS3525_CCU_BASE**		**0x00**
Offset: 0x000Ch		**IO Configuration Registers**		
		With this read/write registers the functionality of IOs are controlled which provides several different functions		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
8:7	naf_ce_sel[1:0]	0	R/W	these bits select which output is used for NAF ce_n. 0: naf_ce0_n 1: naf_ce1_n 2: naf_ce2_n 3: naf_ce3_n
6	pll_probe_en	0	R/W	test mode: 1: pll output clock is available at a GPIO Pin
5	ide_sel	0	R/W	1: the IDE input/output configuration is set
4	spi_flash_mode	0	R/W	SPI used in master mode: 1: pin SSP_FSSOUT always 0 0: pin SSP_FSSOUT generated by SSP hardware block SPI used in slave mode: spi_flash_mode hast to be switched to 0
3:2	xpd_func_sel(1:0)	0	R/W	00: XPD works as general purpose IO 01: SD-MCI interface 10: the XPD[5:0] are configured to support MS, XPD[7:6} are general IO pins 11: reserved (XPD works as general IO)
1	i2c_ms_sel	0	R/W	1: the I2C master/slave IO configuration is set
0	uart_sel	0	R/W	1: the uart IO configuration is set

5.3.15.3 Other CCU functions

With the CCU_MEMMAP register, the remap(r/w) and int_boot_sel (read only) bits are accessible.

Table 79 Memory Map Register

Name		Base		Default
CCU_MEMMAP		**AS3525_CCU_BASE**		**N/A**
Offset: 0x0008h		**Memory Map Register**		
		With the register the remap(r/w) and int_boot_sel (r only) bits are accessible.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
1	INT_BOOT_SEL	external pin XPC[0]	R	Boot selection 1: internal ROM 0: external memory interface
0	REMAP	0	R/W	Defines memory mapping 1: RAM 0: ROM

If the INIT_BOOT_SEL is 0 (boot from external memory interface), following pins will be latched at startup to define the MPMC interface settings:

• mpmc_stcs1mw[0]	• mpmc_stcs1pb
• mpmc_stcs1pol	• mpmc_rel1config

5.3.15.4 Additional Chip Control Unit Registers

Table 80 System Configuration Register

Name		Base		Default
CCU_SCON		AS3525_CCU_BASE		0x00
Offset: 0x0010h		System Configuration Register		
		This read/write register controls system parameters.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
0	priority_config	0	R/W	AHB master's priority configuration: 0: Configuration A (default) Highest priority: TIC (Test Interface Controller) – for production test only 2nd highest priority: ARM922T 3rd highest priority: DMA 4th highest priority: USB lowest priority: IDE 1: Configuration B Highest priority: TIC (Test Interface Controller) – for production test only 2nd highest priority: DMA 3rd highest priority: USB 4th highest priority: IDE lowest priority: ARM922T

Table 81 Chip Version Register

Name		Base		Default
CCU_VERS		AS3525_CCU_BASE		0x09
Offset: 0x0014h		Chip Version Register		
		Version information can be read from this register.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
31:12	main_version_id(19:0)	0x2	R	main version ID
11:0	sub_version_id(11:0)	0x1	R	sub version ID

Table 82 Spare Register 1

Name		Base		Default
CCU_SPARE1		AS3525_CCU_BASE		0x00
Offset: 0x0018h		Metal ECO Spare Register		
		This register implements 32bit spare FF's. Use for metal ECO redesign.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
31:9	spare	0x00	R/W	spare bits to be used for metal ECO redesign if SET
8	dma_sreq_SSPRX_off	0x00	R/W	disableDMA single request of SSPRX module if SET
7	dma_sreq_SSPTX_off	0x00	R/W	disable DMA single request of SSPTX module if SET
6	dma_sreq_DBOP_off	0x00	R/W	disable DMA single request of DBOP module if SET
5	dma_sreq_I2Sin_off	0x00	R/W	disable DMA single request of I2Sin module if SET
4	dma_sreq_I2Sout_off	0x00	R/W	disable DMA single request of I2Sout module if SET
3:2	spare	0x00	R/W	spare bits to be used for metal ECO redesign if SET
1:0	mpmc_clk_inv	0x00	R/W	spare bits used to invert output clocks mpmc_clk(1:0) if SET

Table 83 Spare Register 2

Name		Base		Default
CCU_SPARE2		**AS3525_CCU_BASE**		**0x00**
Offset: 0x001Ch		**Metal ECO Spare Register**		
		This register implements 32bit spare FF's. Use for metal ECO redesign.		
Bit	**Bit Name**	**Default**	**Access**	**Bit Description**
31:3	spare	0x00	R/W	spare bits to be used for metal ECO redesign if SET
2:0	bist_idle_cycle_ctrl	0x00	R/W	internal RAM refresh cycle control bits of BIST_MGR module 000: idle every 32nd cycle (default) 100: idle every 16th cycle 110: idle every 8th cycle 111: idle every 4th cycle

6 Pinout and Packaging

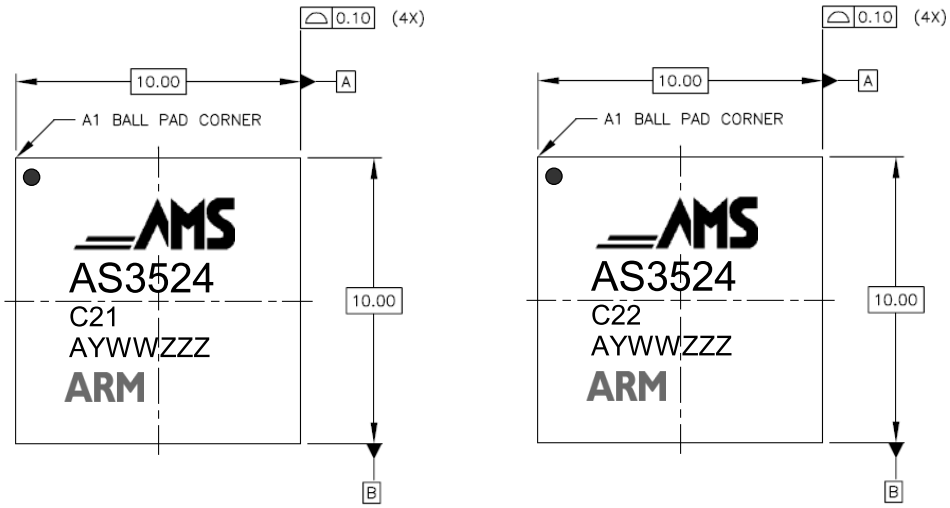
6.1 Package Variants

CTBGA (Thin ChipArray BGA) package technology is used for multi-chip-module (MCM) packaging.

