3.4

Programming the CY7C960

The VMEbus board that is designed to use the CY7C960 has what might be called a personality. This encompasses such things as DRAM (speed and amount), I/O circuitry, SRAM, and other matters under the control of the board designer. To a large extent, this personality has to be known to the CY7C960. Some portion of this personality is determined when the board is developed, and is not changed during the board's lifetime. Other aspects of the personality could conceivably change from time to time, such as the address to which the board responds, or the type of VMEbus transaction that it recognizes. All of this information must be provided to the CY7C960 each time the power is cycled or SYSRESET* is asserted. The designer might choose to provide switches that convey part of the personality, or he may rely upon some other VMEbus board to download the information. The CY7C960 has been designed to support these diverse programming requirements.

First consider the address on the VMEbus to which the board responds. This may be a very simple application which has only one address range within which to respond; or it may be a complex address map involving AM Code discrimination, and multiple address ranges. In either case, the external address comparator circuitry needs to be programmed with the slave address ranges in which the application will respond. The CY7C960 supports the loading of this address comparator circuitry from either local circuitry, or from the VMEbus.

For configuration of the other parameters of the user's application a serial bit stream is used. The CY7C960 contains approximately 380 programmable bits, each of which must be set to the right value for the user's application in order to ensure correct functionality (see *Figure 3−9*). There are two methods that can accomplish this: the VMEbus can be used to pass parameters from a VMEbus master to the CY7C960, or a local serial PROM can be used.

Thus in summary, the slave address map can be obtained either from local circuitry or the VMEbus, and the configuration data can likewise be obtained either from the VMEbus or from local circuitry.

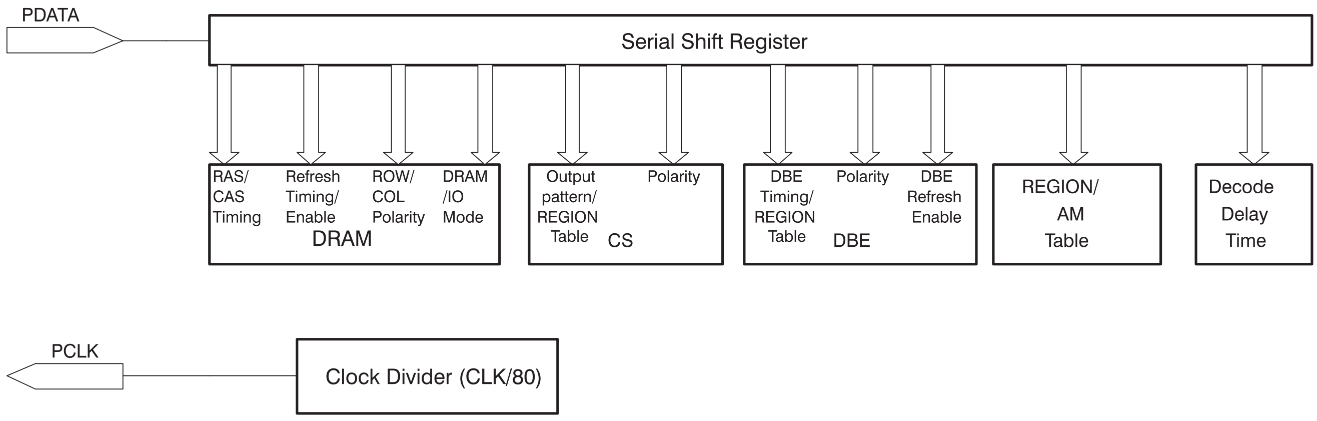

Figure 3−9. CY7C960 Initialization Block Diagram

The CY7C960 is designed for use with the CY7C964 device. The CY7C964's address and mask registers can be used during the configuration process and the CY7C960 provides timing and control signals during initialization that are intended to load these address and mask registers. If the target application does not use the CY7C964's registers as part of the slave address decoder, these signals can be used by user-defined circuitry to configure the address decoder. Refer to Section 4, The CY7C964 Bus Interface Logic Circuit, for more information regarding the signals used to configure the address and mask registers.

The LAEN signal is programmed during the power-on reset period. This signal is used by the CY7C960 to control the external three-state buffer connected to LA[7:0], enabling the CY7C960 itself to drive LA[7:0] at certain times. If using the CY7C964, LAEN is required to be active High; other devices may require LAEN to be active Low. During the power-on reset period the LAEN pin is sampled to determine whether it is pulled up externally, and if so the LAEN output becomes an active Low signal.

3.4.1 Configuration Bit Stream

The 380 bits that make up the configuration data are provided to the CY7C960 by either a series of VMEbus transactions, or by a local serial device such as a serial PROM. The various fields within the bit stream are used to determine the behavior of the CY7C960, and hence the application, after configuration is complete. For the duration of the configuration process two pins, LA[2:1], assume the roles of serial clock and serial data signals. Another pin, PREN*, is used to enable the external serial device during configuration.

The CY7C960 expects the serial data to arrive on pin LA[2] so if the VMEbus configuration method is used, it is the responsibility of the remote VMEbus master to encode the desired configuration stream into 380 VMEbus cycles whose VME address A[2] is controlled by the desired configuration data bit. Likewise, if using a local serial device, the data output from the device shall be connected to LA[2].

