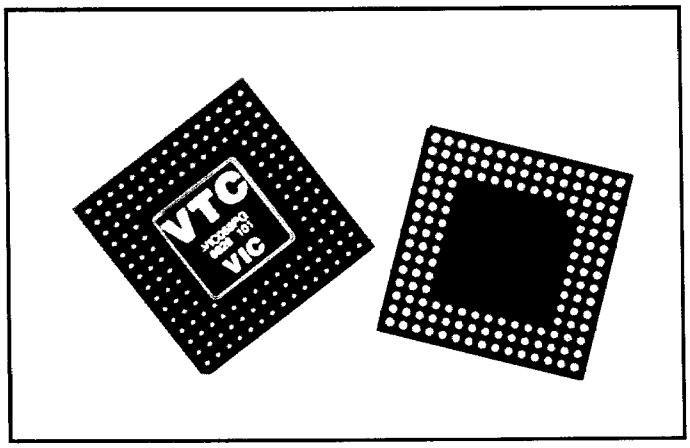
VTC Incorporated

VIC068
VMEbus INTERFACE CONTROLLER

FEATURES

- **Complete VMEbus Interface Controller and Arbiter**
  - 58 Internal Registers Provide Configuration Control and Status of VME and Local Operations
  - Drives Arbitration, Interrupt, Address Modifier Utility, Strobe, Address Lines A07-A01 and Data Lines D07-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 (7.8 ns granularity)
  - Programmable Metastability Delay
  - Programmable Delays for DSACK to DTACK
  - Provides 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
  - Also 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 VME Interrupts: Interrupter and Interrupt Handler
  - Seven Local Interrupt Lines
  - 8-level Interrupt Priority Encode
  - Total of 29 Interrupts Mapped through the VIC068.
- **Miscellaneous Features**
  - Refresh Option for Local DRAM
  - Option to Drive DTACK High upon Release
  - Four Broadcast Location Monitors
  - Four Module-specific Location Monitors
  - Eight Interprocessor Communications Registers
  - TAS/CAS/CAS2 Instruction Support for 68020
  - Available in 144-pin Plastic or Ceramic PGA Package.



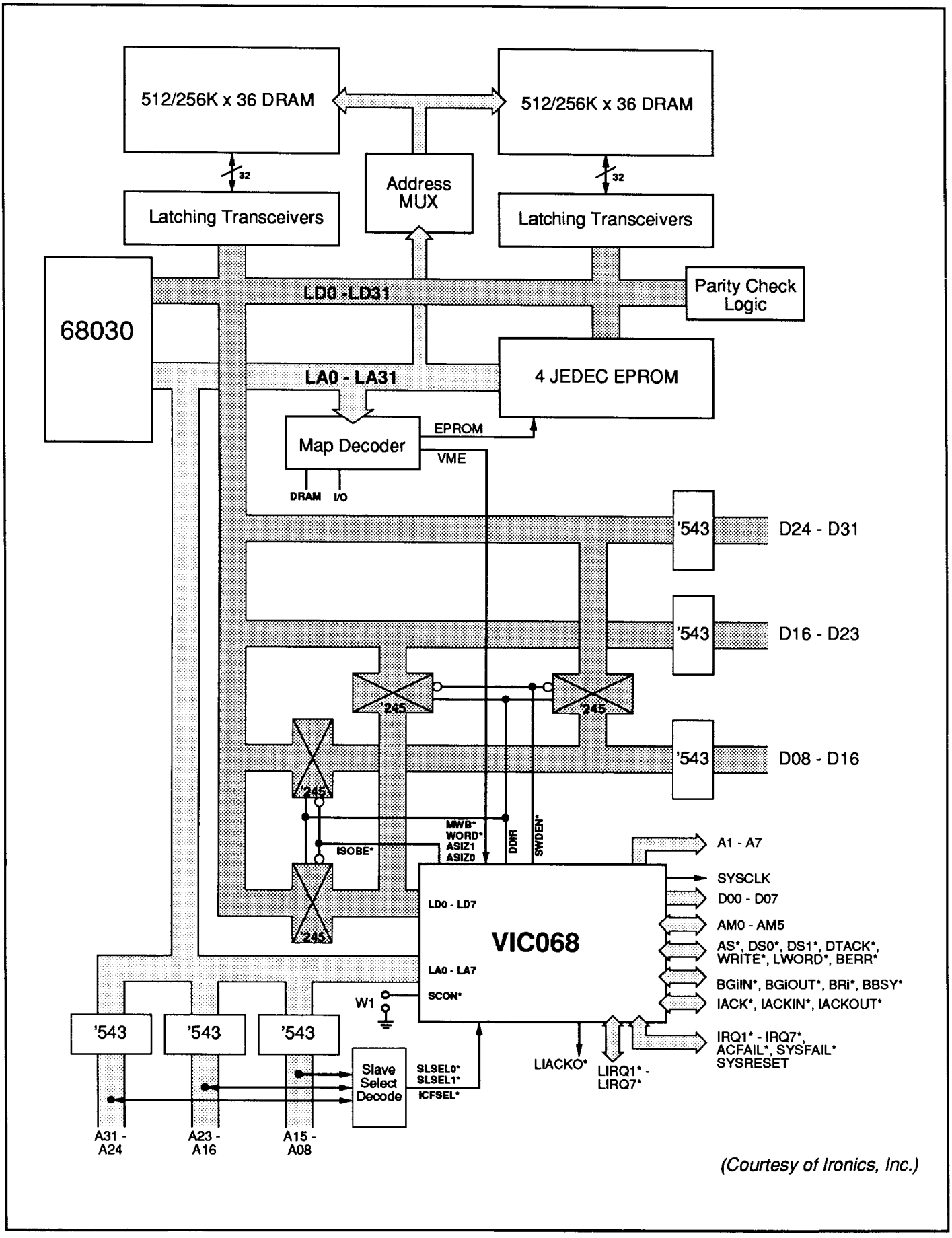
VTC's VIC068 Interface Controller chip is available in a 144-pin pin grid array (PGA) package.

DESCRIPTION

The VME Interface Controller (VIC068) 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 either a 8-bit, 16-bit, or 32-bit system. The VIC068 was designed using VTC's high-performance standard cells on an advanced 1 micron CMOS process. The VIC068 provide all VMEbus system controller functions plus many other features which simplify the development of a VMEbus interface. The VIC068 utilizes output buffers based on VTC's patented and military-approved ACL product family. These CMOS high drive buffers provide direct connection to the address and data lines. It addition to these signals, the VIC068 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 VIC068 was developed through the joint efforts of both VTC and a consortium of board vendors, under the auspices of the VMEbus International Trade Association (VITA). The VIC068 thus offers a implementation that provides inputs from a wide array of users which maximizes the number of applications. This also provides compatibility between boards designed by different manufacturers.

VIC068 ON 68030 BOARD



(Courtesy of Ironics, Inc.)

