RadiSys® R380EX

High-Integration Intel386 System Controller

Complete PC-compatible system controller with support for EDO DRAM, ISAbus and power management

System Overview

The RadiSys R380EX Embedded System Controller is a member of the RadiSys family of embedded core logic specifically designed to support the Intel386* EX processor. Providing the necessary circuitry for a PC-compatible embedded system design, it supplies a simple, low-cost, “glueless” interface to additional chips like a video controller or a PCMCIA controller.

The functional design of the R380EX is directly derived from PC architecture. The R380EX includes a DRAM controller, keyboard/mouse controller, real time clock, enhanced IDE interface, and ISAbus controller. The DRAM controller is compatible with both fast-page-mode (FPM) and extended-data-out (EDO) DRAM. It can be configured to support both DRAM SIMMs and flash SIMMs for greater system flexibility. The enhanced IDE interface supports a maximum transfer rate of 8.33MB per second. The ISAbus controller has a separate data bus, and manages the ISA signals to ensure a “quiet” bus for cycles not directed to the ISA address space.

The power management capabilities of the R380EX include clock source switching, halt detection, SMI event generation, and a programmable clock restart

delay to ensure clock frequency settling when restarting the oscillator from a powerdown state. The clock source for the Intel386EX processor can be switched between the R380EX CLK2OSC input and the 32.768KHz real time clock oscillator, thereby reducing the system power consumption.

The R380EX has 4 user-programmable I/O chip selects in addition to the Intel386EX processor’s chip selects. There are 16 bits of individually programmable digital I/O along with 6 bits of digital output. Additional functional blocks handle the Intel386EX processor halt and shutdown cycles, and control the speaker and external LEDs. The main and alternate functions of many of the pins are controlled through registers within the R380EX.

Feature Summary

- Supports Intel386* EX CPU
- True single-chip ISA implementation
- DRAM controller, 512KB to 64MB
- Flash SIMM controller
- Integrated real-time clock
- Enhanced IDE interface
- Keyboard and mouse controller
- Power management
- Supports Intel386EX chip DMA
- 4 programmable I/O chip selects
- 16-bit digital I/O port plus 6-bit output port
- Supports local bus implementation
- ROM or flash ROM interface
- Speaker interface
- Test mode
- 5V or 3.3V operation
- SMI support
- BIOS shadowing
- 208-pin PQFP
- Reference design available

Table of Contents

I. Technical Overview

INTRODUCTION

The RadiSys® R380EX™ is a companion chip for the Intel386™ EX that performs many of the accompanying functions necessary in a PC-compatible embedded system design. The major objectives of the R380EX are low system cost, flexibility, and ease of use in system designs. The R380EX doesn't attempt to incorporate every feature needed in every design. Instead, it incorporates the features needed in most designs, and then provides a simple, low-cost, "glueless" interface to other chips for additional needed functions. Examples include a video controller or a PCMCIA controller.

RECOMMENDED ADDITIONAL DOCUMENTATION

The following reference is a prerequisite for this specification:

- *Intel386 EX Embedded Microprocessor User's Manual*; order number 272485-001

An important feature of the R380EX is support for the IBM PC/AT architecture and for the ISA bus. The primary defining reference for the architecture and the bus is the following:

- *IBM Technical Reference, Personal Computer AT*

The following is an even better reference:

- Edward Solari, *ISA & EISA Theory & Operation*, ISBN 0-929392-15-9

  Note: Please see Errata. This document, in conjunction with the errata, accurately describes the Rev 1.0 production silicon.

INTEL386 EX PROCESSOR SUPPORT

The R380EX connects directly to the Intel386 EX CPU local bus and supports 3.3V and 5V Intel386 EX processors. It supports the C-step Intel386 EX device up through 33 MHz, and also the 5V B-step device up through 25 MHz.

DRAM/FLASH MEMORY CONTROLLER

The memory controller has two operating modes enabling maximum DRAM configurations of 16MB per bank or 64MB per bank. Both modes support two banks of DRAM or flash SIMMs. Each bank can support up to a 32 bit wide Fast Page Mode (FPM) DRAM, Extended Data Out (EDO) DRAM, or a flash SIMM. Each bank can be any size from 512KB to 16MB (or up to 64MB).

The timing parameters for DRAM accesses are configured through the R380EX DRAM control registers.

ROM/FLASH ROM INTERFACE

The ROM/flash ROM interface supports a variety of x8 and x16 ROM/flash devices. Most PC compatible implementations shadow an 8-bit BIOS into DRAM.

The R380EX is specifically designed to support the Intel 28F00xBV-T or similar flash devices. The upper 128KB is normally used for the BIOS and includes a 16KB boot block. The lower 384KB can contain user code. If user code is not needed, then a smaller device could be used (e.g., 28F001BV-T). The BIOS typically consists of a 16KB boot block, and a 112KB System BIOS. Once shadowed, the BIOS running in DRAM can occupy upto 256KB just below 1MB of memory.

ISA BUS INTERFACE

The ISA bus controller manages a separate ISA data bus and the ISA control signals. The controller supports 8- and 16-bit transfers, including translation for 16-bit requests from the CPU to 8-bit ISA devices, and 8-bit requests to 16-bit devices.

The R380EX implements the "quiet ISA bus" feature. When the R380EX sees an access that is not destined for the ISA bus (i.e., a local bus cycle, a non-DMA DRAM cycle, or a cycle internal to the R380EX), it does not drive the ISA bus data or control signals. In addition, ISA bus refresh cycles may be disabled. This reduces overall system power consumption.

The controller also produces the ISA bus clock (SYSCLK) which can be the 2X CPU clock divided by 4, 6, or 8.

POWER MANAGEMENT

The R380EX contains support for System Management Mode (SMM) and for power management of the system environment. The R380EX allows the system designer several different power conservation strategies:

A Halt instruction can be issued from an application to place the system into a very low power mode that supports self-refresh DRAMs. While in this mode, the system's high speed clock can be shutdown, further reducing power. System power consumption can typically be kept to less than 10mA, while still retaining the state of the system. All of this can be done without having to place the Intel386 EX into Powerdown Mode. This allows local-bus peripherals to keep track of the bus phase through halt-resume cycles.

An application can write to the R380EX CPU clock divider control register in to reduce the Intel386 EX processors's clock speed.

An SMI (System Management Interrupt) can be generated in response to an external event or an interval timer expiring in the R380EX. SMI accesses to 20000 to 3FFFF can be automatically remapped to A0000 to BFFFF. The SMI idle latch and timer can be set up to provide a time-out from 1 millisecond to 256 seconds.

The digital I/O or digital outputs may be used in conjunction with external PFETs to selectively power-down peripherals.

KEYBOARD/MOUSE CONTROLLER

The R380EX contains a PC/AT compatible keyboard controller with PS/2 compatible mouse controller extensions. Response to keyboard commands is immediate because the keyboard controller function is implemented as a hard-wired state machine. Two input port pins associated with the keyboard controller are made available on the R380EX pins. If desired, an external keyboard controller can be supported by the R380EX.

REAL TIME CLOCK

The R380EX integrates a real-time clock providing the functions of the PC date/time clock including: an alarm, programmable periodic interrupt, 114 bytes of battery backed CMOS RAM, PC I/O registers 070h and 071h, and the crystal and battery inputs. If desired, an external RTC can be supported by the R380EX.

ENHANCED IDE INTERFACE

The R380EX supports the ATA-2 specified programmed I/O mode 3 and 4 for IDE drives at a maximum transfer rate of 6MB/second. The R380EX powers up with mode 0 timing to support both older IDE drives that are incapable of interfacing at higher speeds and to support the power-on mode of the newer EIDE drives.

DMA CYCLE

The R380EX coordinates the ISA bus and the DRAM controllers state machines enabling the use of the Intel386 EX 's internal DMA channels with external DMA-controlled devices. The R380EX provides the proper control signals for mating the memory access with the ISA I/O access.

PROGRAMMABLE CHIP SELECTS

The R380EX contains four user-programmable I/O chip selects in addition to those provided by the Intel386 EX embedded processor. The chip selects can be programmed and used individually or they can be "daisy-chained" together to form one extended chip select output that can decode up to four regions.

16-BIT DIGITAL I/O PORT

The R380EX has a 16-bit digital I/O port, and a 6-bit digital output port. Four registers are associated with the DIO port: the DIO Data Input Register, the DIO Data Output Register, the DIO Direction Control Register, and the DIO Mux Register. The DIO Direction and Mux registers can be initialized at power-up by configuration resistors.

PC ENGINE LOGIC

The R380EX performs support functions for the Intel386 EX chip and the PC hardware architecture. The R380EX has two types of internal configuration registers: PC compatible registers, and R380EX specific registers. The PC compatible registers are defined by the PC architecture and are addressable at their standard I/O addresses.

The R380EX generates clock signals for the Intel386 EX embedded processor. The R380EX also handles the variety of reset signals and states expected in a PC compatible system, including a cold reset (PWRGOOD) input, and a watchdog time-out (WDTIN) input. It also contains the PC-compatible Port B Register (061h).

PC-COMPATIBLE R380EX REFERENCE DESIGN

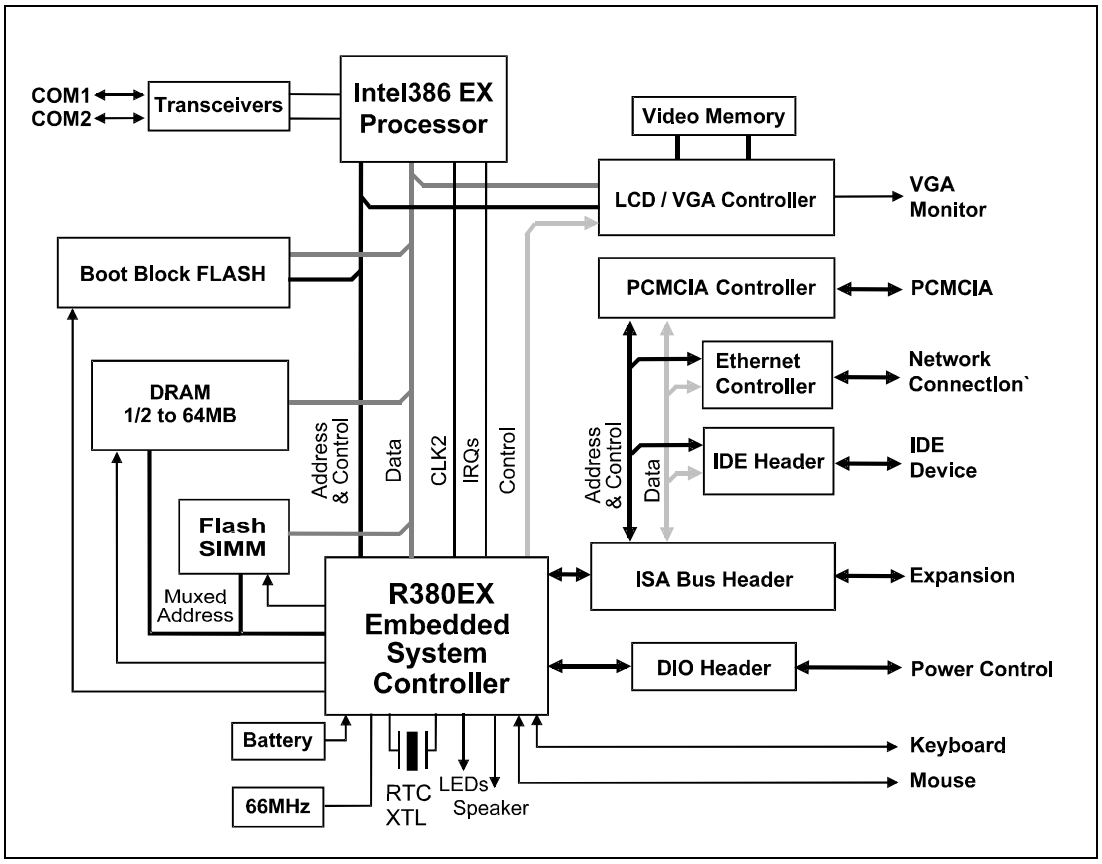
RadiSys has worked with several vendors in developing a PC-compatible EXPLR2 reference design. RadiSys highly recommends use of this reference design as the starting point for any system using the R380EX. By starting with this proven reference design your path to success is shortened, and your BIOS options are increased. Several BIOS vendors have developed R380EX BIOS adaptation kits based upon this reference design. The EXPLR2 reference design is available on the RadiSys home page at http://www.radisys.com.

The RadiSys Advantage

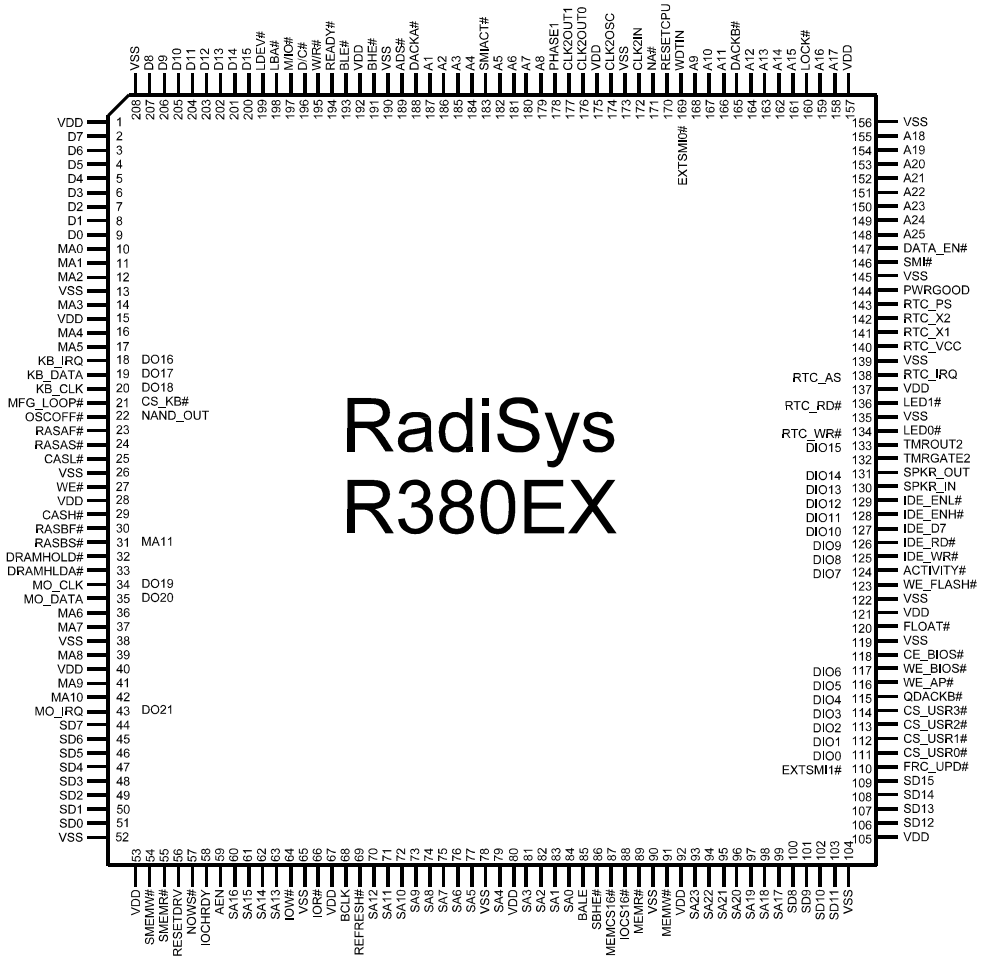
At RadiSys our core competency is the creation of embedded PC-compatible computers. We developed the R380EX to address the challenges that we experience in creating these designs. RadiSys is sensitive to the issues of long-term support for embedded system applications and we are commited to long-term support of the R380EX.

Unlike designing with short-lived parts from the commodity PC chipset market, this commitment makes the RadiSys R380EX a safe design choice.

SAMPLE SYSTEM USING THE R380EX



RADISYS R380EX PINOUT



II. Signal Definitions

Table 2.1: Signal Type Abbreviations

Abbreviation	Description
trailing #	The named signal is active low.
cmos in	CMOS level input.
ttl in	The input threshold for these pins is switchable, as a single group, between TTL and CMOS thresholds, by a power-up configuration resistor.
out	CMOS level output. Most outputs can drive a DC load of 8 mA. The ones that can't are labeled 2mA. All outputs, except for NAND_OUT, are actually implemented as bi-directional pins for test purposes.
tristate out	CMOS level output that can become tristated under certain conditions.
cmos io	combined: cmos in out
ttl io	combined: ttl in out
2mA	Output buffer has only 2mA drive strength. Most output buffers can drive 8mA.
csr	The output buffer has a controlled slew rate. This reduces ground bounce at the expense of an increase in propagation delay.
open drain	Open drain output buffer. Actively drives low; requires an external pull-up resistor.
bus keeper	The pin has a bus keeper circuit that maintains the previous pin state. This uses active CMOS buffers, rather than resistors, to save power. These buffers are very weak and can easily be overdriven.
pull-up	The input has an internal pull-up device.
schmitt	The input has a Schmitt trigger.
power	Power pin.

Table 2-2: Clock and Reset Signals

Signal	Type	Description
CLK2OSC	cmos in	2x Clock from Oscillator: The double-frequency input clock from the system oscillator. This input must be 66 MHz for systems that run 33 MHz bus cycles. The R380EX rebuffers this input and outputs it (or a sub-multiple) on CLK2OUT[1:0]. It is of paramount importance to have a high quality signal on this input. The clock must be free of excessive overshoot and undershoot and must be monotonic, especially at the CMOS switching threshold of the input buffer (nominally 50% of VCC). Series termination or equivalent design techniques should be employed.
CLK2OUT[1:0]	out 16mA	2x Output Clocks: Two identical double-frequency outputs which should be used for system clock distribution. These two outputs have low skew between each other when they are terminated to equivalent loads. Connect one of these outputs to the Intel386 EX CLK2 input and also to the R380EX CLK2IN input. In normal operation the R380EX generates these outputs from CLK2OSC or from a sub-multiple thereof. In standby mode the R380EX can optionally generate these outputs from the 32.768 KHz clock on RTC_X1.

Table 2-2: Clock and Reset Signals

Signal	Type	Description
CLK2IN	cmos in	2x Clock Input: The double-frequency master clock. This is the primary clock input for the R380EX. This must input must be connected to one of the CLK2OUT outputs to insure phase synchronization to the Intel386 EX It is of paramount importance to have a high quality signal on this input. The clock must be free of excessive overshoot and undershoot and must be monotonic, especially about the CMOS input switching threshold (nominally 50% of VCC). Series termination or equivalent design techniques should be employed.
PHASE1	out csr	Intel386 EX Phase1: This output indicates which clock cycle of the double-frequency CLK2OUT output corresponds to phase PH1 of the Intel386 EX internal clock. It transitions a few nanoseconds after the rising edges of CLK2OUT[1:0], when both outputs are terminated to equivalent loads. PHASE1 may be used as a qualifier for external logic clocked by CLK2OUT.
PWRGOOD	cmos in schmitt	Power Good: The assertion of this input indicates that system power is stable and that normal operation may begin. While PWRGOOD is negated, RESETCPU and RESETDRV outputs are asserted, and the internal Real Time Clock is placed in standby. The R380EX internally latches its power-up configuration from MA[10:0] as PWRGOOD becomes asserted. This input buffer is powered by RTC_VCC and is active even when system power is off.
RESETCPU	out	Reset Intel386 EX CPU: Connect this output to the Intel386 EX RESET input. It is called RESETCPU to indicate that its primary purpose is to reset the CPU, but may also be used to reset other logic. RESETCPU is also used internally by the R380EX to initialize various registers and state machines. This output is asserted asynchronously in response to PWRGOOD, WDTIN, an Intel386 EX shutdown cycle, or a PORT64 reset, if enabled. RESETCPU is held asserted for 32 CLK2IN cycles past the initiating input. It is negated synchronously to CLK2IN and is used by the R380EX to synchronize to the bus phase of the Intel386 EX. While RESETCPU is asserted the R380EX switches the CLK2OUT output frequency to one- quarter the CLK2OSC input frequency. This also changes the CLK2IN frequency, because it is normally connected externally to CLK2OUT. This allows the R380EX to deassert the RESETCPU signal after the *falling* edge of CLK2IN and easily meet both the *rising* edge CLK2 setup and hold time parameters required by the Intel386 EX and other local bus logic. Within sixteen bus cycles after the deassertion of RESETCPU, the R380EX automatically glitchlessly switches the CLK2OUT frequency back to the undivided CLK2OSC input frequency. The R380EX internally latches its power-up configuration only when PWRGOOD becomes asserted. None of the other sources of reset change this configuration, nor do they change whether the DIO pins, or DO pins, are I/O or the alternate functions. The ISA Bus interface signal RESETDRV is an output that has similar timing but has slightly different functionality. The differences are that RESETDRV will not be asserted as a result of a PORT64 reset, and that RESETDRV is negated synchronously to BCLK. Note: This output isn't the same as the CPU Reset function that is controlled by the Intel386 EX PORT92 Register. That register resets only the CPU core and is handled internally by the Intel386 EX.

Table 2-2: Clock and Reset Signals

Signal	Type	Description
WDTIN	ttl in	Watchdog Timer Input: If enabled, this input causes RESETCPU and RESETDRV to be asserted. Typically, it should be connected to the Intel386 EX WDTOUT output. Doing so will initiate a reset of the system if the CPU watchdog timer expires. Alternate Function: EXTSMI0# -- See IDE and Miscellaneous Signals

Table 2.3: Intel386 EX CPU Local Bus Interface Signals

Signal	Type	Description
A[25:1]	ttl in	Address Bus: Connect directly to the corresponding Intel386 EX address bus outputs. The address, in conjunction with BHE#, BLE#, M/IO#, D/C#, W/R# and optionally SMIACT#, selects a physical memory or I/O location. Certain address ranges can optionally be remapped in various power management modes.
ADS#	ttl in	Address Status: Connect directly to the corresponding Intel386 EX output. Indicates that the CPU is driving a valid address and bus cycle definition. The R380EX tracks the state of the local bus by monitoring this input at all times. When the Intel386 EX leaves the special HALT POWERDOWN mode and bit 4 of the Clock/Reset Control Register is set, the R380EX uses the ADS# input to re-synchronize the R380EX's PHASE1 output to the Intel386 EX clock phase PH1. This is easily accomplished because the system clock frequency is 32.768 KHz during powerdown mode and thus the ADS# signal is guaranteed to be valid very early in PH1 of the Intel386 EX's bus cycle.. This resynchronization allows the R380EX, and other local bus peripherals that look at the R380EX's PHASE1 output, to stay in phase with even an A- or B-step Intel386 EX as it exits its lowest power state.
BHE# BLE#	ttl in	Byte High Enable, Byte Low Enable: Connect directly to the corresponding Intel386 EX outputs. These inputs indicate which data bytes are valid. BHE# BLE# 0 0 word transfer 0 1 upper byte transfer (D[15:8]) 1 0 lower byte transfer (D[7:0]) 1 1 memory refresh (not acted on) Note: The R380EX uses the 32.768 KHz clock to initiate DRAM refresh. It completely ignores CPU generated refresh cycles, and does not drive READY# for those cycles. Refresh cycle generation should not be enabled in the Intel386 EX, otherwise the refresh cycle will never complete and the system will hang.
D[15:0]	ttl io csr bus keeper	Data Bus: Connect directly to the corresponding Intel386 EX signals. The R380EX links the CPU data bus with the ISA data bus. Except for D0, these signals have bus keepers that keep them from floating when the data bus is not being driven. D0 does not have a bus keeper. This makes it easy to use D0, together with an external pull-up resistor, to detect the presence of EDO-type DRAM.

Table 2.3: Intel386 EX CPU Local Bus Interface Signals

Signal	Type	Description
DACKA# DACKB#	cmos in	DMA Acknowledge: Connect directly to the Intel386 EX DACK0# and DACK1# outputs. These inputs indicate that the current bus cycle is a DMA transfer. The CPU must be programmed to perform fly-by DMA transfers. The address during DMA cycles is interpreted as a memory address. The R380EX will mate a memory read to a concurrent ISA bus I/O write, and will mate a memory write to a concurrent ISA bus I/O read. DACKB# is re-driven on the QDACKB# (Labeled QDACKA# on Rev 0 Silicon) output after being internally qualified to prevent it from becoming active during an ISA refresh cycle. There is no corresponding QDACKA# output (QDACKB# on Rev 0 Silicon). Note: These inputs are not called by the same names as the Intel386 EX outputs. This is to make it clearer that either Intel386 EX output may be connected to either input, depending on which channel needs a QDACK output.
LBA#	cmos in	Local Bus Access: Connect directly to the corresponding Intel386 EX output. This input indicates that the CPU is either doing an internal register access or is doing a local bus access as selected by the Intel386 EX chip-select unit. The Intel386 EX will assert READY# itself to complete the access. The R380EX tracks the state of the local bus by monitoring this input at all times. The R380EX does not use LBA# as a qualifier for deciding when to assert CE_BIOS# or WE_FLASH#. Except for these two outputs, the R380EX ignores local bus cycles when LBA# is asserted. The R380EX will not access any of its internal peripherals or registers, nor will it perform a DRAM or ISA access during cycles with LBA# asserted.
LDEV#	ttl in	Local Device: An external device that wishes to claim the current local bus access for itself must assert this input to notify the R380EX. The external device must generate READY# to complete the access. The R380EX tracks the state of the local bus by monitoring this input at all times. The R380EX completely ignores bus cycles for which LDEV# is asserted. It will not assert CE_BIOS# or WE_FLASH# for these cycles. It will not access any of its internal peripherals or registers, nor will it perform a DRAM or ISA access during cycles with LDEV# asserted.
LOCK#	ttl in	Bus Lock: Connect directly to the corresponding Intel386 EX output. This input indicates that the current bus access is part of a Read-Modify-Write sequence from the CPU. The R380EX will not relinquish control of the DRAM via the DRAMHOLD# and DRAMHLDA# protocol while this signal is asserted.

Table 2.3: Intel386 EX CPU Local Bus Interface Signals

Signal	Type	Description
M/IO# D/C# W/R#	ttl in	Bus Cycle Definitions (Memory/IO, Data/Control, Write/Read): Connect directly to the corresponding Intel386 EX outputs. These three inputs define the current bus cycle type: M/IO# D/C# W/R# 0 0 0 interrupt acknowledge 0 0 1 Intel Reserved † 0 1 0 I/O data read 0 1 1 I/O data write 1 0 0 memory code read 1 0 1 halt or shutdown* 1 1 0 memory data read** 1 1 1 memory data write * For halt or shutdown cycles: A1 = 0 indicates Shutdown Cycle A1 = 1 indicates Halt Cycle ** The Intel386 EX defines refresh cycles as memory data read with REFRESH# asserted and both BHE# and BLE# negated. The R380EX does not connect to REFRESH# and does not respond to Intel386 EX refresh cycles. The R380EX distinguishes reads from refreshes by the BHE# and BLE# inputs. † I/O Code Write cycles are sometimes used on microprocessor emulators to keep track of "branch taken messages."
NA#	out 16mA	Next Address: Connect directly to the corresponding Intel386 EX input. This output requests address pipelining from the CPU. It indicates that the R380EX can accept a new address and bus cycle definition prior to completing the current DRAM access. NA# is asserted only for normal DRAM accesses, never for ISA accesses, DMA accesses or internal register or peripheral accesses.
READY#	ttl io csr	Bus Ready: Connect directly to the corresponding Intel386 EX signal. The R380EX asserts this bi-directional signal to indicate that the current bus cycle has completed. The R380EX tracks the state of the local bus by monitoring READY# at all times, but it does not drive READY# for cycles with LBA# or LDEV# asserted.
SMI#	out 2 mA	System Management Interrupt: Connect directly to the corresponding Intel386 EX input. This output is asserted whenever any system management event occurs, or whenever either of the EXTSMI[1:0]# inputs is asserted. It remains asserted for a minimum of eight CLK2 cycles. ***** Note *****: The exact behavior of this output is to be determined with respect to multiple concurrent SMIs.
SMIACT#	cmos in	System Management Interrupt Active: Connect directly to the corresponding Intel386 EX input. This input is the acknowledgment from the Intel386 EX that it has entered system management mode. Various CPU address ranges may be remapped in this mode through the SMI Control/Status Register in the R380EX.

Table 2.4: DRAM Interface Signals

Signal	Type	Description
CASH# CASL#	tristate out	Column Address Strobe: These outputs strobe a column address into DRAM. CASL# enables reading or writing the lower data byte, bits D[7:0], and corresponds to BLE#. CASH# enables reading or writing the upper data byte, bits D[15:8], and corresponds to BHE#.
DATA_EN#	out csr	Data Enable: This output may be connected to the OE# input of an optional external data transceiver between the CPU data bus and the DRAM and BIOS data busses. The DIR input of the transceiver should be connected to the Intel386 EX RD# output. Register control bits can selectively enable this output for accesses to either the DRAM or the BIOS. By providing controls for an external transceiver, this output facilitates using DRAM and/or BIOS chips that have slow output disable times that may encroach into a subsequent Intel386 EX write cycle and consequently fight with the CPU data outputs. Typical DRAM output disable timings are good enough that a transceiver is not needed and is not recommended. However, some EPROMs may need it. Alternate Function: DRAMHLDA# (described below)
DRAMHOLD#	ttl in	DRAM Hold Request: An external device may share the DRAM by asserting DRAMHOLD#. After the R380EX completes any pending CPU accesses and memory refreshes it tristates its memory address and control outputs and asserts DRAMHLDA#, provided that LOCK# is not asserted. The external device may then access the DRAM. The external device must keep DRAMHOLD# asserted for the duration of its DRAM access(es) and must negate it after its final DRAM access. If the external device continuously maintains control of the DRAM for longer than one memory refresh interval (typically 15.3 microseconds), it must perform distributed CAS-before-RAS refreshes at suitable intervals. Note: The R380EX actively negates all control outputs before tristating them and relinquishing the DRAM. Nevertheless, system design practice suggests that pull-up resistors be added to the control outputs to prevent stray noise from affecting the DRAM.
DRAMHLDA#	out csr	DRAM Hold Acknowledge: An external device sharing the DRAM must request access by asserting DRAMHOLD#. The R380EX responds with DRAMHLDA# after pending CPU accesses have completed.
MA[11:0]	cmos io	Memory Address: There are 11 dedicated multiplexed DRAM memory address outputs. The 12th output, MA11, is multiplexed with RASBS#, and its function is selected at power-up by detecting the state of MA3. If MA3 is low, the RAS_BS# pin function is selected. If it is high, then the MA11 pin function is selected. During normal operation the 11 (or 12) outputs form a 22-bit (or 24-bit) multiplexed memory address bus. Those DRAMs which have the same number of row and column address bits, or those with one more row address bit than column address bit are supported. The extra row address bit is the most significant address bit. These pins are uses as inputs, while PWRGOOD is negated, to set the power-up configuration of the R380EX. These pins don't have bus keepers or internal pull-up resistors. External programming resistors are required for all pins. When the R380EX is in low-power standby, it redrives all these pins with the values latched at the assertion of PWRGOOD. Alternate Function: MA11 is multiplexed with RASBS#.

Table 2.4: DRAM Interface Signals

Signal	Type	Description
RASAF# RASAS# RASBF# RASBS#	tristate out	Row Address Strobe: These outputs strobe a row address into DRAM. There are four RASxx# signals, which are used to control two independent DRAM banks. Only one RASxx# is asserted at a time. The R380EX operates the DRAM in Fast Page Mode. The two memory banks, bank A and bank B, may have one or two 16-bit wide DRAM sub-bank, called the First and the Second sub-bank. For each sub-bank the upper and lower data bytes are controlled by the two CASx# outputs. If a bank has only a single 16-bit wide DRAM sub-bank installed, the R380EX controls it with the RASxF# output. If a bank has two 16-bit wide DRAM sub-banks installed, the R380EX controls them using both the RASxF# and RASxS# outputs. The R380EX multiplexes the RASBS# function with the MA11 function on the same pin. This function is selected at power-up by detecting the state of MA3. This means that when the MA11 function is selected during power-up configuration, the R380EX only supports a total of three memory sub-banks, and the bank B second DRAM sub-bank is not available. Alternate Function: RASBS# is multiplexed with MA11.
WE#	tristate out	DRAM Write Enable: This output directs the DRAMs to prepare for an Early Write cycle. There is only one WE# output because the data bytes to be written to the 16-bit-wide banks are selected by the subsequent assertion of CASL# or CASH#. If either bank of DRAM is programmed to be Extended Data Out, the R380EX will assert WE# briefly after every read cycle to disable the DRAM output buffers. This allows EDO DRAMs to be used should normal Fast Page Mode DRAMs become more expensive or more difficult to obtain.

