VIC64

VMEbus Interface Controller with D64 Functionality

Features

- **An enhanced VIC068A**
  - **64-bit MBLT operation**
  - **Higher transfer rate**
- **Complete VMEbus interface controller and arbiter**
  - **58 internal registers for configuration control and VMEbus and local operations status**
  - **Drives arbitration, interrupt, address modifier, utility, strobe, address line A[7:1], and data line D[7:0] directly and provides control signals to drive remaining address and data lines**
  - **Direct connection to 68K family and mappable to non-68K processors**
- **Complete master/slave capability**
  - **Supports read, write, write posting, and block transfers**
  - **Accommodates VMEbus timing requirements with internal digital delay line with half-clock granularity**
  - **Programmable metastability delay**
  - **Programmable data acquisition delays**
  - **Provides programmable timeout timers for local bus and VMEbus transactions**
- **Interleaved block transfers**
  - **D64 block transfer capability in conformance with VME64 proposal**
  - **Can act as DMA master on local bus**
  - **Programmable burst counter, transfer length, and interleave period**
  - **Allows master and slave transfer to occur during interleave period**
  - **Also supports local module-based DMA**
- **Arbitration support**
  - **Supports single-level, priority, and round-robin arbitration**
  - **Support fair request option as requester**
- **Interrupt support**
  - **Complete support for the VMEbus interrupts; interrupters and interrupt handler**
  - **Seven local interrupt lines**
  - **8-level interrupt priority encoded**
  - **Total of 29 interrupts mapped through the VIC64**
- **Miscellaneous features**
  - **Refresh option for local DRAM**
  - **Four broadcast location monitors**
  - **Four module-specific location monitors**
  - **Eight interprocessor communication registers**
- **See the *VMEbus Interface Handbook* for more information**

Functional Description

Cypress's VIC64 VMEbus Interface Controller with D64 functionality is a single chip designed to minimize the cost and board area requirements and to maximize the performance of a VMEbus master/slave module. Data transfers of 70 Mbyte/sec are possible between boards using VIC64.

In addition to D8, D16, and D32 operations, the VIC64 performs D64 data transfer. On-chip output buffers are used to provide direct connection to address and data lines.

The VIC64 is based on the industry-standard VIC068A. For most applications, the VIC64 is fully software and plug compatible with the VIC068A. (As VIC64 uses register bits that are unassigned in VIC068A, user code may require simple rework to insure compatibility.)

The local bus interface of the VIC64 emulates Motorola's family of 32-bit 68K processor interfaces. Other processors can easily be adapted to interface to the VIC64 using appropriate logic.

Resetting the VIC64

The VIC64 can be reset by any of three distinct reset conditions:

- Internal Reset. This reset is the most common means of resetting the VIC64. It resets selected register values and logic within the device.
- System Reset. This reset provides a means of resetting the VIC64 through the VMEbus backplane. The VIC64 may also initiate a system reset by writing a configuration register.
- Global Reset. This provides the most complete reset of the VIC64. It resets all of the VIC64's configuration registers.

All three reset options are implemented in a different manner and have different effect on the VIC64 configuration registers.

VIC64 VMEbus System Controller

The VIC64 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 VIC64. This pin is sampled during Reset and if LOW, VIC64 performs as system controller. After Reset the pin becomes an output signifying a D64 transfer.

VIC64 VMEbus Master Cycles

The VIC64 is capable of becoming the VMEbus master in response to a request from local resources. In this situation, the local resource requests a VMEbus transfer. The VIC64 makes a request for the VMEbus. When the VMEbus is granted to the VIC64, it then performs the transfer and acknowledges the lo-

cal resource and the cycle is complete. The VIC64 is capable of all four VMEbus request levels. In addition, 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)