To assist in the generation of the correct configuration data file, there is a configuration program available, called WinSvic, that runs on a PC using Windows. This program has extensive help menus which can be used to better understand the configuration bit stream. See section 3.4.6 for more details on this development tool.

For applications using a local serial device the serial clock may be generated by the CY7C960, or it may be provided to the CY7C960 by external circuitry.

The CY7C960 determines which method of configuration is desired, and whether to drive a serial clock or not, as part of the power-on process. This is described in the next section.

3.4.2 Operation at Power-On or Reset

The CY7C960 contains a power-on reset circuit. When power is applied to the device this circuit causes the internal state to be reset. All outputs become three-state until the first rising edge of the clock input is recognized after the internal power-on reset circuit terminates. (The time period of the on-chip power-on-reset circuit depends upon the characteristics of the power supply, but is guaranteed by design to be less than the time period of the rising power ramp.) Also, if a Low level is applied to the SYSRESET* input the CY7C960 immediately three-states its output drivers, resets its internal state and waits for the first clock edge before taking any action. SYSRESET* does not have to be High for the CY7C960 to commence the configuration operation, but must be High for normal operation to proceed after configuration.

PREN* is asserted Low four clocks after the assertion of SYSRESET*, or after a power-on reset. If an external serial device is connected, the first bit must be programmed to be zero. If the VMEbus is to be used, an external pull-up must provide a High level. The PDATA signal (LA[2]) is sensed 40 CLK periods after the assertion of PREN*. If PDATA is sensed to be High, PREN* is driven High one CLK period later.

Also, the CY7C960 senses the level of two pins: LAEN and PCLK (LA[1]). LAEN is sensed to determine whether the signal is desired to be high or low true. PCLK is sensed to determine whether the pin is to be an input or an output. The point at which these signals are sensed is 40 CLK periods after the assertion of PREN*. LAEN has an internal pull-down of approximately 470 kohms. PCLK has no internal resistor. Whichever level is sensed on LAEN is driven from the pin one clock period after it has been sensed. If PCLK is determined to be pulled High, then it is driven Low by the CY7C960 one CLK period after it was sensed: subsequently the CY7C960 will provide a clock output (of frequency CLK÷80). If PCLK is pulled Low, then external circuitry must provide the serial clock signal and must not drive the PCLK input High until at least 40 CLK periods following the falling edge of PREN*.

If the CY7C960 is providing the PCLK signal, the period of each half-cycle is 40 CLK periods, leading to a PCLK frequency of 1 MHz (CLK÷80). PDATA is required to be stable at each falling edge on PCLK, when the PDATA is clocked into the serial configuration register inside the CY7C960. Similarly, if the PCLK signal is provided externally, the PDATA input must be stable at the falling edge of PCLK. The frequency applied to the PCLK pin from an external source does not have to be synchronous to the CLK input as the PCLK input is synchronized internally. The minimum half cycle time for the PCLK input is 50 ns: there is no maximum time as the design is static.

If SYSRESET* is driven Low by a device on the VMEbus, then the CY7C960 reacts in a manner similar to the power-on reset: all outputs become three-state until the first rising edge of the clock (following the falling edge of SYSRESET*), then the PREN* signal is driven Low and the initialization cycle commences. Note that the configuration cycle commences prior to the SYSRESET* signal going inactive: this allows the CY7C960 to complete the configuration cycle within the 200-ms VMEbus specification for SYSRESET* minimum active time if the Serial PROM method is used.

If a front panel reset switch is to be used, then external circuitry combines the two sources of reset (VMEbus SYSRESET* and switch reset), and connects the result to the SYSRESET* pin. The behavior of the CY7C960 is, of course, identical in the two different methods of applying reset. (Note, however, that if a manual reset is performed while VMEbus activity is taking place, the CY7C960 will ignore any VMEbus transactions to any and all Regions until the configuration has been loaded. This may cause the system timer to timeout, and the transaction will cause a BERR*.)

Following the reset and configuration sequence, the CY7C960 performs a DRAM refresh operation (if enabled to do so in the configuration sequence). This is illustrated in *Figure 3−10*. This allows a "warm" reset whereby data in DRAM is preserved across reset operations that do not involve power-down. The length of the refresh depends upon the DRAM configuration selected during initialization: the possible values are from zero bursts to 128 bursts, each burst being four CAS*-before-RAS* cycles whose cycle timing is taken from the configured values. During the long refresh operation, the CY7C960 monitors the VMEbus: if a transaction is directed to the CY7C960 which does not require the use of the DRAM circuitry (for example, a transaction addressed to a Region where DRAM is not enabled), the transaction is completed normally. If the DRAM access is required, the cycle cannot complete until the refresh burst is ended. The detail of the behavior depends upon the transaction: a read operation of any sort cannot commence until refresh is complete; for write operations, the first write cycle is DTACK'ed because the CY7C960 always posts write data, but subsequent write cycles will not be acknowledged until the DRAM refresh burst has completed and the posted data has been written to DRAM. In all cases, the CY7C960 completes the transaction correctly.

Following the power-on or reset configuration sequence, and the refresh burst (if enabled), the CY7C960 will respond to any VMEbus transfers that are enabled by the configuration.

