**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 | Fixed Point |
| Interface | SPI, SSP, UART |
| Clock Rate | 500MHz |
| Non-Volatile Memory | External |
| On-Chip RAM | 328kB |
| Voltage - I/O | 2.50V, 3.30V |
| Voltage - Core | 1.25V |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 297-BGA |
| Supplier Device Package | 297-PBGA (27x27) |
| Purchase URL | https://www.e-xfl.com/product-detail/analog-devices/adsp-bf561sbbz500 |

# ADSP-BF561* PRODUCT PAGE QUICK LINKS

**Last Content Update: 02/23/2017**

## COMPARABLE PARTS

View a parametric search of comparable parts.

## EVALUATION KITS

- Low Cost ICE-1000 and High Performance ICE-2000 USB-based JTAG Emulators
- Multimedia Starter Kit
- The ADSP-BF561 EZ-Kit Lite evaluation hardware provides a low-cost hardware solution for evaluating the ADSP-BF561 Blackfin processor.
- USB-Based Emulator and High Performance USB-Based Emulator

## DOCUMENTATION

**Application Notes**

- AN-813: Interfacing the ADSP-BF533/ADSP-BF561 Blackfin®; Processors to High Speed Parallel ADCs
- EE-120: Interfacing Assembly Language Programs to C
- EE-126: The ABCs of SDRAMemories
- EE-175: Emulator and Evaluation Hardware Troubleshooting Guide for VisualDSP++ Users
- EE-183: Rational Sample Rate Conversion with Blackfin® Processors
- EE-185: Fast Floating-Point Arithmetic Emulation on Blackfin® Processors
- EE-228: Switching Regulator Design Considerations for ADSP-BF533 Blackfin® Processors
- EE-261: Understanding Jitter Requirements of PLL-Based Processors
- EE-269: A Beginner's Guide to Ethernet 802.3
- EE-281: Hardware Design Checklist for the Blackfin® Processors
- EE-289: Implementing FAT32 File Systems on ADSP-BF533 Blackfin® Processors
- EE-293: Estimating Power for ADSP-BF561 Blackfin® Processors
- EE-294: Energy-Aware Programming on Blackfin Processors
- EE-300: Interfacing Blackfin® EZ-KIT Lite® Boards to CMOS Image Sensors
- EE-314: Booting the ADSP-BF561 Blackfin® Processor
- EE-323: Implementing Dynamically Loaded Software Modules
- EE-326: Blackfin® Processor and SDRAM Technology
- EE-330: Windows Vista Compatibility in VisualDSP++ 5.0 Development Tools
- EE-332: Cycle Counting and Profiling
- EE-336: Putting ADSP-BF54x Blackfin® Processor Booting into Practice
- EE-339: Using External Switching Regulators with Blackfin® Processors
- EE-340: Connecting SHARC® and Blackfin® Processors over SPI
- EE-356: Emulator and Evaluation Hardware Troubleshooting Guide for CCES Users

**Data Sheet**

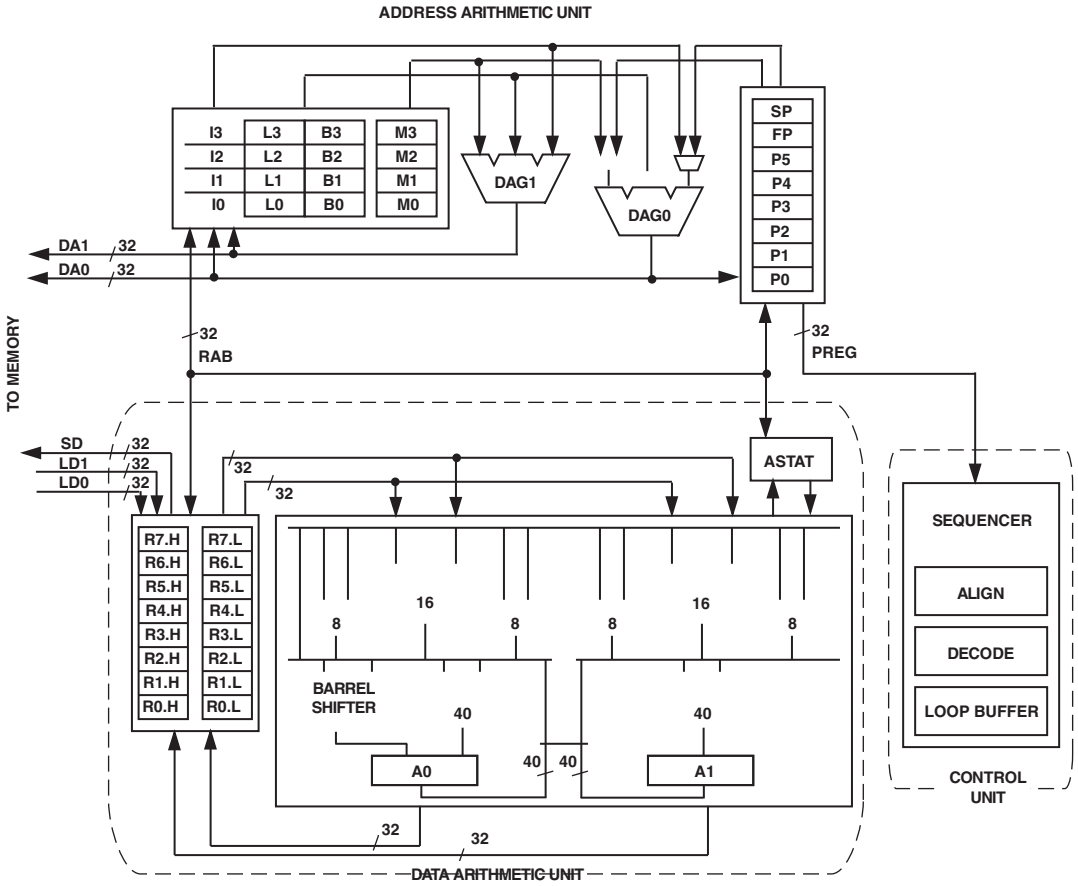


*Figure 2. Blackfin Processor Core*

## MEMORY ARCHITECTURE

The ADSP-BF561 views memory as a single unified 4G byte address space, using 32-bit addresses. All resources including internal memory, external memory, and I/O control registers occupy separate sections of this common address space. The memory portions of this address space are arranged in a hierarchical structure to provide a good cost/performance balance of some very fast, low latency memory as cache or SRAM very close to the processor, and larger, lower cost and performance memory systems farther away from the processor. The ADSP-BF561 memory map is shown in Figure 3.

The L1 memory system in each core is the highest performance memory available to each Blackfin core. The L2 memory provides additional capacity with lower performance. Lastly, the off-chip memory system, accessed through the External Bus Interface Unit (EBIU), provides expansion with SDRAM, flash memory, and SRAM, optionally accessing more than 768M bytes of physical memory. The memory DMA controllers provide high bandwidth data movement capability. They can perform block transfers of code or data between the internal L1/L2 memories and the external memory spaces.

### *Internal (On-Chip) Memory*

The ADSP-BF561 has four blocks of on-chip memory providing high bandwidth access to the core.

The first is the L1 instruction memory of each Blackfin core consisting of 16K bytes of four-way set-associative cache memory and 16K bytes of SRAM. The cache memory may also be configured as an SRAM. This memory is accessed at full processor speed. When configured as SRAM, each of the two 16K banks of memory is broken into 4K sub-banks which can be independently accessed by the processor and DMA.

The second on-chip memory block is the L1 data memory of each Blackfin core which consists of four banks of 16K bytes each. Two of the L1 data memory banks can be configured as one way of a two-way set-associative cache or as an SRAM. The other two banks are configured as SRAM. All banks are accessed at full processor speed. When configured as SRAM, each of the four 16K banks of memory is broken into 4K sub-banks which can be independently accessed by the processor and DMA.

The third memory block associated with each core is a 4K byte scratchpad SRAM which runs at the same speed as the L1 memories, but is only accessible as data SRAM (it cannot be configured as cache memory and is not accessible via DMA).

writing the appropriate values into the Interrupt Assignment Registers (SIC_IAR7–0). Table 2 describes the inputs into the SIC and the default mappings into the CEC.

**Table 2. System Interrupt Controller (SIC)**

| Peripheral Interrupt Event | Default Mapping |
|---|---|
| PLL Wakeup | IVG7 |
| DMA1 Error (Generic) | IVG7 |
| DMA2 Error (Generic) | IVG7 |
| IMDMA Error | IVG7 |
| PPI0 Error | IVG7 |
| PPI1 Error | IVG7 |
| SPORT0 Error | IVG7 |
| SPORT1 Error | IVG7 |
| SPI Error | IVG7 |
| UART Error | IVG7 |
| Reserved | IVG7 |
| DMA1 Channel 0 Interrupt (PPI0) | IVG8 |
| DMA1 Channel 1 Interrupt (PPI1) | IVG8 |
| DMA1 Channel 2 Interrupt | IVG8 |
| DMA1 Channel 3 Interrupt | IVG8 |
| DMA1 Channel 4 Interrupt | IVG8 |
| DMA1 Channel 5 Interrupt | IVG8 |
| DMA1 Channel 6 Interrupt | IVG8 |
| DMA1 Channel 7 Interrupt | IVG8 |
| DMA1 Channel 8 Interrupt | IVG8 |
| DMA1 Channel 9 Interrupt | IVG8 |
| DMA1 Channel 10 Interrupt | IVG8 |
| DMA1 Channel 11 Interrupt | IVG8 |
| DMA2 Channel 0 Interrupt (SPORT0 Rx) | IVG9 |
| DMA2 Channel 1 Interrupt (SPORT0 Tx) | IVG9 |
| DMA2 Channel 2 Interrupt (SPORT1 Rx) | IVG9 |
| DMA2 Channel 3 Interrupt (SPORT1 Tx) | IVG9 |
| DMA2 Channel 4 Interrupt (SPI) | IVG9 |
| DMA2 Channel 5 Interrupt (UART Rx) | IVG9 |
| DMA2 Channel 6 Interrupt (UART Tx) | IVG9 |
| DMA2 Channel 7 Interrupt | IVG9 |
| DMA2 Channel 8 Interrupt | IVG9 |
| DMA2 Channel 9 Interrupt | IVG9 |
| DMA2 Channel 10 Interrupt | IVG9 |
| DMA2 Channel 11 Interrupt | IVG9 |
| Timer0 Interrupt | IVG10 |
| Timer1 Interrupt | IVG10 |
| Timer2 Interrupt | IVG10 |
| Timer3 Interrupt | IVG10 |
| Timer4 Interrupt | IVG10 |
| Timer5 Interrupt | IVG10 |
| Timer6 Interrupt | IVG10 |
| Timer7 Interrupt | IVG10 |
| Timer8 Interrupt | IVG10 |
| Timer9 Interrupt | IVG10 |
| Timer10 Interrupt | IVG10 |
| Timer11 Interrupt | IVG10 |
| Programmable Flags 15–0 Interrupt A | IVG11 |
| Programmable Flags 15–0 Interrupt B | IVG11 |
| Programmable Flags 31–16 Interrupt A | IVG11 |
| Programmable Flags 31–16 Interrupt B | IVG11 |
| Programmable Flags 47–32 Interrupt A | IVG11 |
| Programmable Flags 47–32 Interrupt B | IVG11 |
| DMA1 Channel 12/13 Interrupt (Memory DMA/Stream 0) | IVG8 |
| DMA1 Channel 14/15 Interrupt (Memory DMA/Stream 1) | IVG8 |
| DMA2 Channel 12/13 Interrupt (Memory DMA/Stream 0) | IVG9 |
| DMA2 Channel 14/15 Interrupt (Memory DMA/Stream 1) | IVG9 |
| IMDMA Stream 0 Interrupt | IVG12 |
| IMDMA Stream 1 Interrupt | IVG12 |
| Watchdog Timer Interrupt | IVG13 |
| Reserved | IVG7 |
| Reserved | IVG7 |
| Supplemental Interrupt 0 | IVG7 |
| Supplemental Interrupt 1 | IVG7 |

### *Event Control*

The ADSP-BF561 provides the user with a very flexible mechanism to control the processing of events. In the CEC, three registers are used to coordinate and control events. Each of the registers is 16 bits wide, while each bit represents a particular event class.

