LIBERTY CHIP™ IC

Am29X305/Am29X305A

8-Bit Microcontroller

PRELIMINARY 108207

Am29X305/Am29X305A

Advanced Micro Devices

March 1986

DISTINCTIVE CHARACTERISTICS

- Fetch, decode, and execute a 16-bit instruction in a single machine cycle
- Bit-oriented instruction set
  - Addressable single- or multiple-bit subfields
- Separate buses for instruction, instruction addresses and three-state I/O
- Thirteen 8-bit general-purpose working registers
- On-chip oscillator and timing generator
- Pin-for-pin and functionally compatible to the Signetics 8X305
- FAST
  - The Am29X305A is 20% faster than the Am29X305

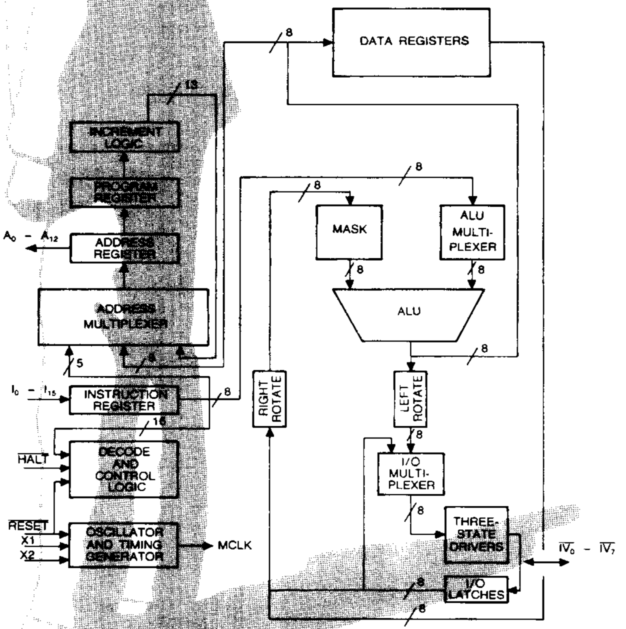
GENERAL DESCRIPTION

The Am29X305 Microcontroller is a high-speed bipolar microprocessor. In a single chip, the Am29X305 combines speed, flexibility, and a bit-oriented instruction set. These features and other basic characteristics of the chip combine to provide cost-effective solutions for a broad range of applications. The Am29X305 is particularly useful in systems that require high-speed bit manipulations, sophisticated controllers, data communications, very fast interface control, and other applications of a similar nature.

The Am29X305 can fetch, decode and execute a 16-bit instruction word in a single machine cycle. Within one instruction cycle the 8-bit data processing path can be programmed to rotate, mask, shift, and/or merge single- or multiple-bit subfields, and in addition, perform an ALU operation. In the same instruction, an internal data field can be input, processed, and output to a specified destination. Likewise, a single- or multiple-bit data field can be internally moved from a given source to a given destination. To interface with I/O and program memory, the Am29X305 uses a 13-bit instruction address bus, a 16-bit instruction bus, and a 5-bit I/O control bus.

The Am29X305A is a plug-in replacement for the Am29X305, featuring AMD's IMOX™ processing.

SIMPLIFIED BLOCK DIAGRAM



BD005921

Orig

AMD

001393

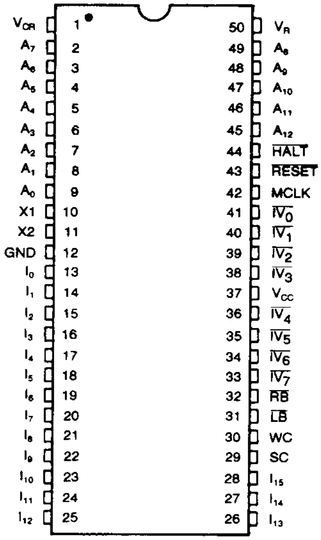
5-3

T 1393

IMOX is a trademark of Advanced Micro Devices, Inc.
LIBERTY CHIP is a trademark of Advanced Micro Devices, Inc.

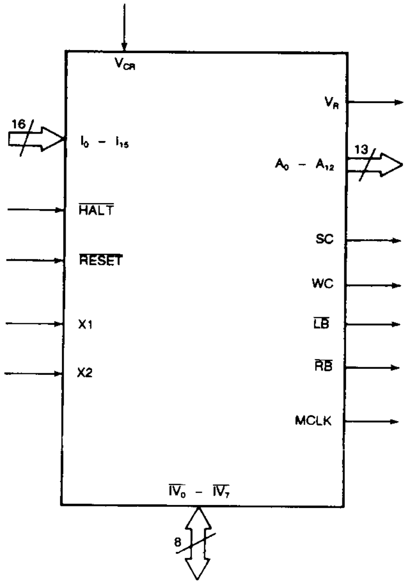
Order # 07639A

CONNECTION DIAGRAM
Top View



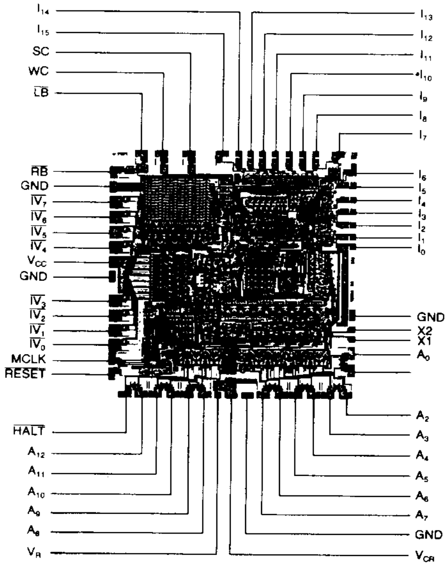
CD009180

LOGIC SYMBOL



LS002210

METALLIZATION AND PAD LAYOUT



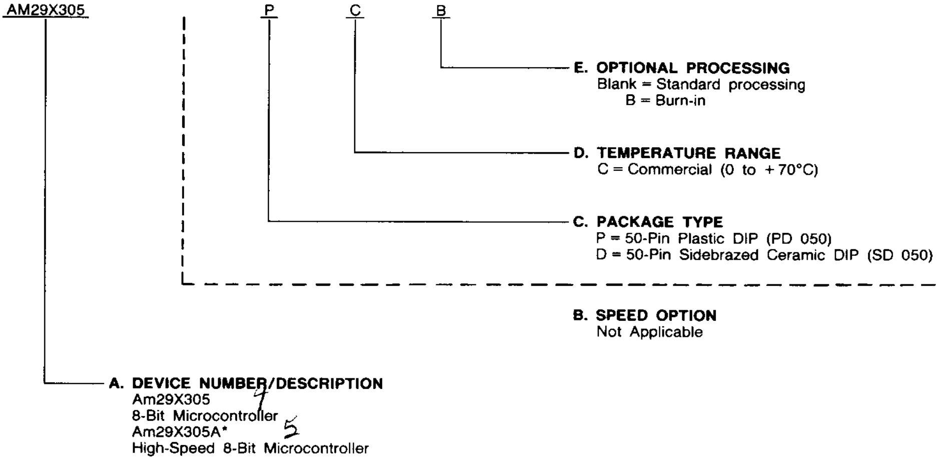
Die Size: 0.203″ x 0.218″

V_{CC} = Power Supply
GND = Ground

ORDERING INFORMATION

Standard Products

AMD products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **A. Device Number**
B. Speed Option (if applicable)
C. Package Type
D. Temperature Range
E. Optional Processing



AM29X305 P C B

E. OPTIONAL PROCESSING
Blank = Standard processing
B = Burn-in

D. TEMPERATURE RANGE
C = Commercial (0 to +70°C)

C. PACKAGE TYPE
P = 50-Pin Plastic DIP (PD 050)
D = 50-Pin Sidebrazed Ceramic DIP (SD 050)

B. SPEED OPTION
Not Applicable

A. DEVICE NUMBER/DESCRIPTION
Am29X305
8-Bit Microcontroller
Am29X305A*
High-Speed 8-Bit Microcontroller

Valid Combinations	
AM29X305, AM29X305A	PC, PCB, DC, DCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*The high-speed 8-bit microcontroller is also planned to be available on 52-pin Leadless Chip Carriers — consult the local AMD sales office for further information.

?

PIN DESCRIPTION

PIN NO.	PIN NAME	I/O	DESCRIPTION
1	V_{CR}	I	Regulated voltage input from series-pass transistor (2N5320 or equivalent).
2 – 9, 45 – 49	$A_0 - A_{12}$	O	Program Address Lines. These active HIGH outputs permit direct addressing of up to 8192 words of program storage. A_{12} is the Least Significant Bit (LSB).
10, 11	X1, X2	I	Timing generator connections for a capacitor, a series resonant crystal, or an external clock source with complementary outputs.
12	GND		Ground
13 – 28	$I_0 - I_{15}$	I	Instruction Lines. These active HIGH input lines receive 16-bit instructions from Program Storage. I_{15} is the LSB.
29	SC	O	Select Command. When HIGH (binary 1), an address is being output on pins $\overline{IV}_0$ through $\overline{IV}_7$.
30	WC	O	Write Command. When HIGH (binary 1), data is being output on pins $\overline{IV}_0$ through $\overline{IV}_7$.
31	$\overline{LB}$	O	Left Bank Control. When LOW (binary 0), devices connected to the Left Bank are accessed (Note: Typically, the $\overline{LB}$ signal is tied to the $\overline{ME}$ input pin of I/O peripherals).
32	$\overline{RB}$	O	Right Bank Control. When LOW (binary 0), devices connected to the Right Bank are accessed (Note: Typically, the $\overline{RB}$ signal is tied to the $\overline{ME}$ input pin of I/O peripherals).
33 – 36, 38 – 41	$\overline{IV}_0 - \overline{IV}_7$	I/O	Interface Vector (I/O Bus). These bidirectional active LOW, three-state lines communicate data and/or addresses to I/O devices and memory locations. A LOW voltage level equals a binary "1." $\overline{IV}_7$ is the LSB.
37	V_{CC}		+ 5-volt Power Supply
42	MCLK	O	Master Clock. This active HIGH output signal is used for clocking I/O devices and/or synchronization of external logic.
43	$\overline{RESET}$	I	When $\overline{RESET}$ input is LOW (binary 0), the Am29X305 is initialized — sets Program Counter/Address Register to zero and inhibits MCLK. For the period of time $\overline{RESET}$ is LOW, the Left Bank and Right Bank signals ($\overline{LB}$ and $\overline{RB}$) are forced HIGH asynchronously.
44	$\overline{HALT}$	I	When $\overline{HALT}$ input is LOW (binary 0), internal operation of the Am29X305 stops at the start of next instruction. MCLK is not inhibited, nor is any internal register affected. However, both $\overline{LB}$ and $\overline{RB}$ signals are synchronously driven HIGH during the first quarter of the instruction cycle time and remain HIGH during the time $\overline{HALT}$ is LOW.
50	V_R	O	Internally generated reference output voltage for external series-pass regulator transistor.

FUNCTIONAL DESCRIPTION

Typical System Configuration

Although the system hookup diagram shown in Figure 1 is of the simplest form, it provides a fundamental look at the Am29X305 Microcontroller and peripheral relationships (see Figure 2 for Detailed Block Diagram). As indicated, the Am29X305 can directly address up to 8K words of Program Storage – either ROM or PROM. The user interface ($\overline{IV}_0 - \overline{IV}_7$) is capable of uniquely addressing 256 Input/Output locations and, with additional bank bits ($\overline{LB}$, $\overline{RB}$), this number is expanded to 512 – each bank comprising 256 addressable locations. The addressable locations of each bank can be used in a variety of ways. A simple method of implementation is shown in Figure 2. When $\overline{LB}$ is active LOW, the left bank is enabled and any one of 256 locations within the RAM memory can be accessed for input/output operations. A similar set of "enable/access" conditions are applicable to the right bank when $\overline{RB}$ is active LOW.

