

Am29C331

CMOS 16-Bit Microprogram Sequencer



Am29C331

PRELIMINARY

DISTINCTIVE CHARACTERISTICS

- **16-Bits Address up to 64K Words**
Supports 110-ns microcycle time for a 32-bit high-performance system when used with the other members of the Am29C300 Family.
- **Speed Select**
Supports 80-ns system cycle time.
- **Real-Time Interrupt Support**
Micro-trap and interrupts are handled transparently at any microinstruction boundary.
- **Built-In Conditional Test Logic**
Has twelve external test inputs, four of which are used to internally generate an additional four test conditions. Test multiplexer selects one out of 16 test inputs.
- **Break-Point Logic**
Built-in address comparator allows break-points in the microcode for debugging and statistics collection.
- **Master/Slave Error Checking**
Two sequencers can operate in parallel as a master and a slave. The slave generates a fault flag for unequal results.
- **33-Level Stack**
Provides support for interrupts, loops, and subroutine nesting. It can be accessed through the D-bus to support diagnostics.

GENERAL DESCRIPTION

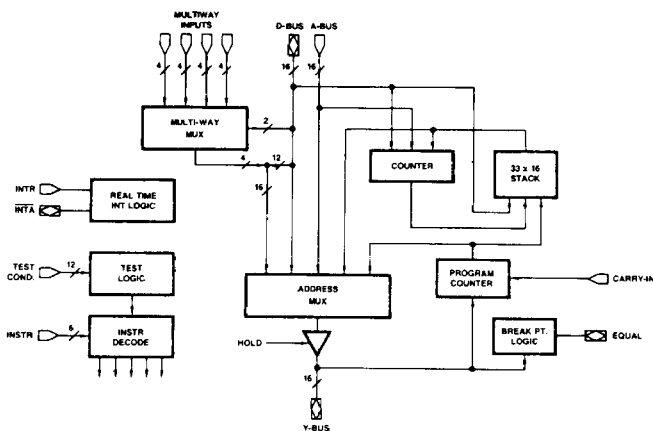
The Am29C331 is a 16-bit wide, high-speed single-chip sequencer designed to control the execution sequence of microinstructions stored in the microprogram memory. The instruction set is designed to resemble high-level language constructs, thereby bringing high-level language programming to the micro level.

The Am29C331 is interruptible at any microinstruction boundary to support real-time interrupts. Interrupts are handled transparently to the microprogrammer as an unexpected procedure call. Traps are also handled transparently at any microinstruction boundary. This feature allows re-execution of the prior microinstruction. Two separate buses are provided to bring a branch address directly into the chip from two sources to avoid slow turn-on and turn-off times for different sources connected to the data-input bus. Four

sets of multiway inputs are also provided to avoid slow turn-on and turn-off times for different branch-address sources. This feature allows implementation of table look-up or use of external conditions as part of a branch address. The 33-deep stack provides the ability to support interrupts, loops, and subroutine nesting. The stack can be read through the D-bus to support diagnostics or to implement multitasking at the micro-architecture level. The master/slave mode provides a complete function check capability for the device.

Fabricated using Advanced Micro Devices' 1.6 micron CMOS process, the Am29C331 is powered by a single 5-volt supply. The device is housed in a 120-terminal pin-grid array package.

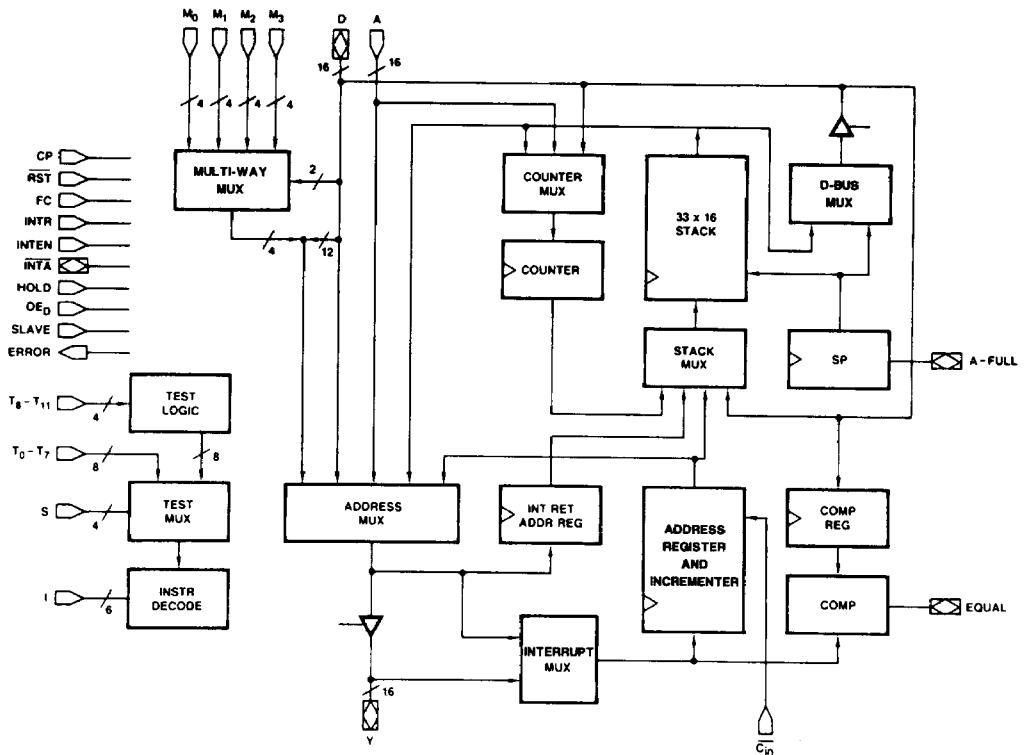
SIMPLIFIED BLOCK DIAGRAM



BD006091

RELATED AMD PRODUCTS

Part No.	Description
Am29114	Vectored Priority Interrupt Controller
Am29116	High-Performance Bipolar 16-Bit Microprocessor
Am29C116	High-Performance CMOS 16-Bit Microprocessor
Am29PL141	Field-Programmable Controller
Am29C323	CMOS 32-Bit Parallel Multiplier
Am29325	32-Bit Floating-Point Processor
Am29C325	CMOS 32-Bit Floating-Point Processor
Am29332	32-Bit Extended Function ALU
Am29C332	CMOS 32-Bit Extended Function ALU
Am29334	64 x 18 Four-Port, Dual-Access Register File
Am29C334	CMOS 64 x 18 Four-Port Dual-Access Register File
Am29337	16-Bit Bounds Checker
Am29338	Byte Queue



BD006102

Figure 1. Am29C331 Detailed Block Diagram

CONNECTION DIAGRAM

120-Lead PGA*

	A	B	C	D	E	F	G	H	J	K	L	M	N
1	M0,0	M1,0	M2,0	M2,1	\overline{CTN}	M1,2	M1,3	M2,3	GND	\overline{RST}	INTR	SLAVE	D15
2	D0	A0	M3,0	M1,1	M0,2	M2,2	M0,3	M3,3	EQUAL	OED	INTEN	HOLD	A15
3	VCC	Y0	D1	M0,1	M3,1	GND	M3,2	VCC	A-FULL	ERROR	\overline{INTA}	Y15	VCC
4	A1	Y1	D2								D14	A14	Y14
5	GND	A2	Y2								D13	A13	GND
6	A3	D3	GND								GND	D12	Y13
7	Y3	D4	A4								A12	Y12	D11
8	D5	Y4	VCC								VCC	Y11	A11
9	GND	A5	Y5								D10	A10	GND
10	D6	A6	Y6								Y10	D9	A9
11	VCC	D7	T3	T6	GND	T10	T11	I0	VCC	I3	Y9	D8	VCC
12	A7	T1	T2	T5	GND	T7	S0	S1	VCC	I2	I4	A8	Y8
13	Y7	T0	T9	T4	GND	T8	CP	S3	VCC	I1	S2	I5	FC

CD010380

*Pins facing up.

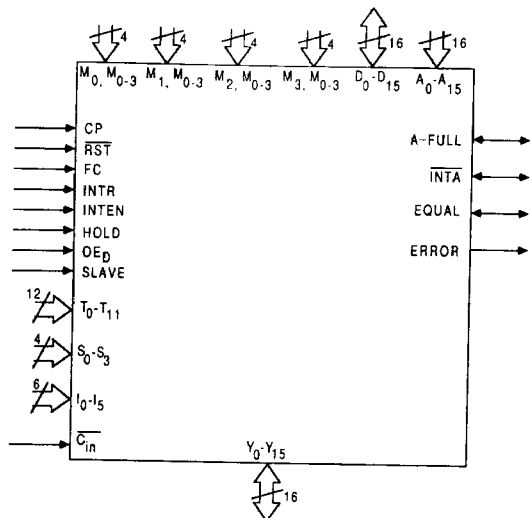
PIN DESIGNATIONS
(Sorted by Pin No.)

PIN NO.	PIN NAME	PAD NO.	PIN NO.	PIN NAME	PAD NO.	PIN NO.	PIN NAME	PAD NO.	PIN NO.	PIN NAME	PAD NO.
			C-5	Y ₂	115	H-2	M _{3, 3}	10	M-5	A ₁₃	80
			C-6	GND	113	H-3	VCC	68	M-6	D ₁₂	81
			C-7	A ₄	52	H-11	I ₀	34	M-7	Y ₁₂	82
			C-8	VCC	53	H-12	S ₁	95	M-8	Y ₁₁	25
A-1	M _{0, 0}	1	C-9	Y ₅	109	H-13	S ₃	94	M-9	A ₁₀	86
A-2	D ₀	120	C-10	Y ₆	48	J-1	GND	11	M-10	D ₉	87
A-3	VCC	59	C-11	T ₃	44	J-2	EQUAL	71	M-11	D ₈	89
A-4	A ₁	58	C-12	T ₂	104	J-3	A-FULL	70	M-12	A ₈	30
A-5	GND	56	C-13	T ₉	41	J-11	VCC	37	M-13	I ₅	91
A-6	A ₃	114	D-1	M _{2, 1}	4	J-12	VCC	38	N-1	D ₁₅	16
A-7	Y ₃	54	D-2	M _{1, 1}	63	J-13	VCC	39	N-2	A ₁₅	76
A-8	D ₅	51	D-3	M _{0, 1}	3	K-1	RST	13	N-3	VCC	17
A-9	GND	50	D-11	T ₆	102	K-2	OE _D	72	N-4	Y ₁₄	19
A-10	D ₆	49	D-12	T ₅	43	K-3	ERROR	12	N-5	GND	20
A-11	VCC	47	D-13	T ₄	103	K-11	I ₃	92	N-6	Y ₁₃	21
A-12	A ₇	106	E-1	C _{in}	5	K-12	I ₂	33	N-7	D ₁₁	24
A-13	Y ₇	46	E-2	M _{0, 2}	65	K-13	I ₁	93	N-8	A ₁₁	84
B-1	M _{1, 0}	61	E-3	M _{3, 1}	64	L-1	INTR	14	N-9	GND	26
B-2	A ₀	60	E-11	GND	97	L-2	INTEN	74	N-10	A ₉	28
B-3	Y ₀	119	E-12	GND	98	L-3	INTA	73	N-11	VCC	29
B-4	Y ₁	117	E-13	GND	99	L-4	D ₁₄	18	N-12	Y ₈	90
B-5	A ₂	116	F-1	M _{1, 2}	6	L-5	D ₁₃	79	N-13	FC	31
B-6	D ₃	55	F-2	M _{2, 2}	66	L-6	GND	23			
B-7	D ₄	112	F-3	GND	8	L-7	A ₁₂	22			
B-8	Y ₄	111	F-11	T ₁₀	100	L-8	VCC	83			
B-9	A ₅	110	F-12	T ₇	42	L-9	D ₁₀	85			
B-10	A ₆	108	F-13	T ₈	101	L-10	Y ₁₀	27			
B-11	D ₇	107	G-1	M _{1, 3}	9	L-11	Y ₉	88			
B-12	T ₁	45	G-2	M _{0, 3}	67	L-12	I ₄	32			
B-13	T ₀	105	G-3	M _{3, 2}	7	L-13	S ₂	35			
C-1	M _{2, 0}	2	G-11	T ₁₁	40	M-1	SLAVE	75			
C-2	M _{3, 0}	62	G-12	S ₀	36	M-2	HOLD	15			
C-3	D ₁	118	G-13	CP	96	M-3	Y ₁₅	77			
C-4	D ₂	57	H-1	M _{2, 3}	69	M-4	A ₁₄	78			

PIN DESIGNATIONS
(Sorted by Pin Name)

