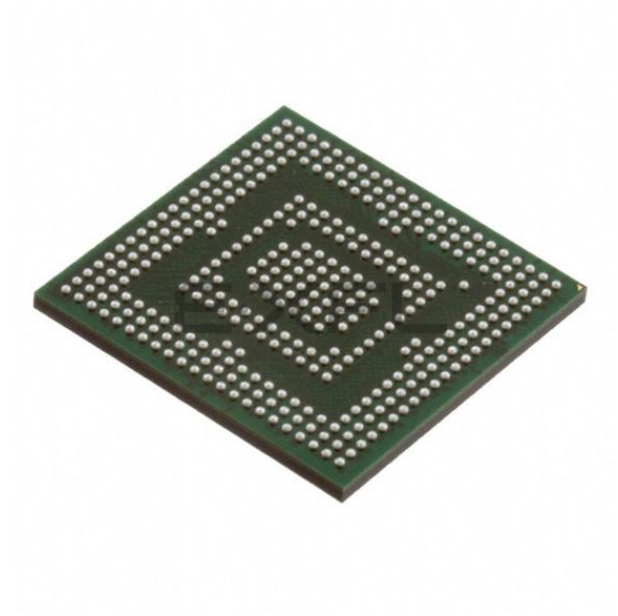


Welcome to E-XFL.COM

Understanding Embedded - DSP (Digital Signal Processors)

Embedded - DSP (Digital Signal Processors) are specialized microprocessors designed to perform complex mathematical computations on digital signals in real-time. Unlike general-purpose processors, DSPs are optimized for high-speed numeric processing tasks, making them ideal for applications that require efficient and precise manipulation of digital data. These processors are fundamental in converting and processing signals in various forms, including audio, video, and communication signals, ensuring that data is accurately interpreted and utilized in embedded systems.

Applications of Embedded - DSP (Digital Signal Processors)

Details	
Product Status	Active
Type	Dual Core
Interface	CAN, EBI/EMI, Ethernet, I²C, SPI, SPORT, UART/USART, USB OTG
Clock Rate	500MHz
Non-Volatile Memory	ROM (64kB)
On-Chip RAM	808K x 8
Voltage - I/O	1.8V, 3.3V
Voltage - Core	1.25V
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	349-LFBGA, CSPBGA
Supplier Device Package	349-CSPBGA (19x19)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adbf607wcbcz502

a very small final memory size. The instruction set also provides fully featured multifunction instructions that allow the programmer to use many of the processor core resources in a single instruction. Coupled with many features more often seen on microcontrollers, this instruction set is very efficient when compiling C and C++ source code. In addition, the architecture supports both user (algorithm/application code) and supervisor (O/S kernel, device drivers, debuggers, ISRs) modes of operation, allowing multiple levels of access to core processor resources.

The assembly language, which takes advantage of the processor's unique architecture, offers the following advantages:

- Seamlessly integrated DSP/MCU features are optimized for both 8-bit and 16-bit operations.
- A multi-issue load/store modified-Harvard architecture, which supports two 16-bit MAC or four 8-bit ALU + two load/store + two pointer updates per cycle.
- All registers, I/O, and memory are mapped into a unified 4G byte memory space, providing a simplified programming model.
- Control of all asynchronous and synchronous events to the processor is handled by two subsystems: the Core Event Controller (CEC) and the System Event Controller (SEC).
- Microcontroller features, such as arbitrary bit and bit-field manipulation, insertion, and extraction; integer operations on 8-, 16-, and 32-bit data-types; and separate user and supervisor stack pointers.
- Code density enhancements, which include intermixing of 16-bit and 32-bit instructions (no mode switching, no code segregation). Frequently used instructions are encoded in 16 bits.

PROCESSOR INFRASTRUCTURE

The following sections provide information on the primary infrastructure components of the ADSP-BF609 processor.

DMA Controllers

The processor uses Direct Memory Access (DMA) to transfer data within memory spaces or between a memory space and a peripheral. The processor can specify data transfer operations and return to normal processing while the fully integrated DMA controller carries out the data transfers independent of processor activity.

DMA transfers can occur between memory and a peripheral or between one memory and another memory. Each Memory-to-memory DMA stream uses two channels, where one channel is the source channel, and the second is the destination channel.

All DMAs can transport data to and from all on-chip and off-chip memories. Programs can use two types of DMA transfers, descriptor-based or register-based. Register-based DMA allows the processor to directly program DMA control registers to initiate a DMA transfer. On completion, the control registers may be automatically updated with their original setup values for continuous transfer. Descriptor-based DMA transfers require a set of parameters stored within memory to initiate a DMA sequence. Descriptor-based DMA transfers allow multiple DMA sequences to be chained together and a DMA channel can be programmed to automatically set up and start another DMA transfer after the current sequence completes.

The DMA controller supports the following DMA operations.

- A single linear buffer that stops on completion.
- A linear buffer with negative, positive or zero stride length.
- A circular, auto-refreshing buffer that interrupts when each buffer becomes full.
- A similar buffer that interrupts on fractional buffers (for example, 1/2, 1/4).
- 1D DMA – uses a set of identical ping-pong buffers defined by a linked ring of two-word descriptor sets, each containing a link pointer and an address.
- 1D DMA – uses a linked list of 4 word descriptor sets containing a link pointer, an address, a length, and a configuration.
- 2D DMA – uses an array of one-word descriptor sets, specifying only the base DMA address.
- 2D DMA – uses a linked list of multi-word descriptor sets, specifying everything.

CRC Protection

The two CRC protection modules allow system software to periodically calculate the signature of code and/or data in memory, the content of memory-mapped registers, or communication message objects. Dedicated hardware circuitry compares the signature with pre calculated values and triggers appropriate fault events.

For example, every 100 ms the system software might initiate the signature calculation of the entire memory contents and compare these contents with expected, pre calculated values. If a mismatch occurs, a fault condition can be generated (via the processor core or the trigger routing unit).

The CRC is a hardware module based on a CRC32 engine that computes the CRC value of the 32-bit data words presented to it. Data is provided by the source channel of the memory-to-memory DMA (in memory scan mode) and is optionally forwarded to the destination channel (memory transfer mode). The main features of the CRC peripheral are:

- Memory scan mode
- Memory transfer mode
- Data verify mode
- Data fill mode
- User-programmable CRC32 polynomial
- Bit/byte mirroring option (endianness)
- Fault/error interrupt mechanisms
- 1D and 2D fill block to initialize array with constants.
- 32-bit CRC signature of a block of a memory or MMR block.

- Support for remote frames.
- Active or passive network support.
- CAN wakeup from hibernation mode (lowest static power consumption mode).
- Interrupts, including: TX complete, RX complete, error and global.

An additional crystal is not required to supply the CAN clock, as the CAN clock is derived from a system clock through a programmable divider.

10/100 Ethernet MAC

The processor can directly connect to a network by way of an embedded fast Ethernet media access controller (MAC) that supports both 10-BaseT (10M bits/sec) and 100-BaseT (100M bits/sec) operation. The 10/100 Ethernet MAC peripheral on the processor is fully compliant to the IEEE 802.3-2002 standard and it provides programmable features designed to minimize supervision, bus use, or message processing by the rest of the processor system.

