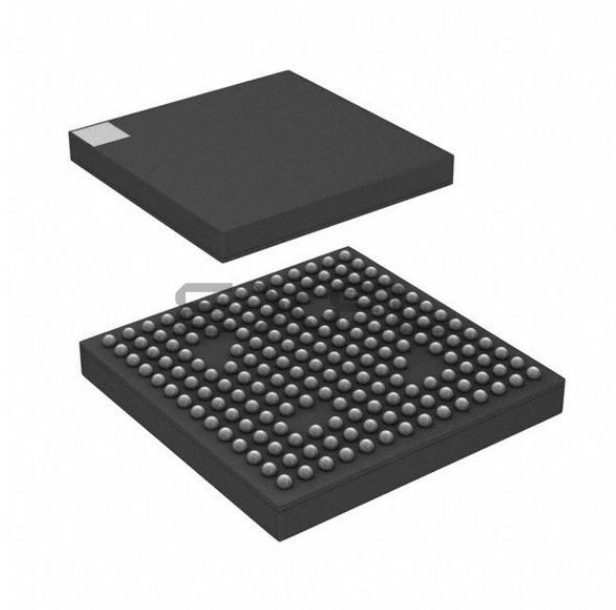


Welcome to E-XFL.COM

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	Blackfin+
Interface	CAN, DSPI, EBI/EMI, I²C, PPI, QSPI, SD/SDIO, SPI, SPORT, UART/USART, USB OTG
Clock Rate	400MHz
Non-Volatile Memory	ROM (512kB)
On-Chip RAM	1MB
Voltage - I/O	1.8V, 3.3V
Voltage - Core	1.10V
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	184-LFBGA, CSPBGA
Supplier Device Package	184-CSPBGA (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adsp-bf707kbcz-4

Static Memory Controller (SMC)

The SMC can be programmed to control up to two blocks of external memories or memory-mapped devices, with very flexible timing parameters. Each block occupies a 8K byte segment regardless of the size of the device used.

Dynamic Memory Controller (DMC)

The DMC includes a controller that supports JESD79-2E compatible double-data-rate (DDR2) SDRAM and JESD209A low-power DDR (LPDDR) SDRAM devices. The DMC PHY features on-die termination on all data and data strobe pins that can be used during reads.

I/O Memory Space

The processor does not define a separate I/O space. All resources are mapped through the flat 32-bit address space. On-chip I/O devices have their control registers mapped into memory-mapped registers (MMRs) at addresses in a region of the 4G byte address space. These are separated into two smaller blocks, one which contains the control MMRs for all core functions, and the other which contains the registers needed for setup and control of the on-chip peripherals outside of the core. The MMRs are accessible only in supervisor mode and appear as reserved space to on-chip peripherals.

Booting

The processor has several mechanisms for automatically loading internal and external memory after a reset. The boot mode is defined by the SYS_BMODE input pins dedicated for this purpose. There are two categories of boot modes. In master boot mode, the processor actively loads data from serial memories. In slave boot modes, the processor receives data from external host devices.

The boot modes are shown in Table 2. These modes are implemented by the SYS_BMODE bits of the reset configuration register and are sampled during power-on resets and software-initiated resets.

Table 2. Boot Modes

SYS_BMODE Setting	Boot Mode
00	No Boot/Idle
01	SPI2 Master
10	SPI2 Slave
11	UART0 Slave

SECURITY FEATURES

The ADSP-BF70x processor supports standards-based hardware-accelerated encryption, decryption, authentication, and true random number generation.

The following hardware-accelerated cryptographic ciphers are supported:

- AES in ECB, CBC, ICM, and CTR modes with 128-, 192-, and 256-bit keys
- DES in ECB and CBC mode with 56-bit key
- 3DES in ECB and CBC mode with 3x 56-bit key

The following hardware-accelerated hash functions are supported:

- SHA-1
- SHA-2 with 224-bit and 256-bit digest
- HMAC transforms for SHA-1 and SHA-2

Public key accelerator is available to offload computation-intensive public key cryptography operations.

Both a hardware-based nondeterministic random number generator and pseudo-random number generator are available. The TRNG also provides HW post-processing to meet NIST requirements of FIPS 140-2, while the PRNG is ANSI X9.31 compliant.

Secure boot is also available with 224-bit elliptic curve digital signatures ensuring integrity and authenticity of the boot stream. Optionally, confidentiality is also ensured through AES-128 encryption.

CAUTION
This product includes security features that can be used to protect embedded nonvolatile memory contents and prevent execution of unauthorized code. When security is enabled on this device (either by the ordering party or the subsequent receiving parties), the ability of Analog Devices to conduct failure analysis on returned devices is limited. Contact Analog Devices for details on the failure analysis limitations for this device.

Secure debug is also employed to allow only trusted users to access the system with debug tools.

PROCESSOR SAFETY FEATURES

The ADSP-BF70x processor has been designed for functional safety applications. While the level of safety is mainly dominated by the system concept, the following primitives are provided by the devices to build a robust safety concept.

Multi-Parity-Bit-Protected L1 Memories

In the processor's L1 memory space, whether SRAM or cache, each word is protected by multiple parity bits to detect the single event upsets that occur in all RAMs. This applies both to L1 instruction and data memory spaces.

ECC-Protected L2 Memories

Error correcting codes (ECC) are used to correct single event upsets. The L2 memory is protected with a single error correct-double error detect (SEC-DED) code. By default ECC is enabled, but it can be disabled on a per-bank basis. Single-bit errors are transparently corrected. Dual-bit errors can issue a

General-Purpose Timers

There is one GP timer unit, and it provides eight general-purpose programmable timers. Each timer has an external pin that can be configured either as a pulse width modulator (PWM) or timer output, as an input to clock the timer, or as a mechanism for measuring pulse widths and periods of external events. These timers can be synchronized to an external clock input on the TIMER_TMRx pins, an external TIMER_CLK input pin, or to the internal SCLK0.

These timer units can be used in conjunction with the UARTs and the CAN controller to measure the width of the pulses in the data stream to provide a software auto-baud detect function for the respective serial channels.

The GP timers can generate interrupts to the processor core, providing periodic events for synchronization to either the system clock or to external signals. Timer events can also trigger other peripherals through the TRU (for instance, to signal a fault). Each timer may also be started and/or stopped by any TRU master without core intervention.

Core Timer

The processor core also has its own dedicated timer. This extra timer is clocked by the internal processor clock and is typically used as a system tick clock for generating periodic operating system interrupts.

Watchdog Timer

The core includes a 32-bit timer, which may be used to implement a software watchdog function. A software watchdog can improve system availability by forcing the processor to a known state, through generation of a hardware reset, nonmaskable interrupt (NMI), or general-purpose interrupt, if the timer expires before being reset by software. The programmer initializes the count value of the timer, enables the appropriate interrupt, then enables the timer. Thereafter, the software must reload the counter before it counts down to zero from the programmed value. This protects the system from remaining in an unknown state where software that would normally reset the timer has stopped running due to an external noise condition or software error.

After a reset, software can determine if the watchdog was the source of the hardware reset by interrogating a status bit in its timer control register that is set only upon a watchdog-generated reset.

Serial Ports (SPORTs)

Two synchronous serial ports (comprised of four half-SPORTs) provide an inexpensive interface to a wide variety of digital and mixed-signal peripheral devices such as Analog Devices' audio codecs, ADCs, and DACs. Each half-SPORT is made up of two data lines, a clock, and frame sync. The data lines can be programmed to either transmit or receive and each data line has a dedicated DMA channel.

Serial port data can be automatically transferred to and from on-chip memory/external memory through dedicated DMA channels. Each of the serial ports can work in conjunction with another serial port to provide TDM support. In this configuration, one SPORT provides two transmit signals while the other SPORT provides the two receive signals. The frame sync and clock are shared.

Serial ports operate in six modes:

- Standard DSP serial mode
- Multichannel (TDM) mode
- I^2S mode
- Packed I^2S mode
- Left-justified mode
- Right-justified mode

General-Purpose Counters

A 32-bit counter is provided that can operate in general-purpose up/down count modes and can sense 2-bit quadrature or binary codes as typically emitted by industrial drives or manual thumbwheels. Count direction is either controlled by a level-sensitive input pin or by two edge detectors.

A third counter input can provide flexible zero marker support and can alternatively be used to input the push-button signal of thumbwheel devices. All three pins have a programmable debouncing circuit.