Table 2.5: ISA Bus Interface Signals

Signal	Type	Description
AEN	out csr	Address Enable: This output is asserted only during DMA cycles. It indicates that the SA outputs to the ISA bus are driving a memory address and not an I/O address. (ISA DMA cycles are fly-by and never have an I/O address associated with them.) ISA peripherals accessed via I/O instructions must disable their address decode while this output is asserted, since otherwise they would incorrectly respond to memory addresses when the IOR# or IOW# signal becomes asserted together with MEMR# or MEMW# during a DMA cycle. I/O devices participating in a DMA cycle should select themselves using DACKx# directly from the Intel386 EX.
BALE	out	Bus Address Latch Enable: This output is pulsed asserted at the beginning of a normal ISA cycle. It is continuously asserted during DMA cycles. It indicates that a valid address is present on the bus. In a generic ISA system this signal is used to latch the high order bits of the address bus. The R380EX keeps all addresses stable for the duration of the ISA cycle, so they don't need to be latched by BALE.
BCLK	out	Bus Clock: The ISA bus clock signal. This can be set to CLK2IN divided by 4, 6 or 8 as determined by bits 2 and 3 of the Clock/Reset Control Register. For all divisors the output duty cycle is ~50%. The ISA bus is generally asynchronous, and BCLK shouldn't be used to sample inputs or drive outputs. The one exception is NOWS#, which is synchronously sampled by the R380EX. Note: The IBM Technical Reference, Personal Computer AT refers to this output as CLK in the pin descriptions, but refers to it as SYSCLK in the actual schematics. This document uses the name chosen by Solari in ISA & EISA Theory & Operation.
IOCHRDY	ttl in	I/O Channel Ready: In order to extend a cycle, an ISA device must negate IOCHRDY within a specific time after the assertion of MEMR#, MEMW#, IOR#, or IOW#. When the device is ready for the cycle to complete, it must release IOCHRDY and allow the pull-up resistor to assert it. Alternatively, to speed up the completion of the cycle, the device may actively drive IOCHRDY asserted and then tristate it. On the ISA bus this is an open-collector signal implementation, held asserted by a pull-up resistor. It may be negated by an ISA memory or I/O device to extend the duration of the current ISA cycle. IOCHRDY negated overrides NOWS# asserted. Normal ISA cycles are extended by integral numbers of BCLK cycles. ISA DMA cycles are extended by an even number of BCLK cycles. Because the DRAM refresh controller only remembers a single pending refresh, if IOCHRDY is held negated for too long, ISA and DRAM refresh cycles may be lost.
IOCS16#	ttl in	I/O Chip Select 16: On the ISA bus this is an open-collector signal implementation, held negated by a pull-up resistor. An ISA bus I/O device that can support 16-bit accesses may assert IOCS16# via a straight decode of SA[9:0] without qualification by any control strobes. When this input is asserted, the R380EX performs, a 16-bit access to the device, if possible. The PCS Registers inside the R380EX may also assert an internal version of this signal and thus save an external open-collector buffer otherwise would be required by 16-bit wide I/O mapped peripherals. IOCS16# is ignored by the R380EX except during ISA I/O cycles.

Table 2.5: ISA Bus Interface Signals

Signal	Type	Description
IOR#	out	I/O Read Strobe: This command strobe output instructs the selected ISA I/O device to drive data onto the data bus. An I/O device is selected by an address during an I/O cycle and by a DACKx# during a DMA cycle.
IOW#	out	I/O Write Strobe: This command strobe output instructs the selected ISA I/O device to accept data from the data bus. An I/O device is selected by an address during an I/O cycle and by a DACKx# during a DMA cycle.
MEMCS16#	ttl in	Memory Chip Select 16: On the ISA bus this is an open-collector signal implementation, held negated by a pull-up resistor. An ISA bus memory device that can support 16-bit accesses may assert MEMCS16# via a straight decode of SA[23:0] without qualification by any control strobes. This tells the R380EX to perform, if possible, a 16-bit access to the device. MEMCS16# is ignored by the R380EX except during ISA memory cycles. In a generic ISA bus, SA[16:0] don't have the proper timing to allow them to participate in decoding MEMCS16#. The R380EX drives all SA lines with the same timing, so all 24 may be used. Or, for backward compatibility with generic ISA, the decode may be limited to using SA[23:17].
MEMR#	out	Memory Read Strobe: This command strobe output instructs the selected ISA memory device to drive data onto the data bus. A memory device is always selected by an address, never by a DACKx#. This output is asserted together with REFRESH# during an ISA refresh cycle.
MEMW#	out	Memory Write Strobe: This command strobe output instructs the selected ISA memory device to accept data from the data bus. A memory device is always selected by an address, never by a DACKx#.
NOWS#	ttl in	No Wait State: On the ISA bus this is an open-collector signal implementation, held negated by a pull-up resistor. Unlike all other inputs from the ISA bus, NOWS# is synchronous. It must meet specified setup and hold times to BCLK. The sampling point varies with the cycle type. Solari, ISA & EISA Theory & Operation, should be consulted in order to understand the subtleties of designing with this signal. IOCHRDY negated overrides NOWS# asserted. NOWS# is sampled only in memory and I/O cycles and is ignored during DMA and refresh cycles. Note: The IBM Technical Reference, Personal Computer AT refers to this pin as 0WS. This document uses the name chosen by Solari in ISA & EISA Theory & Operation. An alternate name used interchangeably by Solari is SRDY#.

Table 2.5: ISA Bus Interface Signals

Signal	Type	Description
QDACKB# (Labeled QDACKA# on Rev 0 Silicon)	out	Qualified DACKB: The QDACKB# output (Labeled QDACKA# on Rev 0 Silicon) is an R380EX internally qualified version of DACKB#. It is qualified to prevent it from becoming active during an ISA refresh cycle. This output is guaranteed to never become active while the R380EX is doing a refresh cycle on the ISA bus. It may be used by a DMA device instead of a DACK signal directly from the Intel386 EX. If ISA refresh is enabled, then a DACK output from the Intel386 EX could potentially become active while the R380EX is performing a refresh cycle on the ISA bus. This could cause problems with some ISA DMA devices. Note: There is no corresponding QDACKA# output. (QDACKB# on Rev 0 Silicon) Therefore, the R380EX can only qualify one of the Intel386 EX DACK pins. However, if ISA refresh is disabled then both Intel386 EX DACK# outputs may be used directly, without any need for qualification. Alternate Function: DIO4
REFRESH#	out csr	Refresh: This output indicates that the current ISA cycle is a refresh cycle. Since ISA refresh is rarely used, it may be disabled by clearing a bit in the DRAM Timing Register. Note: The R380EX does not support add-on bus masters (i.e. Master Mode DMA). Therefore, contrary to the ISA spec, the R380EX REFRESH# signal is a totem pole output, not an open collector output.
RESETDRV	out csr	Reset Drive: This output indicates an ISA bus reset. It has the same function as RESETCPU, but has higher drive capability and allows RESETCPU to be isolated from the noisy ISA bus. RESETDRV is guaranteed to be asserted for at least 10 BCLK cycles. Note: Solari in ISA & EISA Theory & Operation refers to this output as RESET. To avoid confusion with any other reset signals in the system, this document uses the name specified in the IBM Technical Reference, Personal Computer AT.
SA[23:0]	out csr	System Address: These 24 outputs, together with AEN and SBHE#, provide a memory address or an I/O address for the ISA bus. The R380EX converts the address inputs from the Intel386 EX into the ISA bus SA outputs. Generally, there is no way of knowing in advance whether the ISA cycle will be to memory or to I/O until one of the command strobes becomes asserted. Each ISA device must wait for a command strobe before acting on the address. IOCS16# and MEMCS16# are exceptions. The device must generate them by decoding SA in advance of a command strobe. The timing of all 24 address lines is identical, and any or all of them may be used to generate MEMCS16#. The x86 architecture supports 16 I/O address lines, and any or all of SA[15:0] may be used to generate IOCS16#. However, the ISA bus generally uses only SA[9:0] to decode I/O addresses. Note: A generic ISA bus has two types of addresses: SA[19:0] are stable through out a transfer, and LA[23:16] may change before a transfer completes. BALE is used in a generic ISA bus to latch the LA signals before they change. The R380EX keeps all 24 addresses stable through out an entire ISA cycle, so there is no need to latch them. For this same reason, the high order address bits are all named SA, even though a generic ISA bus doesn't have SA[23:20]. The SA outputs from the R380EX may be used as either SA addresses or LA addresses, as desired.

Table 2.5: ISA Bus Interface Signals

Signal	Type	Description
SBHE#	out csr	System Byte High Enable: This output indicates a data transfer on SD[15:8], the upper byte of the ISA data bus. 16-bit devices use SBHE# to condition their upper data bus buffers.
SD[15:0]	ttl io csr	System Data: These signals comprise the ISA data bus. The R380EX links the ISA data bus to the CPU data bus. 8-bit devices always use SD[7:0] only, regardless of whether they are at an odd or an even address. 16-bit devices use SD[15:0], at an even address only, and must inform the R380EX of their capability by asserting either IOCS16# or MEMCS16#.
SMEMR#	out	Small Memory Read Strobe: This command strobe output has the identical function and timing as MEMR#, except that it is further qualified by being asserted only in the lowest megabyte of ISA memory address space (SA[23:20] == 0).
SMEMW#	out	Small Memory Write Strobe: This command strobe output has the identical function and timing as MEMW#, except that it is further qualified by being asserted only in the lowest megabyte of ISA memory address space (SA[23:20] == 0).

Table 2.6: Real Time Clock Interface Signals

Signal	Type	Description
RTC_AS	out 2 mA	Real Time Clock Address Strobe: When the internal Real Time Clock megacell is disabled, this output is the address latch strobe for the external RTC chip. It typically is connected to the AS input of an MC146818 compatible RTC chip. Alternate Function: RTC_IRQ
RTC_IRQ	out 2 mA	Real Time Clock Interrupt: When the internal Real Time Clock megacell is enabled, this output is the interrupt request from the megacell to the Intel386 EX. It is commonly connected to the INT4 input of the Intel386 EX (corresponds to ISA IRQ8). Alternate Function: RTC_AS
RTC_PS	cmos in	Real Time Clock Power Sense: This input, when negated, clears the VRT (valid RAM and time) bit in Register D of the internal Real Time Clock megacell. This indicates that RTC battery power has failed, and that the RTC no longer has valid RAM or time. This input buffer is powered by RTC_VCC and is active even when system power is off.
RTC_RD#	out csr	Real Time Clock Read Strobe: When the internal Real Time Clock megacell is disabled, this output is the read strobe for an external RTC chip. It typically is connected to the DS input of an MC146818 compatible RTC chip. Alternate Function: LED0#

Table 2.6: Real Time Clock Interface Signals

Signal	Type	Description
RTC_WR#	out csr	Real Time Clock Write Strobe: When the internal Real Time Clock megacell is disabled, this output is the write strobe for the external RTC chip. It typically is connected to the R/W input of an MC146818 compatible RTC chip. Alternate Function: LED1#
RTC_VCC	power	Real Time Clock VCC: Provides power only to the internal Real Time Clock megacell, its associated battery-backed CMOS RAM, and a few associated input and output buffers. If using the internal RTC, connect this pin to a power management circuit that switches between system VCC and a backup battery. If not using the internal RTC, connect this pin directly to the system VCC plane.
RTC_X1	cmos in	Crystal input: The input side of the 32.768 KHz crystal oscillator for the internal Real Time Clock megacell. This input must be grounded when supplying an external clock to RTC_X2. This input buffer is powered by RTC_VCC and is active even when system power is off. Note: Pay careful attention to how RTC_X1 and RTC_X2 are hooked up. This arrangement is somewhat different from other oscillator circuits.
RTC_X2	cmos io drives crystal only	Crystal input/output: The output side of the 32.768 KHz crystal oscillator for the internal Real Time Clock megacell. This output should only be connected to a crystal, since it doesn't have the ability to drive other loads. To use the internal oscillator, connect a parallel resonant crystal between RTC_X1 and RTC_X2. Also, connect a 1 Megohm resistor between RTC_X2 and RTC_VCC. Finally, a 10-20 Megohm resistor between RTC_X1 and RTC_X2 must be installed to maintain oscillation when operating from a battery, especially through the system power-up and power-down transitions. To supply an external 32.768 KHz clock, ground RTC_X1 and use RTC_X2 as the clock input. The R380EX needs a 32.768 KHz clock even if using an external RTC chip. The clock is needed for DRAM refresh, as the source of CLK2OUT when the Intel386 EX is in powerdown mode, and as the clock for the SMI timer. This output buffer is powered by RTC_VCC and is active even when system power is off. Note: Pay careful attention to how RTC_X1 and RTC_X2 are hooked up. This arrangement is somewhat different from other oscillator circuits.

Table 2.7: User Chip Selects and Flash BIOS Interface

Signal	Type	Description
CE_BIOS#	out csr	BIOS Chip Enable: This output is the chip select for the system BIOS. It typically is connected to the CE# input of the BIOS chip. The BIOS must reside on the Intel386 EX local bus, since the R380EX does not pass BIOS address or data between the local bus and the ISA bus. The R380EX may be configured for the range of local bus addresses for which it will assert CE_BIOS#.
CS_USR[3:0]#	out csr	User Programmable Chip Selects: These outputs are programmable chip selects for I/O devices located on the ISA bus. Each output is asserted based on a decode of a variable size ISA I/O address, between 1 and 1024 bytes. Each decode may optionally be qualified the address alone, or by the address being logically ANDed with IOR#, or with IOW#, or with [IOR# or IOW#]. In addition, each decode may also optionally assert IOCS16#. This assertion of IOCS16# is used internally to select either 8-bit or 16-bit I/O devices and the corresponding data steering, but it is not driven out on the IOCS16# pin. Also, combinations of the four qualified decodes may optionally be ORed together to create a chip select(s) with "holes" in the decoded address range(s). Finally, the CS_USR0# output may optionally be internally connected as an input that restarts the internal SMI activity timer. Alternate Functions: CS_USR3# is multiplexed with DIO3. CS_USR2# is multiplexed with DIO2. CS_USR1# is multiplexed with DIO1. CS_USR0# is multiplexed with DIO0.
FRC_UPD#	ttl in	Force Update: This input may be read from bit 13 of the SMI Control/Status Register. Typically, a pull-up resistor is connected to the input, and a ground-shunt jumper is inserted to notify the BIOS to force a re-flash update of the BIOS contents even though the BIOS checksum may be valid. Note: Because the BIOS examines this input only after a reset, it is also used an external SMI input. There is a simple way to use both functions together. The ground shunt is still placed directly at the input pin, but instead of using a pull-up resistor, a series resistor is connected between the input pin and the external logic that asserts EXTSMI1#. After reset, if the external logic keeps EXTSMI1# negated (high) until after the BIOS has examined the FRC_UPD# input, then the external logic and series resistor together act like a pull-up resistor. Alternate Function: EXTSMI1#
WE_AP#	ttl in	Application Flash Write Enable: This input, in conjunction with bit 8 of the BIOS Control Register, enables writes to Flash memory in the region not occupied by the BIOS. At the top of the Intel386 EX address space, this affects addresses 3F00000h through 3FDFFFFh if the Flash size is selected to be 1MB or 3000000h through 3FDFFFFh if the Flash size is selected to be 16MB Below the 1MB address boundary, this affects the memory region at addresses C0000h through DFFFFh. Alternate Function: DIO5

Table 2.7: User Chip Selects and Flash BIOS Interface

Signal	Type	Description
WE_BIOS#	ttl in	BIOS Write Enable: This input, in conjunction with bit 9 of the BIOS Control Register, enables writes to the BIOS region of the Flash device. The BIOS region is the top 128K of the Flash memory, both at the top of the Intel386 EX address space and just below the 1MB address boundary. In other words, addresses 3FE0000h through 3FFFFFFh and 00E0000h through 00FFFFFh. Note: Don't confuse this signal with similarly named signal WE_FLASH#. Alternate Function: DIO6
WE_FLASH#	out csr	Flash Write Enable: This output is the write enable for the system BIOS and application Flash memory. It typically is connected to the WE# input of the BIOS chip. It is gated by WE_AP#, and WE_BIOS#, and their corresponding register bits. Note: Don't confuse this signal with similarly named signal WE_BIOS#.

Table 2.8: Keyboard Interface Signals

Signal	Type	Description
CS_KB#	out 2 mA open drain	Keyboard Controller Chip Select: This pin is both the BIOS manufacturing loop input and the chip select output for an external keyboard controller. Regardless of whether the internal keyboard controller megacell is enabled or not, this output is asserted for ISA I/O accesses to addresses 60h or 64h. Because this output is open drain, there is no need for a mode bit to select between the MFG_LOOP# input and the CS_KB# output functions. The state of this pin may be read from bit 14 of SMI Control/Status Register. When using an external keyboard controller, connect a pull-up resistor to this output and connect this output to the chip select input of the external controller. Alternate Function: MFG_LOOP#
KB_CLK	cmos io csr open drain schmitt	Keyboard Clock: This bi-directional open drain signal implements the IBM PS/2 protocol for the keyboard clock. Alternate Function: DO18
KB_DATA	cmos io csr open drain schmitt	Keyboard Data: This bi-directional open drain signal implements the IBM PS/2 protocol for keyboard data. Alternate Function: DO17
KB_IRQ	out csr	Keyboard Interrupt: This output is the interrupt from the internal keyboard controller. It typically is connected to the INTO input of the Intel386 EX, which corresponds to ISA IRQ1. Alternate Function: DO16

Table 2.8: Keyboard Interface Signals

Signal	Type	Description
MFG_LOOP#	cmos in	Manufacturing loop: This pin is both the BIOS manufacturing loop input and the chip select output for an external keyboard controller. This input may be read from bit 14 of the SMI Control/Status Register or from the port P1.5 of the internal keyboard controller. Typically, a pull-up resistor is connected to the input, and a ground shunt jumper is inserted to notify the BIOS to perform a manufacturing test loop. Note: When using an external keyboard controller, a ground shunt would interfere with the chip select output function of this pin. In this case, a pull-up resistor and ground shunt may instead be connected to P1.5 of the external keyboard controller. The value of P1.5 may then be read by issuing appropriate commands to the external controller. Alternate Function: CS_KB#
MO_CLK	cmos io csr open drain schmitt	Mouse Clock: This bi-directional open drain signal implements the IBM PS/2 protocol for the mouse clock. . Alternate Function: DO19
MO_DATA	cmos io csr open drain schmitt	Mouse Data: This bi-directional open drain signal implements the IBM PS/2 protocol for mouse data. Alternate Function: DO20
MO_IRQ	out csr	Mouse Interrupt: This output is the interrupt from the internal mouse controller. This output should normally be hooked to the Intel386EX's IRQ12/13 pin which is available as an IRQ12 input on Intel386 EX B-step and later devices. Alternate Function: DO21

Table 2.9: IDE and Miscellaneous Signals

Signal	Type	Description
ACTIVITY#	ttl in	Activity: When asserted, this input indicates that system activity is occurring. This reloads the R380EX SMI timer with its starting value. Alternate Function: DIO7
EXTSMI[1:0]#	ttl in	External System Management Interrupt: These two inputs allow external devices to OR their interrupt requests into the SMI# output from the R380EX to the Intel386 EX input. They may be masked by bits 0 and 1 of the SMI Control/Status Register. Alternate Functions: EXTSMI1# is multiplexed with FRC_UPD#. EXTSMI0# is multiplexed with WDTIN.
FLOAT#	cmos in pull-up	Float All Outputs: When asserted, this input places the R380EX into a test mode. All output and I/O pins are tristated. The internal NAND chain is enabled, connecting all package pins except for power and ground pins. This allows parametric testing of all signals at chip fabrication time, and also allows ATE testing of the R380EX in a system. The output of the chain is NAND_OUT.
IDE_D7	ttl io csr	IDE Data bit 7: This signal connects to IDE interface data bit 7. ISA I/O address 3F7h is shared between an IDE device and a floppy disk controller. The R380EX resolves this conflict for bit D7 by bi-directionally buffering D7 between the SD bus and the IDE interface for all IDE I/O addresses, except for the shared address, 3F7h. For this case, D7 is bi-directionally buffered, through the R380EX , to SD7 to support a floppy disk controller. If the IDE interface is enabled, there is no way to disable this rebuffering. Therefore, IDE data bit 7 must always connect through the R380EX, even if the system design is such that all the other IDE data bits are connected directly to the SD bus without an intervening data transceiver. Alternate Function: DIO10
IDE_ENH#	out csr	IDE Enable High: This output enables an optional high-byte data transceiver in the IDE interface. This transceiver connects SD[15:8] to IDE data bus [15:8]. This output is asserted only for 16-bit reads to ISA I/O address 1F0h. Alternate Function: DIO11
IDE_ENL#	out csr	IDE Enable Low: This output enables an optional low-byte data transceiver in the IDE interface. This transceiver connects SD[6:0] to IDE data bus [6:0]. The transceiver must not connect to SD7. The R380EX rebuffers that bit directly, using signal IDE_D7. Alternate Function: DIO12
IDE_RD#	out csr	IDE I/O Read Strobe: When not using IDE PIO mode 3, this output is identical to the IOR# output. In IDE PIO mode 3, this output switches much more quickly than it would for a standard ISA I/O read cycle. This allows much faster transfer rates between the IDE device than would otherwise be possible. Alternate Function: DIO9

Table 2.9: IDE and Miscellaneous Signals

Signal	Type	Description
IDE_WR#	out csr	IDE I/O Write Strobe: When not using IDE PIO mode 3, this output is identical to the IOW# output. In IDE PIO mode 3, this output switches much more quickly than it would for a standard ISA I/O write cycle. This allows much faster transfer rates between the IDE device than would otherwise be possible. Alternate Function: DIO8
NAND_OUT	out 2 mA	NAND Chain Out: If FLOAT# is asserted, this is the output of a NAND chain that connects all other package pins, except for power pins, ground pins, and the Real Time Clock pins that are powered by RTC_VCC. This allows parametric testing of most signals at chip fabrication time, and also allows ATE testing of the R380EX in a system. The exact ordering of the NAND gates is to be specified. If FLOAT# is negated, this output is OSCOFF#. Alternate Function: OSCOFF#
OSCOFF#	out 2 mA	Disable External CLK2 Oscillator: When FLOAT# is negated, this output indicates that the R380EX has entered a low-power standby mode, and that the external CLK2 oscillator may be switched off to conserve power. When FLOAT# is asserted, this output is NAND_OUT. Alternate Function: NAND_OUT
SPKR_IN	ttl in	Speaker In: This input feeds into an internal XOR gate that generates SPKR_OUT. It may be used by external devices, such as PCMCIA modem cards, that want to combine their audio output with the audio output from the Intel386 EX 8254 timer 2. Alternate Function: DIO13
SPKR_OUT	out 8mA csr	Speaker Out: This output is (bit 1 of the R380EX PortB Register ANDed with the TMROUT2 input from the Intel386 EX 8254 timer 2) XORed with the SPKR_IN input. The AND is an enable function defined by the AT architecture, while the XOR allows another device to combine its audio with Intel386 EX 8254 timer 2. Alternate Function: DIO14
TMRGATE2	out 2 mA	Timer Gate 2: This output is bit 0 of the R380EX PortB Register. Connect this signal directly to the corresponding Intel386 EX input. This provides an easy way for an application to control Intel386 EX 8254 timer 2, and is compatible with the AT architecture.
TMROUT2	ttl in	Timer Out 2: Connect this input directly to the corresponding Intel386 EX output. This input is readable in the R380EX PortB Register, and also combines with internal logic before being output on SPKR_OUT. Alternate Function: DIO15

Table 2.10: Digital I/O, Digital Outputs, LED Outputs

Signal	Type	Description
DIO[15:0]	DIO9 and DIO8 are cmos io All others are ttl io csr	Digital Inputs/Outputs: The R380EX has up to 16 digital I/O pins. In order to save package pins, they are each multiplexed with an alternate function. Each pin may individually be configured for digital I/O or for the alternate function through the DIO_CFG Register. This configuration register is initialized, in three groups of bits, by the state of MA[7-5] at power-up. Each I/O signal is individually programmable as either an input or an output through the DIO_DIR Register. This register is cleared at power-up, so all I/O signals start out as inputs. Most pins may be dynamically switched between digital I/O and their alternate functions. There is one exception. If DIO[6:5] are NOT configured at power-up to be I/O, then they can not be dynamically switched to I/O later. This is because their alternate function is to allow external jumpers to selectively protect various regions of the system BIOS and Flash memory from alteration. If these pins were allowed to become I/O, an errant application program could bypass the protection mechanism. Alternate Functions (selected by MA7): DIO15 is multiplexed with TMROUT2. DIO14 is multiplexed with SPKR_OUT. DIO13 is multiplexed with SPKR_IN. DIO12 is multiplexed with IDE_ENL#. DIO11 is multiplexed with IDE_ENH#. DIO10 is multiplexed with IDE_D7. DIO9 is multiplexed with IDE_RD#. DIO8 is multiplexed with IDE_WR#. Alternate Functions (selected by MA6): DIO7 is multiplexed with ACTIVITY#. DIO6 is multiplexed with WE_BIOS#. DIO5 is multiplexed with WE_AP#. DIO4 is multiplexed with QDACK0#. Alternate Functions (selected by MA5): DIO3 is multiplexed with CS_USR3#. DIO2 is multiplexed with CS_USR2#. DIO1 is multiplexed with CS_USR1#. DIO0 is multiplexed with CS_USR0#.
DO[21:16]	out csr	Digital Outputs: The R380EX has up to six digital output pins. In order to save package pins, they are all multiplexed with alternate functions. The pins may be selected in two groups for use as digital output or for use in their alternate functions. If the internal Keyboard Controller megacell is enabled, then all pins initialize to keyboard or mouse functions. If the megacell is disabled, all pins initialize to digital outputs and their initial output level may be set, as a group, by the state of MA[8] at power-up. Alternate Functions (selected by MA8): DO21 is multiplexed with MO_IRQ. DO20 is multiplexed with MO_DATA. DO19 is multiplexed with MO_CLK. DO18 is multiplexed with KB_CLK. DO17 is multiplexed with KB_DATA. DO16 is multiplexed with KB_IRQ.

Table 2.10: Digital I/O, Digital Outputs, LED Outputs

Signal	Type	Description
LED[1:0]#	out csr	LED Outputs: When the internal Real Time Clock megacell is enabled, these outputs are controlled by the LED Register. LED0# is initialized by a hardware reset to buffer and output the RTC_X2 input (normally 32.768KHz). The LED Control Register determines the exact LED drive/control functions performed by these pins. Alternate Function: LED0# is multiplexed with RTC_WR#. LED1# is multiplexed with RTC_RD#.

III. Intel386 EX Local Bus Interface

The R380EX connects directly to the Intel386 EX CPU local bus. There are three types of bus cycles from the Intel386 EX, and the R380EX supports them all:

1. LBA cycles
2. LDEV cycles
3. Bus cycles handled by the R380EX:
   - DRAM references
   - Intel386 EX initiated Refresh
   - Internal registers
   - ISA/Xbus cycles
   - Halt/shutdown
   - Interrupt Acknowledge
   - DMA cycle

LBA CYCLES

The Intel386 EX completely defines and controls LBA cycles. They are denoted by the assertion of LBA# by the Intel386 EX. LBA# cycles are accesses to either the internal Intel386 EX peripherals, or devices selected by the Intel386 EX chip-select unit. LBA# indicates that the Intel386 EX will assert READY# itself to terminate the cycle.

The R380EX ignores cycles with LBA# asserted and waits for the next ADS. The timing of LBA# from the Intel386 EX is such that the R380EX must wait to sample it until the end of the first T2 state of the cycle. Because the R380EX starts DRAM cycles at the end of the first T1 state which is much earlier, LBA must **not** be asserted for memory addresses that would otherwise access the DRAM. ISA cycles (including accesses to the R380EX internal peripherals and registers) are not started until the second T2 state of a cycle. LBA# will inhibit those ISA cycles.

If the address is in the range defined by the BIOS Control Register, the R380EX will assert CE_BIOS#. This allows CE_BIOS# to be used during initial power-up, before the BIOS has been able to configure the R380EX. However, the R380EX will never assert WE_FLASH# during an LBA# cycle.

LDEV CYCLES

LDEV# cycles are controlled primarily by another local bus device. These cycles are denoted by the assertion of LDEV# by another local bus device (such as a video controller). In this case, the local bus device must meet the Intel386 EX READY timing and drive READY directly with a 3-state driver.

LDEV# cycles can also be any type of cycle (read, write, I/O, or memory). The R380EX only monitors these cycles but does not act on them.

All non-LBA# or non-LDEV# cycles are handled by the R380EX. These include DRAM references, DRAM refresh, R380EX internal register references, ISA/Xbus cycles (both memory and IO), halt/shutdown, interrupt acknowledge, and DMA cycles. Each of these is described in more detail below.

ADDRESS PIPELINING

The R380EX supports address pipelining for DRAM cycles by asserting NA# shortly before each DRAM access completes. Local bus devices may request address pipelining by asserting NA# also. However, the Intel386 EX imposes very stringent setup and hold requirements on this signal. It will be difficult for the system designer to use an external OR gate to combine these two sources of NA#. For this reason it is unlikely that local bus devices will be able to use NA#, except at low clock frequencies.

ALTERNATE LOCAL BUS MASTERS

The R380EX also supports other local bus masters. These devices need to acquire the bus using HOLD/HLDA protocol and then emulate the timing and protocol of Intel386 EX bus cycles. The R380EX cannot tell the difference between a Intel386 EX cycle and another local bus master cycle that properly mimics an Intel386 EX bus cycle.

DRAM REFERENCES

Accesses to memory addresses that span the DRAM address range are handled by the R380EX. DRAM should be autosized by the BIOS on boot-up. DRAM access cycles can be either 8- or 16-bit. The cycle is terminated with the assertion of READY by the R380EX. Timing parameters are dependent on the values in the DRAM Control Registers.

INTERNAL REGISTERS

Internal register references are handled by the R380EX. R380EX registers should be accessed using 16-bit I/O read or write cycles. R380EX internal register access cycles are echoed on the ISA bus. The cycle is terminated with the R380EX asserting READY. Accesses to R380EX internal registers use 16-bit ISA I/O timing. 8-bit references to these 16-bit registers will terminate correctly, but the data is undefined for the other byte.

ISA CYCLES

Any cycle that does not access DRAM or R380EX internal registers, and is not a local bus access, or a halt/shutdown cycle, or refresh cycle is routed to the ISA bus.

HALT OR SHUTDOWN

The R380EX can be configured to respond to an Intel386 EX Halt cycle in the following ways:

1. Return READY.
2. Return READY and switch the 2xCPU clock from CLK2OSC source to the 32KHz (RTC_CLK) clock. A subsequent ADS# assertion from the Intel386 EX initiates the switch back to the high-speed CLK2OSC source after a programmable delay is satisfied. While the 32KHz clock is being sourced to CLK2, the DRAMs will be placed into a self-refresh mode. Thus to use this mode, self-refresh capable DRAMs must be used in the system design.
3. Return READY and generate an SMI event.
4. Return READY, generate and SMI event and switch to the 32KHz clock as described in (2) above.

Mode 2 above allows the system designer to bring the whole system power consumption down to a few milliamps, or even less for properly designed systems, but requires the use of self-refresh capable DRAMs.

To achieve the lowest power-consumption, the CPU clock oscillator must be powered down by either the POWERDOWN signal from the Intel386 EX or the OSC_OFF# signal from the R380EX. The POWERDOWN signal from the Intel386 EX is asserted when a HALT is executed and Power Control Register bits PC1:0 of the Intel386 EX are set to the 01b state. By using the OSC_OFF# signal to disable the oscillator, the Intel386 EX does not need to enter its POWERDOWN state, a state in which its CLK2 input is disabled and bus phase information is lost.

If the Intel386 EX does enter the POWERDOWN state, a PHASE1 signal is also provided by the R380EX to allow local bus peripherals to track the bus phase correctly when the halt state is exited. If the POWERDOWN state of the Intel386 EX is not used, the R380EX still supports a very low power HALT state just by switching the 32KHz clock into the Intel386 EX's CLK2 input. This keeps the Intel386 EX's power consumption to less than 1mA, even without the Intel386 EX going into the POWERDOWN state.

In the case that the Intel386 EX is placed into the POWERDOWN mode, the R380EX automatically detects any phase change injected by the Intel386 EX when it exits the POWERDOWN state. The R380EX does this detection by monitoring the assertion of ADS#. ADS# is guaranteed, by the Intel386 EX, to be asserted during PH1 while it is operated at a 32KHz frequency. The PHASE1 output from the R380EX will also be corrected glitch-lessly. Thus the Intel386 EX's POWERDOWN mode may also be used as long as the other local bus devices monitor the R380EX's PHASE1 signal and can deal with the phase change as well.