Some standard features are:

- Support and RMII protocols for external PHYs
- Full duplex and half duplex modes
- Media access management (in half-duplex operation)
- Flow control
- Station management: generation of MDC/MDIO frames for read-write access to PHY registers

Some advanced features are:

- Automatic checksum computation of IP header and IP payload fields of RX frames
- Independent 32-bit descriptor-driven receive and transmit DMA channels
- Frame status delivery to memory through DMA, including frame completion semaphores for efficient buffer queue management in software
- TX DMA support for separate descriptors for MAC header and payload to eliminate buffer copy operations
- Convenient frame alignment modes
- 47 MAC management statistics counters with selectable clear-on-read behavior and programmable interrupts on half maximum value
- Advanced power management
- Magic packet detection and wakeup frame filtering
- Support for 802.3Q tagged VLAN frames
- Programmable MDC clock rate and preamble suppression

IEEE 1588 Support

The IEEE 1588 standard is a precision clock synchronization protocol for networked measurement and control systems. The processor includes hardware support for IEEE 1588 with an integrated precision time protocol synchronization engine (PTP_TSYNC). This engine provides hardware assisted time stamping to improve the accuracy of clock synchronization between PTP nodes. The main features of the engine are:

- Support for both IEEE 1588-2002 and IEEE 1588-2008 protocol standards
- Hardware assisted time stamping capable of up to 12.5 ns resolution
- Lock adjustment
- Automatic detection of IPv4 and IPv6 packets, as well as PTP messages
- Multiple input clock sources (SCLK0, RMII clock, external clock)
- Programmable pulse per second (PPS) output
- Auxiliary snapshot to time stamp external events

USB 2.0 On-the-Go Dual-Role Device Controller

The USB 2.0 OTG dual-role device controller provides a low-cost connectivity solution for the growing adoption of this bus standard in industrial applications, as well as consumer mobile devices such as cell phones, digital still cameras, and MP3 players. The USB 2.0 controller allows these devices to transfer data using a point-to-point USB connection without the need for a PC host. The module can operate in a traditional USB peripheral-only mode as well as the host mode presented in the On-the-Go (OTG) supplement to the USB 2.0 specification.

The USB clock (USB_CLKIN) is provided through a dedicated external crystal or crystal oscillator.

The USB On-the-Go dual-role device controller includes a Phase Locked Loop with programmable multipliers to generate the necessary internal clocking frequency for USB.

POWER AND CLOCK MANAGEMENT

The processor provides four operating modes, each with a different performance/power profile. When configured for a 0 V internal supply voltage (V_{DD_INT}), the processor enters the hibernate state. Control of clocking to each of the processor peripherals also reduces power consumption. See Table 5 for a summary of the power settings for each mode.

Crystal Oscillator (SYS_XTAL)

The processor can be clocked by an external crystal (Figure 6), a sine wave input, or a buffered, shaped clock derived from an external clock oscillator. If an external clock is used, it should be a TTL compatible signal and must not be halted, changed, or operated below the specified frequency during normal operation. This signal is connected to the processor's SYS_CLKIN pin. When an external clock is used, the SYS_XTAL pin must be left unconnected. Alternatively, because the processor includes an on-chip oscillator circuit, an external crystal may be used.

For fundamental frequency operation, use the circuit shown in Figure 6. A parallel-resonant, fundamental frequency, microprocessor grade crystal is connected across the SYS_CLKIN and XTAL pins. The on-chip resistance between SYS_CLKIN and the XTAL pin is in the 500 kΩ range. Further parallel resistors are typically not recommended.

Table 6. Detailed Signal Descriptions (Continued)

Signal Name	Direction	Description
RSI_CLK	Output	**Clock** The clock signal applied to the connected device from the RSI.
RSI_CMD	I/O	**Command** Used to send commands to and receive responses from the connected device.
RSI_Dn	I/O	**Data n** Bidirectional data bus.
SMC_ABEn	Output	**Byte Enable n** Indicate whether the lower or upper byte of a memory is being accessed. When an asynchronous write is made to the upper byte of a 16-bit memory, SMC_ABE1 =0 and SMC_ABE0 =1. When an asynchronous write is made to the lower byte of a 16-bit memory, SMC_ABE1 =1 and SMC_ABE0 =0.
SMC_AMSn	Output	**Memory Select n** Typically connects to the chip select of a memory device.
SMC_Ann	Output	**Address n** Address bus.
SMC_AOE	Output	**Output Enable** Asserts at the beginning of the setup period of a read access.
SMC_ARDY	Input	**Asynchronous Ready** Flow control signal used by memory devices to indicate to the SMC when further transactions may proceed.
SMC_ARE	Output	**Read Enable** Asserts at the beginning of a read access.
SMC_AWE	Output	**Write Enable** Asserts for the duration of a write access period.
SMC_BG	Output	**Bus Grant** Output used to indicate to an external device that it has been granted control of the SMC buses.
SMC_BGH	Output	**Bus Grant Hang** Output used to indicate that the SMC has a pending transaction which requires control of the bus to be restored before it can be completed.
SMC_BR	Input	**Bus Request** Input used by an external device to indicate that it is requesting control of the SMC buses.
SMC_Dnn	I/O	**Data n** Bidirectional data bus.
SMC_NORCLK	Output	**NOR Clock** Clock for synchronous burst mode.
SMC_NORDV	Output	**NOR Data Valid** Asserts for the duration of a synchronous burst mode read setup period.
SMC_NORWT	Input	**NOR Wait** Flow control signal used by memory devices in synchronous burst mode to indicate to the SMC when further transactions may proceed.
SPI_CLK	I/O	**Clock** Input in slave mode, output in master mode.
SPI_D2	I/O	**Data 2** Used to transfer serial data in quad mode. Open drain in ODM mode.
SPI_D3	I/O	**Data 3** Used to transfer serial data in quad mode. Open drain in ODM mode.
SPI_MISO	I/O	**Master In, Slave Out** Used to transfer serial data. Operates in the same direction as SPI_MOSI in dual and quad modes. Open drain in ODM mode.
SPI_MOSI	I/O	**Master Out, Slave In** Used to transfer serial data. Operates in the same direction as SPI_MISO in dual and quad modes. Open drain in ODM mode.
SPI_RDY	I/O	**Ready** Optional flow signal. Output in slave mode, input in master mode.
SPI_SELn	Output	**Slave Select Output n** Used in master mode to enable the desired slave.
SPI_SS	Input	**Slave Select Input** Slave mode: acts as the slave select input. Master mode: optionally serves as an error detection input for the SPI when there are multiple masters.
SPT_ACLK	I/O	**Channel A Clock** Data and frame sync are driven/sampled with respect to this clock. This signal can be either internally or externally generated.
SPT_AD0	I/O	**Channel A Data 0** Primary bidirectional data I/O. This signal can be configured as an output to transmit serial data, or as an input to receive serial data.
SPT_AD1	I/O	**Channel A Data 1** Secondary bidirectional data I/O. This signal can be configured as an output to transmit serial data, or as an input to receive serial data.
SPT_AFS	I/O	**Channel A Frame Sync** The frame sync pulse initiates shifting of serial data. This signal is either generated internally or externally.
SPT_ATDV	Output	**Channel A Transmit Data Valid** This signal is optional and only active when SPORT is configured in multi-channel transmit mode. It is asserted during enabled slots.
SPT_BCLK	I/O	**Channel B Clock** Data and frame sync are driven/sampled with respect to this clock. This signal can be either internally or externally generated.
SPT_BD0	I/O	**Channel B Data 0** Primary bidirectional data I/O. This signal can be configured as an output to transmit serial data, or as an input to receive serial data.

349-BALL CSP_BGA SIGNAL DESCRIPTIONS

The processors' pin definitions are shown in the table. The columns in this table provide the following information:

- Signal Name: The Signal Name column in the table includes the Signal Name for every pin.
- Description: The Description column in the table provides a verbose (descriptive) name for the signal.
- Port: The General-Purpose I/O Port column in the table shows whether or not the signal is multiplexed with other signals on a general-purpose I/O port pin.
- Pin Name: The Pin Name column in the table identifies the name of the package pin (at power-on reset) on which the signal is located (if a single function pin) or is multiplexed (if a general-purpose I/O pin).