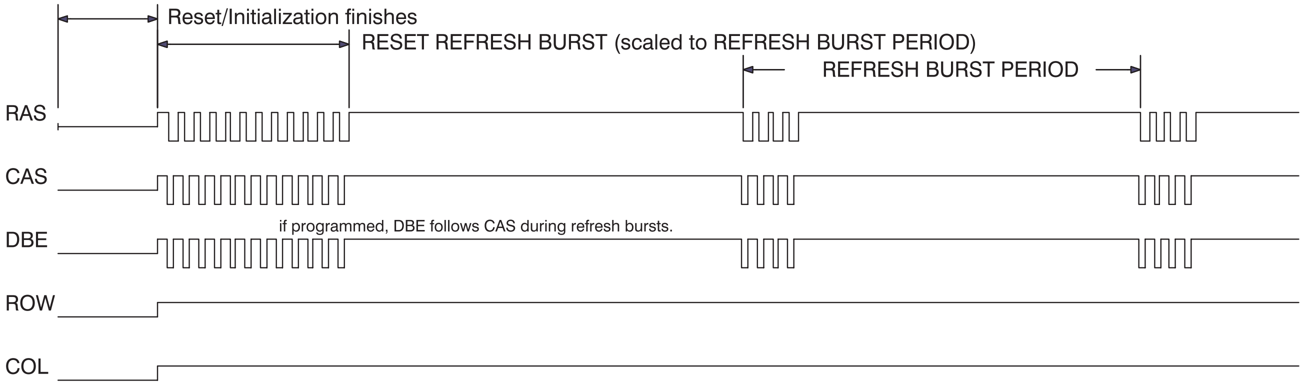
Reset/Initialization finishes
RESET REFRESH BURST (scaled to REFRESH BURST PERIOD)
REFRESH BURST PERIOD
RAS
CAS
DBE
if programmed, DBE follows CAS during refresh bursts.
ROW
COL

Figure 3−10. Refresh Burst after Initialization

3.4.3 VMEbus Method

The VMEbus initialization method is automatically invoked by the CY7C960 if its PDATA pin is at logic 1 at power on or during a system reset. (This may be easily accomplished by using a pull-up resistor for the local address bus.) If PDATA is indeed High at this time, then after SYSRESET* is deasserted on the VME backplane, CY7C960 asserts its interrupt request line. This interrupt should be connected to IRQ2* of the VMEbus backplane as the initialization interrupter inside CY7C960 is hard-coded to respond to that level. (After configuration is complete, the user-programmed interrupt level is used, but this must match the actual hardware level connected on the backplane. Therefore if the user wishes to use a level other than 2, external circuitry is necessary to switch the IRQ output pin to the different VMEbus IRQ signal after initialization.) See section 3.5.6.

The VMEbus Interrupt Handler must provide a level 2 interrupt acknowledge cycle: the interrupt acknowledge daisy-chain will ensure that the CY7C960 closest to the handler will respond to the IACK cycle. When the IACK cycle is detected, CY7C960 asserts its LDEN pin, which provides a signal for enabling initialization interrupt status ID (user defined) onto the local data bus. This status comes from user-defined external circuitry such as jumpers or latches. The CY7C964s are controlled by the CY7C960, and hence pass the vector to the VMEbus. CY7C960 will acknowledge the next VMEbus master write qualified by the A16 (supervisory or nonprivileged) or CR/CSR AM code and write the VMEbus data to the address registers of the CY7C964s. Thus, the CY7C964s are now primed to react to that VMEbus address, unmasked, as the action of writing the compare register clears the mask register in the CY7C964. The interrupt acknowledge cycle and the following single master write cycle must be indivisible for guaranteeing system integrity (the board has yet to be positioned in the VMEbus slave address map, and therefore will respond to ANY A16 or CR/CSR write cycle). If this indivisibility is not possible, then the local method of configuration must be used.

From this point, CY7C960 expects a series of A16 (or CR/CSR) master write transactions to be written to the address just programmed. As the board address has now been set in A16 or CR/CSR space, there is no requirement for indivisibility. (There is a requirement that the AM code for each write transaction be the same as that used in the first transaction: start with A16, stay with A16; or start with CR/CSR, stay with CR/CSR.) Each transaction carries a single configuration bit (just like the serial PROM method). The bit must be encoded on the VMEbus A[2] signal. CY7C960 counts the transactions. After the last transaction of the serial stream is acknowledged, two more transactions are required, with the same address and AM code. The data from these transactions are loaded into the compare and mask registers of the CY7C964s to complete the initialization transaction sequence: the signals that control the loading of the CY7C964 registers are LDS and STROBE (driven by the CY7C960), and MWB* and BLT*, (pulled High).

The REGION[3:0] inputs to the CY7C960 should be driven by the user's external slave address map decoder. For the very first VMEbus write cycle following the IACK cycle, the REGION inputs are ignored. Since the write cycle is not qualified with address (i.e., REGION), it must immediately follow the IACK cycle. For subsequent configuration cycles the REGION inputs must be driven with "xx00" if the CR/CSR AM Code is used; and with "xxx0" (x = don't-care) if one of the two A16 AM Codes is used. Note that this system allows the user to connect the VCOMP (compare) outputs from the appropriate CY7C964 directly to the related REGION input as a simple configuration address map decoder. If the user requires a more complex address map, the configuration requirement is easily accomplished.

It should be understood that certain device behavior is not available during the configuration sequence. Obviously, no VMEbus slave operations can take place other than the configuration sequence itself. The CY7C960 does not provide AM Codes from the LA[7:2] pins until after the configuration sequence has completed as the bit that enables this operation is embedded in the configuration sequence. Similarly, the first IACK cycle is self timed because the bit which enables IACK cycles to be locally handshaked is embedded in the configuration data.

