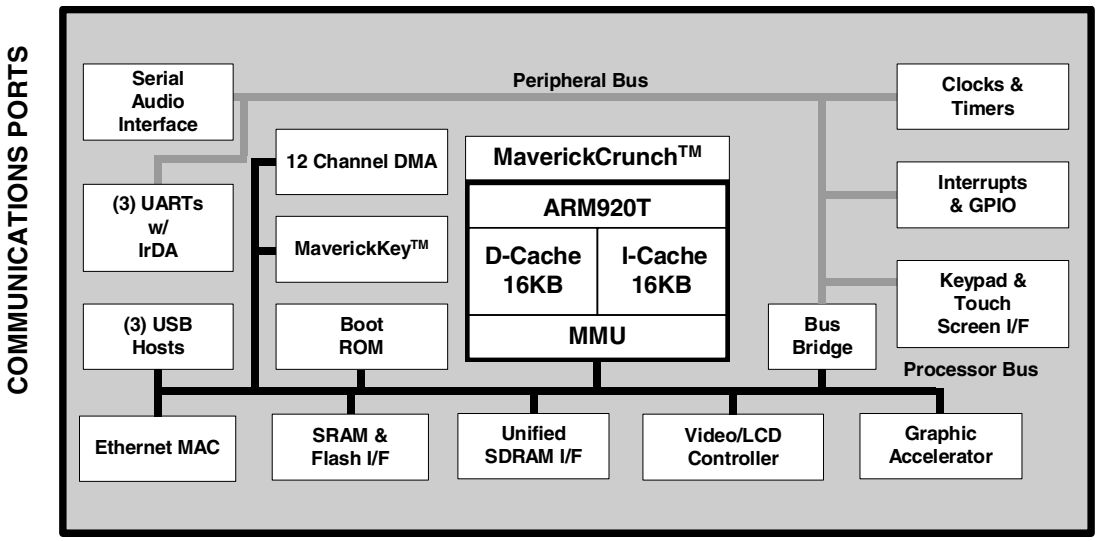
FEATURES

ARM9 SOC with Ethernet, USB, Display and Touchscreen

- 200 MHz ARM920T Processor
  - 16 Kbyte Instruction Cache
  - 16 Kbyte Data Cache
  - Linux®, Microsoft® Windows® CE enabled MMU
  - 100 MHz System Bus
- MaverickCrunch™ Math Engine
  - Floating point, integer and signal processing instructions
  - Optimized for digital music compression and decompression algorithms
  - Hardware interlocks allow in-line coding
- MaverickKey™ IDs
  - 32-bit unique ID can be used for DRM compliance 128-bit random ID
- Integrated Peripheral Interfaces
  - 32-bit SDRAM Interface up to 4 banks
  - 32/16-bit SRAM/FLASH/ROM
  - Serial EEPROM Interface
  - 1/10/100 Mbps Ethernet MAC
  - Three UARTs
  - Three-port USB 2.0 Full Speed Host (OHCI) (12 Mbits per second)
  - IrDA Interface
  - LCD and Raster Interface with Graphics Accelerator
  - Touchscreen Interface with ADC
  - 8 x 8 Keypad Scanner
  - One Serial Peripheral Interface (SPI) Port
  - 6-channel or 2-channel Serial Audio Interface (I^2S)
  - 2-channel low-cost Serial Audio Interface (AC'97)
- Internal Peripherals
  - 12 Direct Memory Access (DMA) Channels
  - Real-time Clock with software Trim
  - Dual PLL controls all clock domains
  - Watchdog Timer
  - Two general purpose 16-bit timers
  - One general purpose 32-bit timer
  - One 40-bit Debug Timer
  - Interrupt Controller
  - Boot ROM
- Package
  - 272 pin TFBGA







© Copyright 2004 Cirrus Logic (All Rights Reserved)

OVERVIEW

The EP9307 is an ARM920T-based system-on-a-chip (SOC) design with a large peripheral set targeted to a variety of applications:

- Thin client computers for business and home
- Internet radio
- Internet access devices
- Industrial computers
- Specialized terminals
- Point of sale terminals
- Test and measurement equipment

The EP9307 is one of a series of ARM920T-based devices.

The ARM920T microprocessor core with separate 16 Kbyte, 64-way set-associative instruction and data caches is augmented by the MaverickCrunch™ co-processor enabling high-speed floating point calculations.

MaverickKey™ unique hardware programmed IDs are a solution to the growing concern over secure web content and commerce. With Internet security playing an important role in the delivery of digital media such as books or music, traditional software methods are quickly becoming unreliable. The MaverickKey unique IDs provide OEMs with a method of utilizing specific hardware IDs such as those assigned for SDMI (Secure Digital Music Initiative) or any other authentication mechanism.

A high-performance 1/10/100 Mbps Ethernet Media Access Controller (MAC) is included along with external interfaces to SPI, I^2S audio, Raster/LCD, keypad and touchscreen. A three-port USB 2.0 Full Speed Host (OHCI) (12 Mbits per second) and three UARTs are included as well.

The EP9307 is a high-performance, low-power RISC-based single-chip computer built around an ARM920T microprocessor core with a maximum operating clock rate of 200 MHz (184 MHz for industrial conditions). The ARM core operates from a 1.8 V supply, while the I/O operates at 3.3 V with power usage between 100 mW and 750 mW (dependent on speed).

Table A. Change History

Revision	Date	Changes
1	July 2004	Initial Release.
2	August 2004	Correct error in pin out table, pages 42 & 43.

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Table of Contents

FEATURES 1
OVERVIEW 2
Processor Core - ARM920T 6
MaverickCrunch™ Math Engine 6
MaverickKey™ Unique ID 6
General Purpose Memory Interface (SDRAM, SRAM, ROM, FLASH) 6
Ethernet Media Access Controller (MAC) 7
Serial Interfaces (SPI, I2S and AC '97) 7
Raster/LCD Interface 7
Graphics Accelerator 8
Touch Screen Interface with 12-bit Analog-to-Digital Converter (ADC) 8
64-Keypad Interface 8
Universal Asynchronous Receiver/Transmitters (UARTs) 9
Internal Boot ROM 9
Triple Port USB Host 9
Two-Wire Interface With EEPROM Support 9
Real-Time Clock with Software Trim 10
PLL and Clocking 10
Timers 10
Interrupt Controller 10
Dual LED Drivers 10
General Purpose Input/Output (GPIO) 10
Reset and Power Management 11
Hardware Debug Interface 11
12-Channel DMA Controller 11
Electrical Specifications 12
Absolute Maximum Ratings 12
Recommended Operating Conditions 12
DC Characteristics 13
Timings 14
Memory Interface 15
Ethernet MAC Interface 29
Audio Interface 31
AC'97 35
LCD Interface 36
ADC 37
JTAG 38
272 Pin TFBGA Package Outline 39
272 TFBGA Diagram 39
272 Pin TFBGA Pinout (Bottom View) 40
Acronyms and Abbreviations 47
Units of Measurement 47
ORDERING INFORMATION 48

© Copyright 2004 Cirrus Logic (All Rights Reserved)

List of Figures

Figure 1. Timing Diagram Drawing Key 14
Figure 2. SDRAM Load Mode Register Cycle Timing Measurement 15
Figure 3. SDRAM Burst Read Cycle Timing Measurement 16
Figure 4. SDRAM Burst Write Cycle Timing Measurement 17
Figure 5. SDRAM Auto Refresh Cycle Timing Measurement 18
Figure 6. Static Memory Single Word Read Cycle Timing Measurement 19
Figure 7. Static Memory Single Word Write Cycle Timing Measurement 20
Figure 8. Static Memory Multiple Word Read 8 Bit Cycle Timing Measurement 21
Figure 9. Static Memory Multiple Word Write 8 bit Cycle Timing Measurement 22
Figure 10. Static Memory Multiple Word Read 16 Bit Cycle Timing Measurement 23
Figure 11. Static Memory Multiple Word Write 16 bit Cycle Timing Measurement 24
Figure 12. Static Memory Burst Read Cycle Timing Measurement 25
Figure 13. Static Memory Single Read Wait Cycle Timing Measurement 26
Figure 14. Static Memory Single Write Wait Cycle Timing Measurement 27
Figure 15. Static Memory Turnaround Cycle Timing Measurement 28
Figure 16. Ethernet MAC Timing Measurement 30
Figure 17. SPI Single Transfer Timing Measurement 32
Figure 18. Microwire Frame Format, Single Transfer 32
Figure 19. SPI Format with SPH=1 Timing Measurement 33
Figure 20. Inter-IC Sound (I2S) Timing Measurement 34
Figure 21. AC ‘97 Configuration Timing Measurement 35
Figure 22. LCD Timing Measurement 36
Figure 23. ADC Transfer Function 37
Figure 24. JTAG Timing Measurement 38
Figure 25. 272 Pin TFBGA Diagram 39
Figure 26. 272 Pin TFBGA Pinout 41

 © Copyright 2004 Cirrus Logic (All Rights Reserved)

List of Tables

Table A. Change History 2
Table B. General Purpose Memory Interface Pin Assignments 6
Table C. Ethernet Media Access Controller Pin Assignments 7
Table D. Audio Interfaces Pin Assignment 7
Table E. LCD Interface Pin Assignments 8
Table F. Touch Screen Interface with 12-bit Analog-to-Digital Converter Pin Assignments ... 8
Table G. 64-Key Keypad Interface Pin Assignments 8
Table H. Universal Asynchronous Receiver / Transmitters Pin Assignments 9
Table I. Triple Port USB Host Pin Assignments 9
Table J. Two-Wire Port with EEPROM Support Pin Assignments 9
Table K. Real-Time Clock with Pin Assignments 10
Table L. PLL and Clocking Pin Assignments 10
Table M. Interrupt Controller Pin Assignment 10
Table N. Dual LED Pin Assignments 10
Table O. General Purpose Input/Output Pin Assignment 11
Table P. Reset and Power Management Pin Assignments 11
Table Q. Hardware Debug Interface 11
Table R. 272 Pin Diagram Dimensions 40
Table S. Pin Descriptions 44
Table T. Pin Multiplex Usage Information 46

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Processor Core - ARM920T

The ARM920T is a Harvard architecture processor with separate 16 Kbyte instruction and data caches with an 8-word line length but a unified memory. The processor utilizes a five-stage pipeline consisting of fetch, decode, execute, memory and write stages. Key features include:

- ARM (32-bit) and Thumb (16-bit compressed) instruction sets
- 32-bit Advanced Micro-Controller Bus Architecture (AMBA)
- 16 Kbyte Instruction Cache with lockdown
- 16 Kbyte Data Cache (programmable write-through or write-back) with lockdown
- MMU for Linux®, Microsoft® Windows® CE and other operating systems
- Translation Look Aside Buffers with 64 Data and 64 Instruction Entries
- Programmable Page Sizes of 64 Kbyte, 4 Kbyte, and 1 Kbyte
- Independent lockdown of TLB Entries

MaverickCrunch™ Math Engine

The MaverickCrunch Engine is a mixed-mode coprocessor designed primarily to accelerate the math processing required to rapidly encode digital audio formats. It accelerates single and double precision integer and floating point operations plus an integer multiply-accumulate (MAC) instruction that is considerably faster than the ARM920T's native MAC instruction. The ARM920T coprocessor interface is utilized thereby sharing its memory interface and instruction stream. Hardware forwarding and interlock allows the ARM to handle looping and addressing while MaverickCrunch handles computation. Features include:

- IEEE-754 single and double precision floating point
- 32/64-bit integer
- Add/multiply/compare
- Integer MAC 32-bit input with 72-bit accumulate
- Integer Shifts
- Floating point to/from integer conversion
- Sixteen 64-bit register files
- Four 72-bit accumulators

MaverickKey™ Unique ID

MaverickKey unique hardware programmed IDs are a solution to the growing concern over secure web content and commerce. With Internet security playing an important role in the delivery of digital media such as books or music, traditional software methods are quickly becoming unreliable. The MaverickKey unique IDs provide OEMs with a method of utilizing specific hardware IDs such as those assigned for SDMI (Secure Digital Music Initiative) or any other authentication mechanism.

Both a specific 32-bit ID as well as a 128-bit random ID is programmed into the EP9307 through the use of laser probing technology. These IDs can then be used to match secure copyrighted content with the ID of the target device the EP9307 is powering, and then deliver the copyrighted information over a secure connection. In addition, secure transactions can benefit by also matching device IDs to server IDs. MaverickKey IDs provide a level of hardware security required for today's Internet appliances.

General Purpose Memory Interface (SDRAM, SRAM, ROM, FLASH)

The EP9307 features a unified memory address model where all memory devices are accessed over a common address/data bus. A separate internal port is dedicated to the read-only Raster/LCD refresh engine, while the rest of the memory accesses are performed via the Processor bus. The SRAM memory controller supports 8, 16 and 32-bit devices and accommodates an internal boot ROM concurrently with 32-bit SDRAM memory.

- 1-4 banks of 32-bit, 100 MHz SDRAM
- One internal port dedicated to the Raster/LCD Refresh Engine (Read Only)
- One internal port dedicated to the rest of the chip via the Processor bus
- Address and data bus shared between SDRAM, SRAM, ROM, and FLASH memory
- Both NAND and NOR FLASH memory supported

Table B. General Purpose Memory Interface Pin Assignments

Pin Mnemonic	Pin Description
SDCLK	SDRAM Clock
SDCLKEN	SDRAM Clock Enable
SDCSn[3:0]	SDRAM Chip Selects 3-0
RASn	SDRAM RAS
CASn	SDRAM CAS
SDWEn	SDRAM Write Enable
CSn[7:6] and CSn[3:0]	Chip Selects 7, 6, 3, 2, 1, 0
AD[25:0]	Address Bus 25-0
DA[31:0]	Data Bus 31-0
DQMn[3:0]	SDRAM Output Enables / Data Masks
WRn	SRAM Write Strobe
RDn	SRAM Read/OE Strobe
WAITn	SRAM Wait Input

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Ethernet Media Access Controller (MAC)

The MAC subsystem is compliant with the ISO/TEC 802.3 topology for a single shared medium with several stations. Multiple MII-compliant PHYs are supported. Features include:

- Supports 1/10/100 Mbps transfer rates for home/small-business/large-business applications
- Interfaces to an off-chip PHY through industry standard Media Independent Interface (MII)

Table C. Ethernet Media Access Controller Pin Assignments

Pin Mnemonic	Pin Description
MDC	Management Data Clock
MDIO	Management Data I/O
RXCLK	Receive Clock
MIIRXD[3:0]	Receive Data
RXDVAL	Receive Data Valid
RXERR	Receive Data Error
TXCLK	Transmit Clock
MIITXD[3:0]	Transmit Data
TXEN	Transmit Enable
TXERR	Transmit Error
CRS	Carrier Sense
CLD	Collision Detect

Serial Interfaces (SPI, I^{2}S and AC ’97)

The SPI port can be configured as a master or a slave, supporting the National Semiconductor®, Motorola® and Texas Instruments® signaling protocols.

The AC'97 port supports multiple codecs for multichannel audio output with a single stereo input. The I^{2}S port can be configured to support two channel, 24 bit audio.

These ports are multiplexed so that I^{2}S port 0 will take over either the AC'97 pins or the SPI pins. The second and third I2S ports' serial input and serial output pins are multiplexed with EGPIO[4,5,6,13]. The clocks supplied in the first I2S port are also used for the second and third I2S ports.

- Normal Mode: One SPI Port and one AC’97 Port
- I^{2}S on SSP Mode: One AC’97 Port and up to three I^{2}S Ports
- I^{2}S on AC’97 Mode: One SPI Port and up to three I^{2}S Ports

Note: I^{2}S may not be output on AC’97 and SSP ports at the same time.

Table D. Audio Interfaces Pin Assignment

Pin Name	Normal Mode	I2S on SSP Mode	I2S on AC'97 Mode
	Pin Description	Pin Description	Pin Description
SCLK1	SPI Bit Clock	I2S Serial Clock	SPI Bit Clock
SFRM1	SPI Frame Clock	I2S Frame Clock	SPI Frame Clock
SSPRX1	SPI Serial Input	I2S Serial Input	SPI Serial Input
SSPTX1	SPI Serial Output	I2S Serial Output	SPI Serial Output
		(No I2S Master Clock)	
ARSTn	AC'97 Reset	AC'97 Reset	I2S Master Clock
ABITCLK	AC'97 Bit Clock	AC'97 Bit Clock	I2S Serial Clock
ASYNC	AC'97 Frame Clock	AC'97 Frame Clock	I2S Frame Clock
ASDI	AC'97 Serial Input	AC'97 Serial Input	I2S Serial Input
ASDO	AC'97 Serial Output	AC'97 Serial Output	I2S Serial Output

Raster/LCD Interface

The Raster/LCD interface provides data and interface signals for a variety of display types. It features fully programmable video interface timing for non-interlaced flat panel or dual scan displays. Resolutions up to 1280 x 1024 are supported from a unified SDRAM based frame buffer. A 16-bit PWM provides control for LCD panel contrast. LCD specific features include:

- Timing and interface signals for digital LCD and TFT displays
- Full programmability for either non-interlaced or dual-scan color and grayscale flat panel displays
- Dedicated data path to SDRAM controller for improved system performance
- Pixel depths of 4, 8, 16, or 18-bits per pixel or 256 levels of grayscale
- Hardware Cursor up to 64 x 64 pixels
- 256 x 18 Color Lookup Table
- Hardware Blinking
- 8-bit interface to low end panel

©Copyright 2004 Cirrus Logic (All Rights Reserved)

Table E. LCD Interface Pin Assignments

Pin Mnemonic	Pin Description
SPCLK	Pixel Clock
P[17:0]	Pixel Data Bus [17:0]
HSYNC/LP	Horizontal Synchronization/Line Pulse
VCSYNC/FP	Vertical or Composite Synchronization / Frame Pulse
BLANK	Composite Blank
BRIGHT	Pulse Width Modulated Brightness

Graphics Accelerator

The EP9307 contains a hardware graphics acceleration engine that improves graphic performance by handling block copy, block fill and hardware line draw operations. The Graphics Accelerator is used in the system to off-load graphics operations from the processor.

Pixel depths supported by the Graphics Accelerator are 4, 8, 16 or 24 bits per pixel (bpp). The 24 bits per pixel mode can be operated as packed (4 pixels every 3 words) or unpacked (1 pixel per word with the high byte unused.)

The block copy operations of the Graphics Accelerator are similar to a DMA (Direct Memory Access) transfer that understands pixel organization, block width, transparency, and transformation from 1bpp to higher 4, 8, 16 or 24 bpp.

The line draw operations also allow for solid lines or dashed lines. The colors for line drawing can be either foreground color and background color or foreground color with the background being transparent.

Touch Screen Interface with 12-bit Analog-to-Digital Converter (ADC)

The touch screen interface performs all sampling, averaging, ADC range checking, and control for a wide variety of analog resistive touch screens. This controller only interrupts the processor when a meaningful change occurs. The touch screen hardware may be disabled and the switch matrix and ADC controlled directly if desired. Features include:

- Support for 4, 5, 7, or 8-wire analog resistive touch screens.
- Flexibility - unused lines may be used for temperature sensing or other functions.
- Touch screen interrupt function.

Table F. Touch Screen Interface with 12-bit Analog-to-Digital Converter Pin Assignments

Pin Mnemonic	Pin Description
Xp, Xm	Touch screen ADC X Axis
Yp, Ym	Touch screen ADC Y Axis
SXp, SXm	Touch screen ADC X Axis Voltage Feedback
SYp, SYm	Touch screen ADC Y Axis Voltage Feedback

64-Keypad Interface

The keypad circuitry scans an 8 x 8 array of 64 normally open, single pole switches. Any one or two keys depressed will be de-bounced and decoded. An interrupt is generated whenever a stable set of depressed keys is detected. If the keypad is not utilized, the 16 column/row pins may be used as general purpose I/O. The Keypad interface:

- Provides scanning, debounce and decoding for a 64-key array.
- Scans an 8-row by 8-column matrix.
- May decode 2 keys at once.
- Generates an interrupt when a new stable key is determined.
- Also generates a 3-key reset interrupt.

