

## AM2904

### *Status and Shift Control Unit*

The AM2904 is designed to perform all the miscellaneous functions which are usually performed in MSI around an ALU. These include the generation of the carry-in signal to the ALU and carry lookahead unit; the interconnection of the data path, auxiliary register, and carry flip-flop during shift operations; and the storage and testing of ALU status flags. These tasks are accomplished in the AM2904 by three nearly independent blocks of logic. The carry-in is generated by a multiplexer. The shift linkages are established by four three-state multiplexers. There are two registers for storing the carry, overflow, zero, and negative status flags. The condition code multiplexer on the AM2904 can lock at true or complement of any of the four status bits and certain combinations of status bits from either of the storage registers or directly from the ALU.

---

#### **Rochester Electronics Manufactured Components**

Rochester branded components are manufactured using either die/wafers purchased from the original suppliers or Rochester wafers recreated from the original IP. All recreations are done with the approval of the OCM.

Parts are tested using original factory test programs or Rochester developed test solutions to guarantee product meets or exceeds the OCM data sheet.

#### **Quality Overview**

- ISO-9001
- AS9120 certification
- Qualified Manufacturers List (QML) MIL-PRF-38535
  - Class Q Military
  - Class V Space Level
- Qualified Suppliers List of Distributors (QSLD)
  - Rochester is a critical supplier to DLA and meets all industry and DLA standards.

Rochester Electronics, LLC is committed to supplying products that satisfy customer expectations for quality and are equal to those originally supplied by industry manufacturers.

---

*The original manufacturer's datasheet accompanying this document reflects the performance and specifications of the Rochester manufactured version of this device. Rochester Electronics guarantees the performance of its semiconductor products to the original OEM specifications. 'Typical' values are for reference purposes only. Certain minimum or maximum ratings may be based on product characterization, design, simulation, or sample testing.*

# Am2904

Status and Shift Control Unit



Am2904

Advanced Micro Devices

## DISTINCTIVE CHARACTERISTICS

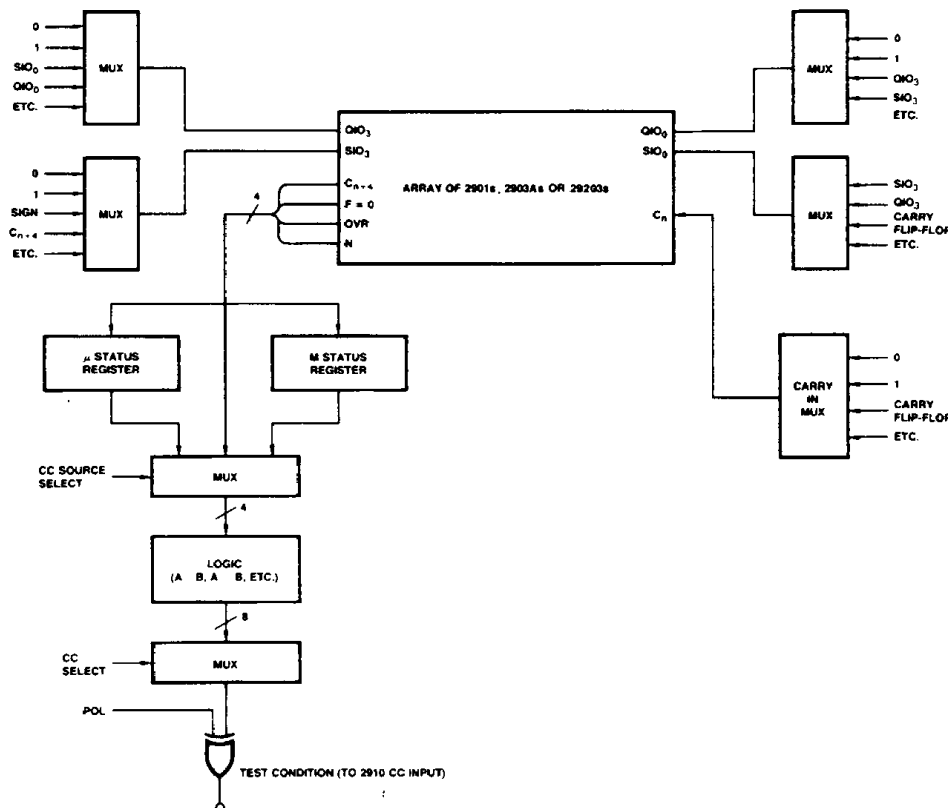
- **Replaces most MSI used around any ALU**
  - Including the Am2901, Am2903A, Am29203 and MSI ALUs.
- **Generates Carry-in to the ALU**
  - Carry signal is selectable from 7 different sources.
- **Contains shift linkage multiplexers**
  - Connects to shift lines at the ends of an Am2901, Am2903A, or Am29203 array to implement single and double length arithmetic and logical shifts and rotates — 32 different modes in all.
- **Contains two edge-triggered status registers**
  - Use for foreground/background registers in controllers or as microlevel and machine level status registers. Bit manipulating instructions are provided.
- **Condition Code Multiplexer on chip**
  - Single cycle tests for any of 16 different conditions. Tests can be performed on either of the two status registers or directly on the ALU output.

## GENERAL DESCRIPTION

The Am2904 is designed to perform all the miscellaneous functions which are usually performed in MSI around an ALU. These include the generation of the carry-in signal to the ALU and carry lookahead unit; the interconnection of the data path, auxiliary register, and carry flip-flop during shift operations; and the storage and testing of ALU status flags. These tasks are accomplished in the Am2904 by three nearly independent blocks of logic. The carry-in is

generated by a multiplexer. The shift linkages are established by four three-state multiplexers. There are two registers for storing the carry, overflow, zero, and negative status flags. The condition code multiplexer on the Am2904 can look at true or complement of any of the four status bits and certain combinations of status bits from either of the storage registers or directly from the ALU.

## SIMPLIFIED BLOCK DIAGRAM



## BASIC FEATURES OF Am2904

All the logic shown except the array of 2901s, 2903As, or 29203s is contained in the Am2904.

DDU1003

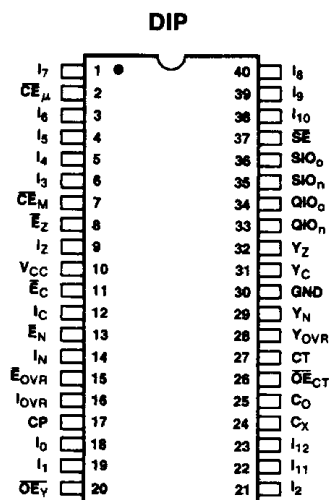
Publication #	Rev.	Amendment
03582	C	/0
Issue Date: January 1987		

## RELATED AMD PRODUCTS

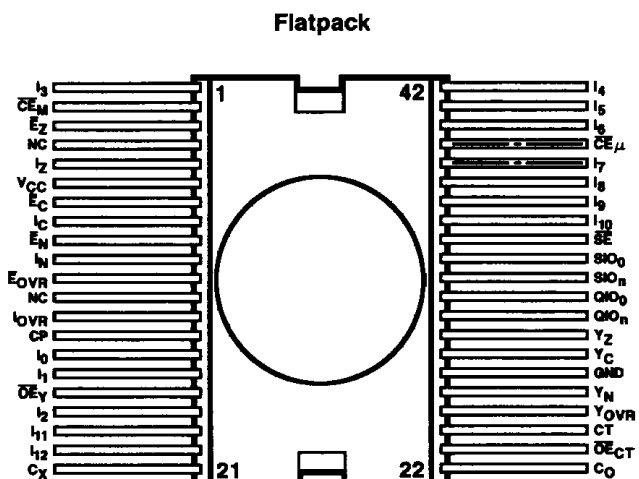
Part No.	Description
Am2901	4-Bit Microprocessor Slice
Am2903A	Advanced 4-Bit Bipolar Microprocessor Slice
Am29203	4-Bit Bipolar Microprocessor Slice
Am2910A	Microprogram Controller
Am29811	Next Address Control Unit for 2909A/11A

For additional applications refer to Chapter 4 of **Bit Slice Microprocessor Design**, Mick & Brick, McGraw Hill Publications.

## CONNECTION DIAGRAMS Top View



CD004131

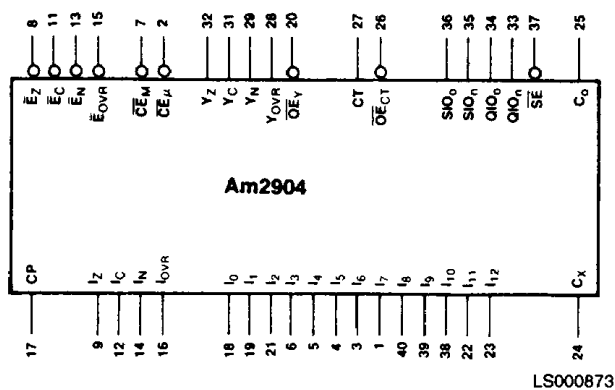


CD008033

Note: Pin 1 is marked for orientation.

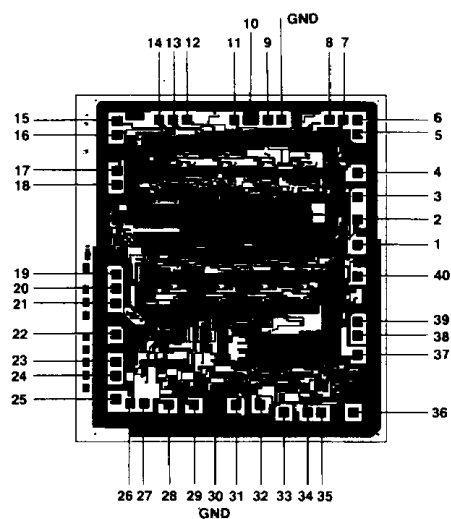
NC = No Connection.