PIN NAME	PIN NO.	PAD NO.	PIN NAME	PIN NO.	PAD NO.	PIN NAME	PIN NO.	PAD NO.	PIN NAME	PIN NO.	PAD NO.
-	-	37	D ₈	M-11	89	INTEN	L-2	74	T ₆	D-11	102
-	-	39	D ₉	M-10	87	INTR	L-1	14	T ₇	F-12	42
-	-	97	D ₁₀	L-9	85	M _{0, 0}	A-1	1	T ₈	F-13	101
-	-	99	D ₁₁	N-7	24	M _{0, 1}	D-3	3	T ₉	C-13	41
A-FULL	J-3	70	D ₁₂	M-6	81	M _{0, 2}	E-2	65	T ₁₀	F-11	100
A ₀	B-2	60	D ₁₃	L-5	79	M _{0, 3}	G-2	67	T ₁₁	G-11	40
A ₁	A-4	58	D ₁₄	L-4	18	M _{1, 0}	B-1	61	GND	J-1	11
A ₂	B-5	116	D ₁₅	N-1	16	M _{1, 1}	D-2	63	GND	N-5	20
A ₃	A-6	114	GND	E-12	97	M _{1, 2}	F-1	6	GND	A-9	50
A ₄	C-7	52	GND	E-13	98	M _{1, 3}	G-1	9	GND	N-9	26
A ₅	B-9	110	GND	E-11	99	M _{2, 0}	C-1	2	GND	A-5	56
A ₆	B-10	108	GND	F-3	8	M _{2, 1}	D-1	4	VCC	N-3	17
A ₇	A-12	106	GND	L-6	23	M _{2, 2}	F-2	66	VCC	N-11	29
A ₈	M-12	30	GND	C-6	113	M _{2, 3}	H-1	69	VCC	A-3	59
A ₉	N-10	28	VCC	J-13	38	M _{3, 0}	C-2	62	VCC	A-11	47
A ₁₀	M-9	86	VCC	H-3	68	M _{3, 1}	E-3	64	Y ₀	B-3	119
A ₁₁	N-8	84	VCC	C-8	53	M _{3, 2}	G-3	7	Y ₁	B-4	117
A ₁₂	L-7	22	VCC	L-8	83	M _{3, 3}	H-2	10	Y ₂	C-5	115
A ₁₃	M-5	80	VCC	J-12	37	OED	K-2	72	Y ₃	A-7	54
A ₁₄	M-4	78	VCC	J-11	39	RST	K-1	13	Y ₄	B-8	111
A ₁₅	N-2	76	EQUAL	J-2	71	S ₀	G-12	36	Y ₅	C-9	109
C _{in}	E-1	5	ERROR	K-3	12	S ₁	H-12	95	Y ₆	C-10	48
CP	G-13	96	FC	N-13	31	S ₂	L-13	35	Y ₇	A-13	46
D ₀	A-2	120	HOLD	M-2	15	S ₃	H-13	94	Y ₈	N-12	90
D ₁	C-3	118	I ₀	H-11	34	SLAVE	M-1	75	Y ₉	L-11	88
D ₂	C-4	57	I ₁	K-13	93	T ₀	B-13	105	Y ₁₀	L-10	27
D ₃	B-6	55	I ₂	K-12	33	T ₁	B-12	45	Y ₁₁	M-8	25
D ₄	B-7	112	I ₃	K-11	92	T ₂	C-12	104	Y ₁₂	M-7	82
D ₅	A-8	51	I ₄	L-12	32	T ₃	C-11	44	Y ₁₃	N-6	21
D ₆	A-10	49	I ₅	M-13	91	T ₄	D-13	103	Y ₁₄	N-4	19
D ₇	B-11	107	INTA	L-3	73	T ₅	D-12	43	Y ₁₅	M-3	77

LOGIC SYMBOL



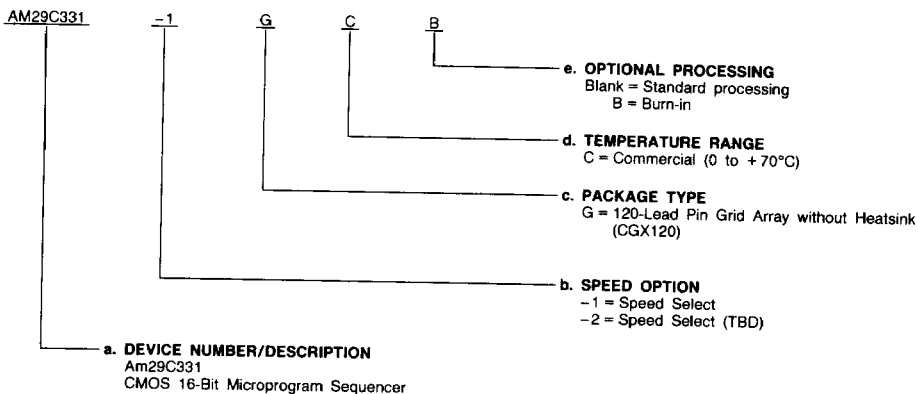
LS002872

ORDERING INFORMATION

Standard Products

AMD standard 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**



Valid Combinations

Valid Combinations	
AM29C331	GC, GCB
AM29C331-1	

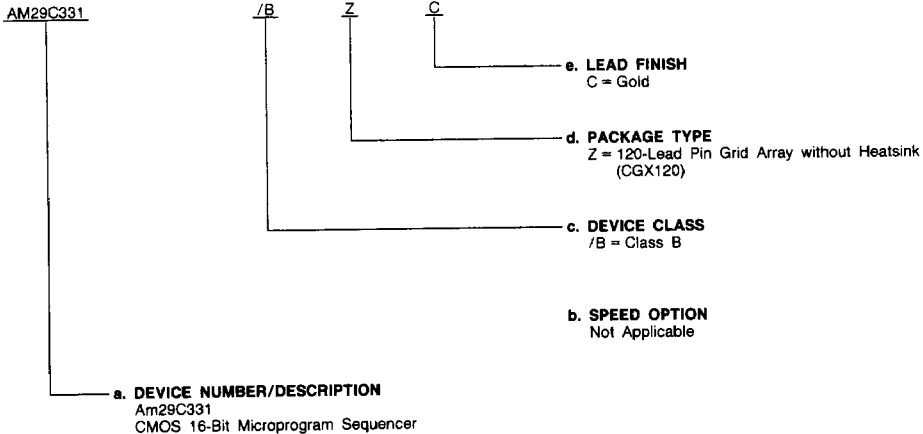
Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

MILITARY ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of:

- a. **Device Number**
- b. **Speed Option** (if applicable)
- c. **Device Class**
- d. **Package Type**
- e. **Lead Finish**



Valid Combinations	
AM29C331	/BZC

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 or to check for newly released valid combinations.

Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

A₀ - A₁₅ Alternate Data (Input)

Input to address multiplexer and counter.

A-FULL Almost Full (Bidirectional; Three-State)

Indicates that $28 \leq SP \leq 63$ (meaning there are five or less empty locations left on stack). Also active during stack underflow.

C_{in} Carry In (Input, Active LOW)

Carry-in to the incrementer.

CP Clock Pulse (Input)

Clocks sequencer at the LOW-to-HIGH transition.

D₀ - D₁₅ Data (Bidirectional, Three-State)

Input to address multiplexer, counter, stack, and comparator register. Output for stack and stack pointer.

EQUAL Equal (Bidirectional, Three-State)

Indicates that the address comparator is enabled and has found a match.

ERROR Error (Output)

Indicates a master/slave error in the slave mode. Indicates a malfunctioning driver or contention of any output in the master mode.

FC Force Continue (Input)

Overrides instruction with CONTINUE.

HOLD Hold (Input)

Stops the sequencer and three-states the outputs.

I₀ - I₅ Instruction (Input)

Selects one of 64 instructions.

INTA Interrupt Acknowledge (Bidirectional; Three-State, Active LOW)

Indicates that an interrupt is accepted.

INTEN Interrupt Enable (Input)

Enables interrupts.

INTR Interrupt Request (Input)

Requests the sequencer to interrupt execution.

M₀₋₃, 0-3 Multiway (Input)

Four sets of multiway inputs providing 16-way branches. The first index refers to the set number.

OE_D Output Enable - D-Bus (Input)

Enables the D-bus driver, provided that the sequencer is not in the hold or slave mode.

RST Reset (Input; Active LOW)

Resets the sequencer.

S₀ - S₃ Select (Input)

Selects one of 16 test conditions.

SLAVE Slave (Input)

Makes the sequencer a slave.

T₀ - T₁₁ Test (Input)

Provides external test inputs.

Y₀ - Y₁₅ Address (Bidirectional; Three-State)

Output of microcode address. Input for interrupt address.

FUNCTIONAL DESCRIPTION

Architecture

The major blocks of the sequencer are the address multiplexer, the address register (AR), the stack (with the top of stack denoted TOS), the counter (C), the test multiplexer with logic, and the address comparison register (R) (Figure 1). The bidirectional D-bus provides branch addresses and iteration counts; it also allows access to the stack from the outside. The A-bus may be used for map addresses. There are four sets of four-bit multiway branch inputs (M). The bidirectional Y-bus either outputs microprogram addresses or inputs interrupt addresses. The buses are all 16 bits wide. Figure 1 shows a detailed block diagram of the sequencer.

Address Multiplexer

The address multiplexer can select an address from any of five sources:

- 1) A branch address supplied by the D-bus
- 2) A branch address supplied by the A-bus

3) A multiway-branch address

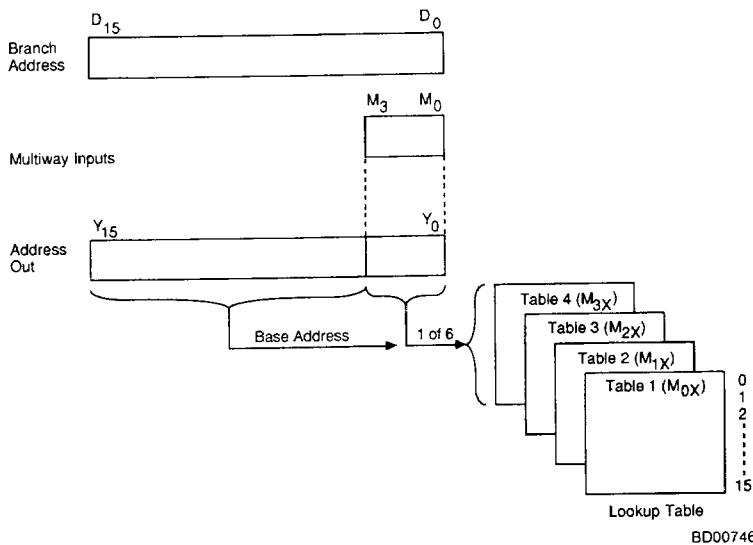
4) A return or loop address from the top of stack

5) The next sequential address from the incrementer

Multiway-Branch Address

A multiway-branch address is formed by substituting the lower four bits of the address on the D-bus (D₃, D₂, D₁, D₀) with one of the four sets (M_{0X}, M_{1X}, M_{2X}, or M_{3X}) of four-bit multiway-branch addresses. The multiway-branch set is selected by the number D₁D₀, while the bits D₃ and D₂ are "don't cares" (see Figure 2).

D ₁	D ₀	Multiway Set Selected
0	0	M _{0X}
0	1	M _{1X}
1	0	M _{2X}
1	1	M _{3X}



- Notes:
1. D₁ and D₀ select one out of four multiway sets. D₃ and D₂ are "don't cares."
 2. Each set of M_{3X} - M_{0X} can select one of sixteen locations. The multiway-branch address is the concatenation of D₁₅ - D₄ (base address) and M_{3X} - M_{0X}.
 3. For a given base address, there can be four look-up tables, each sixteen deep.

Figure 2. Multiway Branch

Address Register and Incrementer

The address register contains the current address. It is loaded from the interrupt multiplexer and feeds the incrementer. The incrementer is inhibited if \overline{CIN} is taken HIGH.

Stack

A 33-word-deep and 16 bit-wide stack provides first-in last-out storage for return addresses, loop addresses, and counter values. Items to be pushed come from the incrementer, the interrupt-return-address register, the counter, or the D-bus. Items popped go to the address multiplexer, the counter, or the D-bus.

The access to the stack via the D-bus may be used for context switching, stack extension, or diagnostics. As the stack is only accessible from the top, stack extension is done by temporarily storing the whole or some lower part of the stack outside the sequencer. The save and the later restore are done with pop and push operations, respectively, at balanced points in the microprogram; for example, points with the same stack depth. The internal D-bus driver must be turned on when popping an item to the D-bus; if the driver is off, the item will be unstacked instead. The driver is normally turned on when the Output Enable signal is asserted and the sequencer is not being reset ($OE_D = 1$, $RST = 1$).

The stack pointer is a modulo 64 counter, which is incremented on each push and decremented on each pop. The stack pointer is reset to zero when the sequencer is reset, but the pointer may also be reset by instruction. Thus, the stack pointer indicates the number of items on the stack as long as stack overflow or underflow has not occurred. Overflow happens when an item is pushed onto a full stack, whereby the item at the bottom of the stack is overwritten. Underflow

happens when an item is popped from an empty stack; in this case the item is undefined.

