Am2930

Program Control Unit

DISTINCTIVE CHARACTERISTICS

- **Powerful, 4-bit slice address controller for memories**
  Useful with both main memory and microprogram memory
  Expandable to generate any address length
- **Executes 32 instructions**
  Capable of executing branch and subroutine call and return
- **Twelve different relative address instructions**
  Including JUMP-TO-SUBROUTINE relative and RETURN-FROM-SUBROUTINE relative
- **Built-in condition code input**
  Sixteen instructions are dependent on external condition control
- **Seventeen-level push/pop stack**
  On-chip storage of subroutine return addresses nested up to 17 levels deep
- **Separate incrementer for program counter**
  A relative address may be computed and PC may be incremented by one on a single cycle

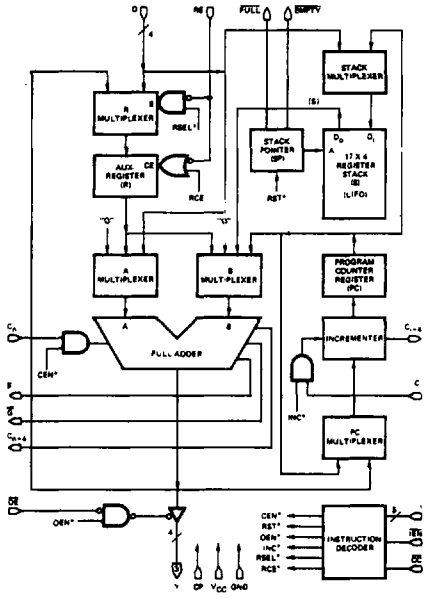
GENERAL DESCRIPTION

The Am2930 is a four-bit wide Program Control Unit intended to perform machine level addressing functions, although the device can also be used as a microprogram sequencer. Four Am2930's may be interconnected to generate a 16-bit address (64K words). The Am2930 contains a program counter, a subroutine stack, an auxiliary register, and a full adder for computing relative addresses.

The Am2930 performs five types of instructions. These are: 1) Unconditional Fetch; 2) Conditional Jump; 3) Conditional Jump-to-Subroutine; 4) Conditional Return-from-Subroutine; and 5) Miscellaneous Instructions.

There are four sources of data for the adder which generates the Address outputs (Y_0–Y_3). These are: 1) the Program Counter (PC); 2) the Stack (S); 3) the auxiliary Register (R); and 4) the Direct inputs (D). Under control of the Instruction inputs (I_0–I_4), the multiplexers at the adder inputs allow various combinations of these terms to be generated at the three-state Y address outputs. The instruction lines also control the updating of the program counter and the auxiliary register. A condition code input is provided for conditional instructions.

BLOCK DIAGRAM



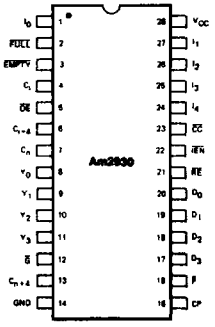
BD002460

03642B

Refer to Page 13-1 for Essential Information on Military Devices

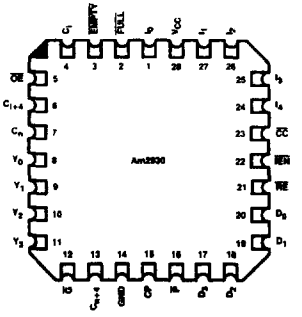
CONNECTION DIAGRAM
Top View

D-28
F-28*



CD004350

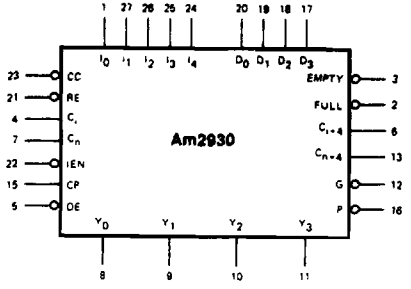
Chip-Pak™
L-28-1



CD004730

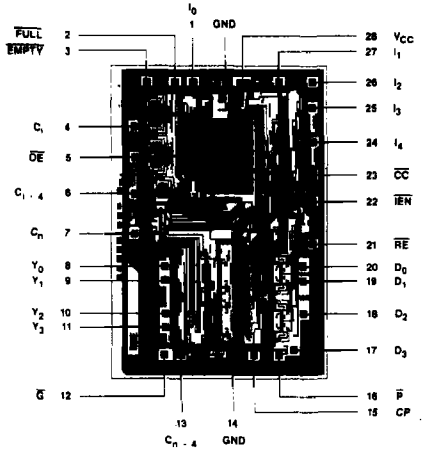
*F-28 pin configuration identical to D-28.
Note: Pin 1 is marked for orientation

LOGIC SYMBOL



LS001050

METALLIZATION AND PAD LAYOUT



DIE SIZE: 0.133" x 0.200"
Pad numbers correspond to DIP pinout

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: *Device number, speed option (if applicable), package type, operating range and screening option (if desired).*

Am2930 **D** **C** **B**

- **B** — *Screening Option*
  - Blank – Standard processing
  - B – Burn-in
- **C** — Temperature (See Operating Range)
  - C – Commercial (0°C to +70°C)
  - M – Military (–55°C to +125°C)
- **D** — Package
  - D – 28-pin CERDIP
  - F – 28-pin flatpak
  - L – 28-pin leadless chip carrier
  - X – Dice
- **Am2930** — Device type
  Program Control Unit