Figure 3−11 shows the timing of the start of the VMEbus initialization sequence, and *Figure 3−12* shows the end.

In summary, the sequence starts by setting the address of the CR/CSR space, then the CR/CSR is used to load the CY7C960, and finally the board's Slave Address is loaded. CY7C960 exits initialization mode and, after the refresh burst if so enabled, is ready for service. (Note that this initialization sequence is compliant with the Auto Slot ID utility of the VME64 specification.)

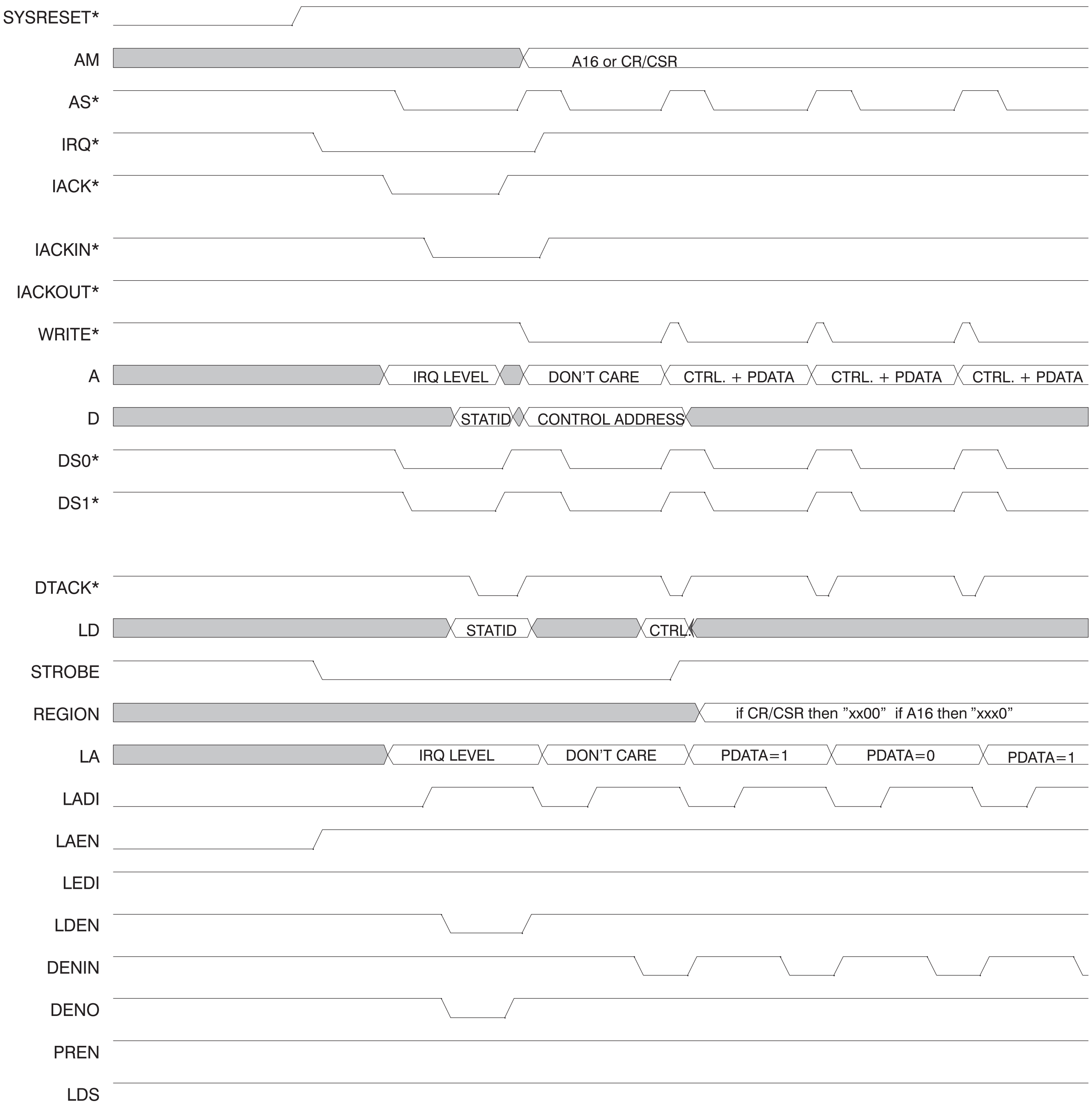

Figure 3−11. VMEbus Initialization Start

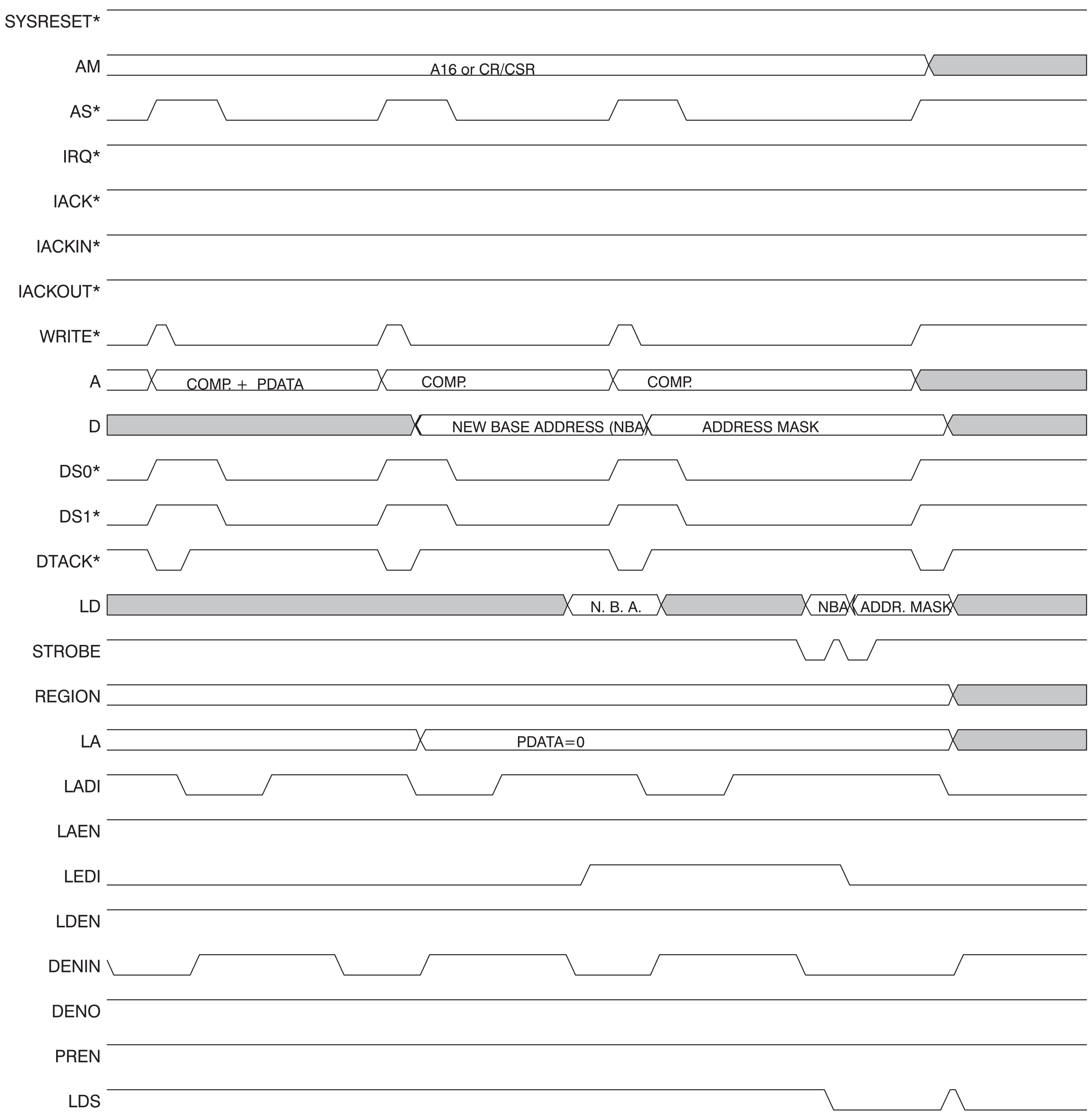

Figure 3−12. VMEbus Initialization End

3.4.4 Serial PROM Method

Following the assertion of SYSRESET*, or a power-on reset, CY7C960 drives PREN* Low and, after 40 CLK periods, senses the state of PDATA and PCLK. If the PDATA pin is Low, the serial method of configuration is used. The PCLK pin is sampled at this same point to determine clock direction. If PCLK is High (as controlled by an external resistor), CY7C960 will drive PCLK to strobe serial data into the CY7C960 from a serial PROM. If PCLK is sensed Low (as controlled by an external resistor), PCLK becomes the clock input of the CY7C960 serial program register. This option is to allow for programming the CY7C960 from a local microprocessor.