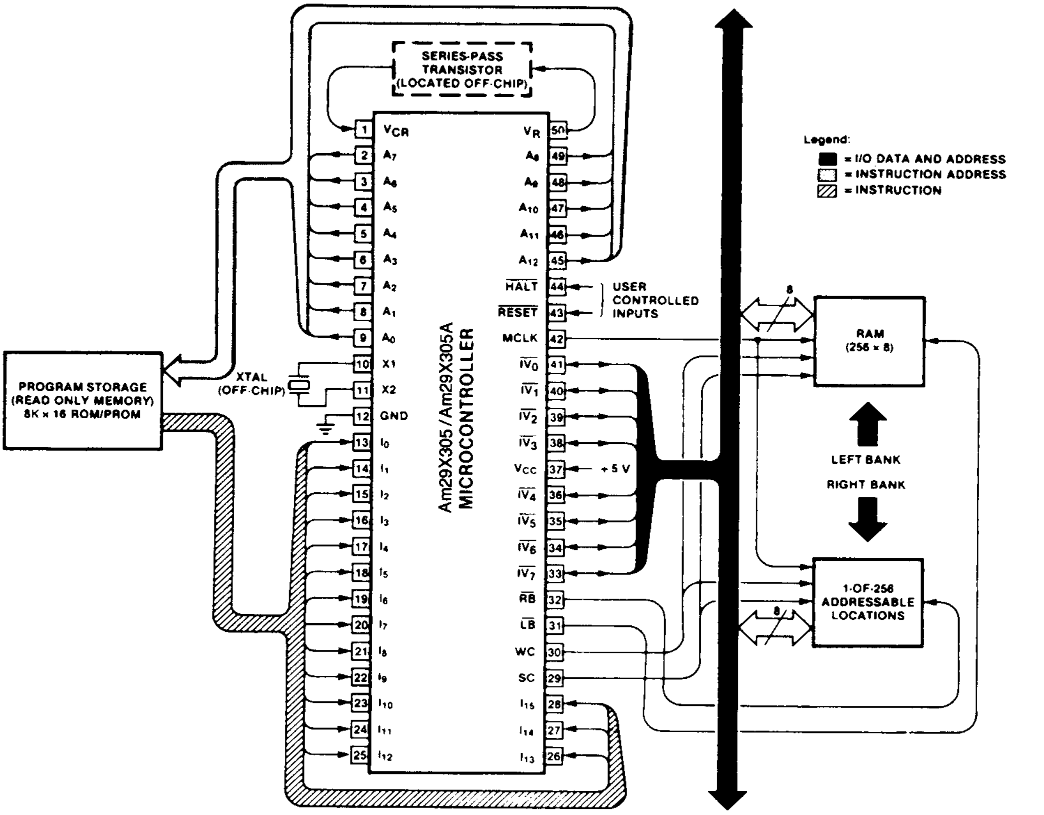


Figure 1. Typical System Hookup

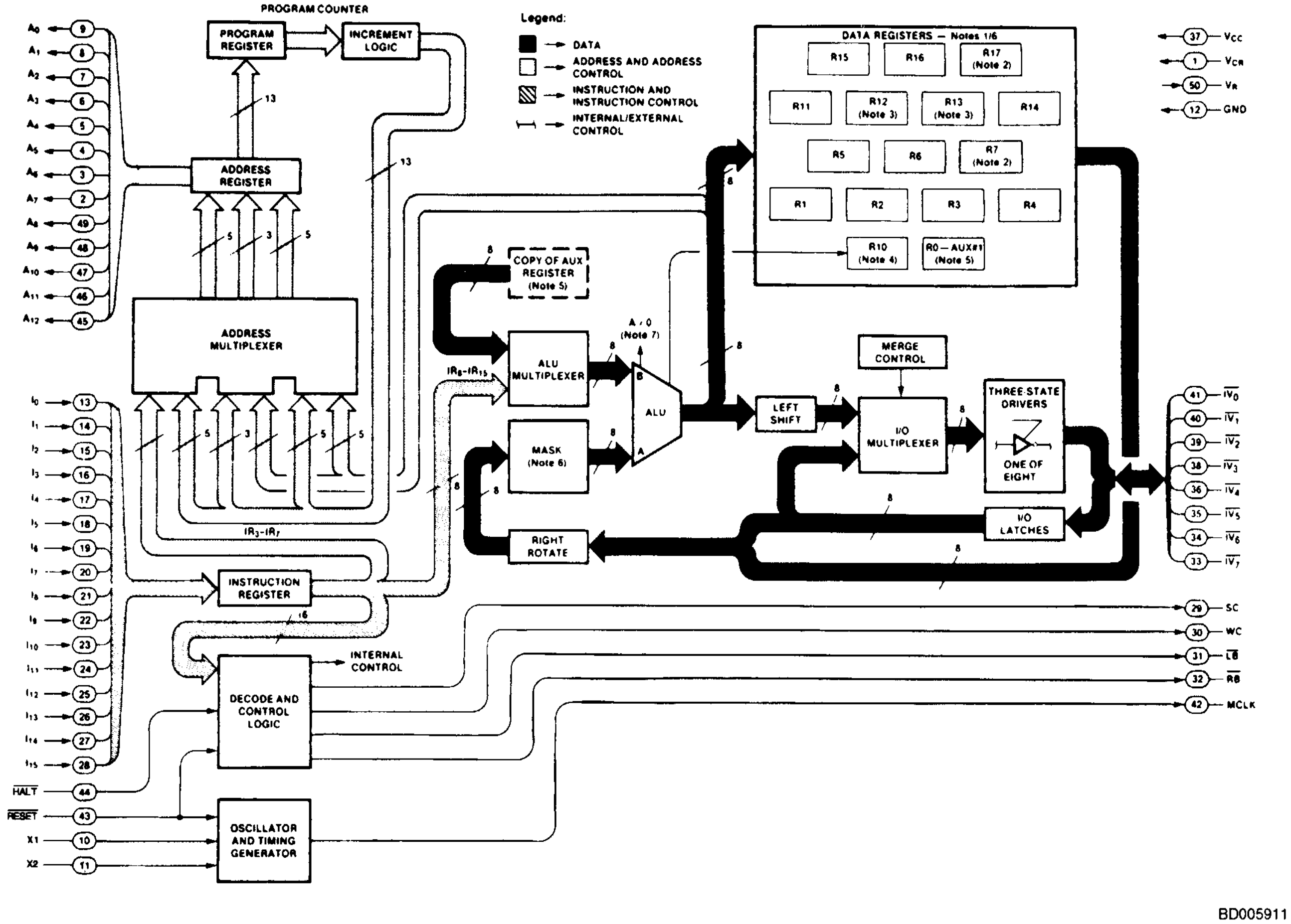


Figure 2. Detailed Block Diagram

See Notes on next page

Notes:
1. Registers R1 – R6, R11 and R14 – R16 are general-purpose working registers.
2. In any instruction where R7 (IVL) or R17 (IVR) is specified as the destination, the 8-bit value is output on the $\overline{IV}$ bus as an $\overline{IV}$ device enable address (SC = HIGH) — R7 = Left Bank and R17 = Right Bank. The results are also stored into the specified internal register and may later be accessed as source data.
3. R12 and R13 are general-purpose working registers for all operations except transmit (XMIT).
4. The least significant bit of register R10 (OVF) is used to reflect the carry-out status resulting from the most recent ADD operation.
5. Auxiliary register R0 #1 is a general-purpose working register that holds the implied operand for Arithmetic and Logical operations. The content of this register is repeated in AUX #2 (shown dotted). The duplicate register is physically part of the ALU and is shown separate only for layout convenience.
6. Internal working registers cannot be operated on by the MASK logic.
7. During NZT instructions the ALU tests for all bits equal to "0" (Transfer if ≠ 0) — Refer to Basic Operations of Am29X305.

Basic Operations of Am29X305

Refer to later discussion of "Instruction Fields" for a detailed examination of all operand fields and subdivisions thereof — "S" (S_0, S_1), "D" (D_0, D_1), "R," "L," "J," and "A."

MOVE Operations

REGISTER-TO-REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA-PROCESSING (POST-ALU)	DESTINATION
R0 – R17 as specified by "S" field of instruction. →	Right-rotate as specified by "R" field of instruction. →	No operation. →	No operation. →	R0 – R7, R11 – R17 asspecified by "D" field of instruction.

REGISTER-TO-$\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA-PROCESSING (POST-ALU)	DESTINATION
R0 – R17 as specified by "S" field of instruction. →	No operation. →	No operation. →	Shift and merge as specified by "D_0" and "L" fields of instruction. →	Variable length field of $\overline{IV}$ bus — Left Bank ($\overline{LB}$) or Right Bank ($\overline{RB}$) as specified by "D_0" and "D_1" fields of instruction.

$\overline{IV}$ BUS-TO-REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA-PROCESSING (POST-ALU)	DESTINATION
Variable length field of $\overline{IV}$ bus — Left Bank ($\overline{LB}$) or Right Bank ($\overline{RB}$) as specified by "S_1" and "S_0" fields of instruction. →	Right-rotate and mask as specified by "S_0" and "L" fields of instruction. →	No operation. →	No operation. →	R0 – R7, R11 – R17 as specified by "D" field of instruction.

$\overline{IV}$ BUS-TO-$\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA-PROCESSING (POST-ALU)	DESTINATION
Variable length field of $\overline{IV}$ bus — Left Bank ($\overline{LB}$) or Right Bank ($\overline{RB}$) as specified by "S_1" and "S_0" fields of instruction. →	Right-rotate and mask as specified by "S_0" and "L" fields of instruction. →	No operation. →	Shift and merge as specified by "D_0" and "L" fields of instruction. →	Variable length field of $\overline{IV}$ bus — Left Bank ($\overline{LB}$) or Right Bank ($\overline{RB}$) as specified by "D_0" and "D_1" fields of instruction.

ADD Operations

REGISTER-to-REGISTER	→ SOURCE → PRE-ALU	Same as MOVE operations, except source data is ADDed to contents of AUXiliary Register R0 via the ALU, if appropriate. Overflow Register R10 (OVF) is also set.	POST-ALU → DEST
REGISTER-to-$\overline{IV}$ BUS	→ SOURCE → PRE-ALU		POST-ALU → DEST
$\overline{IV}$ BUS-to-REGISTER	→ SOURCE → PRE-ALU		POST-ALU → DEST
$\overline{IV}$ BUS-to-$\overline{IV}$ BUS	→ SOURCE → PRE-ALU		POST-ALU → DEST

AND Operations

REGISTER-to-REGISTER	→ SOURCE → PRE-ALU	Same as MOVE operations, except source data is ANDed with contents of AUXiliary Register R0 via the ALU.	POST-ALU → DEST
REGISTER-to-$\overline{IV}$ BUS	→ SOURCE → PRE-ALU		POST-ALU → DEST
$\overline{IV}$ BUS-to-REGISTER	→ SOURCE → PRE-ALU		POST-ALU → DEST
$\overline{IV}$ BUS-to-$\overline{IV}$ BUS	→ SOURCE → PRE-ALU		POST-ALU → DEST

Exclusive OR (XOR) Operations

REGISTER-to-REGISTER	→ SOURCE → PRE-ALU	Same as MOVE operations, except source data is Exclusively ORed with contents of AUXiliary Register R0 via the ALU.	POST-ALU → DEST
REGISTER-to-$\overline{IV}$ BUS	→ SOURCE → PRE-ALU		POST-ALU → DEST
$\overline{IV}$ BUS-to-REGISTER	→ SOURCE → PRE-ALU		POST-ALU → DEST
$\overline{IV}$ BUS-to-$\overline{IV}$ BUS	→ SOURCE → PRE-ALU		POST-ALU → DEST

Execute (XEC) Operations

XEC, REGISTER

SOURCE		DATA PROCESSING (PRE-ALU)		ALU		DATA-PROCESSING (POST-ALU)		DESTINATION
R0 – R17 as specified by "S" field of instruction.	→	No operation.	→	Add source data to 8-bit field specified by instruction literal ($0 \leqslant J \leqslant 377_8$).	→	No operation.	→	Replace 8 LSB of Address Register* with 8-bit sum from ALU. *PGM CTR unchanged.

XEC, $\overline{IV}$ BUS

SOURCE		DATA PROCESSING (PRE-ALU)		ALU		DATA-PROCESSING (POST-ALU)		DESTINATION
Left or Right Bank of $\overline{IV}$ bus as specified by "S" field of instruction.	→	Rotate and mask as specified by "S_0" and "L" fields of instruction.	→	Add masked field of source data to 5-bit literal specified by "J" field of instruction ($0 \leqslant J \leqslant 37_8$).	→	No operation.	→	Replace 5 LSB of Address Register* with 5-bit sum from ALU. *PGM CTR unchanged.

Non-Zero Transfer (NZT) Operations

NZT, REGISTER

SOURCE		DATA PROCESSING (PRE-ALU)		ALU		DESTINATION
R0 – R17 as specified by "S" field of instruction.	→	No operation.	→	Test contents of source register for all zeroes.	→ – – – – – – →	If S = 0, increment PC by 1; if S ≠ 0, replace 8 LSB of AR and PC with literal specified by "J" field of instruction.

NZT, $\overline{IV}$ BUS

SOURCE		DATA PROCESSING (PRE-ALU)		ALU		DESTINATION
Left of Right Bank of $\overline{IV}$ bus as specified by "S" field of instruction.	→	Rotate and mask as specified by "S_0" and "L" fields of instruction.	→	Test contents of masked field for all zeroes.	→ – – – – – – →	If S = 0, increment PC by 1; if S ≠ 0, replace 5 LSB of AR and PC with literal specified by "J" field of instruction.

Transmit (XMIT) Operations

XMIT, REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA-PROCESSING (POST-ALU)	DESTINATION
$0 \leqslant J \leqslant 377_8$–value specified by "J" field of instruction.	No operation.	No operation.	No operation.	R0 – R7, R11, R14 – R17. Load 8-bit integer specified by "J" field into register specified by "D" field.

XMIT 8-BITS IMMEDIATE, $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA-PROCESSING (POST-ALU)	DESTINATION
$0 \leqslant J \leqslant 377_8$–value specified by "J" field of instruction.	No operation.	No operation.	No operation.	Left (R12) or Right (R13) Bank of $\overline{IV}$ bus as specified by "D" field of instruction.

XMIT VARIABLE-BIT FIELD IMMEDIATE $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA-PROCESSING (POST-ALU)	DESTINATION
$0 \leqslant J \leqslant 377_8$–value specified by "J" field of instruction.	No operation.	No operation.	Shift and merge source data as specified by "D_0" and "L" fields of instruction.	Left or Right Bank of $\overline{IV}$ bus as specified by "D" field of instruction.

Jump (JMP) Operations

JMP, ADDRESS

SOURCE	DESTINATION
$0 \leqslant A \leqslant 17777_8$ – value specified by "A" field of instruction.	Load Address Register and Program Counter with contents of "A" field.

Program Storage Interface

As shown in Figure 1, Program Storage is connected to output address lines A_0 through A_{12} (A_{12} = LSB) and input instruction lines I_0 through I_{15}. An address output on A_0/A_{12} identifies one 16-bit instruction word in Program Storage. The instruction word is subsequently input on I_0/I_{15} and defines the Microcontroller operation which is to follow — one instruction word equals one completed operation. Any TTL-compatible memory can be used for Program Storage provided the worst-case access time is compatible with the instruction cycle time used for the application (see timing section for appropriate calculations).

I/O Interface and Control

An 8-bit bidirectional I/O bus, referred to as the Interface Vector ($\overline{IV}$) bus, provides a communication link between the Microcontroller and the two banks of I/O devices. The $\overline{LB}$ (Left Bank) and $\overline{RB}$ (Right Bank) control signals identify which bank is enabled. When both $\overline{LB}$ and $\overline{RB}$ are HIGH (positive), neither bank is enabled and the $\overline{IV}$ bus is inactive (three-state). A functional analysis of the Left and Right Bank signals is shown in Table 1.