Table 7. ADSP-BF60x 349-Ball CSP_BGA Signal Descriptions

Signal Name	Description	Port	Pin Name
ACM0_A0	ACM0 Address 0	F	PF_14
ACM0_A1	ACM0 Address 1	F	PF_15
ACM0_A2	ACM0 Address 2	F	PF_12
ACM0_A3	ACM0 Address 3	F	PF_13
ACM0_A4	ACM0 Address 4	F	PF_10
ACM0_CLK	ACM0 Clock	E	PE_04
ACM0_FS	ACM0 Frame Sync	E	PE_03
ACM0_T0	ACM0 External Trigger 0	E	PE_08
ACM0_T1	ACM0 External Trigger 1	G	PG_05
CAN0_RX	CAN0 Receive	G	PG_04
CAN0_TX	CAN0 Transmit	G	PG_01
CNT0_DG	CNT0 Count Down and Gate	G	PG_12
CNT0_UD	CNT0 Count Up and Direction	G	PG_11
CNT0_ZM	CNT0 Count Zero Marker	G	PG_07
DMC0_A00	DMC Address 0	Not Muxed	DMC0_A00
DMC0_A01	DMC Address 1	Not Muxed	DMC0_A01
DMC0_A02	DMC Address 2	Not Muxed	DMC0_A02
DMC0_A03	DMC Address 3	Not Muxed	DMC0_A03
DMC0_A04	DMC Address 4	Not Muxed	DMC0_A04
DMC0_A05	DMC Address 5	Not Muxed	DMC0_A05
DMC0_A06	DMC Address 6	Not Muxed	DMC0_A06
DMC0_A07	DMC Address 7	Not Muxed	DMC0_A07
DMC0_A08	DMC Address 8	Not Muxed	DMC0_A08
DMC0_A09	DMC Address 9	Not Muxed	DMC0_A09
DMC0_A10	DMC Address 10	Not Muxed	DMC0_A10
DMC0_A11	DMC Address 11	Not Muxed	DMC0_A11
DMC0_A12	DMC Address 12	Not Muxed	DMC0_A12
DMC0_A13	DMC Address 13	Not Muxed	DMC0_A13
DMC0_BA0	DMC Bank Address Input 0	Not Muxed	DMC0_BA0
DMC0_BA1	DMC Bank Address Input 1	Not Muxed	DMC0_BA1
DMC0_BA2	DMC Bank Address Input 2	Not Muxed	DMC0_BA2
$\overline{\text{DMC0\_CAS}}$	DMC Column Address Strobe	Not Muxed	$\overline{\text{DMC0\_CAS}}$
DMC0_CK	DMC Clock	Not Muxed	DMC0_CK
DMC0_CKE	DMC Clock Enable	Not Muxed	DMC0_CKE
$\overline{\text{DMC0\_CK}}$	DMC Clock (complement)	Not Muxed	$\overline{\text{DMC0\_CK}}$
$\overline{\text{DMC0\_CS0}}$	DMC Chip Select 0	Not Muxed	$\overline{\text{DMC0\_CS0}}$

Table 7. ADSP-BF60x 349-Ball CSP_BGA Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
ETH_PTPCLKIN	EMAC0/EMAC1 PTP Clock Input	C	PC_13
GND	Ground	Not Muxed	GND
$\overline{\text{JTG\_EMU}}$	Emulation Output	Not Muxed	$\overline{\text{JTG\_EMU}}$
JTG_TCK	JTAG Clock	Not Muxed	JTG_TCK
JTG_TDI	JTAG Serial Data Input	Not Muxed	JTG_TDI
JTG_TDO	JTAG Serial Data Output	Not Muxed	JTG_TDO
JTG_TMS	JTAG Mode Select	Not Muxed	JTG_TMS
$\overline{\text{JTG\_TRST}}$	JTAG Reset	Not Muxed	$\overline{\text{JTG\_TRST}}$
LP0_ACK	LP0 Acknowledge	B	PB_01
LP0_CLK	LP0 Clock	B	PB_00
LP0_D0	LP0 Data 0	A	PA_00
LP0_D1	LP0 Data 1	A	PA_01
LP0_D2	LP0 Data 2	A	PA_02
LP0_D3	LP0 Data 3	A	PA_03
LP0_D4	LP0 Data 4	A	PA_04
LP0_D5	LP0 Data 5	A	PA_05
LP0_D6	LP0 Data 6	A	PA_06
LP0_D7	LP0 Data 7	A	PA_07
LP1_ACK	LP1 Acknowledge	B	PB_02
LP1_CLK	LP1 Clock	B	PB_03
LP1_D0	LP1 Data 0	A	PA_08
LP1_D1	LP1 Data 1	A	PA_09
LP1_D2	LP1 Data 2	A	PA_10
LP1_D3	LP1 Data 3	A	PA_11
LP1_D4	LP1 Data 4	A	PA_12
LP1_D5	LP1 Data 5	A	PA_13
LP1_D6	LP1 Data 6	A	PA_14
LP1_D7	LP1 Data 7	A	PA_15
LP2_ACK	LP2 Acknowledge	E	PE_08
LP2_CLK	LP2 Clock	E	PE_09
LP2_D0	LP2 Data 0	F	PF_00
LP2_D1	LP2 Data 1	F	PF_01
LP2_D2	LP2 Data 2	F	PF_02
LP2_D3	LP2 Data 3	F	PF_03
LP2_D4	LP2 Data 4	F	PF_04
LP2_D5	LP2 Data 5	F	PF_05
LP2_D6	LP2 Data 6	F	PF_06
LP2_D7	LP2 Data 7	F	PF_07
LP3_ACK	LP3 Acknowledge	E	PE_07
LP3_CLK	LP3 Clock	E	PE_06
LP3_D0	LP3 Data 0	F	PF_08
LP3_D1	LP3 Data 1	F	PF_09
LP3_D2	LP3 Data 2	F	PF_10
LP3_D3	LP3 Data 3	F	PF_11
LP3_D4	LP3 Data 4	F	PF_12
LP3_D5	LP3 Data 5	F	PF_13

Table 7. ADSP-BF60x 349-Ball CSP_BGA Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
UART0_RX	UART0 Receive	D	PD_08
UART0_TX	UART0 Transmit	D	PD_07
UART1_CTS	UART1 Clear to Send	G	PG_13
UART1_RTS	UART1 Request to Send	G	PG_10
UART1_RX	UART1 Receive	G	PG_14
UART1_TX	UART1 Transmit	G	PG_15
USB0_CLKIN	USB0 Clock/Crystal Input	Not Muxed	USB0_CLKIN
USB0_DM	USB0 Data –	Not Muxed	USB0_DM
USB0_DP	USB0 Data +	Not Muxed	USB0_DP
USB0_ID	USB0 OTG ID	Not Muxed	USB0_ID
USB0_VBC	USB0 VBUS Control	Not Muxed	USB0_VBC
USB0_VBUS	USB0 Bus Voltage	Not Muxed	USB0_VBUS
VDD_DMC	VDD for DMC	Not Muxed	VDD_DMC
VDD_EXT	External VDD	Not Muxed	VDD_EXT
VDD_INT	Internal VDD	Not Muxed	VDD_INT
VDD_TD	VDD for Thermal Diode	Not Muxed	VDD_TD
VDD_USB	VDD for USB	Not Muxed	VDD_USB
VREF_DMC	VREF for DMC	Not Muxed	VREF_DMC