Pin Configurations

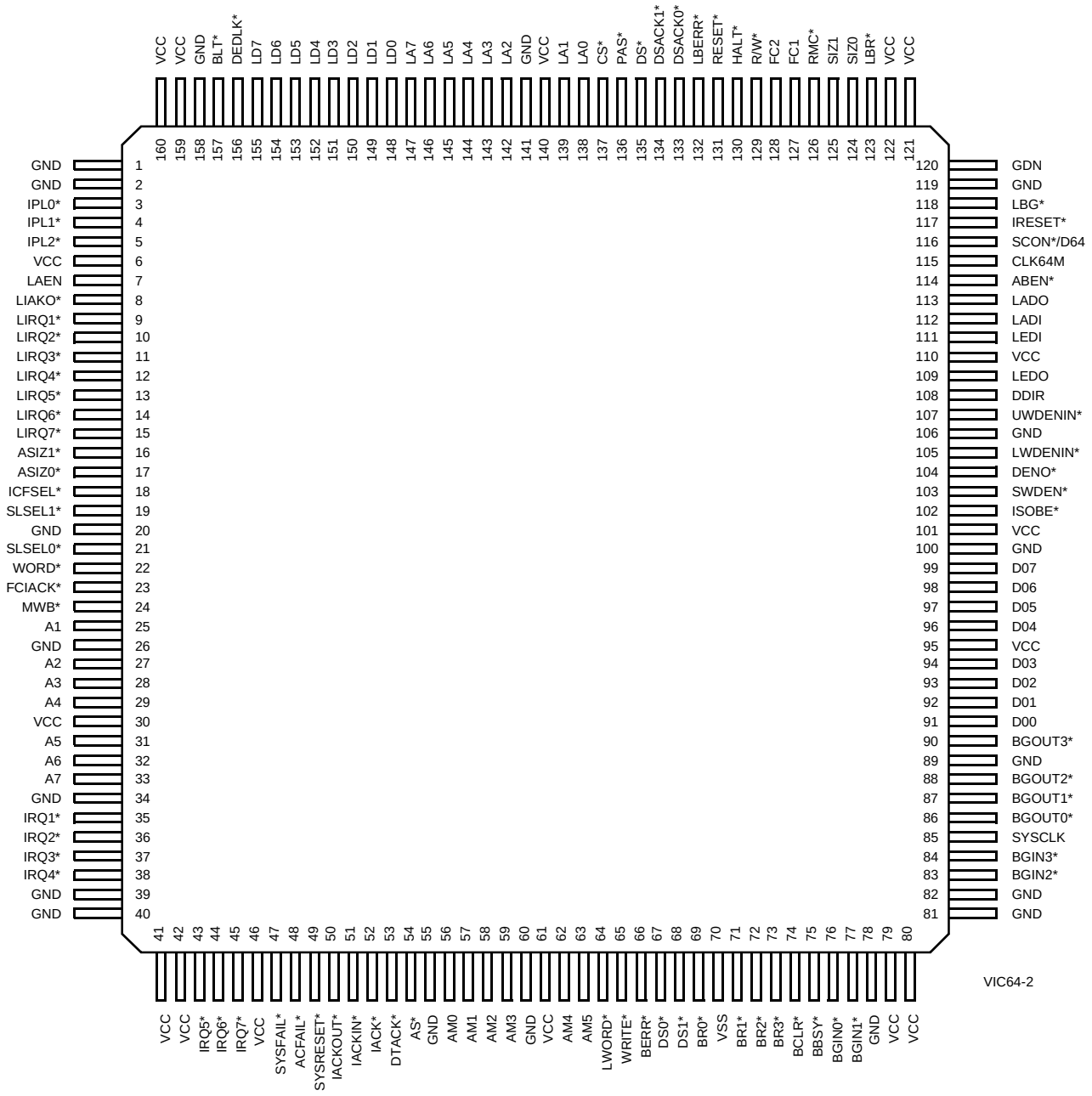
Pin Grid Array (PGA)
Bottom View

A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	
GND	IPL2*	LIACKO*	LIRQ2*	LIRQ5*	ASIZ1	ASIZ0	SLSEL1*	WORD*	FCIACK*	A02	A04	VCC	GND	IRQ4*	1
LD6	BLT*	IPL1*	VCC	LIRQ1*	LIRQ4*	LIRQ6*	ICFSEL*	MWB*	A01	A03	A05	A07	IRQ3*	IRQ7*	2
LD2	LD5	DEDLK*	IPL0*	LAEN	LIRQ3*	LIRQ7*	GND	SLSEL0*	GND	A06	IRQ1*	IRQ2*	IRQ6*	ACFAIL*	3
LD1	LD3	LD7	LOCATOR PIN									IRQ5*	VCC	IACKOUT*	4
LA7	LD0	LD4										SYSFAIL*	SYSRESET*	DTACK*	5
LA3	LA5	LA6										IACKIN*	IACK*	AM0	6
LA2	LA4	GND										GND	AS*	AM1	7
LA1	LA0	VCC										GND	AM2	AM3	8
CS*	DSACK1*	DS*										VCC	LWORD*	AM4	9
PAS*	LBERR*	RESET*										BERR*	WRITE*	AM5	10
DSACK0*	R/W*	FC1										BR2*	DS1*	DS0*	11
HALT*	RMC*	LBR*										BBSY*	BR1*	BR0*	12
FC2	SIZ0	SCON*/D64	CLK64M	LADI	GND	VCC	GND	VCC	D00	BGOUT1*	BGIN2*	BGIN0*	BR3*	GND	13
SIZ1	IRESET*	LADO	LEDI	DDIR	LWDENIN*	DENO*	D06	D03	D01	GND	BGOUT0*	BGIN3*	BGIN1*	BCLR*	14
LBG*	ABEN*	VCC	LEDO	UWDENIN*	SWDEN*	ISOBE*	D07	D05	D04	D02	BGOUT3*	BGOUT2*	SYSCLK	GND	15