Valid Combinations	
Am2930	DC, DCB, DMB FMB LC, LMB XC, XM

Valid Combinations

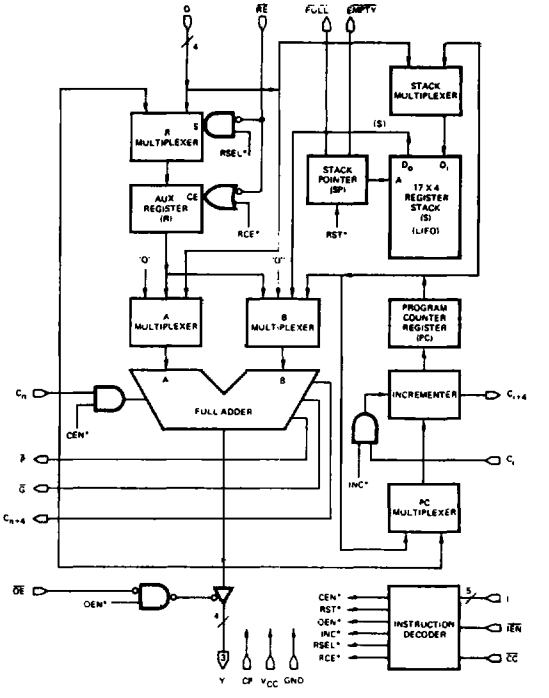
Consult the AMD sales office in your area to determine if a device is currently available in the combination you wish.

Chip-Pak is a trademark of Advanced Micro Devices, Inc.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
	I_{0-4}		The five Instruction control lines to the Am2930, used to establish data paths and enable internal registers.
22	$\overline{IEN}$	I	The Instruction Enable Input, used to enable and disable internal registers. When $\overline{IEN}$ is LOW, all internal registers are under control of the Instruction inputs. When $\overline{IEN}$ is HIGH, all internal registers except R are inhibited from changing state. R is controlled by the $\overline{RE}$ input. The $\overline{IEN}$ input does not affect the combinatorial data paths and the outputs established by the Instruction inputs.
23	$\overline{CC}$	I	The Condition Code input determines whether or not a conditional instruction (Instructions 16-31) is performed. If $\overline{CC}$ is LOW, the conditional instruction is executed. If $\overline{CC}$ is HIGH, Fetch PC (Instruction 1) is executed. The $\overline{CC}$ input may be either HIGH or LOW for unconditional instructions (Instructions 0-15).
21	$\overline{RE}$	I	The Register Enable input for the Auxiliary Register (R). A LOW on $\overline{RE}$ causes the Auxiliary Register (R) to be loaded from the D inputs unless Instruction 8 or 9 is being executed and $\overline{IEN}$ is LOW.
7	C_n	I	The carry-in to the Full Adder.
13	C_{n+4}	O	The carry-out of the Full Adder.
16,12	$\overline{P}$, $\overline{G}$	O	The carry generate and propagate outputs of the Full Adder.
4	C_i	I	The carry-in to the program counter incrementer.
6	C_{i+4}	O	The carry-out of the program counter incrementer.
	Y_{0-3}	O	The four address outputs of the Am2930. These are three-state output lines. When enabled, they display the outputs of the Full Adder.
5	$\overline{OE}$	I	Output Enable. When $\overline{OE}$ is HIGH, the Y outputs are OFF (high-impedance); when $\overline{OE}$ is LOW, the Y outputs are active (HIGH or LOW).
	D_{0-3}	I	The four Direct inputs which are used as inputs to the Auxiliary Register, the RAM, and the Full Adder, under instruction control.
3	$\overline{Empty}$	O	The Empty output is LOW when the Stack is empty.
2	$\overline{Full}$	O	The Full output is LOW when the LIFO stack is full – during and after the 17th push operation.
15	CP	I	The clock input to the Am2930. All internal registers (R, SP, PC) and the RAM are updated on the LOW-to-HIGH transition of the clock input.

BLOCK DIAGRAM



BD002230

*INTERNAL

ARCHITECTURE OF THE Am2930

The Am2930 is a bipolar Program Control Unit intended for use in high-speed microprocessor applications. The device is a cascadable, four-bit slice such that three devices allow addressing of up to 4K words of memory and four devices allow addressing of up to 64K words of memory.

As shown in the Block Diagram, the device consists of the following:

1. A full adder with input multiplexers
2. A Program Counter Register with an incrementer and an input multiplexer
3. A 17 x 4 Last-In, First-Out (LIFO) stack consisting of an input multiplexer, a 17 x 4 RAM, and a Stack Pointer
4. An auxiliary register with an input multiplexer
5. An instruction decoder
6. Four 3-state output buffers on the address outputs

The following paragraphs describe each of these blocks in detail.

Full Adder

The Full Adder is a binary device with full lookahead carry logic for high-speed addition and provision is made for further lookahead by including both carry propagate ($\overline{P}$) and carry generate ($\overline{G}$) outputs. In slower systems, the carry output (C_{n+4}) can be connected to the next higher C_n to provide ripple block arithmetic. The carry input to the adder (C_n) is internally inhibited during those instructions which do not require an addition to be performed. For these instructions, the data is passed directly through the adder, independent of the state of C_n.