## LOGIC SYMBOL (DIP)



LS000873

## METALLIZATION AND PAD LAYOUT



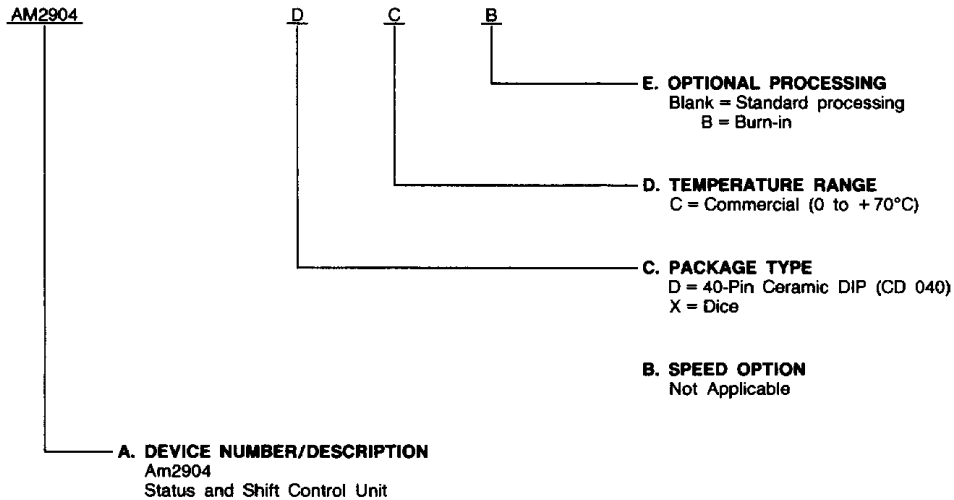
DIE SIZE 0.140" x 0.161"  
Pad Numbers correspond to DIP pinout

## ORDERING INFORMATION

### Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

- A. Device Number**
- B. Speed Option** (if applicable)
- C. Package Type**
- D. Temperature Range**
- E. Optional Processing**



Valid Combinations	
AM2904	DC, DCB, XC

### Valid Combinations

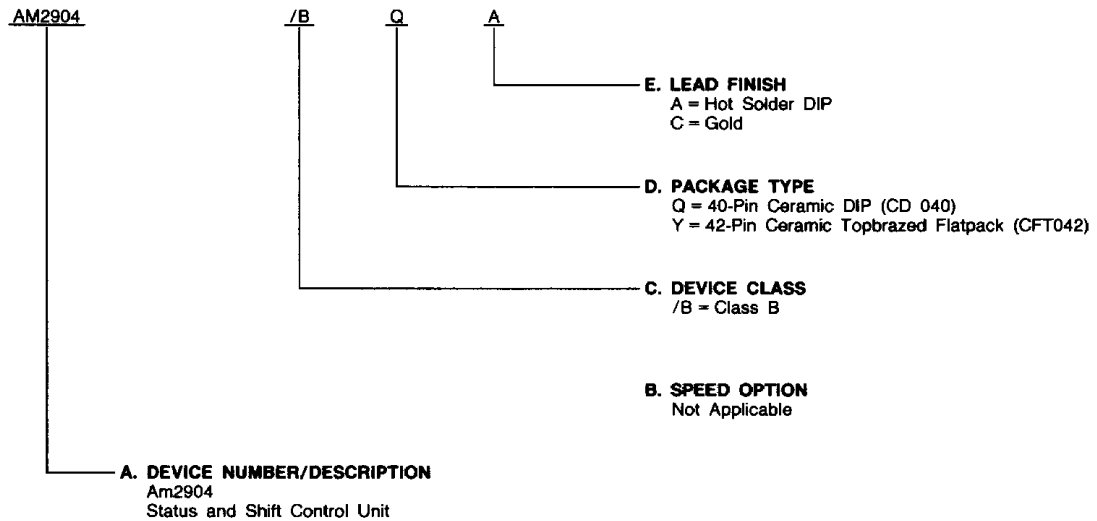
Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

## ORDERING INFORMATION (Cont'd.)

### 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. CPL (Controlled Products List) products are processed in accordance with MIL-STD-883C, but are inherently non-compliant because of package, solderability, or surface treatment exceptions to those specifications. The order number (Valid Combination) for APL products is formed by a combination of:

- A. Device Number**
- B. Speed Option** (if applicable)
- C. Device Class**
- D. Package Type**
- E. Lead Finish**



#### Valid Combinations

Valid Combinations	
AM2904	/BQA, /BYC

#### Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check for newly released valid combinations.

#### Group A Tests

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

## PIN DESCRIPTION

### **$\overline{CE}_M$ Machine Status Register Enable (Input; Active LOW)**

This pin, used in conjunction with  $\overline{E}_Z$ ,  $\overline{E}_C$ ,  $\overline{E}_N$ , and  $\overline{E}_{OVR}$ , acts as the overall enable for the Machine Status Register (MSR). When this pin is LOW, MSR bits may be modified, according to the states of  $\overline{E}_Z$ ,  $\overline{E}_C$ ,  $\overline{E}_N$ , and  $\overline{E}_{OVR}$ . When HIGH, the MSR will retain the present state, regardless of the state of  $\overline{E}_Z$ ,  $\overline{E}_C$ ,  $\overline{E}_N$ , and  $\overline{E}_{OVR}$ .

### **$\overline{CE}_\mu$ Micro Status Register Enable (Input; Active LOW)**

This pin, when LOW, enables all four bits of the Micro Status Register ( $\mu$ SR). When this pin is HIGH, the  $\mu$ SR will not change state.

### **$C_O$ Carry Multiplexer Out (Output)**

This is the output of the Carry-In Control Multiplexer. It connects to the  $C_n$  input of the least significant ALU slice, and the  $C_n$  input of the Am2902A.

### **$C_P$ Clock Pulse (Input)**

The clock input to the device. The  $\mu$ SR and MSR are modified on the LOW-to-HIGH transition of the clock input. All other portions of the Am2904 are combinational and are unaffected by  $C_P$ .

### **$C_T$ Conditional Test (Output)**

The output of the Condition Code Multiplexer appears here.

### **$C_X$ Carry Multiplexer In (Input)**

This pin is used as an input to the Carry-In Control Multiplexer and can route it to the  $C_O$  pin. The  $C_X$  pin is intended for connection to the Z output of the Am2903A to facilitate some of the Am2903A special instructions.

### **$\overline{E}_Z$ , $\overline{E}_C$ , $\overline{E}_N$ , and $\overline{E}_{OVR}$ Zero, Carry, Sign, and Overflow Status Enable (Input; Active LOW)**

These pins, when LOW, enable the corresponding bits in the MSR. When HIGH, they will prevent the corresponding bits from changing state. By using these pins together with the  $\overline{CE}_M$  pin, MSR bits can be selectively modified.

### **$I_0 - I_{12}$ Instruction Pins (Input)**

The thirteen instruction pins that select the operation the Am2904 is to perform.

### **$I_C$ Carry Status (Input)**

Intended for connection to the  $C_n + 4$  output of the most significant ALU slice.

### **$I_{OVR}$ Overflow Status (Input)**

Intended for connection to the OVR pin on the most significant ALU slice.

### **$I_N$ Sign Status (Input)**

Intended for connection to the most significant ALU slice. The connection is to the N pin on the Am2903A, and the  $F_3$  pin on the Am2901.

### **$I_Z$ Zero Status (Input)**

Intended for connection to the Z outputs of the Am2903A or the  $F = 0$  outputs of the Am2901.

### **$\overline{OE}_{CT}$ CT Output Enable (Input; Active LOW)**

When this pin is LOW, the CT pin is active. When HIGH the CT pin is in the high-impedance state.

### **$\overline{OE}_Y$ Y Output Enable (Input; Active LOW)**

When LOW, this pin enables the Y pins as outputs. When HIGH, the Y outputs are in the high-impedance state.

### **$\overline{SE}$ Shift Enable (Input; Active LOW)**

This pin controls the state of the shift outputs. When LOW, the shift outputs are enabled. When HIGH, the shift outputs are in the high-impedance state.

### **$SIO_0$ , $SIO_n$ , $QIO_0$ , $QIO_n$ Serial Shift (Input/Output)**

These pins complete the linking for the various shift and rotate conditions.  $SIO_0$  is intended for connection to the  $SIO_0$  pin of the least significant Am2903A slice ( $RAM_0$  for Am2901).  $SIO_n$  connects to the  $SIO_3$  pin of the most significant Am2903A slice ( $RAM_3$  for Am2901).  $QIO_0$  connects to the  $QIO_0$  pin of the least significant Am2903A slice ( $QIO_0$  for Am2901) and  $QIO_n$  connects to the  $QIO_3$  pin of the most significant Am2903A slice ( $Q_3$  for Am2901).

### **$Y_Z$ , $Y_C$ , $Y_N$ , and $Y_{OVR}$ Zero, Carry, Negative and Overflow Status (Input/Output; Three State)**

These pins form a three-state bidirectional bus over which MSR and  $\mu$ SR status can be read out or the MSR can be loaded in parallel.

## FUNCTIONAL DESCRIPTION

### Am2904 Architecture

The Am2904 Status and Shift Control Unit provides four functions which are included in all processors. These are: a) Status Register, b) Condition Code Multiplexer, c) Shift Linkages and d) Carry-in Control. The architecture and instruction codes have been designed to complement the flexibility of the 2900 Family.

#### Status Register

The Am2904 contains two four-bit registers which can store the status outputs of an ALU: Carry (C), Negative (N), Zero (Z), and Overflow (OVR). They are designated Micro Status Register ( $\mu$ SR) and Machine Status Register (MSR). Each register can be independently controlled. The registers use edge-triggered D-type flip-flops which change state on the LOW to HIGH transition of the Clock Input.

The  $\mu$ SR can be loaded from the four status inputs ( $I_C$ ,  $I_N$ ,  $I_Z$ ,  $I_{OVR}$ ) or from the MSR under instruction control ( $I_{0-5}$ ). The bits in the  $\mu$ SR can also be individually set or reset under instruction control ( $I_{0-5}$ ). When the  $\overline{CE}_\mu$  input is HIGH, the  $\mu$ SR is inhibited from changing, independent of the  $I_{0-5}$  inputs.

The MSR can be loaded from the four status inputs ( $I_C$ ,  $I_N$ ,  $I_Z$ ,  $I_{OVR}$ ), from the  $\mu$ SR, and from the four parallel input/output pins ( $Y_C$ ,  $Y_N$ ,  $Y_Z$ ,  $Y_{OVR}$ ) under instruction control ( $I_{0-5}$ ). The MSR can also be set, reset or complemented under instruction control ( $I_{0-5}$ ). The bits in the MSR can be selectively updated by controlling the four bit-enable inputs ( $\overline{E}_Z$ ,  $\overline{E}_N$ ,  $\overline{E}_C$ ,  $\overline{E}_{OVR}$ ) and the  $\overline{CE}_M$  input. A LOW on both the  $\overline{CE}_M$  input and the bit enable input for a specific bit enables updating that bit. A HIGH on a given bit enable input prevents the corresponding bit changing in the MSR. A HIGH on  $\overline{CE}_M$  prevents any bits changing in the MSR.

The four parallel bidirectional input/output pins ( $Y_Z$ ,  $Y_N$ ,  $Y_C$ ,  $Y_{OVR}$ ) allow the contents of both the  $\mu$ SR and the MSR to be transferred to the system data bus and also allows the MSR to be loaded from the system data bus. This capability is used to save and restore the status registers during certain subroutines and when servicing interrupts.

