VIA

Table of Contents

Table of Figures

1.0 Overview

The VIA Cell is implemented as a VTI VGT200 1.5µ CMOS portable library gate array. It is designed as a block to be utilized within a larger gate array or standard cell part. The design implements all 6523 VIA features which have been used on Macintoshes from the SE onward and is otherwise fully software compatible with the original VIA. The VIA Cell generates its own internal clock to which it self-synchronizes. It also provides a simple bus interface which helps eliminate the need for the bus interface glue traditionally used with the 6523 in 68000-based systems. It is composed of approximately 2,100 VTI equivalent gates. Refer to Figure 1, a block diagram of the VIA Cell.

Much of the text of this document is identical to that found in the original 6523 VIA spec. All ways in which the VIA Cell differs from the 6523 VIA have been noted in this specification by enclosing the descriptions in a box. The reader familiar with the original 6523 VIA may easily scan this document, reading each of the paragraphs in boxes to determine if any needed 6523 features are not supported by the VIA Cell or if they are implemented differently. Also, the reader is encouraged to closely examine the timing specifications and diagrams as these are, in many cases, different from those of the original VIA.

2.0 Guidelines for using the VIA Cell within a larger chip

First of all, care should be exercised when re-routing the Cell to insure that no particularly long paths are inadvertently created. Healthy timing margins were provided throughout the VIA Cell so that the design would be less sensitive to routing changes, but even so, a greater than usual level of care should attend the routing of the Cell since it is meant to be used as a "black-box" block and therefore the user can not know all the details of the internal timing and how they might be affected by adverse routing. After routing, the entire test program should be re-simulated as usual to insure proper operation of the circuit.

All of the timing specifications of the VIA Cell are based upon the signal levels at the edge of the Cell with an assumed loading (of 0.5 pF). If the user should wish to replace an existing buffer in one of the VIA Cell outputs with a different buffer, obviously the changed circuit will have to be simulated to calculate the new timing. In most cases, changing any parts of the VIA Cell will probably not be worth the time and effort involved and is therefore discouraged.

Most of the VIA Cell's I/O's are needed directly in order for the test program to be run. This means that those VIA Cell inputs or outputs which are not passed directly to a chip I/O will probably have to be muxed through to some pin so that they are available for running the VIA Cell test program. The details of how the VIA Cell should be connected within a larger chip and what support is needed for testing are provided below in Section 11.0, "Testing the VIA Cell."

3.0 Global Reset (Reset_)

Reset_ is an active low signal which clears all registers in the VIA Cell and initializes all state machines. The Shift Register counter is initialized but the actual shift register itself is not reset except for the 9th bit (CB2out) which is cleared. The T1 and T2 counters and latches are also not cleared upon reset. All internal registers (ACR, PCR, DDRA, DDRB, ORA, ORB, IER, IFR) are all cleared to 0. All chip operations are disabled while Reset_ is low.

Note that the only difference between the 6523 and the VIA Cell as far as Reset_ is concerned is that the VIA Cell clears the CB2out bit of the Shift Register upon Reset_.

4.0 Internal C783K Clock

Unlike the 6523 VIA, a 783.35 KHz clock need not be supplied to the VIA Cell from outside. The VIA Cell generates its own internal clock off of the standard C16M clock (15.667 MHz) common to all Macintoshes.

5.0 VIA Cell Bus Interface

The VIA Cell is designed to work in conjunction with 68000, 68020, and 68030 based machines, however it should be simple to incorporate it into a 68040 based machine given appropriate glue logic. As indicated in the timing specifications, the VIA Cell can be used in designs with fast processor clock rates. Only certain relatively limited constraints are made upon the bus interface timing for accesses to and from the Cell.

5.1 DSACK

Because accesses to the VIA Cell must be synchronized to the internal C783K clock (in order to make operation identical to the 6523 VIA which is required for some time-critical software), a variable number of wait states must be inserted on each access. Thus, the VIA Cell generates its own DSACK_ (or DTACK_) signal based upon CS and AS_.

5.2 Chip Select (CS)

Accesses to the VIA Cell are activated by the active-high CS (Chip Select) signal. CS should be generated by decoding the address space of the VIA Cell and ANDing with the inverse of the 68000-family bus signal AS_ (Address Strobe). In other words, CS should be active when AS_ (which is active-low) is active.

5.3 Read/Write (RW)

RW controls the direction of transfer between the CPU and the VIA. When RW is high, it indicates a Read operation (if CS is asserted). When RW is low it indicates a Write (also if CS is asserted).

5.4 Data Bus (Din[7:0] and Dout[7:0])

The Din and Dout lines are used to write to and read from the VIA. Two bytes of unidirectional lines are provided rather than one byte of bidirectional lines in order to provide more flexibility in incorporating the Cell into a larger chip. In the simplest configuration the Din lines would come from input buffers and the Dout lines would drive tristatable output buffers whose output enables would be driven by RW_ (i.e. the negation of RW would drive the active low OE's).

5.5 RegSel[3:0] Register Selects

The RegSel address lines select which register within the Cell to access.

5.6 Interrupt Request (IRQ)

The Interrupt Request output is an active low signal which is asserted when one of the Interrupt Flag Register bits is set and the corresponding bit in the Interrupt Enable Register is also set.

6.0 Peripheral Data Ports (Ports A and B)

Each Peripheral Data Port operates in conjunction with a Data Direction Register (DDRA or DDRB). Under program control, the Data Direction Registers specify which lines within the port bus are to be designated as inputs or outputs. A Logic 0 in any bit position of the register will cause the corresponding line to serve as an input, while a Logic 1 will cause the line to serve as an output. Refer to Figure 4.

When a line is programmed as an output, it is controlled by a corresponding flip-flop in the Output Register (ORA and ORB). A Logic 1 in the Output Register will cause the corresponding output line to go high, while a Logic 0 will cause the line to go low. Upon a write operation to one of the ports, data is written into only those port bit positions which have been programmed as outputs. Should data be written into bit positions corresponding to lines which have been programmed as inputs, the Output Register flip-flops will be unaffected.

When reading a Peripheral Data Port, if a given port bit has been programmed as an input, the value on the corresponding Input Register line (IRA or IRB) is directly transferred into the Read Latch and onto the Dout[7:0] lines (See Block Diagram, Figure 1). Note that there are no flip-flops associated with the Input Register lines (IRA[7:0] and IRB[7:0]). There are however, flip-flops for each bit of a port's Output Register. If a port bit is programmed as an output, then the contents of the corresponding Output Register flip-flop will be transferred into the appropriate Read Latch bit and onto the correct Dout[7:0] line (again, see Figure 1).

Input Register B operates in an identical fashion to Input Register A.

Unlike the 6523 VIA, with the VIA Cell the values read from Port A bits programmed as outputs are never ambiguous. For either Port A or B, with a Read operation, if a given bit is programmed as an output, the data placed in the 8 bits of the Read Latch is always the same as the values on the individual Port bit outputs. Therefore, the values returned upon a Read are always unambiguous. The values read with the 6523 VIA depended upon the loading on the port lines.

6.1 Peripheral Data Port A (PA7-PA0)

Peripheral Data Port A is an 8-line, bidirectional bus used for the transfer of data, control and status information between the VIA Cell and a peripheral device. Each Peripheral Data Port bus line may be individually programmed as either an input or output under control of a bit of Data Direction Register A. Data flow direction may be selected on a line-by-line basis with intermixed input and output lines within the same port. When an "0" is written to any bit position of Data Direction Register A, the corresponding line will be programmed as an input. Likewise, when a "1" is written into any bit position of the register, the corresponding data line will serve as an output. Polarity of the data output is determined by the Output Register.

Unlike the 6523 VIA, the VIA Cell does not provide the capability to latch input data into the Input Register under control of the CA1 line. As mentioned above, since the latching feature is not provided in the VIA Cell, there are no flip-flops associated with the Input Registers.

All supported modes are program controlled by the CPU by way of the VIA Cell's internal control registers.

6.2 Peripheral Data Port A Control Lines (CA1, CA2)

Control lines CA1 and CA2 always serve as interrupt inputs in the VIA Cell. The 6523 VIA allowed these lines to be used for handshaking but this feature is not supported in the VIA Cell.

CA1 and CA2 each control an internal Interrupt Flag (in the IFR) with a corresponding Interrupt Enable bit (in the IER). These inputs are either positive or negative edge sensitive depending upon the setting of the PCR (See Figure 5).

