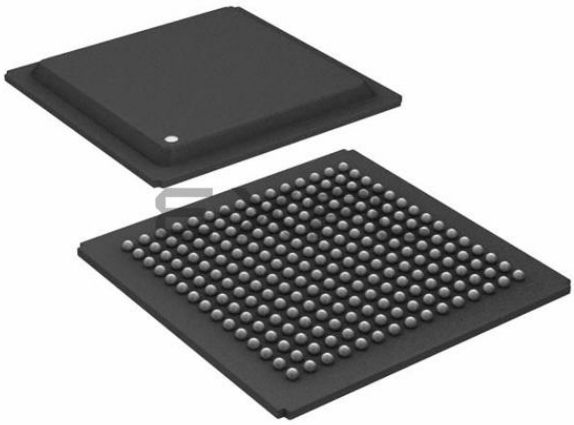

**Welcome to E-XFL.COM**

## 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 | Obsolete |
| Type | Floating Point |
| Interface | Host Interface, Link Port, Serial Port |
| Clock Rate | 33MHz |
| Non-Volatile Memory | External |
| On-Chip RAM | 512kB |
| Voltage - I/O | 3.30V |
| Voltage - Core | 3.30V |
| Operating Temperature | -40°C ~ 100°C (TC) |
| Mounting Type | Surface Mount |
| Package / Case | 225-BBGA |
| Supplier Device Package | 225-PBGA (23x23) |
| Purchase URL | https://www.e-xfl.com/product-detail/analog-devices/adsp-21060lcbz-133 |

### PARALLEL COMPUTATIONS

**Single-cycle multiply and ALU operations in parallel with dual memory read/writes and instruction fetch**

**Multiply with add and subtract for accelerated FFT butterfly computation**

### UP TO 4M BIT ON-CHIP SRAM

**Dual-ported for independent access by core processor and DMA**

### OFF-CHIP MEMORY INTERFACING

**4 gigawords addressable**

**Programmable wait state generation, page-mode DRAM support**

### DMA CONTROLLER

**10 DMA channels for transfers between ADSP-2106x internal memory and external memory, external peripherals, host processor, serial ports, or link ports**

**Background DMA transfers at up to 40 MHz, in parallel with full-speed processor execution**

### HOST PROCESSOR INTERFACE TO 16- AND 32-BIT MICROPROCESSORS

**Host can directly read/write ADSP-2106x internal memory and IOP registers**

### MULTIPROCESSING

**Glueless connection for scalable DSP multiprocessing architecture**

**Distributed on-chip bus arbitration for parallel bus connect of up to six ADSP-2106xs plus host**

**Six link ports for point-to-point connectivity and array multiprocessing**

**240 MBps transfer rate over parallel bus**

**240 MBps transfer rate over link ports**

### SERIAL PORTS

**Two 40 Mbps synchronous serial ports with companding hardware**

**Independent transmit and receive functions**

**Table 1. ADSP-2106x SHARC Processor Family Features**

| Feature | ADSP-21060 | ADSP-21062 | ADSP-21060L | ADSP-21062L | ADSP-21060C | ADSP-21060LC |
|---|---|---|---|---|---|---|
| SRAM | 4M bits | 2M bits | 4M bits | 2M bits | 4M bits | 4M bits |
| Operating Voltage | 5 V | 5 V | 3.3 V | 3.3 V | 5 V | 3.3 V |
| Instruction Rate | 33 MHz<br>40 MHz | 33 MHz<br>40 MHz | 33 MHz<br>40 MHz | 33 MHz<br>40 MHz | 33 MHz<br>40 MHz | 33 MHz<br>40 MHz |
| Package | MQFP_PQ4<br>PBGA | MQFP_PQ4<br>PBGA | MQFP_PQ4<br>PBGA | MQFP_PQ4<br>PBGA | CQFP | CQFP |

# CONTENTS

Summary ..... 1
General Description ..... 4
SHARC Family Core Architecture ..... 4
Memory and I/O Interface Features ..... 5
Development Tools ..... 8
Additional Information ..... 9
Related Signal Chains ..... 9
Pin Function Descriptions ..... 10
Target Board Connector for EZ-ICE Probe ..... 13
ADSP-21060/ADSP-21062 Specifications ..... 15
Operating Conditions (5 V) ..... 15
Electrical Characteristics (5 V) ..... 15
Internal Power Dissipation (5 V) ..... 16
External Power Dissipation (5 V) ..... 17
ADSP-21060L/ADSP-21062L Specifications ..... 18
Operating Conditions (3.3 V) ..... 18
Electrical Characteristics (3.3 V) ..... 18
Internal Power Dissipation (3.3 V) ..... 19
External Power Dissipation (3.3 V) ..... 20
Absolute Maximum Ratings ..... 20
ESD Caution ..... 21
Package Marking Information ..... 21
Timing Specifications ..... 21
Test Conditions ..... 48
Environmental Conditions ..... 51
225-Ball PBGA Ball Configuration ..... 52
240-Lead MQFP_PQ4/CQFP Pin Configuration ..... 54
Outline Dimensions ..... 56
Surface-Mount Design ..... 61
Ordering Guide ..... 62

## REVISION HISTORY

**3/13—Rev. G to Rev. H**

Updated Development Tools ..... 8

Corrected the power dissipation equation from $P_{TOTAL} = P_{EXT} + (I_{DDIN2} \times 5.0\ V)$ to $P_{TOTAL} = P_{EXT} + (I_{DDIN2} \times 3.3\ V)$
External Power Dissipation (3.3 V) ..... 20

### *Single-Cycle Fetch of Instruction and Two Operands*

The ADSP-2106x features an enhanced Harvard architecture in which the data memory (DM) bus transfers data and the program memory (PM) bus transfers both instructions and data (see Figure 1 on Page 1). With its separate program and data memory buses and on-chip instruction cache, the processor can simultaneously fetch two operands and an instruction (from the cache), all in a single cycle.

### *Instruction Cache*

The ADSP-2106x includes an on-chip instruction cache that enables three-bus operation for fetching an instruction and two data values. The cache is selective—only the instructions whose fetches conflict with PM bus data accesses are cached. This allows full-speed execution of core, looped operations such as digital filter multiply-accumulates and FFT butterfly processing.

### *Data Address Generators with Hardware Circular Buffers*

The ADSP-2106x's two data address generators (DAGs) implement circular data buffers in hardware. Circular buffers allow efficient programming of delay lines and other data structures required in digital signal processing, and are commonly used in digital filters and Fourier transforms. The two DAGs of the ADSP-2106x contain sufficient registers to allow the creation of up to 32 circular buffers (16 primary register sets, 16 secondary). The DAGs automatically handle address pointer wraparound, reducing overhead, increasing performance and simplifying implementation. Circular buffers can start and end at any memory location.

### *Flexible Instruction Set*

The 48-bit instruction word accommodates a variety of parallel operations, for concise programming. For example, the ADSP-2106x can conditionally execute a multiply, an add, a subtract and a branch, all in a single instruction.

## MEMORY AND I/O INTERFACE FEATURES

The ADSP-2106x processors add the following architectural features to the SHARC family core.

### *Dual-Ported On-Chip Memory*

The ADSP-21062/ADSP-21062L contains two megabits of on-chip SRAM, and the ADSP-21060/ADSP-21060L contains 4M bits of on-chip SRAM. The internal memory is organized as two equal sized blocks of 1M bit each for the ADSP-21062/ADSP-21062L and two equal sized blocks of 2M bits each for the ADSP-21060/ADSP-21060L. Each can be configured for different combinations of code and data storage. Each memory block is dual-ported for single-cycle, independent accesses by the core processor and I/O processor or DMA controller. The dual-ported memory and separate on-chip buses allow two data transfers from the core and one from I/O, all in a single cycle.

On the ADSP-21062/ADSP-21062L, the memory can be configured as a maximum of 64k words of 32-bit data, 128k words of 16-bit data, 40k words of 48-bit instructions (or 40-bit data), or combinations of different word sizes up to two megabits. All of the memory can be accessed as 16-bit, 32-bit, or 48-bit words.

On the ADSP-21060/ADSP-21060L, the memory can be configured as a maximum of 128k words of 32-bit data, 256k words of 16-bit data, 80k words of 48-bit instructions (or 40-bit data), or combinations of different word sizes up to four megabits. All of the memory can be accessed as 16-bit, 32-bit or 48-bit words.

A 16-bit floating-point storage format is supported, which effectively doubles the amount of data that can be stored on-chip. Conversion between the 32-bit floating-point and 16-bit floating-point formats is done in a single instruction.

While each memory block can store combinations of code and data, accesses are most efficient when one block stores data, using the DM bus for transfers, and the other block stores instructions and data, using the PM bus for transfers. Using the DM bus and PM bus in this way, with one dedicated to each memory block, assures single-cycle execution with two data transfers. In this case, the instruction must be available in the cache. Single-cycle execution is also maintained when one of the data operands is transferred to or from off-chip, via the ADSP-2106x's external port.

### *On-Chip Memory and Peripherals Interface*

The ADSP-2106x's external port provides the processor's interface to off-chip memory and peripherals. The 4-gigaword off-chip address space is included in the ADSP-2106x's unified address space. The separate on-chip buses—for PM addresses, PM data, DM addresses, DM data, I/O addresses, and I/O data—are multiplexed at the external port to create an external system bus with a single 32-bit address bus and a single 48-bit (or 32-bit) data bus.

Addressing of external memory devices is facilitated by on-chip decoding of high-order address lines to generate memory bank select signals. Separate control lines are also generated for simplified addressing of page-mode DRAM. The ADSP-2106x provides programmable memory wait states and external memory acknowledge controls to allow interfacing to DRAM and peripherals with variable access, hold and disable time requirements.