#### Condition Code Multiplexer

The Condition Code Multiplexer output, CT, can be selected from 16 different functions. These include the true and complemented state of each of the status bits and combinations of these bits to detect such conditions as "greater than", "greater than or equal to", "less than" or "less than or equal to" for unsigned or two's complement numbers.

The Am2904 can perform these tests on the contents of the  $\mu$ SR, the MSR or the direct status inputs ( $I_Z$ ,  $I_N$ ,  $I_C$ ,  $I_{OVR}$ ). The CT output is used as the test ( $\overline{CC}$ ) input of the Am2910A and is provided with an output enable,  $\overline{OE}_{CT}$  to make the addition of other condition inputs to this point easy.

#### Shift Linkage Multiplexer

The Shift Linkage Multiplexer generates the necessary linkages to allow the ALU to perform 32 different shift and rotate functions. Both single length and double length shifts and rotates, with and without carry ( $M_C$ ), are provided. When the  $\overline{SE}$  input is HIGH, the four input/output pins ( $SIO_0$ ,  $SIO_n$ ,

$QIO_0$ ,  $QIO_n$ ) are disabled. The  $SIO_0$ ,  $SIO_n$ ,  $QIO_0$ ,  $QIO_n$  pins of the Am2904 are intended to be directly connected to the  $RAM_0$ ,  $RAM_3$ ,  $Q_0$  and  $Q_3$  pins of the Am2901 or the  $SIO_0$ ,  $SIO_3$ ,  $QIO_0$ ,  $QIO_3$  pins of the Am2903A.

#### Carry-In Control Multiplexer

The Carry-In Control Multiplexer generates the  $C_0$  output which can be selected from 7 functions (0, 1,  $C_X$ ,  $\mu_C$ ,  $M_C$ ,  $\overline{\mu_C}$ ,  $\overline{M_C}$ ). These functions allow easy implementation of both single length and double length addition and subtraction. The  $C_X$  input is intended to be connected to the Z output of the Am2903A to facilitate execution of some of the Am2903A special instructions. The  $C_0$  pin is to be connected to the  $C_n$  pin of the least significant Am2901 or Am2903A and the  $C_n$  pin of the Am2902A.

#### Am2904 Instruction Set

The Am2904 is controlled by manipulating the 13 instruction lines,  $I_{0-12}$ , together with the nine enable lines,  $\overline{CE}_M$ ,  $\overline{CE}_\mu$ ,  $\overline{E}_Z$ ,  $\overline{E}_C$ ,  $\overline{E}_N$ ,  $\overline{E}_{OVR}$ ,  $\overline{OE}_Y$ ,  $\overline{OE}_{CT}$ ,  $\overline{SE}$ . Most systems will save on microword bits by tying some of these lines to a fixed level or by connecting certain lines together, or by decoding microinstructions to generate appropriate Am2904 controls.

#### Status Registers

Instruction lines  $I_5$ ,  $I_4$ ,  $I_3$ ,  $I_2$ ,  $I_1$ ,  $I_0$  control the Status Registers. Below, these lines are referred to as two octal digits.

#### Micro Status Register ( $\mu$ SR)

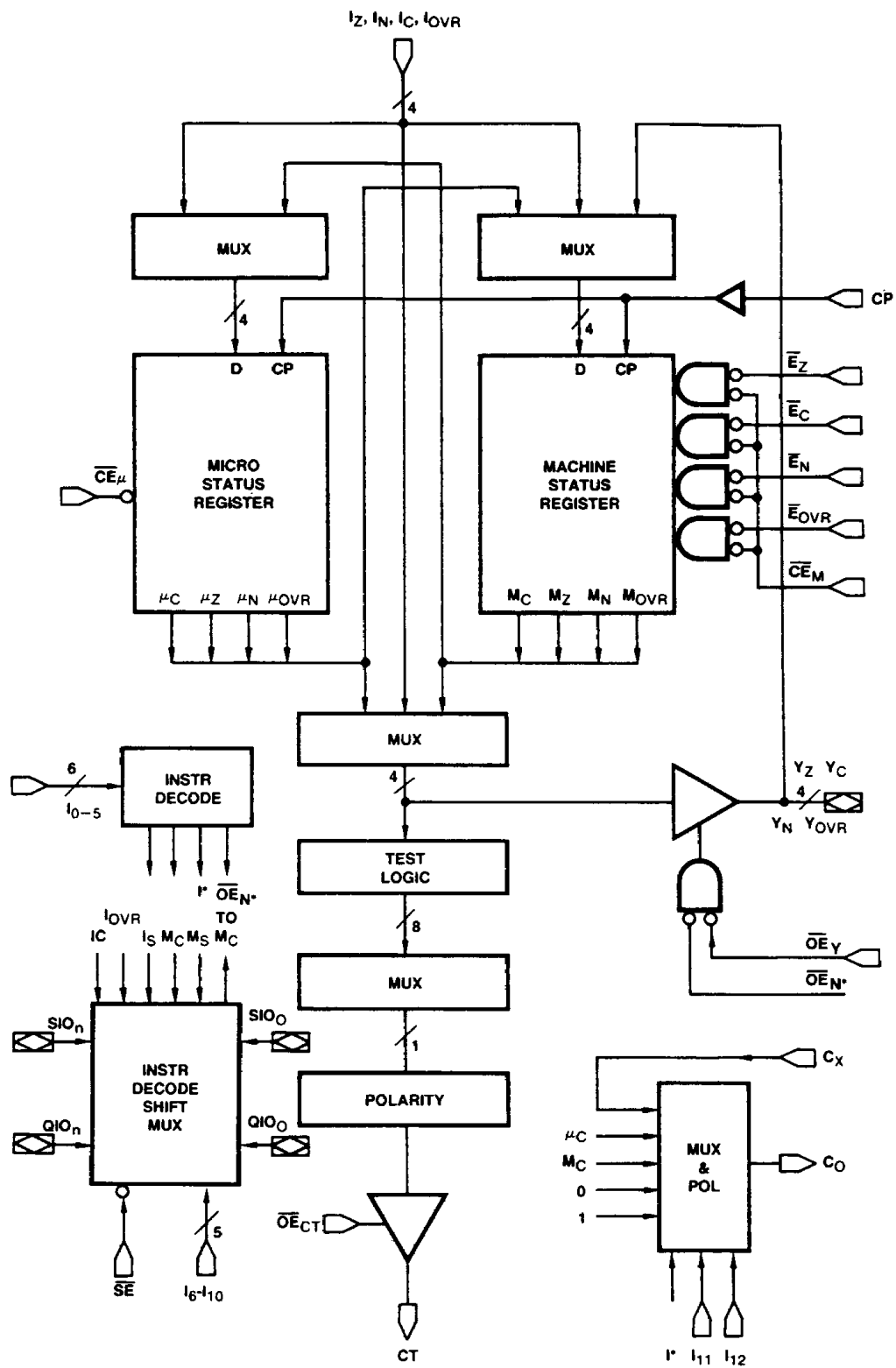
The instruction codes for the Micro Status Register fall into three groups: Bit Operations, Register Operations and Load Operations (See Table 1 and Map 1). All operations require that  $\overline{CE}_\mu$  be LOW to operate.

TABLE 1. MICRO STATUS REGISTER  
INSTRUCTION CODES

Bit Operations		
$I_{543210}$ Octal	$\mu$ SR Operation	Comments
10	$0 \rightarrow \mu Z$	RESET ZERO BIT
11	$1 \rightarrow \mu Z$	SET ZERO BIT
12	$0 \rightarrow \mu C$	RESET CARRY BIT
13	$1 \rightarrow \mu C$	SET CARRY BIT
14	$0 \rightarrow \mu N$	RESET SIGN BIT
15	$1 \rightarrow \mu N$	SET SIGN BIT
16	$0 \rightarrow \mu OVR$	RESET OVERFLOW BIT
17	$1 \rightarrow \mu OVR$	SET OVERFLOW BIT

Register Operations		
$I_{543210}$ Octal	$\mu$ SR Operation	Comments
00	$M_X \rightarrow \mu X$	LOAD MSR TO $\mu$ SR
01	$1 \rightarrow \mu X$	SET $\mu$ SR
02	$M_X \rightarrow \mu X$	REGISTER SWAP
03	$0 \rightarrow \mu X$	RESET $\mu$ SR





\*INTERNAL

BD001841

Figure 1. Detailed Block Diagram

TABLE 1. (Cont'd.)

Load Operations		
I543210 Octal	$\mu$ SR Operation	Comments
06, 07	$I_Z \rightarrow \mu Z$ $I_C \rightarrow \mu C$ $I_N \rightarrow \mu N$ $I_{OVR} + \mu OVR \rightarrow \mu OVR$	LOAD WITH OVERFLOW RETAIN
30, 31 50, 51 70, 71	$I_Z \rightarrow \mu Z$ $I_C \rightarrow \mu C$ $I_N \rightarrow \mu N$ $I_{OVR} \rightarrow \mu OVR$	LOAD WITH CARRY INVERT
04, 05 20-27 32-47 52-67 72-77	$I_Z \rightarrow \mu Z$ $I_C \rightarrow \mu C$ $I_N \rightarrow \mu N$ $I_{OVR} \rightarrow \mu OVR$	LOAD DIRECTLY FROM $I_Z, I_C, I_N, I_{OVR}$

Note: The above table assumes  $\overline{CE}_{\mu}$  is LOW.

MAP 1. MICRO STATUS REGISTER  
INSTRUCTION CODES

I543	000	001	010	011	100	101	110	111
000	LOAD MSR TO $\mu$ SR	SET $\mu$ SR	REG SWAP	RESET $\mu$ SR				LOAD WITH OVERFLOW RETAIN
001	RESET $\mu Z$	SET $\mu Z$	RESET $\mu C$	SET $\mu C$	RESET $\mu N$	SET $\mu N$	RESET $\mu OVR$	SET $\mu OVR$
010	LOAD WITH CARRY INVERT							
011								
100								
101								
110	LOAD WITH CARRY INVERT							
111								

DF000890

Notes: 1. All unmarked locations are a load direct from  $I_Z, I_C, I_N, I_{OVR}$ .

Instruction Codes 10<sub>8</sub> to 17<sub>8</sub> are BIT operations. These operations set or reset the individual bits in the  $\mu$ SR.

Instruction Codes 00<sub>8</sub> to 03<sub>8</sub> are REGISTER operations. These operations affect all bits in the  $\mu$ SR.

00<sub>8</sub> This instruction loads the  $\mu$ SR with the contents of the MSR while loading the MSR from the Y inputs and is further explained under "INTERRUPTS".

01<sub>8</sub> This instruction SETS all  $\mu$ SR bits.

02<sub>8</sub> This instruction SWAPS the contents of the  $\mu$ SR and the MSR. It will also COPY one register to the other if the register to be copied is not enabled.