6.3 Peripheral Data Port B (PB7-PB0)

Peripheral Data Port B is an 8-line, bidirectional bus which is controlled by Data Direction Register B in a manner identical to Data Port A.

However, the VIA Cell does not support the use of Timer 2 to count pulses on the PB6 line as does the 6523 VIA. Also, unlike the 6523 VIA, the output signal on line PB7 may not be controlled by Timer 1. See the section on Timer 1 below for details.

6.4 Peripheral Data Port B Control Lines (CB1, CB2)

Control lines CB1 and CB2 serve as interrupt inputs for Peripheral Data Port B. Like Port A, these two control lines control an Interrupt Flag with a corresponding Interrupt Enable bit. These lines also serve as a serial data port under control of the Shift Register (SR). Like CA1 and CA2, they are edge sensitive and the active edge is set in the PCR (See Figure 5). Note that the dedicated Interrupt Flags corresponding to CB1 and CB2 (IFR bits 4 and 3 respectively) will continue to be set on the proper edges of CB1 and CB2 even if the Shift Register is in use. In other words, operation of the Shift Register and the setting of these two Interrupt Flags is independent.

Note that the VIA Cell does not support the use of CB1 and CB2 as handshake outputs for Port B as does the 6523 VIA.

7.0 Timer 1 - Operation

Interval Timer T1 consists of a two 8-bit latches and a 16 bit counter. The latches serve to store data which is to be loaded into the counter. Once the counter is loaded under program control, it decrements at the C783K clock rate. Thus each clock count is 1.27655 µsec long. Upon reaching zero, the counter rolls around to $FFFF and Interrupt Flag Register bit 6 is set. This will cause Interrupt Request (IRQ_) to go low if IER (Interrupt Enable Register) bit 6 is set. Once the Timer reaches a count of $FFFF, it will automatically transfer the contents of the latches into the counter and proceed to decrement again. Also it will disable any further setting of Interrupt Flag 6 if the counter has been programmed to operate in One-Shot Mode (see below).

Note that Timer 1 times out N+1 cycles after loading with the value N (See Figure 8). For example, if the Timer is loaded with $0003, it will set the interrupt flag 4.5 C783K cycles later, which is 4.5 times 1.276µsec or approx. 5.7µsec later. If the Timer is loaded with $FFFF, it will

time out 65536.5 C783K cycles later; if loaded with $0000, it will time out 1.5 C783K cycles later.

The VIA Cell does not support inversion of the output signal on PB7 each time the Timer reaches a count of $FFFF. Therefore, Auxiliary Control Register bit 7 which controls this option is hard-wired to a "0". Thus, when the ACR is read, a "0" always appears in bit 7 which indicates that this function is disabled and cannot be enabled.

The T1 counter format and operation is shown in Figure 7 with corresponding latch format and operation also shown. An additional control bit is provided in the Auxiliary Control Register (ACR bit 6) to allow selection of Timer T1 operation modes. The modes available are shown in Figure 6.

Again, note that the VIA Cell supports only two of the four T1 modes available in the 6523 VIA.

It should also be noted that the CPU does not write directly into the low-order counter (T1CL). Instead, this half of the counter is loaded automatically from the low-order latch when the CPU writes into the high-order latch and counter. In fact, it may not be necessary to write to the low-order latch in some applications since the timing operation is triggered by writing to the high-order latch and counter.

7.1 Timer 1 - One-Shot Mode

Interval Timer T1 may operate in the One-Shot Mode which allows the generation of a single Interrupt Flag each time the Timer is loaded.

Note that the VIA Cell cannot be programmed to generate a single negative pulse on Data Port line PB7.

To generate a single interrupt, it is required that bit 6 of the Auxiliary Control Register be low (Refer to Figure 6). The low-order T1 counter (T1CL) or the low-order T1 latch (T1LL) must then be loaded with the low-order count value. Note that a load to T1CL is effectively a load to T1LL. Next, the high-order count value must be loaded into the high-order T1 counter (T1CH) at which time the value is simultaneously loaded into high-order T1 latch (T1LH). During this load sequence, the contents of T1LL is transferred to T1CL. The counter will start counting down on the next C783K clock pulse following the load sequence into T1CH, and will decrement at the C783K clock rate. Once the T1 counter reaches a count of $FFFF, Interrupt Flag 6 is set.

Once set, the T1 Interrupt Flag (Interrupt Flag Register bit 6) is reset by either writing to T1LH, loading T1CH (which starts a new count), or by reading T1CL. Refer to Figure 8 for One-Shot Mode timing information.

Note that T1 is always running and that the value in the latches is always loaded into the counter after a count of $FFFF. However, when in One-Shot Mode, Interrupt Flag 6 (IFR bit 6) is only set the first time that the count reaches $FFFF after writing to T1CH. Even if Interrupt Flag Register bit 6 is cleared by reading T1CL, IFR bit 6 will not be set again until after the Timer is reinitialized by writing to T1CH and a count of $FFFF occurs. In this way, Timer 1 acts like a One-Shot timer. Also note that no Interrupt Request (IRQ_) will be generated unless Interrupt Enable Register bit 6 is set.

7.2 Timer 1 - Free-Run Mode

When operating in Free-Run Mode, Interrupt Flag 6 is set each time the counter reaches a count of $FFFF. As in One-Shot Mode, the counter is always running and transfers the contents of the latches into the counter one C783K cycle after a count of $FFFF is reached. Then the counter decrements at the C783K clock rate from the new count value just loaded from the latches. Every time that a count of $FFFF is reached, Interrupt Flag Register bit 6 is set if it hasn't previously been set. There is no need to write to T1CH to re-enable the setting of Interrupt Flag Register bit 6 as there is in One-Shot Mode. Thus, a constant stream of Interrupt Requests (IRQ_) can be generated in Free-Run Mode if Interrupt Enable Register bit 6 has been set.

Since the interval timers are all retriggerable, reloading the counter will always reinitialize the time-out period. Should the CPU continue to reload the counter before it reaches $FFFF, counter time-out can be prevented. Timer 1 is able to operate in this manner provided the CPU writes into the high-order counter (T1CH). By loading the latches only, the CPU can access the timer during each count-down operation without affecting the time-out in progress. In this way, data loaded into the latches will determine the length of the next subsequent time-out period.

Note again that the VIA Cell does not support the inversion of PB7 every time that a count of $FFFF is reached. Instead, PB7 is hardwired as a normal output port bit.

8.0 Timer 2 - Operation

Timer 2 operates in the One-Shot Mode only (as an interval timer). T2 is made up of a write-only low-order latch (T2LL), a read-only low-order counter (T2CL), and a read/write high-order counter (T2CH). This 16-bit counter decrements at the C783K rate. Thus each count is 1.27655 µsec long. Refer to Figure 7 for T2 counter format and operation.

Note that like Timer 1, Timer 2 times out N+1 cycles after loading with the value N (See Figure 8). In identical fashion to Timer 1, if the Timer is loaded with $0003, it will set the interrupt flag 4.5 C783K cycles later, which is 4.5 times 1.276µsec or approx. 5.7µsec later. If the Timer is loaded with $FFFF, it will time out 65536.5 C783K cycles later; if loaded with $0000, it will time out 1.5 cycles later.

The VIA Cell does not support the pulse counting mode available in the 6523 VIA. This mode allowed T2 to count each time a negative edge was detected on Data Port Line PB6. Instead PB6 is hardwired as a normal input/output port bit.

8.1 Timer 2 - One-Shot Mode

Operation of Timer 2 in the One-Shot Mode is similar to Timer 1. That is, for each load T2CH operation, Timer 2 sets Interrupt Flag bit 5 for each countdown to $FFFF. Like Timer 1, after a timeout, the T2 counter rolls over to all 1s ($FFFF). However, T2 is different from T1, because T1 always loads in the value contained in the latches upon time-out whereas T2 simply rolls over from $FFFF to $FFFE, and then $FFFD, etc. This two's complement decrement allows the user to determine how long the T2 Interrupt Flag has been set. Since the logic for Interrupt Flag bit 5 is disabled after the initial interrupt (at a count of $FFFF), further interrupts cannot be set by a subsequent count down to $FFFF. To enable the Interrupt Flag logic for bit 5, the CPU must

reload T2CH. Interrupt Flag bit 5 is cleared by either reading T2CL or by loading T2CH. Refer to Figure 8 for timing information on the One-Shot Mode.

