Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	ARM920T
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	200MHz
Co-Processors/DSP	Math Engine; MaverickCrunch™
RAM Controllers	SDRAM
Graphics Acceleration	Yes
Display & Interface Controllers	Keypad, LCD, Touchscreen
Ethernet	1/10/100Mbps (1)
SATA	-
USB	USB 2.0 (3)
Voltage - I/O	1.8V, 3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	Hardware ID
Package / Case	352-BBGA
Supplier Device Package	352-PBGA (27x27)
Purchase URL	https://www.e-xfl.com/product-detail/cirrus-logic/ep9312-ibz

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 Microcontroller 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 1 Mbyte, 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 are programmed into the EP9312 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 EP9312 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 EP9312 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 to 4 banks of 32-bit, 66- or 100-MHz SDRAM
- One internal port dedicated to the Raster/LCD Refresh Engine (Read Only)
- Address and data bus shared between SDRAM, SRAM, ROM, and FLASH memory
- 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

Table M. 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 ensures 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 64 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. External 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
- Four dedicated off-chip interrupt lines INT[3:0] operate as level-sensitive interrupts
- Any of the 16 GPIO lines maybe configured to generate interrupts
- Software-supported priority mask for all FIQs and IRQs

Table N. External Interrupt Controller Pin Assignment

Pin Mnemonic	Pin Name - Description
INT[3:0]	External Interrupt 3-0

Dual LED Drivers

Two pins are assigned specifically to drive external LEDs.

Table O. 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 16 EGPIO pins may each be configured individually as an output, an input, or an interrupt input.

There are 23 pins that may alternatively be used as input, output, but do not support interrupts. These pins are:

- Key Matrix ROW[7:0], COL[7:0]
- Ethernet MDIO
- Both LED Outputs
- Two-wire Clock and Data
- SLA [1:0]

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

Table P. General Purpose Input/Output Pin Assignment

Pin Mnemonic	Pin Name - Description
EGPIO[15:0]	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 Q. 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

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.

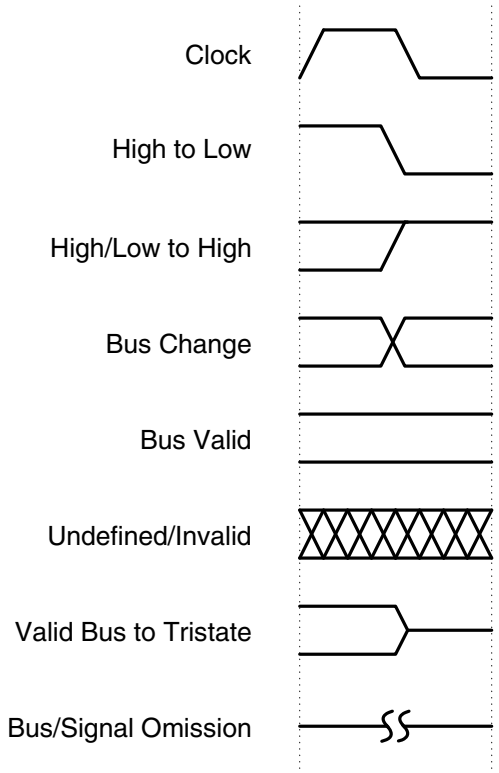


Figure 1. Timing Diagram Drawing Key

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).

SDRAM Burst Write Cycle

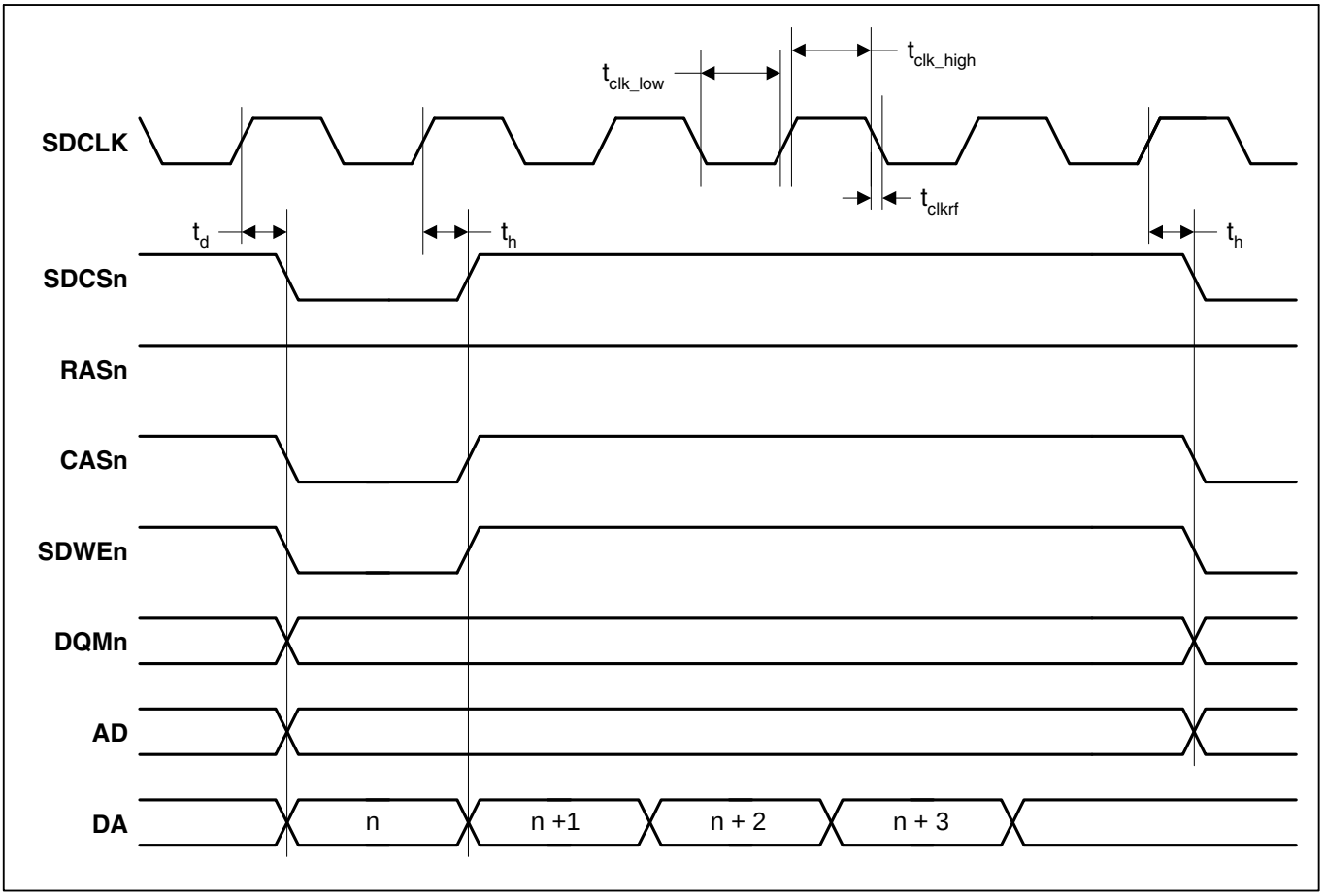


Figure 4. SDRAM Burst Write Cycle Timing Measurement

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

Parameter	Symbol	Min	Typ	Max	Unit
AD setup to CSn assert time	t_{ADs}	t_{HCLK}	-	-	ns
CSn assert to Address transition time	t_{AD1}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
Address assert time	t_{AD2}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
AD transition to CSn deassert time	t_{AD3}	-	$t_{HCLK} \times (WST1 + 2)$	-	ns
AD hold from CSn deassert time	t_{ADh}	t_{HCLK}	-	-	ns
RDn assert time	t_{RDpwL}	-	$t_{HCLK} \times (4 \times WST1 + 5)$	-	ns
CSn to RDn delay time	t_{RDd}	-	-	3	ns
CSn assert to DQMn assert delay time	t_{DQMd}	-	-	1	ns
DA setup to AD transition time	t_{DAs1}	15	-	-	ns
DA setup to RDn deassert time	t_{DAs2}	$t_{HCLK} + 12$	-	-	ns
DA hold from AD transition time	t_{DAh1}	0	-	-	ns
DA hold from RDn deassert time	t_{DAh2}	0	-	-	ns

