fax id: 5708

Frequently Asked Questions about the VMEbus Products

The following questions are frequently asked by customers who are evaluating and using Cypress VMEbus Interface products. These answers will serve as an introduction for each topic. Separate application notes cover these topics in more complete detail.

Section I. Questions Regarding Reset

1. What are the requirements to reset the VIC at power-up?

To properly reset the VIC at power-up, it is required that the VIC see a falling edge on the $\overline{\text{IRESET}}$ signal after the following criteria have been met:

1. The input voltage has reached 5V.
2. The CLK64M clock input is operating within the required specifications.
3. All VMEbus signals are within VMEbus specifications.
4. Local input and three-state I/O signals are driven to a deasserted value (LD[7:0] and LA[7:0] may be left floating).

$\overline{\text{IPL0}}$ must be asserted no earlier than 16 ns (20 ns for military devices) after $\overline{\text{IRESET}}$ has been asserted. This will initiate a global reset. The minimum pulse width for $\overline{\text{IPL0}}$ is 50 ns. See section 12 of the *VIC068A User's Guide* for more details.

2. What is the best way to implement a power-up reset?

Best results have been obtained when the power-up reset is initiated through software during system boot. That is, dedicate two external register bits to be tied to the $\overline{\text{IRESET}}$ and $\overline{\text{IPL0}}$ signals. During system boot-up, have the processor write to these bits in a way that first asserts the $\overline{\text{IRESET}}$ signal, then asserts the $\overline{\text{IPL0}}$ signal, then negates the $\overline{\text{IPL0}}$ signal, and finally negates the $\overline{\text{IRESET}}$ signal. Since the processor must be operational before the VIC, this implies that the $\overline{\text{RESET}}$ output signal may not be used to reset the processor. Sample SPARC™ assembler code for this type of reset may be found in the application note "Software Considerations for the VIC64."

As the VIC must see a falling edge on $\overline{\text{IRESET}}$ when the system is stable (see question 1, above), an RC network should not be used to reset the VIC on power-up.

3. Can the VIC or the local module be remotely reset over the VMEbus?

The assertion of $\overline{\text{SYSRESET}}$ on the VMEbus will reset the internal circuitry and selected internal register bit fields on the VIC. This is referred to as a system reset because $\overline{\text{SYSRESET}}$ is typically used to reset all modules on the VMEbus.

If an individual module reset is desired (without resetting the entire system), ICR7 (Interprocessor Communication Register 7) bit 6 can be set. This will assert $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ from the VIC, which can be used to reset local bus devices on a specific module. However, when this bit is set, no external VMEbus masters can access the VIC, so provisions must be made to issue an $\overline{\text{IRESET}}$ from the local side. Asserting $\overline{\text{IRESET}}$ (for a minimum of 20 ns) will cause the VIC to initiate an internal reset. Upon being granted the local bus (or if no grant is asserted to the VIC within 1 ms, a timer will expire and the VIC will proceed as if it had been granted), the VIC will drive $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ for 200 ms intervals until $\overline{\text{IRESET}}$ is deasserted. When the VIC detects $\overline{\text{IRESET}}$ deasserted at the end of the 200 ms timeout period, it will deassert $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$, bringing the local module out of reset. Upon the assertion of $\overline{\text{IRESET}}$, the VIC will change the state of its internal registers. The internal registers must be reloaded.

For power-up reset, a global reset must be used (to ensure that all internal VIC registers are set to their default values). *See questions 1 and 2.*

4. Does the VIC drive the local bus when $\overline{\text{IRESET}}$ is asserted?

No. After $\overline{\text{IRESET}}$ is asserted, the VIC attempts to arbitrate for the local bus. If the VIC is granted the bus or a 1 ms timer expires, the VIC will assert $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$, deassert its local bus request, place all three-state outputs in high-Z, and begin a 200-ms timeout period. If $\overline{\text{IRESET}}$ is still asserted after 200 ms, additional 200-ms timeout periods follow until $\overline{\text{IRESET}}$ is deasserted.

Cypress Semiconductor Corporation • 3901 North First Street • San Jose • CA 95134 • 408-943-2600
December 10, 1995

Section II. Questions Regarding Interrupts

5. Can the VIC queue up multiple interrupts with the same IPL value?

No. The VIC will queue all pending interrupts that are on different levels. If back-to-back interrupts are required on the same level, the first interrupt will have to be handled before the second interrupt is recognized. It is legal for the VIC to continue to drive the IPL lines to the same level if back-to-back local interrupts are requested on the same level, but the interrupts must be requested sequentially.