8.2 Timer 2 - Pulse Counting Mode

The 6523 VIA Timer 2 Pulse Counting Mode is not implemented in the VIA Cell, and therefore, all bits in the Auxiliary Control Register corresponding to this function are hardwired to indicate that it is always disabled.

9.0 Shift Register Operation

The Shift Register performs bidirectional serial data transfers on line CB2 (CB2in and CB2out when used in conjunction with SDataOE_ as a tristate output enable, are collectively called CB2). These transfers are controlled by an internal modulo-8 counter. Shift pulses are applied to the CB1 line from an external source only. Refer to Figure 6 for format and control bit information. Refer to Figures 14 and 15 for timing information.

Note that the 6523 VIA sampled the CB1 shift clock on the edges of the 783K Hz clock (the E clock). The VIA Cell, on the other hand, synchronizes both CB1 and CB2 to C16M, and shifting is therefore done at half the C16M rate, not at half the C783K rate. The result is that the VIA Cell's Shift Register has much higher performance than the shift register in the original 6523 VIA. Of course the VIA Cell Shift Register may still be run at 6523 VIA speeds, but it may also be used to transfer data at a rate of approximately 893KBytes/sec, which makes the Shift Register port more versatile.

The VIA Cell does not support any of the internal shift clock sources provided by the 6523 VIA. The clock source and the Shift Register's operating modes are controlled by bits in the Auxiliary Control Register. In the VIA Cell, these bits are hardwired to indicate that the Shift Register is either in the mode in which it runs off of an external clock, or it is disabled. The modes supported by the VIA Cell are:

Shift Register Disabled (000)
Shift In - External CB1 Clock Control (011)
Shift Out - External CB1 Clock Control (111)

9.1 Shift Register Disabled (000)

In the 000 mode, the Shift Register is disabled from all operation. The CPU can read or write the Shift Register, but shifting is disabled and both CB1 and CB2 are controlled by bits in the Peripheral Control Register (PCR). The Shift Register Interrupt Flag (IFR bit 2) cannot be set while the Shift Register is disabled. On the other hand however, it is not cleared if previously set before disabling the Shift Register.

9.2 Shift In - External CB1 Clock Control (011)

In this mode, CB1 serves as a clock input to the Shift Register. In this way, an external device can load the Shift Register at its own pace. The Shift Register counter will set Interrupt Flag Register bit 2 after each eight bits have been shifted in. The Shift Register counter does not stop the shifting operation. Its function is simply that of a Shift Clock (CB1) counter. Reading or writing the Shift Register resets Interrupt Flag Register bit 2 and initializes the counter to count another eight pulses.

Refer to Figure 15 for timing requirements of the shift in data (on line CB2) with respect to the shift clock (on line CB1).

Note that when shifting in, bits initially enter bit 0 and are shifted towards bit 7.

9.3 Shift Out - External CB1 Clock Control (111)

In this mode, shifting is controlled by external pulses applied to the CB1 line. The shift Register counter sets Interrupt Flag Register bit 2 for each eight-pulse count, but does not disable the shifting function. Each time the CPU reads or writes the Shift Register, Interrupt Flag 2 is reset and the counter is initialized to begin counting the next eight pulses on the CB1 line. After eight shift pulses, the Interrupt Flag is set. The CPU can then load the Shift Register with the next eight bits of data. Refer to Figure 14.

Note that when shifting out, bit 7 is the first bit out and it is simultaneously rotated back into bit 0. The Shift Register is actually composed of 9 flip-flops and the last bit (SR[8]) contains the value that appears on the CB2out line. SR[7] is shifted into SR[8] upon a shift clock (CB1) active edge. Thus, after parallel loading the Shift Register with data, one Shift must be done before any of the just-loaded data will appear on CB2out. Seven more shifts bring out the rest of the data serially on the CB2out line.

10.0 Interrupt Operation

There are three basic interrupt operations, including: setting the interrupt flag within the Interrupt Flag Register (IFR), enabling the interrupt by way of a corresponding bit in the Interrupt Enable Register (IER), and signaling the CPU with an Interrupt Request (IRQ_). An Interrupt Flag can be set by conditions internal to the chip or by inputs to the chip from external sources. Normally, an Interrupt Flag will remain set until the interrupt is serviced. To determine the source of an interrupt, the CPU must examine each flag in order, from highest to lowest priority. Each Interrupt Flag has a corresponding Interrupt Enable bit in the Interrupt Enable Register. The enable bits are controlled by the CPU (set or cleared). If an Interrupt Flag is high (Logic 1) and the corresponding Interrupt Enable bit is high (Logic 1), the Interrupt Request (IRQ_) will go low (Logic 0).

All Interrupt Flags are contained within a single Interrupt Flag Register. Bit 7 of this register will be high (Logic 1) whenever an Interrupt Flag is set, thus allowing convenient polling of several devices within a system to determine the source of the interrupt.

The Interrupt Flag Register (IFR) and Interrupt Enable Register (IER) format and operation is shown in Figures 9 and 10 respectively. The Interrupt Flag Register may be read directly by the CPU, and individual flag bits may be cleared by writing a "1" into the appropriate bit of the IFR. Bit 7 of the IFR indicates the status of the Interrupt Request (IRQ_) output. Bit 7 corresponds to the following logic function: IRQ=IFR6 x IER6 + IFR5 x IER5 + IFR4 x IER4+ IFR3 x IER3 + IFR2 x IER2 + IFR1 x IER1 + IFR0 x IER0. Note: "x" --> Logical AND, "+" --> Logical OR.

Bit 7 is not a flag. This means that when one writes to the IFR, bit 7 of the data written is a don't care. Thus the value of bit 7 on writes to the IFR has no effect on the write at all. Since IFR 7 is not a flag and therefore has no associated flip-flop, it is not directly cleared by writing a "1" into its bit position. It can only be cleared by clearing all the flags within the register, or by disabling all active interrupts as presented in the next section.

Each Interrupt Flag within the IFR has a corresponding enable bit in the Interrupt Enable Register (IER). The CPU can set or clear selected bits within the IER. This allows control of individual interrupts without affecting others. To set or clear a particular Interrupt Enable bit, the CPU must write directly to the address assigned to the IER (RegSel3..RegSel0 = 11110). During this write operation, if bit 7 on the Data Bus is a "0", each "1" in bits 6 through 0 will clear the corresponding bit in the Interrupt Enable Register. For each "0" in bits 6 through 0, the corresponding bit in the IER will be unaffected.

Setting selected bits in the IER is accomplished by writing to the same address with bit 7 on the Data Bus set to a "1". In this case, each "1" in bits 6 through 0 will set the corresponding bit to a "1". For each "0" the corresponding bit will be unaffected. This method of controlling the bits in the Interrupt Enable Register allows convenient user control of interrupts during system operation. The CPU can also read the contents of the IER. Bit 7 will always read as a "1".

When the 6523 VIA receives an interrupt from an external source (CA1, CA2, CB1, or CB2), IRQ_ goes low very quickly after the triggering event occurs (provided of course that that interrupt has been enabled). This is because the 6523 VIA has direct-wired logic to asynchronously set the Interrupt Flags. The VIA Cell has fully synchronous logic and therefore has to wait for the positive edge of C16M in order to effect a change on the IRQ_ line. Therefore, the delay from an Interrupt Input to IRQ_ low may be as much as 210 nsec (refer to Figure 16 and its associated timing specifications).

Although this latency may be perceived at first as a disadvantage, the Interrupt Inputs to the VIA are traditionally used for things such as Vertical Blanking and 60 Hz interrupts; events whose timing is rather coarse. Therefore, the latency introduced by the VIA Cell is usually insignificant compared to the length of the time interval associated with these signals and consequently the delay is negligible.

11.0 Testing the VIA Cell

In order to insure that the VIA Cell test program will run when incorporated into a larger design, certain guidelines must be followed. As mentioned in Section 2.0, the primary guideline is that most of the I/Os need to be muxed out to input or output pads so that they are directly available to the tester. The required interconnections are illustrated in Figure 17. Following these guidelines should make testing the VIA Cell with the existing test vector set relatively easy.

The VIA Cell has several test modes which facilitate testing the logic. The three test inputs, Test[2:0] are used to select the appropriate test mode. The modes provided and their effects are given below:

Test[2:0] = 000 or 100 ---> NORMAL_MODE: The Cell is in its standard operational mode and is software compatible with the 6523 VIA.