6.2 CTBGA180 Package Drawings

6.2.1 Marking

Figure 43 CTBGA180 TOP View and Package Marking

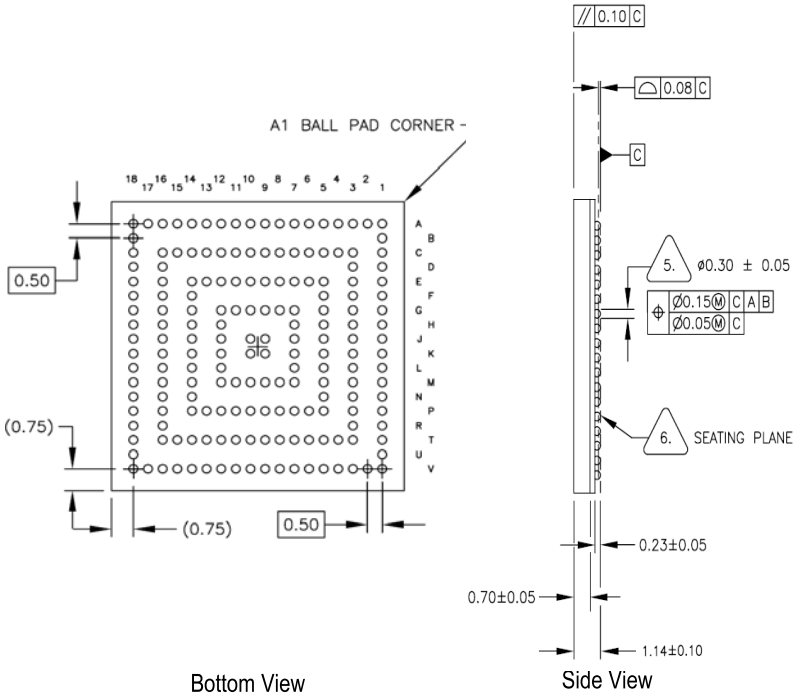


Top View (for both versions C21 and C22)

Package Code AYYWWZZZ

A	Y	WW	ZZZ
A … for PB free	Year	working week assembly/packaging	Free choice

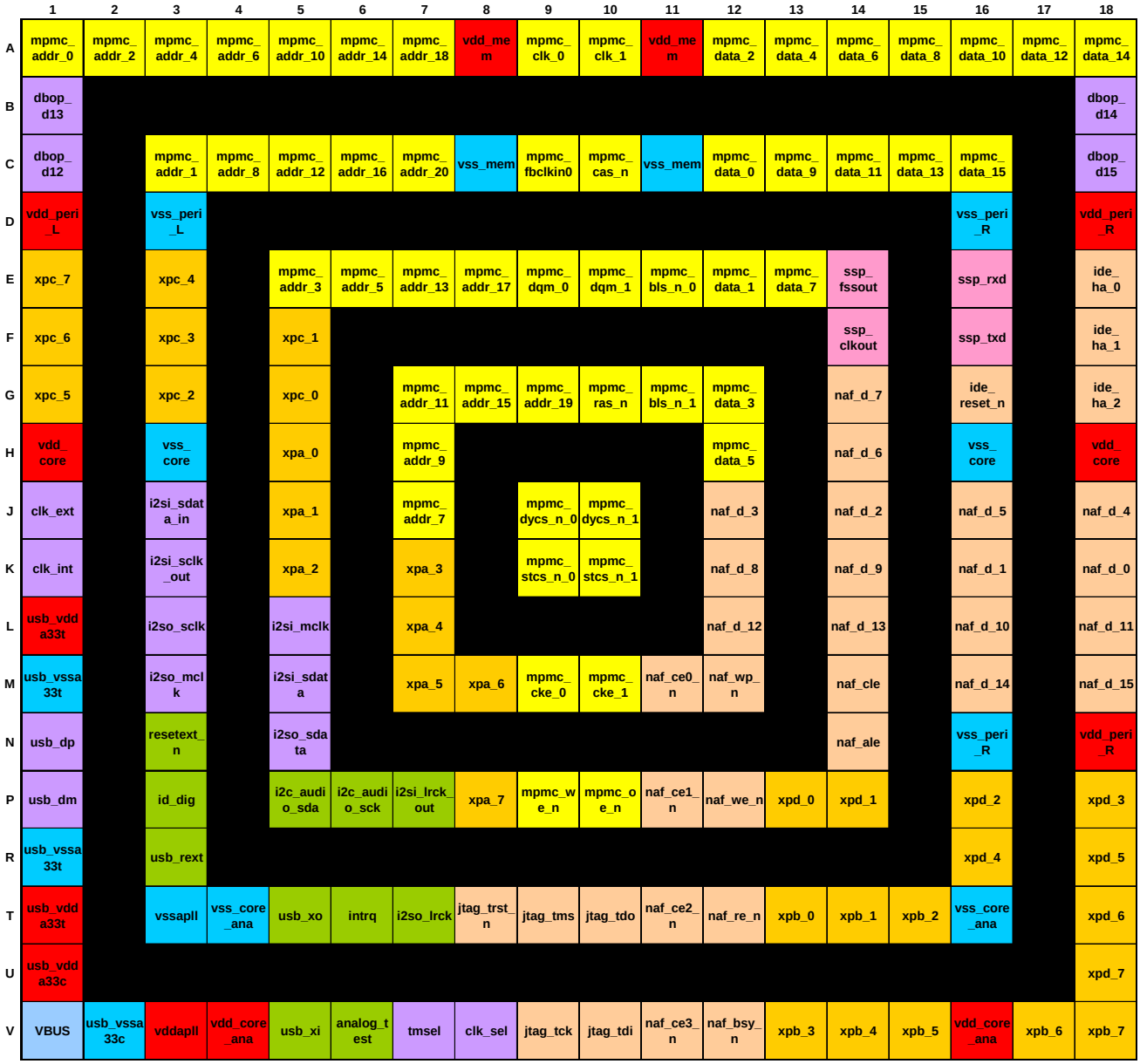
Figure 44 CTBGA224 Package Drawing Bottom/Side View