Internal signals forwarded to a GP timer enable this timer to measure the intervals between count events. Boundary registers enable auto-zero operation or simple system warning by interrupts when programmed count values are exceeded.

Parallel Peripheral Interface (PPI)

The processor provides a parallel peripheral interface (PPI) that supports data widths up to 18 bits. The PPI supports direct connection to TFT LCD panels, parallel analog-to-digital and digital-to-analog converters, video encoders and decoders, image sensor modules, and other general-purpose peripherals.

The following features are supported in the PPI module:

- Programmable data length: 8 bits, 10 bits, 12 bits, 14 bits, 16 bits, and 18 bits per clock.
- Various framed, non-framed, and general-purpose operating modes. Frame syncs can be generated internally or can be supplied by an external device.
- ITU-656 status word error detection and correction for ITU-656 receive modes and ITU-656 preamble and status word decode.
- Optional packing and unpacking of data to/from 32 bits from/to 8 bits, 16 bits and 24 bits. If packing/unpacking is enabled, endianness can be configured to change the order of packing/unpacking of bytes/words.
- RGB888 can be converted to RGB666 or RGB565 for transmit modes.
- Various de-interleaving/interleaving modes for receiving/transmitting 4:2:2 YCrCb data.
- Configurable LCD data enable (DEN) output available on Frame Sync 3.

Serial Peripheral Interface (SPI) Ports

The processors have three industry-standard SPI-compatible ports that allow it to communicate with multiple SPI-compatible devices.

The baseline SPI peripheral is a synchronous, four-wire interface consisting of two data pins, one device select pin, and a gated clock pin. The two data pins allow full-duplex operation to other SPI-compatible devices. An additional two (optional) data pins are provided to support quad SPI operation. Enhanced modes of operation such as flow control, fast mode, and dual I/O mode (DIOM) are also supported. In addition, a direct memory access (DMA) mode allows for transferring several words with minimal CPU interaction.

With a range of configurable options, the SPI ports provide a glueless hardware interface with other SPI-compatible devices in master mode, slave mode, and multimaster environments. The SPI peripheral includes programmable baud rates, clock phase, and clock polarity. The peripheral can operate in a multimaster environment by interfacing with several other devices, acting as either a master device or a slave device. In a multimaster environment, the SPI peripheral uses open-drain outputs to avoid data bus contention. The flow control features enable slow slave devices to interface with fast master devices by providing an SPI Ready pin which flexibly controls the transfers.

The SPI port's baud rate and clock phase/polarities are programmable, and it has integrated DMA channels for both transmit and receive data streams.

SPI Host Port (SPIHP)

The processor includes one SPI host port which may be used in conjunction with any available SPI port to enhance its SPI slave mode capabilities. The SPIHP allows a SPI host device access to memory-mapped resources of the processor through a SPI SRAM/FLASH style protocol. The following features are included:

- Direct read/write of memory and memory-mapped registers
- Support for pre-fetch for faster reads
- Support for SPI controllers that implement hardware-based SPI memory protocol
- Error capture and reporting for protocol errors, bus errors, and over/underflow

UART Ports

The processor provides two full-duplex universal asynchronous receiver/transmitter (UART) ports, which are fully compatible with PC-standard UARTs. Each UART port provides a simplified UART interface to other peripherals or hosts, supporting full-duplex, DMA-supported, asynchronous transfers of serial data. A UART port includes support for five to eight data bits, and none, even, or odd parity. Optionally, an additional address bit can be transferred to interrupt only addressed nodes in multi-drop bus (MDB) systems. A frame is terminated by a configurable number of stop bits.

The UART ports support automatic hardware flow control through the clear to send (CTS) input and request to send (RTS) output with programmable assertion FIFO levels.

To help support the local interconnect network (LIN) protocols, a special command causes the transmitter to queue a break command of programmable bit length into the transmit buffer. Similarly, the number of stop bits can be extended by a programmable inter-frame space.

The capabilities of the UARTs are further extended with support for the Infrared Data Association (IrDA®) serial infrared physical layer link specification (SIR) protocol.

2-Wire Controller Interface (TWI)

The processor includes a 2-wire interface (TWI) module for providing a simple exchange method of control data between multiple devices. The TWI module is compatible with the widely used I^2C bus standard. The TWI module offers the capabilities of simultaneous master and slave operation and support for both 7-bit addressing and multimedia data arbitration. The TWI interface utilizes two pins for transferring clock (TWI_SCL) and data (TWI_SDA) and supports the protocol at speeds up to 400k bits/sec. The TWI interface pins are compatible with 5 V logic levels.

Additionally, the TWI module is fully compatible with serial camera control bus (SCCB) functionality for easier control of various CMOS camera sensor devices.

Mobile Storage Interface (MSI)

The mobile storage interface (MSI) controller acts as the host interface for multimedia cards (MMC), secure digital memory cards (SD), and secure digital input/output cards (SDIO). The following list describes the main features of the MSI controller:

- Support for a single MMC, SD memory, and SDIO card
- Support for 1-bit and 4-bit SD modes
- Support for 1-bit, 4-bit, and 8-bit MMC modes
- Support for eMMC 4.5 embedded NAND flash devices
- Support for power management and clock control
- An eleven-signal external interface with clock, command, optional interrupt, and up to eight data lines
- Card interface clock generation from SCLK0 or SCLK1
- SDIO interrupt and read wait features

Controller Area Network (CAN)

A CAN controller implements the CAN 2.0B (active) protocol. This protocol is an asynchronous communications protocol used in both industrial and automotive control systems. The CAN protocol is well suited for control applications due to its capability to communicate reliably over a network. This is because the protocol incorporates CRC checking, message error tracking, and fault node confinement.

Table 3. Clock Dividers

Clock Source	Divider (if Available on SYS_CLKOUT)
CCLK (Core Clock)	By 16
SYSCLK (System Clock)	By 8
SCLK0 (System Clock, All Peripherals not Covered by SCLK1)	Not available on SYS_CLKOUT
SCLK1 (System Clock for Crypto Engines and MDMA)	By 8
DCLK (LPDDR/DDR2 Clock)	By 8
OCLK (Output Clock)	Programmable
CLKBUF	None, direct from SYS_CLKIN

Power Management

As shown in Table 4, the processor supports multiple power domains, which maximizes flexibility while maintaining compliance with industry standards and conventions. There are no sequencing requirements for the various power domains, but all domains must be powered according to the appropriate Specifications table for processor operating conditions; even if the feature/peripheral is not used.

Table 4. Power Domains

Power Domain	V_{DD} Range
All Internal Logic	V_{DD_INT}
DDR2/LPDDR	V_{DD_DMC}
USB	V_{DD_USB}
OTP Memory	V_{DD_OTP}
HADC	V_{DD_HADC}
RTC	V_{DD_RTC}
All Other I/O (Includes SYS, JTAG, and Ports Pins)	V_{DD_EXT}

The dynamic power management feature of the processor allows the processor's core clock frequency (f_{CCLK}) to be dynamically controlled.

The power dissipated by a processor is largely a function of its clock frequency and the square of the operating voltage. For example, reducing the clock frequency by 25% results in a 25% reduction in dynamic power dissipation.

See Table 5 for a summary of the power settings for each mode.

Full-On Operating Mode—Maximum Performance

In the full-on mode, the PLL is enabled and is not bypassed, providing capability for maximum operational frequency. This is the power-up default execution state in which maximum performance can be achieved. The processor core and all enabled peripherals run at full speed.

Deep Sleep Operating Mode—Maximum Dynamic Power Savings

The deep sleep mode maximizes dynamic power savings by disabling the clocks to the processor core and to all synchronous peripherals. Asynchronous peripherals may still be running but cannot access internal resources or external memory.

Table 5. Power Settings

Mode/State	PLL	PLL Bypassed	f_{CCLK}	f_{SYSCLK}, f_{DCLK}, f_{SCLK0}, f_{SCLK1}	Core Power
Full On	Enabled	No	Enabled	Enabled	On
Deep Sleep	Disabled	—	Disabled	Disabled	On
Hibernate	Disabled	—	Disabled	Disabled	Off

Hibernate State—Maximum Static Power Savings

The hibernate state maximizes static power savings by disabling the voltage and clocks to the processor core and to all of the peripherals. This setting signals the external voltage regulator supplying the VDD_INT pins to shut off using the SYS_EXTWAKE signal, which provides the lowest static power dissipation.

Any critical information stored internally (for example, memory contents, register contents, and other information) must be written to a nonvolatile storage device (or self-refreshed DRAM) prior to removing power if the processor state is to be preserved.