The multiplexers at the A and B inputs of the adder are controlled by the Instruction decoder which selects the appropriate adder inputs for the selected instruction.

Program Counter

The program counter consists of a register preceded by an incrementer. The Program Counter Register (PC) is a four-bit, edge-triggered, D-type register which is loaded from the incrementer output on the LOW-to-HIGH transition of the clock input (CP) at the end of every instruction.

The incrementer utilizes full lookahead logic for high speed. For cascading devices, the carry output of the incrementer (C_{i+4}) is connected to the incrementer carry input (C_i) of the next higher device. The output of the incrementer, which is loaded into the PC, is equal to the incrementer input plus C_i. Therefore, it is possible to control the entire cascaded incrementer from the C_i input of the least significant device; a LOW on the C_i input of the least significant device will simply pass the data from the multiplexer output to the inputs of PC; a HIGH will cause the outputs of the multiplexer to be incremented before they are loaded into PC. During three instructions (unconditional Hold and conditional Hold and Suspend when the $\overline{CC}$ input is LOW), the C_i input is internally inhibited; therefore, data is passed from the multiplexer output to the PC without incrementing. The multiplexer selects the input to the incrementer from either PC or the output of the Full Adder, depending upon the instruction being executed. During the Jump, Jump-to-Subroutine, and Return instructions, the multiplexer chooses the Full Adder outputs as the input to the incrementer if the $\overline{CC}$ input is LOW. The Full Adder output is also selected for the Reset instruction. For all other instructions, the PC is selected as the input to the incrementer.

17 x 4 LIFO Stack

The 17 x 4 LIFO stack consists of a multiplexer, a 17 x 4 RAM, and a Stack Pointer (SP) which address the words in the RAM.

The SP always points to the last word written into the RAM (Top of the Stack). The Top of the Stack (S) is available at the output of the RAM.

Data is pushed onto the Top of the Stack from either D or PC. It is written into memory location SP+1. The SP is incremented on the LOW-to-HIGH clock transition at the end of the cycle so that it still points to the last data written into the RAM.

For a Pop operation, the contents of the RAM are not changed, but the SP is decremented at the end of the cycle so that it then points to the new Top of the Stack.

The SP is an up/down counter which changes state on the LOW-to-HIGH transition of the Clock input. It is internally prevented from incrementing when the stack is full and from decrementing when the Stack is empty. When the Stack is full, the RAM write circuitry is also inhibited.

The active LOW Empty output ($\overline{EMPTY}$) is LOW when the stack is empty (after the Reset instruction and after the last word has been Popped from the stack); the active LOW Full output ($\overline{FULL}$) is LOW either when the stack is full or when the current instruction being executed will fill the stack (during and after the 17th Push).

Auxiliary Register (R)

The Auxiliary Register (R) can be loaded from either the Direct inputs (D) or the output of the Full Adder. It is loaded on the LOW-to-HIGH transition of the clock input (CP) if the Register Enable input ($\overline{RE}$) is LOW or if the Instruction inputs call for it to be loaded. When $\overline{RE}$ is LOW, R is loaded from the D inputs unless the Instruction dictates that R be loaded from the output of the Full Adder.

Instruction Decoder

The Instruction Decoder generates the signals necessary to establish the data paths and to enable the loading of the PC, R, SP, and RAM.

For unconditional instructions, the $\overline{CC}$ input is not utilized; it may be either HIGH or LOW. For conditional instructions, if $\overline{CC}$ is LOW, the condition is met and the conditional operation is performed; if $\overline{CC}$ is HIGH, a Fetch PC is performed.

Output Buffers

The Address outputs (Y_0–Y_3) are three-state drivers which may be disabled either under Instruction control or by a HIGH on the Output Enable input ($\overline{OE}$). Disabling the Y outputs does not affect the execution of instructions inside the Am2930.

Instruction Enable

When HIGH, the Instruction Enable input ($\overline{IEN}$) forces PC and SP into the hold mode and disables the write circuitry to the RAM. The auxiliary register (R) is under control of the $\overline{RE}$ input when $\overline{IEN}$ is HIGH, independent of the state of the Instruction inputs. The $\overline{IEN}$ input does not affect the combinatorial data paths or Y outputs in the Am2930. The data paths are selected by the Instruction and $\overline{CC}$ inputs and are not affected by $\overline{IEN}$.

Am2930 INSTRUCTION SET

The Am2930 Instruction set can be divided into five types of instructions. These are:

- Unconditional Fetches
- Conditional Jumps
- Conditional Jumps-to-Subroutine
- Conditional Returns-from-Subroutine
- Miscellaneous Instructions

The following paragraphs describe each of these types in detail.

Unconditional Fetches

As can be seen from Table 1, there are nine unconditional Fetch instructions (Instructions 1–9). Under control of the Instruction inputs, the desired value is placed at the Y outputs. For all Fetch instructions, PC is incremented if C_i of the least significant device is HIGH. For Instructions 1 through 7, the auxiliary register is under control of the RE input. For Instructions 8 and 9, R is loaded with PC and R + D, respectively. The RAM and Stack Pointer are not changed during a Fetch instruction.

Conditional Jumps

There are six conditional Jump instructions (Instructions 16 through 21). Under control of the Instruction inputs, the desired value is placed at the Y outputs. Additionally, the value is incremented if C_i of the least significant device is HIGH and loaded into PC. During these instructions, R is controlled by $\overline{RE}$. The RAM and Stack Pointer are not changed during these instructions. The above operations are performed if the $\overline{CC}$ input is LOW; if $\overline{CC}$ is HIGH, a Fetch PC operation is performed.