- CEC Interrupt Latch Register (ILAT) – The ILAT register indicates when events have been latched. The appropriate bit is set when the processor has latched the event and cleared when the event has been accepted into the system. This register is updated automatically by the controller, but may also be written to clear (cancel) latched events. This register may be read while in supervisor mode and may only be written while in supervisor mode when the corresponding IMASK bit is cleared.
- CEC Interrupt Mask Register (IMASK) – The IMASK register controls the masking and unmasking of individual events. When a bit is set in the IMASK register, that event is unmasked and will be processed by the CEC when asserted. A cleared bit in the IMASK register masks the event, thereby preventing the processor from servicing the event

even though the event may be latched in the ILAT register. This register may be read from or written to while in supervisor mode.

Note that general-purpose interrupts can be globally enabled and disabled with the STI and CLI instructions, respectively.

- CEC Interrupt Pending Register (IPEND) – The IPEND register keeps track of all nested events. A set bit in the IPEND register indicates the event is currently active or nested at some level. This register is updated automatically by the controller but may be read while in supervisor mode.

The SIC allows further control of event processing by providing six 32-bit interrupt control and status registers. Each register contains a bit corresponding to each of the peripheral interrupt events shown in Table 2.

- SIC Interrupt Mask Registers (SIC_IMASKx) – These registers control the masking and unmasking of each peripheral interrupt event. When a bit is set in these registers, that peripheral event is unmasked and will be processed by the system when asserted. A cleared bit in these registers masks the peripheral event, thereby preventing the processor from servicing the event.
- SIC Interrupt Status Registers (SIC_ISRx) – As multiple peripherals can be mapped to a single event, these registers allow the software to determine which peripheral event source triggered the interrupt. A set bit indicates the peripheral is asserting the interrupt; a cleared bit indicates the peripheral is not asserting the event.
- SIC Interrupt Wakeup Enable Registers (SIC_IWRx) – By enabling the corresponding bit in these registers, each peripheral can be configured to wake up the processor, should the processor be in a powered-down mode when the event is generated.

Because multiple interrupt sources can map to a single general-purpose interrupt, multiple pulse assertions can occur simultaneously, before or during interrupt processing for an interrupt event already detected on this interrupt input. The IPEND register contents are monitored by the SIC as the interrupt acknowledgement.

The appropriate ILAT register bit is set when an interrupt rising edge is detected (detection requires two core clock cycles). The bit is cleared when the respective IPEND register bit is set. The IPEND bit indicates that the event has entered into the processor pipeline. At this point the CEC will recognize and queue the next rising edge event on the corresponding event input. The minimum latency from the rising edge transition of the general-purpose interrupt to the IPEND output asserted is three core clock cycles; however, the latency can be much higher, depending on the activity within and the mode of the processor.

## DMA CONTROLLERS

The ADSP-BF561 has two independent DMA controllers that support automated data transfers with minimal overhead for the DSP cores. DMA transfers can occur between the ADSP-BF561 internal memories and any of its DMA-capable peripherals. Additionally, DMA transfers can be accomplished between any of the DMA-capable peripherals and external devices connected to the external memory interfaces, including the SDRAM controller and the asynchronous memory controller. DMA-capable peripherals include the SPORTs, SPI port, UART, and PPIs. Each individual DMA-capable peripheral has at least one dedicated DMA channel.

The ADSP-BF561 DMA controllers support both 1-dimensional (1-D) and 2-dimensional (2-D) DMA transfers. DMA transfer initialization can be implemented from registers or from sets of parameters called descriptor blocks.

The 2-D DMA capability supports arbitrary row and column sizes up to 64K elements by 64K elements, and arbitrary row and column step sizes up to ± 32K elements. Furthermore, the column step size can be less than the row step size, allowing implementation of interleaved data streams. This feature is especially useful in video applications where data can be de-interleaved on the fly.

Examples of DMA types supported by the ADSP-BF561 DMA controllers include:

- A single linear buffer that stops upon completion.
- A circular autorefreshing buffer that interrupts on each full or fractionally full buffer.
- 1-D or 2-D DMA using a linked list of descriptors.
- 2-D DMA using an array of descriptors, specifying only the base DMA address within a common page.

In addition to the dedicated peripheral DMA channels, each DMA Controller has four memory DMA channels provided for transfers between the various memories of the ADSP-BF561 system. These enable transfers of blocks of data between any of the memories—including external SDRAM, ROM, SRAM, and flash memory—with minimal processor intervention. Memory DMA transfers can be controlled by a very flexible descriptor-based methodology or by a standard register-based autobuffer mechanism.

Further, the ADSP-BF561 has a four channel Internal Memory DMA (IMDMA) Controller. The IMDMA Controller allows data transfers between any of the internal L1 and L2 memories.

## WATCHDOG TIMER

Each ADSP-BF561 core includes a 32-bit timer, which can be used to implement a software watchdog function. A software watchdog can improve system availability by forcing the processor to a known state, via 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 to zero from the programmed value. This protects the system from remaining in an unknown state where software, which would normally reset the timer, has stopped running due to an external noise condition or software error.

regulator for the processor can be shut off by writing b#00 to the FREQ bits of the VR_CTL register. This disables both CCLK and SCLK. Furthermore, it sets the internal power supply voltage ($V_{DDINT}$) to 0 V to provide the lowest static power dissipation. Any critical information stored internally (memory contents, register contents, etc.) must be written to a nonvolatile storage device prior to removing power if the processor state is to be preserved. Since $V_{DDEXT}$ is still 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 have power still applied without drawing unwanted current. The internal supply regulator can be woken up by asserting the $\overline{\text{RESET}}$ pin.

### Power Savings

As shown in Table 4, the ADSP-BF561 supports two different power domains. The use of multiple power domains maximizes flexibility, while maintaining compliance with industry standards and conventions. By isolating the internal logic of the ADSP-BF561 into its own power domain, separate from the I/O, the processor can take advantage of Dynamic Power Management, without affecting the I/O devices. There are no sequencing requirements for the various power domains.

**Table 4. ADSP-BF561 Power Domains**

| Power Domain | $V_{DD}$ Range |
|---|---|
| All internal logic | $V_{DDINT}$ |
| I/O | $V_{DDEXT}$ |

The power dissipated by a processor is largely a function of the clock frequency of the processor and the square of the operating voltage. For example, reducing the clock frequency by 25% results in a 25% reduction in dynamic power dissipation, while reducing the voltage by 25% reduces dynamic power dissipation by more than 40%. Further, these power savings are additive, in that if the clock frequency and supply voltage are both reduced, the power savings can be dramatic.

The dynamic power management feature of the ADSP-BF561 allows both the processor's input voltage ($V_{DDINT}$) and clock frequency ($f_{CCLK}$) to be dynamically controlled.

The savings in power dissipation can be modeled using the power savings factor and % power savings calculations.

The power savings factor is calculated as:

$$\textit{power savings factor} = \frac{f_{CCLKRED}}{f_{CCLKNOM}} \times \left(\frac{V_{DDINTRED}}{V_{DDINTNOM}}\right)^2 \times \left(\frac{t_{RED}}{t_{NOM}}\right)$$

where the variables in the equations are:

$f_{CCLKNOM}$ is the nominal core clock frequency

$f_{CCLKRED}$ is the reduced core clock frequency

$V_{DDINTNOM}$ is the nominal internal supply voltage

$V_{DDINTRED}$ is the reduced internal supply voltage

$t_{NOM}$ is the duration running at $f_{CCLKNOM}$

$t_{RED}$ is the duration running at $f_{CCLKRED}$

The percent power savings is calculated as:

$$\%\ \textit{power savings} = (1 - \textit{power savings factor}) \times 100\%$$

## VOLTAGE REGULATION

The ADSP-BF561 processor provides an on-chip voltage regulator that can generate appropriate $V_{DDINT}$ voltage levels from the $V_{DDEXT}$ supply. See Operating Conditions on Page 20 for regulator tolerances and acceptable $V_{DDEXT}$ ranges for specific models.

Figure 4 shows the typical external components required to complete the power management system. The regulator controls the internal logic voltage levels and is programmable with the voltage regulator control register (VR_CTL) in increments of 50 mV. To reduce standby power consumption, the internal voltage regulator can be programmed to remove power to the processor core while keeping I/O power ($V_{DDEXT}$) supplied. While in the hibernate state, $V_{DDEXT}$ can still be applied, thus eliminating the need for external buffers. The voltage regulator can be activated from this power-down state by asserting $\overline{\text{RESET}}$, which will then initiate a boot sequence. The regulator can also be disabled and bypassed at the user's discretion.

The internal voltage regulation feature is not available on any of the 600 MHz speed grade models or automotive grade models. External voltage regulation is required to ensure correct operation of these parts at 600 MHz.