### *Host Processor Interface*

The ADSP-2106x's host interface allows easy connection to standard microprocessor buses, both 16-bit and 32-bit, with little additional hardware required. Asynchronous transfers at speeds up to the full clock rate of the processor are supported. The host interface is accessed through the ADSP-2106x's external port and is memory-mapped into the unified address space. Four channels of DMA are available for the host interface; code and data transfers are accomplished with low software overhead.

The host processor requests the ADSP-2106x's external bus with the host bus request ($\overline{\text{HBR}}$), host bus grant ($\overline{\text{HBG}}$), and ready (REDY) signals. The host can directly read and write the internal memory of the ADSP-2106x, and can access the DMA channel setup and mailbox registers. Vector interrupt support is provided for efficient execution of host commands.

### Link Ports

The ADSP-2106x features six 4-bit link ports that provide additional I/O capabilities. The link ports can be clocked twice per cycle, allowing each to transfer eight bits of data per cycle. Link port I/O is especially useful for point-to-point interprocessor communication in multiprocessing systems.

The link ports can operate independently and simultaneously, with a maximum data throughput of 240M bytes/s. Link port data is packed into 32- or 48-bit words, and can be directly read by the core processor or DMA-transferred to on-chip memory.

Each link port has its own double-buffered input and output registers. Clock/acknowledge handshaking controls link port transfers. Transfers are programmable as either transmit or receive.

### Program Booting

The internal memory of the ADSP-2106x can be booted at system power-up from an 8-bit EPROM, a host processor, or through one of the link ports. Selection of the boot source is controlled by the BMS (boot memory select), EBOOT (EPROM Boot), and LBOOT (link/host boot) pins. 32-bit and 16-bit host processors can be used for booting. The processor also supports a no-boot mode in which instruction execution is sourced from the external memory.

## DEVELOPMENT TOOLS

Analog Devices supports its processors with a complete line of software and hardware development tools, including integrated development environments (which include CrossCore® Embedded Studio and/or VisualDSP++®), evaluation products, emulators, and a wide variety of software add-ins.

### Integrated Development Environments (IDEs)

For C/C++ software writing and editing, code generation, and debug support, Analog Devices offers two IDEs.

The newest IDE, CrossCore Embedded Studio, is based on the Eclipse™ framework. Supporting most Analog Devices processor families, it is the IDE of choice for future processors, including multicore devices. CrossCore Embedded Studio seamlessly integrates available software add-ins to support real time operating systems, file systems, TCP/IP stacks, USB stacks, algorithmic software modules, and evaluation hardware board support packages. For more information visit www.analog.com/cces.

The other Analog Devices IDE, VisualDSP++, supports processor families introduced prior to the release of CrossCore Embedded Studio. This IDE includes the Analog Devices VDK real time operating system and an open source TCP/IP stack. For more information visit www.analog.com/visualdsp. Note that VisualDSP++ will not support future Analog Devices processors.

### EZ-KIT Lite Evaluation Board

For processor evaluation, Analog Devices provides wide range of EZ-KIT Lite® evaluation boards. Including the processor and key peripherals, the evaluation board also supports on-chip emulation capabilities and other evaluation and development features. Also available are various EZ-Extenders®, which are daughter cards delivering additional specialized functionality, including audio and video processing. For more information visit www.analog.com and search on “ezkit” or “ezextender”.

### EZ-KIT Lite Evaluation Kits

For a cost-effective way to learn more about developing with Analog Devices processors, Analog Devices offer a range of EZ-KIT Lite evaluation kits. Each evaluation kit includes an EZ-KIT Lite evaluation board, directions for downloading an evaluation version of the available IDE(s), a USB cable, and a power supply. The USB controller on the EZ-KIT Lite board connects to the USB port of the user’s PC, enabling the chosen IDE evaluation suite to emulate the on-board processor in-circuit. This permits the customer to download, execute, and debug programs for the EZ-KIT Lite system. It also supports in-circuit programming of the on-board Flash device to store user-specific boot code, enabling standalone operation. With the full version of CrossCore Embedded Studio or VisualDSP++ installed (sold separately), engineers can develop software for supported EZ-KITs or any custom system utilizing supported Analog Devices processors.

### Software Add-Ins for CrossCore Embedded Studio

Analog Devices offers software add-ins which seamlessly integrate with CrossCore Embedded Studio to extend its capabilities and reduce development time. Add-ins include board support packages for evaluation hardware, various middleware packages, and algorithmic modules. Documentation, help, configuration dialogs, and coding examples present in these add-ins are viewable through the CrossCore Embedded Studio IDE once the add-in is installed.

### Board Support Packages for Evaluation Hardware

Software support for the EZ-KIT Lite evaluation boards and EZ-Extender daughter cards is provided by software add-ins called Board Support Packages (BSPs). The BSPs contain the required drivers, pertinent release notes, and select example code for the given evaluation hardware. A download link for a specific BSP is located on the web page for the associated EZ-KIT or EZ-Extender product. The link is found in the *Product Download* area of the product web page.

### Middleware Packages

Analog Devices separately offers middleware add-ins such as real time operating systems, file systems, USB stacks, and TCP/IP stacks. For more information see the following web pages:

- www.analog.com/ucos3
- www.analog.com/ucfs
- www.analog.com/ucusbd
- www.analog.com/lwip

## INTERNAL POWER DISSIPATION (5 V)

These specifications apply to the internal power portion of $V_{DD}$ only. For a complete discussion of the code used to measure power dissipation, see the technical note "SHARC Power Dissipation Measurements."

Specifications are based on the operating scenarios.

| Operation | Peak Activity ($I_{DDINPEAK}$) | High Activity ($I_{DDINHIGH}$) | Low Activity ($I_{DDINLOW}$) |
|---|---|---|---|
| Instruction Type | Multifunction | Multifunction | Single Function |
| Instruction Fetch | Cache | Internal Memory | Internal Memory |
| Core memory Access | 2 Per Cycle (DM and PM) | 1 Per Cycle (DM) | None |
| Internal Memory DMA | 1 Per Cycle | 1 Per 2 Cycles | 1 Per 2 Cycles |

To estimate power consumption for a specific application, use the following equation where% is the amount of time your program spends in that state:

$$\%PEAK\ I_{DDINPEAK} + \%HIGH\ I_{DDINHIGH} + \%LOW\ I_{DDINLOW} + \%IDLE\ I_{DDIDLE} = Power\ Consumption$$

| Parameter | Test Conditions | Max | Unit |
|---|---|---|---|
| $I_{DDINPEAK}$ Supply Current (Internal)[1] | $t_{CK}$ = 30 ns, $V_{DD}$ = Max | 745 | mA |
| | $t_{CK}$ = 25 ns, $V_{DD}$ = Max | 850 | mA |
| $I_{DDINHIGH}$ Supply Current (Internal)[2] | $t_{CK}$ = 30 ns, $V_{DD}$ = Max | 575 | mA |
| | $t_{CK}$ = 25 ns, $V_{DD}$ = Max | 670 | mA |
| $I_{DDINLOW}$ Supply Current (Internal)[2] | $t_{CK}$ = 30 ns, $V_{DD}$ = Max | 340 | mA |
| | $t_{CK}$ = 25 ns, $V_{DD}$ = Max | 390 | mA |
| $I_{DDIDLE}$ Supply Current (Idle)[3] | $V_{DD}$ = Max | 200 | mA |

[1] The test program used to measure $I_{DDINPEAK}$ represents worst case processor operation and is not sustainable under normal application conditions. Actual internal power measurements made using typical applications are less than specified.

[2] $I_{DDINHIGH}$ is a composite average based on a range of high activity code. $I_{DDINLOW}$ is a composite average based on a range of low activity code.

[3] Idle denotes ADSP-2106x state during execution of IDLE instruction.

## INTERNAL POWER DISSIPATION (3.3 V)

These specifications apply to the internal power portion of $V_{DD}$ only. For a complete discussion of the code used to measure power dissipation, see the technical note "SHARC Power Dissipation Measurements."

Specifications are based on the operating scenarios.

| Operation | Peak Activity ($I_{DDINPEAK}$) | High Activity ($I_{DDINHIGH}$) | Low Activity ($I_{DDINLOW}$) |
|---|---|---|---|
| Instruction Type | Multifunction | Multifunction | Single Function |
| Instruction Fetch | Cache | Internal Memory | Internal Memory |
| Core memory Access | 2 Per Cycle (DM and PM) | 1 Per Cycle (DM) | None |
| Internal Memory DMA | 1 Per Cycle | 1 Per 2 Cycles | 1 Per 2 Cycles |

To estimate power consumption for a specific application, use the following equation where % is the amount of time your program spends in that state:

$$\%PEAK\ I_{DDINPEAK} + \%HIGH\ I_{DDINHIGH} + \%LOW\ I_{DDINLOW} + \%IDLE\ I_{DDIDLE} = Power\ Consumption$$

| Parameter | Test Conditions | Max | Unit |
|---|---|---|---|
| $I_{DDINPEAK}$ Supply Current (Internal)[1] | $t_{CK}$ = 30 ns, $V_{DD}$ = Max | 540 | mA |
| | $t_{CK}$ = 25 ns, $V_{DD}$ = Max | 600 | mA |
| $I_{DDINHIGH}$ Supply Current (Internal)[2] | $t_{CK}$ = 30 ns, $V_{DD}$ = Max | 425 | mA |
| | $t_{CK}$ = 25 ns, $V_{DD}$ = Max | 475 | mA |
| $I_{DDINLOW}$ Supply Current (Internal)[2] | $t_{CK}$ = 30 ns, $V_{DD}$ = Max | 250 | mA |
| | $t_{CK}$ = 25 ns, $V_{DD}$ = Max | 275 | mA |
| $I_{DDIDLE}$ Supply Current (Idle)[3] | $V_{DD}$ = Max | 180 | mA |

[1] The test program used to measure $I_{DDINPEAK}$ represents worst case processor operation and is not sustainable under normal application conditions. Actual internal power measurements made using typical applications are less than specified.