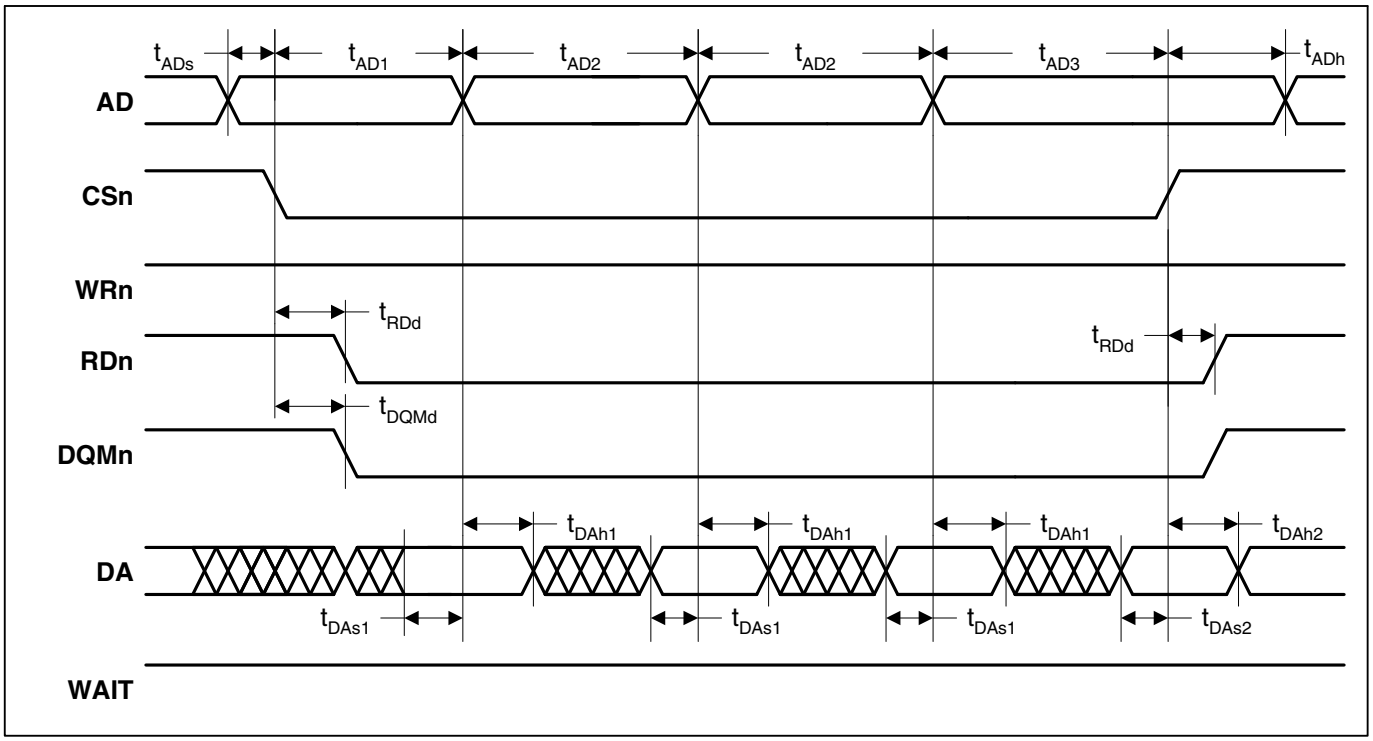


Figure 8. Static Memory Multiple Word Read 8-bit Cycle Timing Measurement

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

Parameter	Symbol	Min	Typ	Max	Unit
AD setup to WRn assert time	t_{ADs}	$t_{HCLK} - 3$	-	-	ns
WRn/DQMn deassert to AD transition time	t_{ADd}	-	-	$t_{HCLK} + 6$	ns
AD hold from WRn deassert time	t_{ADh}	$t_{HCLK} \times 2$	-	-	ns
CSn hold from WRn deassert time	t_{CSh}	7	-	-	ns
CSn to WRn assert delay time	t_{WRd}	-	-	2	ns
WRn assert time	t_{WRpwL}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
WRn deassert time	t_{WRpwH}	-	$t_{HCLK} \times 2$	$(t_{HCLK} \times 2) + 14$	ns
CSn to DQMn assert delay time	t_{DQMd}	-	-	1	ns
DQMn assert time	t_{DQMpwL}	-	$t_{HCLK} \times (WST1 + 1)$	-	ns
DQMn deassert time	t_{DQMpwH}	-	-	$(t_{HCLK} \times 2) + 7$	ns
WRn / DQMn deassert to DA transition time	t_{DAh}	t_{HCLK}	-	-	ns
WRn / DQMn assert to DA valid time	t_{DAV}	-	-	8	ns