In the case of stack overflow, the SP is incremented for every push after overflow. Thus, immediately after the first occurrence of stack overflow, the SP will be equal to 34. Subsequent pushes will increment the SP to 35, 36 ... 61, 62, 63, 0, 1, etc. In the case of stack underflow, the SP is decremented for every pop after underflow. Thus, immediately after the first occurrence of stack underflow, the SP will be equal to 63. Subsequent pops will decrement the SP to 62, 61, ... 2, 1, 0, 63, etc.

The contents of the stack pointer are present on the D-bus for all instructions except POP D, provided the driver is turned on. The output signal, A-FULL, is active under the following condition: $28 \leq SP \leq 63$.

Counter

The counter may be used as a loop counter. It may be loaded from the D-bus, the A-bus, or via a pop from the stack. Its contents may also be pushed onto the stack.

A normal for-loop is set up by a FOR instruction, which loads the counter from the D- or A-bus with the desired number of iterations; the instruction also pushes onto the stack a loop address that points to the next sequential instruction. The end of the loop is given by an unconditional END FOR instruction, which tests the counter value against the value one and then decrements the counter. If the values differ, the loop is repeated by selecting the address at the stack as the next address. If the values are equal, the loop is terminated by popping the stack, thereby removing the loop address, and selecting the address from the incrementer as the next address. The number of iterations is a 16-bit unsigned number, except that the number zero corresponds to 65,536 iterations.

By pushing and popping counter values it is possible to handle nested loops.

Address Comparison

The sequencer is able to compare the address from the interrupt multiplexer with the contents of the comparator register. The instruction SET loads the comparator register with the address on the D-bus and enables the comparison, while CLEAR disables it. The comparison is disabled at reset. A HIGH is present at the output EQUAL if the comparison is enabled and the two addresses are equal. The comparison is useful for detection of a break point or counting the number of times a microinstruction at a specific address is executed.

Instruction Set

The sequencer has 64 instructions that are divided into four classes of 16 instructions each. The instruction lines $I_0 - I_5$ use I_5 and I_4 to select a class, and $I_0 - I_3$ to select an instruction within a class. The classes are:

I_5	I_4	Classes
0	0	Conditional sequence control,
0	1	Conditional sequence control with inverted polarity,
1	0	Unconditional sequence control, and
1	1	Special function with implicit continue.

Note that for the first three classes I_5 forces the condition to be true and I_4 inverts the condition. The basic instructions of the first three classes are shown in Table 1 and the instructions of the fourth class in Table 2.

Structured microprogramming is supported by sequencer instructions that singly or in pairs correspond to high-level language control constructs. Examples are FOR I: = D DOWN TO 1 DO . . . END FOR and CASE N OF . . . END CASE. The instructions have been given high-level language names where appropriate. Figure 2 shows how to microprogram important control constructs; the high-level language is on the left and the microcode on the right.

Test Conditions

The condition for a conditional instruction is supplied by a test multiplexer, which selects one out of sixteen tests with the select lines $S_0 - S_3$. Twelve of these are supplied directly by the inputs $T_0 - T_{11}$, while the remaining four tests are generated by the test logic from the inputs $T_8 - T_{11}$. The following table shows the assignments.

$(S_0 - S_3)_H$	Test	Intended Use
0-7	$T_0 - T_7$	General
8	T_8	C (Carry)
9	T_9	N (Negative)
A	T_{10}	V (Overflow)
B	T_{11}	Z (Zero or equal)
C	$T_8 + T_{11}$	C + Z (Unsigned less than or equal, borrow mode)
D	$\overline{T_8} + T_{11}$	$\overline{C} + Z$ (Unsigned less than or equal)
E	$T_9 \oplus T_{10}$	$N \oplus V$ (Signed less than)
F	$(T_9 \oplus T_{10}) + T_{11}$	$(N \oplus V) + Z$ (Signed less than or equal)

Force Continue

The sequencer has a force continue (FC) input, which overrides the instruction inputs $I_0 - I_5$ with a CONTINUE instruction. This makes it possible to share the microinstruction field for the sequencer instruction with some other control or to initialize a writable control store.

Reset

In order to start a microprogram properly, the sequencer must be reset. The reset works like an instruction overriding both the instruction input and the force continue input. The reset selects the address 0 at the address multiplexer, forces the EQUAL output to LOW, and disregards a potential interrupt request. It synchronously disables the address comparison and initializes the stack pointer to 0. The contents of the stack are invalid after a reset.

TABLE 1. INSTRUCTION SET for I₅I₄ = 00, 01, 10

I ₅ - I ₀	Instruction	Cond.: Fail		Cond.: Pass		Counter	Comp.	D-Mux
		Y	Stack	Y	Stack			
00, 10, 20	Goto D	INC	-	D	-	-	-	SP
01, 11, 21	Call D	INC	-	D	Push INC	-	-	SP
02, 12, 22	Exit D	INC	-	D	Pop	-	-	SP
03, 13, 23	End for D, C ≠ 1	INC	-	D	-	C←C-1	-	SP
	End for D, C = 1	INC	-	INC	-	C←C-1	-	SP
04, 14, 24	Goto A	INC	-	A	-	-	-	SP
05, 15, 25	Call A	INC	-	A	Push INC	-	-	SP
06, 16, 26	Exit A	INC	-	A	Pop	-	-	SP
07, 17, 27	End for A, C ≠ 1	INC	-	A	-	C←C-1	-	SP
	End for A, C = 1	INC	-	INC	-	C←C-1	-	SP
08, 18, 28	Goto M	INC	-	D:M	-	-	-	SP
09, 19, 29	Call M	INC	-	D:M	Push INC	-	-	SP
0A, 1A, 2A	Exit M	INC	-	D:M	Pop	-	-	SP
0B, 1B, 2B	End for M, C ≠ 1	INC	-	D:M	-	C←C-1	-	SP
	End for M, C = 1	INC	-	INC	-	C←C-1	-	SP
0C, 1C, 2C	End Loop	INC	Pop	TOS	-	-	-	SP
0D, 1D, 2D	Call Coroutine	INC	-	TOS	Pop & Push INC	-	-	SP
0E, 1E, 2E	Return	INC	-	TOS	Pop	-	-	SP
0F, 1F, 2F	End for, C ≠ 1	INC	Pop	TOS	-	C←C-1	-	SP
	End for, C = 1	INC	Pop	INC	Pop	C←C-1	-	SP

Cond. = (Test [S] OR I₅) XOR I₄

: = Concatination

C = Counter

INC = Output of Incrementer = AR + 1 (if $\overline{C_{in}}$ = LOW)

Note: For unconditional instructions, the action marked under "Cond: Pass" is taken.

TABLE 2. INSTRUCTION SET for I₅I₄ = 11

I ₅ - I ₀	Instruction	Y	Stack	Counter	Comp.	D-Mux
30	Continue	INC	-	-	-	SP
31	For D	INC	Push INC	C←D	-	SP
32	Decrement	INC	-	C←C-1	-	SP
33	Loop	INC	Push INC	-	-	SP
34	Pop D	INC	Pop	-	-	TOS
35	Push D	INC	Push D	-	-	SP
36	Reset SP	INC	SP←0	-	-	SP
37	For A	INC	Push INC	C←A	-	SP
38	Pop C	INC	Pop	C←TOS	-	SP
39	Push C	INC	Push C	-	-	SP
3A	Swap	INC	TOS←C	C←TOS	-	SP
3B	Push C Load D	INC	Push C	C←D	-	SP
3C	Load D	INC	-	C←D	-	SP
3D	Load A	INC	-	C←A	-	SP
3E	Set	INC	-	-	R←D, Enable	SP
3F	Clear	INC	-	-	Disable	SP

R = Comp. Register

Interrupts

The sequencer may be interrupted at the completion of the current microcycle by asserting the interrupt request input INTR. The return address of the interrupted routine is saved on the stack so that nested interrupts can be easily implemented. An interrupt is accepted if interrupts are enabled and the sequencer is not being reset or held (INTEN = HIGH, RST = HIGH, and HOLD = LOW). The interrupt-acknowledge output (INTA) goes LOW when an interrupt is accepted.

When there is no interrupt, addresses go from the address multiplexer to the Y-bus via the driver, and to the address register and the comparator via the interrupt multiplexer. When there is an interrupt, the driver of the sequencer is turned off, an external driver is turned on, and the interrupt multiplexer is switched. The interrupt address is supplied via the external driver to the Y-bus, the address register, and the comparator (Figure 4). In order to save the address from the address multiplexer, the address is stored in the interrupt return address register, which for simplicity is clocked every cycle. The next microinstruction is the first microinstruction of the interrupt routine (Figure 5).

In this cycle the address in the interrupt return address register is automatically pushed onto the stack. Therefore the microinstruction in this cycle must not use the stack; if a stack operation is programmed, the result is undefined. The instructions that do not use the stack are GOTO D, GOTO A, GOTO M, CONTINUE, DECREMENT, LOAD D, LOAD A, SET and CLEAR. A RETURN instruction terminates the interrupt routine and the interrupted routine is resumed. Interrupts only work with a single-level control path.

Traps

A trap is an unexpected situation linked to current microinstruction that must be handled before the microinstruction completes and changes the state of the system. An example of such a situation is an attempt to read a word from memory across a word boundary in a single cycle. When a trap occurs, the current microinstruction must be aborted and re-executed after the execution of a trap routine, which in the meantime will take corrective measures. An interrupt, on the other hand, is not linked directly to the current microinstruction that can complete safely before an interrupt routine is executed.

Execution of a trap requires that the sequencer ignore the current microinstruction, select the trap return address at the address multiplexer, and initiate an interrupt. This will save the trap return address on the stack and issue the trap address from an external source (Figure 6). The address register

contains the address of the microinstruction in the pipeline register, thus the address register already contains the trap return address when a trap occurs. This address can be selected by the address multiplexer by disabling the incrementer ($\overline{CIN} = 1$), and using the force continue mode (FC = 1). In this mode the sequencer ignores the current microinstruction. The remaining part of the trap handling is done by the interrupt (Figure 7), thus the section on interrupts also applies to traps. There is one exception, however. The interrupt enable cannot be used as a trap enable as it does not control the force continue mode and the carry-in to the incrementer.

Hold Mode

The sequencer has a hold mode in which the operation is suspended.

The outputs (Y, INTA, A-FULL & EQUAL) are disabled and the sequencer enters the hold mode immediately after the HOLD signal goes active. While the sequencer is in this mode, the internal state is left unchanged and the D-bus is disabled. The outputs (Y, INTA, A-FULL & EQUAL) are enabled again and the sequencer leaves the hold mode after the cycle immediately after the HOLD signal goes inactive.

In a time-multiplexed multi-microprocess system there may be one sequencer for all processes with microprogrammed context save and restore, or there may be one sequencer per microprocess permitting fast process switch. In the latter case the Y-buses of the sequencers are tied together and connected to a single microprogram store. A control unit decides on a cycle-by-cycle basis what sequencer should be running, and activates the HOLD signal to the remaining sequencers. The hold mode has higher priority than interrupts, and works independently of the reset. The hold mode can only be used with a single-level control path.

Master/Slave Configuration

In some systems reliability is very important. The master/slave configuration that consists of two sequencers operated in parallel is able to detect faults in both the interconnect and the internal function of the sequencers. One sequencer is the master and operates normally. The other is the slave, i.e., all outputs except the signal ERROR are turned into inputs and connected to the outputs of the master. Since the slave is operated in parallel with the master, it can compare its result with the result of the master and signal an error if they differ. The error signal from the master indicates a malfunctioning driver or contention. Because a TTL output goes HIGH when power is missing, the ERROR signal also indicates power failure.