Conditional Jumps-to-Subroutine

There are six conditional Jump-to-Subroutine instructions (Instructions 22 through 27). Under control of the Instruction inputs, the desired value is placed on the Y outputs. On the rising edge of the clock the value is incremented* and loaded into PC; PC is loaded into the RAM at location SP + 1, and SP is incremented.

As with Conditional Jump Instructions, R is controlled by $\overline{RE}$ and whether the Jump-to-Subroutine or Fetch PC is performed depends upon the state of the $\overline{CC}$ input.

Conditional Returns-from-Subroutine

There are two conditional Return-from-Subroutine instructions (Instructions 28 and 29). Under control of the instruction inputs, either S or S + D is placed at the Y outputs. Additionally, the selected value is incremented* and loaded into PC and SP is decremented at the end of the cycle (on the rising edge of the clock).

*If C_i of the least significant device is HIGH.

As with the Condition Jump and Jump-to-Subroutine Instructions, R is controlled by $\overline{RE}$ and whether the Return-from-Subroutine or Fetch PC is performed depends upon the state of the $\overline{CC}$ input.

Miscellaneous Instructions

Each of the nine miscellaneous instructions is described individually.

Reset (Instruction 0)

The Reset instruction forces the Y outputs to zero, loads either zero or one into PC, depending upon the C_i input of the least significant device, and resets SP. The RAM is unchanged and R is controlled by $\overline{RE}$.

Load R (Instruction 10)

This instruction loads the data on the D inputs into R. PC is either incremented or held depending upon C_i of the least significant device. The SP and RAM are not changed.

Push PC (Instruction 11)

This instruction is the same as Fetch PC except that PC is loaded into RAM and SP is incremented at the end of the cycle; i.e., the current PC is Pushed onto the stack.

Push D (Instruction 12)

This instruction is the same as Fetch PC except that D is loaded into the RAM and SP is incremented at the end of the cycle; i.e., external data is Pushed onto the stack.

Pop S (Instruction 13)

This instruction places the Top of the Stack (S) at the Y outputs and decrements SP at the end of the cycle. The PC is incremented if the C_i input of the least significant device is HIGH. R is controlled by $\overline{RE}$.

Pop PC (Instruction 14)

This instruction is the same as Fetch PC except SP is decremented at the end of the cycle, causing the data at the top of the stack to be lost.

Hold (Instruction 15)

This instruction places PC at the Y outputs and inhibits any change in PC, SP, and RAM. R is controlled by $\overline{RE}$.

Conditional Hold (Instruction 30)

This instruction is the same as Hold except $\overline{CC}$ must be LOW. If $\overline{CC}$ is HIGH, the Fetch PC instruction is performed.

Suspend (Instruction 31)

The Suspend instruction is the same as the Conditional Hold instruction except the Y outputs are forced into the high-impedance state if $\overline{CC}$ is LOW.