TABLE 1. $\overline{LB}$ AND $\overline{RB}$ FUNCTIONAL ANALYSIS

$\overline{LB}$	$\overline{RB}$	FUNCTION
LOW	LOW	This state is not generated by the Am29X305
LOW	HIGH	Enable Left Bank devices
HIGH	LOW	Enable Right Bank devices
HIGH	HIGH	Disable all devices. $\overline{IV}$ bus is three-state.

Both data and I/O address information are multiplexed on the $\overline{IV}$ bus. The SC (Select Command) and WC (Write Command) signals distinguish between data and I/O address information as shown in Table 2.

TABLE 2. SC AND WC FUNCTION TABLE

$\overline{LB}/\overline{RB}$	WC	WC	FUNCTION
HIGH	LOW	LOW	The $\overline{IV}$ bus is three-state and not looking for input data.
LOW	LOW	LOW	The $\overline{IV}$ bus is reading input data.
LOW	LOW	HIGH	Data is being output.
LOW	HIGH	LOW	Address is being output.
X	HIGH	HIGH	This condition is never generated.

Data Processing

Basically, the data processing path of the Am29X305 consists of the Rotate/Mask logic, the Arithmetic Logic Unit (ALU), the Shift/Merge functions, on-chip memory (sixteen 8-bit registers), and the bidirectional $\overline{IV}$ bus interface with its associated driver circuits and internal latches. The on-board memory and the $\overline{IV}$ bus are connected to both inputs and outputs of the ALU via internal 8-bit data paths (see Figure 2). Inputs to the ALU are preceded by right-rotate and data-mask functions. The ALU output is followed by the left-shift and merge operations. Depending on the desired operation, any one or all of the functions (Rotate/Mask/Shift/Merge) can operate on 8 bits of data in a single instruction cycle.

Instruction Cycle

Each operation of the Am29X305 is executed in a single instruction cycle. The instruction cycle is internally divided into four equal parts — each part being as short as 62.5 nanoseconds. Figure 3 shows the general functions that occur during each quarter cycle. Specifics regarding Minimum/Maximum timing and other critical values are described later in this data sheet. During the first quarter cycle, a new instruction from Program Storage is input via $I_0 - I_{15}$ and decoded. If an I/O operation is indicated, new data is fetched from a specified internal register or via the $\overline{IV}$ bus. At the end of the first quarter cycle, the new instruction is latched into the instruction register.

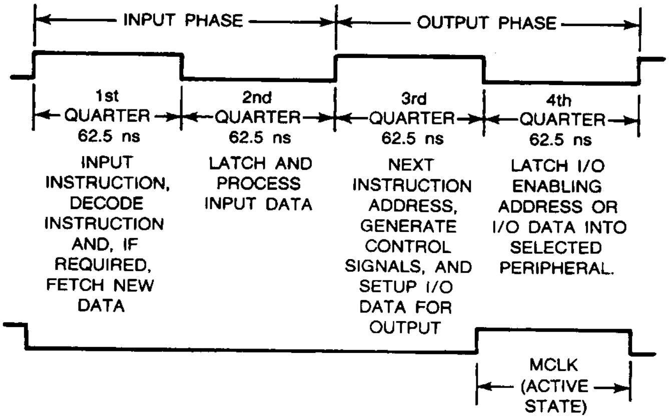

INPUT PHASE
OUTPUT PHASE
1st QUARTER 62.5 ns
2nd QUARTER 62.5 ns
3rd QUARTER 62.5 ns
4th QUARTER 62.5 ns
INPUT INSTRUCTION, DECODE INSTRUCTION AND, IF REQUIRED, FETCH NEW DATA
LATCH AND PROCESS INPUT DATA
NEXT INSTRUCTION ADDRESS, GENERATE CONTROL SIGNALS, AND SETUP I/O DATA FOR OUTPUT
LATCH I/O ENABLING ADDRESS OR I/O DATA INTO SELECTED PERIPHERAL.
MCLK (ACTIVE STATE)
WF021200

Figure 3. Instruction Cycle and MCLK
(With: Crystal = 8 MHz and Cycle Time = 250 ns)

Notes: 1. New instruction must be accepted and latched at end of first quarter cycle.
2. The I/O data latches are open for the first two quarter cycles, that is, for 125 ns.
3. The address changes during third quarter cycle.
4. $\overline{IV}$ bus drivers are active (turned on) during third and fourth quarter cycles.

In the second quarter cycle, the I/O input data stabilizes and preliminary processing is completed. At the end of this quarter, the $\overline{IV}$ latches close and final processing can be accomplished, thus completing the input phase of the instruction cycle. During the third quarter cycle, the address for the next instruction is output to the instruction address bus, $\overline{IV}$ control signals are generated, and both data and destination are set up for the remainder of the output phase. During the fourth quarter cycle, a master clock signal (MCLK) generated by the Am29X305 is used to latch either the I/O-enabling address or the I/O data into peripheral devices connected to the $\overline{IV}$ bus. MCLK can also be used to synchronize any external logic with timing circuits of the Am29X305. To summarize the action, the first half of the instruction cycle deals primarily with input functions and the second half is mostly concerned with output functions.

Instruction Set

General Format and Operating Principles

The 16-bit instruction word (I_0 through I_{15}) from Program Storage is input to the instruction register (see Figure 2) and is subsequently decoded to implement the events to occur during the current instruction cycle. The general format for each instruction word is as follows:

	↓ **MSB**	**LSB** ↓
Bit Positions →	0 1 2	3 4 5 6 7 8 9 10 11 12 13 14 15
	Opcode	Operand(s)

The 3-bit operation code (OPCODE) defines any one of eight classes of instructions. Variations within each class are specified by the remaining thirteen operand bits. The eight instruction classes can be separated into two control areas – Data and Program. General functions within these areas are:

- Data Control
  ADD
  AND } Arithmetic and Logic Operations
  XOR

  MOVE
  XMIT } Movement of Data and Constants

- Program Control
  XEC
  NZT } Branch or Test
  JMP

Instruction Fields

As shown in Table 3, each instruction word consists of an operation code (OPCODE) field and from one to three operand fields. The possible operand fields are Source (S), Destination (D), Rotate/Length (R/L), Literal (J), and Address (A). The OPCODE and operand fields are described in the paragraphs that follow the table.

TABLE 3. FUNCTIONAL DESCRIPTION OF INSTRUCTION SET (Cont'd.)

INSTRUCTION WORD	DESCRIPTION	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE (see Figure 3)		
		CONTROL SIGNAL	INPUT PHASE	OUTPUT PHASE

CLASS = MOVE OPCODE = 0 OPERATION = (S)→D

Register-to-Register

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OPCODE			S					R			D				

$S = 00_8 - 17_8$ $D = 00_8 - 07_8,\ 11_8 - 17_8$

Move content of internal register specified by S-field to internal register specified by D-field. Prior to the "MOVE" operation, right-rotate contents of internal source register by octal value (0 through 7) defined by the R-field.

CONTROL SIGNAL	INPUT PHASE	OUTPUT PHASE
SC	L	H if $D = 07_8,\ 17_8$
WC	L	L
$\overline{\text{LB}}$	H	L if $D = 07_8$
$\overline{\text{RB}}$	H	L if $D = 17_8$

Register-to-$\overline{\text{IV}}$ Bus (Note)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OPCODE			S					L			D				
											D_1		:	D_0	

$S = 00_8 - 17_8$ $D = 20_8 - 37_8$

Move contents of internal register specified by the S-field to the $\overline{\text{IV}}$ bus. Before outputting on $\overline{\text{IV}}$ bus, data is shifted as specified by the least significant octal digit of the D-field and the bits specified by the L-field are merged with the latched I/O data.

CONTROL SIGNAL	INPUT PHASE	OUTPUT PHASE
SC	L	L
WC	L	H
$\overline{\text{LB}}$	L if $D = 20_8 - 27_8$	L if $D = 20_8 - 27_8$
$\overline{\text{RB}}$	L if $D = 30_8 - 37_8$	L if $D = 30_8 - 37_8$

$\overline{\text{IV}}$ Bus-to-Register (Note)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OPCODE			S					L			D				
			S_1		:	S_0									

$S = 20_8 - 37_8$ $D = 00_8 - 07_8,\ 11_8 - 17_8$

Move right-rotated $\overline{\text{IV}}$ bus (source) data specified by the S-field to internal register specified by the D-field. The L-field specifies the length of source data starting from the LSB-position and, if less than 8 bits, the remaining bits are filled with zeros.

CONTROL SIGNAL	INPUT PHASE	OUTPUT PHASE
SC	L	H if $D = 07_8,\ 17_8$
WC	L	L
$\overline{\text{LB}}$	L if $S = 20_8 - 27_8$	L if $D = 07_8$
$\overline{\text{RB}}$	L if $S = 30_8 - 37_8$	L if $D = 17_8$

$\overline{\text{IV}}$ Bus-to-$\overline{\text{IV}}$ Bus (Note)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OPCODE			S					L			D				
			S_1		:	S_0					D_1		:	D_0	

$S = 20_8 - 37_8$ $D = 20_8 - 37_8$

Move right-rotated $\overline{\text{IV}}$ bus (source) data specified by the S-field to the I/O latches. Before outputting on $\overline{\text{IV}}$ bus, shift data as specified by the D-field; then merge source and latched I/O data as specified by the L (length) field.

CONTROL SIGNAL	INPUT PHASE	OUTPUT PHASE
SC	L	L
WC	L	H
$\overline{\text{LB}}$	L if $S = 20_8 - 27_8$	L if $D = 20_8 - 27_8$
$\overline{\text{RB}}$	L if $S = 30_8 - 37_8$	L if $D = 30_8 - 37_8$

CLASS = ADD OPCODE = 1 OPERATION = (S) + (AUX)→D

INSTRUCTION WORD	DESCRIPTION	STATE OF CONTROL SIGNAL
Same as MOVE instruction class	Same as MOVE instruction class except that contents of AUX (R0) register are ADDed to the source data. If there is a "carry" from MSB, then R10 (OVF) = 1 (overflow), otherwise OVF = 0.	Same as MOVE instruction class

CLASS = AND OPCODE = 2 OPERATION = (S) ∧ (AUX)→D

INSTRUCTION WORD	DESCRIPTION	STATE OF CONTROL SIGNAL
Same as MOVE instruction class	Same as MOVE instruction class except that contents of AUX (R0) register are ANDed with source data.	Same as MOVE instruction class

CLASS = XOR OPCODE = 3 OPERATION = (S) ⊕ (AUX)→D

INSTRUCTION WORD	DESCRIPTION	STATE OF CONTROL SIGNAL
Same as MOVE instruction class	Same as MOVE instruction class except that contents of AUX (R0) register are Exclusively ORed with source data.	Same as MOVE instruction class

TABLE 3. FUNCTIONAL DESCRIPTION OF INSTRUCTION SET (Cont'd.)

<table>
<tr><th rowspan="2">INSTRUCTION WORD</th><th rowspan="2">DESCRIPTION</th><th colspan="3">STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE (see Figure 3)</th></tr>
<tr><th>CONTROL SIGNAL</th><th>INPUT PHASE</th><th>OUTPUT PHASE</th></tr>
<tr><td colspan="5">CLASS = <b>XEC</b> OPCODE = <b>4</b> OPERATION = <b>Refer to Description</b></td></tr>
<tr><td rowspan="5"><b>Register Immediate</b>
<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="3">OPCODE</td><td colspan="5">S</td><td colspan="8">J</td></tr>
</table>
$S = 00_8 - 17_8$ $J = 000_8 - 377_8$</td>
<td rowspan="5">Execute instruction at current page address offset by J (literal) + (S). Return to normal instruction flow unless a branch is encountered.

Execute instruction at an address determined by replacing the low-order 8 bits of the Address Register with the following derived sum:

Value of literal (J-field) plus contents of internal register specified by S-field.

The PC is not incremented and the overflow status (OVF) is not changed.</td>
<td>SC</td><td>L</td><td>L</td></tr>
<tr><td>WC</td><td>L</td><td>L</td></tr>
<tr><td>$\overline{\text{LB}}$</td><td>H</td><td>H</td></tr>
<tr><td>$\overline{\text{RB}}$</td><td>H</td><td>H</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td rowspan="5">$\overline{\text{IV}}$ <b>Bus Immediate (Note)</b>
<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="3" rowspan="2">OPCODE</td><td colspan="5">S</td><td colspan="3" rowspan="2">L</td><td colspan="5" rowspan="2">J</td></tr>
<tr><td colspan="5">S_1 : S_0</td></tr>
</table>
$S = 20_8 - 37_8$ $J = 00_8 - 37_8$</td>
<td rowspan="5">Execute instruction at an address determined by replacing the low-order 5 bits of Address Register with the following derived sum: 5-bit value of literal (J-field) plus value of rotated source data specified by S-field. The L-field specifies the length of source data starting from the LSB position and, if less than 8 bits, the remaining bits are filled with zeros; the Program Counter is not incremented and the overflow status (OVF) is not changed.</td>
<td>SC</td><td>L</td><td>L</td></tr>
<tr><td>WC</td><td>L</td><td>L</td></tr>
<tr><td>$\overline{\text{LB}}$</td><td>L if
$S = 20_8 - 27_8$</td><td>H</td></tr>
<tr><td>$\overline{\text{RB}}$</td><td>L if
$S = 30_8 - 37_8$</td><td>H</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="5">CLASS = <b>NZT</b> OPCODE = <b>5</b> OPERATION = <b>Refer to Description</b></td></tr>
<tr><td rowspan="5"><b>Register Immediate</b>
<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="3">OPCODE</td><td colspan="5">S</td><td colspan="8">J</td></tr>
</table>
$S = 00_8 - 17_8$ $J = 000_8 - 377_8$</td>
<td rowspan="5">If data specified by the S-field is not equal to zero, jump to current page address offset by value of J-field; otherwise, increment the Program Counter.

