# VIC068A

# VMEbus Interface Controller

## Features

- **Complete VMEbus interface controller and arbiter**
  - **58 internal registers provide configuration control and status of VMEbus and local operations**
  - **Drives arbitration, interrupt, address modifier utility, strobe, address lines A07 through A01 and data lines D07 through D00 directly, and provides signals for control logic to drive remaining address and data lines**
  - **Direct connection to 68xxx family and mappable to non-68xxx processors**
- **Complete master/slave capability**
  - **Supports read, write, write posting, and block transfers**
  - **Accommodates VMEbus timing requirements with internal digital delay line ($^1/_2$-clock granularity)**
  - **Programmable metastability delay**
  - **Programmable data acquisition delays**
  - **Provides timeout timers for local bus and VMEbus transactions**
- **Interleaved block transfers over VMEbus**
  - **Acts as DMA master on local bus**
  - **Programmable burst count, transfer length, and interleaved period interval**
  - **Supports local module-based DMA**
- **Arbitration support**
  - **Supports single-level, priority and round robin arbitration**
  - **Supports fair request option as requester**
- **Interrupt support**
  - **Complete support for the VMEbus interrupts: interrupter and interrupt handler**
  - **Seven local interrupt lines**
  - **8-level interrupt priority encode**
  - **Total of 29 interrupts mapped through the VIC068A**
- **Miscellaneous features**
  - **Refresh option for local DRAM**
  - **Four broadcast location monitors**
  - **Four module-specific location monitors**
  - **Eight interprocessor communications registers**
  - **PGA or QFP packages**
  - **Compatible with IEEE Specification 1014, Rev. C**
  - **Supports RMC operations**
- **See the *VMEbus Interface Handbook* for more information**

## Functional Description

The VMEbus interface controller (VIC068A) is a single chip designed to minimize the cost and board area requirements and to maximize performance of the VMEbus interface of a VMEbus master/slave module. This can be implemented on VIC068A either an 8-bit, 16-bit, or 32-bit VMEbus system. The VIC068A performs all VMEbus system controller functions plus many others, which simplify the development of VIC068Aa VMEbus interface. The VIC068A utilizes patented on-chip output buffers. These CMOS high-drive buffers provide direct connection to the address and data lines. In addition to these signals, the VIC068A connects directly to the arbitration, interrupt, address modifier, utility and strobe lines. Signals are provided which control data direction and latch functions needed for a 32-bit implementation.

The VIC068A was developed through the efforts of a consortium of board vendors, under the auspices of the VMEbus International Trade Association (VITA). The VIC068A thus insures compatibility between boards designed by different manufacturers.