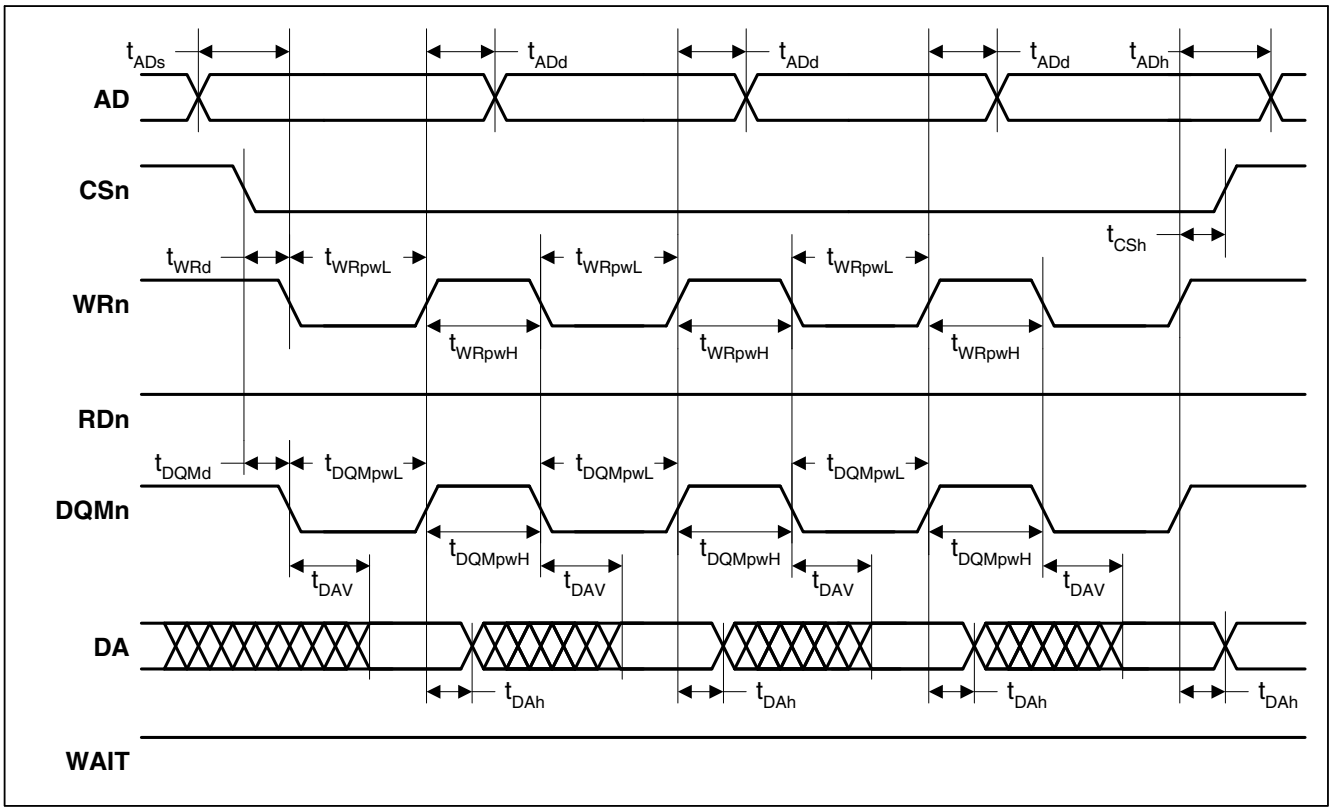


Figure 9. Static Memory Multiple Word Write 8-bit Cycle Timing Measurement

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

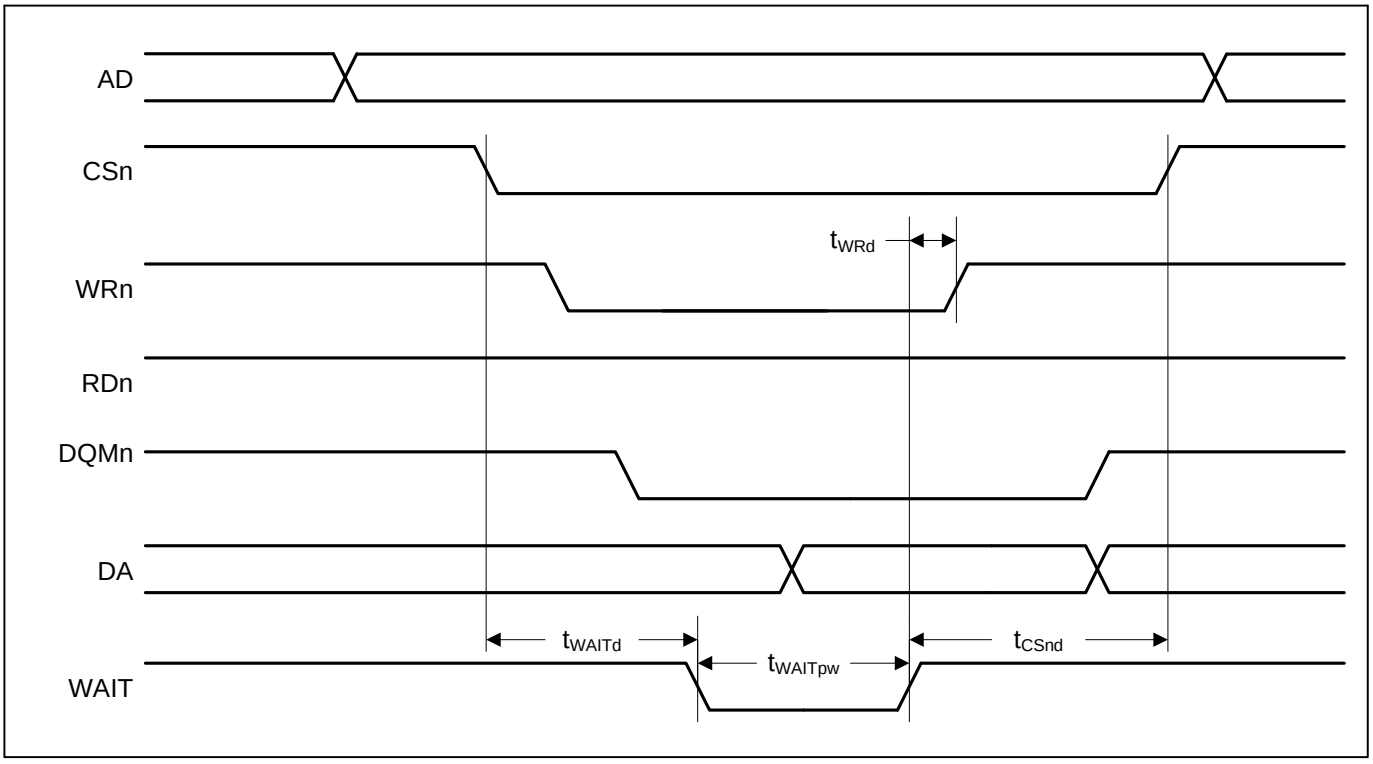


Figure 15. Static Memory Single Write Wait Cycle Timing Measurement

IDE Interface

Register Transfers

Parameter			Symbol	Mode 0 (in ns)	Mode 1 (in ns)	Mode 2 (in ns)	Mode 3 (in ns)	Mode 4 (in ns)
Cycle time	(min)	(Notes 1, 4, 5)	t_0	600	383	330	180	120
Address valid to DIORn / DIOWn setup	(min)	(Note 4)	t_1	70	50	30	30	25
DIORn / DIOWn pulse width 8-bit	(min)	(Note 1, 4)	t_2	290	290	290	80	70
DIORn / DIOWn recovery time	(min)	(Note 1, 4)	t_{2i}	-	-	-	70	25
DIOWn data setup	(min)	(Note 4)	t_3	60	45	30	30	20
DIOWn data hold	(min)		t_4	0	0	0	0	0
DIORn data setup	(min)		t_5	20	20	20	20	20
DIORn data hold	(min)		t_6	0	0	0	0	0
DIORn data high impedance state	(max)	(Note 2, 4)	t_{6z}	30	30	30	30	30
DIORn / DIOWn to address valid hold	(min)	(Note 4)	t_9	20	15	10	10	10
Read Data Valid to IORDY active (if IORDY initially low after t_A)	(min)	(Note 4)	t_{RD}	0	0	0	0	0
IORDY Setup time		(Note 3, 4)	t_A	35	35	35	35	35
IORDY Pulse Width	(max)	(Note 4)	t_B	1250	1250	1250	1250	1250
IORDY assertion to release	(max)		t_C	5	5	5	5	5
DIOWn assert to data valid	(max)		t_{DDV}	10	10	10	10	10

Note: 1. t_0 is the minimum total cycle time, t_2 is the minimum DIORn / DIOWn assertion time, and t_{2i} is the minimum DIORn / DIOWn negation time. A host implementation shall lengthen t_2 and/or t_{2i} to ensure that t_0 is equal to or greater than the value reported in the devices IDENTIFY DEVICE data. A device implementation shall support any legal host implementation.