Bottom View Side View

6.2.2 CTBGA180 Package Ball-out

Figure 45 CTBGA180 Package Ball--out



6.2.3 CTBGA180 Ball List

Table 84 CTBGA180 Ball List

Ball Nr. BGA180	Ball Name	PAD Type	I/O	Ball Description
N3	resetext_n	D IN ST	I	reset input (active low)
K1	clk_int	D IN ST	I	clock input (10-26MHz)
J1	clk_ext	D IN ST PD	I	clock input (10-26MHz)
V8	clk_sel	D IN ST PD	I	clock select 0 (low): clock from clk_int is used for internal clk_main 1 (high): clock from pad clk_ext is used for internal clk_main
V7	tmsel	D IN ST PD	I	test mode select

Ball Nr. BGA180	Ball Name	PAD Type	I/O	Ball Description
				For testing purpose only, has to be set to “0”.
P3	id_dig	D IN ST (PU)	I	USB mini receptacle identifier Has to be connected to USB jack ID pin.
	Port A			
H5	xpa[0]	D IO ST PD LSR	IO	GPIO IO, Port A
J5	xpa[1]	D IO ST PD LSR	IO	GPIO IO, Port A
K5	xpa[2]	D IO ST PD LSR	IO	GPIO IO, Port A
K7	xpa[3]	D IO ST PD LSR	IO	GPIO IO, Port A
L7	xpa[4]	D IO ST PD LSR	IO	GPIO IO, Port A
M7	xpa[5]	D IO ST PD LSR	IO	GPIO IO, Port A
M8	xpa[6]	D IO ST PD LSR	IO	GPIO IO, Port A
P8	xpa[7]	D IO ST PD LSR	IO	GPIO IO, Port A
	Port B / DISPLAY / UART			
T13	xpb[0]	D IO ST PD LSR	IO	GPIO IO, Port B
	mpmc_stcs1mw[0]*		I	static memory chip memory width setting for boot loader 0: 8 bit data bus 1: 16 bit data bus The value is latched at reset.
	dbop_c0		O	DISPLAY control output
T14	xpb[1]	D IO ST PD LSR	IO	GPIO IO, Port B
	mpmc_stcs1pol*		I	static memory chip select polarity setting for boot loader 0: active LOW chip select 1: active high chip select The value is latched at reset.
	dbop_c1		O	DISPLAY control output
T15	xpb[2]	D IO ST PD LSR	IO	GPIO IO, Port B
	mpmc_stcs1pb*		I	static memory byte lane polarity setting for boot loader 0: HIGH for reads, LOW for writes, used for we_n access 1: LOW for reads, LOW for writes, used for upper and lower byte access The value is latched at reset.
	dbop_c2		O	DISPLAY control output
V13	xpb[3]	D IO ST PD LSR	IO	GPIO IO, Port B
	mpmc_rel1config*		I	test mode configuration (for testing purpose only !!!) The value is latched at reset.
	dbop_c3		O	DISPLAY control output
V14	xpb[4]	D IO ST PD LSR	IO	GPIO IO, Port B
	dbop_d[8]		IO	DISPLAY data input/output (high byte)
V15	xpb[5]	D IO ST PD LSR	IO	GPIO IO, Port B
	dbop_d[9]		IO	DISPLAY data input/output (high byte)
V17	xpb[6]	D IO ST PU LSR	IO	GPIO IO, Port B
	uart_rxd		I	UART receive line
	dbop_d[10]		IO	DISPLAY data input/output (high byte)
V18	xpb[7]	D IO ST PU LSR	IO	GPIO IO, Port B
	uart_txd		O	UART transmit line
	dbop_d[11]		IO	DISPLAY data input/output (high byte)
	Port C / DISPLAY / 2-WIRE SERIAL			
G5	xpc[0]	D IO ST PD LSR	IO	GPIO IO, Port C

Ball Nr. BGA180	Ball Name	PAD Type	I/O	Ball Description
	IntBootSel*		I	BOOT LOADER source select input 1: internal ROM 0: external ROM/Flash
	dbop_d[0]		IO	DISPLAY data input/output (low byte)
F5	xpc[1]	D IO ST PD LSR	IO	GPIO IO, Port C
	boot_sel[0]		I	BOOT LOADER type select input
	dbop_d[1]		IO	DISPLAY data input/output (low byte)
G3	xpc[2]	D IO ST PD LSR	IO	GPIO IO, Port C
	boot_sel[1]		I	BOOT LOADER type select input
	dbop_d[2]		IO	DISPLAY data input/output (low byte)
F3	xpc[3]	D IO ST PD LSR	IO	GPIO IO, Port C
	boot_sel[2]		I	BOOT LOADER type select input
	dbop_d[3]		IO	DISPLAY data input/output (low byte)
E3	xpc[4]	D IO ST PD LSR	IO	GPIO IO, Port C
	dbop_d[4]		IO	DISPLAY data input/output (low byte)
G1	xpc[5]	D IO ST PD LSR	IO	GPIO IO, Port C
	dbop_d[5]		IO	DISPLAY data input/output (low byte)
F1	xpc[6]	D IO ST PU LSR	IO	GPIO IO, Port C
	cmd_ms_sck		IO	2-WIRE SERIAL master/slave clock line
	dbop_d[6]		IO	DISPLAY data input/output (low byte)
E1	xpc[7]	D IO ST PU LSR	IO	GPIO IO, Port C
	cmd_ms_sda		IO	2-WIRE SERIAL master/slave data line
	dbop_d[7]		IO	DISPLAY data input/output (low byte)
	Port D / SD Card / Memory Stick			
P13	xpd[0]	D IO ST LSR	IO	GPIO IO, Port D
	mci_dat[0]		IO	MMC/SD data line
	ms_sdio[0]		IO	MEMORY STICK data line
P14	xpd[1]	D IO ST LSR	IO	GPIO IO, Port D
	mci_dat[1]		IO	MMC/SD data line
	ms_sdio[1]		IO	MEMORY STICK data line
P16	xpd[2]	D IO ST LSR	IO	GPIO IO, Port D
	mci_dat[2]		IO	MMC/SD data line
	ms_sdio[2]		IO	MEMORY STICK data line
P18	xpd[3]	D IO ST LSR	IO	GPIO IO, Port D
	mci_dat[3]		IO	MMC/SD data line
	ms_sdio[3]		IO	MEMORY STICK data line
R16	xpd[4]	D IO ST LSR	IO	GPIO IO, Port D
	mci_cmd		O	MMC/SD command line
	ms_sclk		O	MEMORY STICK clock line
R18	xpd[5]	D IO ST LSR	IO	GPIO IO, Port D
	mci_clk		O	MMC/SD clock line
	ms_bs		O	MEMORY STICK bus state
T18	xpd[6]	D IO ST LSR	IO	GPIO IO, Port D
	mci_fbclk		I	MMC/SD feedback clock
	ms_fbclk		I	MEMORY STICK feedback clock
U18	xpd[7]	D IO ST LSR	IO	GPIO IO, Port D
	mci_rod		O	MMC/SD resistor open drain control
	2-wire serial Audio Master			

