VIA C3™ in EBGA

Datasheet

July 25, 2003

VIA TECHNOLOGIES, INC.

LIFE SUPPORT POLICY

VIA processor products are not authorized for use as components in life support or other medical devices or systems (hereinafter life support devices) unless a specific written agreement pertaining to such intended use is executed between the manufacturer and an officer of VIA.

1. **Life support devices are devices which (a) are intended for surgical implant into the body or (b) support or sustain life and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.**
2. **This policy covers any component of a life support device or system whose failure to perform can cause the failure of the life support device or system, or to affect its safety or effectiveness.**

Table of Contents

List of Figures

List of Tables

SECTION

1

INTRODUCTION

The VIA C3™ processor in Enhanced Ball Grid Array (EBGA) packaging is based upon a unique internal architecture and is manufactured using advanced 0.15µ or 0.13µ CMOS technology. The C3 architecture and process technology provide a highly compatible, high-performance, low-cost, and low-power solution for the desktop PC, notebook, and Internet Appliance markets. The VIA C3 processor in EBGA is available in several MHz versions.

When considered individually, the compatibility, function, performance, cost, and power dissipation of the VIA C3 processor family are all very competitive. Furthermore, the value added from the advanced EBGA packaging includes remarkable compactness, cost efficiency and excellent thermal characteristics. The VIA C3 package in EBGA represents a breakthrough combination for enabling high-value, high-performance, low-power, small form factor x86-based solutions. When considered as a whole, the VIA C3 processor family in EBGA offers a peerless level of *value.*

1.1 DATASHEET OUTLINE

The intent of this datasheet is to make it easy for a direct user—a board designer, a system designer, or a BIOS developer—to use the VIA C3 processor in EBGA packaging.

Section 1 of the datasheet summarizes the key features of the VIA C3 processor in EBGA. **Section 2** provides a detailed description of the internal architecture of the VIA C3 In EBGA processor. **Section 3** specifies the primary programming interface. **Section 4** does the same for the bus interface. **Sections 5, 6, and 7** specify the classical datasheet topics of AC timings, ballouts, and mechanical specifications.

Appendix A documents the VIA C3 In EBGA processor machine specific registers (MSRs).

1.2 BASIC FEATURES

The VIA C3 processor family in EBGA currently consists of two basic models with several different MHz versions. Due to their low power dissipation, either model is ideally suited for both desktop and mobile applications. These chips can also be divided into two categories: "S" versions are clock multiplier selectable; EBGA VIA C3 processors that lack the "S" designation are multiplier fixed. All versions share the following common features:

- A proprietary Enhanced Ball Grid Array (EBGA) package that shares with Socket 370 processors features such as bus protocol and electrical interface
- Seamlessly software compatible with the thousands of available x86 software applications
- MMX-compatible instructions for enhanced media performance
- AMD-compatible 3DNow! Instructions for turbocharging games, photo processing and media applications
- Two large (64-KB each, 4-way) on-chip Level 1 caches
- 64-KB Level 2 victim cache
- Two large TLBs (128 entries each, 8-way) with two page directory caches
- Unique and sophisticated branch prediction mechanisms
- Bus speeds up to 133 MHz
- Extremely low power dissipation
- Very small die (52 mm^2 in TSMC 0.13µ technology)
- Compact and economical EBGA packaging with excellent thermal dissipation characteristics

1.3 PROCESSOR VERSIONS

Typically, there are five specification parameters that characterize different versions of a processor family: package, voltage, maximum case temperature, external bus speed, and internal MHz.

The VIA C3 processor in EBGA is differentiated from the rest of the C3 family by its low-profile Enhance Ball Grid Array packaging. The EBGA package is not only compact and economical, but delivers excellent thermal properties for truly Cool Processing.

The internal MHz of a particular VIA C3 in EBGA processor chip is defined by two parameters: the specified external bus speed and the internal bus-clock multiplier. EBGA VIA C3 chips come in two bus-speed versions: either 100MHz bus or 133 MHz bus. (Either version can also operate at 66MHz bus speeds.) Bus frequency select balls (BSEL 0 and BSEL 1) identify the appropriate bus speed (100 MHz or 133 MHz).

The bus-clock multiplier is hardwired into each VIA C3 in EBGA chip that do not feature the "S" designation. For these chips, the ratio of the internal processor clock speed to the externally supplied bus clock is frozen[1]. Several different clock-multiplier versions are currently offered.

EBGA VIA C3 processors that carry the "S" designation do not have the bus multiplier locked. In these chips, the clock ratio is set using the external bus ratio balls, *BR[4:0]*.

More information on these topics is included in Sections 4, 5, and 6 of this datasheet

- The VIA C3 in EBGA processor is available in a variety of speed grades and voltages:
  - 667 MHz (5.0 x 133-MHz bus)
  - 733 MHz (5.5 x 133-MHz bus)
  - 800 MHz (6.0 x 133-MHz bus)
  - 933 MHz (7.0 x 133-MHz bus)
  - 1.00 GHz (7.5 x 133-MHz bus)
- Future versions of the VIA C3 in EBGA processor may provide other speed grades and bus speed combinations.
- Future versions of the VIA C3 in EBGA processor may have different voltages.

[1] Actually, it is semi-frozen. A VIA-unique machine specific register allows programming to temporarily change the hard-wired multiplier. This capability, documented in Appendix A, is intended for special BIOS situations and test and debug usage.

1.4 COMPETITIVE COMPARISONS

Section 2 of this datasheet contains considerable detail about the unique internal design of the VIA C3 processor in EBGA. *If not indicated otherwise, specific information relates to the EBGA VIA C3 Ezra.*

Such an internal architecture comparison, however, does not directly address the most important considerations of users: compatibility, availability, price, MHz, application performance, and power dissipation. For that reason, we do not provide a detailed architecture comparison with other processors in this datasheet. Following instead is the high-level summary of an internal-architecture comparison:

- The VIA C3 in EBGA processor has better cache and TLB capabilities. These are critical to system performance for modern PC operating systems and applications.
- The VIA C3 in EBGA has a generally simpler internal architecture. However, the VIA C3 implements many advanced features critical to performance (such as sophisticated branch prediction) and is highly tuned for good application performance, especially on “integer” or office applications.
- The VIA C3 in EBGA has a much smaller die size (52 mm^2).
- The VIA C3 in EBGA has much lower power dissipation at the same MHz than its competitors.
- No competitive processor matches the VIA C3 in terms of combined low-power, modest thermal demands, performance, flexible packaging and attractive pricing.

1.5 COMPATIBILITY

The VIA C3 in EBGA packaging shares much with it conventionally packaged siblings. “Standard” VIA C3’s can plug into existing Socket 370 motherboards and can operate without requiring changes to the system hardware (with one theoretical difference discussed in the next paragraph). No special jumpers or different board wiring is required. In most cases, however, a special BIOS is needed. Currently, BIOS support for the VIA C3 is available from Award, AMI, Phoenix, and Insyde.

The VIA C3 requires external termination of bus signals. Physical and bus compatibility is covered in more detail in Section 4 of this datasheet.

Obviously the VIA C3 in EBGA packaging will not fit into a Socket 370 motherboard since the CPU is mounted directly to the motherboard. By eliminating the need for a CPU socket, the manufacturing costs are reduced while elevating the intrinsic value of the motherboard. The excellent thermal dissipation characteristics of the EBGA package combined with the C3’s class leading low power requirements facilitate the use of small and cheap CPU cooling solutions – even fanless designs are possible.

The VIA C3 supports 3DNow! instructions. The VIA C3 does NOT support multiple processors. These functions are defined as optional by, and are identified to software via, the CPUID instruction. The VIA C3 in EBGA carefully follows the protocol for defining the availability of these optional features. Both the additional and omitted optional features are covered in more detail in Section 3 of this datasheet.

To verify compatibility of the VIA C3 in EBGA with real PC applications and hardware, VIA has performed extensive testing of hundreds of PC boards and peripherals, thousands of software applications, and all known operating systems.

Indicative of this high compatibility, the VIA C3 *has obtained both Windows 98 and Windows NT certification.*

This page is intentionally left blank.

SECTION

2

ARCHITECTURE

2.1 INTRODUCTION

The VIA C3 in EBGA architecture can be either that of the VIA C3 Samuel 2 or the VIA C3 Ezra. The C3, formerly known as the VIA Cyrix III, is different from any other x86 processor architecture. This unique processor design provides a significantly smaller die size using less power than any other x86 CPU. The VIA C3 EBGA features cores that improve performance (MHz and CPI) and further reduces die size and power over the base VIA C3 Samuel. (The major differences between the VIA C3 Samuel processor and the VIA C3 Ezra are highlighted in the descriptions below.) For Nehemiah architecture, please refer to the C3 Nehemiah datasheet.

The VIA C3 architecture is based on, and directly exploits, basic "facts" about the current x86 market, applications, and bus environment. While seemingly straightforward, these concepts are not exploited in other processor architectures. The major concepts that shape the VIA C3 architecture are:

- **Only a few instructions dominate x86 instruction execution time**. On typical applications, over 90% of instruction execution time is due to a handful of basic x86 instructions. Most x86 instructions have no significant impact on performance.

  The VIA C3 design optimizes performance of the most important x86 instructions while minimizing the hardware provided for the little-used x86 functions (infrequently used x86 instructions are primarily implemented in microcode). The resulting instruction execution speed of highly used instructions is as good or better than comparable processors. For example, **the VIA C3 executes x86 load-ALU-store instructions in only one clock** as compared to several clocks on other processors. The execution time of little-used instructions is impacted to reduce die size, but this has no effect on real application performance.

- **Improving clock frequency has higher leverage than improving CPI**. The result of advanced computer design approaches over the last few years has been that the improvements in cycles-per-instruction (CPI) often impact MHz improvements, and certainly impact die size. Our belief is that the best way to improve total performance and keep a small low-power die is to improve MHz.

  Thus, the VIA C3 family architecture provides improved performance by optimizing MHz via a 12-stage pipeline while maintaining a good CPI. Complex CPI-driven features such as out-of-order instruction execution are not implemented because they (1) impact MHz, (2) require a lot of die area and power, and (3) have little impact on real performance since… (the next point)

- **Memory performance is the limiting CPI performance factor.** In modern PCs, the processor bus is slow compared to the internal clock frequency. Thus, off-chip memory-accesses dominate processor CPI as opposed to internal instruction execution performance.

  The VIA C3 addresses this phenomenon by providing many specific features designed to reduce bus activity: large primary caches, large TLBs, aggressive prefetching, an efficient level-2 cache (new to the VIA C3 Samuel 2 and Ezra), and so forth.

- **Different market segments have different workload characteristics.** The hardware, operating systems, and applications used in our target market (low-end desktop and mobile PCs) have different technical characteristics than those in the high-end, workstation, or server market.

  The VIA C3 family exploits these differences by implementing very specific design tradeoffs, providing high performance with low cost in the target environments. These optimizations are based on extensive analysis of actual behavior of target-market hardware and software.

- **Small is beautiful.** VIA C3 processors are highly optimized for small die size. In addition to the obvious cost benefits, this small size reduces power consumption and improves reliability.

2.2 COMPONENT SUMMARY

2.2.1 GENERAL ARCHITECTURE & FEATURES

Figure 2-1 illustrates the basic 12-stage (integer) pipeline structure of the VIA C3 family. At a high level, there are four major functional groups: I-fetch, decode and translate, execution, and data cache.

The *I-fetch* components deliver x86 instruction bytes from the large I-cache or the external bus. Large buffers allow fetching to proceed asynchronously to other operations. The *decode and translate* components convert x86 instruction bytes into internal execution forms. Branches are also identified, predicted, and the targets prefetched. Large buffers allow decoding and translating to proceed asynchronously to other operations. The *execution* components issue, execute, and retire internal instructions. The *data cache* components manage the efficient loading and storing of execution data to and from the caches, bus, and internal components.

At one level the VIA C3 architecture seems simple: instructions are issued, executed, and retired in order, only one instruction can be issued per clock, and most data cache misses stall the pipeline. However, the design is actually highly optimized and highly tuned to achieve high performance in the targeted environment. Some of the significant features providing this performance are:

- **High internal clock frequency.** The 12-stage pipeline facilitates high MHz
- **Large on-chip caches and TLBs.** The VIA C3 implements large caches and TLBs that significantly reduce stalls due to bus traffic. These primary caches and TLBs are larger than those on any other x86 processor:
  - Two 64-KB primary (L1) caches with 4-way associativity
  - A (new to the VIA C3 Samuel 2 and Ezra) 4-way 64-KB unified level-2 (L2) cache
  - Two 128-entry TLBs with 8-way associativity
  - Two 8-entry page directory caches that effectively eliminate loads of page directory entries upon TLB misses
  - A four-entry (8 bytes each) store queue that also forwards store data to subsequent loads
  - A four-entry write buffer that also performs write combining
- **Extensive features to further minimize bus stalls. These include:**
  - Full memory type range registers (MTRRs)
  - A non-stalling write-allocate implementation
  - Implementation of the "cache lock" feature
  - Non-blocking out-of-order retirement of pipeline stores
  - Implementation of x86 prefetch instruction (3DNow!)
  - Implicit speculative *data* prefetch into D-cache
  - Aggressive implicit instruction prefetch into I-cache
  - Highly asynchronous execution with extensive queuing to allow fetching, decoding and translating, executing, and data movement to proceed in parallel
- **High-performance bus implementation.** The VIA C3 (socket 370 and EBGA) compatible bus implementation includes the following performance enhancements:
  - No stalls on snoops
  - Up to 8 transactions can be initiated by the processor
  - Aggressive write pipelining
  - Smart bus allocation prioritization
  - 100MHz and 133MHz bus operation
- **Good performance for highly used instructions.** Heavily used instructions—including complex functions such as protect-mode segment-register loads and string instructions—are executed fast. In particular, the pipeline is arranged to provide one-clock execution of the heavily used register–memory and memory–register forms of x86 instructions. Many instructions require fewer pipeline clocks than on comparable processors.

