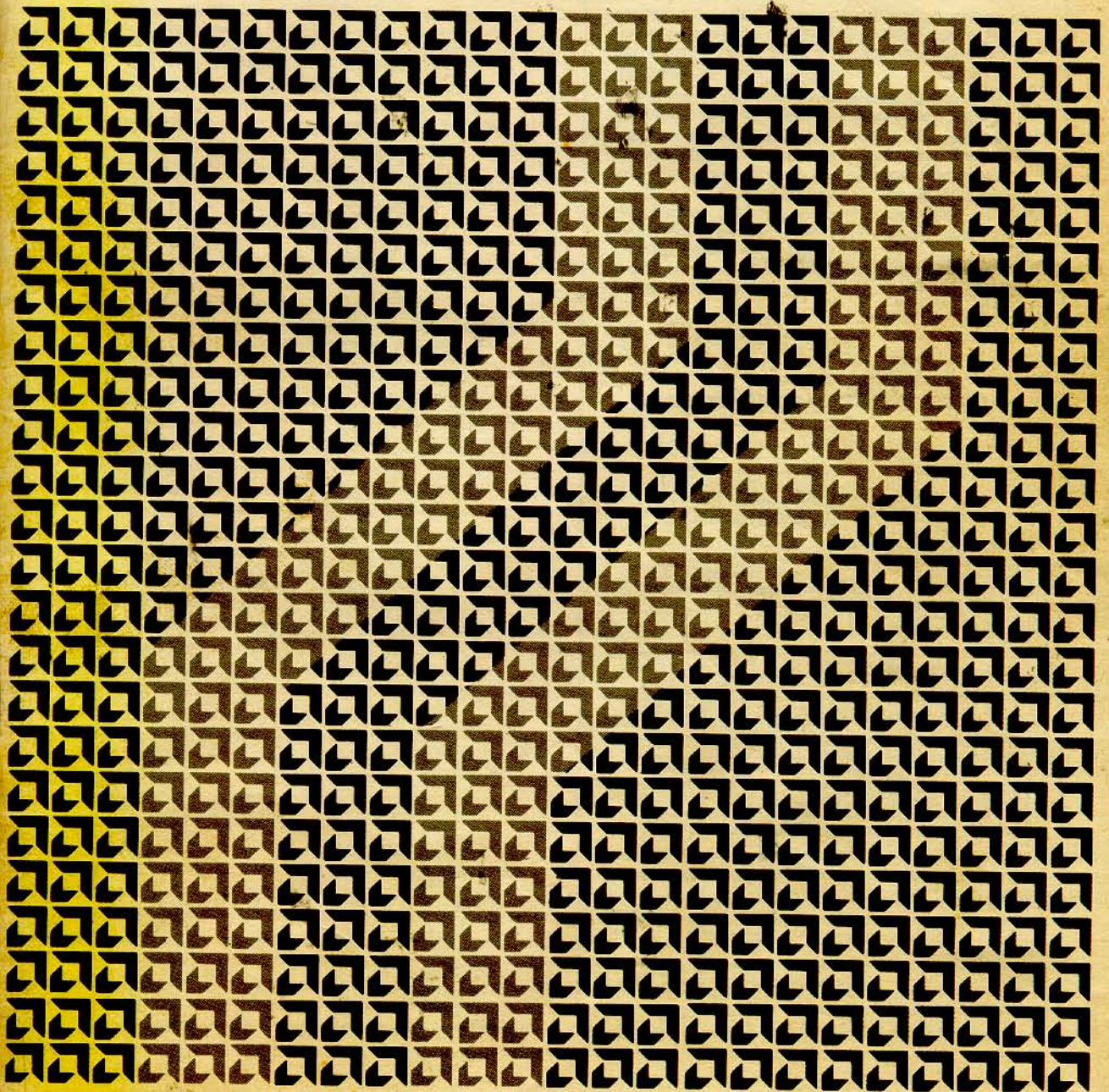


March 1976

A Baldwin



Am2901, Am2909 Technical Data

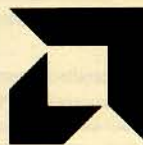


Am2901

Four-Bit Bipolar Microprocessor Slice

Advanced Micro Devices

Bipolar Microprocessor Circuits



DISTINCTIVE CHARACTERISTICS

- Two-address architecture – Independent simultaneous access to two working registers saves machine cycles.
- Eight-function ALU – Performs addition, two subtraction operations, and five logic functions on two source operands.
- Flexible data source selection – ALU data is selected from five source ports for a total of 203 source operand pairs for every ALU function.
- Left/right shift independent of ALU – Add and shift operations take only one cycle.
- Four status flags – Carry, overflow, zero, and negative.
- Expandable – Connect any number of Am2901's together for longer word lengths.
- Microprogrammable – Three groups of three bits each for source operand, ALU function, and destination control.