Table G. 64-Key Keypad Interface Pin Assignments

Pin Mnemonic	Pin Description	Alternative Usage
COL[7:0]	Key Matrix Column Inputs	General Purpose I/O
ROW[7:0]	Key Matrix Row Inputs	General Purpose I/O

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Universal Asynchronous Receiver/Transmitters (UARTs)

Three 16550-compatible UARTs are supplied. Two provide asynchronous HDLC (High-level Data Link Control) protocol support for full duplex transmit and receive. The HDLC receiver handles framing, address matching, CRC checking, control-octet transparency, and optionally passes the CRC to the host at the end of the packet. The HDLC transmitter handles framing, CRC generation, and control-octet transparency. The host must assemble the frame in memory before transmission. The HDLC receiver and transmitter use the UART FIFOs to buffer the data streams. A third IrDA® compatible UART is also supplied.

- UART1 supports modem bit rates up to 115.2 Kbps, supports HDLC and includes a 16 byte FIFO for receive and a 16 byte FIFO for transmit. Interrupts are generated on Rx, Tx and modem status change.
- UART2 contains an IrDA encoder operating at either the slow (up to 115 Kbps), medium (0.576 or 1.152 Mbps), or fast (4 Mbps) IR data rates. It also has a 16 byte FIFO for receive and a 16 byte FIFO for transmit.
- UART3 supports HDLC and includes a 16 byte FIFO for receive and a 16 byte FIFO for transmit. Interrupts are generated on Rx and Tx.

Table H. Universal Asynchronous Receiver / Transmitters Pin Assignments

Pin Mnemonic	Pin Name - Description
TXD0	UART1 Transmit
RXD0	UART1 Receive
CTSn	UART1 Clear To Send / Transmit Enable
DSRn/DCDn	UART1 Data Set Ready / Data Carrier Detect
DTRn	UART1 Data Terminal Ready
RTSn	UART1 Ready To Send
EGPIO[0]/RI	UART1 Ring Indicator
TXD1/SIROUT	UART2 Transmit / IrDA Output
RXD1/SIRIN	UART2 Receive / IrDA Input
TXD2	UART3 Transmit
RXD2	UART3 Receive
TENn	HDLC3 Transmit Enable

Internal Boot ROM

The Internal 16 Kbyte ROM allows booting from FLASH memory, SPI or UART.

Triple Port USB Host

The USB Open Host Controller Interface (Open HCI) provides full speed serial communications ports at a baud rate of 12 Mbits/sec. Up to 127 USB devices (printer, mouse, camera, keyboard, etc.) and USB hubs can be connected to the USB host in the USB “tiered-start” topology.

This includes the following features:

- Compliance with the USB 2.0 specification
- Compliance with the Open HCI Rev 1.0 specification
- Supports both low speed (1.5 Mbps) and full speed (12 Mbps) USB device connections
- Root HUB integrated with 3 downstream USB ports
- Transceiver buffers integrated, over-current protection on ports
- Supports power management
- Operates as a master on the bus

The Open HCI host controller initializes the master DMA transfer with the AHB bus:

- Fetches endpoint descriptors and transfer descriptors
- Accesses endpoint data from system memory
- Accesses the HC communication area
- Writes status and retire transfer descriptor

Table I. Triple Port USB Host Pin Assignments

Pin Mnemonic	Pin Name - Description
USBp[2:0]	USB Positive signals
USBm[2:0]	USB Negative Signals

Two-Wire Interface With EEPROM Support

The two-wire interface provides communication and control for EEPROM devices.

Table J. Two-Wire Port with EEPROM Support Pin Assignments

Pin Mnemonic	Pin Name - Description	Alternative Usage
EECLK	EEPROM / Two-Wire Interface Clock	General Purpose I/O
EEDATA	EEPROM / Two-Wire Interface Data	General Purpose I/O

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Real-Time Clock with Software Trim

The software trim feature on the real time clock (RTC) provides software controlled digital compensation of the 32.768 KHz crystal oscillator. This compensation is accurate to ±1.24 sec/month.

Table K. Real-Time Clock with Pin Assignments

Pin Mnemonic	Pin Name - Description
RTCXTALI	Real-Time Clock Oscillator Input
RTCXTALO	Real-Time Clock Oscillator Output

PLL and Clocking

The Processor and the Peripheral Clocks operate from a single 14.7456 MHz crystal.

The Real Time Clock operates from a 32.768 KHz crystal oscillator.

Table L. PLL and Clocking Pin Assignments

Pin Mnemonic	Pin Name - Description
XTALI	Main Oscillator Input
XTALO	Main Oscillator Output
VDD_PLL	Main Oscillator Power
GND_PLL	Main Oscillator Ground

Timers

The Watchdog Timer insures proper operation by requiring periodic attention to prevent a reset-on-time-out.

Two 16-bit timers operate as free running down-counters or as periodic timers for fixed interval interrupts and have a range of 0.03 ms to 4.27 seconds.

One 32-bit timer, plus a 6-bit prescale counter, has a range of 0.03 µs to 73.3 hours.

One 40-bit debug timer, plus a 6-bit prescale counter, has a range of 1.0 µs to 12.7 days.

Interrupt Controller

The interrupt controller allows up to 62 interrupts to generate an Interrupt Request (IRQ) or Fast Interrupt Request (FIQ) signal to the processor core. Thirty-two hardware priority assignments are provided for assisting IRQ vectoring, and two levels are provided for FIQ vectoring. This allows time critical interrupts to be processed in the shortest time possible. Internal interrupts may be programmed as active high or active low level sensitive inputs. GPIO pins programmed as interrupts may be programmed as active high level sensitive, active low level sensitive, rising edge triggered, falling edge triggered, or combined rising/falling edge triggered.

- Supports 64 interrupts from a variety of sources (such as UARTs, GPIO, and key matrix)
- Routes interrupt sources to either the ARM920T's IRQ or FIQ (Fast IRQ) inputs
- Three dedicated off-chip interrupt lines operate as active high level sensitive interrupts
- Any of the 16 GPIO lines maybe configured to generate interrupts
- Software supported priority mask for all FIQs and IRQs

Table M. Interrupt Controller Pin Assignment

Pin Mnemonic	Pin Name - Description
INT[2:0]	External Interrupts 2, 1, 0

Dual LED Drivers

Two pins are assigned specifically to drive external LEDs.

Table N. Dual LED Pin Assignments

Pin Mnemonic	Pin Name - Description	Alternative Usage
GRLED	Green LED	General Purpose I/O
REDLED	Red LED	General Purpose I/O

General Purpose Input/Output (GPIO)

The 14 EGPIO pins may each be configured individually as an output, an input, or an interrupt input.

There are 22 pins that may alternatively be used as input, output, or open-drain pins, but do not support interrupts. These pins are:

- Key Matrix ROW[7:0], COL[7:0]
- Ethernet MDIO
- Both LED Outputs
- EEPROM Clock and Data
- GGPIO[2]
- HGPIO[7:2]

6 pins may alternatively be used as inputs only:

- CTSn, DSRn/DCDn
- 4 Interrupt Lines

2 pins may alternatively be used as outputs only:

- RTSn
- ARSTn

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Table O. General Purpose Input/Output Pin Assignment

Pin Mnemonic	Pin Name - Description
EGPIO[15] EGPIO[13:0]	Expanded General Purpose Input / Output Pins with Interrupts
FGPIO[7] FGPIO[5] FGPIO[0]	Expanded General Purpose Input / Output Pins with Interrupts

Reset and Power Management

The chip may be reset through the PRSTn pin or through the open drain common reset pin, RSTOn.

Clocks are managed on a peripheral-by-peripheral basis and may be turned off to conserve power.

The processor clock is dynamically adjustable from 0 to 200 MHz (184 MHz for industrial conditions).

Table P. Reset and Power Management Pin Assignments

Pin Mnemonic	Pin Name - Description
PRSTn	Power On Reset
RSTOn	User Reset In/Out – Open Drain – Preserves Real Time Clock value

Hardware Debug Interface

The JTAG interface allows use of ARM's Multi-ICE or other in-circuit emulators.

Table Q. Hardware Debug Interface

Pin Mnemonic	Pin Name - Description
TCK	JTAG Clock
TDI	JTAG Data In
TDO	JTAG Data Out
TMS	JTAG Test Mode Select
TRSTn	JTAG Port Reset

12-Channel DMA Controller

The DMA module contains 12 separate DMA channels. These may be used for peripheral-to-memory or memory-to-peripheral access. Two of these are dedicated to memory-to-memory transfers. Each DMA channel is connected to the 16-bit DMA request bus.

The request bus is a collection of requests, Serial Audio and UARTs. Each DMA channel can be used independently or dedicated to any request signal. For each DMA channel, source and destination addressing can be independently programmed to increment, decrement, or stay at the same value. All DMA addresses are physical, not virtual addresses.

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Electrical Specifications

Absolute Maximum Ratings

(All grounds = 0 V, all voltages with respect to 0 V)

Parameter		Symbol	Min	Max	Unit
Power Supplies		RVDD CVDD VDD_PLL VDD_ADC	- - - -	3.96 2.16 2.16 3.96	V V V V
Total Power Dissipation	(Note 1)		-	2	W
Input Current per Pin, DC (Except supply pins)			-	±10	mA
Output current per pin, DC			-	±50	mA
Digital Input voltage	(Note 2)		-0.3	RVDD+0.3	V
Storage temperature			-40	+125	°C

Note: 1. Includes all power generated due to AC and/or DC output loading.
2. The power supply pins are at maximum values listed in "Recommended Operating Conditions", below.
3. At ambient temperatures above 70° C, total power dissipation must be limited to less than 2.5 Watts.

WARNING: Operation beyond these limits may result in permanent damage to the device. Normal operation is not guaranteed at these extremes.

Recommended Operating Conditions

(All grounds = 0 V, all voltages with respect to 0 V)

Parameter	Symbol	Min	Typ	Max	Unit
Power Supplies	RVDD CVDD VDD_PLL VDD_ADC	3.0 1.65 1.65 3.0	3.3 1.80 1.80 3.3	3.6 1.94 1.94 3.6	V V V V
Operating Ambient Temperature - Commercial	T_A	0	+25	+70	°C
Operating Ambient Temperature - Industrial	T_A	-40	+25	+85	°C
Processor Clock Speed - Commercial	FCLK	-	-	200	MHz
Processor Clock Speed - Industrial	FCLK	-	-	184	MHz
System Clock Speed - Commercial	HCLK	-	-	100	MHz
System Clock Speed - Industrial	HCLK	-	-	92	MHz

© Copyright 2004 Cirrus Logic (All Rights Reserved)

DC Characteristics

(T_A = 0 to 70° C; CVDD = VDD_PLL = 1.8; RVDD = 3.3 V;
All grounds = 0 V; all voltages with respect to 0 V unless otherwise noted)

Parameter			Symbol	Min	Max	Unit
High level output voltage	Iout = -4 mA	(Note 4)	V_{oh}	0.85 × RVDD	-	V
Low level output voltage	Iout = 4 mA		V_{ol}	-	0.15 × RVDD	V
High level input voltage		(Note 5)	V_{ih}	0.65 × RVDD	VDD + 0.3	V
Low level input voltage		(Note 5)	V_{il}	–0.3	0.35 × RVDD	V
High level leakage current	Vin = 3.3 V	(Note 5)	I_{ih}	-	10	µA
Low level leakage current	Vin = 0	(Note 5)	I_{il}	-	-10	µA

Parameter		Min	Typ	Max	Unit
Power Supply Pins (Outputs Unloaded)					
Power Supply Current:	CVDD/VDD_PLL Total RVDD	- -	200 20	- -	mA mA
Low-Power Mode Supply Current	CVDD/VDD_PLL Total RVDD	- -	2.5 1.0	- -	mA mA

Note: 4. For open drain pins, high level output voltage is dependent on the external load.
5. All inputs that do not include internal pull-ups or pull-downs, must be externally driven for proper operation (See Table S on page 44). If an input is not driven, it should be tied to power or ground, depending on the particular function. If an I/O pin is not driven and programmed as an input, it should be tied to power or ground through its own resistor.

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Timings

Timing Diagram Conventions

This data sheet contains one or more timing diagrams. The following key explains the components used in these diagrams. Any variations are clearly labelled when they occur. Therefore, no additional meaning should be attached unless specifically stated.

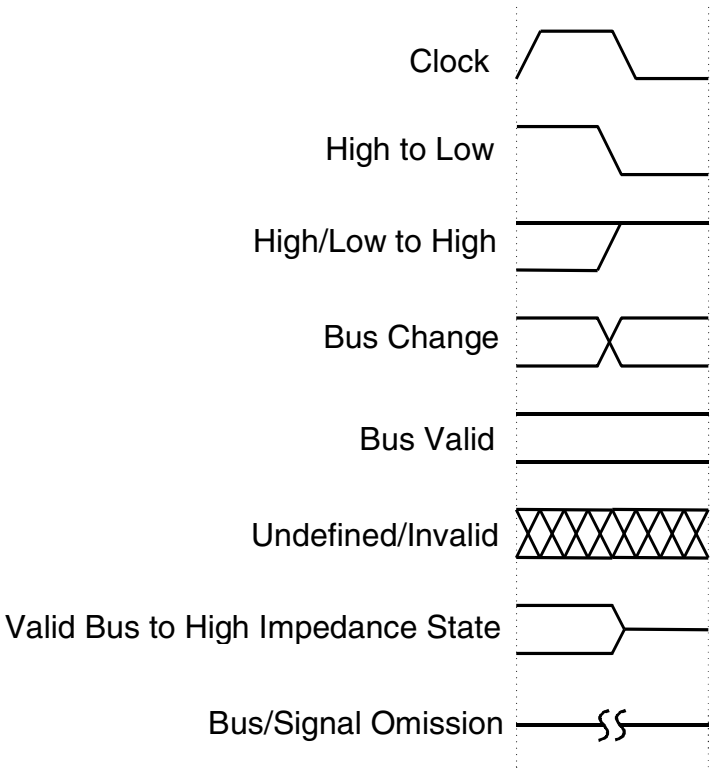


Figure 1. Timing Diagram Drawing Key

Timing Conditions

Unless specified otherwise, the following conditions are true for all timing measurements.

- T_A = 0 to 70° C
- CVDD = VDD_PLL = 1.8V
- RVDD = 3.3 V
- All grounds = 0 V
- Logic 0 = 0 V, Logic 1 = 3.3 V
- Output loading = 50 pF
- Timing reference levels = 1.5 V
- The Processor Bus Clock (HCLK) is programmable and is set by the user. The frequency is typically between 33 MHz and 100 MHz (92 MHz for industrial conditions).

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Memory Interface

Figure 2 through Figure 5 define the timings associated with all phases of the SDRAM. The following table contains the values for the timings of each of the SDRAM modes.

Parameter	Symbol	Min	Typ	Max	Unit
SDCLK high time	t_{clk_high}	-	$(t_{HCLK})/2$	-	ns
SDCLK low time	t_{clk_low}	-	$(t_{HCLK})/2$	-	ns
SDCLK rise/fall time	t_{clkrf}	-	3	-	ns
Signal delay from SDCLK rising edge time	t_d	-	8	-	ns
Signal hold from SDCLK rising edge time	t_h	-	4	-	ns
DQMn delay from SDCLK rising edge time	t_{DQd}	-	6	-	ns
DQMn hold from SDCLK rising edge time	t_{DQh}	-	6	-	ns
DA valid setup to SDCLK rising edge time	t_{DAs}	-	2	-	ns
DA valid hold from SDCLK rising edge time	t_{DAh}	-	2	-	ns

SDRAM Load Mode Register Cycle

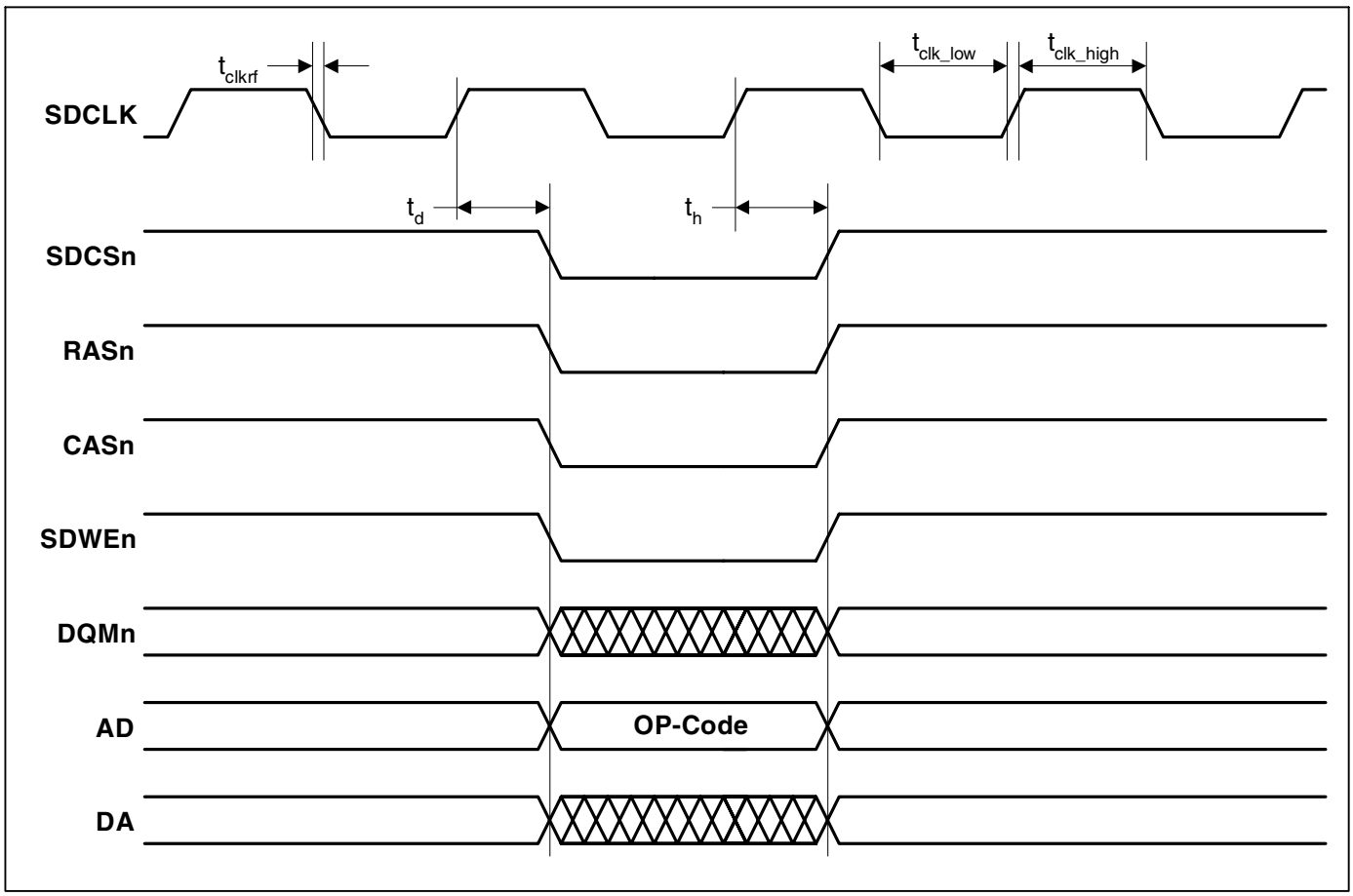


Figure 2. SDRAM Load Mode Register Cycle Timing Measurement

©Copyright 2004 Cirrus Logic (All Rights Reserved)