TABLE I – Am2930 INSTRUCTION SET

Mnemonic	Instruction Number	I_4	I_3	I_2	I_1	I_0	$\overline{CC}$	$\overline{IEN}$	Instruction	Y_0-Y_3	Next State (after CP ⎍) (Note 3) PC	R $\overline{RE}$ = L	R $\overline{RE}$ = H	RAM	SP
		X	X	X	X	X	X	H	Instruction Disable	Note 1	–	D	–	–	–
PRST	0	L	L	L	L	L	X	L	RESET	"0"	"0" + C_i	D	–	–	Reset
FPC	1	L	L	L	L	H	X	L	FETCH PC	PC	PC + C_i	D	–	–	–
FR	2	L	L	L	H	L	X	L	FETCH R	R	PC + C_i	D	–	–	–
FD	3	L	L	L	H	H	X	L	FETCH D	D	PC + C_i	D	–	–	–
FRD	4	L	L	H	L	L	X	L	FETCH R + D	P + D + C_n	PC + C_i	D	–	–	–
FPD	5	L	L	H	L	H	X	L	FETCH PC + D	PC + D + C_n	PC + C_i	D	–	–	–
FPR	6	L	L	H	H	L	X	L	FETCH PC + R	PC + R + C_n	PC + C_i	D	–	–	–
FSD	7	L	L	H	H	H	X	L	FETCH S + D	S + D + C_n	PC + C_i	D	–	–	–
FPLR	8	L	H	L	L	L	X	L	FETCH PC → R	PC	PC + C_i	PC	PC	–	–
FRDR	9	L	H	L	L	H	X	L	FETCH R + D → R	R + D + C_n	PC + C_i	R + D + C_n	R + D + C_n	–	–
PLDR	10	L	H	L	H	L	X	L	LOAD R	PC	PC + C_i	D	D	–	–
PSHP	11	L	H	L	H	H	X	L	PUSH PC	PC	PC + C_i	D	–	PC→Loc SP + 1	SP + 1
PSHD	12	L	H	H	L	L	X	L	PUSH D	PC	PC + C_i	D	–	D→Loc SP + 1	SP + 1
POPS	13	L	H	H	L	H	X	L	POP S	S	PC+ C_i	D	–	–	SP–1
POPP	14	L	H	H	H	L	X	L	POP PC	PC	PC + C_i	D	–	–	SP–1
PHLD	15	L	H	H	H	H	X	L	HOLD	PC	–	D	–	–	–
	16-31	H	X	X	X	X	H	L	FAIL COND'L TEST (FETCH PC)	PC	PC + C_i	D	–	–	–
JMPR	16	H	L	L	L	L	L	L	JUMP R	R	R + C_i	D	–	–	–
JMPD	17	H	L	L	L	H	L	L	JUMP D	D	D + C_i	D	–	–	–
JMPZ	18	H	L	L	H	L	L	L	JUMP "O"	"O"	"O" + C_i	D	–	–	–
JPRD	19	H	L	L	H	H	L	L	JUMP R + D	R + D + C_n	R + D + C_n + C_i	D	–	–	–
JPPD	20	H	L	H	L	L	L	L	JUMP PC + D	PC + D + C_n	PC + D + C_n + C_i	D	–	–	–
JPPR	21	H	L	H	L	H	L	L	JUMP PC + R	PC + R + C_n	PC + R + C_n + C_i	D	–	–	–
JSBR	22	H	L	H	H	L	L	L	JSB R	R	R + C_i	D	–	PC→Loc SP + 1	SP + 1
JSBD	23	H	L	H	H	H	L	L	JSB D	D	D + C_i	D	–	PC→Loc SP + 1	SP + 1
JSBZ	24	H	H	L	L	L	L	L	JSB "O"	"O"	"O" + C_i	D	–	PC→Loc SP + 1	SP + 1
JSRD	25	H	H	L	L	H	L	L	JSB R + D	R + D + C_n	R + D + C_n + C_i	D	–	PC→Loc SP + 1	SP + 1
JSPD	26	H	H	L	H	L	L	L	JSB PC + D	PC + D + C_n	PC + D + C_n + C_i	D	–	PC→Loc SP + 1	SP + 3
JSPR	27	H	H	L	H	H	L	L	JSB PC + R	PC + R + C_n	PC + R + C_n + C_i	D	–	PC→Loc SP + 1	SP + 1
RTS	28	H	H	H	L	L	L	L	RETURN S	S	S + C_i	D	–	–	SP–1
RTSD	29	H	H	H	L	H	L	L	RETURN S + D	S + D + C_n	S + D + C_n + C_i	D	–	–	SP–1
CHLD	30	H	H	H	H	L	L	L	HOLD	PC	–	D	–	–	–
PSUS	31	H	H	H	H	H	L	L	SUSPEND	Z (Note 2)	–	D	–	–	–

PC – Program Counter
R – Auxiliary Register
SP – Stack Pointer
D – Direct Inputs
S – Stack Top

Notes: 1. When $\overline{IEN}$ is HIGH, the Y_0–Y_3 outputs contain the same data as when $\overline{IEN}$ is LOW, as determined by I_0–I_4 and $\overline{CC}$.
2. Z = High impedance state (outputs "OFF").
3. – = No change.

APPLICATIONS

The Am2930 is shown in a typical 16-bit, 2900 Microcomputer design in Figure 1.

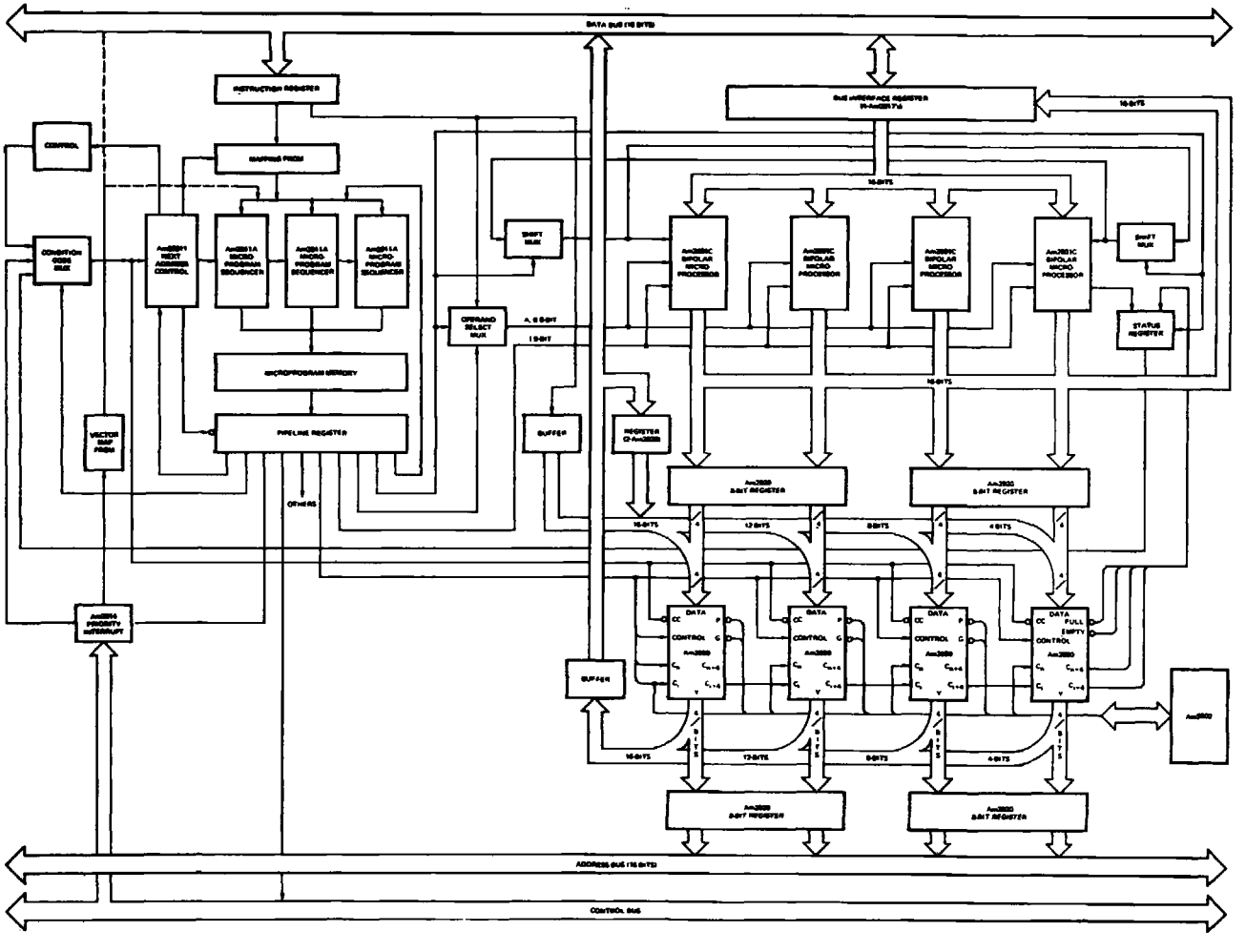
The Direct inputs (D) of the Am2930 are derived from one of three sources: the Instruction Register, the Data Bus via a 16-bit register (two Am2920 8-bit Registers), and the output of the Am2901's via a 16-bit register.