The Intel386 EX exits the HALT and/or POWERDOWN state in response to an interrupt. This is indicated by the assertion of ADS# after a period of ADS# inactivity. On the first access the CPU makes after exiting the HALT state (ADS# is detected again), a programmable delay of up to 64 milliseconds is used. During this delay, the CPU CLK2 line (CLK2OUT pins) is held high before switching back to the normal CPU clock oscillator (CLK2OSC pin). This delay allows the CPU clock oscillator to restart and get back up to speed after the POWERDOWN signal has been removed. The delay can be used, for example, so that subsequent DRAM cycles don't violate RAS/CAS pulse widths.

When returning from this low power state in a properly designed system, the only state information that needs to be updated should be the time of day clock which is normally updated as a result of an 8254 timer tick interrupt. Correction of the time of day is facilitated through the use of a HLTS status bit in the R380EX Clock Control Register. This bit is set when a HALT instruction is detected by the R380EX. The time of day timer-tick interrupt may check this bit to see if the time of day needs to be reloaded from the RTC.

For further details please see the R380EX Clock/Reset Control Register and the SMI Control Register definitions.

In response to a Shutdown bus cycle, the R380EX simply initiates a hardware reset. In this case, both CPURESET and RESETDRV pins are driven asserted for a minimum of 32 CLK2 periods.

INTEL386-EX-INITIATED REFRESH CYCLES

Refresh cycles that are initiated by the Intel386 EX are basically just terminated by the R380EX. The DRAM controller has its own refresh counter and it is used for both the DRAM refresh and to cause ISA refresh cycles.

INTERRUPT ACKNOWLEDGE

Interrupt acknowledge cycles are unique cycles that are used to respond to an 8259A programmable interrupt controller. There are two cycles emitted from the Intel386 EX. The first is an LBA cycle which

acknowledges the master 8259A internal to the Intel386 EX. The second is also an LBA cycle if the interrupt source is from the internal 8259A. If not, it is treated as a normal IO cycle. Cascading of external interrupt controllers is not supported.

DMA CYCLE

In order to use the DMA channels in the Intel386 EX for external DMA-controlled devices, logic is included in the R380EX to coordinate the ISA bus and DRAM controller state machines.

The R380EX has two DACK input pins which should driven by the Intel386 EX and allow the R380EX to see that the Intel386 EX is doing a DMA cycle. They indicate that the Intel386 EX is driving a memory address and that the R380EX needs to either mate a slow DRAM read with a simultaneous ISA write, or a slow DRAM write from a simultaneous ISA read.

When a DMA bus cycle occurs and the Intel386 EX is indicating a memory read, the R380EX generates an I/O write on the ISA bus and gates the local data bus (onto which the DRAM data will be enabled) onto the ISA data bus. The cycle terminates as any regular ISA cycle would.

When such a bus cycle occurs and the Intel386 EX is indicating a memory write, the R380EX generates an I/O read on the ISA bus and gates the ISA data bus to the local bus. The cycle terminates as any regular ISA cycle would.

The above is true if the memory address points to the area of DRAM controlled by the R380EX. The Data is gated onto the ISA bus if the memory address does not match the area of DRAM controlled by the R380EX. In this case, the appropriate ISA memory strobe, MEMR# or MEMW#, is asserted in conjunction with the IO strobe and the DMA transfer is completed entirely on the ISA bus.

IV. DRAM/Flash Memory Controller

The R380EX DRAM/Flash controller can handle two independent 32-bit wide memory banks of DRAM/Flash SIMMs, bank A and bank B. They may have one or two 16-bit wide DRAM sub-banks per bank, called the First and the Second sub-banks. For each sub-bank, the upper and lower data bytes are controlled by the two CASx# outputs

If a bank has only a single 16-bit wide DRAM sub-bank, the R380EX controls it with the RASxF# output. If a bank has two 16-bit wide DRAM sub-banks, the R380EX controls them using both the RASxF# and RASxS# outputs. Only one RASxx# is asserted at a time. The R380EX supports both Fast Page Mode (FPM) and Extended Data Out (EDO) DRAM.

The R380EX multiplexes either the RASBS# function or the MA11 function onto a single pin. The pin function is selected at power-up by detecting the state of MA3. This means that when the MA11 function is selected during power-up configuration, the R380EX only supports a total of three memory sub-banks - sub-bank A-first, sub-bank A-second, and sub-bank B-first. Sub-bank B of the second DRAM bank is not available because the pin for RASBS# is configured for use as MA11. This restriction is enforced in hardware, since violation can cause improper DRAM operation and/or data bus contention.

When configured for RASBS#, 64MB and 32MB banks cannot be used because MA11 is not available. Note that setting the bank size to 16MB or 32MB when MA11 is disabled will be interpreted by the R380EX as meaning that the bank is empty.

In either case, each bank can support up to a 32-bit wide Fast Page Mode (FPM) DRAM, Extended Data Out (EDO) DRAM, or a Flash SIMM, each of which is configured as two x16 sub-banks. Each bank can be any size from 512KB to 16MB (64MB if configured for MA11). Within a bank, if both 16-bit sub-banks are populated, the DRAM size and DRAM type (EDO or FPM) of both sub-banks must be the same. However, bank A and bank B memories may differ from each other in both size and type of DRAM.

Bank A's first memory sub-bank always begins at address 0. Within a bank, the second memory sub-bank always contiguously follows the first. Bank B can contiguously follow Bank A, or it can start at the 32MB boundary (0x2000000). This boundary could be useful if bank B holds Intel DRAM style flash. However, if bank A is 64MB in size, then bank B is not allowed to start at 32MB.

The R380EX allows each bank to be put into a mode that supports software detection of the type of DRAM (EDO or FPM) installed in each bank, without requiring presence detect or other external hardware means. The only extra system hardware required to support this capability is a single pull-up resistor on DRAM data bit 0. To conserve power in the HALT and IDLE modes, the R380EX has "bus keepers" on all of the other data bus pins (D1-D15) that would prevent the detection of EDO versus FPM DRAM.

A second, externally controlled, DRAM port can be implemented with R380EX. DRAMHOLD# is an input that is used to request use of the DRAM interface. An external controller assumes a grant when DRAMHOLD# is asserted and all RAS and CAS lines are unasserted. The external controller is responsible for generating the required timing for the memory that it is accessing. Both the R380EX and the secondary external DRAM controller must monitor LOCK# from the Intel386 EX and not take the bus from the other device if LOCK was asserted during the previous DRAM cycle.

USING TWO SIMM SOCKETS WITH THE R380EX

The R380EX provides two CASxx# and four RASx# signals. This combination can support one or two "single-sided" SIMM modules. In addition, it can also support one "double-sided" SIMM module. Each SIMM module can have one or two banks, and each bank can have one or two 16-bit sub-banks, for a total of up to four sub-banks in one "double-sided" SIMM module. If a user wants to have the option of using one or two SIMMs in two SIMM sockets, care must be take not to exceed the two "Single-Sided" or one "Double-Sided" SIMM requirement.

CONNECTING TWO SIMM SOCKETS

The table below shows the connection scheme for using two SIMM modules with the R380EX. The connections between SIMM sockets are identical except for the connection of the RASx# signals. Table 4.1 highlights the connection of the RASxx# signals from the R380EX to the two SIMM sockets.

Table 4.1: RASxx-to-SIMM Signals

R380EX Signal	SIMM "A" Signal	SIMM "B" Signal
RASAF#	RAS0#	RAS1#
RASAS#	RAS2#	RAS3#
RASBF#	RAS1#	RAS0#
RASBS# *	RAS3#	RAS2#

Note: MA11 is needed for 16M-deep devices. The alternate function of MA11 is RASBS#. These functions are mutually exclusive and result in some "holes" in the SIMM matrix below. These "holes"are noted by NA in the table below.

DRAM CONFIGURATIONS USING ONE OR TWO "SINGLE-SIDED" SIMMS

Table 4.2 shows the valid SIMM size combinations which can be created from 256K, 1M, and 16M density devices in x16- and x32-bit widths when using one or two "single-sided" SIMMs.

Table 4.2: SIMM Matrix

			SIMM "A"								
		RASAx#	-	F	F, S	F	F, S	F	F, S	*F*	*F, S*
	RASBx#	Size	None	256K x 16	256K x 32	1M x 16	1M x 32	4M x 16	4M x 32	*16M x 16*	*16M x 32*
SIMM "B"	-	None	0K	512K	1M	2M	4M	8M	16M	*32M*	*64M*
	F	256K x 16	512K	1M	1.5M	2.5M	4.5M	8.5M	16.5M	*32.5M*	*NA*
	F, S	256K x 32	1M	1.5M	2M	3M	5M	9M	17M	*NA**	*NA*
	F	1M x 16	2M	2.5M	3M	4M	6M	10M	18M	*34M*	*NA*
	F, S	1M x 32	4M	4.5M	5M	6M	8M	12M	20M	*NA**	*NA*
	F	4M x 16	8M	8.5M	9M	10M	12M	16M	24M	*40M*	*NA*
	F, S	4M x 32	16M	16.5M	17M	18M	20M	24M	32M	*NA**	*NA*
	F	*16M x 16*	*32M*	*32.5M*	*33M*	*34M*	*36M*	*40M*	*48M*	*64M*	*NA*
	F, S	*16M x 32*	*NA**	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*

Note: *Italics* indicate configurations which require MA11

DRAM Configurations Using One "Double-Sided" SIMM

Table 4.3 shows the valid SIMM configurations which can be created using "Double Sided" SIMMs. Only one SIMM socket may be populated at a time when using "Double Sided" SIMMs. This table assumes that the 512K x 32 SIMM is constructed as two banks of 256K x 32, 2M x 32 SIMM is constructed as two banks of 1M x32, and 8M x 32 is constructed as two banks of 4M x32. Use of "double sided" SIMMs reqires all four RASxx# signals.

Table 4.3: Double-sided SIMM

		SIMM "A"			
	Size	None	512K x 32	2M x 32	8M x 32
SIMM "B"	None	0K	2M	8M	32M
	512K x 32	2M	NA	NA	NA
	2M x 32	8M	NA	NA	NA
	8M x 32	32M	NA	NA	NA

V. Real Time Clock

The R380EX contains an integrated real-time clock, providing the PC function of the date/time clock, alarm, programmable periodic interrupt, 114 bytes of battery backed CMOS RAM, I/O registers 070h and 071h, and the crystal and battery input.

The RTC has isolated power pins for the battery source (RTCVCC and RCTGND), a reset signal for the CMOS RAM (RTCRES#), two crystal pins (X1, and X2) for a 32.768KHz crystal and an interrupt output (RTCIRQ). The interrupt should be connected to IRQ8 (INT4) on the Intel386 EX.

The RTC is integrated into the R380EX because it is part of most PC-compatible embedded designs.

However, since some designs have a need to use an external RTC, the ability to completely disable the RTC. This is accomplished by pulling MA1 high through a resistor during power-on reset (when PWRGOOD is deasserted). In this case, the R380EX contains logic to generate control signals RTCAS, RTCRD#, and RTCWR# for an external RTC chip. RTCAS is asserted for an I/O write access to address 070h. RTCRD# is asserted for an I/O read access of address 071h. RTCWR# is asserted for an I/O write access to address 071h. RTCRD# has the same timing as IOR#. RTCAS and RTCWR# have the same timing as IOW#.

VI. Power Management

The R380EX contains some basic support for System Management Mode (SMM) and for power management of the system environment. The R380EX allows the system designer to pursue several different power conservation strategies individually or in combination:

1. Application or OS level software can issue a Halt instruction to place the system into the low power mode described in the Halt/shutdown section.
2. Software can write to bits 1 and 0 of the Clock/Reset Control Register to reduce the Intel386 EX's system's clock speed.
3. An SMI (System Management Interrupt) can be generated in response to an external event or the R380EX interval timer expiring. The SMI routine can then handle powering down system peripherals or entering the hardware supported Halt state, as in step (1) above, or reducing the system clock speed as in step (2).

The R380EX contains all of the Halt detection logic, SMI generation logic, and glitchless clock switching logic to perform these functions. The SMI logic and clock switching logic are described in the following paragraphs.

VII. SMI and Clock Switching

Some power management software requires support hardware to determine when a system has become inactive. To make response to this condition be as flexible as possible, an SMI interrupt is normally asserted to allow an SMI service routine to decide how to power-down various parts of the system. The R380EX supports this style of power management with several input pins and a single SMI idle latch and timer. If the idle timer is allowed to count down to zero, and if the SMI function is enabled, then the SMI# pin is asserted, thus alerting the Intel386 EX to this condition.

The idle timer counts down at one of two programmable frequencies. If the idle timer reaches zero, and if SMI# generation is enabled (by bit 2 of the SMI Control/Status Register), then an SMI# is generated and the idle timer is stopped.

Independent of the current value of the idle timer, when any of the following events occur, the idle timer is reset to the idle latch value and resumes counting down:

1. SMIACK# is asserted by the Intel386 EX, (in response to the idle timer reaching 0 and asserting SMI#)
2. The external ACTIVITY pin is asserted for at least 2 CLK2 cycles, (if it is enabled by bit 7 of the DIO Configuration Register being 0)
3. A programmable chip select is asserted (if it is enabled by bit 6 of SMI Control/Status Register being 1)
4. A value is written into the SMI timer control register (i.e. the idle timer's latch)

Both timer reloads and count-down operations are suspended during a read of the SMI Timer Control Register so that a consistent and correct timer value is always returned by a read of this register. If a reload or countdown occurs during the register read, it is not lost. The reload or countdown is injected after the read has finished. In the case that both a reload and a countdown event occur during the read, only the reload occurs after the read.

The idle timer is 12 bits wide and may be clocked by either a 1024 hertz or a 16 hertz clock. Both of these frequencies are derived from the RTC_X2 (normally 32,768 hertz) input clock. When clocked at 1024 hertz, the idle timer can be set to expire from about 1 millisecond up to a total of about 4 seconds with about 1 millisecond granularity. When clocked at 16 hertz, an idle timer expiration time from about 62.5 milliseconds up to about 256 seconds can be achieved with about a 62.5 millisecond granularity.

The R380EX contains an SMI interrupt output pin. The following events cause assertion of this interrupt:

1. The idle timer decrementing to zero (if enabled to assert SMI)
2. The assertion of either the EXTSMI0 or EXTSMI1 input pins (if enabled to assert SMI)
3. The detection of a HALT cycle (when enabled to assert SMI)
4. The setting of a software-triggerable SMI Register bit

The EXTSMI[0:1] pins have individual enable register bits, and are useful for things such as a low-battery warning in hand-held applications for example, and for PCMCIA system-management events. Status bits also exist to show what source(s) initiated the SMI interrupt. An overall SMI enable bit (bit 5 of the SMI Control/Status Register), when clear, disables the assertion of the SMI# pin. However, the individual SMI status bits will latch any individually enabled SMI events that occur while the overall SMI enable bit is clear. This allows these bits to be polled by software instead of causing a System Management Interrupt.

The R380EX has an SMIACT input pin, which is driven by the corresponding Intel386 EX pin to indicate that the Intel386 EX is in SMM. When in SMM, the Intel386 EX generates bus cycles for addresses in the range 038000h - 03FFFFh. When enabled by the SMIMAP[1:0] bits in the SMI Status/Control Register, the R380EX remaps addresses in the range 02xxxx-5xxxx as indicated in the following table.

SMIMAP[1:0]	Address Mapping Function
00	Address mapping disabled in all modes
01	Address mapping disabled in all modes
10	When SMI is active, remap 2xxxx → Axxxx and 3xxxx → Bxxxx
11	2xxxx → Axxxx , 3xxxx→ Bxxxx and if SMI is active, also remap 4xxxx → 2xxxx and 5xxxx → 3xxxx

For “normal” operation, address's 2xxxx/3xxxx are mapped by the R380EX into the DRAM. However, when in SMM, these addresses map into the “Axxxx/Bxxxx” page, which is usually a section of “wasted” memory because video controllers, if present, often use this address range for video memory. The Intel386 EX, in SMM, uses the 3xxxx addresses to store system state and to fetch code from. Thus, some software (e.g., BIOS), at initialization, needs a way to copy code into the DRAM “Axxxx/Bxxxx” page. This is the purpose of the last row of the table. This last line also allows an SMM task to look at the 2xxxx/3xxxx region, if necessary.

Software may reduce the CPU/system “CLK2” clock frequency by writing bits 0 and 1 of the Clock/Reset Control Register . These bits allow the input oscillator clock to be divided by 1,4, or 16 to supply the CLK2OUT pin. Upon reset, the clock is automatically divided by four to allow the system components to easily track the bus phase as reset is released. The Intel386 EX and other local bus peripherals determine the bus phase by watching for the deassertion of reset. Since the R380EX is responsible for releasing the reset line, it also automatically changes the clock divider control bits to perform a divide by 1 within a few bus cycles after deassertion of RESETCPU. Software may also subsequently set the two divider control bits in the Clock/Reset Control Register to speed-up or slow-down the clock speed, depending upon whether performance or power conservation is required.

The SMI idle timer can also be used to perform any system function that needs to happen at guaranteed exact time intervals because the SMI interrupt is the highest priority interrupt in the system that cannot be disabled.

The idle timer can be used as a source of a periodic interrupt that can occur at 1ms intervals up to a maximum of 256 second intervals. This is done by setting the overall SMI enable and the SMI timer enable bits (bits 5 and 2 of the SMI Control/Status register), and by disabling the ACTIVITY pin, (setting bit 7 in the DIO configuration register) and disabling the USR_CS0# idle timer reset sources. (clearing bit 6 in the SMI Control/status register), and by clearing all other SMI enable bits in the SMI Control/Status register (bits 4,3,1, and 0). In this mode, the SMI Timer Control register is programmed to the desired interrupt time period.

For example, if the system needs to update a display at exactly 8 second intervals, the code to update the display would be placed in the SMI handler and the SMI timer control register would be set to 0x8080, the SMI Control/status register would be set to 0x0164, and bit 7 of the DIO Configuration register would be set.

With this configuration, every eight seconds the SMI# pin is asserted and the idle timer is automatically set back to the 0x8080 value (in response to the SMIACK# pin) and starts counting down to cause the next SMI# assertion, etc. This same technique could be used to "wake-up" a halted, power-downed system at guaranteed intervals to check for system activity.

VIII. ROM/Flash ROM Interface

The ROM/Flash ROM interface supports a variety of ROM/Flash devices. Most PC compatible implementations use an 8-bit BIOS device that is shadowed in DRAM. This type of BIOS interface as well as many others are supported by the R380EX.

The registers in the R380EX are specifically designed to support the Intel 28F004BR-T or similar Flash devices. The upper 128KB, which includes a 16KB boot block, is normally used for the BIOS. The lower 384KB can be used for user code such as a real time OS kernel. If user code is not needed, then a smaller device can be used (e.g., 28F001BX-T). The BIOS usually consists of at least a 16KB boot block, and a 112KB System BIOS. Once shadowed, the BIOS running in DRAM typically only occupies 64K in the F-page of the low 1MB of memory.

While ROMs do not need write protection, write protection of the Flash is an important system issue. The R380EX provides extensive write protection. There are two independent pins and associated register bits, WE_BIOS and WE_AP in the BIOS Control Register, that are part of this protection. The intent is to allow one to independently protect the system BIOS (BIOS) and the application program (AP).

To allow writes to either Flash region, the associated pin must be asserted and the appropriate register bit must be set. This allows manufacturers to update the boot block, while giving others the ability to update the system BIOS and the Application Programs in the field without the possibility of corrupting the other sections of the Flash device.

The register bits give software control over these features and the pins give the hardware designer the capability of a mechanical protection mechanism, such as a jumper or a pin strap. For instance, to ensure that no one can corrupt the BIOS, just tie the WE_BIOS# pin high.

Please note that the two write enable pins have DIO as alternate functions for the pins. If the DIO Configuration Register specifies that a particular pin is serving as a DIO pin, then the write enable function is enabled or disabled only by the enable bits in the BIOS Control Register. The FRC_UPD pin can be used to force the BIOS to perform a system BIOS update regardless of the state of the code or the checksum in the BIOS. This is a mechanism that is used by certain BIOS vendors (e.g., Phoenix) that allows the update program to determine whether to process an update. This pin is a read-only bit in the SMI Control/Status Register.

IX. Keyboard/Mouse Controller

The R380EX contains a PC/AT compatible keyboard controller with PS/2 compatible mouse controller extensions. The keyboard controller is clocked by BCLK (also referred to as SYSCLK), at the ISA bus system clock frequency. Response to keyboard commands is nearly immediate, usually within one BCLK, due to the fact that the keyboard controller function is internally implemented as a hard-wired state machine.

If it is not needed,or an external keyboard is desired, the internal keyboard/mouse controller may be disabled. (see power-up options and Miscellaneous configuration register.) When disabled the keyboard and mouse pins can be used for other functions or an external keyboard can be supported.

The keyboard controller responds to two I/O addresses: 0x60 (data) and 0x64 (Command/status). Commands are issued to the keyboard controller by performing I/O writes to 0x64. An I/O write to 0x60 will fill the keyboard controller's "Input Buffer". An I/O read from 0x60 will return the contents of the keyboard controller's "Output Buffer". The keyboard controller contains two other registers called the "Output Port" and "Input" Port that are associated with "port" pins that connect to other signals inside the R380EX. (Don't confuse these with the similarly named Output Buffer and Input Buffer.)

The integrated keyboard controller will support the following commands. All commands except Fx and D4 will be executed in the same bus cycle that they are issued.

Command	Function
20	Read Command Byte
21-2F	Read RAM
60	Write Command Byte
61-6F	Write RAM
A7	Disable Mouse
A8	Enable Mouse
A9	Mouse Interface Test
AA	Self Test
AB	Keyboard Interface Test
AD	Disable Keyboard
AE	Enable Keyboard
C0	Read Input Port
D0	Read Output Port
D1	Write Output Port
D4	Transmit to Mouse
E0	Read Test Inputs
Fx	Pulse Output Port

READ COMMAND BYTE (20)
A Read Command Byte will place the Command Byte in the Output buffer. An I/O read to an address of 0x60 returns the output buffer contents.

READ RAM (21-2F)
The keyboard controller contains one extra byte of RAM beyond the Command byte. Execution of a 0x21 to 0x2F command will cause the contents of this RAM to be dumped into the Output buffer.

WRITE COMMAND BYTE (60)
The Write Command Byte will place the next data byte written to I/O address 0x60 into the Command Byte. Upon reset, the Command Byte is set to 70. The bits within the Command Byte are defined as follows:

Bit	Function
0	Enable Key Interrupt
1	Enable Mouse Interrupt
2	System Flag
3	Inhibit Override
4	Disable Keyboard
5	Disable Mouse
6	Enable Scan Code Translation
7	0

WRITE RAM (61-6F)
The internal keyboard controller contains one byte of RAM. A command of 0x21to 0x2F will cause this RAM location to be updated by the next I/O write to address 0x60.

DISABLE MOUSE (A7)
A Disable Mouse command will clear bit 5 of the Command Byte.

ENABLE MOUSE (A8)
A Enable Mouse command will set bit 5 of the Command Byte.

MOUSE INTERFACE TEST (A9)
A Test Mouse Interface command will place a 00 in the Output Buffer. An Interface test will not actually occur.

SELF TEST (AA)
A Self Test Command will place a 55 in the Output Buffer. A self test will not actually occur.

KEYBOARD INTERFACE TEST (AB)
A Test Keyboard Interface command will place a 00 in the Output Buffer. An Interface test will not actually occur.

DISABLE KEYBOARD (AD)
A Disable Keyboard command will set bit 4 of the Command Byte.

ENABLE KEYBOARD (AE)
A Enable Keyboard command will clear bit 4 of the Command Byte.

READ INPUT PORT (C0)
A Read Input Port command will place the contents of the Input Port in the Output Buffer. The Input Port is connected internally to the R380EX.

READ OUTPUT PORT (D0)
A Read Output Port command will place the contents of the Output Port in the Output Buffer. The output Port is connected internally to the R380EX.

WRITE OUTPUT PORT (D1)
The Write Output Port command will place the next data byte written to I/O 0x60 to the Output Port. The output Port is connected internally to the R380EX.

TRANSMIT TO MOUSE (D4)
This command will cause the next byte written to the data port to be sent to the mouse. The execution time of this of this command will depend upon the mouse.

READ TEST INPUTS (E0)
A Read Test Inputs command will place T0 and T1 in D0 and D1, respectively, of the Output Buffer. Bits D2 - D7 will be 0.

PULSE OUTPUT PORT (FX)
The Pulse Output Port Command will pulse bits 0-3 of the Output Port low for at least 6 usec. Bits 0-3 of this command determine which bits will be pulsed. A command of FF will execute in the same bus cycle it is issued.

GATEA20 AND RESET
Since the D1 command takes place in the same bus cycle it is issued, a change in GATEA20 will happen immediately. An Fx command will take from 6.4 to 128 us to generate a reset depending on the clock speed. An FF command will take place in the same bus cycle it is issued.

KEYBOARD STATUS REGISTER

Bit	Function
0	Output Buffer Full
1	Input Buffer Full
2	System Flag
3	Command/Data
4	Inhibit Switch
5	Mouse Data
6	Time-out
7	0

Upon reset the Keyboard Status Register is set to 00.

KEYBOARD AND MOUSE INTERFACE

The keyboard interface supports a standard AT keyboard. The mouse interface supports a standard PS/2 mouse. The internal port pins are configured as follows:

Signal	Function
P1(0)	Key data
P1(1)	Mouse data
P2(0)	CPURESET
P2(1)	GATEA20
P2(4)	IRQ1
P2(5)	IRQ12

Upon Reset all P2 port bits are set high.

TIMINGS

The keyboard controller clock is driven at the BCLK frequency. The clock speed will determine certain timings as shown below. Note that the FF command will execute in one bus cycle.

Timing	BCLK @ 8MHz
Fx Command execution (except FF)	5.1 us
Transmission backoff	80 us
Transmission Time-out	2.5 ms

X. PC Speaker Interface

The R380EX contains a speaker output (SPKROUT), which is the XOR of {the SPKRIN input pin} with {the logical ANDing of [bit 1 of the port B register] with [the OUT2 timer 2 input pin]}. The rationale for the extra XOR and the SPKRIN input pin is that this is a common “glue” function needed in PCMCIA systems for use with modem cards. SPKRIN shares the same pin as DIO13. If the DIO Configuration Register DEFINES this pin to be DIO13, then the SPKRIN function is disabled.

X. ISA Bus Interface

The ISA bus controller manages the ISA control signals and a separate data bus. It generates cycles on the ISA bus for memory and I/O cycles that are not mapped to devices on the local bus, or in the processor, or in the R380EX chip itself. The ISA controller supports 8- and 16-bit transfers, including translation for 16-bit CPU requests to 8-bit ISA devices, and 8-bit CPU requests to 16-bit devices.

The controller also produces the ISA (AT) bus clock. The ISA clock can be the CPU's CLK2 clock divided by 4, 6, or 8.

The R380EX does not expand the interrupt or DMA capabilities of the Intel386 EX. Thus the interrupt and DMA channels available in a system using the R380EX are exactly the same as the Intel386 EX.

The exception to this is that fly-by DMA cycles are supported by the R380EX ISA interface logic. Assertion of the DACK inputs to the R380EX indicate that the current cycle is a fly-by DMA cycle.

The R380EX implements a “quiet ISA” feature. When the R380EX sees an access that is not destined for the ISA bus (i.e., a local bus cycle, a non-DMA DRAM cycle, it does not drive the ISAbus data and control signals. This reduces overall system power consumption.

The R380EX ISA bus controller does not support multiple bus masters and I/O channel check, which are two almost-never-used aspects of the ISA bus.

XII. PC Engine Logic

The R380EX performs a number of functions in support of the Intel386 EX chip and in support of the PC hardware architecture.

The R380EX generates the clock signals for the Intel386 EX. The R380EX expects a 2x CLK input (CLK2OSC) and generates two, low skew, 2x clock outputs -- CLK2OUT[1:0]. It also generates a 1x clock phase indicator (PHASE1) that is in-phase with the Intel386 EX internal 1x clock.

One of the R380EX 2x clock outputs needs to drive the R380EX CLK2IN input and also the CLK2 input of the Intel386 EX. This clock is used for all timing reference for bus activity in the R380EX.

The R380EX also handles the variety of reset signals and conditions expected in a PC compatible system. It has a cold reset input (PWRGOOD) and a watchdog time-out input (WDTIN) which result in generation of resets for the Intel386 EX (RESET) and the ISA bus (RESETDRV).

The R380EX contains the PC-compatible “port B” register at 061h. There is no port A register at 092h because this function is already included in the Intel386 EX chip.

The R380EX has a TMRGATE2 output and a TMROUT2 input for interfacing to the corresponding Intel386 EX timer pins and are commonly used as part of the speaker output function.

XIII. 16-bit DIO Port

The R380EX has a 16-bit digital I/O port, and 6 bits of digital output. There are four registers associated with the DIO port: the DIO input register, the DIO output register, the DIO direction control register and the DIO Configuration Register. A DIO pin reflects the associated data output register bit if the corresponding bit in the direction register is set to OUTPUT, and the bit in the mux register specifies using the pin as DIO.

The DIO pins can be read as inputs regardless of the settings of the DIO Direction Register or the DIO Configuration Register. For each bit in the DIO port, the Configuration Register defines whether the pin has the DIO function or the alternate function. The DIO Configuration Register is initialized at power-up from the MA[7:5] pins.

Note that DIO bits 5 and 6 receive special treatment and must be set to DIO (by a pull-up on MA6) at power-up if they are to be used as digital I/O pins. If MA6 is pulled-down at power-up, these pins are set to the flash write enable functions and these functions cannot be overridden by writing a "1" to the respective DIO configuration bit.

The DIO output register is a read/write register. Writing data to this register will cause the pins to change state if they are setup as outputs. Reading the data output register returns the value in the register, and not the value of the DIO pins.

The data input register is an input only register, and reading this register returns the actual state of the DIO pins.

Thus, with all bits set as outputs, it is possible to write a value to the DIO data output register and read the actual state of the pins by reading this input register. If bits differs from the contents of the output register, then contention has been detected on the pins.

XIV. Programmable Chip Selects

There are four user-programmable I/O chip selects -- CS_USR[3:0]. Each can be programmed and used individually or they can be “daisy-chained” together to form one extended chip select output that can decode up to four discontiguous regions. They can be combined as a group of four, three, or two chip selects. These configurations include: one extended chip select with four discontiguous regions, one extended chip select with three discontiguous regions and one regular chip select, two extended chip selects each with two discontiguous regions, or one extended chip selects each with two discontiguous regions and two regular chip select.

Further these programmable chip selects can be used to reset the SMI activity timer, even if the external programmable chip select pins are being used as digital I/O.

Each chip select has a corresponding 16 bit mask and compare register (see PCS registers). The upper two bits of the register control whether the chip select is enabled, how it should be decoded, and whether IOCS16 should be asserted by the chip select.

The lower 10 bits are compared with ISA address bits SA[9:0] as specified by the MODE bits in the register. SA15 is always compared with bit 10 of the address portion of the register.

The remaining bits determine whether IOR, and/or IOW are included in the decode, and whether the chip select's output is daisy-chained into the next lower-numbered chip select. These chip selects are valid for I/O accesses only.

XV. IDE Interface

The IDE interface consists of IDEWR#, IDERD#, two buffer enables (IDE_ENHI#, IDE_ENLO#) and the IDED7 pin. IDEWR# and IDERD# connect directly to the IDE connector. An IDE interface also requires two chip selects to be generated: HDCS[1:0]#. These can be generated by using two of the four user programmable chip selects (CS_USR[3:0]#) on the R380EX. HDCS0# is the primary hard disk chip select and should be asserted for accesses to I/O addresses 1F0-1F7h. HDCS1# is the secondary hard disk chip select and should be asserted for accesses to I/O addresses 3F6h and 3F7h. IDE_ENHI# and IDE_ENLO# connect to the enable input of a '245-style bi-directional buffer for the 16-bit data path to the IDE connector.

The IDEWR# and IDERD# signals allow the R380EX to support the higher transfer rates of Enhanced IDE drives without requiring the ISA bus IOR# and IOW# signals to violate the ISA bus pulse widths.

The IDEMD bit, bit 4 of the Output Configuration Register, defines whether IDE interface operates at enhanced data rates or at the original IDE data transfer rates. If set, IDEMD causes the IDEWR# and IDERD# signaling to occur at EIDE mode 3/4 transfer rates.

IDEMD is cleared by a hardware reset to enable working with both older IDE drives and the newer EIDE drives that also reset themselves initially to the older IDE interface style.

IDED7 connects to the IDE connector data bit 7. This pin resolves conflicts with floppy disk interface access to I/O address 3F7.