Ball Nr. BGA180	Ball Name	PAD Type	I/O	Ball Description
P6	i2c_audio_sck	D IO ST PU LSR	O	2-wire serial audio master clock line used for controlling the audio/PMU sub system
P5	i2c_audio_sda	D IO ST PU LSR	O	2-wire serial audio master data line used for controlling the audio/PMU sub system
	Serial Synchronous Port			
E14	ssp_fssout	D IO ST PU LSR	O	SSP master, frame or slave select
	ssp_fssin		I	SSP slave, frame select
F14	ssp_clkout	D IO ST PU LSR	O	SSP master, clock line
	ssp_clkin		I	SSP slave, clock line
E16	ssp_rxd	D IO ST PU LSR	I	SSP receive data input
F16	ssp_txd	D IO ST PU LSR	O	SSP transmit data output
	NandFlash / IDE			
K18	naf_d[0]	D IO ST PD LSR	IO	NAND FLASH data line (low byte)
	ide_hd[0]		IO	IDE data line (low byte)
K16	naf_d[1]	D IO ST PD LSR	IO	NAND FLASH data line (low byte)
	ide_hd[1]		IO	IDE data line (low byte)
J14	naf_d[2]	D IO ST PD LSR	IO	NAND FLASH data line (low byte)
	ide_hd[2]		IO	IDE data line (low byte)
J12	naf_d[3]	D IO ST PD LSR	IO	NAND FLASH data line (low byte)
	ide_hd[3]		IO	IDE data line (low byte)
J18	naf_d[4]	D IO ST PD LSR	IO	NAND FLASH data line (low byte)
	ide_hd[4]		IO	IDE data line (low byte)
J16	naf_d[5]	D IO ST PD LSR	IO	NAND FLASH data line (low byte)
	ide_hd[5]		IO	IDE data line (low byte)
H14	naf_d[6]	D IO ST PD LSR	IO	NAND FLASH data line (low byte)
	ide_hd[6]		IO	IDE data line (low byte)
G14	naf_d[7]	D IO ST PD LSR	IO	NAND FLASH data line (low byte)
	ide_hd[7]		IO	IDE data line (low byte)
K12	naf_d[8]	D IO ST PD LSR	IO	NAND FLASH data line (high byte)
	ide_hd[8]		IO	IDE data line (high byte)
K14	naf_d[9]	D IO ST PD LSR	IO	NAND FLASH data line (high byte)
	ide_hd[9]		IO	IDE data line (high byte)
L16	naf_d[10]	D IO ST PD LSR	IO	NAND FLASH data line (high byte)
	ide_hd[10]		IO	IDE data line (high byte)
L18	naf_d[11]	D IO ST PD LSR	IO	NAND FLASH data line (high byte)
	ide_hd[11]		IO	IDE data line (high byte)
L12	naf_d[12]	D IO ST PD LSR	IO	NAND FLASH data line (high byte)
	ide_hd[12]		IO	IDE data line (high byte)
L14	naf_d[13]	D IO ST PD LSR	IO	NAND FLASH data line (high byte)
	ide_hd[13]		IO	IDE data line (high byte)
M16	naf_d[14]	D IO ST PD LSR	IO	NAND FLASH data line (high byte)
	ide_hd[14]		IO	IDE data line (high byte)
M18	naf_d[15]	D IO ST PD LSR	IO	NAND FLASH data line (high byte)
	ide_hd[15]		IO	IDE data line (high byte)
M14	naf_cle	D IO ST LSR	O	NAND FLASH command latch enable
	ide_dmarq		I	IDE DMA request used for DMA data transfers between host and device
N14	naf_ale	D IO ST LSR	O	NAND FLASH address latch enable