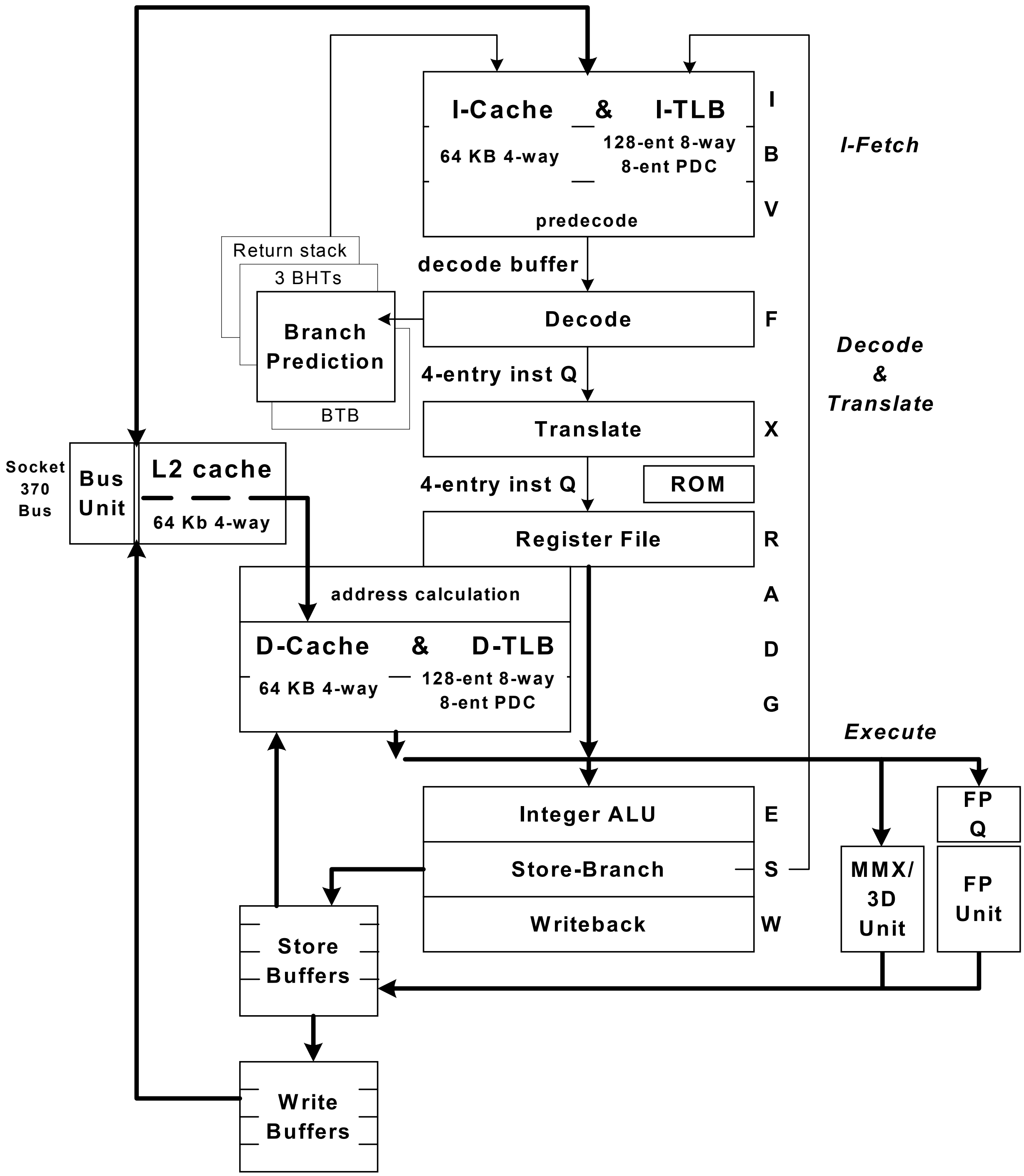


Figure 2-1. The VIA C3 Ezra Processor Pipeline Structure

2.2.2 INSTRUCTION FETCH

The first three pipeline stages (I, B, V) deliver aligned instruction data from the I-cache or external bus into the instruction decode buffers. These stages are fully pipelined such that on each clock cycle a new instruction fetch address is fetched from the I-cache and either 16 bytes (I-cache) or 8-bytes (bus) of instruction data is delivered for decode.

The primary I-cache contains 64 KB organized as four-way set associative with 32-byte lines. The associated large I-TLB contains 128 entries organized as 8-way set associative. In addition, the I-TLB includes an eight-entry page directory cache that significantly reduces the TLB miss penalty. The cache, TLB, and page directory cache all use a pseudo-LRU replacement algorithm.

As opposed to many other contemporary x86 processors, the data in the I-cache is exactly what came from the bus; that is, there are no “hidden” predecode bits. This allows a large cache capacity in a small physical size. The instruction data is predecoded as it comes out of the cache; this predecode is overlapped with other required operations and, thus, effectively takes no time.

The fetched instruction data is placed sequentially into multiple buffers. Starting with a branch, the first branch-target byte is left adjusted into the instruction decode buffer (the *XIB)*. As instructions are decoded, they are removed from the XIB. Each clock, new incoming fetch data is concatenated with the remaining data in the XIB. Once the XIB is filled, fetching continues to fill three 16-byte buffers that feed the XIB.

If an I-cache miss occurs during the filling of the fetch buffers, up to four 32-byte lines are speculatively prefetched from the external bus into the cache. These prefetches are pipelined on the bus for optimal bus efficiency. This aggressive prefetch algorithm exploits the large size and four-way structure of the I-cache, as well as the high-bandwidth bus implementation.

2.2.3 INSTRUCTION DECODE

Instruction bytes in the XIB are decoded and translated into the internal format by two pipeline stages (F, X). These units are fully pipelined such that, in general, an x86 instruction is decoded and translated every clock cycle.

The F stage decodes and "formats" an x86 instruction (completely contained in the XIB) into an intermediate x86 format. This process requires only one clock cycle for every x86 instruction. However, instruction prefixes other than 0F require an additional cycle for each prefix. The internal-format instructions are placed into a five-deep FIFO queue: the *FIQ*. The FIQ facilitates the asynchronous fetch and "lookahead" capability of the formatter such that the extra cycles for prefixes rarely result in a bubble in the execution pipeline. The FIQ also allows for branch prediction to proceed even if the pipeline ahead of the branch is stalled.

The X-stage "translates" an intermediate-form x86 instruction from the FIQ into the internal micro-instruction format. The output of the translator contains: (1) the internal micro-instruction stream to perform the x86 instruction function, (2) the immediate data fields from the x86 instruction, and (3) various x86 state information used to control execution (for example, operand size). The internal micro-instruction stream consists of micro-instructions directly generated by the translator plus, in some cases, an on-chip ROM address. The translator can generate up to three micro-instructions per clock plus a ROM address. Most x86 instructions except for repetitive string operations are translated in one clock.

Executable micro-instructions can come from either the XIQ or the on-chip ROM. For performance-sensitive instructions, there is no delay due to access of micro-code from ROM. The microcode ROM capacity is larger than most x86 microcode ROMs to allow unimportant (relative to performance) functions to be performed in microcode (versus in hardware). The large size also allows the inclusion of extensive self-test microcode and extensive built-in debugging aids (for processor design debug).

Instruction fetch, decode, and translation are made asynchronous from execution via a five-entry FIFO queue (the *XIQ*) between the translator and the execution unit. This queue allows the formatter and translator to "look-ahead" and continue translating x86 instructions even though the execution unit is stalled.

2.2.4 BRANCH PREDICTION

The VIA C3 implements a very unique and sophisticated, yet relatively small, branch prediction mechanism. Several different mechanisms predict all basic types of branches: jumps and calls using a displacement, returns, indirect jumps and calls, and "far" (inter-segment) branches (new to the VIA Ezra). These types of jumps represent almost 25% of all instructions executed.

The prediction starts when the branch is decoded in the F stage. If a branch is predicted taken, the target is fetched such that a four-clock "bubble" appears to occur. In practice however, this bubble averages less than two clocks due to the extensive instruction queuing below the F stage; the delay in fetching the predicted branch target is absorbed by execution of these queued instructions. There is also a special case: short forward branches that "hit" in the XIB require only one clock delay.

For displacement-from branches, the target address is directly calculated. This direct calculation of the target address eliminates the need for a large branch-target buffer (a *BTB* or *BTAC*). The direct calculation always produces the correct address whereas a BTB often mispredicts the address, or fails to predict at all.

Two different prediction algorithms are used to predict the direction of conditional branches. The underlying theory behind this approach is that, in practice, branches tend to fall into two categories. Most branches can be accurately predicted by a classic mechanism using a counter associated with the branch location. The associated VIA C3 mechanism is a simple array of predictors with 8K entries (versus 4K in the VIA C3 Samuel processor).

However, about 20% of branches are hard to predict, because whether they branch or not depends on dynamic execution state. The VIA C3 mechanism for predicting these pattern-based branches uses a 13-bit global branch history XOR'd with the branch instruction address to index a branch history table with 8K entries (versus 4K in the VIA C3 Samuel processor). This branch-history mechanism is called *g-share* in the technical literature. A third 4K-entry table selects which predictor (simple or history-based) to use for a particular branch based on the previous behavior of the branch.

All three predictor tables actually use a one-bit "agree/disagree" predictor rather than the conventional two-bit counter. This unique approach uses a static predictor that considers the type of conditional branch and what type of instruction previously set the condition flag being tested. Rather than indicate taken or not, the one-bit entry in the predictor tables predicts whether the conditional branch direction agrees with the static prediction. The static predictor is accurate about 70% of the time, and the dynamic predictor is accurate about 95% of the time.

In addition, x86 return instructions are predicted by a 16-entry return-address stack. This prediction is about 90% accurate.

Finally, indirect jump and call instructions are predicted using a conventional 128-entry (8-way) BTB (versus 64 and 4-way in the VIA C3 Samuel processor). This small array predicts about 75% of indirect branches correctly. The BTB is also used to predict inter-segment ("far") branches.

2.2.5 INTEGER UNIT

Internal micro-instructions are executed in a tightly coupled seven-stage (integer) pipeline that is similar in structure to a basic four-stage RISC pipeline with extra stages for memory loads and stores inserted. These extra stages allow instructions that perform a load and ALU operation, or a load, ALU, and store operation to execute in one clock.

The micro-instructions and associated execution units are highly tuned for the x86 architecture. The micro-instructions closely resemble highly used x86 instructions. Examples of specialized hardware features supporting the x86 architecture are: hardware handling of the x86 condition codes, segment descriptor manipulation instructions, and hardware to automatically save the x86 floating-point environment.

The integer execution stages are:

- **Decode stage (R):** Micro-instructions are decoded, integer register files are accessed and resource dependencies are evaluated.
- **Addressing stage (A):** Memory addresses are calculated and sent to the D-cache. The VIA C3 in EBGA is capable of calculating most x86 instruction address forms in one clock; a few forms containing two registers or a shifted index register require two clocks. Addresses can be automatically incremented or decremented for implementing stack and string operations.

  Some ALU operations are duplicated in this stage to reduce AGI (address generation interlock) stalls. For example, the sequence of adding to the stack pointer register followed by using it in a return does not cause an AGI. In general, register adds and moves are performed in the A-stage.
- **Cache Access stages (D, G):** The D-cache and D-TLB are accessed and aligned load data returned at the end of the G-stage. The cache is fully pipelined such that a new load or store can be started and data returned each clock.
- **Execute stage (E):** Integer ALU operations are performed. All basic ALU functions take one clock except multiply and divide. Load-ALU and Load-ALU-store instructions require only one clock (the data is loaded before the ALU stage). Results from the ALU can be forwarded to any instruction following the ALU instruction using a unique result-forwarding cache.

  Branch instructions are evaluated in this stage and, if incorrectly predicted, the corrected branch address is sent to the I-cache during the next clock. No additional action is required for correctly predicted branches.

  During this stage the floating-point, MMX, and 3DNow! execution units access their registers or consume the load data just delivered. These execution units “hang off” the end of the main execution unit so that load-ALU operations for these units can be pipelined in one clock.
- **Store stage (S):** Integer store data is grabbed in this stage and placed in a store buffer. Positioning stores in this stage allows a one-clock load–ALU–store instruction. MMX and floating-point stores, however, are placed in the store buffer in a subsequent cycle due to their deep pipelines.
- **Write-back stage (W):** The results of operations are committed to the register file.

2.2.6 D-CACHE & DATAPATH

The D-cache contains 64 KB organized as four-way set associative with 32-byte lines. The associated large D-TLB contains 128 entries organized as 8-way set associative. In addition, the D-TLB includes an eight-entry page directory cache that significantly reduces the TLB miss penalty. The cache, TLB, and page directory cache all use a pseudo-LRU replacement algorithm.

Associated with the D-cache are store and write buffers. All stores that pass the W-stage are placed in a four-entry store buffer (8 bytes each). Stores from the store buffer are drained to the D-cache or bus in actual program order, regardless of when the data appeared in the store buffers. Stores that miss the D-cache percolate to the write buffers where they wait for the bus. If the memory region is defined as write-combining, the stores can combine in the write buffers.

2.2.7 L2 CACHE

The VIA C3 contains an efficient level-2 cache. This L2 cache is *exclusive* (this approach is sometimes called a "victim" cache). This means that the contents of the L2 cache at any point in time are not contained in the two 64-KB L1 caches. As lines are displaced from the L1 caches (due to bringing in new lines from memory), the displaced lines are placed in the L2 cache. Thus, a future L1-cache miss on this displaced line can be satisfied by returning the line from the L2 cache instead of having to access the external memory.

Thus, the total effective cache size on the VIA C3 is 192 KB (two 64-KB L1 caches and the 64-KB L2). Processors, such as the Intel Celeron processor, that have an *inclusive* L2 cache have a total effective cache size equal to the L2 size (128 KB in the case of the current Intel Celeron processor). This is because the contents of the L1 caches are duplicated within an inclusive L2 cache.

The L2 cache is also "unified;" that is, it contains lines displaced from both the L1 I-cache and the L1 D-cache. Thus the L2 cache provides a significant assist for the cases where an I-fetch hits in the L1 data cache, or a data reference hits in the L1 I-cache. To provide correct execution semantics, these cases require the hit line to be ejected from the L1 cache.

The VIA C3 in EBGA L2 cache is 64 KB organized as four-way set associative with 32-byte lines and uses a pseudo-LRU replacement algorithm.

2.2.8 FP UNIT

In addition to the integer execution unit, the VIA C3 has a separate 80-bit floating-point execution unit that can execute x86 floating-point instructions in parallel with integer instructions. Since the VIA C3 is designed for mainstream applications that stress integer performance, floating-point throughput is not a priority. The FP unit is clocked at ½ the processor clock speed; that is, in an 800-MHz VIA C3, the FP-unit runs at 400 MHz.

Floating-point instructions proceed through the integer R, A, D, and G stages. Thus, load-ALU x86 floating-point instructions do not require an extra clock for the load. Floating-point instructions are then passed from the integer pipeline to the FP-unit through an 8-instruction FIFO queue (new to the VIA C3 in EBGA). This queue, which runs at the processor clock speed, decouples the slower running FP unit from the integer pipeline so that the integer pipeline can continue to process instructions overlapped with up to eight FP instructions.

Basic arithmetic floating-point instructions (add, multiply, divide, square root, compare, etc.) are represented by a single internal floating-point instruction. Certain little-used and complex floating point instructions (sin, tan, etc.), however, are implemented in microcode and are represented by a long stream of instructions coming from the ROM. These instructions "tie up" the integer instruction pipeline such that integer execution cannot proceed until they complete.

One floating-point instruction can issue form the FP queue to the FP unit every two clocks. The FP pipeline is six-stages deep following the E stage. It is partially pipelined: a non-dependent add or multiply can start every two clocks.