Test[2:0] = 001 ---> TEST_T1_SELECT: The internal C783K clock is now accelerated to 7.83MHz and Timer 1's counter bytes are muxed out onto the ORA and ORB lines such that ORA[7:0] = T1CL_[7:0] and ORB[7:0] = T1CH_[7:0]. In this mode, the bus interface is essentially "frozen" so that any reads or writes in progress or started while in this mode will have to wait until NORMAL_MODE is entered before completing.

Test[2:0] = 010 or 110 ---> TEST_T2_SELECT: This mode is identical to TEST_T1_SELECT except that ORA = T2CL and ORB = T2CH.

Test[2:0] = 011 or 111 ---> T2_ACCELERATE_MODE: This mode is the same as TEST_T2_SELECT except that T2's counter is accelerated in the following fashion. Timer 2 is composed of four, 4-bit nybbles, each of which has a terminal count. Counting in a higher nybble occurs only if the terminal counts from all lower nybbles are asserted. In T2_ACCELERATE_MODE, these terminal counts are always forced to true so each nybble counts on every C783K pulse (and recall that C783K is now running at 7.83MHz).

Test[2:0] = 101 ---> T1_ACCELERATE_MODE: This mode is the same as TEST_T1_SELECT except that T1's counter is accelerated in the following fashion. Timer 1 is composed of four, 4-bit nybbles, each of which has a terminal count. Counting in a higher nybble occurs only if the terminal counts from all lower nybbles are asserted. In T1_ACCELERATE_MODE, these terminal counts are always forced to true so each nybble counts on every C783K pulse (and recall that C783K is now running at 7.83MHz).

Thus the count sequence would be: (start in TEST_T1_SELECT mode) 0001, 0000, FFFF, FFFE, FFFD, FFFC, FFFB, FFFA, FFF9, FFF8, FFF7, FFF6, FFF5, FFF4, FFF3, FFF2, FFF1, FFF0, (now enter T1_ACCELERATE_MODE) EEEF, DDDE, CCCD, BBBC, AAAB, 999A, 8889, 7778, 6667, 5556, 4445, 3334, 2223, 1112, 0001, (now return to TEST_T1_SELECT mode) 0000, FFFF.

In the case of Timer 2, this rapid acceleration of the counter is used so that it's interrupt flag setting circuit can be tested (i.e. with this, T2's One-Shot operation can be tested). Since T2 cannot be run in Free-Run mode like T1, it would normally take over 65,536 clocks to see two time-outs. With this test mode, the same two time-outs can be seen in 34 clocks. In the case of both Timer 1 and Timer 2, these test modes allow the test program to achieve better fault coverage in less vectors.

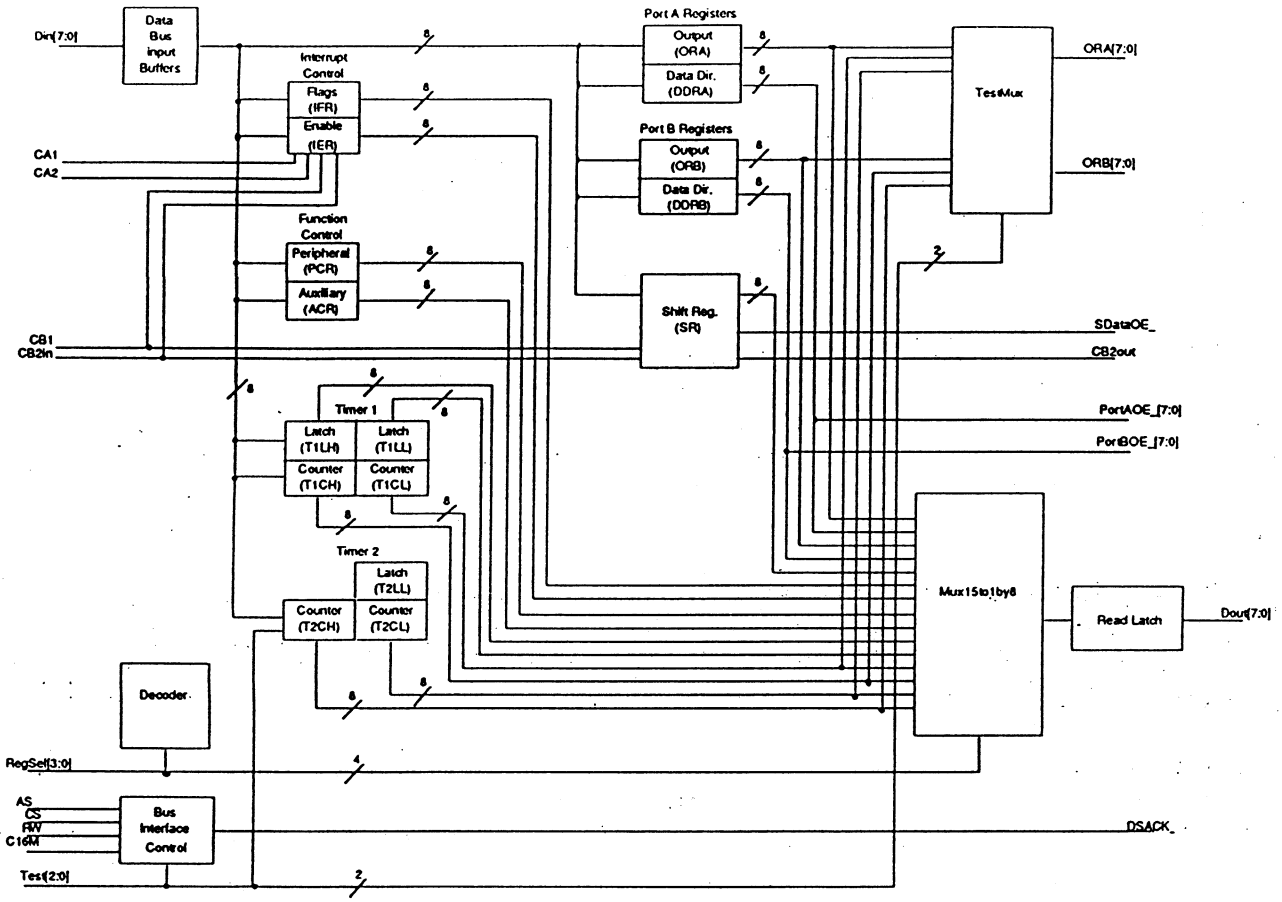


Figure 1 - Simplified VIA Cell Block Diagram

Register Number	RegSel[3:0]	Register Designation	Register/Description Write (RW = "0") / Read (RW = "1)
0	0000	PORTB	Output Register B / Input Register B
1	0001	PORTA	Output Register A / Input Register A
2	0010	DDRB	Data Direction Register B
3	0011	DDRA	Data Direction Register A
4	0100	T1CL	T1 Low-Order Latches / T1 Low-Order Counter
5	0101	T1CH	T1 High-Order Counter
6	0110	T1LL	T1 Low-Order Latch
7	0111	T1LH	T1 High-Order Latch
8	1000	T2CL	T2 Low-Order Latch / T2 Low-Order Counter
9	1001	T2CH	T2 High-Order Counter
10	1010	SR	Shift Register
11	1011	ACR	Auxiliary Control Register
12	1100	PCR	Peripheral Control Register
13	1101	IFR	Interrupt Flag Register
14	1110	IER	Interrupt Enable Register
15	1111	PORTA	Output Register A / Input Register A

Figure 2 - VIA Cell Register Addressing

Signal Name	Type[1]	Input Loading/ Output Drive[2]	Description
Reset_	Input	0.29 pF input loading.	Global VIA Cell Reset.
C16M	Input	0.2 pF input loading.	15.667 MHz clock.
RW	Input	0.1 pF input loading.	680X0 Read/Write_ signal.
CS	Input	0.1 pF input loading.	VIA Cell chip select.
AS_	Input	0.1 pF input loading.	680X0 active low Address Strobe.
RegSel[3:0]	Input	0.3 pF input loading.	VIA Cell register select lines.
Din[7:0]	Input	0.1 pF input loading.	Data input lines.
Test[2:0]	Input	0.1 pF on Test[1:0] and 0.2 pF on Test[2].	VIA Cell Test Mode control lines.
CA1	Input	0.1 pF input loading.	CA1 interrupt input line.
CA2	Input	0.1 pF input loading.	CA2 interrupt input line.
CB1	Input	0.1 pF input loading.	CB1 Shift Register Clock.
CB2in	Input	0.1 pF input loading.	CB2 Shift Register Serial Data in.
IRA[7:0]	Input	0.1 pF input loading.	Port A register input lines.
IRB[7:0]	Input	0.1 pF input loading.	Port B register input lines.
ORA[7:0]	Output	3x output drive.	Port A register output lines.
ORB[7:0]	Output	3x output drive.	Port B register output lines.
PortAOE_[7:0]	Output	3x output drive.	Port A register output enable lines.
PortBOE_[7:0]	Output	3x output drive.	Port B register output enable lines.
IRQ_	Output	3x output drive.	VIA interrupt output line.
CB2out	Output	3x output drive.	CB2 Shift Register Serial Data out.
SDataOE_	Output	3x output drive.	CB2 Shift Register Data output enable.
DSACK_	Output	3x output drive.	VIA Cell's Data Transfer Acknowledge line.
Dout[7:0]	Output	3x output drive.	Data output lines.
C16M3[3]	Output	Do not load with more than 1pF.	One of the three internal C16M clock tree branches.