Because the V_{DD_EXT} pins can still be supplied in this mode, all of the external pins three-state, unless otherwise specified. This allows other devices that may be connected to the processor to still have power applied without drawing unwanted current.

Reset Control Unit

Reset is the initial state of the whole processor or the core and is the result of a hardware- or software-triggered event. In this state, all control registers are set to their default values and functional units are idle. Exiting a full system reset starts with the core being ready to boot.

The reset control unit (RCU) controls how all the functional units enter and exit reset. Differences in functional requirements and clocking constraints define how reset signals are generated. Programs must guarantee that none of the reset functions puts the system into an undefined state or causes resources to stall. This is particularly important when the core is reset (programs must ensure that there is no pending system activity involving the core when it is being reset).

From a system perspective, reset is defined by both the reset target and the reset source described as follows in the following list.

Table 6. ADSP-BF70x Detailed Signal Descriptions (Continued)

Port Name	Direction	Description
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.
SPT_BD1	I/O	**Channel B 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_BFS	I/O	**Channel B Frame Sync.** The frame sync pulse initiates shifting of serial data. This signal is either generated internally or externally.
SPT_BTDV	Output	**Channel B 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.
SYS_BMODEn	Input	**Boot Mode Control n.** Selects the boot mode of the processor.
SYS_CLKIN	Input	**Clock/Crystal Input.** Connect to an external clock source or crystal.
SYS_CLKOUT	Output	**Processor Clock Output.** Outputs internal clocks. Clocks may be divided down. See the CGU chapter of the HRM for more details.
SYS_EXTWAKE	Output	**External Wake Control.** Drives low during hibernate and high all other times. Typically connected to the enable input of the voltage regulator controlling the VDD_INT supply.
$\overline{\text{SYS\_FAULT}}$	I/O	**Active-Low Fault Output.** Indicates internal faults or senses external faults depending on the operating mode.
$\overline{\text{SYS\_HWRST}}$	Input	**Processor Hardware Reset Control.** Resets the device when asserted.
$\overline{\text{SYS\_NMI}}$	Input	**Non-maskable Interrupt.** See the processor hardware and programming references for more details.
$\overline{\text{SYS\_RESOUT}}$	Output	**Reset Output.** Indicates that the device is in the reset or hibernate state.
SYS_WAKEn	Input	**Power Saving Mode Wakeup n.** Wake-up source input for deep sleep and/or hibernate mode.
SYS_XTAL	Output	**Crystal Output.** Drives an external crystal. Must be left unconnected if an external clock is driving CLKIN.
JTG_SWCLK	I/O	**Serial Wire Clock.** Clocks data into and out of the target during debug.
JTG_SWDIO	I/O	**Serial Wire DIO.** Sends and receives serial data to and from the target during debug.
JTG_SWO	Output	**Serial Wire Out.** Provides trace data to the emulator.
JTG_TCK	Input	**JTAG Clock.** JTAG test access port clock.
JTG_TDI	Input	**JTAG Serial Data In.** JTAG test access port data input.
JTG_TDO	Output	**JTAG Serial Data Out.** JTAG test access port data output.
JTG_TMS	Input	**JTAG Mode Select.** JTAG test access port mode select.
$\overline{\text{JTG\_TRST}}$	Input	**JTAG Reset.** JTAG test access port reset.
TM_ACIn	Input	**Alternate Capture Input n.** Provides an additional input for WIDCAP, WATCHDOG, and PININT modes.
TM_ACLKn	Input	**Alternate Clock n.** Provides an additional time base for use by an individual timer.
TM_CLK	Input	**Clock.** Provides an additional global time base for use by all the GP timers.
TM_TMRn	I/O	**Timer n.** The main input/output signal for each timer.
TRACE_CLK	Output	**Trace Clock.** Clock output.
TRACE_Dnn	Output	**Trace Data n.** Unidirectional data bus.
TWI_SCL	I/O	**Serial Clock.** Clock output when master, clock input when slave.
TWI_SDA	I/O	**Serial Data.** Receives or transmits data.
$\overline{\text{UART\_CTS}}$	Input	**Clear to Send.** Flow control signal.
$\overline{\text{UART\_RTS}}$	Output	**Request to Send.** Flow control signal.
$\overline{\text{UART\_RX}}$	Input	**Receive.** Receive input. Typically connects to a transceiver that meets the electrical requirements of the device being communicated with.
$\overline{\text{UART\_TX}}$	Output	**Transmit.** Transmit output. Typically connects to a transceiver that meets the electrical requirements of the device being communicated with.
USB_CLKIN	Input	**Clock/Crystal Input.** This clock input is multiplied by a PLL to form the USB clock. See data sheet specifications for frequency/tolerance information.

Table 11. ADSP-BF70x 12 mm × 12 mm 88-Lead LFCSP (QFN) Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
SMC0_D11	SMC0 Data 11	B	PB_11
SMC0_D12	SMC0 Data 12	B	PB_12
SMC0_D13	SMC0 Data 13	B	PB_13
SMC0_D14	SMC0 Data 14	B	PB_14
SMC0_D15	SMC0 Data 15	B	PB_15
SPI0_CLK	SPI0 Clock	B	PB_00
SPI0_CLK	SPI0 Clock	C	PC_04
SPI0_D2	SPI0 Data 2	B	PB_03
SPI0_D2	SPI0 Data 2	C	PC_08
SPI0_D3	SPI0 Data 3	B	PB_07
SPI0_D3	SPI0 Data 3	C	PC_09
SPI0_MISO	SPI0 Master In, Slave Out	B	PB_01
SPI0_MISO	SPI0 Master In, Slave Out	C	PC_06
SPI0_MOSI	SPI0 Master Out, Slave In	B	PB_02
SPI0_MOSI	SPI0 Master Out, Slave In	C	PC_07
SPI0_RDY	SPI0 Ready	A	PA_06
$\overline{\text{SPI0\_SEL1}}$	SPI0 Slave Select Output 1	A	PA_05
$\overline{\text{SPI0\_SEL2}}$	SPI0 Slave Select Output 2	A	PA_06
$\overline{\text{SPI0\_SEL4}}$	SPI0 Slave Select Output 4	B	PB_04
$\overline{\text{SPI0\_SEL5}}$	SPI0 Slave Select Output 5	B	PB_05
$\overline{\text{SPI0\_SEL6}}$	SPI0 Slave Select Output 6	B	PB_06
$\overline{\text{SPI0\_SS}}$	SPI0 Slave Select Input	A	PA_05
SPI1_CLK	SPI1 Clock	A	PA_00
SPI1_MISO	SPI1 Master In, Slave Out	A	PA_01
SPI1_MOSI	SPI1 Master Out, Slave In	A	PA_02
SPI1_RDY	SPI1 Ready	A	PA_03
$\overline{\text{SPI1\_SEL1}}$	SPI1 Slave Select Output 1	A	PA_04
$\overline{\text{SPI1\_SEL2}}$	SPI1 Slave Select Output 2	A	PA_03
$\overline{\text{SPI1\_SEL3}}$	SPI1 Slave Select Output 3	C	PC_10
$\overline{\text{SPI1\_SEL4}}$	SPI1 Slave Select Output 4	A	PA_14
$\overline{\text{SPI1\_SS}}$	SPI1 Slave Select Input	A	PA_04
SPI2_CLK	SPI2 Clock	B	PB_10
SPI2_D2	SPI2 Data 2	B	PB_13
SPI2_D3	SPI2 Data 3	B	PB_14
SPI2_MISO	SPI2 Master In, Slave Out	B	PB_11
SPI2_MOSI	SPI2 Master Out, Slave In	B	PB_12
SPI2_RDY	SPI2 Ready	A	PA_04
$\overline{\text{SPI2\_SEL1}}$	SPI2 Slave Select Output 1	B	PB_15
$\overline{\text{SPI2\_SEL2}}$	SPI2 Slave Select Output 2	B	PB_08
$\overline{\text{SPI2\_SEL3}}$	SPI2 Slave Select Output 3	B	PB_09
$\overline{\text{SPI2\_SS}}$	SPI2 Slave Select Input	B	PB_15
SPT0_ACLK	SPORT0 Channel A Clock	A	PA_13
SPT0_ACLK	SPORT0 Channel A Clock	C	PC_09
SPT0_AD0	SPORT0 Channel A Data 0	A	PA_14
SPT0_AD0	SPORT0 Channel A Data 0	C	PC_08
SPT0_AD1	SPORT0 Channel A Data 1	C	PC_00