The Address outputs (Y) of the Am2930 are loaded into a 16-bit Memory Address Register (MAR). Although the MAR is shown as part of the CPU, in some applications it may be part of the memory.

An Am2902 High-Speed Lookahead Carry Generator is utilized to provide high-speed relative and indexed addressing. In slower systems, the C_{n+4} output can be wired to the next higher C_n input to provide ripple block arithmetic.

The Condition Code input ($\overline{CC}$) is derived from the same condition code multiplexer which generates the condition code input for the microprogram sequencer.

The control inputs of the Am2930 (I_{0-4}, $\overline{IEN}$, $\overline{RE}$, $\overline{OE}$, and C_i and C_n of the least significant device) are shown originating at the Pipeline Register. Although it is not shown in Figure 1, it is possible to share the Pipeline Register outputs which go to these pins with another device. This can be accomplished if both the Am2930 and the other device do not operate on the same microcycle. Forcing the $\overline{IEN}$ input HIGH inhibits any changes in the Am2930 internal registers, independent of the state of these seven inputs. This allows the Am2930 to be placed in a hold mode while the other device is using the same Pipeline Register outputs as control signals.



AF001791

Figure 1. Typical 16-Bit Microcomputer Design.

Refer to Page 13-1 for Essential Information on Military Devices

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
(Ambient) Temperature Under Bias ..... −55°C to +125°C
Supply Voltage to Ground Potential
Continuous ..... −0.5V to +7.0V
DC Voltage Applied to Outputs For
High Output State ..... −0.5V to +V_{CC} max
DC Input Voltage ..... −0.5V to +5.5V
DC Output Current, Into Outputs 30mA
DC Input Current ..... −30mA to +5.0mA

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 ..... 0°C to +70°C
Supply Voltage ..... +4.75V to +5.25V

Military (M) Devices
Temperature ..... −55°C to +125°C
Supply Voltage ..... +4.5V to +5.5V

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

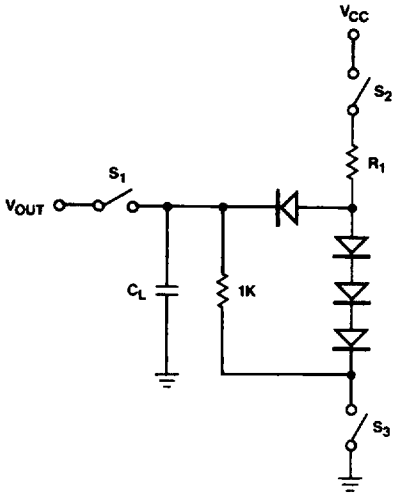
DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions (Note 2)			Min	Typ (Note 1)	Max	Units
V_{OH}	Output HIGH Voltage	V_{CC} = MIN, V_{IN} = V_{IL} or V_{IH}	Y_0, Y_1, Y_2, Y_3 $\overline{G}$, C_{n+4} C_{i+4}	I_{OH} = −1.6mA	2.4			Volts
			P, $\overline{FULL}$, $\overline{EMPTY}$	I_{OH} = −1.2mA	2.4			
V_{OL}	Output LOW Voltage	V_{CC} = MIN V_{IN} = V_{IL} or V_{IH}	Y_0, Y_1, Y_2, Y_3	I_{OL} = 20mA (COM'L)			0.5	Volts
				I_{OL} = 16mA (MIL)			0.5	
			$\overline{G}$, C_{n+4} C_{i+4}	I_{OL} = 16mA			0.5	
			P, $\overline{FULL}$ $\overline{EMPTY}$	I_{OL} = 12mA			0.5	
V_{IH}	Input HIGH Level (Note 4)				2.0			Volts
V_{IL}	Input LOW Level (Note 4)						0.8	Volts
V_I	Input Clamp Voltage	V_{CC} = MIN, I_{IN} = −18mA					−1.5	Volts
I_{IL}	Input LOW Current	V_{CC} = MAX, V_{IN} = 0.5V	D_{0-3}				−.360	mA
			I_{0-4}, $\overline{RE}$, $\overline{IEN}$, CP, $\overline{OE}$				−.702	
			$\overline{CC}$				−.657	
			C_i				−2.31	
			C_n				−3.25	
I_{IH}	Input HIGH Current	V_{CC} = MAX, V_{IN} = 2.7V	D_{0-3}				20	µA
			I_{0-4}, $\overline{RE}$, $\overline{IEN}$, CP, $\overline{OE}$				40	
			$\overline{CC}$				50	
			C_i				90	
			C_n				250	
I_I	Input HIGH Current	V_{CC} = MAX, V_{IN} = 5.5V					1.0	mA
I_{SC}	Output Short Circuit Current (Note 3)	V_{CC} = MAX			−30		−85	mA
I_{OZL}	Output OFF Current	V_{CC} = MAX, $\overline{OE}$ = 2.4V	Y_{0-3}	V_{OUT} = 0.5V			−50	µA
I_{OZH}				V_{OUT} = 2.4V			50	
I_{CC}	Power Supply Current (Note 5)	V_{CC} = MAX		T_C = −55 to +125°C			239	mA
				T_C = +125°C			170	
				T_A = 0 to 70°C			220	
				T_A = 70°C			185	