2.2.9 MMX UNIT

The VIA C3 contains a separate execution unit for the MMX-compatible instructions. MMX instructions proceed through the integer R, A, D, and G stages. Thus, load-ALU x86 MMX instructions do not require an extra clock for the load. One MMX instruction can issue into the MMX unit every clock.

The MMX multiplier is fully pipelined and can start one non-dependent MMX multiply[-add] instruction (which consists of up to four separate multiplies) every clock. Other MMX instructions execute in one clock. Multiplies followed by a dependent MMX instruction require two clocks.

Architecturally, the MMX registers are the same as the floating-point registers. However, there are actually two different register files (one in the FP-unit and one in the MMX units) that are kept synchronized by hardware.

2.2.10 3DNOW! UNIT

The VIA C3 contains a separate execution unit for the 3DNow! instructions. These instructions are compatible with the AMD K6-II™ processor 3DNow! instructions and provide performance assists for graphics transformations via new SIMD single-precision floating-point capabilities. 3DNow! instructions proceed through the integer R, A, D, and G stages. Thus, load-ALU x86 3DNow! instructions do not require an extra clock for the load. One 3DNow! instruction can issue into the 3DNow! unit every clock

The 3DNow! unit has two single-precision floating-point multipliers and two single-precision floating-point adders. Other functions such as conversions, reciprocal, and reciprocal square root are provided.

The multiplier and adder are fully pipelined and can start any non-dependent 3DNow! instruction every clock.

2.2.11 BUS UNIT

The VIA C3 in EBGA bus unit provides an external bus interface, supporting bus speeds of 100 MHz and 133 MHz.

The VIA C3 in EBGA bus implementation has no stalls on snoops, up to eight transactions can be generated (versus four), and stores are more heavily pipelined on the bus.

2.2.12 POWER MANAGEMENT

Power management is not a component but rather a pervasive feature in all components. There are two major modes: powering down components based on bus signals such as stop clock and sleep, and dynamically powering off components during execution when they are not needed.

The VIA C3 implements the bus-controlled power management modes including: autoHALT power down state, stop grant state, sleep state, and deep sleep, or stop clock, state.

While in normal running state, all major functional components are powered off when not in use. All caches, tags and TLBs are turned on and off on a clock-by-clock basis based on their usage. Other major components such as the MMX unit, the 3DNow! unit, the FP unit, and the ROM are turned on and off as used. In addition, even in blocks that are powered up, internal buses and latches hold their values when not needed to reduce dynamic power consumption.

This page is intentionally left blank.

SECTION

3

PROGRAMMING INTERFACE

3.1 GENERAL

The VIA C3's functions include:

- All basic x86 instructions, registers, and functions
- All floating-point (numeric processor) instructions, registers and functions
- All basic operating modes: real mode, protect mode, virtual-8086 mode
- System Management Interrupt (SMI) and the associated System Management Mode (SMM)
- All interrupt and exception functions
- All debug functions (including the new I/O breakpoint function)
- All input/output functions
- All tasking functions (TSS, task switch, etc.)
- Processor initialization behavior
- Page Global Enable feature

The VIA C3 in EBGA, in addition to the MMX instructions, also includes instructions to boost the performance of 3D graphics compatible with the AMD 3DNow! Technology.

However, there are some differences between the VIA C3 in EBGA and the Celeron processor. These differences fall into three groups:

- **Implementation-specific differences**. Examples are cache and TLB testing features, and performance monitoring features that expose the internal implementation features. These types of functions are incompatible among *all* different x86 implementations.
- **Omitted functions**. Some Intel Celeron processor functions are not provided on the VIA C3 in EBGA because they are not used or are not needed in the targeted PC systems. Examples are some specific bus functions such as functional redundancy checking and performance monitoring. Other examples are architectural extensions such as support for 4-MByte pages, Page Attribute Tables, etc.

  These types of differences are similar to those among various versions of the processors. The CPUID instruction is used by system software to determine whether these features are supported.
- **Low-level behavioral differences**. A few low-level VIA C3 in EBGA functions are different from Intel Celeron because the results are (1) documented in the documentation as *undefined*, and (2) known to be different for different x86 implementations. That is, compatibility with the Intel Celeron processor for these functions is clearly not needed for software compatibility (or they would not be different across different implementations).
- **EBGA package**. The VIA C3 is exclusively produced in its advanced, inexpensive, compact and thermally efficient EBGA package.

This chapter summarizes the first three types of differences: additional functions, implementation-specific functions, and omitted functions. *Appendix A* contains more details on machine-specific functions.

3.2 ADDITIONAL FUNCTIONS

The VIA C3 includes AMD-compatible 3DNow! instructions to boost the performance of 3D applications. These instructions are not defined in this datasheet but are defined in the appropriate AMD documentation.

3.3 MACHINE-SPECIFIC FUNCTIONS

3.3.1 GENERAL

All x86 processor implementations provide a variety of *machine-specific functions*. Examples are cache and TLB testing features and performance monitoring features that expose the internal implementation features.

This section describes the VIA C3 in EBGA machine-specific functions that are most likely used by software, and compares them to related processors where applicable. *Appendix A* describes the VIA C3 in EBGA machine-specific registers (*MSRs*).

This section covers those features of Intel Pentium-compatible processors that are used to commonly identify and control processor features. All Pentium-compatible processors have the same mechanisms, but the bit-specific data values often differ.

3.3.2 STANDARD CPUID INSTRUCTION FUNCTIONS

The CPUID instruction is available on all contemporary x86 processors. The CPUID instruction has two standard functions requested via the EAX register. The first function returns a vendor identification string in registers EBX, ECX, and EDX. The second CPUID function returns an assortment of bits in EAX and EDX that identify the chip version and describe the specific features available.

The EAX:EBX:ECX:EDX return values of the CPUID instruction executed with EAX == 0 are:

Table 3-1. CPUID Return Values (EAX = 0)

REGISTER[BITS] – MEANING	*VIA C3*
EAX (highest EAX input value understood by CPUID)	1
EBX:EDX:ECX (vendor ID string)	“Centaur Hauls”

The EAX return values of the CPUID instruction executed with EAX == 1 are:

Table 3-2. CPUID EAX Return Values (EAX = 1)

EAX BITS - MEANING	*VIA C3*
3:0 - Stepping ID	
7:4 - Model ID	Same as the return value in EDX after Reset (see next section)
11:8 - Family ID	
13:12 – Type ID	

The EDX return values of the CPUID instruction with EAX == 1 are:

Table 3-3. CPUID EDX Return Values (EAX = 1)

EDX BITS – MEANING	*VIA C3*	*NOTES*
0 - FPU present	1	
1 - Virtual Mode Extension	0	
2 - Debugging Extensions	1	
3 - Page Size Extensions (4MB)	0	
4 - Time Stamp Counter (TSC) supported	1	
5 - Model Specific Registers present	1	
6 - Physical Address Extension	0	
7 - Machine Check Exception	0/1	
8 - CMPXCHG8B instruction	0/1	1
9 - APIC supported	0	2
10- Reserved		
11- Fast System Call	0	
12- Memory Range Registers	1	
13 - PTE Global Bit supported	1/0	3
14- Machine Check Architecture supported	0	
15- Conditional Move supported	0	
16- Page Attribute Table	0	
17- 36-bit Page Size Extension	0	
18- Processor serial number	0	
19:22 – Reserved		
23- MMX supported	1	
24- FXSR	0	
25- Streaming SIMD Extension supported	0	
26:31 – Reserved		

Notes On CPUID Feature Flags:

General: an “x/y” entry means that the default setting of this bit is x but the bit (and the underlying function) can be set to y using the FCR MSR.

1. The CMPXCHG8B instruction is provided and always enabled, however, it appears disabled in the corresponding CPUID function bit 0 to avoid a bug in an early version of Windows NT. However, this default can be changed via a bit in the FCR MSR.
2. SMP is not supported; it has no utility in the target system environment.
3. The VIA C3’s support for Page Global Enable can be enabled or disabled by a bit in the FCR. The CPUID bit reports the current setting of this enable control.

3.3.3 EXTENDED CPUID INSTRUCTION FUNCTIONS

The VIA C3 in EBGA supports extended CPUID functions. These functions provide additional information about the VIA C3. Extended CPUID functions are requested by executing CPUID with EAX set to any value in the range 0x80000000 through 0x80000006.

The following table summarizes the extended CPUID functions.

Table 3-4. Extended CPUID Functions

EAX	*TITLE*	*OUTPUT*
80000000	Largest Extended Function Input Value	EAX=80000006 EBX,ECX,EDX=Reserved
80000001	Processor Signature and Feature Flags	EAX=Processor Signature EBX,ECX=Reserved EDX=Extended Feature Flags
80000002	Processor Name String	EAX,EBX,ECX,EDX
80000003	Processor Name String	EAX,EBX,ECX,EDX
80000004	Processor Name String	EAX,EBX,ECX,EDX
80000005	TLB and L1 Cache Information	EAX = Reserved EBX = TLB Information ECX = L1 Data Cache Information EDX = L1 Instruction Cache Information
80000006	L2 Cache Information	EAX, EBX, EDX = Reserved ECX = L2 Cache Information

Largest Extended Function Input Value (EAX==0x80000000)
Returns 0x80000006 in EAX, the largest extended function input value.

Processor Signature and Feature Flags (EAX==0x80000001)
Returns processor version information in EAX, this value is identical to the value of EDX after RESET.

Returns feature flags in EDX, this value is identical to the value in EDX after CPUID standard function 1, with the exception of bit 31:

EDX[31]=0 3DNow! instructions not supported.

EDX[31]=1 3DNow! instructions supported.

Note that if FCR[20]=0 then 3DNow! instructions are not supported and EDX[31] will be 0.

Processor Name String (EAX==0x80000002–0x80000004)

Returns the name of the processor, suitable for BIOS to display on the screen (ASCII). The string can be up to 48 characters in length. If the string is shorter, the rightmost characters are padded with zero. The leftmost characters go in EAX, then EBX, ECX, and EDX. The leftmost character goes in least significant byte (little endian).

For example, the string “VIA Ezra” would be returned by extended function EAX=0x80000002 as follows:

EAX = 0x20414956

EBX = 0x61727A45

ECX = 0x00000000

EDX = 0x00000000

Since the string is less than 17 bytes, the extended functions EAX=0x80000003 and EAX=0x80000004 return zero in EAX, EBX, ECX, and EDX.

L1 Cache Information (EAX == 0x80000005)

Returns information about the implementation of the TLBs and caches:

Table 3-5. L1 Cache & TLB Configuration Encoding

REGISTER	*DESCRIPTION*	*VALUE*
EAX	Reserved	
EBX	TLB Information	
EBX[31:24]	D-TLB associativity	8
EBX[23:16]	D-TLB # entries	128
EBX[15: 8]	I-TLB associativity	8
EBX[7: 0]	I-TLB # entries	128
ECX	L1 Data Cache Information	
ECX[31:24]	Size (Kbytes)	64
ECX[23:16]	Associativity	4
ECX[15: 8]	Lines per Tag	1
ECX[7: 0]	Line Size (bytes)	32
EDX	L1 Instruction Cache Information	
EDX[31:24]	Size (Kbytes)	64
EDX[23:16]	Associativity	4
EDX[15: 8]	Lines per Tag	1
EDX[7: 0]	Line Size (bytes)	32

L2 Cache Information (EAX == 0x80000006)

Returns information about the implementation of the L2 cache:

Table 3-6. L2 Cache Configuration Encoding

REGISTER	DESCRIPTION	VALUE
EAX, EBX, EDX	Reserved	
ECX	L2 Data Cache Information	
ECX[31:24]	Size (Kbytes)	64
ECX[23:16]	Associativity	4
ECX[15: 8]	Lines per Tag	1
ECX[7: 0]	Line Size (bytes)	32

3.3.4 PROCESSOR IDENTIFICATION

The VIA C3 in EBGA provides several machine-specific features. These features are identified by the standard CPUID function EAX=1.

Other machine-specific features are controlled by VIA C3 MSRs. Some of these features are not backward-compatible with the predecessors in the IDT WinChip family.

System software must not assume that all future processors in the VIA processor family will implement all of the same machine-specific features, or even that these features will be implemented in a backward-compatible manner. In order to determine if the processor supports particular machine-specific features, system software should follow the following procedure.

Identify the processor as a member of the VIA processor family by checking for a Vendor Identification String of "CentaurHauls" using CPUID with EAX=0. Once this has been verified, system software must determine the processor version in order to properly configure the machine-specific registers.

In general system software can determine the processor version by comparing the Family and Model Identification fields returned by the CPUID standard function EAX=1.

If the processor version is not recognized then system software must not attempt to activate any machine-specific feature.

3.3.5 EDX VALUE AFTER RESET

After reset the EDX register holds a component identification number as follows:

	31:14	13:12	11:8	7:4	3:0
EDX	Reserved	Type ID	Family ID	Model ID	Stepping ID
	18	2	4	4	4

The specific values for the VIA C3 Samuel 2 are:

PROCESSOR	***TYPE ID***	***FAMILY ID***	***MODEL ID***	***STEPPING ID***
VIA C3 Samuel 2	0	6	7	Begins at 0

The specific values for the VIA C3 Ezra are:

PROCESSOR	***TYPE ID***	***FAMILY ID***	***MODEL ID***	***STEPPING ID***
VIA C3 Ezra	0	6	7	Begins at 8

The specific values for the VIA C3 Ezra-T are:

PROCESSOR	***TYPE ID***	***FAMILY ID***	***MODEL ID***	***STEPPING ID***
VIA C3 Ezra-T	0	6	8	Begins at 0

3.3.6 CONTROL REGISTER 4 (CR4)

Control register 4 (CR4) controls some of the advanced features of the Celeron processor. The VIA C3 in EBGA provides a CR4 with the following specifics:

Table 3-7. CR4 Bits

CR4 BITS – MEANING	*VIA C3 EZRA*	*CELERON MODEL 6*	*CELERON MODEL 8*	*NOTES*
0: VME: Enables VME feature	0	0/1	0/1	1
1: PVI: Enables PVI feature	0	0/1	0/1	1
2: TSD: Makes RDTSC inst privileged	0/1	0/1	0/1	
3: DE: Enables I/O breakpoints	0/1	0/1	0/1	
4: PSE: Enables 4-MB pages	0	0/1	0/1	1
5: PAE: Enables address extensions	r	r	r	
6: MCE: Enables machine check exception	0/1	0/1	0/1	2
7: PGE: Enables global page feature	0/1	0/1	0/1	
8: PCE: Enables RDPMC for all levels	0/1	0/1	0/1	
9: OSFXSR: Enables FXSAVE//FXRSTOR Support	r	r	0/1	
10: OSXMMEXCPT: O/S Unmasked Exception Support	r	r	0/1	
31:11 – reserved	r	r	r	

Notes On CR4

General: a “0/1” means that the default setting of this bit is 0 but the bit can be set to (1). A “0” means that the bit is always 0; it cannot be set. An “r” means that this bit is reserved. It appears as a 0 when read, and a GP exception is signaled if an attempt is made to write a 1 to this bit.