PIN DESCRIPTIONS

VMEbus Signals

SYSRESET* The System Reset signal. This signal is both an input and an output collector output. A low level on this signal resets the internal logic of the VIC and asserts the signals RESET* and HALT*; those signals remain asserted for a minimum of 200 ms. If the VIC is configured as a VMEbus system controller, a low level input on signal IRESET* asserts SYSRESET* as an output for a minimum of 200 ms.

ACFAIL* This signal is an input only. It indicates that the power has failed. VIC may be programmed to generate a local interrupt when ACFAIL* is detected.

SYSFAIL* The System Fail signal. It indicates that it is not in full working order. This signal is both an input and an open collector output. The output is asserted under two conditions. First, if self-testing is not complete after startup, the SYSFAIL* signal is asserted—ostensilby to be deasserted by the onboard CPU after completion of diagnostics. Second, if a module has identified itself as faulty, the SYSFAIL* signal is asserted as a warning when HALT* is detected low for 4 ms. The VIC may be programmed to generate a local interrupt when SYSFAIL* is asserted. SYSFAIL* assertion may be inhibited by a control bit in the Inter processor Communications Registers.

SYSCLK The system clock output at 16 MHz. It is driven as an output only while the VIC is the VMEbus system controller and tristated otherwise.

BR3*-BR0* The Bus Request signals for arbitration. These signals are both inputs and open collector outputs. These signals may be asserted by the VIC as a requester. They are used as inputs during arbitration. Only one of these signals are asserted by VIC at one time.

BG3IN*-BG0IN* The four Bus Grant In signals in the daisy-chained arbitration scheme. These signals are inputs only and have internal active pullups so that they may be left unconnected.

BG3OUT*-BG0OUT* The Bus Grant Out signals in the daisy-chained arbitration scheme. These signals are output only.

BBSY* The Bus Busy signal. This signal is both an input and an open collector output.

BCLR* The Bus Clear signal. This signal is both an input and an output.

D07-D00 The low order byte of the VME data bus. The VIC implements a transparently latching bidirectional transceiver on these lines to allow write posting.

A07-A01 The VMEbus Address lines A7-A1. A transparent latching bidirectional I/O buffer is used on the VIC for these lines to allow write posting and compatibility with masters employing address pipelining.

AS* The VMEbus Address Strobe. This signal is both an input and an output. It is asserted by VIC when VIC has mastership of the VME bus to initiate a transaction. For slave accesses, VIC uses AS* input to qualify the SLSEL0*, SLSEL1*, or ICFSEL* select signals to indicate to the module the presence of a valid slave address.

DS1*-DS0* The Bus Data Strobes. These signals are both inputs and outputs. They are asserted by VIC when it is bus master to initiate data transfers and used by VIC to indicate to the module logic of the presence of a request to transfer data while VIC is acting as a slave.

DTACK* The Bus Data Acknowledge. This signal is both an input and an open collector or rescinding output. It is asserted by the slave in a transaction to indicate to the Current Master the termination of the current data transfer.

BERR* The Bus Error signal. This signal is both an input and an open collector output. It is asserted by the slave as an alternative to Data Acknowledge (DTACK*) to indicate an error in the current data transfer. The VIC may also assert this signal while it is the system controller if a slave does not DTACK* within the time programmed into the transfer timeout register ($A3) for VMEbus timeout. VIC will issue BERR* for self access operation.

WRITE* The Bus Write signal. This signal is both an input and an output. It is asserted by the current master with the same timing as VMEbus address lines to indicate the direction of data transfers.

LWORD* This signal is both an input and an output. It is driven by current bus master to indicate the size of data transaction.

AM5-AM0 The Address Modifiers. These signals are both inputs and outputs. These are driven by VIC as current bus master. VIC uses these to qualify slave accesses during slave select cycles. (It is presumed that VIC will not be presented with more than one valid slave access.)

IACK* The Bus Interrupt Acknowledge signal. This signal is both an input and an output. It is asserted by VIC as a current VMEbus master if the tansaction is an interrupt acknowledge cycle.

IACKIN* The Interrupt Acknowledge In signal in the daisy-chained Interrupt Acknowledge priority scheme. This signal is an input only with an internal active pullup. (Note that this pull-up resistor is to take care of floating input types. It cannot provide any significant current.) A low level input to this signal may occur as a result of an Interrupt Acknowledge transaction on VMEbus. The VIC may either consume it, if it has an interrupt request pending at that level, to become a slave supplying a status ID, or pass it on by way of the Interrupt Acknowledge Out (IACKOUT*).

IACKOUT* The Interrupt Acknowledge Out signal in the daisy chained Interrupt Acknowledge scheme. This signal is an output only. The VIC asserts this output if IACKIN*, IACK*, and either data strobe is asserted, 40ns has elapsed since assertion of a data strobe, and the VIC has no interrupt requests pending on the acknowledged level.

IRQ7*-IRQ1* The VMEbus Interrupt Request signals. These signals are both inputs and open collector outputs. Any combination of these signals may be asserted by the VIC to interrupt other modules on VMEbus. Any combination of these lines may also be monitored by the VIC to generate interrupts to the processor at the programmed level.

CPU Interface

LD7-LD0 The low order byte of data lines to the local bus. These signals are both inputs and tristate outputs. VIC register accesses are performed by the onboard CPU using these lines and the CS* input. Typically they are connected to the processor data lines D7-D0 through an isolation buffer.

LA7-LA0 These signals are both inputs and tristate outputs. They are input address lines for VME master operations, register selection and interrupt priority level recognition. When the VIC is acting as a slave VMEbus transaction, these lines become outputs. These lines also become outputs in support of VME master block transfers with local DMA.

For 32-bit 68020/030 processors, these signals connect to processor addresses A7–A0. For 16-bit 68000/010 processors, these signals connect to processor addresses A7–A1 and LA0 becomes the lower data strobe, LDS*. The VIC decodes the interrupt priority level (IPL(2–9) from its inputs LA3–LA1. During an interrupt acknowledge cycle, FCIACK* assertion indicates the interrupt acknowledge cycle.

CS* The Chip Select input. This signal is an input only, and should be driven to the low level to access VIC internal registers.

PAS* The Physical Address Strobe signal. This signal is both an input and a tri-state output. The VIC drives this signal as an output while it is acting as a slave to a VMEbus transaction and for local DMA and refresh mode.

DS* The Data Strobe signal from the processor. This signal is both an input and a tri-state output. The VIC drives this signal as an output while it is acting as a slave to a VMEbus transaction and for local DMA. When the VIC is configured to control 16-bit 68000/010 proccessors, this signal acts as the Upper Data Strobe (UDS*).

DSACK1*-DSACK0* The Data Acknowledge signals. These signals are both inputs and rescinding outputs. The VIC asserts one or both signals in response to register accesses. When the VIC initiates a transaction as the Bus Master on the VMEbus, the cycle on the local bus side is terminated by assertion of the appropriate acknowledge signals (DSACK0*-DSACK1*or LBERR*). When DSACK0* is strapped to ground, the VIC adopts the 68000/68010 local bus protocol by disabling its DSACK0* output driver; DSACK1* is used for the local processor's data transfer acknowledge.