Once serial configuration has commenced, it continues for a predetermined number of clock pulses on the PCLK pin. If PCLK is sourced by the CY7C960, the clock frequency is CLK divided by 80 (maximum rate of 1 MHz), allowing industry-standard serial PROMS to be used. If PCLK is an input, the CY7C960 expects the PCLK signal to begin in the Low state, then toggle at a rate not to exceed 10 MHz. Regardless of PCLK direction, the rising edge on PCLK is assumed to clock the external serial device, and the falling edge is used by the CY7C960 to sample the PDATA input into the internal configuration data stream. PDATA is taken by the CY7C960 one CLK period after the High-to-Low transition of PCLK. When the correct number of data bits have been sampled by the CY7C960, PREN* is driven High. This PREN* transition can be used to disable external clock circuitry.

Once the entire stream of configuration data has been read into the CY7C960, the CY7C964s (or the external slave address decoders) are now to be configured. If the last bit of the serial bit stream was a 0 (Low), the CY7C960 assumes that the CY7C964s are to be configured using local resources. Therefore, the user-defined address and mask data (from latches or jumpers) are enabled onto the local bus by a combination of the LDEN*, STROBE, and LDS signals. LDEN* is driven Low when the CY7C960 detects that the serial PROM data has been loaded, and then LDS selects first the address register (High) then the mask register (Low). The CY7C960 controls the signal, STROBE, to ensure correct operation. Following the register load operation, LDEN* is driven High to disable the user’s latches.