1. The VIA C3 Ezra does not provide this “Appendix H” function and this CR4 bit cannot be set. However, no GP exception occurs if an attempt is made to set this bit. The Cyrix 6x86MX processor also does not provide this function.
2. The VIA C3 Ezra Machine Check has different specifics than the Machine Check function of compatible processors.

3.3.7 MACHINE-SPECIFIC REGISTERS

The VIA C3 in EBGA implements the concept of Machine Specific Registers (MSRs). RDMSR and WRMSR instructions are provided and the CPUID instruction identifies that the VIA C3 supports MSRs.

In general, the MSRs have no usefulness to application or operating system software and are not used. (This is to be expected since the MSRs are different on each processor.) *Appendix A* contains a detailed description of the VIA C3’s MSRs.

3.4 OMITTED FUNCTIONS

This section summarizes those functions that are not in the EBGA VIA C3.

Symmetric Multiprocessing Support: APIC

This bus function is omitted since the target market for the VIA C3 in EBGA is portables and typical desktop systems (which do not support APIC multiprocessing).

A bit in the feature identification returned from the CPUID instruction indicates whether this feature is present or not. This enhancement is not provided on the VIA C3 in EBGA.

Other Functions

There are other functions that are not implemented in the VIA C3 in EBGA. These are identified accordingly in the CPUID feature flags, Conditional Move instructions, Page Attribute Tables, 4-Mbyte pages, 36-bit Page Size Extension, and FXSAVE/FXRSTOR instructions.

The VIA C3 contains a 64kB L2 cache. Model specific registers pertaining to the L2 cache are detailed in Appendix A. Model specific registers pertaining to APIC, Machine Check, and Debug and Trace features are not supported.

SECTION

4

HARDWARE INTERFACE

4.1 BUS INTERFACE

The VIA C3 in EBGA bus interface is functionally similar to the Socket 370 interface.

The majority of the balls within the bus interface are involved with the physical memory and I/O interface. The remaining balls are for power and ground, test and debug support, and various ancillary control functions. The balls and associated functions are listed and described in this section.

4.1.1 DIFFERENCES

The areas where the VIA C3 in EBGA differs from compatible processors should not cause operational compatibility issues. These differences are:

- Bus-to-core Ratio Control
- Bus Frequency Control
- Probe Mode / JTAG / TAP Port (see *Test and Debug Section*)

Bus-to-Core Frequency Ratio Control

The VIA C3 supports both fused and software control of the bus-to-core frequency ratio. At reset, the factory-set, fused ratio is used. This ratio can then be adjusted via software. This adjustment lasts until the next reset.

Software can adjust the bus-to-core ratio using the VIA C3 processor's LongHaul extensions. These are documented separately in the VIA C3 LongHaul Specification.

Bus Frequency Selection

The VIA C3 in EBGA supports automated bus frequency selection through the BSEL balls. The BSEL balls are used as a mechanism whereby the processor and the system board can negotiate to support high frequency bus frequencies. The standard BSEL decoding is shown in Table 4-1.

While the VIA C3 in EBGA is designed to operate at bus frequencies of 66, 100, or 133 MHz, performance is improved by running at higher bus frequencies. Various speed bins preclude 133 MHz operation because the available bus-to-core ratios do not permit operation at the desired core MHz.

Processors from these speed bins indicate this by shorting the BSEL[1] ball to ground internal to the package. For these processors the BSEL[0] ball is left floating. Processors from speed bins which permit 133 MHz bus operation indicate this by allowing both BSEL[1] and BSEL[0] to float.

It is anticipated that motherboards will pull up both BSEL balls. The resulting BSEL-indicated bus frequency will then be either 100 MHz or 133 MHz according to speed bin. Bus operation at 66MHz is not desirable.

Table 4-1. BSEL Frequency Mapping

BSEL[1]	***BSEL[0]***	***BUS FREQUENCY***
0	0	66 MHz
0	1	100 MHz
1	0	Reserved
1	1	133 MHz

4.1.2 CLARIFICATIONS

Power Supply Voltage

The VIA C3 in EBGA automatically controls its core processor power supply voltage with the VID balls.

The VID mapping for the VIA C3 in EBGA is in Table 4-2. This mapping corresponds to the VRM8.5 specification.

Table 4-2. Core Voltage Settings

VID4	*VID3*	*VID2*	*VID1*	*VID0*	*VCORE*
0	0	1	0	0	1.050V
1	0	1	0	0	1.075V
0	0	0	1	1	1.100V
1	0	0	1	1	1.125
0	0	0	1	0	1.150V
1	0	0	1	0	1.175V
0	0	0	0	1	1.200V
1	0	0	0	1	1.225V
0	0	0	0	0	1.250V
1	0	0	0	0	1.275V
0	1	1	1	1	1.300V
1	1	1	1	1	1.325V
0	1	1	1	0	1.350V
1	1	1	1	0	1.375V
0	1	1	0	1	1.400V
1	1	1	0	1	1.425V
0	1	1	0	0	1.450V
1	1	1	0	0	1.475V
0	1	0	1	1	1.500V
1	1	0	1	1	1.525V
0	1	0	1	0	1.550V
1	1	0	1	0	1.575V
0	1	0	0	1	1.600V
1	1	0	0	1	1.625V
0	1	0	0	0	1.650V
1	1	0	0	0	1.675V
0	0	1	1	1	1.700V
1	0	1	1	1	1.725V
0	0	1	1	0	1.750V
1	0	1	1	0	1.775V
0	0	1	0	1	1.800V
1	0	1	0	1	1.825V

VCC$_{CMOS}$

The VIA C3 in EBGA renames VCC$_{1.5}$ and VCC$_{CMOS}$ as VTT power pins. This signal is not used by the processor, but is intended to be used by the system as the power supply for CMOS level signals. VCC$_{CMOS}$ should not be expected to source more than 250mA.

RESET#

The VIA C3 in EBGA is reset by the assertion of the RESET# ball, T-3.

Thermal Monitoring

The VIA C3 in EBGA supports thermal monitoring via the THERMDN and THERMDP balls.

4.1.3 OMISSIONS

Driver Termination

Some VIA C3 in EBGA requires external termination. The VIA C3 Ezra-T core supports on-die termination.

System boards must terminate signals to ensure correct operation.

Advanced Peripheral Interrupt Controller (APIC)

The VIA C3 in EBGA does not support the APIC. The APIC balls (PICCLK, PICD0, and PICD1) are specified as reserved.

Breakpoint and Performance Monitoring Signals

The VIA C3 in EBGA internally supports instruction and data breakpoints. However, the VIA C3 in EBGA does not support the external indication of breakpoint matches. Similarly, the VIA C3 in EBGA contains performance monitoring hooks internally, but it does not support the indication of performance monitoring events.

Error Checking

The VIA C3 in EBGA does not support error checking. The BERR#, BINIT#, AERR#, AP#[1:0], DEP#[7:0], IERR#, RP#, and RSP# signals do not exist.

RTTCTRL

Some VIA C3 processors in EBGA do not connect to the RTTCTRL ball. VIA C3 Ezra-T processors use the RTTCTRL ball to control integrated I/O pull-ups.

NCHCTRL

Some VIA C3 processors in EBGA do not connect to the NCHCTRL ball. VIA C3 Ezra-T processors use the NCHCTRL ball to control the output impedance.

4.2 BALL DESCRIPTION

Table 4-3. Ball Descriptions

Ball Name	*Description*	*I/O*	*Clock*
A[31:3]#	The address Bus provides addresses for physical memory and external I/O devices. During cache inquiry cycles, A31#-A3# are used as inputs to perform snoop cycles.	I/O	BCLK
A20M#	A20 Mask causes the CPU to make (force to 0) the A20 address bit when driving the external address bus or performing an internal cache access. A20M# is provided to emulate the 1 MByte address wrap-around that occurs on the 8086. Snoop addressing is not affected.	I (VTT)	ASYNC
ADS#	Address Strobe begins a memory/I/O cycle and indicates the address bus (A31#-A3#) and transaction request signals (REQ#) are valid.	I/O	BCLK
BCLK	Bus Clock provides the fundamental timing for the VIA C3 Ezra CPU. The frequency of the VIA C3 Ezra CPU input clock determines the operating frequency of the CPU's bus. External timing is defined referenced to the rising edge of CLK.	I (2.5V)	--
BNR#	Block Next Request signals a bus stall by a bus agent unable to accept new transactions.	I/O	BCLK
BPRI#	Priority Agent Bus Request arbitrates for ownership of the system bus.	I	BCLK
BR[4:0]	Hardware strapping options for setting the processors internal clock multiplier. Select VIA C3 processors do not have their clock multiplier set to a factory default value. Use jumpers or populate 0Ω resistors to select the rated multiplier. The BR[4:0] balls should be wired to VSS for a value of "0" or wired to OPEN for setting of "1." See Table 4-4 for ratio values.	I	BCLK
BSEL[1:0]	Bus Selection Bus provides system bus frequency data to the CPU.	O (3.3V)	BCLK
BR0#	BR0# drives the BREQ[0]# signal in the system to request access to the system bus.	I/O	None
D[63:0]#	Data Bus signals are bi-directional signals which provide the data path between the VIA C3 Ezra CPU and external memory and I/O devices. The data bus must assert DRDY# to indicate valid data transfer.	I/O	BCLK
DBSY#	Data Bus Busy is asserted by the data bus driver to indicate data bus is in use.	I/O	BCLK
DEFER#	Defer is asserted by target agent (e.g., north bridge) and indicates the transaction cannot be guaranteed as an in-order completion.	I	BCLK
DRDY#	Data Ready is asserted by data driver to indicate that a valid signal is on the data bus.	I/O	BCLK
FERR#	FPU Error Status indicates an unmasked floating-point error has occurred. FERR# is asserted during execution of the FPU instruction that caused the error.	O (VTT)	ASYNC
FLUSH#	Flush Internal Caches writing back all data in the modified state.	I (VTT))	ASYNC
HIT#	Snoop Hit indicates that the current cache inquiry address has been found in the cache (exclusive or shared states).	I/O	BCLK
HITM#	Snoop Hit Modified indicates that the current cache inquiry address has been found in the cache and dirty data exists in the cache line (modified state).	I/O	BCLK
INIT#	Initialization resets integer registers and does not affect internal cache or floating point registers.	I (VTT))	ASYNC
INTR	Maskable Interrupt	I (VTT))	ASYNC
NMI	Non-Maskable Interrupt	I (VTT)	ASYNC
LOCK#	Lock Status is used by the CPU to signal to the target that the operation is atomic.	I/O	BCLK
NCHCTRL	VIA C3 Ezra-T processors use the RTTCTRL ball to control integrated I/O pull-ups. Connect this signal to VTT with a 14Ω resistor.	I	ASYNC
PWRGD	Indicates that the processor's VCC is stable.	I	ASYNC

Ball Name	Description	I/O	Clock
REQ[4:0]#	Request Command is asserted by bus driver to define current transaction type.	I/O	BCLK
RESET#	Resets the processor and invalidates internal cache without writing back.	I	BCLK
RTTCTRL	VIA C3 Ezra-T processors use the NCHCTRL ball to control the output impedance. Connect this signal to VSS with a 56Ω resistor if relying upon on-die termination. Connect this signal to VSS with a 110Ω resistor if relying upon board termination.	I	ASYNC
RS[2:0]#	Response Status signals the completion status of the current transaction when the CPU is the response agent.	I	BCLK
SLP#	Sleep, when asserted in the stop grant state, causes the CPU to enter the sleep state.	I (VTT)	ASYNC
SMI#	System Management (SMM) Interrupt forces the processor to save the CPU state to the top of SMM memory and to begin execution of the SMI services routine at the beginning of the defined SMM memory space. An SMI is a high-priority interrupt than NMI.	I (VTT)	ASYNC
STPCLK#	Stop Clock causes the CPU to enter the stop grant state.	I (VTT)	ASYNC
TRDY#	Target Ready indicates that the target is ready to receive a write or write-back transfer from the CPU.	I	BCLK
VID[3:0]	Voltage Identification Bus informs the regulatory system on the motherboard of the CPU Core voltage requirements.	O (1.5V)	ASYNC

Table 4-4. Clock Ratio

BR[4]	BR[3]	BR[2]	BR[1]	BR[0]	Bus Ratio
0	0	0	0	0	9.0X
0	0	0	0	1	3.0X
0	0	0	1	0	4.0X
0	0	0	1	1	10.0X
0	0	1	0	0	5.5X
0	0	1	0	1	3.5X
0	0	1	1	0	4.5X
0	0	1	1	1	9.5X
0	1	0	0	0	5.0X
0	1	0	0	1	7.0X
0	1	0	1	0	8.0X
0	1	0	1	1	6.0X
0	1	1	0	0	12.0X
0	1	1	0	1	7.5X
0	1	1	1	0	8.5X
0	1	1	1	1	6.5X
1	0	0	0	0	Reserved
1	0	0	0	1	11.0X
1	0	0	1	0	12.0X
1	0	0	1	1	Reserved
1	0	1	0	0	13.5X
1	0	1	0	1	11.5X
1	0	1	1	0	12.5X
1	0	1	1	1	10.5X
1	1	0	0	0	13.0X
1	1	0	0	1	15.0X
1	1	0	1	0	16.0X
1	1	0	1	1	14.0X
1	1	1	0	0	Reserved
1	1	1	0	1	15.5X
1	1	1	1	0	Reserved
1	1	1	1	1	14.5X

4.3 POWER MANAGEMENT

The VIA C3 in EBGA provides both static and dynamic power management.

The VIA C3 in EBGA supports five power management modes: NORMAL state, STOP GRANT state, AUTOHALT state, SLEEP state, and DEEPSLEEP state.

The VIA C3 in EBGA uses dynamic power management techniques to reduce power consumption in the NORMAL state. In NORMAL state, the on-chip arrays, selected datapaths, and the associated control logic are powered down when not in use. In addition, units which are in use attempt to minimize switching of inactive nodes.

- NORMAL state is the normal operating state for the processor.
- STOP GRANT state is the low power state where most of the processor clocks do not toggle. It is entered when the STPCLK# signal is asserted. Snoop cycles are supported in this state.
- AUTO HALT state is the low power state where most of the processor clocks do not toggle. It is entered when the processor executes the HALT instruction. Snoop cycles are supported in this state.
- SLEEP state is the very low power state where only the processor's PLL (phase lock loop) toggles. It is entered from STOP GRANT state when the processor samples the SLP# signal asserted. Snoop cycles that occur while in SLEEP state or during a transition into or out of SLEEP state will cause unpredictable behavior.
- DEEP SLEEP state is the lowest power state. It is entered when the BCLK signal is stopped while the processor in is the SLEEP state. Snoop cycles will be completely ignored in this state.

4.4 TEST & DEBUG

4.4.1 BIST

A Built-in Self-Test (BIST) can be requested as part of the VIA C3 in EBGA reset sequence by holding INIT# asserted as RESET# is de-asserted.