Ball Nr. BGA180	Ball Name	PAD Type	I/O	Ball Description
	ide_iordy		I	IDE IO ready signal used by device to extend host data transfer cycles
M12	naf_wp_n	D IO ST PD LSR	O	NAND FLASH write protect not
	ide_intrq		I	IDE interrupt request used by device to interrupt the host controller
M11	naf_ce0_n	D IO ST LSR	O	NAND FLASH chip enable
	ide_cs0_n		O	IDE chip select 0 used by the host to select command block registers in the device
P11	naf_ce1_n	D IO ST LSR	O	NAND FLASH chip enable
	ide_cs1_n		O	IDE chip select 1 used by the host to select control block registers in the device
T11	naf_ce2_n	D IO ST LSR	O	NAND FLASH chip enable
	ide_hiown		O	IDE host IO write strobe
V11	naf_ce3_n	D IO ST LSR	O	NAND FLASH chip enable
	ide_hiorn		O	IDE host IO read strobe
P12	naf_we_n	D IO ST LSR	O	NAND FLASH write enable not
	ide_dackn		O	IDE DMA acknowledge used by the host to initiate DMA data transfers
T12	naf_re_n	D IO ST LSR	O	NAND FLASH read enable not
	ide_npcblid		I	IDE primary channel cable ID detect
V12	naf_bsy_n	D IO ST LSR	I	NAND FLASH ready / busy not
	ide_nscblid		I	IDE secondary channel cable ID select
E18	ide_ha[0]	D OUT LSR	O	IDE host address
F18	ide_ha[1]	D OUT LSR	O	IDE host address
G18	ide_ha[2]	D OUT LSR	O	IDE host address
G16	ide_reset_n	D OUT LSR	O	IDE reset not, used by the host to reset the device
	I2S Output			
N5	i2so_sdata	D OUT LSR	O	IS2 data output, data output from digital core to audio sub system
L3	i2so_sclk	D OUT LSR	O	I2S serial clock, clock output from digital core to audio sub system
T7	i2so_lrck	D OUT LSR	O	I2S left/right clock, clock output from digital core to audio sub system
M3	i2so_mclk	D OUT LSR	O	I2S master clock, clock output from digital core to audio sub system
	I2S Input			
M5	i2si_sdata	D IN ST	I	I2S data input, data output from audio sub system to digital core
K3	i2si_sclk_out	D IO ST LSR	O	I2S master serial clock serial clock output for external ADC if AS3525 is I2S master
	i2si_sclk_in		I	I2S slave serial clock serial clock input for external ADC if AS3525 is I2S slave
P7	i2si_lrck_out	D IO ST LSR	O	I2S master, left/right clock left/right clock output for external ADC if AS3525 is I2S master
	i2si_lrck_in		I	I2S slave, left/right clock left/right clock input for external ADC if AS3525 is I2S master
L5	i2si_mclk	D OUT LSR	O	I2S master, master clock
J3	i2si_sdata_in	D IN ST PD	I	I2S data input data input from external audio ADC

Ball Nr. BGA180	Ball Name	PAD Type	I/O	Ball Description
	spdif_data_in		I	SPDIF data input data input for SPDIF to I2S conversion
	Audio Subsystem IRQ			
T6	INTRQ	D IN ST	I	interrupt input
	JTAG Debugging IF			
T8	jtag_trst_n	D IN ST PD	I	JTAG reset not
T9	jtag_tms	D IN ST PU	I	JTAG mode select
V9	jtag_tck	D IN ST PU	I	JTAG clock
V10	jtag_tdi	D IN ST PU	I	JTAG data input
T10	jtag_tdo	D IO ST PU LSR	O	JTAG data output
	External Memory IF			
A1	mpmc_addr[0]	D OUT LSR LV	O	EXT. MEMORY address line
C3	mpmc_addr[1]	D OUT LSR LV	O	EXT. MEMORY address line
A2	mpmc_addr[2]	D OUT LSR LV	O	EXT. MEMORY address line
E5	mpmc_addr[3]	D OUT LSR LV	O	EXT. MEMORY address line
A3	mpmc_addr[4]	D OUT LSR LV	O	EXT. MEMORY address line
E6	mpmc_addr[5]	D OUT LSR LV	O	EXT. MEMORY address line
A4	mpmc_addr[6]	D OUT LSR LV	O	EXT. MEMORY address line
J7	mpmc_addr[7]	D OUT LSR LV	O	EXT. MEMORY address line
C4	mpmc_addr[8]	D OUT LSR LV	O	EXT. MEMORY address line
H7	mpmc_addr[9]	D OUT LSR LV	O	EXT. MEMORY address line
A5	mpmc_addr[10]	D OUT LSR LV	O	EXT. MEMORY address line
G7	mpmc_addr[11]	D OUT LSR LV	O	EXT. MEMORY address line
C5	mpmc_addr[12]	D OUT LSR LV	O	EXT. MEMORY address line
E7	mpmc_addr[13]	D OUT LSR LV	O	EXT. MEMORY address line
A6	mpmc_addr[14]	D OUT LSR LV	O	EXT. MEMORY address line
G8	mpmc_addr[15]	D OUT LSR LV	O	EXT. MEMORY address line
C6	mpmc_addr[16]	D OUT LSR LV	O	EXT. MEMORY address line
E8	mpmc_addr[17]	D OUT LSR LV	O	EXT. MEMORY address line
A7	mpmc_addr[18]	D OUT LSR LV	O	EXT. MEMORY address line
G9	mpmc_addr[19]	D OUT LSR LV	O	EXT. MEMORY address line
C7	mpmc_addr[20]	D OUT LSR LV	O	EXT. MEMORY address line
M9	mpmc_cke[0]	D OUT LSR LV	O	EXT. MEMORY clock enable0 used for SDRAM devices only
M10	mpmc_cke[1]	D OUT LSR LV	O	EXT. MEMORY clock enable 1 used for SDRAM devices only
A9	mpmc_clk[0]	D OUT LSR LV	O	EXT. MEMORY clock 0 used for SDRAM devices only
A10	mpmc_clk[1]	D OUT LSR LV	O	EXT. MEMORY clock 1 used for SDRAM devices only
C9	mpmc_fbclkin	D IO ST PD LSR LV	O	EXT. MEMORY feedback clock used for SDRAM devices only
E9	mpmc_dqm[0]	D OUT LSR LV	O	EXT. MEMORY data mask 0 used for SDRAM devices and static memories
E10	mpmc_dqm[1]	D OUT LSR LV	O	EXT. MEMORY data mask 1 used for SDRAM devices and static memories
C10	mpmc_cas_n	D OUT LSR LV	O	EXT. MEMORY column address strobe not used for SDRAM devices only
J9	mpmc_dycs_n[0]	D OUT LSR LV	O	EXT. MEMORY dynamic memory chip select 0 not used for SDRAM devices only