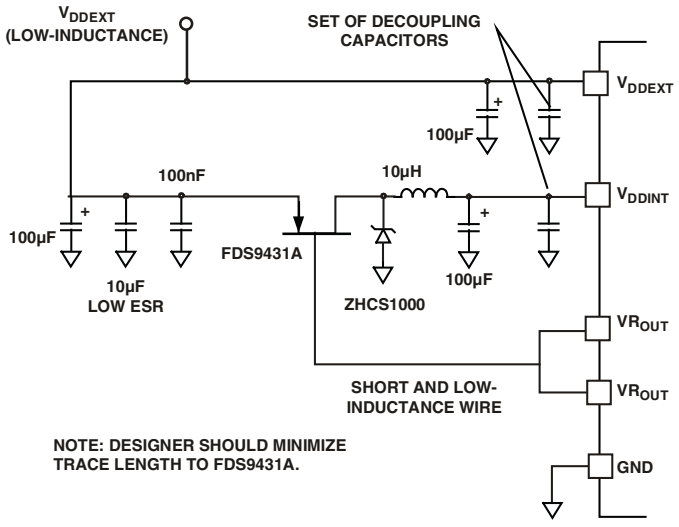


*Figure 4. Voltage Regulator Circuit*

### Voltage Regulator Layout Guidelines

Regulator external component placement, board routing, and bypass capacitors all have a significant effect on noise injected into the other analog circuits on-chip. The VROUT1–0 traces and voltage regulator external components should be considered as noise sources when doing board layout and should not be routed or placed near sensitive circuits or components on the

# PIN DESCRIPTIONS

ADSP-BF561 pin definitions are listed in Table 8. In order to maintain maximum function and reduce package size and pin count, some pins have multiple functions. In cases where pin function is reconfigurable, the default state is shown in plain text, while alternate functionality is shown in italics.

All pins are three-stated during and immediately after reset, except the external memory interface, asynchronous memory control, and synchronous memory control pins. These pins are all driven high, with the exception of CLKOUT, which toggles at the system clock rate. However if $\overline{BR}$ is active, the memory pins are also three-stated.

All I/O pins have their input buffers disabled, with the exception of the pins that need pull-ups or pull-downs if unused, as noted in Table 8.

**Table 8. Pin Descriptions**

| Pin Name | Type | Function | Driver Type[1] |
|---|---|---|---|
| *EBIU* | | | |
| ADDR25–2 | O | Address Bus for Async/Sync Access | A |
| DATA31–0 | I/O | Data Bus for Async/Sync Access | A |
| $\overline{ABE3–0}$/SDQM3–0 | O | Byte Enables/Data Masks for Async/Sync Access | A |
| $\overline{BR}$ | I | Bus Request (This pin should be pulled HIGH if not used.) | |
| $\overline{BG}$ | O | Bus Grant | A |
| $\overline{BGH}$ | O | Bus Grant Hang | A |
| *EBIU (ASYNC)* | | | |
| $\overline{AMS3–0}$ | O | Bank Select | A |
| ARDY | I | Hardware Ready Control (This pin should be pulled HIGH if not used.) | |
| $\overline{AOE}$ | O | Output Enable | A |
| $\overline{AWE}$ | O | Write Enable | A |
| $\overline{ARE}$ | O | Read Enable | A |
| *EBIU (SDRAM)* | | | |
| $\overline{SRAS}$ | O | Row Address Strobe | A |
| $\overline{SCAS}$ | O | Column Address Strobe | A |
| $\overline{SWE}$ | O | Write Enable | A |
| SCKE | O | Clock Enable | A |
| SCLK0/CLKOUT | O | Clock Output Pin 0 | B |
| SCLK1 | O | Clock Output Pin 1 | B |
| SA10 | O | SDRAM A10 Pin | A |
| $\overline{SMS3–0}$ | O | Bank Select | A |

# SPECIFICATIONS

Component specifications are subject to change without notice.

## OPERATING CONDITIONS

| Parameter | | Conditions | Min | Nominal | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{DDINT}$ | Internal Supply Voltage[1] | Non automotive 500 MHz and 533 MHz speed grade models[2] | 0.8 | 1.25 | 1.375 | V |
| $V_{DDINT}$ | Internal Supply Voltage[3] | 600 MHz speed grade models[2] | 0.8 | 1.35 | 1.4185 | V |
| $V_{DDINT}$ | Internal Supply Voltage[3] | Automotive grade models[2] | 0.95 | 1.25 | 1.375 | V |
| $V_{DDEXT}$ | External Supply Voltage | Non automotive grade models[2] | 2.25 | 2.5, or 3.3 | 3.6 | V |
| $V_{DDEXT}$ | External Supply Voltage | Automotive grade models[2] | 2.7 | 3.3 | 3.6 | V |
| $V_{IH}$ | High Level Input Voltage[4, 5] | | 2.0 | | 3.6 | V |
| $V_{IL}$ | Low Level Input Voltage[5] | | –0.3 | | +0.6 | V |
| $T_J$ | Junction Temperature | 256-Ball CSP_BGA (12 mm × 12 mm) @ $T_{AMBIENT}$ = 0°C to +70°C | 0 | | +105 | °C |
| $T_J$ | Junction Temperature | 256-Ball CSP_BGA (17 mm × 17 mm) @ $T_{AMBIENT}$ = 0°C to +70°C | 0 | | +95 | °C |
| $T_J$ | Junction Temperature | 256-Ball CSP_BGA (17 mm × 17 mm) @ $T_{AMBIENT}$ =–40°C to +85°C | –40 | | +115 | °C |
| $T_J$ | Junction Temperature | 297-Ball PBGA @ $T_{AMBIENT}$ = 0°C to +70°C | 0 | | +95 | °C |
| $T_J$ | Junction Temperature | 297-Ball PBGA @ $T_{AMBIENT}$ = –40°C to +85°C | –40 | | +115 | °C |

[1] Internal voltage ($V_{DDINT}$) regulator tolerance is –5% to +10% for all models.
[2] See Ordering Guide on Page 63.
[3] The internal voltage regulation feature is not available. External voltage regulation is required to ensure correct operation.
[4] The ADSP-BF561 is 3.3 V tolerant (always accepts up to 3.6 V maximum $V_{IH}$), but voltage compliance (on outputs, $V_{OH}$) depends on the input $V_{DDEXT}$, because $V_{OH}$ (maximum) approximately equals $V_{DDEXT}$ (maximum). This 3.3 V tolerance applies to bidirectional and input only pins.
[5] Applies to all signal pins.

Table 9 and Table 10 describe the timing requirements for the ADSP-BF561 clocks ($t_{CCLK} = 1/f_{CCLK}$). Take care in selecting MSEL, SSEL, and CSEL ratios so as not to exceed the maximum core clock, system clock, and Voltage Controlled Oscillator (VCO) operating frequencies, as described in Absolute Maximum Ratings on Page 22. Table 11 describes phase-locked loop operating conditions.

**Table 9. Core Clock (CCLK) Requirements—500 MHz and 533 MHz Speed Grade Models[1]**

| Parameter | | Max | Unit |
|---|---|---|---|
| $f_{CCLK}$ | CCLK Frequency ($V_{DDINT}$ = 1.235 Vminimum)[2] | 533 | MHz |
| $f_{CCLK}$ | CCLK Frequency ($V_{DDINT}$ = 1.1875 Vminimum) | 500 | MHz |
| $f_{CCLK}$ | CCLK Frequency ($V_{DDINT}$ = 1.045 Vminimum) | 444 | MHz |
| $f_{CCLK}$ | CCLK Frequency ($V_{DDINT}$ = 0.95 Vminimum) | 350 | MHz |
| $f_{CCLK}$ | CCLK Frequency ($V_{DDINT}$ = 0.855 Vminimum)[3] | 300 | MHz |
| $f_{CCLK}$ | CCLK Frequency ($V_{DDINT}$ = 0.8 V minimum)[3] | 250 | MHz |

[1] See Ordering Guide on Page 63.
[2] External Voltage regulation is required on automotive grade models (see Ordering Guide on Page 63) to ensure correct operation.
[3] Not applicable to automotive grade models. See Ordering Guide on Page 63.

**Table 10. Core Clock (CCLK) Requirements—600 MHz Speed Grade Models[1]**

| Parameter | | Max | Unit |
|---|---|---|---|
| $f_{CCLK}$ | CCLK Frequency ($V_{DDINT}$ = 1.2825 V minimum)[2] | 600 | MHz |
| $f_{CCLK}$ | CCLK Frequency ($V_{DDINT}$ = 1.235 V minimum) | 533 | MHz |
| $f_{CCLK}$ | CCLK Frequency ($V_{DDINT}$ = 1.1875 V minimum) | 500 | MHz |
| $f_{CCLK}$ | CCLK Frequency ($V_{DDINT}$ = 1.045 V minimum) | 444 | MHz |
| $f_{CCLK}$ | CCLK Frequency ($V_{DDINT}$ = 0.95 V minimum) | 350 | MHz |
| $f_{CCLK}$ | CCLK Frequency ($V_{DDINT}$ = 0.855 V minimum) | 300 | MHz |
| $f_{CCLK}$ | CCLK Frequency ($V_{DDINT}$ = 0.8 V minimum) | 250 | MHz |

[1] See Ordering Guide on Page 63.
[2] External voltage regulator required to ensure proper operation at 600 MHz.

**Table 11. Phase-Locked Loop Operating Conditions**

| Parameter | Min | Max | Unit |
|---|---|---|---|
| Voltage Controlled Oscillator (VCO) Frequency | 50 | Maximum $f_{CCLK}$ | MHz |

**Table 12. System Clock (SCLK) Requirements**

| Parameter[1] | | Max $V_{DDEXT}$ = 2.5V/3.3V | Unit |
|---|---|---|---|
| $f_{SCLK}$ | CLKOUT/SCLK Frequency ($V_{DDINT} \geq 1.14$ V) | 133[2] | MHz |
| $f_{SCLK}$ | CLKOUT/SCLK Frequency ($V_{DDINT} < 1.14$ V) | 100 | MHz |

[1] $t_{SCLK}$ (= $1/f_{SCLK}$) must be greater than or equal to $t_{CCLK}$.
[2] Rounded number. Guaranteed to $t_{SCLK}$ = 7.5 ns. See Table 20 on Page 26.

