80296SA MICROCONTROLLER SPECIFICATION UPDATE

Release Date: February, 1997
Order Number: 272908-006

The 80296SA may contain design defects or errors known as errata. Characterized errata that may cause the 80296SA's behavior to deviate from published specifications are documented in this specification update.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation
P.O. Box 7641
Mt. Prospect IL 60056-7641

or call in North America 1-800-879-4683, Europe 44-0-1793-431-155,
France 44-0-1793-421-777, Germany 44-0-1793-421-333, other countries 708-296-9333

CONTENTS

REVISION HISTORY

Rev. Date	Version	Description
07/01/96	001	This is the new Specification Update document. It contains all identified errata published prior to this date. Release for 80296SA A0 including spec changes, spec clarifications, and documentation changes.
08/29/96	002	Added errata 9600016 and 9600017, specification change 002, and documentation changes 003 and 004.
10/21/96	003	Deleted marking information for SQFP package. Added documentation changes 005 through 011.
12/11/96	004	Added specification changes 003 and 004, specification clarification 002, and documentation changes 012 through 022. Corrected documentation change 011.
1/09/97	005	Clarified erratum 9600015.
2/05/97	006	Added specification clarification 004 and documentation change 024.

PREFACE

As of July, 1996, Intel's Semiconductor Products Group has consolidated available historical device and documentation errata into this new document type called the Specification Update. We have endeavored to include all documented errata in the consolidation process, however, we make no representations or warranties concerning the completeness of the Specification Update.

This document is an update to the specifications contained in the Affected Documents/Related Documents table below. This document is a compilation of device and documentation errata, specification clarifications and changes. It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools.

Information types defined in Nomenclature are consolidated into the specification update and are no longer published in other documents.

This document may also contain information that was not previously published.

Affected Documents/Related Documents

Title	Order
80296SA Commercial CHMOS 16-bit Microcontroller datasheet	272748-002
80296SA Microcontroller User's Manual	272803-001

Nomenclature

Errata are design defects or errors. These may cause the published (component, board, system) behavior to deviate from published specifications. Hardware and software designed to be used with any component, board, and system must consider all errata documented.

Specification Changes are modifications to the current published specifications. These changes will be incorporated in any new release of the specification.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in any new release of the specification.

Documentation Changes include typos, errors, or omissions from the current published specifications. These will be incorporated in any new release of the specification.

NOTE:

Errata remain in the specification update throughout the product's lifecycle, or until a particular stepping is no longer commercially available. Under these circumstances, errata removed from the specification update are archived and available upon request. Specification changes, specification clarifications and documentation changes are removed from the specification update when the appropriate changes are made to the appropriate product specification or user documentation (datasheets, manuals, etc.).

SUMMARY TABLE OF CHANGES

The following table indicates the errata, specification changes, specification clarifications, or documentation changes which apply to the 80296SA product. Intel may fix some of the errata in a future stepping of the component, and account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

Codes Used in Summary Table

Stepping

X:	Errata exists in the stepping indicated. Specification Change or Clarification that applies to this stepping.
(No mark) or (Blank box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

Page

(Page):	Page location of item in this document.

Status

Doc:	Document change or update will be implemented.
Fix:	This erratum is intended to be fixed in a future step of the component.
Fixed:	This erratum has been previously fixed.
NoFix:	There are no plans to fix this erratum.
Eval:	Plans to fix this erratum are under evaluation.

Row

▌	Change bar to left of table row indicates this erratum is either new or modified from the previous version of the document.

Errata

No.	Steppings			Page	Status	Errata
	A0					
9600001	X			9	NoFix	Register File Boundary and Rotate Instructions
9600002	X			9	NoFix	Indirect Addressing with CPU SFRs
9600003	X			10	Fix	Address, Addr/Data, RD#, and BHE# May Remain Active During Idle, Standby, and Powerdown Modes
9600004	X			11	Fix	Address, Data, EPORT Pins Weakly Held High During Hold
9600005	X			11	Fix	Address, Data Pins Driven Early Before Controller Enters Hold
9600006	X			13	Fix	Address, Data Pins Driven Early Before Controller Exits Hold
9600007	X			14	Fix	Generation of Timer Overflow
9600008	X			15	Fix	False NMI Interrupt Request Generated Upon Exiting Reset
9600009	X			15	Fix	When TAG1-Ing a Protected Instruction, the Controller Deadlocks [ICE-specific Erratum Only]
9600010	X			16	Fix	Double Data Read With Hold
9600011	X			16	Fix	Interruptible BMOV Repeat Counter
9600012	X			17	Fix	8-Bit Bus ICX Add With Carry
9600013	X			17	Fix	Premature Exit of Hold When POPA Is Used
9600014	X			17	Fix	Data Hold Timing For CCB1
9600015	X			18	Fix	Edge/Level Sensitive External Interrupt Exit from Powerdown **and Standby**
9600016	X			18	Eval	RETI Instruction in Programmable Priority Mode May Clear IN_PROG Bit
9600017	X			19	Eval	External Code Fetch During Read-Modify-Write Instruction Overwrites Interrupt Pending Register

Specification Changes

No.	Steppings			Page	Status	Specification Changes
	A0					
001	X			20	Doc	Index Register ICX1 Usage With ST Instruction
002	X			21	Eval	V_{CC} Specification for 50 MHz Operation
003	X			21	Doc	T_{RXDX} Specification Change to 0 ns (from 2 ns)
004	X			21	Doc	T_{CHYX} Specification Replaced by T_{CH1YX} and T_{CH2YX}, including new READY timing diagrams

Specification Clarifications

No.	Steppings			Page	Status	Specification Clarifications
	A0					
001	X			22	Doc	Use of RPD to Exit Powerdown
002	X			22	Eval	P2.3/BREQ# is automatically configured as BREQ# during hold.
003	X			22	Doc	Interrupt priority scheme clarification.
004	X			22	Doc	WRH#/BHE# automatically configured as BHE# during ICE mode.

Documentation Changes

No.	Document Revision	Page	Status	Documentation Changes
001	-001	22	Doc	Address, Data, EPORT Pins Weakly Held High During Hold (User's Manual)
002	-001	23	Doc	Index Register ICX1 Usage With ST Instruction (User's Manual)
003	-002	23	Doc	V_{CC} Specification for 50 MHz Operation (Datasheet)
004	-001	23	Doc	Read-modify-write and MAC Instructions Are Not Protected (User's Manual)
005	-001	23	Doc	AP-445, *8XC196KR Peripherals: A User's Point of View* is obsolete; no longer available. (User's Manual)

Documentation Changes (Continued)

