Am29CPL142/Am29CPL152

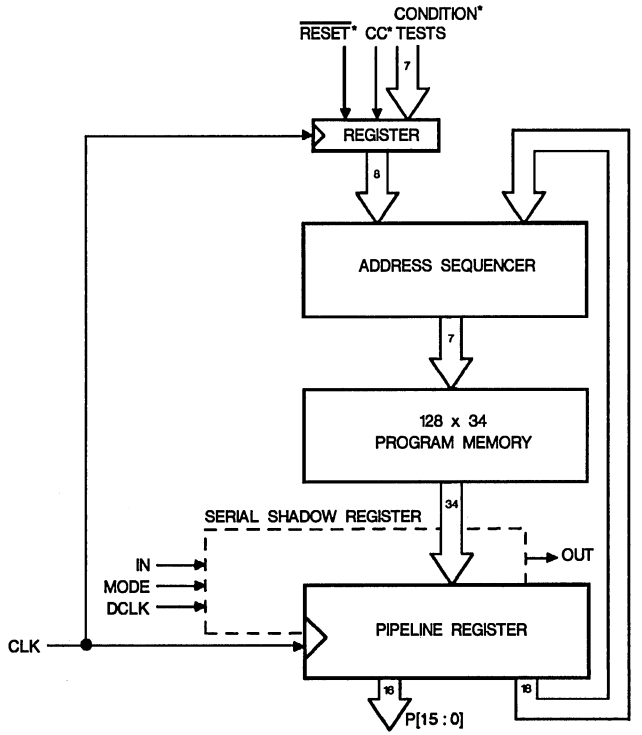
CMOS Field-Programmable Controller (FPC)

ADVANCE INFORMATION

DISTINCTIVE CHARACTERISTICS

- Implements complex state machines
- High speed, low power CMOS EPROM technology
- Eight conditional inputs, 16 outputs
- Each input can be registered or left unregistered as a programmable option
- 128-word by 34-bit CMOS EPROM
- Up to 25-MHz clock rate, 28-pin DIP and PLCC
- 28 instructions
  - Conditional branching
  - Conditional looping
  - Conditional subroutine call
  - Multiway branch
- Output instruction presents counter contents at the control outputs for implementing a larger class of state-machine designs
- A controller-expansion (EXP) cell provides address to external registered PROMs allowing more than 16 outputs
- Am29CPL142 is packaged in a 28-pin 0.6" DIP for upgrade of existing designs
- Am29CPL152 is packaged in a space-saving 28-pin 0.3" DIP or 28-pin PLCC for new designs

SIMPLIFIED BLOCK DIAGRAM



BD007542

* Each test input can be individually unregistered or left registered as a programmable option. The $\overline{\text{RESET}}$ input can also be unregistered as a programmable option.

Publication #	Rev.	Amendment
10796	A	/0

Issue Date: **November 1988**

GENERAL DESCRIPTION

The Am29CPL142 is a CMOS single-chip Field-Programmable Controller (FPC) that allows implementation of complex state machines and controllers by programming the appropriate sequence of instructions. Jumps, loops, and subroutine calls, conditionally executed based on the test inputs, provide the designer with powerful control flow primitives.

Intelligent control may be distributed throughout the system by using FPCs to control various self-contained functional units, such as register file/ALU, I/O, interrupt, diagnostic, and bus-control units.

An address sequencer, the heart of the FPC, provides the address to an internal 128-word by 34-bit EPROM. This UV-erasable and reprogrammable device utilizes proven floating-gate CMOS EPROM technology to ensure high reliability, easy programming, and better than 99.9% programming yields.

The Am29CPL142 can be expanded to address external registered memories by using the EXP option to output the program-memory address through the control output pins P[14:8].

A counter register is provided and an instruction is available to present the counter-register contents at the control outputs P[14:8]. Using this, the control outputs can be dynamically modified for implementing a larger class of state machines.

As an option, the Am29CPL142 may be programmed to have on-chip SSR diagnostics capability. Instructions can be serially shifted in, executed, and the results shifted out to facilitate system diagnostics.

A space-saving package version of the device is numbered Am29CPL152. Both ceramic windowed and plastic OTP 28-pin 0.3" DIP packages are offered, as well as 28-pin PLCC and 28-pin LCC (military) versions.

RELATED AMD PRODUCTS
Am29PL100 FAMILY MEMBERS
Field-Programmable Controllers

Part No.	Technology	Memory	Words	Inputs	Outputs	Package
Am29CPL141*	CMOS	EPROM	64	7	16	28-Pin
Am29CPL151	CMOS	EPROM	64	7	16	28-Pin x 0.3" DIP 28-Pin PLCC
Am29CPL142**	CMOS	EPROM	128	8	16	28-Pin
Am29CPL152	CMOS	EPROM	128	8	16	28-Pin x 0.3" DIP 28-Pin PLCC
Am29CPL144	CMOS	EPROM	512	8	16	28-Pin
Am29CPL154	CMOS	EPROM	512	8	16	28-Pin x 0.3" DIP 28-Pin PLCC

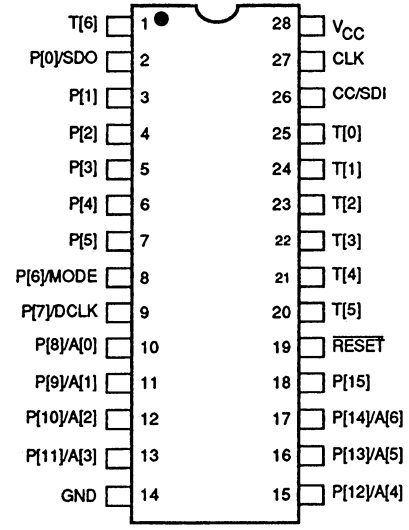
* Direct plug-in replacement for the bipolar Am29PL141
** Direct plug-in replacement for the bipolar Am29PL142

OTHER RELATED AMD PRODUCTS

Part No.	Description
Am29116A	High-Performance 16-Bit Bipolar Microprocessor
Am29C116	High-Performance 16-Bit CMOS Microprocessor
Am29C117	Two-Port Version of the Am29C116
Am2914	Vectored Priority Interrupt Controller
Am29300/C300	CMOS and Bipolar 32-Bit Microprogrammable Products Family
Am29C327	CMOS Double-Precision Floating-Point Processor
Am29C818	CMOS SSR Diagnostics Pipeline Register

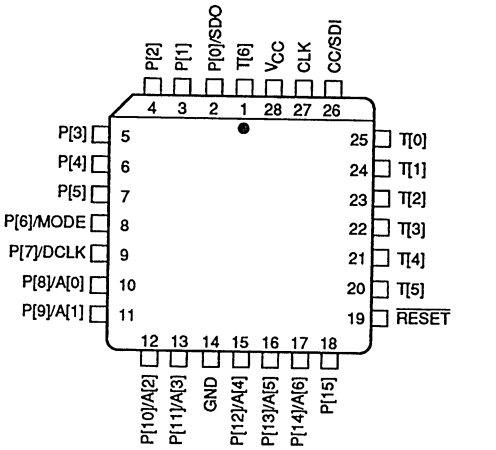
CONNECTION DIAGRAMS
Top View

DIPs*



CD011100

PLCC*



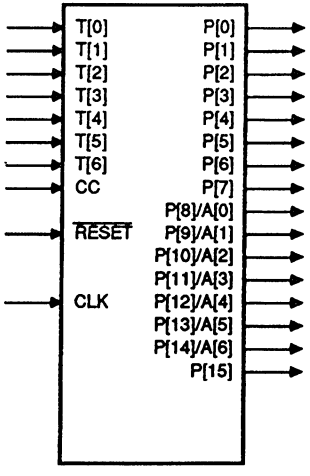
10796-001A

CD011760

Note: Pin 1 is marked for orientation.

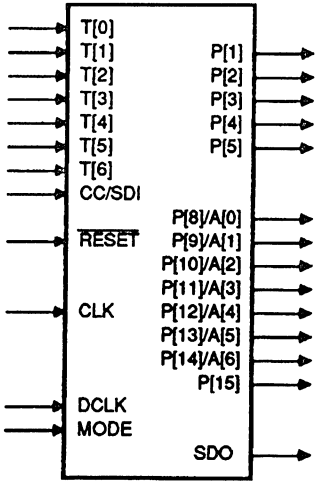
* Will also be available in 28-pin LCC; pinout identical to the PLCC. Plastic DIP and PLCC are One-Time Programmable (OTP), non-windowed packages.

LOGIC SYMBOLS



CD011230

Normal Configuration



CD011241

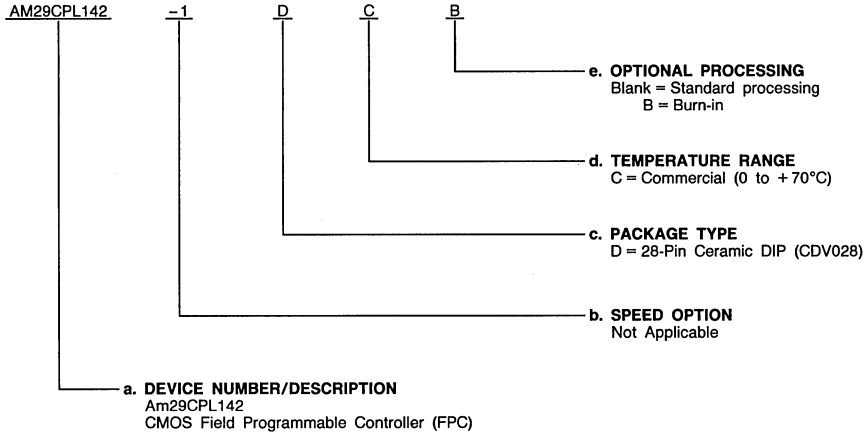
SSR Diagnostics Configuration

ORDERING INFORMATION
(Am29CPL142)

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	
AM29CPL142	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.

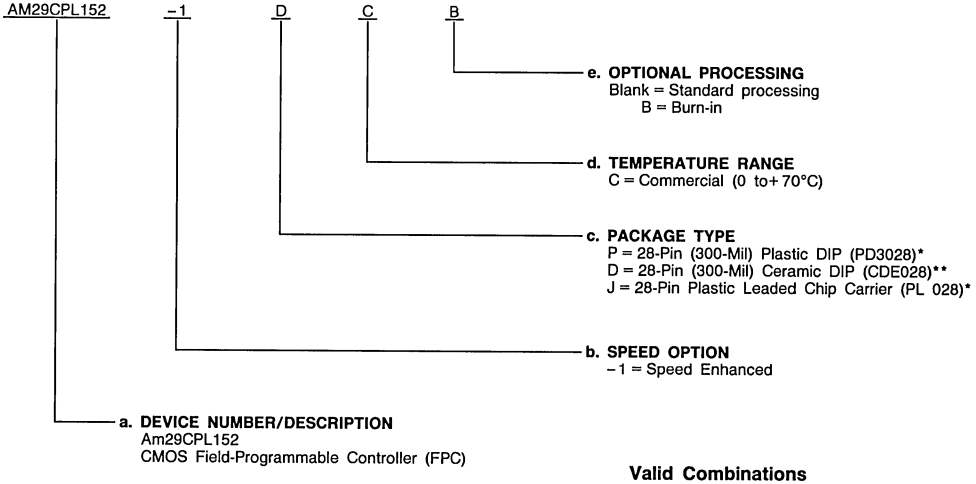
The Am29CPL142 is the ordering part number for devices packaged in 28-pin x 0.6″ ceramic windowed DIP packages. All specifications and functional descriptions in this data sheet refer equally to the Am29CPL142 and AmCPL152, except for package drawings.

ORDERING INFORMATION
(Am29CPL152)

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	
Am29CPL152	PC, DC, DCB, JC
Am29CPL152-1	

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 combinations, and to obtain additional data on AMD's standard military grade products.

* Plastic DIP and PLCC are One-Time Programmable (OTP) non-windowed packages.