## ELECTRICAL CHARACTERISTICS

| Parameter | | Test Conditions | Min | Typical | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{OH}$ | High Level Output Voltage[1] | $V_{DDEXT}$ = 3.0 V, $I_{OH}$ = –0.5 mA | 2.4 | | | V |
| $V_{OL}$ | Low Level Output Voltage[1] | $V_{DDEXT}$ = 3.0 V, $I_{OL}$ = 2.0 mA | | | 0.4 | V |
| $I_{IH}$ | High Level Input Current[2] | $V_{DDEXT}$ = Maximum, $V_{IN}$ = $V_{DD}$ Maximum | | | 10.0 | µA |
| $I_{IHP}$ | High Level Input Current JTAG[3] | $V_{DDEXT}$ = Maximum, $V_{IN}$ = $V_{DD}$ Maximum | | | 50.0 | µA |
| $I_{IL}$[4] | Low Level Input Current[2] | $V_{DDEXT}$ = Maximum, $V_{IN}$ = 0 V | | | 10.0 | µA |
| $I_{OZH}$ | Three-State Leakage Current[5] | $V_{DDEXT}$ = Maximum, $V_{IN}$ = $V_{DD}$ Maximum | | | 10.0 | µA |
| $I_{OZL}$[4] | Three-State Leakage Current[5] | $V_{DDEXT}$ = Maximum, $V_{IN}$ = 0 V | | | 10.0 | µA |
| $C_{IN}$ | Input Capacitance[6] | $f_{IN}$ = 1 MHz, $T_{AMBIENT}$ = 25°C, $V_{IN}$ = 2.5 V | | 4 | 8[7] | pF |
| $I_{DDHIBERNATE}$[8] | $V_{DDEXT}$ Current in Hibernate Mode | CLKIN=0 MHz, $V_{DDEXT}$ = 3.65 V with Voltage Regulator Off ($V_{DDINT}$ = 0 V) | | 50 | | µA |
| $I_{DDDEEPSLEEP}$[9] | $V_{DDINT}$ Current in Deep Sleep Mode | $V_{DDINT}$ = 0.8 V, $T_{JUNCTION}$ = 25°C | | 70 | | mA |
| $I_{DD\_TYP}$[9, 10] | $V_{DDINT}$ Current | $V_{DDINT}$ = 0.8 V, $f_{CCLK}$ = 50 MHz, $T_{JUNCTION}$ = 25°C | | 127 | | mA |
| $I_{DD\_TYP}$[9, 10] | $V_{DDINT}$ Current | $V_{DDINT}$ = 1.25 V, $f_{CCLK}$ = 500 MHz, $T_{JUNCTION}$ = 25°C | | 660 | | mA |
| $I_{DD\_TYP}$[9, 10] | $V_{DDINT}$ Current | $V_{DDINT}$ = 1.35 V, $f_{CCLK}$ = 600 MHz, $T_{JUNCTION}$ = 25°C | | 818 | | mA |

[1] Applies to output and bidirectional pins.
[2] Applies to input pins except JTAG inputs.
[3] Applies to JTAG input pins (TCK, TDI, TMS, $\overline{TRST}$).
[4] Absolute value.
[5] Applies to three-statable pins.
[6] Applies to all signal pins.
[7] Guaranteed, but not tested.
[8] CLKIN must be tied to $V_{DDEXT}$ or GND during hibernate.
[9] Maximum current drawn. See *Estimating Power for ADSP-BF561 Blackfin Processors (EE-293)* on the Analog Devices website (www.analog.com)—use site search on "EE-293".
[10] Both cores executing 75% dual MAC, 25% ADD instructions with moderate data bus activity.

System designers should refer to *Estimating Power for the ADSP-BF561 (EE-293)*, which provides detailed information for optimizing designs for lowest power. All topics discussed in this section are described in detail in EE-293. Total power dissipation has two components:

1. Static, including leakage current
2. Dynamic, due to transistor switching characteristics

Many operating conditions can also affect power dissipation, including temperature, voltage, operating frequency, and processor activity. Electrical Characteristics on Page 21 shows the current dissipation for internal circuitry ($V_{DDINT}$).

### *Asynchronous Memory Read Cycle Timing*

**Table 18. Asynchronous Memory Read Cycle Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SDAT}$ | DATA31–0 Setup Before CLKOUT | 2.1 | | ns |
| $t_{HDAT}$ | DATA31–0 Hold After CLKOUT | 0.8 | | ns |
| $t_{SARDY}$ | ARDY Setup Before CLKOUT | 4.0 | | ns |
| $t_{HARDY}$ | ARDY Hold After CLKOUT | 0.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DO}$ | Output Delay After CLKOUT[1] | | 6.0 | ns |
| $t_{HO}$ | Output Hold After CLKOUT [1] | 0.8 | | ns |

[1] Output pins include $\overline{AMS3-0}$, $\overline{ABE3-0}$, ADDR25–2, $\overline{AOE}$, $\overline{ARE}$.