No.	Document Revision	Page	Status	Documentation Changes
006	-001	23	Doc	AP-711, *EMI Design Techniques for Microcontrollers in Automotive Applications*, is order number 272637. (User's Manual)
007	-002	24	Doc	SQFP package is not available. (Datasheet)
008	-002	24	Doc	ROM option is not available. (Datasheet)
009	-002	24	Doc	Marking for 40 MHz is "40." (Datasheet)
010	-001	24	Doc	SQFP package is not available (User's Manual)
011	-001	24	Doc	DCD and BGD bit explanations (User's Manual)
012	-002	25	Doc	Change T_{RXDX} specification to 0 ns (Datasheet)
013	-001	26	Doc	Deferred bus cycle mode on page 2-17 (User's Manual)
014	-001	26	Doc	Port 2 register addresses on page 8-4 (User's Manual)
015	-001	26	Doc	EPA register addresses on page 10-3 (User's Manual)
016	-001	26	Doc	P3_PIN register address on page 12-3 (User's Manual)
017	-001	26	Doc	Example for selecting capacitor C_1 on page 12-12 (User's Manual)
018	-001	27	Doc	Remove Note 1 from Table A-1 on page A-3 and from Table A-7 on page A-63 (User's Manual)
019	-001	27	Doc	Correct note for MSAC instruction on page A-32, MVAC instruction on page A-36, and SHLL instruction on page A-45 (User's Manual)
020	-001	27	Doc	Correct description of SUBB instruction on page A-53 (User's Manual)
021	-001	28	Doc	Correct reset value of CON_REG0 register on pages C-2 and C-16 (User's Manual)
022	-002	25	Doc	Replace T_{CHYX} specification with $T_{CH1\ YX}$ and T_{CH2YX} for both multiplexed and demultiplexed modes; replace READY timing diagrams (Datasheet)
023	-002	31	Doc	Add Jump Penalty Table
024	-002	32	Doc	Correction to the deferred timing description.

IDENTIFICATION INFORMATION

Markings

QFP

S80296SA

A0 STEPPING

ERRATA

9600001. Register File Boundary and Rotate Instructions

PROBLEM: When using the rotate instructions, the 80296SA cannot rotate across the register file boundary into external memory. The rotate instructions (SMACR, MACR, SMACRZ, MACRZ) can operate within the register file and within the external memory region, but the rotation cannot occur across the boundary of the register file and external memory boundary. This is a design consideration and will not be changed on future steppings of the device.

IMPLICATION: Rotate instructions close to the upper boundary of the register file cannot be used.

WORKAROUND: When using the rotate instructions close to the upper boundary of the register file, the firmware must monitor when the next instruction will be a rotate across the register file boundary, and not allow the rotate to occur using the instructions given. Rather the firmware must handle moving the data to be rotated without using the rotate instructions.

STATUS: NoFix. Refer to Summary Table of Changes to determine the affected stepping(s).

9600002. Indirect Addressing with CPU SFRs

PROBLEM: CPU registers 19h or below cannot be accessed with indirect addressing. For example:

```
LD    30h, #12h
LD    20h, [30h]
```

will not load 12H (the INT_PEND1 register) into 20H since the INT_PEND1 register cannot be indirectly addressed. This is a design consideration and will not be changed on future steppings of the device.

IMPLICATION: Registers 19h or below should not be indirectly addressed.

WORKAROUND: Ensure that all CPU SFRs (00-19h) are directly addressed.

Example:

```
LD 20h, 12h
```

STATUS: NoFix. Refer to Summary Table of Changes to determine the affected stepping(s).

9600003. Address, Addr/Data, RD#, and BHE# May Remain Active During Idle, Standby, and Powerdown Modes

PROBLEM: The RD# and BHE# pins are active low, and these signals should be driven inactive during idle, standby and powerdown modes. However, under some circumstances, when reading the IDLPD instruction from external memory these pins remain active during these modes. Also, when entering powerdown, idle, and standby, the address/data bus may continue to drive the last state of the bus.

IMPLICATION: While in powerdown, idle or standby, RD# and BHE# may remain active. This means that external memory may be active even during powerdown. Also the address/data bus may continue to drive the last state of the bus.

WORKAROUND: To avoid driving the bus during entry into powerdown, standby and idle, code must be executing and fetching from internal code RAM. Therefore, if the instruction (IDLPD #x) to enter into powerdown, idle or standby modes occurs from external memory, the bus will remain driving. However, you can have a call to a routine in internal code RAM that executes the IDLPD# instruction and a RET. It could look something like the following:

```
CSEG at EXT_MEM
    ....
    ecallpowerdown
    ....
CSEG at CODE_RAM
powerdown:
    IDLPD #2
    RET
standby:
    IDLPD #3
    RET
idle:
    IDLPD #1
    RET
```

Do not locate these routines on the top boundary of code RAM or the 80296SA could begin to fetch externally before the IDLPD instruction is executed and the bus could be stuck driving. Locate these routines in the middle or beginning of the code RAM.

STATUS: Fix. Refer to Summary Table of Changes to determine the affected stepping(s).

9600004. Address, Data, EPORT Pins Weakly Held High During Hold

PROBLEM: The pin status table in the *80296SA Microcontroller User's Manual* states that the address bus, EPORT, and address/data bus should be high-impedance during hold, but these pins are weakly held high instead.

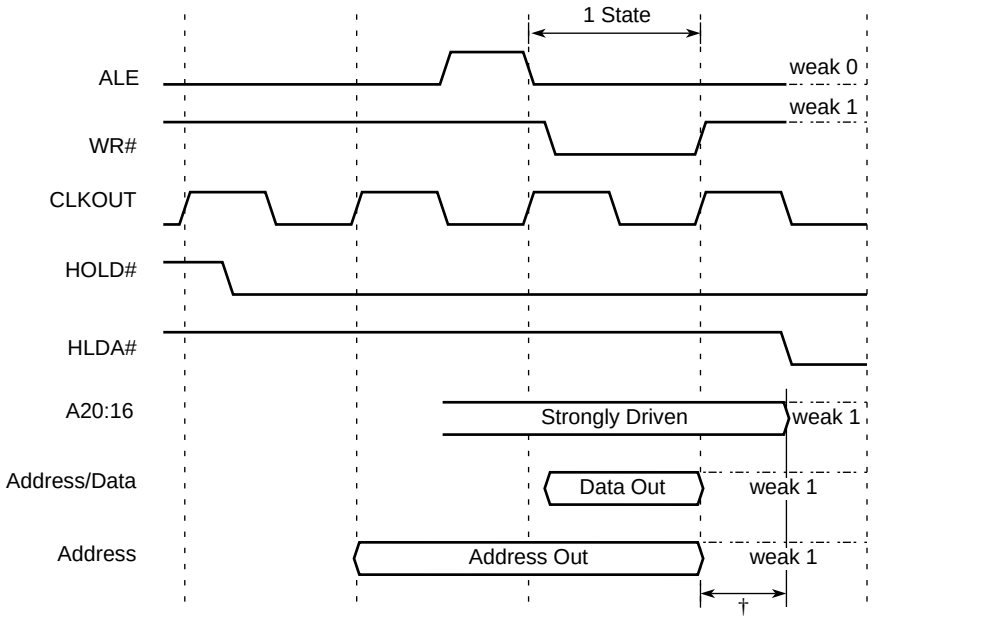
IMPLICATION: Potential bus contention could occur if the device driving the bus is not strong enough to drive the weakly held high signals to the desired state.