Table 12. Signal Multiplexing for Port E

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function Input Tap
PE_00	SPI1_D3	PPI0_D18	SPT1_BD1	
PE_01	SPI1_D2	PPI0_D19	SPT1_BD0	
PE_02	SPI1_RDY	PPI0_D22	SPT1_ACLK	
PE_03		PPI0_D16	ACM0_FS/SPT1_BFS	
PE_04		PPI0_D17	ACM0_CLK/SPT1_BCLK	
PE_05		PPI0_D23	SPT1_AFS	
PE_06	SPT1_ATDV	PPI0_FS3	LP3_CLK	
PE_07	SPT1_BTDV	PPI0_FS2	LP3_ACK	
PE_08	PWM0_SYNC	PPI0_FS1	LP2_ACK	ACM0_T0
PE_09		PPI0_CLK	LP2_CLK	$\overline{\text{PWM0\_TRIP0}}$
PE_10	ETH1_MDC	PWM1_DL	RSI0_D6	
PE_11	ETH1_MDIO	PWM1_DH	RSI0_D7	
PE_12		PWM1_CL	RSI0_D5	
PE_13	ETH1_CRS	PWM1_CH	RSI0_D4	
PE_14		SPT2_ATDV	TM0_TMR0	
PE_15	ETH1_RXD1	PWM1_BL	RSI0_D3	

Table 13. Signal Multiplexing for Port F

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function Input Tap
PF_00	PWM0_AL	PPI0_D00	LP2_D0	
PF_01	PWM0_AH	PPI0_D01	LP2_D1	
PF_02	PWM0_BL	PPI0_D02	LP2_D2	
PF_03	PWM0_BH	PPI0_D03	LP2_D3	
PF_04	PWM0_CL	PPI0_D04	LP2_D4	
PF_05	PWM0_CH	PPI0_D05	LP2_D5	
PF_06	PWM0_DL	PPI0_D06	LP2_D6	
PF_07	PWM0_DH	PPI0_D07	LP2_D7	
PF_08	$\overline{\text{SPI1\_SEL5}}$	PPI0_D08	LP3_D0	
PF_09	$\overline{\text{SPI1\_SEL6}}$	PPI0_D09	LP3_D1	
PF_10	ACM0_A4	PPI0_D10	LP3_D2	
PF_11		PPI0_D11	LP3_D3	$\overline{\text{PWM0\_TRIP1}}$
PF_12	ACM0_A2	PPI0_D12	LP3_D4	
PF_13	ACM0_A3	PPI0_D13	LP3_D5	
PF_14	ACM0_A0	PPI0_D14	LP3_D6	
PF_15	ACM0_A1	PPI0_D15	LP3_D7	

Table 15. ADSP-BF60x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Int Term	Reset Term	Reset Drive	Hiber Term	Hiber Drive	Power Domain	Description and Notes
PB_13	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PB Position 13 | EPPI1 Frame Sync 1 (HSYNC) | ETH0 Transmit Enable | TIMER0 Alternate Capture Input 6. Notes: No notes.
PB_14	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PB Position 14 | EPPI1 Clock | ETH0 Reference Clock. Notes: No notes.
PB_15	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PB Position 15 | EPPI1 Frame Sync 3 (FIELD) | ETH0 PTP Pulse-Per-Second Output. Notes: May be used to wake the processor from hibernate or deep sleep mode.
PC_00	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 0 | EPPI1 Data 0 | ETH0 Receive Data 0. Notes: No notes.
PC_01	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 1 | EPPI1 Data 1 | ETH0 Receive Data 1. Notes: No notes.
PC_02	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 2 | EPPI1 Data 2 | ETH0 Transmit Data 0. Notes: No notes.
PC_03	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 3 | EPPI1 Data 3 | ETH0 Transmit Data 1. Notes: No notes.
PC_04	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 4 | EPPI1 Data 4 | ETH0 Receive Error. Notes: No notes.
PC_05	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 5 | EPPI1 Data 5 | ETH0 Carrier Sense/RMII Receive Data Valid. Notes: No notes.
PC_06	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 6 | EPPI1 Data 6 | ETH0 Management Channel Clock. Notes: No notes.
PC_07	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 7 | EPPI1 Data 7 | ETH0 Management Channel Serial Data. Notes: No notes.
PC_08	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 8 | EPPI1 Data 8. Notes: No notes.
PC_09	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 9 | EPPI1 Data 9 | ETH1 PTP Pulse-Per-Second Output. Notes: No notes.
PC_10	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 10 | EPPI1 Data 10. Notes: No notes.
PC_11	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 11 | EPPI1 Data 11 | ETH PTP Auxiliary Trigger Input. Notes: No notes.
PC_12	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 12 | SPI0 Slave Select Output b | EPPI1 Data 12. Notes: No notes.

Table 15. ADSP-BF60x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Int Term	Reset Term	Reset Drive	Hiber Term	Hiber Drive	Power Domain	Description and Notes
PC_13	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 13 | SPI0 Slave Select Output b | EPPI1 Data 13 | ETH PTP Clock Input. Notes: No notes.
PC_14	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 14 | SPI1 Slave Select Output b | EPPI1 Data 14. Notes: No notes.
PC_15	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PC Position 15 | SPI0 Slave Select Output b | EPPI1 Data 15. Notes: May be used to wake the processor from hibernate or deep sleep mode.
PD_00	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PD Position 0 | SPI0 Data 2 | EPPI1 Data 16 | SPI0 Slave Select Output b. Notes: No notes.
PD_01	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PD Position 1 | SPI0 Data 3 | EPPI1 Data 17 | SPI0 Slave Select Output b. Notes: No notes.
PD_02	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PD Position 2 | SPI0 Master In, Slave Out. Notes: No notes.
PD_03	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PD Position 3 | SPI0 Master Out, Slave In. Notes: No notes.
PD_04	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PD Position 4 | SPI0 Clock. Notes: No notes.
PD_05	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PD Position 5 | SPI1 Clock | TIMER0 Alternate Clock 7. Notes: No notes.
PD_06	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PD Position 6 | EPPI1 Frame Sync 2 (VSYNC) | ETH0 RMII Management Data Interrupt | TIMER0 Alternate Capture Input 5. Notes: May be used to wake the processor from hibernate or deep sleep mode.
PD_07	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PD Position 7 | UART0 Transmit | TIMER0 Alternate Capture Input 3. Notes: No notes.
PD_08	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PD Position 8 | UART0 Receive | TIMER0 Alternate Capture Input 0. Notes: No notes.
PD_09	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PD Position 9 | SPI1 Slave Select Output b | UART0 Request to Send | SPI0 Slave Select Output b. Notes: No notes.
PD_10	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PD Position 10 | SPI0 Ready | UART0 Clear to Send | SPI1 Slave Select Output b. Notes: No notes.
PD_11	I/O	A	wk	wk	none	wk	none	VDD_EXT	Desc: PD Position 11 | SPI0 Slave Select Output b | SPI0 Slave Select Input. Notes: No notes.

Power-Up Reset Timing

In Figure 11, $V_{DD\_SUPPLIES}$ are V_{DD_INT}, V_{DD_EXT}, V_{DD_DMC}, V_{DD_USB}, and V_{DD_TD}.