The CY7C960 performs the refresh burst, if so enabled, and is ready for service. The serial configuration portion of this operation takes approximately 380 μsec if the CY7C960 is providing the PCLK signal, and hence can be completed within the period of a normal VMEbus

SYSRESET*. If the configuration sequence is invoked from a manual reset, it should be noted that the CY7C960 ignores any VMEbus activity until the entire configuration sequence is completed. VMEbus response during the long refresh burst is as described above.

Figure 3−13 shows the start of the Serial programming sequence, and *Figure 3−14* shows the end. SYSRESET* is Low as the sequence starts. This illustrates that serial programming can be completed before the VMEbus SYSRESET* period (200 ms) ends. There is no actual requirement that SYSRESET* be Low.

The first bit of PROM data after PREN* goes Low is 0: subsequent bits contain appropriate programming information. After the final bit of PROM Data has been read as a 0, the CY7C964s are programmed. First LDEN* and STROBE go Low, then STROBE goes High to load the Address Compare Register. While LDEN* remains Low, one clock after STROBE goes High, LDS is driven Low to select the Mask Data. A second STROBE pulse is issued to complete the process. The pulsewidth of STROBE assertion is 40 CLKs High/40 CLKs Low. See Section 4, The CY7C964 Bus Interface Logic Circuit, if using CY7C964s.

The CY7C960 board is now ready for service, once SYSRESET* goes High.

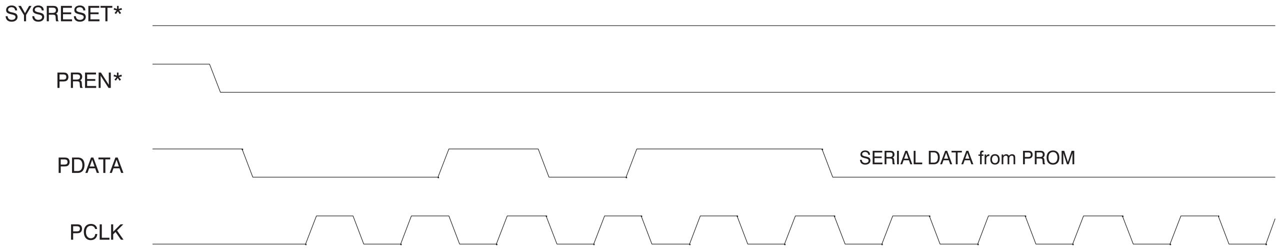

Figure 3−13. Serial Method Start

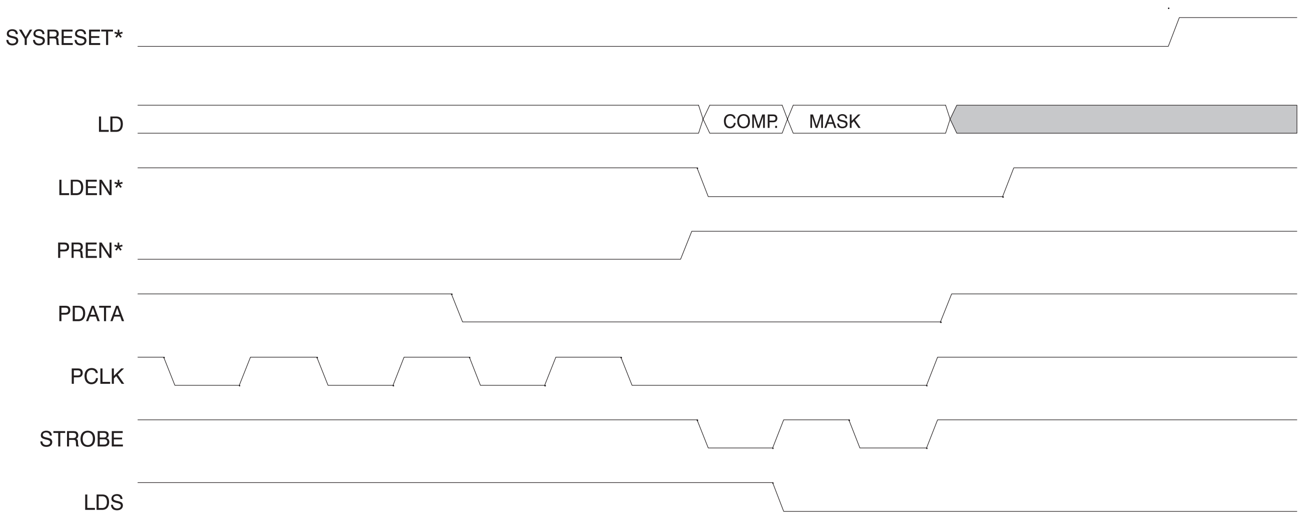

Figure 3−14. Serial Method End

3.4.5 Combination Method

The third method of initializing the CY7C960 is simply a combination of the above two methods. The serial PROM is used to load the configuration data stream, then VMEbus transactions are used to load the address and mask registers of the CY7C964s. The serial device operates as described above, except that the final bit is programmed to 1. This is a flag to the CY7C960 to use the VMEbus for the CY7C964 configuration. First the CY7C960 drives PREN* High and asserts the IRQ* output, which must be connected to the VMEbus IRQ line designated in the configuration data stream that was just loaded. This interrupt request must be interpreted by the interrupt handler in a similar manner to that described earlier for the VMEbus method: the interrupt acknowledge cycle causes CY7C960 to enable the initialization status vector using LDEN*. CY7C960 then interprets the next A16 (or CR/CSR) write transaction to load the CY7C964 address register with the control address. Then it expects the next two A16 (or CR/CSR) write transactions (to the address just loaded) to be the address and mask data for the CY7C964s. See section 3.5.6.