WORKAROUND: Do not use hold or ensure that the device driving the bus can overcome the weak driver.

STATUS: Fix. Refer to Summary Table of Changes to determine the affected stepping(s).

9600005. Address, Data Pins Driven Early Before Controller Enters Hold

PROBLEM: When going into bus hold, the address/data bus and address bus pins go to their hold values about 1/2 state early before HLDA# goes low. The other bus signals, such as the EPORT, ALE, RD#, WR#, etc., remain strongly driven for about 1/2 state longer until HLDA# goes active. Please refer to the diagram:



Behavior of Bus Signals When Entering Bus Hold

IMPLICATION: In the figure above, a write cycle is occurring just before the 80296SA releases the bus and goes into hold mode. This write cycle cannot be relied upon, since the 80296SA stops driving valid information on the address and address/data bus at nearly the same time as WR# goes high. The address behaves the same as in normal bus cycles. The data is held longer, though, in the normal bus cycle.

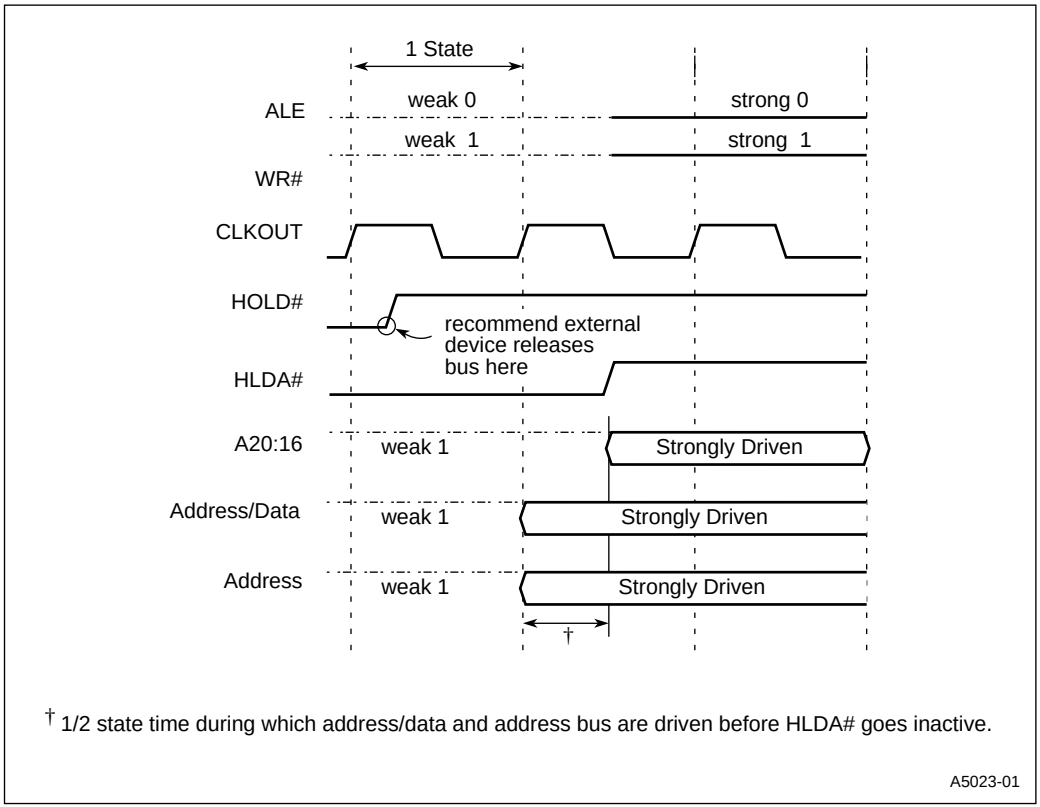
This may make bus hold unreliable and unusable.

WORKAROUND: There is no workaround unless the device receiving data does not expect the data to be held after rising edge of WR#.

STATUS: Fix. Refer to Summary Table of Changes to determine the affected stepping(s).

9600006. Address, Data Pins Driven Early Before Controller Exits Hold

PROBLEM: When coming out of bus hold, the 80296SA starts to strongly drive the address/data and address bus pins about 1/2 state early before HLDA# goes inactive. The other bus signals such as EPORT, ALE, RD#, WR#, etc. start being driven about the same time as HLDA# gets released, but the address/data and address bus start being driven about 1/2 state before HLDA# gets released. Please refer to the diagram:



Behavior Of Bus Signals When Exiting Bus Hold

IMPLICATION: There may be contention on the address/data and address bus, but this is not likely, since HOLD# has already been released.

WORKAROUND: If the external device releases the bus when HOLD# goes inactive, the signals driven before HLDA# goes inactive are not an issue and should not cause bus contention. It is recommended that HOLD# be used as the signal to inform devices to release the bus.

STATUS: Fix. Refer to Summary Table of Changes to determine the affected stepping(s).

9600007. Generation of Timer Overflow

PROBLEM: The EPA's two up/down timers/counters set the underflow interrupt pending bit on the 0000 to 0FFFFh boundary and the overflow pending bit on the 0FFFFh to 0000h boundary. Previous MCS® 96 controllers generated the overflow interrupt on the 0FFFEh to 0FFFFh and 0001h to 0000h boundaries. This was fixed on the 80296SA.

However, the fix used on the 80296SA did not consider the case where 0FFFFh is loaded into the timer which is counting up. The 80296SA will not generate a timer overflow interrupt if the timer is loaded with 0FFFFh and counts up one to 0000h. Similarly the 80296SA will not generate a timer underflow interrupt if the timer is loaded with 0000h and counts down to 0FFFFh. The interrupt does not occur because the 80296SA actually triggers the interrupt pending signal on the 0FFFEh to 0FFFFh or 0001h to 0000h boundary. Then the 80296SA delays the overflow/underflow interrupt to when the timer counts from 0000h to 0FFFFh or 0FFFFh to 0000h.

Therefore, loading the register with 0FFFFh (timer counting up) or 0000h (timer counting down) will not generate the overflow interrupt because the timer has not counted through 0FFFEh to 0FFFFh or through 0001h to 0000h to generate the interrupt pending signal.

IMPLICATION: If code loads the timer with 0FFFFh while it is counting up or with 0000h while it is counting down, the 80296SA will not generate an overflow/underflow interrupt.

WORKAROUND: If your code loads the timer, verify whether it is 0000h or 0FFFFh. If timer is 0000h or 0FFFFh, the code must check whether the timer counts over the boundary. Then the code must force the interrupt by setting the timer overflow interrupt bit in the interrupt pending register or call the interrupt service routine.

STATUS: Fix. Refer to Summary Table of Changes to determine the affected stepping(s). The next stepping will generate a timer overflow interrupt including the following cases:

1. The timer is loaded with 0FFFFh and counts up one to 0000h.
2. The timer is loaded with 0000h and counts down to 0FFFFh.