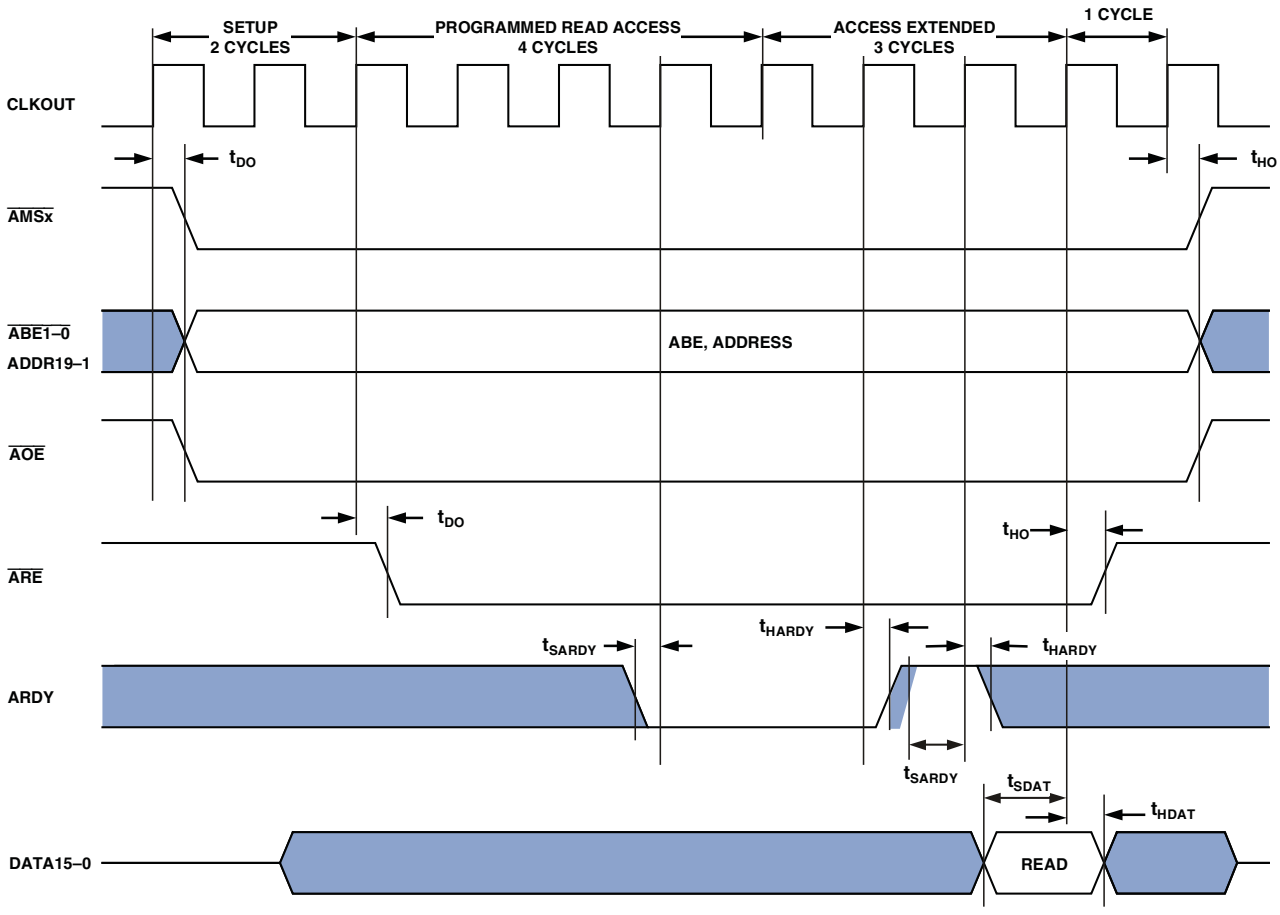


*Figure 10. Asynchronous Memory Read Cycle Timing*

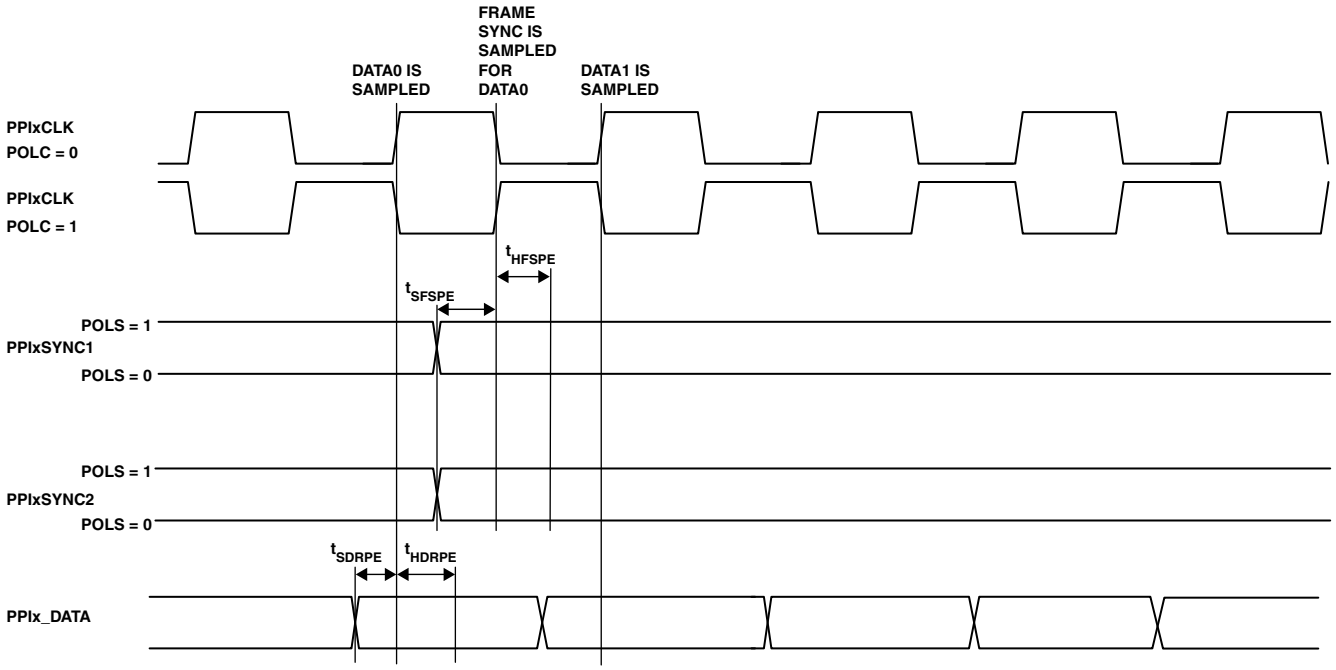


*Figure 15. PPI GP Rx Mode with External Frame Sync Timing (Default)*

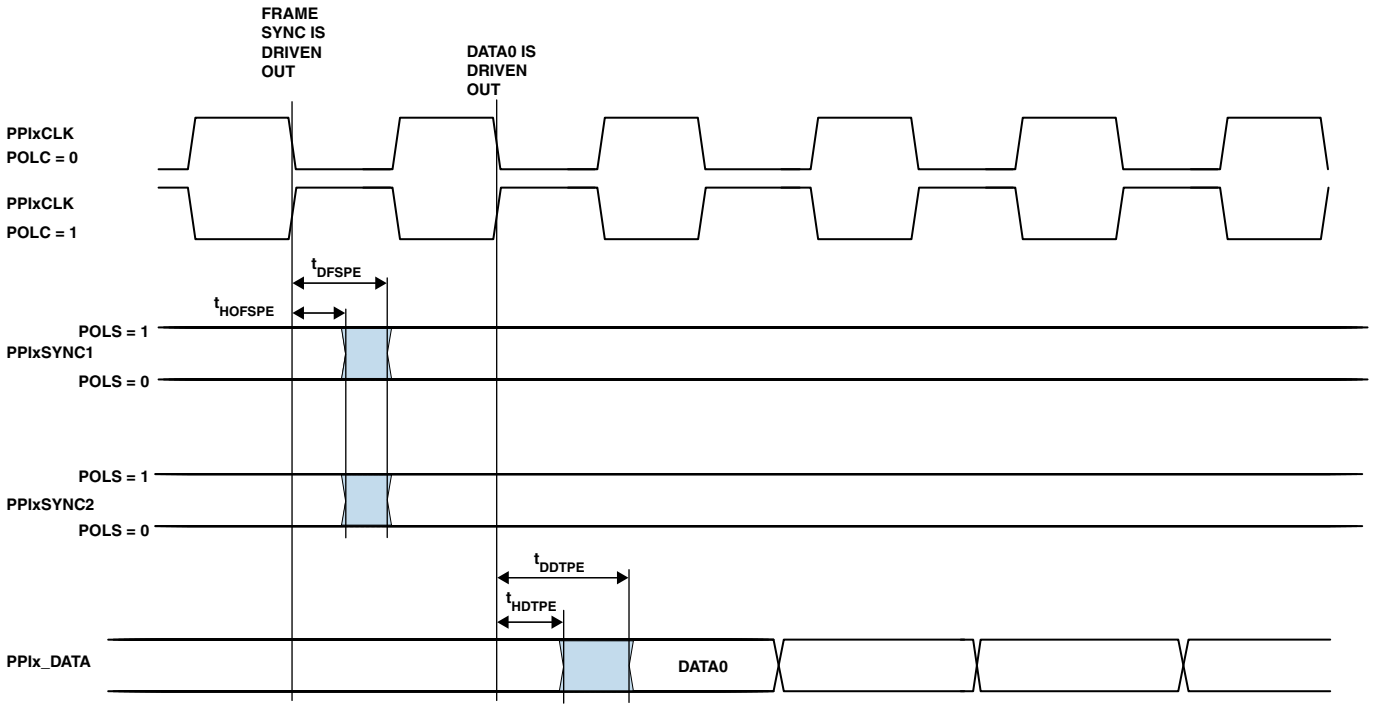


*Figure 16. PPI GP Tx Mode with Internal Frame Sync Timing (Default)*

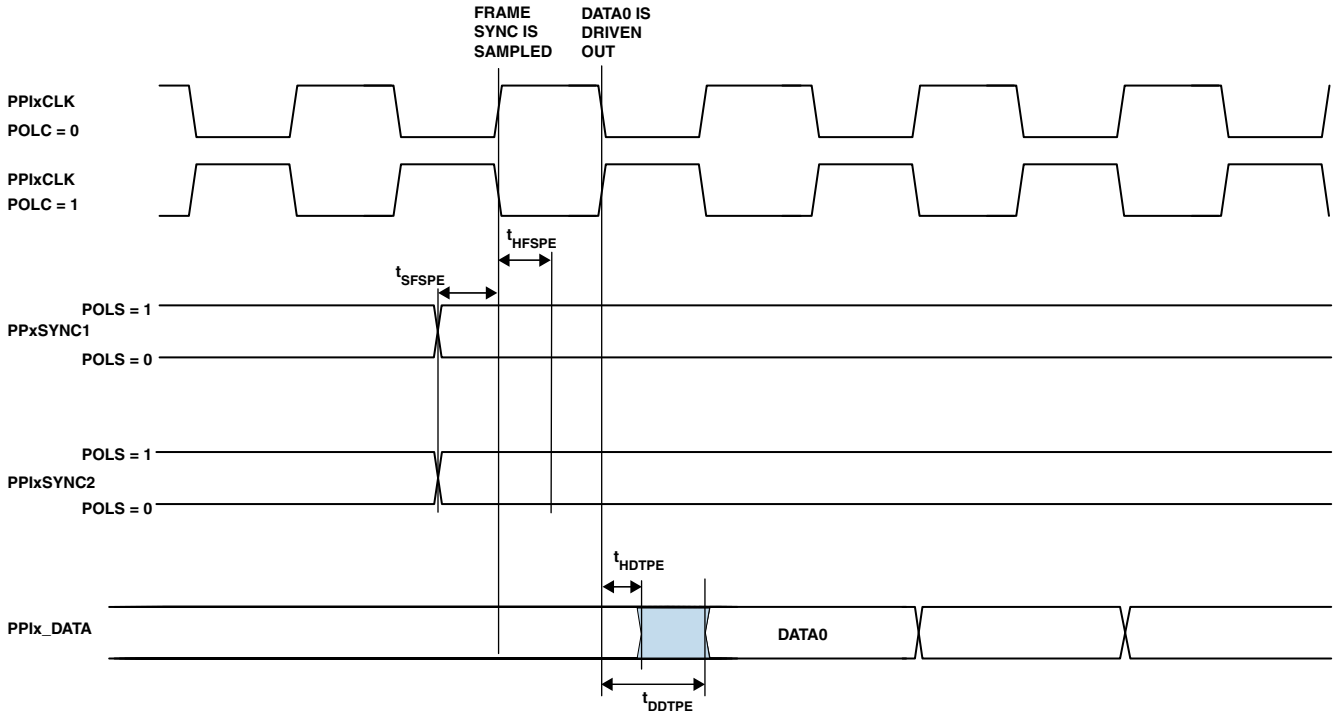


*Figure 17. PPI GP Tx Mode with External Frame Sync Timing (Default)*

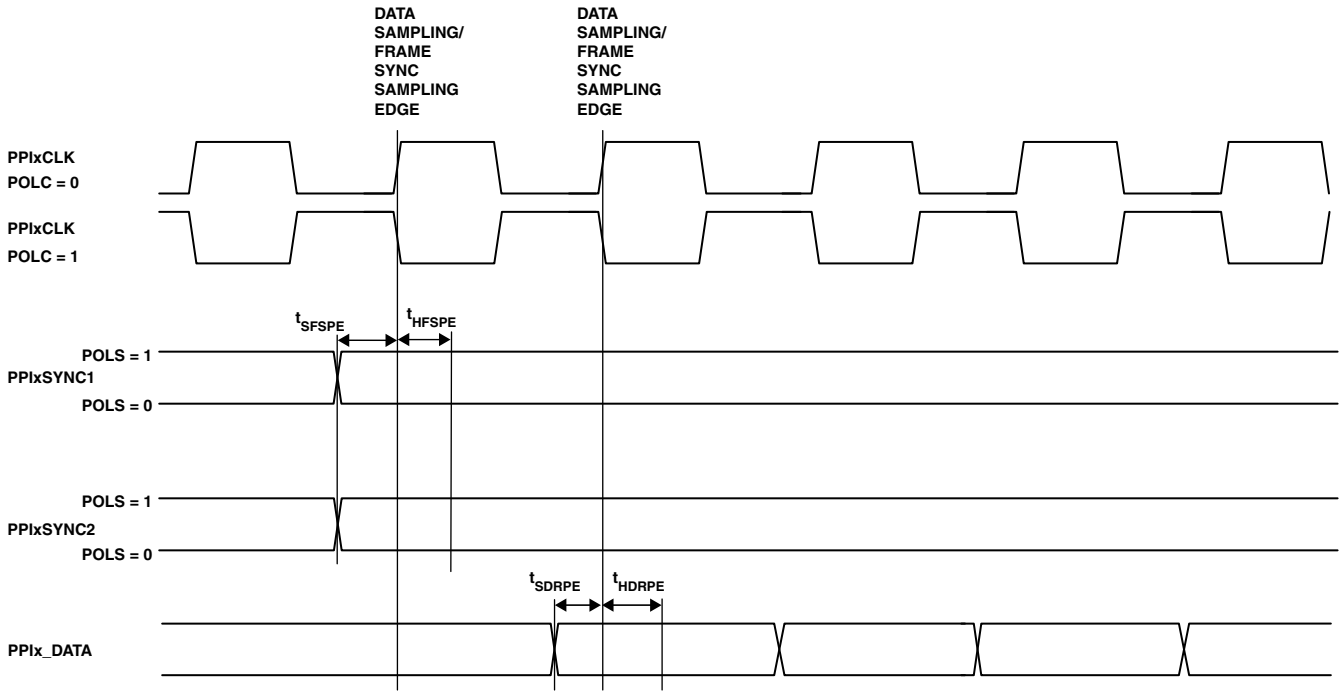


*Figure 18. PPI GP Rx Mode with External Frame Sync Timing (Bit 4 of PLL_CTL Set)*

**Table 25. Serial Ports—Enable and Three-State**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{DTENE}$ | Data Enable Delay from External TSCLKx[1] | 0 | | ns |
| $t_{DDTTE}$ | Data Disable Delay from External TSCLKx[1] | | 10.0 | ns |
| $t_{DTENI}$ | Data Enable Delay from Internal TSCLKx[1] | –2.0 | | ns |
| $t_{DDTTI}$ | Data Disable Delay from Internal TSCLKx[1] | | 3.0 | ns |

[1] Referenced to drive edge.

**Table 26. External Late Frame Sync**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{DDTLFSE}$ | Data Delay from Late External TFSx or External RFSx with MCMEN = 1, MFD = 0[1, 2] | | 10.0 | ns |
| $t_{DTENLFS}$ | Data Enable from Late FS or MCMEN = 1, MFD = 0[1, 2] | 0 | | ns |

[1] MCMEN = 1, TFSx enable and TFSx valid follow $t_{DTENLFS}$ and $t_{DDTLFSE}$.
[2] If external RFSx/TFSx setup to RSCLKx/TSCLKx > $t_{SCLKE}/2$, then $t_{DDTTE/I}$ and $t_{DTENE/I}$ apply; otherwise $t_{DDTLFSE}$ and $t_{DTENLFS}$ apply.