03<sub>8</sub> This instruction RESETS all  $\mu$ SR bits.

All instruction codes except those mentioned in the above two sections cause a LOAD operation from the  $I_Z, I_C, I_N, I_{OVR}$  inputs.

06<sub>8</sub>, 07<sub>8</sub> When a series of arithmetic operations are being executed sometimes it is not necessary to test for an overflow condition after each operation, but rather it is sufficient simply to know that an overflow occurred during any one of the operations. Use of these instructions captures the overflow condition by loading the  $\mu$ SR overflow bit with the LOGICAL-OR of its present state and  $I_{OVR}$ . Thus, once an overflow occurs,  $\mu OVR$  will remain set throughout the remaining operations.

30<sub>8</sub>, 31<sub>8</sub>, 50<sub>8</sub>, 51<sub>8</sub>, 70<sub>8</sub>, 71<sub>8</sub> These instructions cause a load from the I inputs, but invert the carry bit. The reason for this is explained more fully under the "BORROW SAVE" section.

All others The remaining instructions load the  $\mu$ SR directly from the  $I_Z, I_C, I_N, I_{OVR}$  inputs.

#### Machine Status Register (MSR)

The instruction codes for the MSR fall into two groups; REGISTER Operations and LOAD Operations. All operations require that  $\overline{CE}_M$  be LOW to operate (See Table 2 and Map 2). BIT operations are accomplished by the use of Register or Load Operations with the  $E_Z, E_C, E_N, E_{OVR}$  inputs selectively set LOW.

Instruction codes 00<sub>8</sub>-03<sub>8</sub> and 05<sub>8</sub> are REGISTER operations. They affect only those bits enabled by  $E_Z, E_C, E_N, E_{OVR}$ .

00<sub>8</sub> This instruction loads the MSR from the Y inputs while transferring the present contents to the  $\mu$ SR. The use of this instruction is further explained under "INTERRUPTS".

01<sub>8</sub> This instruction SETS all enabled MSR bits.

02<sub>8</sub> This instruction SWAPS the contents of the  $\mu$ SR and the MSR. It will also COPY one register to the other if the register to be copied is not enabled.

03<sub>8</sub> This instruction RESETS all enabled MSR bits.

05<sub>8</sub> This instruction COMPLEMENTS all enabled MSR bits.

All instruction codes except those mentioned in the above section cause a LOAD operation from the  $I_Z, I_C, I_N, I_{OVR}$  inputs.

04<sub>8</sub> The Am2904 Shift Linkage Multiplexer allows for shifts and rotates through the MSR CARRY bit. Some machines require a shift or rotate through the OVERFLOW bit. By using this code, which swaps the contents of the MSR CARRY bit ( $M_C$ ) and OVERFLOW bit ( $M_{OVR}$ ), the shift or rotate can be made to appear to take place through the OVERFLOW bit. The procedure is to swap the bits, shift or rotate (any number or positions) then swap the bits again.

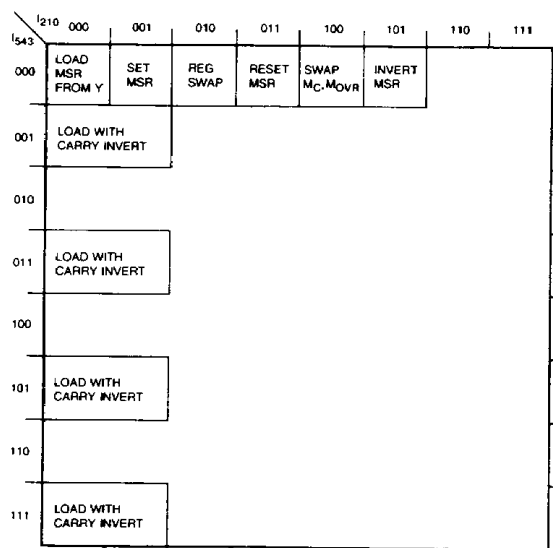
**TABLE 2. MACHINE STATUS REGISTER INSTRUCTION CODES**

Register Operations		
I <sub>5</sub> 43210 Octal	MSR Operation	Comments
00	Y <sub>X</sub> → M <sub>X</sub>	LOAD Y <sub>Z</sub> , Y <sub>C</sub> , Y <sub>N</sub> , Y <sub>OV</sub> R TO MSR
01	1 → M <sub>X</sub>	SET MSR
02	μ <sub>X</sub> → M <sub>X</sub>	REGISTER SWAP
03	0 → M <sub>X</sub>	RESET MSR
05	M̄ <sub>X</sub> → M <sub>X</sub>	INVERT MSR

Load Operations		
I <sub>5</sub> 43210 Octal	MSR Operation	Comments
04	I <sub>Z</sub> → M <sub>Z</sub> M <sub>OV</sub> R → M <sub>C</sub> I <sub>N</sub> → M <sub>N</sub> M <sub>C</sub> → M <sub>OV</sub> R	LOAD FOR SHIFT THROUGH OVERFLOW OPERATION
10, 11 30, 31 50, 51 70, 71	I <sub>Z</sub> → M <sub>Z</sub> I <sub>C</sub> → M <sub>C</sub> I <sub>N</sub> → M <sub>N</sub> I <sub>OV</sub> R → M <sub>OV</sub> R	LOAD WITH CARRY INVERT
06, 07 12-27 32-47 52-67 72-77	I <sub>Z</sub> → M <sub>Z</sub> I <sub>C</sub> → M <sub>C</sub> I <sub>N</sub> → M <sub>N</sub> I <sub>OV</sub> R → M <sub>OV</sub> R	LOAD DIRECTLY FROM I <sub>Z</sub> , I <sub>C</sub> I <sub>N</sub> , I <sub>OV</sub> R

- Notes: 1. The above tables assume  $\overline{CE}_M$ ,  $\overline{E}_Z$ ,  $\overline{E}_C$ ,  $\overline{E}_N$ ,  $\overline{E}_{OV}$  are LOW.  
2. A shift-through-carry instruction loads M<sub>C</sub> irrespective of I<sub>5</sub>-I<sub>0</sub>.

**MAP 2. MACHINE STATUS REGISTER INSTRUCTION CODES**



DF000900

06<sub>8</sub>, 07<sub>8</sub> These instructions load the MSR directly from the I<sub>Z</sub>, I<sub>C</sub>, I<sub>N</sub>, I<sub>OV</sub>R inputs.  
12<sub>8</sub>-27<sub>8</sub>  
32<sub>8</sub>-47<sub>8</sub>  
52<sub>8</sub>-67<sub>8</sub>  
72<sub>8</sub>-77<sub>8</sub>

10<sub>8</sub>, 11<sub>8</sub> These instructions cause a load from the I inputs  
30<sub>8</sub>, 31<sub>8</sub> but invert the CARRY bit. The reason for this is  
50<sub>8</sub>, 51<sub>8</sub> explained more fully under the 'BORROW SAVE'  
70<sub>8</sub>, 71<sub>8</sub> section.

### Condition Code Multiplexer

The two instruction lines I<sub>4</sub>, I<sub>5</sub> select whether the μSR, the MSR or the direct inputs I<sub>Z</sub>, I<sub>C</sub>, I<sub>N</sub>, I<sub>OV</sub>R are used as the inputs to the Y output buffer and the CT output (see Tables 3 and 4).

The four instruction lines I<sub>3</sub>, I<sub>2</sub>, I<sub>1</sub>, I<sub>0</sub> will select one of 16 possible operations to be carried out on the input bits, the result being routed to the Conditional Test Output (CT). Eight of the operations supply an individual status bit or its complement to the CT output. Another four do more complex operations while the remaining four are the complemented results of these (See Table 4).

The more complex operations are intended to follow the calculation A-B to give an indication of which is the larger (A, B unsigned) or more positive (A, B in 2's complement form). See Table 5.

Instruction codes 16<sub>8</sub> and 17<sub>8</sub> form the EXCLUSIVE-OR and the EXCLUSIVE-NOR functions of M<sub>N</sub> and I<sub>N</sub>. The use of these instructions is explained under "NORMALIZING".

**TABLE 3. Y OUTPUT INSTRUCTION CODES**

$\overline{OE}_Y$	I <sub>5</sub>	I <sub>4</sub>	Y Output	Comment
H	X	X	Z	Output Off High Impedance
L	L	X	μ <sub>i</sub> → Y <sub>i</sub>	See Note 1
L	H	L	M <sub>i</sub> → Y <sub>i</sub>	
L	H	H	I <sub>i</sub> → Y <sub>i</sub>	

- Notes: 1. For the conditions:  
I<sub>5</sub>, I<sub>4</sub>, I<sub>3</sub>, I<sub>2</sub>, I<sub>1</sub>, I<sub>0</sub> are LOW, Y is an input.  
 $\overline{OE}_Y$  is "Don't Care" for this condition.  
2. X is "Don't Care" condition.

**TABLE 4. CONDITION CODE OUTPUT (CT) INSTRUCTION CODES**

I <sub>3-0</sub> HEX	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>	I <sub>5</sub> = I <sub>4</sub> = 0	I <sub>5</sub> = 0, I <sub>4</sub> = 1	I <sub>5</sub> = 1, I <sub>4</sub> = 0	I <sub>5</sub> = I <sub>4</sub> = 1
0	0	0	0	0	( $\mu N \oplus \mu OVR$ ) + $\mu Z$	( $\mu N \oplus \mu OVR$ ) + $\mu Z$	( $\mu N \oplus \mu OVR$ ) + $\mu Z$	( $\mu N \oplus \mu OVR$ ) + $\mu Z$
1	0	0	0	1	( $\mu N \oplus \mu OVR$ ) + $\mu Z$	( $\mu N \oplus \mu OVR$ ) + $\mu Z$	( $\mu N \oplus \mu OVR$ ) + $\mu Z$	( $\mu N \oplus \mu OVR$ ) + $\mu Z$
2	0	0	1	0	$\mu N \oplus \mu OVR$	$\mu N \oplus \mu OVR$	$\mu N \oplus \mu OVR$	$\mu N \oplus \mu OVR$
3	0	0	1	1	$\mu N \oplus \mu OVR$	$\mu N \oplus \mu OVR$	$\mu N \oplus \mu OVR$	$\mu N \oplus \mu OVR$
4	0	1	0	0	$\mu Z$	$\mu Z$	$\mu Z$	$\mu Z$
5	0	1	0	1	$\mu Z$	$\mu Z$	$\mu Z$	$\mu Z$
6	0	1	1	0	$\mu OVR$	$\mu OVR$	$\mu OVR$	$\mu OVR$
7	0	1	1	1	$\mu OVR$	$\mu OVR$	$\mu OVR$	$\mu OVR$
8	1	0	0	0	$\mu C + \mu Z$	$\mu C + \mu Z$	$\mu C + \mu Z$	$\mu C + \mu Z$
9	1	0	0	1	$\mu C + \mu Z$	$\mu C + \mu Z$	$\mu C + \mu Z$	$\mu C + \mu Z$
A	1	0	1	0	$\mu C$	$\mu C$	$\mu C$	$\mu C$
B	1	0	1	1	$\mu C$	$\mu C$	$\mu C$	$\mu C$
C	1	1	0	0	$\mu C + \mu Z$	$\mu C + \mu Z$	$\mu C + \mu Z$	$\mu C + \mu Z$
D	1	1	0	1	$\mu C + \mu Z$	$\mu C + \mu Z$	$\mu C + \mu Z$	$\mu C + \mu Z$
E	1	1	1	0	$\mu N \oplus \mu N$	$\mu N$	$\mu N$	$\mu N$
F	1	1	1	1	$\mu N \oplus \mu N$	$\mu N$	$\mu N$	$\mu N$

NOTES: 1.  $\oplus$  Represents EXCLUSIVE-OR 2. Correct code as stated.  
 $\odot$  Represents EXCLUSIVE-NOR or coincidence.