SDRAM Burst Read Cycle

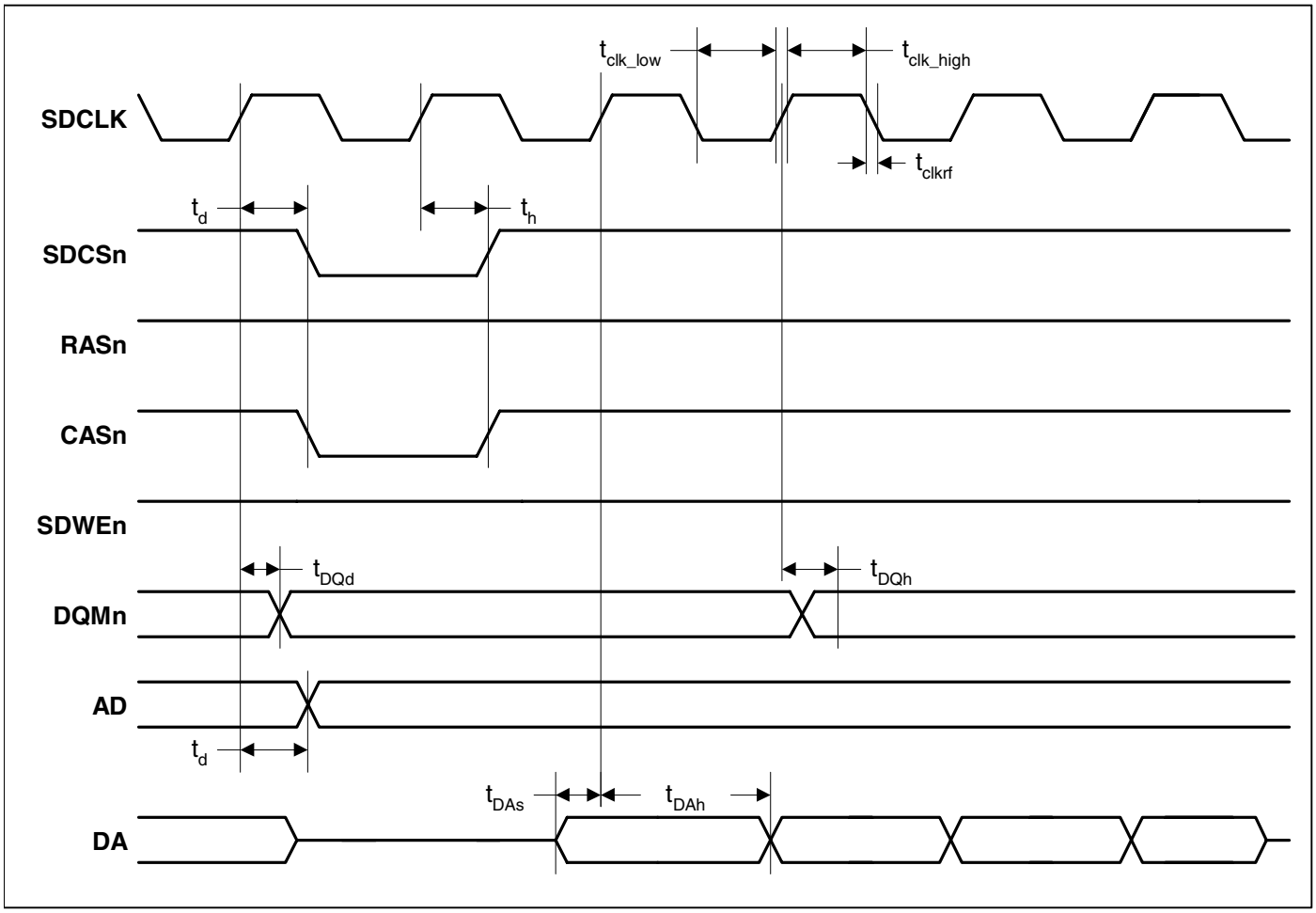


Figure 3. SDRAM Burst Read Cycle Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

SDRAM Burst Write Cycle

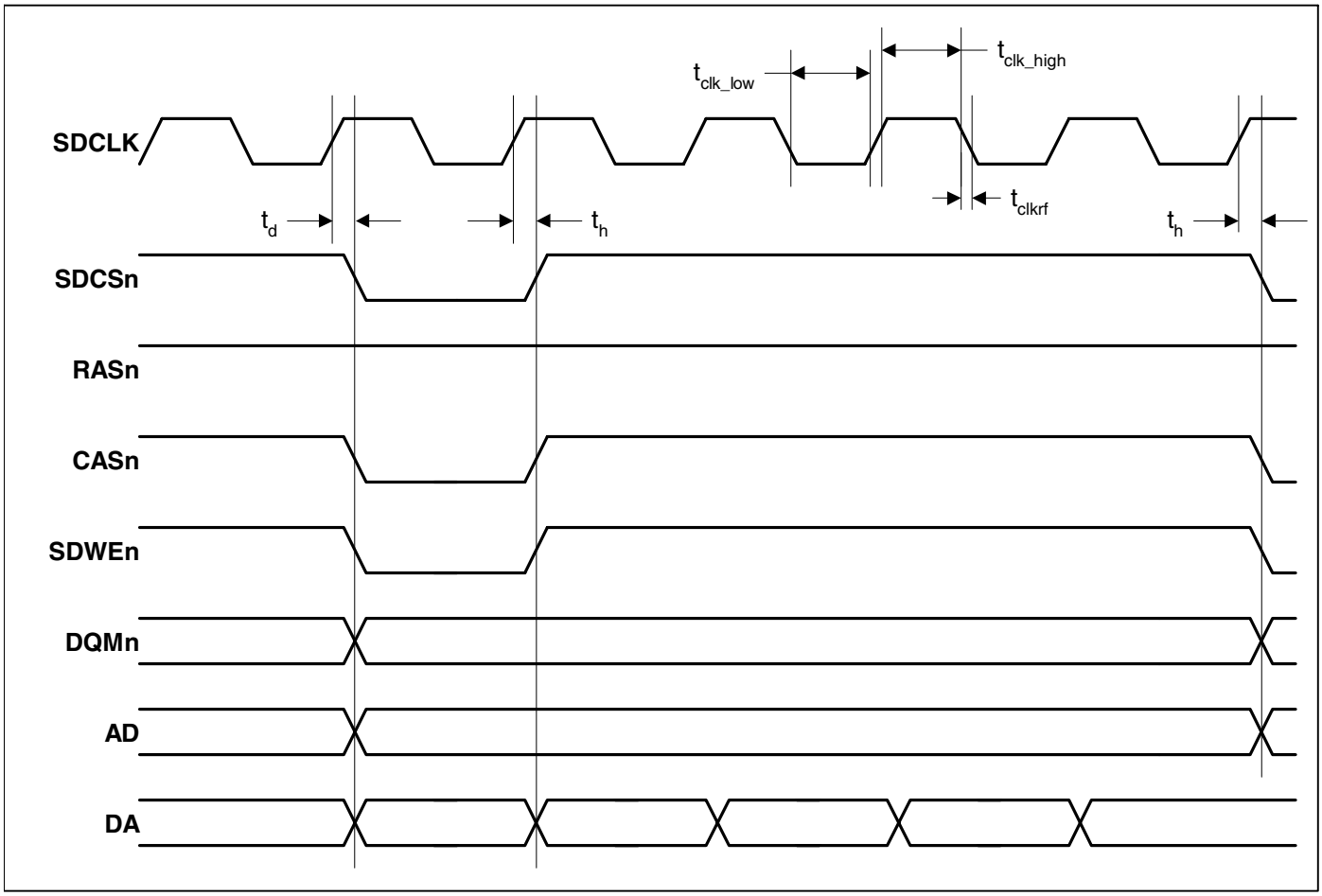
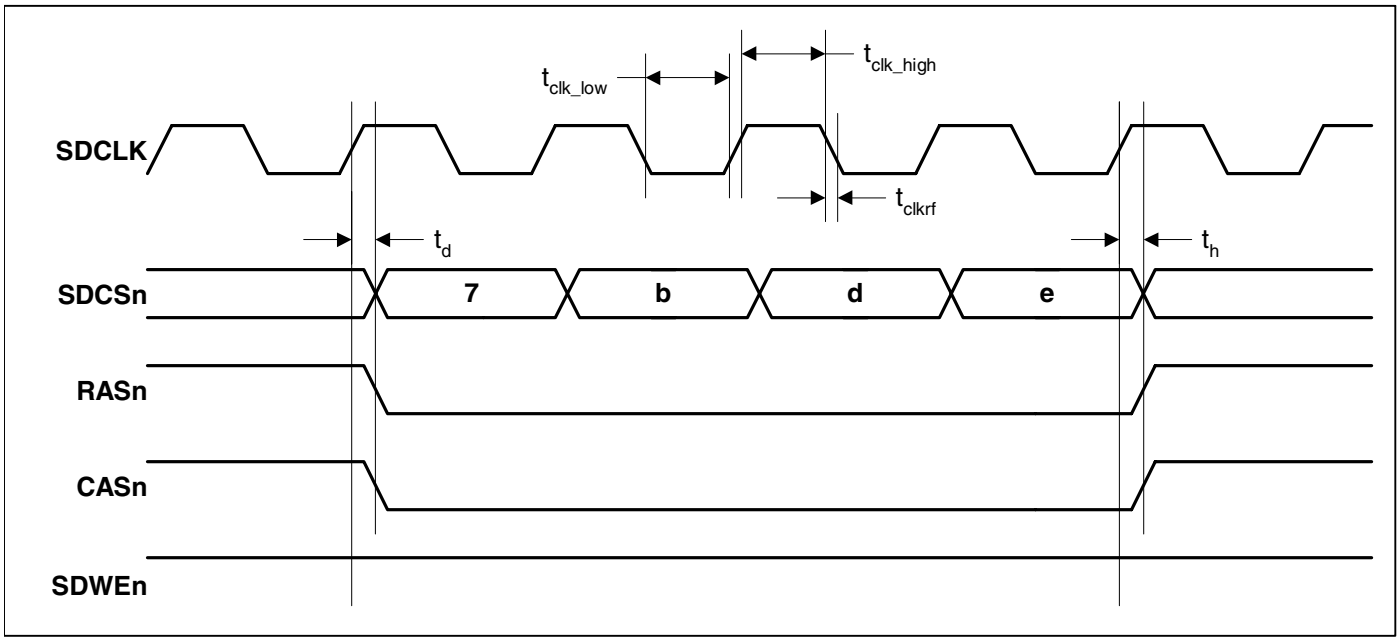


Figure 4. SDRAM Burst Write Cycle Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

SDRAM Auto Refresh Cycle



Note: Chip select shown as bus to illustrate multiple devices being put into auto refresh in one access

Figure 5. SDRAM Auto Refresh Cycle Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Static Memory Single Word Read Cycle

Parameter	Symbol	Min	Typ	Max	Unit
AD setup to RDn assert time	t_{ADs}	-	5	-	ns
AD hold from RDn deassert time	t_{ADh}	-	$t_{HCLK} \times 2$	-	ns
RDn assert time	t_{RDpw}	-	$t_{HCLK} \times (WST1 + 2)$	$t_{HCLK} \times 33$	ns
CSn assert to RDn assert delay time	t_{RDd}	-	0	-	ns
CSn deassert to RDn deassert delay time	t_{RDh}	-	0	-	ns
CSn assert to DQMn assert delay time	t_{DQMd}	-	0	-	ns
CSn deassert to DQMn deassert delay time	t_{DQMh}	-	0	-	ns
DA setup to RDn deassert time	t_{DAs}	-	$t_{HCLK} + 6$	-	ns
DA hold from RDn deassert time	t_{DAh}	0	0	-	ns

See “Timing Conditions” on page 14 for definition of HCLK.

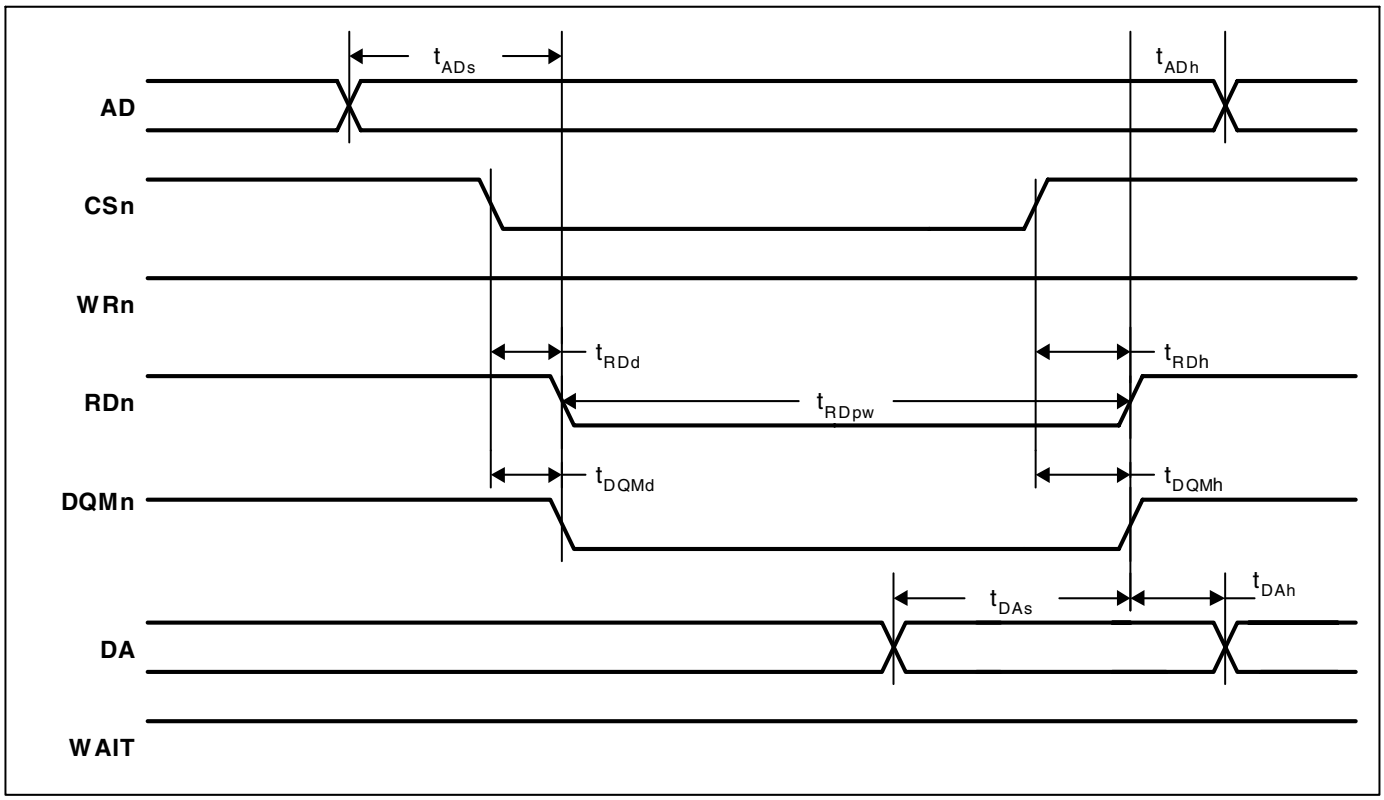


Figure 6. Static Memory Single Word Read Cycle Timing Measurement

©Copyright 2004 Cirrus Logic (All Rights Reserved)

Static Memory Single Word Write Cycle

Parameter	Symbol	Min	Typ	Max	Unit
AD setup to WRn assert time	t_{ADs}	-	t_{HCLK}	-	ns
AD hold from WRn deassert time	t_{ADh}	-	$t_{HCLK} \times 3$	-	ns
WRn deassert to CSn deassert time	t_{CSh}	-	t_{HCLK}	-	ns
CSn to WRn assert delay time	t_{WRd}	-	0	-	ns
WRn assert time	t_{WRpw}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
CSn to DQMn assert delay time	t_{DQMd}	-	0	-	ns
WRn deassert to DQMn deassert time	t_{DQMh}	-	0	-	ns
WRn deassert to DA transition time	t_{DAh}	-	t_{HCLK}	-	ns

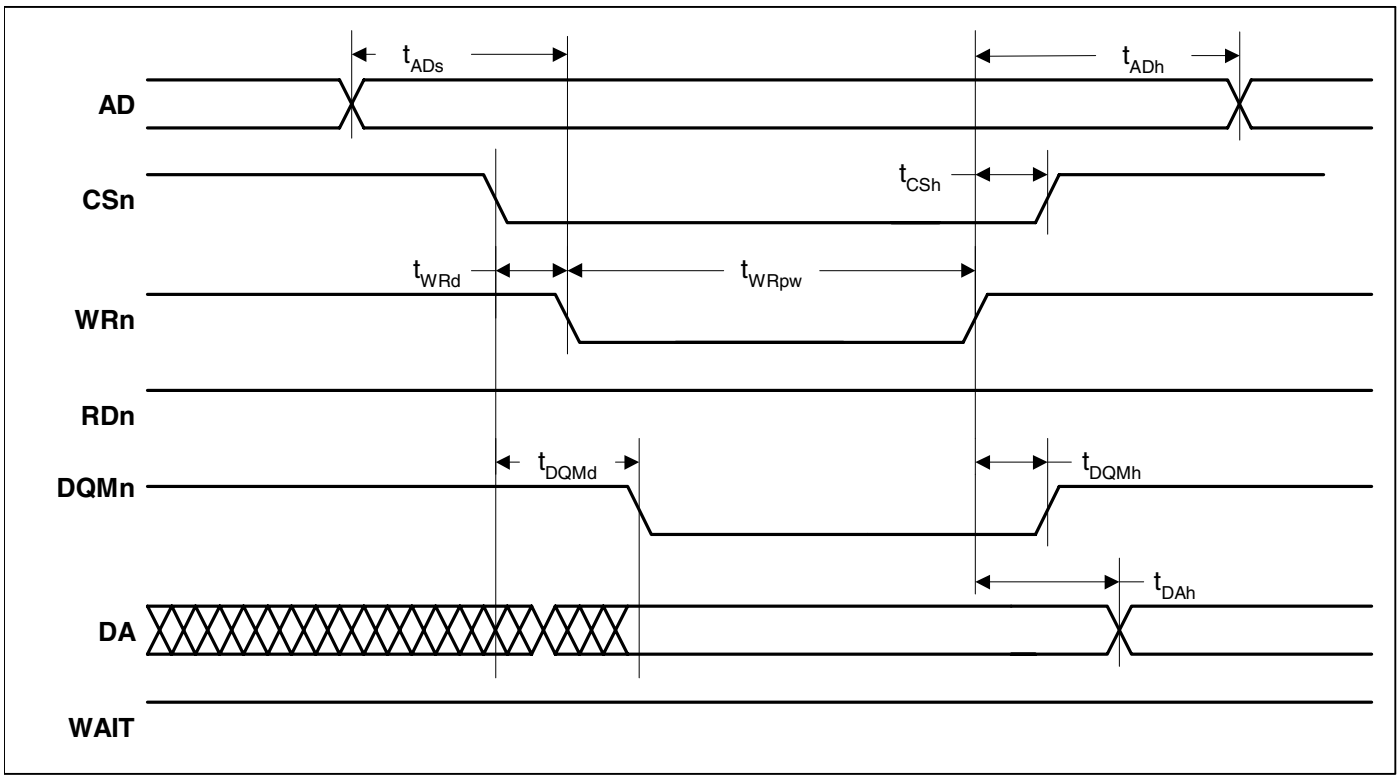


Figure 7. Static Memory Single Word Write Cycle Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Static Memory 32-bit Read on 8-bit External Bus

Parameter	Symbol	Min	Typ	Max	Unit
AD setup to RDn assert time	t_{ADs}	-	t_{HCLK}	-	ns
RDn assert to Address 1 transition time	t_{AD1}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
Address 2 assert time	t_{AD2}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
Address 3 assert time	t_{AD3}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
AD transition to RDn deassert time	t_{AD4}	-	$t_{HCLK} \times (WST1 + 2)$	-	ns
AD hold from RDn deassert time	t_{ADh}	-	$t_{HCLK} \times 2$	-	ns
RDn assert time	t_{RDpwL}	-	$t_{HCLK} \times (4 \times WST1 + 5)$	-	ns
CSn assert to RDn assert delay time	t_{RDd}	-	0	-	ns
CSn deassert to RDn deassert delay time	t_{RDh}	-	0	-	ns
CSn assert to DQMn assert delay time	t_{DQMd}	-	0	-	ns
CSn deassert to DQMn deassert delay time	t_{DQMh}	-	0	-	ns
DA setup to AD transition time	t_{DAs1}	-	6	-	ns
DA to RDn setup time	t_{DAs2}	-	$t_{HCLK} + 6$	-	ns
AD transition to DA transition hold time	t_{DAh1}	-	0	-	ns
RDn deassert to DA transition hold time	t_{DAh2}	-	0	-	ns