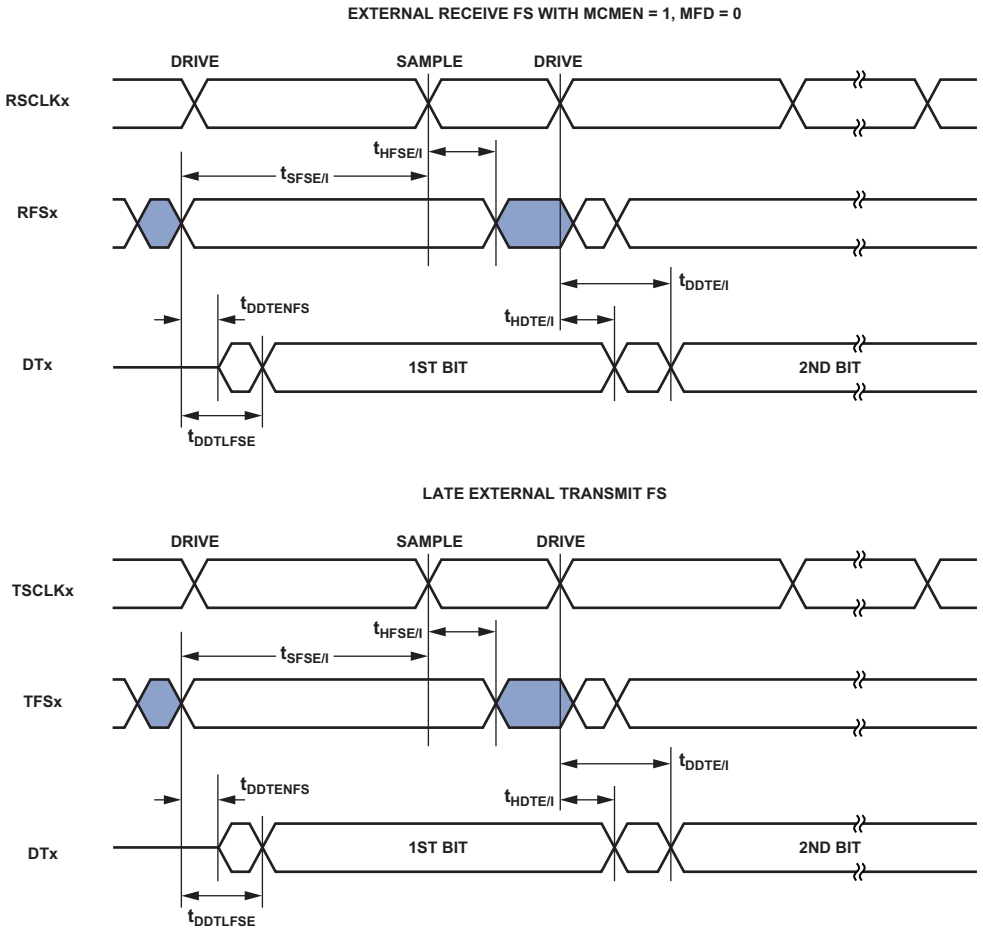


*Figure 22. External Late Frame Sync*

### *Universal Asynchronous Receiver Transmitter (UART) Port—Receive and Transmit Timing*

Figure 25 describes UART port receive and transmit operations. The maximum baud rate is SCLK/16. As shown in Figure 25, there is some latency between the generation internal UART interrupts and the external data operations. These latencies are negligible at the data transmission rates for the UART.

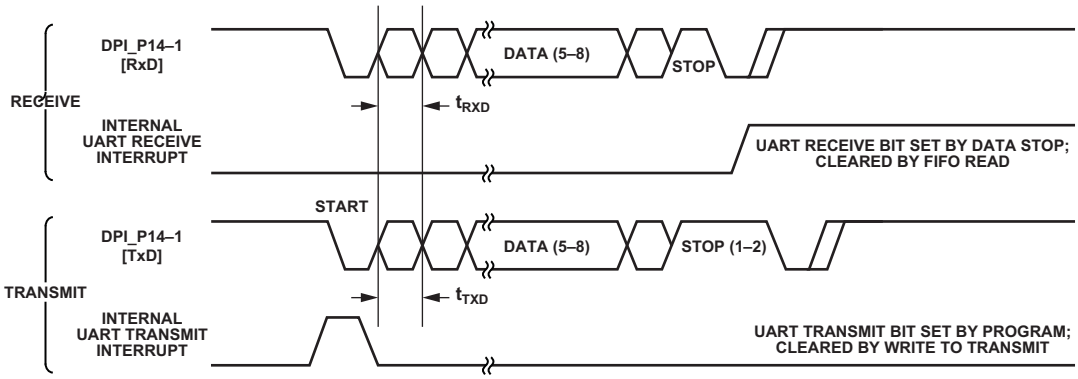


*Figure 25. UART Port—Receive and Transmit Timing*

# 256-BALL CSP_BGA (17 mm) BALL ASSIGNMENT

Table 35 lists the 256-Ball CSP_BGA (17 mm × 17 mm) ball assignment by ball number. Table 36 on Page 48 lists the ball assignment alphabetically by signal.

**Table 35. 256-Ball CSP_BGA (17 mm × 17 mm) Ball Assignment (Numerically by Ball Number)**

| Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal |
|---|---|---|---|---|---|---|---|---|---|
| A1 | VDDEXT | C9 | $\overline{\text{SMS3}}$ | F1 | CLKIN | H9 | GND | L1 | PPI0D3 |
| A2 | ADDR22 | C10 | $\overline{\text{SWE}}$ | F2 | PPI0D10 | H10 | GND | L2 | PPI0D2 |
| A3 | ADDR18 | C11 | SA10 | F3 | $\overline{\text{RESET}}$ | H11 | GND | L3 | PPI0D1 |
| A4 | ADDR14 | C12 | $\overline{\text{ABE0}}$ | F4 | BYPASS | H12 | GND | L4 | PPI0D0 |
| A5 | ADDR11 | C13 | ADDR07 | F5 | VDDEXT | H13 | GND | L5 | VDDEXT |
| A6 | $\overline{\text{AMS3}}$ | C14 | ADDR04 | F6 | VDDEXT | H14 | DATA21 | L6 | VDDEXT |
| A7 | $\overline{\text{AMS0}}$ | C15 | DATA0 | F7 | VDDEXT | H15 | DATA19 | L7 | VDDEXT |
| A8 | ARDY | C16 | DATA05 | F8 | GND | H16 | DATA23 | L8 | VDDEXT |
| A9 | $\overline{\text{SMS2}}$ | D1 | PPI0D15 | F9 | GND | J1 | VROUT1 | L9 | GND |
| A10 | SCLK0 | D2 | PPI0SYNC3 | F10 | VDDEXT | J2 | PPI0D8 | L10 | VDDEXT |
| A11 | SCLK1 | D3 | PPI0SYNC2 | F11 | VDDEXT | J3 | PPI0D7 | L11 | VDDEXT |
| A12 | $\overline{\text{ABE2}}$ | D4 | ADDR21 | F12 | VDDEXT | J4 | PPI0D9 | L12 | VDDEXT |
| A13 | $\overline{\text{ABE3}}$ | D5 | ADDR15 | F13 | DATA11 | J5 | GND | L13 | NC |
| A14 | ADDR06 | D6 | ADDR09 | F14 | DATA08 | J6 | GND | L14 | DT0PRI |
| A15 | ADDR03 | D7 | $\overline{\text{AWE}}$ | F15 | DATA10 | J7 | GND | L15 | DATA31 |
| A16 | VDDEXT | D8 | $\overline{\text{SMS0}}$ | F16 | DATA16 | J8 | GND | L16 | DATA28 |
| B1 | ADDR24 | D9 | $\overline{\text{SRAS}}$ | G1 | XTAL | J9 | GND | M1 | PPI1SYNC2 |
| B2 | ADDR23 | D10 | $\overline{\text{SCAS}}$ | G2 | VDDEXT | J10 | GND | M2 | PPI1D15 |
| B3 | ADDR19 | D11 | $\overline{\text{BGH}}$ | G3 | VDDEXT | J11 | GND | M3 | PPI1D14 |
| B4 | ADDR17 | D12 | $\overline{\text{ABE1}}$ | G4 | GND | J12 | VDDINT | M4 | PPI1D9 |
| B5 | ADDR12 | D13 | DATA02 | G5 | GND | J13 | VDDINT | M5 | VDDINT |
| B6 | ADDR10 | D14 | DATA01 | G6 | VDDEXT | J14 | DATA20 | M6 | VDDINT |
| B7 | $\overline{\text{AMS1}}$ | D15 | DATA03 | G7 | GND | J15 | DATA22 | M7 | GND |
| B8 | $\overline{\text{AOE}}$ | D16 | DATA07 | G8 | GND | J16 | DATA24 | M8 | VDDINT |
| B9 | $\overline{\text{SMS1}}$ | E1 | PPI0D11 | G9 | GND | K1 | PPI0D6 | M9 | GND |
| B10 | SCKE | E2 | PPI0D13 | G10 | GND | K2 | PPI0D5 | M10 | VDDINT |
| B11 | $\overline{\text{BR}}$ | E3 | PPI0D12 | G11 | VDDEXT | K3 | PPI0D4 | M11 | GND |
| B12 | $\overline{\text{BG}}$ | E4 | PPI0D14 | G12 | VDDEXT | K4 | PPI1SYNC3 | M12 | VDDINT |
| B13 | ADDR08 | E5 | PPI1CLK | G13 | DATA17 | K5 | VDDEXT | M13 | RSCLK0 |
| B14 | ADDR05 | E6 | VDDINT | G14 | DATA14 | K6 | VDDEXT | M14 | DR0PRI |
| B15 | ADDR02 | E7 | GND | G15 | DATA15 | K7 | GND | M15 | TSCLK0 |
| B16 | DATA04 | E8 | VDDINT | G16 | DATA18 | K8 | GND | M16 | DATA29 |
| C1 | PPI0SYNC1 | E9 | GND | H1 | VROUT0 | K9 | GND | N1 | PPI1SYNC1 |
| C2 | ADDR25 | E10 | VDDINT | H2 | GND | K10 | GND | N2 | PPI1D10 |
| C3 | PPI0CLK | E11 | GND | H3 | GND | K11 | VDDEXT | N3 | PPI1D7 |
| C4 | ADDR20 | E12 | VDDINT | H4 | VDDINT | K12 | GND | N4 | PPI1D5 |
| C5 | ADDR16 | E13 | DATA06 | H5 | VDDINT | K13 | GND | N5 | PF0 |
| C6 | ADDR13 | E14 | DATA13 | H6 | GND | K14 | DATA26 | N6 | PF04 |
| C7 | $\overline{\text{AMS2}}$ | E15 | DATA09 | H7 | GND | K15 | DATA25 | N7 | PF09 |
| C8 | $\overline{\text{ARE}}$ | E16 | DATA12 | H8 | GND | K16 | DATA27 | N8 | PF12 |