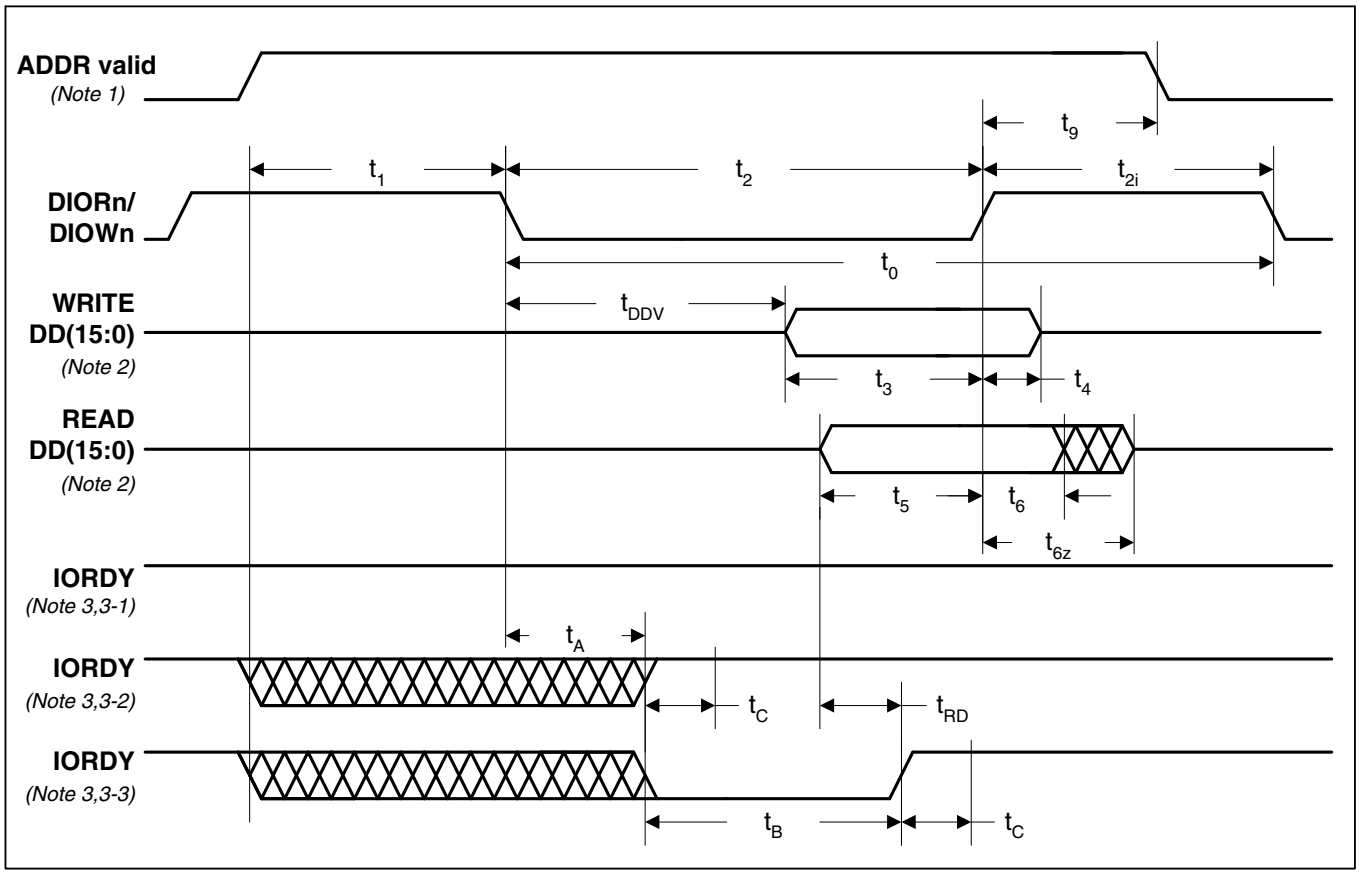
2. This parameter specifies the time from the negation edge of DIORn to the time that the data bus is released by the device.

3. The delay from the activation of DIORn or DIOWn until the state of IORDY is first sampled. If IORDY is inactive then the host shall wait until IORDY is active before the register transfer cycle is completed. If the device is not driving IORDY negated at the t_A after the activation of DIORn or DIOWn, then t_5 shall be met and t_{RD} is not applicable. If the device is driving IORDY negated at the time t_A after the activation of DIORn or DIOWn, then t_{RD} shall be met and t_5 is not applicable.

4. Timings based upon software control. See User's Guide.

5. ATA / ATAPI standards prior to ATA / ATAPI-5 inadvertently specified an incorrect value for mode 2 time t_0 by utilizing the 16-bit PIO value.

6. All IDE timing is based upon HCLK = 100 MHz.



Note: 1. Device address consists of signals IDECS0n, IDECS1n and IDEDA (2:0)

2. Data consists of DD (7:0)

3. The negation of IORDY by the device is used to extend the register transfer cycle. The determination of whether the cycle is to be extended is made by the host after t_A from the assertion of DIORn or DIOWn. The assertion and negation or IORDY are described in the following three cases:

3-1 Device never negates IORDY, devices keeps IORDY released: no wait is generated.

3-2 Device negates IORDY before t_A, but causes IORDY to be asserted before t_A. IORDY is released prior to negation and may be asserted for no more than t_C before release: no wait generated.

3-3 Device negates IORDY before t_A. IORDY is released prior to negation and may be asserted for no more than t_C before release: wait generated. The cycle completes after IORDY is reasserted. For cycles where a wait is generated and DIORn is asserted, the device shall place read data on DD (7:0) for t_{RD} before asserting IORDY.

Figure 17. Register Transfer to/from Device

PIO Data Transfers

Parameter			Symbol	Mode 0 (in ns)	Mode 1 (in ns)	Mode 2 (in ns)	Mode 3 (in ns)	Mode 4 (in ns)
Cycle time	(min)	(Note 1, 4)	t_0	600	383	240	180	120
Address valid to DIORn / DIOWn setup	(min)	(Note 4)	t_1	70	50	30	30	25
DIORn / DIOWn 16-bit	(min)	(Note 1, 4)	t_2	165	125	100	80	70
DIORn / DIOWn recovery time	(min)	(Note 1, 4)	t_{2i}	-	-	-	70	25
DIOWn data setup	(min)	(Note 4)	t_3	60	45	30	30	20
DIOWn data hold	(min)		t_4	0	0	0	0	0
DIORn data setup	(min)		t_5	20	20	20	20	20
DIORn data hold	(min)		t_6	0	0	0	0	0
DIORn data high impedance state	(max)	(Note 2, 4)	t_{6z}	30	30	30	30	30
DIORn / DIOWn to address valid hold	(min)	(Note 4)	t_9	20	15	10	10	10
Read Data Valid to IORDY active (if IORDY initially low after t_A)	(min)	(Note 4)	t_{RD}	0	0	0	0	0
IORDY Setup time		(Note 3, 4)	t_A	35	35	35	35	35
IORDY Pulse Width	(max)	(Note 4)	t_B	1250	1250	1250	1250	1250
IORDY assertion to release	(max)		t_C	5	5	5	5	5
DIOWn assert to data valid	(max)		t_{DDV}	10	10	10	10	10

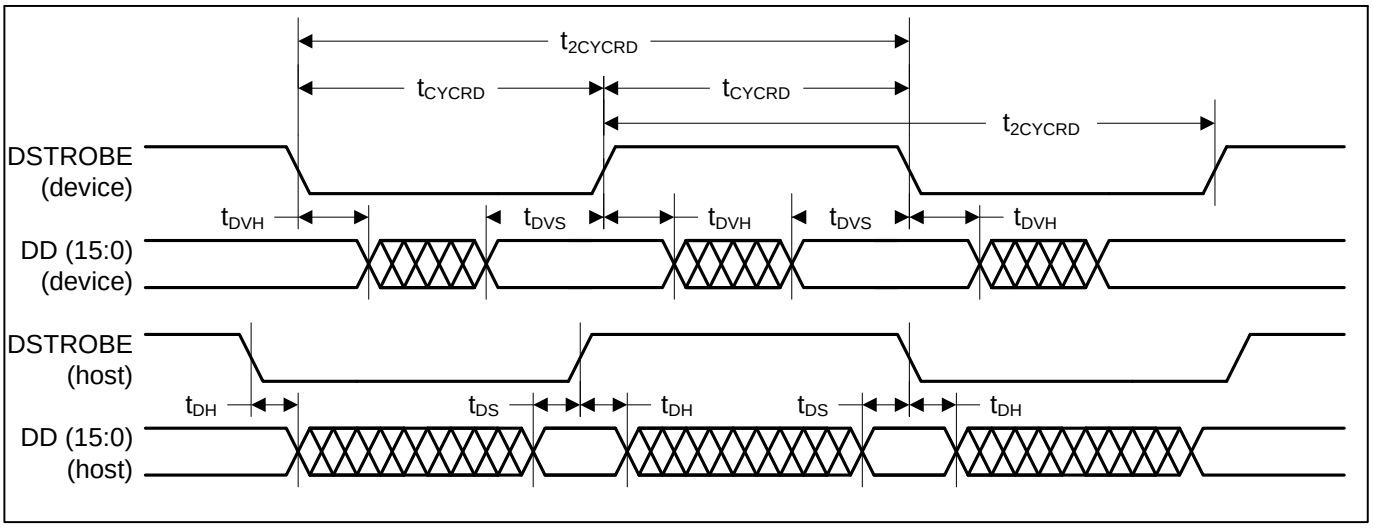
Note: 1. t_0 is the minimum total cycle time, t_2 is the minimum DIORn / DIOWn assertion time, and t_{2i} is the minimum DIORn / DIOWn negation time. A host implementation shall lengthen t_2 and/or t_{2i} to ensure that t_0 is equal to or greater than the value reported in the devices IDENTIFY DEVICE data. A device implementation shall support any legal host implementation.