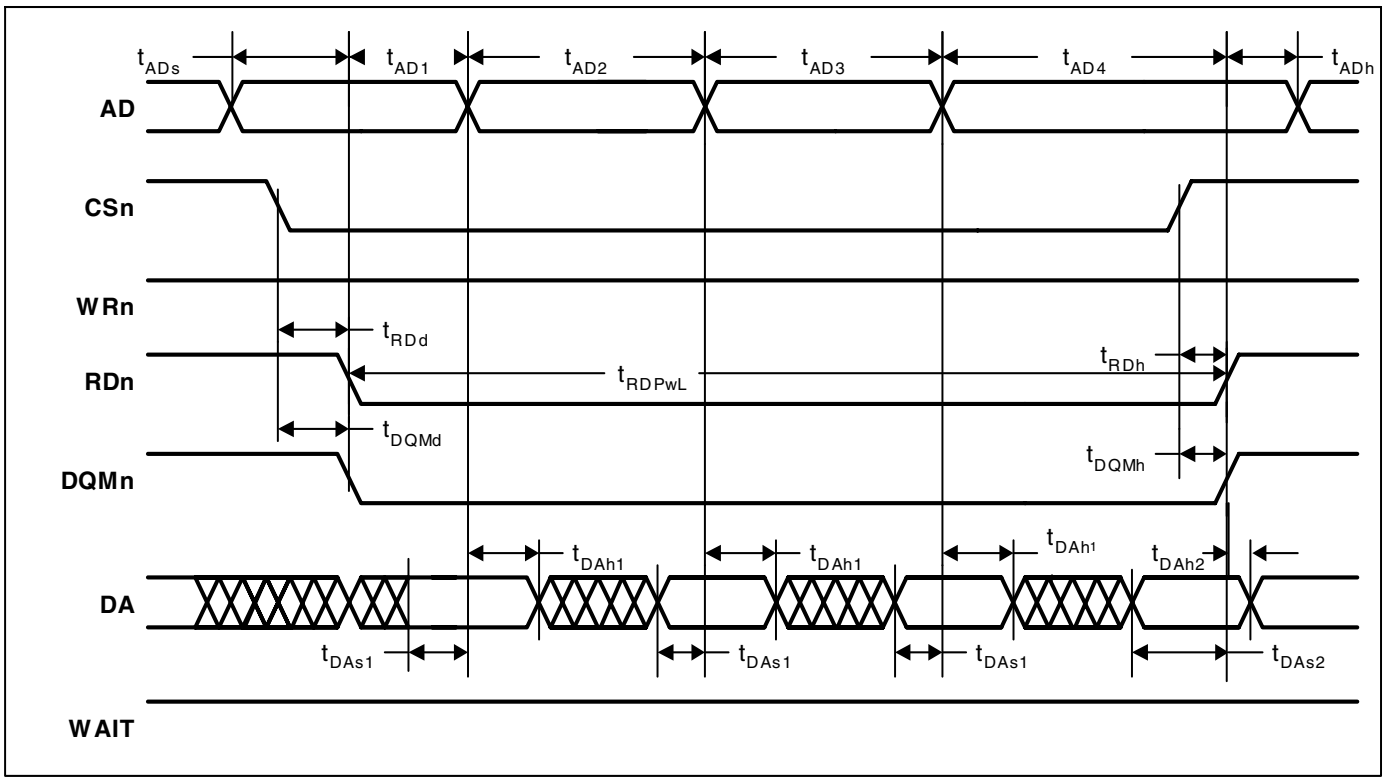


Figure 8. Static Memory Multiple Word Read 8 Bit Cycle Timing Measurement

©Copyright 2004 Cirrus Logic (All Rights Reserved)

Static Memory 32-bit Write on 8-bit External Bus

Parameter	Symbol	Min	Typ	Max	Unit
AD setup to WRn assert time	t_{ADs}	-	t_{HCLK}	-	ns
WRn deassert to AD transition time	t_{ADd}	-	t_{HCLK}	-	ns
AD hold from WRn deassert time	t_{ADh}	-	$t_{HCLK} \times 3$	-	ns
CSn hold from WRn deassert time	t_{CSh}	-	t_{HCLK}	-	ns
CSn to WRn assert delay time	t_{WRd}	-	0	-	ns
WRn assert time	t_{WRpwL}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
WRn deassert time	t_{WRpwH}	-	$t_{HCLK} \times 2$	-	ns
CSn to DQMn assert delay time	t_{DQMd}	-	0	-	ns
DQMn assert time	t_{DQMpwL}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
DQMn deassert time	t_{DQMpwH}	-	$t_{HCLK} \times 2$	-	ns
WRn/DQMn deassert to DA transition time	t_{DAh}	-	t_{HCLK}	-	ns

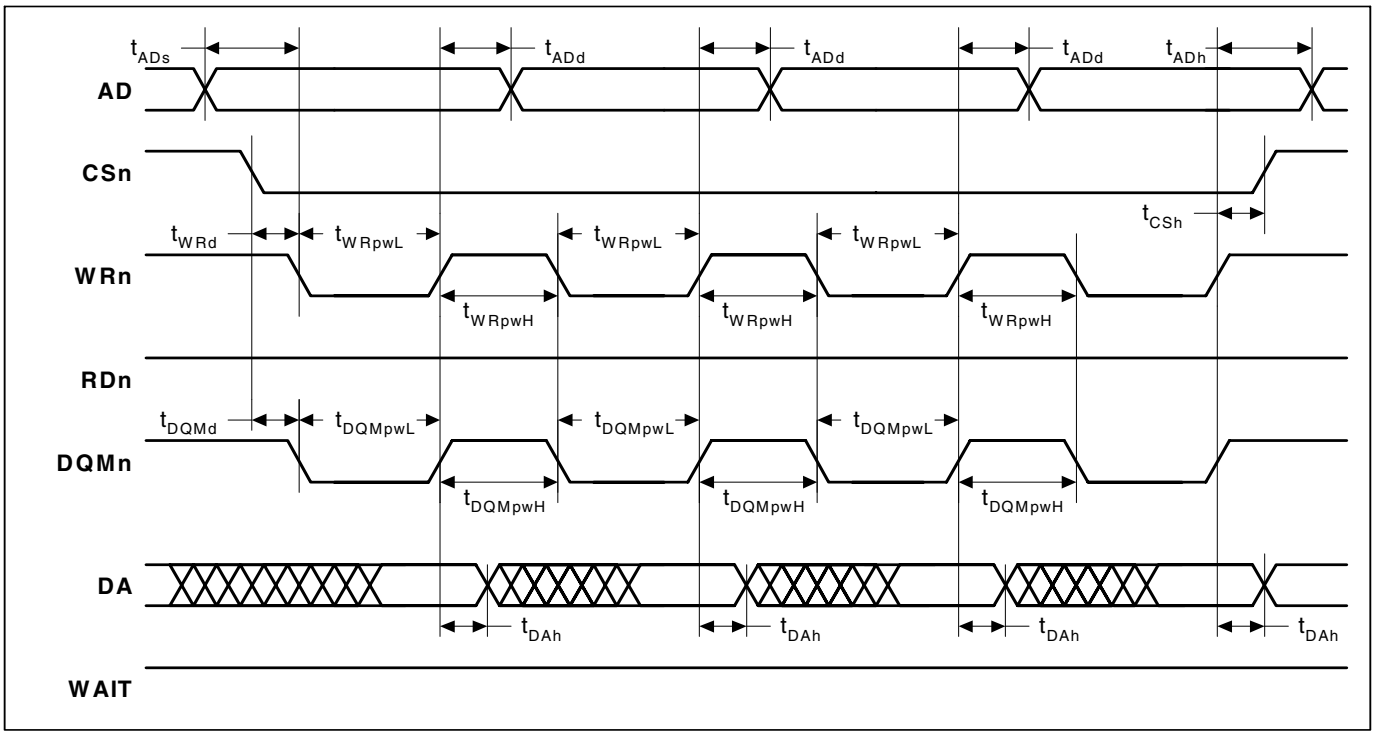


Figure 9. Static Memory Multiple Word Write 8 bit Cycle Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Static Memory 32-bit Read on 16-bit External Bus

Parameter	Symbol	Min	Typ	Max	Unit
AD setup to RDn assert time	t_{ADs}	-	t_{HCLK}	-	ns
RDn assert to AD transition time	t_{ADd1}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
AD transition to RDn deassert time	t_{ADd2}	-	$t_{HCLK} \times (WST1 + 2)$	-	ns
AD hold from RDn deassert time	t_{ADh}	-	$t_{HCLK} \times 2$	-	ns
RDn assert time	t_{RDpwL}	-	$t_{HCLK} \times (2 \times WST1 + 3)$	-	ns
CSn to RDn assert delay time	t_{RDd}	-	0	-	ns
CSn to RDn deassert delay time	t_{RDh}	-	0	-	ns
CSn to DQMn assert delay time	t_{DQMd}	-	0	-	ns
CSn to DQMn deassert delay time	t_{DQMh}	-	0	-	ns
DA to ADsetup time	t_{DAs1}	-	6	-	ns
DA to RDn setup time	t_{DAs2}	-	$t_{HCLK} + 6$	-	ns
AD transition to DA transition hold time	t_{DAh1}	-	0	-	ns
RDn deassert to DA transition hold time	t_{DAh2}	-	0	-	ns

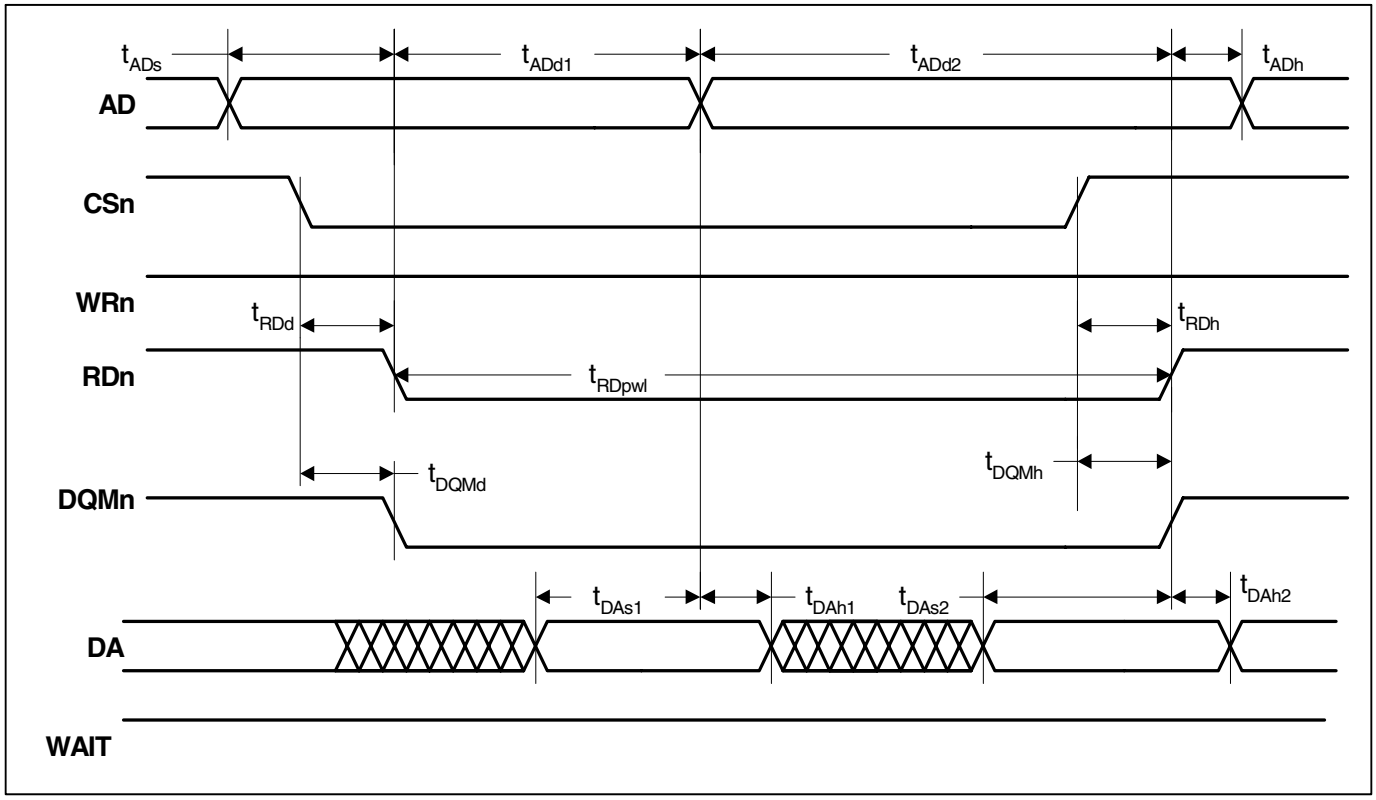


Figure 10. Static Memory Multiple Word Read 16 Bit Cycle Timing Measurement

©Copyright 2004 Cirrus Logic (All Rights Reserved)

Static Memory 32-bit Write on 16-bit External Bus

Parameter	Symbol	Min	Typ	Max	Unit
AD setup to WRn assert time	t_{ADs}	-	t_{HCLK}	-	ns
WRn deassert to AD transition time	t_{ADd}	-	t_{HCLK}	-	ns
AD hold from WRn deassert time	t_{ADh}	-	$2 \times t_{HCLK}$	-	ns
CSn hold from WRn deassert time	t_{CSh}	-	t_{HCLK}	-	ns
CSn to WRn assert delay time	t_{WRd}	-	0	-	ns
WRn assert time	t_{WRpwL}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
WRn deassert time	t_{WRpwH}	-	$t_{HCLK} \times 2$	-	ns
CSn to DQMn assert delay time	t_{DQMd}	-	0	-	ns
DQMn assert time	t_{DQMpwL}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
DQMn deassert time	t_{DQMpwH}	-	$t_{HCLK} \times 2$	-	ns
WRn/DQMn deassert to DA transition time	t_{DAh1}	-	t_{HCLK}	-	ns
WRn/DQMn deassert to DA transition time	t_{DAh2}	-	t_{HCLK}	-	ns

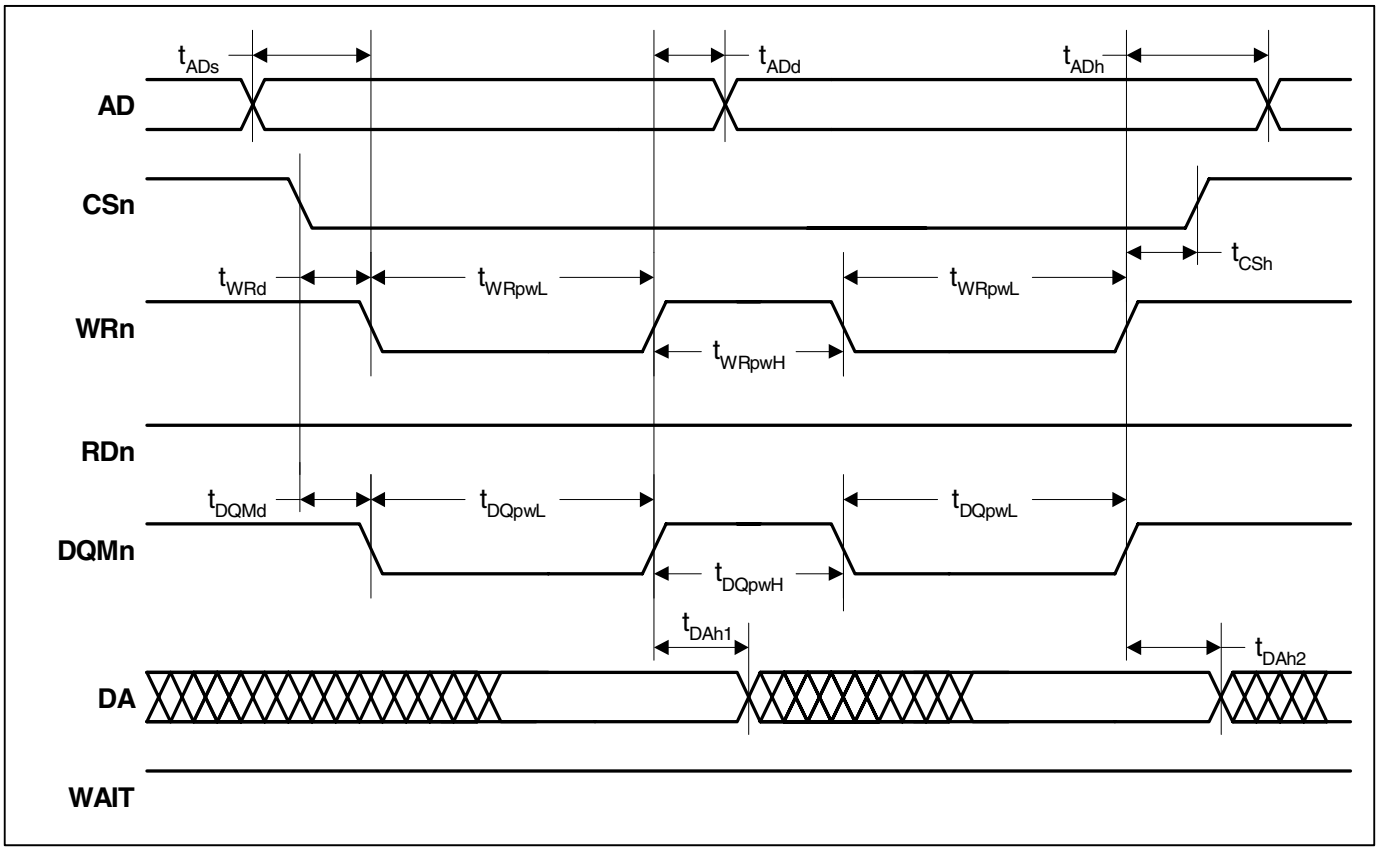


Figure 11. Static Memory Multiple Word Write 16 bit Cycle Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Static Memory Burst Read Cycle

Parameter	Symbol	Min	Typ	Max	Unit
CSn assert to Address 1 transition time	t_{ADd1}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
Address 2 assert time	t_{ADd2}	-	$t_{HCLK} \times (WST2 + 1)$	-	ns
AD hold from CSn deassert time	t_{ADh}	-	$t_{HCLK} \times 2$	-	ns
CSn assert time	t_{CSpw}	-	$t_{HCLK} \times ((WST1 + 1) + 4(WST2 + 1))$	-	ns
CSn to RDn assert delay time	t_{RDd}	-	0	-	ns
RDn assert time	t_{RDpw}	-	$t_{HCLK} \times ((WST1 + 1) + 4(WST2 + 1))$	-	ns
CSn to DQMn assert delay time	t_{DQMd}	-	4	-	ns
DQMn assert time	t_{DQMpw}	-	$t_{HCLK} \times ((WST1 + 1) + 4(WST2 + 1))$	-	ns
DA to AD setup time	t_{DAs1}	-	6	-	ns
DA to CSn setup time	t_{DAs2}	-	$t_{HCLK} + 6$	-	ns
AD transition to DA transition hold time	t_{DAh1}	-	0	-	ns
CSn deassert to DA transition hold time	t_{DAh2}	0	0	-	ns

Note: These characteristics are valid when the Page Mode Enable (Burst Mode) bit is set. See the User's Guide for details.