Following receipt of these three transactions the CY7C960 initialization is complete.

Regarding the interrupt level used in this combination method: though any level can be used, if it is desired to be compliant with auto slot ID level 2 should be used.

Figure 3−15 shows the timing of the signals involved in the CY7C964 Address and Mask register programming that occurs at the end of the serial load sequence. Following the re-

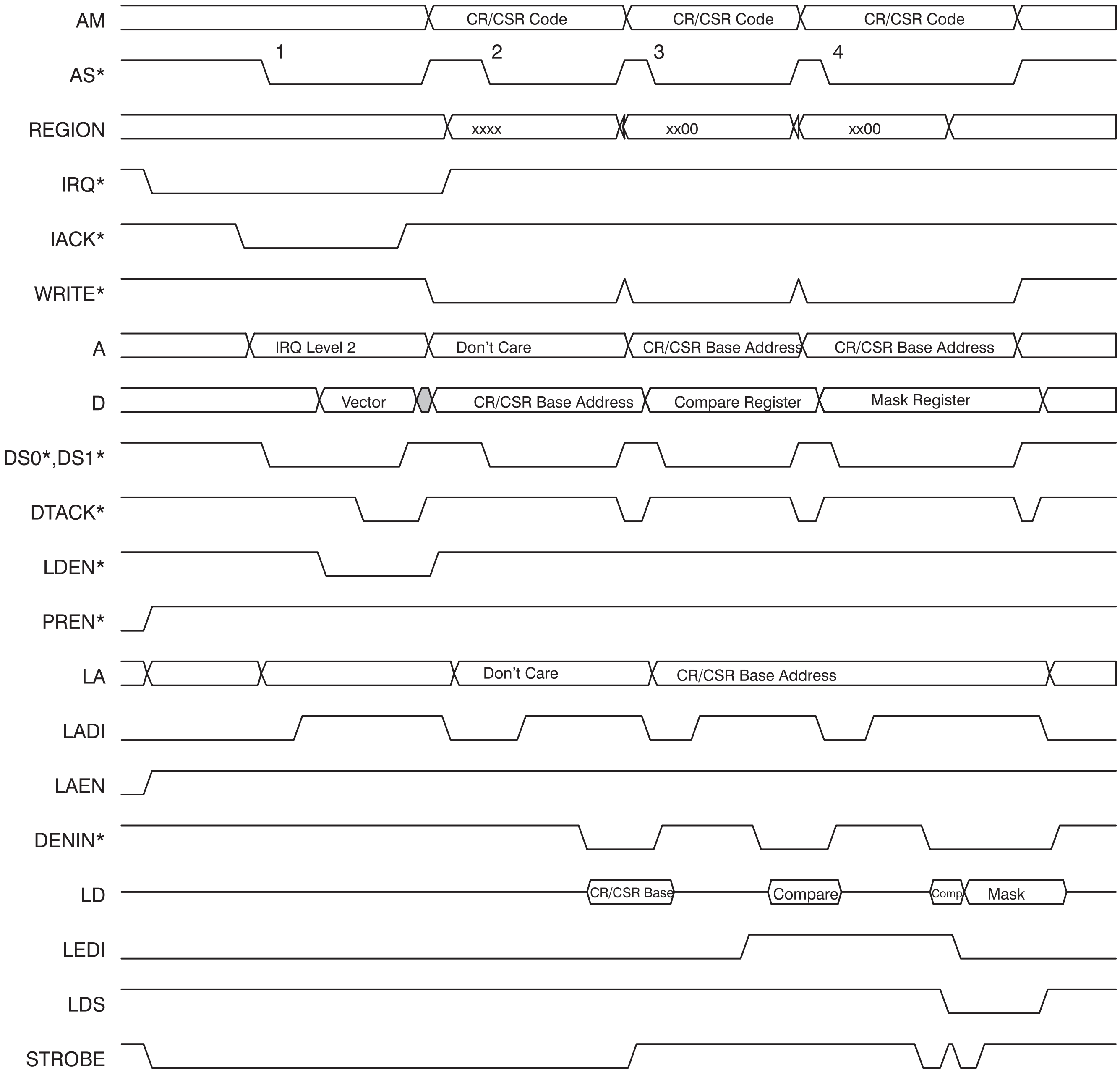

Figure 3−15. Combination Method Timing

ceipt of the final bit of serial data, CY7C960 waits for a High on SYSRESET* (if SYSRESET* is Low). Then IRQ* is driven Low. The resultant IACK cycle (AS* #1) causes CY7C960 to drive LDEN* Low, which causes local circuitry to drive a vector that is passed to the VMEbus. The interrupt handler must be programmed to then write the CR/CSR base address (AS* #2). This can use the CR/CSR AM Code, or one of the two A16 AM Codes.