LBERR* The Bus Error signal to the local bus. This signal is both an input and a rescinding output. When the VIC initiates a transaction as Bus Master on VMEbus, the cycle is terminated by the asertion of LBERR* when a VMEbus error is detected. If this signal is asserted on the local side while the VIC is acting as a slave to a VMEbus transaction, a Bus Error on VMEbus is generated. (Note: A retry is initiated by 680x0 processors if HALT and LBERR* are received asserted.)

RESET* This is an output signal. A low level on IRESET* causes RESET* to be asserted for a minimum period of 200 ms. A low level on SYSRESET* causes RESET* to be asserted to the module logic for the duration of the low on SYSRESET*. If bit 6 of Interprocessor Communication Register ($7F) is set. RESET* and HALT* will be asserted until this bit gets cleared.

HALT* The Halt signal. This signal is an input and an open collector output. A low level on IRESET* or SYSRESET* causes RESET* and HALT* to be asserted

Output		
FC2	FC1	Function
0	0	Slave Block Transfer
0	1	Local DMA
1	0	Standard Slave Access
1	1	DRAM Refresh

3

for a minimum period of 200 ms. Assertion of HALT* for greater than 4 ms by anything other than the VIC causes the VIC to assert SYSFAIL*. (Note: A retry is initiated by 680x0 processors if HALT and LBERR* are received asserted.)

R/W* The Read/Write signal. This signal is both an input and a tri-state output. This signal is driven as an output when the VIC is acting as a slave to a VMEbus transaction with the same timing as the local address lines and when the VIC is acting as a DMA controller on the local bus. As an output, R/W* has the same

SIZ1	SIZ0	Function
0	0	LWORD
0	1	BYTE
1	0	WORD
1	1	3-BYTE

timing as a local address line; it is set up before PAS* and DS* assertion and remains stable until the VIC negates DS*.

FC2-FC1 The Function Code signals. These signals are both inputs and tri-state outputs. As inputs, they are used to encode the VMEbus address modifier outputs. These signals are driven as outputs when the VIC owns the local bus to identify refresh cycles, local DMA cycles, slave, and slave block transfer cycles.

RMC* The Read Modify Write signal. This signal is an input only. The VIC uses this signal to ensure the indivisibility of transactions on VMEbus by inhibiting bus release and (optionally) stretching AS*, and (optionally) requesting the VMEbus. This signal and bit 5-7 of register $AF control various modes as indicated in the RMC Control table.

SIZ1-SIZ0 The size signals. These signals are both inputs and tri-state outputs. When acting as a slave to a VMEbus transaction, the VIC drives these signals as outputs to indicate the size of the transfer requested.

LBR* The Local Bus Request signal to the processor. This signal is an output only. When managing shared resources, as a VMEbus Contoller, the VIC asserts this signal to allow slave access to the local bus. LBR* remains asserted for the duration of the slave transaction to eliminate the need for a Local Bus Grant Acknowledge signal.

LBG* The Local Bus Grant signal. This signal is an input only. In response to a low level on this signal while Local Bus Request (LBR*) is asserted, the VIC assumes control of the local bus and begins slave transfers or local DMA for block transfers. Once asserted this signal should remain asserted throughout the time LBR* is asserted.

MWB* The Module Wants Bus signal. This signal is an input only. The VIC requests the VMEbus in response to a low level on this input. This signal is required for all VMEbus transactions initiated from the VIC module.

FCIACK* This signal indicates that the current access is an interrupt acknowledge cycle. This signal is an input only. For host CPUs which do not follow the 68xxx family interrupt acknowledge protocol, this signal can be used as a "chip select" for requesting a read of a status/ID byte.

SLSEL0* This signal is one of three slave select lines. This input causes the VIC to initiate dual-port arbitration with the modules local bus arbiter when asserted. This input is generated by external A16, A24, or A32 VMEbus map decoder. VIC qualifies this select signal with AS* and appropriate AM codes programmed in Slave Select0 Control Register0.

SLSEL1* This signal is one of three slave select lines. This input causes the VIC to initiate dual-port arbitration with the modules local bus arbiter when asserted. This input is generated by external A16, A24, or A32 VMEbus map decoder. VIC qualifies this select signal with AS* and appropriate AM codes programmed in Slave Select1 Control Register0.

ICFSEL* This signal is one of three slave select lines. This input requests access to one of the VICs interprocessor communications facilities via VMEbus: the registers, global switches or module switches. This input is generated by an external A16 VMEbus map decoder.

ASIZ0* This input, when asserted in conjunction with MWB* or BLT*, indicates that the VIC should use the address modifier code for 16-bit addressing. This input serves as a data acknowledge signal, (analogous to DTACK*), when the VIC is performing local DMA without VME transfers. (Refer to Section 11.)

ASIZ1* This input, when asserted in conjunction with MWB* or BLT*, indicates that the VIC should use the address modifier code for 32-bit addressing. If neither ASIZ0* nor ASIZ1* is asserted, then the address modifiers for standard addressing, A24, will be used. This signal serves as a data error signal, analogous to BERR*, when the VIC is peforming local DMA without VME transfers. (Refer to Section 11.)

WORD* When this input is asserted, the VMEbus is treated as a 16-bit data path. Otherwise, it is considered a 32-bit data path. If WORD* is strapped to ground at power up and stays that way, it configures VIC in D16 mode on the VMEbus.

BLT* This is an input and an open collector output. As an output, this signal acknowledges that a VME block transfer with local bus DMA is in progress. BLT* in conjunction with LAEN & FC(2:1) controls the loading, enabling, and incrementing of an external latch for LA8+ of the local address. If BLT* is asserted by external circuitry while MWB* remains high and bit 7 of register $D7 is set, the VIC initiates and performs non-VME related local DMA cycles using the local bus portion of the block transfer with local DMA protocol.

DEDLK* This signal indicates deadlock, a slave access request concurrent with local CPU request for VMEbus. It should be used by the module logic to remove the local CPU's request for VMEbus. This signal will get activated for "self access" of the VIC module along with LBERR*.

IPL2*-IPL1* These pins function as interrupt request level outputs 2 and 1. They are output only. These lines drive the processor Interrupt Priority Lines IPL*(2–1).

IPL0* This pin is both an output and an input. As output it functions as interrupt request level output 0. During assertion of IRESET* this pin may be driven as an input to provide a "global" reset to the VIC.

LIRQ7*-LIRQ1* The Local Interrupt Request signals. These signals are inputs only (except LIRQ2*). They receive interrupt requests from the module which are merged by the VIC with other interrupt request sources to produce the three encoded Interrupt Priority Lines (IPL2*-IPL0*). All LIRQx inputs have internal pull-up resistors so that they may be left unterminated. LIRQ2* functions as a timer output when VIC's programmable clock tick timer is enabled.