The VIA C3 in EBGA BIST performs the following general functions:

- A hardware-implemented exhaustive test of (1) all internal microcode ROM, and (2) the X86 instruction decode, instruction generation, and entry point generation logic.
- An extensive microcode test of all internal registers and datapaths.
- An extensive microcode test of data and instruction caches, their tags, and associated TLBs.

BIST requires about four million internal clocks.

EAX Value After Reset

The result of a BIST is indicated by a code in EAX. Normally EAX is zero after reset. If a BIST is requested as part of the Reset sequence, EAX contains the BIST results. A 0 in EAX after BIST Reset means that no failures were detected. Any value other than zero indicates an error has occurred during BIST.

4.4.2 JTAG

The VIA C3 in EBGA has a JTAG scan interface that is used for test functions and the proprietary Debug Port. However, the VIA C3 in EBGA does not provide a fully compatible IEEE 1149.1 JTAG function.

From a practical user viewpoint, JTAG does not exist and the associated balls (TCLK, and so forth) should not be used.

4.4.3 DEBUG PORT

Certain processors have a proprietary Debug Port that uses the JTAG scan mechanism to control internal debug features ("probe mode"). These interfaces are not documented and are available (if at all) only under a non-disclosure agreement.

The VIA C3 in EBGA does not have a debug interface.

This page is intentionally left blank.

SECTION

5

ELECTRICAL SPECIFICATIONS

5.1 AC TIMING TABLES

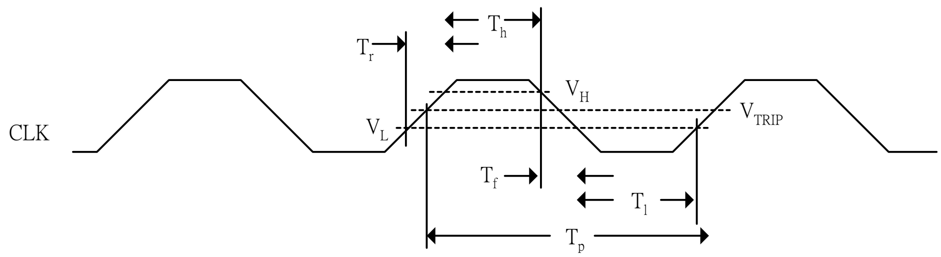
Table 5-1. Clock Switching Characteristics

SYMBOL	PARAMETER	MIN	MAX	UNIT	FIGURE	NOTES
F	BCLK Frequency	66	133	MHz		
t_1	BCLK Period	7.5	15	ns		
t_2	BCLK High Time	1.4		ns		2V
t_3	BCLK Low Time	1.4		ps		8V
t_4	BCLK Fall Time	0.4	1.6	ns		2V ~ 0.8V
t_5	BCLK Rise Time	0.4	1.6	ns		2V ~ 0.8V
t_7	BCLK Period Stability		$\pm$250	ps		

Table 5-2. Output Delay Timings

SYMBOL	PARAMETER	MIN	MAX	UNIT	FIGURE	NOTES
t_{7A}	Output Valid Delay H->L	0.40	3.25	ns		
t_{7B}	Output Valid Delay L->H	0.40	3.25	ns		
T_8	Input Setup Time	0.95		ns		
T_9	Input Hold Time	1.0		ns		
T_{10}	Reset# Pulse Width	1.0		ms		

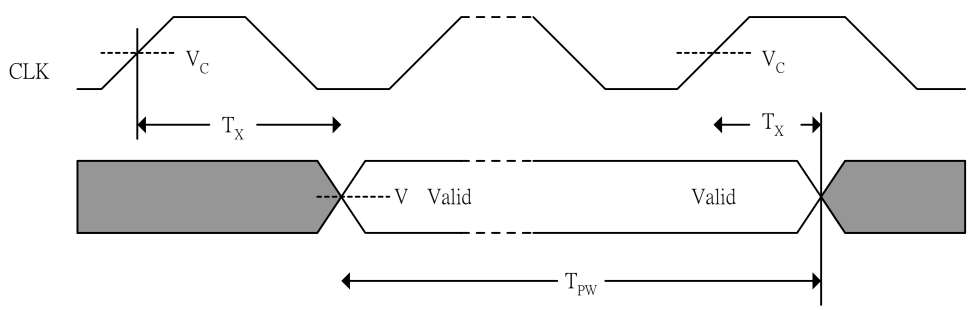
Note: Valid delay timings for these signals are specified into an idealized 50-ohm resistor to 1.5V with VREF at 1.0V. Minimum values guaranteed by design.



NOTES:

T_r=T5S, T5S1 (Rise Time)
T_f=T6S, T6S1 (Fall Time)
T_h=T3S, T3S1 (High Time)
T_l=T4S, T4S1 (Low Time)
T_p=T1S, T1S1 (Period)
V_{TRIP}=1.25v for BCLK
V_L=0.5v for BCLK
V_H=2.0v for BCLK

Figure 5-1. BCLK Generic Clock Timing Waveform



NOTES:

T_X=T7 (Valid Delay)
T_{PW}=T14 (Pulse Width)
V=V_{REF} for bus signal group; 1.0v for CMOS, Open-drain
V_C= 1.25v

Figure 5-2. Valid Delay Timings

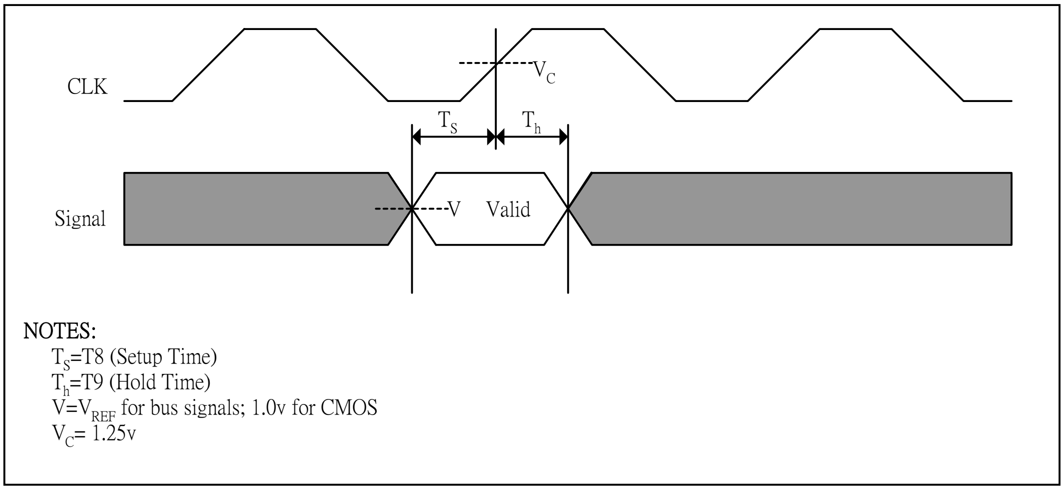


Figure 5-3. Setup and Hold Timings

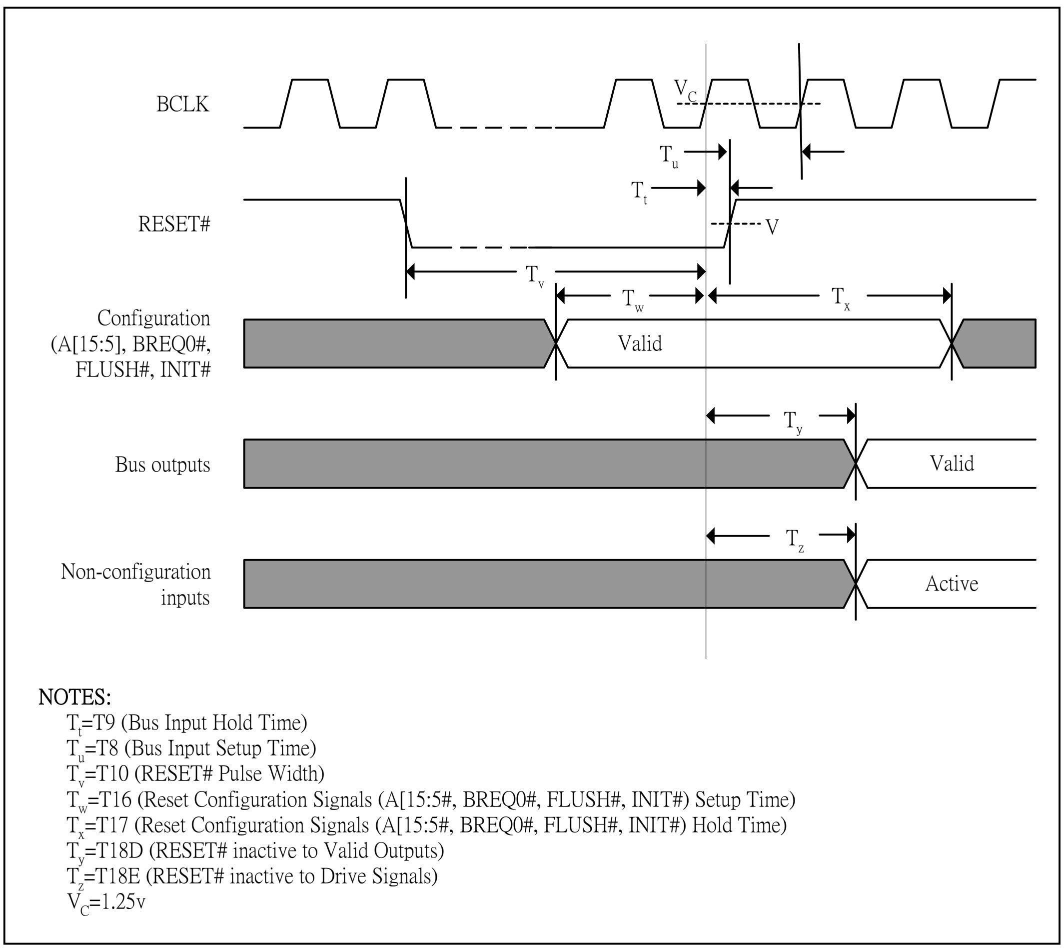
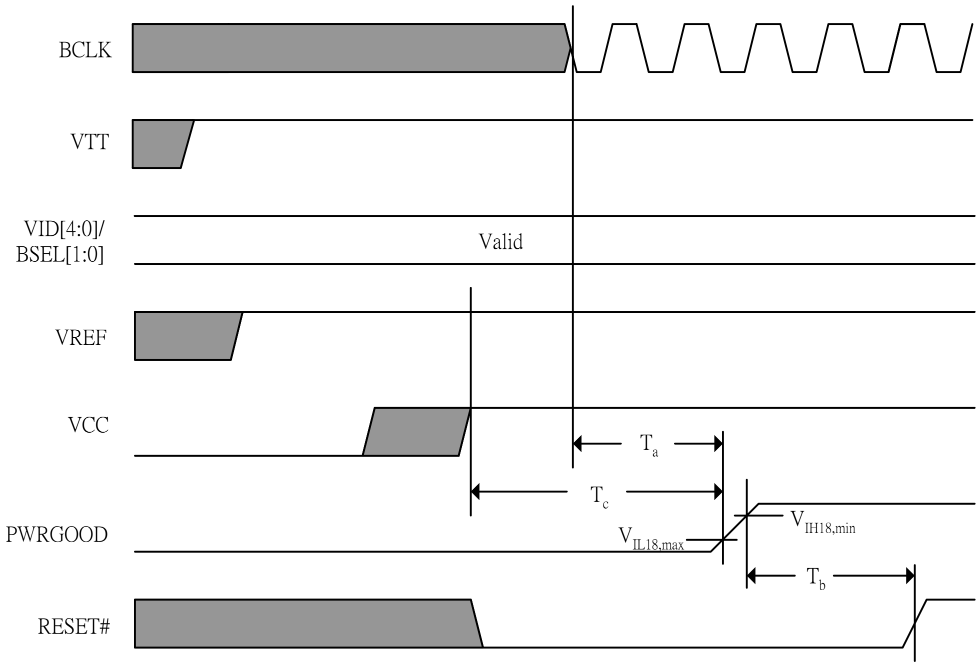


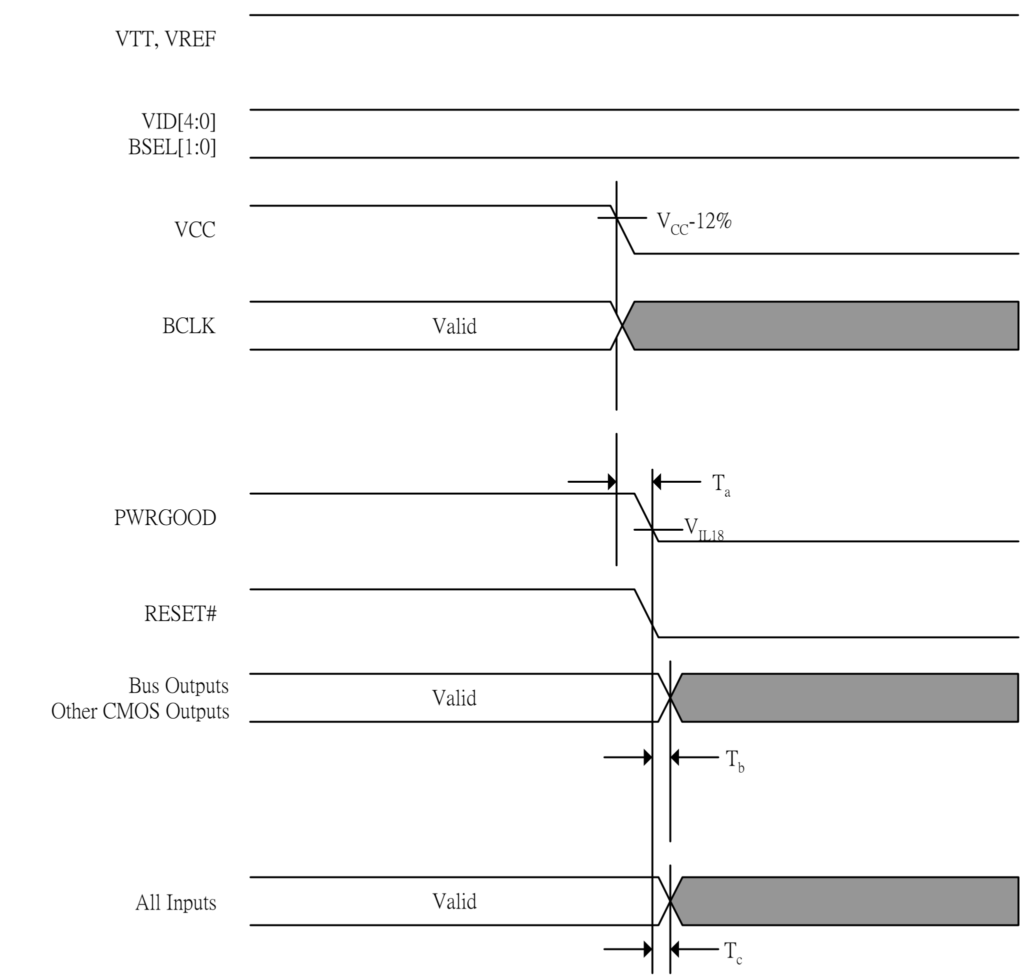
Figure 5-4. Cold/Warm Reset and Configuration Timings