GENERAL DESCRIPTION

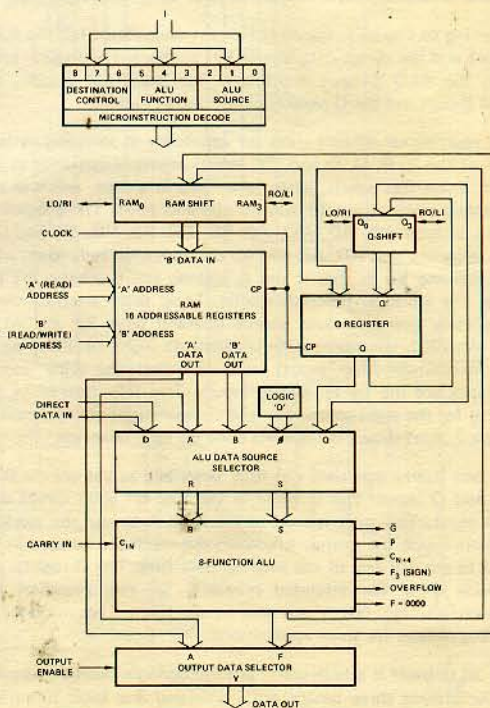
The four-bit bipolar microprocessor slice is designed as a high-speed cascadable element intended for use in CPU's, peripheral controllers, programmable microprocessors and numerous other applications. The microinstruction flexibility of the Am2901 will allow efficient emulation of almost any digital computing machine.

The device, as shown in the block diagram below, consists of a 16-word by 4-bit two-port RAM, a high-speed ALU, and the associated shifting, decoding and multiplexing circuitry. The nine-bit microinstruction word is organized into three groups of three bits each and selects the ALU source operands, the ALU function, and the ALU destination register. The microprocessor is cascadable with full look-ahead or with ripple carry, has three-state outputs, and provides various status flag outputs from the ALU. Advanced low-power Schottky processing is used to fabricate this 40-lead LSI chip.

TABLE OF CONTENTS

Block Diagram	4
Function Tables	5
Package Outline	7
Connection Diagram	8
Pin Definitions	8
Screening	9
Order Codes	9
DC Characteristics	10
AC Characteristics	11
Switching Waveforms	12
Applications	13

MICROPROCESSOR SLICE BLOCK DIAGRAM



ARCHITECTURE

A detailed block diagram of the bipolar microprogrammable microprocessor structure is shown in Figure 1. The circuit is a four-bit slice cascadable to any number of bits. Therefore, all data paths within the circuit are four bits wide. The two key elements in the Figure 1 block diagram are the 16-word by 4-bit 2-port RAM and the high-speed ALU.

Data in any of the 16 words of the Random Access Memory (RAM) can be read from the A-port of the RAM as controlled by the 4-bit A address field input. Likewise, data in any of the 16 words of the RAM as defined by the B address field input can be simultaneously read from the B-port of the RAM. The same code can be applied to the A select field and B select field in which case the identical file data will appear at both the RAM A-port and B-port outputs simultaneously.

When enabled by the RAM write enable (RAM EN), new data is always written into the file (word) defined by the B address field of the RAM. The RAM data input field is driven by a 3-input multiplexer. This configuration is used to shift the ALU output data (F) if desired. This three-input multiplexer scheme allows the data to be shifted up (right) one bit position, shifted down (left) one bit position, or not shifted in either direction.

The RAM A-port data outputs and RAM B-port data outputs drive separate 4-bit latches. These latches hold the RAM data while the clock input is LOW. This eliminates any possible race conditions that could occur while new data is being written into the RAM.

The high-speed Arithmetic Logic Unit (ALU) can perform three binary arithmetic and five logic operations on the two 4-bit input words R and S. The R input field is driven from a 2-input multiplexer, while the S input field is driven from a 3-input multiplexer. Both multiplexers also have an inhibit capability; that is, no data is passed. This is equivalent to a "zero" source operand.

Referring to Figure 1, the ALU R-input multiplexer has the RAM A-port and the direct data inputs (D) connected as inputs. Likewise, the ALU S-input multiplexer has the RAM A-port, the RAM B-port and the Q register connected as inputs.