**TABLE 5. CRITERIA FOR COMPARING TWO NUMBERS FOLLOWING "A MINUS B" OPERATION**

Relation	For Unsigned Numbers			For 2's Complement Numbers		
	Status	I <sub>3-0</sub>		Status	I <sub>3-0</sub>	
		CT = H	CT = L		CT = H	CT = L
A = B	Z = 1	4	5	Z = 1	4	5
A $\neq$ B	Z = 0	5	4	Z = 0	5	4
A $\geq$ B	C = 1	A	B	N $\odot$ OVR = 1	3	2
A < B	C = 0	B	A	N $\oplus$ OVR = 1	2	3
A > B	C $\cdot$ Z = 1	D	C	(N $\odot$ OVR) $\cdot$ Z = 1	1	0
A $\leq$ B	$\overline{C} + Z = 1$	C	D	(N $\oplus$ OVR) + Z = 1	0	1

$\oplus$  = Exclusive OR H = HIGH Note: For Am2910A the CC input is active LOW, so use I<sub>3-0</sub> code to produce CT = L for the desired test.  
 $\odot$  = Exclusive NOR L = LOW

### Shift Linkage Multiplexer

The five instruction lines I<sub>10</sub>, I<sub>9</sub>, I<sub>8</sub>, I<sub>7</sub>, I<sub>6</sub> control the SHIFT LINKAGE multiplexer. All instructions set up the linkages for both the ALU shifter (RAM shifter on the Am2901C) and the Q register.

UP and DOWN shifts are decided by I<sub>10</sub> which should be connected to I<sub>8</sub> of the Am2903A's instruction lines or I<sub>7</sub> of the Am2901's instruction lines. A wide range of input and output connections are provided, allowing for single or double length shifting or rotating with or without the use of the MSR CARRY or SIGN bits (See Table 6).

In the following discussion of some of the shifts the instruction codes are given as two octal digits AB; A represents I<sub>10</sub>, I<sub>9</sub>, B represents I<sub>8</sub>, I<sub>7</sub>, I<sub>6</sub>.

When adding and down shifting on the same microcycle, (i.e., when doing multiplication or averaging) the shifter input must be the present CARRY, I<sub>C</sub>, rather than the carry resulting from the last cycle (M<sub>C</sub>). Instruction Code 13<sub>8</sub> accomplishes this for unsigned arithmetic. For 2's complement arithmetic, the required shifter input is: I<sub>N</sub>  $\oplus$  I<sub>OVR</sub>. This is provided by Instruction Code 16<sub>8</sub>.

Instruction Codes 14<sub>8</sub>, 15<sub>8</sub>, 17<sub>8</sub> provide the RIGHT ROTATE THROUGH CARRY, ROTATE BRANCH CARRY and ROTATE WITHOUT CARRY functions respectively.

Instruction Codes 34<sub>8</sub>, 35<sub>8</sub>, 37<sub>8</sub> provide the LEFT ROTATE THROUGH CARRY, ROTATE BRANCH CARRY and ROTATE WITHOUT CARRY functions respectively.

The shift outputs are in the high-impedance state unless  $\overline{SE}$  is LOW.

Loading of the M<sub>C</sub> bit by a shift operation overrides any loading or holding of the M<sub>C</sub> bit by MSR Instructions (I<sub>0-5</sub>,  $\overline{CE}_M$  and  $\overline{EC}$ ).

### "CARRY-IN" Control Multiplexer

The two instruction lines I<sub>12</sub>, I<sub>11</sub> control the source of the CARRY output (C<sub>0</sub>).

When I<sub>12</sub> = 0 C<sub>0</sub> = I<sub>11</sub>

When I<sub>12</sub> = 1 and I<sub>11</sub> = 0, the external carry input C<sub>X</sub> is presented to the carry output.

When I<sub>12</sub> = I<sub>11</sub> = 1 the carry output is selected from  $\mu C$ ,  $\overline{\mu C}$ , M<sub>C</sub> or  $\overline{M}_C$  as defined by I<sub>5</sub>, I<sub>3</sub>, I<sub>2</sub>, I<sub>1</sub> (See Table 7).

## APPLICATIONS

### Borrow - Save

One of the capabilities of the Am2900 Family is the complete emulation of other processing machines. One requirement of an emulator is that, when a calculation is being performed, not only must the answer obtained from the Am2900 chips be the same as that from the machine being emulated, but after each machine level instruction, the status bits must be identical.

**TABLE 6. SHIFT LINKAGE MULTIPLEXER INSTRUCTION CODES**

$I_{10}$	$I_9$	$I_8$	$I_7$	$I_6$	$M_C$	RAM	Q	$SIO_0$	$SIO_n$	$QIO_0$	$QIO_n$	Loaded into $M_C$
0	0	0	0	0				Z	0	Z	0	SIO <sub>0</sub>
0	0	0	0	1				Z	1	Z	1	
0	0	0	1	0				Z	0	Z	$M_n$	
0	0	0	1	1				Z	1	Z	$SIO_0$	
0	0	1	0	0				Z	$M_C$	Z	$SIO_0$	
0	0	1	0	1				Z	$M_N$	Z	$SIO_0$	
0	0	1	1	0				Z	0	Z	$SIO_0$	
0	0	1	1	1				Z	0	Z	$SIO_0$	
0	1	0	0	0				Z	$SIO_0$	Z	$QIO_0$	QIO <sub>0</sub>
0	1	0	0	1				Z	$M_C$	Z	$QIO_0$	
0	1	0	1	0				Z	$SIO_0$	Z	$QIO_0$	
0	1	0	1	1				Z	$I_C$	Z	$SIO_0$	
0	1	1	0	0				Z	$M_C$	Z	$SIO_0$	
0	1	1	0	1				Z	$QIO_0$	Z	$SIO_0$	
0	1	1	1	0				Z	$I_N \oplus IOVR$	Z	$SIO_0$	
0	1	1	1	1				Z	$QIO_0$	Z	$SIO_0$	
1	0	0	0	0				0	Z	0	Z	SIO <sub>n</sub>
1	0	0	0	1				1	Z	1	Z	SIO <sub>n</sub>
1	0	0	1	0				0	Z	0	Z	SIO <sub>n</sub>
1	0	0	1	1				1	Z	1	Z	
1	0	1	0	0				$QIO_n$	Z	0	Z	
1	0	1	0	1				$QIO_n$	Z	1	Z	
1	0	1	1	0				$QIO_n$	Z	0	Z	
1	0	1	1	1				$QIO_n$	Z	1	Z	
1	1	0	0	0				$SIO_n$	Z	$QIO_n$	Z	
1	1	0	0	1				$M_C$	Z	$QIO_n$	Z	
1	1	0	1	0				$SIO_n$	Z	$QIO_n$	Z	SIO <sub>n</sub>
1	1	0	1	1				$M_C$	Z	0	Z	
1	1	1	0	0				$QIO_n$	Z	$M_C$	Z	
1	1	1	0	1				$QIO_n$	Z	$SIO_n$	Z	
1	1	1	1	0				$QIO_n$	Z	$M_C$	Z	
1	1	1	1	1				$QIO_n$	Z	$SIO_n$	Z	

Notes: 1. Z = High-impedance (outputs off) state.  
2. Outputs enabled and  $M_C$  loaded only if  $\overline{SE}$  is LOW.  
3. Loading of  $M_C$  from  $I_{10-6}$  overrides control from  $I_{5-0}, \overline{CE}_M, \overline{E}_C$ .

**TABLE 7. CARRY-IN CONTROL MULTIPLEXER INSTRUCTION CODES**

I <sub>12</sub>	I <sub>11</sub>	I <sub>5</sub>	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	C <sub>0</sub>
0	0	X	X	X	X	0
0	1	X	X	X	X	1
1	0	X	X	X	X	C <sub>X</sub>
1	1	0	0	X	X	$\mu C$
1	1	0	X	1	X	$\mu C$
1	1	0	X	X	1	$\mu C$
1	1	0	1	0	0	$\overline{\mu C}$
1	1	1	0	X	X	M <sub>C</sub>
1	1	1	X	1	X	M <sub>C</sub>
1	1	1	X	X	1	M <sub>C</sub>
1	1	1	1	0	0	$\overline{M_C}$

There are alternative methods for subtracting in a digital machine and the state of the CARRY after the calculation depends on the method. For instance, the subtraction of 0100 from 1010 by the 2's complement add method generates a result of 0110 with a CARRY. Direct subtraction however, yields an answer of 0110 with no BORROW.

Many machines store the state of the CARRY for subtract operations, and this is the recommended method for maximum effective use of the Am2904, but, to allow those machines which store the BORROW to be efficiently emulated, the Am2904 has allocated special instructions. Using these codes causes the CARRY bit to be inverted before storage in the status registers and also re-inverts these status bits before using them as carry inputs. These codes are 10<sub>8</sub>, 11<sub>8</sub>, 30<sub>8</sub>, 31<sub>8</sub>, 50<sub>8</sub>, 51<sub>8</sub>, 70<sub>8</sub>, 71<sub>8</sub> (I<sub>5-0</sub>).

Notice that when these codes are used to load the inverted CARRY to either of the status registers, the CT output selected by the Condition Code Multiplexer assumes the CARRY is inverted and still defines whether  $A > B$  or  $A \leq B$  (See Table 4).

Similarly, when doing a compare on a machine which saves the BORROW, testing for  $A > B$ ,  $A \leq B$  forces the complement of the CARRY to be stored in the status registers (See Tables 1 and 2).