Ball Nr. BGA180	Ball Name	PAD Type	I/O	Ball Description
J10	mpmc_dycs_n[1]	D OUT LSR LV	O	EXT. MEMORY dynamic memory chip select 1 not used for SDRAM devices only
G10	mpmc_ras_n	D OUT LSR LV	O	EXT. MEMORY row address strobe not used for SDRAM devices only
P9	mpmc_we_n	D OUT LSR LV	O	EXT. MEMORY write enable not used for SDRAM devices and static memories
K9	mpmc_stcs_n[0]	D OUT LSR LV	O	EXT. MEMORY static memory chip select 0 not used for static memory devices only
K10	mpmc_stcs_n[1]	D OUT LSR LV	O	EXT. MEMORY static memory chip select 0 not used for static memory devices only
E11	mpmc_bls_n[0]	D OUT LSR LV	O	EXT. MEMORY byte lane select 0 not used for static memory devices only
G11	mpmc_bls_n[1]	D OUT LSR LV	O	EXT. MEMORY byte lane select 1 not used for static memory devices only
P10	mpmc_oe_n	D OUT LSR LV	O	EXT. MEMORY output enable not used for static memory devices only
C12	mpmc_data[0]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
E12	mpmc_data[1]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
A12	mpmc_data[2]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
G12	mpmc_data[3]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
A13	mpmc_data[4]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
H12	mpmc_data[5]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
A14	mpmc_data[6]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
E13	mpmc_data[7]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
A15	mpmc_data[8]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
C13	mpmc_data[9]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
A16	mpmc_data[10]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
C14	mpmc_data[11]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
A17	mpmc_data[12]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
C15	mpmc_data[13]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
A18	mpmc_data[14]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
C16	mpmc_data[15]	D IO ST PD LSR LV	IO	EXT. MEMORY data line
	DBOP			
C1	dbop_d[12]	D IO ST PD LSR	IO	DISPLAY data input/output (high byte)
B1	dbop_d[13]	D IO ST PD LSR	IO	DISPLAY data input/output (high byte)
B18	dbop_d[14]	D IO ST PD LSR	IO	DISPLAY data input/output (high byte)
C18	dbop_d[15]	D IO ST PD LSR	IO	DISPLAY data input/output (high byte)
	USB 2.0 OTG			
T1	usb_vdda33t	PWP_VD_RDO_3V	P	USB 3.3V analog power supply for OTG transceiver block
L1	usb_vdda33t	PWP_VD_RDO_3V	P	USB 3.3V analog power supply for OTG transceiver block
M1	usb_vssa33t	PWP_VS_RDO_3V	P	USB 3.3V analog ground supply for OTG transceiver block
R1	usb_vssa33t	PWP_VS_RDO_3V	P	USB 3.3V analog ground supply for OTG transceiver block
N1	usb_dp	USB_ESD_5VT	A	USB D+ signal from USB cable
P1	usb_dm	USB_ESD_5VT	A	USB D- signal from USB cable
V5	usb_xi	ANA_BI_DNR_3V	A	USB crystal oscillator xi pin used for using external crystal for USB clock generation for testing purpose only, can be tied to ground or left floating
T5	usb_xo	ANA_BI_DNR_3V	A	USB crystal oscillator xo pin used for using external crystal for USB clock generation for testing purpose only, can be tied to ground or left floating
R3	usb_rext	ANA_BI_RXT_3V	A	USB external resistor connect

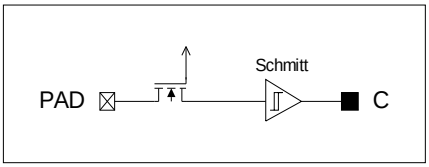
Ball Nr. BGA180	Ball Name	PAD Type	I/O	Ball Description
				analog signal to the external resistor for setting the bias current of the USB 2.0 OTG PHY, voltage level is 1.1-1.3V
U1	vdda33c	PWP_VD_ANA_3V	P	USB 3.3V analog power supply for common block
V2	usb_vssa33c	PWP_VS_RDO_3V	P	USB 3.3V analog ground supply for common block
V6	analog_test	ANA_BI_DNR_3V	A	USB analog test input/output for testing purpose only, has to be left open
V1	VBUS	USB20_VBUS_5VT_OTG	AIO	USB VBUS analog input
	Supply Balls			
D1	vdd_peri_l	P	P	3.3V peripheral power supply
D18	vdd_peri_r	P	P	3.3V peripheral power supply
N18	vdd_peri_r	P	P	3.3V peripheral power supply
A8	vdd_mem	P	P	3.3V/2.5V/1.8V external memory power supply
A11	vdd_mem	P	P	3.3V/2.5V/1.8V external memory power supply
D3	vss_peri_l	P	P	3.3V peripheral ground supply
D16	vss_peri_r	P	P	3.3V peripheral ground supply
N16	vss_peri_r	P	P	3.3V peripheral ground supply
C8	vss_mem	P	P	3.3V (2.5V) external memory ground supply
C11	vss_mem	P	P	3.3V (2.5V) external memory ground supply
H1	vdd_core	P	P	1.2V core power supply
H18	vdd_core	P	P	1.2V core power supply
V4	vdd_core_ana	P	P	1.2V core power supply (analog blocks)
V16	vdd_core_ana	P	P	1.2V core power supply (analog blocks)
V3	vddapll	P	P	1.2V PLL power supply
H3	vss_core	P	P	1.2V core ground supply
H16	vss_core	P	P	1.2V core ground supply
T4	vss_core_ana	P	P	1.2V core ground supply (analog blocks)
T16	vss_core_ana	P	P	1.2V core ground supply (analog blocks)
T3	vssapll	P	P	1.2V PLL ground suppl

6.3 Pad Cell Description

6.3.1 Digital Pads

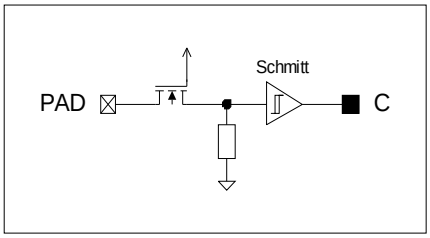
6.3.1.1 D IN ST

Figure 46 Digital Input with Schmitt Trigger



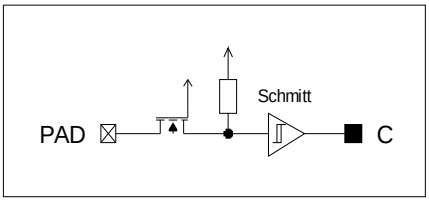
6.3.1.2 D IN PD ST

Figure 47 Digital Input with Schmitt Trigger and Pull-Down



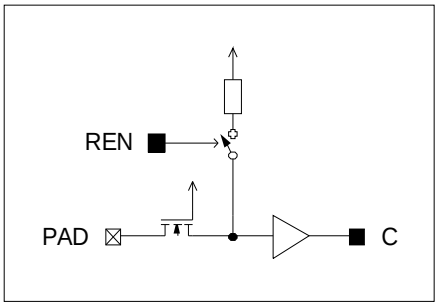
6.3.1.3 D IN PU ST

Figure 48 Digital Input with Schmitt Trigger and Pull-Up



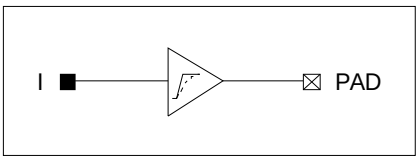
6.3.1.4 D IN (PU)