6. Is there a way to check the level of VMEbus interrupts in the VIC?

If the interrupt was generated by writing to the VIRSR (VMEbus Interrupt Request/Status Register), the level can be checked by reading the VIRSR. Otherwise, the only way to check the level is to allow the local processor to perform the interrupt acknowledge cycle. The proper vector will be generated, which should allow software to determine the interrupt level by jumping to the specific interrupt handler. The vector can also be seen with a logic analyzer during the interrupt acknowledge cycle.

7. What is the minimum pulse width for the LIRQ signals?

One CLK64M clock period. The LIRQ lines are internally registered by the VIC. Therefore, if the local interrupt request lines are asserted for at least one 64-MHz clock period, the VIC is guaranteed to sample and recognize the asserted request lines on a CLK64M clock edge.

8. When does the VIC latch the IPL lines?

$\overline{\text{IPL2}}$, $\overline{\text{IPL1}}$, and $\overline{\text{IPL0}}$ are the local priority encoded interrupt request signals. They are used to interrupt the local processor. These signals emulate the Motorola 68K interrupt mechanism. The IPL lines are latched on the assertion of the $\overline{\text{FCIACK}}$ signal. $\overline{\text{FCIACK}}$ should be asserted by the processor to tell the VIC that an interrupt is being acknowledged. Once the VIC detects the assertion of $\overline{\text{FCIACK}}$, it samples LA[3:1] to determine whether the interrupt acknowledge is for the VIC's pending interrupt. If the acknowledge was intended for the VIC, it will either pass the acknowledge to the VMEbus (for VMEbus initiated interrupts) or provide the appropriate acknowledge signals to the local bus (for local bus initiated interrupts). The IPL lines can change after the $\overline{\text{FCIACK}}$ signal is deasserted. The assertion of $\overline{\text{DSACK0}}$ or $\overline{\text{DSACK1}}$ by the VIC indicates that the acknowledge matches the interrupt level that the VIC is currently requesting.

Section III. Questions Regarding Register Operations

9. Can the VIC registers be programmed over the VMEbus?

VIC registers (other than the ICF registers) cannot be directly programmed over the VMEbus. They can be accessed, however, by having the address decoder drive CS to the VIC.

Section IV. Questions Regarding Arbitration

10. How must the local bus arbiter operate?

The VIC (or any other local bus master) will assert its own $\overline{\text{LBR}}$ whenever it needs to access the local bus. The arbiter must assert a specific $\overline{\text{LBG}}$ for one master allowing the access to occur. The VIC will maintain its $\overline{\text{LBR}}$ until it no longer wants the local bus. It is up to the system designer to pick an arbitration scheme (assigning priorities to each master, insuring that no master will be "starved" off of the bus, etc.). Arbiters must also monitor the $\overline{\text{DEDLK}}$ signal to prioritize the local bus grant to the VIC during deadlock situations.

Once the VIC has been granted the local bus, it is important that the $\overline{\text{LBG}}$ signal to the VIC not be removed until its $\overline{\text{LBR}}$ is deasserted. The VIC will keep its $\overline{\text{LBR}}$ asserted through its entire cycle.

11. Can $\overline{\text{LBG}}$ on the VIC be tied HIGH?

Only if the designer can insure that the VIC will never be the local bus master. The VIC requires local bus mastership when there are VME slave accesses, VME block transfers, or VME DRAM refreshes performed on the board.

12. Does the VIC support early release of $\overline{\text{BBSY}}$?

Yes. If the Release When Done release mode has been selected, the VIC will deassert $\overline{\text{BBSY}}$ upon the last assertion of $\overline{\text{AS}}$.

Section V. Questions Regarding Deadlock

13. When is $\overline{\text{DEDLK}}$ asserted?

When the $\overline{\text{MWB}}$ signal or $\overline{\text{FCIACK}}$ and a valid slave select occur at the same time, the VIC will assert $\overline{\text{DEDLK}}$ to force the processor to remove $\overline{\text{MWB}}$ or $\overline{\text{FCIACK}}$ and retry the transaction later. The VIC will not detect a deadlock situation when $\overline{\text{CS}}$ or $\overline{\text{IFCSEL}}$ is asserted (a VIC register access) at the same time as a valid slave transaction to the VIC.