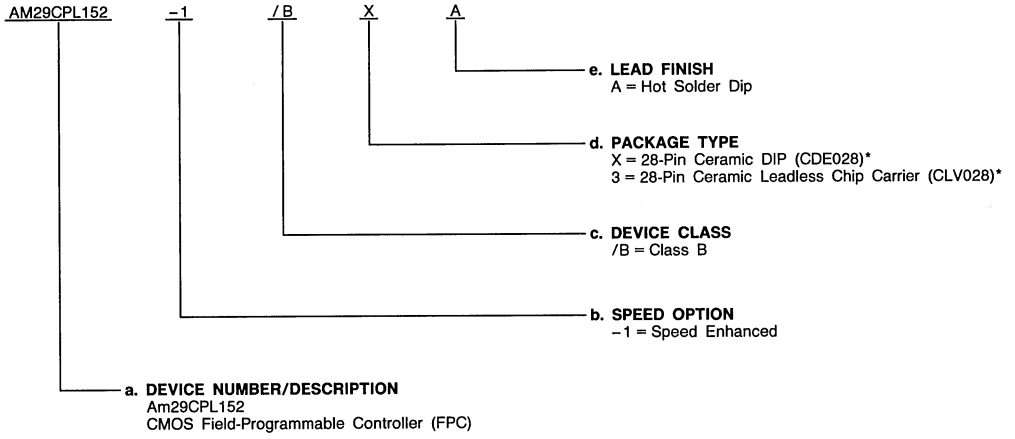
**Package is in development. Consult MIP Product Marketing for information.

MILITARY ORDERING INFORMATION
(Am29CPL152)

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) 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	
AM29CPL152	/BXA, /B3A
AM29CPL152-1	

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 consists of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

*Preliminary; package in development.

PIN DESCRIPTION

CC[SDI] Condition Code — TEST (Input)
Internally synchronized condition code test input is selected through the 4-bit test-condition select field. In the SSR mode, CC is also the Serial Data Input (SDI).

CLK Clock (Input)
The rising edge of the clock latches the PC register, count register, stack register, instruction-pipeline register, test-input register, CC register, reset input register, and the EQ flag.

P[15], P[14:8]/A[6:0] Upper General-Purpose Control (Outputs)
The upper eight general-purpose control outputs are enabled by the OE bit from the instruction-pipeline register. When OE is HIGH, these outputs are enabled; when OE is LOW, they are three stated.

A controller Expansion (EXP) cell can be programmed to set pins P[14:8] to output the program address A[6:0] from the PC MUX. These can be used to address external registered memories to provide more control outputs.

The contents of the internal count register (CREG) can also be routed to the control output pins P[14:8], using the OUTPUT instruction. Thus, the control outputs can be changed dynamically.

P[7:0] Lower General-Purpose Control (Outputs)
The lower eight general-purpose control outputs are permanently enabled. In the SSR diagnostics mode, P[7] becomes the diagnostic clock input (DCLK), P[6] becomes the diagnostic control input (MODE), and P[0] becomes the Serial Data Output (SDO). Both SDO and SDI are clocked by DCLK under control of the MODE bit.

$\overline{\text{RESET}}$ Optionally Registered Reset (input; Active LOW)
In registered mode, the first clock edge after $\overline{\text{RESET}}$ goes low latches $\overline{\text{RESET}}$ internally. The next clock edge loads the contents of location 127 decimal into the instruction-pipeline register and clears the EQ Flag. A programmable configuration bit allows the option of bypassing the synchronizing register. In this case, after $\overline{\text{RESET}}$ goes low, the output of the PC MUX is forced to all '1's (address 127 decimal) during the setup time, and the first clock edge loads the contents of location 127 decimal into the instruction-pipeline register and clears the EQ Flag. Note: by default, the $\overline{\text{RESET}}$ input is registered.

T[6:0] Optionally Registered Test (Inputs)
These Test inputs are internally synchronized. A set of programmable configuration bits allow the option of individually bypassing the synchronizing registers on the test inputs, so that inputs which are already synchronous need not experience an unnecessary delay. In conditional instructions, one of the inputs is selected according to the four bit test condition select field. T[6:0] can also be used as a branch address or as a value to be loaded into the counter.

Note: By default, the test inputs are registered.

FUNCTIONAL DESCRIPTION

Figure 1, the detailed block diagram of the Am29CPL142 FPC, shows logic blocks and interconnecting buses that permit parallel performance of different operations in a single instruction. The FPC consists of four main logic blocks: the program memory, address control logic, condition code selection logic, and instruction decode. A fifth optional block is the Serial Shadow Register (SSR).

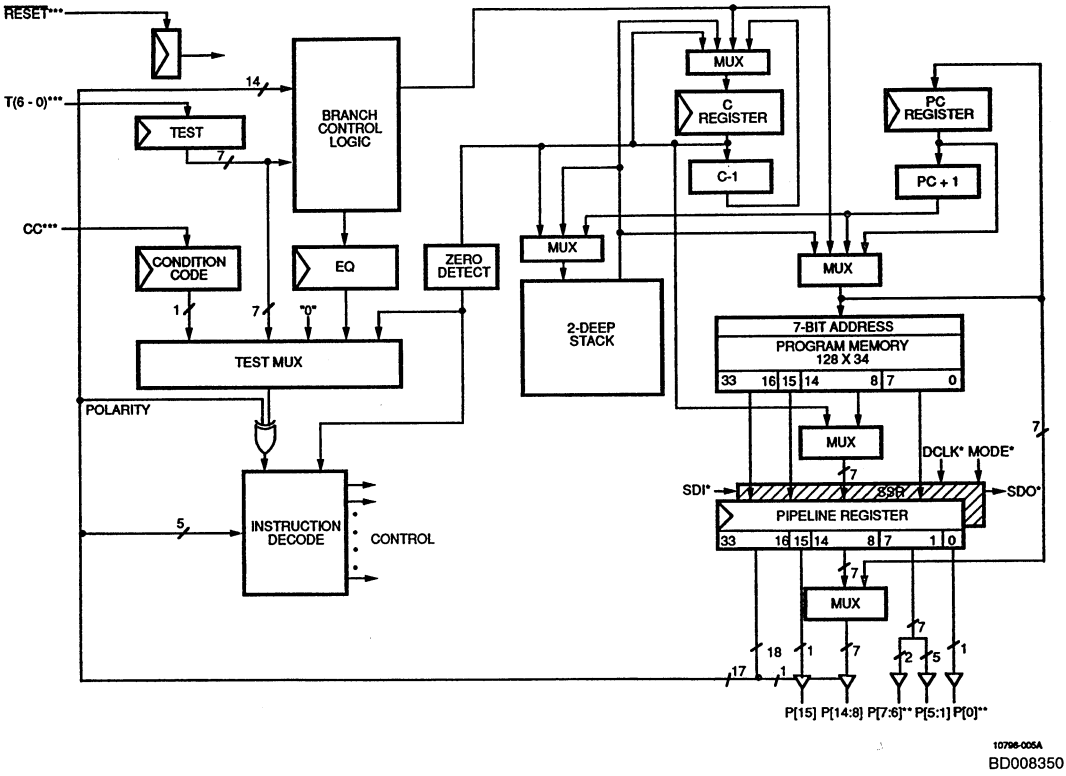
The program memory contains the user-defined instruction flow and output sequence. The address control logic addresses the program memory. This control logic supports high-level instruction functions including conditional branches, subroutine calls and returns, loops, and multiway branches. The condition code selection logic selects the condition code input to be tested when a conditional instruction is executed. The polarity of the selected condition code input is controlled by the POL bit in the microword. The instruction decode generates the control signals necessary to perform the instruction specified by the instruction part (P[33 : 16]) of the microword. The SSR enables in-system testing to isolate problems down to the IC level.

Program Memory

The FPC program memory is a 128-word by 34-bit EPROM with a 34-bit pipeline register at its output. The upper 18 bits (P[33:16]) of the pipeline register are internal to the FPC and form the instruction to control address sequencing. The format for instructions is: a 1-bit synchronous Output Enable OE, a 5-bit OPCODE, a 1-bit test polarity select POL, a 4-bit TEST condition select field, and a 7-bit immediate DATA field. The DATA field is used to provide branch addresses, test input masks, and counter values.

The lower 16 bits (P[15:0]) of the pipeline register are brought out as user-defined, general purpose control outputs. The upper eight control outputs (P[15:8]) are three-stated when OE is programmed as a LOW. The lower eight control bits (P[7:0]) are always enabled.

Outputs P[14:8] will contain the next instruction address when the optional bit EXP is set. The contents of the Counter Register can be sent to outputs P[14:8], regardless of whether the EXP fuse is programmed.



* These pins available only in SSR mode.

** These pins available only in normal mode.

*** Each of T[6:0], CC, and $\overline{\text{RESET}}$ inputs can be individually registered or left unregistered as a programmable option.

Figure 1. Am29CPL142 Detailed Block Diagram

Address Control Logic

The address control logic consists of four logic blocks:

PC GRP – the program counter multiplexer (PC MUX), program counter register (PC) and combinatorial incrementer (PC + 1)

STACK – subroutine multiplexer (S MUX) with a 2-word by 7-bit-wide stack

CNTR – count register (CREG) with counter multiplexer (C MUX), combinatorial decrementer (CREG – 1), and zero detect on count register

GOTO – multifunction branch control logic

PC GRP

The PC GRP consists of a 4:1 multiplexer, a program counter (PC) register, and a 7-bit combinatorial incrementer (PC + 1). It selects the PC, PC + 1, the branch address, or the top of stack as the next instruction address input to the program memory and the PC.

In registered mode, the first clock edge after $\overline{\text{RESET}}$ goes low latches $\overline{\text{RESET}}$ internally. The next clock edge loads the contents of location 127 decimal into the instruction-pipeline register and clears the EQ Flag. A programmable configuration bit allows the option of bypassing the synchronizing register. In this case, after $\overline{\text{RESET}}$ goes low, the output of the PC MUX is forced to all '1's (address 127 decimal) during the setup time, and the first clock edge loads the contents of location 127 decimal into the instruction-pipeline register and clears the EQ Flag. Note: by default, the $\overline{\text{RESET}}$ input is registered.

STACK

The stack block consists of a 3:1 multiplexer (S MUX) that stores the data into the topmost location of the stack. The S MUX chooses from three sources: PC + 1, count register, and the top of the stack (for holding). PC + 1 is the input source when doing subroutine calls. PC MUX is the output destination when a return-from-subroutine instruction is performed.

CNTR

The CNTR block consists of a 4:1 multiplexer (C MUX) feeding a 7-bit count register (CREG) which, in turn, outputs to a combinatorial decrementer (CREG – 1) and a zero-detect circuit. The CNTR block is typically used for timing functions and iterative loop counting.

The C MUX has the following input sources: top of stack, the branch-logic output, CREG – 1, and the CREG (for holding).

GOTO

The GOTO (branch control) logic block performs three functions:

(1) It provides a 7-bit value directly from the DATA field in the instruction pipeline register.

(2) It provides a 7-bit branch address from the TEST inputs T[6:0] masked by the DATA field from the instruction pipeline register.

(3) It compares the TEST inputs T[6:0] (masked by the MASK field in the instruction pipeline register) with the constant field comparison (see Instruction Format) in the pipeline register.

The EQ flip-flop is set according to the following expression:

EQ = [(TEST .AND. MASK) .XNOR. constant] .OR. EQ

The EQ flag can be tested by the condition-code selection logic. Multiple tests of any group of T inputs in a manner analogous to sum-of-products can be performed, since a no-match comparison does not reset the EQ flag. Any conditional branch on EQ will reset the EQ flag. Conditional returns on EQ will not change the EQ flag. $\overline{\text{RESET}}$ input LOW will reset the EQ flag.

NOTE: A zero in the MASK field blocks the corresponding bit in the TEST field; a one activates the corresponding bit.

The constant field bits that correspond to masked test field bits must be zero. A zero is substituted for masked test field bits. This masking operation is independent of the value of the POL bit.

Condition Code Selection Logic

The condition code selection logic consists of a 16:1 multiplexer. The 16 condition inputs are the seven test bits, the condition code input CC, the EQ flag, CREG ZERO status, and six UNCOND test conditions connected to zero for the unconditional mode. The TEST field in the pipeline register (P[26:23]) selects one of the 16 conditions. If one of the UNCOND is chosen, and the POL bit is a one, the instruction is executed with a "forced PASS" condition. If one of the UNCOND is chosen, and the POL bit is a zero, the instruction is executed with a "forced FAIL" condition. See opcode descriptions for more details.