## Pin Configurations

**Pin Grid Array (PGA)**
**Bottom View**

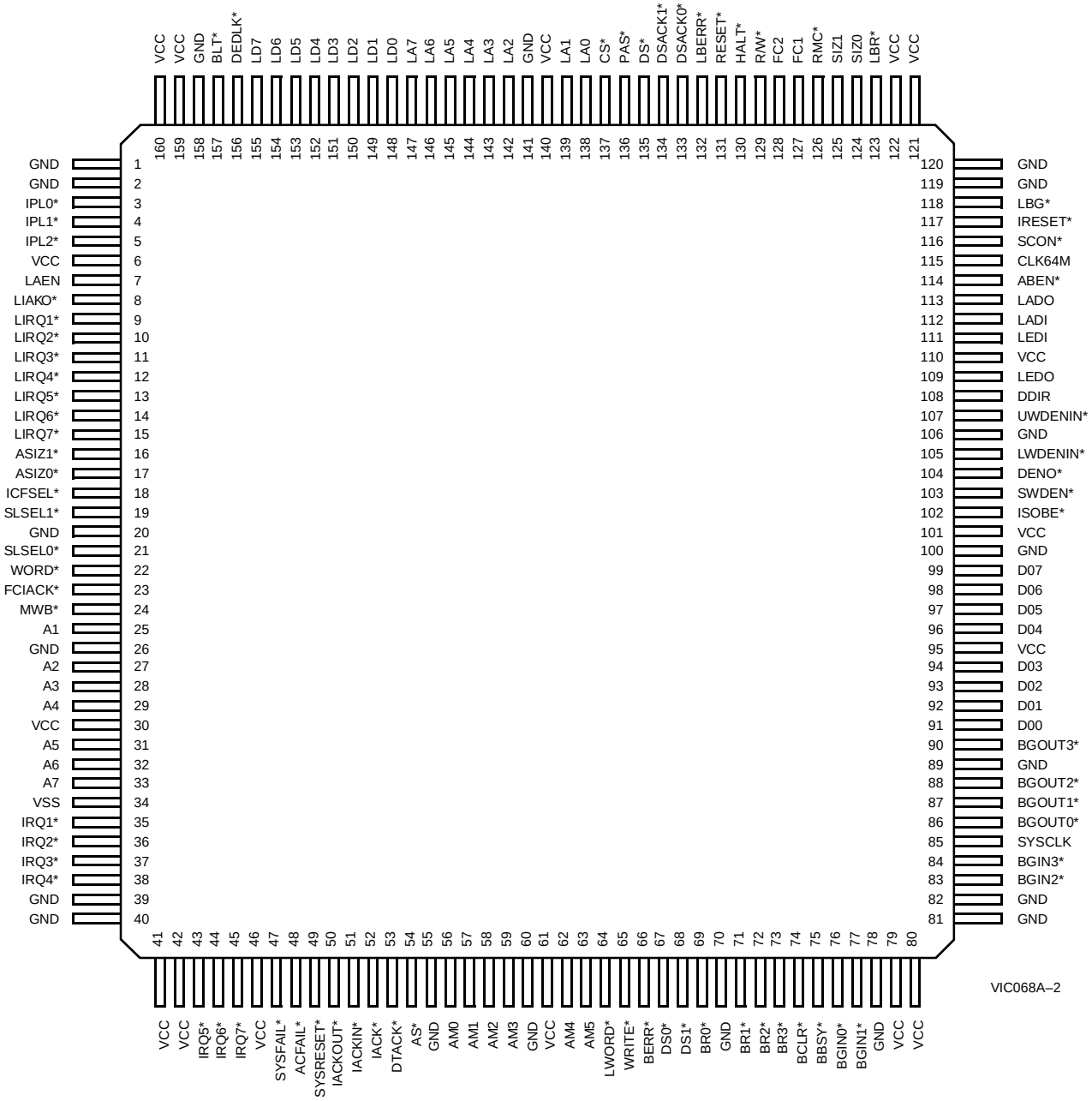
| A | B | C | D | E | F | G | H | J | K | L | M | N | P | R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GND | IPL2* | LIACKO* | LIRQ2* | LIRQ5* | ASIZ1 | ASIZ0 | SLSEL1* | WORD* | FIACK* | A02 | A04 | VCC | GND | IRQ4* |
| LD6 | BLT* | IPL1* | VCC | LIRQ1* | LIRQ4* | LIRQ6* | ICFSEL* | MWB* | A01 | A03 | A05 | A07 | IRQ3* | IRQ7* |
| LD2 | LD5 | DEDLK* | IPL0* | LAEN | LIRQ3* | LIRQ7* | GND | SLSEL0* | GND | A06 | IRQ1* | IRQ2* | IRQ6* | ACFAIL* |
| LD1 | LD3 | LD7 | LOCATOR PIN | | | | | | | | | IRQ5* | VCC | IACKOUT* |
| LA7 | LD0 | LD4 | | | | | | | | | | SYSFAIL* | SYSRESET* | DTACK* |
| LA3 | LA5 | LA6 | | | | | | | | | | IACKIN* | IACK* | AM0 |
| LA2 | LA4 | GND | | | | | | | | | | GND | AS* | AM1 |
| LA1 | LA0 | VCC | | | | | | | | | | GND | AM2 | AM3 |
| CS* | DSACK1* | DS* | | | | | | | | | | VCC | LWORD* | AM4 |
| PAS* | LBERR* | RESET* | | | | | | | | | | BERR* | WRITE* | AM5 |
| DSACK0* | R/W* | FC1 | | | | | | | | | | BR2* | DS1* | DS0* |
| HALT* | RMC* | LBR* | | | | | | | | | | BBSY* | BR1* | BR0* |
| FC2 | SIZ0 | SCON* | CLK64M | LADI | GND | VCC | GND | VCC | D00 | BGOUT1* | BGIN2* | BGIN0* | BR3* | GND |
| SIZ1 | IRESET* | LADO | LEDI | DDIR | LWDENIN* | DENO* | D06 | D03 | D01 | GND | BGOUT0* | BGIN3* | BGIN1* | BCLR* |
| LBG* | ABEN* | VCC | LEDO | UWDENIN* | SWDEN* | ISOBE* | D07 | D05 | D04 | D02 | BGOUT3* | BGOUT2* | SYSCLK | GND |