VIC64-1

Pin Configurations (continued)

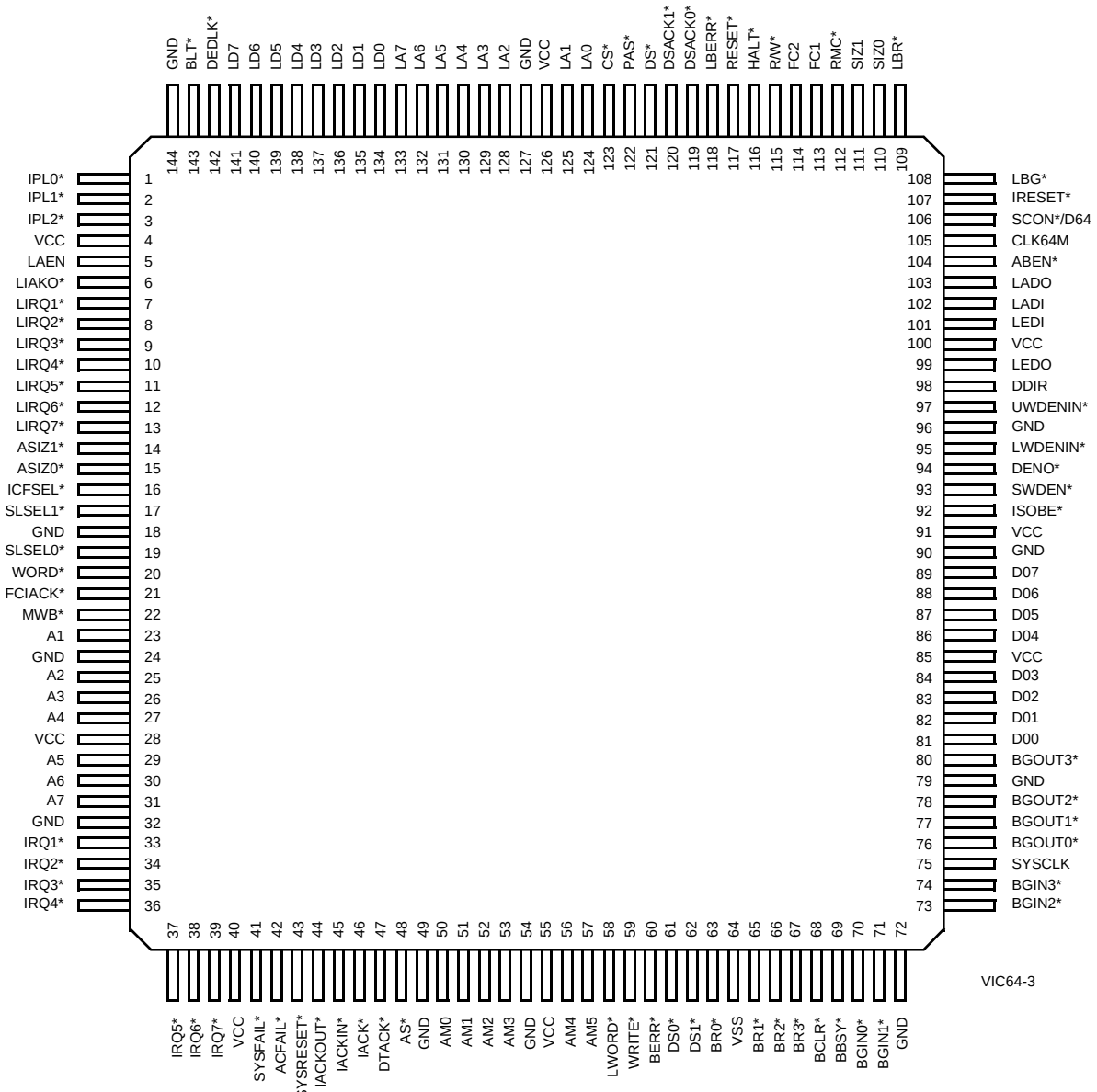
160-Pin Quad Flatpack (QFP)
Top View



VIC64-2

Pin Configurations (continued)

144-Pin Thin Quad Flatpack (TQFP)
Top View



VIC64-3

Functional Description (continued)

The VIC64 supports A32, A24, and A16, as well as user-defined address spaces.