ISA I/O address 3F7h is shared between an IDE device and a floppy disk controller. The R380EX resolves this conflict for bit D7 by bi-directionally buffering D7 between the SD bus and the IDE interface, except in one special case, an I/O read from address 3F7h. In this case, the R380EX does not pass IDE data bit 7 to the SD bus, but instead tristates SD7. This allows a floppy disk controller to drive SD7 as per ISA requirements.

If the IDE interface is enabled, there is no way to disable this rebuffering. Therefore, IDE data bit 7 must always connect through the R380EX, even if the system design is such that all the other IDE data bits are connected directly to the SD bus without an intervening data transceiver.

XVI. LED Control

The R380EX contains a control register and two pins, LED[1:0]#, for controlling two external indicator LEDs. This function can be useful in communications applications and hand-held applications (e.g., indicating system state or battery warning).

Alternatively these two pins can generate a clock with a frequency of 1 Hz, 128Hz, or 256Hz. The latter two frequencies can be obtained by placing the LED control logic into a test mode as detailed in the LED Control Register bit descriptions.

In addition, LED0 powers-up driving a buffered version of the RTC_X2 input clock, normally 32.768KHz. If the R380EX supplies the decode for an external RTC, then the LED pins must instead be configured to supply the RTC read and write strobes.

If used for external activity indicators, each of the two pins can drive an LED. Two register bits in the LED Control Register are associated with each pin and control the output.

00	LED off
01	LED on
10	LED flashes once (on for ~0.5 second) after bits written as 10
11	LED flashing (a 1Hz rate - 0.5 sec on, 0.5 sec off)

XVII. Power-up Configuration Options

The R380EX is highly configurable through use of internal registers, and power-up configuration sensing. While the PWRGOOD input signal is negated, the MA pins are used as configuration inputs that select various operating modes. By reading the Power-up Options register, software can also determine how the board was configured at power-up. Once operating, additional modes are selected through the use of DIO and DO configuration, direction, and data register bits.

None of the MA pins have internal pull-up or pull-down resistors. Therefore, external configuration resistors are required for all of the MA pins. Certain MA lines need not be pulled-up or down, if the system designer does not care how the configuration bit powers-up.

For example, if the Digital Outputs aren't being used then MA8 does not need an external resistor. When the R380EX is in low-power standby, it redrives all of the MA pins with the values latched at the assertion of PWRGOOD. This reduces standby power consumption by eliminating DC current through the resistors.

Table 17.1 enumerates the power-up configuration options.

Table 17.1: Power-up Configuration Options

Pin	State	Option
MA0		R380EX operating voltage.
	0	3.0 or 3.3 volt VCC operation. 'ttl in' pins have CMOS input thresholds.
	1	5.0 volt VCC operation. 'ttl in' pins have TTL input thresholds.
MA1		Internal Real Time Clock megacell. This bit is also written into bit 3 (EXRTC bit) of the Output register at power-up.
	0	Enabled.
	1	Disabled.
MA2		Internal keyboard and mouse controller megacell. This bit is also written into bits 2-0 of the Output configuration register at power-up.
	0	Enabled.
	1	Disabled.
MA3		Select between RASBS# and MA11 output.
	0	RAS_BS# output is enabled.
	1	MA11 output is enabled.
MA4		DRAM interface.
	0	DRAMHLDA# output is enabled.
	1	DATA_EN# output is enabled.
MA5		Digital I/O [3:0].
	0	CS_USR[3:0]# outputs are enabled.
	1	DIO[3:0] are enabled as inputs.
MA6		Digital I/O [7:4].
	0	ACTIVITY#, WE_BIOS#, WE_AP#, and QDACKB are enabled.
	1	DIO[7:4] are enabled as inputs.
MA7		Digital I/O [15:8].
	0	TMROUT2, SPKR_OUT, SPKR_IN, IDE_ENL#, IDE_ENH#, IDE_D7, IDE_CS1#, IDE_CS0# are enabled.
	1	DIO[15:8] are enabled as inputs.

Table 17.1: Power-up Configuration Options

Pin	State	Option
MA8		Initial level of Digital Outputs [21:16]. Applies only if the internal keyboard and mouse controller megacell is disabled.
	0	All outputs initialize to 0.
	1	All outputs initialize to 1.
MA9, MA10		User definable bits. These bits are sampled with reset and are available for software to read from the power-up option register. Thus, these two bits can be used to tell software about system-level (as opposed to R380EX level) configuration options.

XVIII. Register Descriptions

Note: Please refer to the RadiSys R380EX BIOS Adaptation Implementation Notes *for more detailed information and suggestions on register use.*

There are two types of internal registers in the R380EX:

- PC-Compatible Registers. These registers are defined by the PC architecture and are addressable at standard I/O addresses. This includes the RTC registers and the PortB register. The RTC registers are referenced with an index and data register at I/O addresses 070h and 071h respectively. The PortB register is addressed at I/O address 061h.
- R380EX-Specific Registers. These registers are referenced with an index and data register at I/O addresses 024h, and 026h respectively.

All PC-architecture defined registers (PortB, 0x60, 0x64, 0x70, 0x71, etc.) use 10-bit address decoding . All other registers (e.g. 024h and 026h) use 16-bit decoding.

R380EX INDEX/DATA REGISTERS

These registers form an index/data register pair for accessing R380EX registers. Only the lower six bits of the index register are implemented. Reads or writes to the R380EX Internal Data Register at address 026 return or modify the register that is selected indirectly through the six-bit R380EX Internal Index Register's content. Attempting to write to a single byte of the Index or Data register (e.g. writing only a single byte to 026h or 027h) results in both bytes of the selected register being written.

*

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Adr
Index																024
Data																026

KEYBOARD DATA REGISTER

The Keyboard Data Register implements the standard PC-Compatible keyboard data functions.

7	6	5	4	3	2	1	0	Adr
Keyboard data								060

PORTB REGISTER

The PortB Register supports speaker control and the timer counter2 functions which are available in conjunction with the Intel386 EX.

7	6	5	4	3	2	1	0	Adr
0	0	Timer Counter 2 OUT Status	Refresh Toggle	RAM	RAM	Speaker Data Enable	Timer Counter 2 GATE Enable	061
0	0							Reset

The Timer Counter 2 GATE Enable bit controls the GATE input to Counter 2, and enables counting when set. This bit is cleared on HW reset.

The state of the Speaker Data Enable bit, is ANDed with the Counter 2 OUT signal from the Intel386 EX to drive the SPKR output signal.

Bits 2 and 3 are RAM bits. These bits normally enable the IOCHCK and parity features. They are included to preserve PC compatibility, but these functions that are not supported in the R380EX.

The Refresh Toggle bit toggles every time a refresh occurs. This bit is cleared on HW reset.

Timer Counter 2 OUT Status reflects the state of the TIM2_OUT pin.

Bits 6 and 7 are read only bits. In the PC architecture, they reflect parity and IOCHCK errors, but these functions are not supported in the R380EX.

KEYBOARD STATUS/COMMAND REGISTER

The Keyboard Status/Command Register implements the standard PC-Compatible keyboard controller functions.

7	6	5	4	3	2	1	0	Adr
Keyboard status/command								064

RTC REGISTERS

These registers are used for an Index/Data Register pair for accessing the RTC and CMOS RAM.

7	6	5	4	3	2	1	0	Adr
RAM	RTC index							070
RTC data								071

Bit 7 of register 70 is the RAM bit and is included to preserve PC compatibility. It normally preserves the NMI enable function, but this function is not supported in the R380EX.

IDENTIFICATION REGISTER

The upper 12 bits of the ID Register is a read only constant identifying the part and its revision number.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
0	0	1	1	1	0	0	0	0	0	0	0	BIOS Revision				0
0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	Reset

Bits 7-4 are intended to encode the stepping revision. The remaining four register bits are read/writeable and are reserved for BIOS use only. These bits are reset to 0 by hardware reset.

POWER-UP OPTION REGISTER

The Power-up Option Register is a read-only register that latches the state of the MA[10:0] pins at power-up.

10	9	8	7	6	5	4	3	2	1	0	Idx
MA[10:0], latched while PWRGOOD is deasserted.											01

Bits 0-8 control the initial configuration of the R380EX (see Power-up Configuration Options). Bits 9 and 10 are also latched at power-up and can be used by the system designer to indicate system-level options. When the R380EX enters the low-power halt state, it redrives the contents of this register out to the MA[10:0] lines in order to minimize the static power consumption that the external pull-up or pull-down resistors could otherwise potentially consume.

BIOS CONTROL REGISTER

The BIOS Control Register is used to control access to the Flash boot device through use of the CE_BIOS# and WE_FLASH# pins. When using the CE_BIOS# and WE_FLASH# pins the Flash device is required to reside on the Intel386 EX's local bus.

15	14	13	12	11	10	9	8	Idx
	BIOS wait states			BIOSSIZE	LFLSHWE	BIOSWE	APPWE	02
0	1	1	1	0	0	0	0	Reset

7	6	5	4	3	2	1	0	Idx
F8000-FFFFF	F0000-7FFFF	E8000-EFFFF	E0000-E7FFF	D8000-DFFFF	D0000-D7FFF	C8000-CFFFF	C0000-C7FFF	02
1	1	1	1	0	0	0	0	Reset

The least-significant 8 bits enable are used to set the overall address range for CE_BIOS#. When a bit is set, the associated address region is included in the overall address decoding for assertion of the CE_BIOS# signal. These bits, when set, take priority over the Shadow 1/2 Register bits.

The APPWE bit is used in conjunction with the external WE_AP# pin, and the LFLSHWE bit. When the APPWE bit is set it allows WE_FLASH# to be asserted for all CE_BIOS# address regions except the top 128KB where the BIOS normally resides.

The BIOSWE bit is used in conjunction with the external WE_BIOS# pin. When the BIOSWE bit is set, it enables writes to the BIOS system code portion of the Flash device which resides at both the top of memory (upper 128KB just below 64MB) and the 128KB just below 1MB. Additionally, the portion just below 1MB is only written as allowed by the LFLSHWE bit. . These enable bits are cleared on HW reset.

The LFLSHWE bit when set, enables writes to the flash region below 1MB. This bit is cleared on HW reset. Both LFLSHWE and BIOSWE must be set to be able to write to the low BIOS region (just below 1MB). LFLSHWE and APPWE must both bet set to be able to write the application flash below 1MB.

The APPWE enable bit and the BIOSWE enable bit both have corresponding pins that affect these functions as well. Both the pin and the register bit must be set to enable writes to the associated flash region. The exception is if the DIO Mux Register specifies a pin as a DIO bit, then that pin is ignored for purposes of enabling writes to the Flash device.

The BIOSSIZE bit, when set, enables address decoder recognition of the top 16MB (0x30_0000-0x3f_ffff) and results in assertion of CE_BIOS for this address range. If the BIOSSIZE bit is clear, then only accesses to the top 1MB causes CE_BIOS to be asserted.

The BIOS wait states field determines the number of wait states inserted into a BIOS read access as follows:

Encoding	BIOS access Wait states
000	2
001	3
010	4
011	5
100	6
101	7
110	8
111	9

Writes to the BIOS region always require nine wait states.

SHADOW 1 REGISTER

The Shadow 1 Register contains two shadow control bits for each of the indicated memory address regions shown in the following illustration. (The Shadow 2 Register has an identical function for a different set of address ranges.)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
D4000-D7FFF		D0000-D3FFF		CC000-CFFFF		C8000-CBFFF		C4000-C7FFF		C0000-C3FFF		B0000-BFFFF		A0000-AFFFF		03
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset

The Shadow 1 Register is cleared to 0 by reset. The two bit pairs control the routing of memory accesses, for their respective regions, if the region is not already mapped to the flash BIOS by the BIOS Control Register. The Shadow Register bits are ignored when the corresponding region has already been designated for assertion of CE_BIOS#. When CE_BIOS# is not asserted for the region, then the Shadow Register bits are interpreted as shown in the following table:

Encoding	Function
00	Read/write ISA
01	Read ISA, write to DRAM
10	Read DRAM, write to the bit-bucket (neither an ISA nor a DRAM cycle is generated, but ready is returned to Intel386 EX)
11	Read/write DRAM

SHADOW 2 REGISTER

The Shadow 2 Register control bits are interpreted in the same manner as for the Shadow 1 Register, except that the address ranges covered are different.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
F8000-FFFFF		F0000-F7FFF		EC000-EFFFF		E8000-EBFFF		E4000-E7FFF		E0000-E3FFF		DC000-DFFFF		D8000-DBFFF		04
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset

DIO OUTPUT REGISTER

When the corresponding direction bit in the DIO Direction Register is set to output, and the corresponding mux bit in the DIO Mux Register specifies using the pin as a DIO pin, a DIO pin reflects a bit in the Data Output Register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
DIO Register Data																05
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset

The DIO Output Register is a read/write register. Writing to this register will cause pins to change if they are set up as outputs. Reading this register returns the value in the register, not the value of the DIO pins. This register is cleared on HW reset.

DIO INPUT REGISTER

The DIO Input Register reflects the actual state of the DIO pins. This is true regardless of the state of the DIO Direction Register or the DIO Mux Register. This is a read only register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
DIO pin data																06

DIO DIRECTION REGISTER

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
DIO direction data																07
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset

Each bit in the DIO Output Register defines whether or not to enable the output driver for that pin. An output driver is only enabled to drive the pin if the associated bit in the DIO Direction Register is a 1, and the corresponding bit in the DIO Mux Register specifies its use as a DIO pin. This register is cleared on HW reset.

DIO CONFIGURATION REGISTER

The DIO Configuration Register defines, for each bit in the register, whether the associated pin is used as a DIO pin or for the alternate function.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
DIO configuration data																08
MA7									MA6				MA5			At Reset

A 1 in a particular bit defines the associated pin be used as a DIO pin. During a hardware reset, each bit of this register is initialized to the value of the MA(x) signal shown in the register template above.

The state of MA7 at reset is reflected in bits 15-8. The state of MA6 at reset is reflected in bits 7-4. The state of MA5 at reset is reflected in bits 3-0. Please note that bits 5 and 6 of this register may not be written to a "1" by software unless MA6 was sampled high during a hardware reset. This is done to protect the flash write enable functions that are multiplexed onto the same pins as DIO5 and DIO6.

DO OUTPUT REGISTER

A DO pin reflects the associated bit in the Data Output Register when the corresponding bit in the DO Configuration Register specifies using this pin as a DO pin.

7	6	5	4	3	2	1	0	Idx
Reserved		DO Register Data						09
X		MA8						At Reset

The DO Output Register is a read/write register. Writing to the DO Output Register will cause pins to change state if they are setup as outputs. Reading this register returns the value in the register which are not necessarily the actual value of the DO pins. Each bit of this register is initialized with the value contained on MA8 during a HW reset.

OUTPUT CONFIGURATION REGISTER

The Output Configuration Register defines, for each pin in the DO port, whether it is used as a DO pin or for an alternate function.

7	6	5	4	3	2	1	0	Idx
X	X	X	IDEMD	EXRTC	EXTKB	DISMO	DISKB	0A
0	0	0	0	MA1	MA2	MA2	MA2	Reset

To enable DO outputs 16, 17, & 18, the DISKB bit is set. When the DISKB bit is set it causes the keyboard clock, data, and interrupt to be disabled. To enable DO outputs 19, 20, & 21, the DISMO bit is set. When the DISMO bit is set it causes the mouse clock, data and interrupt to be disabled.

If an external keyboard will be used, the EXTKB bit should be set. This will enable the keyboard chip select function for the pin on the R380EX.

When set, the EXTRTC bit enables the RTC_AS, RTC_RD#, RTC_WR# pin functions of the R380EX. When clear, RTC_IRQ, LED1#, and LED0# pin functions are enabled instead.

The IDEMD bit, when set, causes the R380EX to generate much shorter bus cycles to the I/O address space occupied by IDE. This allows the R380EX to support EIDE drives, in PIO mode 3 or 4, at a much higher transfer rate. Not all IDE drives capable of PIO mode 3 will be able to support the minimum read/write cycle time of the R380EX (180ns), but all IDE drives capable of PIO mode 4 should be capable of supporting this cycle time. The specific EIDE drive specifications must be checked for compatibility. When the IDEMD bit is clear, PIO mode 0 (the old IDE timing) is supported.

SMI TIMER CONTROL REGISTER

A write to the SMI Timer Control Register sets the Clock Prescaler Bit (PRE) state and the value in the 12-bit Idle Timer Latch. Writes to the Idle Timer Latch write through the latch and into the counter as well. Reads from the SMI Timer Control Register return the value in the Idle Timer Counter as well as the state of the PRE bit.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
PRE	Reserved			Idle timer latch (wr), timer/counter (rd)												0B
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset

If an Idle Timer countdown or reload event occur during a read of the Idle Timer (i.e., an SMI Timer Control register read), the countdown and/or the reload event is delayed until after the read is completed. If both events are delayed during the read, only the reload is performed after the read. If PRE is set, a 16 hertz clock is used to decrement the timer counter, and the counter can be set to a value in the range from about 63 milliseconds to about 256 seconds. When PRE is 0, a 1024 hertz clock is used to decrement the timer/counter, and the counter can be set to a value in the range from about 1ms to about 4 seconds.

The timer/counter is set to the Idle Timer Latch Value whenever any of the following events occur: the ACTIVITY# pin is asserted and bit 7 of DIO Configuration Register is 0 (enabling ACTIVITY# instead of DIO7 onto pin 124), the internal programmable chip select is enabled (see bit 6 of the SMI Control/Status Register) and asserted, the idle timer latch (SMI Timer Control) is written to by software, or the counter is at zero and the SMI has been acknowledged by the Intel386 EX (SMIACT is asserted).

If the timer/counter reaches zero, an SMI event is signaled (if SMI is enabled) and the timer is stopped. In this case, the timer is reloaded in response to SMIACT being asserted. Both the Idle Timer Latch and the timer/counter are cleared on HW reset. The prescaling counter (counts down) is also preset to all 1's by reset or a write to this register.

SMI CONTROL/STATUS REGISTER

The SMI Control/Status Register contains the global SMI enable and the SMM RAM Map enable, and enables the sources for SMI generation. It also contains several read-only status bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
RSV	MFGLP	FUPD	HLTS	TEXP	EXT1	EXT0	SMIMAP		ENPCS	ENSMI	SWSMI	ENHLT	ENTIM	EEXT1	EEXT0	0C
0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	Reset

The EEXT[1:0] bits enable an SMI event when one of the corresponding EXTSMIx pins is asserted. These bits are cleared on HW reset.

The ENTIM bit enables the SMI timing event when the SMI timer times out. This bit is cleared on HW reset.

The ENHLT bit enables an SMI event when a HALT bus cycle is detected on the Intel386 EX bus. This bit is cleared on HW reset.

The SWSMI bit is a software triggerable SMI event bit. When set, an SMI event is signaled. This bit is cleared on HW reset.

The ENSMI bit is the overall SMI event enable. This bit must be set to allow any SMI event to be signaled to the Intel386 EX. This bit is cleared on HW reset.

The ENPCS bit, when set, enables the CS_USR0 programmable chip select's output, when asserted, to set the idle timer back to its latch value. The chip select's daisy chaining capability works for this application as well. When CS_USR0 is used in this mode, the IOR and/or the IOW qualification bits of the CS_USR0 chip select register (and other chip select registers daisy-chained into CS_USR0) would normally be set to avoid spurious assertions of CS_USR0 in response to ISA bus address transitions.

The SMIMAP bits allow the SMI memory to be mapped to different regions of the DRAM when SMIACT is asserted as described in the following table. These bits are cleared on HW reset.

SMIMAP[1:0]	Address Mapping Function
0-	Address mapping disabled in all modes
10	When SMI is active, then remap 2xxxx → Axxxx and 3xxxx → Bxxxx
11	2xxxx → Axxxx , 3xxxx → Bxxxx normally, and if SMI is active, then also remap 4xxxx → 2xxxx and 5xxxx → 3xxxx

Status Indication of the SMI Control/Status Register. The TEXP, EXT1, EXT0, HLTS bits are status bits used to indicate which of the respective SMI sources caused the SMI interrupt. When an SMI event is signaled to the Intel386 EX, the corresponding bit in the register is set to indicate the source of the SMI event. These bits can be set without asserting an SMI#, if the overall SMI enable bit, ENSMI, is set to a 0; however, the individual SMI source enable bits must be set to allow a software polled mode of operation of the respective status bits. These status bits are cleared on HW reset, or can be cleared by writing a "0" to the corresponding bit. Writing a "1" into a bit will not change the state of the bit, and enables multiple event status to be maintained.

The MFGLP and FUPD bits are read-only status bits that follow the state of the corresponding R380EX external pin.

PROGRAMMABLE CHIP SELECT 0 (PCS 0) COMPARE AND MASK REGISTER

The four user-programmable I/O chip selects can be programmed and used individually or they can be "daisy-chained" together to form one or two extended chip select outputs that can decode up to four discontiguous regions. They can be combined as a group of four, three, or two chip selects.

Further, these programmable chip selects can be used to reset the SMI activity timer, even if the external programmable chip select pins are being used as digital I/O.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
MODE		OR	IOR	IOW	C15	Compare and mask bits										0D
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset

When set, the OR bit in the PCSx Register normally causes CS_USR(x+1) to be OR'ed into CS_USR(x). To eliminate building a circular latch circuit, CS_USR3 is a "boundary case" that does not implement the normal OR bit function. The OR bit in this case is just a read/writeable bit and serves no other useful function. For PCS0, the OR bit, when set causes CS_USR1 to be OR'ed into CS_USR0.

When set, the IOR bit in the PCSx Register requires IOR to be asserted along with the appropriate address to assert the chip select.

When set, the IOW bit in the PCSx Register requires IOW to be asserted along with the appropriate address to assert this chip select.

If both IOR and IOW bits are set, then the chip select is asserted whenever either IOR or IOW is asserted along with appropriate address. If neither bit is set, the chip select is asserted on the basis of the address decode alone.

The C15 bit in the PCSx Register is always compared to the address bit 15. If it does not match the chip select is not asserted.

The 10 bit compare and mask bit field in the PCSx Register is interpreted on the basis of the MODE bits as follows:

Mode	Decode function
00	Disable the chip select.
01	Decode a single byte address using the compare and mask bits as A[9:0] comparison bits only.
10	Decode a power of 2 region of address space using the compare and mask bits as described below.
11	Same as Mode 10, but, also assert IOCS16.

PCSx MODE 10, 11: Chip Select Block Decoding. When the MODE bits are set to "10" or "11", the Compare And Mask bits are used as follows. The least significant bit of the Compare And Mask Field bits that is a "0" specifies the demarcation line that separates the Compare Field bits from the Mask Field bits. The "first 0" and the bits of lower significance (all must be "1s") than this "first 0" cause the corresponding address bit(s) to be masked off. Only the Compare And Mask Field bits "to the left" of the "first 0" are used to compare to the corresponding address bits to determine if the chip select is to be asserted.

For example, if the Compare And Mask Field contains a "01 1111 0011" , then the Mask Field would be the three LSBs: "011", and the Compare Field bits would be "01 1111 0xxx" where the xxx fills in the mask field bits. With the Compare And Mask Field programmed as described, the address range 0x1F0-0x1F7 would cause the chip select to be asserted.

PCS 1 COMPARE AND MASK

The PCS 1 register performs the same function as PCS0 except that CS_USR1 is asserted when the comparitor matches and the OR bit causes the CS_USR2 to be OR'ed into CS_USR1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
MODE		OR	IOR	IOW	C15	Compare and mask bits										0E
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset

The PCS 1 Register is cleared on HW reset.

PCS 2 COMPARE AND MASK

The PCS 2 register performs the same function as PCS0 except that CS_USR2 is asserted when the comparitor matches and the OR bit causes the CS_USR3 to be OR'ed into CS_USR2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
MODE		OR	IOR	IOW	C15	Compare and mask bits										0F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset

The PCS 2 Register is cleared on HW reset.

PCS 3 COMPARE AND MASK

The PCS 3 register performs the same function as PCS0 except that CS_USR3 is asserted when the comparitor matches The OR bit serves no useful purpose for this "boundary" case.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
MODE		OR	IOR	IOW	C15	Compare and mask bits										10
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset

The PCS 3 Register is cleared on HW reset.

LED CONTROL REGISTER

The LED Control Register controls the LED pins.

7	6	5	4	3	2	1	0	Idx
Reserved		Test control		LED1 control		LED0 control		11
0	0	0	1	0	0	0	0	Reset

The low order two bits control the LED0 pin. The next two bits control the LED1 pin.

The Test control bits allow the LED0 and LED1 logic circuitry to be tested quickly. In a system, these bits would normally be set to zero and never changed.

All bits in the LED Control Register are cleared on HW reset.

Each of the two pins can drive an LED. The two register bits associated with each pin control the output in the following way:

0-	LED off (output is high)
01	LED on (output is low)
10	LED flashes once (on for about 0.5 seconds) after bits written as 10 *
11	LED flashing (about 1 Hz rate - 0.5 second on, 0.5 second off)

*The 10 encoding will be reset back to 00 encoding, once the LED has flashed for 0.5 seconds. This prevents the LED from continued flashing and allows software to once again write a 10 encoding into the bits to cause the LED to flash for another 0.5 seconds.

Please note that the changing the above bits from a 11 or a 01 encoding to a 10 encoding will result in an unpredictable LED blink pattern. The LED should be in the "LED off" state (00) before writing these bits to the 10 encoding to cause a single flash. Also, all timing for the LEDs is based upon the RTC-X1 crystal/clock input and assumes a 32,768 hertz signal is supplied.

The Test Control Register bits are defined as follows:

00	Normal operation. LEDs are controlled as described above.
01	Normal operation for LED1, LED0 outputs RTC_X2 input (normally 32,768 Hertz).
10	Test LED0. The LED0 timer/counter is subdivided into 8 bit and 7 bit sections whose outputs are driven on the LED0 (256Hz) and LED1 (128Hz) outputs assuming a 32,768 Hz RTC_X2 input.
11	Test LED1. The LED1 timer/counter is subdivided into 8 bit and 7 bit sections whose outputs are driven on the LED0 (128Hz) and LED1 (256Hz) outputs assuming a 32,768 Hz RTC_X2 input.

Except for the 01 encoding above, the test encodings must be used in conjunction with the LED1/0 control encodings to test an LED counter timer. To test LED0, for example, the test control bits would be set to 10 and one must additionally set the LED0 control to the 10 or 11 setting to enable counting.

CLOCK/RESET CONTROL REGISTER

All Clock/Reset Control Register bits are cleared by HW reset, except CDIV1, as described in the following paragraphs.

9	8	7	6	5	4	3	2	1	0	Idx
HLTS	ENWDT	ENKBR	DEL1	DEL0	ENHLT	SDIV1	SDIV0	CDIV1	CDIV0	12
0	0	0	0	0	0	0	0	1	0	Reset

CDIV1 and CDIV0 specify the divide ratio for CLK2OSC that is used to generate the CLK2OUT signals as shown in the followng table.

CDIV1	CDIV0	CLK2OUTx output
0	0	CLK2OSC divided by 4
0	1	CLK2OSC divided by 16
1	0/1	CLK2OSC divided by 1

CDIV1 and CDIV0 are initially reset to 0, but CDIV1 is set to 1 immediately after RESETCPU is driven deasserted. This allows phase synchronization to occur, at a lower local bus clock frequency, when RESETCPU is deasserted. This is done so that the setup and hold time parameters are met with very conservative timings and, yet, still allow the Intel386 EX to begin execution immediately following reset at the highest clock rate. In order to guarantee a

single CLK2OUT frequency transition the system's software must make only gray-code type transitions when changing these bits.

SDIV1 and SDIV0 specify the CLK2IN divide ratio used to generate the ISA bus BCLK signal according to the following table.

SDIV1	SDIV0	BCLK output
0	0	CLK2IN divided by 8
0	1	CLK2IN divided by 6
1	0	CLK2IN divided by 4
1	1	reserved

For 66, 50 and 32MHz CPU clock, one should set this field to 00, 01, and 10 respectively. This will yield 8.25, 8.33, and 8MHz for the BCLK frequency. When the CLK2OUTx output is connected to the CLK2IN input on the R380EX as recommended, then the BCLK and CLK2OUT dividers act in "series". For example, if all four clock divide bits are set to 0, then a 66MHz CLK2OSC input would be divided down to 16.5MHz to supply the CLK2OUT pin. The BCLK output would be 2.0625 MHz in this case.

When set, the ENHLT bit enables glitchless switching of the CLK2OUT signal to a 32KHz (RTC_X1) frequency and allows the system to conserve power while the Intel386 EX is in the HALT state. In addition, this bit enables generating an active low OSC_OFF signal. OSC_OFF is a shared function of the NAND_OUT pin. OSC_OFF can be used to powerdown an external oscillator or PLL clock synthesis chip to further reduce system power consumption. When the external oscillator is powered-down by this signal, then the DEL1 and DEL0 bits should be set to a non-zero setting as described below.

When exiting the HALT state, the DEL1 and DEL0 bits allow a programmable start-up delay before the CLK2OUT output is again sourced by the CLK2OSC clock input. The delay starts when an ADS# assertion is detected while the R380EX is in the HALT state. The OSC_OFF# signal is also deasserted upon detection of the ADS#. During the delay, the CLK2OUT output pin is held high. After the delay expires, the CLK2OUT output is again sourced from the CLK2OSC input. This allows an oscillator or PLL clock synthesis chip, that has been powered-down under control of the OSC_OFF# signal, time to return to its rated output frequency before it is again supplied through the CLK2OUT pin.

The encoding of the DEL1 and DEL0 control bits is described in the table below.

DEL1	DEL0	Delay
0	0	0
0	1	2ms
1	0	8ms
1	1	64 ms

When set, the ENKBR bit enables CPURESET to be asserted whenever a "keyboard reset" is issued. This is typically an I/O write to 0x64 with data of 0xFE. The RESETDRV signal is unaffected by this bit and will not be driven in response to a keyboard reset. The initial edge of this reset starts a CLK2IN timer in the R380EX that guarantees that the CPURESET signal is driven active for a minimum of 32 CLK2IN cycles.

When set, the ENWDT bit enables CPURESET and RESETDRV to be asserted whenever the WDTIN pin is asserted. The initial edge of this reset starts a CLK2IN timer in the R380EX that guarantees that the CPURESET signal is driven active for a minimum of 32 CLK2IN cycles.

The HLTS bit is a status bit that is set when a HALT instruction is executed by the Intel386 EX. This bit can be cleared by writing a "0" into this bit. Writing a "1" into this bit retains the previous value of HLTS. This bit could be used by system software (e.g., DOS) that uses a time of day interrupt to keep time. If the HLTS bit is set, the interrupt routine could copy the RTC data to the DOS time of day and then reset the bit. With the HLTS bit clear, the time of day should be correct, and the routine would update the time of day using the normal procedure.

DRAM CONTROL REGISTER

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
Reserved						PB	SB	B2	BANKBSIZE			A2	BANKASIZE			13
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	Reset

The BANKBSIZE and BANKASIZE bits each use the 3-bit encoding in the table below to determine their respective bank sizes:

BANKxSIZE	Bank Size	# Row Addr.	# Col. Addr.
000	Bank is not present	-	-
001	Bank's sub-bank is 256K x 16	9	9
010	Bank's sub-bank is 512K x 16	10	9
011	Bank's sub-bank is 1024K x 16	10	10
100	Bank's sub-bank is 2048K x 16	11	10
101	Bank's sub-bank is 4096K x 16	11	11
110	Bank's sub-bank is 8192K x 16, only if MA11 enabled	12	11
111	Bank's sub-bank is 16384K x 16, only if MA11 enabled	12	12

The MA11 signal must be enabled for 110 and 111 encodings to be valid. The R380EX will consider the bank to not be present if MA11 is disabled and one of these final two encodings has been selected. A hardware reset causes BANKASIZE to default to the 001 encoding above (256Kx16). BANKBSIZE defaults to the 000 encoding (bank not present) on reset.

The A2 and B2 bits determine if the second sub-bank of their respective bank are present (RASAS# and RASBS# are used). If present, these sub-banks must be the same size as first sub-bank in the bank (i.e. if present, the bank size bits also specify the size of the second sub-bank in the bank), and, the second sub-bank's starting address will follow immediately after the ending address of the first sub-bank in the bank. If A2 is set, then the R380EX will assert RASAS# for the appropriate address range. The B2 also helps determine if the second bank of sub-bank B is present, but is only valid when MA11 is not enabled. The R380EX will only support the second half of sub-bank B DRAMs when the B2 bit is set and MA11 is not enabled. All other conditions will result in disabling second bank of sub-bank B.

The SB bit determines the starting address of bank B DRAM (or DRAM-style flash). If SB is set, then, regardless of bank A's size, bank B's starting address is 0x200_0000 (i.e., it starts at 32MB). If SB is clear, then bank B's starting address immediately follows the last address that is consumed by bank A's memory.