High-Level Language Constructs

An example of high-level language constructs using Am29C331 instructions is given in Figure 3 (3-1, 3-2, 3-3, and 3-4).

```

REPEAT          LOOP
-              -
-              -
UNTIL CC       END LOOP NOT CC

WHILE CC DO    LOOP
-              IF NOT CC THEN EXIT L
-              -
-              -
END WHILE      END LOOP
L:

LOOP           LOOP
-             -
IF CC THEN EXIT IF CC THEN EXIT L
-             -
END LOOP      END LOOP
L:
    
```

Figure 3-1. Loops with Unknown Number of Iterations

```

FOR CNT: = 10 DOWN TO 1 DO  FOR D 10
-                             -
-                             -
END FOR                      END FOR
    
```

Figure 3-2. Loop with Known Number of Iterations

```

CASE I OF      PUSH D B
0: -          GOTO M
-           A: -
1: -          -, RETURN (TO B)
-           A + 2: -
2: -          -, RETURN (TO B)
-           A + 4: -
3: -          -, RETURN (TO B)
-           A + 6: -
-           -, RETURN
END CASE      B:
    
```

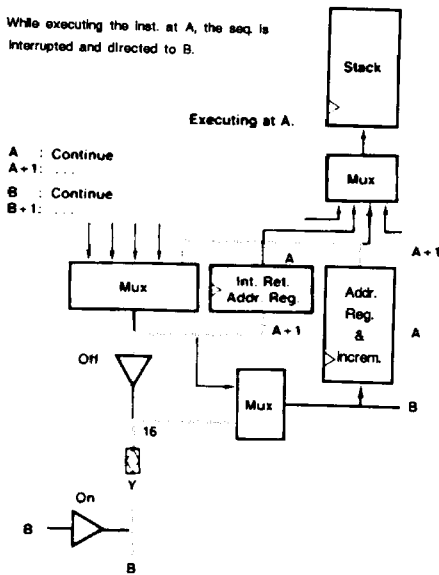
Figure 3-3. Case Statement
(with $D = A_{15} \dots A_4XX00$ and $M_{0,0-3} = A_{31}I_00$ during the GOTO M instruction. A_1A_0 must be 00, and X signifies a don't care.)

```

IF X THEN     PUSH D C
IF Y THEN     IF NOT X THEN GOTO A
IF NOT Y THEN IF NOT Y THEN GOTO B
-             -
-             -, RETURN (TO C)
ELSE          B:
-             -
-             -, RETURN (TO C)
END IF
ELSE          A:
IF Z THEN     IF NOT Z THEN GOTO D
-             -
-             -, RETURN (TO D)
ELSE          D:
-             -
-             -, RETURN (TO C)
END IF
END IF        C:
    
```

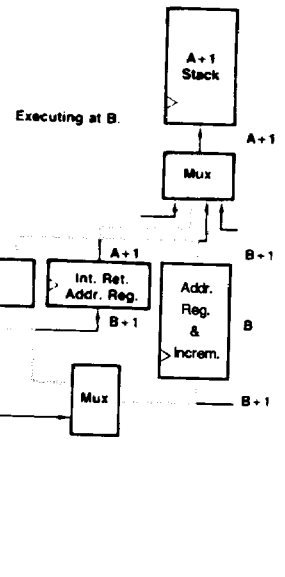
Figure 3-4. Double-Nested If Statement

While executing the inst. at A, the seq. is interrupted and directed to B.



AF004191

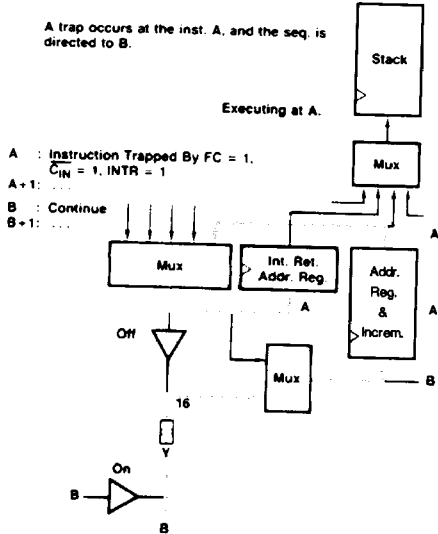
Figure 4. Am29C331 Interrupt Cycle 1



AF004211

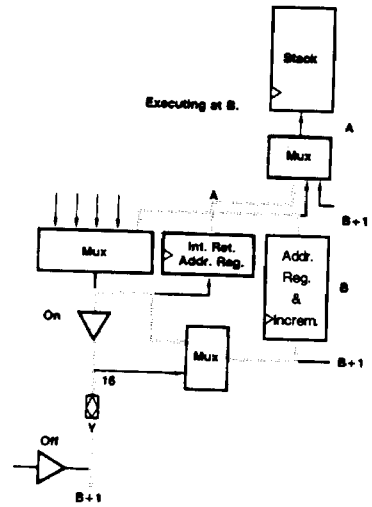
Figure 5. Am29C331 Interrupt Cycle 2

A trap occurs at the inst. A, and the seq. is directed to B.



AF004201

Figure 6. Am29C331 Traps Cycle 1



AF004181

Figure 7. Am29C331 Traps Cycle 2

Instruction Set Definition

Legend: ● = Other instruction
 ⊙ = Instruction being described
 CC = (Test [S₃ - S₀])

P = Test pass
 F = Test fail
 ○ = Register in part

Opcode (I ₅ - I ₀)	Mnemonics	Description	Execution Example
20H	BRA__D	GOTO D Unconditional branch to the address specified by the D inputs. The D port must be disabled to avoid bus contention.	
24H	BRA__A	GOTO A Unconditional branch to the address specified by the A inputs.	
28H	BRA__M	GOTO Multiway (D ₁₅ - D ₄ M _{X3} - M _{X0}) Unconditional branch to the address specified by the M inputs concatenated with the D input. The lower four bits on the D bus (D ₃ - D ₀) are replaced by one of the four sets of the four-bit multiway branch addresses. The multiway branch set is selected by bits D ₁ and D ₀ while bits D ₃ and D ₂ are "don't cares."	
2CH	BRA__S	GOTO TOS Unconditional branch to the address on the top of the stack.	
00H	BRCC_D	IF CC THEN GOTO D ELSE CONTINUE If CC is HIGH (pass), branch to the address specified by D. If CC is LOW (fail), continue. The D port must be disabled to avoid bus contention.	
04H	BRCC_A	IF CC THEN GOTO A ELSE CONTINUE If CC is HIGH (pass), branch to the address specified by A. If CC is LOW (fail), continue.	
08H	BRCC_M	IF CC THEN GOTO Multiway (D ₁₅ - D ₄ M _{X3} - M _{X0}) ELSE CONTINUE If CC is HIGH (pass), branch to the address specified by D inputs concatenated with the M inputs. If CC is LOW (fail) continue. The lower four bits on the D bus (D ₃ - D ₀) are replaced by one of the four sets of the 4-bit multiway branch addresses. The multiway branch set is selected by bits D ₁ and D ₀ while bits D ₃ and D ₂ are "don't cares."	
0CH	BRCC_S	IF CC THEN GOTO TOS ELSE POP STACK CONTINUE If CC is HIGH (pass), branch to the address on the top of the stack. If CC is LOW (fail), pop the stack and continue.	

Note: Opcode numbers are in hexadecimal notation.

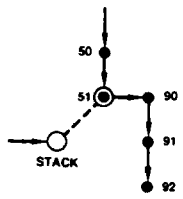
Opcode (15 - 10)	Mnemonics	Description	Execution Example
10H	BRNC_D	IF NOT CC THEN GOTO D ELSE CONTINUE If CC is LOW (pass), branch to the address specified by D. If CC is HIGH (fail), continue. The D Port must be disabled to avoid Bus contention.	
14H	BRNC_A	IF NOT CC THEN GOTO A ELSE CONTINUE If CC is LOW (pass), branch to the address specified by A. If CC is HIGH (fail), continue.	
18H	BRNC_M	IF NOT CC THEN GOTO Multiway (D ₁₅ - D ₄ M _{X3} - M _{X0}) ELSE CONTINUE If CC is LOW (pass), branch to the address specified by D inputs concatenated with the M inputs. If CC is HIGH (fail), continue. The lower four bits on the D bus (D ₃ - D ₀) are replaced by one of the four sets of the 4-bit multiway branch addresses. The multiway branch set is selected by bits D ₁ and D ₀ while bits D ₃ and D ₂ are "don't cares."	
1CH	BRNC_S	IF NOT CC THEN GOTO TOS ELSE POP STACK CONTINUE If CC is LOW (pass), branch to the address on the top of the stack. If CC is HIGH (fail), pop the stack and continue.	

21H	CALL_D	CALL D Unconditional branch to the subroutine specified by the D inputs. Push the return address (address Reg. + 1) on the stack. The D port must be disabled to avoid bus contention.	
25H	CALL_A	CALL A Unconditional branch to the subroutine specified by the A inputs. Push the return address (Address Reg. + 1) on the stack.	
29H	CALL_M	CALL Multiway (D ₁₅ - D ₄ M _{X3} - M _{X0}) Unconditional branch to the subroutine specified by the D inputs concatenated with the multiway inputs. Push the return address (Address Reg. + 1) on the stack. The lower four bits on the D bus (D ₃ - D ₀) are replaced by one of the four sets of the 4-bit multiway branch addresses. The multiway branch set is selected by bits D ₁ and D ₀ while bits D ₃ and D ₂ are "don't cares."	
2DH	CALL_S	CALL TOS Unconditional branch to the subroutine specified by the address on the top of the stack. The stack is popped and the return address (Address Reg. + 1) is then pushed onto the stack.	

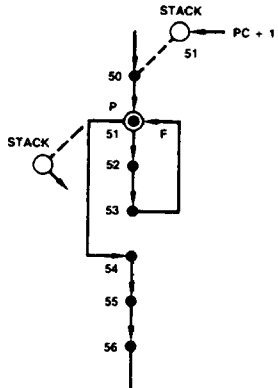
Note: Opcode numbers are in hexadecimal notation.

Opcode (I ₅ - I ₀)	Mnemonics	Description	Execution Example
01H	CCC_D	<p>IF CC, THEN CALL D ELSE CONTINUE</p> <p>If CC is HIGH (pass), call the subroutine specified by the D inputs. Push the return address (Address Reg. + 1) on the stack. If CC is LOW (fail), continue. The D port must be disabled to avoid bus contention.</p>	
05H	CCC_A	<p>IF CC, THEN CALL A ELSE CONTINUE</p> <p>If CC is HIGH (pass), call the subroutine specified by the A inputs. Push the return address (Address Reg. + 1) on the stack. If CC is LOW (fail), continue.</p>	
09H	CCC_M	<p>IF CC, THEN CALL Multiway (D₁₅ - D₄ M_{X3} - M_{X0}) ELSE CONTINUE</p> <p>If CC is HIGH (pass), call the subroutine specified by the D inputs concatenated with the M inputs. Push the return address (Address Reg. + 1) on the stack. The lower four bits on the D bus (D₃ - D₀) are replaced by one of the four sets of the 4-bit multiway branch addresses. The multiway branch set is selected by bits D₁ and D₀ while bits D₃ and D₂ are "don't cares."</p>	
0DH	CCC_S	<p>IF CC, THEN CALL TOS ELSE CONTINUE</p> <p>If CC is HIGH (pass), call the subroutine specified by the address on the top of the stack. The stack is popped and the return address (Address Reg. + 1) is pushed onto the stack. If CC is LOW (fail), continue.</p>	
11H	CNC_D	<p>IF NOT CC, THEN CALL D ELSE CONTINUE</p> <p>If CC is LOW (pass), call the subroutine specified by the D inputs. Push the return address (Address Reg. + 1) on the stack. If CC is HIGH (fail), continue. The D port must be disabled to avoid bus contention.</p>	
15H	CNC_A	<p>IF NOT CC, THEN CALL A ELSE CONTINUE</p> <p>If CC is LOW (pass), call the subroutine specified by the A inputs. Push the return address (Address Reg. + 1) on the stack. If CC is HIGH (fail), continue.</p>	
19H	CNC_M	<p>IF NOT CC, THEN CALL Multiway (D₁₅ - D₄ M_{X3} - M_{X0}) ELSE CONTINUE</p> <p>If CC is LOW (pass), call the subroutine specified by the D inputs concatenated with the M inputs. Push the return address (Address Reg. + 1) on the stack. The lower four bits on the D bus (D₃ - D₀) are replaced by one of the four sets of the 4-bit multiway branch addresses. The multiway branch set is selected by bits D₁ and D₀ while bits D₃ and D₂ are "don't cares."</p>	
1DH	CNC_S	<p>IF NOT CC, THEN CALL TOS ELSE CONTINUE</p> <p>If CC is LOW (pass), call the subroutine specified by the address on the top of the stack. The stack is popped and the return address (Address Reg. + 1) is pushed onto the stack.</p>	

Note: Opcode numbers are in hexadecimal notation.

Opcode (I5 - I0)	Mnemonics	Description	Execution Example
22H	EXIT_D	EXIT TO D Unconditional branch to the address specified by the D inputs and pop the stack. The D port must be disabled to avoid bus contention.	
26H	EXIT_A	EXIT TO A Unconditional branch to the address specified by the A inputs and pop the stack.	
2AH	EXIT_M	EXIT TO Multiway (D15 - D4 Mx3 - Mx0) Unconditional branch to the address specified by the D inputs concatenated with the M inputs and pop the stack. The lower four bits on the D bus (D3 - D0) are replaced by one of the four sets of the 4-bit multiway branch addresses. The multiway branch set is selected by bits D1 and D0 while D3 and D2 are "don't cares."	
2EH	EXIT_S	EXIT TO TOS Unconditional branch to the address on the top of the stack and pop the stack. Also used for unconditional returns.	