VIC068A–1

## Pin Configurations (continued)

**160-Pin Quad Flatpack (QFP)**

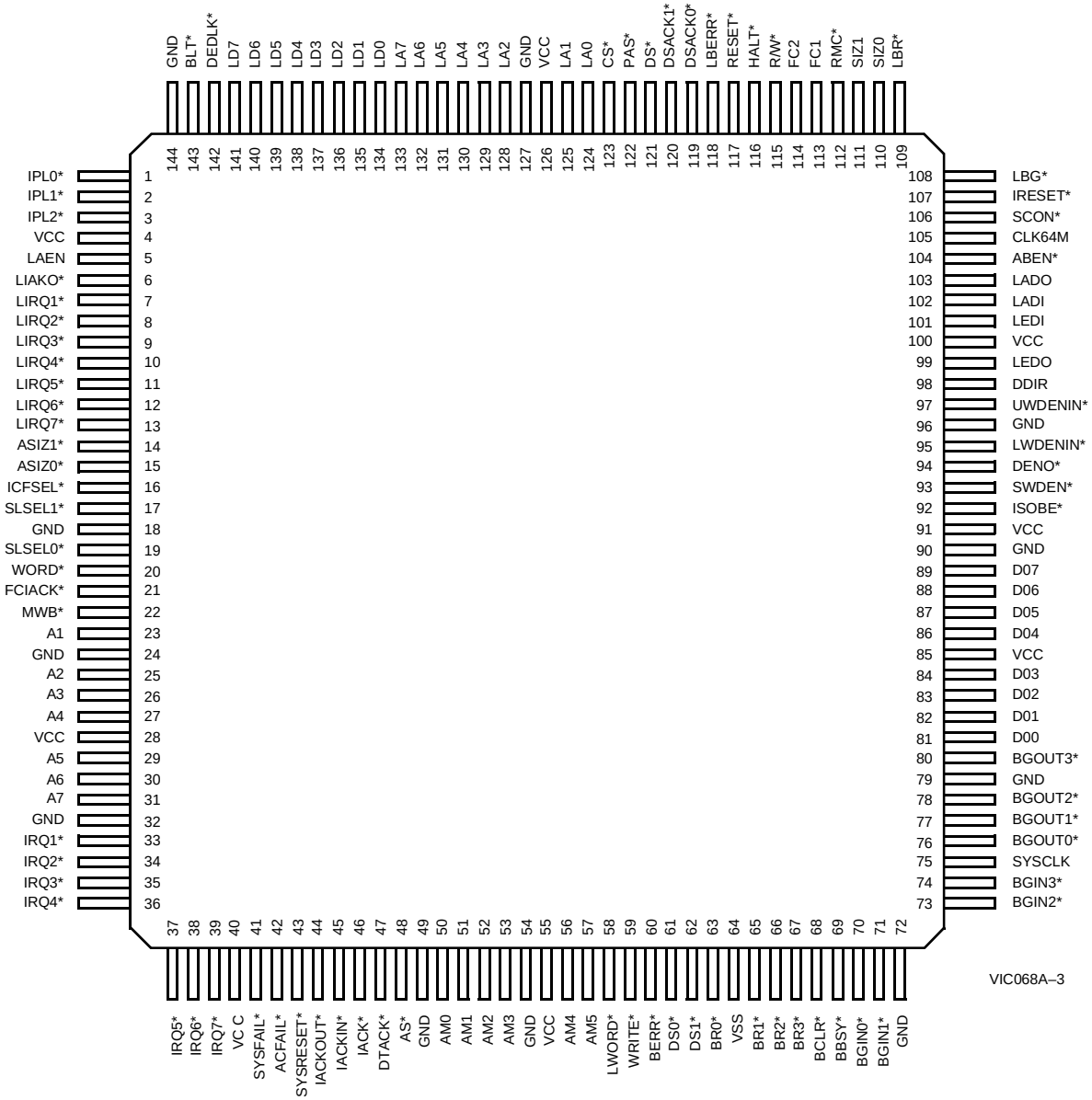
**Top View**




VIC068A–2

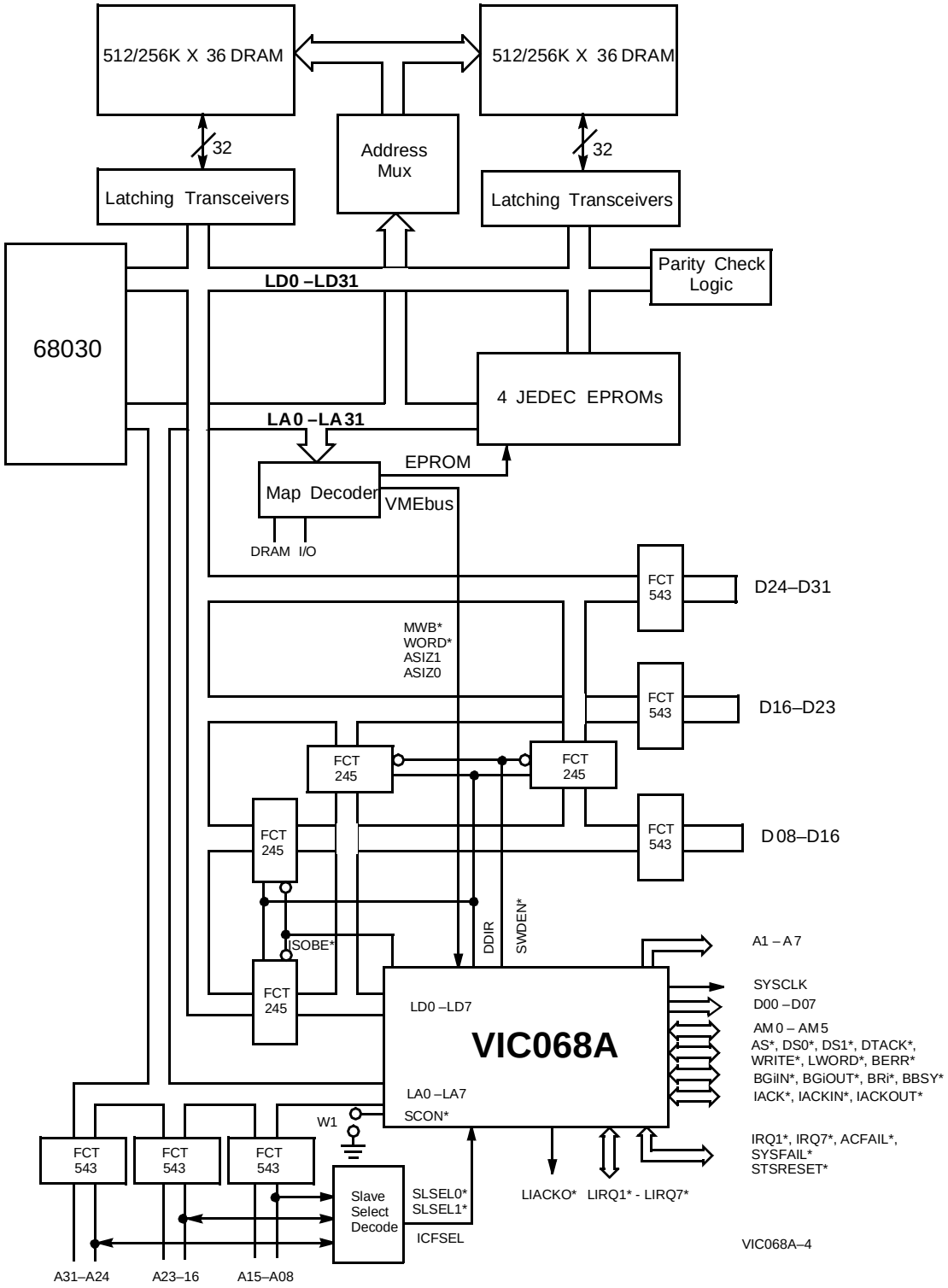
## Pin Configurations (continued)