2. This parameter specifies the time from the negation edge of DIORn to the time that the data bus is released by the device.

3. The delay from the activation of DIORn or DIOWn until the state of IORDY is first sampled. If IORDY is inactive then the host shall wait until IORDY is active before the register transfer cycle is completed. If the device is not driving IORDY negated at the t_A after the activation of DIORn or DIOWn, then t_5 shall be met and t_{RD} is not applicable. If the device is driving IORDY negated at the time t_A after the activation of DIORn or DIOWn, then t_{RD} shall be met and t_5 is not applicable.

4. Timings based upon software control. See User's Guide.

5. All IDE timing is based upon HCLK = 100 MHz.



Note: DD (15:0) and DSTROBE signals are shown at both the host and the device to emphasize that cable settling time as well as cable propagation delay shall not allow the data signals to be considered stable at the host until some time after they are driven by the device.

Figure 20. Sustained Ultra DMA data-in Burst

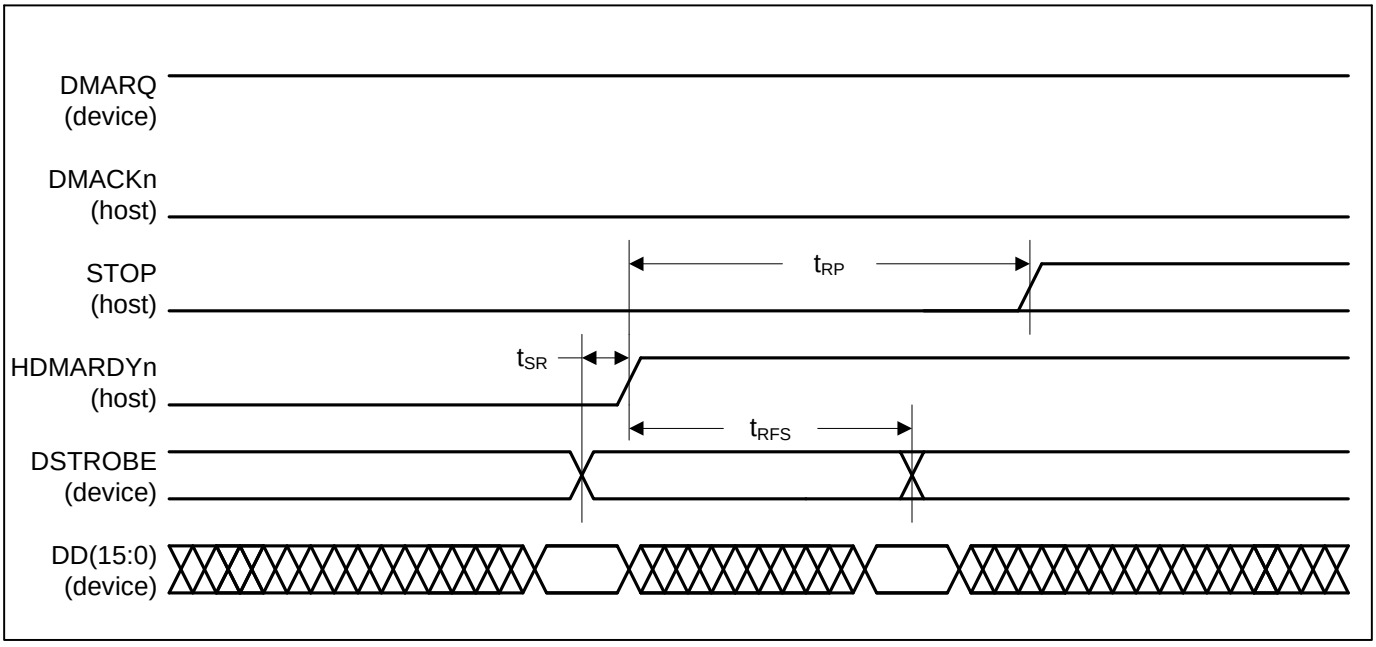
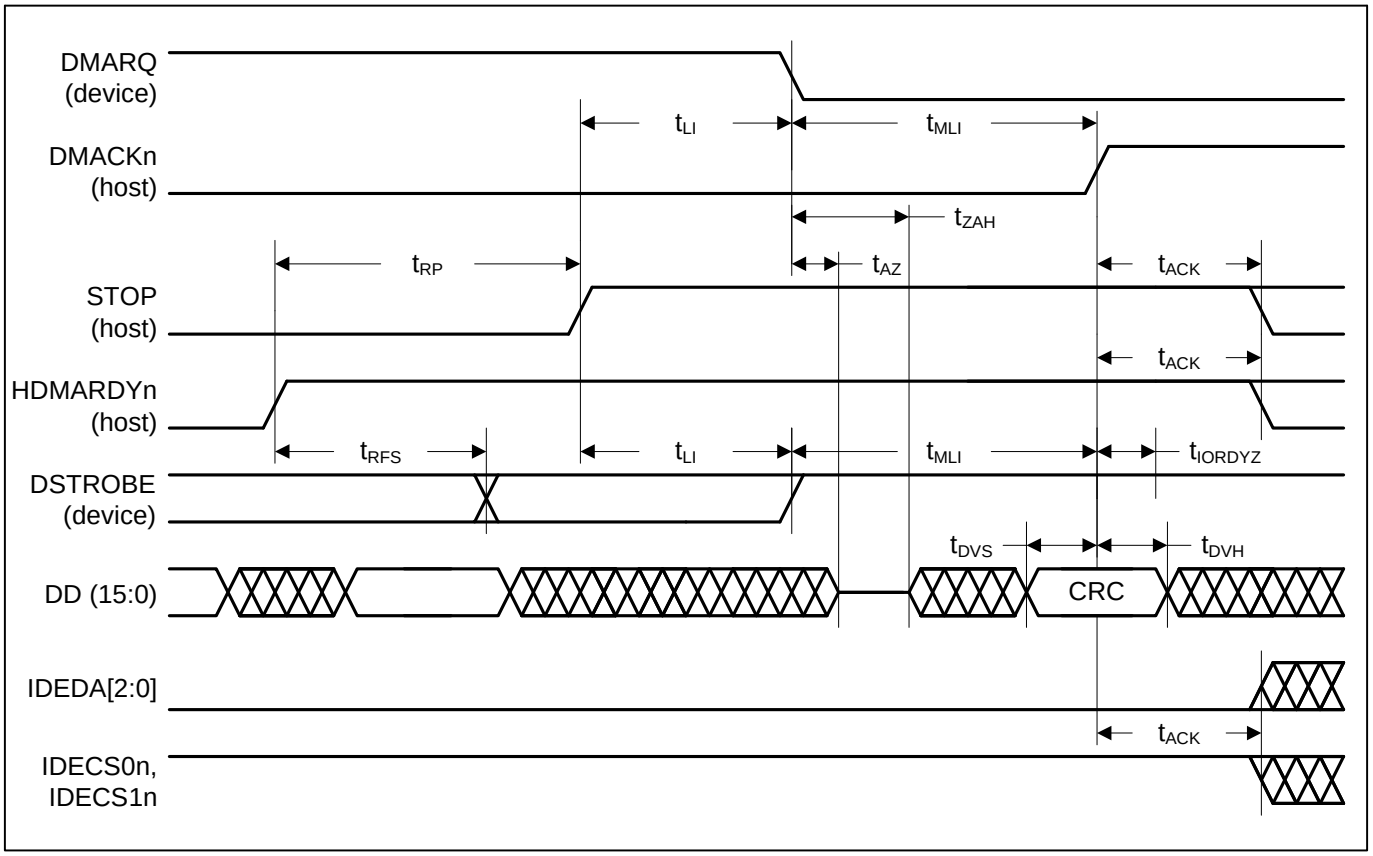
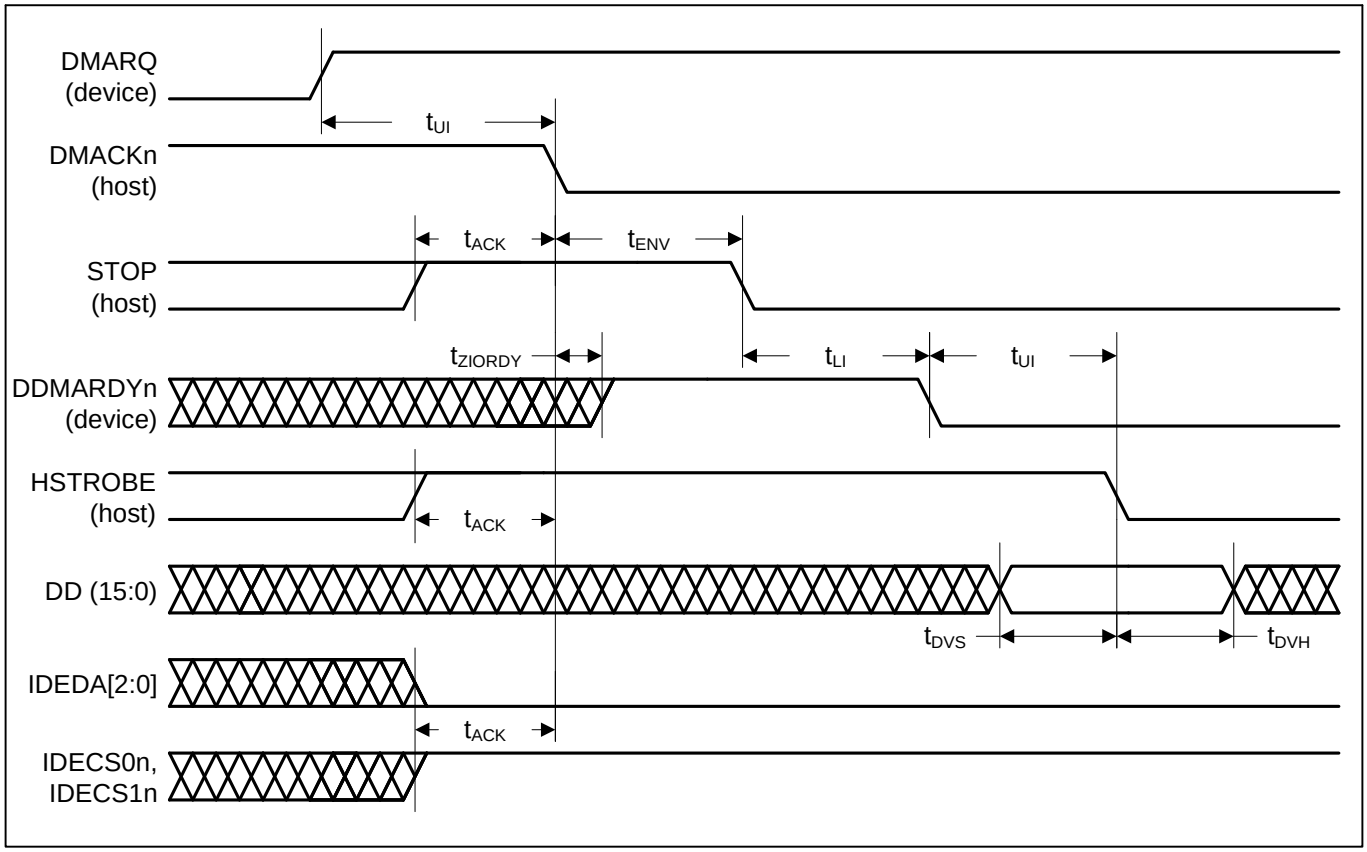