9600008. False NMI Interrupt Request Generated Upon Exiting Reset

PROBLEM: A false nonmaskable interrupt (NMI) will be generated by the rising edge of the RESET# signal if the external NMI pin is held high during a reset sequence. This will cause an undesired branch to NMI address vector FF203EH shortly after the start of code execution.

IMPLICATION: If the NMI pin is held high during a reset sequence, an undesired branch to the NMI vector address will occur shortly after the user code begins executing.

WORKAROUND:

1. Clear the NMI interrupt pending bit in the **first** instruction after exiting reset. This will cause the NMI interrupt request to be ignored and code will continue to execute normally.
2. Hold the NMI pin low during a reset sequence. This will prevent the false NMI interrupt.

STATUS: Fix. Refer to Summary Table of Changes to determine the affected stepping(s).

9600009. When TAG1-Ing a Protected Instruction, the Controller Deadlocks [ICE-specific Erratum Only]

PROBLEM: When using the in-circuit emulator (ICE) under a specific condition, the model can deadlock. The problem arises because there is a conflict in protocol in the following sequence:

```
ei

nop  @TAG1
```

By definition, the EI instruction is protected which means that no interrupt will occur between EI and the next instruction. However, the TAG1 indicates that instead of executing the next instruction, an interrupt should be executed. The conflict is that one protocol says “do not allow interrupts,” and the other says “unconditionally interrupt the current instruction.”

IMPLICATION: This can cause the controller to deadlock.

WORKAROUND: None.

STATUS: Fix. Refer to Summary Table of Changes to determine the affected stepping(s).

9600010. Double Data Read With Hold

PROBLEM: When a hold cycle begins after a read instruction has been latched, a double read can be detected upon completion of the hold cycle.

IMPLICATION: The double read can cause incorrect data to be read.

WORKAROUND: There is no workaround. However, unless the data is read to a self modifying register, the double read will have no impact because the correct data on the second read will replace corrupted data.

STATUS: Fix. Refer to Summary Table of Changes to determine the affected stepping(s).

9600011. Interruptible BMOV Repeat Counter

PROBLEM: When a BMOVI or EBMOVI instruction is interrupted with two or fewer iterations left, the repeat counter is not preserved. And when the 80296SA returns from the interrupt service routine, the repeat counter performs FFFFh moves.

IMPLICATION: The 80296SA will move data that is not intended to be moved, and it may lock up if code is unintentionally moved.

WORKAROUND:

1. Check the value of the repeat counter. If the value is not zero, then the move was interrupted. Code must reload the counter and complete the move.

Sample code:

```
      ld      r20, repeat_counter
move: rpti    r20
      ld      ICX0,ICX1                 ; move data
      cmp   repeat_counter,zero_reg     ; check if rpt counter is 0
      je    move_done                   ; all done if 0
      ld    r20,repeat_counter          ; rpt was interrupted reload count
                                        ; with remaining iterations
      sjmp    move                      ; jmp back to rpt prefix to continue
move_done:
```

2. Use only the uninterruptible block move instruction (BMOV).

STATUS: Fix. Refer to Summary Table of Changes to determine the affected stepping(s).

9600012. 8-Bit Bus ICX Add With Carry

PROBLEM: An ADDC or SUBC instruction that is executed from slow memory can cause a carry error. If the instruction generates a carry and the carry bit was not previously set, then the carry will be included in the current operation and cause an incorrect result.

IMPLICATION: Result can be incorrect, off by one.

WORKAROUND: Run code and fetch data from internal memory.

STATUS: Fix. Refer to Summary Table of Changes to determine the affected stepping(s).

9600013. Premature Exit of Hold When POPA Is Used

PROBLEM: When a hold cycle begins after a POPA instruction is latched, incorrect data is momentarily written into the WSR. This causes the 80296SA to terminate the hold cycle prematurely by releasing HLDA# before the external device releases HOLD#.

IMPLICATION: The result is improper hold cycles, which could inhibit external use of the bus and could cause corrupt data.

WORKAROUND: A possible workaround is to use POP WSR instead of POPA. Because the POP WSR instruction executes more quickly than POPA, the momentary incorrect data is not an issue.

STATUS: Fix. Refer to Summary Table of Changes to determine the affected stepping(s).

9600014. Data Hold Timing For CCB1

PROBLEM: When loading CCB1, data must be held until the start of the first code fetch. CCB1 data is not latched until the address of the first code fetch appears on the address bus.

IMPLICATION: MODE64 could be set incorrectly. If the data bus is driven before the first code fetch, CCB1 data could be corrupted before it is latched.

WORKAROUND: Do not run the 80296SA at extremely low frequencies. Do not change the CCB1 data bus before the first code fetch data comes across the data bus.

STATUS: Fix. Refer to Summary Table of Changes to determine the affected stepping(s).

9600015. Edge/Level Sensitive External Interrupt Exit from Powerdown and Standby

PROBLEM: External interrupts used to exit powerdown **and standby** are edge triggered. If the external interrupt edge occurs while the device is executing the powerdown **or standby** instruction, but before it is completed, the edge may be missed.

IMPLICATION: If the external interrupt edge used to exit powerdown **or standby** is asserted after the time the powerdown **or standby** instruction is executed but before it is completed, the edge will be missed and the 80296SA may get stuck in powerdown **or standby**.

WORKAROUND: Ensure that code causes the external interrupt to occur so that the edge appears after the 80296SA is in powerdown.

STATUS: Fix (external interrupts will be changed to level sensitive **during standby and powerdown**). **This will correct the problem, provided that the incoming interrupt's minimum pulse width is greater than twice the total execution time of the powerdown or standby (IDLPD) instruction.** Refer to Summary Table of Changes to determine the affected stepping(s).

9600016. RETI Instruction in Programmable Priority Mode May Clear IN_PROG Bit

PROBLEM: If a jump, branch, call, or interrupt request occurs immediately before a RETI instruction, the microcontroller should fetch the RETI instruction but should not execute it. However, the RETI instruction is partially executed during the prefetch. The portion of the RETI instruction that is executed clears the highest-priority IN_PROG bit.

IMPLICATION: In programmable-priority mode, lower-priority interrupts are not allowed to interrupt a higher-priority interrupt. However, a lower-priority interrupt can interrupt a higher-priority interrupt after the RETI instruction is prefetched but before the higher-priority interrupt service routine completes.

WORKAROUND: Place an uninterruptible instruction before the RETI as shown in these examples:

example 1:

```
BR      $
DCB     0FEH
RETI
```

example 2:

```
JNE     label
DCB     0FEH
RETI
```

The DCB 0FEH is the optional, signed-prefix opcode for the two-byte, signed multiply and divide instructions. Adding this instruction does two things:

1. The DCB 0FEH instruction is prefetched before the RETI. The time delay prevents the microcontroller from partially executing the RETI instruction.
2. The uninterruptible DCB 0FEH instruction prevents the microcontroller from acknowledging an interrupt request between the added instruction and the RETI instruction. An uninterruptible instruction such as DCB0FEH prevents the microcontroller form clearing an IN_PROG bit until the complete RETI instruction is executed.