Table 27. Power-Up Reset Timing

Parameter		Min	Max	Unit
Timing Requirement				
$t_{RST\_IN\_PWR}$	$\overline{SYS\_HWRST}$ Deasserted after V_{DD_INT}, V_{DD_EXT}, V_{DD_DMC}, V_{DD_USB}, V_{DD_TD}, and SYS_CLKIN are Stable and Within Specification	$11 \times t_{CKIN}$		ns

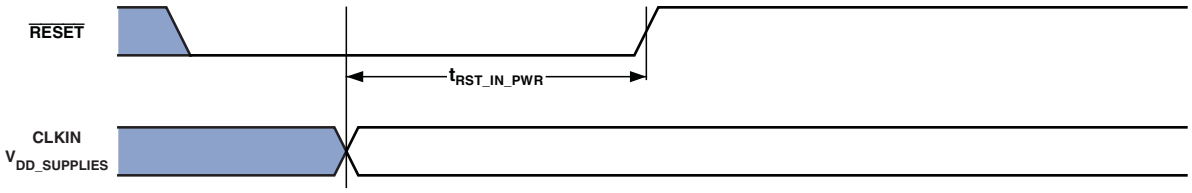


Figure 11. Power-Up Reset Timing

DDR2 SDRAM Write Cycle Timing

Table 38. DDR2 SDRAM Write Cycle Timing, V_{DD_DMC} Nominal 1.8 V

Parameter		250 MHz[1] Min	250 MHz[1] Max	Unit
Switching Characteristics				
t_{DQSS}[2]	DMC0_DQS Latching Rising Transitions to Associated Clock Edges	–0.15	0.15	t_{CK}
t_{DS}	Last Data Valid to DMC0_DQS Delay	0.15		ns
t_{DH}	DMC0_DQS to First Data Invalid Delay	0.3		ns
t_{DSS}	DMC0_DQS Falling Edge to Clock Setup Time	0.25		t_{CK}
t_{DSH}	DMC0_DQS Falling Edge Hold Time From DMC0_CK	0.25		t_{CK}
t_{DQSH}	DMC0_DQS Input High Pulse Width	0.35		t_{CK}
t_{DQSL}	DMC0_DQS Input Low Pulse Width	0.35		t_{CK}
t_{WPRE}	Write Preamble	0.35		t_{CK}
t_{WPST}	Write Postamble	0.4		t_{CK}
t_{IPW}	Address and Control Output Pulse Width	0.6		t_{CK}
t_{DIPW}	DMC0_DQ and DMC0_DM Output Pulse Width	0.35		t_{CK}

[1] In order to ensure proper operation of the DDR2, all the DDR2 guidelines have to be strictly followed.
[2] Write command to first DMC0_DQS delay = WL × t_{CK} + t_{DQSS}.

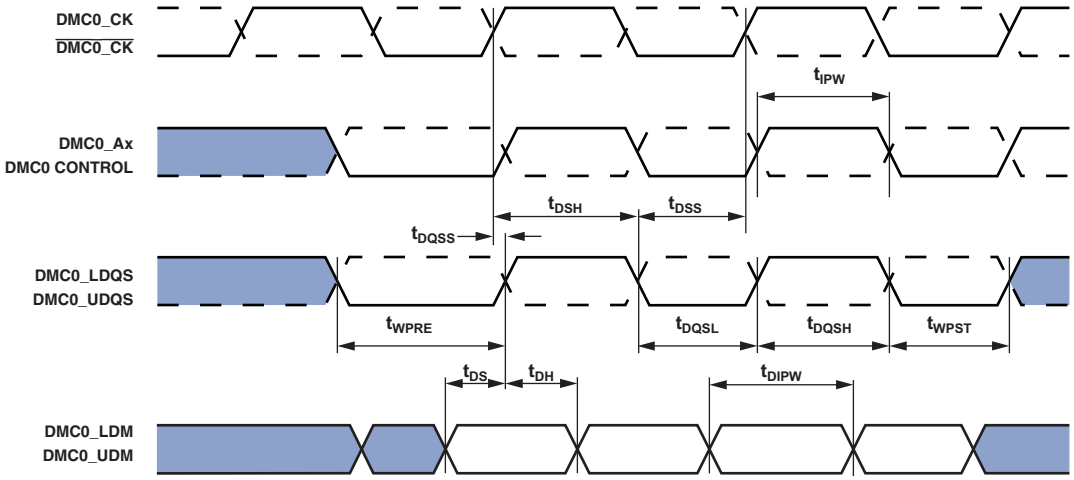
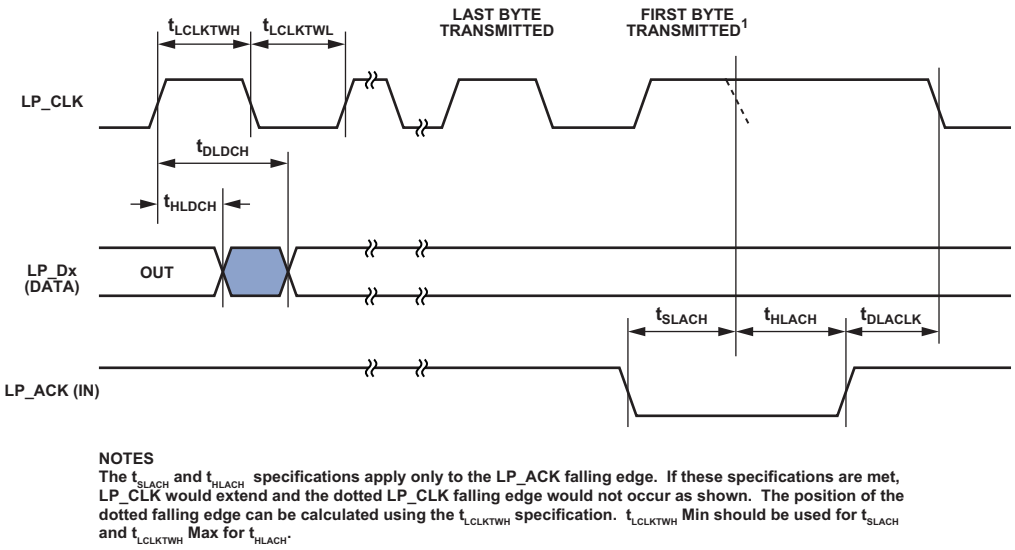


Figure 21. DDR2 SDRAM Controller Output AC Timing

Table 45. Link Ports—Transmit

Parameter		V_{DD_EXT} 1.8 V Nominal Min	V_{DD_EXT} 1.8 V Nominal Max	V_{DD_EXT} 3.3 V Nominal Min	V_{DD_EXT} 3.3 V Nominal Max	Unit
Timing Requirements						
t_{SLACH}	LP_ACK Setup Before LP_CLK Low	$2 \times t_{SCLK0} + 17.5$		$2 \times t_{SCLK0} + 13.5$		ns
t_{HLACH}	LP_ACK Hold After LP_CLK Low	0		0		ns
Switching Characteristics						
t_{DLDCH}	Data Delay After LP_CLK High		2.5		2.5	ns
t_{HLDCH}	Data Hold After LP_CLK High	–1.5		–1.5		ns
$t_{LCLKTWL}$[1]	LP_CLK Width Low	$0.4 \times t_{LCLKTPROG}$	$0.6 \times t_{LCLKTPROG}$	$0.4 \times t_{LCLKTPROG}$	$0.6 \times t_{LCLKTPROG}$	ns
$t_{LCLKTWH}$[1]	LP_CLK Width High	$0.4 \times t_{LCLKTPROG}$	$0.6 \times t_{LCLKTPROG}$	$0.4 \times t_{LCLKTPROG}$	$0.6 \times t_{LCLKTPROG}$	ns
t_{LCLKTW}[1]	LP_CLK Period	$t_{LCLKTPROG} - 1.2$		$t_{LCLKTPROG} - 1.2$		ns
t_{DLACLK}	LP_CLK Low Delay After LP_ACK High	$t_{SCLK0} + 4$	$(2 \times t_{SCLK0}) + t_{LCLK} + 10$	$t_{SCLK0} + 4$	$(2 \times t_{SCLK0}) + t_{LCLK} + 10$	ns

[1] See Table 17 on Page 53 in Clock Related Operating Conditions for details on the minimum period that may be programmed for $t_{LCLKTPROG}$.