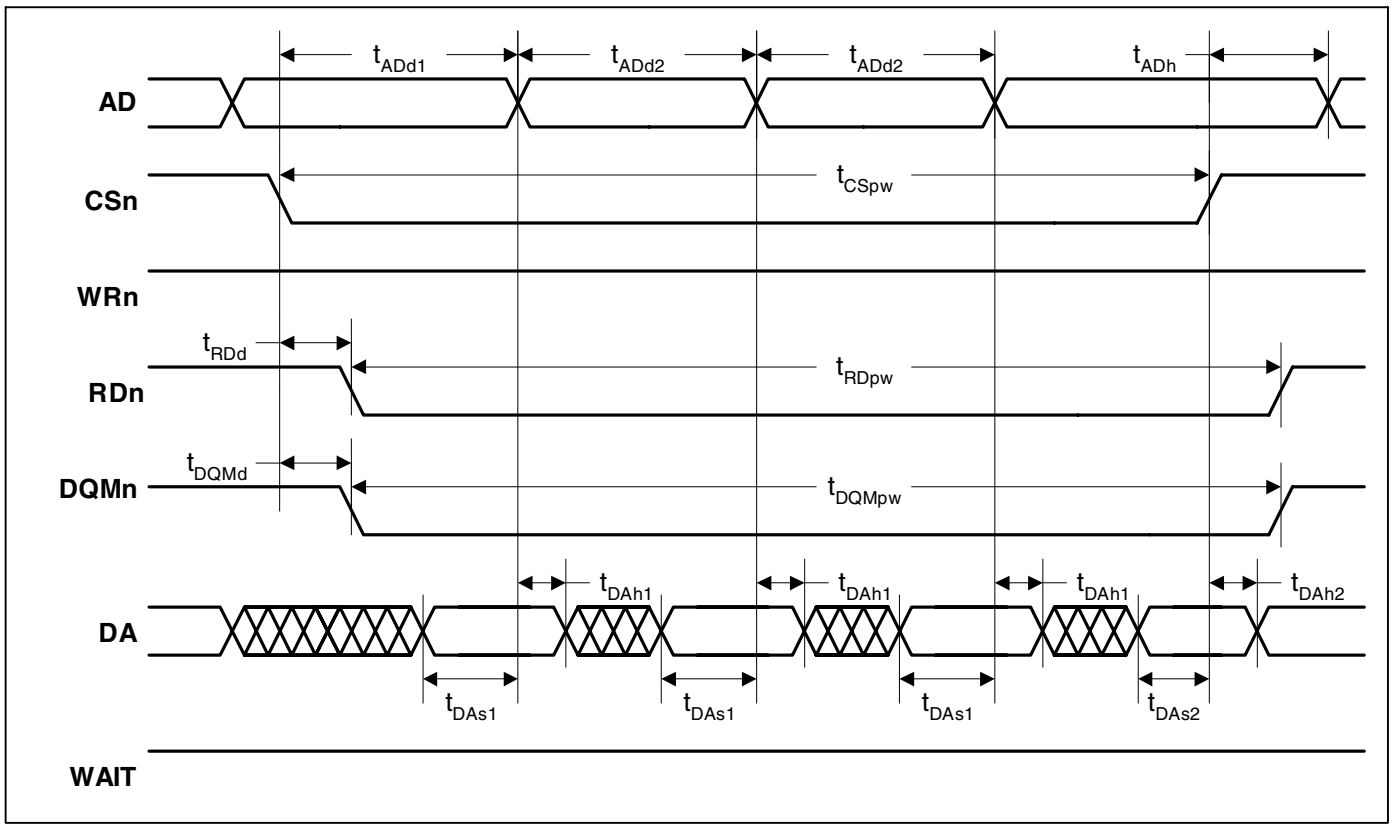


Figure 12. Static Memory Burst Read Cycle Timing Measurement

©Copyright 2004 Cirrus Logic (All Rights Reserved)

Static Memory Single Read Wait Cycle

Parameter	Symbol	Min	Typ	Max	Unit
CSn assert to WAIT time	t_{WAITd}	-	-	$t_{HCLK} \times (WST1 - 2)$	ns
WAIT assert time	t_{WAITpw}	$t_{HCLK} \times 2$	-	$t_{HCLK} \times 510$	ns
WAIT to CSn deassert delay time	t_{CSnd}	$t_{HCLK} \times 3$	-	$t_{HCLK} \times 5$	ns

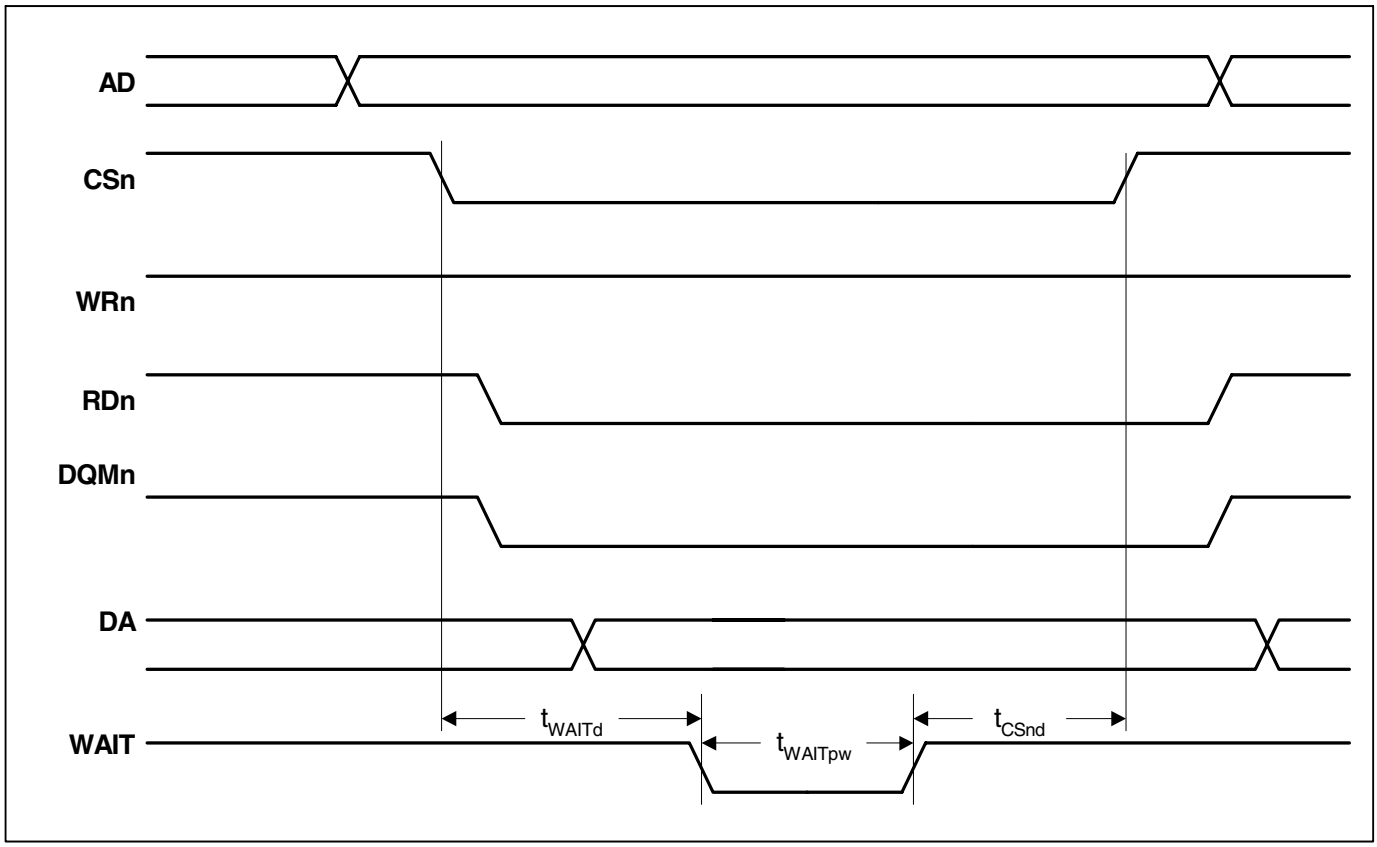


Figure 13. Static Memory Single Read Wait Cycle Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Static Memory Single Write Wait Cycle

Parameter	Symbol	Min	Typ	Max	Unit
WAIT to WRn deassert delay time	t_{WRd}	$t_{HCLK} \times 2$	-	$t_{HCLK} \times 4$	ns
CSn assert to WAIT time	t_{WAITd}	-	-	$t_{HCLK} \times$ (WST1 - 2)	ns
WAIT assert time	t_{WAITpw}	$t_{HCLK} \times 2$	-	$t_{HCLK} \times 510$	ns
WAIT to CSn deassert delay time	t_{CSnd}	$t_{HCLK} \times 3$	-	$t_{HCLK} \times 5$	ns

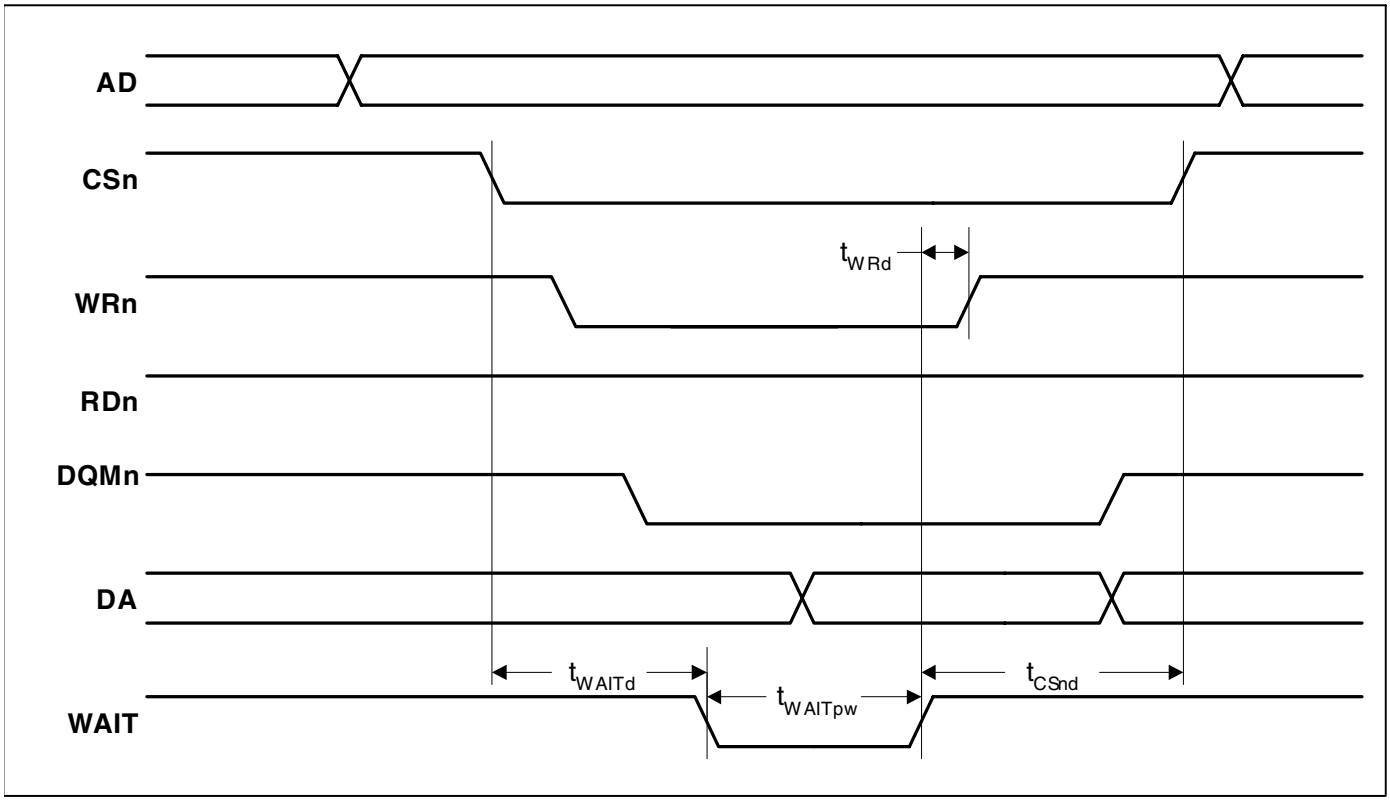


Figure 14. Static Memory Single Write Wait Cycle Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Static Memory Turnaround Cycle

Parameter	Symbol	Min	Typ	Max	Unit
CSnX deassert to CSnY assert time	t_{BTcyc}	-	$t_{HCLK} \times (IDCY+1)$	-	ns

Note: X and Y represent any two chip select numbers.

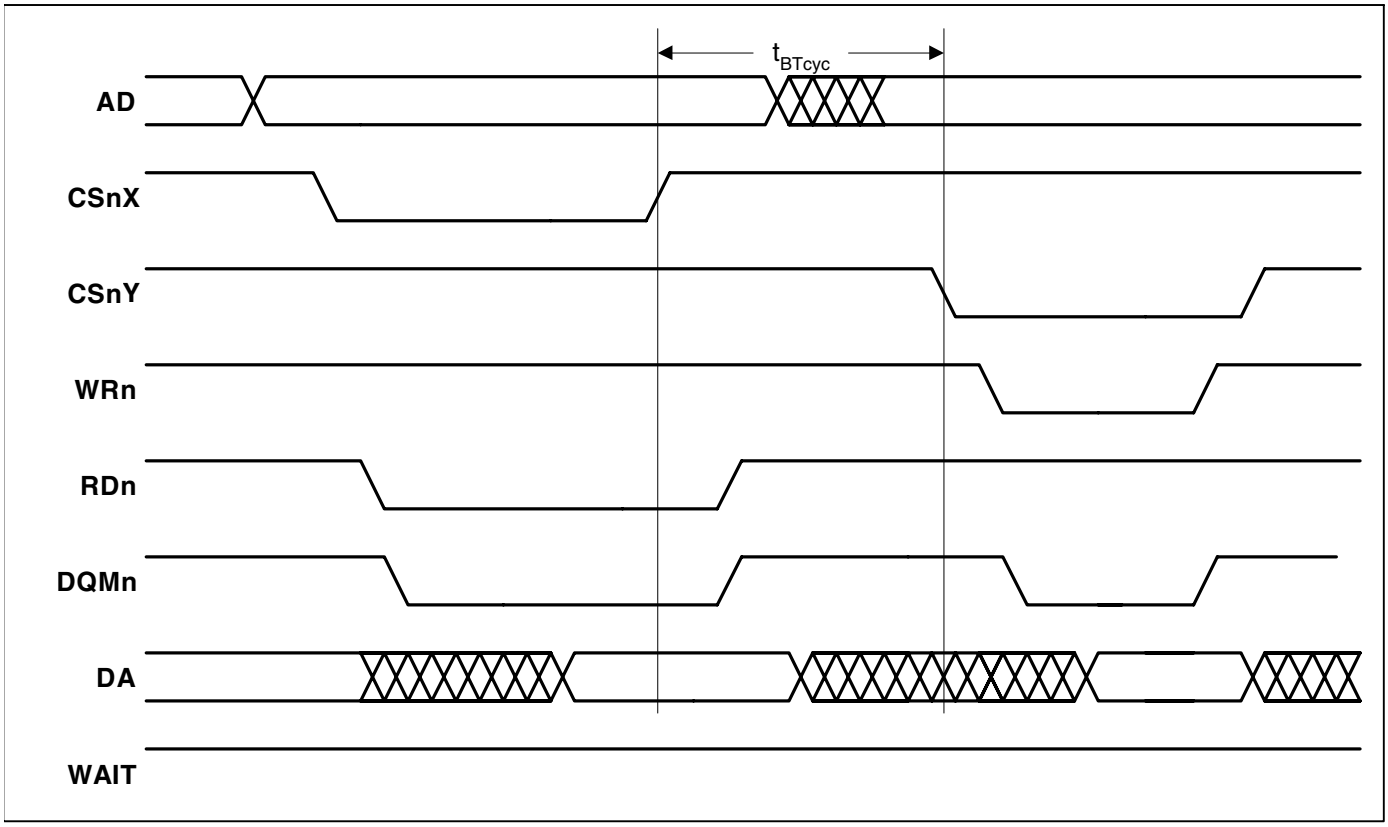


Figure 15. Static Memory Turnaround Cycle Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Ethernet MAC Interface

Parameter	Symbol	Min		Typ		Max		Unit
		10 Mbit mode	100 Mbit mode	10 Mbit mode	100 Mbit mode	10 Mbit mode	100 Mbit mode	
TXCLK cycle time	t_{TX_per}	-	-	400	40	-	-	ns
TXCLK high time	t_{TX_high}	140	14	200	20	260	26	ns
TXCLK low time	t_{TX_low}	140	14	200	20	260	26	ns
TXCLK to signal transition delay time	t_{TXd}	0	0	10	10	25	25	ns
TXCLK rise/fall time	t_{TXrf}	-	-	-	-	5	5	ns
RXCLK cycle time	t_{RX_per}	-	-	400	40	-	-	ns
RXCLK high time	t_{RX_high}	140	14	200	20	260	26	ns
RXCLK low time	t_{RX_low}	140	14	200	20	260	26	ns
RXDVAL/RXERR setup time	t_{RXs}	10	10	-	-	-	-	ns
RXDVAL/RXERR hold time	t_{RXh}	10	10	-	-	-	-	ns
RXCLK rise/fall time	t_{RXrf}	-	-	-	-	5	5	ns
MDC cycle time	t_{MDC_per}	400	400	-	-	-	-	ns
MDC high time	t_{MDC_high}	160	160	-	-	-	-	ns
MDC low time	t_{MDC_low}	160	160	-	-	-	-	ns
MDC rise/fall time	t_{MDCrf}	-	-	-	-	5	5	ns
MDIO setup time (STA sourced)	t_{MDIOs}	10	10	15	15	-	-	ns
MDIO hold time (STA sourced)	t_{MDIOh}	10	10	15	15	-	-	ns
MDC to MDIO signal transition delay time (PHY sourced)	t_{MDIOd}	-	-	-	-	300	300	ns

STA - Station - Any device that contains an IEEE 802.11 conforming Medium Access Control (MAC) and physical layer (PHY) interface to the wireless medium.

PHY - Ethernet physical layer interface.

©Copyright 2004 Cirrus Logic (All Rights Reserved)

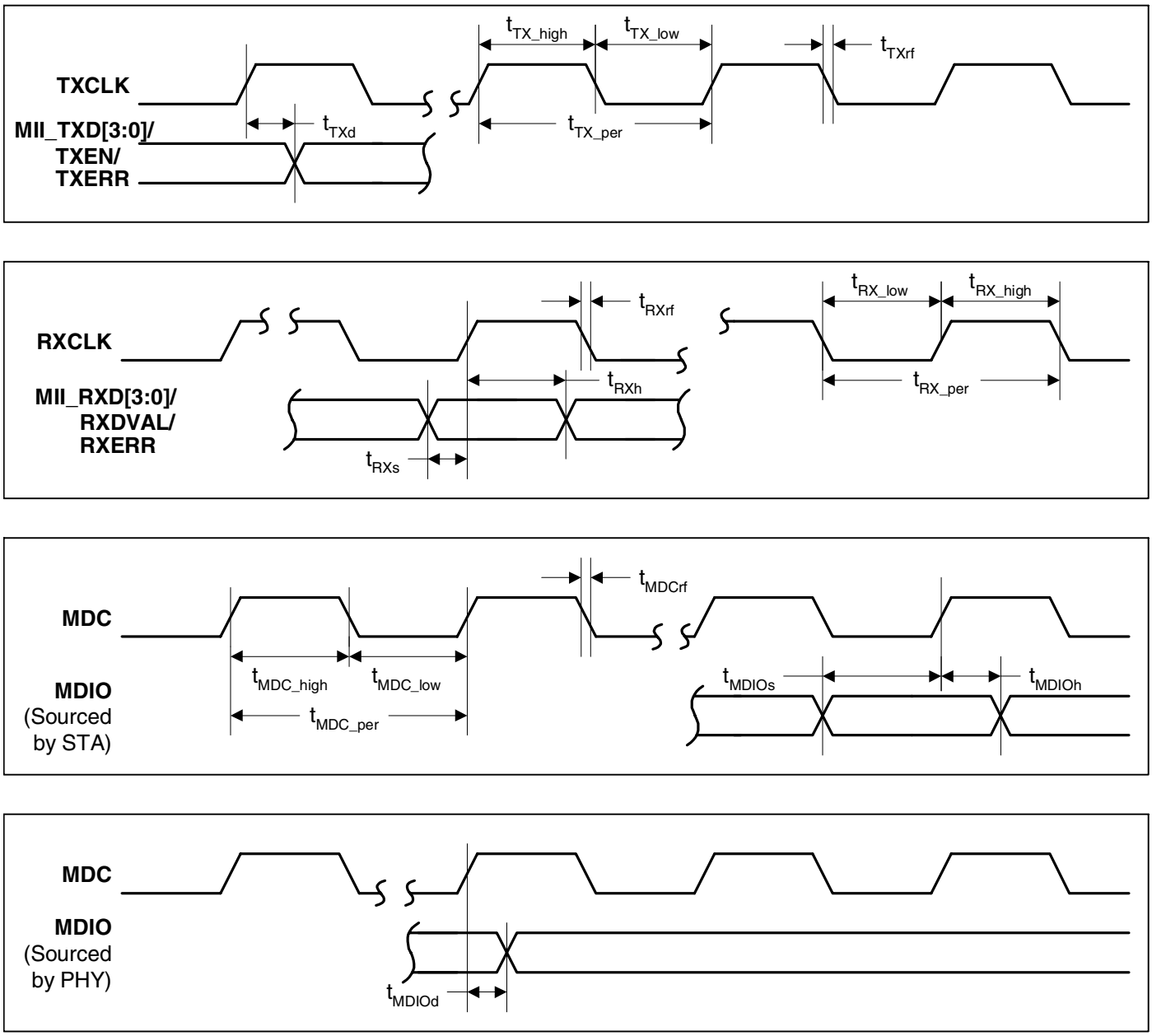


Figure 16. Ethernet MAC Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Audio Interface

The following table contains the values for the timings of each of the SPI modes.