14. How does the system recover from a deadlock?

If a deadlock occurs, the VIC will assert the $\overline{\text{DEDLK}}$ signal (or a combination of $\overline{\text{DEDLK}}$ and $\overline{\text{LBERR}}$ and/or $\overline{\text{HALT}}$, which can be programmed to occur on deadlocks). $\overline{\text{DEDLK}}$ must go to the arbiter to prioritize the local bus grant to the VIC (so it can perform the slave access). During a deadlock the processor will not have access to the VME bus as a master until the slave transaction has been completed. All other local transactions will not be affected by the deadlock.

15. Can deadlocks be disallowed?

If the system designer can guarantee that no master will try to access local memory on a VMEbus board, the board does not have to support deadlocks. Otherwise, they cannot be disallowed.

Section VI. Questions Regarding Block Transfers

16. Can block transfers be interrupted or aborted?

The only way to abort a block transfer is by asserting $\overline{\text{LBERR}}$. However, when $\overline{\text{LBERR}}$ is asserted, the status will be saved (bits in the DMASR, etc.). Also the assertion of $\overline{\text{LBERR}}$ will cause the VIC to assert VMEbus $\overline{\text{BERR}}$, which can have severe system ramifications. If block transfers are taking too much local bus/VMEbus bandwidth, the block size should be shortened or the block should be broken up using interleaving. Breaking up the block is a cleaner solution.

17. What is the maximum block transfer?

The VMEbus specification prohibits the crossing of 256-byte boundaries during block transfers (2K-byte boundaries for VME64). The VIC allows for larger block transfers by deasserting AS, incrementing the address, and reasserting AS without relinquishing the VMEbus whenever a boundary is crossed. The boundary crossing feature is enabled by setting bit 2 in the BTDR, Block Transfer Definition Register (bit 7 for the VIC64 with 2K-byte boundaries).

Without using CY7C964s or the VAC with the VIC, the maximum block transfer is 256 bytes (2^8). This is because the VIC only has direct control over the lower order VME address lines (A[7:1]).

If a VAC or CY7C964s are used in conjunction with the VIC068, 64K bytes (2^{16}) can be transferred in a block. For the VIC64, the maximum block size is 16M bytes (2^{24}). The increase in block size is due to the fact that the VAC or CY7C964s give complete access to the 32 VME address signals so the block address can be incremented past A7. The 64K-byte VIC and 16M byte VIC64 constraints are due to the fact that there are two eight-bit registers in the VIC068 (BTLR0 and BTLR1) and three eight-bit registers in the VIC64 (BTLR0, BTLR1, and BTLR2) to define and control the block transfer length.

18. Can the VIC perform D8 block transfers?

No. The least significant bit of BTLR0 should be cleared. If the least significant bit is set, the block transfer length is ignored and only one burst is performed.

Section VII. Questions Regarding Slave Operations

19. Can the VIC be used to implement a slave-only interface without using a microprocessor?

This can be done, but external logic must be provided to load the VIC's internal registers. Please see the Application Note entitled "Using VIC068A on a Board Without a Microprocessor," *Cypress Applications Handbook*, 1993. Cypress also offers slave-only interface chips, CY7C960 and CY7C961.

20. Can SLSEL0 and SLSEL1 be programmed to respond to more than one address space each?

No. Each slave select signals can only respond to one address space at a time.

Section VIII. Questions Regarding Modeling/Schematic Capture

21. Are schematic capture libraries available for the VIC?

A VIC schematic in OrCAD is available on the Cypress BBS (408-934-2954).

22. Are simulation libraries available for the VIC?

Verilog models are available for the VIC068A, VIC64, VAC068A, and CY7C964. Verilog behavioral models of standard VMEbus transactions are available as well. They work with Cadence's Verilog package. Contact your local Cypress Field Applications Engineer to obtain them.

Section IX. Questions Regarding Electrical Characteristics

23. What are the thermal characteristics for Cypresses VMEbus products?

Package	Theta JC (Degrees C/Watt)	Theta JA (Degrees C/Watt)	Description
B144	11.0	38.0	144-Pin Plastic PGA
G145	4.0	24.0	144-Pin Ceramic PGA
N160	13.0	34.3	160-Pin PQFP
A144	7.2	45.1	144-Pin TQFP
U162	6.5	26.0	160-Pin CQFP
N65	17.7	81.3	64-Pin TQFP 14mm
A64	18.2	108.0	64-Pin TQFP 10mm
U65	3.0	80.7	64-Pin CQFP
G68	4.0	28.4	68-Pin Ceramic PGA

24. What is the maximum power consumption for the VIC?

The VIC and the VAC consume 0.75W max each. The I_{CC} is rated at 150 mA max. The parts typically consume 50 mA.