The PB bit, when set, write protects bank B. When clear, writing to bank B is enabled. This may be useful if bank B holds Intel DRAM-style flash.

The table below shows the fixed DRAM address multiplexing scheme that is used by the R380EX. The Ax entries refer to the corresponding Intel386 EX address bit.

MA bit position	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
Column address bits	A24	A22	A20	A9	A8	A7	A6	A5	A4	A3	A2	A1
Row address bits	A23	A21	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10

DRAM TIMING REGISTER

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
MODE		EB	EA	REFTIM		IR	CD	RPB	CPB	MDB	CAB	RPA	CPA	MDA	CAA	14
0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	Reset

The CAA and CAB bits determine CAS assertion widths for the A and B banks respectively. When the CAx bit is 0, CAS is asserted for 3 CLK2 cycles on a read and for 2 CLK2 cycles on a write. When the bit is a 1, CAS is asserted for 5 CLK2 cycles on a read, and for 4 CLK2 cycles on a write. These bits are overridden when the MODE field is set to 01 as described below.

The MDA and MDB bits determine the delay between RAS# and the MAx (multiplexed address), and also the MAx to CAS# delay. When these bits are set, the default state, MDA/B cause a two CLK2 cycle delay on each side of MAx address row-to-column transition When these bits are clear, there is only a single CLK2 delay on each side of an MAx address transition. Even though a single CLK2 delay will usually satisfy the row address hold and column address setup parameters of even the slowest DRAMs, setting this parameter to a two CLK2 delay only affects the access time of a "new" page access. Thus, this parameter may be used to get extra RAS# access time for the first access in a page without adding wait-states to subsequent page-hit accesses.

The CPA and CPB bits determine the minimum CAS precharge time for each bank. When set, the default state, CPA/B cause the CAS# precharge time of the respective bank to be set to two CLK2 cycles in width. When clear, CAS# precharge time is limited to a single CLK2 cycle.

The RPA and RPB bits determine the minimum RAS precharge time, and the active time during refresh. When set, the default state, these bits cause the RAS# precharge time of the respective bank to be set to five CLK2s. Also the refresh RAS# active time is set to seven CLK2s. When clear, RAS# precharge time is set to 3 CLK2s and active refresh time is set to five CLK2s.

The CD bit sets the CAS assertion delay from the time when write data is enabled by the Intel386 EX, for both banks of DRAM. When set, the default state, the delay is set to four CLK2s. When clear, the delay is set to two CLK2s.

The IR bit determines whether ISA bus refresh is enabled. When set, the default state, ISA bus refresh is enabled. When clear, ISA bus refresh is disabled.

The REFTIM bits determine the refresh strategy and period to be used on the DRAM. The table below details the refresh functions supported.

REFTIM encoding	Refresh function
00	refresh disabled
01	initiated by Intel386 EX refresh cycles
10	every 15.3 usec (32768 Hz * 2) (The default after reset)
11	every 91 usec (32768 Hz / 3)

The EA and EB bits determine the type of DRAM installed in each bank. When set, EA/B cause the R380EX to generate DRAM timing that supports EDO DRAMs for the respective bank. When clear, the default state, the R380EX generates timing to support FPM DRAMs.

The MODE bits control the overall DRAM access strategy as shown in the following table.

MODE	Function
00	no Page Mode, don't assert NA#, default after reset.
01	no Page Mode, don't assert NA#, slow reads to allow FPM vs. EDO detection
10	Page Mode, assert NA#, normal operation mode for either FPM or EDO DRAM
11	Reserved

RESERVED REGISTER

This register is only useful for testing.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
Reserved																15

Writes to this register will corrupt (clear parts of) the RTC counter. Reads returned undefined data.

FULL I/O DECODE REGISTER

Note: This register is a new feature in the Rev. 2 silicon.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Idx
Reserved															Full Decode	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset

Bit 0 of the Full I/O Decode Register is used to select the partial or full decoding of R380EX internal PC-compatible resource I/O addresses. When set, the internal PC-compatible resources (RTC, Keyboard/Mouse controller, IDE interface, Port B register) are decoded using the full 16 bit I/O address. This results in no aliasing above 0x3ff. When clear, the addresses for the internal PC-compatible resources are decoded using address bits 0-9 only and will be aliased above 0x3ff.

The reset condition is clear, and backward compatible with Rev 0/1 silicon.

XIX. Electrical Characteristics

MAXIMUM RATINGS

Table 19.1 Maximum Ratings

Condition	Rating
Case Temperature Under Bias	-65C to 110C.
Storage Temperature	-65C to 150C.
Supply Voltage with Respect to Ground	-0.5V to Vcc + 0.5V
Voltage on any pin	-0.5V to Vcc + 0.5V

Caution: Stressing the device beyond the Absolute Maximum Ratings may cause permanent damage. These are stress ratings only. Operation beyond the Operating Conditions is not recommended and extended exposure beyond the Operating Conditions may affect reliability.

DC CHARACTERISTICS: 5.0 VOLT

Table 19.2 DC Characteristics: 5.0 Volt (V_{CC} = 5.0V ± 10%, Tc = -40°C to +85°C)

Symbol	Parameter	Min	Max	Units	Test Conditions	Notes
V_{IL}	Input low voltage		0.8	V		
V_{IH}	Input high voltage	2.0		V		
V_{OL1}	Output low voltage		0.4	V	I_{OL} = 16mA	1
V_{OH1}	Output high voltage	2.4		V	I_{OH} = -16mA	1
V_{OL2}	Output low voltage		0.4	V	I_{OL} = 8mA	2
V_{OH2}	Output high voltage	2.4		V	I_{OH} = -8mA	2
V_{OL3}	Output low voltage		0.4	V	I_{OL} = 2mA	3
V_{OH3}	Output high voltage	2.4		V	I_{OH} = -2mA	3
I_{IL1}	Input leakage current		±15	µA		
I_{IL2}	Input leakage current		±350	µA		4
I_{LO}	Output leakage current		±15	µA		
C_{IN}	Input capacitance		8	pF		
C_{OUT}	Output or I/O capacitance		15	pF	@ 1MHz	
I_{CC}	V_{CC} supply current		200	mA	@33MHz, V_{CC} = 5.25V	

Notes:

1. Applies to outputs with 16mA drivers.
2. Applies to outputs with 8mA drivers.
3. Applies to outputs with 4mA drivers.
4. This applies to inputs that have weak internal pull-ups or pull downs.

DC CHARACTERISTICS: 3.3 VOLT

Table 19.3 DC Characteristics: 3.3 Volt (VCC = 3.3V ± 10%, Tc = -40°C to +85°C)

Symbol	Parameter	Min	Max	Units	Test Conditions	Notes
V_{IL}	Input low voltage		0.8	V		
V_{IH}	Input high voltage	2.0		V		
V_{OL1}	Output low voltage		0.4	V	I_{OL} = 8mA	1
V_{OH1}	Output high voltage	2.4		V	I_{OH} = -8mA	1
V_{OL2}	Output low voltage		0.4	V	I_{OL} = 2mA	2
V_{OH2}	Output high voltage	2.4		V	I_{OH} = -2mA	2
V_{OL3}	Output low voltage		0.4	V	I_{OL} = 1mA	3
V_{OH3}	Output high voltage	2.4		V	I_{OH} = -1mA	3
I_{IL1}	Input leakage current		±15	µA		
I_{IL2}	Input leakage current		±350	µA		4
I_{LO}	Output leakage current		±15	µA		
C_{IN}	Input capacitance		8	pF		
C_{OUT}	Output or I/O capacitance		15	pF	@ 1MHz	
I_{CC}	V_{CC} supply current		130	mA	@33MHz, V_{CC} = 3.45V	

Notes:
1. Applies to outputs with 16mA drivers.
2. Applies to outputs with 8mA drivers.
3. Applies to outputs with 4mA drivers.
4. This applies to inputs that have weak internal pull-ups or pull-downs.

REAL-TIME CLOCK CHARACTERISTICS:

Table 19.4 Real-Time Clock Characteristics: (Tc = -40°C to +85°C)

Symbol	Parameter	Min	Max	Units	Test Conditions	Notes
V_{RTCVCC}	Real Time Clock Supply Voltage	2.4	VCC	V	VCC = 0: Powerdown	1

Notes:
1. Required for continued operation of real time clock. Active inputs on RTC_X1 and RTC_X2 required for time keeping.

INPUT CAPACITANCE

Input capacitance for any output or I/O pin: 15pF max.

Input capacitance for any input pin: 10pF max.

AC CHARACTERISTICS

To Be Specified

AC TIMING DIAGRAMS

To Be Specified

XX. Mechanical Specifications

PACKAGING AND DIMENSIONS

The R380EX is packaged in a 208-pin plastic quad flat pack, as shown in the following illustration. Package dimensions are listed in Table 20.1.

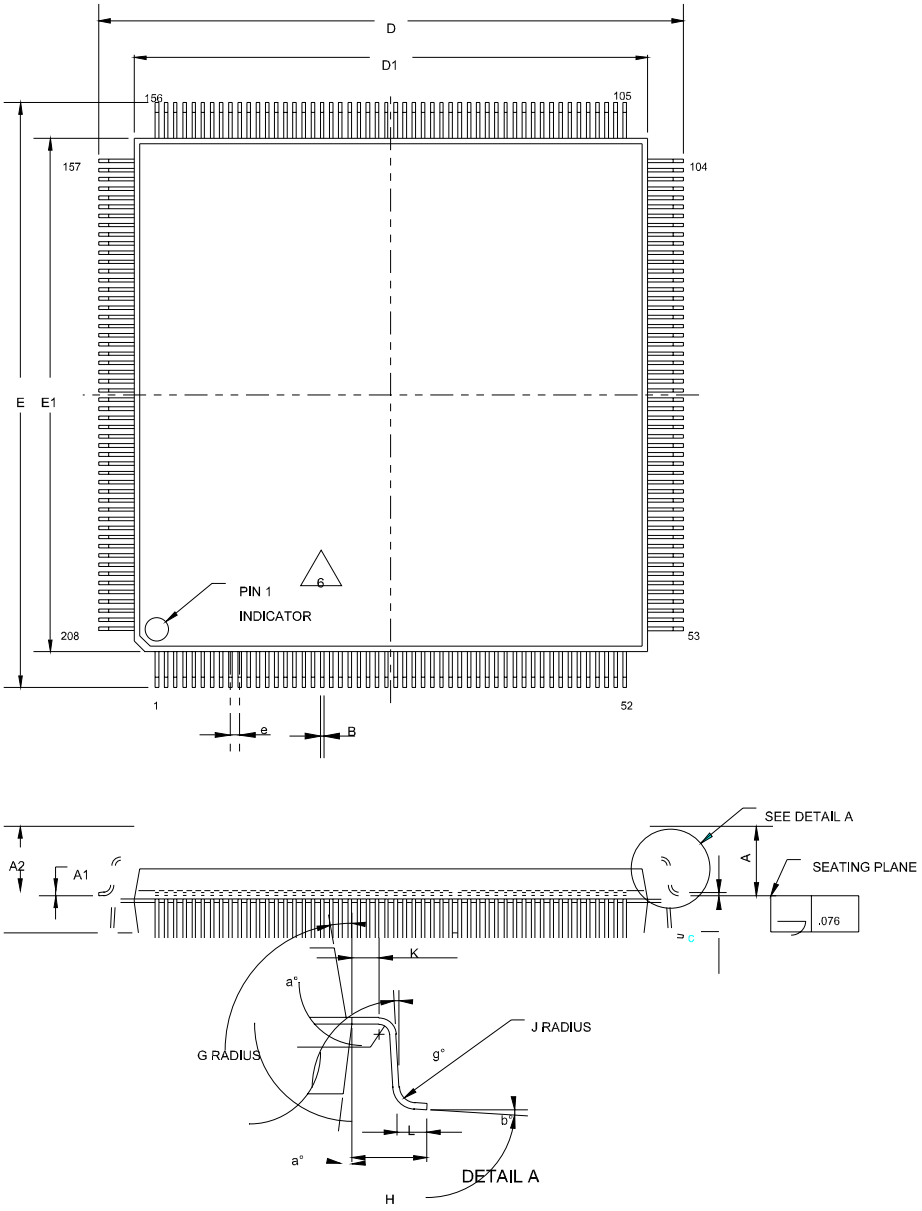


Figure 20.1: R380EX Dimensions

Table 20.1: R380EX Dimensions

Symbol	Minimum	Nominal	Maximum
A	3.40		4.07
A1	0.000		0.43
A2	3.17	3.37	3.67
D	30.35	30.60	30.85
D1	27.90	28.00	28.10
E	30.35	30.60	30.85
E1	27.90	28.00	28.10
L	0.40	0.50	0.60
e	0.42	0.50	0.58
B	0.13		0.28
c	0.10	0.15	0.20
a°	8		16
b°	0		7
g°	0		10
G	0.13	0.23	
H		1.30 Ref	
J	0.13		0.30
K	0.40		
2H		2.6	
Note: All dimensions in mm.			

THERMAL SPECIFICATIONS

The R380EX operates over the case temperature range of -40 to +85 °C. The thermal characteristics of the package are to be specified. Under no circumstance should the case temperature of the package exceed 105 °C.

XXI. Testability

GLOBAL TRISTATE

Asserting the FLOAT# input causes all of the output and I/O pins, except for the NAND output, to be tristate and enables the NAND tree. If any one input is low and all other inputs are high then the NAND signal is high, otherwise it is low. The FLOAT# input should be unasserted for normal operation.

NAND TREE ORDER

The NAND Tree starts with pin 2 and connects to all applicable pins sequentially through pin 114. The order is then 114, 116, 115, 117, 118. It then continues on linearly from pin 118 through pin 207. The following pins are excluded from the NAND Tree: Power (Vdd) and Ground (Vss) pins including RTC_VCC, FLOAT#, MA0, PWRGD, RTC_X1 and RTC_X2.

Appendix A: R380EX Errata List For Revision 0 Silicon

This document contains all known errata items, as of February 23, 1996, for the revision 0 silicon of the RadiSys R380EX Embedded System Controller chip.

The Table A.1 summarizes the errata items which are then explained in detail. Work-around suggestions are presented as appropriate.

Table A.1: R380EX Errata List for Revision 1 Silicon

REV 0 Errata #	Description	Fixed In Rev
1	32KHz oscillator does not oscillate when powered from a battery.	1
2	Keyboard IRQ and Mouse IRQ pins are swapped.	1
3	Internal keyboard does not transmit codes to keyboard correctly.	1
4	Internal keyboard's scan translation bit in the command byte is inverted.	1
5	Internal RTC (146818) power sense (PS) input pin input and IRQ output pin are inverted	1
6	Internal PS/2-compatible mouse control is non-functional	1
7	Address pipelining malfunctions in several modes	1
8	Ready (READY#) pin signaling limits operation to 25Mhz	1
9	QDACKA output pin is inverted	1
10	QDACKA pin is sourced from DACKB# pin.	No change
11	DMA cycles that follow immediately after an ISA bus split transaction malfunction	1
12	Refresh bit (bit 4 of the PortB, i/o address 0x61) does not toggle	1
13	IDED7 pin does not work properly	1
14	DRAMHLDA#/DATA_EN# Power-up configuration reversed.	No change

Errata #1: 32KHz oscillator does not oscillate when powered from a battery.

When battery operation of the RTC is attempted with the chip's main VDD power removed, the RTC's 32KHz oscillator is unable to oscillate. The RTC_X1, RTC_X2, and RTC_PS pin's input protection diodes are tied to the R380EX's core VDD. When VDD is removed, with RTC_VCC powered, the current path through the diodes attempts to power the core and everything else attached to the R380EX VDD pins.

Revision 1 silicon will correct this problem.

Workaround: If battery operation of the RTC is required, use an external RTC.

Errata #2: Keyboard IRQ and Mouse IRQ pins are swapped.

In Revision 0 silicon, KB_IRQ is on pin 43 and MO_IRQ to pin 18. As a result, when disabling the keyboard signals to enable digital outputs DO16-18, the signals KB_DATA, KB_CLK, and MO_IRQ are actually disabled. When disabling the mouse signals to enable digital outputs DO19-21, the signals MO_CLK, MO_DATA, and KB_IRQ are actually disabled.

Revision 1 silicon will route KB_IRQ to pin 18 and MO_IRQ to pin 43 as shown on the data sheets, and the keyboard and mouse signals will be disabled with the proper grouping..

Errata #3: Internal keyboard does not transmit codes to keyboard correctly.

The internal keyboard controller does not transmit commands to the keyboard correctly. The R380EX keyboard controller does correctly receive scan codes from the keyboard. This does prevent R380EX host system from correctly issuing any keyboard commands. Users will notice the transmit problem when pressing the Caps Lock, Num Lock, or Scroll Lock. Although these keys will toggle their respective functions normally, the LED's associated with these keys will not change state.

Revision 1 silicon will correct this problem.

Workaround: Use an external keyboard controller, if command transmission to the keyboard is required by the system.

Errata #4: Internal keyboard's scan translation bit in the command byte is inverted.
Keyboard controllers have a scan translation table selection bit used to select raw or translated keycodes. In the Revision 0 silicon, this bit is incorrectly inverted.

Revision 1 silicon will correct this problem.

Work around
The scan translation bit is normally only accessed at power-on by the BIOS. A BIOS is available from RadiSys which automatically detects the R380EX revision and flips this bit, if necessary.

Errata #5: Internal RTC (146818) power sense (PS) input pin input and IRQ output pin are inverted.
The RTC_PS input clears the valid RAM and time indicator (VRT) bit in the RTC when it is deasserted. De-assertion usually happens when the battery power fails. On revision 0 silicon, the RTC_PS input pin must be taken high, rather than low, to clear the VRT bit. On revision 0 silicon, the RTC_IRQ bit is asserted low rather than high.

Revision 1 silicon will correct this problem.

Work-around
The RTC_IRQ pin must be inverted externally before it is routed into the Intel386 EX. Due to errata #1, the RTC_PS pin should be grounded when using Rev. 0 silicon. Optionally, an external RTC may be used.

Errata #6: Internal PS/2-compatible mouse control does not function properly.
The PS/2-compatible mouse controller portion of the R380EX internal keyboard/mouse controller does not function properly, and should not be used.

Workaround: A mouse may be hooked to one of the Intel386 EX's serial ports. If a PS/2-compatible mouse is desired, an external keyboard/mouse controller (82PC42) will be required in conjunction with Revision 0 R380EX silicon.

Errata #7: Address pipelining malfunctions in several modes.
The R380EX does not properly assert the NA# output when configured to operate with Fast Page Mode DRAM. In the DRAM Timing Register (index 14h), FPM is selected when EB and EA (bits 13 and 12) are clear.

The R380EX only asserts NA# properly when configured to operate with Extended Data Out DRAM (the EB and EA bits are set). In addition, CD (bit 8) in the DRAM Timing Register must be clear for proper pipelined operation.

Finally, the R380EX does not properly control the DATA_EN# output when it encounters address pipelining.

Revision 1 silicon will correct this problem.

Workaround: Disable address pipelining by leaving the R380EX's NA# output unconnected and tying the Intel386 EX's NA# input to VCC. Or, if pipelining is desired, use EDO DRAM, and make sure the CD bit is clear. Also, do not use data transceivers between the DRAM data bus and the Intel386 EX data bus.

> *Note: The NA# input to the Intel386 EX may be tied to VCC. This completely disables address pipelining. However, the NA# input must not be grounded in an attempt to unconditionally enable address pipelining. The R380EX is not designed to function correctly if address pipelining is forced on. Instead, connect the NA# input may instead to the corresponding R380EX output. This allows the R380EX to initiate address pipelining when appropriate, subject to the limitations described in this errata.*

Errata #8: Ready (READY#) signal limits operation to 25Mhz.
On Revision 0 silicon, READY# is returned to the Intel386 EX in Phase 2 for ISA bus or internal R380EX register accesses. This limits operation of the Revision 0 silicon to 25Mhz (50MHz CLK2)

Revision 1 silicon will correct this problem by returning READY in Phase 1.

Workaround: None: Operation is limited to 25MHz.

Errata #9: QDACKA output pin is inverted.
The QDACKA pin is driven high to signal a DMA acknowledge cycle. The ISA bus requires an active low version of this signal.

Revision 1 silicon will correct this problem.

Workaround: Add an external inverter to this signal.

Errata #10: QDACKA pin is sourced from DACKB# input pin.
Initial documentation indicates that DACKA# input pin is used to derive the QDACKA output. The documentation will be changed to reflect the renaming of QDACKA to QDACKB.

Revision 1 silicon will not change this function, the documentation will reflect the name change.

Workaround: Use the DACKB# input when QDACKB is desired.

Errata #11: DMA cycles that immediately follow an ISA bus split transaction malfunction.
When an 8-bit DRQ/DACK# cycle follows immediately after a Intel386 EX code-initiated (as opposed to a DMA-initiated) 16 bit I/O read or 16 bit memory transaction on the ISA bus, then the byte steering logic is incorrectly forced to always transmit DMA data between the Intel386 EX's data byte 1 lane (D8-D15) and the ISA bus's byte 0 lane (SD0-SD7).

Revision 1 silicon will correct this problem.

Workaround: If there is no 8-bit memory (ROM or RAM) on the ISA bus, it is possible to workaround this bug by tying the MEMCS16# pin low and insuring that all 8-bit I/O peripherals are accessed with 8-bit I/O instructions.

Errata #12: Refresh bit (bit 4 of the PortB, I/O address 0x61) does not toggle.
Bit 4 of the PortB register, known as the "refresh bit", does not toggle, even though refresh is occurring.

Revision 0 silicon will always return this bit as "0" when read.

Revision 1 silicon will correct this problem.

Workaround: Insure that software does not attempt to use this bit to time a function.

Errata #13: IDED7 pin does not work properly.
During IDE reads (0x1F0-1F7, 0x3F6-3F7) processor data bit 7 should be read from pin IDED7; however during the READY# cycle, the ISA bus signal SD7 is sourced instead.

Revision 1 silicon will correct this problem.

Workaround: Treat the IDE's 7th data bit should be connected to SD7 either directly or through '245 buffers as required by your application. This hooks IDE data bit 7 in the same manner as all other IDE data bits; however if a standard floppy controller is present in the design, floppy controller reads of 0x3F7 (floppy controller drives only bit 7 of the ISA bus with disk change information)will fight the IDE controller's data bit 7 (or the '245 buffer acting on its behalf). By inserting a 150 ohm resistor in series with the IDE's SD7 bit driver, it is possible to make this work as well. Please consult the R380EX reference schematics for more details.

Errata #14: DRAMHLDA#/DATA_EN# Power-up configuration reversed.
The multiplexed DRAMHLDA# / DATA_EN# output function selected by MA power-up input bit 4 is the opposite of what is shown. The actual configuration is:

MA4 = 0 at power-up selects DATA_EN#

MA4 = 1 at power-up selects DRAMHLDA#

Revision 1 silicon will not change this function, the documentation will reflect the name change.

Workaround. Use the polarity indicated above to select the desired function. The DATA_EN# function on pin 147 should be used instead. The DATA_EN# alternate function of pin 33 is included for historical reasons and should be ignored. The final R380EX documentation will remove references to this alternate function.

Appendix B: R380EX Errata For Revision 1 Silicon

This document contains all known errata items, as of June 12, 1996, for the Revision 1 silicon of the RadiSys R380EX Embedded System Controller chip.

Table B.1 summarizes the errata items which are then explained in detail. Work-around suggestions are presented as appropriate.

Table B.1: R380EX Errata List for Revision 1 Silicon

Rev 1 Errata #	Description	Rev. fix
1	SMI mode interferes with flyby DMA cycle	2
2	First SMM fetch may be from “normal” rather than remapped DRAM	2
3	Flyby DMA with address 0xx061h incorrectly updates PORTB register	2
4	During an ISA memory read data is read from IDED7 rather than AT.SD7	2
5	DRAM access followed by DMA to same DRAM Page can cause READY hang	2
6	In EIDE mode, ISA memory cycles in 0x1f0-0x1f7 and 0x3f6-0x3f7 are sequenced as EIDE transfers instead of memory transfers.	2

Errata #1: SMI mode interferes with flyby DMA cycle

If a flyby DMA cycle occurs while the Intel386 EX is in SMI mode *and* SMI address remapping is enabled in the R380EX *and* the DMA address matches the SMI address region (0x20000-0x3FFFF), then the DMA access will be remapped according to the SMI remapping (a DMA cycle should not be remapped).

Revision 2 silicon will correct this problem.

Workarounds:

- Disable SMI remapping.
- If SMI remapping is required, then the system software must insure that the DMA address never matches the SMI remapping address region.

Errata #2: First SMM fetch may be from “normal” rather than remapped DRAM

If the first access in SMM (initiated by an System Management Interrupt) occurs immediately after a normal (non SMM) Intel386 EX access to a memory address between 0x003fc 00-003ffff, *and* the R380EX DRAM controller is operating in page-mode *and* SMI remapping is enabled, then the SMM handler, whose first access is always to 0x003fffe, will fetch the data from "normal" address 0x003fffe, instead of from the remapped DRAM address (0x0bfffe).

Revision 2 silicon will correct this problem.

Workarounds: Use one of the following workarounds:

- Disable page mode operation. (reduces system performance by 15-40%).
- Don’t remap SMI memory. Software must insure that at least the region between 0x38000-3FFFF is never occupied by anything but the SMI handler.
- Fix in hardware by detecting SMIACT# pin change and signalling a DRAMHOLD to the R380EX until DRAMHLDA is returned. This causes the R380EX to remove itself from page mode before the first access occurs in SMM mode and also before the first access occurs after leaving SMM mode. The details for a PAL for this workaround follow.

```
module fixsmi
title 'Pal used to fix SMI. (c) Copyright 1996, RadiSys Corporation'

"This pal fixes an error in the R380 REV1 silicon.
"
"The REV1 R380 has a problem when entering and leaving system management
"mode (SMM) in that the DRAM page is not invalidated.
"
"This PAL forces a page miss when entering and leaving SMM, by asserting
"a request for DRAM via DRAM hold request.  This invalidates the current
"page.

fixsmi device 'P16V8R'; "Uses 16V8-type PAL or GAL

"inputs
clk2 pin 1;             "CLK2 that drives CPU
!smiact pin 2;          "smi active input
!drmhlda pin 3;         "dram hold acknowledge
!outen pin 11;          "output enable

"outputs
!drmhold  pin 17 istype 'reg';  "dram hold output
smiactd  pin 16 istype 'reg';   "delayed smi active
smiactdd  pin 15 istype 'reg';  "

equations

smiactd := smiact;      "smi active delayed one clock
smiactd.clk = clk2;

smiactdd := smiactd;    "smi active delayed two clocks
smiactdd.clk = clk2;

drmhold := smiactd & !smiactdd   "entering SMI
         # !smiactd & smiactdd   "leaving SMI
         # drmhold & !drmhlda;   "hold until holda from CPU
drmhold.clk = clk2;

end fixsmi
```

Errata #3: Flyby DMA with address 0xx061h incorrectly updates PORTB register

Bits 3, 2, and 1 of the PORTB register incorrectly get updated on a flyby DMA read access (DRAM read, IOW), when the DMA address goes to 0x061 modulo 1024. Bits 3 and 2 are RAM bits only and do not perform any function inside the R380EX. (In most PCs, these bits are used to enable IOCHK and DRAM parity check. These two functions are not supported directly by the R380EX). Bit 1 is the speaker data enable. Thus a flyby DMA read could cause the speaker to start or stop a speaker tone as a side effect.

Note that the R380EX does correctly support two-cycle DMAs provided the DACK0/1# signals are not connected to the R380EX (and DMAs do not occur to the SMI address region when the SMI capability is being used by the system unless SMI remapping is not used).

Workarounds: To remove the speaker side effects:

- Insure that bit 0 of PORTB is 0 (Timer/Counter 2 GATE enable bit) during any DMA activity period. This works provided speaker tones are never generated during DMA activity.
- Implement PORTB bit 1 (enable speaker) and SPKROUT function gate externally. This can be done by ORing TMR2OUT from Intel386 EX with the inverted version of PORTB bit 1.

Revision 2 silicon will correct this problem.

Errata #4: During an ISA memory read data is read from IDED7 rather than AT.SD7

During an ISA memory read operation, the R380EX incorrectly routes information from IDED7 to processor data bus bit 7. AT.SD7 should be routed to processor data bit 7 instead.

Workaround: Use a ~100Ω resistor connected between IDED7 and AT.SD7. The IDE drive will be 3-stated during this operation and no contention between these bits should occur for all other cases as well. The R380EX validation/reference board has this fix.

Revision 2 silicon will correct this problem.

Errata #5 DRAM access followed by DMA to same DRAM Page can cause READY hang

When a normal Intel386 EX data access is followed by a DMA access that hits the same DRAM page as the previous access, the Intel386 EX becomes READY# hung.

Workaround: This can be worked around in hardware by disabling page mode operation. (R380EX register 0x14, bits 15 and 14 must be set to 0).

Revision 2 silicon will correct this problem.

Errata #6: In EIDE mode, ISA memory cycles to 0x1f0-0x1f7 and 0x3f6-0x3f7 are sequenced as EIDE transfers instead of memory transfers

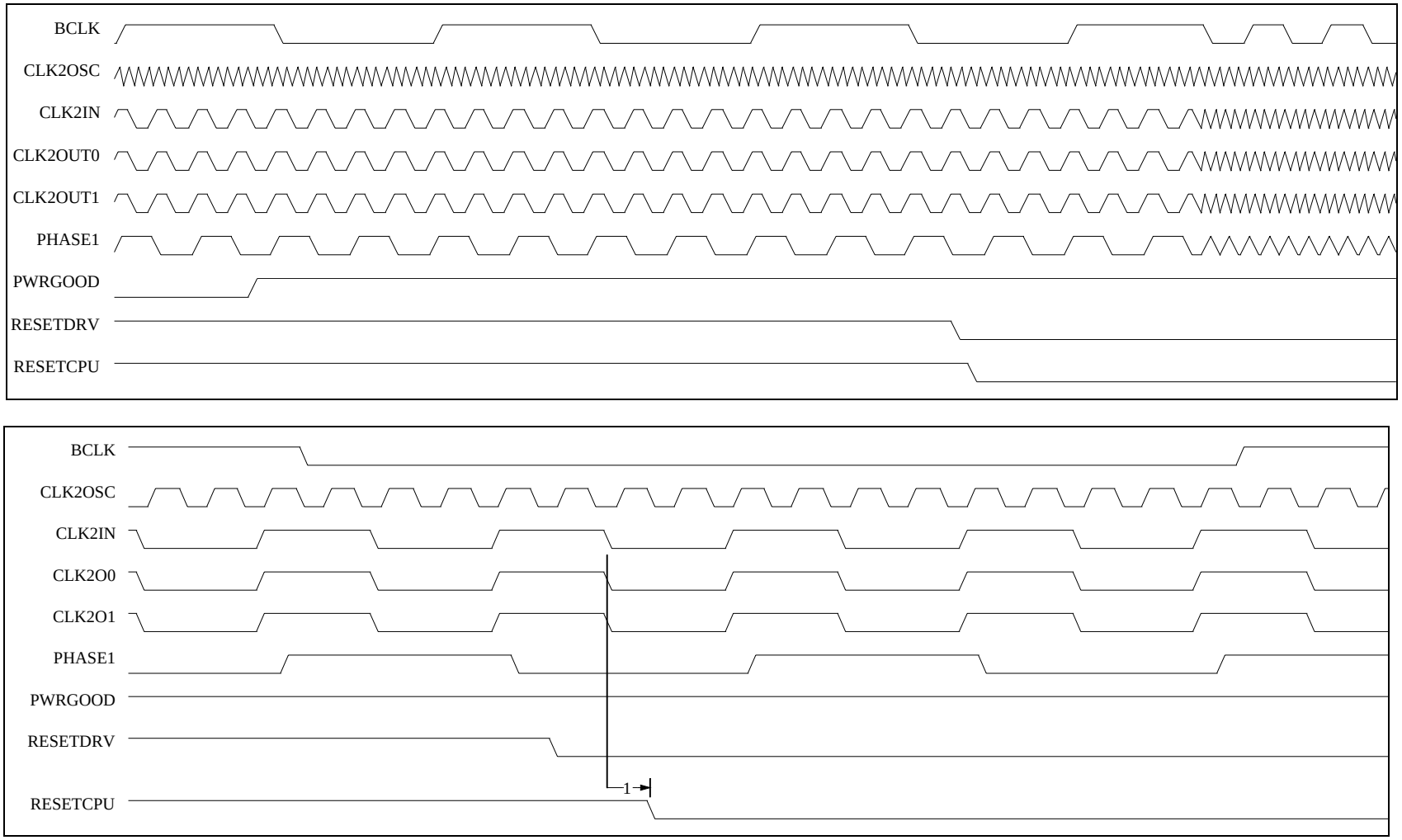
In enhanced IDE mode (IDEMD = 1, bit 4 of register 0xA), ISA memory transfers whose lower 10 address bits match the IDE decode range (0x1f0-0x1f7 and 0x3f6-0x3f7) are sequenced as EIDE transfers instead of memory transfers.