Master Write-Posting

The VIC64 is capable of performing master write-posting (bus decoupling). In this situation, the VIC64 acknowledges the local resource immediately after the request to the VIC64 is made, thus freeing the local bus. The VIC64 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 (IMACs) are easily performed using the VIC64. Significant control is allowed for:

- 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

If a master operation is attempted when a slave operation to the same module is in progress, a deadlock condition occurs. The VIC64 signals a deadlock condition by asserting the DEDLOCK* signal. This should be used by the local resource requesting the VMEbus to try the transfer after the slave access has completed.

Self-Access

If the VIC64, while it is VMEbus master, has a slave select signaled, a self-access has occurred. The VIC64 asserts BERR* and LBERR*.

VIC64 VMEbus Slave Cycles

The VIC64 is capable of operating as a VMEbus slave controller. The VIC64 contains a highly programmable environment to allow for a wide variety of slave configurations. The VIC64 allows for:

- D64, 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 VIC64 requests the local bus. When local bus mastership is obtained, the VIC64 reads or writes the data to/from the local resource and asserts the DTACK* signal to complete the transfer.

Slave Write-Posting

The VIC64 is capable of performing a slave write-post operation (bus decoupling). When enabled, the VIC64 latches the data to be written, and acknowledges the VMEbus (asserts DTACK*) immediately thereafter. This prevents the VMEbus from having to wait for local bus access.

Address Modifier (AM) Codes

The VIC64 encodes and decodes the VMEbus address modifier codes. For VMEbus master accesses, the VIC64 encodes the appropriate AM codes through the VIC64 FCi and ASIZi signals, as well as the block transfer status. For slave accesses, the VIC64 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 VIC64 also supports user-defined AM codes; that is, the VIC64 can be made to assert and respond to user-defined AM codes.

VIC64 VMEbus Block Transfers

The VIC64 is capable of both master and slave block transfers. The master VIC64 performs a block transfer in one of two modes:

- The Master Block Transfer with Local DMA (D16, D32, and D64)
- The MOVEM-type Block Transfer (D16 and D32)

In addition to these VMEbus block transfers, the VIC64 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.

For D32 block transfers, 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 VIC64 allows for easy implementation of block transfers that exceed the 256-byte restriction by releasing the VMEbus at the appropriate time and re-arbitrating 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. For D64 block transfers, the VMEbus specification allows for bursts of up to 2048 bytes.

The VIC64 contains two separate address counters for the VMEbus and local address buses. In addition, a separate address counter is provided for slave block transfers. The VIC64 address counters are 8-bit up-counters that provide for transfers up to 256 bytes. For transfers that exceed the 256 byte limit, the external counters and latches are required.

The VIC64 is capable of performing A32/A16:D64/D32/D16 master block transfers. For D64 transfers, external logic is required for the multiplexing of the data and address signals for the upper 24 address/data lines. The CY7C964 is specifically designed for this purpose. Multiplexing for the lower 8 bits is done within the VIC64.

The VIC64 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 VIC64 for a MOVEM block transfer and proceeds with the consecutive-address cycles (such as a 68K MOVEM instruction). The local resource continues as the local bus master in this mode.

Master Block Transfers with Local DMA

In this mode, the VIC64 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.

D64 block transfers are not supported by MOVEM protocol.

VIC64 Slave Block Transfer

The VIC64 is capable of decoding the address modifier codes to determine that a slave block transfer is desired. In this mode, the VIC64 captures the VMEbus address, and latches it into internal counters. For subsequent cycles, the VIC64 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.

For D64 slave block transfers, the SCON*/D64 signal is asserted to indicate a D64 transfer is in progress. External logic is required to de-multiplex the data from the VMEbus address bus for the upper 24 address/data lines. The lower 8 bits are done within the VIC64.

Module-Based DMA Transfers

The VIC64 can act 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 source or destination.