**Table 36. 256-Ball CSP_BGA (17 mm × 17 mm) Ball Assignment (Alphabetically by Signal) (Continued)**

| Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. |
|---|---|---|---|---|---|---|---|---|---|
| $\overline{SMS1}$ | B9 | TSCLK0 | M15 | VDDEXT | G3 | VDDEXT | L11 | VDDINT | M5 |
| $\overline{SMS2}$ | A9 | TSCLK1 | R14 | VDDEXT | G6 | VDDEXT | L12 | VDDINT | M6 |
| $\overline{SMS3}$ | C9 | TX | T13 | VDDEXT | G11 | VDDEXT | T1 | VDDINT | M8 |
| $\overline{SRAS}$ | D9 | VDDEXT | A1 | VDDEXT | G12 | VDDEXT | T16 | VDDINT | M10 |
| $\overline{SWE}$ | C10 | VDDEXT | A16 | VDDEXT | K5 | VDDINT | E6 | VDDINT | M12 |
| TCK | R9 | VDDEXT | F5 | VDDEXT | K6 | VDDINT | E8 | VROUT0 | H1 |
| TDI | T10 | VDDEXT | F6 | VDDEXT | K11 | VDDINT | E10 | VROUT1 | J1 |
| TDO | T9 | VDDEXT | F7 | VDDEXT | L5 | VDDINT | E12 | XTAL | G1 |
| TFS0 | R16 | VDDEXT | F10 | VDDEXT | L6 | VDDINT | H4 | | |
| TFS1 | P14 | VDDEXT | F11 | VDDEXT | L7 | VDDINT | H5 | | |
| TMS | P10 | VDDEXT | F12 | VDDEXT | L8 | VDDINT | J12 | | |
| $\overline{TRST}$ | R10 | VDDEXT | G2 | VDDEXT | L10 | VDDINT | J13 | | |

# 256-BALL CSP_BGA (12 mm) BALL ASSIGNMENT

Table 37 lists the 256-Ball CSP_BGA (12 mm × 12 mm) ball assignment by ball number. Table 38 on Page 53 lists the ball assignment alphabetically by signal.

**Table 37. 256-Ball CSP_BGA (12 mm × 12 mm) Ball Assignment (Numerically by Ball Number)**

| Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal |
|---|---|---|---|---|---|---|---|---|---|
| A01 | VDDEXT | C09 | $\overline{\text{SMS2}}$ | F01 | CLKIN | H09 | GND | L01 | PPI0D0 |
| A02 | ADDR24 | C10 | $\overline{\text{SRAS}}$ | F02 | VDDEXT | H10 | GND | L02 | PPI1SYNC2 |
| A03 | ADDR20 | C11 | GND | F03 | $\overline{\text{RESET}}$ | H11 | VDDINT | L03 | GND |
| A04 | VDDEXT | C12 | $\overline{\text{BGH}}$ | F04 | PPI0D10 | H12 | DATA16 | L04 | PPI1SYNC3 |
| A05 | ADDR14 | C13 | GND | F05 | ADDR21 | H13 | DATA18 | L05 | VDDEXT |
| A06 | ADDR10 | C14 | ADDR07 | F06 | ADDR17 | H14 | DATA20 | L06 | PPI1D11 |
| A07 | $\overline{\text{AMS3}}$ | C15 | DATA1 | F07 | VDDINT | H15 | DATA17 | L07 | GND |
| A08 | $\overline{\text{AWE}}$ | C16 | DATA3 | F08 | GND | H16 | DATA19 | L08 | VDDINT |
| A09 | VDDEXT | D01 | PPI0D13 | F09 | VDDINT | J01 | VROUT0 | L09 | GND |
| A10 | $\overline{\text{SMS3}}$ | D02 | PPI0D15 | F10 | GND | J02 | VROUT1 | L10 | VDDEXT |
| A11 | SCLK0 | D03 | PPI0SYNC3 | F11 | ADDR08 | J03 | PPI0D2 | L11 | GND |
| A12 | SCLK1 | D04 | ADDR23 | F12 | DATA10 | J04 | PPI0D3 | L12 | DR0PRI |
| A13 | $\overline{\text{BG}}$ | D05 | GND | F13 | DATA8 | J05 | PPI0D1 | L13 | TFS0 |
| A14 | $\overline{\text{ABE2}}$ | D06 | GND | F14 | DATA12 | J06 | VDDEXT | L14 | GND |
| A15 | $\overline{\text{ABE3}}$ | D07 | ADDR09 | F15 | DATA9 | J07 | GND | L15 | DATA27 |
| A16 | VDDEXT | D08 | GND | F16 | DATA11 | J08 | VDDINT | L16 | DATA29 |
| B01 | PPI1CLK | D09 | ARDY | G01 | XTAL | J09 | VDDINT | M01 | PPI1D15 |
| B02 | ADDR22 | D10 | $\overline{\text{SCAS}}$ | G02 | GND | J10 | VDDINT | M02 | PPI1D13 |
| B03 | ADDR18 | D11 | SA10 | G03 | VDDEXT | J11 | GND | M03 | PPI1D9 |
| B04 | ADDR16 | D12 | VDDEXT | G04 | BYPASS | J12 | DATA30 | M04 | GND |
| B05 | ADDR12 | D13 | ADDR02 | G05 | PPI0D14 | J13 | DATA22 | M05 | NC |
| B06 | VDDEXT | D14 | GND | G06 | GND | J14 | GND | M06 | PF3 |
| B07 | $\overline{\text{AMS1}}$ | D15 | DATA5 | G07 | GND | J15 | DATA21 | M07 | PF7 |
| B08 | $\overline{\text{ARE}}$ | D16 | DATA6 | G08 | GND | J16 | DATA23 | M08 | VDDINT |
| B09 | $\overline{\text{SMS1}}$ | E01 | GND | G09 | VDDINT | K01 | PPI0D6 | M09 | GND |
| B10 | SCKE | E02 | PPI0D11 | G10 | ADDR05 | K02 | PPI0D4 | M10 | BMODE0 |
| B11 | VDDEXT | E03 | PPI0D12 | G11 | ADDR03 | K03 | PPI0D8 | M11 | SCK |
| B12 | $\overline{\text{BR}}$ | E04 | PPI0SYNC1 | G12 | DATA15 | K04 | PPI1SYNC1 | M12 | DR1PRI |
| B13 | $\overline{\text{ABE1}}$ | E05 | ADDR15 | G13 | DATA14 | K05 | PPI1D14 | M13 | NC |
| B14 | ADDR06 | E06 | ADDR13 | G14 | GND | K06 | VDDEXT | M14 | VDDEXT |
| B15 | ADDR04 | E07 | $\overline{\text{AMS2}}$ | G15 | DATA13 | K07 | GND | M15 | DATA31 |
| B16 | DATA0 | E08 | VDDINT | G16 | VDDEXT | K08 | VDDINT | M16 | DT0PRI |
| C01 | PPI0SYNC2 | E09 | $\overline{\text{SMS0}}$ | H01 | GND | K09 | GND | N01 | PPI1D12 |
| C02 | PPI0CLK | E10 | $\overline{\text{SWE}}$ | H02 | GND | K10 | GND | N02 | PPI1D10 |
| C03 | ADDR25 | E11 | $\overline{\text{ABE0}}$ | H03 | PPI0D9 | K11 | VDDINT | N03 | PPI1D3 |
| C04 | ADDR19 | E12 | DATA2 | H04 | PPI0D7 | K12 | DATA28 | N04 | PPI1D1 |
| C05 | GND | E13 | GND | H05 | PPI0D5 | K13 | DATA26 | N05 | PF1 |
| C06 | ADDR11 | E14 | DATA4 | H06 | VDDINT | K14 | DATA24 | N06 | PF9 |
| C07 | $\overline{\text{AOE}}$ | E15 | DATA7 | H07 | VDDINT | K15 | DATA25 | N07 | GND |
| C08 | $\overline{\text{AMS0}}$ | E16 | VDDEXT | H08 | GND | K16 | VDDEXT | N08 | PF13 |

# 297-BALL PBGA BALL ASSIGNMENT

Table 39 lists the 297-Ball PBGA ball assignment numerically by ball number. Table 40 on Page 58 lists the ball assignment alphabetically by signal.