Parameter	Symbol	Min	Typ	Max	Unit
SCLK cycle time	t_{clk_per}	-	tspix_clk	-	ns
SCLK high time	t_{clk_high}	-	(tspix_clk)/2	-	ns
SCLK low time	t_{clk_low}	-	(tspix_clk)/2	-	ns
SCLK rise/fall time	t_{clkrf}	-	4.5 / 1.5	-	ns
Data from master valid delay time	t_{DMd}	-	2	-	ns
Data from master setup time	t_{DMs}	-	20	-	ns
Data from master hold time	t_{DMh}	-	40	-	ns
Data from slave valid delay time	t_{DSd}	-	2	-	ns
Data from slave setup time	t_{DSs}	-	20	-	ns
Data from slave hold time	t_{DSh}	-	40	-	ns

Note: tspix_clk is programmable by the user.

©Copyright 2004 Cirrus Logic (All Rights Reserved)

Texas Instruments' Synchronous Serial Format

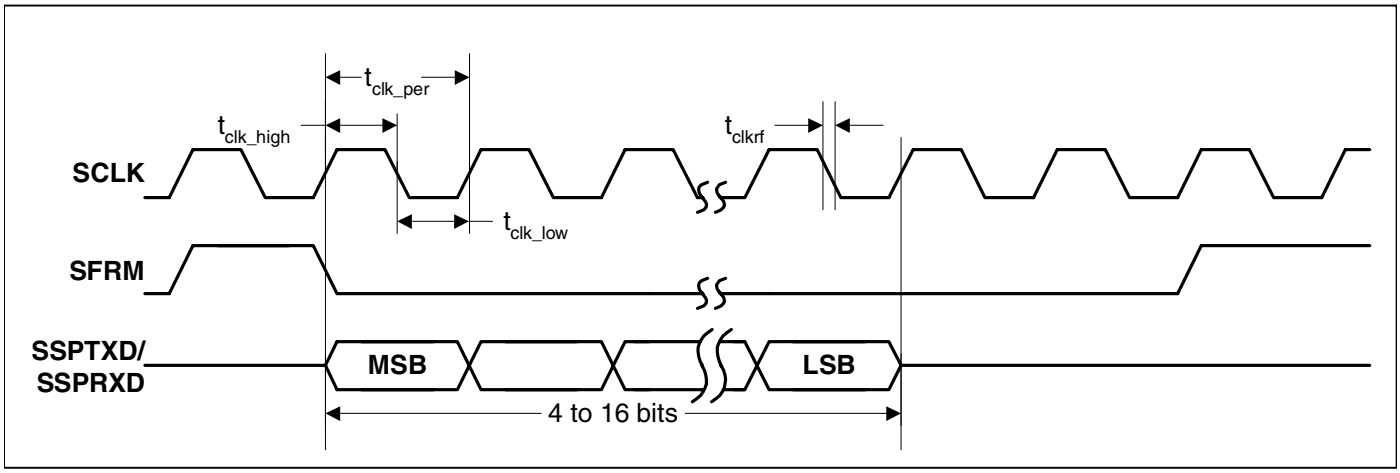


Figure 17. SPI Single Transfer Timing Measurement

Microwire

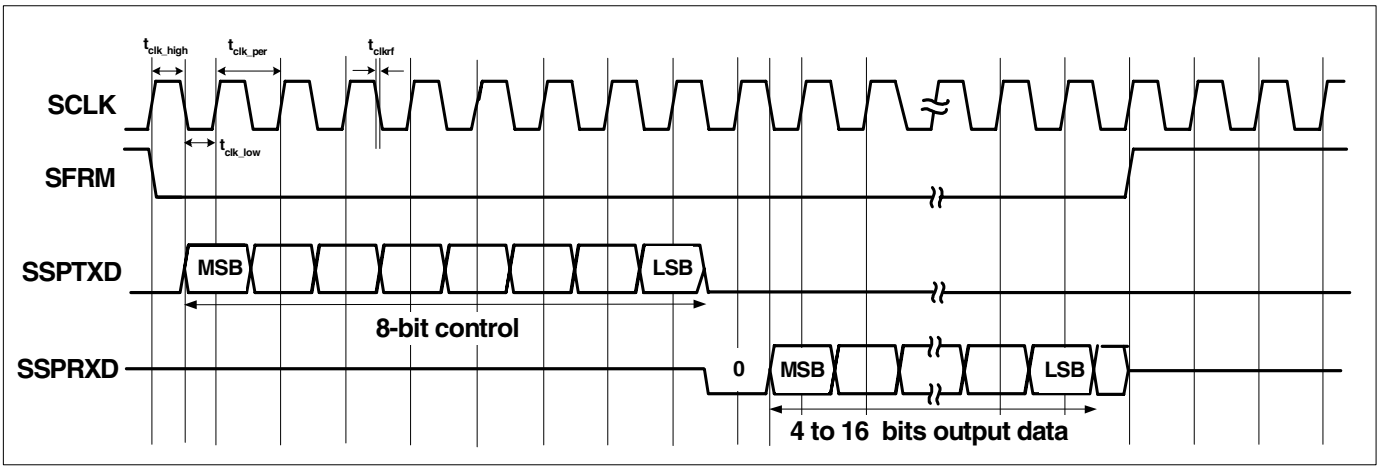


Figure 18. Microwire Frame Format, Single Transfer

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Motorola SPI

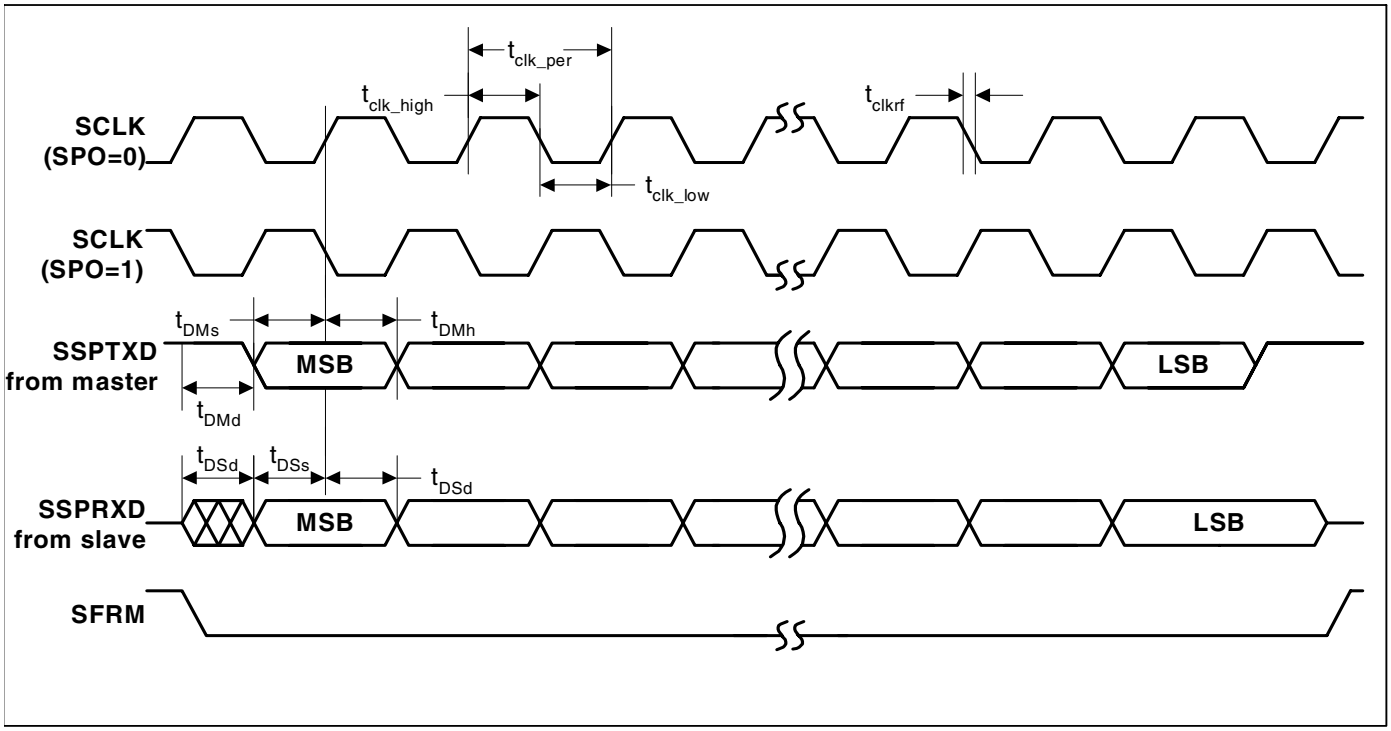


Figure 19. SPI Format with SPH=1 Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Inter-IC Sound - I²S

Parameter	Symbol	Min	Typ	Max	Unit
SCLK cycle time	t_{clk_per}	-	t_{i2s_clk}	-	ns
SCLK high time	t_{clk_high}	-	$(t_{i2s_clk}) / 2$	-	ns
SCLK low time	t_{clk_low}	-	$(t_{i2s_clk}) / 2$	-	ns
SCLK rise/fall time	t_{clkrf}	-	4	-	ns
SCLK to LRCLK assert delay time	t_{LRs}	-	1.5	-	ns
LRCLK from SCLK assert hold time	t_{LRh}	-	1.5	-	ns
SDI to SCLK deassert setup time	t_{SDIs}	-	20	-	ns
SDI from SCLK deassert hold time	t_{SDIh}	-	10	-	ns
SCLK to SDO assert delay time	t_{SDOd}	-	4	-	ns
SDO from SCLK assert hold time	t_{SDOh}	-	4	-	ns

Note: t_{i2s_clk} is programmable by the user.

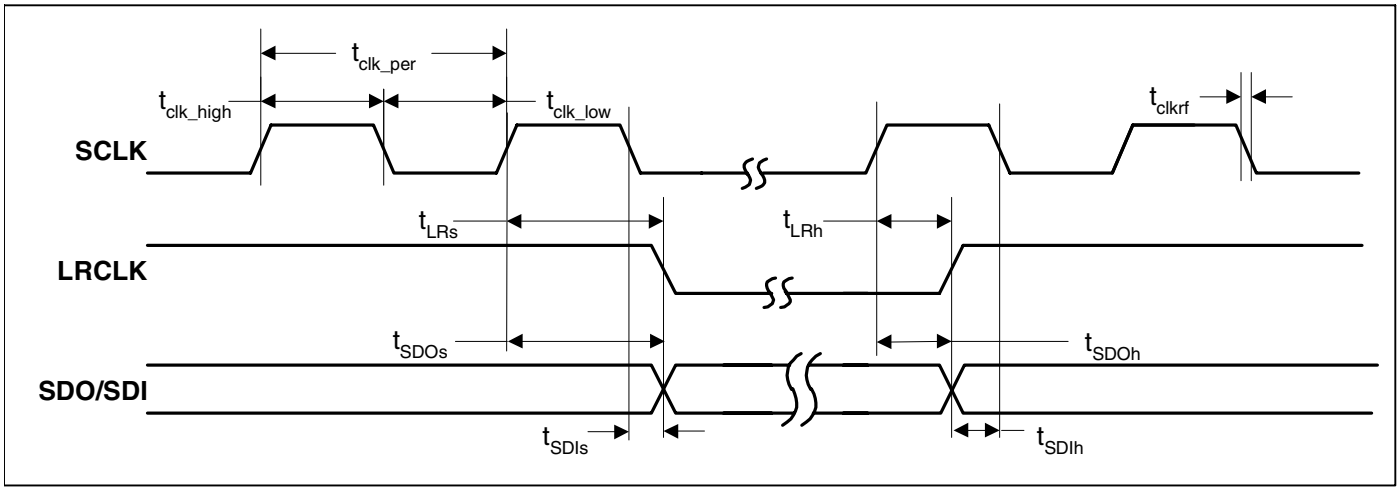


Figure 20. Inter-IC Sound (I²S) Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

AC'97

Parameter	Symbol	Min	Typ	Max	Unit
ABITCLK input cycle time	t_{clk_per}	-	81.4	-	ns
ABITCLK input high time	t_{clk_high}	36	-	45	ns
ABITCLK input low time	t_{clk_low}	36	-	45	ns
ABITCLK input rise time	t_{clkr}	2	-	6	ns
ABITCLK input fall time	t_{clkf}	2	-	6	ns
ASDI setup to ABITCLK falling	t_s	10	23	-	ns
ASDI hold after ABITCLK falling	t_h	10	53	-	ns
ASDI input rise/fall time	t_{rfin}	2	-	6	ns
ABITCLK rising to ASDO/ASYNC valid, C_L = 55 pF	t_{co}	2	-	15	ns
ASYNC/ASDO rise time, C_L = 55 pF	t_{rout}	2	-	6	ns
ASYNC/ASDO fall time, C_L = 55 pF	t_{fout}	2	-	6	ns

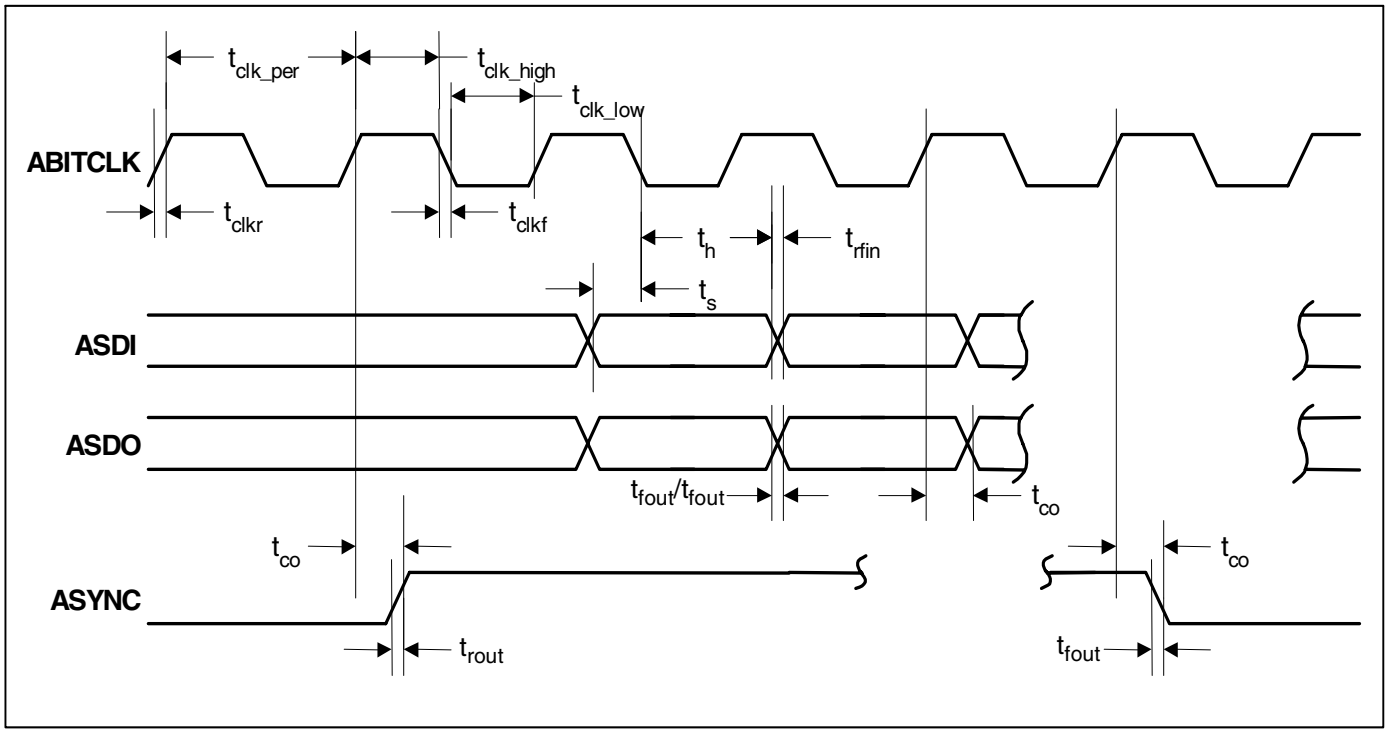


Figure 21. AC '97 Configuration Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

LCD Interface

Parameter	Symbol	Min	Typ	Max	Unit
SPCLK rising time	t_{clkr}	-	5	-	ns
SPCLK falling time	t_{clkf}	-	5	-	ns
SPCLK rising edge to control signal transition time	t_{CD}	-	1	-	ns
SPCLK rising edge to data transition time	t_{DD}	-	0	-	ns
SPCLK falling edge to control signal transition time	t_{CDi}	-	$(t_{SPCLK})/2$	-	ns
SPCLK falling edge to data transition time	t_{DDi}	-	$(t_{SPCLK})/2$	-	ns
Data valid time	t_{Dv}	-	t_{SPCLK}	-	ns