NOTES
The t_{SLACH} and t_{HLACH} specifications apply only to the LP_ACK falling edge. If these specifications are met, LP_CLK would extend and the dotted LP_CLK falling edge would not occur as shown. The position of the dotted falling edge can be calculated using the $t_{LCLKTWH}$ specification. $t_{LCLKTWH}$ Min should be used for t_{SLACH} and $t_{LCLKTWH}$ Max for t_{HLACH}.

Figure 35. Link Ports—Transmit

Serial Ports

To determine whether communication is possible between two devices at clock speed n, the following specifications must be confirmed: 1) frame sync delay and frame sync setup and hold, 2) data delay and data setup and hold, and 3) serial clock (SPT_CLK) width. In Figure 36 either the rising edge or the falling edge of SPT_CLK (external or internal) can be used as the active sampling edge.

When externally generated the SPORT clock is called $f_{SPTCLKEXT}$:

$$t_{SPTCLKEXT} = \frac{1}{f_{SPTCLKEXT}}$$

When internally generated, the programmed SPORT clock ($f_{SPTCLKPROG}$) frequency in MHz is set by the following equation where CLKDIV is a field in the SPORT_DIV register that can be set from 0 to 65535:

$$f_{SPTCLKPROG} = \frac{f_{SCLK1}}{(CLKDIV + 1)}$$

$$t_{SPTCLKPROG} = \frac{1}{f_{SPTCLKPROG}}$$

Table 46. Serial Ports—External Clock

Parameter		V_{DD_EXT} 1.8 V Nominal Min	V_{DD_EXT} 1.8 V Nominal Max	V_{DD_EXT} 3.3 V Nominal Min	V_{DD_EXT} 3.3 V Nominal Max	Unit
Timing Requirements						
t_{SFSE}	Frame Sync Setup Before SPT_CLK (Externally Generated Frame Sync in either Transmit or Receive Mode)[1]	2		2		ns
t_{HFSE}	Frame Sync Hold After SPT_CLK (Externally Generated Frame Sync in either Transmit or Receive Mode)[1]	2.7		2.7		ns
t_{SDRE}	Receive Data Setup Before Receive SPT_CLK[1]	2		2		ns
t_{HDRE}	Receive Data Hold After SPT_CLK[1]	2.7		2.7		ns
t_{SCLKW}	SPT_CLK Width[2]	$(0.5 \times t_{SPTCLKEXT}) - 1.5$		$(0.5 \times t_{SPTCLKEXT}) - 1.5$		ns
t_{SPTCLK}	SPT_CLK Period[2]	$t_{SPTCLKEXT} - 1.5$		$t_{SPTCLKEXT} - 1.5$		ns
Switching Characteristics						
t_{DFSE}	Frame Sync Delay After SPT_CLK (Internally Generated Frame Sync in either Transmit or Receive Mode)[3]		19.3		14.5	ns
t_{HOFSE}	Frame Sync Hold After SPT_CLK (Internally Generated Frame Sync in either Transmit or Receive Mode)[3]	2		2		ns
t_{DDTE}	Transmit Data Delay After Transmit SPT_CLK[3]		18.8		14	ns
t_{HDTE}	Transmit Data Hold After Transmit SPT_CLK[3]	2		2		ns

[1] Referenced to sample edge.

[2] This specification indicates the minimum instantaneous width or period that can be tolerated due to duty cycle variation or jitter on the external SPT_CLK. For the external SPT_CLK ideal maximum frequency see the $f_{SPTCLKEXT}$ specification in Table 17 on Page 53 in Clock Related Operating Conditions.

[3] Referenced to drive edge.

Table 47. Serial Ports—Internal Clock

Parameter		V_{DD_EXT} 1.8 V Nominal Min	V_{DD_EXT} 1.8 V Nominal Max	V_{DD_EXT} 3.3 V Nominal Min	V_{DD_EXT} 3.3 V Nominal Max	Unit
Timing Requirements						
t_{SFSI}	Frame Sync Setup Before SPT_CLK (Externally Generated Frame Sync in either Transmit or Receive Mode)[1]	16.8		12		ns
t_{HFSI}	Frame Sync Hold After SPT_CLK (Externally Generated Frame Sync in either Transmit or Receive Mode)[1]	0		–0.5		ns
t_{SDRI}	Receive Data Setup Before SPT_CLK[1]	4.8		3.4		ns
t_{HDRI}	Receive Data Hold After SPT_CLK[1]	1.5		1.5		ns
Switching Characteristics						
t_{DFSI}	Frame Sync Delay After SPT_CLK (Internally Generated Frame Sync in Transmit or Receive Mode)[2]		3.5		3.5	ns
t_{HOFSI}	Frame Sync Hold After SPT_CLK (Internally Generated Frame Sync in Transmit or Receive Mode)[2]	–1		–1		ns
t_{DDTI}	Transmit Data Delay After SPT_CLK[2]		3.5		3.5	ns
t_{HDTI}	Transmit Data Hold After SPT_CLK[2]	–1		–1		ns
t_{SCLKIW}	SPT_CLK Width[3]	$0.5 \times t_{SPTCLKPROG} - 1.5$		$0.5 \times t_{SPTCLKPROG} - 1.5$		ns
t_{SPTCLK}	SPT_CLK Period[3]	$t_{SPTCLKPROG} - 1.5$		$t_{SPTCLKPROG} - 1.5$		ns

[1] Referenced to the sample edge.
[2] Referenced to drive edge.
[3] See Table 17 on Page 53 in Clock Related Operating Conditions for details on the minimum period that may be programmed for $t_{SPTCLKPROG}$.

Table 48. Serial Ports—Enable and Three-State

Parameter		V_{DD_EXT} 1.8 V Nominal Min	V_{DD_EXT} 1.8 V Nominal Max	V_{DD_EXT} 3.3 V Nominal Min	V_{DD_EXT} 3.3 V Nominal Max	Unit
Switching Characteristics						
t_{DDTEN}	Data Enable from External Transmit SPT_CLK[1]	1		1		ns
t_{DDTTE}	Data Disable from External Transmit SPT_CLK[1]		18.8		14	ns
t_{DDTIN}	Data Enable from Internal Transmit SPT_CLK[1]	–1		–1		ns
t_{DDTTI}	Data Disable from Internal Transmit SPT_CLK[1]		2.8		2.8	ns

[1] Referenced to drive edge.

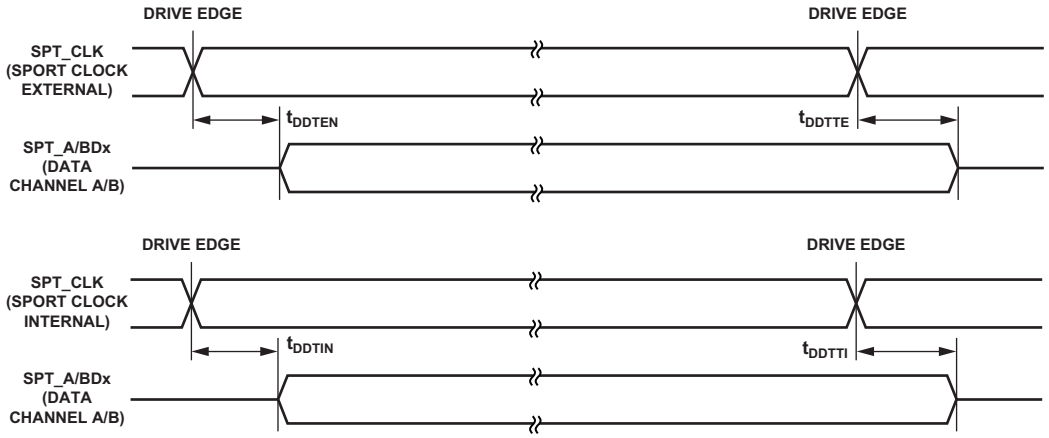


Figure 37. Serial Ports—Enable and Three-State

Table 50. Serial Ports—External Late Frame Sync

Parameter		V_{DD_EXT} 1.8 V Nominal Min	V_{DD_EXT} 1.8 V Nominal Max	V_{DD_EXT} 3.3 V Nominal Min	V_{DD_EXT} 3.3 V Nominal Max	Unit
Switching Characteristics						
$t_{DDTLFSE}$	Data Delay from Late External Transmit Frame Sync or External Receive Frame Sync with MCE = 1, MFD = 0[1]		18.8		14	ns
$t_{DDTENFS}$	Data Enable for MCE = 1, MFD = 0[1]	0.5		0.5		ns

[1] The $t_{DDTLFSE}$ and $t_{DDTENFS}$ parameters apply to left-justified as well as standard serial mode, and MCE = 1, MFD = 0.