NOTES:

[1]The VIA Cell has only inputs and outputs. There are no bidirectionals. With those signals which are meant to be used as bidirectionals, the inputs, outputs, and output enables are provided so that the user may attach them in the most convenient fashion possible; for example, by using a bidirectional pad driver.

[2]All output buffers are "3x." This refers to VGT200 library 3x inverting buffers. Consult the VTI VGT200 library manual or the timing specifications in this spec for further details on timing characteristics of these output buffers and a derating formula for varying loads.

[3]This signal is the actual C16M which drives the flip-flops within the VIA Cell after it has been buffered and distributed. It is provided for synchronization or other system uses.

Figure 3 - VIA Cell Pin List

REG 2 - DDRB & REG 3 - DDRA

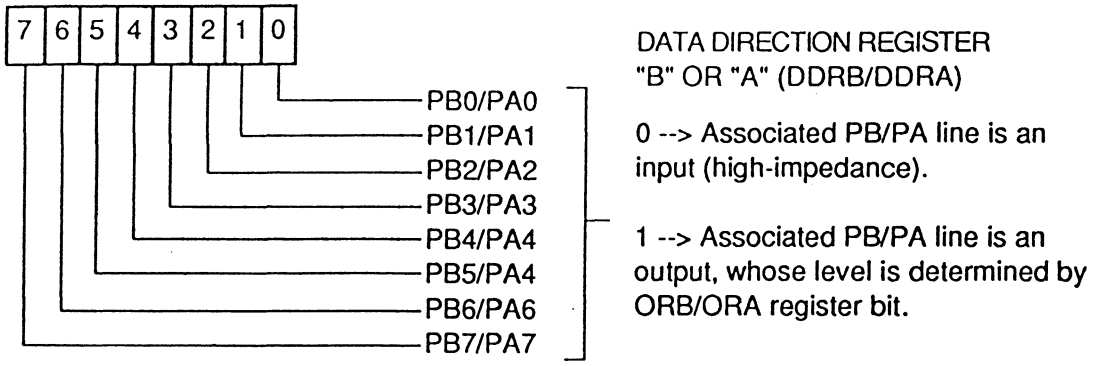
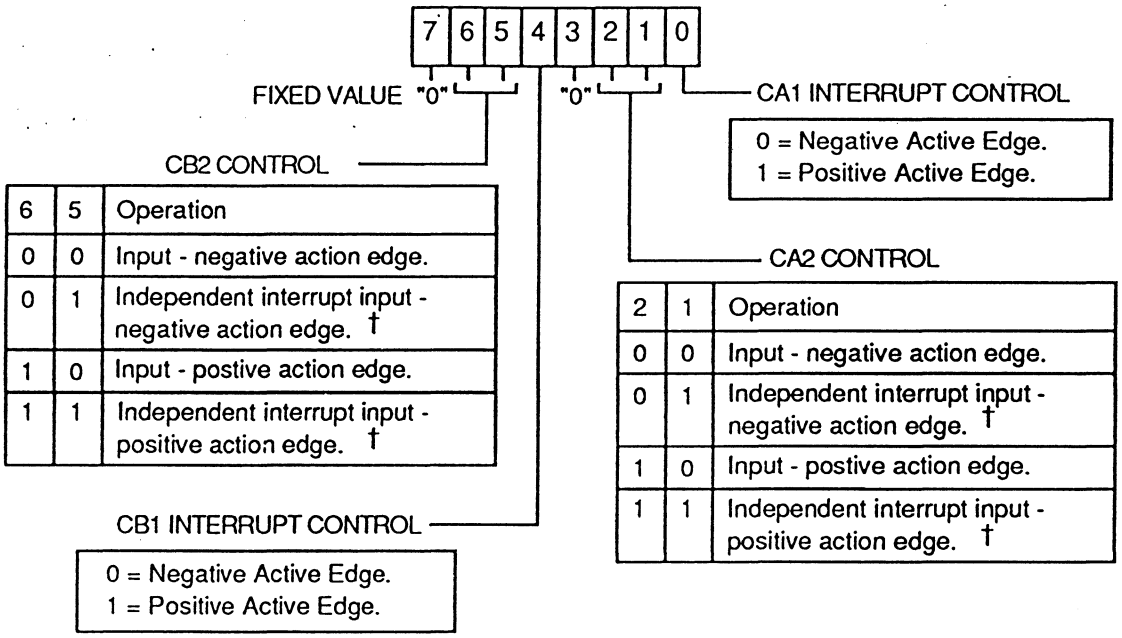


Figure 4 - Data Direction Registers (DDRA, DDRB)

REG 12 - PERIPHERAL CONTROL REGISTER



6	5	Operation
0	0	Input - negative action edge.
0	1	Independent interrupt input - negative action edge. †
1	0	Input - postive action edge.
1	1	Independent interrupt input - positive action edge. †

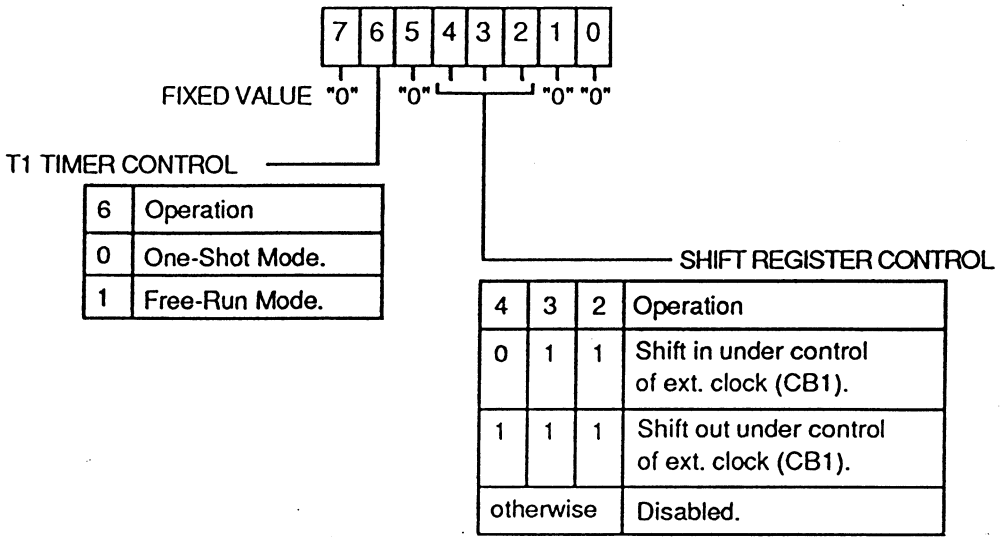
2	1	Operation
0	0	Input - negative action edge.
0	1	Independent interrupt input - negative action edge. †
1	0	Input - postive action edge.
1	1	Independent interrupt input - positive action edge. †

† If the CA2/CB2 control in the PCR is selected as an "independent" interrupt input, then reading or writing Port A or Port B will not clear the corresponding Interrupt Flag Register bit. Instead, the bit must be cleared by writing into the IFR, as described previously.

Note that bits 7 and 3 of the PCR are hardwired to "0". These bits are used in the 6523 VIA to select modes which are not supported in the VIA Cell.