PF001790

02H	XTCC_D	IF CC, THEN EXIT TO D ELSE CONTINUE If CC is HIGH (pass), exit to the address specified by the D inputs and pop the stack. If CC is LOW (fail), continue with no pop. The D port must be disabled to avoid bus contention.	
06H	XTCC_A	IF CC, THEN EXIT TO A ELSE CONTINUE If CC is HIGH (pass), exit to the address specified by the A inputs and pop the stack. If CC is LOW (fail), continue with no pop.	
0AH	XTCC_M	IF CC, THEN EXIT TO Multiway (D15 - D4 Mx3 - Mx0) ELSE CONTINUE If CC is HIGH (pass), exit to the address specified by the D inputs concatenated with the M inputs and pop the stack. The lower four bits on the D bus (D3 - D0) are replaced by one of the four sets of the 4-bit multiway branch addresses. The multiway branch set is selected by bits D1 and D0 while D3 and D2 are "don't cares."	
0EH	XTCC_S	IF CC, THEN EXIT TO TOS ELSE CONTINUE If CC is HIGH (pass), exit to the address on the top of the stack and pop the stack. If CC is LOW (fail), continue with no pop. Also used for conditional returns.	

PF001800

Note: Opcode numbers are in hexadecimal notation.

Opcode (I ₅ - I ₀)	Mnemonics	Description	Execution Example
12H	XTNC_D	<p>IF NOT CC, THEN EXIT TO D ELSE CONTINUE</p> <p>If CC is LOW (pass), exit to the address specified by the D inputs and pop the stack. If CC is HIGH (fail), continue with no pop. The D port must be disabled to avoid bus contention.</p>	
16H	XTNC_A	<p>IF NOT CC, THEN EXIT TO A ELSE CONTINUE</p> <p>If CC is LOW (pass), exit to the address specified by the A inputs and pop the stack. If CC is HIGH (fail), continue with no pop.</p>	
1AH	XTNC_M	<p>IF NOT CC, THEN EXIT TO Multiway (D₁₅ - D₄ M_{X3} - M_{X0}) ELSE CONTINUE</p> <p>If CC is LOW (pass), exit to the address specified by the D inputs concatenated with the M inputs and pop the stack. The lower four bits on the D bus (D₃ - D₀) are replaced by one of the four sets of the 4-bit multiply branch addresses. The multiway branch set is selected by bits D₁ and D₀ while bits D₃ and D₂ are "don't cares."</p>	
1EH	XTNC_S	<p>IF NOT CC, THEN EXIT TO TOS ELSE CONTINUE</p> <p>If CC is LOW (pass), exit to the address on the top of the stack and pop the stack. If CC is HIGH (fail), continue with no pop. Also used for conditional returns.</p>	PF001810
23H	DJMP_D	<p>IF CNT ≠ 1 THEN CNT: = CNT - 1 GOTO D ELSE CNT: = CNT - 1 CONTINUE</p> <p>If the counter is not equal to one, decrement the counter and branch to the address specified by the D inputs. If the counter is equal to one, then decrement the counter and continue. The D port must be disabled to avoid bus contention.</p>	
27H	DJMP_A	<p>IF CNT ≠ 1 THEN CNT: = CNT - 1 GOTO A ELSE CNT: = CNT - 1 CONTINUE</p> <p>If the counter is not equal to one, decrement the counter and branch to the address specified by the A inputs. If the counter is equal to one, then decrement the counter and continue.</p>	
2BH	DJMP_M	<p>IF CNT ≠ 1 THEN CNT: = CNT - 1 GOTO Multiway (D₁₅ - D₄ M_{X3} - M_{X0}) ELSE CNT: = CNT - 1 CONTINUE</p> <p>If the counter is not equal to one, decrement the counter and branch to the address specified by the D inputs concatenated with the M inputs. The lower four bits on the D bus (D₃ - D₀) are replaced by one of the four sets of the 4-bit multiway branch addresses. The multiway branch set is selected by bits D₁ and D₀ while bits D₃ and D₂ are "don't cares."</p>	PF001820
2FH	DJMP_S	<p>IF CNT ≠ 1 THEN CNT: = CNT - 1 GOTO TOS ELSE CNT: = CNT - 1 POP STACK CONTINUE</p> <p>If the counter is not equal to one, decrement the counter and branch to the address on the top of the stack. If the counter is equal to one, then decrement the counter, pop the stack and continue.</p>	

Note: Opcode numbers are in hexadecimal notation.

Opcode (15 - 10)	Mnemonics	Description	Execution Example
03H	DJCC_D	<p>IF CC AND CNT \neq 1 THEN CNT: = CNT - 1 GOTO D ELSE CNT: = CNT - 1 CONTINUE</p> <p>If CC is HIGH (pass) and the counter is not equal to one, decrement the counter and branch to the address specified by the D inputs. If CC is LOW (fail) or the counter is equal to one, then decrement the counter and continue. The D port must be disabled to avoid bus contention.</p>	
07H	DJCC_A	<p>IF CC AND CNT \neq 1 THEN CNT: = CNT - 1 GOTO A ELSE CNT: = CNT - 1 CONTINUE</p> <p>If CC is HIGH (pass) and the counter is not equal to one, decrement the counter and branch to the address specified by the A inputs. If CC is LOW (fail) or the counter is equal to one, then decrement the counter and continue.</p>	
0BH	DJCC_M	<p>IF CC AND CNT \neq 1 THEN CNT: = CNT - 1 GOTO Multiway (D₁₅ - D₄ M_{X3} - M_{X0}) ELSE CNT: = CNT - 1 CONTINUE</p> <p>If CC is HIGH (pass) and the counter is not equal to one, decrement the counter and branch to the address specified by the D inputs concatenated with the M inputs. The lower four bits on the D bus (D₃ - D₀) are replaced by one of the four sets of the 4-bit multiway branch addresses. The multiway branch set is selected by bits D₁ and D₀ while bits D₃ and D₂ are "don't cares."</p>	
0FH	DJCC_S	<p>IF CC AND CNT \neq 1 THEN CNT: = CNT - 1 GOTO TOS ELSE CNT: = CNT - 1 POP STACK CONTINUE</p> <p>If CC is HIGH (pass) and the counter is not equal to one, decrement the counter and branch to the address on the top of the stack. If CC is LOW (fail) or the counter is equal to one, then decrement the counter, pop the stack and continue.</p>	

PF001830

Note: Opcode numbers are in hexadecimal notation.

Opcode (I ₅ - I ₀)	Mnemonics	Description	Execution Example	
13H	DJNCC_D	<p>IF NOT CC AND CNT ≠ 1 THEN CNT: = CNT - 1 GOTO D ELSE CNT: = CNT - 1 CONTINUE</p> <p>If CC is LOW (pass) and the counter is not equal to one, decrement the counter and branch to the address specified by the D inputs. If CC is HIGH (fail) or the counter is equal to one, then decrement the counter and continue. The D port must be disabled to avoid bus contention.</p>		
17H	DJNCC_A	<p>IF NOT CC AND CNT ≠ 1 THEN CNT: = CNT - 1 GOTO A ELSE CNT: = CNT - 1 CONTINUE</p> <p>If CC is LOW (pass) and the counter is not equal to one, decrement the counter and branch to the address specified by the A inputs. The content of the interrupt return address register and the address register is replaced by the A address in this case. If CC is HIGH (fail) or the counter is equal to one, the current address is incremented, appears on the bus for continue, and is stored into the above two registers.</p>		
1BH	DJNCC_M	<p>IF NOT CC AND CNT ≠ 1 THEN CNT: = CNT - 1 GOTO Multiway (D₁₅ - D₄ M₃ - M₀) ELSE CONTINUE</p> <p>If CC is LOW (pass) and the counter is not equal to one, decrement the counter and branch to the address specified by the D inputs concatenated with the M inputs. The lower four bits on the D bus (D₃ - D₀) are replaced by one of the four sets of the 4-bit multiway branch addresses. The multiway branch set is selected by bits D₁ and D₀ while bits D₃ and D₂ are "don't cares."</p>		PF001840
1FH	DJNCC_S	<p>IF NOT CC AND CNT ≠ 1 THEN CNT: = CNT - 1 GOTO TOS ELSE CNT: = CNT - 1 POP STACK CONTINUE</p> <p>If CC is LOW (pass) and the counter is not equal to one, decrement the counter and branch to the address on the top of the stack. If CC is HIGH (fail) or the counter is equal to one, then decrement the counter, pop the stack and continue.</p>		
2EH	RET	<p>RETURN</p> <p>Unconditional return from subroutine. The return address is popped from the stack.</p>		
0EH	RETCC	<p>IF CC THEN RETURN ELSE CONTINUE</p> <p>If CC is HIGH (pass), return from subroutine. The return address is popped from the stack. If CC is LOW (fail), continue.</p>		
1EH	RETNC	<p>IF NOT CC THEN RETURN ELSE CONTINUE</p> <p>If CC is LOW (pass), return from subroutine. The return address is popped from the stack. If CC is HIGH (fail), continue.</p>		PF001850

Note: Opcode numbers are in hexadecimal notation.

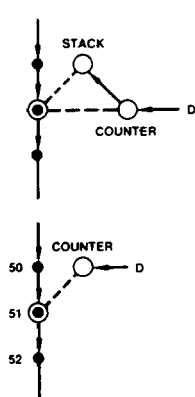
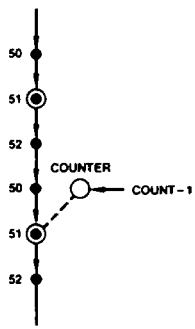
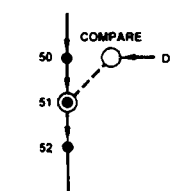
Opcode (I5 - I0)	Mnemonics	Description	Execution Example
31H	FOR_D	INITIALIZE LOOP Push the Address Reg. + 1 on the stack, load the counter from the D inputs and continue. Use with DJUMP_S for FOR...NEXT loops. The D port must be disabled to avoid bus contention.	
37H	FOR_A	INITIALIZE LOOP Push the Address Reg. + 1 on the stack, load the counter from the A inputs and continue. Use with DJUMP_S for FOR...NEXT loops.	
33H	LOOP	INITIALIZE LOOP Push the Address Reg. + 1 on the stack and continue. Use with BRCC_S for REPEAT...UNTIL loops, or with XTCC_D and BRA_S for WHILE...END WHILE loops.	

PF001860

34H	POP_D	Pop the stack and output the value on the D outputs and continue. The D port must be enabled.	
38H	POP_C	Pop the stack and store the value in the counter and continue.	
35H	PUSH_D	Push the D inputs on the stack and continue. The D port must be disabled to avoid bus contention.	
39H	PUSH_C	Push the counter on the stack and continue.	
3AH	SWAP	Exchange the counter and the top of stack and continue.	

PF001870

Note: Opcode numbers are in hexadecimal notation.

Opcode (I ₅ - I ₀)	Mnemonics	Description	Execution Example
3BH 3CH 3DH	STACK_C LOAD_D LOAD_A	Push the counter on the stack and load the counter with the value of the D inputs and continue. Load the counter with the value of the D inputs and continue. The D port must be disabled to avoid bus contention. Load the counter with the value of the A inputs and continue.	 <p>The diagram illustrates the execution of three opcodes. In the first part, the counter (a circle) is pushed onto the stack (a vertical line of circles). The counter is then loaded with the value from the D input (a circle). In the second part, the counter is loaded with the value from the A input (a circle).</p>
30H 32H 36H	CONT DECR RESET_SP	Continue. Decrement the counter and continue. Reset the stack pointer and continue.	 <p>The diagram illustrates the execution of three opcodes. The counter (a circle) is decremented by one (COUNT - 1). The stack pointer (a circle) is reset to the value 51.</p>
3EH 3FH	SET CLEAR	Load the comparison register with the value of the D inputs, enable the comparator and continue. Disable the comparator and continue.	 <p>The diagram illustrates the execution of two opcodes. The comparison register (a circle) is loaded with the value from the D input (a circle). The comparison register is then disabled.</p>

Note: Opcode numbers are in hexadecimal notation.