[2] $I_{DDINHIGH}$ is a composite average based on a range of high activity code. $I_{DDINLOW}$ is a composite average based on a range of low activity code.

[3] Idle denotes ADSP-2106xL state during execution of IDLE instruction.

## EXTERNAL POWER DISSIPATION (3.3 V)

Total power dissipation has two components, one due to internal circuitry and one due to the switching of external output drivers. Internal power dissipation is dependent on the instruction execution sequence and the data operands involved. Internal power dissipation is calculated in the following way:

$$P_{INT} = I_{DDIN} \times V_{DD}$$

The external component of total power dissipation is caused by the switching of output pins. Its magnitude depends on:

- the number of output pins that switch during each cycle (O)
- the maximum frequency at which they can switch (f)
- their load capacitance (C)
- their voltage swing ($V_{DD}$)

and is calculated by:

$$P_{EXT} = O \times C \times V_{DD}^2 \times f$$

The load capacitance should include the processor's package capacitance (CIN). The switching frequency includes driving the load high and then back low. Address and data pins can drive high and low at a maximum rate of $1/(2t_{CK})$. The write strobe can switch every cycle at a frequency of $1/t_{CK}$. Select pins switch at $1/(2t_{CK})$, but selects can switch on each cycle.

*Example:* Estimate $P_{EXT}$ with the following assumptions:

- A system with one bank of external data memory RAM (32-bit)
- Four 128K × 8 RAM chips are used, each with a load of 10 pF
- External data memory writes occur every other cycle, a rate of $1/(4t_{CK})$, with 50% of the pins switching
- The instruction cycle rate is 40 MHz ($t_{CK}$ = 25 ns)

The $P_{EXT}$ equation is calculated for each class of pins that can drive:

A typical power consumption can now be calculated for these conditions by adding a typical internal power dissipation:

$$P_{TOTAL} = P_{EXT} + (I_{DDIN2} \times 3.3\ V)$$

Note that the conditions causing a worst-case $P_{EXT}$ are different from those causing a worst-case $P_{INT}$. Maximum $P_{INT}$ cannot occur while 100% of the output pins are switching from all ones to all zeros. Note also that it is not common for an application to have 100% or even 50% of the outputs switching simultaneously.

**Table 6. External Power Calculations (3.3 V Devices)**

| Pin Type | No. of Pins | % Switching | × C | × f | × $V_{DD}^2$ | = $P_{EXT}$ |
|---|---|---|---|---|---|---|
| Address | 15 | 50 | × 44.7 pF | × 10 MHz | × 10.9 V | = 0.037 W |
| $\overline{MS0}$ | 1 | 0 | × 44.7 pF | × 10 MHz | × 10.9 V | = 0.000 W |
| $\overline{WR}$ | 1 | – | × 44.7 pF | × 20 MHz | × 10.9 V | = 0.010 W |
| Data | 32 | 50 | × 14.7 pF | × 10 MHz | × 10.9 V | = 0.026 W |
| ADDRCLK | 1 | – | × 4.7 pF | × 20 MHz | × 10.9 V | = 0.001 W |
| | | | | | | **$P_{EXT}$ = 0.074 W** |

## ABSOLUTE MAXIMUM RATINGS

Stresses greater than those listed Table 7 may cause permanent damage to the device. These are stress ratings only; functional operation of the device at these or any other conditions greater than those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Table 7. Absolute Maximum Ratings**

| Parameter | ADSP-21060/ADSP-21060C ADSP-21062 5 V | ADSP-21060L/ADSP-21060LC ADSP-21062L 3.3 V |
|---|---|---|
| Supply Voltage ($V_{DD}$) | –0.3 V to +7.0 V | –0.3 V to +4.6 V |
| Input Voltage | –0.5 V to $V_{DD}$ + 0.5 V | –0.5 V to $V_{DD}$ +0.5 V |
| Output Voltage Swing | –0.5 V to $V_{DD}$ + 0.5 V | –0.5 V to $V_{DD}$ + 0.5 V |
| Load Capacitance | 200 pF | 200 pF |
| Storage Temperature Range | –65°C to +150°C | –65°C to +150°C |
| Lead Temperature (5 seconds) | 280°C | 280°C |
| Junction Temperature Under Bias | 130°C | 130°C |

### *Clock Input*

**Table 9. Clock Input**

| Parameter | | ADSP-21060 ADSP-21062 40 MHz, 5 V Min | ADSP-21060 ADSP-21062 40 MHz, 5 V Max | ADSP-21060 ADSP-21062 33 MHz, 5 V Min | ADSP-21060 ADSP-21062 33 MHz, 5 V Max | ADSP-21060L ADSP-21062L 40 MHz, 3.3 V Min | ADSP-21060L ADSP-21062L 40 MHz, 3.3 V Max | ADSP-21060L ADSP-21062L 33 MHz, 3.3 V Min | ADSP-21060L ADSP-21062L 33 MHz, 3.3 V Max | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | | | | | |
| $t_{CK}$ | CLKIN Period | 25 | 100 | 30 | 100 | 25 | 100 | 30 | 100 | ns |
| $t_{CKL}$ | CLKIN Width Low | 7 | | 7 | | 8.75 | | 8.75[1] | | ns |
| $t_{CKH}$ | CLKIN Width High | 5 | | 5 | | 5 | | 5 | | ns |
| $t_{CKRF}$ | CLKIN Rise/Fall (0.4 V to 2.0 V) | | 3 | | 3 | | 3 | | 3 | ns |

[1] For the ADSP-21060LC, this specification is 9.5 ns min.

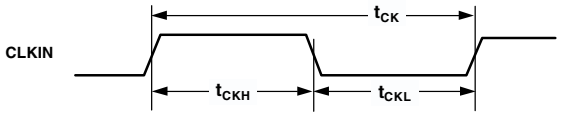


*Figure 9. Clock Input*

### *Reset*

**Table 10. Reset**

| Parameter | | 5 V and 3.3 V Min | 5 V and 3.3 V Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{WRST}$ | $\overline{\text{RESET}}$ Pulse Width Low[1] | $4t_{CK}$ | | ns |
| $t_{SRST}$ | $\overline{\text{RESET}}$ Setup Before CLKIN High[2] | 14 + DT/2 | $t_{CK}$ | ns |

[1] Applies after the power-up sequence is complete. At power-up, the processor's internal phase-locked loop requires no more than 100 µs while $\overline{\text{RESET}}$ is low, assuming stable $V_{DD}$ and CLKIN (not including start-up time of external clock oscillator).

[2] Only required if multiple ADSP-2106xs must come out of reset synchronous to CLKIN with program counters (PC) equal. Not required for multiple ADSP-2106xs communicating over the shared bus (through the external port), because the bus arbitration logic automatically synchronizes itself after reset.

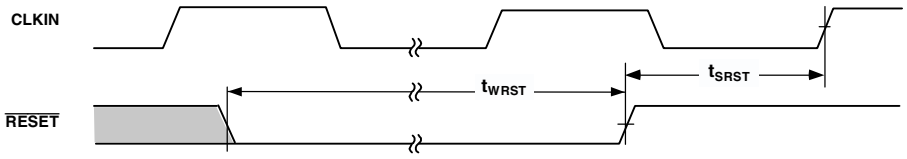


*Figure 10. Reset*

### *Synchronous Read/Write—Bus Slave*

Use these specifications for bus master accesses of a slave's IOP registers or internal memory (in multiprocessor memory space). The bus master must meet the bus slave timing requirements.

**Table 17. Synchronous Read/Write—Bus Slave**

| Parameter | | 5 V and 3.3 V Min | 5 V and 3.3 V Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SADRI}$ | Address, $\overline{SW}$ Setup Before CLKIN | 15 + DT/2 | | ns |
| $t_{HADRI}$ | Address, $\overline{SW}$ Hold After CLKIN | | 5 + DT/2 | ns |
| $t_{SRWLI}$ | $\overline{RD}$/$\overline{WR}$ Low Setup Before CLKIN[1] | 9.5 + 5DT/16 | | ns |
| $t_{HRWLI}$ | $\overline{RD}$/$\overline{WR}$ Low Hold After CLKIN[2] | –4 – 5DT/16 | 8 + 7DT/16 | ns |
| $t_{RWHPI}$ | $\overline{RD}$/$\overline{WR}$ Pulse High | 3 | | ns |
| $t_{SDATWH}$ | Data Setup Before $\overline{WR}$ High | 5 | | ns |
| $t_{HDATWH}$ | Data Hold After $\overline{WR}$ High | 1 | | ns |
| *Switching Characteristics* | | | | |
| $t_{SDDATO}$ | Data Delay After CLKIN[3] | | 18 + 5DT/16 | ns |
| $t_{DATTR}$ | Data Disable After CLKIN[4] | 0 – DT/8 | 7 – DT/8 | ns |
| $t_{DACKAD}$ | ACK Delay After Address, $\overline{SW}$[5] | | 9 | ns |
| $t_{ACKTR}$ | ACK Disable After CLKIN[5] | –1 – DT/8 | 6 – DT/8 | ns |

[1] $t_{SRWLI}$ (min) = 9.5 + 5DT/16 when Multiprocessor Memory Space Wait State (MMSWS bit in WAIT register) is disabled; when MMSWS is enabled, $t_{SRWLI}$ (min)= 4 + DT/8.
[2] For ADSP-21060C specification is –3.5 – 5DT/16 ns min, 8 + 7DT/16 ns max; for ADSP-21060LC specification is –3.75 – 5DT/16 ns min, 8 + 7DT/16 ns max.
[3] For ADSP-21062/ADSP-21062L/ADSP-21060C specification is 19 + 5DT/16 ns max; for ADSP-21060LC specification is 19.25 + 5DT/16 ns max.
[4] See Example System Hold Time Calculation on Page 48 for calculation of hold times given capacitive and dc loads.
[5] $t_{DACKAD}$ is true only if the address and $\overline{SW}$ inputs have setup times (before CLKIN) greater than 10 + DT/8 and less than 19 + 3DT/4. If the address and inputs have setup times greater than 19 + 3DT/4, then ACK is valid 14 + DT/4 (max) after CLKIN. A slave that sees an address with an M field match will respond with ACK regardless of the state of MMSWS or strobes. A slave will three-state ACK every cycle with $t_{ACKTR}$.