If contents of internal register specified by S-field are non-zero, transfer to address determined by replacing the low-order 8 bits of Address Register and Program Counter with "J"; otherwise, increment PC.</td>
<td>SC</td><td>L</td><td>L</td></tr>
<tr><td>WC</td><td>L</td><td>L</td></tr>
<tr><td>$\overline{\text{LB}}$</td><td>H</td><td>H</td></tr>
<tr><td>$\overline{\text{RB}}$</td><td>H</td><td>H</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td rowspan="5">$\overline{\text{IV}}$ <b>Bus Immediate (Note)</b>
<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="3" rowspan="2">OPCODE</td><td colspan="5">S</td><td colspan="3" rowspan="2">L</td><td colspan="5" rowspan="2">J</td></tr>
<tr><td colspan="5">S_1 : S_0</td></tr>
</table>
$S = 20_8 - 37_8$ $J = 00_8 - 37_8$</td>
<td rowspan="5">If right-rotated and masked $\overline{\text{IV}}$ bus is non-zero, transfer to address determined by replacing low-order 5 bits of Address Register and Program Counter with "J"; otherwise, increment PC. (The L-field specifies the length of source I/O data starting from the LSB-position and, if less than 8 bits, the remaining bits are filled with zeros.)</td>
<td>SC</td><td>L</td><td>L</td></tr>
<tr><td>WC</td><td>L</td><td>L</td></tr>
<tr><td>$\overline{\text{LB}}$</td><td>L if
$S = 20_8 - 27_8$</td><td>H</td></tr>
<tr><td>$\overline{\text{RB}}$</td><td>L if
$S = 30_8 - 37_8$</td><td>H</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

TABLE 3. FUNCTIONAL DESCRIPTION OF INSTRUCTION SET

<table>
<tr><th rowspan="2">INSTRUCTION WORD</th><th rowspan="2">DESCRIPTION</th><th colspan="3">STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE (see Figure 3)</th></tr>
<tr><th>CONTROL SIGNAL</th><th>INPUT PHASE</th><th>OUTPUT PHASE</th></tr>
<tr><td colspan="5">CLASS = <b>XMIT</b> OPCODE = <b>6</b> OPERATION = <b>J → D</b></td></tr>
<tr><td rowspan="4"><b>XMIT, Register</b>
<table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="3">OPCODE</td><td colspan="5">D</td><td colspan="8">J</td></tr></table>
S = $00_8 - 06_8$, 11_8, $14_8 - 16_8$ J = $000_8 - 377_8$</td>
<td rowspan="4">Store 8-bit value specified by "J" into register specified by "D."</td><td>SC</td><td>L</td><td>L</td></tr>
<tr><td>WC</td><td>L</td><td>L</td></tr>
<tr><td>$\overline{LB}$</td><td>H</td><td>H</td></tr>
<tr><td>$\overline{RB}$</td><td>H</td><td>H</td></tr>
<tr><td rowspan="4"><b>XMIT, $\overline{IV}$ Bus Address</b>
<table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="3">OPCODE</td><td colspan="5">D</td><td colspan="8">J</td></tr></table>
S = 07_8, 17_8 J = $000_8 - 377_8$</td>
<td rowspan="4">Enable I/O device on the bank specified by "D," whose address is the 8-bit integer specified by "J." Address "J" is stored in register "D."</td><td>SC</td><td>L</td><td>H</td></tr>
<tr><td>WC</td><td>L</td><td>L</td></tr>
<tr><td>$\overline{LB}$</td><td>H</td><td>L if D = 07_8</td></tr>
<tr><td>$\overline{RB}$</td><td>H</td><td>L if D = 17_8</td></tr>
<tr><td rowspan="4"><b>XMIT 8 Bits Immediate, $\overline{IV}$ Bus (Note)</b>
<table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="3">OPCODE</td><td colspan="5">D</td><td colspan="8">J</td></tr></table>
S = $12_8 - 13_8$ J = $000_8 - 377_8$</td>
<td rowspan="4">Store value of 8-bit integer in the previously enabled I/O port, at the bank destination ($\overline{LB}$ or $\overline{RB}$) specified by "D." Contents of R12 or R13 remain unchanged.</td><td>SC</td><td>L</td><td>L</td></tr>
<tr><td>WC</td><td>L</td><td>H</td></tr>
<tr><td>$\overline{LB}$</td><td>H</td><td>L if D = 12_8</td></tr>
<tr><td>$\overline{RB}$</td><td>H</td><td>L if D = 13_8</td></tr>
<tr><td rowspan="5"><b>XMIT Variable Bit Field Immediate, $\overline{IV}$ Bus (Note)</b>
<table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="3" rowspan="2">OPCODE</td><td colspan="5">D</td><td colspan="3" rowspan="2">L</td><td colspan="5" rowspan="2">J</td></tr>
<tr><td colspan="5">D_1 ⋮ D_0</td></tr></table>
D = $20_8 - 37_8$ J = $00_8 - 37_8$</td>
<td rowspan="5">Transmit Least Significant "L" bits of "J" field to "L-bit" field of $\overline{IV}$ bus specified by "D." If "L" is greater than 5 bits, the MSB bits of destination field are filled with zeros.</td><td>SC</td><td>L</td><td>L</td></tr>
<tr><td>WC</td><td>L</td><td>H</td></tr>
<tr><td>$\overline{LB}$</td><td>L if
D = $20_8 - 27_8$</td><td>L if
D = $20_8 - 27_8$</td></tr>
<tr><td>$\overline{RB}$</td><td>L if
D = $30_8 - 37_8$</td><td>L if
D = $30_8 - 37_8$</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="5">CLASS = <b>JMP</b> OPCODE = <b>7</b> OPERATION = <b>Refer to Description</b></td></tr>
<tr><td rowspan="4"><b>Address Immediate</b>
<table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="3">OPCODE</td><td colspan="13">A</td></tr></table>
A = $00000_8 - 17777_8$</td>
<td rowspan="4">Jump to address in Program Storage specified by A-field; this address is loaded into the Address Register and the Program Counter.</td><td>SC</td><td>L</td><td>L</td></tr>
<tr><td>WC</td><td>L</td><td>L</td></tr>
<tr><td>$\overline{LB}$</td><td>H</td><td>H</td></tr>
<tr><td>$\overline{RB}$</td><td>H</td><td>H</td></tr>
</table>

Note: S_0 specifies the LSB of rotated input data field,
S_1 specifies the bank of $\overline{IV}$ bus from which source data will be input,
D_0 specifies bit position in I/O device with which LSB of processed data will be aligned, and
D_1 specifies the bank of $\overline{IV}$ bus which will be the destination.

Operations Code Field: The 3-bit OPCODE field specifies one of eight classes of instructions. Octal designations for this field and operands for each instruction are shown in the preceding table.

Source (S) and Destination (D) Fields: The 5-bit "S" and "D" fields specify the source and destination, respectively, for whatever operation is defined by the OPCODE. The "S" and/or "D" fields can specify an internal Am29X305 register or any one-to-eight bit field within an I/O device. Octal values and source/destination field assignments for all internal registers are shown in Table 4.

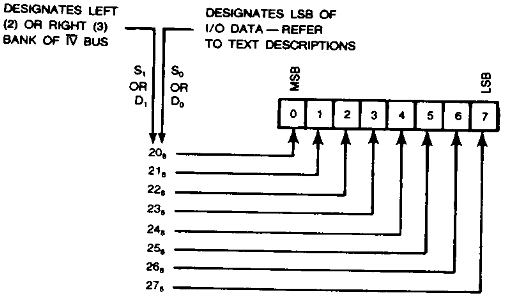
TABLE 4. OCTAL ADDRESSES AND SOURCE/DESTINATION FIELDS FOR Am29X305

ADDRESS	REGISTER DESIGNATION	SOURCE	DESTINATION	ADDRESS	REGISTER DESIGNATION	SOURCE	DESTINATION
00_8	R0 (AUX) — General purpose register	X	X	10_8	R10 (OVF — Overflow register)	X	
01_8	R1 — General purpose register	X	X	11_8	R11 — General purpose register	X	X
02_8	R2 — General purpose register	X	X	12_8	R12 — General purpose register (Note)	X	X
03_8	R3 — General purpose register	X	X	13_8	R13 — General purpose register (Note)	X	X
04_8	R4 — General purpose register	X	X	14_8	R14 — General purpose register	X	X
05_8	R5 — General purpose register	X	X	15_8	R15 — General purpose register	X	X
06_8	R6 — General purpose register	X	X	16_8	R16 — General purpose register	X	X
07_8	R7 — Special purpose register (refer to next paragraph)	X	X	17_8	R12 — Special purpose register (refer to next paragraph)	X	X

Note: R12 and R13 function as general-purpose working registers for all except transmit (XMIT). During a transmit instruction where R12 or R13 is the destination, the 8-bit 'J' field is immediately transferred to the $\overline{IV}$ bus (for this operation, the contents of the designated register remain unchanged).

In instructions where $R7_8$ (IVL) or $R17_8$ (IVR) is specified as the destination, the 8-bit value is output on the $\overline{IV}$ bus as an I/O device address or memory location. Register R7 selects the Left Bank and register R17 selects the Right Bank. The results are also stored into the specified internal register ($R7_8$ or $R17_8$) and may later be accessed as source data. When the $\overline{IV}$ bus is specified as a source and/or destination, the "S" and "D" fields are split into two parts; that is,

- Source (S) = S_1, S_0 and Destination (D) = D_1, D_0 where,
  S_0 specifies the LSB of rotated input data field,
  S_1 specifies the bank of $\overline{IV}$ bus from which source data will be input,
  D_0 specifies bit position in I/O device with which LSB of processed data will be aligned, and,
  D_1 specifies the bank of $\overline{IV}$ bus which will be the destination.


DESIGNATES LEFT (2) OR RIGHT (3) BANK OF $\overline{IV}$ BUS
DESIGNATES LSB OF I/O DATA — REFER TO TEXT DESCRIPTIONS
S_1 OR D_1
S_0 OR D_0
MSB
LSB
0
1
2
3
4
5
6
7
20_8
21_8
22_8
23_8
24_8
25_8
26_8
27_8

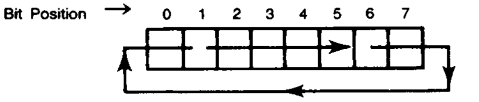
DF006020

Figure 4. Source and/or Destination Code

Notes: 1. The field length of 0-to-8 bits is specified by the "L" field.
2. For the Right Bank, $30_8 - 37_8$ perform equivalent I/O functions.

Rotate (R) and Length (L) Field: The 3-bit R/L field performs one of two functions, specifying either the field Length (L) for I/O operations, or a right-Rotate (R) for internal operations. For a given instruction, the specified function depends upon the contents of the Source (S) and Destination (D) fields.

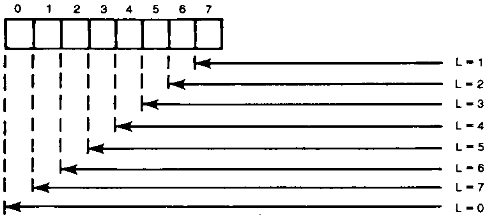
When an internal register is specified by both the Source and Destination fields, the "R" field is invoked and it specifies a right-rotate of the data specified in the "S" field (see Figure 5). The source-register data (up to 8 bits) is right-rotated during the "input phase" of the instruction cycle (see Figure 3) and this function is always performed prior to any ALU operation. (Note: The right-rotate function is implemented on the bus and not in the source register.)



DF006030

Figure 5. Right-Rotate Function

When either or both of the source and destination fields specify a variable-length I/O data field, the "L" field specifies the length of the data field (see Figure 6). If the source field specifies an $\overline{\text{IV}}$ address ($20_8 - 37_8$) and the destination field specifies an internal register ($00_8 - 07_8$, $11_8 - 17_8$), the "L" field specifies the length of source data. The source data is formed by right-rotating the $\overline{\text{IV}}$ bus data according to the source address and then masking the result as specified by the "L" field. If length is less than 8 bits, all remaining bits are set to zero prior to processing data in the ALU. If the source field specifies an internal register ($00_8 - 17_8$) and the destination field specifies $\overline{\text{IV}}$ bus data ($20_8 - 37_8$), the "L" field specifies the length of the destination data. To form the destination data, the ALU output is left-shifted according to the destination address and then masked to the required length (see Figure 6). The destination data is merged with data in the I/O latches to finalize the $\overline{\text{IV}}$ bus data. Hence, a one-to-eight bit destination data field can be inserted into the existing 8-bit I/O port without modifying surrounding bits. If both the source and destination fields specify $\overline{\text{IV}}$ bus data ($20_8 - 37_8$), the "L" field specifies the length of both the source and destination data.



DF006040

Figure 6. $\overline{\text{IV}}$ Data Length Specification
(No Rotate Function Specified)

To form the source data, the $\overline{\text{IV}}$ bus input data is right-rotated according to the source address and then masked to the required length (see Figure 6). If length is less than 8 bits, all remaining bits are set to zero before processing in the ALU. To form the destination data, the ALU output is left-shifted according to the destination address and masked to the required length specification. The destination data is then merged into the $\overline{\text{IV}}$ bus data that was used to obtain the source. Thus, if the source and destination addresses are on the same bank, the $\overline{\text{IV}}$ bus data written to the destination I/O port appears unmodified, except for bits changed during the shift-and-mask operations. If the source and destination addresses refer to different banks, the destination I/O port is changed to contain the contents of the source I/O port in those bit positions not affected by the destination data.

J Field: The 5-bit or 8-bit "J" field is used to load a literal value (contained in the instruction) into a register, into a variable I/O data field, or to modify the low-order bits of the Program Counter. The bit length of the "J" field is implied by the "S" and "L" fields in the XEC, NZT, and XMIT instructions, based on the following conditions:

- When the Source (S) field specifies an internal register, the literal value of the "J" field is an 8-bit binary number.
- When the Source (S) field specifies a variable I/O data field, the literal value of the "J" field is a 5-bit binary number.