Figure 5 - CA1, CA2, CB1, CB2 Control

REG 11 - AUXILIARY CONTROL REGISTER



6	Operation
0	One-Shot Mode.
1	Free-Run Mode.

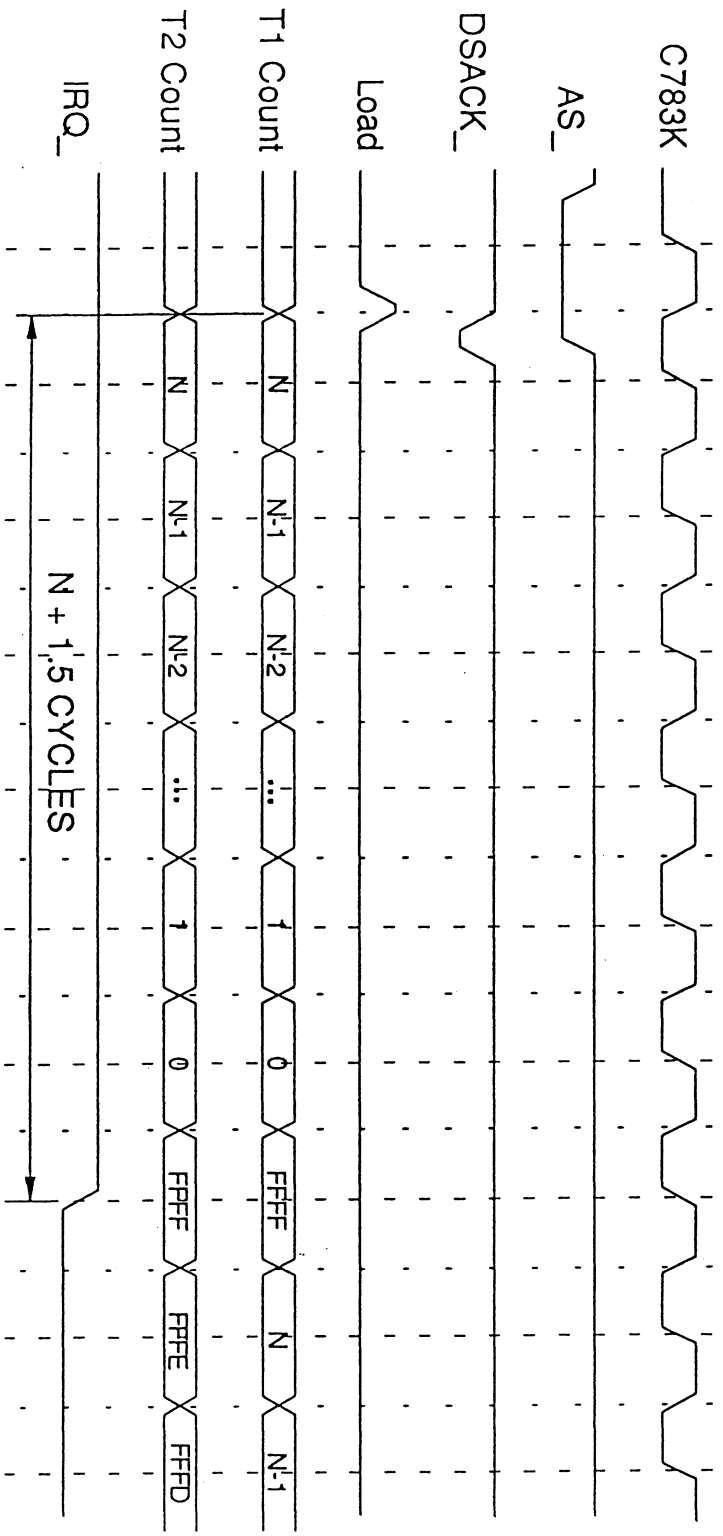
4	3	2	Operation
0	1	1	Shift in under control of ext. clock (CB1).
1	1	1	Shift out under control of ext. clock (CB1).
otherwise			Disabled.

Note that bits 7, 5, 1, and 0 of the ACR are hardwired to "0". These bits are used in the 6523 VIA to select modes which are not supported in the VIA Cell.

Figure 6 - Auxiliary Control Register Format & Operation (ACR)

	WRITE	READ
TIMER 1 Low-order Counter (T1CL), Reg. 4	8 bits loaded into T1 low-order latches. Latch contents are transferred into low-order counter at the time the high-order counter is loaded (Reg. 5).	8 bits from T1 low-order counter transferred to CPU. In addition, T1 interrupt flag is cleared (IFR bit 6).
TIMER 1 High-order Counter (T1CH), Reg. 5	8 bits loaded into T1 high-order latches. Also, at this time both high and low-order latches are transferred into T1 counter and initiates countdown. T1 interrupt flag also is cleared.	8 bits from T1 high-order counter transferred to CPU.
TIMER 1 Low-order Latch (T1LL), Reg. 6	8 bits loaded into T1 low-order latches. This operation is no different than a write to Reg. 4.	8 bits from T1 low-order latch transferred to CPU. Unlike Reg. 4 operation, this does not cause the clearing of T1's interrupt flag.
TIMER 1 High-order Latch (T1LH), Reg. 7	8 bits loaded into T1 high-order latches. Unlike Reg. 4 operation no latch-to-counter transfer takes place. T1 interrupt flag is cleared.	8 bits from T1 high-order latches transferred to CPU.
TIMER 2 Low-order Counter (T2CL), Reg. 8	8 bits loaded into T2 low-order latches.	8 bits from T2 low-order counter transferred to CPU. T2 interrupt flag is cleared.
TIMER 2 High-order Counter (T2CH), Reg. 9	8 bits loaded into T2 high-order counter. Also, low-order latches transferred to low-order counter. In addition, T2 interrupt flag is cleared.	8 bits from T2 high-order counter transferred to CPU.

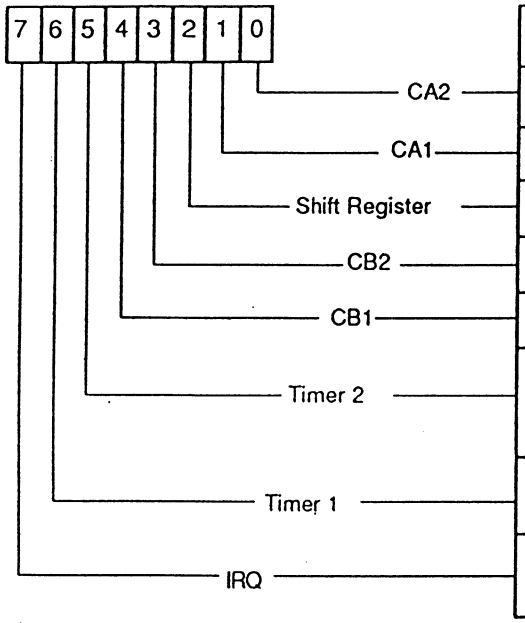
Figure 7 - Timer 1 and Timer 2 Control



NOTE: Load is an internal signal which is used to load the various registers within the VIA Cell, in this case, the Timers.

Figure 8 - Timer 1 and Timer 2 One-Shot Mode Operation

REG 13 - INTERRUPT FLAG REGISTER

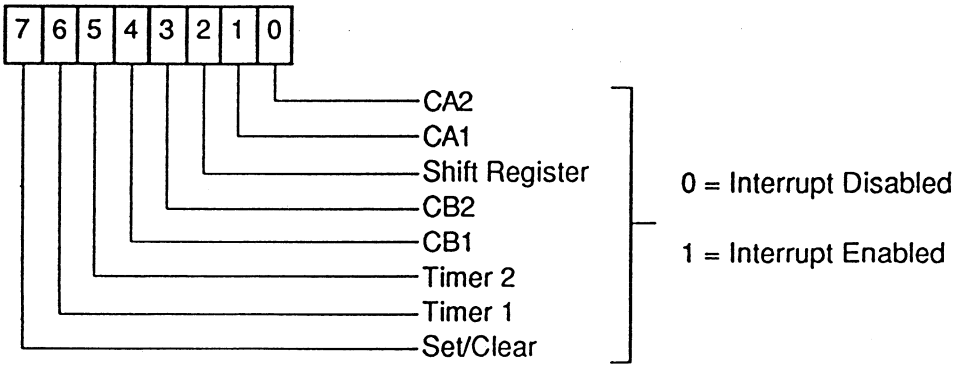


SET BY	CLEARED BY
CA2 active edge.	Read or write Port A. †
CA1 active edge.	Read or write Port A.
Complete 8 shifts in or out.	Read or write Shift Register.
CB2 active edge.	Read or write Port B. †
CB1 active edge.	Read or write Port B.
Time-out of T2.	Read T2 low or write T2 high (read T2CL or write T2CH).
Time-out of T1.	Read T1CL or write T1CH or write T1LH.
Any enabled interrupt flag which is set.	Clearing all interrupt flags.