NOTES:

T_a=T15 (PWRGOOD Inactive Pulse Width)
T_b=T18 (RESET#/PWRGOOD Setup Time)
T_c=T18B (Setup time from VCC valid until PWRGOOD assertion)

Figure 5-5. Power-on Sequence and Reset Timings



NOTES:

T_a=T19A (Time from VCC(nominal)-12% to PWRGOOD low)
T_b=T19B (All outputs valid after PWRGOOD low)
T_c=T19C (All inputs required valid after PWRGOOD low)

Figure 5-6. Power Down Sequencing and Timings (VCC Leading)

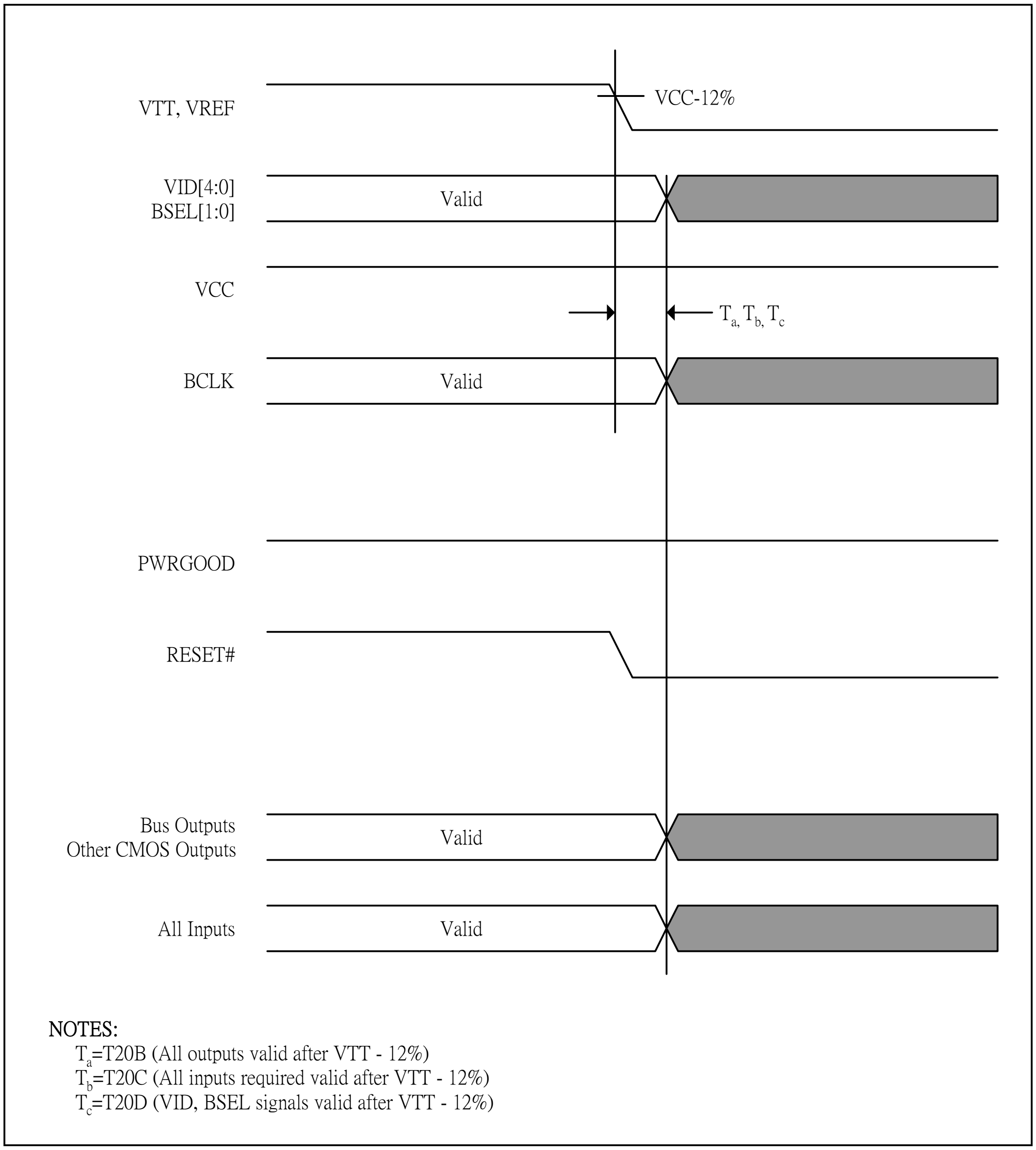


Figure 5-7. Power Down Sequencing and Timings (VTT Leading)

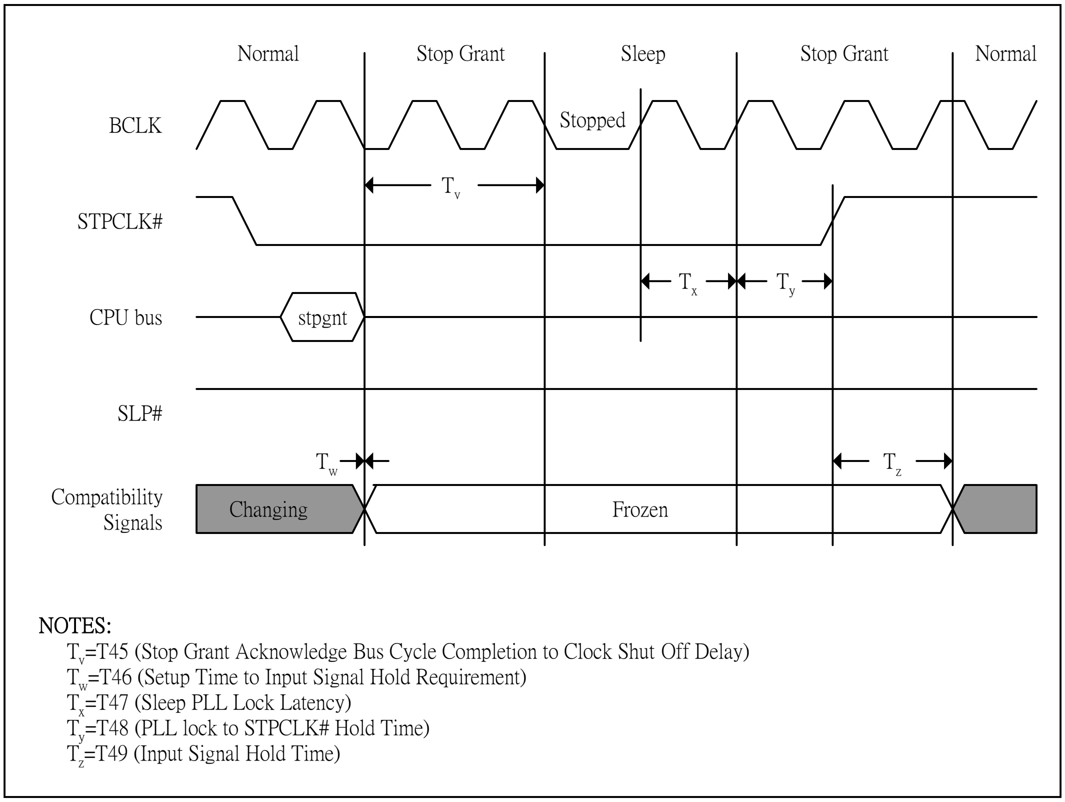
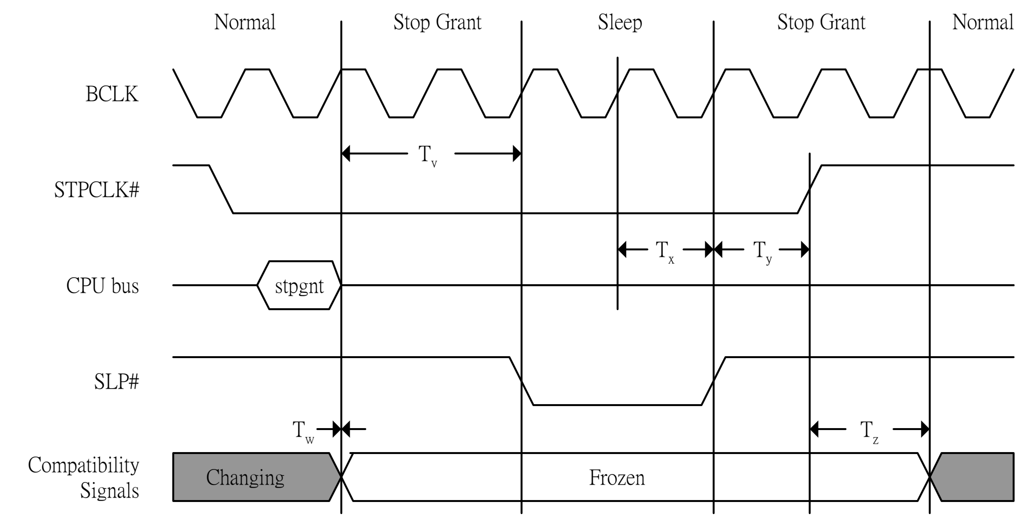


Figure 5-8. Stop Grant/Sleep Timing (BCLK Stopping Method)



NOTES:
T_v=T45 (Stop Grant Acknowledge Bus Cycle Completion to DPSLP# assertion)
T_w=T46 (Setup Time to Input Signal Hold Requirement)
T_x=T47 (Sleep PLL Lock Latency)
T_y=T48 (PLL lock to STPCLK# Hold Time)
T_z=T49 (Input Signal Hold Time)

Figure 5-9. Stop Grant/Sleep Timing (SLP# assertion method)

5.2 DC SPECIFICATIONS

5.2.1 RECOMMENDED OPERATING CONDITIONS

Functional operation of the VIA C3 in EBGA is guaranteed if the conditions in Table 5-3are met. Sustained operation outside of the recommended operating conditions may damage the device.

Table 5-3. Recommended Operating Conditions

PARAMETER	MIN	NOM	MAX	UNITS	NOTES
Operating Case Temperature	0		85	°C	
V_{CORE} Voltage		1.60 / 1.65		V	Samuel 2
V_{CORE} Voltage		1.35 / 1.40		V	Ezra and Ezra-T
V_{CORE} Voltage		1.45		V	Ezra-T
V_{CORE} Static Tolerance	See Table 5-4			V	(1)
V_{CORE} Dynamic Tolerance				V	(2)
V_{TT} Voltage	1.365	1.50	1.635	V	Ezra, Ezra-T
IV_{TT} Termination Supply Current			800	mA	Ezra-T (3)
R_{TT}	50	56	115	Ω	(4)
V_{REF}	-2%	2/3 V_{TT}	+2%	V	
$V_{2.5}$ – 2.5 V Supply Voltage	2.375		2.625	V	
$V_{1.5}$ – 1.5V Supply Voltage	1.365		1.635	V	

Notes:

1. DC measurement
2. AC noise measured with bandwidth limited to 20MHz
3. DC measurement. Measured with 250 µs sampling rate
4. Necessary for Ezra-T. R_{TT} is controlled by RTTCTRL ball. RTTCTRL should be 56Ω when relying upon on-die bus termination. RTTCTRL should be 110Ω when relying upon board termination.

Table 5-4. V_{CC} Static and Transient Tolerance

Icc (A)	Voltage Deviation from VID Setting (mV)				
	Static Tolerance			Transient Tolerance	
	Min	Nom	Max	Min	Max
0	**15**	**40**	**65**	**5**	**85**
2	**5**	**30**	**55**	**-5**	**74**
4	**-5**	**20**	**45**	**-15**	**62**
6	**-15**	**10**	**35**	**-25**	**51**
8	**-25**	**0**	**25**	**-35**	**40**
10	**-35**	**-10**	**15**	**-45**	**28**
12	**-45**	**-20**	**5**	**-55**	**17**
14	**-55**	**-30**	**-5**	**-65**	**6**
16	**-65**	**-40**	**-15**	**-76**	**-5**

5.2.2 MAXIMUM RATINGS

While functional operation is not guaranteed beyond the operating ranges listed in Table 5-4, the device may be subjected to the limits specified in Table 5-5 without causing long-term damage.

These conditions must not be imposed on the device for a sustained period—any such sustained imposition may damage the device. Likewise exposure to conditions in excess of the maximum ratings may damage the device.

Table 5-5. Maximum Ratings

PARAMETER	***MIN***	***MAX***	***UNITS***	***NOTES***
Storage Temperature	-40	85	°C	
Supply Voltage (V_{CC})	-0.5	2.0	V	
CMOS I/O Voltage	-0.5	V_{CMOS} +0.5	V	
I/O Voltage	-0.5	V_{TT} +0.5	V	

5.2.3 DC CHARACTERISTICS

Table 5-6. DC Characteristics

PARAMETER	MIN	MAX	UNITS	NOTES
I_{OL} – Low level output current	-9.0		mA	@ V = $V_{OL(max)}$
V_{OH} – High Level Output Voltage		V_{TT}	V	
V_{OL} – Low Level Output Voltage	0	0.4	V	@ I_{ol} = -8mA
I_L – Input Leakage Current		± 100	µA	
I_{LU} – Input Leakage Current for inputs with pull-ups		200	µA	
I_{LD} – Input Leakage Current for inputs with pull-downs		-400	µA	

Table 5-7. CMOS DC Characteristics

PARAMETER	MIN	MAX	UNITS	NOTES
V_{IL} -- Input Low Voltage	-0.58	0.700	V	
$V_{IH1.5}$ – Input High Voltage	V_{REF} + 0.2	V_{TT}	V	(2)
$V_{IH2.5}$ – Input High Voltage	2.0	3.18	V	(3)
V_{OL} – Low Level Output Voltage		0.40	V	@ I_{OL}
V_{OH} – High Level Output Voltage		V_{CMOS}	V	(1)
I_{OL} – Low Level Output Current	9		mA	@ V_{OL}
I_{LI} – Input Leakage Current		±100	µA	
I_{LO} – Output Leakage Current		±100	µA	

Notes:

1. All CMOS signals are open drain.
2. Applies to all CMOS signals except **BCLK**.
3. Applies only to **BCLK**.

5.2.4 POWER DISSIPATION

Table 5-8 gives the core power consumption for the VIA C3 in EBGA at the various operating frequencies and voltages. Note that this does not include the power consumed by the I/O pads.