A Field: The 13-bit "A" field is an address field which allows the Am29X305 to directly branch to any of the 8192 locations in Program Storage memory.

Formation of Instruction Address

The Address Register and Program Counter are used to generate addresses for accessing an instruction from Program Storage. The instruction address is formed in one of the following ways:

- For all except the JMP, XEC, and a "satisfied" NZT instruction, the Program Counter is incremented by one and placed in the Address Register.
- For the JMP instruction, the 13-bit "A" field contained in the JMP instruction word replaces the contents of both the Address Register and the Program Counter.
- For the XEC instruction, the Address Register is loaded with bits from the Program Counter modified as follows:
  - XEC using $\overline{\text{IV}}$ Bus Data: low-order 5 bits of ALU output replaces counterpart bits in Address Register.
  - XEC using Data from Internal Register: low-order 8 bits of ALU output replaces counterpart bits in Address Register.

The Program Counter is not modified for either of the above conditions.

- For a "satisfied" NZT instruction, the low-order 5 bits (NZT source is $\overline{\text{IV}}$ bus data) or low-order 8 bits (NZT source is an internal register) of both the Address Register and Program Counter are loaded with the literal value specified by the "J" field of instruction word.

Data Addressing

The source and/or destination addresses of the data to be operated upon are specified as part of the instruction word. As shown earlier, source/destination addresses are specified using a 5-bit code ($00_8 - 37_8$). When the most significant octal digit is a "0" or a "1," the source and/or destination address is an internal register. If the most significant digit is a 2 or 3, an $\overline{\text{IV}}$ bus operation is indicated — 2 specifying a Left-Bank ($\overline{\text{LB}}$) operation and 3 specifying a Right-Bank ($\overline{\text{RB}}$) operation. The least significant octal digit (0 through 7) indicates either a specific internal register address or positioning information for the LSB when specifying $\overline{\text{IV}}$ bus data. Referring to Table 3, AUXiliary Register R0 (00_8) is the implied source of the second argument for the ADD, AND, and XOR operations. IVL register R7 and IVR register R17 (destination addresses 07_8 and 17_8, respectively) provide a means of routing enabling address information to I/O peripherals. With IVL or IVR specified as the destination address, data is placed on the $\overline{\text{IV}}$

bus during the output phase of the instruction cycle. Simultaneously, a Select Command (SC) is generated to inform all I/O devices that information on the $\overline{IV}$ bus is to be considered as an I/O address. Since the contents of IVL and IVR are preserved, either register may later be accessed as a source of data.

Control outputs $\overline{LB}$ and $\overline{RB}$ are used to partition I/O bus devices into two fields of 256 addresses. With $\overline{LB}$ in the active LOW state and a source address of $20_8 - 27_8$, the left bank of I/O devices are enabled during the input phase of the instruction cycle. With $\overline{RB}$ in the active LOW state and a source address of $30_8 - 37_8$, the right bank of devices is enabled. During the output phase, $\overline{LB}$ is LOW if the destination address is 07_8 or $20_8 - 27_8$, whereas $\overline{RB}$ is LOW if the destination address is 17_8 or $30_8 - 37_8$. Each address field ($\overline{LB}$ and $\overline{RB}$) can have a different I/O device selected. That is, data can be transferred from a device in one bank to a device in the other in one instruction cycle.

Design Parameters

Hardware design of an Am29X305-based system largely consists of the following operations:

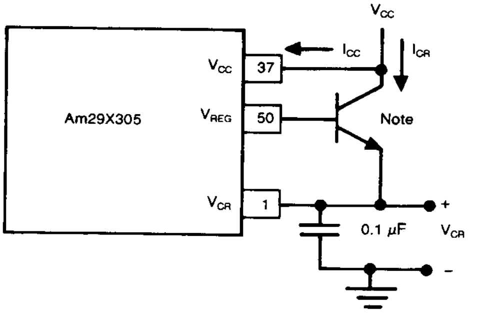
- Selecting and interfacing a Program Storage device — ROM, PROM, etc.
- Selecting and interfacing input/output devices — RAM, Ports, and other 8-bit addressable I/O devices.
- Choosing and implementing System Clock — Capacitor-Controlled, Crystal-Controlled, or Externally-Driven.
- Selection of an off-chip, series-pass transistor.

All information required for easy implementation of these design requirements is provided under the following captions:

- Ordering Information
- Voltage Regulator
- DC Characteristics
- Switching Characteristics
- Timing Considerations
- Clock Considerations
- $\overline{HALT}/\overline{RESET}$ Logic

Voltage Regulator

All internal logic of the Am29X305 is powered by an on-chip Voltage Regulator that requires an external series-pass transistor. Electrical specifications for the off-chip power transistor and a typical hookup diagram are shown in Figure 7. To minimize lead inductance, the transistor should be as close as possible to the Am29X305 package and the emitter should be AC-grounded via a 0.1 microfarad ceramic capacitor.



TC003031

Figure 7. Off-Chip Power Transistor
(Electrical Specifications
and Typical Hookup)

Parameter	Conditions	Limits
h_{fe}	$V_{CE} = 2$ V; 100 mA $< I_C <$ 500 mA	> 50
VBE_{ON}	$V_{CE} = 5$ V; $I_C = 500$ mA	< 1.0 V
V_{CESAT}	$I_C = 500$ mA; $I_B = 50$ mA	< 0.5 V
BV_{CEO}		> 15.0 V
f_t		> 30 MHz

Note: Typical approved parts — 2N5320, 2N5337

ABSOLUTE MAXIMUM RATINGS

Storage Temperature .... −65 to +150°C
Ambient Temperature
with Power Applied .... −0.5 to +7.0 V
Crystal Input Voltage
on X1, X2 .... +2.0 V
Logic Input Voltage
on All Other Pins .... +5.5 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) .... 0 to +70°C
Supply Voltages (V_{CC}) .... +4.75 to +5.25 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified (Notes 1 & 2)

Parameter Symbol	Parameter Description	Test Conditions		Min.	Typ.	Max.	Units
V_{CC}	Supply Voltage			4.75	5.0	5.25	V
V_{IH}	Input HIGH Voltage — X1 and X2			0.6		2.0	V
	All Other Pins			2.0		5.5	
V_{IL}	Input LOW Voltage — X1 and X2					0.4	V
	All Other Pins					0.8	
V_{OH}	Output HIGH Voltage	V_{CC} = Min., I_{OH} = −3 mA		2.4			V
V_{OL}	Output LOW Voltage — $A_0 - A_{12}$	V_{CC} = Min., I_{OL} = 6 mA				0.55	V
	All Other Outputs	V_{CC} = Min., I_{OL} = 16 mA				0.55	
V_{CR}	Regulator Voltage	V_{CC} = 5 V	T_A = 0°C		3.1		V
			T_A = 70°C		2.9		
V_{IC}	Input Clamp Voltage (Note 3)	V_{CC} = Min., I_{IN} = −10 mA				−1.5	V
I_{IH}	Input HIGH Current — X1 and X2	V_{CC} = Max.	V_{IH} = 0.6 V			4.0	mA
	All Other Pins		V_{IH} = 4.5 V			50.0	μA
I_{IL}	Input LOW Current — X1 and X2	V_{CC} = Max., V_{IL} = 0.4 V				−3.0	mA
	$\overline{IV_0} - \overline{IV_7}$					−0.2	
	$I_0 - I_{15}$					−1.6	
	$\overline{HALT}$ and $\overline{RESET}$					−0.4	
I_{OS}	Output Short-Circuit Current (All Output Pins)	V_{CC} = Max. (Note 4)		−30.0		−140.0	mA
I_{CC}	Supply Current	V_{CC} = Max.	T_A = 0°C			195.0	mA
			T_A = 70°C			180.0	
I_{REG}	Regulator Control (Note 5)	V_{CC} = 5.0 V		−10.0		−25.0	mA
I_{CR}	Regulator Current	V_{CC} = Max.	T_A = 0°C			230.0	mA
			T_A = 70°C			200.0	

Notes:
1. Operating temperature ranges are guaranteed after thermal equilibrium has been reached.
2. All voltages measured with respect to ground terminal.
3. Crystal Inputs X1 and X2 do not have internal clamp diodes.
4. At any time, no more than one output should be connected to ground.
5. Maximum available base drive for series-pass transistor.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified (Note 1) (Cont'd.)

Am29X305 Standard

No.	Parameter Symbol	Parameter Description (Note 2)	Limits (Note 3)				Units
			= 250 ns		> 250 ns		
			Min.	Max.	Min.	Max.	
1	T_{PC}	Processor Cycle Time	250.0		250.0		ns
2	T_{CP}	X1 Clock Period	125.0		125.0		ns
3	T_{CH}	X1 Clock HIGH Time	62.5		62.5		ns
4	T_{CL}	X1 Clock LOW Time	62.5		62.5		ns
5	T_{MCL}	MCLK LOW Delay	11.0	44.0	11.0	44.0	ns
6	T_{W}	MCLK Pulse Width (Note 4)	38.5	72.5	$T_{4Q} - 24$	$T_{4Q} + 10$	ns
7	T_{MODO}	Output Driver Turn On (Time MCLK Falling Edge) (Note 11)	145.0	200.0	$T_{1Q} + T_{2Q} + 20$	$T_{1Q} + T_{2Q} + 75$	ns
8	T_{DI}	Output Driver Turn On Time (SC/WC Rising Edge) (Note 12)	20.0		20.0		ns
9	T_{DD}	Input Data to Output Data	69.0	150.0	69.0	150.0	ns
10	T_{MHS}	MCLK Falling Edge to $\overline{\text{HALT}}$ Falling Edge (Note 4)		42.5		$T_{1Q} - 20$	ns
11	T_{MHH}	$\overline{\text{HALT}}$ Hold Time (MCLK Falling Edge) (Note 4)	77.5		$T_{1Q} + 15$		ns
12	T_{ACC}	Program Storage Access Time		60.0		60.0	ns
13	T_{IO}	I/O Port Output Enable Time ($\overline{\text{LB}}/\overline{\text{RB}}$ to Valid IV Data Input)		30.0		30.0	ns
14	T_{MAS}	MCLK Falling Edge to Address Stable (Notes 4, 5 & 6)		173.0		$T_{1Q} + T_{2Q} + 48$	ns
15	T_{IA}	Instruction to Address (Notes 4, 5 & 7)		152.5		$T_{2Q} + 90$	ns
16	T_{IVA}	Input Data to Address (Notes 5 & 8)		96.0		96.0	ns
17	T_{MIS}	MCLK Falling Edge to Instruction Stable (Notes 4 & 12)		36.5		$T_{1Q} - 26$	ns
18	T_{MIH}	Instruction Hold Time (MCLK Falling Edge) (Notes 4 & 10)	67.5		$T_{1Q} + 5$		ns
19	T_{MWH}	MCLK Falling Edge to SC/WC Rising Edge (Note 4)	129.0	160.0	$T_{1Q} + T_{2Q} + 4$	$T_{1Q} + T_{2Q} + 35$	ns
20	T_{MWL}	MCLK Falling Edge to SC/WC Falling Edge	−5.0	20.0	−5.0	20.0	ns
21	T_{MIBS}	MCLK Falling Edge to LB/RB (Input Phase)	5.0	40.0	5.0	40.0	ns
22	T_{IIBS}	Instruction to LB/RB (Input Phase)		32.0		32.0	ns
23	T_{MOBS}	MCLK Falling Edge to LB/RB (Output Phase) (Note 4)	132.0	173.0	$T_{1Q} + T_{2Q} + 7$	$T_{1Q} + T_{2Q} + 48$	ns
24	T_{MIDS}	MCLK Falling Edge to Input Data Stable (Note 4)		70.0		$T_{1Q} + T_{2Q} - 55$	ns
25	T_{MIDH}	Input Data Hold Time (MCLK Falling Edge) (Note 4)	140.0		$T_{1Q} + T_{2Q} + 15$		ns
26	T_{MODH}	Output Data Hold Time (MCLK Falling Edge)	1.0		1.0		ns
27	T_{MODS}	Output Data Stable (MCLK Falling Edge) (Note 4)	145.0	201.0	$T_{1Q} + T_{2Q} + 20$	$T_{1Q} + T_{2Q} + 76$	ns

Notes:
1. See Switching Test Circuits for Output Test Loads.
2. X1 and X2 inputs are driven by an external pulse generator with an amplitude of 1.5 V. All Timing parameters are measured at this voltage level.
3. Instruction Cycle Time Limits.
4. Respectively, T_{1Q}, T_{2Q}, T_{3Q} and T_{4Q} represent time intervals for the first, second, third, and fourth quarter cycles.
5. Capacitive loading for the address bus is 150 pF.
6. T_{MAS} is obtained by forcing a valid instruction and an I/O bus input to occur earlier than the specified minimum setup time.
7. T_{IA} is obtained by forcing a valid instruction input to occur earlier than the minimum setup time.
8. T_{IVA} is obtained by forcing a valid I/O bus input to meet the minimum setup time.
9. T_{MIS} represents the setup time required by internal latches of the Am29X305. In system applications, the instruction input may have to be valid before the worst-case setup time in order for the system to respond with a valid I/O bus input that meets the I/O bus input setup time (T_{IDS} and T_{MIDS}).
10. T_{MIH} represents the hold time required by internal latches of the Am29X305. To generate proper $\overline{\text{LB}}/\overline{\text{RB}}$ signals, the instruction must be held valid until the address bus changes.
11. The minimum figure for these parameters represents the earliest time that I/O bus output drivers of the Am29X305 will turn on.
12. This parameter represents the latest time that the output drivers of the input device should be turned off.

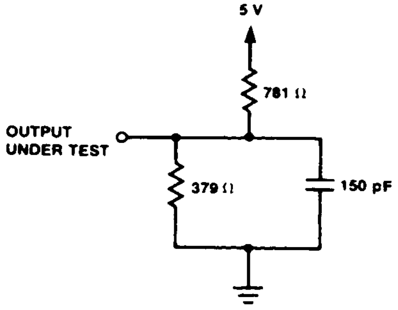
SWITCHING CHARACTERISTICS over operating range unless otherwise specified (Note 1) (Cont'd.)