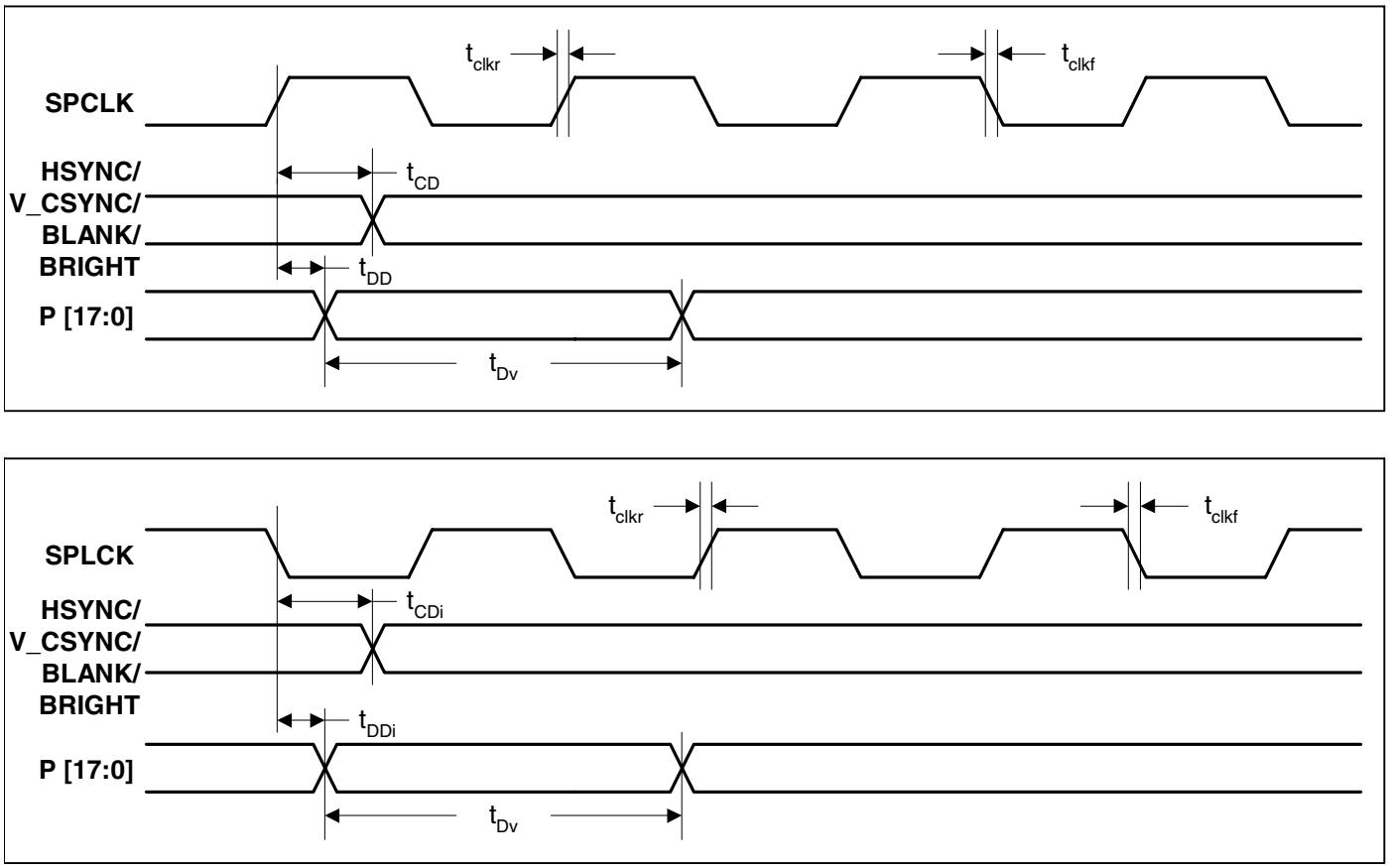


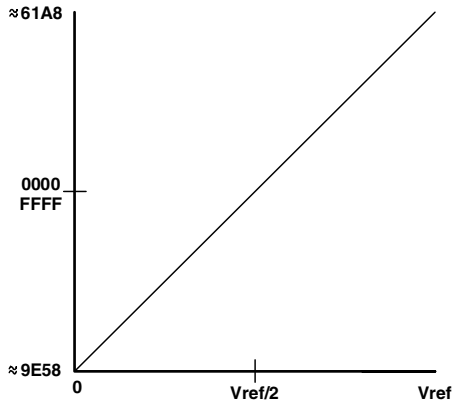
Figure 22. LCD Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

ADC

Parameter	Comment	Value	Units
Resolution	No missing codes Range of 0 to 3.3 V	50K counts (approximate)	
Integral non-linearity		0.01%	
Offset error		±15	mV
Full scale error		0.2%	
Maximum sample rate	ADIV = 0 ADIV = 1	3750 925	Samples per second Samples per second
Channel switch settling time	ADIV = 0 ADIV = 1	500 2	µs ms
Noise (RMS) - typical		120	µV

Note: ADIV refers to bit 16 in the KeyTchClkDiv register.
ADIV = 0 means the input clock to the ADC module is equal to the external 14.7456 MHz clock divided by 4.
ADIV = 1 means the input clock to the ADC module is equal to the external 14.7456 MHz clock divided by 16.



A/D Converter Transfer Function
(approximately ±25,000 counts)

Figure 23. ADC Transfer Function

Using the ADC:

This ADC has a state-machine based conversion engine that automates the conversion process. The initiator for a conversion is the read access of the TSXYResult register by the CPU. The data returned from reading this register contains the result as well as the status bit indicating the state of the ADC. However, this peripheral requires a delay between each successful conversion and the issue of the next conversion command, or else the returned value of successive samples may not reflect the analog input. Since the state of the ADC state machine is returned through the same channel used to initiate the conversion process, there must be a delay inserted after every complete conversion. Note that reading TSXYResult during a conversion will not affect the result of the ongoing process.

The following is a recommended procedure for safely polling the ADC from software:

1. Read the TSXYResult register into a local variable to initiate a conversion.
2. If the value of bit 31 of the local variable is '0', repeat step 1.
3. Delay long enough to meet the maximum sample rate as shown above.
4. Mask the local variable with 0xFFFF to remove extraneous data.
5. If signed mode is used, do a sign extend of the lower halfword.
6. Return the sampled value.

©Copyright 2004 Cirrus Logic (All Rights Reserved)

JTAG

Parameter	Symbol	Min	Max	Units
TCK clock period	t_{clk_per}	100	-	ns
TCK clock high time	t_{clk_high}	50	-	ns
TCK clock low time	t_{clk_low}	50	-	ns
TMS/TDI to clock rising setup time	t_{JPs}	20	-	ns
Clock rising to TMS/TDI hold time	t_{JPh}	45	-	ns
JTAG port clock to output	t_{JPco}	-	30	ns
JTAG port high impedance to valid output	t_{JPzx}	-	30	ns
JTAG port valid output to high impedance	t_{JPxz}	-	30	ns

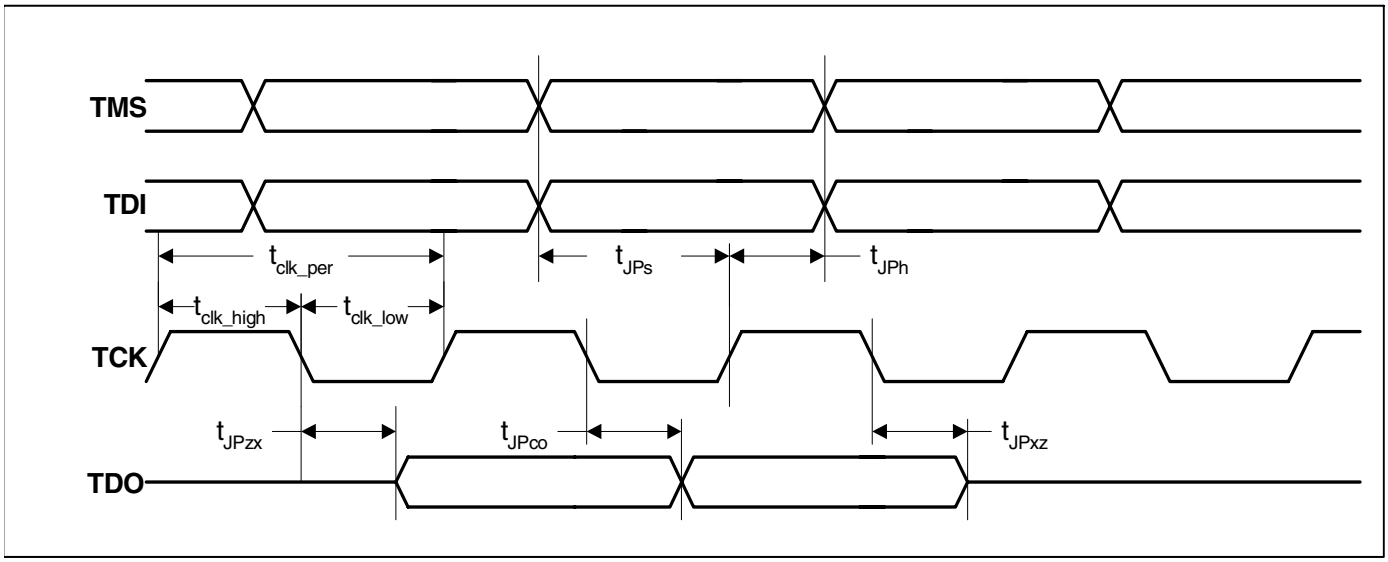


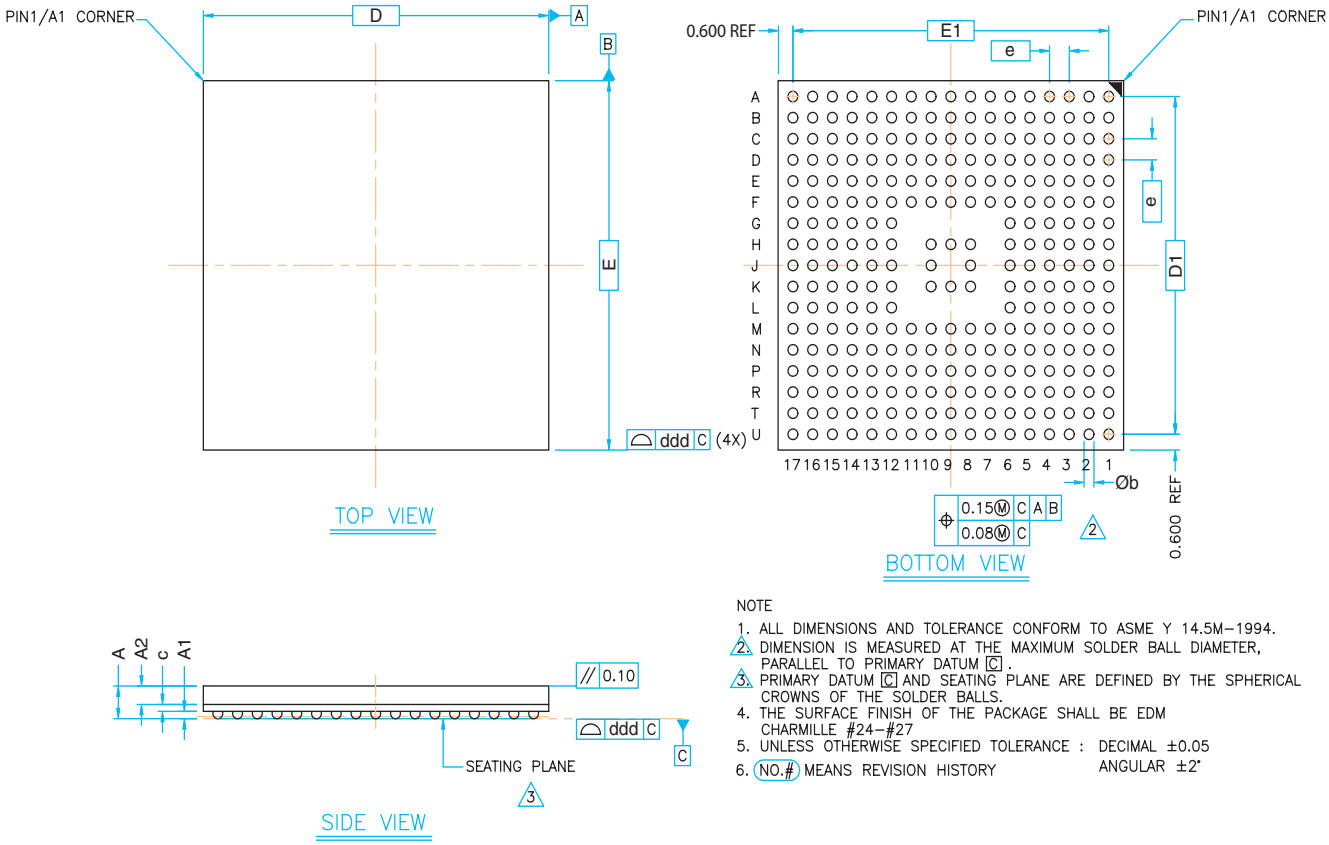
Figure 24. JTAG Timing Measurement

© Copyright 2004 Cirrus Logic (All Rights Reserved)

272 Pin TFBGA Package Outline

272 TFBGA Diagram

Figure 25. 272 Pin TFBGA Diagram



©Copyright 2004 Cirrus Logic (All Rights Reserved)

Table R. 272 Pin Diagram Dimensions

Symbol	dimension in mm			dimension in inches		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.40	1.45	0.053	0.055	0.057
A1	0.23	0.28	0.33	0.009	0.011	0.013
A2	0.65	0.70	0.75	0.026	0.028	0.030
b	0.35	0.40	0.45	0.014	0.016	0.018
c	0.21	0.26	0.31	0.0083	0.0102	0.0122
D	13.95	14.00	14.05	0.549	0.551	0.553
D3	12.75	12.80	12.85	0.502	0.504	0.506
E	13.95	14.00	14.05	0.549	0.551	0.553
E3	12.75	12.80	12.85	0.502	0.504	0.506
e	0.75	0.80	0.85	0.030	0.031	0.033
ddd			0.10			0.004

Note: 1. Controlling Dimension: Millimeter.
2. Primary Datum C and seating plane are defined by the spherical crowns of the solder balls.
3. Dimension b is measured at the maximum solder ball diameter, parallel to Primary Datum C.
4. There shall be a minimum clearance of 0.25 mm between the edge of the solder ball and the body edge.
5. Reference Document: JEDEC MO-151, BAL-2

272 Pin TFBGA Pinout (Bottom View)

The following table shows the 272 pin TFBGA pinout. (For better understanding, compare the coordinates on the x and y axis on Figure 26, "272 Pin TFBGA Pinout", on page 41 with Figure 25, "272 Pin TFBGA Diagram", on page 39.

- VDD_core is vddc.
- VDD_ring is vddr.
- GND_core is gndc.
- GND_ring is gndr.

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Figure 26. 272 Pin TFBGA Pinout

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
U	NC	NC	P[8]	P[4]	P[1]	DA[6]	DA[3]	AD[10]	DA[0]	TDO	NC	SCLK[1]	SSPRX[1]	INT[1]	RTSn	USBm[1]	NC	U
T	NC	NC	V_CSYNC	P[7]	P[2]	DA[7]	AD[11]	AD[9]	DSRn	TMS	gndr	SFRM[1]	INT[2]	INT[0]	USBp[1]	NC	NC	T
R	P[9]	HSYNC	P[6]	P[5]	P[0]	AD[14]	DA[4]	DA[1]	DTRn	TDI	BOOT[0]	ASYNC	SSPTX[1]	PWMOUT	USBm[0]	ABITCLK	USBp[0]	R
P	SPCLK	P[10]	P[11]	P[3]	AD[15]	AD[13]	AD[12]	DA[2]	AD[8]	TCK	BOOT[1]	EEDAT	GRLED	RDLED	GGPIO[2]	RXD[1]	RXD[2]	P
N	P[14]	P[16]	P[15]	P[13]	P[12]	DA[5]	vddr	vddr	vddr	vddr	EECLK	ASDO	CTSn	RXD[0]	TXD[0]	TXD[1]	TXD[2]	N
M	BRIGHT	AD[0]	DQMn[1]	DQMn[2]	P[17]	gndr	gndr	vddc	vddc	gndr	gndr	ROW[6]	ROW[4]	ROW[1]	ROW[0]	ROW[3]	ROW[2]	M
L	DA[9]	AD[2]	AD[1]	DA[8]	BLANK	gndr						gndr	ROW[7]	ROW[5]	PLL_GND	XTALI	XTALO	L
K	AD[4]	DA[12]	DA[10]	DA[11]	vddr	gndr		gndc	gndc	gndc		vddc	COL[4]	PLL_VDD	COL[2]	COL[1]	COL[0]	K
J	AD[6]	DA[14]	AD[7]	DA[13]	vddr	vddc		gndc		gndc		vddc	vddr	COL[5]	COL[6]	CSn[0]	COL[3]	J
H	DA[18]	DA[20]	DA[19]	DA[16]	vddr	vddc		gndc	gndc	gndc		gndr	vddr	EGPIO[8]	PRSTn	COL[7]	RSTOn	H
G	DQMn[0]	CASn	DA[21]	AD[22]	vddr	gndr						gndr	EGPIO[9]	EGPIO[10]	EGPIO[11]	RTCXTALO	RTCXTALI	G
F	RASn	SDCSn[1]	SDCSn[0]	DQMn[3]	AD[5]	gndr	gndr	gndr	vddc	vddc	gndr	EGPIO[7]	EGPIO[5]	ADC_GND	EGPIO[6]	sYm	sYp	F
E	SDCSn[2]	SDWEN	DA[22]	AD[3]	DA[15]	AD[21]	DA[17]	vddr	vddr	vddr	MIIRXD[0]	TXERR	EGPIO[2]	EGPIO[4]	EGPIO[3]	sXp	sXm	E
D	SDCSn[3]	DA[23]	SDCLK	DA[24]	HGPIO[7]	HGPIO[6]	DA[28]	HGPIO[4]	AD[16]	MDC	RXERR	MIITXD[3]	EGPIO[12]	EGPIO[1]	EGPIO[0]	Ym	Yp	D
C	AD[23]	DA[26]	CSn[3]	DA[25]	AD[24]	AD[19]	HGPIO[5]	WRn	MDIO	MIIRXD[2]	TXCLK	MIITXD[0]	CLD	EGPIO[13]	TRSTn	Xp	Xm	C
B	AD[25]	CSn[2]	CSn[6]	AD[20]	DA[30]	AD[18]	HGPIO[3]	AD[17]	RXCLK	MIIRXD[1]	MIITXD[2]	TXEN	FGPIO[5]	EGPIO[15]	USBp[2]	ARSTn	ADC_VDD	B
A	CSn[1]	CSn[7]	SDCLKEN	DA[31]	DA[29]	DA[27]	HGPIO[2]	RDn	MIIRXD[3]	RXDVAL	MIITXD[1]	CRS	FGPIO[7]	FGPIO[0]	WAITn	USBm[2]	ASDI	A
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Pin List

The following Thin-profile Fine-pitch Ball Grid Array (TFBGA) ball assignment table is sorted in order of ball.