VIC64 Interrupt Generation and Handling Facilities

The VIC64 can generate and handle a seven-level prioritized interrupt scheme similar to that used by the Motorola 68K processors. These interrupts include:

- 7 VMEbus interrupts
- 7 local interrupts
- 5 VIC64 error/status interrupts
- 8 interprocessor communication interrupts.

The VIC64 can be configured to act as handler for any of the seven VMEbus interrupts. The VIC64 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 VIC64 drives the IACK* daisy chain.

The local interrupts can be configured with the following:

- User-defined local interrupt priority level (IPL)
- Option for VIC64 to provide the status/ID vector
- Edge or level sensitivity
- Polarity (rising/falling edge, active HIGH/LOW)

The VIC64 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 VIC64 can also interrupt on the setting of a module or global switch in the interprocessor communication facilities.

Interprocessor Communication Facilities

The VIC64 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
- VIC64 version/revision register (read-only)
- VIC64 reset/halt condition (read-only)
- VIC64 interprocessor communication register semaphores

When set through a VMEbus access, these switches can interrupt a local resource. The VIC64 includes module switches that are intended for a single module, and global switches which are intended to be used as a broadcast.

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

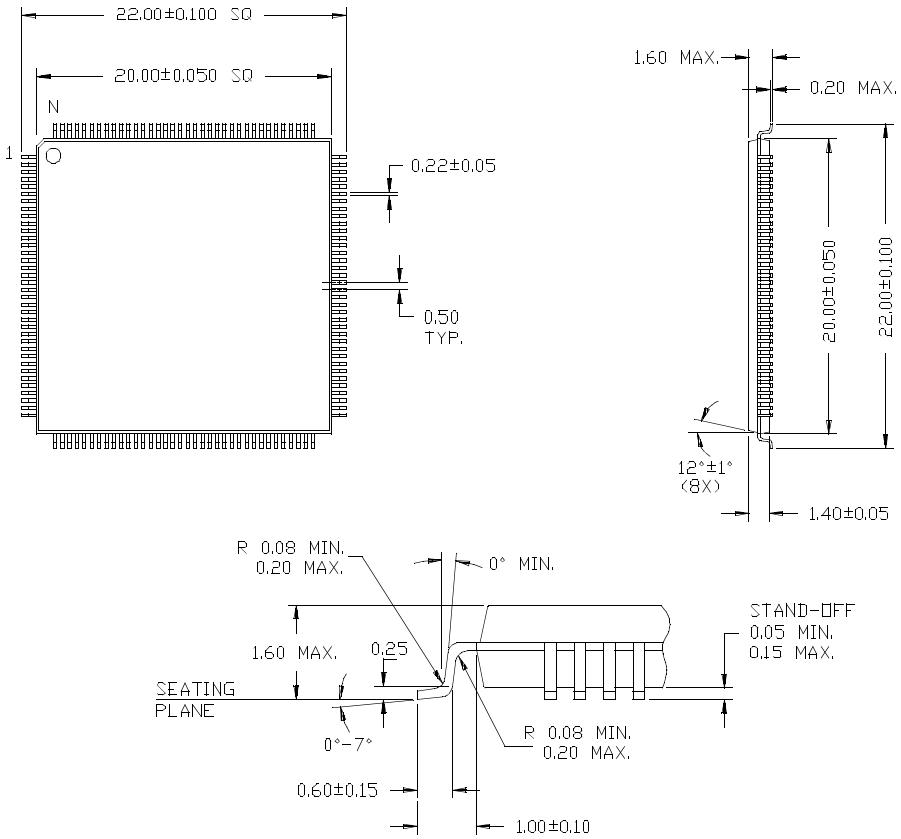
VMEbus Interface Handbook

Ordering Information

Ordering Code	Package Name	Package Type	Operating Range
VIC64–AC	A144	144-Lead Thin Quad Flatpack	Commercial
VIC64–BC	B144	145-Pin Plastic Pin Grid Array	
VIC64–GC	G145	145-Pin Ceramic Pin Grid Array	
VIC64–NC	N160	160-Lead Plastic Quad Flatpack	
VIC64–GI	G145	145-Pin Pin Grid Array	Industrial
VIC64–GM	G145	145-Pin Ceramic Pin Grid Array	Military Temp. Commercial
VIC64–GMB	G145	145-Pin Ceramic Pin Grid Array	MIL-STD-883
VIC64–UMB	U162	160-Lead Ceramic Quad Flatpack	MIL-STD-883
VIC64–UM	U162	160-Lead Ceramic Quad Flatpack	Military Temp. Commercial