### Normalizing

Normalizing is the process of stripping off all leading sign bits until the two most significant bits are complementary. The Am2904 facilitates both single and double length normaliza-

tion in the Am2901 and the Am2903A. When using the NORMALIZE special instructions with the Am2903A, the EXCLUSIVE-OR of the most significant two bits is generated at the C<sub>n+4</sub> pin of the most significant Am2903A. The EXCLUSIVE-OR of the two bits next to the most significant bit is also generated at the OVR pin. The procedure for normalizing then is to loop on the normalize instruction with a branch condition on the C<sub>n+4</sub> state or the OVR state, depending on the architecture employed. The C<sub>n+4</sub> or OVR output is routed to the Am2910 CC input through the Am2904 Condition Code multiplexer. As the contents of the status registers always refer to the last cycle, not the present one, the last operation in normalizing is to downshift, bringing the sign bit (M<sub>N</sub>) back into the most significant bit position. This is achieved using the shift operations 05<sub>8</sub> (I<sub>10-6</sub>) for double length normalizing, and 02<sub>8</sub> for single length normalizing. For more details regarding normalizing with the Am2903A see the Am2903A data sheet.

The Am2901 does not have the EXCLUSIVE-OR gates to help with normalizing, so the Am2904 includes in the Condition Code multiplexer the EXCLUSIVE-OR and EXCLUSIVE-NOR functions of M<sub>N</sub> (the sign bit resulting from the last operation) and I<sub>N</sub> (the sign bit resulting from the present operation). Instruction codes 16<sub>8</sub> and 17<sub>8</sub> (I<sub>5-0</sub>) form the EXCLUSIVE-OR and EXCLUSIVE-NOR functions of M<sub>N</sub> and I<sub>N</sub>.

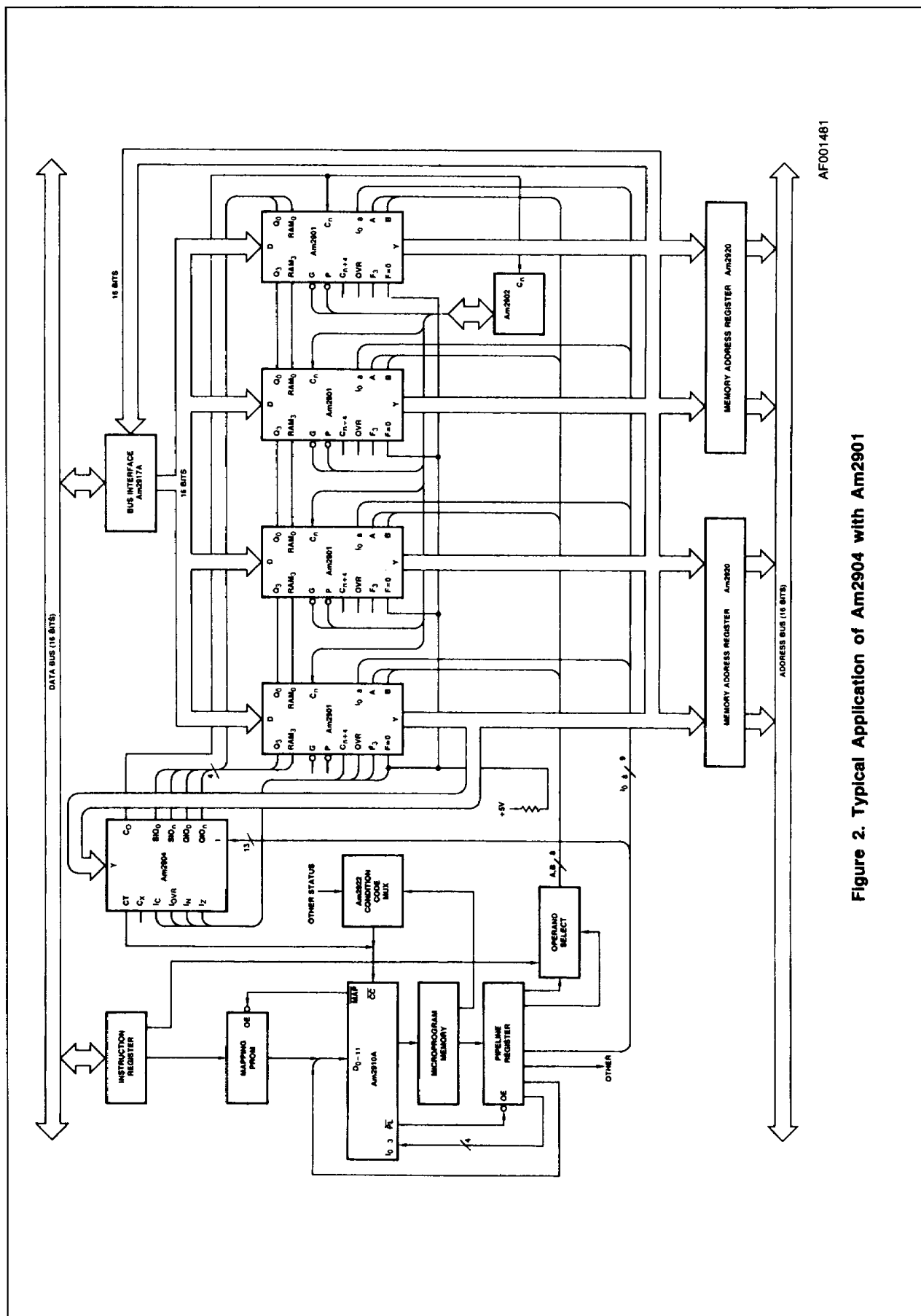
### Interrupts

Some machines allow interrupts only at the machine instruction level while others allow them at the microinstruction level. The Am2904 is designed to handle both cases.

When the machine is interrupted, it is necessary to store the contents of either the MSR (machine instruction level interrupts) or both the status registers (micro instruction level interrupts) into an external store. This transfer is intended to take place over the Y input/output pins (See Table 3).

After the interrupt has been serviced the registers must be restored to their pre-interrupt state. This is accomplished by two operations of instruction 00<sub>8</sub> (I<sub>5-0</sub>) which loads the MSR from the Y inputs while loading the  $\mu$ SR from the MSR. Thus, the pre-interrupt contents of the  $\mu$ SR are first loaded to the MSR (first instruction 00<sub>8</sub>), then this data is transferred to the  $\mu$ SR while the MSR is restored to its pre-interrupt state (second instruction 00<sub>8</sub>).

In controllers and some other microprogrammed machines the applications program itself is often in the microprogram memory; that is, there is no macroinstruction set. These machines require only a microstatus register since there is no separate machine status. The MSR in the Am2904 can be used as a one-level stack on the microstatus register. When an interrupt occurs, the  $\mu$ SR and the MSR are simply swapped (I<sub>5-0</sub> = 02<sub>8</sub>).



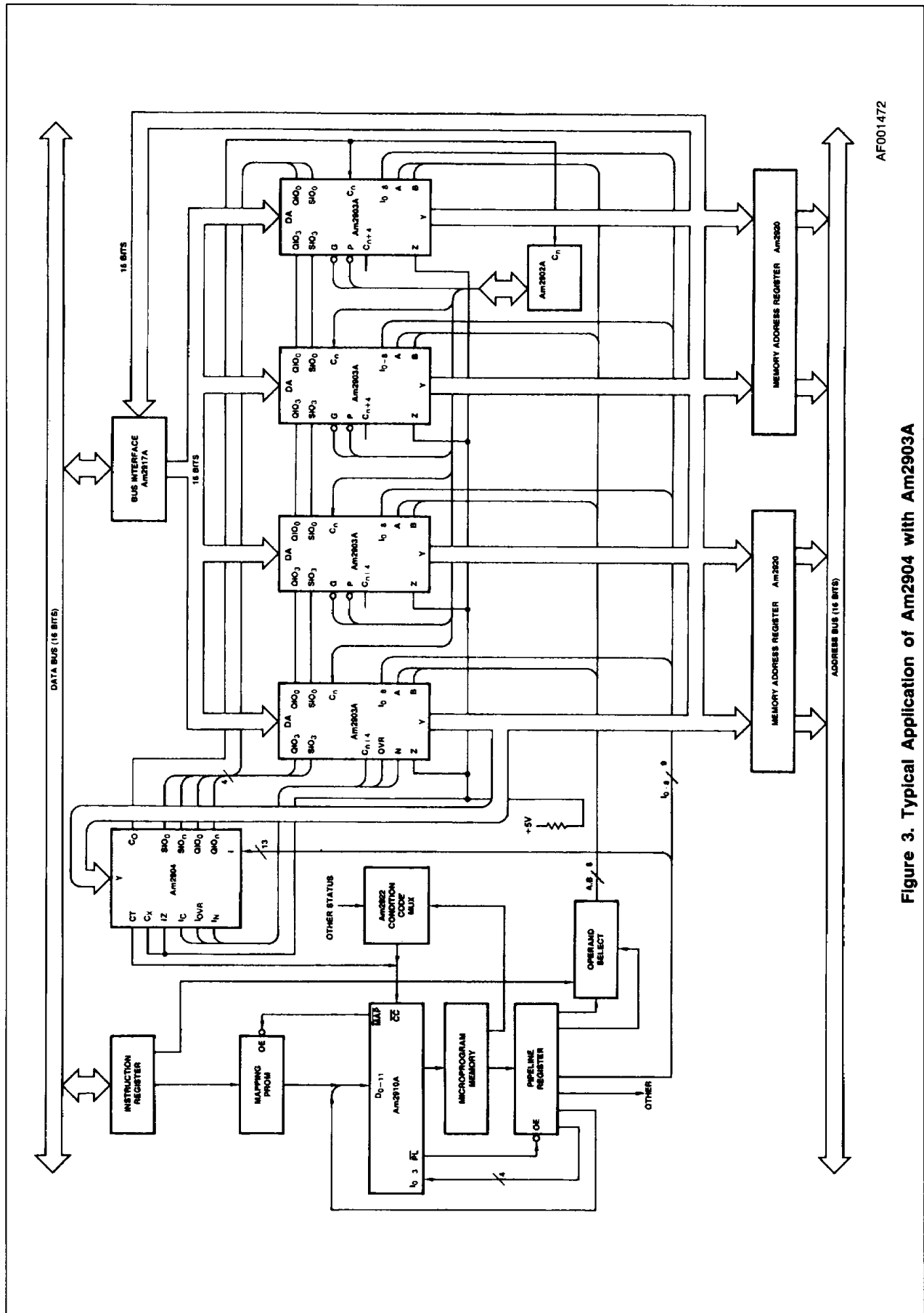


Figure 3. Typical Application of Am2904 with Am2903A

AF001472



## ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... -65 to +150°C  
 (Case) Temperature Under Bias ..... -55 to +125°C  
 Supply Voltage to Ground Potential  
     Continuous ..... -0.5 V to +7.0 V  
 DC Voltage Applied to Outputs for  
     High Output State ..... -0.5 V to +V<sub>CC</sub> Max.  
 DC Input Voltage ..... -0.5 V to +5.5 V  
 DC Output Current, Into Outputs ..... 30 mA  
 DC Input Current ..... -30 mA to +5.0 mA

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

## OPERATING RANGES

Commercial (C) Devices

Temperature (T<sub>A</sub>) ..... 0 to +70°C  
 Supply Voltage (V<sub>CC</sub>) ..... +4.75 V to +5.25 V

Military (M) Devices

Temperature (T<sub>C</sub>) ..... -55 to +125°C  
 Supply Voltage (V<sub>CC</sub>) ..... +4.5 V to +5.5 V

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

**DC CHARACTERISTICS** over operating range unless otherwise specified; Included in Group A, Subgroup 1, 2, 3 tests unless otherwise noted for APL products.