**144-Pin Thin Quad Flatpack (TQFP)**
**Top View**




VIC068A–3

## VIC068A on 68030 Board



## Theory of Operation

The VIC068A is an interface between a local CPU bus and the VMEbus. The local bus interface of the VIC068A emulates Motorola's family of 32-bit CISC processor interfaces. Other processors can easily be adapted to interface to the VIC068A using the appropriate logic.

### Resetting the VIC068A

The VIC068A can be reset by any of three distinct reset conditions:

**Internal Reset.** This reset is the most common means of resetting the VIC068A. It resets select register values and all logic within the device.

**System Reset.** This reset provides a means of resetting the VIC068A through the VMEbus backplane. The VIC068A may also signal a SYSRESET* by writing a configuration register.

**Global Reset.** This provides a complete reset of the VIC068A. This reset resets all of the VIC068A's configuration registers. This reset should be used with caution since SYSCLK is not driven while a global reset is in progress.

All three reset options are implemented in a different manner and have different effects on the VIC068A configuration registers.

### VIC068A VMEbus System Controller

The VIC068A is capable of operating as the VMEbus system controller. It provides VMEbus arbitration functions, including:

- Priority, round-robin, and single-level arbitration schemes
- Driving IACK* Daisy-Chain
- Driving BGiOUT* Daisy-Chain (All four levels)
- Driving SYSCLK output
- VMEbus arbitration timeout timer

The System controller functions are enabled by the SCON* pin of the VIC068A. When strapped LOW, the VIC068A functions as the VMEbus system controller.

### VIC068A VMEbus Master Cycles

The VIC068A is capable of becoming the VMEbus master in response to a request from local resources. In this situation, the local resource requests that a VMEbus transfer is desired. The VIC068A makes a request for the VMEbus. When the VMEbus is granted to the VIC068A, it then performs the transfer and acknowledges the local resource and the cycle is complete. The VIC068A is capable of all four VMEbus request levels. The following release modes are supported:

- Release on request (ROR)
- Release when done (RWD)
- Release on clear (ROC)
- Release under RMC* control
- Bus capture and hold (BCAP)

The VIC068A supports A32, A24, and A16, as well as user-defined address spaces.

### Master Write-Posting

The VIC068A is capable of performing master write-posting (bus decoupling). In this situation, the VIC068A acknowledges the local resource *immediately* after the request to the VIC068A is made, thus freeing the local bus. The VIC068A latches the local data to be written and performs the VMEbus transfer without the local resource having to wait for VMEbus arbitration.

### Indivisible Cycles

Read-modify-write cycles and indivisible multiple-address cycles (EMACS) are easily performed using the VIC068A. Significant control is allowed to:

- Requesting the VMEbus on the assertion of RMC* independent of MWB* (this prevents any slave access from interrupting local indivisible cycles)
- Stretching the VMEbus AS*
- Making the above behaviors dependent on the local SIZi signals

### Deadlock Condition

If a master operation is attempted when a slave operation to the same module is in progress, a deadlock condition has occurred. The VIC068A will signal a deadlock condition by asserting the DEDLK* signal. This should be used by the local resource requesting the VMEbus to try the transfer after the slave access has completed.

### Self-Access Condition

If the VIC068A, while it is VMEbus master, has a slave select signaled, a self access is said to have occurred. The VIC068A will issue a BERR*, which in turn will cause a LBERR* to be asserted.

### VIC068A VMEbus Slave Cycles

The VIC068A is capable of operating as a VMEbus slave controller. The VIC068A contains a highly programmable environment to allow for a wide variety of slave configurations. The VIC068A allows for:

- D32, D16, or D8 configuration
- A32, A24, A16, or user-defined address spaces
- Programmable block transfer support including:
  - — DMA-type block transfer (PAS* and DSACKi* held asserted)
  - — non-DMA-type block transfer (toggle PAS* and DSACKi*)
  - — No support for block transfer
- Programmable data acquisition delays
- Programmable PAS* and DS* timing
- Restricted slave accesses (supervisory accesses only)

When a slave access is required, the VIC068A will request the local bus. When local bus mastership is obtained, the VIC068A will read or write the data to/from the local resource and assert the DTACK* signal to complete the transfer.

### Slave Write-Posting

The VIC068A is capable of performing a slave write-post operation (bus decoupling). When enabled, the VIC068A latches the data to be written and acknowledge the VMEbus (asserts DTACK*) immediately thereafter. This prevents the VMEbus from having to wait for local bus access.

### Address Modifier (AM) Codes

The VIC068A encodes and decodes the VMEbus address modifier codes. For VMEbus master accesses, the VIC068A encodes the appropriate AM codes through the VIC068A FCi and ASIZi signals, as well as the block transfer status. For

slave accesses, the VIC068A decodes the AM codes and checks the slave select control registers to see if the slave request is to be supported with regard to address spaces, supervisory accesses, and block transfers. The VIC068A also supports user-defined AM codes; that is, the VIC068A can be made to assert and respond to user-defined AM codes.