Table 11. ADSP-BF70x 12 mm × 12 mm 88-Lead LFCSP (QFN) Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
TM0_ACLK1	TIMER0 Alternate Clock 1	C	PC_10
TM0_ACLK2	TIMER0 Alternate Clock 2	C	PC_09
TM0_ACLK3	TIMER0 Alternate Clock 3	B	PB_00
TM0_ACLK4	TIMER0 Alternate Clock 4	B	PB_10
TM0_ACLK5	TIMER0 Alternate Clock 5	A	PA_14
TM0_ACLK6	TIMER0 Alternate Clock 6	B	PB_04
TM0_CLK	TIMER0 Clock	B	PB_06
TM0_TMR0	TIMER0 Timer 0	A	PA_05
TM0_TMR1	TIMER0 Timer 1	A	PA_06
TM0_TMR2	TIMER0 Timer 2	A	PA_07
TM0_TMR3	TIMER0 Timer 3	C	PC_05
TM0_TMR4	TIMER0 Timer 4	A	PA_09
TM0_TMR5	TIMER0 Timer 5	A	PA_10
TM0_TMR6	TIMER0 Timer 6	A	PA_11
TM0_TMR7	TIMER0 Timer 7	A	PA_04
TRACE0_CLK	TPIU0 Trace Clock	B	PB_10
TRACE0_D00	TPIU0 Trace Data 0	B	PB_15
TRACE0_D01	TPIU0 Trace Data 1	B	PB_14
TRACE0_D02	TPIU0 Trace Data 2	B	PB_13
TRACE0_D03	TPIU0 Trace Data 3	B	PB_12
TRACE0_D04	TPIU0 Trace Data 4	B	PB_11
TRACE0_D05	TPIU0 Trace Data 5	A	PA_02
TRACE0_D06	TPIU0 Trace Data 6	A	PA_01
TRACE0_D07	TPIU0 Trace Data 7	A	PA_00
TWI0_SCL	TWI0 Serial Clock	Not Muxed	TWI0_SCL
TWI0_SDA	TWI0 Serial Data	Not Muxed	TWI0_SDA
$\overline{\text{UART0\_CTS}}$	UART0 Clear to Send	C	PC_03
$\overline{\text{UART0\_RTS}}$	UART0 Request to Send	C	PC_02
$\overline{\text{UART0\_RX}}$	UART0 Receive	B	PB_09
$\overline{\text{UART0\_TX}}$	UART0 Transmit	B	PB_08
$\overline{\text{UART1\_CTS}}$	UART1 Clear to Send	B	PB_14
$\overline{\text{UART1\_RTS}}$	UART1 Request to Send	B	PB_13
$\overline{\text{UART1\_RX}}$	UART1 Receive	C	PC_01
$\overline{\text{UART1\_TX}}$	UART1 Transmit	C	PC_00
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
USB0_XTAL	USB0 Crystal	Not Muxed	USB0_XTAL
VDD_EXT	External VDD	Not Muxed	VDD_EXT
VDD_INT	Internal VDD	Not Muxed	VDD_INT
VDD_OTP	VDD for OTP	Not Muxed	VDD_OTP
VDD_RTC	VDD for RTC	Not Muxed	VDD_RTC
VDD_USB	VDD for USB	Not Muxed	VDD_USB

GPIO MULTIPLEXING FOR 12 mm × 12 mm 88-LEAD LFCSP (QFN)

Table 12 through Table 14 identify the pin functions that are multiplexed on the general-purpose I/O pins of the 12 mm × 12 mm 88-Lead LFCSP (QFN) package.

Table 12. Signal Multiplexing for Port A

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function Input Tap
PA_00	SPI1_CLK		TRACE0_D07	SMC0_ABE0	
PA_01	SPI1_MISO		TRACE0_D06	SMC0_ABE1	
PA_02	SPI1_MOSI		TRACE0_D05	SMC0_AMS1	
PA_03	SPI1_SEL2	SPI1_RDY		SMC0_ARDY	
PA_04	SPI1_SEL1	TM0_TMR7	SPI2_RDY	SMC0_A08	SPI1_SS
PA_05	TM0_TMR0	SPI0_SEL1		SMC0_A07	SPI0_SS
PA_06	TM0_TMR1	SPI0_SEL2	SPI0_RDY	SMC0_A06	
PA_07	TM0_TMR2	SPT1_BTDV	SPT1_ATDV	SMC0_A05	CNT0_DG
PA_08	PPI0_D11	MSI0_CD	SPT1_ACLK	SMC0_A01	
PA_09	PPI0_D10	TM0_TMR4	SPT1_AFS	SMC0_A02	
PA_10	PPI0_D09	TM0_TMR5	SPT1_AD0	SMC0_A03	
PA_11	PPI0_D08	TM0_TMR6	SPT1_AD1	SMC0_A04	
PA_12	PPI0_FS1	CAN1_RX	SPT0_AFS	SMC0_AOE	TM0_ACI6/SYS_WAKE4
PA_13	PPI0_FS2	CAN1_TX	SPT0_ACLK	SMC0_ARE	CNT0_ZM
PA_14	PPI0_CLK	SPI1_SEL4	SPT0_AD0	SMC0_AWE	TM0_ACLK5
PA_15	PPI0_FS3	SPT0_ATDV	SPT0_BTDV	SMC0_AMS0	CNT0_UD

Table 13. Signal Multiplexing for Port B

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function Input Tap
PB_00	PPI0_D07	SPT1_BCLK	SPI0_CLK	SMC0_D07	TM0_ACLK3
PB_01	PPI0_D06	SPT1_BFS	SPI0_MISO	SMC0_D06	TM0_ACI1
PB_02	PPI0_D05	SPT1_BD0	SPI0_MOSI	SMC0_D05	
PB_03	PPI0_D04	SPT1_BD1	SPI0_D2	SMC0_D04	
PB_04	PPI0_D03	SPT0_BCLK	SPI0_SEL4	SMC0_D03	TM0_ACLK6
PB_05	PPI0_D02	SPT0_BD0	SPI0_SEL5	SMC0_D02	
PB_06	PPI0_D01	SPT0_BFS	SPI0_SEL6	SMC0_D01	TM0_CLK
PB_07	PPI0_D00	SPT0_BD1	SPI0_D3	SMC0_D00	SYS_WAKE0
PB_08	UART0_TX	PPI0_D16	SPI2_SEL2	SMC0_D08	SYS_WAKE1
PB_09	UART0_RX	PPI0_D17	SPI2_SEL3	SMC0_D09	TM0_ACI3
PB_10	SPI2_CLK		TRACE0_CLK	SMC0_D10	TM0_ACLK4
PB_11	SPI2_MISO		TRACE0_D04	SMC0_D11	
PB_12	SPI2_MOSI		TRACE0_D03	SMC0_D12	SYS_WAKE2
PB_13	SPI2_D2	UART1_RTS	TRACE0_D02	SMC0_D13	
PB_14	SPI2_D3	UART1_CTS	TRACE0_D01	SMC0_D14	
PB_15	SPI2_SEL1		TRACE0_D00	SMC0_D15	SPI2_SS

Table 15. ADSP-BF70x 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	none	none	none	none	none	VDD_EXT	Desc: SPT1 Channel B Data 1 | MSI0 Data 7 Notes: An external pull-up may be required for MSI modes, see the MSI chapter in the hardware reference for details.
PC_14	I/O	A	none	none	none	none	none	VDD_EXT	Desc: SPT1 Channel B Transmit Data Valid | MSI0 eSDIO Interrupt Input Notes: No notes.
RTC0_CLKIN	a	na	none	none	none	none	none	VDD_RTC	Desc: RTC0 Crystal input / external oscillator connection Notes: If RTC is not used, connect to ground.
RTC0_XTAL	a	na	none	none	none	none	none	VDD_RTC	Desc: RTC0 Crystal output Notes: No notes.
SYS_BMODE0	I/O	na	none	none	none	none	none	VDD_EXT	Desc: SYS Boot Mode Control 0 Notes: A pull-down is required for setting to 0 and a pull-up is required for setting to 1.
SYS_BMODE1	I/O	na	none	none	none	none	none	VDD_EXT	Desc: SYS Boot Mode Control 1 Notes: A pull-down is required for setting to 0 and a pull-up is required for setting to 1.
SYS_CLKIN	a	na	none	none	none	none	none	VDD_EXT	Desc: SYS Clock/Crystal Input Notes: No notes.
SYS_CLKOUT	I/O	A	none	none	L	none	none	VDD_EXT	Desc: SYS Processor Clock Output Notes: During reset, SYS_CLKOUT drives out SYS_CLKIN Frequency.
SYS_EXTWAKE	I/O	A	none	none	H	none	L	VDD_EXT	Desc: SYS External Wake Control Notes: Drives low during hibernate and high all other times including reset.
$\overline{\text{SYS\_FAULT}}$	I/O	A	none	none	none	none	none	VDD_EXT	Desc: SYS Complementary Fault Output Notes: Open drain, requires an external pull-up resistor.
$\overline{\text{SYS\_HWRST}}$	I/O	na	none	none	none	none	none	VDD_EXT	Desc: SYS Processor Hardware Reset Control Notes: Active during reset, must be externally driven.
$\overline{\text{SYS\_NMI}}$	I/O	na	none	none	none	none	none	VDD_EXT	Desc: SYS Non-maskable Interrupt Notes: Requires an external pull-up resistor.
$\overline{\text{SYS\_RESOUT}}$	I/O	A	none	none	L	none	none	VDD_EXT	Desc: SYS Reset Output Notes: Active during reset.
SYS_XTAL	a	na	none	none	none	none	none	VDD_EXT	Desc: SYS Crystal Output Notes: Leave unconnected if an oscillator is used to provide SYS_CLKIN. Active during reset. State during hibernate is controlled by DPM_HIB_DIS.