Parameter Symbol	Parameter Description	Test Conditions (Note 2)		Min.	Typ. (Note 1)	Max.	Units	
V <sub>OH</sub>	Output HIGH Voltage	V <sub>CC</sub> = Min., V <sub>IN</sub> = V <sub>IH</sub> or V <sub>IL</sub>	I <sub>OH</sub> = -1.6 mA Y <sub>Z</sub> , Y <sub>C</sub> , Y <sub>N</sub> , Y <sub>OV</sub> R	2.4			Volts	
			I <sub>OH</sub> = -0.8 mA SIO <sub>0</sub> , SIO <sub>n</sub> , QIO <sub>0</sub> QIO <sub>n</sub> , CT, CO	2.4			Volts	
V <sub>OL</sub>	Output LOW Voltage	V <sub>CC</sub> = Min., V <sub>IN</sub> = V <sub>IH</sub> or V <sub>IL</sub>	Y <sub>Z</sub> , Y <sub>C</sub>	I <sub>OL</sub> = 24 mA (COM'L)			0.5	Volts
			Y <sub>N</sub> , Y <sub>OV</sub> R	I <sub>OL</sub> = 16 mA (MIL)			0.5	
			SIO <sub>0</sub> , QIO <sub>0</sub> , CT, SIO <sub>n</sub> , QIO <sub>n</sub> , CO	I <sub>OL</sub> = 8mA			0.5	
V <sub>IH</sub>	Input HIGH Voltage	Guaranteed Input Logical HIGH Voltage for all inputs (Note 6)		2.0			Volts	
V <sub>IL</sub>	Input LOW Voltage	Guaranteed Input Logical LOW Voltage for all inputs (Note 6)				0.8	Volts	
V <sub>I</sub>	Input Clamp Voltage	V <sub>CC</sub> = Min., I <sub>IN</sub> = -18 mA				-1.5	Volts	
I <sub>IL</sub>	Input LOW Current	V <sub>CC</sub> = Max., V <sub>IN</sub> = 0.5 V	CP				-0.7	mA
			CE <sub>M</sub> , CE <sub>μ</sub>				-1.8	
			I <sub>Z</sub> , I <sub>C</sub> , I <sub>N</sub> , I <sub>OV</sub> R				-1.2	
			I <sub>0</sub> -I <sub>12</sub> , E <sub>Z</sub> , E <sub>C</sub> , E <sub>N</sub> E <sub>OV</sub> R, OE <sub>Y</sub> , OE <sub>CT</sub> , C <sub>X</sub> , Y <sub>Z</sub> , Y <sub>C</sub> , Y <sub>N</sub> , Y <sub>OV</sub> R				-0.45	
			SE, SIO <sub>0</sub> , SIO <sub>n</sub> QIO <sub>0</sub> , QIO <sub>n</sub>				-1.35	
I <sub>IH</sub>	Input HIGH Current	V <sub>CC</sub> = Max., V <sub>IN</sub> = 2.7 V	CP, I <sub>0</sub> -I <sub>12</sub> , E <sub>Z</sub> , E <sub>C</sub> , E <sub>N</sub> , E <sub>OV</sub> R, OE <sub>Y</sub> , OE <sub>CT</sub> , C <sub>X</sub>				20	μA
			CE <sub>M</sub> , CE <sub>μ</sub>				80	
			I <sub>Z</sub> , I <sub>C</sub> , I <sub>N</sub> , I <sub>OV</sub> R, SE				60	
			SIO <sub>0</sub> , SIO <sub>n</sub> , QIO <sub>0</sub> , QIO <sub>n</sub>				110	
			Y <sub>Z</sub> , Y <sub>C</sub> , Y <sub>N</sub> , Y <sub>OV</sub> R				70	
I <sub>I</sub>	Input HIGH Current	V <sub>CC</sub> = Max., V <sub>IN</sub> = 5.5 V				1.0	mA	
I <sub>OZH</sub> I <sub>OZL</sub>	Off State (High Impedance) Output Current	V <sub>CC</sub> = Max.	CT	V <sub>O</sub> = 2.4			50	μA
				V <sub>O</sub> = 0.5			-50	
			SIO <sub>0</sub> , SIO <sub>n</sub> , QIO <sub>0</sub> , QIO <sub>n</sub> (Note 4)	V <sub>O</sub> = 2.4			110	
				V <sub>O</sub> = 0.5			-1350	
			Y <sub>Z</sub> , Y <sub>C</sub> , Y <sub>N</sub> , Y <sub>OV</sub> R (Note 4)	V <sub>O</sub> = 2.4			70	
				V <sub>O</sub> = 0.5			-450	
I <sub>OS</sub>	Output Short-Circuit Current (Note 3)	V <sub>CC</sub> = 5.75 V, V <sub>O</sub> = 0.5 V		-30		-85	mA	
I <sub>CC</sub>	Power Supply Current (Note 5)	V <sub>CC</sub> = Max.	COM'L	T <sub>A</sub> = 0°C to +70°C			318	mA
				T <sub>A</sub> = +70°C			262	
			MIL	T <sub>C</sub> = -55°C to +125°C			348	
				T <sub>C</sub> = +125°C			222	

Notes: 1. Typical limits are at V<sub>CC</sub> = 5.0 V, 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 are three-state outputs internally connected to TTL inputs. Input Characteristics are measured with output enables HIGH.

5. Worst-case I<sub>CC</sub> is at minimum temperature.

6. These input levels provide zero noise immunity and should only be static tested in a noise-free environment (not functionally tested.)

## SWITCHING CHARACTERISTICS

The following tables define the Am2904 switching characteristics. Tables A are setup and hold times relative to the clock LOW-to-HIGH transition; Tables B are combinational delays; Tables C are clock requirements. All measurements are made at 1.5 V with input levels at 0 V or 3 V. All values are in ns. All outputs have maximum DC loading.

### GUARANTEED CHARACTERISTICS OVER COMMERCIAL OPERATING RANGE

#### A. Setup and Hold Times (ns)

Input	$t_s$	$t_h$
$I_Z, I_N, I_{OVR}$	14	5
$I_C$ ( $I_1 I_2 I_3 = 001$ )	27	5
$I_C$ ( $I_1 I_2 I_3 \neq 001$ )	14	5
$\overline{OE}_\mu$	18	3
$\overline{OE}_M$	23	3
$\overline{E}_Z, \overline{E}_C, \overline{E}_N, \overline{E}_{OVR}$	22	3
$I_0 - I_5$	41	1
$I_6 - I_{10}$	40	1
$\overline{SE}$	36	0
$Y_Z, Y_C, Y_N, Y_{OVR}$ ( $I_{0-5} = \text{LOW}$ )	15	5
$SIO_0, SIO_n,$ $QIO_0, QIO_n$	20	5

#### B. Combinational Delays (ns)

From (Input)	To (Output)	$t_{pd}$
$I_Z, I_C, I_N, I_{OVR}$	$Y_Z, Y_C, Y_N, Y_{OVR}$	38
CP	$Y_Z, Y_C, Y_N, Y_{OVR}$	41
$I_4, I_5$	$Y_Z, Y_C, Y_N, Y_{OVR}$	35
$I_Z, I_C, I_N, I_{OVR}$	CT	33
CP	CT	36
$I_0 - I_5$	CT	33
$C_X$	$C_O$	20
CP	$C_O$	27
$I_1, 2, 3, 5, 11, 12$	$C_O$	39
$SIO_n, QIO_n$	$SIO_0$	19
$SIO_0, QIO_0$	$SIO_n$	19
$I_C, I_N, I_{OVR}$	$SIO_n$	26
$SIO_n, QIO_n$	$QIO_0$	19
$SIO_0, QIO_0$	$QIO_n$	19
CP	$SIO_0, SIO_n, QIO_0,$ $QIO_n$	30
$I_6 - I_{10}$	$SIO_0, SIO_n, QIO_0,$ $QIO_n$	26

#### C. Clock Requirements (ns)

Minimum Clock LOW Time	20
Minimum Clock HIGH Time	20

#### D. Enable/Disable Times (ns)

$C_L = 5.0$  pF for output disable tests measured to 0.5 V change of output voltage level.

From (Input)	To (Output)	Enable	Disable
$\overline{OE}_{CT}$	CT	23	18
$\overline{SE}$	$SIO_0, SIO_n,$ $QIO_0, QIO_n$	30	16
$I_{10}$	$SIO_0, SIO_n,$ $QIO_0, QIO_n$	39	29
$\overline{OE}_Y$	$Y_Z, Y_C, Y_N, Y_{OVR}$	26	21
$I_0 - I_5$	$Y_Z, Y_C, Y_N, Y_{OVR}$	28	40

## GUARANTEED CHARACTERISTICS OVER MILITARY OPERATING RANGE

(Included in Group A, Subgroup 9, 10, 11 tests unless otherwise noted)

### A. Setup and Hold Times (ns)

Input	$t_s$	$t_h$
$I_Z, I_N, I_{OVR}$	15	5
$I_C$ ( $I_1 I_2 I_3 = 001$ )	28	5
$I_C$ ( $I_1 I_2 I_3 \neq 001$ )	15	5
$\overline{OE}_\mu$	20	3
$\overline{OE}_M$	23	4
$\overline{E}_Z, \overline{E}_C, \overline{E}_N, \overline{E}_{OVR}$	23	4
$I_0 - I_5$	48	2
$I_6 - I_{10}$	44	2
$\overline{SE}$	40	0
$Y_Z, Y_C, Y_N, Y_{OVR}$ ( $I_{0-5} = \text{LOW}$ )	16	6
$SIO_0, SIO_n,$ $QIO_0, QIO_n$	20	5