Workarounds:

- Only use IDE mode 0 (IDEMD = 0). The problem only occurs when the faster transfer rate is enabled in the R380EX (IDEMD = 1).
- If EIDE operation is desired and no devices on the ISA bus are memory mapped, a hardware workaround can be performed that insures that the IDE drive is not accessed by memory accesses to the ISA bus. (For example, a memory manager may scan ISA memory before deciding where to install itself, even though it never actually uses ISA memory). To accomplish the fix, simply "OR" the Intel386 EX's M/IO signal into the IDERD# and IDEWR# signals. This will disable them during memory operations (bus pipelining never occurs during I/O operations - the R380EX only asserts NA# during DRAM accesses).

Revision 2 silicon will correct this problem.

Figure A1: R380EX Clock - Reset Timing Waveform



NOTE 1: PWRGOOD signal is sampled asyncrounously

Table A1: R380EX Clock - Reset AC Characteristic

Symbol	Parameter	Min	Max	Description
1	tPD_CLK2IN:RESET	2	11	Prop delay from CLK2 low to Reset low (Up to 50 pf load)

Figure A2: R380EX System Management Interrupt Timing Waveforms

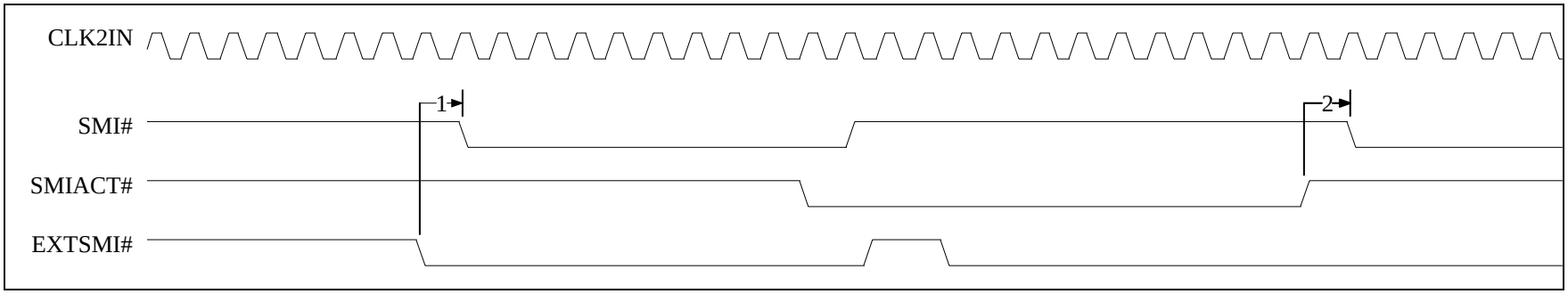


Table A2: R380EX System Management Interrupts Signals Propagation Delays

Symbol	Parameter	Min	Max	Description
1	tPD_EXTSIM#:SMI#	-0	18	Propagation delay from EXTSMI# low to SMI# low
2	tPD_SMIACT#: SMI#	-0	18	Propagation delay from SMIACT# low to SMI# high

Figure A3: R380EX Flash Access Waveform

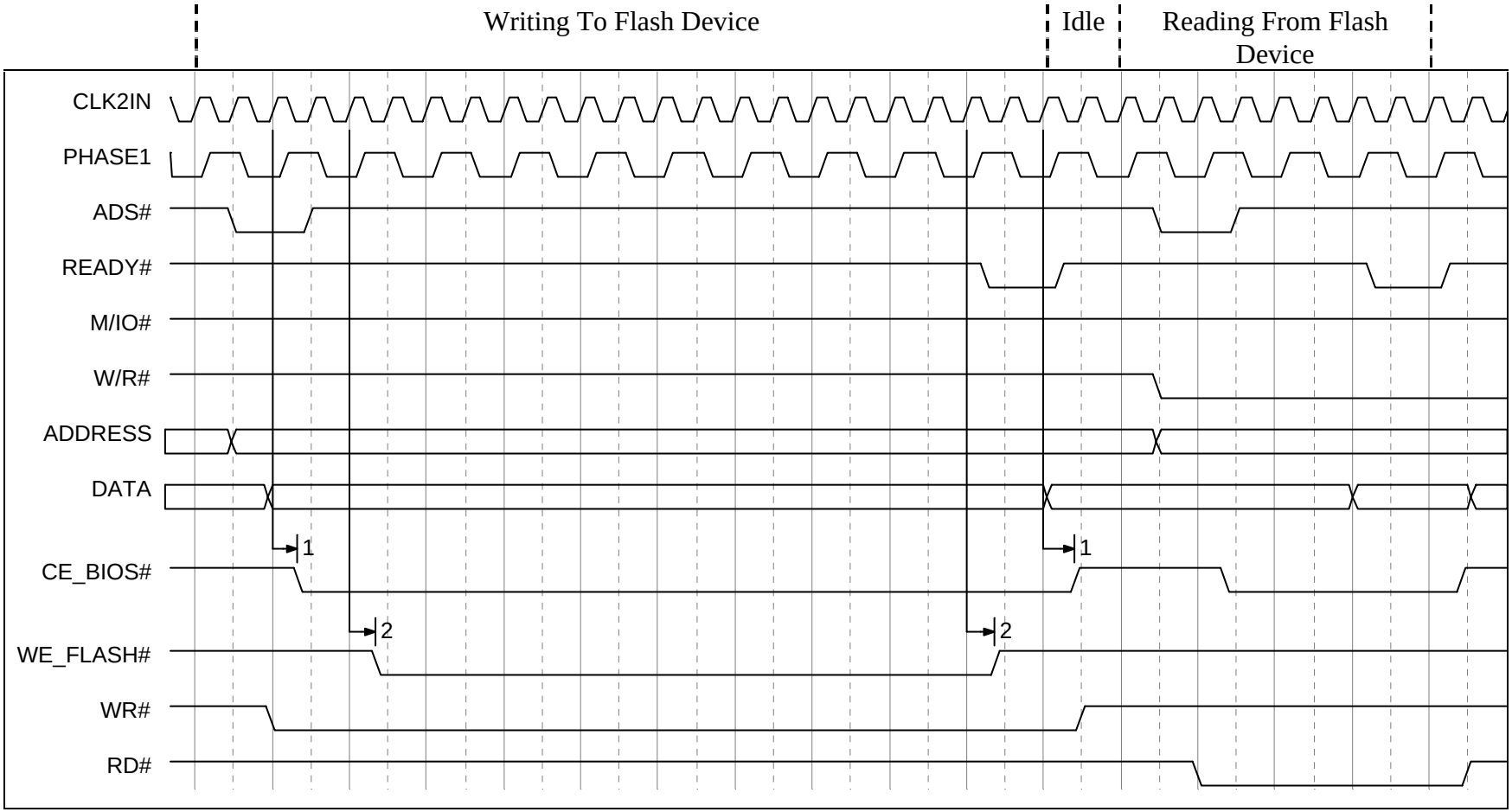


Table A3: R380EX - Flash Memory Access Timing

Symbols	Parameters	Min	Max	Description
1	tCO_CLK2IN : CE_BIOS#	2	12	Prop delay form CLK2IN high to CE_BIOS Valid
2	tCO_CLK2IN : WE_FLASH#	2	12	Prop delay from CLK2IN high to WE_FLASH Valid

NOTE1: For flash write respective write enable input WE_BIOS or WE_AP must be asserted.

Figure A4: R380EX IDE Timing Waveforms Overview

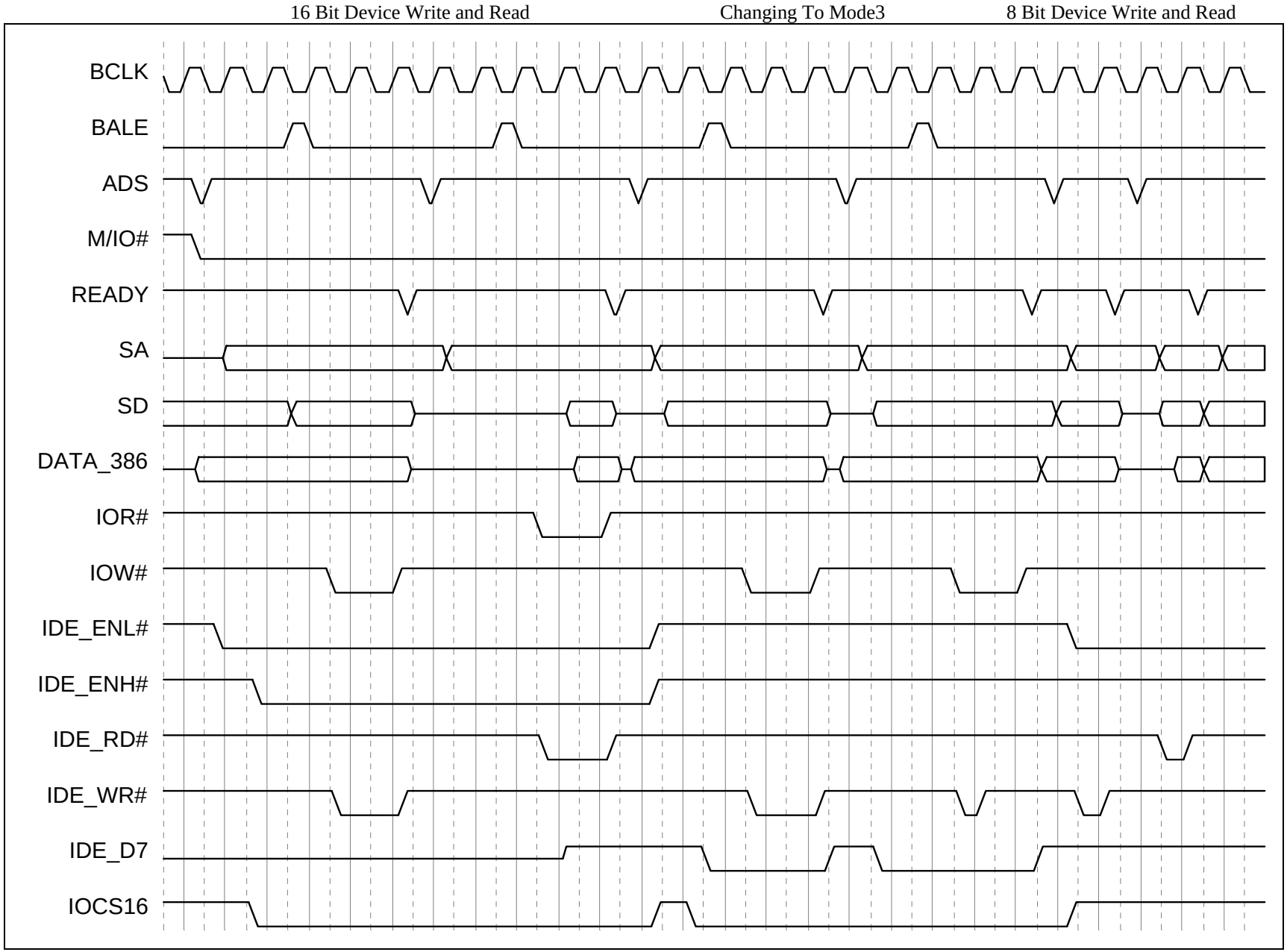


Table A4-A-B-C: R380EX - IDE Drive Access Timing

Symbols	Parameters	Min	Max	Description
1	tPD_ADS:IDE_ENL#l	-	3 CLK2IN	Prop delay from ADS# low to IDE_ENL#l low
2	tPD_IOCS16:IDE_ENL# L	2	6	Prop delay form IOCS16# low to IDE_ENL# low
3	tCO_CLK2IN:IDE_ENL# L	2	15	Prop delay form CLK2IN high to IDE_ENL# low
4	tCO_CLK2IN:IDE_ENL# H	2	15	Prop delay form CLK2IN high to IDE_ENL# high
5	tCO_CLK2IN:IDE_ENH# L	2	15	Prop delay from CLK2IN high to IDE_ENH# low
6	tCO_CLK2IN:IDE_ENH# H	2	15	Prop delay fromCLK2IN high to IDE_ENH# high
7	tCO_CLK2IN:IDE_RD#	2	12	Prop delay fromCLK2IN low to IDE_RD# Valid
8	tCO_BCLK-IDE_WR#	2	12	Prop delay from CLK2IN low to IDE_WR# Valid
9	tSU_IOCS16:CLK2IN	14	-	Setup time from IOCS16# low to CLK2IN
-	t_HLD_IOCS16:CLK2IN	end of cycle	-	Hold time from IOCS16# low to CLK2IN

Figure A4-A: R380EX Mode 0 IDE Timing Waveforms

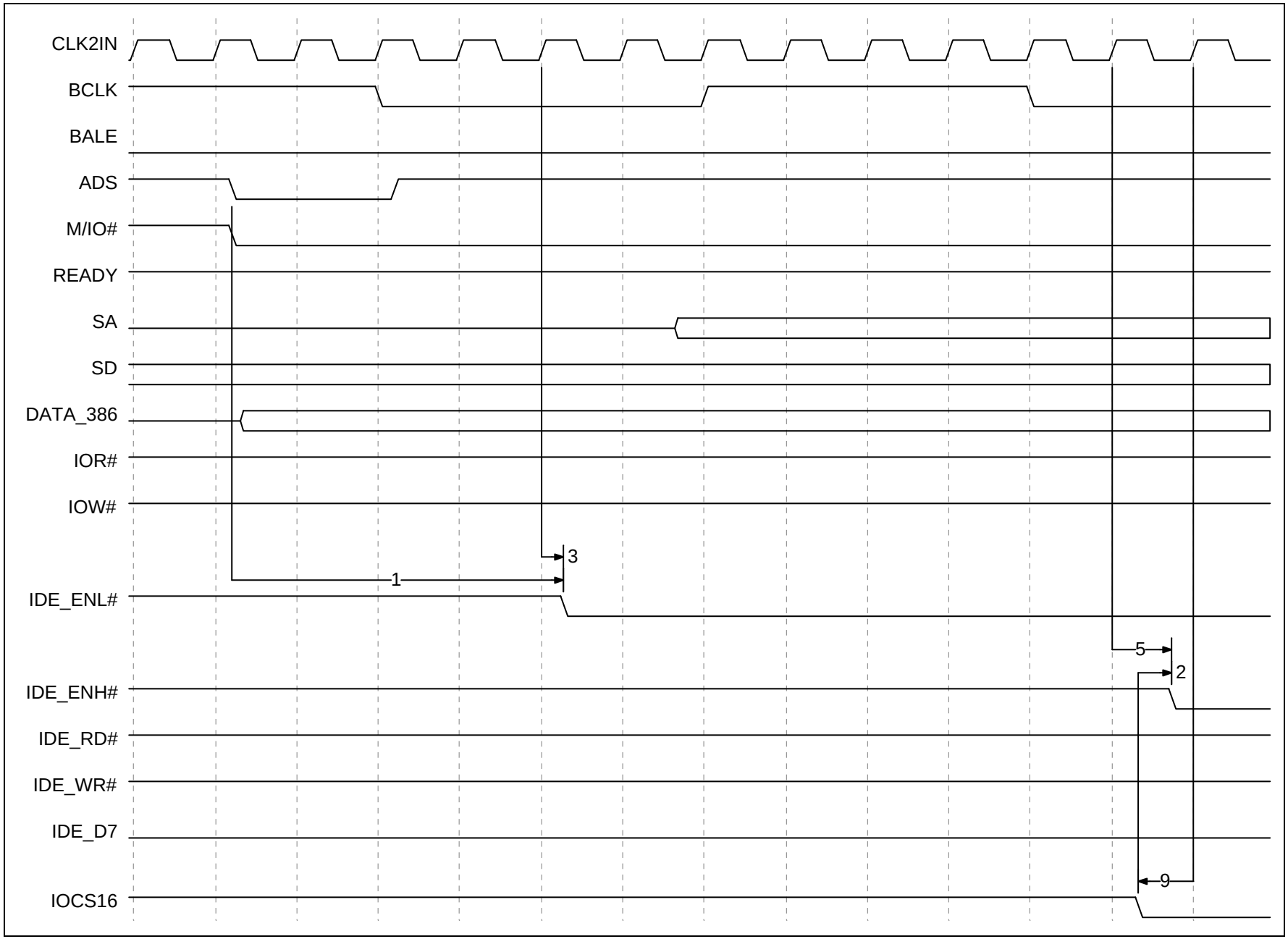


Figure A4-B: R380EX Mode 0 IDE Timing Waveforms

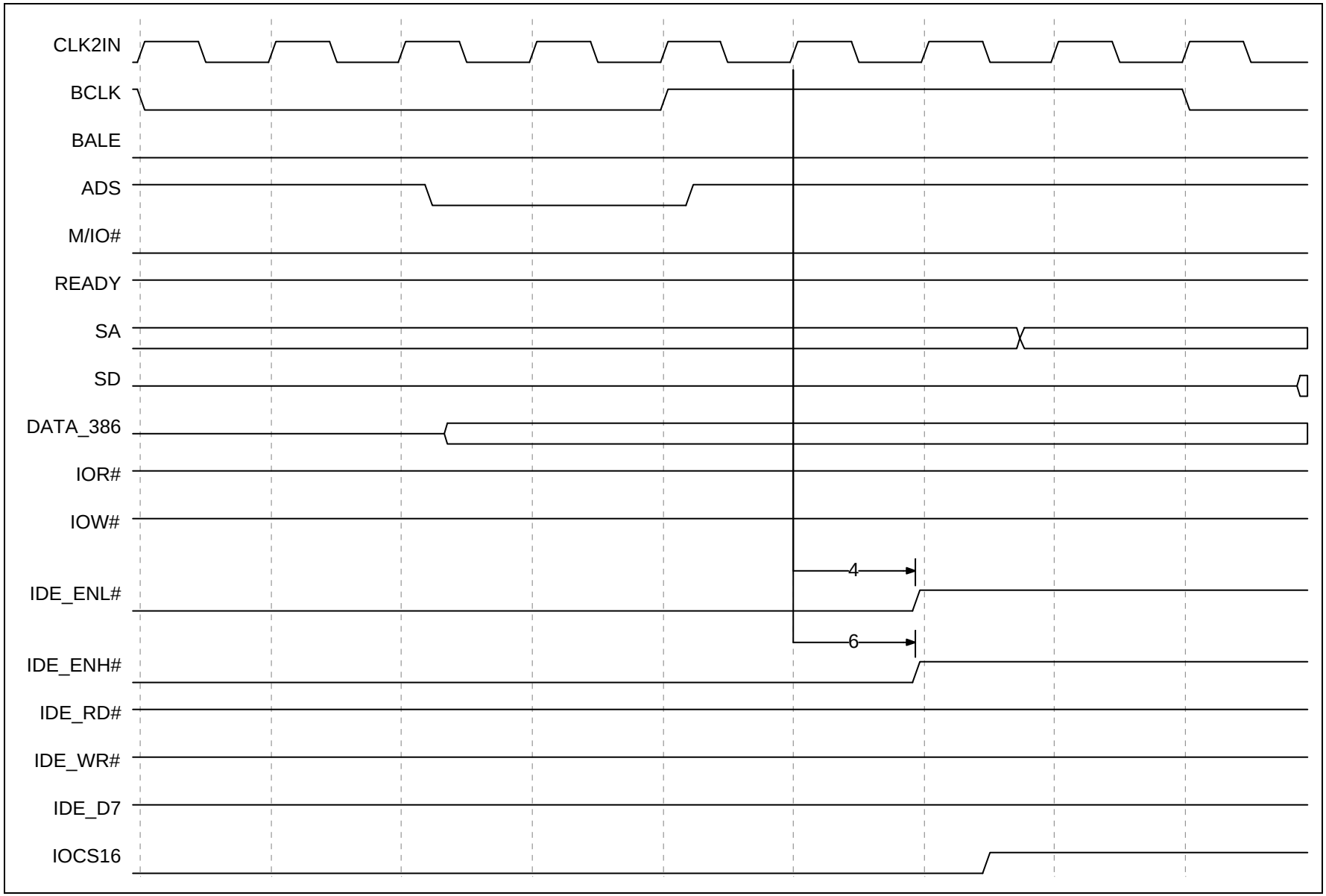


Figure A4-C: R380EX Mode 3 IDE Timing Waveforms

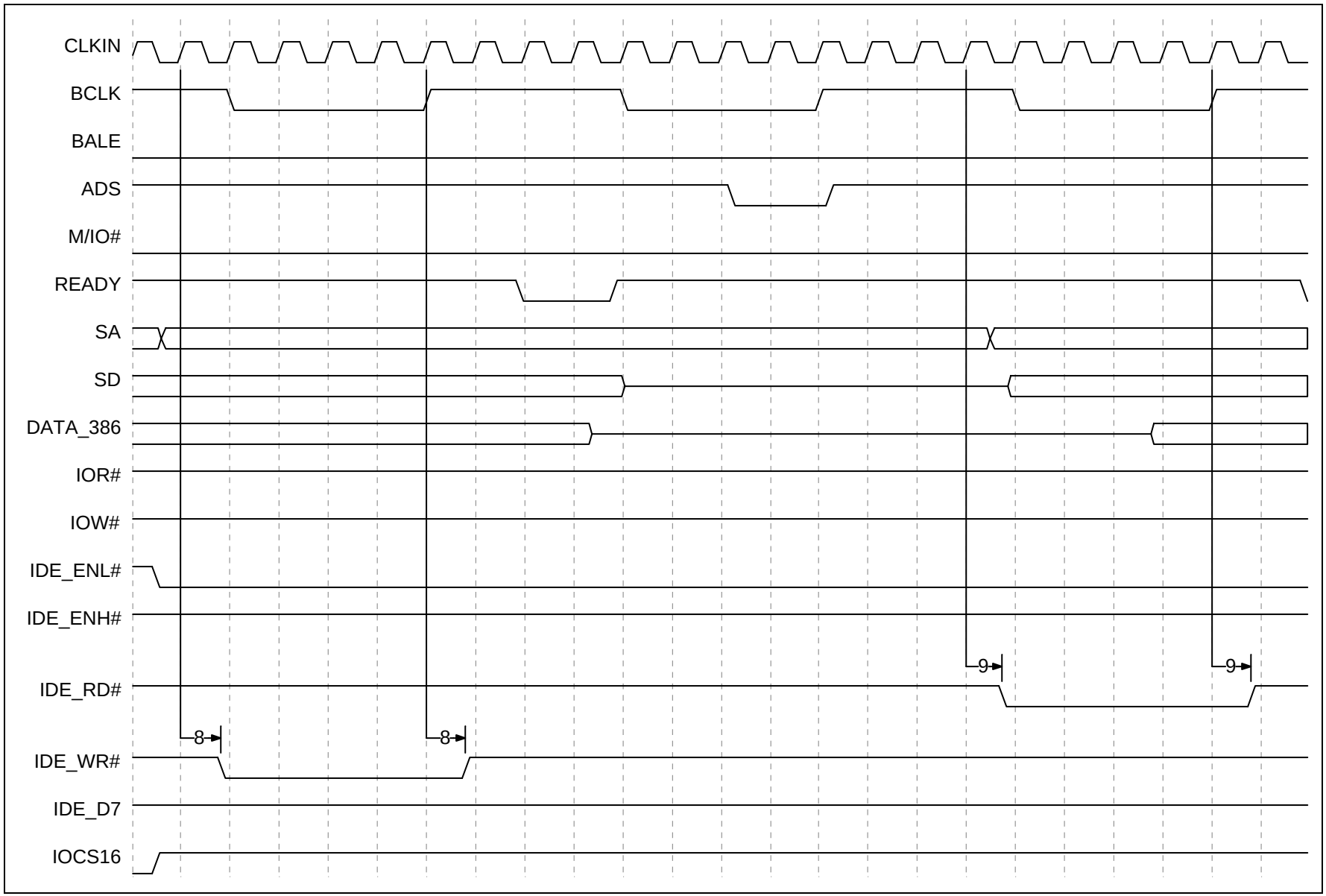


Figure A5- A: R380EX 16 Bit ISA Memory Device Access Overview

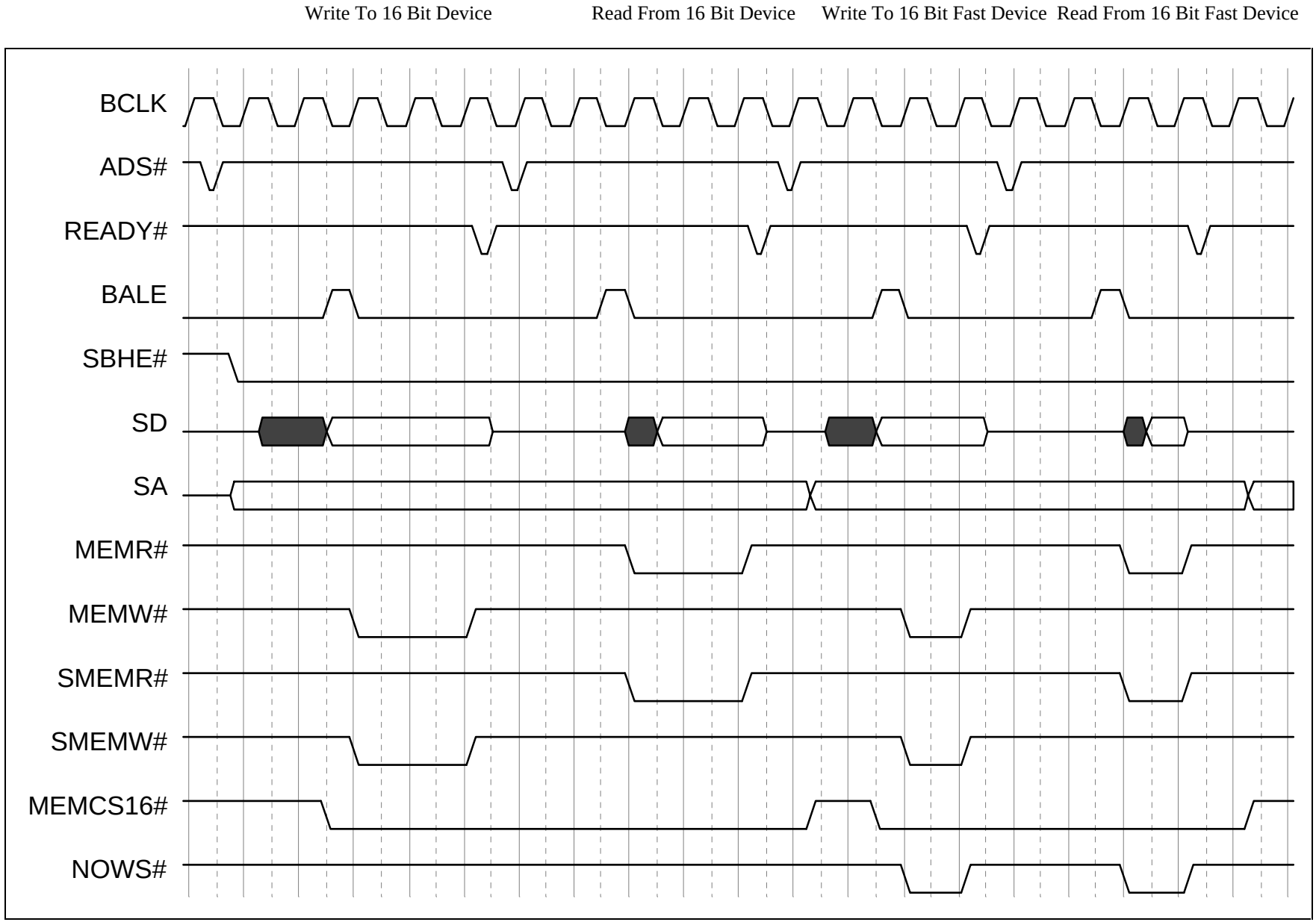


Figure A5- B:R380EX 8 Bit ISA Memory Device Access Overview

Word Write To 8 Bit Device Word Read From 8 Bit Device Fast Write And Read To 8 Bit Device

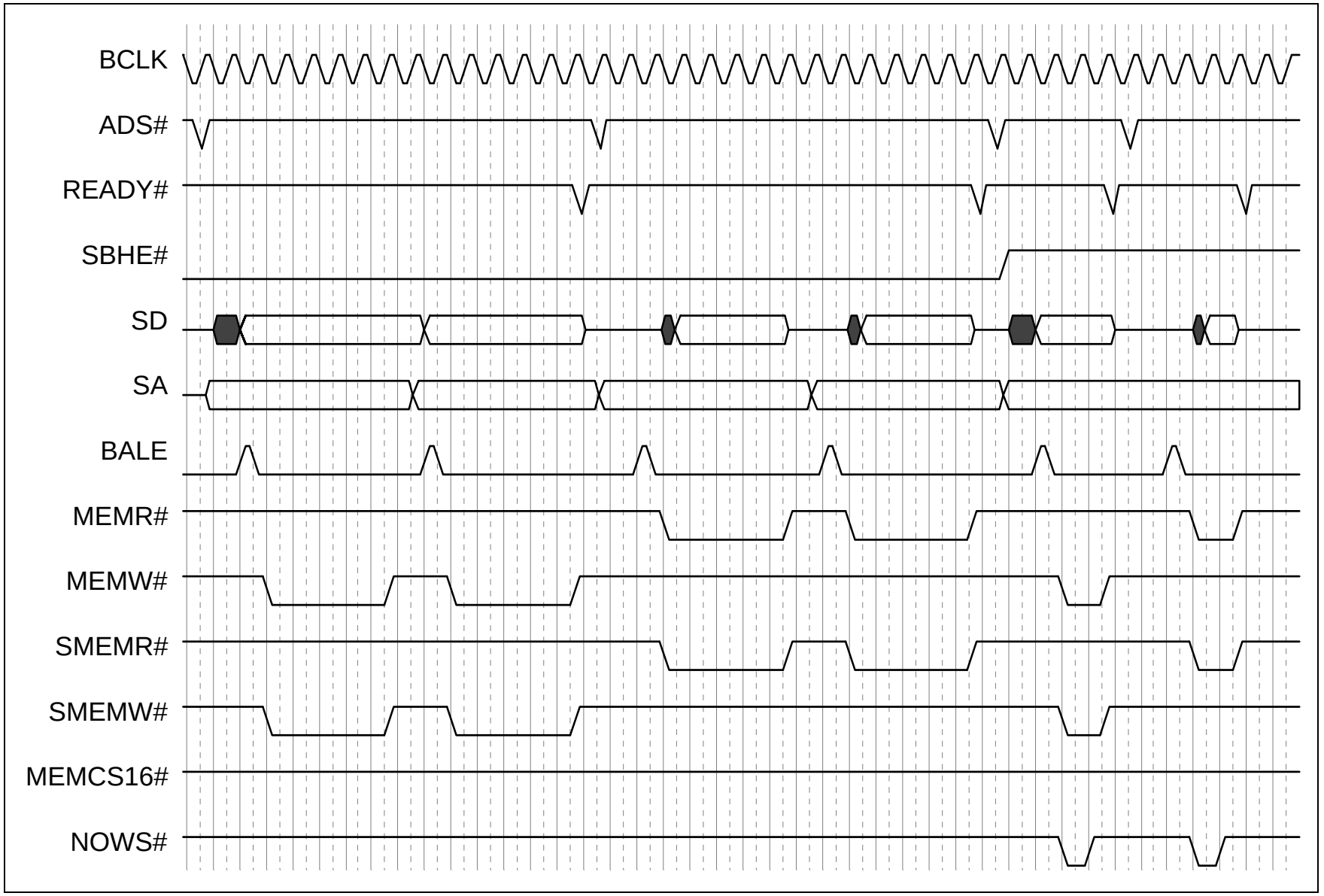


Table A5-CD: R380EX - ISA Memory Access Timing

Symbols	Parameters	Min	Max	Description
1	tCO_CLK2:MEMW#	2	11	Prop delay fromCLK2IN high to MEMW# Valid
2	tCO_CLK2IN:MEMR#	2	11	Prop delay from CLK2IN high to MEMR# Valid
3	tSU_BCLK:MEMCS16	0	-	Setup time rising edge of BALE
4	tHOLD_MEMCS16:BCLK	*****End of Cycle	-	Hold time falling edge of BCLK to end of the cycle
5	tSU_BCLK:NOWS	0	-	Setup time falling edge of BALE
6	tHOLD_NOWS:BCLK	1 BCLK	-	Hold time next rising edge of BCLK
7	tSU_BCLK:NOWS (8 bit only)	0		Setup time falling edge of BCLK after BALE low
8	tHOLD_NOWS:BCLK (8 bit only)	1 BCLK	-	Hold time next rising edge of BCLK
Note 1	tPD_IOCHRDY:{MEMRH,MEMWH}	1 BCLK	-	Setup time assertion of IOCHRDY to deassertion of MEMR or MEMW
Note 2	tCO_CLK2IN:SBHEL	2	14	Prop delay from CLK2IN high to SBHE# low