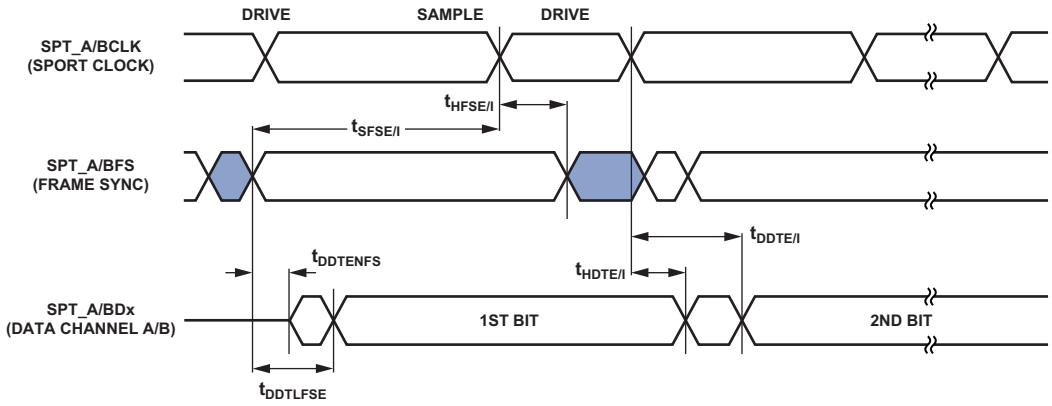


Figure 39. External Late Frame Sync

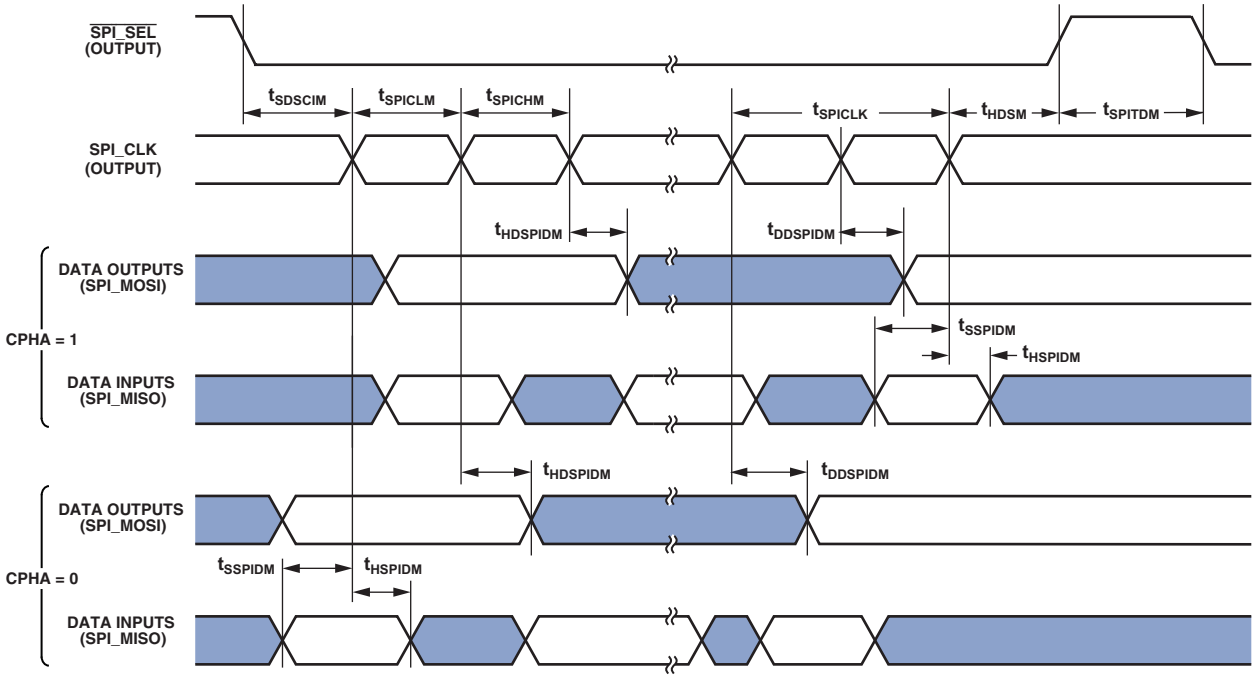


Figure 40. Serial Peripheral Interface (SPI) Port—Master Timing

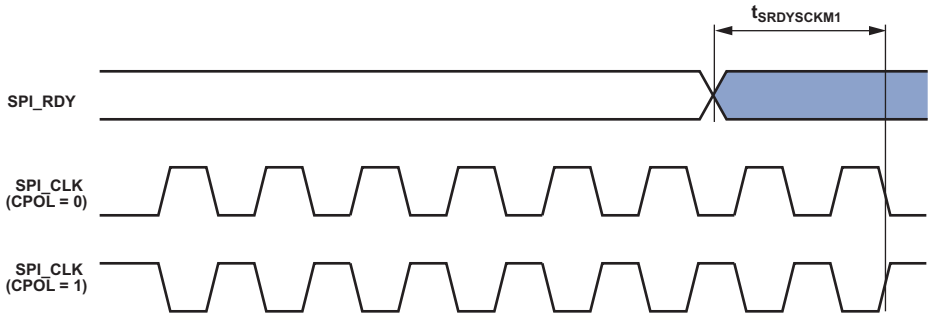


Figure 47. SPI_RDY Setup Before SPI_CLK with CPHA = 1

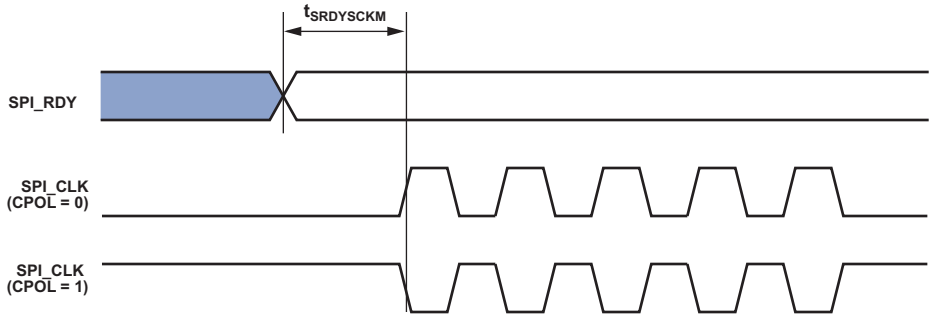


Figure 48. SPI_CLK Switching Diagram after SPI_RDY Assertion, CPHA = x

Table 66. 10/100 Ethernet MAC Controller Timing: RMII Station Management

Parameter[1]		V_{DD_EXT} 1.8 V/3.3 V Nominal Min	Max	Unit
Timing Requirements				
t_{MDIOS}	ETHx_MDIO Input Valid to ETHx_MDC Rising Edge (Setup)	14		ns
t_{MDCIH}	ETHx_MDC Rising Edge to ETHx_MDIO Input Invalid (Hold)	0		ns
Switching Characteristics				
t_{MDCOV}	ETHx_MDC Falling Edge to ETHx_MDIO Output Valid		t_{SCLK0} + 5	ns
t_{MDCOH}	ETHx_MDC Falling Edge to ETHx_MDIO Output Invalid (Hold)	t_{SCLK0} −1		ns

[1] ETHx_MDC/ETHx_MDIO is a 2-wire serial bidirectional port for controlling one or more external PHYs. ETHx_MDC is an output clock whose minimum period is programmable as a multiple of the system clock SCLK0. ETHx_MDIO is a bidirectional data line.