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
A1	CSn[1]	E1	SDCSn[2]	J10	gndc	P1	SPCLK
A2	CSn[7]	E2	SDWEN	J12	vddc	P2	P[10]
A3	SDCLKEN	E3	DA[22]	J13	vddr	P3	P[11]
A4	DA[31]	E4	AD[3]	J14	COL[5]	P4	P[3]
A5	DA[29]	E5	DA[15]	J15	COL[6]	P5	AD[15]
A6	DA[27]	E6	AD[21]	J16	CSn[0]	P6	AD[13]
A7	HGPIO[2]	E7	DA[17]	J17	COL[3]	P7	AD[12]
A8	RDn	E8	vddr	K1	AD[4]	P8	DA[2]
A9	MIIRXD[3]	E9	vddr	K2	DA[12]	P9	AD[8]
A10	RXDVAL	E10	vddr	K3	DA[10]	P10	TCK
A11	MIITXD[1]	E11	MIIRXD[0]	K4	DA[11]	P11	BOOT[1]
A12	CRS	E12	TXERR	K5	vddr	P12	EEDAT
A13	FGPIO[7]	E13	EGPIO[2]	K6	gndr	P13	GRLED
A14	FGPIO[0]	E14	EGPIO[4]	K8	gndc	P14	RDLED
A15	WAITn	E15	EGPIO[3]	K9	gndc	P15	GGPIO[2]
A16	USBm[2]	E16	sXp	K10	gndc	P16	RXD[1]
A17	ASDI	E17	sXm	K12	vddc	P17	RXD[2]
B1	AD[25]	F1	RASn	K13	COL[4]	R1	P[9]
B2	CSn[2]	F2	SDCSn[1]	K14	PLL_VDD	R2	HSYNC
B3	CSn[6]	F3	SDCSn[0]	K15	COL[2]	R3	P[6]
B4	AD[20]	F4	DQMn[3]	K16	COL[1]	R4	P[5]
B5	DA[30]	F5	AD[5]	K17	COL[0]	R5	P[0]
B6	AD[18]	F6	gndr	L1	DA[9]	R6	AD[14]
B7	HGPIO[3]	F7	gndr	L2	AD[2]	R7	DA[4]
B8	AD[17]	F8	gndr	L3	AD[1]	R8	DA[1]
B9	RXCLK	F9	vddc	L4	DA[8]	R9	DTRn
B10	MIIRXD[1]	F10	vddc	L5	BLANK	R10	TDI
B11	MIITXD[2]	F11	gndr	L6	gndr	R11	BOOT[0]
B12	TXEN	F12	EGPIO[7]	L12	gndr	R12	ASYNC
B13	FGPIO[5]	F13	EGPIO[5]	L13	ROW[7]	R13	SSPTX[1]
B14	EGPIO[15]	F14	ADC_GND	L14	ROW[5]	R14	PWMOUT
B15	USBp[2]	F15	EGPIO[6]	L15	PLL_GND	R15	USBm[0]
B16	ARSTn	F16	sYm	L16	XTALI	R16	ABITCLK
B17	ADC_VDD	F17	sYp	L17	XTALO	R17	USBp[0]
C1	AD[23]	G1	DQMn[0]	M1	BRIGHT	T1	NC
C2	DA[26]	G2	CASn	M2	AD[0]	T2	NC
C3	CSn[3]	G3	DA[21]	M3	DQMn[1]	T3	V_CSYNC
C4	DA[25]	G4	AD[22]	M4	DQMn[2]	T4	P[7]
C5	AD[24]	G5	vddr	M5	P[17]	T5	P[2]
C6	AD[19]	G6	gndr	M6	gndr	T6	DA[7]
C7	HGPIO[5]	G12	gndr	M7	gndr	T7	AD[11]
C8	WRn	G13	EGPIO[9]	M8	vddc	T8	AD[9]

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
C9	MDIO	G14	EGPIO[10]	M9	vddc	T9	DSRn
C10	MIIRXD[2]	G15	EGPIO[11]	M10	gndr	T10	TMS
C11	TXCLK	G16	RTCXTALO	M11	gndr	T11	gndr
C12	MIITXD[0]	G17	RTCXTALI	M12	ROW[6]	T12	SFRM[1]
C13	CLD	H1	DA[18]	M13	ROW[4]	T13	INT[2]
C14	EGPIO[13]	H2	DA[20]	M14	ROW[1]	T14	INT[0]
C15	TRSTn	H3	DA[19]	M15	ROW[0]	T15	USBp[1]
C16	Xp	H4	DA[16]	M16	ROW[3]	T16	NC
C17	Xm	H5	vddr	M17	ROW[2]	T17	NC
D1	SDCSn[3]	H6	vddc	N1	P[14]	U1	NC
D2	DA[23]	H8	gndc	N2	P[16]	U2	NC
D3	SDCLK	H9	gndc	N3	P[15]	U3	P[8]
D4	DA[24]	H10	gndc	N4	P[13]	U4	P[4]
D5	HGPIO[7]	H12	gndr	N5	P[12]	U5	P[1]
D6	HGPIO[6]	H13	vddr	N6	DA[5]	U6	DA[6]
D7	DA[28]	H14	EGPIO[8]	N7	vddr	U7	DA[3]
D8	HGPIO[4]	H15	PRSTn	N8	vddr	U8	AD[10]
D9	AD[16]	H16	COL[7]	N9	vddr	U9	DA[0]
D10	MDC	H17	RSTOn	N10	vddr	U10	TDO
D11	RXERR	J1	AD[6]	N11	EECLK	U11	NC
D12	MIITXD[3]	J2	DA[14]	N12	ASDO	U12	SCLK[1]
D13	EGPIO[12]	J3	AD[7]	N13	CTSn	U13	SSPRX[1]
D14	EGPIO[1]	J4	DA[13]	N14	RXD[0]	U14	INT[1]
D15	EGPIO[0]	J5	vddr	N15	TXD[0]	U15	RTSn
D16	Ym	J6	vddc	N16	TXD[1]	U16	USBm[1]
D17	Yp	J8	gndc	N17	TXD[2]	U17	NC

© Copyright 2004 Cirrus Logic (All Rights Reserved)

The following section focuses on the EP9307 pin signals from two viewpoints - the pin usage and pad characteristics, and the pin multiplexing usage. The first table (Table S) is a summary of all the EP9307 pin signals. The second table (Table T) illustrates the pin signal multiplexing and configuration options.

Table S is a summary of the EP9307 pin signals, which illustrates the pad type and pad pull type (if any). The symbols used in the table are defined as follows. (Note: A blank box means Not Applicable (NA) or, for Pull Type, No Pull (NP).)

Under the Pad Type column:

- A - Analog pad
- P - Power pad
- G - Ground pad
- I - Pin is an input only
- I/O - Pin is input/output
- 4mA - Pin is a 4mA output driver
- 8mA - Pin is an 8mA output driver
- 12mA - Pin is an 12mA output driver

See the text description for additional information about bi-directional pins.

Under the Pull Type Column:

- PU - Resistor is a pull up to the RVDD supply
- PD - Resistor is a pull down to the RGND supply

.

Table S. Pin Descriptions

Pin Name	Block	Pad Type	Pull Type	Description
TCK	JTAG	I	PD	JTAG clock in
TDI	JTAG	I	PD	JTAG data in
TDO	JTAG	4ma	-	JTAG data out
TMS	JTAG	I	PD	JTAG test mode select
TRSTn	JTAG	I	PD	JTAG reset
BOOT[1:0]	System	I	PD	Boot mode select in
XTALI	PLL	A	-	Main oscillator input
XTALO	PLL	A	-	Main oscillator output
VDD_PLL	PLL	P	-	Main oscillator power, 1.8V
GND_PLL	PLL	G	-	Main oscillator ground
RTCXTALI	RTC	A	-	RTC oscillator input
RTCXTALO	RTC	A	-	RTC oscillator output
WRn	EBUS	4ma	-	SRAM Write strobe out
RDn	EBUS	4ma	-	SRAM Read/OE strobe out
WAITn	EBUS	I	PU	SRAM Wait in
AD[25:0]	EBUS	8ma	-	Shared Address bus out
DA[31:0]	EBUS	8ma	PU	Shared Data bus in/out
CSn[3:0]	EBUS	4ma	PU	Chip select out
CSn[7:6]	EBUS	4ma	PU	Chip select out
DQMn[3:0]	EBUS	8ma	-	Shared data mask out
SDCLK	SDRAM	8ma	-	SDRAM clock out
SDCLKEN	SDRAM	8ma	-	SDRAM clock enable out
SDCSn[3:0]	SDRAM	4ma	-	SDRAM chip selects out
RASn	SDRAM	8ma	-	SDRAM RAS out
CASn	SDRAM	8ma	-	SDRAM CAS out
SDWEn	SDRAM	8ma	-	SDRAM write enable out
P[17:0]	Raster	4ma	PU	Pixel data bus out
SPCLK	Raster	12ma	PU	Pixel clock in/out
HSYNC	Raster	8ma	PU	Horizontal synchronization/ line pulse out

Table S. Pin Descriptions *(Continued)*

Pin Name	Block	Pad Type	Pull Type	Description
V_CSYNC	Raster	8ma	PU	Vertical or composite synchronization/frame pulse out
BLANK	Raster	8ma	PU	Composite blanking signal out
BRIGHT	Raster	4ma	-	PWM brightness control out
PWMOUT	PWM	8ma		Pulse width modulator output
Xp, Xm	ADC	A	-	Touchscreen ADC X axis
Yp, Ym	ADC	A	-	Touchscreen ADC Y axis
sXp, sXm	ADC	A	-	Touchscreen ADC X axis feedback
sYp, sYm	ADC	A	-	Touchscreen ADC Y axis feedback
VDD_ADC	ADC	P	-	Touchscreen ADC power, 3.3V
GND_ADC	ADC	G	-	Touchscreen ADC ground
COL[7:0]	Key	8ma	PU	Key matrix column inputs
ROW[7:0]	Key	8ma	PU	Key matrix row outputs
USBp[2:0]	USB	A	-	USB positive signals
USBm[2:0]	USB	A	-	USB negative signals
TXD0	UART1	4ma	-	Transmit out
RXD0	UART1	I	PU	Receive in
CTSn	UART1	I	PU	Clear to send/transmit enable
DSRn	UART1	I	PU	Data set ready/Data Carrier Detect
DTRn	UART1	4ma	-	Data Terminal Ready output
RTSn	UART1	4ma	-	Ready to send
TXD1	UART2	4ma	-	Transmit/IrDA output
RXD1	UART2	I	PU	Receive/IrDA input
TXD2	UART3	4ma	-	Transmit
RXD2	UART3	I	PU	Receive
MDC	EMAC	4ma		Management data clock
MDIO	EMAC	4ma	PU	Management data input/output
RXCLK	EMAC	I	PD	Receive clock in
MIIRXD[3:0]	EMAC	I	PD	Receive data in
RXDVAL	EMAC	I	PD	Receive data valid

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Table S. Pin Descriptions *(Continued)*

Pin Name	Block	Pad Type	Pull Type	Description
RXERR	EMAC	I	PD	Receive data error
TXCLK	EMAC	4ma	PU	Transmit clock in
MIITXD[3:0]	EMAC	I	PD	Transmit data out
TXEN	EMAC	4ma	PD	Transmit enable
TXERR	EMAC	4ma	PD	Transmit error
CRS	EMAC	I	PD	Carrier sense
CLD	EMAC	I	PU	Collision detect
GRLED	LED	12ma	-	Green LED
RDLED	LED	12ma	-	Red LED
EECLK	EEPROM	4ma	PU	EEPROM/Two-wire Interface clock
EEDAT	EEPROM	4ma	PU	EEPROM/Two-wire Interface data
ABITCLK	AC97	8ma	PD	AC97 bit clock
ASYNC	AC97	8ma	PD	AC97 frame sync
ASDI	AC97	I	PD	AC97 Primary input
ASDO	AC97	8ma	PU	AC97 output
ARSTn	AC97	8ma	-	AC97 reset
SCLK1	SPI1	8ma	PD	SPI bit clock
SFRM1	SPI1	8ma	PD	SPI Frame Clock
SSPRX1	SPI1	I	PD	SPI input
SSPTX1	SPI1	8ma	-	SPI output
INT[2:0]	INT	I	PD	External interrupts
PRSTn	Syscon	I	PU	Power on reset
RSTOn	Syscon	4ma	-	User Reset in out - open drain
EGPIO[15]	GPIO	I/O, 4ma	PU	Enhanced GPIO
EGPIO[13:0]	GPIO	I/O, 4ma	PU	Enhanced GPIO
FGPIO[7, 5, 0]	GPIO	I/O, 8ma	PU	GPIO
GGPIO[2]	GPIO	I/O, 8ma	PU	GPIO
HGPIO[7:2]	GPIO	I/O, 8ma	PU	GPIO
vddc	Power	P	-	Digital power, 1.8V
vddr	Power	P	-	Digital power, 3.3V
gndc	Ground	G	-	Digital ground
gndr	Ground	G	-	Digital ground

©Copyright 2004 Cirrus Logic (All Rights Reserved)

Table T illustrates the pin signal multiplexing and configuration options.

Table T. Pin Multiplex Usage Information

Physical Pin Name	Description	Multiplex signal name
COL[7:0]	GPIO	GPIO Port D[7:0]
ROW[7:0]	GPIO	GPIO Port C[7:0]
EGPIO[0]	Ring Indicator Input	RI
EGPIO[1]	1Hz clock monitor	CLK1HZ
EGPIO[2]	DMA request	DMARQ
EGPIO[3]	HDLC Clock	HDLCCLK1
EGPIO[4]	I2S Transmit Data 1	SDO1
EGPIO[5]	I2S Receive Data 1	SDI1
EGPIO[6]	I2S Transmit Data 2	SDO2
EGPIO[7]	DMA Request 0	DREQ0
EGPIO[8]	DMA Acknowledge 0	DACK0
EGPIO[9]	DMA EOT 0	DEOT0
EGPIO[10]	DMA Request 1	DREQ1
EGPIO[11]	DMA Acknowledge 1	DACK1
EGPIO[12]	DMA EOT 1	DEOT1
EGPIO[13]	I2S Receive Data 2	SDI2
EGPIO[15]	Device active / present	DASP
ABITCLK	I2S Serial clock	SCLK
ASYNC	I2S Frame Clock	LRCK
ASDO	I2S Transmit Data 0	SDO0
ASDI	I2S Receive Data 0	SDI0
ARSTn	I2S Master clock	MCLK
SCLK1	I2S Serial clock	SCLK
SFRM1	I2S Frame Clock	LRCK
SSPTX1	I2S Transmit Data 0	SDO0
SSPRX1	I2S Receive Data 0	SDI0

© Copyright 2004 Cirrus Logic (All Rights Reserved)

Acronyms and Abbreviations

The following tables list abbreviations and acronyms used in this data sheet.

Term	Definition
ADC	Analog-to-Digital Converter
ALT	Alternative
AMBA	Advanced Micro-controller Bus Architecture
ATAPI	ATA Packet Interface
CODEC	COder/DECoder
CRC	Cyclic Redundancy Check
DAC	Digital-to-Analog Converter
DMA	Direct-Memory Access
EEPROM	Electronically Erasable Programmable Read Only Memory
EMAC	Ethernet Media Access Controller
EBUS	External Bus
FIFO	First In/First Out
FIQ	Fast Interrupt Request
FLASH	Flash memory
GPIO	General Purpose I/O
HDLC	High-level Data Link Control
I/F	Interface
I^2S	Inter-IC Sound
IC	Integrated Circuit
ICE	In-Circuit Emulator
IDE	Integrated Drive Electronics
IEEE	Institute of Electronics and Electrical Engineers
IrDA	Infrared Data Association
IRQ	Standard Interrupt Request
ISO	International Standards Organization
JTAG	Joint Test Action Group
LFSR	Linear Feedback Shift Register
MII	Media Independent Interface
MMU	Memory Management Unit
OHCI	Open Host Controller Interface
PHY	Ethernet PHYsical layer interface
PIO	Programmed I/O
RISC	Reduced Instruction Set Computer
SDMI	Secure Digital Music Initiative
SDRAM	Synchronous Dynamic RAM
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
STA	Station - Any device that contains an IEEE 802.11 conforming Medium Access Control (MAC) and physical layer (PHY) interface to the wireless medium
TFT	Thin Film Transistor
TLB	Translation Lookaside Buffer
USB	Universal Serial Bus

Units of Measurement

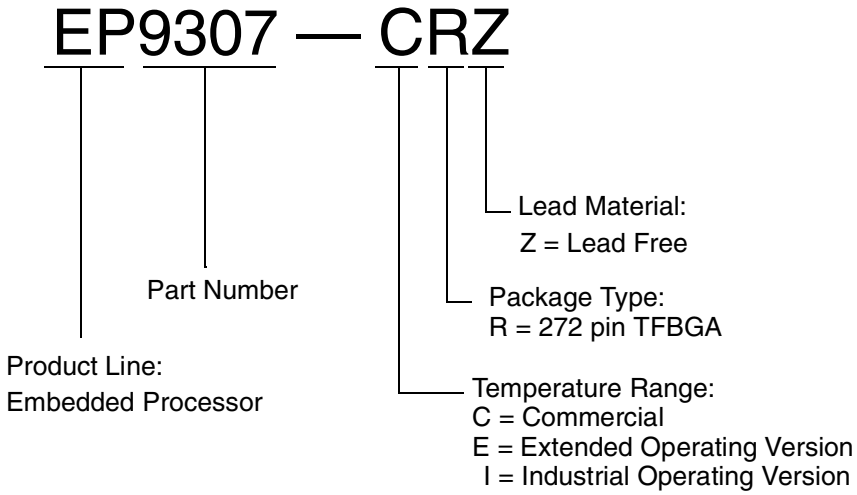
Symbol	Unit of Measure
^{o}C	degree Celsius
Hz	Hertz = cycle per second
Kbps	Kilobits per second
Kbyte	Kilobyte
KHz	KiloHertz = 1000 Hz
Mbps	Megabits per second
MHz	MegaHertz = 1,000 KiloHertz
µA	microAmpere = 10^{-6} Ampere
µs	microsecond = 1,000 nanoseconds = 10^{-6} seconds
mA	milliAmpere = 10^{-3} Ampere
ms	millisecond = 1,000 microseconds = 10^{-3} seconds
mW	milliWatt = 10^{-3} Watts
ns	nanosecond = 10^{-9} seconds
pF	picoFarad = 10^{-12} Farads
V	Volt
W	Watt

©Copyright 2004 Cirrus Logic (All Rights Reserved)

ORDERING INFORMATION

The order numbers for the device are:

EP9307-CR	0°C to +70°C	272 pin TFBGA	
EP9307-CRZ	0°C to +70°C	272 pin TFBGA	Lead Free
EP9307-IR	-40°C to +85°C	272 pin TFBGA	
EP9307-IRZ	-40°C to +85°C	272 pin TFBGA	Lead Free



Note: Go to the Cirrus Logic Internet site at http://www.cirrus.com to find contact information for your local sales representative.

Contacting Cirrus Logic Support

For all product questions and inquiries contact a Cirrus Logic Sales Representative.
To find one nearest you go to **www.cirrus.com**

IMPORTANT NOTICE

"Preliminary" product information describes products that are in production, but for which full characterization data is not yet available. Cirrus Logic, Inc. and its subsidiaries ("Cirrus") believe that the information contained in this document is accurate and reliable. However, the information is subject to change without notice and is provided "AS IS" without warranty of any kind (express or implied). Customers are advised to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability. No responsibility is assumed by Cirrus for the use of this information, including use of this information as the basis for manufacture or sale of any items, or for infringement of patents or other rights of third parties. This document is the property of Cirrus and by furnishing this information, Cirrus grants no license, express or implied under any patents, mask work rights, copyrights, trademarks, trade secrets or other intellectual property rights. Cirrus owns the copyrights associated with the information contained herein and gives consent for copies to be made of the information only for use within your organization with respect to Cirrus integrated circuits or other products of Cirrus. This consent does not extend to other copying such as copying for general distribution, advertising or promotional purposes, or for creating any work for resale.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). CIRRUS PRODUCTS ARE NOT DESIGNED, AUTHORIZED OR WARRANTED FOR USE IN AIRCRAFT SYSTEMS, MILITARY APPLICATIONS, PRODUCTS SURGICALLY IMPLANTED INTO THE BODY, LIFE SUPPORT PRODUCTS OR OTHER CRITICAL APPLICATIONS (INCLUDING MEDICAL DEVICES, AIRCRAFT SYSTEMS OR COMPONENTS AND PERSONAL OR AUTOMOTIVE SAFETY OR SECURITY DEVICES). INCLUSION OF CIRRUS PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK AND CIRRUS DISCLAIMS AND MAKES NO WARRANTY, EXPRESS, STATUTORY OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE, WITH REGARD TO ANY CIRRUS PRODUCT THAT IS USED IN SUCH A MANNER. IF THE CUSTOMER OR CUSTOMER'S CUSTOMER USES OR PERMITS THE USE OF CIRRUS PRODUCTS IN CRITICAL APPLICATIONS, CUSTOMER AGREES, BY SUCH USE, TO FULLY INDEMNIFY CIRRUS, ITS OFFICERS, DIRECTORS, EMPLOYEES, DISTRIBUTORS AND OTHER AGENTS FROM ANY AND ALL LIABILITY, INCLUDING ATTORNEYS' FEES AND COSTS, THAT MAY RESULT FROM OR ARISE IN CONNECTION WITH THESE USES.

Cirrus Logic, Cirrus, MaverickCrunch, MaverickKey, and the Cirrus Logic logo designs are trademarks of Cirrus Logic, Inc. All other brand and product names in this document may be trademarks or service marks of their respective owners.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

Microwire is a trademark of National Semiconductor Corp. National Semiconductor is a registered trademark of National Semiconductor Corp.

Texas Instruments is a registered trademark of Texas Instruments, Inc.

Motorola is a registered trademark of Motorola, Inc.

LINUX is a registered trademark of Linus Torvalds.

 © Copyright 2004 Cirrus Logic (All Rights Reserved)