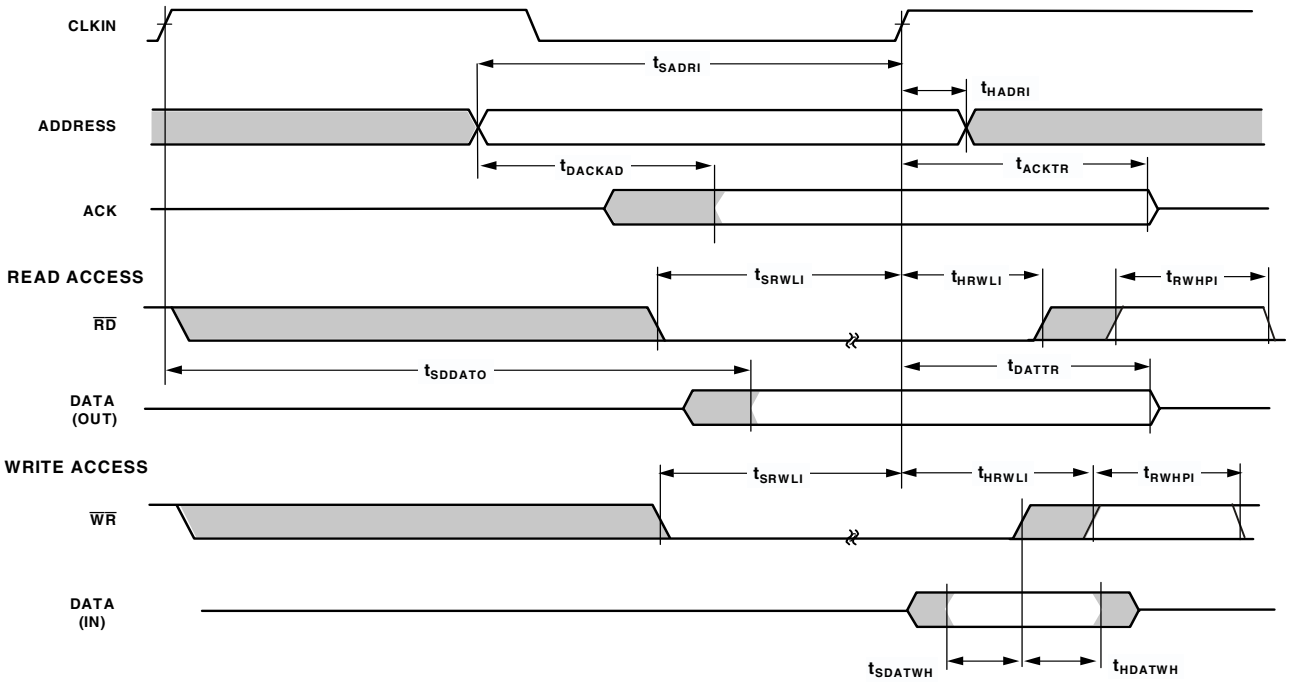


*Figure 17. Synchronous Read/Write—Bus Slave*

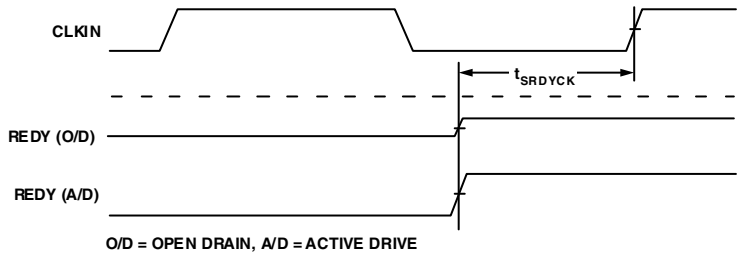


*Figure 19. Synchronous REDY Timing*

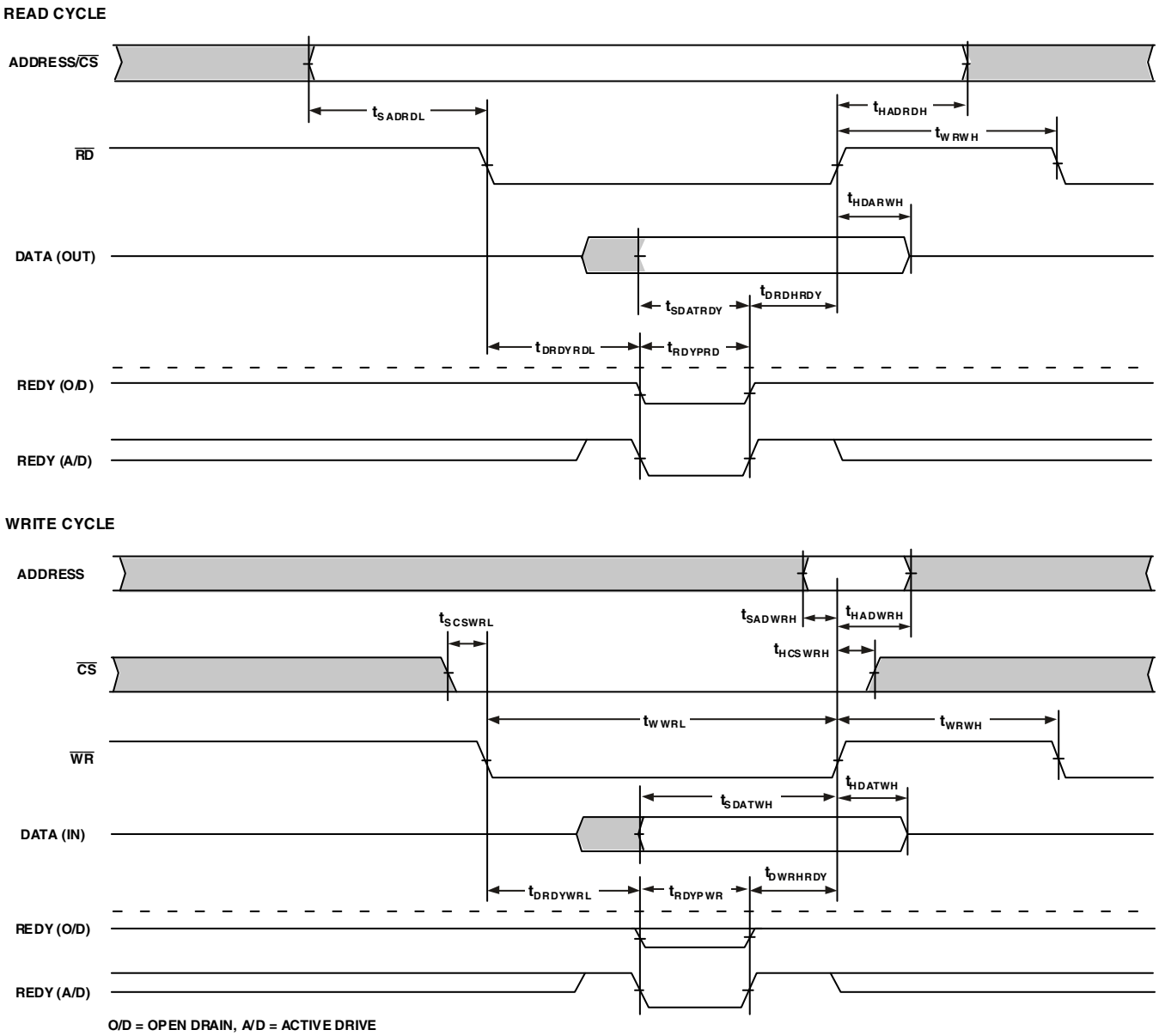


*Figure 20. Asynchronous Read/Write—Host to ADSP-2106x*

### *Link Ports —1 × CLK Speed Operation*

**Table 23. Link Ports—Receive**

| Parameter | | 5 V Min | 5 V Max | 3.3 V Min | 3.3 V Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{SLDCL}$ | Data Setup Before LCLK Low[1] | 3.5 | | 3 | | ns |
| $t_{HLDCL}$ | Data Hold After LCLK Low | 3 | | 3 | | ns |
| $t_{LCLKIW}$ | LCLK Period (1× Operation) | $t_{CK}$ | | $t_{CK}$ | | ns |
| $t_{LCLKRWL}$ | LCLK Width Low | 6 | | 6 | | ns |
| $t_{LCLKRWH}$ | LCLK Width High | 5 | | 5 | | ns |
| *Switching Characteristics* | | | | | | |
| $t_{DLAHC}$ | LACK High Delay After CLKIN High[2, 3] | 18 + DT/2 | 28.5 + DT/2 | 18 + DT/2 | 28.5 + DT/2 | ns |
| $t_{DLALC}$ | LACK Low Delay After LCLK High | –3 | +13 | –3 | +13 | ns |
| $t_{ENDLK}$ | LACK Enable From CLKIN | 5 + DT/2 | | 5 + DT/2 | | ns |
| $t_{TDLK}$ | LACK Disable From CLKIN | | 20 + DT/2 | | 20 + DT/2 | ns |

[1] For ADSP-21062, specification is 3 ns min.
[2] LACK goes low with $t_{DLALC}$ relative to rise of LCLK after first nibble, but does not go low if the receiver's link buffer is not about to fill.
[3] For ADSP-21060C, specification is 18 + DT/2 ns min, 29 + DT/2 ns max.