ELECTRICAL CHARACTERISTICS

Parameter		Test Conditions/Comments	Min	Typ	Max	Unit
V_{OH}[1]	High Level Output Voltage	V_{DD_EXT} = 1.7 V, I_{OH} = –1.0 mA	0.8 × V_{DD_EXT}			V
V_{OH}[1]	High Level Output Voltage	V_{DD_EXT} = 3.13 V, I_{OH} = –2.0 mA	0.9 × V_{DD_EXT}			V
V_{OH_DDR2}[2]	High Level Output Voltage, DDR2, Programmed Impedance = 34 Ω	V_{DD_DMC} = 1.70 V, I_{OH} = –7.1 mA	V_{DD_DMC} – 0.320			V
V_{OH_DDR2}[2]	High Level Output Voltage, DDR2, Programmed Impedance = 40 Ω	V_{DD_DMC} = 1.70 V, I_{OH} = –5.8 mA	V_{DD_DMC} – 0.320			V
V_{OH_DDR2}[2]	High Level Output Voltage, DDR2, Programmed Impedance = 50 Ω	V_{DD_DMC} = 1.70 V, I_{OH} = –4.1 mA	V_{DD_DMC} – 0.320			V
V_{OH_DDR2}[2]	High Level Output Voltage, DDR2, Programmed Impedance = 60 Ω	V_{DD_DMC} = 1.70 V, I_{OH} = –3.4 mA	V_{DD_DMC} – 0.320			V
V_{OH_LPDDR}[2]	High Level Output Voltage, LPDDR	V_{DD_DMC} = 1.70 V, I_{OH} = –2.0 mA	V_{DD_DMC} – 0.320			V
V_{OL}[3]	Low Level Output Voltage	V_{DD_EXT} = 1.7 V, I_{OL} = 1.0 mA			0.400	V
V_{OL}[3]	Low Level Output Voltage	V_{DD_EXT} = 3.13 V, I_{OL} = 2.0 mA			0.400	V
V_{OL_DDR2}[2]	Low Level Output Voltage, DDR2, Programmed Impedance = 34 Ω	V_{DD_DMC} = 1.70 V, I_{OL} = 7.1 mA			0.320	V
V_{OL_DDR2}[2]	Low Level Output Voltage, DDR2, Programmed Impedance = 40 Ω	V_{DD_DMC} = 1.70 V, I_{OL} = 5.8 mA			0.320	V
V_{OL_DDR2}[2]	Low Level Output Voltage, DDR2, Programmed Impedance = 50 Ω	V_{DD_DMC} = 1.70 V, I_{OL} = 4.1 mA			0.320	V
V_{OL_DDR2}[2]	Low Level Output Voltage, DDR2, Programmed Impedance = 60 Ω	V_{DD_DMC} = 1.70 V, I_{OL} = 3.4 mA			0.320	V
V_{OL_LPDDR}[2]	Low Level Output Voltage, LPDDR	V_{DD_DMC} = 1.70 V, I_{OL} = 2.0 mA			0.320	V
I_{IH}[4]	High Level Input Current	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 3.47 V			10	µA
$I_{IH\_DMC0\_VREF}$[5]	High Level Input Current	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 3.47 V			1	µA
I_{IH_PD}[6]	High Level Input Current with Pull-down Resistor	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 3.47 V			100	µA
R_{PD}[6]	Internal Pull-down Resistance	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 3.47 V	57		130	kΩ
I_{IL}[7]	Low Level Input Current	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 0 V			10	µA
$I_{IL\_DMC0\_VREF}$[5]	Low Level Input Current	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 0 V			1	µA
I_{IL_PU}[8]	Low Level Input Current with Pull-up Resistor	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 0 V			100	µA
R_{PU}[8]	Internal Pull-up Resistance	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 0 V	53		129	kΩ
I_{IH_USB0}[9]	High Level Input Current	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 3.47 V			10	µA
I_{IL_USB0}[9]	Low Level Input Current	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 0 V			10	µA
I_{OZH}[10]	Three-State Leakage Current	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 3.47 V			10	µA
I_{OZH}[11]	Three-State Leakage Current	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 1.9 V			10	µA
I_{OZL}[12]	Three-State Leakage Current	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 0 V			10	µA
I_{OZH_PD}[13]	Three-State Leakage Current	V_{DD_EXT} = 3.47 V, V_{DD_DMC} = 1.9 V, V_{DD_USB} = 3.47 V, V_{IN} = 3.47 V			100	µA

Total Internal Power Dissipation

Total power dissipation has two components:

1. Static, including leakage current (deep sleep)
2. Dynamic, due to transistor switching characteristics for each clock domain

Many operating conditions can also affect power dissipation, including temperature, voltage, operating frequency, and processor activity. The following equation describes the internal current consumption.

$$I_{DDINT\_TOT} = I_{DDINT\_DEEPSLEEP} + I_{DDINT\_CCLK\_DYN} + I_{DDINT\_PLLCLK\_DYN} + I_{DDINT\_SYSCLK\_DYN} + I_{DDINT\_SCLK0\_DYN} + I_{DDINT\_SCLK1\_DYN} + I_{DDINT\_DCLK\_DYN} + I_{DDINT\_DMA\_DR\_DYN} + I_{DDINT\_USBCLK\_DYN}$$

$I_{DDINT\_DEEPSLEEP}$ is the only item present that is part of the static power dissipation component. $I_{DDINT\_DEEPSLEEP}$ is specified as a function of voltage (V_{DD_INT}) and temperature (see Table 21).

There are eight different items that contribute to the dynamic power dissipation. These components fall into three broad categories: application-dependent currents, clock currents, and data transmission currents.

Application-Dependent Current

The application-dependent currents include the dynamic current in the core clock domain.

Core clock (CCLK) use is subject to an activity scaling factor (ASF) that represents application code running on the processor cores and L1/L2 memories (Table 22). The ASF is combined with the CCLK frequency and V_{DD_INT} dependent data in Table 23 to calculate this portion.

$I_{DDINT\_CCLK\_DYN}$ (mA) = Table 23 × *ASF*

Clock Current

The dynamic clock currents provide the total power dissipated by all transistors switching in the clock paths. The power dissipated by each clock domain is dependent on voltage (V_{DD_INT}), operating frequency and a unique scaling factor.

$I_{DDINT\_PLLCLK\_DYN}$ (mA) = 0.012 × f_{PLLCLK} (MHz) × V_{DD_INT} (V)

$I_{DDINT\_SYSCLK\_DYN}$ (mA) = 0.120 × f_{SYSCLK} (MHz) × V_{DD_INT} (V)

$I_{DDINT\_SCLK0\_DYN}$ (mA) = 0.110 × f_{SCLK0} (MHz) × V_{DD_INT} (V)

$I_{DDINT\_SCLK1\_DYN}$ (mA) = 0.068 × f_{SCLK1} (MHz) × V_{DD_INT} (V)

$I_{DDINT\_DCLK\_DYN}$ (mA) = 0.055 × f_{DCLK} (MHz) × V_{DD_INT} (V)

The dynamic component of the USB clock is a unique case. The USB clock contributes a near constant current value when used.