**Table 39. 297-Ball PBGA Ball Assignment (Numerically by Ball Number)**

| Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal |
|---|---|---|---|---|---|---|---|
| A01 | GND | B15 | $\overline{\text{SMS1}}$ | G01 | PPI0D11 | L14 | GND |
| A02 | ADDR25 | B16 | $\overline{\text{SMS3}}$ | G02 | PPI0D10 | L15 | GND |
| A03 | ADDR23 | B17 | SCKE | G25 | DATA4 | L16 | GND |
| A04 | ADDR21 | B18 | $\overline{\text{SWE}}$ | G26 | DATA7 | L17 | GND |
| A05 | ADDR19 | B19 | SA10 | H01 | BYPASS | L18 | VDDINT |
| A06 | ADDR17 | B20 | $\overline{\text{BR}}$ | H02 | $\overline{\text{RESET}}$ | L25 | DATA12 |
| A07 | ADDR15 | B21 | $\overline{\text{BG}}$ | H25 | DATA6 | L26 | DATA15 |
| A08 | ADDR13 | B22 | $\overline{\text{ABE1}}$ | H26 | DATA9 | M01 | VROUT0 |
| A09 | ADDR11 | B23 | $\overline{\text{ABE3}}$ | J01 | CLKIN | M02 | GND |
| A10 | ADDR09 | B24 | ADDR07 | J02 | GND | M10 | VDDEXT |
| A11 | $\overline{\text{AMS3}}$ | B25 | GND | J10 | VDDEXT | M11 | GND |
| A12 | $\overline{\text{AMS1}}$ | B26 | ADDR05 | J11 | VDDEXT | M12 | GND |
| A13 | $\overline{\text{AWE}}$ | C01 | PPI0SYNC3 | J12 | VDDEXT | M13 | GND |
| A14 | $\overline{\text{ARE}}$ | C02 | PPI0CLK | J13 | VDDEXT | M14 | GND |
| A15 | $\overline{\text{SMS0}}$ | C03 | GND | J14 | VDDEXT | M15 | GND |
| A16 | $\overline{\text{SMS2}}$ | C04 | GND | J15 | VDDEXT | M16 | GND |
| A17 | $\overline{\text{SRAS}}$ | C05 | GND | J16 | VDDINT | M17 | GND |
| A18 | $\overline{\text{SCAS}}$ | C22 | GND | J17 | VDDINT | M18 | VDDINT |
| A19 | SCLK0 | C23 | GND | J18 | VDDINT | M25 | DATA14 |
| A20 | SCLK1 | C24 | GND | J25 | DATA8 | M26 | DATA17 |
| A21 | $\overline{\text{BGH}}$ | C25 | ADDR04 | J26 | DATA11 | N01 | VROUT1 |
| A22 | $\overline{\text{ABE0}}$ | C26 | ADDR03 | K01 | XTAL | N02 | PPI0D9 |
| A23 | $\overline{\text{ABE2}}$ | D01 | PPI0SYNC1 | K02 | NC | N10 | VDDEXT |
| A24 | ADDR08 | D02 | PPI0SYNC2 | K10 | VDDEXT | N11 | GND |
| A25 | ADDR06 | D03 | GND | K11 | VDDEXT | N12 | GND |
| A26 | GND | D04 | GND | K12 | VDDEXT | N13 | GND |
| B01 | PPI1CLK | D23 | GND | K13 | VDDEXT | N14 | GND |
| B02 | GND | D24 | GND | K14 | VDDEXT | N15 | GND |
| B03 | ADDR24 | D25 | ADDR02 | K15 | VDDEXT | N16 | GND |
| B04 | ADDR22 | D26 | DATA1 | K16 | VDDINT | N17 | GND |
| B05 | ADDR20 | E01 | PPI0D15 | K17 | VDDINT | N18 | VDDINT |
| B06 | ADDR18 | E02 | PPI0D14 | K18 | VDDINT | N25 | DATA16 |
| B07 | ADDR16 | E03 | GND | K25 | DATA10 | N26 | DATA19 |
| B08 | ADDR14 | E24 | GND | K26 | DATA13 | P01 | PPI0D7 |
| B09 | ADDR12 | E25 | DATA0 | L01 | NC | P02 | PPI0D8 |
| B10 | ADDR10 | E26 | DATA3 | L02 | NC | P10 | VDDEXT |
| B11 | $\overline{\text{AMS2}}$ | F01 | PPI0D13 | L10 | VDDEXT | P11 | GND |
| B12 | $\overline{\text{AMS0}}$ | F02 | PPI0D12 | L11 | GND | P12 | GND |
| B13 | $\overline{\text{AOE}}$ | F25 | DATA2 | L12 | GND | P13 | GND |
| B14 | ARDY | F26 | DATA5 | L13 | GND | P14 | GND |

**Table 40. 297-Ball PBGA Ball Assignment (Alphabetically by Signal)**

| Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. |
|---|---|---|---|---|---|---|---|
| $\overline{ABE0}$ | A22 | $\overline{BR}$ | B20 | DT0SEC | AA25 | GND | N15 |
| $\overline{ABE1}$ | B22 | BYPASS | H01 | DT1PRI | AF25 | GND | N16 |
| $\overline{ABE2}$ | A23 | CLKIN | J01 | DT1SEC | AF24 | GND | N17 |
| $\overline{ABE3}$ | B23 | DATA0 | E25 | $\overline{EMU}$ | AE16 | GND | P11 |
| ADDR02 | D25 | DATA1 | D26 | GND | A01 | GND | P12 |
| ADDR03 | C26 | DATA2 | F25 | GND | A26 | GND | P13 |
| ADDR04 | C25 | DATA3 | E26 | GND | B02 | GND | P14 |
| ADDR05 | B26 | DATA4 | G25 | GND | B25 | GND | P15 |
| ADDR06 | A25 | DATA5 | F26 | GND | C03 | GND | P16 |
| ADDR07 | B24 | DATA6 | H25 | GND | C04 | GND | P17 |
| ADDR08 | A24 | DATA7 | G26 | GND | C05 | GND | R11 |
| ADDR09 | A10 | DATA8 | J25 | GND | C22 | GND | R12 |
| ADDR10 | B10 | DATA9 | H26 | GND | C23 | GND | R13 |
| ADDR11 | A09 | DATA10 | K25 | GND | C24 | GND | R14 |
| ADDR12 | B09 | DATA11 | J26 | GND | D03 | GND | R15 |
| ADDR13 | A08 | DATA12 | L25 | GND | D04 | GND | R16 |
| ADDR14 | B08 | DATA13 | K26 | GND | D23 | GND | R17 |
| ADDR15 | A07 | DATA14 | M25 | GND | D24 | GND | T11 |
| ADDR16 | B07 | DATA15 | L26 | GND | E03 | GND | T12 |
| ADDR17 | A06 | DATA16 | N25 | GND | E24 | GND | T13 |
| ADDR18 | B06 | DATA17 | M26 | GND | J02 | GND | T14 |
| ADDR19 | A05 | DATA18 | P25 | GND | L11 | GND | T15 |
| ADDR20 | B05 | DATA19 | N26 | GND | L12 | GND | T16 |
| ADDR21 | A04 | DATA20 | R25 | GND | L13 | GND | T17 |
| ADDR22 | B04 | DATA21 | P26 | GND | L14 | GND | U14 |
| ADDR23 | A03 | DATA22 | T25 | GND | L15 | GND | AB03 |
| ADDR24 | B03 | DATA23 | R26 | GND | L16 | GND | AB24 |
| ADDR25 | A02 | DATA24 | U25 | GND | L17 | GND | AC03 |
| $\overline{AMS0}$ | B12 | DATA25 | T26 | GND | M02 | GND | AC04 |
| $\overline{AMS1}$ | A12 | DATA26 | V25 | GND | M11 | GND | AC23 |
| $\overline{AMS2}$ | B11 | DATA27 | U26 | GND | M12 | GND | AC24 |
| $\overline{AMS3}$ | A11 | DATA28 | W25 | GND | M13 | GND | AD03 |
| $\overline{AOE}$ | B13 | DATA29 | V26 | GND | M14 | GND | AD04 |
| ARDY | B14 | DATA30 | Y25 | GND | M15 | GND | AD05 |
| $\overline{ARE}$ | A14 | DATA31 | W26 | GND | M16 | GND | AD22 |
| $\overline{AWE}$ | A13 | DR0PRI | AB26 | GND | M17 | GND | AD23 |
| $\overline{BG}$ | B21 | DR0SEC | AC25 | GND | N11 | GND | AD24 |
| $\overline{BGH}$ | A21 | DR1PRI | AF22 | GND | N12 | GND | AE02 |
| BMODE0 | AE18 | DR1SEC | AE23 | GND | N13 | GND | AE25 |
| BMODE1 | AE17 | DT0PRI | Y26 | GND | N14 | GND | AF01 |

**Table 40. 297-Ball PBGA Ball Assignment (Alphabetically by Signal) (Continued)**

| Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. |
|---|---|---|---|---|---|---|---|
| GND | AF26 | PPI0D7 | P01 | RSCLK0 | AD26 | VDDEXT | K13 |
| MISO | AE19 | PPI0D8 | P02 | RSCLK1 | AF21 | VDDEXT | K14 |
| MOSI | AE20 | PPI0D9 | N02 | RX | AE21 | VDDEXT | K15 |
| NC | K02 | PPI0D10 | G02 | SA10 | B19 | VDDEXT | L10 |
| NC | L01 | PPI0D11 | G01 | $\overline{\text{SCAS}}$ | A18 | VDDEXT | M10 |
| NC | L02 | PPI0D12 | F02 | SCK | AF19 | VDDEXT | N10 |
| NC | AD25 | PPI0D13 | F01 | SCKE | B17 | VDDEXT | P10 |
| NC | AE13 | PPI0D14 | E02 | SCLK0 | A19 | VDDEXT | R10 |
| NC | AE26 | PPI0D15 | E01 | SCLK1 | A20 | VDDEXT | T10 |
| NMI0 | AF18 | PPI0SYNC1 | D01 | SLEEP | AF17 | VDDEXT | U10 |
| NMI1 | AF13 | PPI0SYNC2 | D02 | $\overline{\text{SMS0}}$ | A15 | VDDEXT | U11 |
| PF0 | AE05 | PPI0SYNC3 | C01 | $\overline{\text{SMS1}}$ | B15 | VDDEXT | U12 |
| PF1 | AF05 | PPI1CLK | B01 | $\overline{\text{SMS2}}$ | A16 | VDDEXT | U13 |
| PF2 | AE06 | PPI1D0 | AF04 | $\overline{\text{SMS3}}$ | B16 | VDDINT | J16 |
| PF3 | AF06 | PPI1D1 | AE04 | $\overline{\text{SRAS}}$ | A17 | VDDINT | J17 |
| PF4 | AE07 | PPI1D2 | AF03 | $\overline{\text{SWE}}$ | B18 | VDDINT | J18 |
| PF5 | AF07 | PPI1D3 | AE03 | TCK | AF14 | VDDINT | K16 |
| PF6 | AE08 | PPI1D4 | AF02 | TDI | AF15 | VDDINT | K17 |
| PF7 | AF08 | PPI1D5 | AE01 | TDO | AE14 | VDDINT | K18 |
| PF8 | AE09 | PPI1D6 | AD02 | TFS0 | AB25 | VDDINT | L18 |
| PF9 | AF09 | PPI1D7 | AD01 | TFS1 | AE24 | VDDINT | M18 |
| PF10 | AE10 | PPI1D8 | AC02 | TMS | AF16 | VDDINT | N18 |
| PF11 | AF10 | PPI1D9 | AC01 | $\overline{\text{TRST}}$ | AE15 | VDDINT | P18 |
| PF12 | AE11 | PPI1D10 | AB02 | TSCLK0 | AA26 | VDDINT | R18 |
| PF13 | AF11 | PPI1D11 | AB01 | TSCLK1 | AF23 | VDDINT | T18 |
| PF14 | AE12 | PPI1D12 | AA02 | TX/PF26 | AF20 | VDDINT | U15 |
| PF15 | AF12 | PPI1D13 | AA01 | VDDEXT | J10 | VDDINT | U16 |
| PPI0CLK | C02 | PPI1D14 | Y02 | VDDEXT | J11 | VDDINT | U17 |
| PPI0D0 | V02 | PPI1D15 | Y01 | VDDEXT | J12 | VDDINT | U18 |
| PPI0D1 | U01 | PPI1SYNC1 | W01 | VDDEXT | J13 | VROUT0 | M01 |
| PPI0D2 | U02 | PPI1SYNC2 | W02 | VDDEXT | J14 | VROUT1 | N01 |
| PPI0D3 | T01 | PPI1SYNC3 | V01 | VDDEXT | J15 | XTAL | K01 |
| PPI0D4 | T02 | $\overline{\text{RESET}}$ | H02 | VDDEXT | K10 | | |
| PPI0D5 | R01 | RFS0 | AC26 | VDDEXT | K11 | | |
| PPI0D6 | R02 | RFS1 | AE22 | VDDEXT | K12 | | |

D04696-0-9/09(E)