**Table 24. Link Ports—Transmit**

| Parameter | | 5 V Min | 5 V Max | 3.3 V Min | 3.3 V Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{SLACH}$ | LACK Setup Before LCLK High[1] | 18 | | 18 | | ns |
| $t_{HLACH}$ | LACK Hold After LCLK High | –7 | | –7 | | ns |
| *Switching Characteristics* | | | | | | |
| $t_{DLCLK}$ | Data Delay After CLKIN (1× Operation)[2] | | 15.5 | | 15.5 | ns |
| $t_{DLDCH}$ | Data Delay After LCLK High[3] | | 3 | | 2.5 | ns |
| $t_{HLDCH}$ | Data Hold After LCLK High | –3 | | –3 | | ns |
| $t_{LCLKTWL}$ | LCLK Width Low[4] | ($t_{CK}$/2) – 2 | ($t_{CK}$/2) + 2 | ($t_{CK}$/2) – 1 | ($t_{CK}$/2) + 1.25 | ns |
| $t_{LCLKTWH}$ | LCLK Width High[5] | ($t_{CK}$/2) – 2 | ($t_{CK}$/2) + 2 | ($t_{CK}$/2) – 1.25 | ($t_{CK}$/2) + 1 | ns |
| $t_{DLACLK}$ | LCLK Low Delay After LACK High[6] | ($t_{CK}$/2) + 8.5 | (3 × $t_{CK}$/2) + 17 | ($t_{CK}$/2) + 8 | (3 × $t_{CK}$/2) + 17.5 | ns |
| $t_{ENDLK}$ | LACK Enable From CLKIN | 5 + DT/2 | | 5 + DT/2 | | ns |
| $t_{TDLK}$ | LACK Disable From CLKIN | | 20 + DT/2 | | 20 + DT/2 | ns |

[1] For ADSP-21060L/ADSP-21060LC, specification is 20 ns min.
[2] For ADSP-21060L, specification is 16.5 ns max; for ADSP-21060LC, specification is 16.75 ns max.
[3] For ADSP-21062, specification is 2.5 ns max.
[4] For ADSP-21062, specification is ($t_{CK}$/2) – 1 ns min, ($t_{CK}$/2) + 1.25 ns max; for ADSP-21062L, specification is ($t_{CK}$/2) – 1 ns min, ($t_{CK}$/2) + 1.5 ns max; for ADSP-21060LC specification is ($t_{CK}$/2) – 1 ns min, ($t_{CK}$/2) + 2.25 ns max.
[5] For ADSP-21062, specification is ($t_{CK}$/2) – 1.25 ns min, ($t_{CK}$/2) + 1 ns max; for ADSP-21062L, specification is ($t_{CK}$/2) – 1.5 ns min, ($t_{CK}$/2) + 1 ns max; for ADSP-21060C specification is ($t_{CK}$/2) – 2.25 ns min, ($t_{CK}$/2) + 1 ns max.
[6] For ADSP-21062, specification is ($t_{CK}$/2) + 8.75 ns min, (3 × $t_{CK}$/2) + 17 ns max; for ADSP-21062L, specification is ($t_{CK}$/2) + 8 ns min, (3 × $t_{CK}$/2) + 17 ns max; for ADSP-21060LC specification is ($t_{CK}$/2) + 8 ns min, (3 × $t_{CK}$/2) + 18.5 ns max.

**Table 25. Link Port Service Request Interrupts: 1× and 2× Speed Operations**

| Parameter | | 5 V Min | 5 V Max | 3.3 V Min | 3.3 V Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{SLCK}$ | LACK/LCLK Setup Before CLKIN Low[1] | 10 | | 10 | | ns |
| $t_{HLCK}$ | LACK/LCLK Hold After CLKIN Low[1] | 2 | | 2 | | ns |

[1] Only required for interrupt recognition in the current cycle.

### *Link Ports —2 × CLK Speed Operation*

Calculation of link receiver data setup and hold relative to link clock is required to determine the maximum allowable skew that can be introduced in the transmission path between LDATA and LCLK. Setup skew is the maximum delay that can be introduced in LDATA relative to LCLK:

$$Setup\ Skew = t_{LCLKTWH}\ \text{min} - t_{DLDCH} - t_{SLDCL}$$

Hold skew is the maximum delay that can be introduced in LCLK relative to LDATA:

$$Hold\ Skew = t_{LCLKTWL}\ \text{min} - t_{HLDCH} - t_{HLDCL}$$

Calculations made directly from 2 speed specifications will result in unrealistically small skew times because they include multiple tester guardbands.

Note that link port transfers at 2× CLK speed at 40 MHz ($t_{CK}$ = 25 ns) may fail. However, 2× CLK speed link port transfers at 33 MHz ($t_{CK}$ = 30 ns) work as specified.

**Table 26. Link Ports—Receive**

| Parameter | | 5 V Min | 5 V Max | 3.3 V Min | 3.3 V Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{SLDCL}$ | Data Setup Before LCLK Low | 2.5 | | 2.25 | | ns |
| $t_{HLDCL}$ | Data Hold After LCLK Low | 2.25 | | 2.25 | | ns |
| $t_{LCLKIW}$ | LCLK Period (2× Operation) | $t_{CK}/2$ | | $t_{CK}/2$ | | ns |
| $t_{LCLKRWL}$ | LCLK Width Low[1] | 4.5 | | 5.25 | | ns |
| $t_{LCLKRWH}$ | LCLK Width High[2] | 4.25 | | 4 | | ns |
| *Switching Characteristics* | | | | | | |
| $t_{DLAHC}$ | LACK High Delay After CLKIN High[3] | 18 + DT/2 | 28.5 + DT/2 | 18 + DT/2 | 29.5 + DT/2 | ns |
| $t_{DLALC}$ | LACK Low Delay After LCLK High[4] | 6 | 16 | 6 | 16 | ns |

[1] For ADSP-21060L, specification is 5 ns min.
[2] For ADSP-21062, specification is 4 ns min, for ADSP-21060LC, specification is 4.5 ns min.
[3] LACK goes low with $t_{DLALC}$ relative to rise of LCLK after first nibble, but does not go low if the receiver's link buffer is not about to fill.
[4] For ADSP-21060L, specification is 6 ns min, 18 ns max. For ADSP-21060C, specification is 6 ns min, 16.5 ns max. For ADSP-21060LC, specification is 6 ns min, 18.5 ns max.

# 225-BALL PBGA BALL CONFIGURATION

**Table 40. ADSP-2106x 225-Ball Metric PBGA Ball Assignments (B-225-2)**

| Ball Name | Ball Number | Ball Name | Ball Number | Ball Name | Ball Number | Ball Name | Ball Number | Ball Name | Ball Number |
|---|---|---|---|---|---|---|---|---|---|
| $\overline{BMS}$ | A01 | ADDR25 | D01 | ADDR14 | G01 | ADDR6 | K01 | $\overline{EMU}$ | N01 |
| ADDR30 | A02 | ADDR26 | D02 | ADDR15 | G02 | ADDR5 | K02 | TDO | N02 |
| $\overline{DMAR2}$ | A03 | $\overline{MS2}$ | D03 | ADDR16 | G03 | ADDR3 | K03 | $\overline{IRQ0}$ | N03 |
| DT1 | A04 | ADDR29 | D04 | ADDR19 | G04 | ADDR0 | K04 | $\overline{IRQ1}$ | N04 |
| RCLK1 | A05 | $\overline{DMAR1}$ | D05 | GND | G05 | ICSA | K05 | ID2 | N05 |
| TCLK0 | A06 | TFS1 | D06 | $V_{DD}$ | G06 | GND | K06 | L5DAT1 | N06 |
| RCLK0 | A07 | $\overline{CPA}$ | D07 | $V_{DD}$ | G07 | $V_{DD}$ | K07 | L4CLK | N07 |
| ADRCLK | A08 | $\overline{HBG}$ | D08 | $V_{DD}$ | G08 | $V_{DD}$ | K08 | L3CLK | N08 |
| $\overline{CS}$ | A09 | $\overline{DMAG2}$ | D09 | $V_{DD}$ | G09 | $V_{DD}$ | K09 | L3DAT3 | N09 |
| CLKIN | A10 | $\overline{BR5}$ | D10 | $V_{DD}$ | G10 | GND | K10 | L2DAT0 | N10 |
| PAGE | A11 | $\overline{BR1}$ | D11 | GND | G11 | GND | K11 | L1ACK | N11 |
| $\overline{BR3}$ | A12 | DATA40 | D12 | DATA22 | G12 | DATA8 | K12 | L1DAT3 | N12 |
| DATA47 | A13 | DATA37 | D13 | DATA25 | G13 | DATA11 | K13 | L0DAT3 | N13 |
| DATA44 | A14 | DATA35 | D14 | DATA24 | G14 | DATA13 | K14 | DATA1 | N14 |
| DATA42 | A15 | DATA34 | D15 | DATA23 | G15 | DATA14 | K15 | DATA3 | N15 |
| $\overline{MS0}$ | B01 | ADDR21 | E01 | ADDR12 | H01 | ADDR2 | L01 | $\overline{TRST}$ | P01 |
| $\overline{SW}$ | B02 | ADDR22 | E02 | ADDR11 | H02 | ADDR1 | L02 | TMS | P02 |
| ADDR31 | B03 | ADDR24 | E03 | ADDR13 | H03 | FLAG0 | L03 | EBOOT | P03 |
| $\overline{HBR}$ | B04 | ADDR27 | E04 | ADDR10 | H04 | FLAG3 | L04 | ID0 | P04 |
| DR1 | B05 | GND | E05 | GND | H05 | RPBA | L05 | L5CLK | P05 |
| DT0 | B06 | GND | E06 | $V_{DD}$ | H06 | GND | L06 | L5DAT3 | P06 |
| DR0 | B07 | GND | E07 | $V_{DD}$ | H07 | GND | L07 | L4DAT0 | P07 |
| REDY | B08 | GND | E08 | $V_{DD}$ | H08 | GND | L08 | L4DAT3 | P08 |
| $\overline{RD}$ | B09 | GND | E09 | $V_{DD}$ | H09 | GND | L09 | L3DAT2 | P09 |
| ACK | B10 | GND | E10 | $V_{DD}$ | H10 | GND | L10 | L2CLK | P10 |
| $\overline{BR6}$ | B11 | NC | E11 | GND | H11 | NC | L11 | L2DAT2 | P11 |
| $\overline{BR2}$ | B12 | DATA33 | E12 | DATA18 | H12 | DATA4 | L12 | L1DAT0 | P12 |
| DATA45 | B13 | DATA30 | E13 | DATA19 | H13 | DATA7 | L13 | L0ACK | P13 |
| DATA43 | B14 | DATA32 | E14 | DATA21 | H14 | DATA9 | L14 | L0DAT1 | P14 |
| DATA39 | B15 | DATA31 | E15 | DATA20 | H15 | DATA10 | L15 | DATA0 | P15 |
| $\overline{MS3}$ | C01 | ADDR17 | F01 | ADDR9 | J01 | FLAG1 | M01 | TCK | R01 |
| $\overline{MS1}$ | C02 | ADDR18 | F02 | ADDR8 | J02 | FLAG2 | M02 | $\overline{IRQ2}$ | R02 |
| ADDR28 | C03 | ADDR20 | F03 | ADDR7 | J03 | TIMEXP | M03 | $\overline{RESET}$ | R03 |
| $\overline{SBTS}$ | C04 | ADDR23 | F04 | ADDR4 | J04 | TDI | M04 | ID1 | R04 |
| TCLK1 | C05 | GND | F05 | GND | J05 | LBOOT | M05 | L5DAT0 | R05 |
| RFS1 | C06 | GND | F06 | $V_{DD}$ | J06 | L5ACK | M06 | L4ACK | R06 |
| TFS0 | C07 | $V_{DD}$ | F07 | $V_{DD}$ | J07 | L5DAT2 | M07 | L4DAT1 | R07 |
| RFS0 | C08 | $V_{DD}$ | F08 | $V_{DD}$ | J08 | L4DAT2 | M08 | L3ACK | R08 |
| $\overline{WR}$ | C09 | $V_{DD}$ | F09 | $V_{DD}$ | J09 | L3DAT0 | M09 | L3DAT1 | R09 |
| $\overline{DMAG1}$ | C10 | GND | F10 | $V_{DD}$ | J10 | L2DAT3 | M10 | L2ACK | R10 |
| $\overline{BR4}$ | C11 | GND | F11 | GND | J11 | L1DAT1 | M11 | L2DAT1 | R11 |
| DATA46 | C12 | DATA29 | F12 | DATA12 | J12 | L0DAT0 | M12 | L1CLK | R12 |
| DATA41 | C13 | DATA26 | F13 | DATA15 | J13 | DATA2 | M13 | L1DAT2 | R13 |
| DATA38 | C14 | DATA28 | F14 | DATA16 | J14 | DATA5 | M14 | L0CLK | R14 |
| DATA36 | C15 | DATA27 | F15 | DATA17 | J15 | DATA6 | M15 | L0DAT2 | R15 |