LIACKO* This signal is a Local Interrupt Acknowledge Output. The assertion of this signal by the VIC either allows a local interrupting device to place the device's interrupt vector on the data bus in response to a CPU interrupt acknowledge cycle or can be used to affect autovectoring.

IRESET* The Internal Reset signal. This input has hysteresis and allows an RC time constant to be connected to an external switch to produce an operator reset. The VIC is reset by the assertion of the IRESET*. A low level on IRESET* causes the signals RESET* and HALT* to be asserted for a minimum period of 200 ms. If SCON* is low, then SYSRESET* is also driven low on VMEbus by the assertion of IRESET*.

SCON* This pin should be tied low to cause the VIC to perform the VMEbus system controller functions.

CLK64M 64 MHz square wave used to clock internal arbitration and time delay functions and to generate SYSCLK. A 60/40 worst case duty cycle must be provided. Higher clock speed could result in set-up/hold time violations of VMEbus signals.

Buffer Control Signals

The signals in this group are outputs only with 8 mA IOL. They provide the control signals for the address buffers and data buffers/latches to VMEbus and to control the swap requirements of 680xxx processors. Their function may vary depending on the states of certain control bits in the VIC. If WORD* is strapped to ground at reset time and stays that way it puts VIC in D16 mode. Refer to the following block diagram for a sample hookup of the buffer control signals.

ABEN* This signal is an output only. This pin functions as Enable Address Out. It is asserted by the VIC when bus mastership is obtained to cause latching address transceivers with OE* pins (i.e., the 543) to drive the VMEbus address lines. LAEN is connected to OEBA* of the latching address transceivers to enable the VMEbus address onto the local bus for slave transactions and for DMA-based block transfers.

LADO This output signal is the Latch Address Out control signal. It should be connected to the LEAB* pin of address latches driving onto the VMEbus from the local bus. When deasserted, the latches are in the fall-through mode. When it is asserted (high), the latch contents are held constant. If a block transfer is suspended in order to release the bus after burst, LADO is asserted until the VIC regains the bus and resumes and completes the Block Level Transfer (BLT) so that the block transfer high order address bits maintained in the external latches will not be lost.

LADI This output signal is the Latch Address In control signal. It should be connected to the LEBA* pin of address latches driving from the VMEbus to the local bus. When low (deasserted), the latches are in the fall-through mode. When it is asserted (high), the latch contents are held constant.

LEDO This output signal is the Latch Enable Data Out control signal. It should be connected to the LE* pins of data latches whose outputs connect to the VMEbus and whose inputs drive the local bus. When deasserted, the latches are in the fall-through mode. When it is asserted (high), the latch contents are held constant.

LEDI This output signal is the Latch Enable Data In control signal. It should be connected to the LE* pins of data latches whose inputs connect to the VMEbus and whose outputs drive the local bus. When deasserted, the latches are in the fall-through mode. When it is asserted (high), the latch contents are held constant.

DDIR* The Data Direction signal. This signal is an output only. This signal controls the direction of data flow for swapping and isolation buffer transceivers as shown in Figure 2.1.

DENO* This is the Data Enable out signal. It should be used to drive the OE* pins of 543 type latching data transceivers (e.g., 543) to enable the buffers to drive onto the VMEbus.

SWDEN* This pin serves as Swap Data Enable control function. It enables the swapping buffers for swapping data between LD(31–16) and LD(15–0).

LWDENIN* With latching buffers, LWDENIN* enables data input from VMEbus D16-D31 onto the local bus LD16–LD31.

UWDENIN* With latching buffers, UWDENIN* enables data input from VMEbus D16–D31 to local bus LD16–LD31.

ISOBE* This is the Isolation Buffer Enable signal. It is asserted for accesses of the VIC and VMEbus by the CPU. If using dual-port memory, the module logic must provide for the enabling of the isolation buffer for a local memory access by its onboard CPU.

LAEN This is asserted (high) to indicate VIC is driving local bus address lines. This pin should be used to enable OEBA* to enable VMEbus addresses onto local bus address lines.

VIC REGISTER VALUES AFTER VARIOUS RESET OPERATIONS

Address	Register Name	Global Reset	System Reset	Internal Reset
$03	VMEbus Interrupter Interrupt Control Register	1 1 1 1 1 0 0 0	1 1 1 1 1 X X X	1 1 1 1 1 X X X
$07 - $1F	VMEbus Interrupter Control Registers 1 thru 7	1 1 1 1 1 0 0 0	1 1 1 1 1 X X X	1 1 1 1 1 X X X
$23	DMA Status Interrupt Control Register	1 1 1 1 1 0 0 0	1 1 1 1 1 X X X	1 1 1 1 1 X X X
$27 - $3F	Local Interrupt Control Registers 1 thru 7	1 0 0 0 X 0 0 0	1 X XX X X XX	1 X XX X X XX
$43	ICGS Interrupt Control Register	1 1 1 1 1 0 0 0	1 1 1 1 1 X X X	1 1 1 1 1 X X X
$47	ICMS Interrupt Control Register	1 1 1 1 1 0 0 0	1 1 1 1 1 X X X	1 1 1 1 1 X X X
$4B	Error Group Interrupt Control Register	1 1 1 1 1 0 0 0	1 1 1 1 1 X X X	1 1 1 1 1 X X X
$4F	ICGS Interrupt Vector Base Register	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1
$53	ICMS Interrupt Vector Base Register	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1
$57	Local Interrupt Vector Base Register	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1
$5B	Error Group Interrupt Vector Base Register	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1
$5F	Interprocessor Communications Switch Register	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	XX X X 0 0 0 0
$63 - $73	Interprocessor Communications Registers 0 thru 4	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$77	Interprocessor Communications Register 5	1 1 1 1 0 0 0 1	1 1 1 1 0 0 0 1	1 1 1 1 0 0 0 1
$7B	Interprocessor Communications Register 6	X 1 1 1 1 1 X X	X 1 1 1 1 1 1 1	X 1 1 1 1 1 1 0
$7F	Interprocessor Communications Register 7	0 0 X 0 0 0 0 0	0 0 X 0 0 0 0 0	X 0 X X X X XX
$83	VMEbus Interrupt Request and Status Register	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	X X X X X X X0
$87 - $9F	VMEbus Interrupt Vector Registers 1 thru 7	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1	X X X X X X XX
$A3	Transfer Timeout Register	0 1 1 0 1 0 0 0	0 1 1 0 1 0 0 0	0 1 1 0 1 0 0 0
$A7	Local Bus Timing Register	0 0 0 0 0 0 0 0	X XX X X X X X	X X X X X X XX
$AB	Block Transfer Definition Register	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$AF	VMEbus Interface Configuration Register 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$B3	Arbiter and Requester Configuration Register	0 1 1 0 0 0 0 0	0 1 1 X 0 0 0 0	0 1 1 X 0 0 0 0
$B7	Address Modifier Source Register	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$BB	Bus Error Status Register	X 0 0 0 0 0 0 0	X 0 0 0 0 0 0 0	X 0 0 0 0 0 0 0
$BF	DMA Status Register	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$C3	Slave Select 0 Control Register 0	0 0 0 0 0 0 0 0	0 0 X X X X XX	0 0 X X X X XX
$C7	Slave Select 0 Control Register 1	0 0 0 0 0 0 0 0	X X X X X X XX	X X X X X X XX
$CB	Slave Select 1 Control Register 0	0 0 0 0 0 0 0 0	0 0 X X X X XX	1 1 X X X X XX
$CF	Slave Select 1 Control Register 1	0 0 0 0 0 0 0 0	X X X X X X XX	X X X X X X XX
$D3	Release Control Register	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$D7	Block Transfer Control Register	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$DB	Block Transfer Length Register 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$DF	Block Transfer Length Register 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$E3	System Reset Register	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
$EB - $FF	Undefined Locations	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1

THEORY OF OPERATION

The VIC068 is a bridge between two buses, the local bus and the VMEbus. The VIC allows some resources to be accessed from the local bus via a bridge to the VME subsystem.