Table 5-8. Thermal Design Power Information

PARAMETER	VOLTAGE	TDP AVERAGEMAX[1,2]	TDP FACTORYMAX[1,3]	UNITS	NOTES
Normal Mode					
VIA C3 Samuel 2 733 MHz (70C Tcase)	1.60V	10.3	17.9	W	
VIA C3 Samuel 2 733 MHz (70C Tcase)	1.65V	11.1	17.9	W	
VIA C3 Samuel 2 800 MHz (70C Tcase)	1.65V	12.0	19.4	W	4
VIA C3 Nehemiah 1.00 GHz (7.5 x 133 MHz)	1.40V	17.8	18.5	W	
VIA C3 LP 1.00 GHz (7.5 x 133 MHz)	1.25V	11.8	12	W	
StopGrant / AutoHalt Mode					
VIA C3 Samuel 2 733 MHz (70C Tcase)	1.60V	-	4.8	W	
VIA C3 Samuel 2 733 MHz (70C Tcase)	1.65V	-	4.8	W	
VIA C3 Samuel 2 800 MHz (70C Tcase)	1.65V	-	4.8	W	4
VIA C3 Nehemiah 1.00 GHz (7.5 x 133 MHz)	1.40V	5.0	6.75	W	
VIA C3 LP 1.00 GHz (7.5 x 133 MHz)	1.25V	-	-	W	
Sleep Mode					
VIA C3 Samuel 2 733 MHz (70C Tcase)	1.60V	-	4.8	W	
VIA C3 Samuel 2 733 MHz (70C Tcase)	1.60V	-	4.8	W	
VIA C3 Samuel 2 800 MHz (70C Tcase)	1.65V	-	4.8	W	4
VIA C3 Nehemiah 1.00 GHz (7.5 x 133 MHz)	1.40V	5.0	6.75	W	
VIA C3 LP 1.00 GHz (7.5 x 133 MHz)	1.25V	-	-	W	
Deep Sleep Mode					
VIA C3 Samuel 2 733 MHz (70C Tcase)	1.60V	-	4.8	W	
VIA C3 Samuel 2 733 MHz (70C Tcase)	1.65V	-	4.8	W	
VIA C3 Samuel 2 800 MHz (70C Tcase)	1.65V	-	4.8	W	4
VIA C3 Nehemiah 1.00 GHz (7.5 x 133 MHz)	1.40V	3.75	6.75	W	
VIA C3 LP 1.00 GHz (7.5 x 133 MHz)	1.25V	-	-	W	

Notes:

1. Maximum power is generated from running publicly available application software that consumes the most power. Synthetic applications or "thermal virus" applications may consume more power..
2. TDP AverageMax is average value of all parts while running the worst case instruction sequence. Not 100% guaranteed or tested. Consider these power numbers as an average of all parts and some deviation is expected..
3. TDP FactoryMax is the factory limit for power consumption while running the worst case instruction sequence. Factory will reject parts that exceed these specified values.
4. The above power consumption is preliminary and based on 85°C case unless otherwise listed.
5. Conservative thermal solutions must be designed to account for worst-case core and I/O power consumption

Table 5-9. $V_{TT\text{-}I/O}$ Power Consumption

PARAMETER	TYPICAL	MAX	UNITS	NOTES
$P_{TT\text{-}I/O}$ – I/O Operating Power Consumption	0.3	1.2	W	

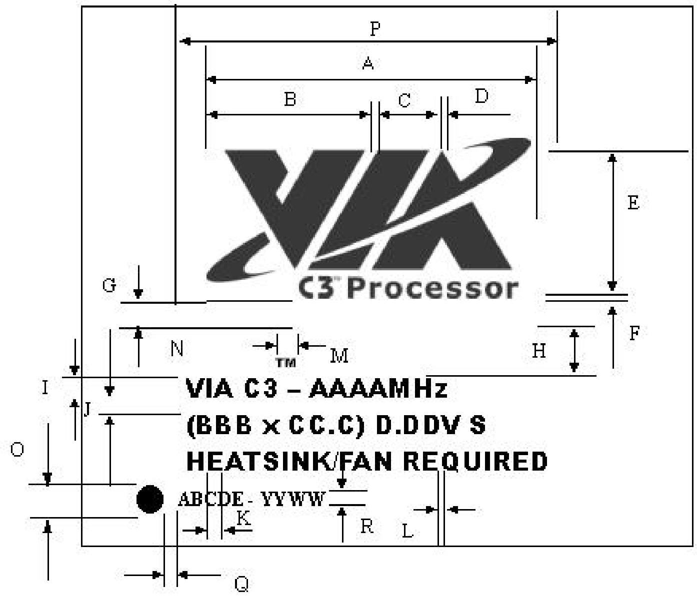
SECTION

6

MECHANICAL SPECIFICATIONS

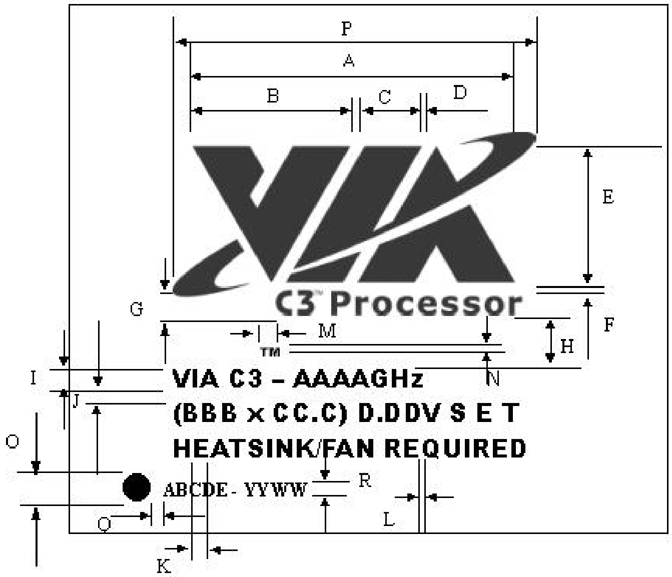
6.1 EBGA PACKAGE

The VIA C3 in EBGA packaged in a unique enhanced ball grid array, 368-ball package, which facilitates compact and economical surface mounting. The VIA C3 in EBGA bus is functionally similar to Socket 370 used by conventional CPGA VIA C3s but is obviously not mechanically compatible.



Symbol Description			
S	Indicates that M/B jumpers should be implemented to adjust the clock-ratio.		
Part Number		**Mechanical Dimensions (mm)**	
A	Wafer Revision	A	18.40
B	Package Method	B	9.18
C	Wafer Vendor	C	3.24
D	Fab#	D	0.81
E	Package Code	E	7.74
Date Code		F	0.50
YY	Year of Assembly	G	1.15
WW	Week of Assembly	H	1.75
Speed Code – core and FSB speed		I	1.28
AAA	866 / 800 / 733 / 700	J	0.65
BBB	133 / 133 / 133 / 100	K	0.78
C.C	6.5 / 6.0 / 5.5 / 7.0	L	0.22
		M	1.54
		N	0.96
		O	1.30
		P	19.7
		Q	0.50
		R	0.80

Figure 6-1. Processor Top Marking – Speeds Below 1.0GHz



Symbol Description	
S	Indicates that M/B jumpers should be implemented to adjust the clock-ratio.
E	Indicates the Tc spec. of 85℃
T	Indicates termination on die

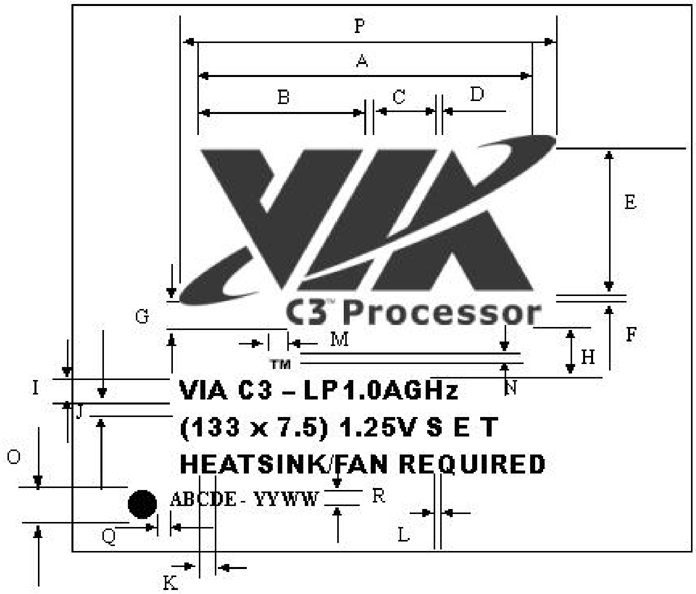
Part Number	
A	Wafer Revision
B	Package Method
C	Wafer Vendor
D	Fab#
E	Package Code

Date Code	
YY	Year of Assembly
WW	Week of Assembly

Speed Code – core and FSB speed			
AAA	BBB	C.C	Description
1.4	100	14.0	1.4AGHz
1.4	133	10.5	1.4AGHz
1.3	133	10.0	1.3AGHz
1.3	100	13.0	1.3AGHz
1.2	133	9.0	1.2AGHz
1.1	100	11.0	1.1AGHz
1.0	100	10.0	1.0AGHz
1.0	133	7.5	1.0AGHz

Mechanical Dimensions (mm)	
A	18.40
B	9.18
C	3.24
D	0.81
E	7.74
F	0.50
G	1.15
H	1.75
I	1.28
J	0.65
K	0.78
L	0.22
M	1.54
N	0.96
O	1.30
P	19.7
Q	0.50
R	0.80

Figure 6-2. Processor Top Marking – Speeds Below 1.0GHz



Symbol Description			
S	Indicates that M/B jumpers should be implemented to adjust the clock-ratio.		
E	Indicates the Tc spec. of 85℃		
T	Indicates termination on die		
LP	Indicates the low power version of VIA C3 1.0GHz		
Part Number		**Mechanical Dimensions (mm)**	
A	Wafer Revision	A	18.40
B	Package Method	B	9.18
C	Wafer Vendor	C	3.24
D	Fab#	D	0.81
E	Package Code	E	7.74
Date Code		F	0.50
YY	Year of Assembly	G	1.15
WW	Week of Assembly	H	1.75
		I	1.28
		J	0.65
		K	0.78
		L	0.22
		M	1.54
		N	0.96
		O	1.30
		P	19.7
		Q	0.50
		R	0.80

Figure 6-3. Processor Top Marking - Low Power C3 1.0GHz

SECTION

7

THERMAL SPECIFICATIONS

7.1 INTRODUCTION

The EBGA VIA C3 is specified for operation with device case temperatures in the range of 0°C to 85°C. Operation outside of this range will result in functional failures and may potentially damage the device.

Care must be taken to ensure that the case temperature remains within the specified range at all times during operation. An effective heat sink with adequate airflow is therefore a requirement during operation.

7.2 TYPICAL ENVIRONMENTS

Typical thermal solutions involve three components: a heat sink, an interface material between the heat sink and the package, and a source of airflow. The best thermal solutions rely on the use of all three components. To the extent that any of these components are not used, the other components must be improved to compensate for such omission. In particular, the use of interface material such as thermal grease, silicone paste, or graphite paper can make a 40°C difference in the case temperature. Likewise, the imposition of airflow is realistically a requirement.

7.3 MEASURING T_C

The case temperature (T_C) should be measured by attaching a thermocouple to the center of the VIA C3 package. The heat produced by the processor is very localized so measuring the case temperature anywhere else will underestimate the case temperature.

The presence of a thermocouple is inherently invasive; effort must be taken to minimize the effect of the measurement. The thermocouple should be attached to the processor through a small hole drilled in the heat sink. Thermal grease should be used to ensure that the thermocouple makes good contact with the package, but the thermocouple should not come in direct contact with the heat sink.

Physical Test Conditions

Case temperature measurements should be made in the worst case operating environments. Ideally, systems should be maximally configured, and tested at the worst-case ambient temperature.

Test Patterns

During normal operation the processor attempts to minimize power consumption. Consequently, normal power consumption is much lower than the maximum power consumption. Thermal testing should be done while running software which causes the processor to operate at its thermal limits.

7.4 MEASURING T_J

The junction temperature of the die can be measured by using the processor's on-chip diode.

7.5 ESTIMATING T_C

The EBGA VIA C3's case temperature can be estimated based on the general characteristics of the thermal environment. This estimate is not intended as a replacement for actual measurement.

Case temperature can be estimated from Table 7-1 below, where,

$T_A \equiv$ Ambient Temperature

$T_C \equiv$ Case Temperature

$\theta_{CA} \equiv$ case-to-ambient thermal resistance

$\theta_{JA} \equiv$ junction-to-ambient thermal resistance

$\theta_{JC} \equiv$ junction-to-case thermal resistance

$P \equiv$ power consumption (Watts)

and,

$T_J = T_C + (P * \theta_{JC})$

$T_A = T_J – (P * \theta_{JA})$

$T_A = T_C – (P * \theta_{CA})$

$\theta_{CA} = \theta_{JA} – \theta_{JC}$

Table 7-1. CPGA θ_{JC} and θ_{JA}

		θ_{JA} (°C/WATT) VS. LAMINAR AIRFLOW (LINEAR FT/MIN)					
Heat Sink in Inches (height)	θ_{JC} (°C/Watt)	0	100	200	400	600	800
0.25	0.9	9.5	8.4	7.0	4.9	4.0	3.4
0.35	0.9	9.2	7.9	6.4	4.4	3.7	3.2
0.45	0.9	8.8	7.4	5.7	4.0	3.3	2.9
0.55	0.9	8.5	6.9	5.1	3.6	3.0	2.7
0.65	0.9	8.1	6.4	4.7	3.4	2.8	2.5
0.80	0.9	7.4	5.7	4.3	3.2	2.6	2.4
1.00	0.9	6.7	5.1	4.0	3.0	2.5	2.2
1.20	0.9	6.3	4.7	3.7	2.8	2.4	2.2
1.40	0.9	5.8	4.3	3.4	2.6	2.3	2.1
No Heat Sink	1.6	14.7	13.4	12.0	9.1	7.7	6.8

Environment: these estimates assume the use of thermal grease between the processor and the heat sink. Heat sinks are 1.95” square.

This page is intentionally left blank.

APPENDIX

A

MACHINE SPECIFIC REGISTERS

A.1 GENERAL

Tables A-1 and A-2 summarize the VIA C3 in EBGA machine-specific registers (MSRs). Further description of each MSR follows the table. MSRs are read using the RDMSR instruction and written using the WRMSR instruction.

There are four basic groups of MSRs (not necessarily with contiguous addresses). Other than as defined below, a reference to an undefined MSR causes a General Protection exception.

1. Generally these registers can have some utility to low-level programs (like BIOS).

   Note that some of the MSRs (address 0 to 0x4FF) have no function in the VIA C3 in EBGA. These MSRs do not cause a GP when used on the VIA C3 in EBGA; instead, reads to these MSRs return zero, and writes are ignored. Some of these undocumented MSRs may have ill side effects when written to indiscriminately. Do not write to undocumented MSRs.
2. There are some undocumented internal-use MSRs used for low-level hardware testing purposes. Attempts to read or write these undocumented MSRs cause unpredictable and disastrous results; so don't use MSRs that are not documented in this datasheet!
3. MSRs used for cache and TLB testing. These use MSR addresses that are not used on compatible processor. These test functions are very low-level and complicated to use. They are not documented in this datasheet but the information will be provided to customers given an appropriate justification

MSRs are not reinitialized by the bus INIT interrupt; the setting of MSRs is preserved across INIT.