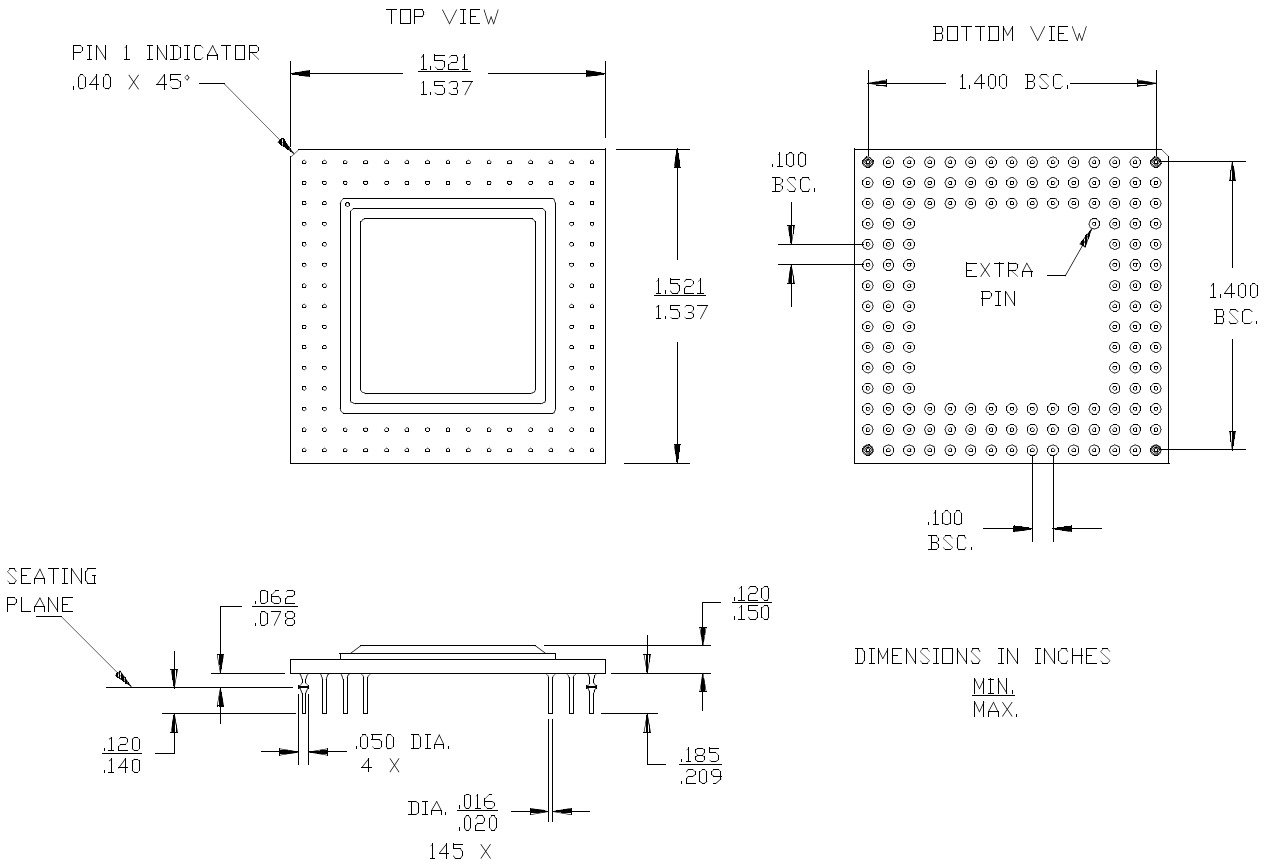
Package Diagrams

144-Pin Thin Quad Flat Pack A144



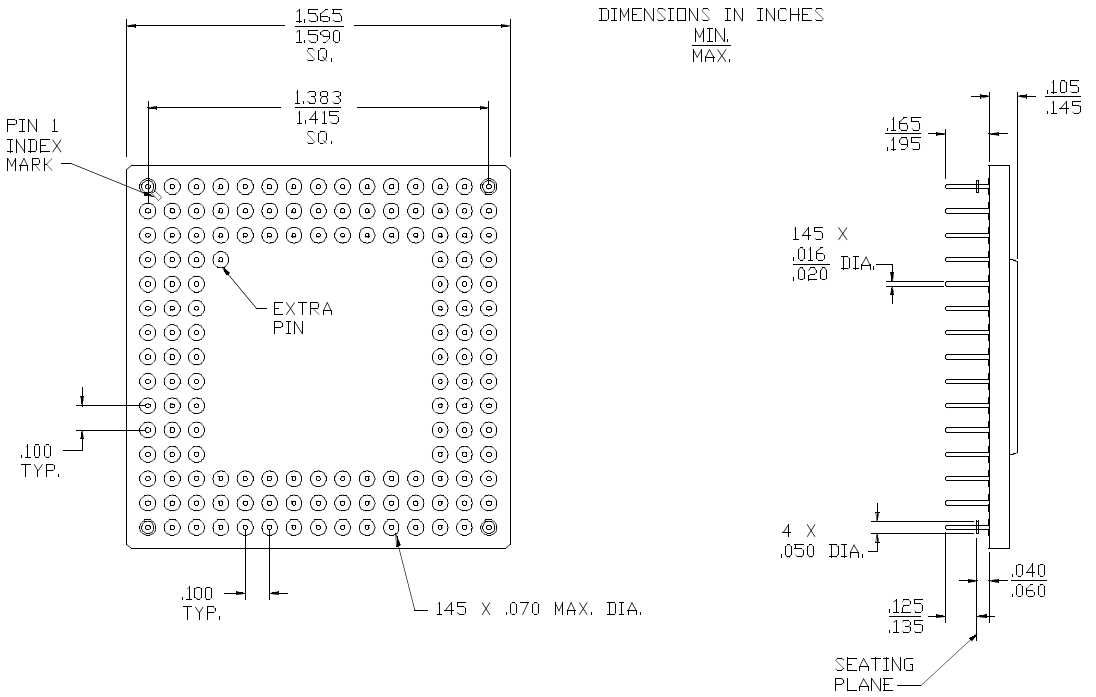