The following uninterruptible instructions can be used in place of the DCB 0FEH instruction:

- a protected instruction: DI, EI, POPA, POPF, PUSHA, PUSHF
- a repeat instruction: RPT, RPTxxx

STATUS: Plans to fix this erratum are under evaluation. Refer to Summary Table of Changes to determine the affected stepping(s).

9600017. External Code Fetch During Read-Modify-Write Instruction Overwrites Interrupt Pending Register

PROBLEM: Certain conditions cause a read-modify-write instruction to overwrite the INT_PEND register. If a code-fetch to external memory occurs between the read and write operations of a read-modify-write instruction and an interrupt request occurs during this time, the INT_PEND register is updated (an interrupt pending bit is set) while the read-modify-write instruction is executing, but is overwritten by the write operation.

IMPLICATION: If an interrupt occurs during a read-modify-write instruction, the INT_PEND register is overwritten by the write and the interrupt is never acknowledged.

WORKAROUND: When interrupts are enabled, any read-modify-write instruction that operates on the INT_PEND register must be executed from code RAM or from 16-bit, zero-wait-state external memory.

STATUS: Plans to fix this erratum are under evaluation. Refer to Summary Table of Changes to determine the affected stepping(s).

SPECIFICATION CHANGES

001. Index Register ICX1 Usage With ST Instruction

PROBLEM: The 80296SA index register post-auto-increment/decrement operation does not take place when the store (ST) instruction is used with the index reference register ICX1 (CPU SFR address 16H) as the source. When using the store instruction, the indirect address in the IDX1 register is stored correctly to the destination address. However, the auto-increment/decrement of the indirect address located in the IDX1 register does not occur at the end of the instruction.

IMPLICATION: The index register auto-increment/decrement operation will not occur if the customer uses the store instruction with the ICX1 index reference register. Index Register IDX0 is not impacted.

WORKAROUND: If the auto-increment/decrement feature is needed for the IDX1 index register, the customer can use the load (LD) instruction with the ICX1 index reference register as the source. For example,

```
LD    IDX0, #3000H
LDB   IDX0+2, #0FFH  ; load IDX0 to point to 0FF3000H

LD    IDX1, #4000H
LDB   IDX1+2, #88H   ; load IDX1 to point to 884000H

LDB   ICB0, #1H      ; set-up auto-increment IDX0 by 1 byte
LDB   ICB1, #1H      ; set-up auto-increment IDX1 by 1 byte

LD    ICX0, ICX1     ; load contents in location 0FF3000H into
                     ; location 884000H
                     ; increment contents of IDX0 and IDX1 by 1 byte
                     ; This instruction used in place of
                     ; ST  ICX1, ICX0
```

STATUS: This operation will be described in future revisions of the user's manual.

002. V_{CC} Specification for 50 MHz Operation

PROBLEM: At 40 MHz (standard) operation, the 80296SA requires that V_{CC} be stable within ±10% of 5.0 volts. At 50 MHz (premium) operation, however, the 80296SA requires that V_{CC} be stable within ±5% of 5.0 volts.

IMPLICATION: Operation at 50 MHz premium requires a more stringent V_{CC} specification and may require a “cleaner” power supply.

WORKAROUND: Depending on the application, a “cleaner” power supply may allow the system to meet the V_{CC} specification.

STATUS: Plans to change this specification are under evaluation. Refer to Summary Table of Changes to determine the affected stepping(s).

003. T_{RXDX} Specification Change to 0 ns (from 2 ns)

PROBLEM: The specification for T_{RXDX} is 0 ns is both multiplexed and demultiplexed modes.

IMPLICATION: This change relaxes the requirement for T_{RXDX} (Data Hold After RD# Inactive).

WORKAROUND: Not applicable.

STATUS: Doc. This specification will be changed in future revisions of the datasheet.

004. T_{CHYX} Specification Replaced by T_{CH1YX} and T_{CH2YX}, including new READY timing diagrams

PROBLEM: The T_{CHYX} specification has been replaced by two specifications, T_{CH1YX} and T_{CH2YX}, to more clearly define the timing requirements for the first READY pulse and for subsequent READY pulses.

IMPLICATION: This change makes it clear that the timing requirements for the first READY pulse differ from those of subsequent READY pulses.

WORKAROUND: Not applicable.

STATUS: Doc. The specifications will be changed in future revisions of the datasheet.

SPECIFICATION CLARIFICATIONS

001. Use of RPD to Exit Powerdown

ITEM: Driving RPD low will not bring the device out of powerdown. It is no longer possible to use the RPD pin to exit from powerdown.

002. P2.3/BREQ# is automatically configured as BREQ# during hold.

ITEM: P2.3/BREQ# is automatically configured as the special-function signal, BREQ#, during bus-hold operations. Plans to change this operation are under evaluation.

003. Interrupt priority scheme clarification.

ITEM: Higher priority interrupts are serviced before lower priority interrupts. A low-priority interrupt is always interrupted by a higher priority interrupt, but not by another interrupt of equal or lower priority. The absolute highest priority interrupt is not interrupted by any other interrupt source.

004. WRH#/BHE# automatically configured as BHE# during ICE mode.

ITEM: CCB0.2 (BHE#) is ignored in the in-circuit (ICE) mode and the microcontroller is forced to BHE mode. Plans to change this such that the WRH#/BHE# pin is configurable while in this mode are under evaluation.

DOCUMENTATION CHANGES

001. Address, Data, EPORT Pins Weakly Held High During Hold (User's Manual)

ITEM: The pin status table in the *80296SA Microcontroller User's Manual* states that the address bus, EPORT and address/data bus should be high-impedance during hold, but these pins are weakly held high instead. This will be changed on page B-11.

002. Index Register ICX1 Usage With ST Instruction (User's Manual)

ITEM: The 80296SA index register post-auto-increment/decrement operation does not take place when the store (ST) instruction is used with the index reference register ICX1 (CPU SFR address 16H) as the source. When using the store instruction, the contents of the indirect address in the IDX1 register is stored correctly to the destination address. However, the auto-increment/decrement of the indirect address located in the IDX1 register will not occur at the end of the instruction. This will be clarified in the *80296SA Microcontroller User's Manual* on pages 3-19 and C-30.

003. V_{CC} Specification for 50 MHz Operation (Datasheet)

ITEM: For 50 MHz operation, V_{CC} must be stable within ± 5% of 5.0 volts. This item will be added to the next revision of the datasheet.

004. Read-modify-write and MAC Instructions Are Not Protected (User's Manual)

ITEM: The read-modify-write instructions (AND, ANDB, OR, ORB, XOR, XORB) and the eight multiply-accumulate (MAC) instructions are **not** protected instructions. If an interrupt request occurs while one of these instructions is executing, the interrupt will be acknowledged after the current instruction completes. This will be corrected in the *80296SA Microcontroller User's Manual* on page 6-12 ("Situations That Increase Interrupt Latency").

005. AP-445, 8XC196KR Peripherals: A User's Point of View is obsolete; no longer available. (User's Manual)

ITEM: Application note AP-445 (order number 270873) is obsolete. This item will be deleted from the table on page 1-7 of the *80296SA Microcontroller User's Manual* .