The local bus interface is similar to the 68xxx family of Motorola's processors. All signal names have a nomenclature corresponding to Motorola's family of micros. Other processors can easily duplicate this interface if they follow the general timing sequence of the local bus cycle as described below.

Any local bus cycle is initiated by a Bus Master. It has to acquire the local bus; the arbitration scheme for the local bus is left open to the board designer.

VIC Bus Arbitration

The VIC is capable of being a Bus Master. To do so, it asserts a signal, LBR* (Local Bus Request), an active low signal. The VIC is granted the local bus by the assertion of LBG* (Local Bus Grant). The onboard logic has to undertake this arbitration function. The VIC expects LBG* to be kept asserted until it deasserts LBR*. Deassertion of LBR* by the VIC implies that it is done with the bus. (Note: If the VIC received LBG* asserted when it does not have LBR* asserted, no action is undertaken by the VIC. LBG* can be removed without any constraints.) See figure below.

Bus Acquisition During VIC Reset

During initialization of the VIC, when IRESET*/SYSRESET* are asserted, the VIC first tries to acquire local bus by asserting LBR*. If LBG* is not received after 1 µsec, the VIC undertakes the reset action regardless of the local bus being granted. The objective of 1 µsec timout is to do an orderly reset on the module. This will allow any local/VME cycle to complete before initialization begins. During initialization the VIC registers will get reset. The VIC needs a software initialization before any VME interface activities can resume in a coherent manner.

Local Bus Cycle Timing

1. Obtain Bus Mastership

2. Master
   - Assert: Address, ASIZ (1-0), FC(2-1), SIZ (1-0), R/W*, WORD*
   - Wait set-up time : minimum 5 ns
   - Assert PAS*
   - If Write Cycle, Assert data
   - Wait set-up time: minimum 10 ns
   - Assert: DS*

3. Slave
   - If Read: Supply data, on appropriate data lines
   - If Write: Capture data
   - Indicate end of operation: Assert DSACK(1-0)*

4. Master
   - Deassert PAS*, DS*
   - Wait min: 10 ns
   - Deassert: Address, Data, ASIZ, FC SIZ, R/W*, WORD*

5. Slave
   - See DS* Deasserted
   - Deassert DSACK (1-0)*
   - Stop driving data lines

Note: The VIC waits a programmed delay (1+ clocks) after last cycle (i.e., PAS* is deasserted and DSACK is high before it the drives bus. Delay is programmable by register $A7).

Various Master Cycles on VIC

Simple master operations start with local bus cycle. The module logic for 'Address MAP' puts out MWB* indicating that the resource is on the VMEbus. Assertion of MWB* signal causes a local cycle to be mapped on to VMEbus as a master cycle for read/write.

If the block transfer control register is programmed, this operation can transform into a VME master block transfer operation.

If RMC* signal gets asserted, it becomes one continuous operation of Read Modify Write, either by stretching AS* of VMEbus, or by continuing to occupy VMEbus during multiple cycles (as required for IMAC instructions of 68xxx family of processors). This is controlled by bits 5, 6, 7 of register $AF.

If CS* signal gets asserted, the local access is only to the VIC register map. The VIC will undertake a register Read/Write as per the local cycle request.

If BLT* signal is asserted and block transfer bit for local BLT is programmed, it will cause the VIC to get into local Block transfer without VMEbus.

Interrupt Acknowledge Cycle

During the master cycle, if FCIACK* signal is asserted by the module logic, it indicates interrupt acknowledge cycle. The VIC has the option of supplying a vector for most of the interrupt services. Only pending VME interrupts cause the VIC to start an interrupt acknowledge cycle on VMEbus. The VIC has to acquire the VMEbus before the VME interrupt handler will take over and obtain the vector from the interrupting device on the the VMEbus. The VIC will supply the vector to the local Bus Master that initiated the FCIACK* cycle.

VMEbus Data Size

The VMEbus data size is programmed and controlled by WORD* and SIZ(1-0) signals. D16 VME interface can be programmed by strapping WORD* low permanently. For D32 interface WORD* value controls the operations that take place on VMEbus and the VIC's response to the CPU in terms of DSACK. The DSACKs will always acknowledge as a 16-bit transfer if WORD* is asserted (low).

Master Writepost

The VIC can be programmed to post a local Bus Master cycle mapped onto VMEbus. If bit 7 of register $CB is set; all master write operations will be posted. A posted operation causes the address and data to be latched by the VIC. DSACKs are immediately asserted by the VIC to end the local bus cycle.

The VIC starts a VMEbus acquisition (if not VMEbus Master) and completes the posted operation. If any error results (i.e., the operation is BERR*ed on VMEbus) it will set up Write Post Fail interrupt. This interrupt should be turned on in the VIC if write posting is being used. The error recovery operation is left up to the implementer. There is no provision to supply the BERR*ed operation's address/data, so use this operation with caution.

If a second master cycle is followed immediately by a write-posted operation, the VIC will not DSACK the new operation until the previously posted write is completed.

AM (5-0) Map Control

ASIZ(2:1) and FC(2:1) control the AM field put out by the VIC during master operation. It is important to note that the ASIZ field is expected to be generated by the module Address Map Logic. It is the task of module logic to indicate a Module Wants VMEbus (MWB*) signal and assert the ASIZ* field. In the 68xxx environment the FC(2:1) is identical to that supplied by the processor function code. The FC field controls user data/program and supervisor data/program fields in the VME (AM(5–0)) address modifier field.