APPLICATIONS

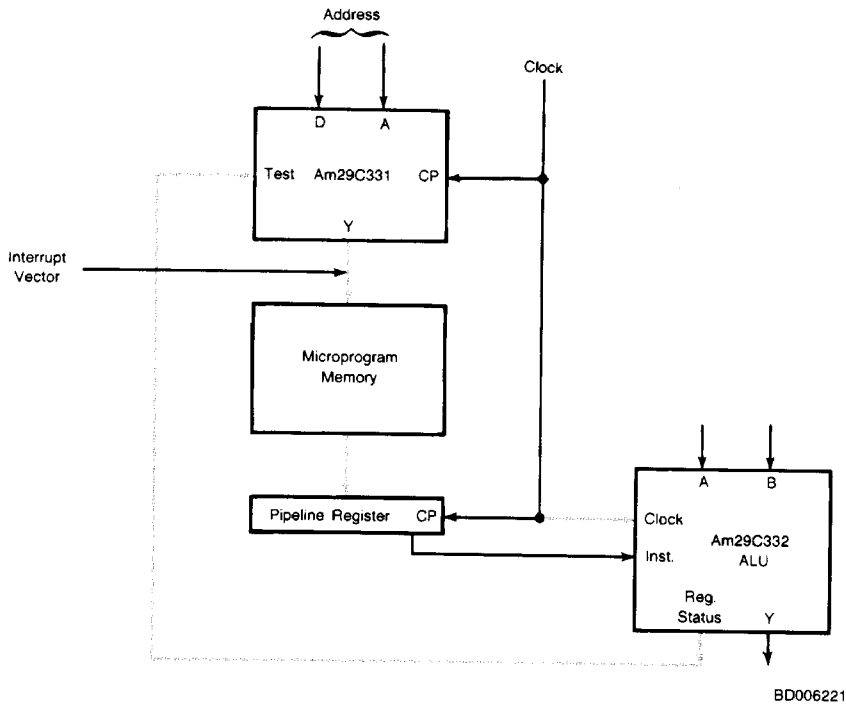


Figure 8. Typical Control-Path Architecture For Am29C300 Family

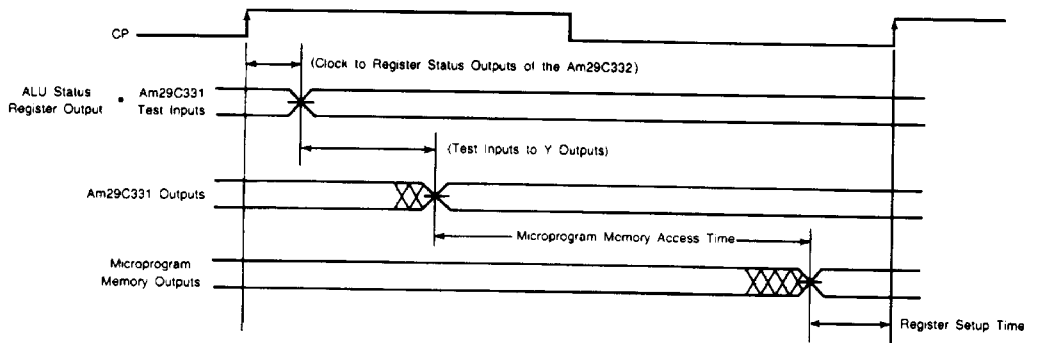


Figure 9. Cycle Timing Waveform*

*This waveform shows the timing relationship for the configuration shown in Figure 8.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature -65 to +150°C
 (Case) Temperature Under Bias -55 to +125°C
 Supply Voltage to
 Ground Potential Continuous -0.3 V to +7.0 V
 DC Voltage Applied to Outputs For
 High Output State -0.3 V to +V_{CC} +0.3 V
 DC Input Voltage -0.3 V to +V_{CC} +0.3 V
 DC Output Current, Into LOW Outputs 30 mA
 DC Input Current -10 mA to +10 mA

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 Voltage (V_{CC}) +4.75 V to +5.25 V

Military* (M) Devices
 Temperature (T_A) -55 to +125°C
 Supply Voltage (V_{CC}) +4.5 V to +5.5 V

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

*Military Product 100% tested at T_A = +25°C, +125°C, and -55°C.

DC CHARACTERISTICS over operating range unless otherwise specified (for APL Products, Group A, Subgroups 1, 2, 3 are tested unless otherwise noted)

Parameter Symbol	Parameter Description	Test Conditions (Note 1)		Min.	Max.	Unit
V _{OH}	Output HIGH Voltage	V _{CC} = Min. V _{IN} = V _{IH} or V _{IL}	I _{OH} = 0.4 mA	2.4		Volts
V _{OL}	Output LOW Voltage	V _{CC} = Min. V _{IN} = V _{IH} or V _{IL}	I _{OL} = 8 mA for Y-BUS = 4 mA for All Other Pins		0.5	Volts
V _{IH}	Guaranteed Input Logical HIGH Voltage (Note 2)			2.0		Volts
V _{IL}	Guaranteed Input Logical LOW Voltage (Note 2)				0.8	Volts
I _{IL}	Input LOW Current	V _{CC} = Max. V _{IN} = 0.5 Volts			-10	μA
I _{IH}	Input HIGH Current	V _{CC} = Max. V _{IN} = V _{CC} - 0.5 V			10	μA
I _{OZH}	Off-State (HIGH Impedance) Output Current	V _{CC} = Max. V _O = 2.4 Volts			10	μA
I _{OZL}	Off-State (HIGH Impedance) Output Current	V _{CC} = Max. V _O = 0.5 Volts			-10	μA
I _{CC}	Static Power Supply Current (Note 3)	V _{CC} = Max., V _{IN} = V _{CC} or GND, I _O = 0 μA	COM'L	29C331	40	mA
				29C331-1/-2	50	
			MIL	29C331 only	50	
C _{PD}	Power Dissipation Capacitance (Note 4)	V _{CC} = 5.0 V T _A = 25°C No Load		pF Typical		

- Notes: 1. V_{CC} conditions shown as Min. or Max. refer to the commercial and military V_{CC} limits.
 2. These input levels provide zero-noise immunity and should only be statically tested in a noise-free environment (not functionally tested).
 3. Worst-case I_{CC} is measured at the lowest temperature in the specified operating range.
 4. C_{PD} determines the no-load dynamic current consumption:
 I_{CC} (Total) = I_{CC} (Static) + C_{PD} V_{CC} f, where f is the switching frequency of the majority of the internal nodes, normally one-half of the clock frequency. This specification is not tested.

SWITCHING CHARACTERISTICS over **COMMERCIAL** operating range

A. COMBINATIONAL PROPAGATION DELAYS

No.	From	To	29C331	29C331-1	29C331-2	Unit
			Max. Delay	Max. Delay	Max. Delay	
1	D15-0	Y15-0	22*	20*	18	ns
	D15-0	EQUAL	32	28	23	ns
	D15-0	ERROR	36	32	26	ns
2	A15-0	Y15-0	20	18	16	ns
	A15-0	EQUAL	31	27	22	ns
	A15-0	ERROR	33	29	24	ns
3	Mx3-x0	Y15-0	19	16	16	ns
	Mx3-x0	EQUAL	29	26	21	ns
	Mx3-x0	ERROR	33	29	24	ns
	Y15-0	EQUAL	31	28	23	ns
	Y15-0	ERROR	26	23	19	ns
4	I5-0	Y31-0	24	22	18	ns
	I5-0	D15-0	29	26	21	ns
	I5-0	EQUAL	36	33	27	ns
5	I5-0	ERROR	40	35	28	ns
	T11-0	Y15-0	24	22	18	ns
	T11-0	EQUAL	35	32	26	ns
	T11-0	ERROR	37	33	27	ns
	S3-0	Y15-0	24	22	18	ns
6	S3-0	EQUAL	35	32	26	ns
	S3-0	ERROR	37	33	27	ns
	CP	Y15-0	28	25	20	ns
7	CP	D15-0	27/Z	25/Z	20/Z	ns
	CP	A-FULL	27	24	20	ns
8	CP	EQUAL	36	32	26	ns
	CP	ERROR	50	45	36	ns
	RST	Y15-0	26/Z	24/Z	20/Z	ns
9	RST	D15-0	Z	Z	Z	ns
	RST	INTA	22	19	17	ns
	RST	EQUAL	35	31	25	ns
10	RST	ERROR	38	34	28	ns
	FC	Y15-0	24	22	18	ns
	FC	D15-0	28	25	20	ns
11	FC	EQUAL	33	30	24	ns
	FC	ERROR	35	31	25	ns
	INTR	Y15-0	Z	Z	Z	ns
12	INTR	INTA	17	16	9	ns
	INTR	EQUAL	(Note 1)	(Note 1)	(Note 1)	ns
	INTR	ERROR	46	21	18	ns
13	INTEN	Y15-0	Z	Z	Z	ns
	INTEN	INTA	16	15	9	ns
	INTEN	EQUAL	(Note 1)	(Note 1)	(Note 1)	ns
14	INTEN	ERROR	46	21	18	ns
	HOLD	Y15-0	Z	Z	Z	ns
	HOLD	INTA	Z	Z	Z	ns
	HOLD	A-FULL	Z	Z	Z	ns
	HOLD	EQUAL	34/Z	31/Z	17/Z	ns
	HOLD	ERROR	46	18	17	ns
	OED	D15-0	Z	17	Z	ns
	OED	ERROR	19	Z	Z	ns
	INTA	ERROR	19**	17**	17	ns
	A-FULL	ERROR	21**	20**	17	ns
15	EQUAL	ERROR	19**	17**	17	ns
	C _m	Y15-0	24	21	18	ns
	C _m	EQUAL	36	33	20	ns
	C _m	ERROR	37	33	21	ns
	SLAVE	Y15-0	Z	Z	Z	ns
	SLAVE	D15-0	Z	Z	Z	ns
	SLAVE	INTA	Z	Z	Z	ns
16	SLAVE	A-FULL	Z	Z	Z	ns
	SLAVE	EQUAL	Z	Z	Z	ns

Notes: See notes following Table D.

*This includes using D as select lines for multiway sets.

**In the slave mode.

SWITCHING CHARACTERISTICS over **COMMERCIAL** operating range (Cont'd.)

B. OUTPUT DISABLE TIME

No.	From	To	Description	29C331	29C331-1	29C331-2	Unit
				Max. Value	Max. Value	Max. Value	
43 44	RST	Y ₁₅₋₀	Reset-to-Address Enable	29	25	25	ns
	RST	Y ₁₅₋₀	Reset-to-Address Disable	29	25	25	ns
	INTR	Y ₁₅₋₀	INTR-to-Address Enable	24	21	21	ns
	INTR	Y ₁₅₋₀	INTR-to-Address Disable	24	21	21	ns
	INTEN	Y ₁₅₋₀	INTEN-to-Address Enable	24	21	21	ns
	INTEN	Y ₁₅₋₀	INTEN-to-Address Disable	24	21	21	ns
	HOLD	Y ₁₅₋₀	HOLD-to-Address Enable	23	20	20	ns
	HOLD	Y ₁₅₋₀	HOLD-to-Address Disable	23	20	20	ns
	SLAVE	Y ₁₅₋₀	SLAVE-to-Address Enable	24	21	21	ns
	SLAVE	Y ₁₅₋₀	SLAVE-to-Address Disable	24	21	21	ns
	OED	Y ₁₅₋₀	OED-to-Data Enable	26	22	22	ns
	OED	D ₁₅₋₀	OED-to-Data Disable	26	22	22	ns
	RST	D ₁₅₋₀	Reset-to-Data Enable	27	23	23	ns
	RST	D ₁₅₋₀	Reset-to-Data Disable	27	23	23	ns
	SLAVE	D ₁₅₋₀	SLAVE-to-Data Enable	26	22	22	ns
	SLAVE	D ₁₅₋₀	SLAVE-to-Data Disable	26	22	22	ns
	CP	D ₁₅₋₀	Clock-to-Data Enable	35	24	24	ns
	CP	D ₁₅₋₀	Clock-to-Data Disable	35	24	24	ns
	HOLD	INTA	HOLD-to-INTA Enable	22	19	19	ns
	HOLD	INTA	HOLD-to-INTA Disable	22	19	19	ns
	HOLD	A-FULL	HOLD-to-A-FULL Enable	21	18	18	ns
	HOLD	A-FULL	HOLD-to-A-FULL Disable	21	18	18	ns
	HOLD	EQUAL	HOLD-to-EQUAL Enable	21	18	18	ns
	HOLD	EQUAL	HOLD-to-EQUAL Disable	21	18	18	ns
	SLAVE	INTA	SLAVE-to-INTA Enable	22	19	19	ns
	SLAVE	INTA	SLAVE-to-INTA Disable	22	19	19	ns
	SLAVE	A-FULL	SLAVE-to-A-FULL Enable	22	19	19	ns
	SLAVE	A-FULL	SLAVE-to-A-FULL Disable	22	19	19	ns
	SLAVE	EQUAL	SLAVE-to-EQUAL Enable	22	19	19	ns
	SLAVE	EQUAL	SLAVE-to-EQUAL Disable	22	19	19	ns

Notes: See notes following Table D.

SWITCHING CHARACTERISTICS over **COMMERCIAL** over operating range (Cont'd.)