Package Diagrams (continued)

145-Pin Plastic Grid Array (Cavity Up) B144



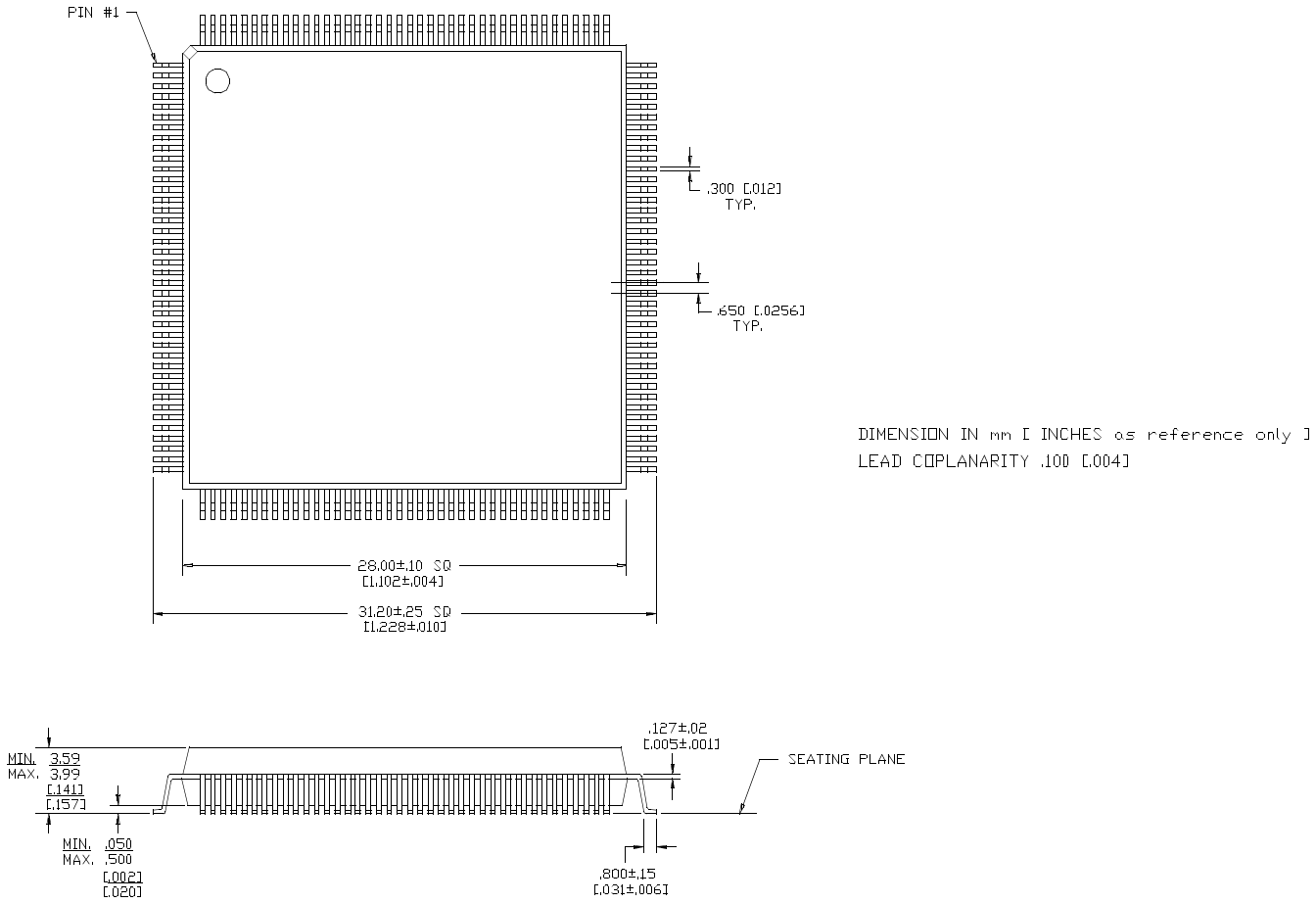
Package Diagrams (continued)