Am29X305A High-Speed

No.	Parameter Symbol	Parameter Description (Note 2)	Limits (Note 3)				Units
			= 200 ns		> 200 ns		
			Min.	Max.	Min.	Max.	
1	T_{PC}	Processor Cycle Time	200		200		ns
2	T_{CP}	X1 Clock Period	100		100		ns
3	T_{CH}	X1 Clock HIGH Time	50		50		ns
4	T_{CL}	X1 Clock LOW Time	50		50		ns
5	T_{MCL}	MCLK LOW Delay	12	32	12	32	ns
6	T_{W}	MCLK Pulse Width (Note 4)	40	59	$T_{4Q} - 10$	$T_{4Q} + 9$	ns
7	T_{MODO}	Output Driver Turn On Time (MCLK Falling Edge) (Note 11)	116	165	$T_{1Q} + T_{2Q} + 16$	$T_{1Q} + T_{2Q} + 65$	ns
8	T_{DI}	Output Driver Turn On Time (SC/WC Rising Edge) (Note 12)	20		20		ns
9	T_{DD}	Input Data to Output Data	69	120	69	120	ns
10	T_{MHS}	MCLK Falling Edge to $\overline{\text{HALT}}$ Falling Edge (Note 4)		29		$T_{1Q} - 21$	ns
11	T_{MHH}	$\overline{\text{HALT}}$ Hold Time (MCLK Falling Edge) (Note 4)	47		$T_{1Q} - 3$		ns
12	T_{ACC}	Program Storage Access Time		60		60	ns
13	T_{IO}	I/O Port Output Enable Time ($\overline{\text{LB}}/\overline{\text{RB}}$ to Valid IV Data Input)		30		30	ns
14	T_{MAS}	MCLK Falling Edge to Address Stable (Notes 4, 5 & 6)		122		$T_{1Q} + T_{2Q} + 22$	ns
15	T_{IA}	Instruction to Address (Notes 4, 5 & 7)		101		$T_{2Q} + 51$	ns
16	T_{IVA}	Input Data to Address (Notes 5 & 8)		60		60	ns
17	T_{MIS}	MCLK Falling Edge to Instruction Stable (Notes 4 & 12)		24		$T_{1Q} - 26$	ns
18	T_{MIH}	Instruction Hold Time (MCLK Falling Edge) (Notes 4 & 10)	49		$T_{1Q} - 1$		ns
19	T_{MWH}	MCLK Falling Edge to SC/WC Rising Edge (Note 4)	104	130	$T_{1Q} + T_{2Q} + 4$	$T_{1Q} + T_{2Q} + 30$	ns
20	T_{MWL}	MCLK Falling Edge to SC/WC Falling Edge	−4	24	−4	24	ns
21	T_{MIBS}	MCLK Falling Edge to LB/RB (Input Phase)	11	35	11	35	ns
22	T_{IIBS}	Instruction to LB/RB (Input Phase)		34		34	ns
23	T_{MOBS}	MCLK Falling Edge to LB/RB (Output Phase) (Note 4)	105	134	$T_{1Q} + T_{2Q} + 5$	$T_{1Q} + T_{2Q} + 34$	ns
24	T_{MIDS}	MCLK Falling Edge to Input Data Stable (Note 4)		65		$T_{1Q} + T_{2Q} - 35$	ns
25	T_{MIDH}	Input Data Hold Time (MCLK Falling Edge) (Note 4)	112		$T_{1Q} + T_{2Q} + 12$		ns
26	T_{MODH}	Output Data Hold Time (MCLK Falling Edge)	4		4		ns
27	T_{MODS}	Output Data Stable (MCLK Falling Edge) (Note 4)	130	167	$T_{1Q} + T_{2Q} + 30$	$T_{1Q} + T_{2Q} + 67$	ns

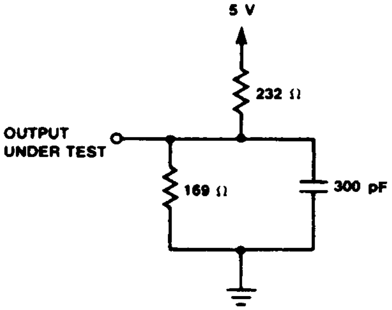
Notes:
1. See Switching Test Circuits for Output Test Loads.
2. X1 and X2 inputs are driven by an external pulse generator with an amplitude of 1.5 V. All Timing parameters are measured at this voltage level.
3. Instruction Cycle Time Limits.
4. Respectively, T_{1Q}, T_{2Q}, T_{3Q} and T_{4Q} represent time intervals for the first, second, third, and fourth quarter cycles.
5. Capacitive loading for the address bus is 150 pF.
6. T_{MAS} is obtained by forcing a valid instruction and an I/O bus input to occur earlier than the specified minimum setup time.
7. T_{IA} is obtained by forcing a valid instruction input to occur earlier than the minimum setup time.
8. T_{IVA} is obtained by forcing a valid I/O bus input to meet the minimum setup time.
9. T_{MIS} represents the setup time required by internal latches of the Am29X305. In system applications, the instruction input may have to be valid before the worst-case setup time in order for the system to respond with a valid I/O bus input that meets the I/O bus input setup time (T_{IDS} and T_{MIDS}).
10. T_{MIH} represents the hold time required by internal latches of the Am29X305. To generate proper $\overline{\text{LB}}/\overline{\text{RB}}$ signals, the instruction must be held valid until the address bus changes.
11. The minimum figure for these parameters represents the earliest time that I/O bus output drivers of the Am29X305 will turn on.
12. This parameter represents the latest time that the output drivers of the input device should be turned off.

SWITCHING TEST CIRCUITS



TC003011

Address



TC003021

Other

Timing Considerations

As shown in the Switching Characteristics table, the minimum instruction cycle time is 250 ns, whereas the maximum is determined by the on-chip oscillator frequency and can be any value the user chooses. With an instruction cycle time of 250 ns, the part can be characterized in terms of absolute values (shown in the first Limits column, designated as: " = 250 ns"). When the instruction cycle time is greater than 250 ns, certain parameters are specified (shown in the second Limits column, designated as: " > 250 ns") in terms of the four quarter cycles (T_{1Q}, T_{2Q}, T_{3Q} and T_{4Q}) that make up one instruction cycle (see Am29X305 Timing Diagram). As the time interval for each instruction cycle increases (becomes greater than 250 ns), the delay for all parameters that are cycle-time dependent is likewise increased. In some cases these delays have a significant impact on timing relationships and other areas of systems design. Subsequent paragraphs describe these timing parameters and reliable methods of calculation.

Timing parameters for the Am29X305 are normally measured with reference to MCLK.

System determinants for the instruction cycle time are:

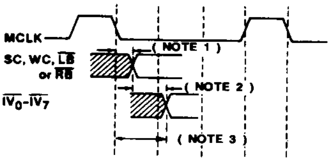
- Propagation delays within the Am29X305
- Access time of Program Storage
- Enable time of the I/O port

Normally the instruction cycle time is constrained by one or more of the following conditions:

Condition 1: Instruction or MCLK to $\overline{LB}/\overline{RB}$ (input phase) plus I/O port access time (T_{IO} $\leqslant$ $\overline{IV}$ data setup time (see Figure 8).

Condition 2: Program Storage access time (T_{ACC}) plus instruction to $\overline{LB}/\overline{RB}$ (input phase) plus I/O port access time (T_{IO}) plus $\overline{IV}$ data (input phase) to address $\leqslant$ instruction cycle time (see Figure 9).

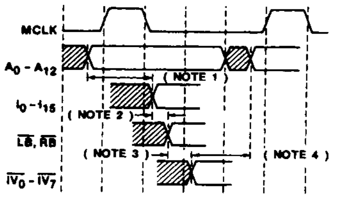
Condition 3: Program Storage· access time plus instruction to address $\leqslant$ instruction cycle time (see Figure 10).


MCLK
SC, WC, $\overline{LB}$ or $\overline{RB}$
$\overline{IV}_0$-$\overline{IV}_7$
(NOTE 1)
(NOTE 2)
(NOTE 3)

WF021060

Figure 8. Constraints of Am29X305 Instruction Cycle Time — Condition #1

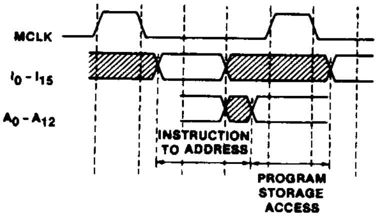
Notes: 1. MCLK to $\overline{LB}/\overline{RB}$ (input phase) or instruction to $\overline{LB}/\overline{RB}$ (input phase).
2. I/O port access (T_{IO}).
3. $\overline{IV}$ data setup time (referenced to MCLK).



WF021070

Figure 9. Constraints of Am29X305 Instruction Cycle Time — Condition #2

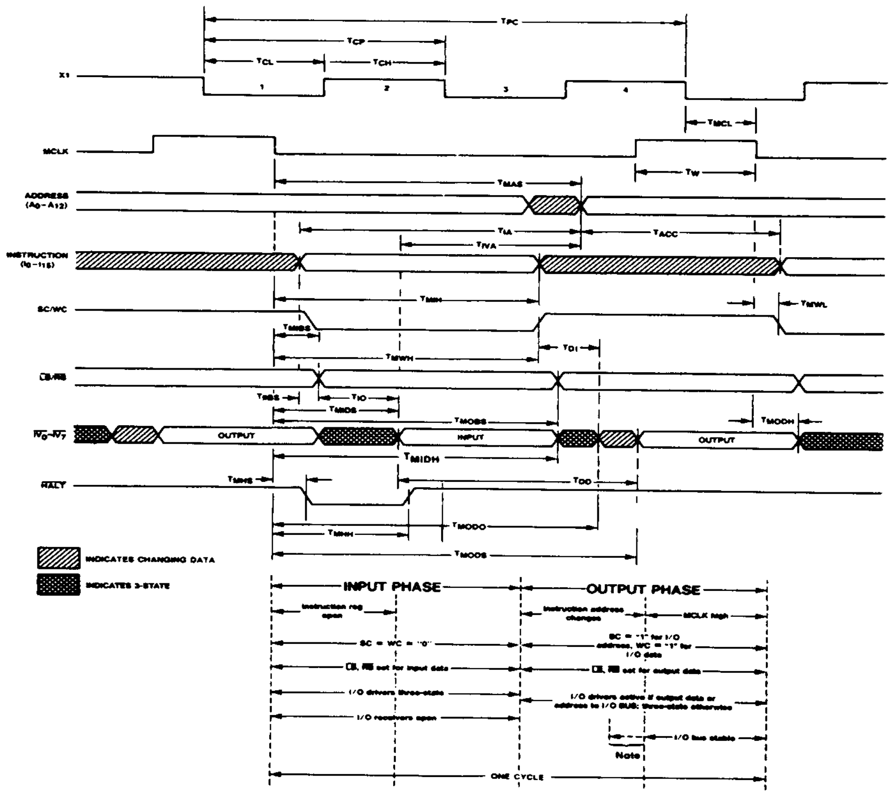
Notes:
1. Program Storage access time.
2. MCLK to $\overline{LB}/\overline{RB}$ (input phase) or instruction to LB/RB (input phase).
3. I/O port access (T_{IO})
4. $\overline{IV}$ data (input phase) to address.



WF021080

Figure 10. Constraints of Am29X305 Instruction Cycle Time — Condition #3

Am29X305 TIMING DIAGRAM



WF021030

Note: For an instruction cycle time greater than 250 ns, the I/O bus can be stable some time within the third quarter (T_{3Q}) cycle.

From Conditions 1, 2 and 3, one can calculate the minimum instruction cycle period as follows:

For Condition #1: $T_{MIBS} + T_{IO} \leqslant T_{MIDS}$
Substituting $40\text{ ns} + 30\text{ ns} \leqslant (T_{1Q} + T_{2Q} - 55)$ ns
Solving for $T_{1Q} + T_{2Q} \geqslant (34 + 30 + 55)$ ns
$1/2\text{ cycle} \geqslant 119$ ns

Using the results of Condition #1, we can check Conditions #2 and #3 to verify the minimum instruction cycle period.

For Condition #2: $T_{ACC} + T_{IIBS} + T_{IO} + T_{IVA} \leqslant 250$ ns
Substituting $60 + 32 + 30 + 96 \leqslant 250$ ns
Verifying $218 \leqslant 250$ ns

For Condition #3: $T_{IA} + T_{ACC} \leqslant$ Instruction Cycle
Substituting $152.5 + 60 \leqslant 250$ ns
Verifying $212.5 \leqslant 250$ ns

The results verify that Condition #1 defines the minimum instruction cycle period.

It is important to note that, during the input phase, the beginning of a valid $\overline{LB}/\overline{RB}$ signal is determined by either the instruction to $\overline{LB}/\overline{RB}$ (T_{IIBS}) or the delay from the falling edge of MCLK to $\overline{LB}/\overline{RB}$ (T_{MIBS}). Assuming the instruction is valid at the falling edge of MCLK and adding the instruction-to-$\overline{LB}/\overline{RB}$ delay ($T_{IIBS} = 32$ ns), the $\overline{LB}/\overline{RB}$ signal will be valid 32 ns after the falling edge of MCLK. With a fast Program Storage memory and with a valid instruction before the falling edge of MCLK — the $\overline{LB}/\overline{RB}$ signal will, due to the T_{MIBS} delay, still be valid 32 ns after the falling edge of MCLK.

Using a worst-case instruction cycle time of 250 ns, the user cannot gain a speed advantage by selecting a memory with faster access time. Under the same conditions, a speed advantage cannot be obtained by using an I/O port with fast access time (T_{IO}) because the address bus will be stable 66 ns (T_{AS}) after the beginning of the third quarter cycle — no matter how early the $\overline{IV}$ data input is valid.