Note

1. Same as parameter 3 in Table A6-CD R380EX ISA I/O Access Timing
2. Same as parameter 4 in Table A6-CD R380EX ISA I/O Access Timing
3. High order SA/LA address lines are stable until the end of the ISA bus cycle
4. IOCHRDY is sampled at the rising edge of the BCLK

Figure A5- C: R380EX ISA Memory Access Timing Waveforms

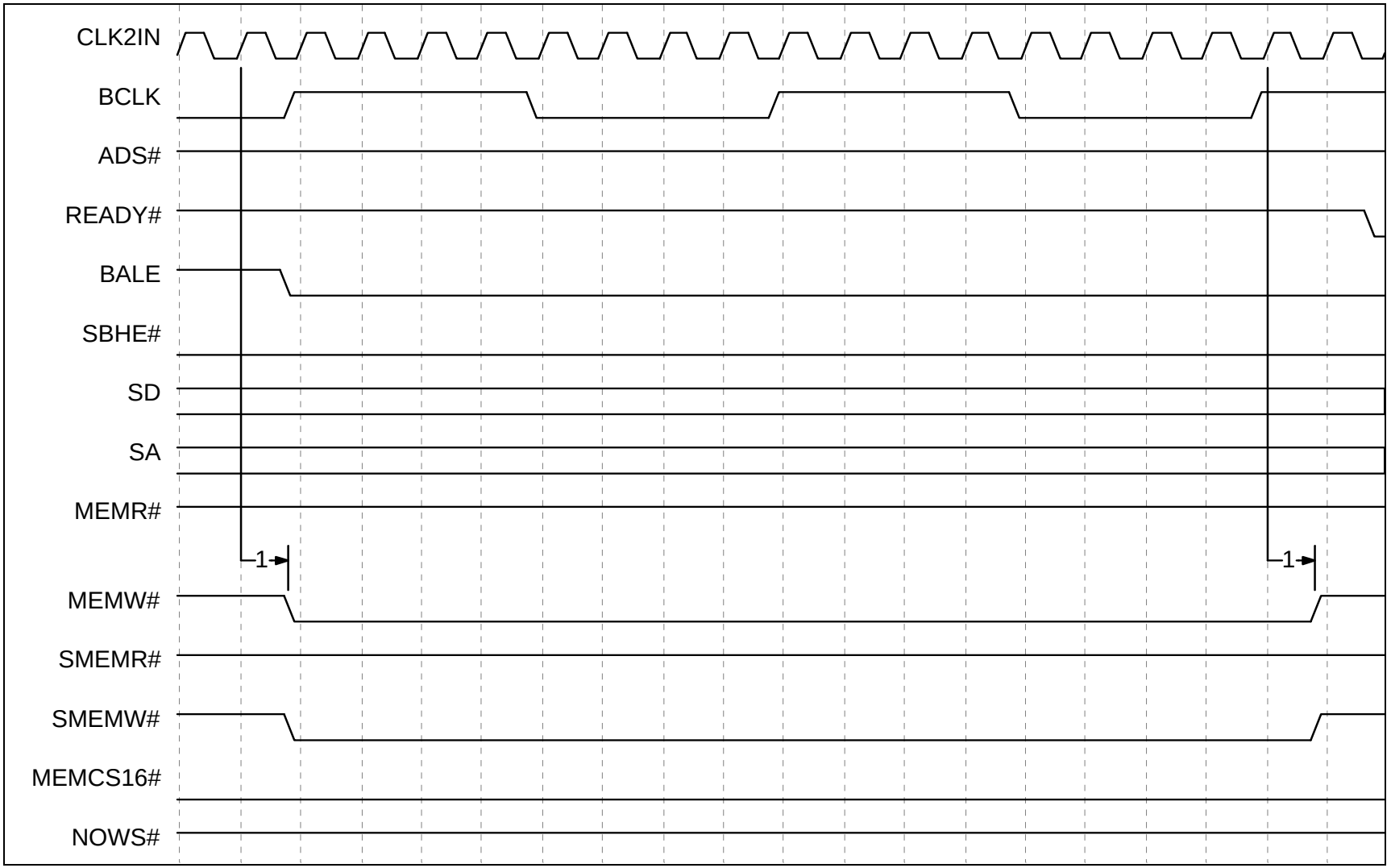


Figure A5- D: R380EX ISA Memory Access Timing Waveforms

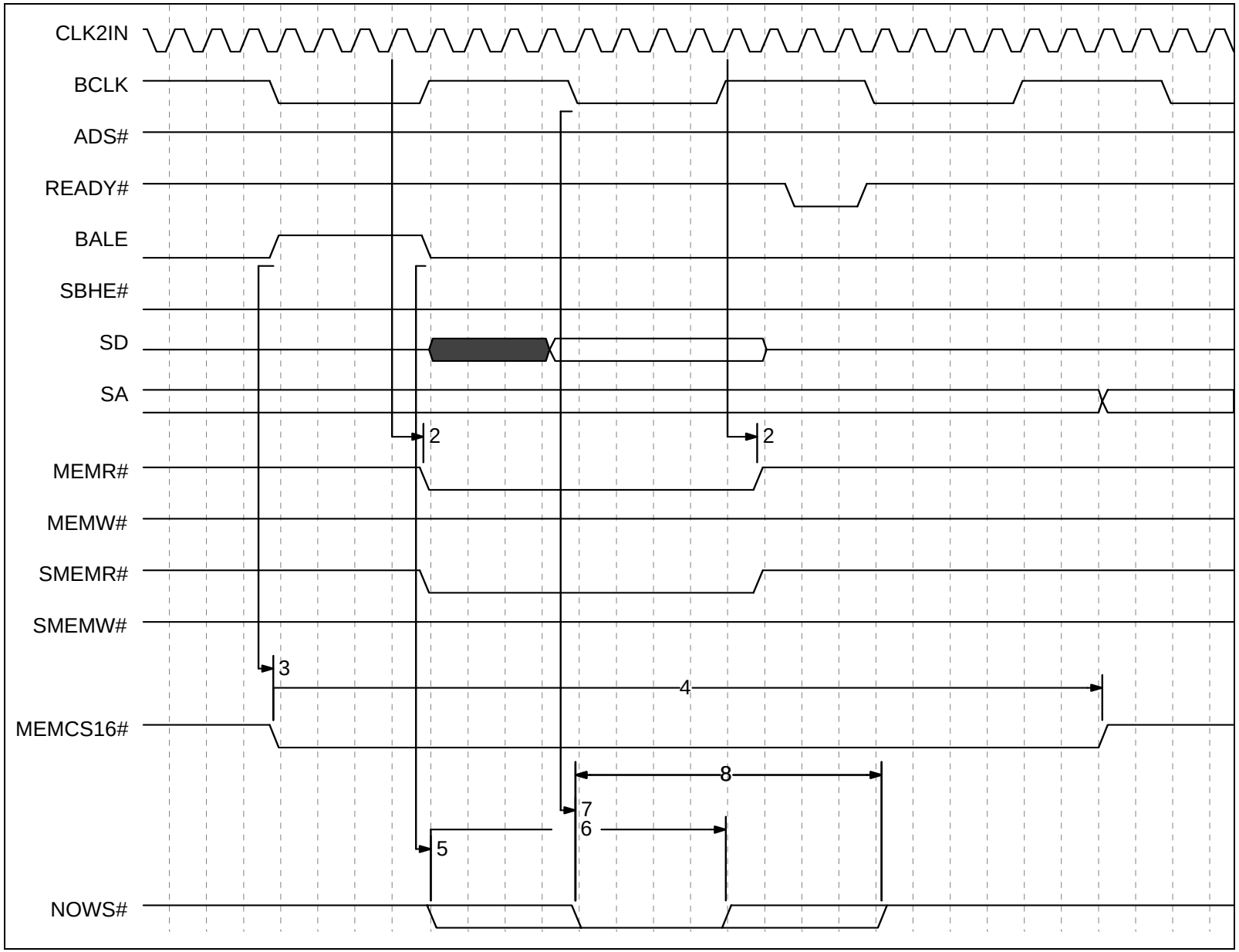


Figure A6- A: R380EX 16 Bit ISA I/O Device Access Overview

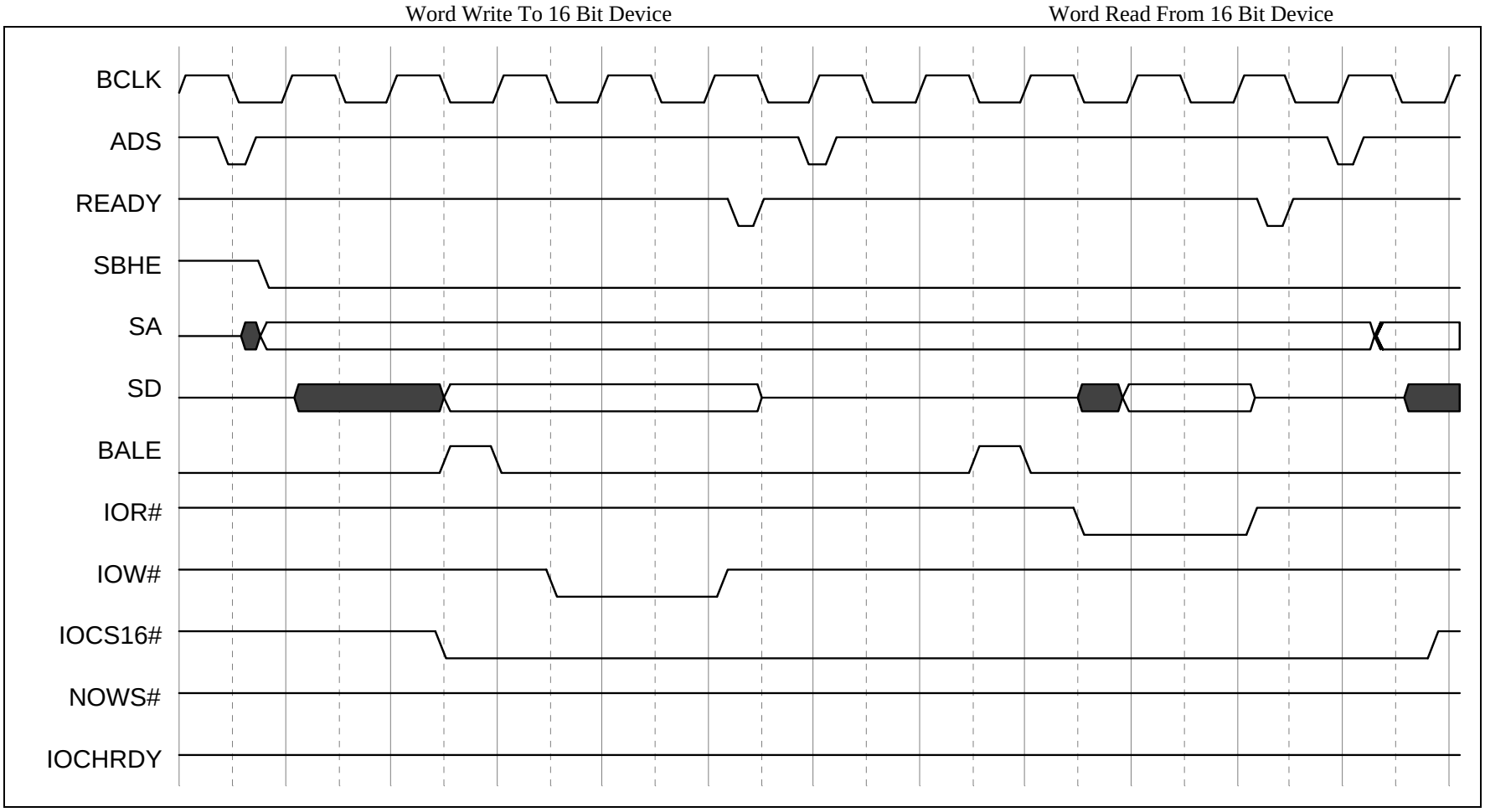


Figure A6- B:R380EX 8 Bit ISA I/O Device Access Overview

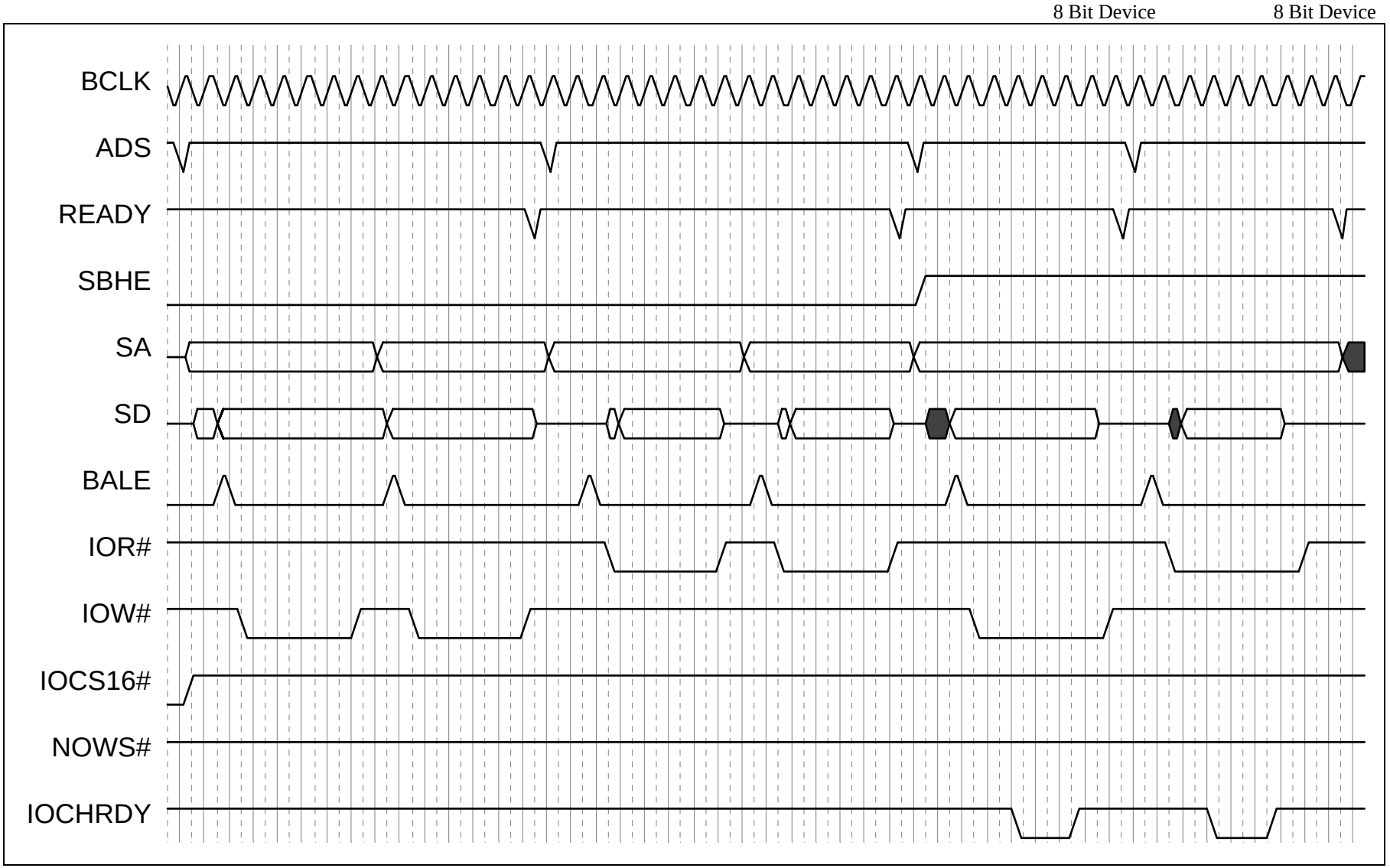


Table A6-CD: R380EX - ISA I/O Access Timing

Symbol	Parameter	Min	Max	Description
1	tCO_CLK2:IOR	2	11	Prop delay fromCLK2IN high to IOR# Valid
2	tCO_CLK2:IOW	2	11	Prop delay fromCLK2IN high to IOW# Valid
3	tPD_IOCHRDY: {IORH,IORH}	1 BCLK	-	Prop delay from assertion of IOCHRDY to deassertion of {IOR,IOW}
4	tCO_CLK2IN:SBHEL	2	14	Prop delay from CLK2IN high to SBHE low
5	tCO_CLK2IN:BALE	2	11	Prop delay from CLK2IN high to BALE# Valid
6	tSU_IOCS16:BCLK	0		Setup time rising edge edge of BALE
7	tHOLD_IOCS16:BCLK	*End of cycle	-	Hold time next falling edge of BCLK to end of the cycle
Note 2	tSU_NOWS:BCLK	0	-	Setup time next falling edge of BCLK after BALE low
Note 2	tHOLD_NOWS:BCLK	1 BCLK	-	Hold time next falling edge of BCLK after BALE low to rising edge of BCLK

Note:

1. The original IBM PC, XT and AT platform did not support 16 bit ISA no-wait state cycle
2. Same as parameter 7 and 8 in Table A5-C R380EX ISA Memoey Access Timing
3. High order SA/LA address lines are stable until the end of the ISA bus cycle