### VIC068A VMEbus Block Transfers

The VIC068A is capable of both master and slave block transfers. The master VIC068A performs a block transfer in one of two modes:

- MOVEM-type Block Transfer
- Master Block Transfer with Local DMA

In addition to these VMEbus block transfers, the VIC068A is also capable of performing block transfers from one local resource to another in a DMA-like fashion. This is referred to as a Module-based DMA transfer.

The VMEbus specification restricts block transfers from crossing 256-byte boundaries without toggling the address strobe, in addition to restricting the maximum length of the transfer to 256 bytes. The VIC068A allows for easy implementation of block transfers that exceed the 256-byte restriction by releasing the VMEbus at the appropriate time and rearbitrating for the bus at a programmed time later (this in-between time is referred to as the interleave period), while at the same time holding both the local and VMEbus addresses with internal latches. All of this is performed without processor/software intervention until the transfer is complete.

The VIC068A contains two separate address counters for the VMEbus and the local address buses. In addition, a separate address is counter-provided for slave block transfers. The VIC068A address counters are 8-bit up-counters that provide for transfers up to 256 bytes. For transfers that exceed the 256-byte limit, the Cypress CY7C964 or external counters and latches are required.

The VIC068A allows slave accesses to occur during the interleave period. Master accesses are also allowed during interleave with programming and external logic. This is referred to as the “dual path” option.

### MOVEM Master Block Transfer

This mode of block transfer provides the simplest implementation of VMEbus block transfers. For this mode, the local resource simply configures the VIC068A for a MOVEM block transfer and proceeds with the consecutive-address cycles (such as a 680X0 MOVEM instruction). The local resource continues as the local bus master in this mode.

### Master Block Transfers with Local DMA

In this mode, the VIC068A becomes the local bus master and reads or writes the local data in a DMA-like fashion. This provides a much faster interface than the MOVEM block transfer, but with less control and fault tolerance.

### VIC068A Slave Block Transfer

The process of receiving a block transfer is referred to as a slave block transfer. The VIC068A is capable of decoding the address modifier codes to determine that a slave block transfer is desired. In this mode, the VIC068A captures the VMEbus address, and latches them into internal counters. For subsequent cycles, the VIC068A simply increments this counter for each transfer. The local protocol for slave block transfers can be configured in a full handshake mode by toggling both PAS* and DS* and expecting DSACKi* to toggle, or in an accelerated mode in which only DS* toggles and PAS* is asserted throughout the cycle.

### Module-Based DMA Transfers

The VIC068A is capable of acting as a DMA controller between two local resources. This mode is similar to that of master block transfers with local DMA, with the exception that the VMEbus is not the second source or destination.

### VIC068A Interrupt Generation and Handling Facilities

The VIC068A is capable of generating and handling a seven-level prioritized interrupt scheme similar to that used by the Motorola CISC processors. These interrupts include the seven VMEbus interrupts, seven local interrupts, five VIC068A error/status interrupts, and eight interprocessor communication interrupts.

The VIC068A can be configured to act as handler for any of the seven VMEbus interrupts. The VIC068A can generate the seven VMEbus interrupts as well as supplying a user-defined status/ID vector. The local priority level (IPL) for VMEbus interrupts is programmable. When configured as the system controller, the VIC068 will drive the IACK daisy-chain.

The local interrupts can be configured with the following:

- User-defined local interrupt priority level (IPL)
- Option for VIC068A to provide the status/ID vector
- Edge or level sensitivity
- Polarity (rising/falling edge, active HIGH/LOW)

The VIC068A is also capable of generating local interrupts on certain error or status conditions. These include:

- ACFAIL* asserted
- SYSFAIL* asserted
- Failed master write-post (BERR* asserted)
- Local DMA completion for block transfers
- Arbitration timeout
- VMEbus interrupter interrupt

The VIC068A can also interrupt on the setting of a module or global switch in the interprocessor communication facilities.