145-Pin Grid Array (Cavity Up) G145



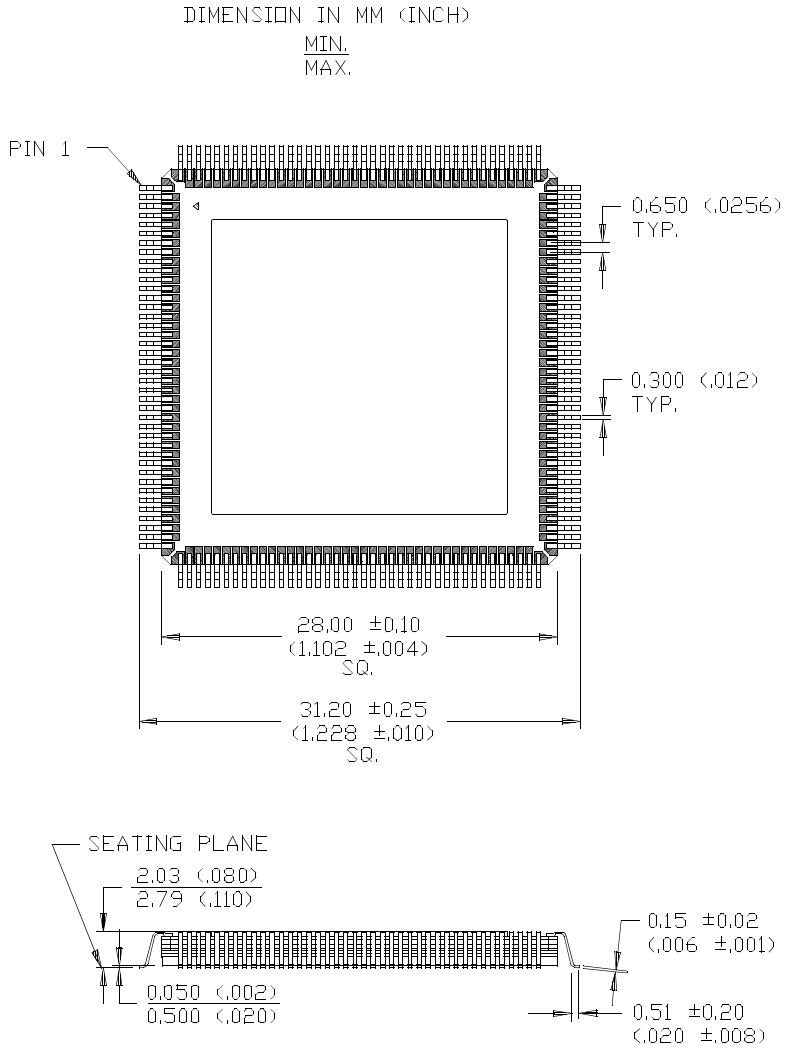
Package Diagrams (continued)

160-Lead Plastic Quad Flatpack N160



Package Diagrams (continued)

160-Lead Ceramic Quad Flatpack U162



Document Title: VIC64 Interface Controller with D64 Functionality Document Number: 38-09002				
REV.	**ECN NO.**	**Issue Date**	**Orig. of Change**	**Description of Change**
**	106237	04/19/01	SZV	Change from Spec number: 38-00196 to 38-09002