Figure 21. Host Pausing an Ultra DMA data-in Burst



Note: The definitions for the DIOWn:STOP, DIORn:HDMARDYn:HSTROBE and IORDY:DDMARDYn:DSTROBE signal lines are no longer in effect after DMARQ and DMACKn are negated.

Figure 23. Host Terminating an Ultra DMA data-in Burst



Note: The definitions for the DIOWn:STOP, DIORn:HDMARDYn:HSTROBE and IORDY:DDMARDYn:DSTROBE signal lines are not in effect until DMARQ and DMACKn are asserted.

Figure 24. Initiating an Ultra DMA data-out Burst

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}	1	4	8	ns
SCLK to LRCLK assert delay time	t_{LRd}	-	-	3	ns
Hold between SCLK assert then LRCLK deassert *or* Hold between LRCLK deassert then SCLK assert	t_{LRh}	0	-	-	ns
SDI to SCLK deassert setup time	t_{SDIs}	12	-	-	ns
SDI from SCLK deassert hold time	t_{SDIh}	0	-	-	ns
SCLK assert to SDO delay time	t_{SDOd}	-	-	9	ns
SDO from SCLK assert hold time	t_{SDOh}	1	-	-	ns

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

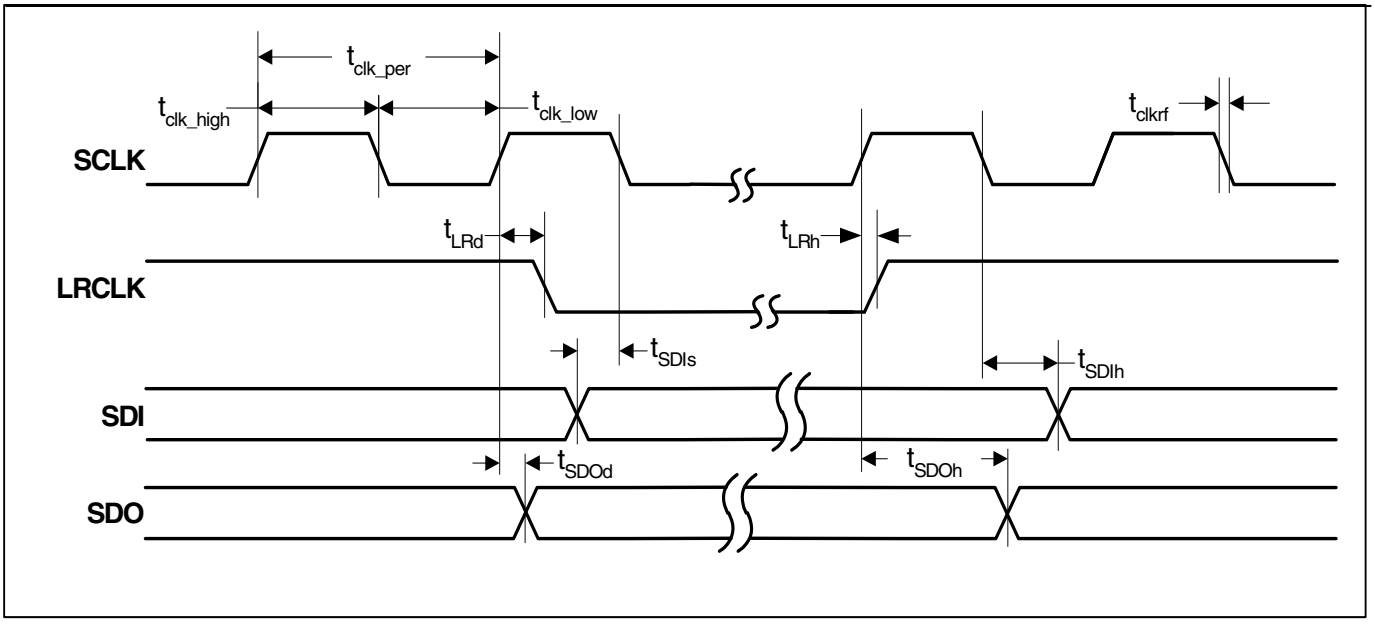


Figure 33. Inter-IC Sound (I^2S) Timing Measurement

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/fall time	t_{clkrf}	2	-	6	ns
ASDI setup to ABITCLK falling	t_s	10	-	-	ns
ASDI hold after ABITCLK falling	t_h	10	-	-	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/fall time, C_L = 55 pF	t_{rfout}	2	-	6	ns