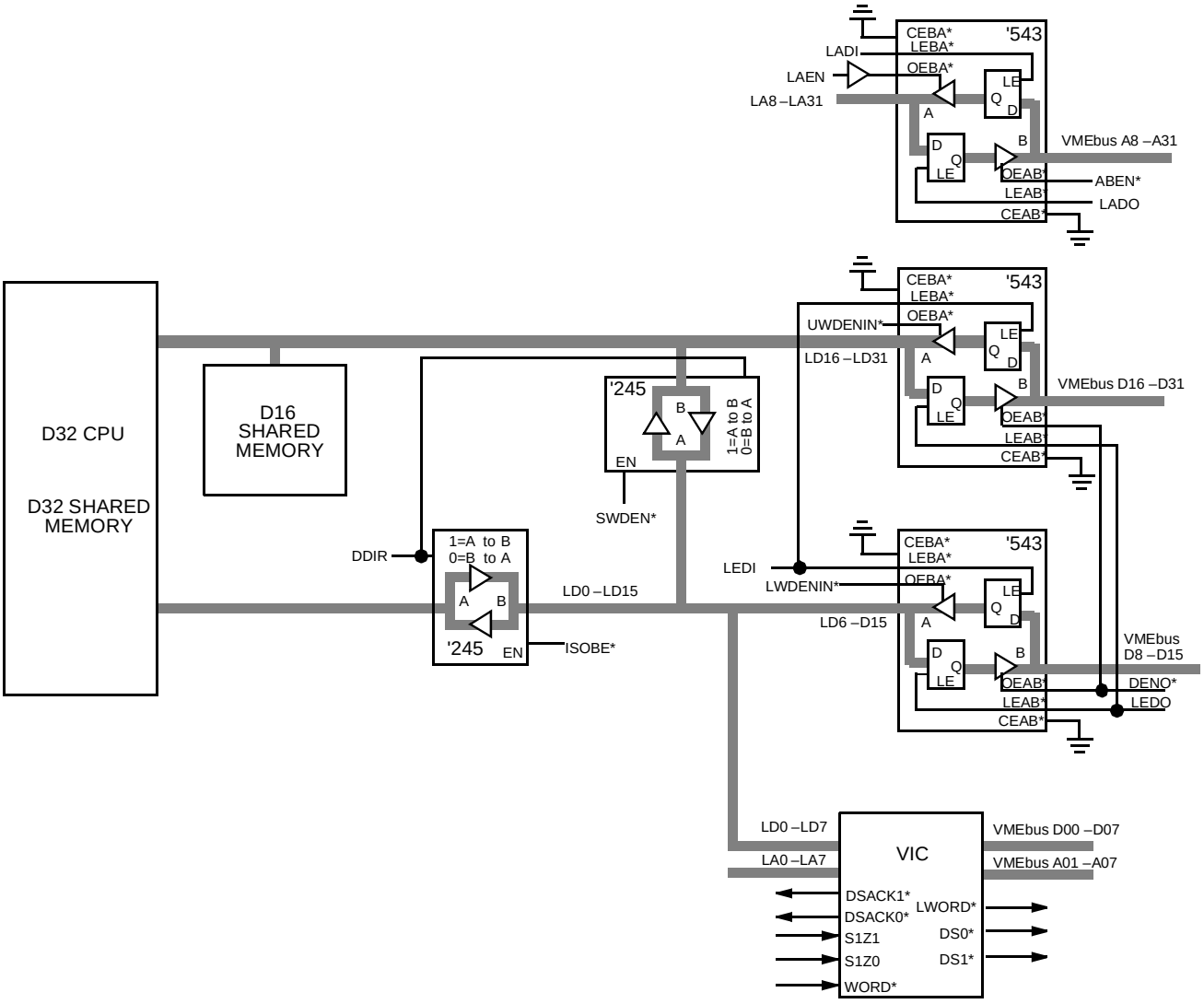
### Interprocessor Communication Facilities

The VIC068A includes interprocessor registers and switches that can be written and read through VMEbus accesses. These are the only such registers that are directly accessible from the VMEbus. Included in the interprocessor communication facilities are:

- Four general purpose 8-bit registers
- Four module switches
- Four global switches
- VIC068A version/revision register (read-only)
- VIC068A Reset/Halt condition (read-only)
- VIC068A interprocessor communication register semaphores

When set through a VMEbus access, these switches can interrupt a local resource. The VIC068A includes module switches that are intended for a single module, and global switches which are intended to be used as a broadcast.

## Buffer Control Signal for Shared Memory Implementation[1]




**Note:**
1. This configuration can support Slave Block Transfers and Master and Slave Write-Post Operation. This buffer configuration cannot support block transfers with DMA.

## Operating Range

| Range | Ambient Temperature | $V_{CC}$ |
|---|---|---|
| Commercial | 0°C to +70°C | 5V ± 5% |
| Industrial | –40°C to +85°C | 5V ± 10% |
| Military | –55°C to +125°C | 5V ± 10% |