006. AP-711, EMI Design Techniques for Microcontrollers in Automotive Applications, is order number 272637. (User's Manual)

ITEM: The order number for AP-711, *EMI Design Techniques for Microcontrollers in Automotive Applications*, is 272637. This item will be corrected in the table on page 1-7 and in the text on page 11-4 of the *80296SA Microcontroller User's Manual* .

007. SQFP package is not available. (Datasheet)

ITEM: The SQFP package is not available. The following changes will be made in the next revision of the datasheet:

- 100-pin SQFP will be deleted from the features list.
- 100-pin SQFP will be deleted from the product nomenclature table.
- Figure 3, the SQFP pinout diagram, will be deleted.
- Tables 2 and 3, the SQFP pin assignment tables, will be deleted.
- 100-pin SQFP will be deleted from Table 17, the thermal characteristics table.

008. ROM option is not available. (Datasheet)

ITEM: The 80296SA microcontroller is not available with ROM. The ROM option will be deleted from Table 1, the product nomenclature table.

009. Marking for 40 MHz is "40." (Datasheet)

ITEM: The speed marking for 40 MHz is "40." The marking will be changed from "no mark" to "40" in Table 1, the product nomenclature table.

010. SQFP package is not available (User's Manual)

ITEM: The SQFP package is not available. Figure B-1, the SQFP package diagram, will be deleted in the next revision of the user's manual.

011. DCD and BGD bit explanations (User's Manual)

ITEM: The explanations of the DCD and BGD bits (page 12-13) contain errors. The next revision of the manual will contain the following corrected text:

The DCD bit in the PWM clock control register (CON_REG0 on page **C-16**) enables and disables the duty-cycle generator. Setting DCD **disables** the duty-cycle generator; clearing DCD **enables** it. The DCD bit is **set** at reset (duty-cycle generator **disabled**).

The **BGD** bit in the serial port control register (SP_CON on page C-**55**) enables and disables the baud-rate generator. Setting **BGD disables** the baud-rate generator; clearing **BGD enables** it. The **BGD** bit is **set** at reset (baud-rate generator disabled).

The bits that implement these new features (DCD in CON_REG0) and **BGD** in SP_CON) are reserved in previous MCS 96 microcontrollers; they are documented as "Reserved; for

compatibility with future devices, write zero to this bit." Therefore, code written for a previous MCS 96 microcontroller system that uses these peripherals will enable the duty-cycle generator and baud-rate generator as part of the initialization.

CON_REG0

Address: 1FB6H
Reset State: FCH

The control (CON_REG0) register controls the clock prescaler for the three pulse-width modulators (PWM0–PWM2) and enables or disables the duty-cycle generator.

7							0
DCD†	—	—	—	—	—	CLK1	CLK0

Bit Number	Bit Mnemonic	Function
7	DCD	Duty Cycle Disable Control This bit allows power conservation when the PWM is not being used. The default disables the duty-cycle generator at power-up or reset. You must clear this bit to enable the duty-cycle generator. 0 = enable the PWM duty cycle generator 1 = disable the PWM duty cycle generator (default at reset)
6:2	—	Reserved; for compatibility with future devices, write zeros to these bits.
1:0	CLK1:0	Enable PWM Clock Prescaler These bits control the PWM output period on the three pulse-width modulators (PWM2:0). **CLK1 CLK0** 0 0 disable clock prescaler 0 1 enable divide-by-two prescaler; PWM output period is 1024 state times 1 X enable divide-by-four prescaler; PWM output period is 2048 state times

012. Change T_{RXDX} specification to 0 ns (Datasheet)

ITEM: The T_{RXDX} specification will be changed to 0 ns (from 2 ns) in the next revision of the datasheet.

013. Deferred bus cycle mode on page 2-17 (User's Manual)

ITEM: The 80296SA automatically invokes the deferred bus cycle mode, while the 8*x*C196NU's chip configuration byte 1 (CCB1) allows you to enable or disable deferred mode.

For a demultiplexed bus cycle, the microcontroller automatically delays the WR# signal (and the next bus cycle) by one state after any bus cycle in which a read or a chip-select change occurs. This delay, called a deferred bus cycle, is designed to reduce bus contention when using slow memory devices.

014. Port 2 register addresses on page 8-4 (User's Manual)

ITEM: The port 2 register addresses are as follows:

P2_DIR	1FD3h
P2_MODE	1FD1h
P2_PIN	1FD7h
P2_REG	1FD5h

015. EPA register addresses on page 10-3 (User's Manual)

ITEM: The EPA_CON and EPA_TIME register addresses are as follows:

EPA0_CON	1F80h
EPA1_CON	1F84h, 1F85h
EPA2_CON	1F88h
EPA3_CON	1F8Ch, 1F8Dh
EPA0_TIME	1F82h
EPA1_TIME	1F86h
EPA2_TIME	1F8Ah
EPA3_TIME	1F8Eh

016. P3_PIN register address on page 12-3 (User's Manual)

ITEM: The P3_PIN register address is 1FDEh.

017. Example for selecting capacitor C_1 on page 12-12 (User's Manual)

ITEM: The sentence following the note on page 12-12 should read as follows:

For example, assume that the oscillator needs at least 12.5 ms to **stabilize** (T_{DIS} = 12.5 ms), V_t is 2.5 V, and the discharge current is 200 µA.

018. Remove Note 1 from Table A-1 on page A-3 and from Table A-7 on page A-63 (User's Manual)

ITEM: Note 1 does not apply to the 80296SA microcontroller. Opcode EE is reserved and it **does** generate an unimplemented opcode interrupt request.

019. Correct note for MSAC instruction on page A-32, MVAC instruction on page A-36, and SHLL instruction on page A-45 (User's Manual)

ITEM: The note in the "Instruction Format" column contains errors (MVAC and "Reserved" are reversedl). The corrected text is as follows:

NOTE: The following table identifies the instruction executed by opcode 0DH with the possible values of lreg.1 and lreg.0:

lreg.1	lreg.0	instruction executed
0	0	SHLL
0	1	**Reserved**
1	0	**MVAC**
1	1	MSAC

020. Correct description of SUBB instruction on page A-53 (User's Manual)

ITEM: The description of the SUBB instruction contains errors. The corrected text is as follows:

SUBTRACT BYTES. Subtracts the **second** source byte operand from the **first**, stores the result in the destination operand, and sets the carry flag as the complement of borrow.

(DEST) ← (SRC1) – (SRC2)

021. Correct reset value of CON_REG0 register on pages C-2 and C-16 (User's Manual)

ITEM: The correct reset value of CON_REG0 is FCh (not 7Ch). The PWM duty-cycle generator is disabled at reset.