**Table 42. ADSP-21060CW/21060LCW CQFP Pin Assignments (QS-240-1A, QS-240-1B)**

| Pin Name | Pin No. | Pin Name | Pin No. | Pin Name | Pin No. | Pin Name | Pin No. | Pin Name | Pin No. | Pin Name | Pin No. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| GND | 1 | DATA29 | 41 | $\overline{\text{DMAG2}}$ | 81 | ADDR28 | 121 | ADDR5 | 161 | GND | 201 |
| DATA0 | 2 | GND | 42 | ACK | 82 | $\overline{\text{BMS}}$ | 122 | GND | 162 | $V_{DD}$ | 202 |
| DATA1 | 3 | DATA30 | 43 | CLKIN | 83 | $\overline{\text{SW}}$ | 123 | ADDR4 | 163 | L4ACK | 203 |
| DATA2 | 4 | DATA31 | 44 | GND | 84 | $\overline{\text{MS0}}$ | 124 | ADDR3 | 164 | L4CLK | 204 |
| $V_{DD}$ | 5 | DATA32 | 45 | $V_{DD}$ | 85 | $\overline{\text{MS1}}$ | 125 | ADDR2 | 165 | L4DAT0 | 205 |
| DATA3 | 6 | GND | 46 | GND | 86 | $\overline{\text{MS2}}$ | 126 | $V_{DD}$ | 166 | L4DAT1 | 206 |
| DATA4 | 7 | $V_{DD}$ | 47 | $\overline{\text{WR}}$ | 87 | $\overline{\text{MS3}}$ | 127 | ADDR1 | 167 | L4DAT2 | 207 |
| DATA5 | 8 | $V_{DD}$ | 48 | $\overline{\text{RD}}$ | 88 | GND | 128 | ADDR0 | 168 | L4DAT3 | 208 |
| GND | 9 | DATA33 | 49 | $\overline{\text{CS}}$ | 89 | ADDR27 | 129 | GND | 169 | GND | 209 |
| DATA6 | 10 | DATA34 | 50 | $\overline{\text{HBG}}$ | 90 | ADDR26 | 130 | FLAG0 | 170 | L3ACK | 210 |
| DATA7 | 11 | DATA35 | 51 | REDY | 91 | ADDR25 | 131 | FLAG1 | 171 | L3CLK | 211 |
| DATA8 | 12 | NC | 52 | ADRCLK | 92 | $V_{DD}$ | 132 | FLAG2 | 172 | L3DAT0 | 212 |
| $V_{DD}$ | 13 | GND | 53 | GND | 93 | GND | 133 | FLAG3 | 173 | L3DAT1 | 213 |
| DATA9 | 14 | DATA36 | 54 | $V_{DD}$ | 94 | $V_{DD}$ | 134 | ICSA | 174 | L3DAT2 | 214 |
| DATA10 | 15 | DATA37 | 55 | $V_{DD}$ | 95 | ADDR24 | 135 | $\overline{\text{EMU}}$ | 175 | L3DAT3 | 215 |
| DATA11 | 16 | DATA38 | 56 | RFS0 | 96 | ADDR23 | 136 | TIMEXP | 176 | $V_{DD}$ | 216 |
| GND | 17 | $V_{DD}$ | 57 | RCLK0 | 97 | ADDR22 | 137 | TDO | 177 | NC | 217 |
| DATA12 | 18 | DATA39 | 58 | DR0 | 98 | GND | 138 | $V_{DD}$ | 178 | L2ACK | 218 |
| DATA13 | 19 | DATA40 | 59 | TFS0 | 99 | ADDR21 | 139 | $\overline{\text{TRST}}$ | 179 | L2CLK | 219 |
| DATA14 | 20 | DATA41 | 60 | TCLK0 | 100 | ADDR20 | 140 | TDI | 180 | L2DAT0 | 220 |
| $V_{DD}$ | 21 | GND | 61 | DT0 | 101 | ADDR19 | 141 | TMS | 181 | L2DAT1 | 221 |
| DATA15 | 22 | DATA42 | 62 | $\overline{\text{CPA}}$ | 102 | $V_{DD}$ | 142 | TCK | 182 | L2DAT2 | 222 |
| DATA16 | 23 | DATA43 | 63 | GND | 103 | $V_{DD}$ | 143 | $\overline{\text{IRQ0}}$ | 183 | L2DAT3 | 223 |
| DATA17 | 24 | DATA44 | 64 | RFS1 | 104 | ADDR18 | 144 | $\overline{\text{IRQ1}}$ | 184 | $V_{DD}$ | 224 |
| GND | 25 | $V_{DD}$ | 65 | RCLK1 | 105 | ADDR17 | 145 | $\overline{\text{IRQ2}}$ | 185 | GND | 225 |
| DATA18 | 26 | DATA45 | 66 | DR1 | 106 | ADDR16 | 146 | EBOOT | 186 | GND | 226 |
| DATA19 | 27 | DATA46 | 67 | TFS1 | 107 | GND | 147 | $\overline{\text{RESET}}$ | 187 | L1ACK | 227 |
| DATA20 | 28 | DATA47 | 68 | TCLK1 | 108 | ADDR15 | 148 | RPBA | 188 | L1CLK | 228 |
| $V_{DD}$ | 29 | GND | 69 | DT1 | 109 | ADDR14 | 149 | LBOOT | 189 | L1DAT0 | 229 |
| DATA21 | 30 | $V_{DD}$ | 70 | $\overline{\text{HBR}}$ | 110 | $V_{DD}$ | 150 | ID0 | 190 | L1DAT1 | 230 |
| DATA22 | 31 | GND | 71 | $\overline{\text{DMAR1}}$ | 111 | ADDR13 | 151 | ID1 | 191 | L1DAT2 | 231 |
| DATA23 | 32 | $\overline{\text{BR1}}$ | 72 | $\overline{\text{DMAR2}}$ | 112 | ADDR12 | 152 | ID2 | 192 | L1DAT3 | 232 |
| GND | 33 | $\overline{\text{BR2}}$ | 73 | $\overline{\text{SBTS}}$ | 113 | ADDR11 | 153 | GND | 193 | $V_{DD}$ | 233 |
| DATA24 | 34 | $\overline{\text{BR3}}$ | 74 | GND | 114 | GND | 154 | L5ACK | 194 | L0ACK | 234 |
| DATA25 | 35 | $\overline{\text{BR4}}$ | 75 | ADDR31 | 115 | ADDR10 | 155 | L5CLK | 195 | L0CLK | 235 |
| DATA26 | 36 | $\overline{\text{BR5}}$ | 76 | ADDR30 | 116 | ADDR9 | 156 | L5DAT0 | 196 | L0DAT0 | 236 |
| $V_{DD}$ | 37 | $\overline{\text{BR6}}$ | 77 | ADDR29 | 117 | ADDR8 | 157 | L5DAT1 | 197 | L0DAT1 | 237 |
| $V_{DD}$ | 38 | $V_{DD}$ | 78 | $V_{DD}$ | 118 | $V_{DD}$ | 158 | L5DAT2 | 198 | L0DAT2 | 238 |
| DATA27 | 39 | PAGE | 79 | $V_{DD}$ | 119 | ADDR7 | 159 | L5DAT3 | 199 | L0DAT3 | 239 |
| DATA28 | 40 | $\overline{\text{DMAG1}}$ | 80 | GND | 120 | ADDR6 | 160 | $V_{DD}$ | 200 | GND | 240 |