Table 20. $I_{DDINT\_USBCLK\_DYN}$ Current

Is USB Enabled?	$I_{DDINT\_USBCLK\_DYN}$ (mA)
Yes – High-Speed Mode	13.94
Yes – Full-Speed Mode	10.83
Yes – Suspend Mode	5.2
No	0.34

Data Transmission Current

The data transmission current represents the power dissipated when transmitting data. This current is expressed in terms of data rate. The calculation is performed by adding the data rate (MB/s) of each DMA-driven access to peripherals, L1, L2, and external memory. This number is then multiplied by a weighted data-rate coefficient and V_{DD_INT}:

$I_{DDINT\_DMADR\_DYN}$ (mA) = *Weighted DRC* × *Total Data Rate* (MB/s) × V_{DD_INT} (V)

A weighted data-rate coefficient is used because different coefficients exist depending on the source and destination of the transfer. For details on using this equation and calculating the weighted DRC, see the related Engineer Zone material. For a quick maximum calculation, the weighted DRC can be assumed to be 0.0497, which is the coefficient for L1 to L1 transfers.

SMC Write Cycle Timing With Reference to SYS_CLKOUT

The following SMC specifications with respect to SYS_CLKOUT are given to accommodate the connection of the SMC to programmable logic devices. These specifications assume that SYS_CLKOUT is outputting a buffered version of SCLK0 by setting CGU_CLKOUTSEL.CLKOUTSEL = 0x3. However, SCLK0 must not run faster than the maximum f_{OCLK} specification. For this example WST = 0x2, WAT = 0x2, and WHT = 0x1.

Table 36. SMC Write Cycle Timing With Reference to SYS_CLKOUT (BxMODE = b#00)

Parameter		V_{DD_EXT} 1.8 V/3.3 V Nominal Min	Max	Unit
Timing Requirements				
t_{SARDY}	SMC0_ARDY Setup Before SYS_CLKOUT	14.4		ns
t_{HARDY}	SMC0_ARDY Hold After SYS_CLKOUT	0.7		ns
Switching Characteristics				
t_{DDAT}	SMC0_Dx Disable After SYS_CLKOUT		7	ns
t_{ENDAT}	SMC0_Dx Enable After SYS_CLKOUT	–2.5		ns
t_{DO}	Output Delay After SYS_CLKOUT[1]		7	ns
t_{HO}	Output Hold After SYS_CLKOUT [1]	–2.5		ns

[1] Output pins/balls include $\overline{\text{SMC0\_AMSx}}$, $\overline{\text{SMC0\_ABEx}}$, SMC0_Ax, SMC0_Dx, $\overline{\text{SMC0\_AOE}}$, and $\overline{\text{SMC0\_AWE}}$.

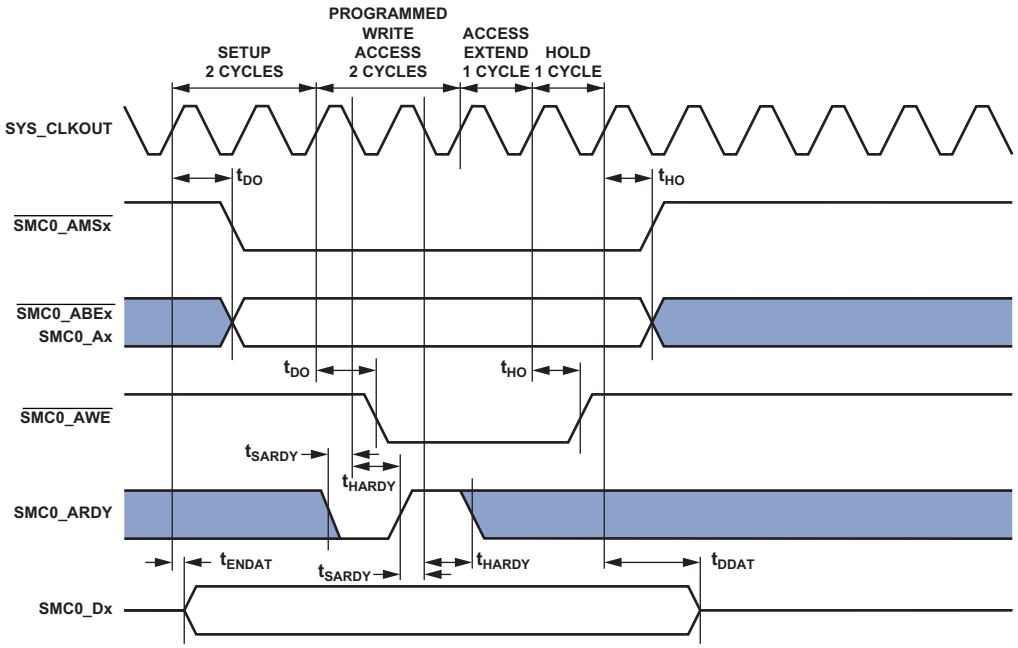


Figure 15. SMC Write Cycle Timing With Reference to SYS_CLKOUT Timing

Asynchronous Flash Write

Table 37 and Figure 16 show asynchronous flash memory write timing, related to the static memory controller (SMC).

Table 37. Asynchronous Flash Write

Parameter		V_{DD_EXT} 1.8 V/3.3 V Nominal Min	Max	Unit
Switching Characteristics				
t_{AMSADV}	SMC0_Ax/$\overline{\text{SMC0\_AMSx}}$ Assertion Before ADV Low[1]	PREST × t_{SCLK0} – 2		ns
$t_{DADVAWE}$	$\overline{\text{SMC0\_AWE}}$ Low Delay From ADV High[2]	PREAT × t_{SCLK0} – 4		ns
t_{WADV}	NR_ADV Active Low Width[3]	WST × t_{SCLK0} – 2		ns
t_{HAWE}	Output[4] Hold After $\overline{\text{SMC0\_AWE}}$ High[5]	WHT × t_{SCLK0}		ns
t_{WAWE}[6]	$\overline{\text{SMC0\_AWE}}$ Active Low Width[7]	WAT × t_{SCLK0} – 2		ns

[1] PREST value set using the SMC_BxETIM.PREST bits.
[2] PREAT value set using the SMC_BxETIM.PREAT bits.
[3] WST value set using the SMC_BxTIM.WST bits.
[4] Output signals are DATA, SMC0_Ax, $\overline{\text{SMC0\_AMSx}}$, $\overline{\text{SMC0\_ABEx}}$.
[5] WHT value set using the SMC_BxTIM.WHT bits.
[6] SMC_BxCTL.ARDYEN bit = 0.
[7] WAT value set using the SMC_BxTIM.WAT bits.

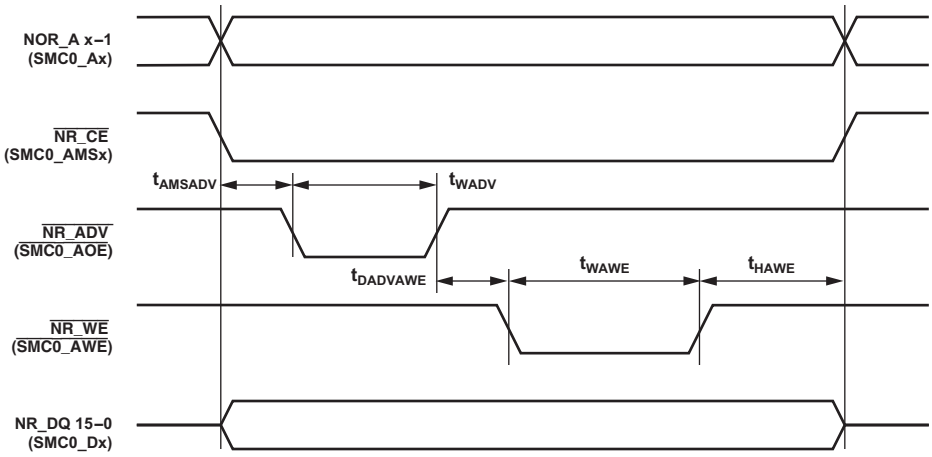


Figure 16. Asynchronous Flash Write

All Accesses

Table 38 describes timing that applies to all memory accesses, related to the static memory controller (SMC).