Clock Considerations

The on-chip oscillator and timing-generation circuits of the Am29X305 can be controlled by any one of the following methods:

Capacitor — if timing is not critical,

Crystal — if precise timing is required, or

External Drive — if application requires that the Am29X305 be driven from a system clock.

Capacitor Timing: A non-polarized ceramic or mica capacitor with a working voltage equal to or greater than 25 volts is recommended. The lead lengths of capacitor should be approximately the same and as short as possible. Also, the timing circuits should not be in close proximity to external sources of noise. For various capacitor (C_X) values, the cycle time can be approximated as follows:

C_X (in pF)	Approximate Cycle Time
100	300 ns
200	500 ns
500	1.1 μs
1000	2.0 μs

Crystal Timing: When a crystal is used, the on-chip oscillator operates at the resonant frequency (f_O) of the crystal. The series-resonant quartz crystal connects to the Am29X305 via pins 10 (X_1) and 11 (X_2). The lead lengths of the crystal should be approximately equal and as short as possible. Also, the timing circuits should not be in close proximity to external sources of noise. The crystal should be hermetically sealed (HC type can) and have the following electrical characteristics:

Type: Fundamental mode, series resonant

Impedance at Fundamental: 35 ohms maximum

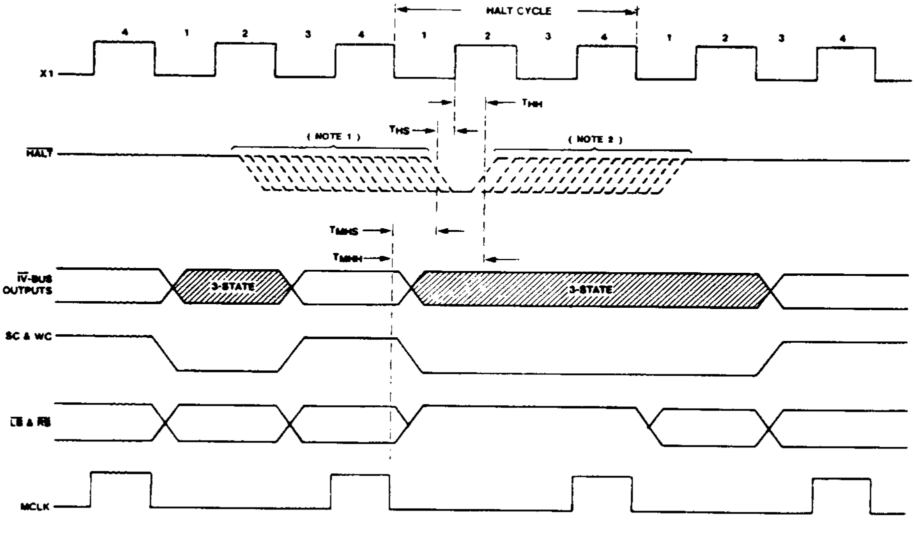
Impedance at Harmonics and Spurs: 50 ohms minimum

The resonant frequency (f_O) of the crystal is related to the desired cycle time (T) by the equation: $f_O = 2/T$. Therefore, for a cycle time of 250 ns, $f_O = 8$ MHz.

$\overline{\text{HALT}}$ Logic

The $\overline{\text{HALT}}$ signal is sampled via internal chip logic at the end of the first internal quarter of each instruction cycle. If, when sampled, the $\overline{\text{HALT}}$ signal is active LOW, a halt is immediately executed and the current instruction cycle is terminated. However, the halt cycle does not inhibit MCLK, nor does it affect any internal registers of the Am29X305. As long as the $\overline{\text{HALT}}$ line is active LOW, the SC and WC lines are LOW (inactive), the Left Bank ($\overline{LB}$)/Right Bank ($\overline{RB}$) signals are HIGH, and the $\overline{IV}$ bus remains in the three-state mode of operation. Normal operation resumes at the next cycle in which $\overline{\text{HALT}}$ is HIGH when sampled (see $\overline{\text{HALT}}$ Timing Diagram).

$\overline{\text{HALT}}$ TIMING DIAGRAM



Notes: 1. The $\overline{\text{HALT}}$ signal can switch from HIGH-to-LOW at any time during this interval.
2. The $\overline{\text{HALT}}$ signal can switch from LOW-to-HIGH at any time during this interval.

Timing Descriptions:

T_{HS} = Setup time from $\overline{\text{HALT}}$ to X1 (independent of instruction cycle time)
T_{HH} = Hold time from X1 to $\overline{\text{HALT}}$ (independent of instruction cycle time)
T_{MHS} = Setup time from MCLK to $\overline{\text{HALT}}$ (dependent upon instruction cycle time)
T_{MHH} = Hold time from MCLK to $\overline{\text{HALT}}$ (dependent upon instruction cycle time)

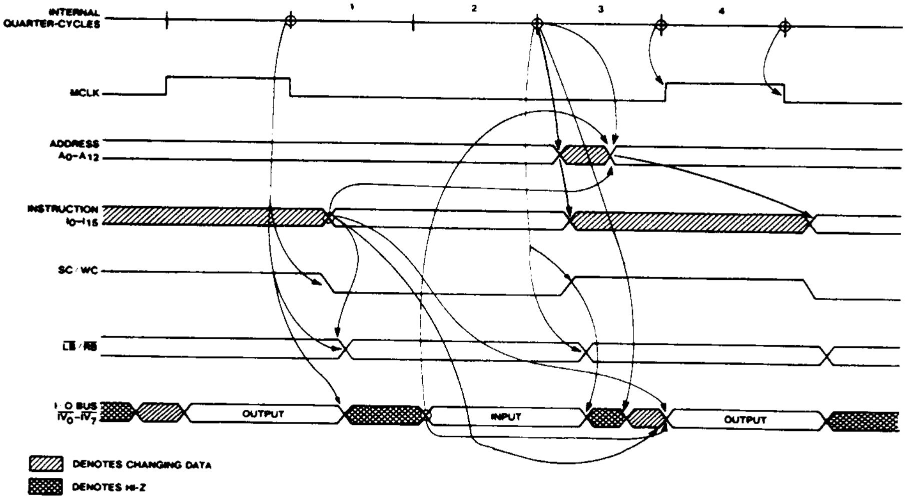
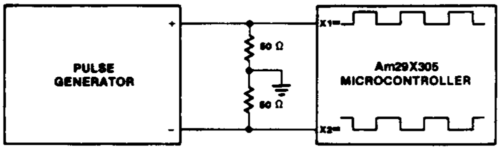


Figure 11. Timing Relationships of Am29X305 I/O Signals

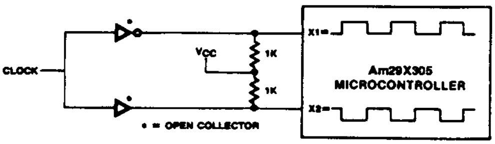
Using an External Clock: The Am29X305 can be synchronized with an external clock by simply connecting appropriate drive circuits to the X1/X2 inputs. Figure 12 shows how the on-chip oscillator can be driven from the complementary outputs of a pulse generator. In applications where the Microcontroller must be driven from a master clock, the X1/X2 lines can be interfaced to TTL logic as shown in Figure 13.



TC002991

Pulse Generator Characteristics:
$Z_{OUT} = 50\ \Omega$ $V_{OUT} = 0 - 1$ V
Risetime $\leqslant$ 10 ns Skew $\leqslant$ 10 ns

Figure 12. Clocking with a Pulse Generator



TC003001

TTL Driver Characteristics:
Fall Time $\leqslant$ 10 ns
Skew between complementary outputs $\leqslant$ 10 ns

Figure 13. Clocking with TTL

$\overline{\text{RESET}}$ Logic

$\overline{\text{RESET}}$ (pin 43) can be driven from a HIGH (inactive) state to a LOW (active) state at any time with respect to the system clock; that is, the reset function is asynchronous. To ensure proper operation, $\overline{\text{RESET}}$ must be held LOW (active) for one full instruction time. When the line is driven from a HIGH state to an active LOW state, several events occur — the precise instant of occurrence is basically a function of the propagation delay for that particular event. As shown in the $\overline{\text{RESET}}$ Timing Diagram, these events are:

- The Program Counter and Address Register are set to address zero and remain in that state as long as the $\overline{\text{RESET}}$ line is LOW. Other than PC and AR, $\overline{\text{RESET}}$ does not affect other internal registers.

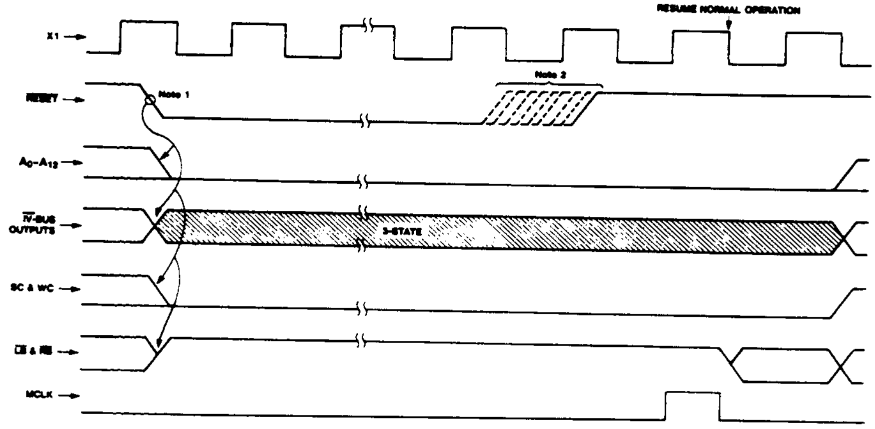
- The Input/Output bus ($\overline{\text{IV}}$) goes three-state and remains in that condition as long as the $\overline{\text{RESET}}$ line is LOW.

- The Select Command (SC) and Write Command (WC) signals are driven LOW and remain LOW as long as the $\overline{\text{RESET}}$ line is LOW.

- The Left Bank/Right Bank ($\overline{\text{LB}}/\overline{\text{RB}}$) signals are forced HIGH asynchronously for the period in which the $\overline{\text{RESET}}$ line is LOW.

During the time $\overline{\text{RESET}}$ is active LOW, MCLK is inhibited. Moreover, if the $\overline{\text{RESET}}$ line is driven LOW during the last two quarter cycles, MCLK may be shortened for that particular machine cycle. When the $\overline{\text{RESET}}$ line is driven HIGH (inactive) — one quarter to one full instruction cycle later, MCLK appears just before normal operation is resumed. The $\overline{\text{RESET}}$/MCLK relationship is clearly shown by "B" in the timing diagram. As long as the $\overline{\text{RESET}}$ line is active LOW, the $\overline{\text{HALT}}$ signal is not sampled by internal logic of the Am29X305.

$\overline{\text{RESET}}$ TIMING DIAGRAM

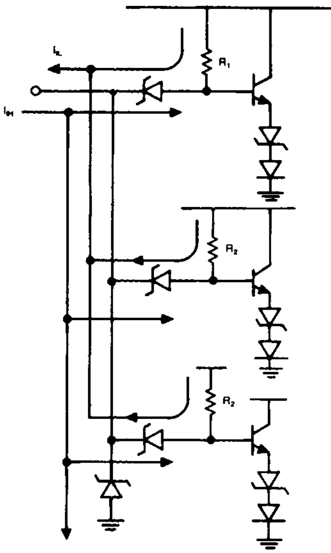


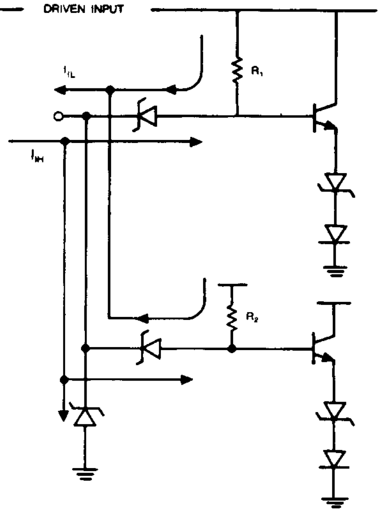
WF021090

Notes: 1. A HIGH-to-LOW transition of the $\overline{\text{RESET}}$ signal will force the Address Bus to an all-zero configuration.

2. The $\overline{\text{RESET}}$ signal can switch from LOW-to-HIGH at any point within this time interval and, in all cases, MCLK will occur at least one quarter cycle time later, as shown.

INPUT/OUTPUT CURRENT DIAGRAMS

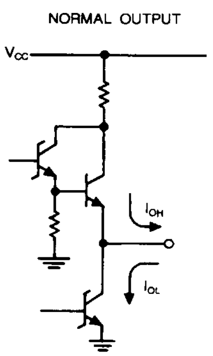


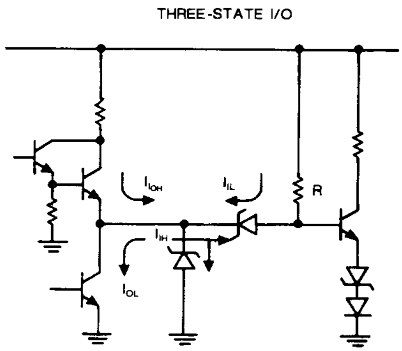


IC000761

I_0, I_3 Inputs
(R_1 = 8K, R_2 = 15K)

I_1, I_2, I_4,
I_{11} and I_{12} Inputs
(R_1 = 12K, R_2 = 15K)



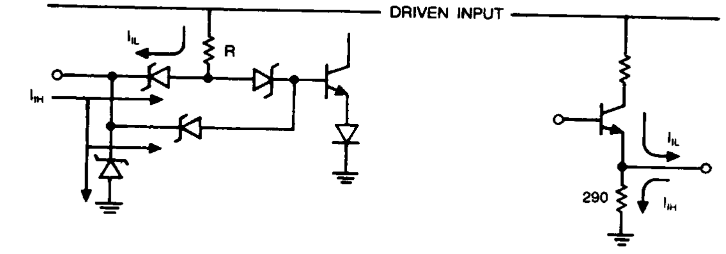


IC000771

MCLK, $A_0 - A_{12}$, WC, SC, $\overline{RB}$ and $\overline{LB}$
Inputs

$\overline{IV}_0 - \overline{IV}_7$ Inputs/Outputs
(R = 30K)

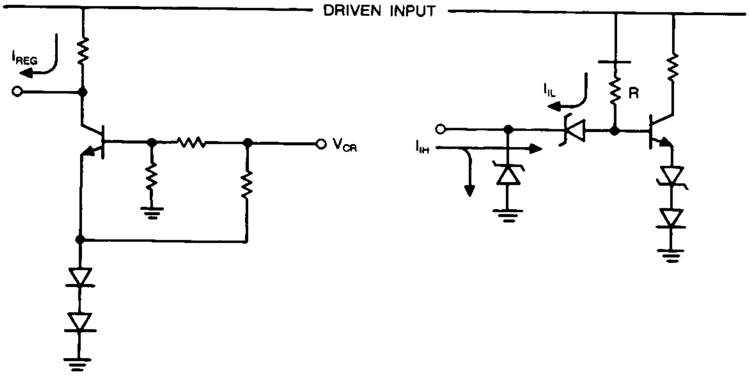
INPUT/OUTPUT CURRENT DIAGRAMS (Cont'd.)