C. SETUP AND HOLD TIMES

No.	Parameter	For	With Respect To	29C331	29C331-1	29C331-2	Unit
				Max. Value	Max. Value	Max. Value	
17	Data Setup	D15-0	CP ↑	21	19	19	ns
18	Data Hold	D15-0	CP ↑	0	0	0	ns
19	Alternate Data Setup	A15-0	CP ↑	23	21	21	ns
20	Alternate Data Hold	A15-0	CP ↑	0	0	0	ns
21	Multiway Setup	Mx3-X0	CP ↑	28	21	21	ns
22	Multiway Hold	Mx3-X0	CP ↑	0	0	0	ns
23	Address Setup	Y15-0	CP ↑	18	17	17	ns
24	Address Hold	Y15-0	CP ↑	0	0	0	ns
25	Instruction Setup	I5-0	CP ↑	24	21	21	ns
26	Instruction Hold	I5-0	CP ↑	0	0	0	ns
27	Forced Continue Setup	FC	CP ↑	21	19	19	ns
28	Forced Continue Hold	FC	CP ↑	0	0	0	ns
29	Test Setup	T11-0	CP ↑	21	20	20	ns
30	Test Hold	T11-0	CP ↑	0	0	0	ns
31	Select Setup	S3-0	CP ↑	22	20	20	ns
32	Select Hold	S3-0	CP ↑	0	0	0	ns
33	Reset Setup	RST	CP ↑	22	20	20	ns
34	Reset Hold	RST	CP ↑	0	0	0	ns
35	Interrupt Request Setup	INTR	CP ↑	20	18	18	ns
36	Interrupt Request Hold	INTR	CP ↑	0	0	0	ns
37	Interrupt Enable Setup	INTEN	CP ↑	18	16	16	ns
38	Interrupt Enable Hold	INTEN	CP ↑	0	0	0	ns
39	Hold Mode Setup	HOLD	CP ↑	21	19	19	ns
40	Hold Mode Hold	HOLD	CP ↑	0	0	0	ns
41	Carry-In Setup	C _{in}	CP ↑	22	20	20	ns
42	Carry-In Hold	C _{in}	CP ↑	0	0	0	ns

D. MINIMUM CLOCK REQUIREMENT

No.	Description	29C331	29C331-1	29C331-2	Unit
		Max. Value	Max. Value	Max. Value	
53	Minimum Clock LOW Time	23	22	22	ns
54	Minimum Clock HIGH Time	19	16	16	ns

Notes: 1. (INTR, INTEN)-to-EQUAL is the sum of (INTR, INTEN)-to-Y disable time and Y-to-EQUAL delay time.

2. C_L = 50 pF; C_L = 5 pF for Disable Time only.

SWITCHING CHARACTERISTICS over **MILITARY** operating range (for APL Products, Group A, Subgroups 9, 10, 11 are tested unless otherwise noted)

A. COMBINATIONAL PROPAGATION DELAYS

No.	From	To	29C331	Unit	
			Max. Delay		
1	D ₁₅₋₀	Y ₁₅₋₀	30*	ns	
	D ₁₅₋₀	EQUAL	48	ns	
	D ₁₅₋₀	ERROR	29**	ns	
2	A ₁₅₋₀	Y ₁₅₋₀	27	ns	
	A ₁₅₋₀	EQUAL	44	ns	
	A ₁₅₋₀	ERROR	50	ns	
3	M _{x3-x0}	Y ₁₅₋₀	30	ns	
	M _{x3-x0}	EQUAL	48	ns	
	M _{x3-x0}	ERROR	55	ns	
	Y ₁₅₋₀	EQUAL	41	ns	
4	Y ₁₅₋₀	ERROR	29**	ns	
	I ₅₋₀	Y ₃₁₋₀	32	ns	
5	I ₅₋₀	D ₁₅₋₀	37	ns	
	I ₅₋₀	EQUAL	48	ns	
	I ₅₋₀	ERROR	55	ns	
6	T ₁₁₋₀	Y ₁₅₋₀	32	ns	
	T ₁₁₋₀	EQUAL	48	ns	
	T ₁₁₋₀	ERROR	55	ns	
	S ₃₋₀	Y ₁₅₋₀	32	ns	
	S ₃₋₀	EQUAL	48	ns	
7	S ₃₋₀	ERROR	55	ns	
	CP	Y ₁₅₋₀	37	ns	
	CP	D ₁₅₋₀	37/Z	ns	
9	CP	A-FULL	32	ns	
	CP	EQUAL	54	ns	
	CP	ERROR	60	ns	
10	RST	Y ₁₅₋₀	32/Z	ns	
	RST	D ₁₅₋₀	Z	ns	
11	RST	INTA	22	ns	
	RST	EQUAL	48	ns	
	RST	ERROR	55	ns	
12	FC	Y ₁₅₋₀	32	ns	
	FC	D ₁₅₋₀	37	ns	
	FC	EQUAL	48	ns	
13	FC	ERROR	55	ns	
	INTR	Y ₁₅₋₀	Z	ns	
	INTR	INTA	21	ns	
14	INTR	EQUAL	(Note 1)	ns	
	INTR	ERROR	49	ns	
	INTEN	Y ₁₅₋₀	Z	ns	
	INTEN	INTA	21	ns	
	INTEN	EQUAL	(Note 1)	ns	
15	INTEN	ERROR	49	ns	
	HOLD	Y ₁₅₋₀	Z	ns	
	HOLD	INTA	Z	ns	
	HOLD	A-FULL	21/Z	ns	
	HOLD	EQUAL	43/Z	ns	
	HOLD	ERROR	49	ns	
	OED	D ₁₅₋₀	26	ns	
	OED	ERROR	Z	ns	
	INTA	ERROR	29**	ns	
	A-FULL	ERROR	29**	ns	
	EQUAL	ERROR	29**	ns	
	16	C _{in}	Y ₁₅₋₀	32	ns
		C _{in}	EQUAL	48	ns
C _{in}		ERROR	55	ns	
SLAVE		Y ₁₅₋₀	Z	ns	
SLAVE		D ₁₅₋₀	Z	ns	
SLAVE		INTA	Z	ns	
SLAVE		A-FULL	Z	ns	
SLAVE	EQUAL	Z	ns		

Notes: See notes following Table D.

*This includes using D as select lines for multiway sets.

**In the slave mode.

SWITCHING CHARACTERISTICS over **MILITARY** operating range (Cont'd.)

B. OUTPUT DISABLE TIME

No.	From	To	Description	29C331	Unit
				Max. Value	
43	RST	Y15-0	Reset-to-Address Enable	26	ns
	RST	Y15-0	Reset-to-Address Disable	26	ns
44	INTR	Y15-0	INTR-to-Address Enable	26	ns
	INTR	Y15-0	INTR-to-Address Disable	26	ns
	INTEN	Y15-0	INTEN-to-Address Enable	26	ns
	INTEN	Y15-0	INTEN-to-Address Disable	26	ns
	HOLD	Y15-0	HOLD-to-Address Enable	26	ns
	HOLD	Y15-0	HOLD-to-Address Disable	26	ns
	SLAVE	Y15-0	SLAVE-to-Address Enable	26	ns
	SLAVE	Y15-0	SLAVE-to-Address Disable	26	ns
	OED	Y15-0	OED-to-Data Enable	26	ns
	OED	D15-0	OED-to-Data Disable	26	ns
	RST	D15-0	Reset-to-Data Enable	26	ns
	RST	D15-0	Reset-to-Data Disable	26	ns
	SLAVE	D15-0	SLAVE-to-Data Enable	26	ns
	SLAVE	D15-0	SLAVE-to-Data Disable	26	ns
	CP	D15-0	Clock-to-Data Enable	23	ns
	CP	D15-0	Clock-to-Data Disable	23	ns
	HOLD	INTA	HOLD-to-INTA Enable	21	ns
	HOLD	INTA	HOLD-to-INTA Disable	21	ns
	HOLD	A-FULL	HOLD-to-A-FULL Enable	21	ns
	HOLD	A-FULL	HOLD-to-A-FULL Disable	21	ns
	HOLD	EQUAL	HOLD-to-EQUAL Enable	21	ns
	HOLD	EQUAL	HOLD-to-EQUAL Disable	21	ns
	SLAVE	INTA	SLAVE-to-INTA Enable	21	ns
	SLAVE	INTA	SLAVE-to-INTA Disable	21	ns
	SLAVE	A-FULL	SLAVE-to-A-FULL Enable	21	ns
	SLAVE	A-FULL	SLAVE-to-A-FULL Disable	21	ns
	SLAVE	EQUAL	SLAVE-to-EQUAL Enable	21	ns
	SLAVE	EQUAL	SLAVE-to-EQUAL Disable	21	ns

Notes: See notes following Table D.

SWITCHING CHARACTERISTICS over **MILITARY** operating range (Cont'd.)

C. SETUP AND HOLD TIMES

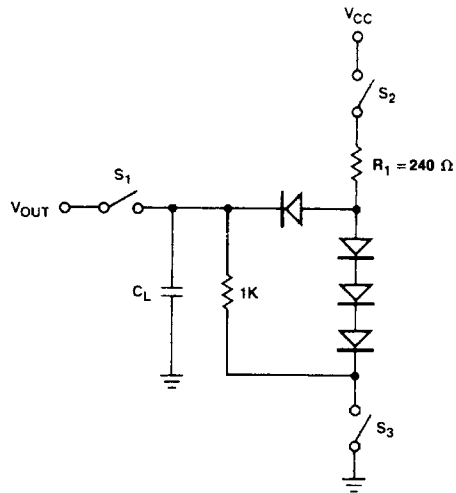
No.	Parameter	For	With Respect To	29C331	Unit
				Max. Value	
17	Data Setup	D ₁₅₋₀	CP ↑	32	ns
18	Data Hold	D ₁₅₋₀	CP ↑	1	ns
19	Alternate Data Setup	A ₁₅₋₀	CP ↑	32	ns
20	Alternate Data Hold	A ₁₅₋₀	CP ↑	1	ns
21	Multiway Setup	M _{X3-X0}	CP ↑	32	ns
22	Multiway Hold	M _{X3-X0}	CP ↑	1	ns
23	Address Setup	Y ₁₅₋₀	CP ↑	27	ns
24	Address Hold	Y ₁₅₋₀	CP ↑	2	ns
25	Instruction Setup	I ₅₋₀	CP ↑	32	ns
26	Instruction Hold	I ₅₋₀	CP ↑	0	ns
27	Forced Continue Setup	FC	CP ↑	32	ns
28	Forced Continue Hold	FC	CP ↑	1	ns
29	Test Setup	T ₁₁₋₀	CP ↑	32	ns
30	Test Hold	T ₁₁₋₀	CP ↑	0	ns
31	Select Setup	S ₃₋₀	CP ↑	32	ns
32	Select Hold	S ₃₋₀	CP ↑	0	ns
33	Reset Setup	RST	CP ↑	32	ns
34	Reset Hold	RST	CP ↑	1	ns
35	Interrupt Request Setup	INTR	CP ↑	27	ns
36	Interrupt Request Hold	INTR	CP ↑	1	ns
37	Interrupt Enable Setup	INTEN	CP ↑	27	ns
38	Interrupt Enable Hold	INTEN	CP ↑	1	ns
39	Hold Mode Setup	HOLD	CP ↑	27	ns
40	Hold Mode Hold	HOLD	CP ↑	1	ns
41	Carry-In Setup	C _{in}	CP ↑	30	ns
42	Carry-In Hold	C _{in}	CP ↑	1	ns

D. MINIMUM CLOCK REQUIREMENTS

No.		29C331	Unit
		Max. Value	
53	Minimum Clock LOW Time	33	ns
54	Minimum Clock HIGH Time	28	ns

- Notes: 1. (INTR, INTEN)-to-EQUAL is the sum of (INTR, INTEN)-to-Y disable time and Y-to-EQUAL delay time.
 2. C_L = 50 pF; C_L = 5 pF for Disable Time only.
 3. The status of I₅₋₀ and FC must not be changed during the clock LOW time.

SWITCHING TEST CIRCUIT

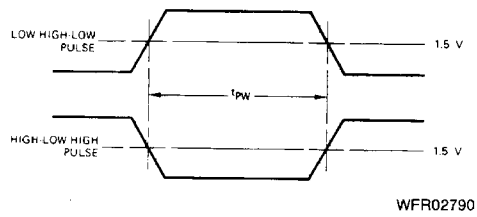
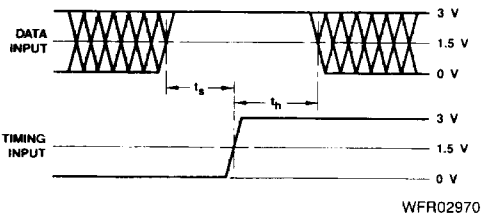


TC003420

A. Three-State Outputs

- Notes:
1. $C_L = 50$ pF includes scope probe, wiring, and stray capacitances without device in test fixture.
 2. S_1 , S_2 , S_3 are closed during function tests and all AC tests except output enable tests.
 3. S_1 and S_3 are closed while S_2 is open for t_{PZH} test.
 S_1 and S_2 are closed while S_3 is open for t_{PZL} test.
 4. $C_L = 5.0$ pF for output disable tests.