The polarity bit POL in P[27] of the instruction pipeline register allows the user to test for either pass/true or fail/false condition. Refer to Table 2 for details.

Instruction Decode

The instruction decoder is a PLA that generates the control for 28 different instructions. The decoder inputs include the OPCODE field P[32:28], the zero detection flag from the CNTR, and the selected test condition code from the condition code select logic.

Operational Modes

The Am29CPL142 operates as a 7 bit microcontroller in normal mode, and there are several configuration bits which can be programmed to modify this normal operation (see figure below). By programming the EXP bit, the output pins P[14:8] can be used as the microprogram address for external registered memories to get a wider control output. The SSR bit allows on-chip diagnostic capabilities for in-system testing. The remaining bits serve to individually select whether the input pins will be internally synchronized or not. The default setting of these bits (unprogrammed, 1) will cause each pin to be internally synchronized, and so programming a given bit (to 0) will cause that corresponding input to bypass the synchronizing register.

EPROM – Configuration Bits

					EXP	SSR	Reset BYPASS

EPROM – Configuration Bits (input register bypass)

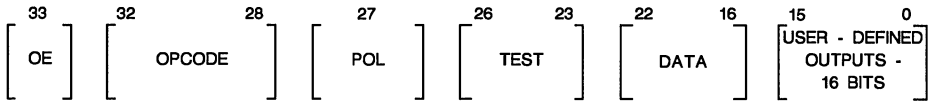
CC BYPASS	T_6 BYPASS	T_5 BYPASS	T_4 BYPASS	T_3 BYPASS	T_2 BYPASS	T_1 BYPASS	T_0 BYPASS

The SSR option provides on-chip diagnostic capabilities for in-system testing. This is accomplished by setting the appropriate conditions on the MODE control input. Note that this can only be done if the SSR diagnostics cell has been programmed. See Table 1.

The SSR diagnostics configuration activates a 34-bit-wide, D-type register, called a "shadow" register, on the pipeline register inputs. The shadow register can be serially loaded from the SDI pin, parallel loaded from the pipeline register, or held. The pipeline register can be loaded from the program memory in normal operation or from the shadow register during diagnostics. A redefinition of four device pins is required to control the different diagnostics functions. CC also functions as the Serial Data Input (SDI), P[0] becomes the serial data output (SDO), P[7] becomes the diagnostic clock (DCLK), and P[6] becomes the diagnostic mode control (MODE). The various diagnostic and normal modes are shown in Table 1.

Serially loading a test instruction into the shadow register and parallel loading the shadow register contents into the pipeline register forces execution of the test instruction. The test result can then be clocked into the pipeline register, as in normal operation mode, parallel loaded into the shadow register, and serially shifted out for system diagnostics.

Am29CPL142 General Instruction Format



DFR00733

WHERE:

OE = Synchronous Output Enable for P[15:8].

OPCODE = A 5-bit opcode field for selecting one of the 27 single data-field instructions.

POL = A 1-bit test condition polarity select. (Refer to Table 2)

TEST = A 4-bit test condition select.

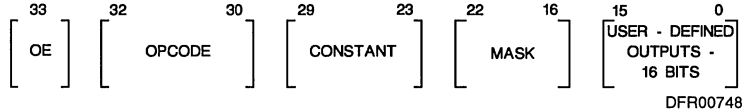
TEST[3:0]	UNDER TEST
0000	T[0]
0001	T[1]
0010	T[2]
0011	T[3]
0100	T[4]
0101	T[5]
0110	T[6]
0111	CC
1000	EQ
1001	CREG ZERO
1010-1111	UNCOND [0]

The polarity bit POL in an instruction allows the user to test for a pass/true or fail/false condition as shown in Table 2. An unconditional true is set by selecting UNCOND and POL = 1.

DATA = A 7-bit conditional branch address, test input mask, or counter value field designated as PL in instruction mnemonics.

The special two data field comparison instruction format is shown below:

Am29CPL142 Comparison Instruction Format



DFR00748

WHERE:

OE = Synchronous Output Enable for P[15:8].

OPCODE = Compare instruction (binary 100).

CONSTANT = A 7-bit constant for equal to comparison with T*M.

MASK = A 7-bit mask field for masking the incoming T[6:0] inputs.

TABLE 1.

Inputs				Outputs			Operation
SDI	MODE	DCLK	CLK	SDO	Shadow Register	Pipeline Register	
D	L	↑	H,L,↓	S_o	$S_{i-1} \leftarrow S_i$ $S_{31} \leftarrow D$	Hold	Serial Right Shift Register
CC**	L	H,L,↓	↑	L	Hold	$P_i \leftarrow EPROM_i$	Load Pipeline Register from EPROM
L	H	↑	H,L,↓	L	$S_i \leftarrow P_i$	Hold	Load Shadow Register from Pipeline* Register
X	H	H,L,↓	↑	SDI	Hold	$P_i \leftarrow S_i$	Load Pipeline Register from Shadow Register
H	H	↑	H,L,↓	H	Hold	Hold	Hold Shadow Register

* S7, S6 are undefined. If P[33] in the microword is a one, S_{15}-S_8 are loaded from the pipeline register. If P[33] in the microword is a zero, $S_{15} - S_8$ are loaded from an external source.

** During normal operation, this pin behaves as the CC test input.

FUNCTION TABLE DEFINITIONS

INPUTS

H = HIGH
L = LOW
X = Don't Care
D = Serial Data
↑ = LOW-to-HIGH transition
↓ = HIGH-to-LOW transition

TABLE 2.

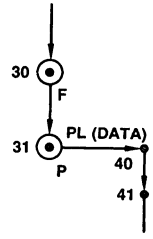
Input Condition Being Tested	POL	Condition
0	0	Fail
0	1	Pass
1	0	Pass
1	1	Fail

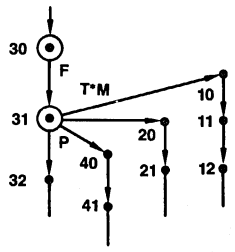
Am29CPL142 INSTRUCTION SET DEFINITION

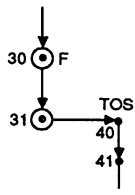
The stack consists of two registers; the top one is labeled Top of Stack (TOS), the bottom one is labeled Bottom of Stack (BOS).

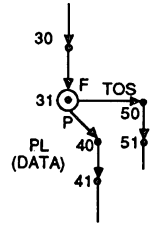
● = Other instruction
⊙ = Instruction being described
O = Register in part
P = Test Pass
F = Test Fail
X, Y are arbitrary values in the CREG or STACK

Opcode	Mnemonic	Description	Execution Example	Register Transfer Description

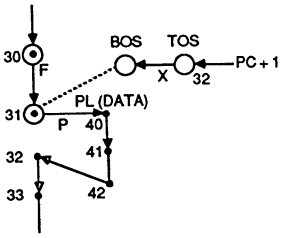
19	GOTOPL	**IF (cond) THEN GOTO PL (data)** Conditional branch to the address in the PL (DATA field). The EQ flag will be reset if the test field selects it and the condition passes. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.	 PF001420	If (cond = true) Then
				PC ← PL(DATA)
				Else
				PC ← PC + 1

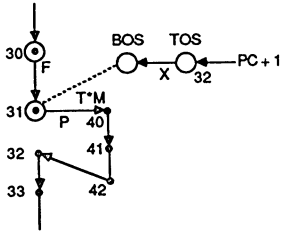
1F	GOTOTM	**IF (cond) THEN GOTO TM (data)** Conditional branch to the address defined by the T*M (T[6:0] under bitwise mask from the DATA field). This instruction is intended for multiway branches. The EQ flag will be reset if the test field selects it and the condition passes. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.	 PF001441	If (cond = true) Then
				PC ← T*M
				Else
				PC ← PC + 1

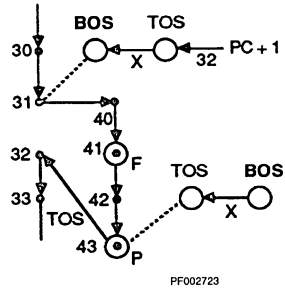
03	GOTOSTK	**IF (cond) THEN GOTO (STACK)** Conditional branch to the address at the top of the stack, or else continue. The EQ flag will be reset if the test field selects it and the condition passes. The instruction is executed with a forced PASS condition if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition if the test field is UNCOND and POL = 0.	 PF002680	If (cond = true) Then
				PC ← TOS
				Else
				PC ← PC + 1

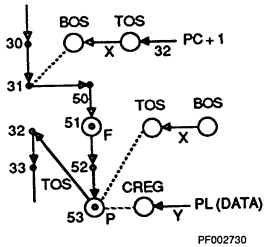
18	FORK	**IF (cond) THEN GOTO PL (data) ELSE GOTO (STACK)** Conditional branch to the address in the PL (DATA field) or the TOS. A branch to PL is taken if the condition is true and a branch to TOS if false. The EQ flag will be reset if the test field selects it and the condition passes. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.	 PF002690	If (cond = true) Then
				PC ← PL(DATA)
				Else
				PC ← TOS

Opcode	Mnemonic	Description	Execution Example	Register Transfer Description

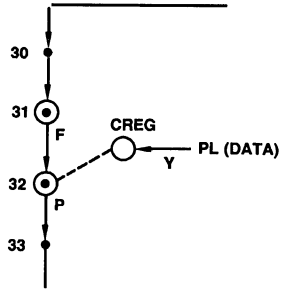
1C	CALPL	**IF (cond) THEN CALL PL (data)**
		Conditional jump to subroutine at the address in the PL (DATA field). The PC + 1 is pushed into the TOS as the return address. The EQ flag will be reset if the test field selects it and the condition passes. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF002700
				If (cond = true) Then
				BOS ← TOS
				TOS ← PC + 1
				PC ← PL(DATA)
				Else
				PC ← PC + 1

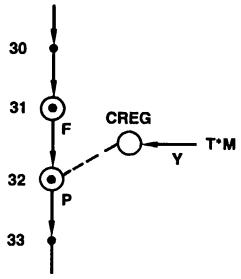
1E	CALTM	**IF (cond) THEN CALL TM (data)**
		Conditional jump to subroutine at the address specified by the T*M (T[6:0] under bitwise mask from the DATA field). The PC + 1 is pushed into the TOS as the return address. The EQ flag will be reset if the test field selects it and the condition passes. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF002710
				If (cond = true) Then
				BOS ← TOS
				TOS ← PC + 1
				PC ← T*M
				Else
				PC ← PC + 1

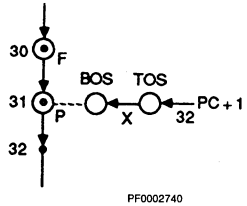
02	RET	**IF (cond) THEN RET**
		Conditional return from subroutine. The TOS provides the return from subroutine address and the stack is popped. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF002723
				If (cond = true) Then
				PC ← TOS
				TOS ← BOS
				Else
				PC ← PC + 1

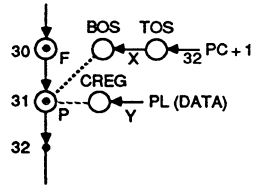
00	RETPL	**IF (cond) THEN RET, LOAD PL (data)**
		Conditional return from subroutine and load the CREG from the PL (DATA field). The TOS provides the return from subroutine address and the stack is popped. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF002730
				If (cond = true) Then
				PC ← TOS
				TOS ← BOS
				CREG ← PL(DATA)
				Else
				PC ← PC + 1

Opcode	Mnemonic	Description	Execution Example	Register Transfer Description

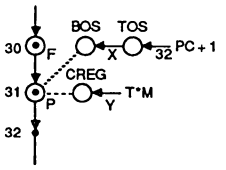
04	LDPL	**IF (cond) THEN LOAD PL (data)**
		Conditional Load the CREG from the PL (DATA field). The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF001511
				If (cond = true) Then
				CREG ← PL(DATA)
				PC ← PC + 1
				Else
				PC ← PC + 1

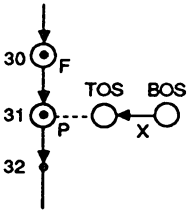
06	LDTM	**IF (cond) THEN LOAD TM (data)**
		Conditional load the CREG from the T*M (T[6:0] inputs under bitwise mask from the DATA field). The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF001521
				If (cond = true) Then
				CREG ← T*M
				PC ← PC + 1
				Else
				PC ← PC + 1