Table 38. All Accesses

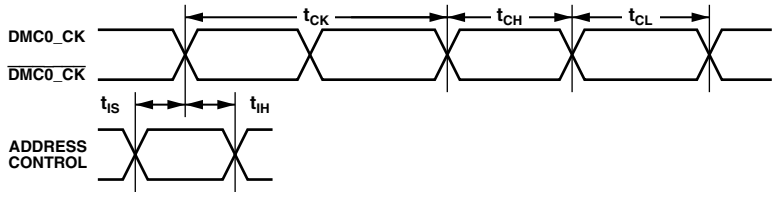
Parameter		V_{DD_EXT} 1.8 V Nominal Min	Max	V_{DD_EXT} 3.3 V Nominal Min	Max	Unit
Switching Characteristic						
t_{TURN}	$\overline{\text{SMC0\_AMSx}}$ Inactive Width	(IT + TT) × t_{SCLK0} – 2		(IT + TT) × t_{SCLK0} – 2		ns

Mobile DDR SDRAM Clock and Control Cycle Timing

Table 42 and Figure 20 show mobile DDR SDRAM clock and control cycle timing, related to the dynamic memory controller (DMC).

Table 42. Mobile DDR SDRAM Clock and Control Cycle Timing, V_{DD_DMC} Nominal 1.8 V

Parameter		200 MHz Min	200 MHz Max	Unit
Switching Characteristics				
t_{CK}	Clock Cycle Time (CL = 2 Not Supported)	5		ns
t_{CH}	Minimum Clock Pulse Width	0.45	0.55	t_{CK}
t_{CL}	Maximum Clock Pulse Width	0.45	0.55	t_{CK}
t_{IS}	Control/Address Setup Relative to DMC0_CK Rise	1.5		ns
t_{IH}	Control/Address Hold Relative to DMC0_CK Rise	1.5		ns



NOTE: CONTROL = $\overline{\text{DMC0\_CS0}}$, DMC0_CKE, $\overline{\text{DMC0\_RAS}}$, $\overline{\text{DMC0\_CAS}}$, AND $\overline{\text{DMC0\_WE}}$.
ADDRESS = DMC0_A00–13, AND DMC0_BA0–2.

Figure 20. Mobile DDR SDRAM Clock and Control Cycle Timing

Serial Ports

To determine whether serial port (SPORT) 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 27 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 65,535:

$$f_{SPTCLKPROG} = \frac{f_{SCLK0}}{(CLKDIV + 1)}$$

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

Table 49. 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]	1.5		1		ns
t_{HFSE}	Frame Sync Hold After SPT_CLK (Externally Generated Frame Sync in Either Transmit or Receive Mode)[1]	3		3		ns
t_{SDRE}	Receive Data Setup Before Receive SPT_CLK[1]	1.5		1		ns
t_{HDRE}	Receive Data Hold After SPT_CLK[1]	3		3		ns
t_{SCLKW}	SPT_CLK Width[2]	$(0.5 \times t_{SPTCLKEXT}) - 1$		$(0.5 \times t_{SPTCLKEXT}) - 1$		ns
$t_{SPTCLKE}$	SPT_CLK Period[2]	$t_{SPTCLKEXT} - 1$		$t_{SPTCLKEXT} - 1$		ns
Switching Characteristics						
t_{DFSE}	Frame Sync Delay After SPT_CLK (Internally Generated Frame Sync in Either Transmit or Receive Mode)[3]		18		15	ns
t_{HOFSE}	Frame Sync Hold After SPT_CLK (Internally Generated Frame Sync in Either Transmit or Receive Mode)[3]	2.5		2.5		ns
t_{DDTE}	Transmit Data Delay After Transmit SPT_CLK[3]		18		15	ns
t_{HDTE}	Transmit Data Hold After Transmit SPT_CLK[3]	2.5		2.5		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 18 on Page 52 in Clock Related Operating Conditions.

[3] Referenced to drive edge.

Table 53. 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]		19		15.5	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.

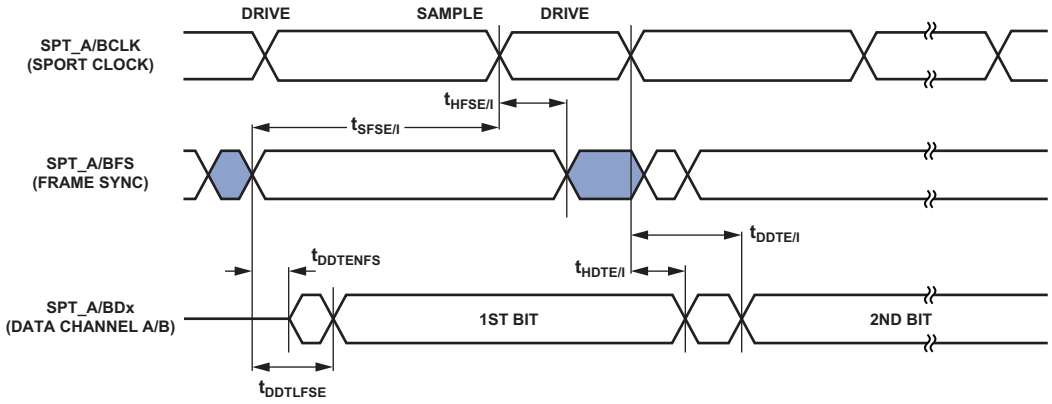


Figure 30. External Late Frame Sync

Enhanced Parallel Peripheral Interface Timing

The following tables and figures describe enhanced parallel peripheral interface timing operations. The POLC bits in the EPPI_CTL register may be used to set the sampling/driving edges of the EPPI clock.

When internally generated, the programmed PPI clock ($f_{PCLKPROG}$) frequency in MHz is set by the following equation where VALUE is a field in the EPPI_CLKDIV register that can be set from 0 to 65,535:

$$f_{PCLKPROG} = \frac{f_{SCLK0}}{(VALUE + 1)}$$

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

When externally generated the EPPI_CLK is called $f_{PCLKEXT}$:

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

Table 60. Enhanced Parallel Peripheral Interface—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_{SFSPI}	External FS Setup Before EPPI_CLK	6.5		5		ns
t_{HFSPI}	External FS Hold After EPPI_CLK	1.5		1		ns
t_{SDRPI}	Receive Data Setup Before EPPI_CLK	6.4		5		ns
t_{HDRPI}	Receive Data Hold After EPPI_CLK	1		1		ns
t_{SFS3GI}	External FS3 Input Setup Before EPPI_CLK Fall Edge in Clock Gating Mode	16.5		14		ns
t_{HFS3GI}	External FS3 Input Hold Before EPPI_CLK Fall Edge in Clock Gating Mode	1.5		0		ns
Switching Characteristics						
t_{PCLKW}	EPPI_CLK Width[1]	$0.5 \times t_{PCLKPROG} - 2$		$0.5 \times t_{PCLKPROG} - 2$		ns
t_{PCLK}	EPPI_CLK Period[1]	$t_{PCLKPROG} - 2$		$t_{PCLKPROG} - 2$		ns
t_{DFSPI}	Internal FS Delay After EPPI_CLK		2		2	ns
t_{HOFSPI}	Internal FS Hold After EPPI_CLK	–4		–3		ns
t_{DDTPI}	Transmit Data Delay After EPPI_CLK		2		2	ns
t_{HDTPI}	Transmit Data Hold After EPPI_CLK	–4		–3		ns

[1] See Table 18 on Page 52 in Clock Related Operating Conditions for details on the minimum period that may be programmed for $t_{PCLKPROG}$.