Figure 49 Digital Input with enable controlled Pull-Up



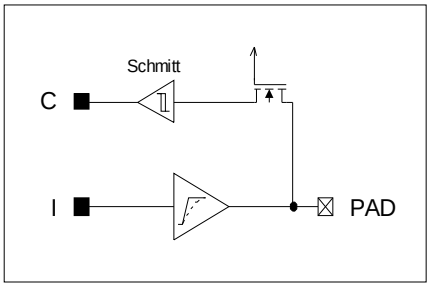
6.3.1.5 D OUT LSR

Figure 50 Digital Output with Limited Slew Rate



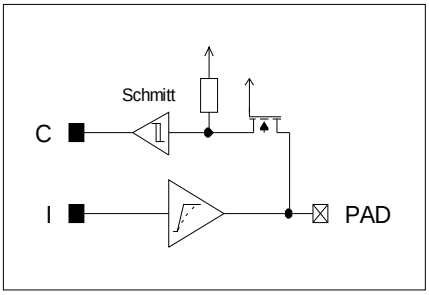
6.3.1.6 D IO ST LSR

Figure 51 Digital Schmitt Trigger Input and Limited Slew Rate Output



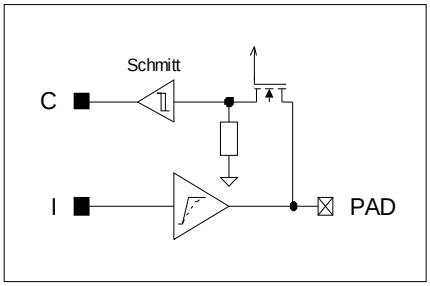
6.3.1.7 D IO ST PU LSR

Figure 52 Digital Schmitt Trigger Input with Pull-Up and Limited Slew Rate Output



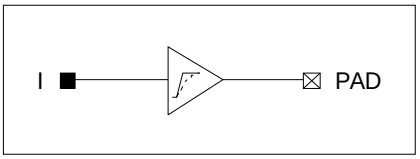
6.3.1.8 D IO ST PD LSR

Figure 53 Digital Schmitt Trigger Input with Pull-Down and Limited Slew Rate Output



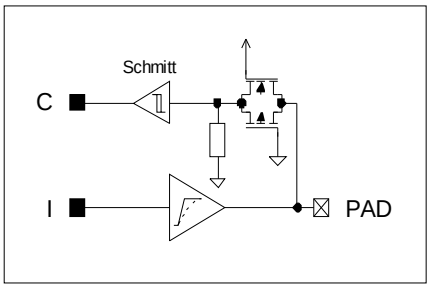
6.3.1.9 D OUT LSR LV

Figure 54 Digital Output with Limited Slew Rate (low voltage)



6.3.1.10 D IO ST PD LSR LV

Figure 55 Digital Schmitt Trigger Input with Pull-Down and Limited Slew Rate Output (low voltage)



7 Appendix

7.1 Memory MAP

ARM922T provides 32-bit address to access the peripherals and memory. With this 32-bit address ARM922T can access up to 4 Giga Bytes of memory. Cocoa does not use the complete 4 GB address space.

Address 0x0000_0000 is mapped to internal ROM or External Memory interface based on the boot ROM selection by the external input pin (Port C, xpc[0] = intBootSel) Pin intBootSel=1 at startup selects the internal ROM, intBootSel = 0 selects the external memory.

The address range starting at 0x0000_0000 is also mapped to internal RAM upon setting of the remap bit. This remap allows the user to select either RAM or ROM at 0x0000_0000.

Table 85 Address Map

S.No.	Start (Base) Address	End Address	Actual Block Size	Peripheral	Comment
				AHB Blocks	
	0x0000_0000	0x0001_FFFF	128 KByte	Internal ROM	Remap = 0 and IntBootSel = 1
	0x0000_0000	0x003F_FFFF	4 MB	External Memory IF (MPMC Bank1 – Ext Flash or Ext ROM)	Remap = 0 and IntBootSel = 0
	0x0000_0000	0x0004_FFFF	320 KByte	Embedded 1T-RAM	Remap = 1
	0x0100_0000	0x0FFF_FFFF		Reserved	
	0x1000_0000	0x103F_FFFF	4 MB	External Memory IF	Aliased