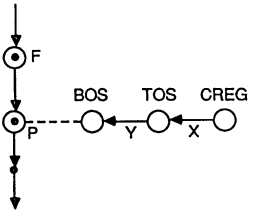
15	PSH	**IF (cond) THEN PUSH**
		Conditional push the PC + 1 into the TOS. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF0002740
				If (cond = true) Then
				BOS ← TOS
				TOS ← PC + 1
				PC ← PC + 1
				Else
				PC ← PC + 1

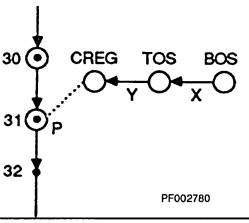
14	PSHPL	**IF (cond) THEN PUSH, LOAD PL (data)**
		Conditional push the PC + 1 into the TOS and load the CREG from the PL (DATA field). The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF002750
				If (cond = true) Then
				BOS ← TOS
				CREG ← PL(DATA)
				TOS ← PC + 1
				PC ← PC + 1
				Else
				PC ← PC + 1

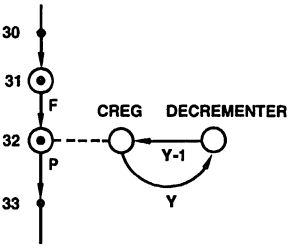
Opcode	Mnemonic	Description	Execution Example	Register Transfer Description

16	PSHTM	**IF (cond) THEN PUSH, LOAD TM (data)**
		Conditional push the PC + 1 into the TOS and load the CREG from the T*M (T[6:0] under bitwise mask from the DATA field). The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF002770
				If (cond = true) Then
				BOS ← TOS
				CREG ← T*M
				TOS ← PC + 1
				PC ← PC + 1
				Else
				PC ← PC + 1

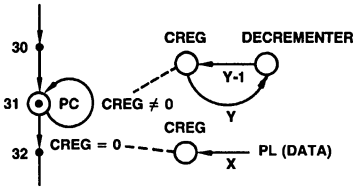
07	POP	**IF (cond) THEN POP**
		Conditional Pop the TOS. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF002760
				If (cond = true) Then
				TOS ← BOS
				PC ← PC + 1
				Else
				PC ← PC + 1

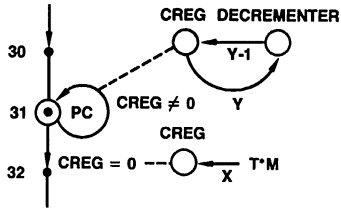
05	PSHCNTR	**IF (cond) THEN PUSH (CREG)**
		Conditional push CREG contents to top of stack. The instruction is executed with a forced PASS condition if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition if the test field is UNCOND and POL = 0.
			
			PF002790
				If (cond = true) Then
				BOS ← TOS
				TOS ← CREG
				PC ← PC + 1
				Else
				PC ← PC + 1

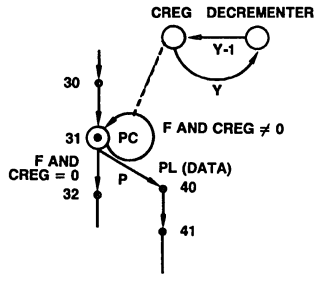
17	POPCNTR	**IF (cond) THEN POP TO (CREG)**
		Conditional pop TOS into CREG. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF002780
				If (cond = true) Then
				CREG ← TOS
				TOS ← BOS
				PC ← PC + 1
				Else
				PC ← PC + 1

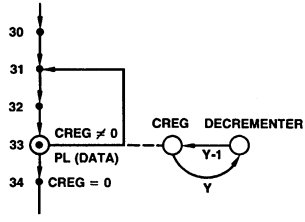
0B	DEC	**IF (cond) THEN DEC**
		Conditional decrement of the CREG. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF001621
				If (cond = true) Then
				CREG ← CREG - 1
				PC ← PC + 1
				Else
				PC ← PC + 1

Opcode	Mnemonic	Description	Execution Example	Register Transfer Description

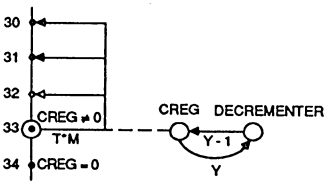
0C	DECPL	**WHILE (CREG <> 0) WAIT ELSE LOAD PL (data)**
		Conditional Hold until the counter is equal to zero, then load CREG from the PL (DATA field). This instruction is intended for timing waveform generation. If the CREG is not equal to zero, the same instruction is refetched while CREG is decremented. Timing is complete when the CREG is equal to zero, causing the next instruction to be fetched and the CREG to be reloaded from PL. This instruction does not depend on the pass/fail condition.
			
			PF001632
				While (CREG ≠ 0)
				CREG ← CREG – 1
				PC ← PC
				End While
				CREG ← PL(DATA)
				PC ← PC + 1

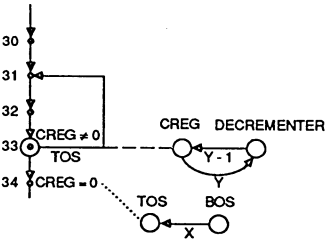
0E	DECTM	**WHILE (CREG <> 0) WAIT ELSE LOAD TM (data)**
		Conditional Hold until the counter is equal to zero, then load CREG from the T*M (T[6:0] under bitwise mask from the DATA field). This instruction is intended for timing waveform generation. If the CREG is not equal to zero, the same instruction is refetched while the CREG is decremented. Timing is complete when the CREG is equal to zero, causing the next instruction to be refetched and the CREG to be reloaded from T*M. This instruction does not depend on the pass/fail condition.
			
			PF001641
				While (CREG ≠ 0)
				CREG ← CREG – 1
				PC ← PC
				End While
				CREG ← T*M
				PC ← PC + 1

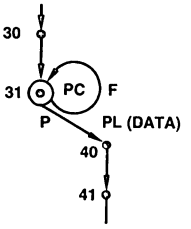
1D	DECGOPL	**IF (cond) THEN GOTO PL (data) ELSE WHILE (CREG <> 0) WAIT**
		Conditional Hold/Count. The current instruction will be refetched and the CREG decremented until the condition under test becomes true or the counter is equal to zero. If the condition becomes true, a branch to the address in the PL (DATA field) is executed. If the counter becomes zero without the condition becoming true, a CONTINUE is executed. The EQ flag will be reset if the test field selects it and the condition passes. The instruction is executed with a forced PASS condition if the TEST field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition if the TEST field is UNCOND and POL = 0.
			
			PF001651
				While (cond = false)
				If (CREG ≠ 0)
				CREG ← CREG – 1
				PC ← PC
				Else
				PC ← PC + 1
				End While
				PC ← PL(DATA)

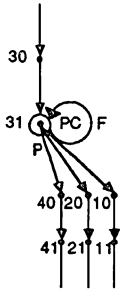
08	LPPL	**WHILE (CREG <> 0) LOOP TO PL (data)**
		Conditional loop to the address in the PL (DATA field). This instruction is intended to be placed at the bottom of an iterative loop. If the CREG is not equal to zero, it is decremented (signifying completion of an iteration), and a branch to the address in the PL (DATA field) (top of the loop) is executed. If the CREG is equal to zero, looping is complete and the next sequential instruction is executed. This instruction does not depend on the pass/fail condition. The EQ flag will be reset if the test field selects it and CREG is not equal to zero.
			
			PF001671
				While (CREG ≠ 0)
				CREG ← CREG – 1
				PC ← PL (DATA)
				End While
				PC ← PC + 1

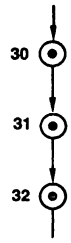
Opcode	Mnemonic	Description	Execution Example	Register Transfer Description

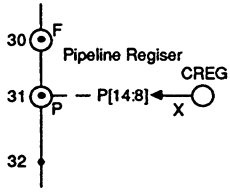
0A	LPTM	**WHILE (CREG <> 0) LOOP TO TM (data)**
		Conditional loop to the address T*M (T[6:0] under bitwise mask from the DATA field). This instruction should be placed at the bottom of an iterative loop. If CREG is not equal to zero, it is decremented (signifying completion of an iteration), and a branch to the address specified by T*M (top of the loop) is executed. If CREG is equal to zero, looping is complete and the next sequential instruction is executed. This does not depend on the pass/fail condition. The EQ flag will be reset if the test field selects it and CREG is not equal to zero.
			
			PF002800
				While (CREG ≠ 0)
				CREG ← CREG - 1
				PC = T*M
				End While
				PC ← PC + 1

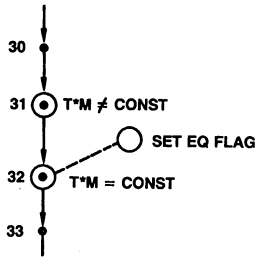
0F	LPSTK	**WHILE (CREG <> 0) LOOP TO (STACK)**
		Conditional loop to the address in the TOS. If CREG ≠ 0, the CREG is decremented and a branch to the TOS address is executed. If the CREG = 0, looping is complete, the stack is popped, and the next sequential instruction is executed. This instruction does not depend on the pass/fail condition. The EQ flag will be reset if the test field selects it and CREG is not equal to zero.
			
			PF002810
				While (CREG ≠ 0)
				CREG ← CREG - 1
				PC ← TOS
				End While
				TOS ← BOS
				PC ← PC + 1

1A	WAITPL	**IF (cond) THEN GOTO PL (data) ELSE WAIT**
		Conditional Hold. The current instruction will be refetched and executed until the condition under test becomes true. When true, a branch to the address in the PL (DATA field) is executed. The EQ flag will be reset if the test field selects it and the condition passes. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instructicn is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF001660
				If (cond = true) Then
				PC ← PL(DATA)
				Else
				PC ← PC

1B	WAITTM	**IF (cond) THEN GOTO TM (data), ELSE WAIT**
		Conditional Hold. The current instruction will be refetched and executed until the condition under test becomes true. When the condition is true, a branch to the T*M address (T[6:0] under bitwise mask from the DATA field) is executed. The EQ flag will be reset if the test field selects it and the condition passes. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF002660
				If (cond = true) Then
				PC ← T*M
				Else
				PC ← PC

Opcode	Mnemonic	Description	Execution Example	Register Transfer Description
0D	CONT	**CONTINUE** The next sequential instruction is fetched unconditionally. This instruction can also be used to reset the EQ flag by selecting EQ in the TEST field.	 PF001690	PC ← PC + 1

01	OUTPUT	**IF (cond) THEN OUTPUT**
		The CREG contents will be output on pins P[14:8] during the next clock cycle. Care should be taken to ensure that the outputs are enabled for the next sequential instruction by setting the microcode bit OE = 1. The instruction is executed with a forced PASS condition, if the test field is UNCOND and POL = 1. The instruction is executed with a forced FAIL condition, if the test field is UNCOND and POL = 0.
			