Notes: 1. Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
2. For conditions shown as MIN or MAX, use the appropriate value specified under Operating Ranges for the applicable device type.
3. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
4. These input levels provide no guaranteed noise immunity and should only be tested in a static-, noise-free environment.
5. Minimum I_{CC} is at maximum temperature.

SWITCHING TEST CIRCUIT

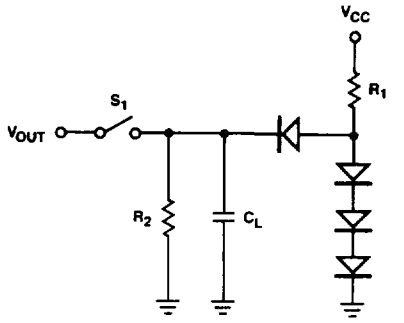
A. THREE STATE OUTPUTS



TCR01350

$$R_1 = \frac{5.0 - V_{BE} - V_{OL}}{I_{OL} + V_{OL}/1K}$$

B. NORMAL OUTPUTS



TCR01370

$$R_2 = \frac{2.4V}{I_{OH}}$$

$$R_1 = \frac{5.0 - V_{BE} - V_{OL}}{I_{OL} + V_{OL}/R_2}$$

Notes: 1. C_L = 50pF 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 – 5.0pF for output disable tests.

TEST OUTPUT LOADS FOR Am2930

Pin # (DIP)	Pin Label	Test Circuit	R_1	R_2
2	$\overline{FULL}$	B	300	2K
3	$\overline{EMPTY}$	B	300	2K
6	C_{i+4}	B	240	1.5K
8–11	Y_{0-3}	A	240	1K
12	$\overline{G}$	B	240	1.5K
13	C_{n+4}	B	240	1.5K
16	P	B	300	2K

For additional information on testing, see section "Guidelines on Testing Am2900 Family Devices."

Am2930 SWITCHING CHARACTERISTICS

Tables A, B, C and D define the timing characteristics of the Am2930. Measurements are made at 1.5V with V_{IL} = 0V and V_{IH} = 3.0V. For three-state disable tests, C_L = 5.0pF and measurement is to 0.5V change on output voltage level.

I. GUARANTEED PERFORMANCE OVER COMMERCIAL OPERATING RANGE

V_{CC} = 4.75 to 5.25V, T_A = 0 to 70°C

TABLE IA
Clock Characteristics.

Minimum Clock LOW Time	31ns
Minimum Clock HIGH Time	33ns

TABLE IB
Output Enable/Disable Times.
All in ns.
C_L = 5.0pF for output disable tests.

From	To	Enable	Disable
$\overline{OE}$	Y	27	26
$\overline{CC}$ (Note 1)	Y	55	37
I_{4-0} (Note 1)	Y	80	55

Note 1: "Suspend" instruction.

TABLE IC
Combinational Propagation Delays.
All in ns.
Outputs fully loaded. C_L = 50pF.

	To Output						
From Input	Y	$\overline{G}$, $\overline{P}$	C_{n+4}	C_{i+4} I_4 = L	C_{i+4} I_4 = H	$\overline{Full}$	$\overline{Empty}$
I_{4-0}	81	67	77	80	91	69	–
$\overline{CC}$	63	45	55	–	72	42	–
C_n	32	–	25	–	45	–	–
C_i	–	–	–	22	22	–	–
CP	69	53	61	43	78	55	55
D	49	33	40	–	59	–	–
$\overline{IEN}$	–	–	–	–	–	40	–

TABLE ID
Set-up and Hold Times. All in ns.
All relative to clock LOW-to-HIGH transition.