## Related Documents

*VMEbus Interface Handbook*

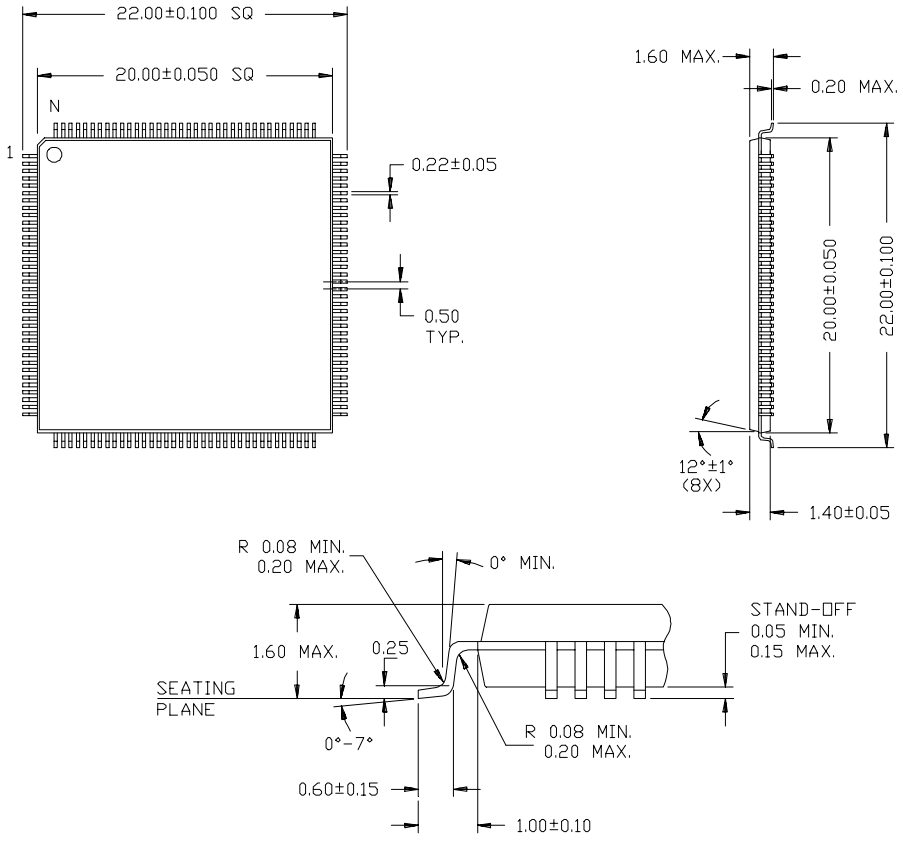
## Ordering Information

| Ordering Code | Package Name | Package Type | Operating Range |
|---|---|---|---|
| VIC068A-AC | A144 | 144-Pin Thin Quad Flatpack | Commercial |
| VIC068A-BC | B144 | 145-Pin Plastic Pin Grid Array | |
| VIC068A-GC | G145 | 145-Pin Ceramic Pin Grid Array | |
| VIC068A-NC | N160 | 160-Lead Plastic Quad Flatpack | |
| VIC068A-GI | G145 | 145-Pin Ceramic Pin Grid Array | Industrial |
| VIC068A-GMB | G145 | 145-Pin Ceramic Pin Grid Array | MIL-STD-883 |
| VIC068A-UM | U162 | 160-Lead Ceramic Quad Flatpack | Military Temp. Commercial |
| VIC068A-UMB | U162 | 160-Lead Ceramic Quad Flatpack | MIL-STD-883 |