			PF002820
				If (cond = true) Then
				P[14 : 8] ← CREG
				PC ← PC + 1
				Else
				PC ← PC + 1

10 – 13 (100XX binary)	CMP	**CMP TM (mask) TO PL (data)** This instruction performs bitwise exclusive-OR of T*M (T[6:0] under bitwise mask from the MASK field P[22:16]) with CONSTANT (P[29:23]). If T*M equals CONSTANT, the EQ flag is set to one, so that a branch may occur in a following instruction. If not equal, the EQ flag is unaffected. This allows sequences of compares, in a manner analogous to sum-of-products, to be performed which can be followed by a single conditional branch if one or more of the comparisons were true. Note: The EQ flag is set to zero on reset or when EQ is selected as the test condition in a branch. Conditional returns on EQ leave the flag unchanged. **Constant field bits that correspond to masked test-input bits must be zero.** This instruction does not depend on the pass/fail condition.	 PF001701	Compare T*M and CONSTANT EQ = ((T [6:0] .AND. MASK) .XNOR. CONSTANT) .OR. EQ PC ← PC + 1

MICROINSTRUCTIONS BASED ON TEST CONDITIONS

Opcode	Mnemonic	Assembler Statement	Condition Pass				Condition Fail				Notes
			PC MUX	STACK	CREG	EQ FLAG	PC MUX	STACK	CREG	EQ FLAG	
00	RETPL	IF (cond) THEN RET, LOAD PL (data)	TOS	Pop	Load PL	NC	PC + 1	Hold	Hold	NC	5
01	OUTPUT	IF (cond) THEN OUTPUT	PC + 1	Hold	Hold	NC	PC + 1	Hold	Hold	NC	1
02	RET	IF (cond) THEN RET	TOS	Pop	Hold	NC	PC + 1	Hold	Hold	NC	5
03	GOTOSTK	IF (cond) THEN GOTO (STACK)	TOS	Hold	Hold	Reset	PC + 1	Hold	Hold	NC	3
04	LDPL	IF (cond) THEN LOAD PL (data)	PC + 1	Hold	Load PL	NC	PC + 1	Hold	Hold	NC	
05	PSHCNTR	IF (cond) THEN PUSH (CREG)	PC + 1	Push CREG	Hold	NC	PC + 1	Hold	Hold	NC	6
06	LDTM	IF (cond) THEN LOAD TM (data)	PC + 1	Hold	Load TM	NC	PC + 1	Hold	Hold	NC	
07	POP	IF (cond) THEN POP	PC + 1	Pop	Hold	NC	PC + 1	Hold	Hold	NC	5
0B	DEC	IF (cond) THEN DEC	PC + 1	Hold	DEC	NC	PC + 1	Hold	Hold	NC	
14	PSHPL	IF (cond) THEN PUSH, LOAD PL (data)	PC + 1	Push PC + 1	Load PL	NC	PC + 1	Hold	Hold	NC	6
15	PSH	IF (cond) THEN PUSH	PC + 1	Push PC + 1	Hold	NC	PC + 1	Hold	Hold	NC	6
16	PSHTM	IF (cond) THEN PUSH, LOAD TM (data)	PC + 1	Push PC + 1	Load TM	NC	PC + 1	Hold	Hold	NC	6
17	POPCNTR	IF (cond) THEN POP TO (CREG)	PC + 1	Pop	Load TOS	NC	PC + 1	Hold	Hold	NC	5
18	FORK	IF (cond) THEN GOTO PL (data) ELSE GOTO (STACK)	PL	Hold	Hold	Reset	TOS	Hold	Hold	NC	3
19	GOTOPL	IF (cond) THEN GOTO PL (data)	PL	Hold	Hold	Reset	PC + 1	Hold	Hold	NC	3
1A	WAITPL	IF (cond) THEN GOTO PL (data) ELSE WAIT	PL	Hold	Hold	Reset	PC	Hold	Hold	NC	3
1B	WAITTM	IF (cond) THEN GOTO TM (data) ELSE WAIT	TM	Hold	Hold	Reset	PC	Hold	Hold	NC	3
1C	CALPL	IF (cond) THEN CALL PL (data)	PL	Push PC + 1	Hold	Reset	PC + 1	Hold	Hold	NC	3, 6
1E	CALTM	IF (cond) THEN CALL TM (data)	TM	Push PC + 1	Hold	Reset	PC + 1	Hold	Hold	NC	3, 6
1F	GOTOTM	IF (cond) THEN GOTO TM (data)	TM	Hold	Hold	Reset	PC + 1	Hold	Hold	NC	3

Key: TOS = Top of Stack
PC = Program Counter
CREG = Counter Register
PL = Pipeline (DATA) Field
TM = Test Inputs Masked by PL (DATA) Field
DEC = Decrement
BOS = Bottom of Stack
NC = No Change

Notes: See notes on next page.

MICROINSTRUCTION DEPENDENT ON CREG

Opcode	Mnemonic	Assembler Statement	CREQ = 0				CREG ≠ 0				Notes
			PC MUX	STACK	CREG	EQ FLAG	PC MUX	STACK	CREG	EQ FLAG	
08	LPPL	WHILE (CREG<>0) LOOP TO PL (data)	PC + 1	Hold	Hold	NC	PL	Hold	DEC	Reset	4
0A	LPTM	WHILE (CREG<>0) LOOP TO TM (data)	PC + 1	Hold	Hold	NC	TM	Hold	DEC	Reset	4
0C	DECPL	WHILE (CREG<>0) WAIT ELSE LOAD PL (data)	PC + 1	Hold	Load PL	NC	PC	Hold	DEC	NC	
0E	DECTM	WHILE (CREG<>0) WAIT ELSE LOAD TM (data)	PC + 1	Hold	Load TM	NC	PC	Hold	DEC	NC	
0F	LPSTK	WHILE (CREG<>0) LOOP TO (STACK)	PC + 1	Pop	Hold	NC	TOS	Hold	DEC	Reset	4

MICROINSTRUCTION DEPENDENT ON TEST CONDITION AND CREG VALUE

Opcode	Mnemonic	Assembler Statement	CREG Content	Condition Pass				Condition Fail				Notes
				PC MUX	STACK	CREG	EQ FLAG	PC MUX	STACK	CREG	EQ FLAG	
1D	DECGOPL	IF (cond) THEN GOTO PL (data) ELSE WHILE (CREG<>0) WAIT	≠0	PL	Hold	Hold	Reset	PC	Hold	DEC	NC	3
			= 0	PL	Hold	Hold	Reset	PC + 1	Hold	Hold	NC	3

UNCONDITIONAL MICROINSTRUCTIONS

Opcode	Mnemonic	Assembler Statement	PC MUX	STACK	CREG	EQ FLAG	Notes
0D	CONT	CONTINUE	PC + 1	Hold	Hold	NC	2
10-13 (100XX) Binary	CMP	CMP TM (mask) TO PL (constant)	PC + 1	Hold	Hold	Set	7

Key: TOS = Top of Stack
BOS = Bottom of Stack
PC = Program Counter
CREG= Counter Register
PL = Pipeline (DATA) Field
TM = Test Inputs Masked by MASK Field
DEC = Decrement
NC = No Change

Notes: 1. If Condition Passes, Output CREG contents on next clock cycle.
2. If Condition = EQ, reset EQ flag.
3. If Condition = EQ and Condition Passes, reset EQ flag.
4. If Condition = EQ and CREG ≠ 0, reset EQ flag.
5. When Stack is popped, BOS is transferred to TOS.
6. When Stack is pushed, TOS is transferred to BOS before value is written into TOS.
7. Set EQ Flag if CONST field = T*M.

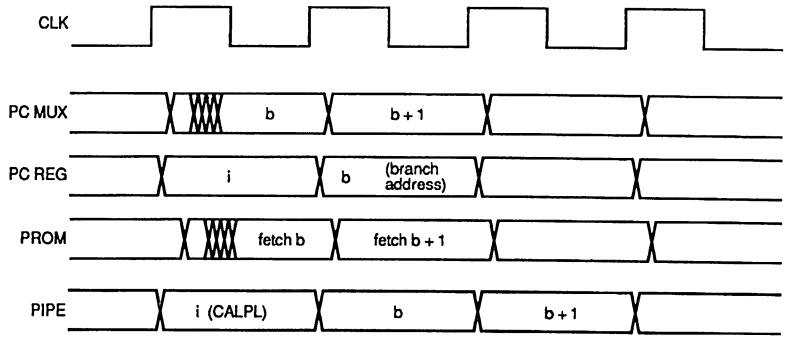


Figure 2. Conditional Branch/Jump Instruction (Example: CALPL)

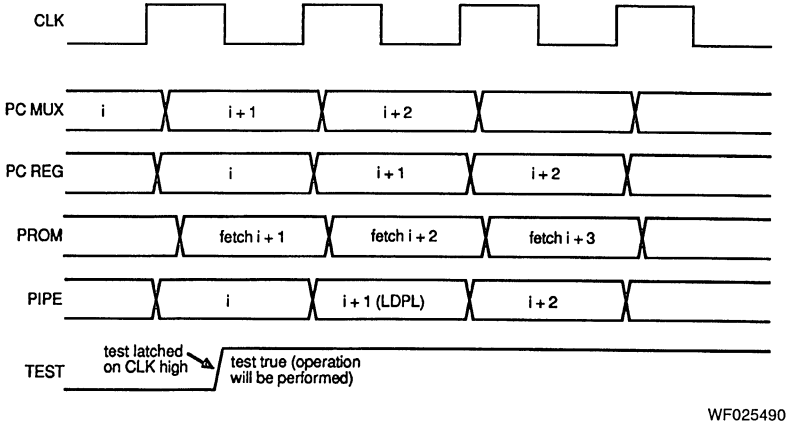


Figure 3. Conditional Instruction (Example: LDPL)

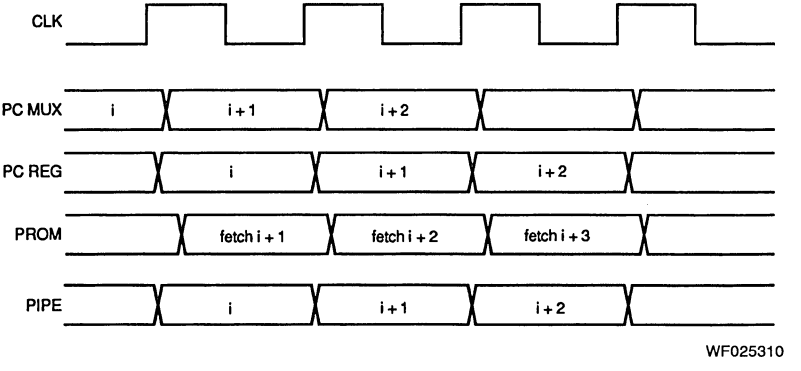


Figure 4. Unconditional Instruction (Example: CONT)

Using The Am29CPL142 To Address External Registered PROM

When the EXP cell is programmed, the program memory MUX is output over pins P[14:8]/A[6:0]. This feature can be used to extend the width of the output control word when external registered PROMs are used. In Figure 5 below, the Am29CPL142 addresses external registered PROMs to provide an output control word (9 + N) bits wide (where N is the bit width of the PROMs).

When the OUTPUT instruction is executed, the CREG contents are output over pins P[14:8]/A[6:0] on the following cycle. Consequently, if the CREG contents must be read after programming the EXP cell, the system design should be modified to handle this exception.

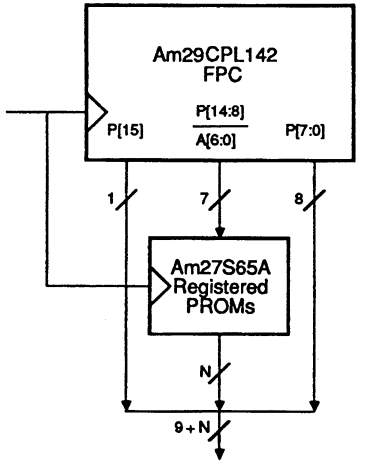


Figure 5. Using Am29CPL142 to Address External Registered PROMs

PROGRAMMING

The Am29CPL142 FPC controller is programmed using a simple algorithm. The internal EPROM is organized as a 128-word by 34-bit array. The array is divided up into five bytes for programming. Data is written byte-wide through pins P[15:8] using a simple sequence of voltages on two pins (CLK and $\overline{\text{RESET}}$). The Am29CPL142 uses pins P[7:5] for byte addressing; the EPROM array resides in the five lower bytes (0 through 4), while the most significant byte (7) is reserved for User Configuration Registers. Bytes 5 and 6 are not used on the Am29CPL142. Pins T[6:0] are used to address the word.

The Am29CPL142 programming cycle is shown in Figure 7. Each programming cycle consists of a program mode followed by a verify mode. To begin programming, the CLK pin is raised from a TTL level to V_{PP}. The AM29CPL142 enters program mode and disables output pins P[15:5] and accepts these pins as data I/O and address inputs. Now that the chip is in program mode, the $\overline{\text{RESET}}$ pin controls the program and verify modes. With a TTL-level HIGH on $\overline{\text{RESET}}$, the data I/O is in high impedance, and the program data (P[15:8]) and address (P[7:5], T[6:0]) can be set up. The data is written into memory by applying V_{PP} to $\overline{\text{RESET}}$ for time $T_{W(PGMi)}$ as described in the Program Parameter table. $\overline{\text{RESET}}$ is then switched back to a TTL-level HIGH, and the program data is removed. The verify (Read) cycle begins when $\overline{\text{RESET}}$ is switched to a TTL LOW level and the data resident at the addressed byte is output on the data I/O (P[15:8]). Raising $\overline{\text{RESET}}$ back to a TTL-level HIGH completes one programming cycle.