	CP:	
Input	Set-up Time	Hold Time
I_{4-0}	114	0
$\overline{CC}$	75	0
$\overline{IEN}$	55	0
C_n	43	0
C_i	32	5
D ($\overline{RE}$ = L I_{4-0} = 0-8 or 10-15)	25	2
$\overline{RE}$	24	4

II. GUARANTEED PERFORMANCE OVER MILITARY OPERATING RANGE

V_{CC} = 4.5 to 5.5V, T_C = −55 to +125°C

TABLE IIA
Clock Characteristics.

Minimum Clock LOW Time	35ns
Minimum Clock HIGH Time	35ns

TABLE IIB
Output Enable/Disable Times.
All in ns.
C_L = 5.0pF for output disable tests.

From	To	Enable	Disable
$\overline{OE}$	Y	32	31
$\overline{CC}$ (Note 1)	Y	60	42
I_{4-0} (Note 1)	Y	85	60

Note 1: "Suspend" instruction.

TABLE IIC
Combinational Propagation Delays.
All in ns.
Outputs fully loaded. C_L = 50pF.

	To Output						
From Input	Y	$\overline{G}$, $\overline{P}$	C_{n+4}	C_{i+4} I_4 = L	C_{i+4} I_4 = H	$\overline{Full}$	$\overline{Empty}$
I_{4-0}	88	74	82	87	97	78	–
$\overline{CC}$	68	52	60	–	78	47	–
C_n	37	–	30	–	46	–	–
C_i	–	–	–	23	23	–	–
CP	74	58	66	48	84	60	60
D	55	38	45	–	65	–	–
$\overline{IEN}$	–	–	–	–	–	45	–

TABLE IID
Setup and Hold Times. All in ns.
All relative to clock LOW-to-HIGH transition.

	CP:	
Input	Set-up Time	Hold Time
I_{4-0}	124	0
$\overline{CC}$	80	0
$\overline{IEN}$	69	0
C_n	52	0
C_i	37	5
D ($\overline{RE}$ = L I_{4-0} = 0-8 or 10-15)	30	2
D (All other conditions)	72	2
$\overline{RE}$	29	4

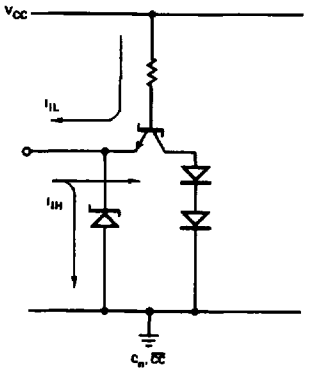
Notes on Testing

Incoming test procedures on this device should be carefully planned, taking into account the high complexity and power levels of the part. The following notes may be useful:

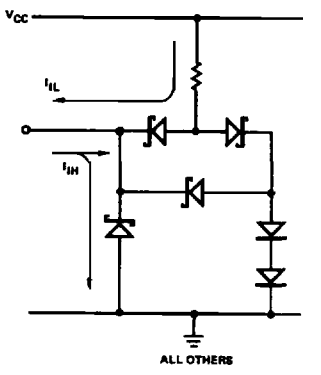
1. Insure the part is adequately decoupled at the test head. Large changes in V_{CC} current when the device switches may cause erroneous function failures due to V_{CC} changes.
2. Do not leave inputs floating during any tests, as they may start to oscillate at high frequency.
3. Do not attempt to perform threshold tests at high speed. Following an input transition, ground current may change by as much as 400mA in 5–8ns. Inductance in the ground cable may allow the ground pin at the device to rise by 100s of millivolts momentarily.
4. Use extreme care in defining input levels for AC tests. Many inputs may be changed at once, so there will be significant noise at the device pins and they may not actually reach V_{IL} or V_{IH} until the noise has settled. AMD recommends using $V_{IL} \leq 0V$ and $V_{IH} \geq 3.0V$ 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. To assist in testing, AMD offers complete documentation on our test procedures and, in most cases, can provide Fairchild Sentry programs, under license.

INPUT/OUTPUT CURRENT INTERFACE CONDITIONS

Driven Inputs



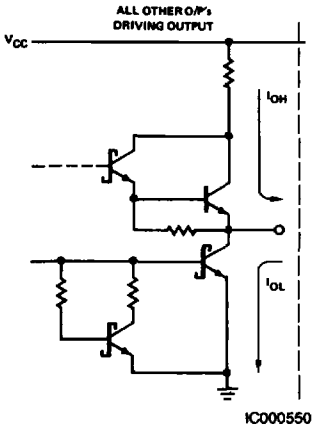
IC000570



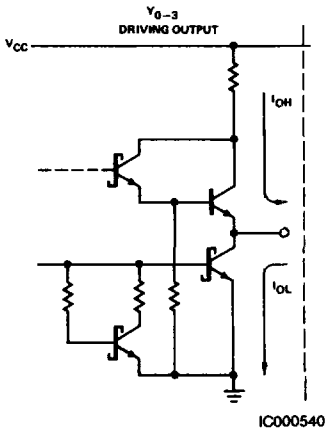
IC000560

Note: C_i input is connected to both configurations in parallel.

Driving Outputs



IC000550



IC000540

Note: Actual current flow direction shown.

RELATED PRODUCTS

Part No.	Description
Am2902A	Carry Look-Ahead Generator
Am2904	Status and Shift Control Unit
Am2920	8-Bit Register
Am2922	Condition Code MUX

For applications information, see Chapter V of **Bit Slice Microprocessor Design**, Mick & Brick, McGraw Hill Publications.