Note that, as the board has not yet been programmed, the VMEbus Address, and hence the REGION inputs to CY7C960 are not significant. The implication of this is that the IACK cycle and the first write cycle must be indivisible on the VMEbus. The CY7C964s are primed to receive the Address Compare data as STROBE previously went Low while LDS was High. DENIN* and DENIN1* go Low, which drives the latched VMEbus data onto the local bus. This is the Address Compare register value, the CR/CSR base address value. The CY7C960 drives STROBE High, latching the value into the CY7C964 Compare register (and clearing the Mask register in the process). Now the CY7C964s have a meaningful value to compare against VMEbus address, and the REGION inputs to the CY7C960 become significant. If A16 cycles are used the CY7C960 expects the value “xxx0” on REGION[3:0], or if CR/CSR cycles are used, “xx00”. The next two write cycles must be of the same AM Code as the first write cycle, but they do not have to be indivisible. AS* #3 provides the “final” address for the board’s slave address map, and AS* #4 provides the associated mask pattern. Note that during cycle #3 LEDI goes High: this latches the value of the VME data bus inside the CY7C964. Then during cycle #4, while the VMEbus is carrying the mask value, the data held from cycle #3 is loaded into the Address Compare register (STROBE going High), then LDS and STROBE toggle, selecting the Mask Register. Finally, LEDI goes Low allowing the Mask data to flow from the VMEbus, and STROBE goes High loading the Mask register. The CY7C964s are now primed to respond to the appropriate address.

3.4.6 Configuration Software

To assist in developing and debugging CY7C960 configurations for specific applications, Cypress has developed a Windows-based program called WinSvic. This program assists in configuration by leading the designer through all the steps needed to successfully generate the file that will be used to configure the CY7C960 during initialization

WinSvic presents device configuration graphically as a set of nested menu forms. The main screen provides for selection of basic options, such as DRAM/IO versus I/O mode, Decode Delay, and the method by which the CY7C960 is to be configured (Serial, Serial/VME, or VME). Selecting options changes the availability of submenu forms. For example, when DRAM/IO mode is selected, the I/O configuration submenu button is greyed out.

Selecting I/O mode allows the designer to move into the IO submenu. Here the polarity of the six available chip select outputs are individually specified, and for each Region, the pattern to be driven from the Chip Selects when that Region is addressed. Chip Select Assert Time is also specified for each individual Region.

A second submenu, the AM Code dialog box, allows specification of which AM codes are to be accepted as valid for each Region. The special AM codes, CR/CSR, USER1, USER2, LOCK, and SERIAL, are also enabled and assigned in this submenu.

If DRAM/IO mode is selected, the designer can access the DRAM submenu to specify various DRAM configuration options. DRAM timing parameters such as, RAS*/CAS* Delay, RAS* PreCharge, and CAS* Assert are specified in terms of CY7C960 CLK periods. Refresh controls, DRAM Region enables, Chip Select pattern and polarity, and ROW/COL control polarity are all selectable from this form.

Once the appropriate submenus have been negotiated, the main menu is revisited, and the configuration file can be written to disk using the appropriate button or menu choice.

Context sensitive help is available, and the program can write existing names for modification and adjustment. WinSvic guides the designer through the configurability of the CY7C960, making clear the relationship between operating modes and programmable features.

3.4.7 Programmable Features

During configuration the CY7C960 functionality is selected, and many timing parameters are set. For more information on the individual functions read the appropriate chapter of this document. The following two tables serve as a brief outline of the various functions.

The CY7C960 can be viewed as having two sections: Local Bus side, and VMEbus side. *Table 3−1* indicates the programmable functions that the user employs to affect the behavior of the local side of the chip. The part has two basic modes of operation: DRAM/IO and I/O. A bit in the configuration bit stream selects between the two modes. *Table 3−2* illustrates what control the user has over the VMEbus responses.

Table 3−1. Local Bus Programmable Functions

Programmable Function	DRAM Mode	I/O Mode
Polarity of data byte enable pins	All 4 either hi or low true	All 4 either hi or low true
Chip Select Output Pattern	3 signals	6 signals
DBE assert timing	If DRAM disabled in region, 3 to 18 clock periods	3 to 18 clock periods
Chip Select Output Polarity	Individually selected hi or low	Individually selected hi or low
Polarity of ROW/COL signals	Individually selected hi or low	Not Available
RAS/CAS Delay	2 to 9 clock periods	Not Available
RAS Precharge	5 to 12 clock periods	Not Available
CAS Assert Time	3 to 10 clock periods	Not Available
CAS Precharge	1 to 8 clock periods	Not Available
Refresh burst period	(1 to 255) * 256 clock periods	Not Available
Enable Refresh	Selectable	Not Available
Use DBE pins for refresh	Selectable	Not Available
Number of Regions	0 to 8	0 to 16
Enable DRAM Region access	Individually	Not Available

Table 3−2. VMEbus Programmable Functions

Function	Programmable Response
Decode delay from Address Strobe	2 to 5 clock periods
User group 1 (AM Code 18 − 1F hex)	Any 1 region, or none
User group 2 (AM Code 10 − 17 hex)	Any 1 region, or none
CR/CSR	Any 1 region, or none
Serial	Any 1 region, or none
Lock	Any 1 region, or none
Block Transfers, including MBLT	Any number of regions
Data Access	Any number of regions
Program Access	Any number of regions
Supervisory Access	Any number of regions
Non Privileged Access	Any number of regions
A64 Transfers	Any number of regions
A40 Transfers	Any number of regions
A32 Transfers	Any number of regions
A24 Transfers	Any number of regions
A16 Transfers	Any number of regions
D32/D64 Transactions	Any number of regions
Interrupt Level for IACK response	1 to 7