### B. Combinational Delays (ns)

From (Input)	To (Output)	$t_{pd}$
$I_Z, I_C, I_N, I_{OVR}$	$Y_Z, Y_C, Y_N, Y_{OVR}$	40
CP	$Y_Z, Y_C, Y_N, Y_{OVR}$	45
$I_4, I_5$	$Y_Z, Y_C, Y_N, Y_{OVR}$	38
$I_Z, I_C, I_N, I_{OVR}$	CT	44
CP	CT	40
$I_0 - I_5$	CT	41
$C_X$	$C_O$	22
CP	$C_O$	28
$I_1, 2, 3, 5, 11, 12$	$C_O$	42
$SIO_n, QIO_n$	$SIO_0$	20
$SIO_0, QIO_0$	$SIO_n$	20
$I_C, I_N, I_{OVR}$	$SIO_n$	29
$SIO_n, QIO_n$	$QIO_0$	20
$SIO_0, QIO_0$	$QIO_n$	20
CP	$SIO_0, SIO_n, QIO_0,$ $QIO_n$	32
$I_6 - I_{10}$	$SIO_0, SIO_n, QIO_0,$ $QIO_n$	31

### C. Clock Requirements (ns)

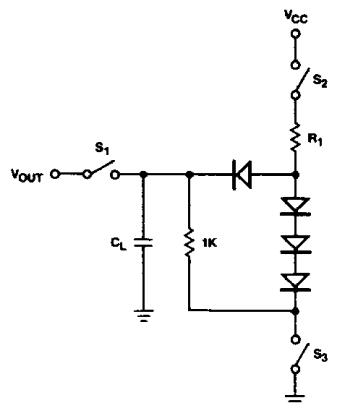
Minimum Clock LOW Time	25
Minimum Clock HIGH Time	25

### D. Enable/Disable Times (ns)

$C_L = 5.0$  pF for output disable tests measured to 0.5 V change of output voltage level.

From (Input)	To (Output)	Enable	Disable
$\overline{OE}_{CT}$	CT	25	18
$\overline{SE}$	$SIO_0, SIO_n,$ $QIO_0, QIO_n$	35	20
$I_{10}$	$SIO_0, SIO_n,$ $QIO_0, QIO_n$	43	32
$\overline{OE}_Y$	$Y_Z, Y_C, Y_N, Y_{OVR}$	28	23
$I_0 - I_5$	$Y_Z, Y_C, Y_N, Y_{OVR}$	30	41

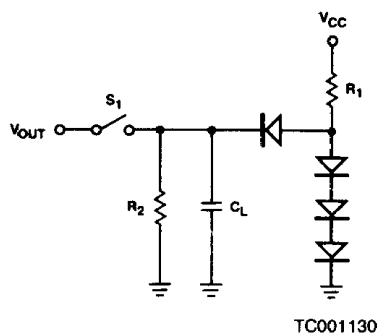
## SWITCHING TEST CIRCUITS



TC001120

### A. Three-State Outputs

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



TC001130

### B. Normal Outputs

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

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

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

## TEST OUTPUT LOADS FOR Am2904

Pin # (DIP)	Pin Label	Test Circuit	$R_1$	$R_2$
25	$C_0$	B	470	3K
27	CT	A	430	1K
28	$Y_{OVR}$	A	220	1K
29	$Y_N$	A	220	1K
31	$Y_C$	A	220	1K
32	$Y_Z$	A	220	1K
33	$QIO_n$	A	430	1K
34	$QIO_0$	A	430	1K
35	$SIO_n$	A	430	1K
36	$SIO_0$	A	430	1K

## 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 as 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 400 mA in 5-8 ns. Inductance in the ground

cable may allow the ground pin at the device to rise by 100s of millivolts momentarily.

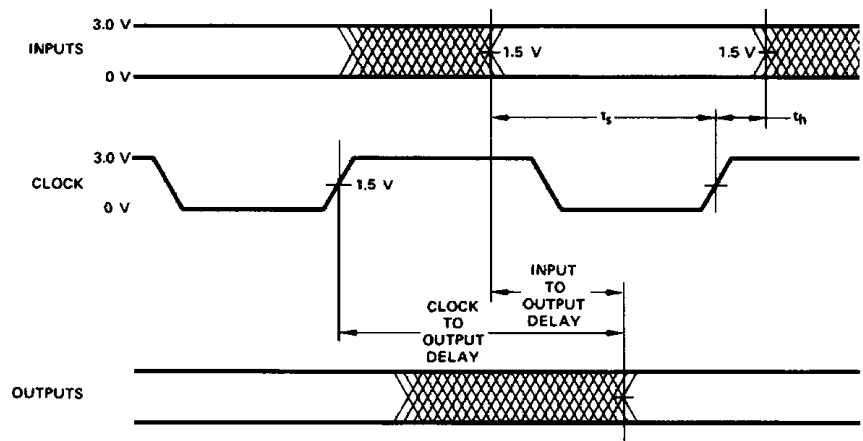
4. Use extreme care in defining input levels for AC tests. Many inputs may be changed at once, so there will be significant noise at the device pins and they may not actually reach  $V_{IL}$  or  $V_{IH}$  until the noise has settled. AMD recommends using  $V_{IL} \leq 0$  V and  $V_{IH} \geq 3.0$  V 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. To assist in testing, AMD offers complete documentation on our test procedures and, in most cases, can provide Fairchild Sentry programs, under license.

# SWITCHING WAVEFORMS

## KEY TO SWITCHING WAVEFORMS

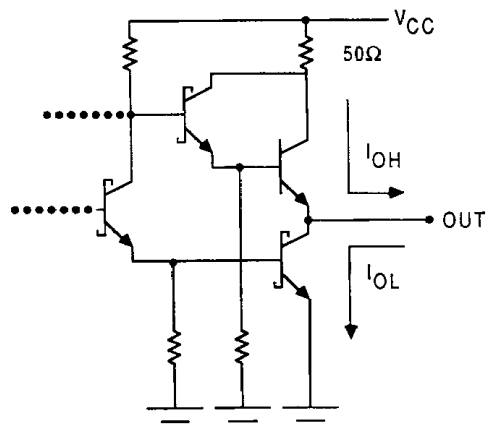
WAVEFORM	INPUTS	OUTPUTS
	MUST BE STEADY	WILL BE STEADY
	MAY CHANGE FROM H TO L	WILL BE CHANGING FROM H TO L
	MAY CHANGE FROM L TO H	WILL BE CHANGING FROM L TO H
	DON'T CARE: ANY CHANGE PERMITTED	CHANGING: STATE UNKNOWN
	DOES NOT APPLY	CENTER LINE IS HIGH IMPEDANCE "OFF" STATE

KS000010



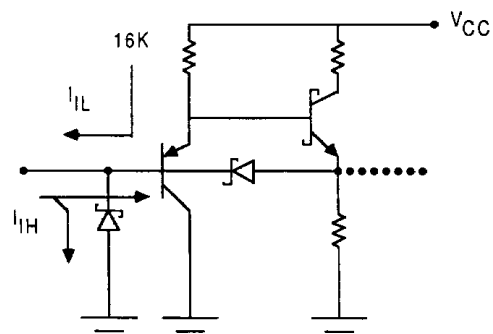
WFR02990

## INPUT/OUTPUT CIRCUIT DIAGRAM



IC000940

### A. Driving Output

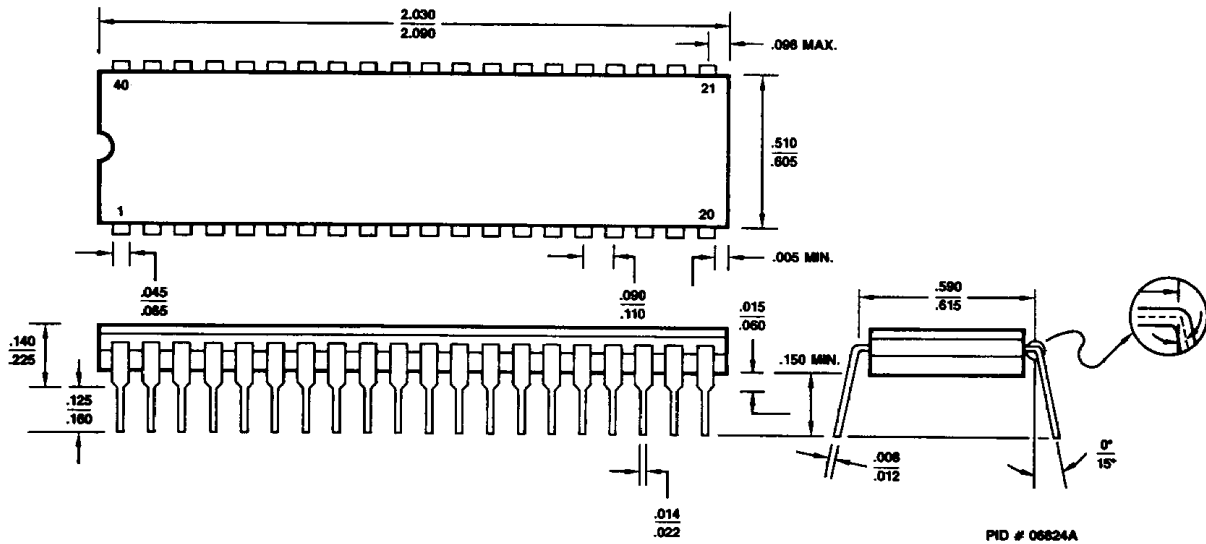


IC000950

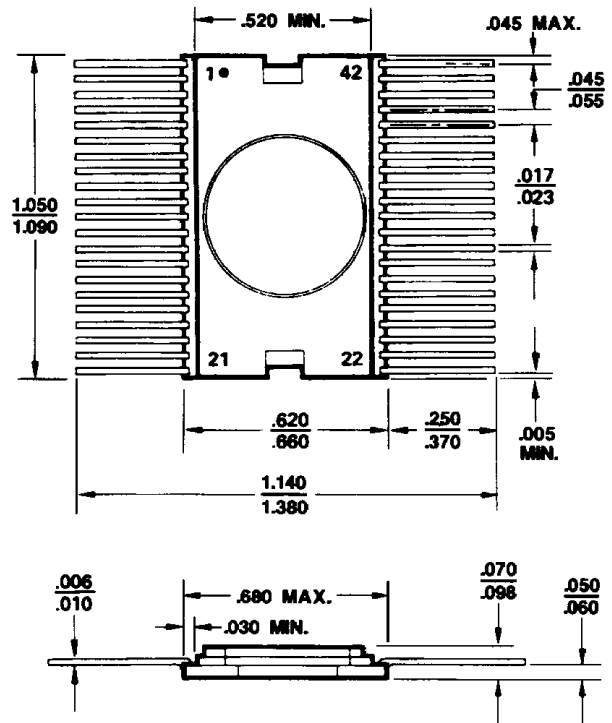
### B. Driven Input

# PHYSICAL DIMENSIONS\*

## CD 040



## CFT042



\*For reference only.