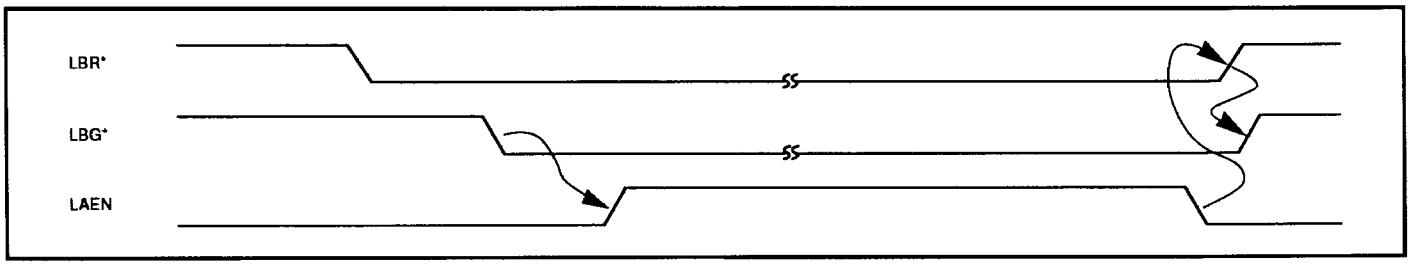
Register $B7, bit 7, controls the user-supplied AM field. Block transfer will use this option if bit 1 of $AB is set. Bit 7 of register $B7 has no effect on block transfer AM codes.

Special Conditions

DEDLK Operation

If, during a master operation, the local bus is mapped to go to VMEbus, an operation on VME is simultaneously in progress, and a valid slave operation comes through, the VIC will assert DEDLK* signal to the local Bus Master. It is expected that assertion of DEDLK* will cause the local Bus Master to abandon the current operation and retry the operation at a later time. This also tells the local bus arbitration logic to give the local bus grant to the VIC in priority mode,

VIC Arbitration of Local Bus



as the VIC needs to complete the local bus cycle. No local bus cycles can go through to the VMEbus during DEDLK* stage. DEDLK* will get deasserted after the slave cycle is complete.

Self-Access

If a VIC-initiated VMEbus cycle results in the selection of the same module's slave select pins, a self-access is said to have occurred. The self-access is BERR*ed and causes LBERR* and DEDLK* to take place on the local side. If LBERR*/DEDLK* combination results in a hardware retry, this could hold up the processor in an infinite loop. Software can undertake self-access operation and then access Error Status registers ($BB). Bits (1:2) indicate if self-access took place. This could be used to decode the Slave Address map of the module if so desired.

VIC as VME Slave/VIC as Local Bus Master

The VIC becomes the local Bus Master in four different ways:

1. Selected as slave from the VMEbus, standard and slave block transfer mode.
2. The VIC undertaking block transfer with VMEbus. The VIC is Bus Master on the local and VMEbus during this operation.
3. The VIC in local block transfers. In this mode the VIC is local Bus Master.
4. The VIC generating DRAM refresh timing (programmable option).

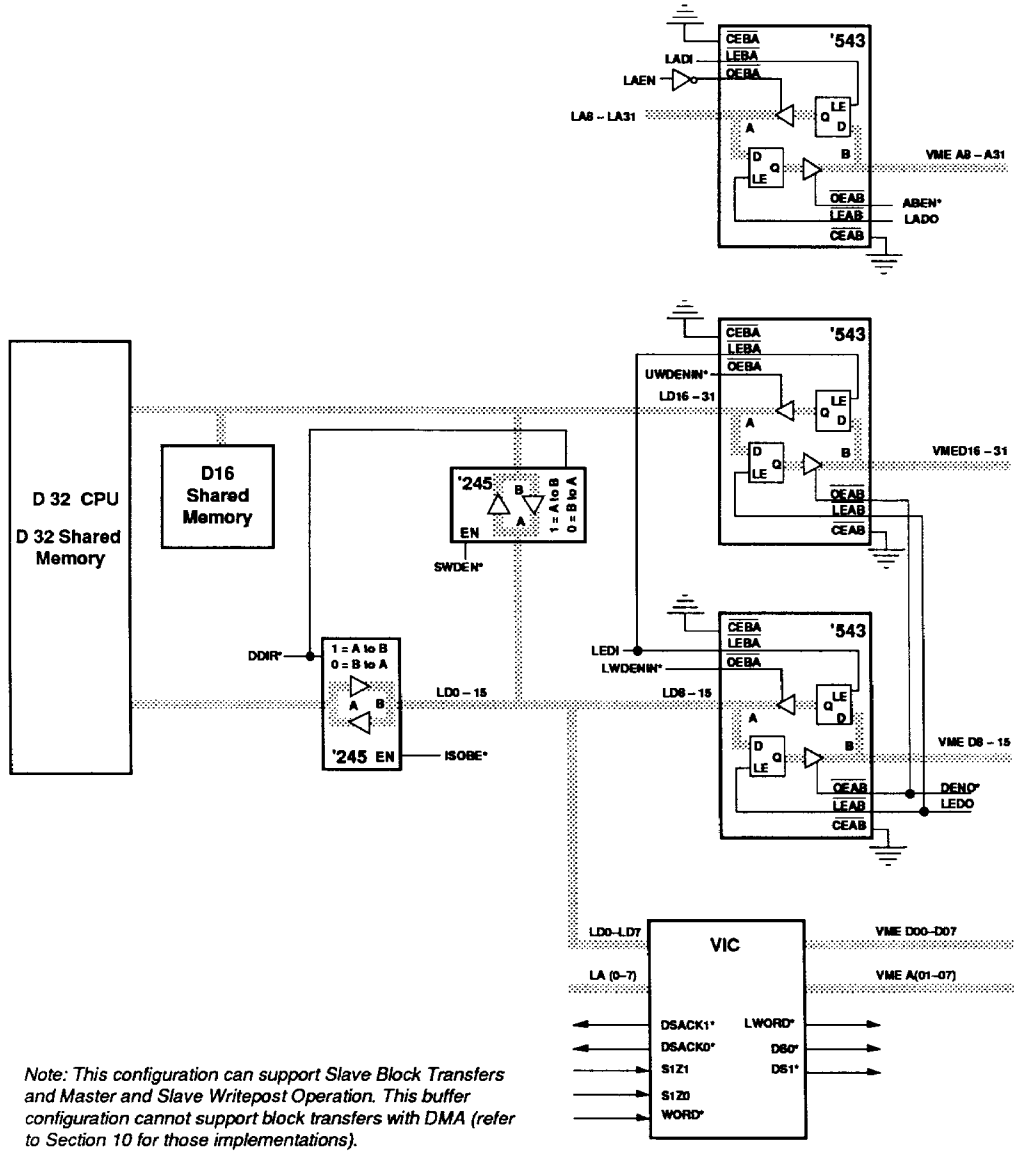
The VIC mode as a local Bus Master can be determined by Function Code pin FC(2:1) as shown in the table below. The VIC drives FC(2:1) when it becomes local Bus Master. It drives FC field along with LA, R/W*, etc. LAEN high indicates the VIC is driving local bus.

When VIC is Bus Master

FC(2)	FC(1)	Function Performed
L	L	Slave Block Transfer
L	H	Local DMA
H	L	Standard Slave Access
H	H	Dynamic RAM Refresh

The VIC shares a local bus which is tightly coupled to a DRAM on the module. The local bus timing is programmable and can support devices from the fastest available DRAMs to 150 ns access time devices. VIC assumes a tight coupling between the local bus address strobe and the DRAMs' RAS* (PAS*<> RAS*) and the local data strobe and the CAS* signals (DS* <> CAS*). Accordingly, it maintains compatible minimum pulse widths in both the high and low states for the strobes, and has a programmable DTACK delay to support advance data acknowledge on the local bus.