Document #: 38-00167-C

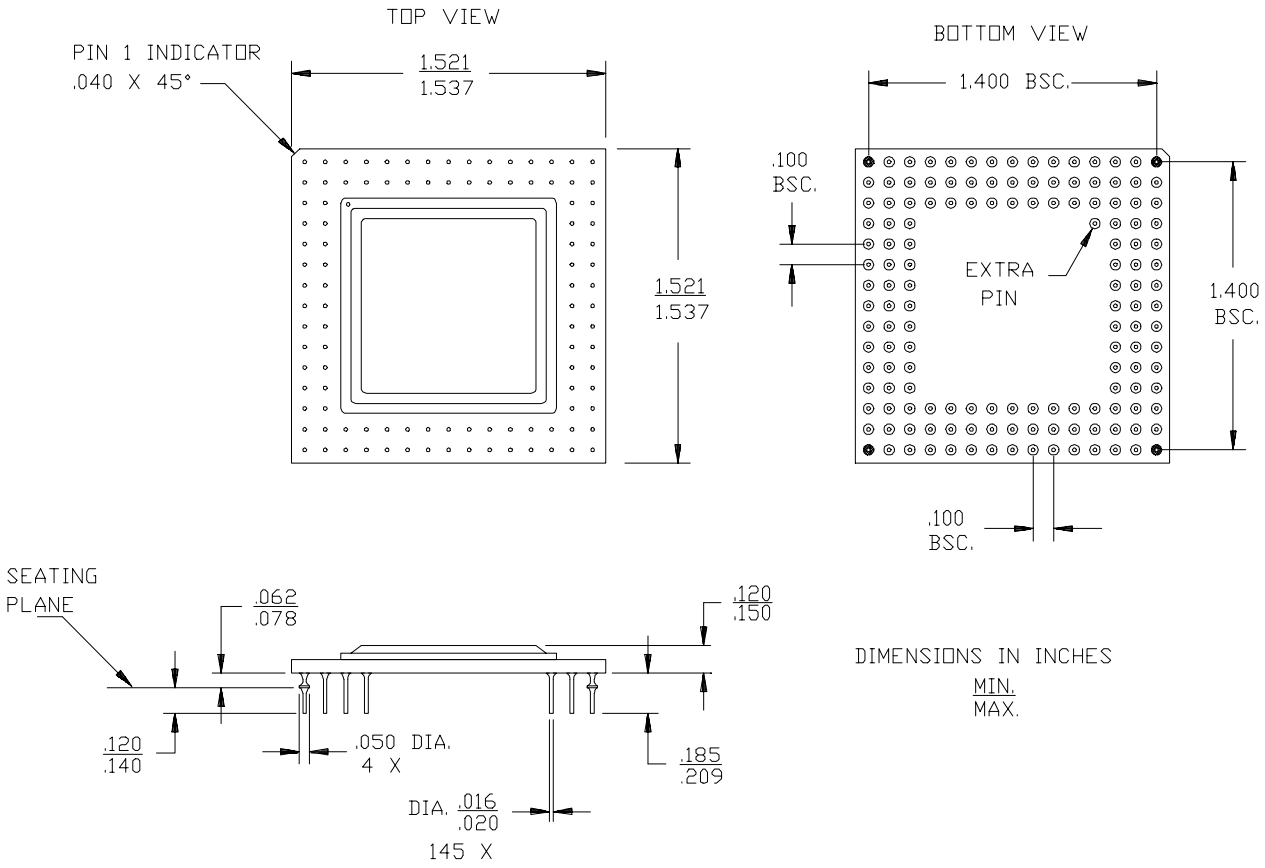
## Package Diagrams

**144-Pin Thin Quad Flat Pack A144**



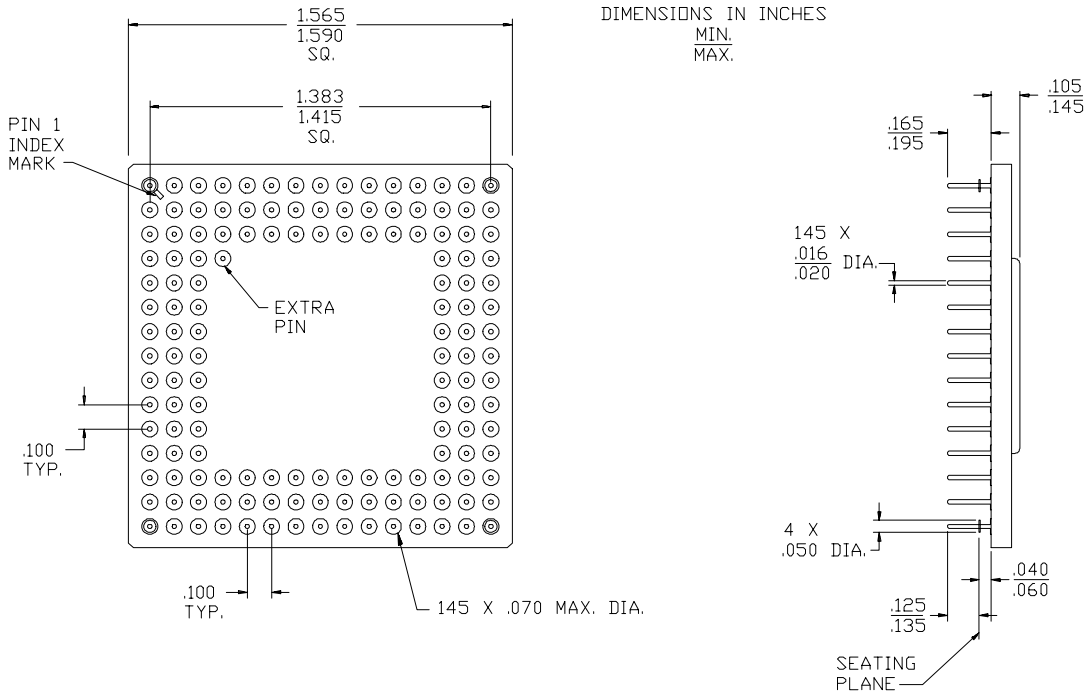
**Package Diagrams** (continued)

**145-Pin Plastic Grid Array (CavityUp) B144**



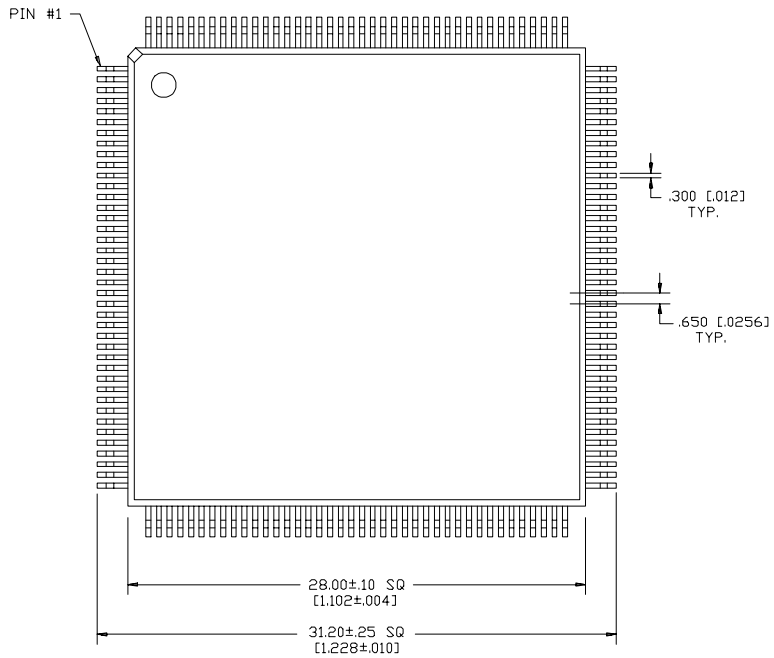
## Package Diagrams (continued)

**145-Pin Grid Array (Cavity Up) G145**

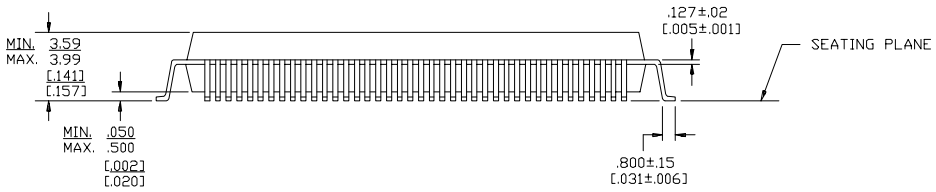


## Package Diagrams (continued)

**160-Lead Plastic Quad Flatpack N160**




DIMENSION IN mm [ INCHES as reference only]
LEAD COPLANARITY .100 [.004]




© Cypress Semiconductor Corporation, 1997. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress Semiconductor product. Nor does it convey or imply any license under patent or other rights. Cypress Semiconductor does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress Semiconductor products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress Semiconductor against all charges.

## Package Diagrams (continued)

**160-Lead Ceramic Quad Flatpack U162**