Figure A6- C: R380EX ISA I/O Access Timing Waveforms

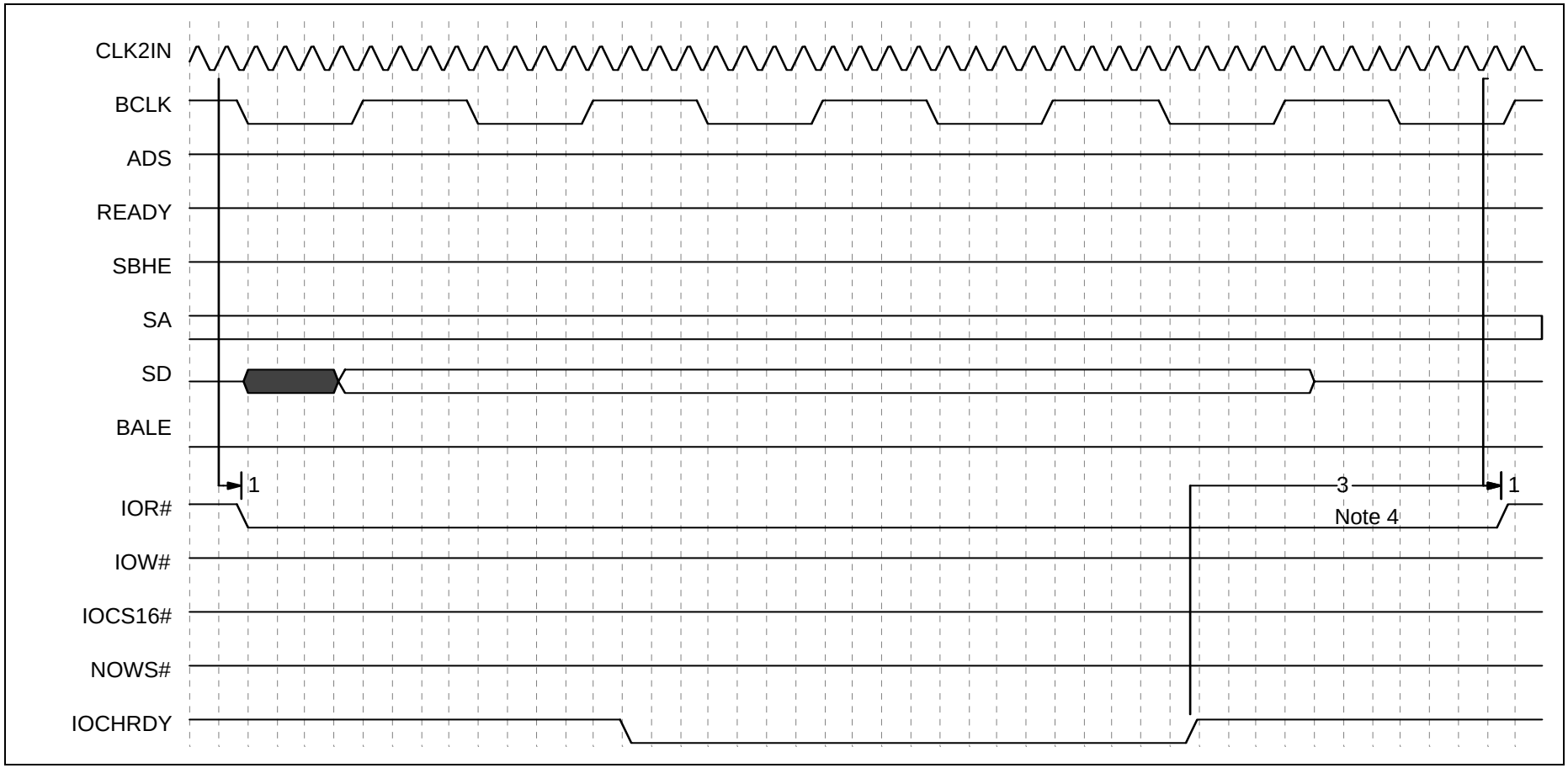


Figure A6- D: R380EX ISA I/O Access Timing Waveforms

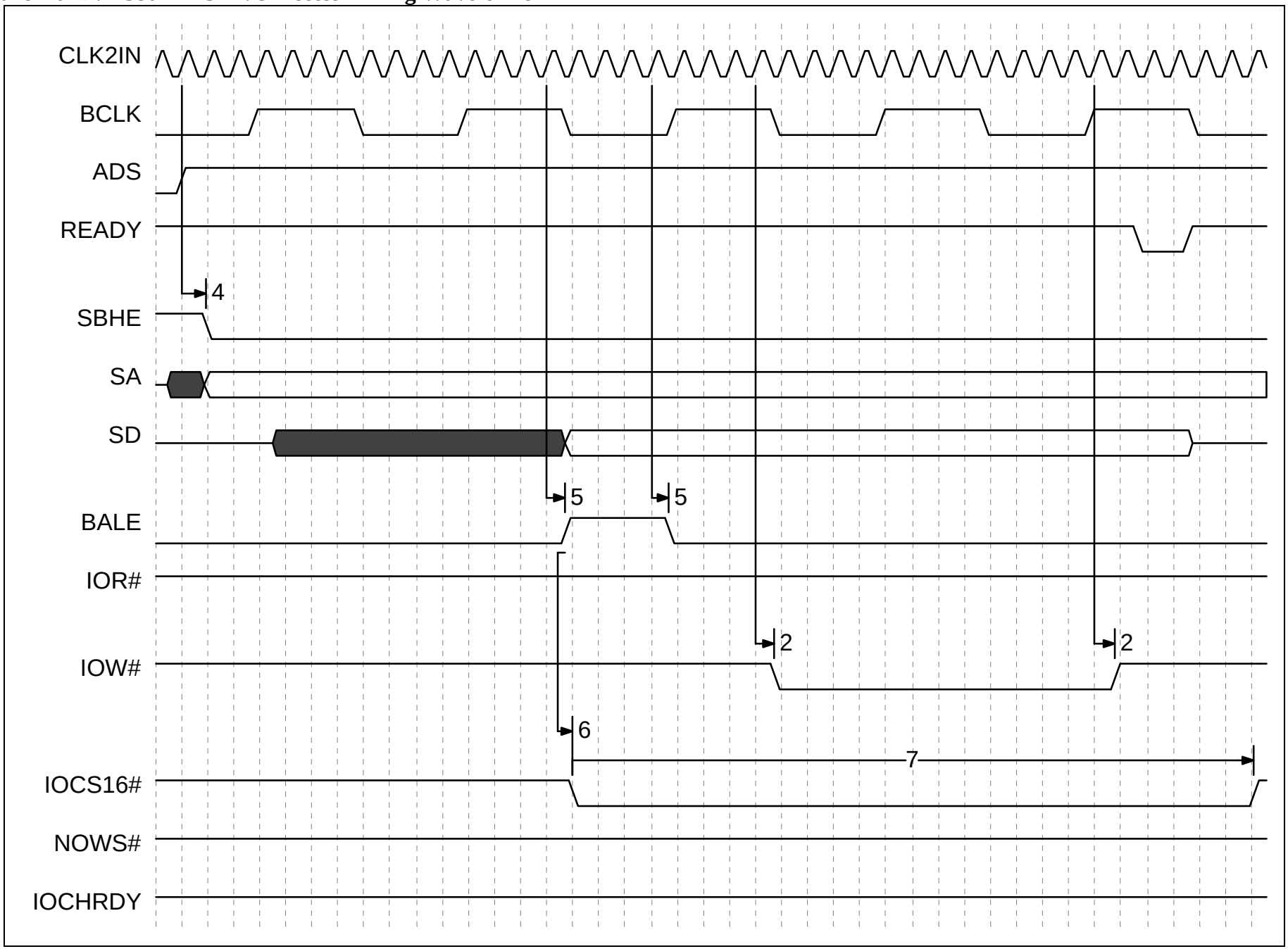


Figure A7-A: R380EX DMA ISA-ISA Memory Read To I/O Write Timing Waveforms

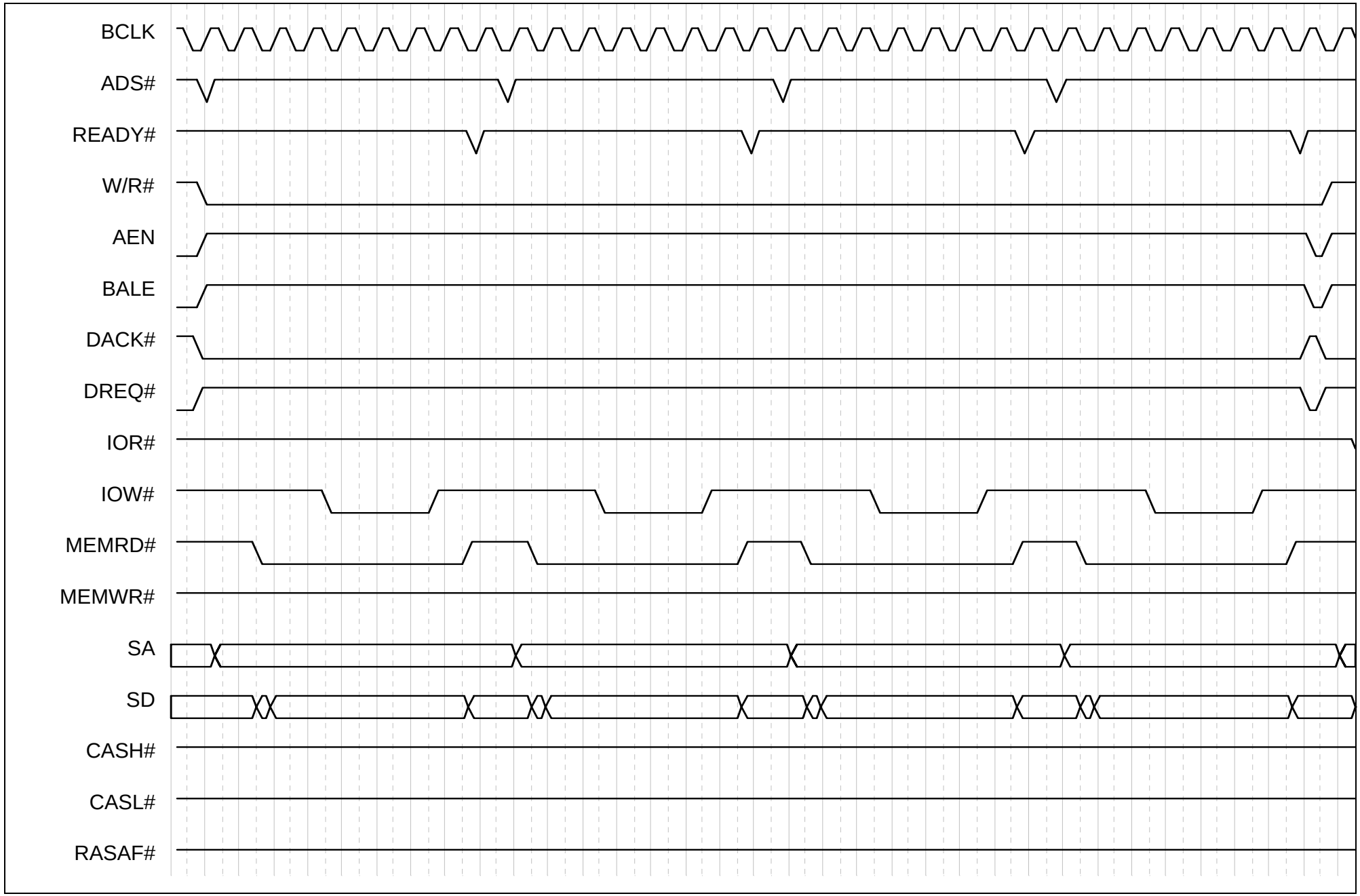


Figure A7-B: R380EX DMA ISA-ISA I/O Read To Memory Write Timing Waveforms

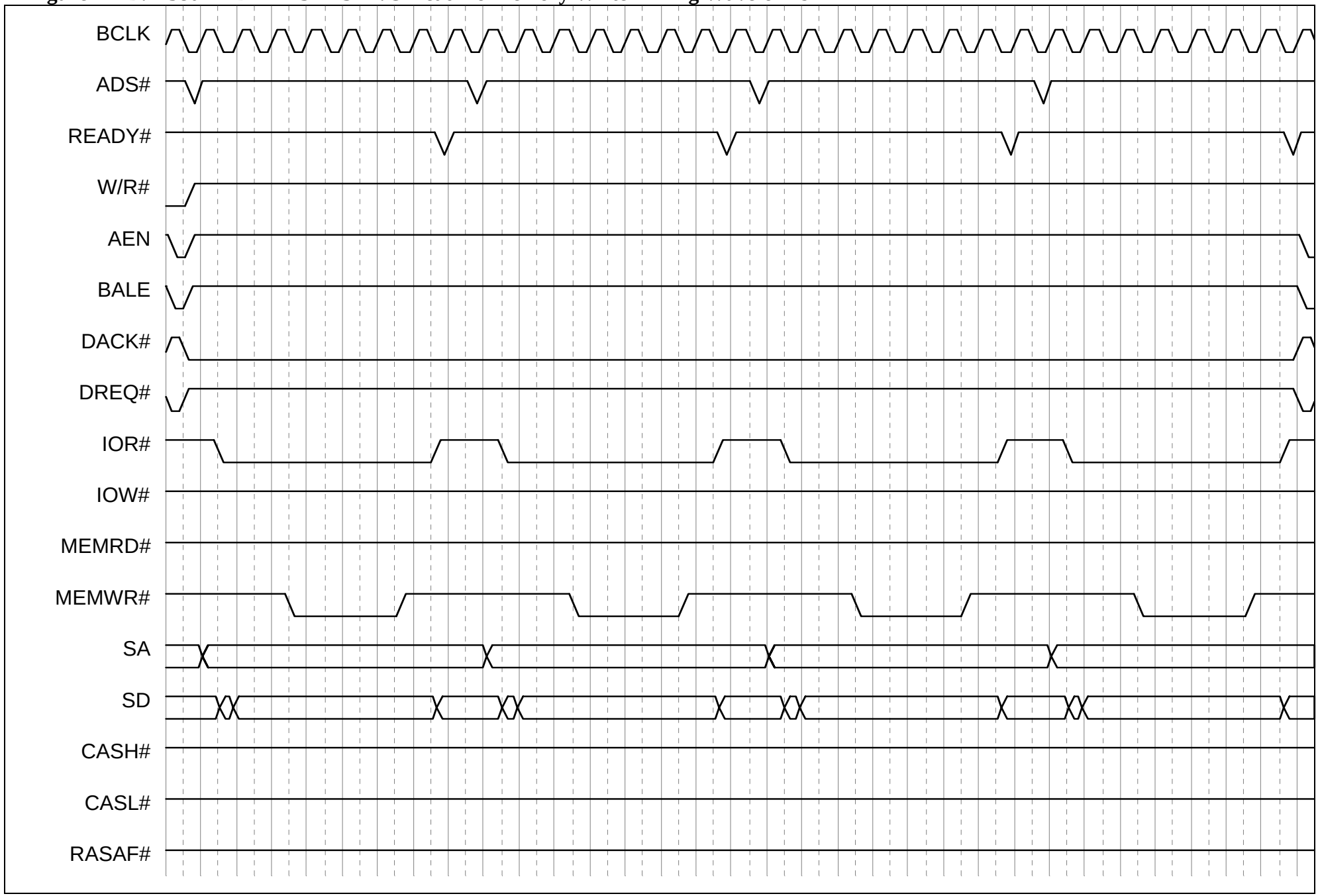


Figure A7-C: R380EX DMA DRAM Read to ISA I/O Write Timing Waveforms

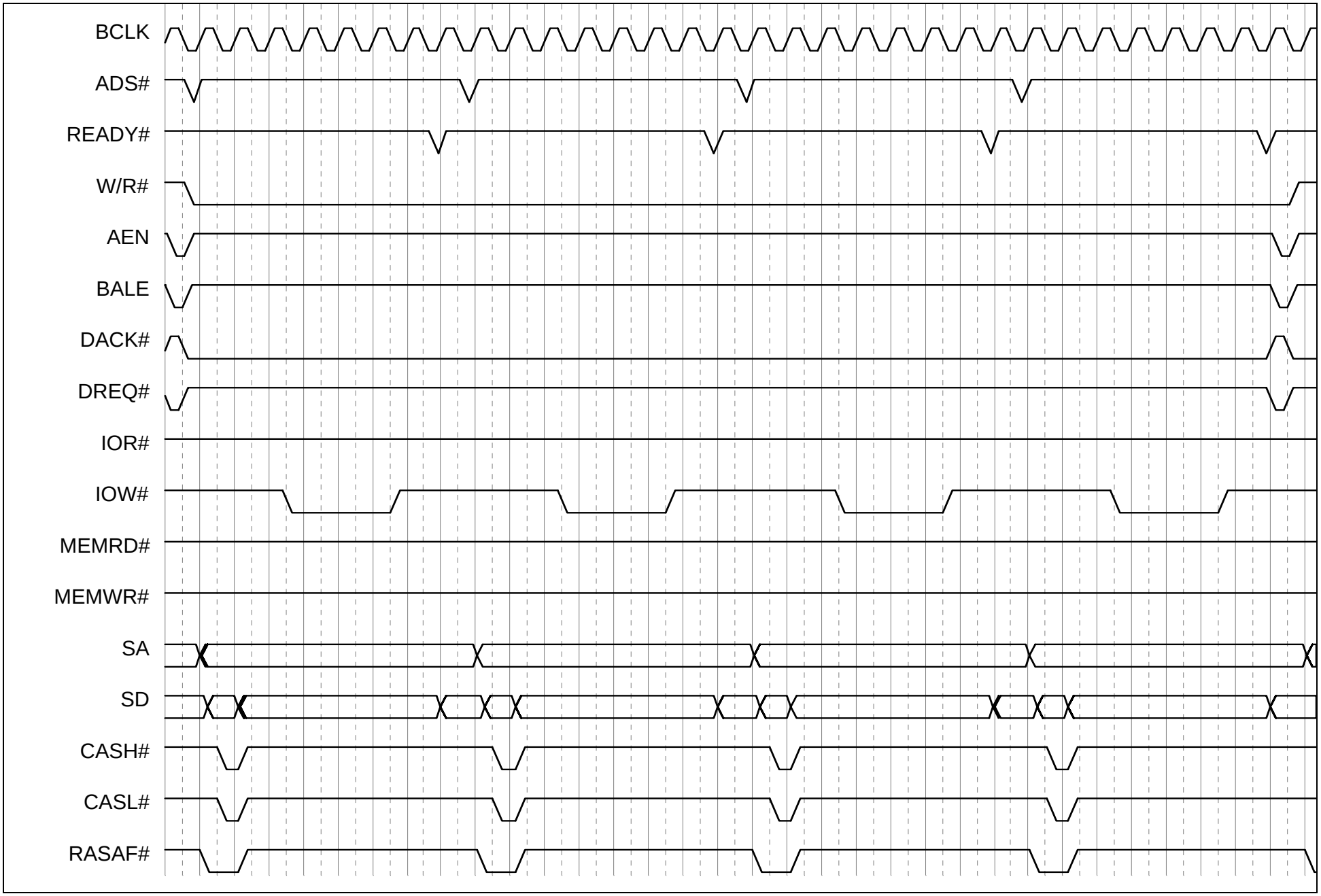


Figure A7-D: R380EX DMA ISA I/O Read to DRAM WriteTiming Waveforms

Figure A8-A: R380EX User Chip Select Timing Waveforms

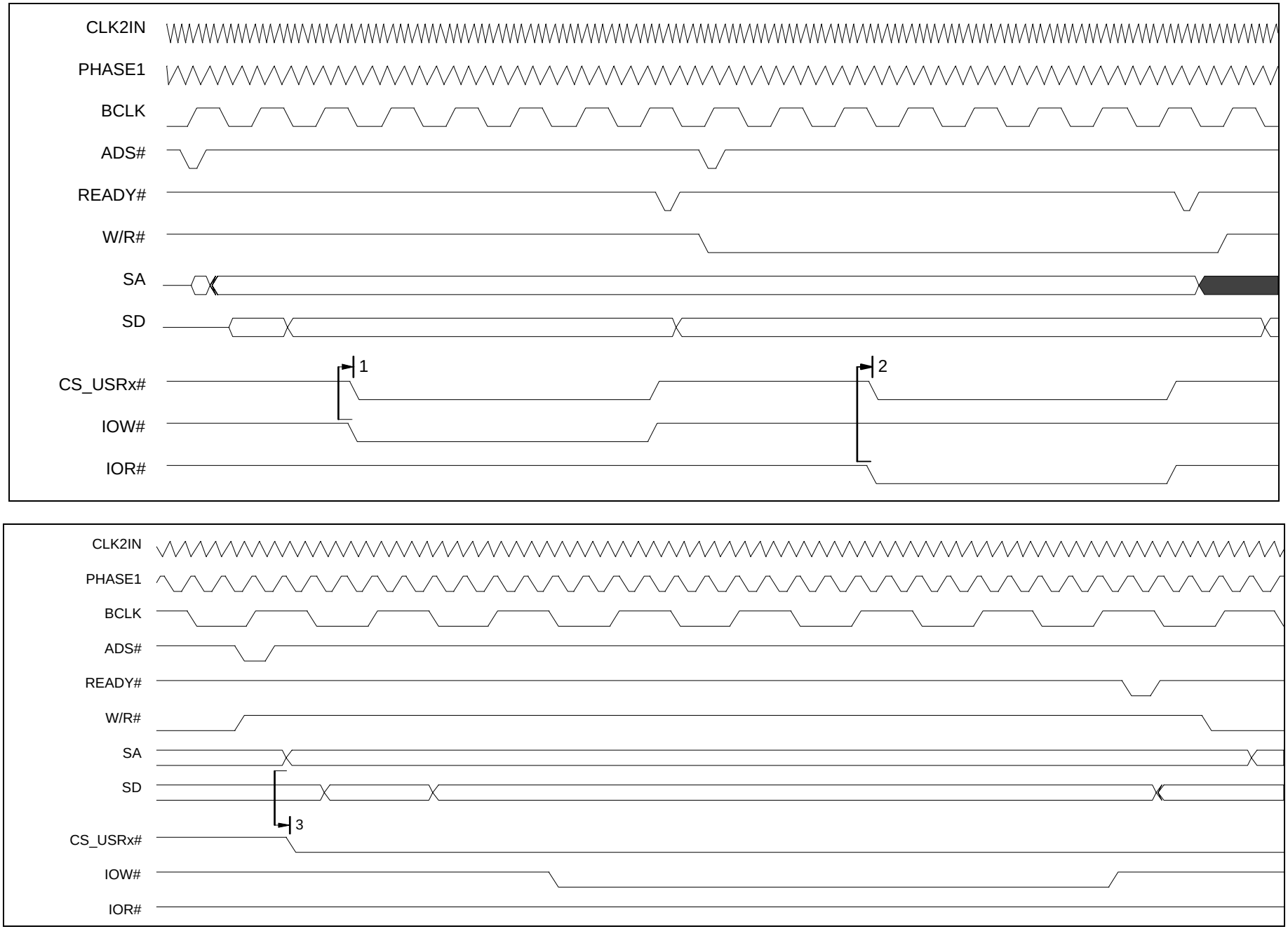


Table A8-A: R380EX - User Chip Select Access Timing

Symbol	Parameter	Min	Max	Description
1	tPD_IOR#:CS_USR[3:0]#	-1	1	Prop delay from IOR# to CS_USR[3:0]#
2	tPD_IOW# CS_USR[3:0]#	-1	1	Prop delay from IOW# to CS_USR[3:0]#
3	tPD_SA:CS_USR[3:0]#	-1	1	Prop delay from SA address valid to CS_USR[3:0]#

Figure A9: R380EX Local Bus Device Access Waveform

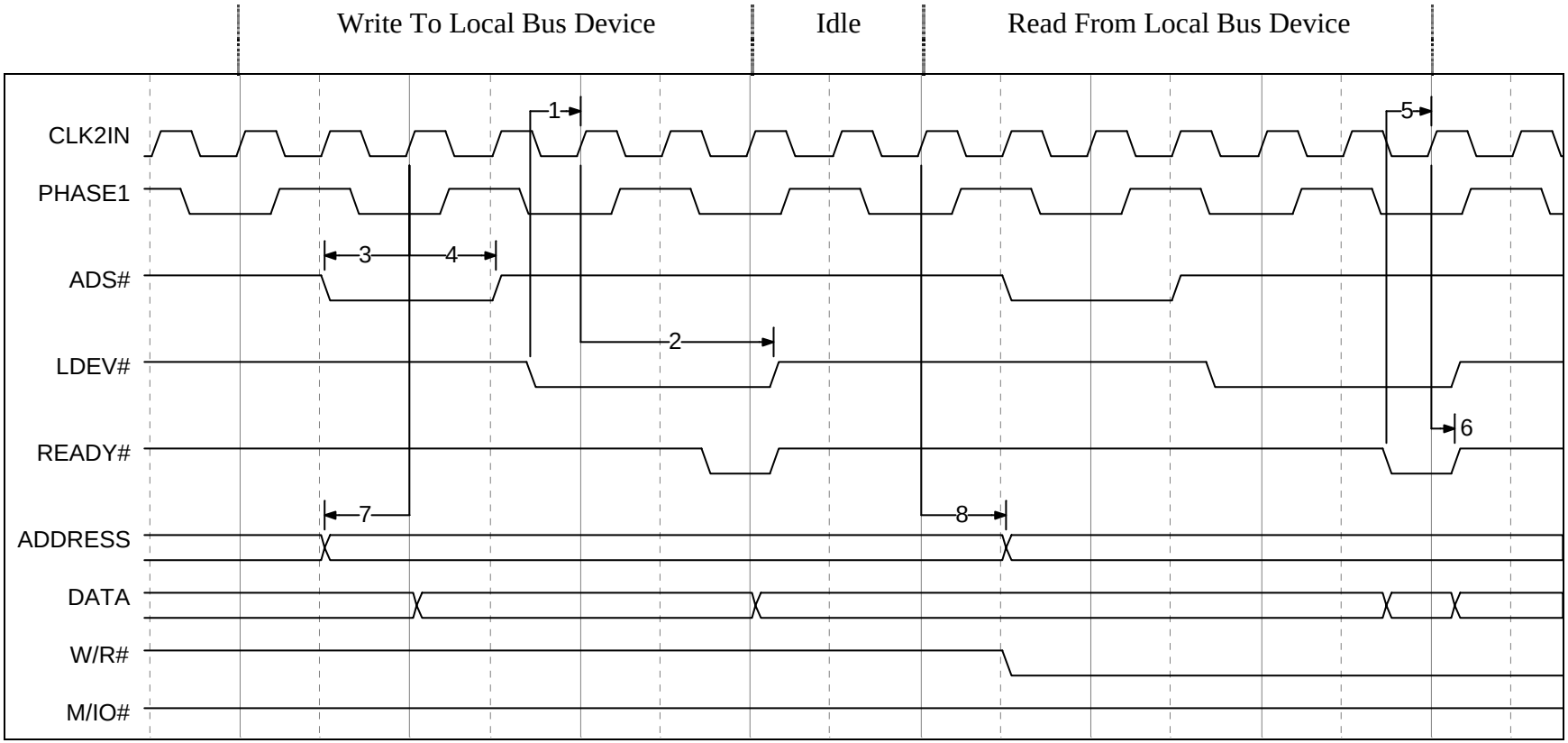


Table A9: R380EX - Local Bus Device Access Table

Symbol	Parameter	Min	Max	Description
1	tSU_LDEV:CLK2IN	8	-	Setup time LDEV# low to CLK2IN high
2	tHOLD_LDEV:CLK2IN	end of the cycle	-	Hold time CLK2IN high to end of the cycle
3	tSU_ADS:CLK2IN	6ns		Setup time ADS# low to CLK2IN high
4	tHOLD_ADS:CLK2IN	2ns		Hold time CLK2IN to ADS# high
5	tSU_READY:CLK2IN	5ns		Setup time READY# low to CLK2IN high
6	tHOLD_READY:CLK2IN	2ns		Hold time CLK2IN to READY# high
7	tSU_ADDRESS:CLK2IN	6ns		Setup time ADDRESS valid to CLK2IN high
8	tHOLD_ADDRESS:CLK2IN	2ns		Hold time CLK2IN to ADDRESS invalid

Figure A10-A: R380EX EDO DRAM Access Waveforms

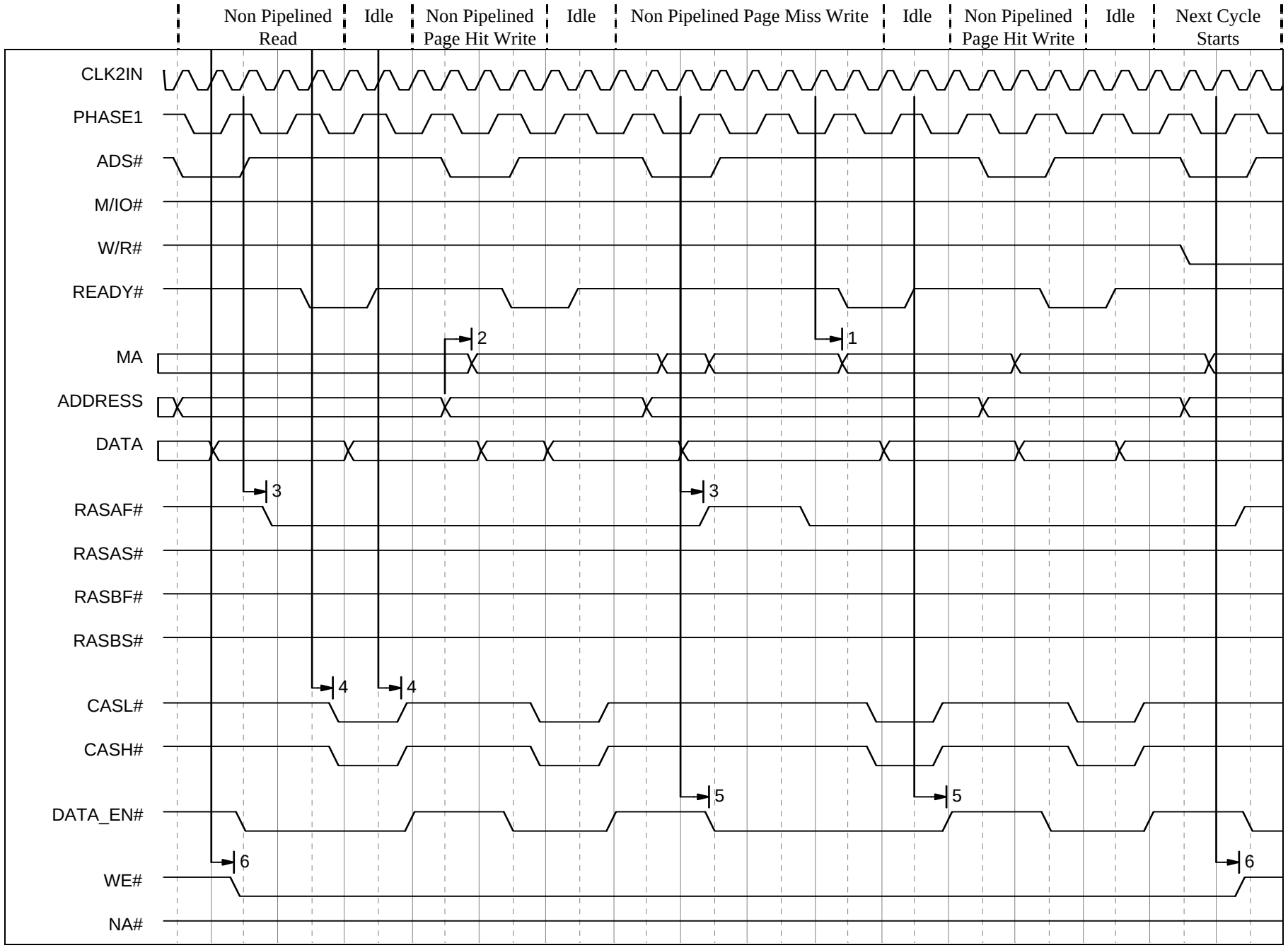

Non Pipelined Read
Idle
Non Pipelined Page Hit Write
Idle
Non Pipelined Page Miss Write
Idle
Non Pipelined Page Hit Write
Idle
Next Cycle Starts
CLK2IN
PHASE1
ADS#
M/IO#
W/R#
READY#
MA
ADDRESS
DATA
RASAF#
RASAS#
RASBF#
RASBS#
CASL#
CASH#
DATA_EN#
WE#
NA#
1
2
3
3
4
4
5
5
6
6

Table A10-AB: R380EX - EDO and FPM Dram Access Timing

Symbols	Parameters	Min	Max	Description
1	tCO_CLK2IN:MA	2	14	Prop delay form CLK2IN high to MA valid
2	tCO_ADDRESS MA	2	12	Prop delay from 386 address valid to MA valid
3	tCO_CLK2IN:RASAF# L	2	10	Prop delay from CLK2IN high to RASAF# low
4	tCO_CLK2IN:CASL# L	2	10	Prop delay from CLK2IN high to CASL# low
5	tCO CLK2IN:DATA_EN# L	2	11	Prop delay from CLK2IN high to DATA_EN# low
6	tCO CLK2IN:WE# L	2	10	Prop delay from CLK2IN high to WE# low
7	tCO CLK2IN:READY# L	2	11	Prop delay from CLK2IN high to READY# low
8	tCO CLK2IN NA# L	2	10	Prop delay from CLK2IN high to NA# low

NOTE:

1. DRAM timing register is set to fastest timing mode
2. Test Condition C_L= 50 pF for all signals except NA# with C_L= 25 pF
3. Temparature at 85C
4. Voltage at 4.5V

Figure A10-B: R380EX EDO Dram Access Waveforms

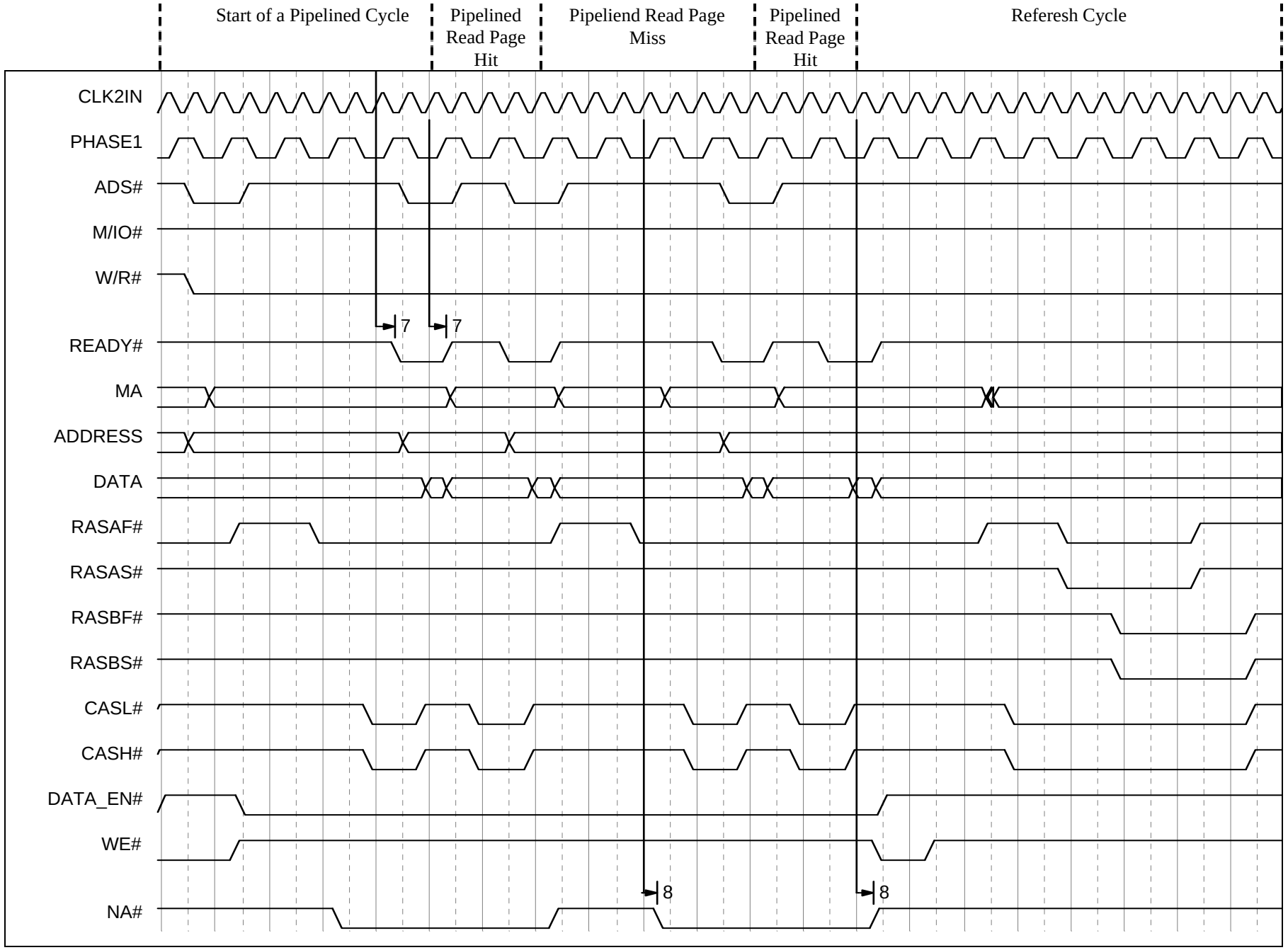


Figure A11-A: R380EX FPM Dram Access Waveforms

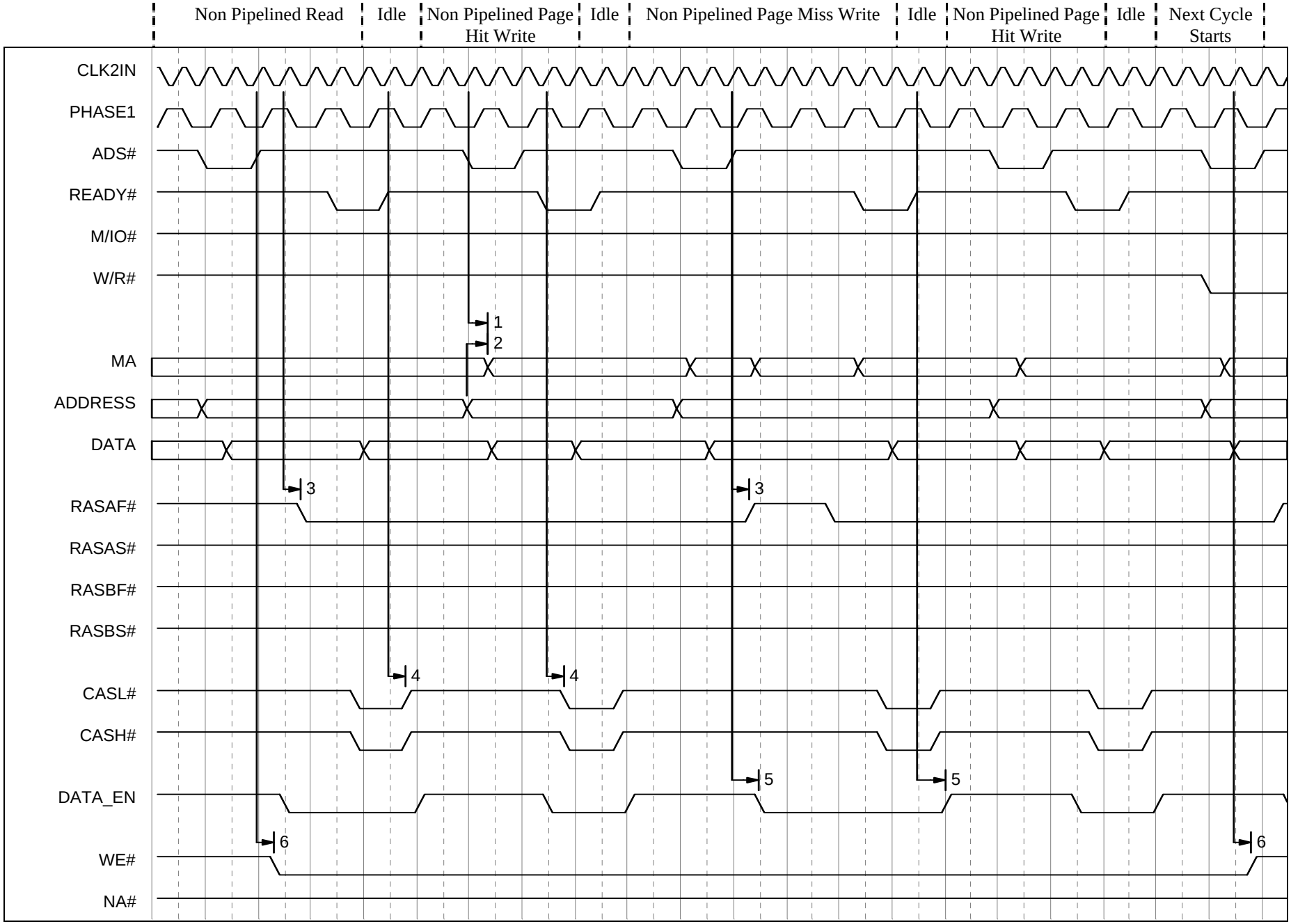


Figure A11-B: FPM Dram Access Waveforms

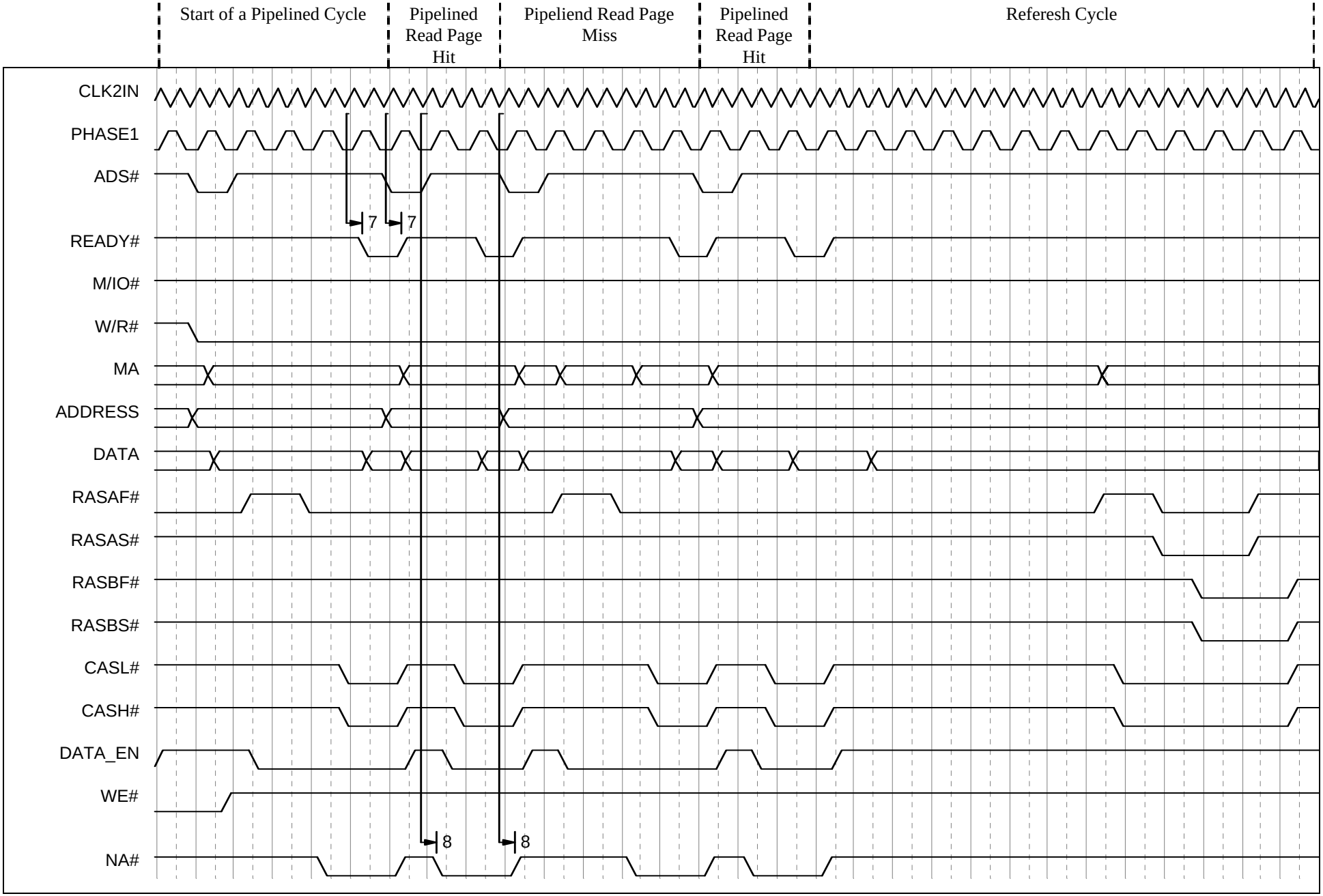
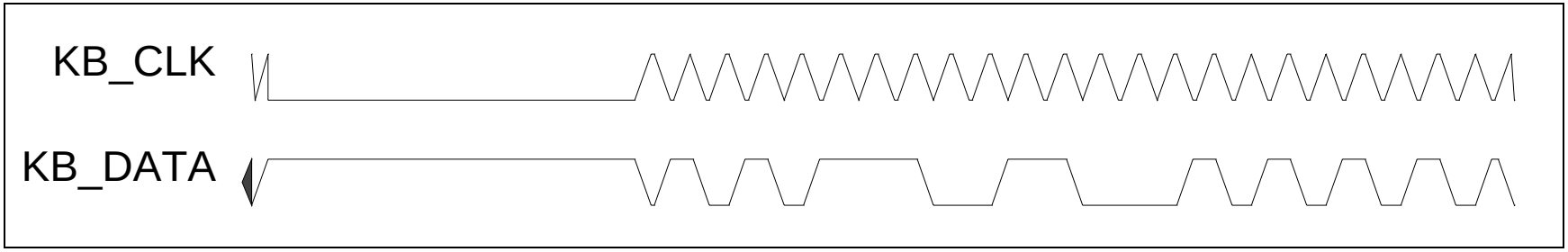
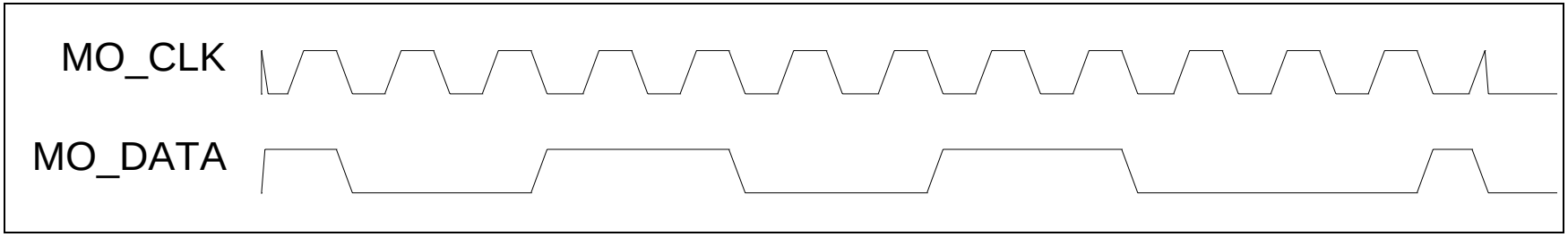


Figure A12-A: R380EX Key Board and Mouse Controller Timing Waveforms



Mouse Sending Data to the Controller



5445 NE Dawson Creek Drive

Hillsboro, OR 97124

Toll Free: (800) 950-0044

Phone: (503) 615-1100

Fax: (503) 615-1150

Internet Email: info@radisys.com

World-Wide Web: http://www/radisys.com

Worldwide Offices:

Japan: +81 3 5496 7891

United Kingdom: +44 1793 411200

The Netherlands: +31 40 2653 644

Germany: +49 811 95817

07-0348-01 2M K2