022. Replace T_{CHYX} specification with $T_{CH1\ YX}$ and T_{CH2YX} for both multiplexed and demultiplexed modes; replace READY timing diagrams (Datasheet)

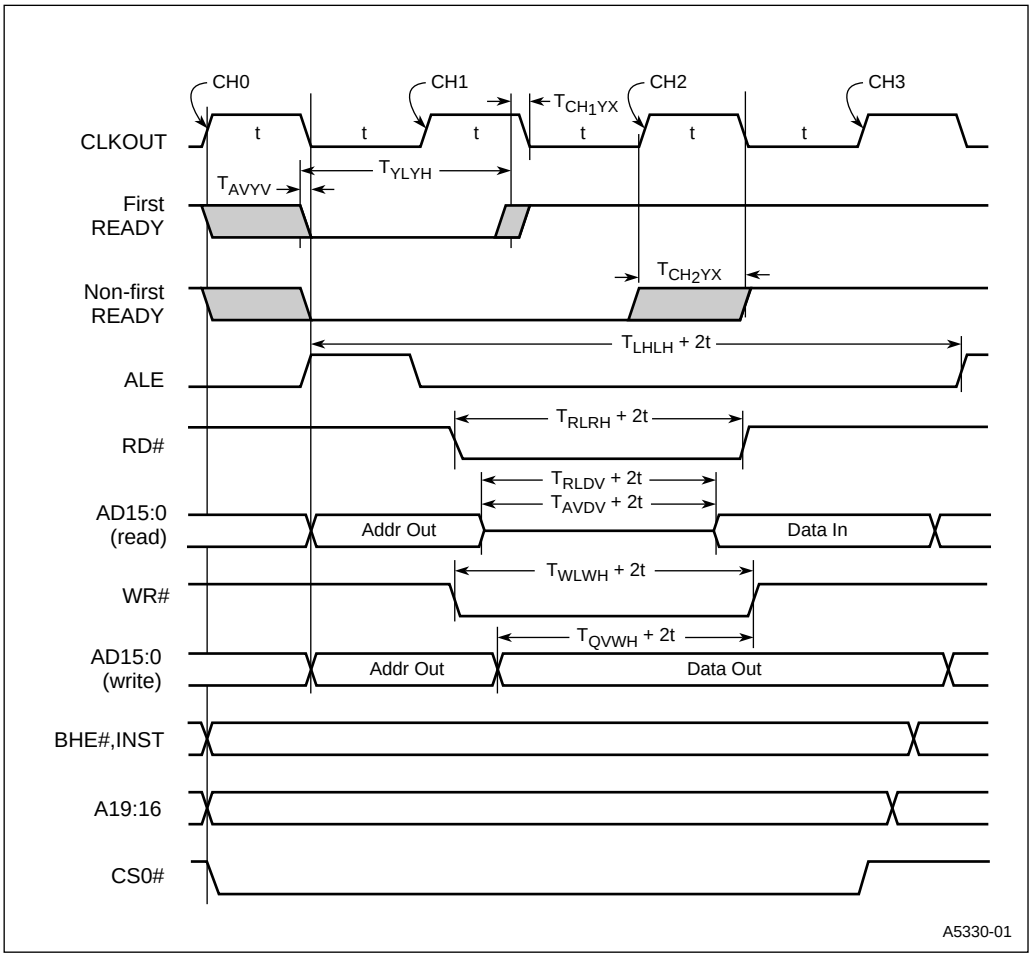
ITEM: The T_{CHYX} specification has been replaced by two specifications, T_{CH1YX} and T_{CH2YX}, to more clearly define the timing requirements for the first READY pulse and for subsequent READY pulses.

The following AC characteristics tables and READY timing diagrams will be included in the next revision of the datasheet.

AC Characteristics the External Memory System Must Meet, Multiplexed Bus Mode

Symbol	Parameter	Min	Max	Units
T_{AVDV}	AD15:0 Valid to Input Data Valid		3t – 32	ns (1, 2)
T_{RLDV}	RD# Active to Input Data Valid		2t – 40	ns (1, 2)
T_{SLDV}	Chip Select Low to Data Valid		4t – 28	ns (1, 2)
T_{CHDV}	CLKOUT High to Input Data Valid		2t – 25	ns
T_{RHDZ}	End of RD# to Input Data Float		t – 3	ns (2)
T_{RXDX}	Data Hold after RD# Inactive	0		ns
T_{AVYV}	AD15:0 Valid to READY (Inactive) Setup		2t – 42	ns (3)
T_{CH1YX}	First READY Hold (Inactive) after CLKOUT High	t – 4	2t – 21	ns (4, 5)
T_{CH2YX}	Non-first READY Hold (Inactive) after CLKOUT High		2t – 21	ns (4)
T_{YLYH}	Non-READY (Inactive) Time	2t	No Upper Limit	ns

NOTES:
1. If using the READY signal to insert wait states, you must program at least one wait state in the BUSCON*x* register because the first falling edge of READY is not synchronized with a CLKOUT edge.
2. If using the BUSCON*x* register **without** the READY signal to insert wait states, add $2t \times n$, where n = number of wait states.
3. If using the BUSCON*x* register to insert wait states, add $2t \times (n–1)$, where n = number of wait states.
4. Exceeding the maximum specification causes additional wait states.
5. If you program two or more wait states in the BUSCON*x* register, the T_{CH1YX} minimum does not apply.