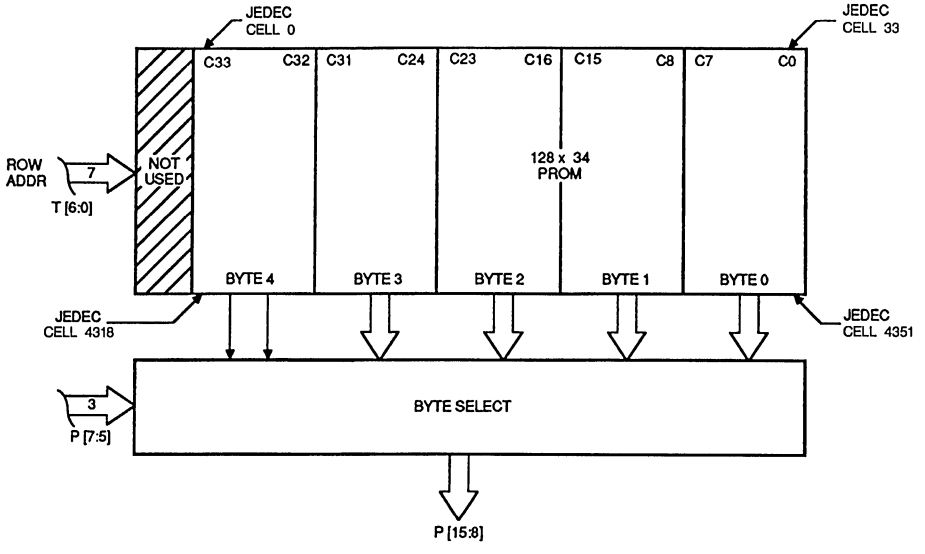
To ensure reliable programming if the data does not verify, the programming cycle could be repeated up to a total of 25 times. After a valid verification, a final overprogramming should be executed using a V_{CC} = 5.0 V and an extended pulse width $T_{W(PGMf)}$ equal to twice the sum of initial programming pulse width. At the conclusion of programming the Am29CPL142, the EPROM memory should be reverified for correct data at all addresses using two supply voltages (V_{CC} = 5.5 V and V_{CC} = 4.5 V).

Erasure

In order to fully erase all memory locations, it is necessary to expose the memory array to an ultraviolet light source having a wavelength of 2537 Angstroms. The minimum recommended dose (UV intensity x exposure time) is 15 Wsec/cm^2. For a UV lamp with a 12 mW/cm^2 power rating, the exposure time would be about 30 minutes. The device should be located one inch from the source in a direct line.

It should be noted that erasure will begin with exposure to light having wavelengths less than 4000 Angstroms. To prevent exposure to sunlight or fluorescent lighting, an opaque label should be affixed over the window after programming.

One-Time Programmable (OTP) Am29CPL142 devices are available in plastic and are ideal for volume production. They can be inventoried unprogrammed and used with current software revisions; there is no window to be covered to prevent light from changing data.



BD007553

JEDEC Cell Numbering: If Byte $\leq$ 3, Cell No. = (34 x (Row_Addr)) + (8 x (3-Byte)) + (7-Bit) + 2
If Byte = 4, Cell No. = (34 x (Row_Addr)) + (1-Bit)

Example Computations:

Row 0, Byte 4, Bit 1 — Cell No. = 34 (0) + (1 – 1) = 0
Row 127, Byte 0, Bit 0 — Cell No. = 34 (127) + 8 (3 – 0) + (7 – 0) + 2 = 4351

The cell numbers for the configuration bits are as follows: (Byte 7 only)
Register bypass for CC and T_6 through T_0 are cells 4352 through 4359
EXP is cell 4360
SSR is cell 4361
Reset register bypass is cell 4362

	Byte Select			Bit Select							
Byte	P[7]	P[6]	P[5]	P[15]	P[14]	P[13]	P[12]	P[11]	P[10]	P[9]	P[8]
0	L	L	L	C7	C6	C5	C4	C3	C2	C1	C0
1	L	L	H	C15	C14	C13	C12	C11	C10	C9	C8
2	L	H	L	C23	C22	C21	C20	C19	C18	C17	C16
3	L	H	H	C31	C30	C29	C28	C27	C26	C25	C24
4	H	L	L	–	–	–	–	–	–	C33	C32
5	H	L	H	–	–	–	–	–	–	–	–
6	H	H	L	–	–	–	–	–	–	–	–
7 (Row 0)	H	H	H	CC	T_6	T_5	T_4	T_3	T_2	T_1	T_0
7 (Row 1)	H	H	H	–	–	–	–	–	EXP	SSR	$\overline{\text{RESET}}$

Row decoding is performed as a straight binary decode of pins T[6:0]. Column decoding is accomplished by the combination of the byte select and the bit select inputs. As an example, when P[7:5] are all zero and select byte 0, and T[6:0] are all zero and select row 0, then P[15:8] can be used to program the 8 bits in byte 0 of row 0. These are identified as C_7 through C_0, and are JEDEC cells 26 through 33 respectively.

Figure 6. Programming Configuration

PROGRAMMING PARAMETERS (T_A = 25°C)

Parameter Symbol	Parameter Description		Min.	Typ.	Max	Unit
V_{CC}	Supply Voltage	I_{CC} = 50 mA	5.75	6.0	6.25	V
V_{PP}	Programming Voltage	I_{PP} = 30 mA	13	13.5	14	V
V_{IH}	Input HIGH Level		2.4		5.5	V
V_{IL}	Input LOW Level		0		0.5	V
t_W (PGMi)	Program Pulse Duration (Initial)		0.95		1.05	ms
t_W (PGMf)	Program Pulse Duration (Final)		2.0		50	ms
t_{AS}	Address Setup Time		2			µs
t_{DS}	Data Setup Time		2			µs
t_{SU} (V_{PP})	V_{PP} Setup Time		2			µs
t_{AH}	Address Hold Time		1			µs
t_{DH}	Data Hold Time		1			µs
t_{DV}	Data Valid from $\overline{RESET}$ LOW				100	ns
t_{DF}	Data Float from $\overline{RESET}$ HIGH		0		100	ns

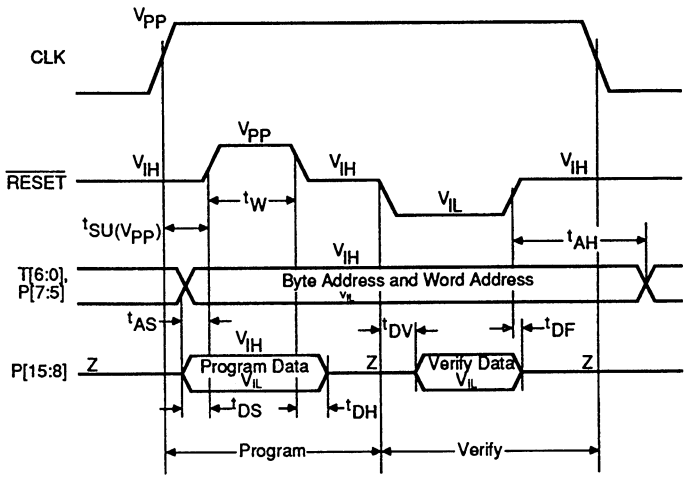


Figure 7. Programming Waveforms

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
(Ambient) Temperature Under Bias ..... −55 to +125°C
Supply Voltage to Ground Potential
(Pin 28 to Pin 14) Continuous ..... −0.5 V to +7.0 V
DC Voltage Applied to Outputs ... −0.3 V to +V_{CC} +0.3 V
DC Input Voltage ..... −0.3 V to +V_{CC} +0.3 V
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
Ambient Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.50 to +5.50 V

Military* (M) Devices
Ambient 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.

Thermal Impedance Values (θ_{JA}), Typical
28-Pin Plastic DIP (PD3028) ..... 50°C/W
28-Pin Ceramic DIP (CDV028) (CDE028) ..... 40°C/W
28-Pin Plastic Leaded Chip Carrier (PL 028) ..... 55°C/W
28-Pin Leadless Chip Carrier CLV028) ..... 55°C/W

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			Min.	Max.	Unit
V_{OH}	Output HIGH Voltage	V_{CC} = Min., V_{IN} = V_{IH} or V_{IL}	COM'L	I_{OH} = −3.0 mA	2.4		V
			MIL	I_{OH} = −1.0 mA			
V_{OL}	Output LOW Voltage	V_{CC} = Min., V_{IN} = V_{IL} or V_{IH}	COM'L	I_{OL} = 16 mA		0.50	V
			MIL	I_{OL} = 12 mA			
V_{IH} (Note 1)	Input HIGH Level	Guaranteed Input Logical HIGH Voltage for All Inputs			2.0		V
V_{IL} (Note 1)	Input LOW Level	Guaranteed Input Logical LOW Voltage for All Inputs				0.8	V
I_{IL}	Input LOW Current	V_{CC} = Max. V_{IN} = 0.5 V				−10	μA
I_{IH}	Input HIGH Current	V_{CC} = Max. V_{IN} = V_{CC} −0.5 V				10	μA
I_{OZH}	Output Leakage Current (Note 3)	V_{CC} = Max., V_{IL} = 0.8 V V_{IH} = 2.0 V		V_O = 2.4 V		10	μA
I_{OZL}				V_O = 0.5 V		−10	
I_{CC}	Power Supply Current	COM'L T_A = 0 to 70°C		CMOS, V_{IN} = V_{CC} or GND		115	mA
				TTL, V_{IN} = 0.5 V or 2.4 V		125	
		MIL T_A = −55 to 125°C		CMOS, V_{IN} = V_{CC} or GND		130	
				TTL, V_{IN} = 0.5 V or 2.4 V		140	
C_{PD}	Power Dissipation Capacitance (Note 4)	V_{CC} = Max. T_A = 25°C No Load			100	pF	Typical

Notes: 1. These are absolute values with respect to device ground and all overshoots due to system or tester noise are included.
2. I/O pin leakage is the worst-case of I_{OZX} or I_{IX} (where X = H or L).
3. Use CMOS I_{CC} when the device is driven by CMOS circuits and TTL I_{CC} when the device is driven by TTL circuits.
4. The dynamic current consumption is:
I_{CC} (Total) = I_{CC} (Static) + (C_{PD} + nC_L) V_{CC} (f/2), where f is the clock frequency, C_L is the load capacitance, and n is the number of loads.