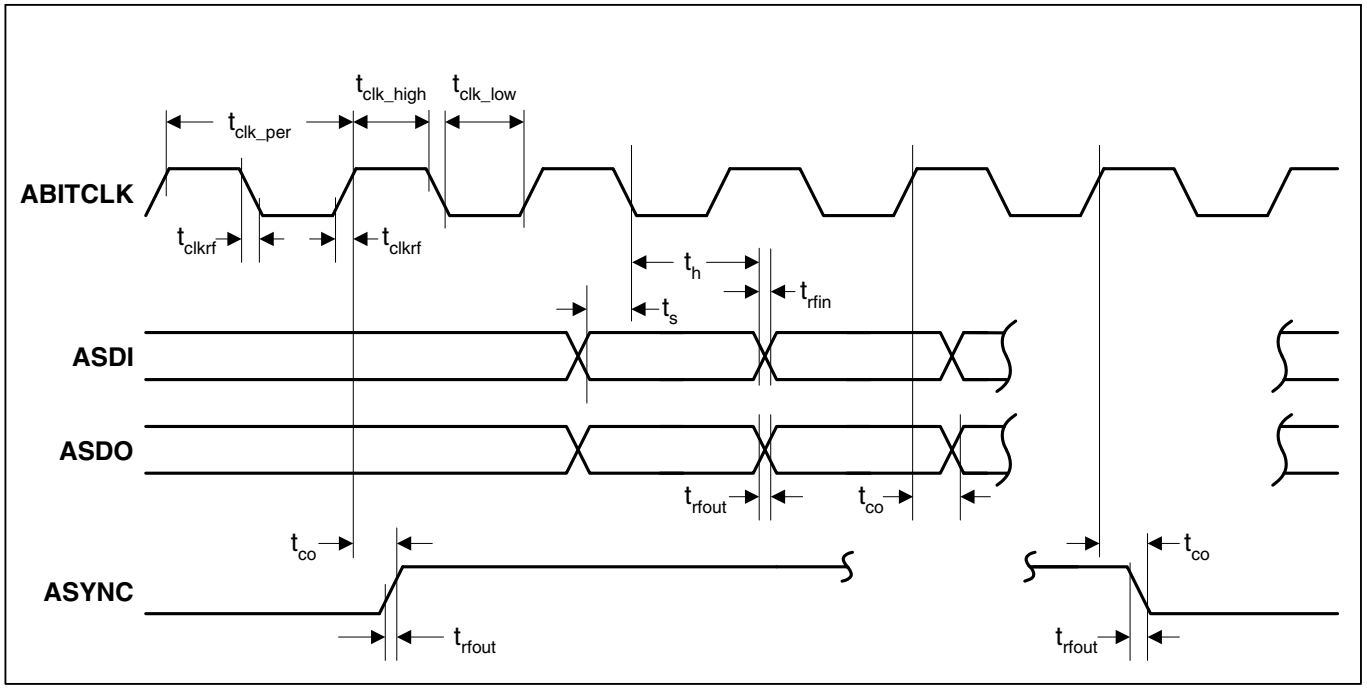


Figure 34. AC '97 Configuration Timing Measurement

LCD Interface

Parameter	Symbol	Min	Typ	Max	Unit
SPCLK rise/fall time	t_{clkr}	2	-	8	ns
SPCLK rising edge to control signal transition time	t_{CD}	-	-	3	ns
SPCLK rising edge to data transition time	t_{DD}	-	-	10	ns
Data valid time	t_{Dv}	t_{SPCLK}	-	-	ns