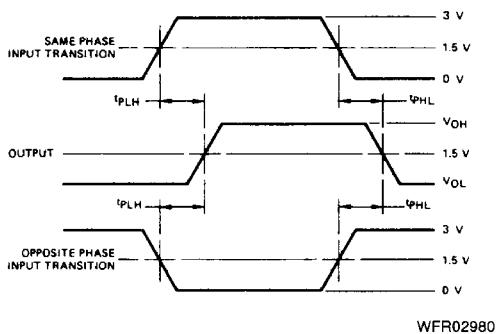
SWITCHING TEST WAVEFORMS



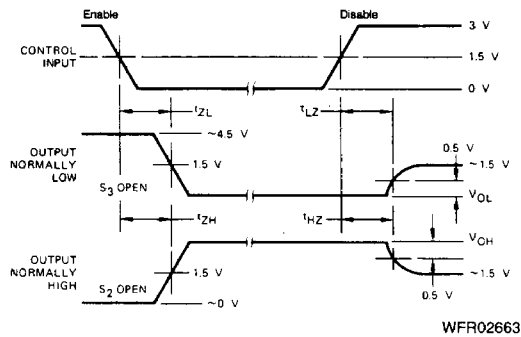
Pulse Width

- Notes: 1. Diagram shown for HIGH data only. Output transition may be opposite sense.
2. Cross hatched area is don't care condition.

Setup, Hold, and Release Times



Propagation Delay



- Notes: 1. Diagram shown for Input Control Enable-LOW and Input Control Disable-HIGH.
2. S_1 , S_2 , and S_3 of Load Circuit are closed except where shown.

Enable and Disable Times

Test Philosophy and Methods

The following points give the general philosophy that we apply to tests that 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 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 5–8 ns. Inductance in the ground cable may allow the ground pin at the device to rise by hundreds of millivolts momentarily. Current level may vary from product to product.
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 $V_{IL} \leq 0$ V and $V_{IH} \geq 3$ V for AC tests.
5. To simplify failure analysis, programs should be designed to perform DC, Function, and AC tests as three distinct groups of tests.
6. Capacitive Loading for AC Testing

Automatic testers and their associated hardware have stray capacitance which varies from one type of tester to another, but is generally around 50 pF. This 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 into 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 setup 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 setup and the knowledge that certain DC measurements (I_{OH} , I_{OL} , for example) have already been taken and are within specification. In some cases, special DC tests are performed in order to facilitate this correlation.

7. 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.

8. AC Testing

Occasionally parameters are specified that 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 these tests with other AC tests that have been performed. These correlations are arrived at by the cognizant engineer 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 that have already been performed. In these cases, the redundant tests are not performed.

9. Output Short-Circuit Current Testing

When performing I_{OS} tests on devices containing RAM or registers, great care must be taken that undershoot caused by grounding the high-state output does not trigger parasitic elements which in turn cause the device to change state. In order to avoid this effect, it is common to make the measurement at a voltage (V_{output}) that is slightly above ground. The V_{CC} is raised by the same amount so that the result (as confirmed by Ohm's law and precise bench testing) is identical to the $V_{OUT} = 0$, $V_{CC} = \text{Max.}$ case.

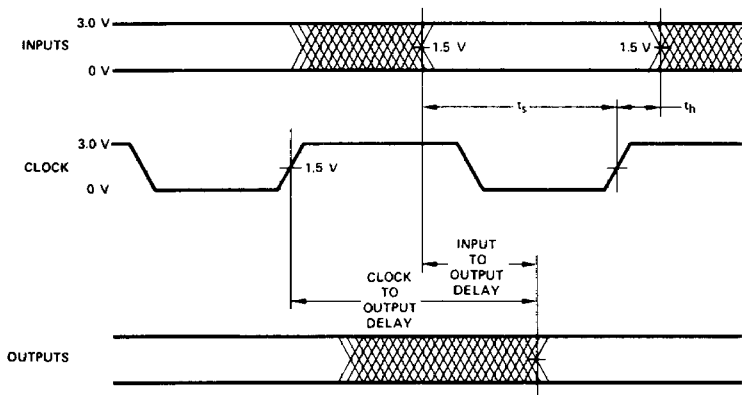
SWITCHING WAVEFORMS

KEY TO SWITCHING WAVEFORMS

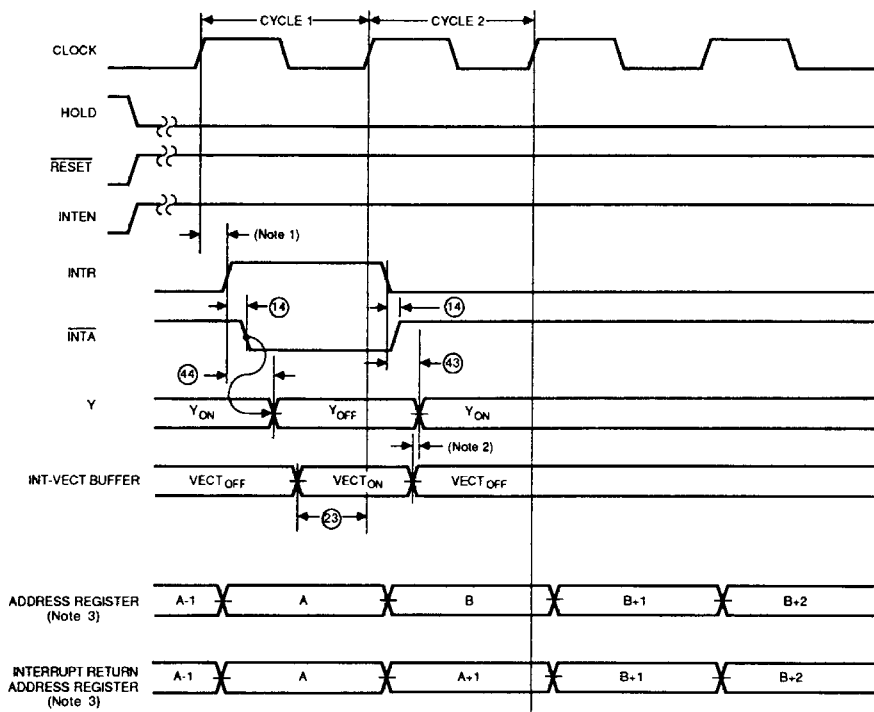
WAVEFORM	INPUTS	OUTPUTS
	MUST BE STEADY	WILL BE STEADY
	MAY CHANGE FROM H TO L	WILL BE CHANGING FROM H TO L
	MAY CHANGE FROM L TO H	WILL BE CHANGING FROM L TO H
	DON'T CARE ANY CHANGE PERMITTED	CHANGING STATE UNKNOWN
	DOES NOT APPLY	CENTER LINE IS HIGH IMPEDANCE 'OFF' STATE

KS000010

SWITCHING WAVEFORMS (Cont'd.)



WFR02990

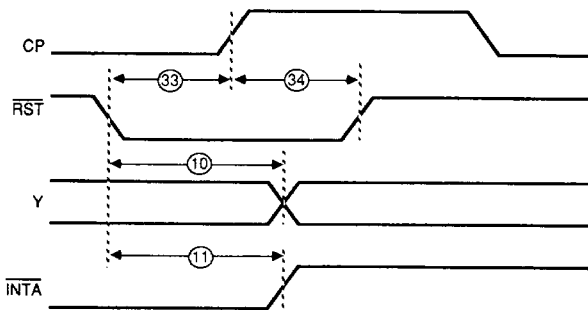


WF025100

Interrupt Timing

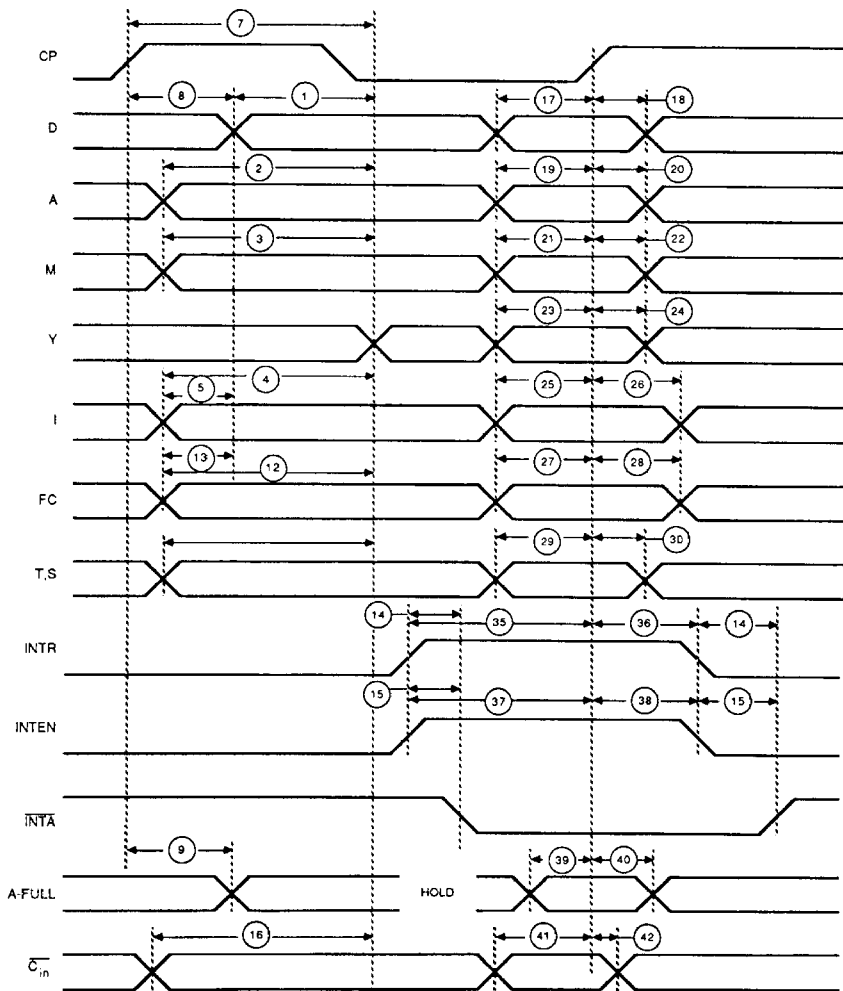
- Notes:
1. Interrupt Request comes from an interrupt-controller register. It reflects the CP \uparrow to INTR time of the interrupt controller.
 2. During Cycle 2, there may be contention on the Y-bus if the Y-bus is turned ON before the INT-VECT buffer is turned OFF.
 3. Refer to Figures 4 and 5 for definition of A and B.

SWITCHING WAVEFORMS (Cont'd.)



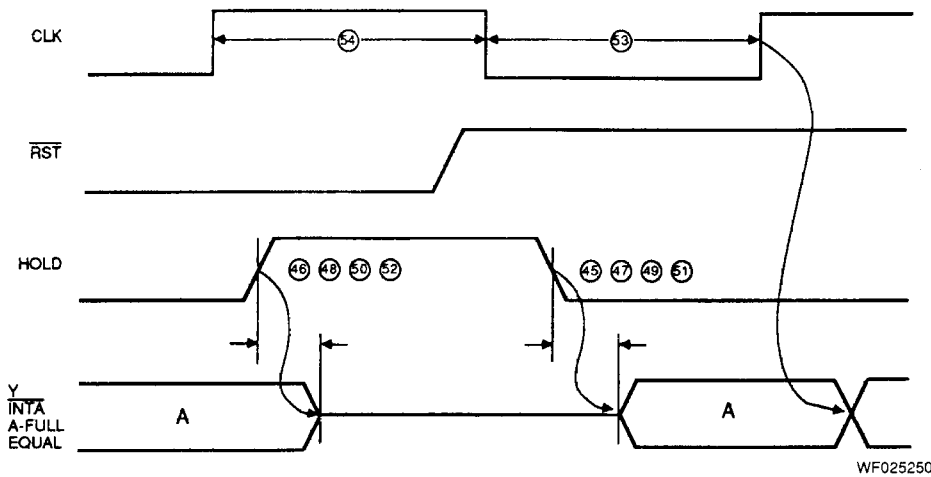
WF024770

Reset Timing



WF025320

SWITCHING WAVEFORMS (Cont'd.)

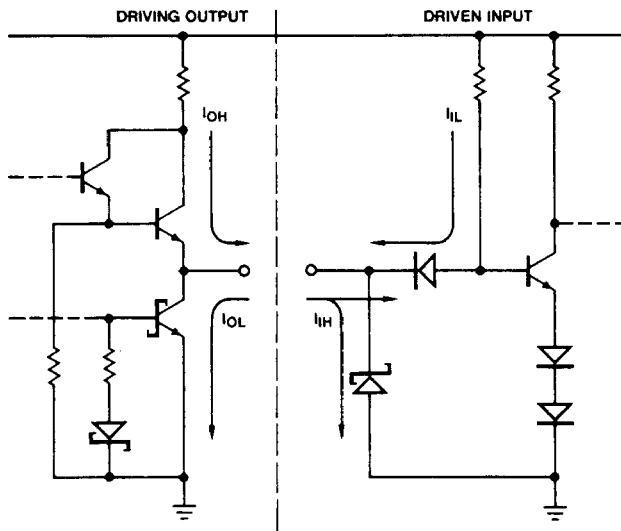


WF025250

Am29331 Hold Timing

INPUT/OUTPUT CIRCUIT DIAGRAM

(All Devices)



ICR00480