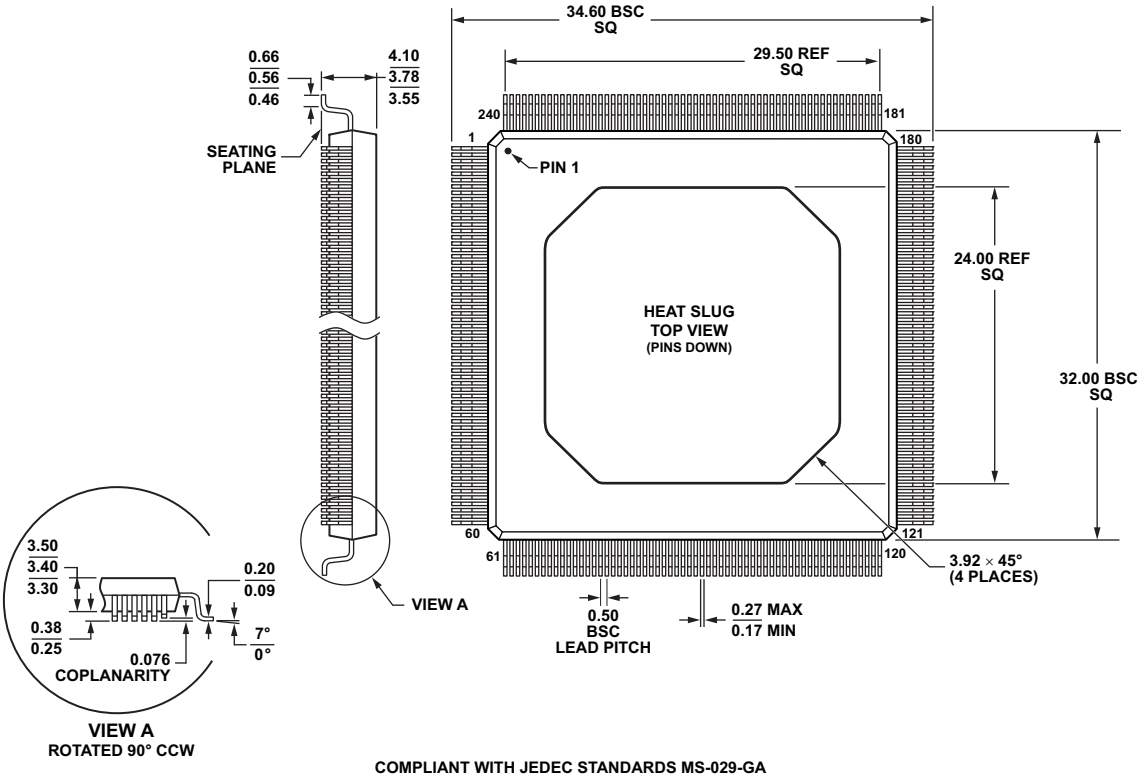


*Figure 41. 240-Lead Metric Quad Flat Package, Thermally Enhanced "PowerQuad" [MQFP_PQ4]*
*(SP-240-2)*
*Dimensions shown in millimeters*

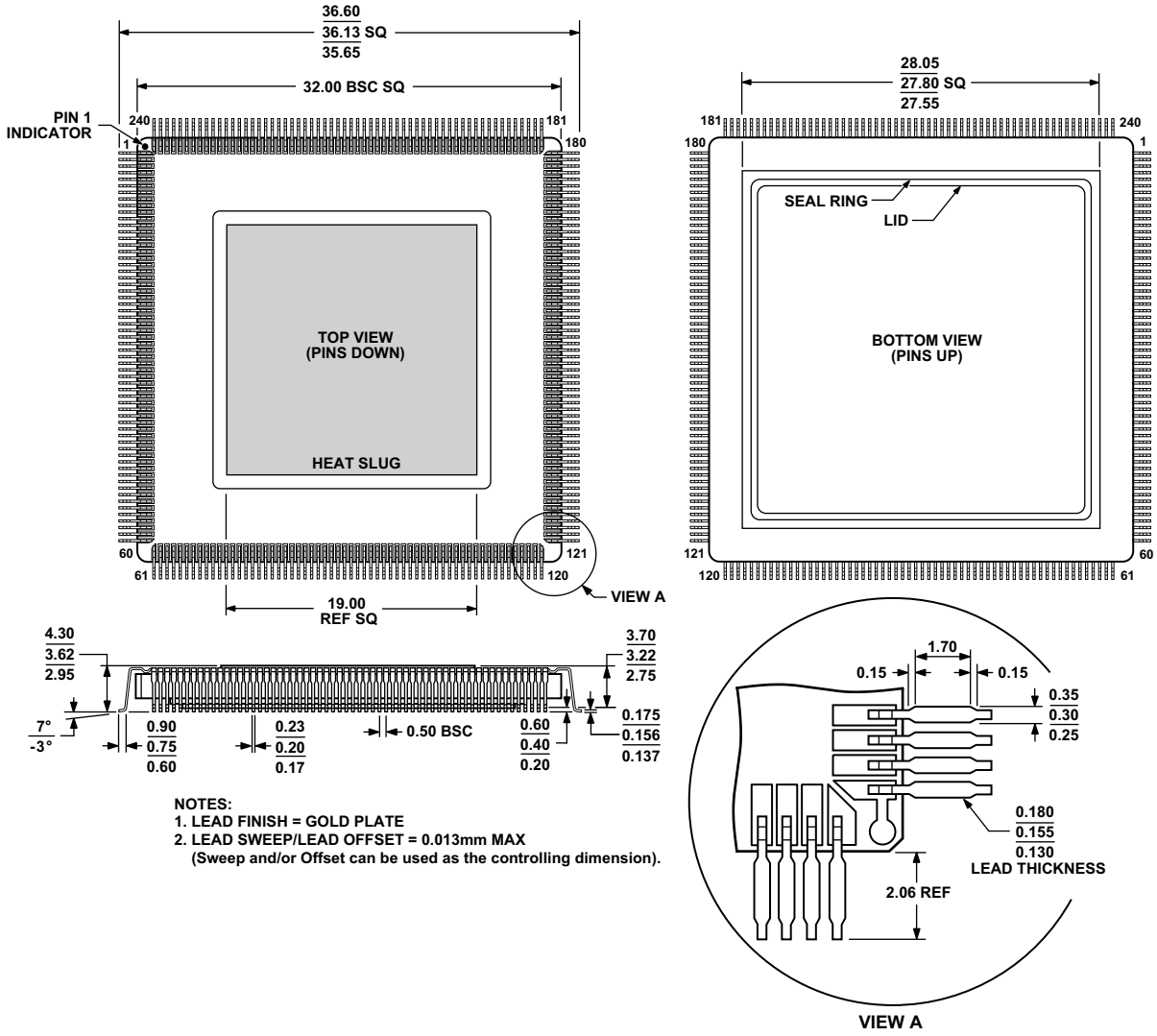


*Figure 42. 240-Lead Ceramic Quad Flat Package, Heat Slug Up [CQFP]*
*(QS-240-2A)*
*Dimensions shown in millimeters*