† If the CA2/CB2 control in the PCR is selected as an "independent" interrupt input, then reading or writing Port A or Port B will not clear the flag bit. Instead, the bit must be cleared by writing into the IFR, as described previously.

Figure 9 - Interrupt Flag Register (IFR)

REG 14 - INTERRUPT ENABLE REGISTER



NOTES:
1. Upon a write, if bit 7 is a "0", then each "1" in bits 0-6 disables the corresponding interrupt.
2. Upon a write, if bit 7 is a "1", then each "1" in bits 0-6 enables the corresponding interrupt.
3. Upon a read of this register, bit 7 will be "1" and all other bits will reflect their enable/disable state.

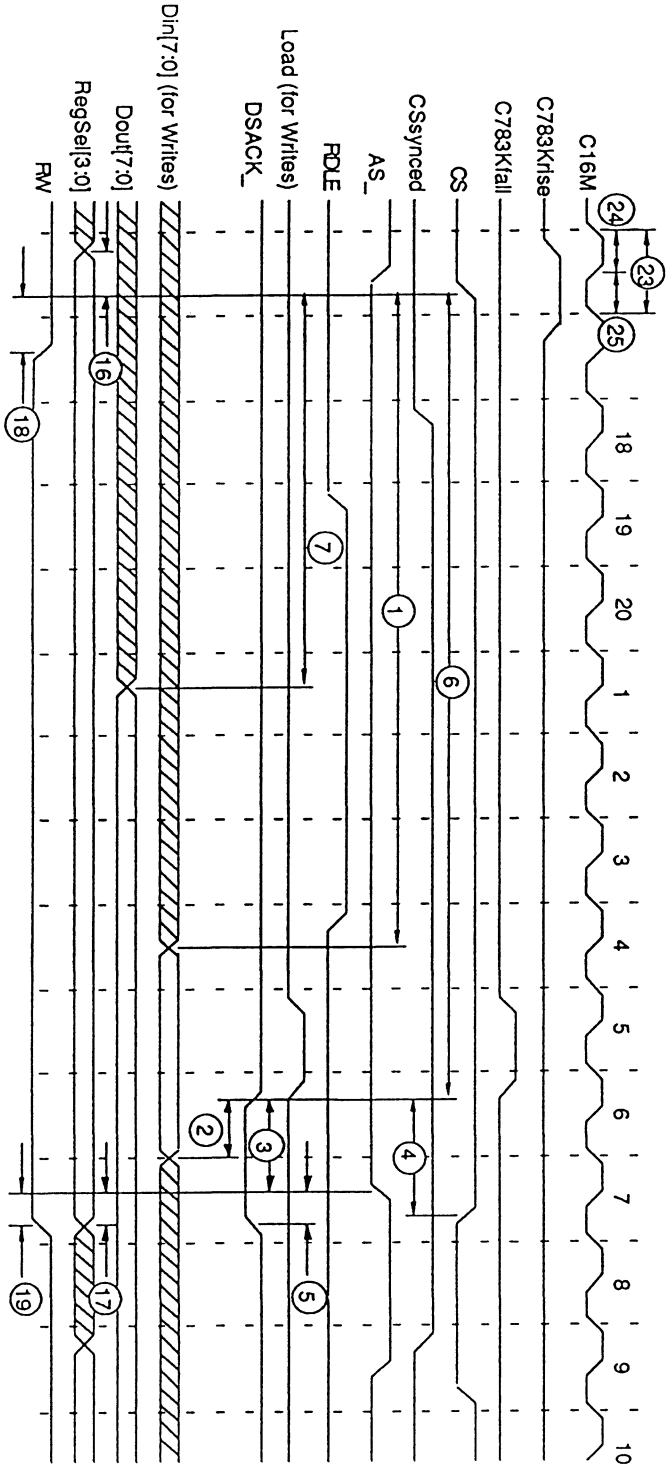
Figure 10 - Interrupt Enable Register (IER)

Number	Characteristic	Min	Max	Unit
1	CS high to Din valid max. allowable delay		500	ns
2	DSACK_ low to Data invalid required hold time	10		ns
3	DSACK_ low to AS_ high required delay	10		ns
4	DSACK_ low to CS low allowable delay		660	ns
5	Delay, AS_ high to DSACK_ high[2]		10	ns
6	Delay, CS high to DSACK_ low[2]	590 (≈9.25 C16M cycles)	1870 (≈29.25 C16M cycles)	ns
7	Delay, CS high to Dout valid[2]	145	1365	ns
8	Delay, C16M high to IRQ_ or CB2out high or low (not shown on timing diagrams)[2]		20	ns
9	CB1 Shift Clock required delay between pos. edges	140		ns
10	CB1 Shift Clock required delay between neg. edges	140		ns
11	Delay, CB1 Shift Clock falling edge to CB2out (shift data out) valid[2]		200	ns
12	CB2in required setup time to CB1 Shift Clock rising edge	70		ns
13	CB2in required hold time from CB1 Shift Clock rising edge	70		ns
14	Required pulse width, CB1 Shift Clock low	70		ns
15	Required pulse width, CB1 Shift Clock high	70		ns
16	RegSel required setup time to CS rising edge	0		ns
17	RegSel required hold time from AS_ rising edge	0		ns
18	Allowable delay, CS high to RW valid		125	ns
19	RW required hold time after AS_ high	0		ns
20	Required delay between active edges on CA1, CA2, CB1, or CB2	140		ns
21	Delay, CA1, CA2, CB1, or CB2 active edge to enabled IRQ_ active[2]		210	ns
22	Required pulse width, CA1, CA2, CB1, or CB2	70		ns
23	Allowable C16M clock period [1]	63.77	63.89	ns
24	Required pulse width, C16M high	25		ns
25	Required pulse width, C16M low	25		ns

[1]The accuracy of the period of C16M determines the accuracy of the Timers. It is therefore necessary to make C16M quite accurate to insure compatibility with existing Macintosh software which uses the Timers. The numbers given above represent 100 ppm deviation in frequency.

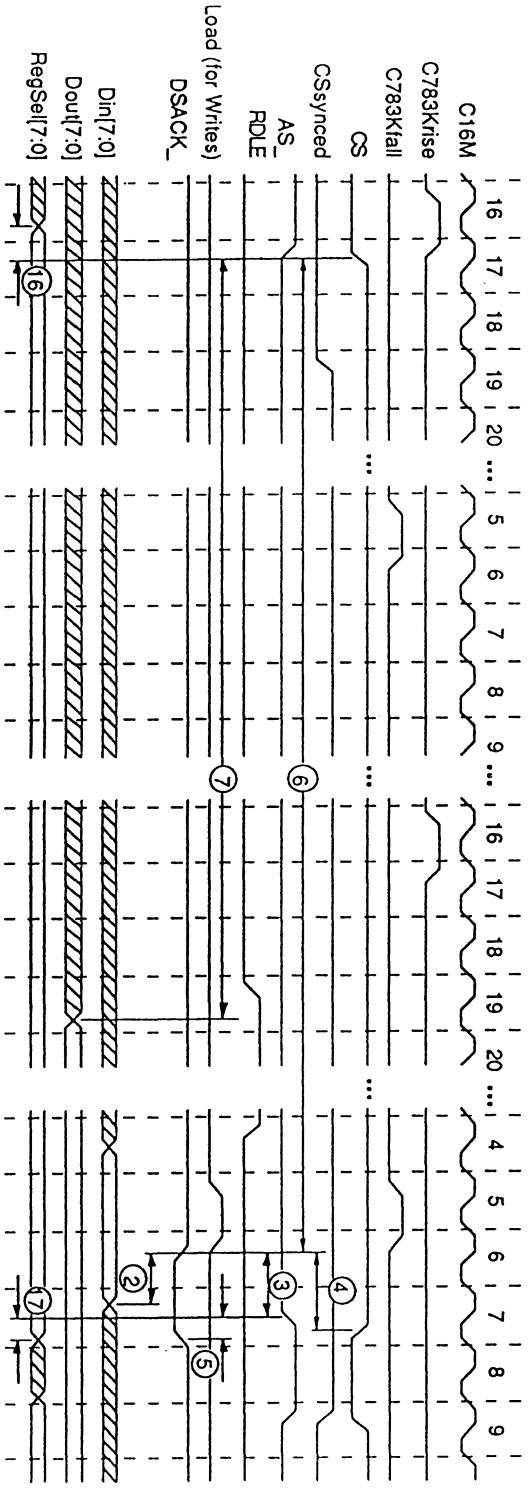
[2]The output delay numbers above represent the worst case delay with a load of 0.5pF. The delay for larger loads is given by the following formula: Actual Delay(ns) = Delay Given Above(ns) + 1.0(ns/pF) * (Actual Loading(pF) - 0.5pF).