Buffer Control Signal for Shared Memory Implementation

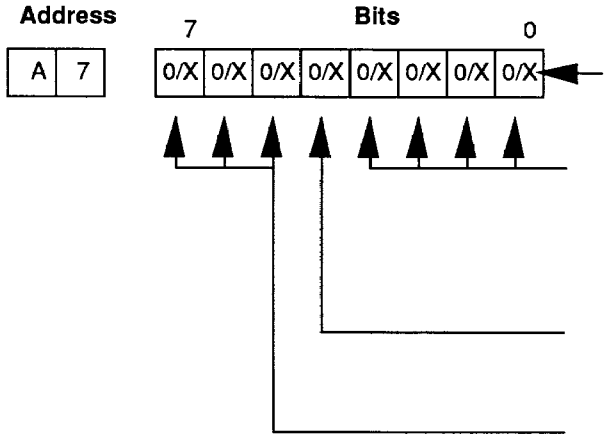


Note: This configuration can support Slave Block Transfers and Master and Slave Writepost Operation. This buffer configuration cannot support block transfers with DMA (refer to Section 10 for those implementations).

7

SAMPLE REGISTERS

Local Bus Timing Register



This field sets the minimum PAS* asserted time for all local bus cycles controlled by VIC. These include refresh, slave access, and DMA. PAS* minimum = (n + 2) 64 MHz clock periods where N is the binary value written to this field. Clock latency can add up to 1 additional 64MHz period to the width.

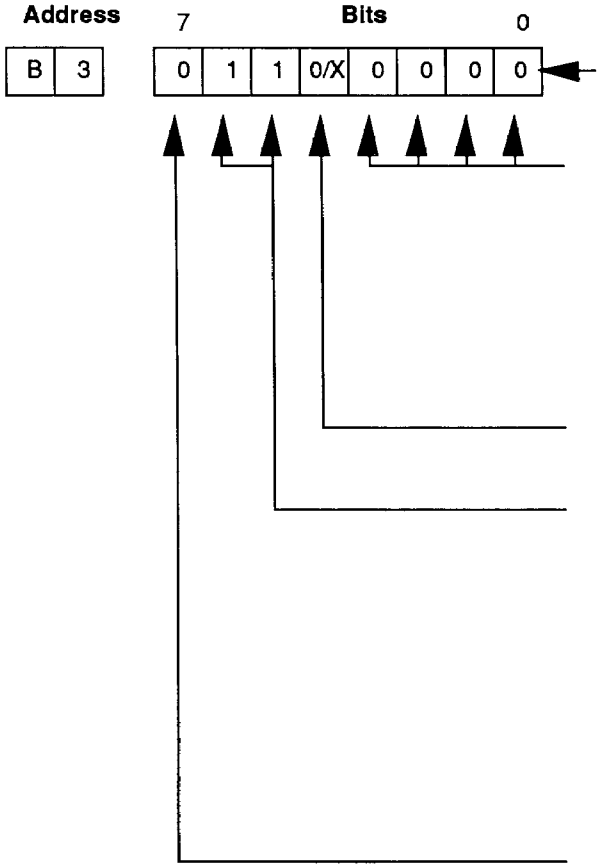
This bit selects the DS* high time. A 0 (zero) selects 1 64MHz period, a 1 (one) selects 2 64MHz periods. Clock latency can add up to 1 additional 64MHz period.

This field programs the PAS* high width. PAS* high = (N+1) 64 MHz clock periods where N is the binary value written to this field. Clock latency can add up to 1 additional 64MHz period to the width.

Register Function:

This register provides for control of local bus timing by providing programmable control of PAS* and DS* when these signals are driven from VIC. This register is unaffected by reset of the VIC. This feature could be used to preserve DRAM data during module resets.

Arbiter/Requester Configuration Register



Fairness Timer:

These bits program the fair request timeout period and control fairness mode according to the following table:

Bit 3	*Bit 2*	*Bit 1*	*Bit 0*	*Period/Mode*
0	0	0	0	0/Fairness disabled
1	1	1	1	Timeout disabled
All other patterns				2µs times the number

This bit enables DRAM refresh.

VME Bus Request Level:

These bits program the VMEbus request level used by VIC and determine which BRn * pin will be used to request the VMEbus:

Bit 6	*Bit 5*	
0	0	Request level BR0*
0	1	Request level BR1*
1	0	Request level BR2*
1	1	Request level BR3*

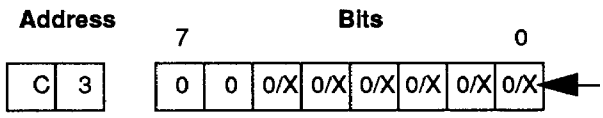
Priority/Round Robin:

This bit selects between priority and round robin arbitration. Setting this bit selects priority arbitration. This bit is relevant only when SCON* pin is asserted (strapped low).

Register Function:

The register configures the VIC VMEbus arbiter. On VIC reset, bits 0-3 and 7 are cleared while bits 5 and 6 are set.

Slave Select 0/Control Register 0



One of three slave block transfer modes is selected according to the following table:

Bit 1	*Bit 0*	*Mode*
0	0	no support for slave block transfer request
0	1	emulate non-block on local bus
1	0	accelerated block transfer
1	1	undefined

These bits specify the address size for above access as follows:

Bit 3	*Bit 2*	*Address Size*
0	0	A32
0	1	A24
1	0	A16
1	1	Use Address Modifier source register

When set, this bit enables D32 slave data size on SLSEL0*.

When set, this bit restricts SLSEL0* access to supervisory only. (checks AM2 bit)

These bits control and enable VIC's clock tick timer according to the following table (Note that LIRQ2* becomes timer output but retains its interrupt characteristics):

Bit 7	*Bit 6*	*Timer Mode*
0	0	Timer disabled
0	1	50Hz output on LIRQ2*
1	0	1000Hz output on LIRQ2*
1	1	100Hz output on LIRQ2*

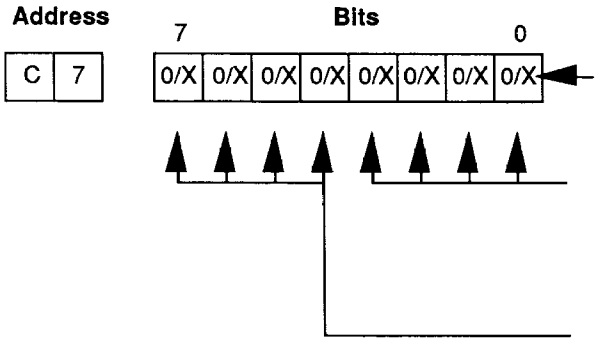
Register Function:

The register provides slave configuration control for slave access in response to SLSEL0*. The register also provides for enable and control of the VIC clock/timer. Bits (5-0) are unaffected by VIC or system reset.