Section X. Miscellaneous Questions

25. Is there a test mode/pin to three-state all of the VIC's outputs for testing purposes?

No.

26. Can all of the VIC's inputs and outputs be treated as synchronous signals clocked off of CLK64M?

No. All inputs and outputs should be treated as asynchronous. There are internal synchronizers to sync the external signals to the CLK64M clk for the purpose of running the VIC's internal state machines synchronously, but there are no guaranteed timing relationships between any of the signals and CLK64M.

27. Does the VIC have internal clamping diodes?

The signals are clamped to 5V (to help prevent overshoot problems). There are no clamping diodes to GND.

28. What values of capacitors are recommended for decoupling?

0.10 µF for AC bypass and 100 pF (or 470 pF) for high frequency decoupling. Four of each is recommended. They should be laid out as close to the V_{CC} pins as possible with wide traces (if possible) to eliminate some of the inductive effects.

29. What kind of throughput can be expected from the VIC?

The design group was able to achieve 61.6 Mbytes per second using the VIC64, 30 Mbytes per second using the VIC068. Over 70 Mbytes per second is possible using the VIC64. This maximum is usually dependent on system constraints rather than interface components.

30. What is the die size for the VIC068?

315x300 mils for the VIC068A and VIC64, 313x300 mils for the VAC068A, and 144x133 mils for the CY7C964.

31. Using the VIC with CY7C964s (or the VAC), is there any way to avoid violating the 2-inch VMEbus rule?

Users should consider this rule as a guideline. The rule is nearly impossible to meet using any standard VMEbus interface chipset. Traces from the VIC/CY7C964s/VAC to the VMEbus connectors should be kept as short as possible.

32. How many CY7C964s should be used with the VIC?

Each CY7C964 controls 8 bits of both address and data. The VIC068A and VIC64 also control 8 bits of address and data. Users can determine how many CY7C964s are needed to complete their interface by determining which address and data transactions will be supported. An A32/D32 interface would require three CY7C964s. See the *VIC64/CY7C964 Design Notes* from Cypress Semiconductor for more information on the CY7C964 and how to connect it to the VIC.

33. How many gates are in the VIC068A/VAC068A?

19,435 in the VIC068A; 21,250 in the VIC64; 18,106 in the VAC06A; 3000 in the CY6C964. The transistor counts are as follows: 80,000 for the VIC068A, 85,000 for the VIC64, 75,000 for the VAC068A, and 12,000 for the CY7C964.

34. What is the capacitive loading on the VIC signal lines?

5 pF on inputs. 7 pF on outputs. 13 pF on bidirectional signals.

35. How many words can be write posted to the VIC from the local and the VMEbus side?

One longword can be write posted from either side.

36. Which VIC signals have metastability protection?

Metastability is a problem with all asynchronous, clocked designs. If a valid level is not reached on the input to a clocked element (flip-flop, etc.) within the specified set-up and hold window, the condition called “metastability” can occur. The output of the clocked element is unpredictable. It may be driven to a valid output level or even oscillate. Eventually the output will settle to a valid level, but the settling time may also be unpredictable. There are several ways to combat metastability problems. One of the most common techniques involves “double clocking” the input. Two clocked elements are placed, in series, in the signal path. Even if the first clocked element goes metastable, the odds are good that the output will have settled to a valid state before the set-up and hold window of the second element is reached.

All of the VMEbus strobe inputs to the VIC are metastability-hardened and carry with them 2–3 CLK64M cycles of synchronization delay. $\overline{\text{DSi}}$, $\overline{\text{DTACK}}$, and $\overline{\text{BERR}}$ are also metastability-hardened. $\overline{\text{AS}}$ has both an asynchronous path and a metastability-protected path. When performing slave transfers, the asynchronous path is used.

The VME data bus, address bus, AM5–0, LWORD, WRITE, and all of the local bus signals are not metastability-hardened.

37. Is there any example “C” code available for programming the VIC?

Yes. A file named SAMPCODE.EXE is available on the Cypress BBS (408) 943-2954. This is a self-extracting file.