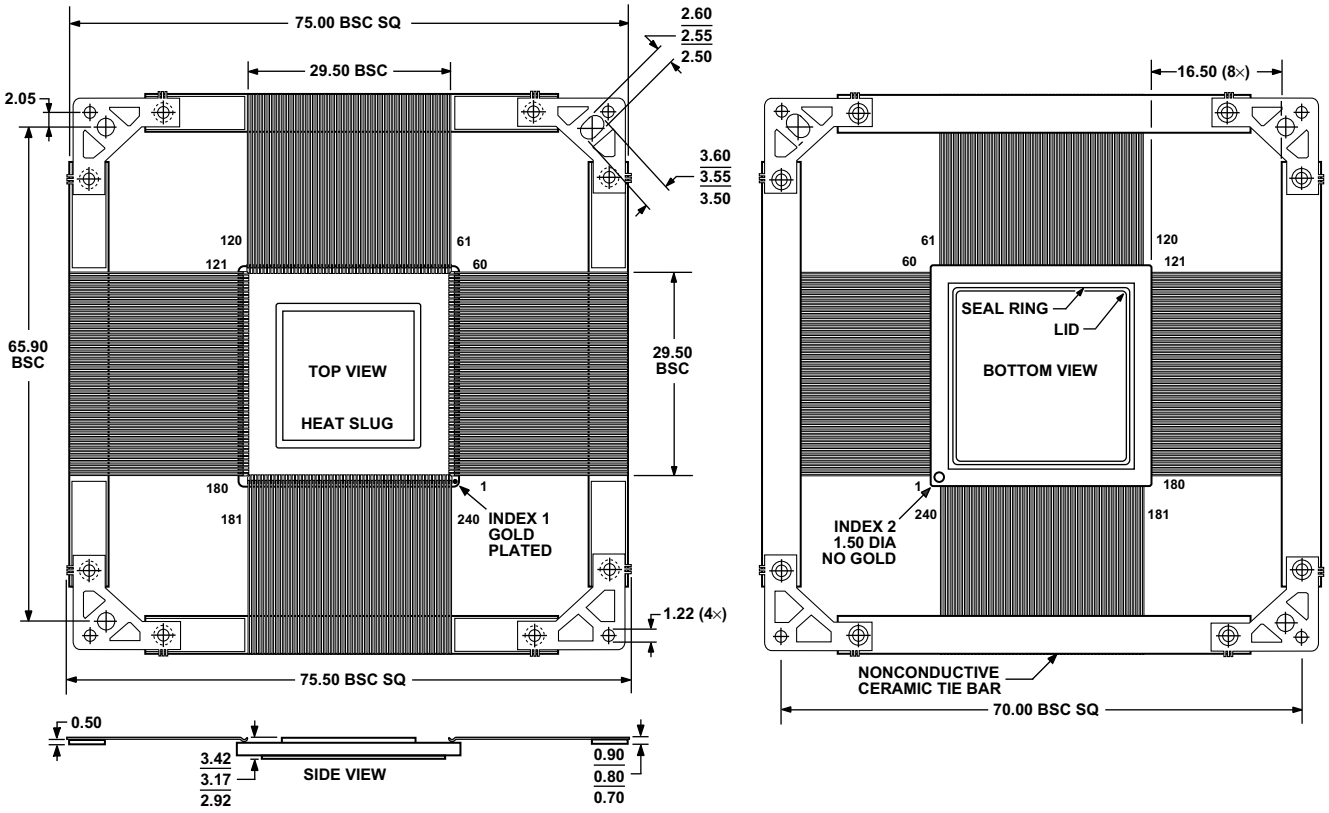


*Figure 43. 240-Lead Ceramic Quad Flat Package, Mounted with Cavity Down [CQFP]*
*(QS-240-2B)*
*Dimensions shown in millimeters*

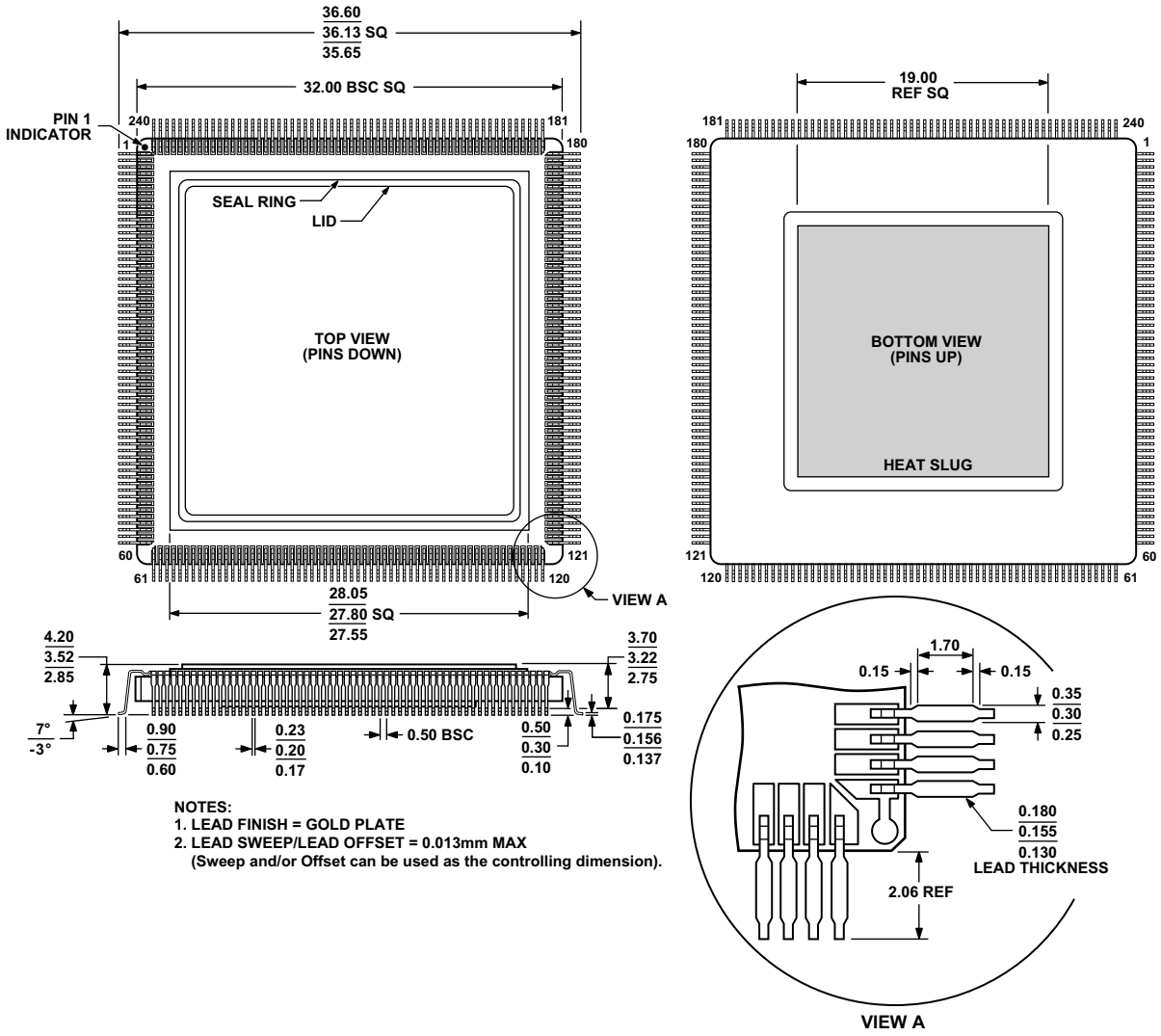


*Figure 44. 240-Lead Ceramic Quad Flat Package, Heat Slug Down [CQFP]*
*(QS-240-1A)*
*Dimensions shown in millimeters*

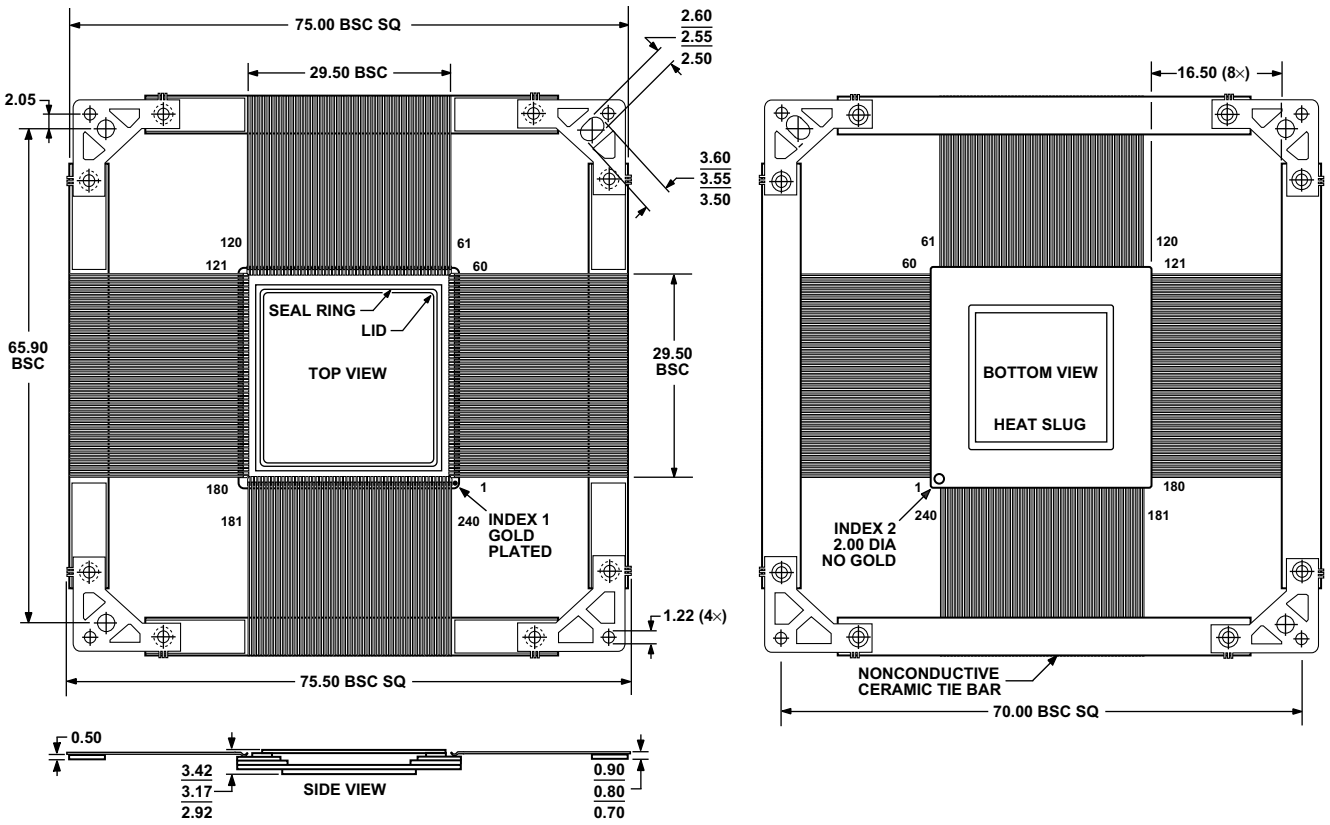


*Figure 45. 240-Lead Ceramic Quad Flat Package, Mounted with Cavity Up [CQFP]*
*(QS-240-1B)*
*Dimensions shown in millimeters*

## SURFACE-MOUNT DESIGN

Table 43 is provided as an aide to PCB design. For industry-standard design recommendations, refer to IPC-7351, *Generic Requirements for Surface-Mount Design and Land Pattern Standard.*

**Table 43. BGA Data for Use with Surface-Mount Design**

| Package | Ball Attach Type | Solder Mask Opening | Ball Pad Size |
|---|---|---|---|
| 225-Ball Grid Array (PBGA) | Solder Mask Defined | 0.63 mm diameter | 0.76 mm diameter |