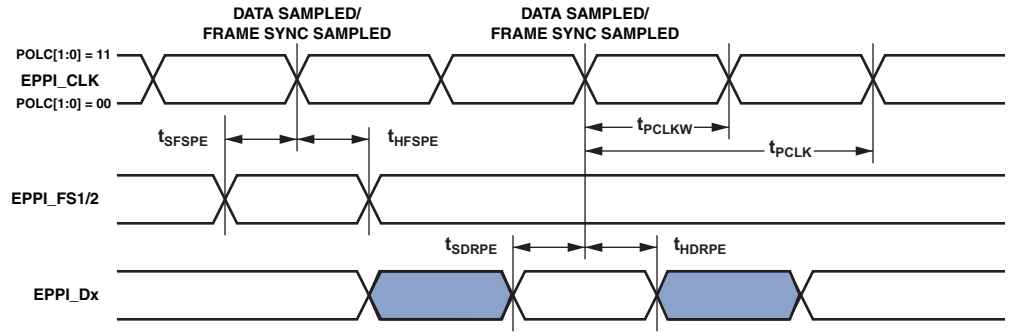


Figure 47. PPI External Clock GP Receive Mode with External Frame Sync Timing

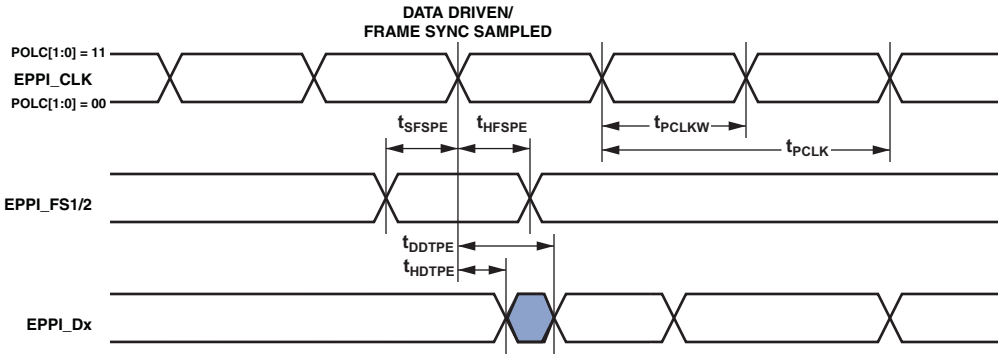


Figure 48. PPI External Clock GP Transmit Mode with External Frame Sync Timing

ENVIRONMENTAL CONDITIONS

To determine the junction temperature on the application printed circuit board, use the following equation:

$$T_J = T_{CASE} + (\Psi_{JT} \times P_D)$$

where:

T_J = Junction temperature (°C).

T_{CASE} = Case temperature (°C) measured by customer at top center of package.

Ψ_{JT} = From Table 65 and Table 66.

P_D = Power dissipation (see Total Internal Power Dissipation on Page 56 for the method to calculate P_D).

Values of θ_{JA} are provided for package comparison and printed circuit board design considerations. θ_{JA} can be used for a first order approximation of T_J by the equation:

$$T_J = T_A + (\theta_{JA} \times P_D)$$

where:

T_A = Ambient temperature (°C).

Values of θ_{JC} are provided for package comparison and printed circuit board design considerations when an external heat sink is required.

In Table 65 and Table 66, airflow measurements comply with JEDEC standards JESD51-2 and JESD51-6. The junction-to-case measurement complies with MIL-STD-883 (Method 1012.1). All measurements use a 2S2P JEDEC test board.

Table 65. Thermal Characteristics for CSP_BGA

Parameter	Condition	Typical	Unit
θ_{JA}	0 linear m/s air flow	28.7	°C/W
θ_{JMA}	1 linear m/s air flow	26.2	°C/W
θ_{JMA}	2 linear m/s air flow	25.2	°C/W
θ_{JC}		10.1	°C/W
Ψ_{JT}	0 linear m/s air flow	0.24	°C/W
Ψ_{JT}	1 linear m/s air flow	0.40	°C/W
Ψ_{JT}	2 linear m/s air flow	0.51	°C/W

Table 66. Thermal Characteristics for LFCSP (QFN)

Parameter	Condition	Typical	Unit
θ_{JA}	0 linear m/s air flow	22.9	°C/W
θ_{JMA}	1 linear m/s air flow	17.9	°C/W
θ_{JMA}	2 linear m/s air flow	16.4	°C/W
θ_{JC}		2.26	°C/W
Ψ_{JT}	0 linear m/s air flow	0.14	°C/W
Ψ_{JT}	1 linear m/s air flow	0.27	°C/W
Ψ_{JT}	2 linear m/s air flow	0.30	°C/W

Table 67. 184-Ball CSP_BGA Ball Assignment (Numerical by Ball Number)

Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name
A01	GND	D08	VDD_DMC	H03	SYS_CLKOUT	L14	GND
A02	DMC0_A09	D09	VDD_DMC	H04	VDD_INT	M01	PC_00
A03	DMC0_BA0	D12	PA_08	H05	GND	M02	RTC0_CLKIN
A04	DMC0_BA1	D13	DMC0_DQ06	H06	GND	M03	PB_15
A05	DMC0_BA2	D14	DMC0_DQ05	H07	GND	M04	PB_12
A06	$\overline{\text{DMC0\_CAS}}$	E01	DMC0_A06	H08	GND	M05	PC_12
A07	$\overline{\text{DMC0\_RAS}}$	E02	DMC0_A05	H09	GND	M06	USB0_VBUS
A08	DMC0_A13	E03	JTG_TDI	H10	GND	M07	USB0_VBC
A09	PA_03	E05	VDD_INT	H11	VDD_DMC	M08	PB_09
A10	DMC0_CK	E06	VDD_DMC	H12	PA_10	M09	PB_05
A11	$\overline{\text{DMC0\_CK}}$	E07	VDD_DMC	H13	PA_11	M10	PB_04
A12	DMC0_LDQS	E08	VDD_DMC	H14	$\overline{\text{DMC0\_UDQS}}$	M11	PB_01
A13	$\overline{\text{DMC0\_LDQS}}$	E09	VDD_DMC	J01	PC_05	M12	PB_03
A14	GND	E10	DMC0_VREF	J02	PC_06	M13	DMC0_LDM
B01	DMC0_A07	E12	SYS_BMODE0	J03	$\overline{\text{SYS\_RESOUT}}$	M14	SYS_CLKIN
B02	DMC0_A08	E13	DMC0_DQ08	J04	VDD_INT	N01	RTC0_XTAL
B03	DMC0_A11	E14	DMC0_DQ07	J05	VDD_RTC	N02	PB_14
B04	DMC0_A10	F01	DMC0_A01	J06	GND	N03	PB_11
B05	DMC0_A12	F02	DMC0_A02	J07	GND	N04	PC_14
B06	$\overline{\text{DMC0\_WE}}$	F03	PC_09	J08	GND	N05	PC_11
B07	$\overline{\text{DMC0\_CS0}}$	F04	VDD_INT	J09	GND	N06	USB0_ID
B08	DMC0_ODT	F05	VDD_INT	J10	GND_HADC	N07	USB0_DP
B09	DMC0_CKE	F06	GND	J11	VDD_OTP	N08	PB_08
B10	DMC0_DQ00	F07	GND	J12	PA_13	N09	PB_06
B11	DMC0_DQ02	F08	GND	J13	DMC0_DQ13	N10	PB_00
B12	DMC0_DQ01	F09	GND	J14	DMC0_UDQS	N11	HADC0_VIN2
B13	DMC0_DQ04	F10	VDD_DMC	K01	PC_04	N12	HADC0_VIN1
B14	DMC0_DQ03	F11	VDD_DMC	K02	PC_01	N13	PA_15
C01	JTG_TDO_SWO	F12	$\overline{\text{SYS\_FAULT}}$	K03	PC_02	N14	SYS_XTAL
C02	JTG_TMS_SWDIO	F13	DMC0_DQ10	K05	VDD_EXT	P01	GND
C03	JTG_TCK_SWCLK	F14	DMC0_DQ09	K06	VDD_EXT	P02	PB_13
C04	PA_01	G01	DMC0_A03	K07	VDD_EXT	P03	PB_10
C05	SYS_EXTWAKE	G02	PA_00	K08	VDD_EXT	P04	PC_13
C06	PA_02	G03	PC_08	K09	VDD_EXT	P05	USB0_XTAL
C07	$\overline{\text{SYS\_NMI}}$	G04	VDD_INT	K10	VDD_HADC	P06	USB0_CLKIN
C08	GND	G05	GND	K12	PA_12	P07	USB0_DM
C09	PA_04	G06	GND	K13	DMC0_DQ15	P08	PB_07
C10	PA_05	G07	GND	K14	DMC0_DQ14	P09	HADC0_VREFN
C11	PA_06	G08	GND	L01	PC_03	P10	HADC0_VREFP
C12	PA_07	G09	GND	L02	TWI0_SDA	P11	HADC0_VIN3
C13	$\overline{\text{SYS\_HWRST}}$	G10	GND	L03	TWI0_SCL	P12	HADC0_VIN0
C14	SYS_BMODE1	G11	VDD_DMC	L06	VDD_USB	P13	PA_14
D01	DMC0_A00	G12	PA_09	L07	VDD_EXT	P14	GND
D02	DMC0_A04	G13	DMC0_DQ11	L08	VDD_EXT		
D03	$\overline{\text{JTG\_TRST}}$	G14	DMC0_DQ12	L09	VDD_EXT		
D06	VDD_DMC	H01	PC_07	L12	PB_02		
D07	VDD_DMC	H02	PC_10	L13	DMC0_UDM		