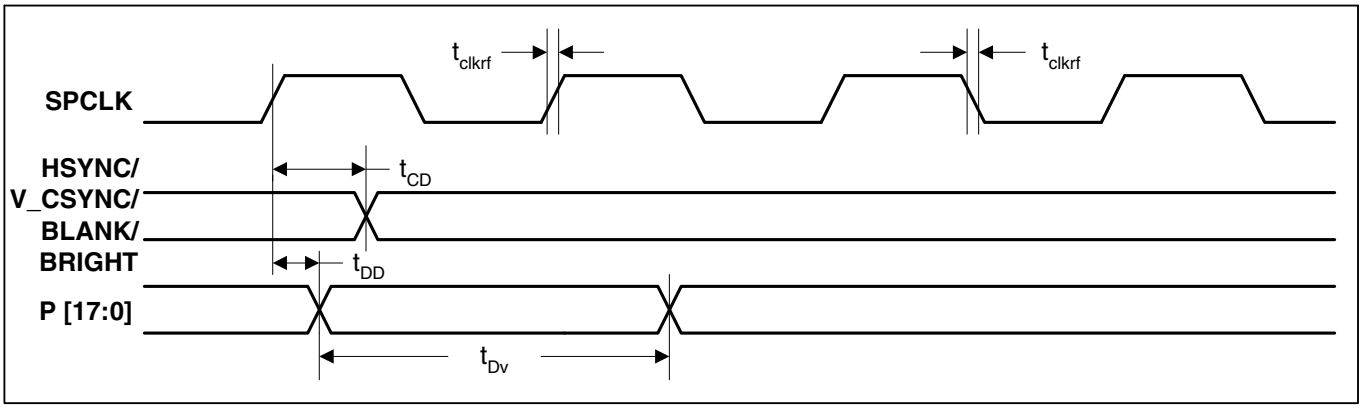


Figure 35. LCD Timing Measurement

352 Pin BGA Package Outline

352-Ball PBGA Diagram

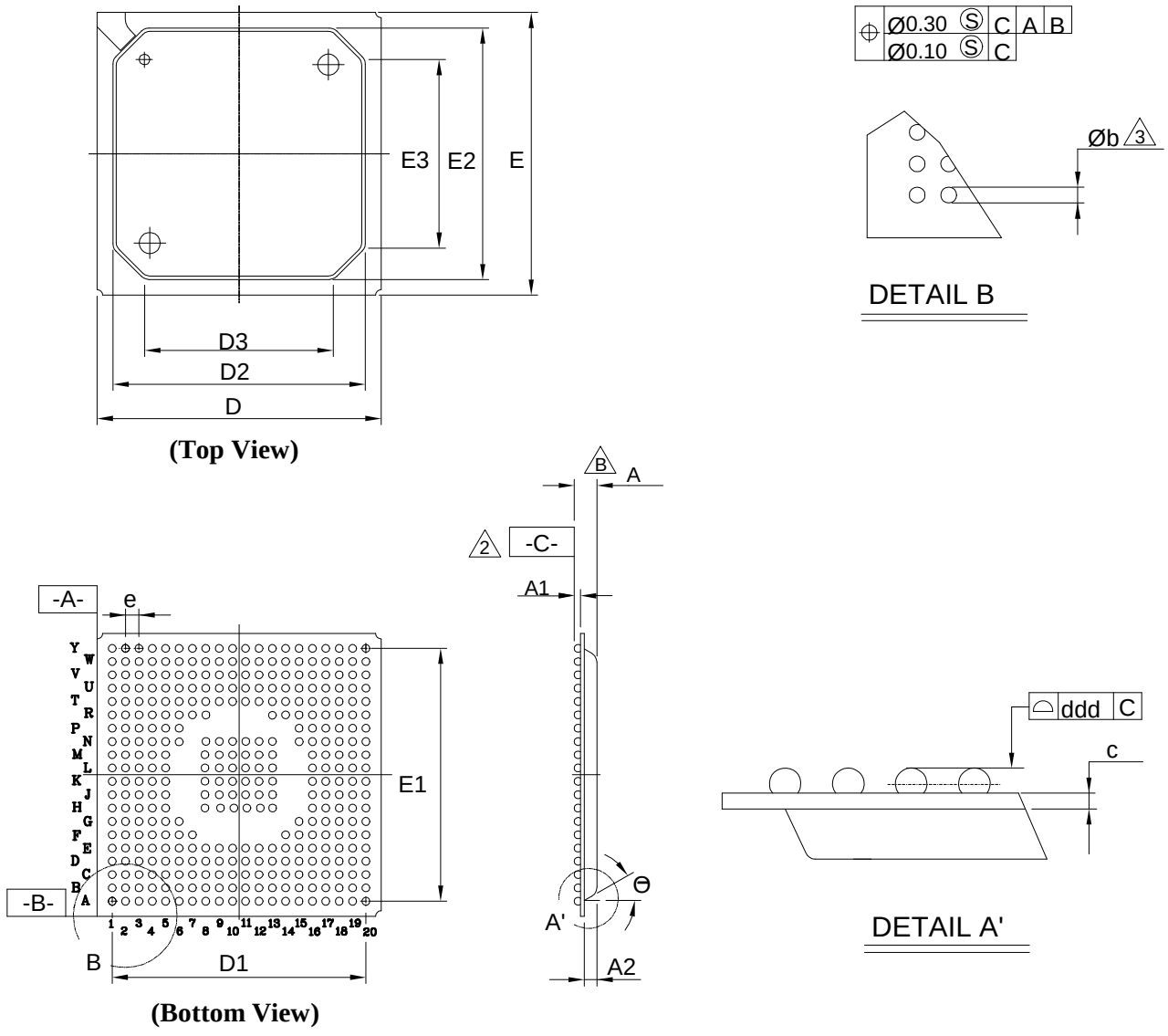


Figure 38. 352 Pin PBGA Pin Diagram

Figure 40. 352 PIN BGA PINOUT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Y	HSYNC	DD[1]	DD[12]	P[2]	AD[15]	DA[6]	DA[4]	AD[10]	DA[1]	AD[8]	IDEDA[0]	DTRN	TDO	BOOT[0]	EEDAT	ASDO	SFRM1	RDLED	USBP[1]	ABITCLK	**Y**
W	P[12]	P[9]	DD[0]	P[5]	P[3]	DA[7]	DA[5]	AD[11]	AD[9]	IDECS1N	IDEDA[1]	TCK	TMS	EECLK	SCLK1	GRLED	INT[3]	SLA[1]	SLA[0]	RXD[2]	**W**
V	P[16]	P[11]	P[8]	DD[15]	DD[13]	P[1]	AD[14]	AD[12]	DA[2]	IDECS0N	IDEDA[2]	TDI	GND	ASYNC	SSPTX1	INT[2]	RTSN	USBP[0]	CTSN	TXD[0]	**V**
U	AD[0]	P[15]	P[10]	P[7]	P[6]	P[4]	P[0]	AD[13]	DA[3]	DA[0]	DSRN	BOOT[1]	NC	SSPRX1	INT[1]	PWMOUT	USBM[0]	RXD[1]	TXD[1]	ROW[1]	**U**
T	DA[8]	BLANK	P[13]	SPCLK	V_CSYNC	DD[14]	GND	CVDD	RVDD	GND	GND	RVDD	CVDD	GND	INT[0]	USBM[1]	RXD[0]	TXD[2]	ROW[2]	ROW[4]	**T**
R	AD[2]	AD[1]	P[17]	P[14]	RVDD	RVDD	GND	CVDD					CVDD	GND	RVDD	RVDD	ROW[0]	ROW[3]	PLL_GND	ROW[5]	**R**
P	AD[4]	DA[10]	DA[9]	BRIGHT	RVDD	RVDD									RVDD	RVDD	XTALI	PLL_VDD	ROW[6]	ROW[7]	**P**
N	DA[13]	DA[12]	DA[11]	AD[3]	CVDD	CVDD		GND	GND	GND	GND	GND	GND		GND	GND	XTALO	COL[0]	COL[1]	COL[2]	**N**
M	AD[7]	DA[14]	AD[6]	AD[5]	CVDD			GND	GND	GND	GND	GND	GND			GND	COL[4]	COL[3]	COL[6]	CSN[0]	**M**
L	DA[18]	DA[17]	DA[16]	DA[15]	GND			GND	GND	GND	GND	GND	GND			CVDD	COL[5]	COL[7]	RSTON	PRSTN	**L**
K	AD[22]	DA[20]	AD[21]	DA[19]	RVDD			GND	GND	GND	GND	GND	GND			CVDD	SYM	SYP	SXM	SXP	**K**
J	DA[21]	DQMN[0]	DQMN[1]	DQMN[2]	GND			GND	GND	GND	GND	GND	GND			CVDD	RTCXTALI	XM	YP	YM	**J**
H	DQMN[3]	CASN	RASN	SDCSN[2]	CVDD			GND	GND	GND	GND	GND	GND			RVDD	RTCXTALO	ADC_VDD	ADC_GND	XP	**H**
G	SDCSN[0]	SDCSN[1]	SDWEN	SDCLK	RVDD	RVDD									RVDD	RVDD	EGPIO[7]	EGPIO[9]	EGPIO[10]	EGPIO[11]	**G**
F	SDCSN[3]	DA[22]	DA[24]	AD[25]	RVDD	GND	CVDD							CVDD	GND	GND	EGPIO[2]	EGPIO[4]	EGPIO[6]	EGPIO[8]	**F**
E	AD[23]	DA[23]	DA[26]	CSN[6]	GND	GND	CVDD	CVDD	RVDD	GND	GND	RVDD	CVDD	CVDD	GND	ASDI	DIOWN	EGPIO[0]	EGPIO[3]	EGPIO[5]	**E**
D	AD[24]	DA[25]	DD[11]	SDCLKEN	AD[19]	DD[9]	DD[5]	AD[16]	MIIRXD[2]	MIITXD[3]	TXEN	NC	NC	NC	EGPIO[14]	NC	USBM[2]	ARSTN	DIORN	EGPIO[1]	**D**
C	CSN[1]	CSN[3]	AD[20]	DA[29]	DD[10]	DD[6]	DD[2]	MDC	MIIRXD[3]	TXCLK	MIITXD[0]	NC	NC	NC	NC	NC	NC	USBP[2]	IORDY	DMACKN	**C**
B	CSN[2]	DA[31]	DA[30]	DA[27]	DD[7]	DD[3]	WRN	MDIO	MIIRXD[1]	RXERR	MIITXD[1]	CRS	NC	NC	NC	NC	EGPIO[13]	NC	WAITN	TRSTN	**B**
A	CSN[7]	DA[28]	AD[18]	DD[8]	DD[4]	AD[17]	RDN	RXCLK	MIIRXD[0]	RXDVAL	MIITXD[2]	TXERR	CLD	NC	NC	NC	EGPIO[12]	EGPIO[15]	NC	NC	**A**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	

The following section focuses on the EP9312 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 EP9312 pin signals. The second table (Table T) illustrates the pin signal multiplexing and configuration options.

Table S is a summary of the EP9312 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 4 mA output driver
- 8mA - Pin is an 8 mA output driver
- 12mA - Pin is an 12 mA 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 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]	IDE DMA request	DMARQ
EGPIO[3]	Transmit Enable output / HDLC clocks	TENn / HDLCCLK1 / HDLCCLK3
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[14]	PWM 1 output	PWMOUT1
EGPIO[15]	IDE 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
IDEDA[2:0]	GPIO	GPIO Port E[7:5]
IDECS0n	GPIO	GPIO Port E[4]
IDECS1n	GPIO	GPIO Port E[3]
DIORn	GPIO	GPIO Port E[2]
DD[7:0]	GPIO	GPIO Port H[7:0]
DD[15:12]	GPIO	GPIO Port G[7:4]
SLA[1:0]	GPIO	GPIO Port G[3:2]
EEDAT	GPIO	GPIO Port G[1]
EECLK	GPIO	GPIO Port G[0]