9

SAMPLE REGISTERS *(continued)*

Slave Select 0/Control Register 1



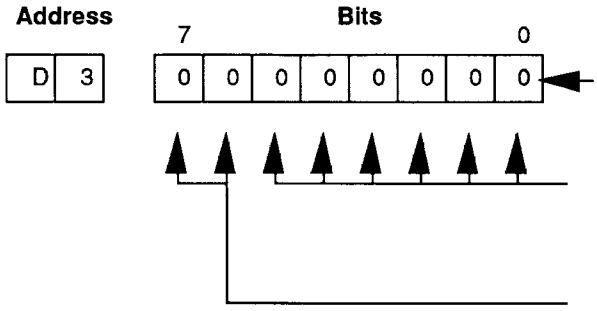
This field established the initial DSACKi* to DTACK* delay for slave access from SLSEL0*. Delay can be programmed to the following multiples of the 64 MHz clock period in ascending binary order: 0, 2, 2.5, 3, 3.5 ... 9.5. Clock latency may add up to one half of a 64MHz period to this delay.

This field establishes the DSACKi* to DTACK* delay for the second and subsequent slave block transfer cycles from the SLSEL0*. Delay can be programmed to the following multiples of the 64 MHz clock period in ascending binary order: 0, 2, 2.5, 3, 3.5 ... 9.5. Clock latency may add up to one half of a 64 MHz period to this delay.

Register Function:

This register provides for programming of slave access delay across the VMEbus/local bus interface for both single cycle slave access and slave block transfers. These programmed delays apply only to SLSEL0* access. This register is unaffected by VIC or system reset. This register is the default register for block transfer operations with local bus DMA.

Release Control Register



Burst Length of DMA:

This field programs the burst length for block transfers. Burst length is the number of VMEbus cycles.

This field defines the release protocol used by VIC in releasing the VMEbus once it has been captured. The following table defines the available modes:

Bit 7	*Bit 6*	*Release Mode*
0	0	ROR — release on request
0	1	RWD — release when done
1	0	ROC — release on BCLR*
1	1	BCAP — bus capture and hold

Register Function:

The register provides for programming VIC's VMEbus release behavior and the burst length for block transfers involving the VMEbus. This register is reset when VIC is reset.

10

SIGNAL LISTS AND PINOUTS

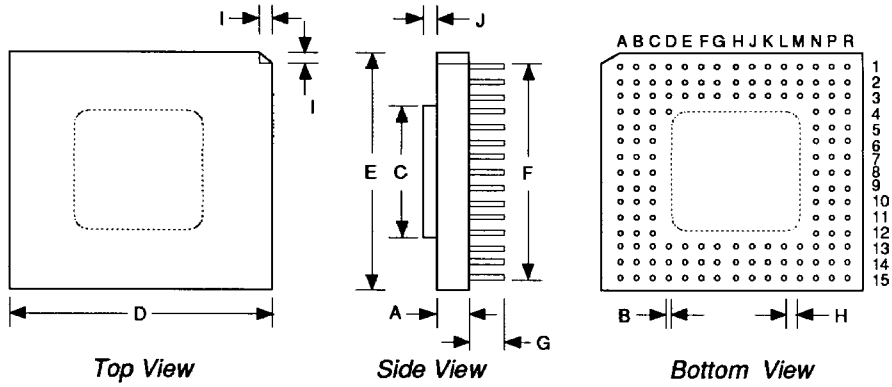
Bottom view

A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	
VSS11	IPL2*	IIACKO*	LIRQ2*	LIRQ5*	ASIZ1*	ASIZ0*	SLSEL1*	WORD*	FCIACK*	A02	A04	VCC2	VSS2	IRQ4*	1
LD6	BLT*	IPL1*	VCC1	LIRQ1*	LIRQ4*	LIRQ6*	ICFSEL*	MWB*	A01	A03	A05	A07	IRQ3*	IRQ7*	2
LD2	LD5	DEDLK*	IPL0*	LAEN	LIRQ3*	LIRQ7*	VSSCORE	SLSEL0*	VSS1	A06	IRQ1*	IRQ2*	IRQ6*	ACFAIL*	3
LD1	LD3	LD7	LOCATOR PIN									IRQ5*	VCC3	IACKOUT*	4
LA7	LD0	LD4										SYSFAIL*	SYSRESET*	DTACK*	5
LA3	LA5	LA6										IIACKIN*	IIACK*	AM0	6
LA2	LA4	VSS10										VSS3	AS*	AM1	7
LA1	LA0	VCC7										VSS4	AM2	AM3	8
CS*	DSACK1*	DS*										VCC4	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*	CLOCK64M	LADI	VSS9	VCCCORE	VSS8	VCC5	D00	BG1OUT*	BG2IN*	BG0IN*	BR3*	VSS5	13
SIZ1	IRESET*	LADO	LEDI	DDIR	LWDENIN*	DENO*	D06	D03	D01	VSS7	BG0OUT*	BG3IN*	BG1IN*	BCLR*	14
LBG*	ABEN*	VCC6	LEDO	UWDENIN*	SWDEN*	ISOBE*	D07	D05	D04	D02	BG3OUT*	BG2OUT*	SYSCLK	VSS6	15

PACKAGE DIMENSIONS

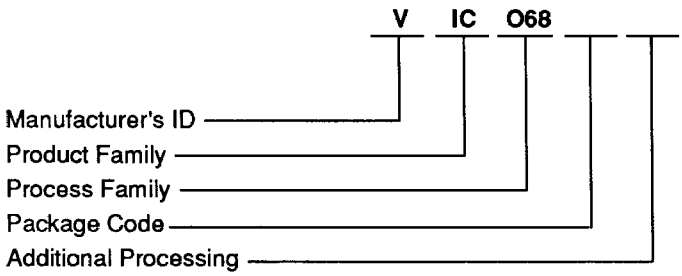
144-pin Pin Grid Array (PGA)				
	Inches		Millimeters	
Sym.	Min.	Max.	Min.	Max.
A	0.059	0.076	1.500	1.930
B	0.016	0.020	0.406	0.508
C	1.030	1.060	26.16	26.92
D	1.400	1.600	35.56	40.64
E	1.35	1.600	35.56	40.64
F	1.35	1.45	34.29	36.83
G	0.185	0.209	4.699	5.309
H	0.095	0.105	2.413	2.667
I	0.010	0.040	.254	1.016
J	0.060	0.075	1.524	1.905

Plastic, cavity up



Top View *Side View* *Bottom View*

ORDERING INFORMATION



ADDITIONAL PROCESSING

J = Screened to MIL-STD-883C Class B

PACKAGE CODE

PG = Plastic Pin Grid Array
DG = Ceramic Pin Grid Array

VTC Incorporated

2401 E. 86th Street
Bloomington, MN 55425
612/851-5200
800/352-6789
Telex 857113