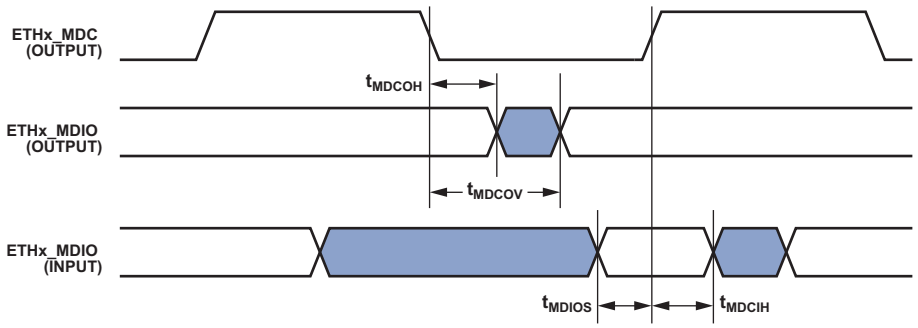


Figure 57. 10/100 Ethernet MAC Controller Timing: RMII Station Management

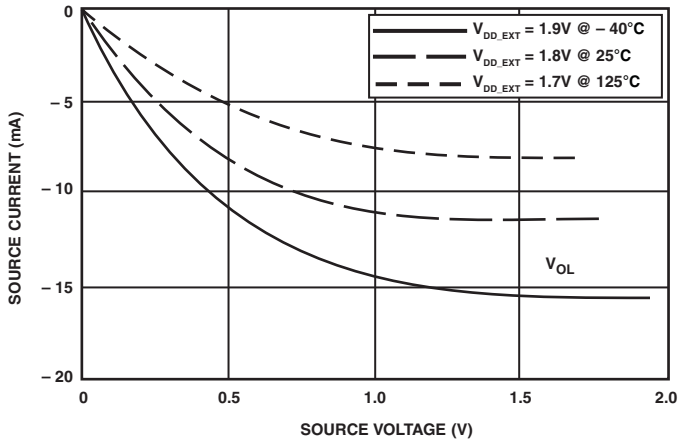


Figure 63. Driver Type D Current (1.8 V V_{DD_EXT})

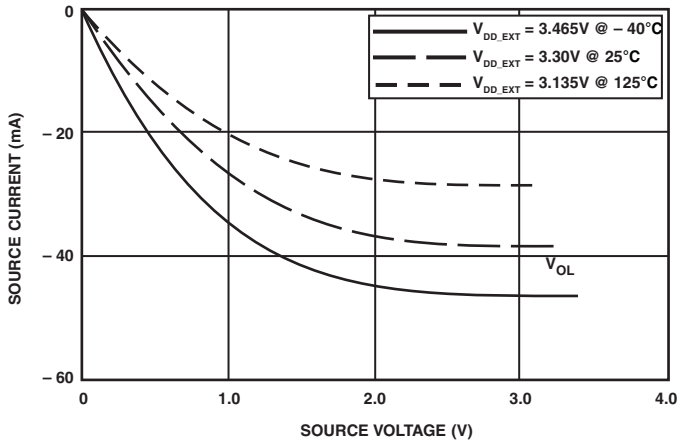


Figure 64. Driver Type D Current (3.3 V V_{DD_EXT})

TEST CONDITIONS

All Timing Requirements appearing in this data sheet were measured under the conditions described in this section. Figure 65 shows the measurement point for AC measurements (except output enable/disable). The measurement point V_{MEAS} is $V_{DDEXT}/2$ or $V_{DDMEM}/2$ for V_{DDEXT}/V_{DDMEM} (nominal) = 1.8 V/ 2.5 V/3.3 V.

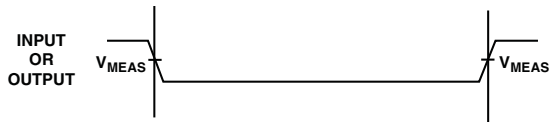


Figure 65. Voltage Reference Levels for AC Measurements (Except Output Enable/Disable)

Output Enable Time Measurement

Output balls are considered to be enabled when they have made a transition from a high impedance state to the point when they start driving.

The output enable time t_{ENA} is the interval from the point when a reference signal reaches a high or low voltage level to the point when the output starts driving as shown on the right side of Figure 66.

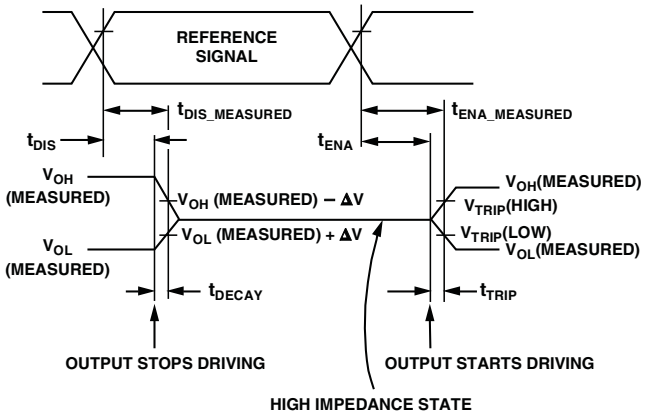


Figure 66. Output Enable/Disable

The time $t_{ENA\_MEASURED}$ is the interval *from when* the reference signal switches *to when* the output voltage reaches V_{TRIP}(high) or V_{TRIP}(low). For V_{DDEXT}/V_{DDMEM} (nominal) = 1.8 V, V_{TRIP} (high) is 1.05 V, and V_{TRIP} (low) is 0.75 V. For V_{DDEXT}/V_{DDMEM} (nominal) = 2.5 V, V_{TRIP} (high) is 1.5 V and V_{TRIP} (low) is 1.0 V. For V_{DDEXT}/V_{DDMEM} (nominal) = 3.3 V, V_{TRIP} (high) is 1.9 V, and V_{TRIP} (low) is 1.4 V. Time t_{TRIP} is the interval *from when* the output starts driving *to when* the output reaches the V_{TRIP}(high) or V_{TRIP}(low) trip voltage.

Time t_{ENA} is calculated as shown in the equation:

$$t_{ENA} = t_{ENA\_MEASURED} - t_{TRIP}$$

If multiple balls (such as the data bus) are enabled, the measurement value is that of the first ball to start driving.

Output Disable Time Measurement

Output balls are considered to be disabled when they stop driving, go into a high impedance state, and start to decay from their output high or low voltage. The output disable time t_{DIS} is the difference between $t_{DIS\_MEASURED}$ and t_{DECAY} as shown on the left side of Figure 66.

$$t_{DIS} = t_{DIS\_MEASURED} - t_{DECAY}$$

The time for the voltage on the bus to decay by ΔV is dependent on the capacitive load C_L and the load current I_L. This decay time can be approximated by the equation:

$$t_{DECAY} = (C_L \Delta V)/I_L$$