Table A-1. Category 1 MSRs

MSR	*MSR NAME*	*ECX*	*EDX*	*EAX*	*TYPE*	*NOTES*

MSR	MSR NAME	ECX	EDX	EAX	TYPE	NOTES
TSC	Time Stamp Counter	10h	TSC[63:32]	TSC[31:0]	RW	
EBL_CR_POWERON	EBL_CR_POWERON	2Ah	n/a	Control bits	RW	
PERFCTR0	Performance counter 0	C1h	TSC[39:32]	TSC[31:0]	RW	1
PERFCTR1	Performance counter 1	C2h	0	Count[31:0]	RW	
BBL_CR_CTL3	L2 Hardware Disabled	11Eh	n/a	00800000h	RO	
EVNTSEL0	Event counter 0 select	186h	n/a	00470079h	RO	1
EVNTSEL1	Event counter 1 select	187h	n/a	Control bits	RW	
MTRR	MTRRphysBase0	200h	Control bits	Control bits	RW	
MTRR	MTRRphysMask0	201h	Control bits	Control bits	RW	
MTRR	MTRRphysBase1	202h	Control bits	Control bits	RW	
MTRR	MTRRphysMask1	203h	Control bits	Control bits	RW	
MTRR	MTRRphysBase2	204h	Control bits	Control bits	RW	
MTRR	MTRRphysMask2	205h	Control bits	Control bits	RW	
MTRR	MTRRphysBase3	206h	Control bits	Control bits	RW	
MTRR	MTRRphysMask3	207h	Control bits	Control bits	RW	
MTRR	MTRRphysBase4	208h	Control bits	Control bits	RW	
MTRR	MTRRphysMask4	209h	Control bits	Control bits	RW	
MTRR	MTRRphysBase5	20Ah	Control bits	Control bits	RW	
MTRR	MTRRphysMask5	20Bh	Control bits	Control bits	RW	
MTRR	MTRRphysBase6	20Ch	Control bits	Control bits	RW	
MTRR	MTRRphysMask6	20Dh	Control bits	Control bits	RW	
MTRR	MTRRphysBase7	20Eh	Control bits	Control bits	RW	
MTRR	MTRRphysMask7	20Fh	Control bits	Control bits	RW	
MTRR	MTRRfix64K_00000	250h	Control bits	Control bits	RW	
MTRR	MTRRfix16K_80000	258h	Control bits	Control bits	RW	
MTRR	MTRRfix16K_A0000	259h	Control bits	Control bits	RW	
MTRR	MTRRfix4K_C0000	268h	Control bits	Control bits	RW	
MTRR	MTRRfix4K_C8000	269h	Control bits	Control bits	RW	
MTRR	MTRRfix4K_D0000	26Ah	Control bits	Control bits	RW	
MTRR	MTRRfix4K_D8000	26Bh	Control bits	Control bits	RW	
MTRR	MTRRfix4K_E0000	26Ch	Control bits	Control bits	RW	
MTRR	MTRRfix4K_E8000	26Dh	Control bits	Control bits	RW	
MTRR	MTRRfix4K_F0000	26Eh	Control bits	Control bits	RW	
MTRR	MTRRfix4K_F8000	26Fh	Control bits	Control bits	RW	
MTRR	MTRRdefType	2FFh	Control bits	Control bits	RW	

Notes:

1. PERFCTR0 is an alias for the lower 40 bits of the Time Stamp Counter. EVNTSEL0 is a read only MSR that reflects this limitation.

Table A-2. Category 2 MSRs

MSR	MSR NAME	ECX	EDX	EAX	TYPE	NOTES
FCR	Feature Control Reg	1107h	n/a	FCR value	RW	
FCR2	Feature Control Reg 2	1108h	FCR2_Hi	FCR2 value	RW	1
FCR3	Feature Control Reg 3	1109h	FCR3_Hi	FCR3 value	WO	1

Notes:

FCR2 and FCR3 provide system software with the ability to specify the Vendor ID string returned by the CPUID instruction.

A.2 CATEGORY 1 MSRS

10H: TSC (TIME STAMP COUNTER)

VIA C3 in EBGA has a 64-bit MSR that materializes the Time Stamp Counter (TSC). System increments the TSC once per processor clock. The TSC is incremented even during AutoHalt or StopClock. A WRMSR to the TSC will clear the upper 32 bits of the TSC.

2AH: EBL_CR_POWERON

31:28	27	26	25:22	21:20	19:18	17:15	14	13	12:0
Res '1100'	*BF High*	LowPow-erEn '1'	BF	*Res*	BSEL	*Res*	1MPOV	IOQDepth	*Reserved (Ignored on write; returns 0 on read)*
4	1	1	4	2	2	3	1	1	13

IOQDepth: 0 = In Order Queue Depth with up to 8 transactions
1 = 1 transaction

1MPOV: 0 = Power on Reset Vector at 0xFFFFFFF0 (4Gbytes)
1 = Power on Reset Vector at 0x000FFFF0 (1 Mbyte)

BSEL: 01 = 133 MHz Bus
10 = 100 MHz Bus

BF_High & BF: Bus Clock Frequency Ratio

BF_High (27)	BF (25:22)	Clock Multiplier
0	0000	5.0
0	0001	3.0
0	0010	4.0
0	0011	10.0
0	0100	5.5
0	0101	3.5
0	0110	4.5
0	0111	9.5
0	1000	9.0
0	1001	7.0
0	1010	8.0
0	1011	6.0
0	1100	12.0
0	1101	7.5
0	1110	8.5
0	1111	6.5
1	0000	Reserved
1	0001	11.0
1	0010	12.0
1	0011	Reserved
1	0100	13.5
1	0101	11.5
1	0110	12.5
1	0111	10.5
1	1000	13.0
1	1001	15.0
1	1010	16.0
1	1011	14.0
1	1100	Reserved
1	1101	15.5
1	1110	Reserved
1	1111	14.5

LowPowerEn: This bit always set to '1'

C1H-C2H: PERFCTR0 & PERFCTR1

These are events counters 0 and 1. VIA C3 in EBGA's PERFCTR0 is an alias for the lower 40 bits of the TSC.

11EH: BBL_CR_CTL3

31:24	23	22:0
Reserved	L2_Hdw_Disable '1'	*Reserved (Ignored on write; returns 0 on read)*
8	1	23

The VIA C3 in EBGA does contain an L2 cache. For compatibility, this read-only MSR indicates to the BIOS or system software that the L2 is disabled even if the L2 is enabled.

L2_Hdw_Disable: This bit always set to '1'

186H: EVNTSEL0 (EVENT COUNTER 0 SELECT)

31:24	23:16	15:9	8:0
Reserved	*Reserved*	*Reserved*	CTR0 Event Select = 79h
8	8	7	9

PERFCTR0 is an alias for the lower 40 bits of the Time Stamp Counter. EVNTSEL0 is a read only MSR which reflects this limitation. The CTR0_Event Select field always returns 0x0079, which corresponds to counting of processor clocks.

187H: EVNTSEL1 (EVENT COUNTER 1 SELECT)

31:24	23:16	15:9	8:0
Reserved	*Reserved*	*Reserved*	CTR1 Event Select
8	8	7	9

VIA C3 in EBGA have two MSRs that contain bits defining the behavior of the two hardware event counters: PERFCTR0 and PERFCTR1.

The CTR1_Event_Select control field defines which of several possible events is counted. The possible Event Select values for PERFCTR1 are listed in the table below. Note that CTR1_Event_Select is a 9-bit field.

The EVNTSEL1 register should be written before PERFCTR1 is written to initialize the counter. The counts are not necessarily perfectly exact; the counters are intended for use over a large number of events and may differ by one or two counts from what might be expected.

Most counter events are internal implementation-dependent debug functions, having no meaning to software. The counters that can have end-user utility are:

EVENT	***DESCRIPTION***
C0h	Instructions executed
1C0h	Instructions executed and string iterations
79h	Internal clocks (default event for CTR0)

A.3 CATEGORY 2 MSRS

1107H: FCR (FEATURE CONTROL REGISTER)

The FCR controls the major optional feature capabilities of the VIA C3 in EBGA. Table A-3 contains the bit values for the FCR. The default settings shown for the FCR bits are not necessarily exact. The actual settings can be changed as part of the manufacturing process and thus a particular VIA C3 in EBGA version can have slightly different default settings than shown here. All reserved bit values of the FCR must be preserved by using a read-modify-write sequence to update the FCR.

Table A-3. FCR Bit Assignments

BIT	***NAME***	***DESCRIPTION***	***DEFAULT***
0	ALTINST	*Reserved for test & special uses*	0
1	ECX8	Enables CPUID reporting CX8	0
2		*Reserved*	0
3		*Reserved*	0
4		*Reserved*	0
5	DSTPCLK	Disables supporting STPCLK	0
6		*Reserved*	0
7	EPGE	Enables CR4.PGE and CPUID.PGE (Page Global Enable)	1
8	DL2	Disables L2 Cache	0
9		*Reserved*	1
10		*Reserved*	0
11	DPDC	Disables Page Directory cache	0
12	EBRPRED	Enables Branch Prediction	1
13	DIC	Disables I-Cache	0
14	DDC	Disables D-Cache	0
15		*Reserved*	1
16		*Reserved*	1
17		*Reserved*	1
18		*Reserved*	0
19		*Reserved*	1
20		*Reserved*	1
21		*Reserved*	1
22:25	SID	Stepping ID	0
26		*Reserved*	0
27		*Reserved*	0
28		*Reserved*	1
29		*Reserved*	0
30		*Reserved*	0
31		*Reserved*	1

ALTINST: 0 = Normal x86 instruction execution.

1 = Alternate instruction set execution is enabled (see details below)

ECX8: 0 **=** The CPUID instruction does not report the presence of the CMPXCHG8B instruction (CX8 = 0)**.** The instruction actually exists and operates correctly, however.

1 = The CPUID instruction reports that the CMPXCHG8B instruction is supported (CX8 = 1).

DSTPCLK: 0 = STPCLK interrupt properly supported.

1 = Ignores SPCLK interrupt.

EPGE: 0 = The processor does not support Page Global Enable and therefore CPUID Feature Flags reports EDX[13]=0; attempts to set CR4.PGE are ignored.

1 = The processor supports Page Global Enable and therefore CPUID Feature Flags reports EDX[13]=1; CR4.PGE can be set to 1.

DL2: 0 = L2 Cache enabled.

1 = L2 Cache disabled.

DPDC: 0 = Enables use of internal Page Directory Cache.

1 = Disables use of internal Page Directory Cache.

EBRPRED: 0 **=** Disables branch prediction function.

1 = Enables branch prediction function.

DIC: 0 = Enables use of I-Cache.

1 = Disables use of I-Cache: cache misses are performed as single transfer bus cycles, PCD is de-asserted. This overrides any setting of CR0.CD and CR0.NW.

DDC: 0 = Enables use of D-Cache.

1 = Disables use of D-Cache: same semantics as for DIC except for D-Cache.

ALTERNATE INSTRUCTION EXECUTION

When set to 1, the ALTINST bit in the FCR enables execution of an alternate (not x86) instruction set. While setting this FCR bit is a privileged operation, executing the alternate instructions can be done from any protection level.

This alternate instruction set includes an extended set of integer, MMX, floating-point, and 3DNow! instructions along with additional registers and some more powerful instruction forms over the x86 instruction architecture. For example, in the alternate instruction set, privileged functions can be used from any protection level, memory descriptor checking can be bypassed, and many x86 exceptions such as alignment check can be bypassed.

This alternate instruction set is intended for testing, debug, and special application usage. Accordingly, it is not documented for general usage. If you have a justified need for access to these instructions, contact your VIA representative.

The mechanism for initiating execution of this alternate set of instructions is as follows:

1. Set the FCR ALTINST bit to 1 using WRMSR instruction (this is a privileged instruction). This should be done using a read-modify-write sequence to preserve the values of other FCR bits.
2. The ALTINST bit enables execution of a new x86 jump instruction that starts execution of alternate instructions. This new jump instruction can be executed from any privilege level at any time that ALTINST is 1. The new jump instruction is a two-byte instruction: 0x0F3F. If ALTINST is 0, the execution of 0x0F3F causes an Invalid Instruction exception.
3. When executed, the new 0x0F3F x86 instruction causes a near branch to CS:EAX. That is, the branch function is the same as the existing x86 instruction
   - jmp [eax]

- In addition to the branch, the 0x0F3F instruction sets the processor into an internal mode where the target bytes are not interpreted as x86 instructions but rather as alternate instruction set instructions.

4. The alternate instructions fetched following the 0x0F3F branch should be of the form
   - 0x8D8400XXXXXXXX where 0xXXXXXXXX is the 32-bit alternate instruction

- That is, the alternate instructions are presented as the 32-bit displacement of a
  - LEA [EAX+EAX+disp]
- instruction. This example assumes that the current code segment size is 32-bits, if it is 16-bits, then an address size prefix (0x67) must be placed in front of the LEA opcode.

5. Upon fetching, the LEA "wrapper" is stripped off and the 32-bit alternate instruction contained in the displacement field is executed.
6. The alternate instruction set contains a special branch instruction that returns control to x86 fetch and execute mode. The x86 state upon return is not necessarily what it was when alternate instruction execution is entered since the alternate instructions can completely modify the x86 state.

While all VIA C3 processor processors contain this alternate instruction feature, the invocation details (e.g., the 0x8D8400 "prefix") may be different between processors. Check the appropriate processor datasheet for details.

1108H: FCR2 (FEATURE CONTROL REGISTER 2)

This MSR contains more feature control bits — many of which are undefined. It is important that all reserved bits are preserved by using a read-modify-write sequence to update the MSR.

63:32
Last 4 characters of Alternate Vendor ID string

31:15	14	13:12	11:8	7:4	3:0
Reserved	AVS	*Res*	Family ID	Model ID	*Res*
17	1	2	4	4	4

AVS: 0 = The CPUID instruction vendor ID is "CentaurHauls"

1 = The CPUID instruction returns the alternate Vendor ID. The first 8 characters of the alternate Vendor ID are stored in FCR3 and the last 4 characters in FCR2[63:32]. These 12 characters are undefined after RESET and may be loaded by system software using WRMSR.

Family ID: This field will be returned as the family ID field by subsequent uses of the CPUID instruction

Model ID: This field will be returned as the model ID field by subsequent uses of the CPUID instruction

1109H: FCR (FEATURE CONTROL REGISTER 3)

This MSR contains the first 8 characters of the alternate Vendor ID. The alternate Vendor ID is returned by the CPUID instruction when FCR2[AVS] is set to '1'. FCR3 is a write-only MSR.

63:32
First 4 characters of Alternate Vendor ID string

31:0
Middle 4 characters of Alternate Vendor ID string