Figure 11 - Timing Specifications



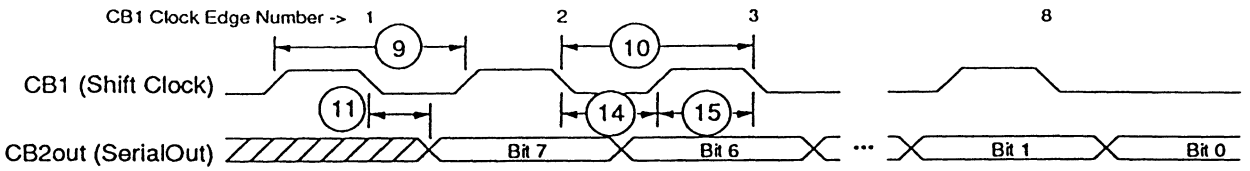
NOTE: C783Krise, C783Kfall, CSsynced, RDLE, and Load are all signals internal to the VIA Cell and are shown only to facilitate an understanding of the schematics. The numbering above the C16M clock refers to BusCntrl state machine cycles.

Figure 12 - Shortest Possible VIA Cell Access Timing



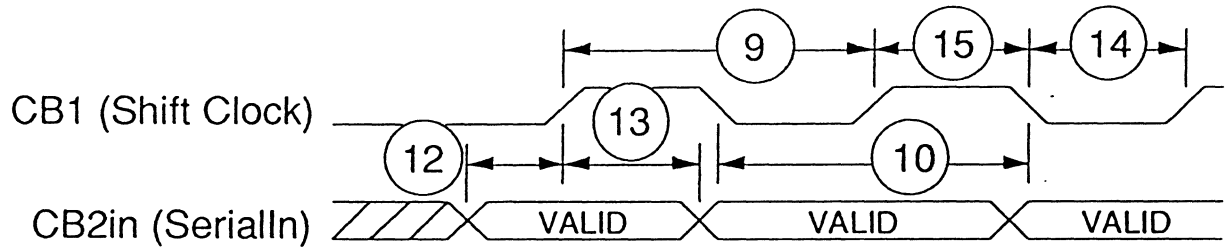
NOTE: C783Krise, C783Kfall, CSsynced, RDLE, and Load are all signals internal to the VIA Cell and are shown only to facilitate an understanding of the schematics. The numbering above the C16M clock refers to BusCntrl state machine cycles.

Figure 13 - Longest Possible VIA Cell Access Timing



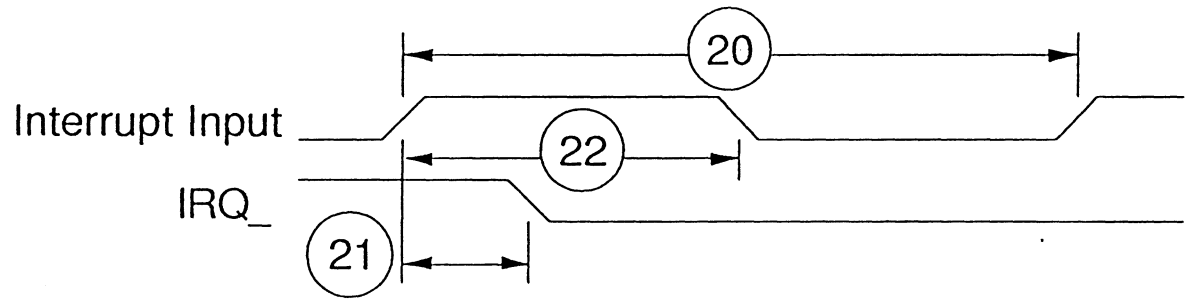
Note that when the Shift Register is programmed to shift out, the falling edge of CB1 is the active edge.

Figure 14 - Shift Register Shift-Out Timing



Note that when the Shift Register is programmed to shift in, the rising edge of CB1 is the active edge.

Figure 15 - Shift Register Shift-In Timing



NOTE: Interrupt Input refers to either CA1, CA2, CB1, or CB2. Although timing spec #20 is shown with the rising edge of the Interrupt Input, it should be understood that this timing spec also applies when the falling edge is the active edge. Also note that timing spec #21 assumes that interrupts have been enabled for the particular Interrupt Input in question, otherwise IRQ_ would not go active. Timing spec #22 specifies the minimum pulse width for active high and active low pulses.

Figure 16 - Interrupt Input Timing

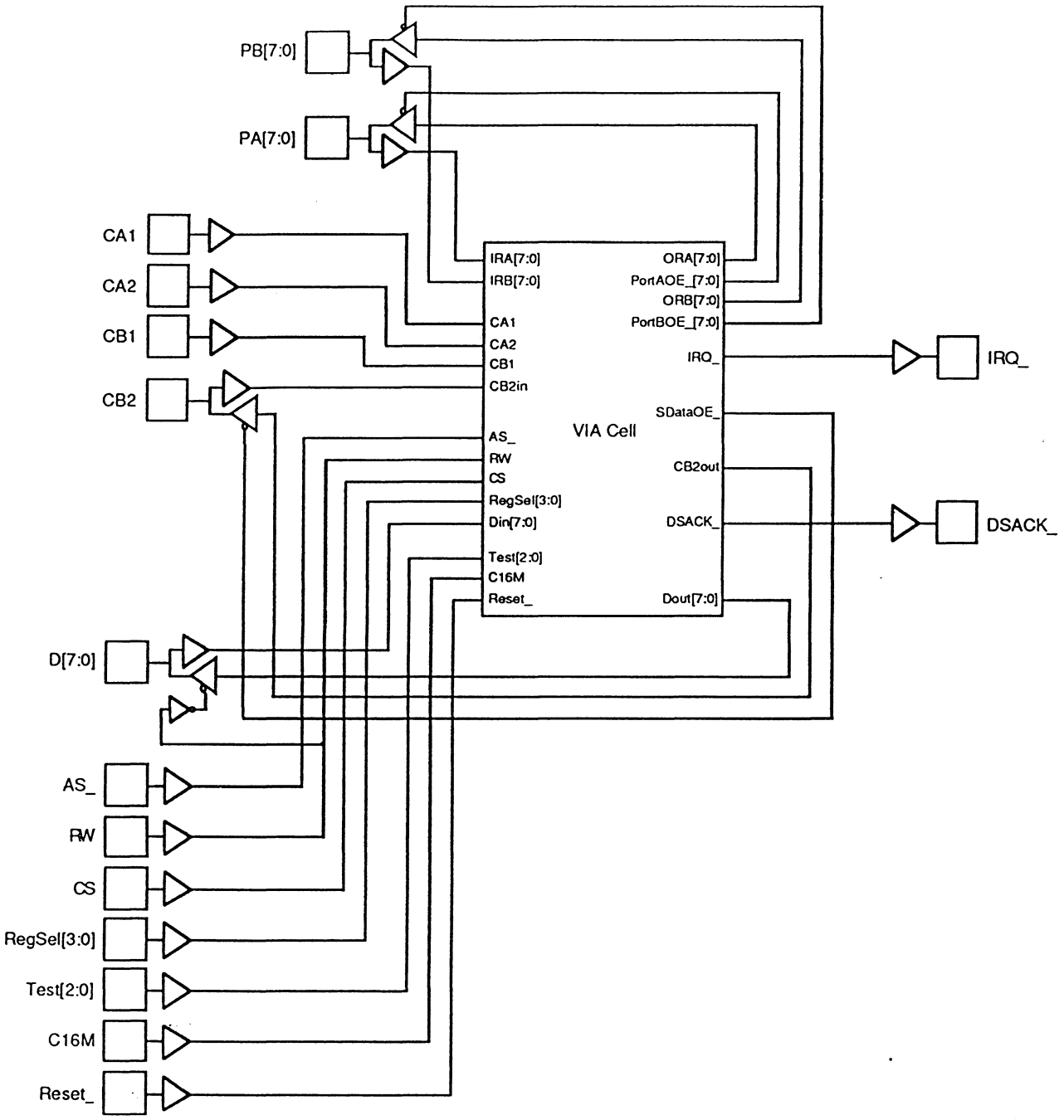


Figure 17 - Using the VIA Cell within a Larger Chip