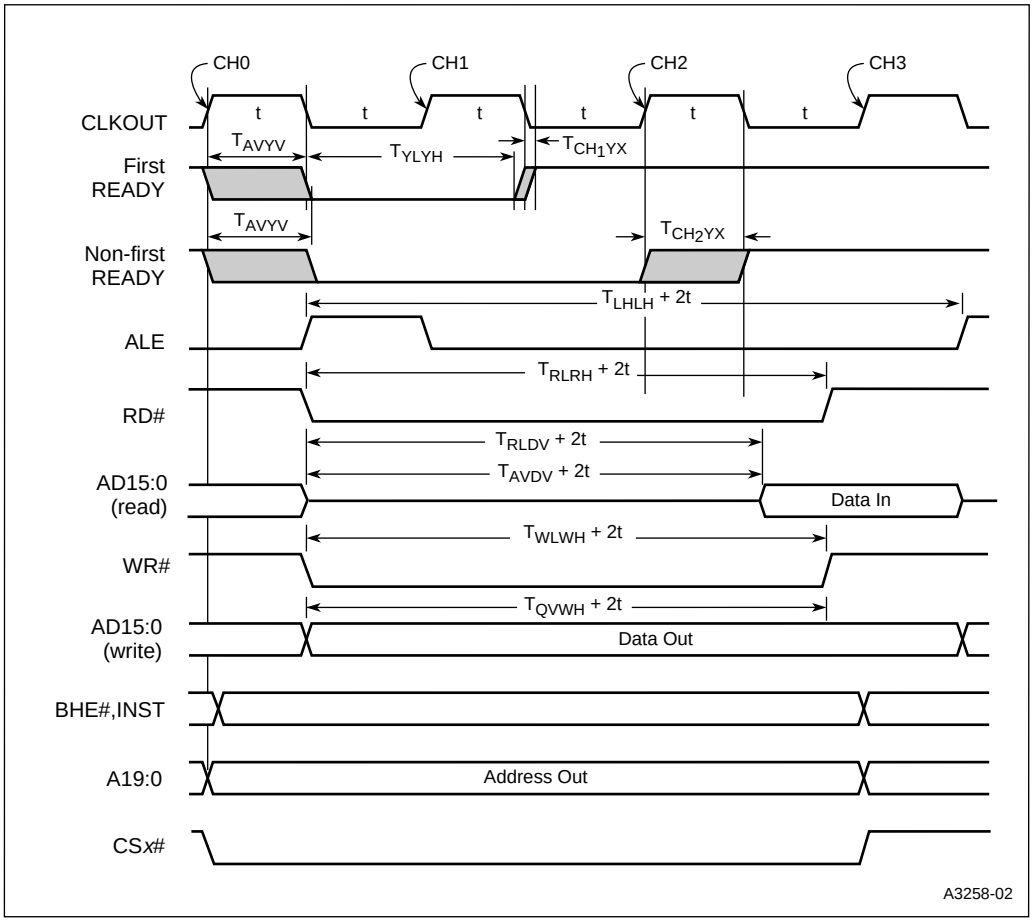


Example READY Timings at 50 MHz, Multiplexed Bus, BUSCON*x* = 1 Wait State

AC Characteristics the External Memory System Must Meet, Demultiplexed Bus Mode

Symbol	Parameter	Min	Max	Units
T_{AVDV}	A19:0 Valid to Input Data Valid		4t – 28	ns (1, 2, 3)
T_{RLDV}	RD# Active to Input Data Valid		3t – 25	ns (1, 2)
T_{SLDV}	Chip Select Low to Data Valid		4t – 28	ns (1, 2, 3)
T_{CHDV}	CLKOUT High to Input Data Valid		2t – 25	ns
T_{RHDZ}	End of RD# to Input Data Float		t	ns (2, 3)
T_{RXDX}	Data Hold after RD# Inactive	0		ns
T_{AVYV}	A19:0 Valid to READY Setup		3t – 45	ns (4)
T_{CH1YX}	First READY Hold after CLKOUT High	t – 4	2t – 21	ns (4, 6)
T_{CH2YX}	Non-first READY Hold after CLKOUT High		2t – 21	ns (4)
T_{YLYH}	Non READY Time	2t	No Upper Limit	ns

NOTES:
1. If using the READY signal to insert wait states, you must program at least one wait state in the BUSCON*x* register because the first falling edge of READY is not synchronized with a CLKOUT edge.
2. If using the BUSCON*x* register **without** the READY signal to insert wait states, add $2t \times n$, where n = number of wait states.
3. If CS*x*# changes or if a write cycle follows a read cycle, add 2t (1 state).
4. If using the BUSCON*x* register to insert wait states, add $2t \times (n–1)$, where n = number of wait states.
5. Exceeding the maximum specification causes additional wait states.
6. If you program two or more wait states in the BUSCON*x* register, the T_{CH1YX} minimum does not apply.



Example READY Timings at 50 MHz, Demultiplexed Bus, BUSCON*x* = 1 Wait State

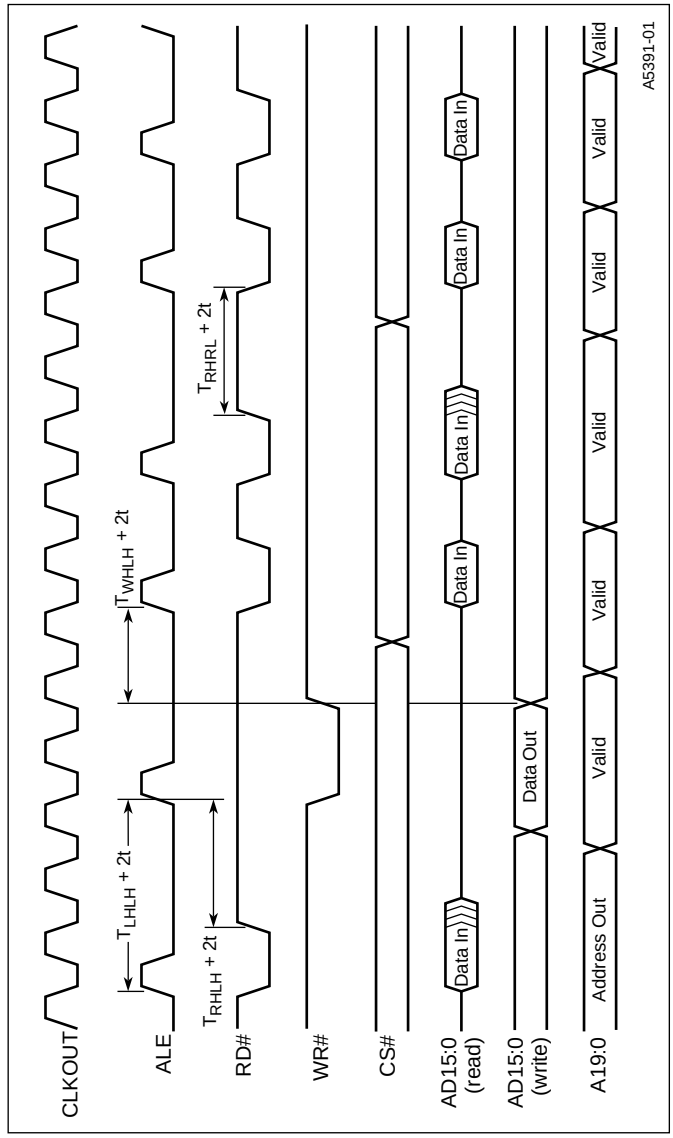
023. Add Jump Penalty Table

Table 0-1. Jump Penalty (in State Times)

Instruction Length (bytes)	Number of Reads to Complete Instruction		Penalty for Jump to Code RAM		Penalty for Jump to External Memory			
					Even Address		Odd Address	
	Even Address	Odd Address	Even Address	Odd Address	Bus Idle	Bus Busy	Bus Idle	Bus Busy
1	1	1	4	4	4	5	4	5
2	1	2	4	5	4	5	6	7
3	2	2	5	5	6	7	6	7
4	2	3	5	6	6	7	8	9
5	3	3	6	6	8	9	8	9
6	3	4	6	7	8	9	10	11

024. Correction to the deferred timing description.

ITEM: In the 80296SA datasheet, first paragraph, the third sentence states, “In the deferred mode, a delay of the WR# signal and the next bus cycle will occur in the first bus cycle following a chip select change and in the first write cycle following a read cycle.” It should state, “Delayed bus cycles cause a 2t delay in the last read bus cycle that occurs prior to a write cycle. A 2t delay will also occur in the last read or write cycle prior to a chip select change.” Refer to the following deferred bus cycle diagram.


CLKOUT
ALE
RD#
WR#
CS#
AD15:0 (read)
AD15:0 (write)
A19:0
T_{LHLH} + 2t
T_{RHLH} + 2t
T_{WHLH} + 2t
T_{RHRL} + 2t
Data In
Data Out
Address Out
Valid
A5391-01