S.No.	Start (Base) Address	End Address	Actual Block Size	Peripheral	Comment
				(MPMC Bank1 – Ext Flash or Ext ROM)	
	0x1100_0000	0x1FFF_FFFF		Reserved	
	0x2000_0000	0x203F_FFFF	4 MB	External Memory IF (MPMC Bank2 – External LCD Controller)	
	0x2100_0000	0x2FFF_FFFF		Reserved	
	0x3000_0000	0x3FFF_FFFF	256 MB	External Memory IF (MPMC Bank 4 – SDRAM)	
	0x4000_0000	0x4FFF_FFFF	256 MB	External Memory If (MPMC Bank5 – SDRAM)	
	0x5000_0000	0x7FFF_FFFF		Reserved	
	0x8000_0000	0x8001_FFFF	128 KByte	Internal ROM	Aliased
	0x8002_0000	0x80FF_FFFF		Reserved	
	0x8100_0000	0x8104_FFFF	320 KByte	Embedded 1T-RAM	Aliased
	0x8105_0000	0xBFFF_FFFF		Reserved	
	0xC000_0000	0xC001_FFFF	128 KByte	Internal ROM	Aliased
	0xC002_0000	0xC0FF_FFFF		Reserved	
	0xC100_0000	0xC104_FFFF	320 KByte	Embedded 1T-RAM	Aliased
	0xC105_0000	0xC5FF_FFFF		Reserved	
	0xC600_0000	0xC600_FFFF	Few	USB2.0 Slave	
	0xC601_0000	0xC601_FFFF	Few	VIC	
	0xC602_0000	0xC602_FFFF	Few	DMAC Slave	
	0xC603_0000	0xC603_FFFF	Few	ExtMemIFSlave	
	0xC604_0000	0xC604_FFFF	Few	MemoryStick Slave	
	0xC605_0000	0xC605_FFFF	Few	CompactFlash/IDE Slave	
	0xC606_0000	0xC606_FFFF	4 KByte	ARM922T Slave	
	0xC607_0000	0xC7FF_FFFF		Reserved	
				APB blocks	
	0xC800_0000	0xC800_FFFF	Few	Nand Flash / Smart Media Interface	
	0xC801_0000	0xC801_FFFF	Few	BistManager	
	0xC802_0000	0xC802_FFFF	Few	SD-MCI	
	0xC803_0000	0xC803_FFFF	Few	Reserved	
	0xC804_0000	0xC804_FFFF	Few	Timer	
	0xC805_0000	0xC805_FFFF	Few	Watchdog Timer	
	0xC806_0000	0xC806_FFFF	Few	I2C Master/Slave	
	0xC807_0000	0xC807_FFFF	Few	I2C Audio Master	
	0xC808_0000	0xC808_FFFF	Few	SSP	
	0xC809_0000	0xC809_FFFF	Few	I2S IN Interface	
	0xC80A_0000	0xC80A_FFFF	Few	I2S OUT Interface	
	0xC80B_0000	0xC80B_FFFF	Few	GPIO A	
	0xC80C_0000	0xC80C_FFFF	Few	GPIO B	
	0xC80D_0000	0xC80D_FFFF	Few	GPIO C	
	0xC80E_0000	0xC80E_FFFF	Few	GPIO D	
	0xC80F_0000	0xC80F_FFFF	Few	Clock Generation Unit	
	0xC810_0000	0xC810_FFFF	Few	Chip Control Unit	
	0xC811_0000	0xC811_FFFF	Few	Debug UART	
	0xC812_0000	0xC812_FFFF		DBOP	
	0xC813_0000	0xC813_FFFF		reserved	

7.2 Register definitions

This section gives a short overview of all module registers.

7.2.1 Base Address definitions

Each module register block starts at a specific base address.

Table 86 Base Addresses

REGISTER Name	Register Address
AS3525_RAM_BASE	0x00000000
AS3525_USB_BASE	0xC6000000
AS3525_VIC_BASE	0xC6010000
AS3525_DMAC_BASE	0xC6020000
AS3525_EXTMEM_ITF_BASE	0xC6030000
AS3525_MEMSTICK_BASE	0xC6040000
AS3525_CF_IDE_BASE	0xC6050000
AS3525_NAND_FLASH_BASE	0xC8000000
AS3525_BIST_MANAGER_BASE	0xC8010000
AS3525_SD_MCI_BASE	0xC8020000
AS3525_TIMER_BASE	0xC8040000
AS3525_WDT_BASE	0xC8050000
AS3525_I2C_MS_BASE	0xC8060000
AS3525_I2C_AUDIO_BASE	0xC8070000
AS3525_SSP_BASE	0xC8080000
AS3525_I2SIN_BASE	0xC8090000
AS3525_I2SOUT_BASE	0xC80A0000
AS3525_GPIO1_BASE	0xC80B0000
AS3525_GPIO2_BASE	0xC80C0000
AS3525_GPIO3_BASE	0xC80D0000
AS3525_GPIO4_BASE	0xC80E0000
AS3525_CGU_BASE	0xC80F0000
AS3525_CCU_BASE	0xC8100000
AS3525_UART_BASE	0xC8110000
AS3525_DBOP_BASE	0xC8120000

8 Ordering Information

Table 87 ordering information

Device ID	Number	Package Type	Delivery Form	Description
AS3524P[-Z] V D	AS3524A-Z C21 TRA	CTBGA 180	Tray	Pb-free
	AS3524A-Z C21 T&R		Tape and Reel	
	AS3524A-Z C22 TRA		Tray	
	AS3524A-Z C22 T&A		Tape and Reel	

Where

V = **Version**

C21: Version with initial Bootloader Version 1

C22: Version with ROM mask update and changed Bootloader Version 2

see chapter

P = **Package Type**:

A: CTBGA 180, Thin ChipArray Ball Grid Array, 10x10mm package size, 0.5mm ball pitch

D = **Delivery Form**:

TRA = Tray

T&R = Tape and Reel

Z = **Pb-free Status:**

Z = Pb-free/ RoHS package type

9 Copyright

10 Disclaimer

Devices sold by austriamicrosystems AG are covered by the warranty and patent identification provisions appearing in its Term of Sale. austriamicrosystems AG makes no warranty, express, statutory, implied, or by description regarding the information set forth herein or regarding the freedom of the described devices from patent infringement. austriamicrosystems AG reserves the right to change specifications and prices at any time and without notice. Therefore, prior to designing this product into a system, it is necessary to check with austriamicrosystems AG for current information. This product is intended for use in normal commercial applications. Applications requiring extended temperature range, unusual environmental requirements, or high reliability applications, such as military, medical life-support or life-sustaining equipment are specifically not recommended without additional processing by austriamicrosystems AG for each application.

The information furnished here by austriamicrosystems AG is believed to be correct and accurate. However, austriamicrosystems AG shall not be liable to recipient or any third party for any damages, including but not limited to personal injury, property damage, loss of profits, loss of use, interruption of business or indirect, special, incidental or consequential damages, of any kind, in connection with or arising out of the furnishing, performance or use of the technical data herein. No obligation or liability to recipient or any third party shall arise or flow out of austriamicrosystems AG rendering of technical or other services.

11 Contact Information

austria*micro*systems AG
Business Unit Communications
A 8141 Schloss Premstätten, Austria
T. +43 (0) 3136 500 0
F. +43 (0) 3136 5692
info@austriamicrosystems.com

For Sales Offices, Distributors and Representatives, please visit:

http://www.austriamicrosystems.com