CAPACITANCE*

Paramcter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
C_I ($\overline{RESET}$)	Input Capacitance	f = 1 MHz T_A = −55°C to +125°C V_{CC} = 4.5 V to 5.5 V		25	pF
C_I (All others)	Input Capacitance			15	
C_0	Output Capacitance			15	

* These capacitances are tested on a sample basis.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range

No.	Parameter Symbol	Parameter Description	Test Conditions	Am29CPL142 Min.	Am29CPL142 Max.	Unit
1		CLK to P[15:0]			20	ns
2		CLK to A[6:0]			30	ns
3	t_{PD}	DCLK to SDO			30	ns
4		Mode to SDO			30	ns
5		SDI to SDO			30	ns
6		T[6:0] to CLK, Registered Mode		6		ns
7		T[6:0] to CLK, Asynch. Mode (Note 1)		15		ns
8		CC to CLK, Registered Mode		6		ns
9		CC to CLK, Asynch. Mode (Note 1)		15		ns
10		$\overline{\text{RESET}}$ to CLK, Registered Mode		6		ns
11		$\overline{\text{RESET}}$ to CLK, Asynch. Mode		15		ns
12	t_S	Mode to CLK		30		ns
13		Mode to DCLK		30		ns
14		SDI to DCLK		30		ns
15		P[15:8] to DCLK		30		ns
16		T[6:0] to CLK	See Test Output Load Conditions	0		ns
17		CC to CLK		0		ns
18		$\overline{\text{RESET}}$ to CLK		0		ns
19	t_H	Mode to CLK		0		ns
20		Mode to DCLK		0		ns
21		SDI to DCLK		0		ns
22		P[15:8] to DCLK		0		ns
23	t_{PZX}	CLK to P[15:8] Enable			20	ns
24	t_{PXZ}	CLK to P[15:8] Disable			30	ns
25	t_{PW}	CLK Pulse Width		15		ns
26	t_{PWD}	DCLK Pulse Width		25		ns
27	t_P	CLK Period (Note 1)		40		ns
28		DCLK Period		50		ns

Notes: 1. These parameters are measured indirectly on unprogrammed devices. They are determined as follows:
a. Measure delay from input (T[6:0], or CLK) to EPROM address out in test mode. This will measure the delay through the sequence logic.
b. Measure setup time from T[6:0] input through EPROM test columns to pipeline register in verify test column mode. This will measure the delay through the EPROM and register setup.
c. Measure delay from T[6:0] input to EPROM address out in verify test column mode. This will measure the delay through the logic and P[15:0] outputs.
To calculate the desired parameter measurement the following formula is used:
Measurement (a) + Measurement (b) – Measurement (c)
CLK PERIOD:
CLK (a) + (b) – (c) = CLK PERIOD

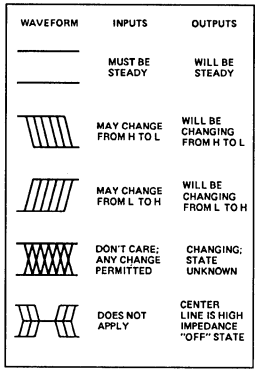
SWITCHING CHARACTERISTICS over MILITARY operating range

No.	Parameter Symbol	Parameter Description	Test Conditions	Am29CPL152		Am29CPL152-1		Unit
				Min.	Max.	Min.	Max.	
1		CLK to P[15:0]			25		20	ns
2		CLK to A[6:0]			35		30	ns
3	t_{PD}	DCLK to SDO			35		30	ns
4		Mode to SDO			35		30	ns
5		SDI to SDO			35		30	ns
6		T[6:0] to CLK, Registered Mode		8		8		ns
7		T[6:0] to CLK, Asynch. Mode (Note 1)		20		20		ns
8		CC to CLK, Registered Mode		8		8		ns
9		CC to CLK, Asynch. Mode (Note 1)		20		20		ns
10		$\overline{\text{RESET}}$ to CLK, Registered Mode		8		8		ns
11		$\overline{\text{RESET}}$ to CLK, Asynch. Mode		20		20		ns
12	t_S	Mode to CLK		35		30		ns
13		Mode to DCLK		35		30		ns
14		SDI to DCLK		35		30		ns
15		P[15:8] to DCLK		35		30		ns
16		T[6:0] to CLK	See Test	0		0		ns
17		CC to CLK	Output Load	0		0		ns
18		$\overline{\text{RESET}}$ to CLK	Conditions	0		0		ns
19	t_H	Mode to CLK		0		0		ns
20		Mode to DCLK		0		0		ns
21		SDI to DCLK		0		0		ns
22		P[15:8] to DCLK		0		0		ns
23	t_{PZX}	CLK to P[15:8] Enable			25		20	ns
24	t_{PXZ}	CLK to P[15:8] Disable			35		30	ns
25	t_{PW}	CLK Pulse Width		20		15		ns
26	t_{PWD}	DCLK Pulse Width		30		25		ns
27	t_P	CLK Period (Note 1)		50		40		ns
28		DCLK Period		60		50		ns

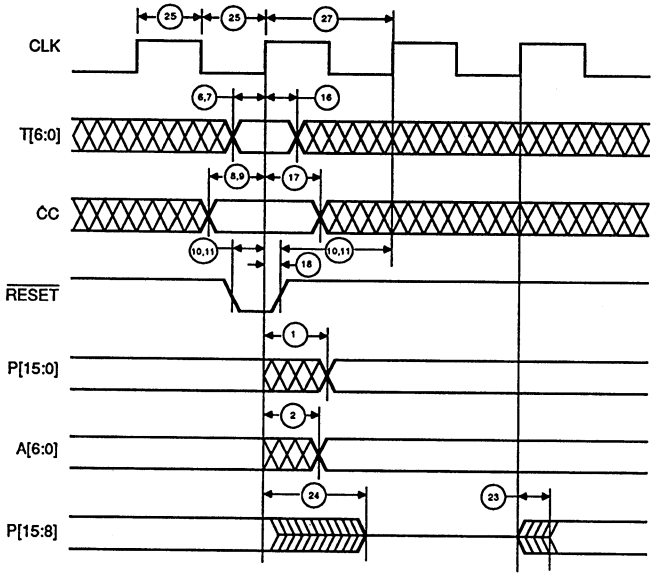
Notes: 1. These parameters are measured indirectly on unprogrammed devices. They are determined as follows:
a. Measure delay from input (T[6:0], or CLK) to PROM address out in test mode. This will measure the delay through the sequence logic.
b. Measure setup time from T[6:0] input through PROM test columns to pipeline register in verify test column mode. This will measure the delay through the PROM and register setup.
c. Measure delay from T[6:0] input to PROM address out in verify test column mode. This will measure the delay through the logic and P[15:0] outputs.
To calculate the desired parameter measurement the following formula is used:
Measurement (a) + Measurement (b) – Measurement (c)
CLK PERIOD:
CLK (a) + (b) – (c) = CLK PERIOD

SWITCHING WAVEFORMS

KEY TO SWITCHING WAVEFORMS



KS000010

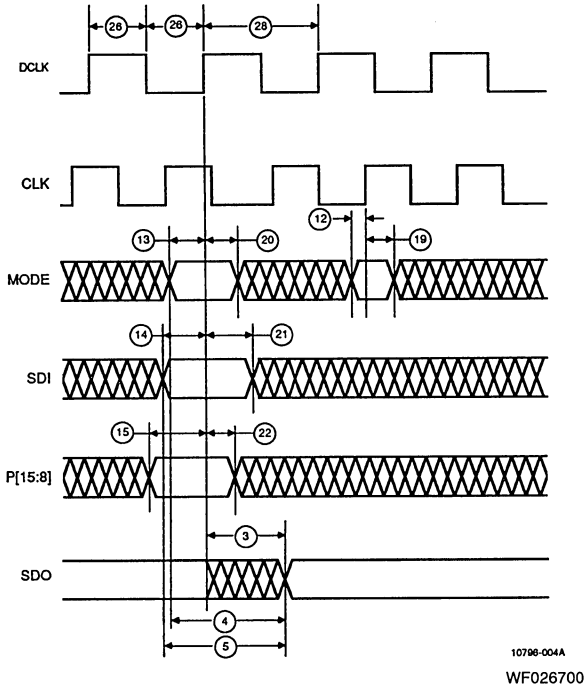


10796-003A

WF026690

Normal Configuration

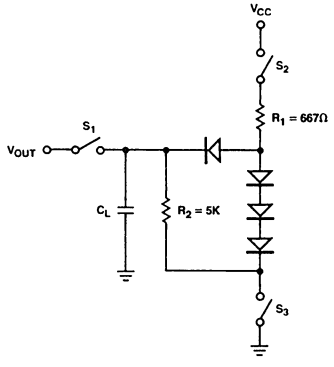
SWITCHING WAVEFORMS (Cont'd.)



WF026700

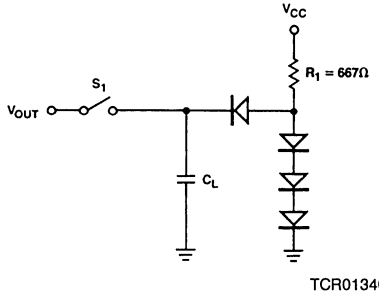
SSR Configuration

SWITCHING TEST CIRCUITS



TCR01330

A. Three State Outputs

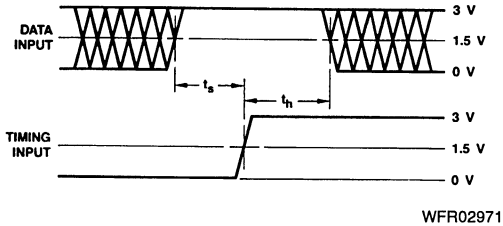


TCR01340

B. Normal Outputs

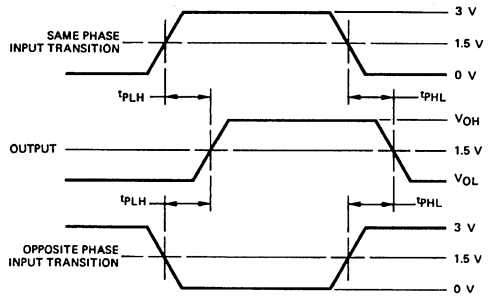
Notes: 1. CL = 50 pF includes scope probe, wiring and stray capacitances without device in test fixture.
2. S_1, S_2, and S_3 are closed during function tests and all AC tests except output enable tests.
3. S_1 and S_3 are closed while S_2 is open for tp_{ZH} test.
4. C_L = 5.0 pF for output disable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

SWITCHING TEST WAVEFORMS



WFR02971

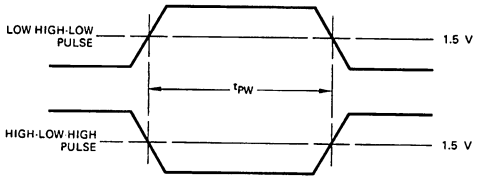
Setup, Hold, and Release Times



WFR02980

Propagation Delay

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



WFR02791

Pulse Width

Test	V_X	Output Waveform – Measurement Level
All t_{PD}s	5.0V	V_{OH}, V_{OL}, 1.5V
t_{PHZ}	0.0V	V_{OH}, 0.5V, 0.0V
t_{PLZ}	5.0V	3V, 0.5V, V_{OL}
t_{PZH}	0.0V	V_{OH}, 1.5V, 0.0V
t_{PZL}	5.0V	3V, 1.5V, V_{OL}

WFR02680

Enable and Disable Times

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.

Note: Pulse generator for all pulses: Rate $\leq$ 1.0 MHz; $Z_0 = 50\ \Omega$: $t_r \leq 2.5$ ns.

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 200 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 that may not actually reach V_{IL} or V_{IH} until the noise has settled. AMD recommends using $V_{IL} \leqslant 0$ V and $V_{IH} \geqslant 3.0$ 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 that varies from one type of tester to another but is generally around 50 pF. This, of course, makes it impossible to make direct measurements of parameters that call for smaller capacitive load than the associated stray capacitance. Typical examples of this are the so-called "float delays," which measure the propagation delays into 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 <u>both</u> 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 (I_{OH}, I_{OL}, for example) have already been taken and are within spec. In some cases, special DC tests are performed in order to facilitate this correlation.

7. Threshold Testing

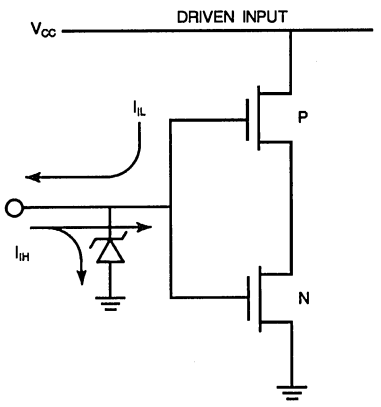
   The noise associated with automatic testing (due to the long inductive cables), and the high gain of the tested device when in the vicinity of the actual device threshold, frequently give rise to oscillations when testing high-speed speed circuits. These oscillations are not indicative of a reject device, but instead, of an overtaxed test system. To minimize this problem, thresholds are tested at least once for <u>each</u> 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

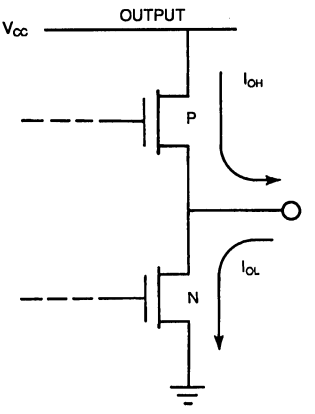
   Occasionally, parameters are specified that cannot be measured directly on automatic testers because of tester limitations. Data input hold times often fall into this category. In these cases, the parameter in question is guaranteed by correlating these tests with other AC tests that have been performed. These correlations are arrived at by the cognizant engineer by using data from precise bench measurements in conjunction with the knowledge that certain DC parameters have already been measured and are within spec.