IC000741

$\overline{\text{RESET}}/\overline{\text{HALT}}$ Inputs
(R = 20K)

X1/X2 Inputs

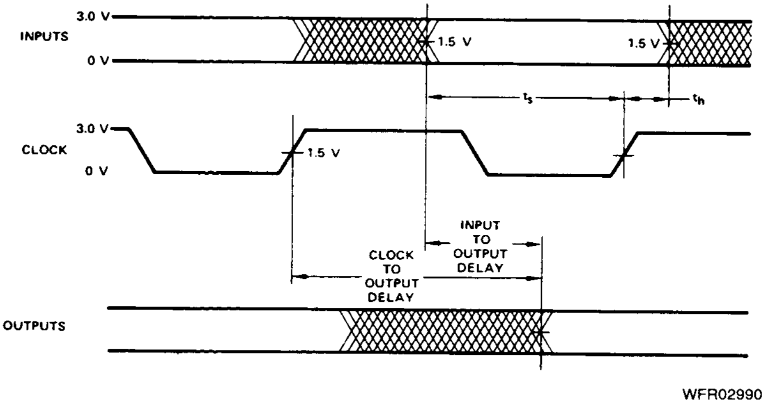


IC000751

V_{CR} Input

$I_5 - I_{10}$ and $I_{13} - I_{15}$
Inputs (R = 12K)

SWITCHING WAVEFORM



Notes on Test Methods

The following points give the general philosophy which we apply to tests which must be properly engineered if they are to be implemented in an automatic environment. The specifics of what philosophies applied to which test are shown.

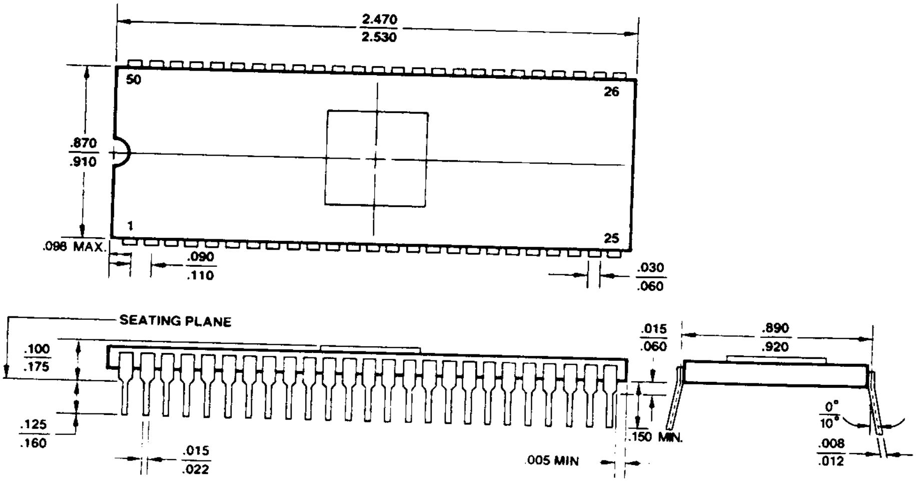
1. Ensure that the part is adequately decoupled at the test head. Large changes in supply current when the device switches may cause function failures due to V_{CC} changes.
2. Do not leave inputs floating during any tests, as they may oscillate at high frequency.
3. Do not attempt to perform threshold tests at high speed. Following an input transition, ground current may change by as much as 400 mA in 5 to 8 ns. Inductance in the ground cable may allow the ground pin at the device to rise by hundreds of millivolts momentarily.
4. Use extreme care in defining input levels for AC tests. Many inputs may be changed at once, so there will be significant noise at the device pins which may not actually reach V_{IL} or V_{IH} until the noise has settled. AMD recommends using $V_{IL} \leqslant 0$ V and $V_{IH} \leqslant 3$ V for AC tests.
5. To simplify failure and analysis, programs should be designed to perform DC, Function, and AC tests as three distinct groups of tests.
6. To assist in testing, AMD offers complete documentation on our test procedures and, in most cases, can provide actual Sentry programs, under license from Sentry.
7. Capacitive Loading for AC Testing: Automatic testers and their associated hardware have stray capacitance which varies from one type of tester to another, but generally around 50 pF. This, of course, makes it impossible to make direct measurements of parameters which call for a smaller capacitive load than the associated stray capacitance. Typical examples of this are the so-called "float delays," which measure the propagation delays in to and out of the high-impedance state, and are usually specified at a load capacitance of 5.0 pF. In these cases the test is performed at the higher load capacitance (typically 50 pF), and engineering correlations based on data taken with a bench set up are used to predict the result at the lower capacitance.

   Similarly, a product may be specified at more than one capacitive load. Since the typical automatic tester is not capable of switching loads in mid-test, it is impossible to make measurements at both capacitances even though they may both be greater than the stray capacitance. In these cases, a measurement is made at one of the two capacitances. The result at the other capacitance is predicted from engineering correlations based on data taken with a bench set up and the knowledge that certain DC measurements (e.g., I_{OH}, I_{OL}) have already been taken and are within specification. In some cases, special DC tests are performed in order to facilitate this correlation.
8. Threshold Testing: The noise associated with automatic testing, the long, inductive cables, and the high gain of bipolar devices when in the vicinity of the actual device threshold, frequently give rise to oscillations when testing high-speed circuits. These oscillations are not indicative of a reject device, but instead, of an overtaxed test system. To minimize this problem, thresholds are tested at least once for each input pin. Thereafter, "hard" high and low levels are used for other tests. Generally this means that function and AC testing are performed at "hard" input levels rather than at V_{IL} Max. and V_{IH} Min.
9. AC Testing: Occasionally, parameters are specified which cannot be measured directly on automatic testers because of tester limitations. Data input hold times often fall into this category. In these cases, the parameter in question is guaranteed by correlating tests with other AC tests which have been performed. These correlations are arrived at by the cognizant engineer by using data from precise bench measurements in conjunction with the knowledge that certain DC parameters have already been measured and are within specification.

   In some cases, certain AC tests are redundant since they can be shown to be predicted by other tests which have already been performed. In these cases, the redundant tests are not performed.

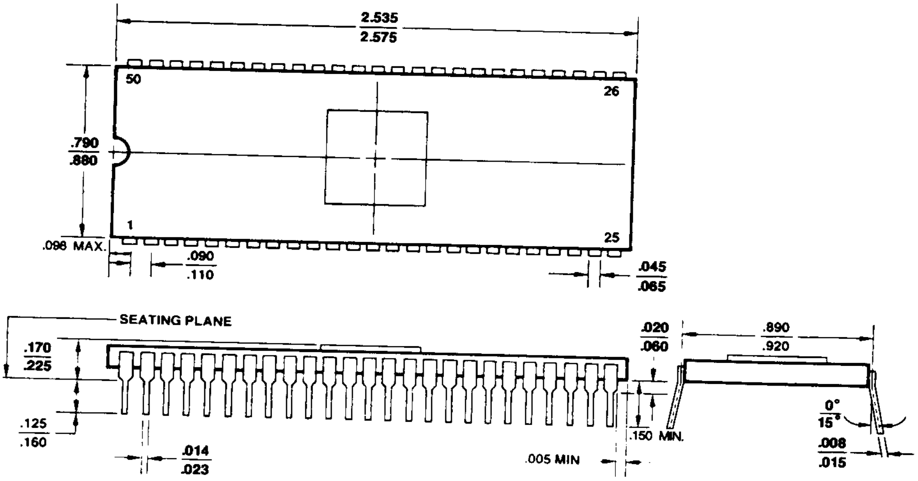
PHYSICAL DIMENSIONS

SD 050



PID # 07644A

PD 050



PID # 07732A

ADVANCED MICRO DEVICES
DOMESTIC SALES OFFICES

ALABAMA	(205) 882-9122
ARIZONA,	
Tempe	(602) 242-4400
Tucson	(602) 792-1200
CALIFORNIA,	
El Segundo	(213) 640-3210
Newport Beach	(714) 752-6262
San Diego	(619) 560-7030
Sunnyvale	(408) 720-8811
Woodland Hills	(818) 992-4155
COLORADO	(303) 691-5100
CONNECTICUT,	
Southbury	(203) 264-7800
FLORIDA,	
Altamonte Springs	(305) 339-5022
Clearwater	(813) 530-9971
Ft. Lauderdale	(305) 484-8600
Melbourne	(305) 254-2915
GEORGIA	(404) 449-7920
ILLINOIS	(312) 773-4422
INDIANA	(317) 244-7207
KANSAS	(913) 451-3115
MARYLAND	(301) 796-9310
MASSACHUSETTS	(617) 273-3970
MINNESOTA	(612) 938-0001
NEW JERSEY	(201) 299-0002
NEW YORK,	
Liverpool	(315) 457-5400
Poughkeepsie (IBM only)	(914) 471-8180
Woodbury	(516) 364-8020
NORTH CAROLINA,	
Charlotte	(704) 525-1875
Raleigh	(919) 847-8471
OREGON	(503) 245-0080
OHIO,	
Columbus	(614) 891-6455
PENNSYLVANIA,	
Allentown (AT&T only)	(215) 398-8006
Willow Grove	(215) 657-3101
TEXAS,	
Austin	(512) 346-7830
Dallas	(214) 934-9099
Houston	(713) 785-9001
WASHINGTON	(206) 455-3600
WISCONSIN	(414) 782-7748

INTERNATIONAL SALES OFFICES

BELGIUM,		
Bruxelles	TEL:	(02) 771 99 93
	FAX:	762-3716
	TLX:	61028
CANADA, Ontario,		
Kanata	TEL:	(613) 592-0090
Willowdale	TEL:	(416) 224-5193
	FAX:	(416) 224-0056
FRANCE,		
Paris	TEL:	(01) 46 87 36 66
	FAX:	(01) 46 86 21 85
	TLX:	202053F
GERMANY,		
Hannover area	TEL:	(05143) 50 55
	FAX:	5553
	TLX:	925287
München	TEL:	(089) 41 14-0
	FAX:	406490
	TLX:	523883
Stuttgart	TEL:	(0711) 62 33 77
	FAX:	625187
	TLX:	721882
HONG KONG,		
Kowloon	TEL:	3-695377
	FAX:	1234276
	TLX:	50426
ITALY, Milano	TEL:	(02) 3390541
	FAX:	3498000
	TLX:	315286
JAPAN, Tokyo	TEL:	(03) 345-8241
	FAX:	3425196
	TLX:	J24064 AMDTKOJ
LATIN AMERICA,		
Ft. Lauderdale,	TEL:	(305) 484-8600
	FAX:	(305) 485-9736
SWEDEN, Stockholm	TEL:	(08) 733 03 50
	FAX:	7332285
	TLX:	11602
UNITED KINGDOM,		
Manchester area	TEL:	(0925) 828008
	FAX:	827693
	TLX:	628524
London area	TEL:	(04862) 22121
	FAX:	22179
	TLX:	859103

NORTH AMERICAN REPRESENTATIVES

CALIFORNIA	
I^2 INC	OEM (408) 988-3400
	DISTI (408) 498-6868
CONNECTICUT	
SCIENTIFIC COMPONENTS	(203) 272-2963
IDAHO	
INTERMOUNTAIN TECH MKGT	(208) 322-5022
INDIANA	
SAI MARKETING CORP	(317) 241-9276
IOWA	
LORENZ SALES	(319) 377-4666
MICHIGAN	
SAI MARKETING CORP	(313) 227-1786
NEBRASKA	
LORENZ SALES	(402) 475-4660
NEW JERSEY	
TAI CORPORATION	(609) 933-2600
NEW MEXICO	
THORSON DESERT STATES	(505) 293-8555
NEW YORK	
NYCOM, INC	(315) 437-8343
OHIO	
Dayton	
DOLFUSS ROOT & CO	(513) 433-6776
Strongsville	
DOLFUSS ROOT & CO	(216) 238-0300
PENNSYLVANIA	
DOLFUSS ROOT & CO	(412) 221-4420
UTAH	
R^2 MARKETING	(801) 595-0631

Advanced Micro Devices reserves the right to make changes in its product without notice in order to improve design or performance characteristics. The performance characteristics listed in this document are guaranteed by specific tests, correlated testing, guard banding, design and other practices common to the industry. For specific testing details, contact your local AMD sales representative. The company assumes no responsibility for the use of any circuits described herein.

ADVANCED MICRO DEVICES 901 Thompson Pl., P.O. Box 3453, Sunnyvale, CA 94088, USA
TEL: (408) 732-2400 ● TWX: 910-339-9280 ● TELEX: 34-6306 ● TOLL FREE: (800) 538-8450

Printed in U.S.A. AIS-WCP-15M-3/86-0