   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.

INPUT/OUTPUT CIRCUIT DIAGRAMS



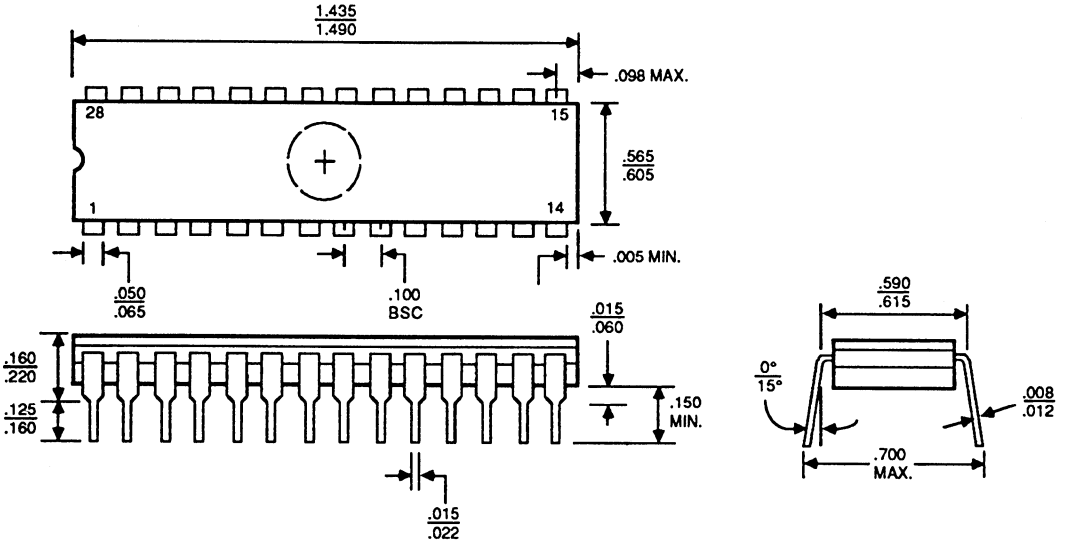
IC000861



IC000871

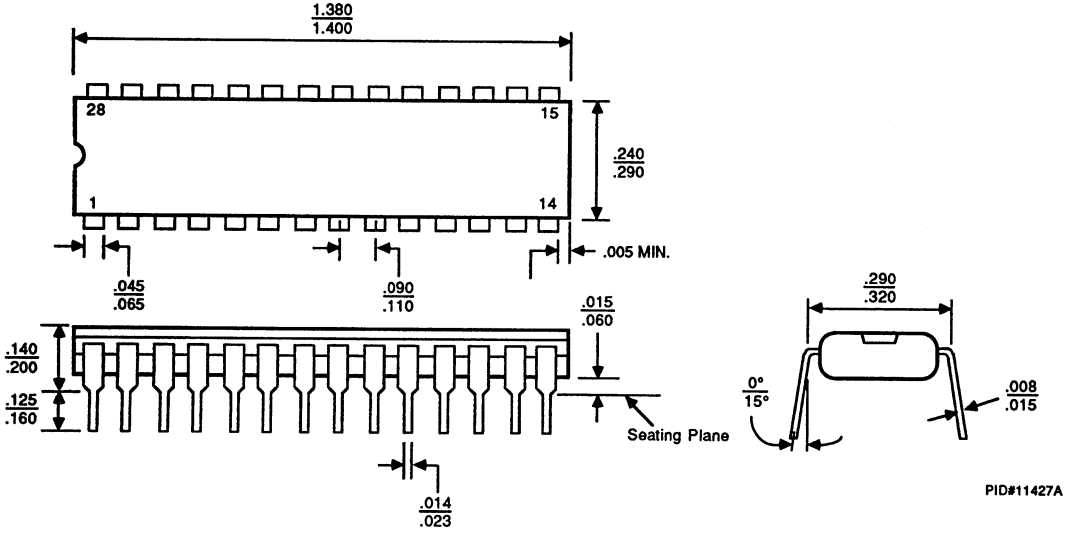
PHYSICAL DIMENSIONS*

CDV028



PID# 08267B

PD3028



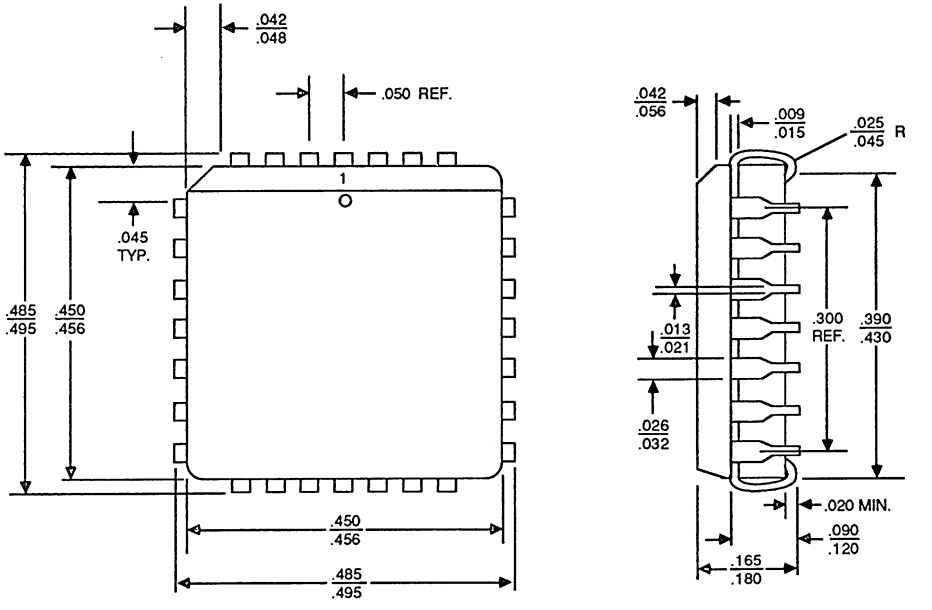
PID#11427A

PID# 11427A

*For reference only. All dimensions are measured in inches. BSC is an ANSI Standard for Basic Space Centering.

PHYSICAL DIMENSIONS* (Cont'd.)

PL 028



PID# 06751E

CDE028*

*Package is in development. Consult MIP Product Marketing for information.

Sales Offices

North American

ALABAMA	(205) 882-9122
ARIZONA	(602) 242-4400
CALIFORNIA,	
Culver City	(213) 645-1524
Newport Beach	(714) 752-6262
San Diego	(619) 560-7030
San Jose	(408) 452-0500
Woodland Hills	(818) 992-4155
CANADA, Ontario,	
Kanata	(613) 592-0060
Willowdale	(416) 224-5193
COLORADO	(303) 741-2900
CONNECTICUT	(203) 264-7800
FLORIDA,	
Clearwater	(813) 530-9971
Ft. Lauderdale	(305) 776-2001
Orlando	(407) 830-8100
GEORGIA	(404) 449-7920
ILLINOIS,	
Chicago	(312) 773-4422
Naperville	(312) 505-9517
INDIANA	(317) 244-7207
KANSAS	(913) 451-3115
MARYLAND	(301) 796-9310
MASSACHUSETTS	(617) 273-3970
MINNESOTA	(612) 938-0001
MISSOURI	(913) 451-3115
NEW JERSEY,	
Cherry Hill	(609) 662-2900
Parsippany	(201) 299-0002
NEW YORK,	
Liverpool	(315) 457-5400
Poughkeepsie	(914) 471-8180
Woodbury	(516) 364-8020
NORTH CAROLINA	(919) 878-8111
OHIO,	
Columbus	(614) 891-6455
Dayton	(513) 439-0470
OREGON	(503) 245-0080
PENNSYLVANIA	(215) 398-8006
SOUTH CAROLINA	(803) 772-6760
TEXAS,	
Austin	(512) 346-7830
Dallas	(214) 934-9099
Houston	(713) 785-9001
WASHINGTON	(206) 455-3600
WISCONSIN	(414) 792-0590

International

BELGIUM, Bruxelles	TEL	(02) 771-91-42
	FAX	(02) 762-37-12
	TLX	61028
FRANCE, Paris	TEL	(1) 49-75-10-10
	FAX	(1) 49-75-10-13
	TLX	263282
WEST GERMANY,		
Hannover area	TEL	(0511) 736085
	FAX	(0511) 721254
	TLX	922850
München	TEL	(089) 4114170
	FAX	(089) 406490
	TLX	523883
Stuttgart	TEL	(0711) 62 33 77
	FAX	(0711) 625187
	TLX	721882
HONG KONG	TEL	852-5-8654525
	FAX	852-5-8654335
	TLX	67955AMDAPHX
ITALY, Milan	TEL	(02) 3390541
		(02) 3533241
	FAX	(02) 3498000
	TLX	315286
JAPAN,		
Kanagawa	TEL	462-47-2911
	FAX	462-47-1729
Tokyo	TEL	(03) 345-8241
	FAX	(03) 342-5196
	TLX	J24064AMDTKOJ
Osaka	TEL	06-243-3250
	FAX	06-243-3253
KOREA, Seoul	TEL	82-2-784-7598
	FAX	82-2-784-8014
LATIN AMERICA,		
Ft. Lauderdale	TEL	(305) 484-8600
	FAX	(305) 485-9736
	TEL	5109554261 AMDFTL
NORWAY, Hovik	TEL	(02) 537810
	FAX	(02) 591959
	TLX	79079
SINGAPORE	TEL	65-2257544
	FAX	65-2246113
	TLX	RS55650 MMI RS
SWEDEN,		
Stockholm	TEL	(08) 733 03 50
	FAX	(08) 733 22 85
	TLX	11602
TAIWAN	TLX	886-2-7122066
	FAX	886-2-7122017
UNITED KINGDOM,		
Manchester area	TEL	(0925) 828008
	FAX	(0925) 827693
	TLX	628524
London area	TEL	(04862) 22121
	FAX	(0483) 756196
	TLX	859103

North American Representatives

CANADA	
Burnaby, B.C.	
DAVETEK MARKETING	(604) 430-3680
Calgary, Alberta	
VITEL ELECTRONICS	(403) 278-5833
Kanata, Ontario	
VITEL ELECTRONICS	(613) 592-0090
Mississauga, Ontario	
VITEL ELECTRONICS	(416) 676-9720
Quebec	
VITEL ELECTRONICS	(514) 636-5951
IDAHO	
INTERMOUNTAIN TECH MKTG	(208) 888-6071
ILLINOIS	
HEARTLAND TECHNICAL MARKETING	(312) 577-9222
INDIANA	
ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3452
IOWA	
LORENZ SALES	(319) 377-4666
KANSAS	
Merriam – LORENZ SALES	(913) 384-6556
Wichita – LORENZ SALES	(316) 721-0500
KENTUCKY	
ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3452
MICHIGAN	
Birmingham – MIKE RAICK ASSOCIATES	(313) 644-5040
Holland – COM-TEK SALES, INC	(616) 399-7273
Novi – COM-TEK SALES, INC	(313) 344-1409
MISSOURI	
LORENZ SALES	(314) 997-4558
NEBRASKA	
LORENZ SALES	(402) 475-4660
NEW MEXICO	
THORSON DESERT STATES	(505) 293-8555
NEW YORK	
NYCOM, INC	(315) 437-8343
OHIO	
Centerville – DOLFUSS ROOT & CO	(513) 433-6776
Columbus – DOLFUSS ROOT & CO	(614) 885-4844
Strongsville – DOLFUSS ROOT & CO	(216) 238-0300
PENNSYLVANIA	
DOLFUSS ROOT & CO	(412) 221-4420
UTAH	
R^2 MARKETING	(801) 595-0631
WISCONSIN	
HEARTLAND TECHNICAL MARKETING	(414) 796-1128

Advanced Micro Devices, Inc. 901 Thompson Place, P.O. Box 3453, Sunnyvale, CA 94088, USA
Tel: (408) 732-2400 • TWX: 910-339-9280 • TELEX: 34-6306 • TOLL FREE: (800) 538-8450
APPLICATIONS HOTLINE TOLL FREE: (800) 222-9323 • (408) 749-5703

11/17/88
AIS-WCP-15M-12/88-0 Printed in USA