This multiplexer scheme gives the capability of selecting various pairs of the A, B, D, Q and "0" inputs as source operands to the ALU. These five inputs, when taken two at a time, result in ten possible combinations of source operand pairs. These combinations include AB, AD, AQ, A0, BD, BQ, B0, DQ, D0 and Q0. It is apparent that AD, AQ and A0 are somewhat redundant with BD, BQ and B0 in that if the A address and B address are the same, the identical function results. Thus, there are only seven completely non-redundant source operand pairs for the ALU. The Am2901 microprocessor implements eight of these pairs. The microinstruction inputs used to select the ALU source operands are the I₀, I₁, and I₂ inputs. The definition of I₀, I₁, and I₂ for the eight source operand combinations are as shown in Figure 2. Also shown is the octal code for each selection.

The two source operands not fully described as yet are the D input and Q input. The D input is the four-bit wide direct data field input. This port is used to insert all data into the working registers inside the device. Likewise, this input can be used in the ALU to modify any of the internal data files. The Q register is a separate 4-bit file intended primarily for multiplication and division routines but it can also be used as an accumulator or holding register for some applications.

The ALU itself is a high-speed arithmetic/logic operator capable of performing three binary arithmetic and five logic functions. The I₃, I₄, and I₅ microinstruction inputs are used to select the

ALU function. The definition of these inputs is shown in Figure 3. The octal code is also shown for reference. The normal technique for cascading the ALU of several devices is in a look-ahead carry mode. Carry generate, \bar{G} , and carry propagate, \bar{P} , are outputs of the device for use with a carry-look-ahead-generator such as the Am2902 ('182). A carry-out, C_{n+4} , is also generated and is available as an output for use as the carry flag in a status register. Both carry-in (C_n) and carry-out (C_{n+4}) are active HIGH.

The ALU has three other status-oriented outputs. These are F₃, F = 0, and overflow (OVR). The F₃ output is the most significant (sign) bit of the ALU and can be used to determine positive or negative results without enabling the three-state data outputs. F₃ is non-inverted with respect to the sign bit output Y₃. The F = 0 output is used for zero detect. It is an open-collector output and can be wire OR'ed between microprocessor slices. F = 0 is HIGH when all F outputs are LOW. The overflow output (OVR) is used to flag arithmetic operations that exceed the available two's complement number range. The overflow output (OVR) is HIGH when overflow exists. That is, when C_{n+3} and C_{n+4} are not the same polarity.

The ALU data output is routed to several destinations. It can be a data output of the device and it can also be stored in the RAM or the Q register. Eight possible combinations of ALU destination functions are available as defined by the I₆, I₇, and I₈ microinstruction inputs. These combinations are shown in Figure 4.

The four-bit data output field (Y) features three-state outputs and can be directly bus organized. An output control (\bar{OE}) is used to enable the three-state outputs. When \bar{OE} is HIGH, the Y outputs are in the high-impedance state.

A two-input multiplexer is also used at the data output such that either the A-port of the RAM or the ALU outputs (F) are selected at the device Y outputs. This selection is controlled by the I₆, I₇, and I₈ microinstruction inputs. Refer to Figure 4 for the selected output for each microinstruction code combination.

As was discussed previously, the RAM inputs are driven from a three-input multiplexer. This allows the ALU outputs to be entered non-shifted, shifted up one position (X2) or shifted down one position ($\div 2$). The shifter has two ports; one is labeled RAM₀-LO/RI and the other is labeled RAM₃-RO/LI. Both of these ports consist of a buffer-driver with a three-state output and an input to the multiplexer. Thus, in the shift up mode, the RO buffer is enabled and the RI multiplexer input is enabled. Likewise, in the shift down mode, the LO buffer and LI input are enabled. In the no-shift mode, both the LO and RO buffers are in the high-impedance state and the multiplexer inputs are not selected. This shifter is controlled from the I₆, I₇, and I₈ microinstruction inputs as defined in Figure 4.

Similarly, the Q register is driven from a 3-input multiplexer. In the no-shift mode, the multiplexer enters the ALU data into the Q register. In either the shift-up or shift-down mode, the multiplexer selects the Q register data appropriately shifted up or down. The Q shifter also has two ports; one is labeled Q₀-LO/RI and the other is Q₃-RO/LI. The operation of these two ports is similar to the RAM shifter and is also controlled from I₆, I₇, and I₈ as shown in Figure 4.

The clock input to the Am2901 controls the RAM, the Q register, and the A and B data latches. When enabled, data is clocked into the Q register on the LOW-to-HIGH transition of the clock. When the clock input is HIGH, the A and B latches are open and will pass whatever data is present at the RAM outputs. When the clock input is LOW, the latches are closed and will retain the last data entered. If the RAM-EN is enabled, new data will be written into the RAM file (word) defined by the B address field when the clock input is LOW.

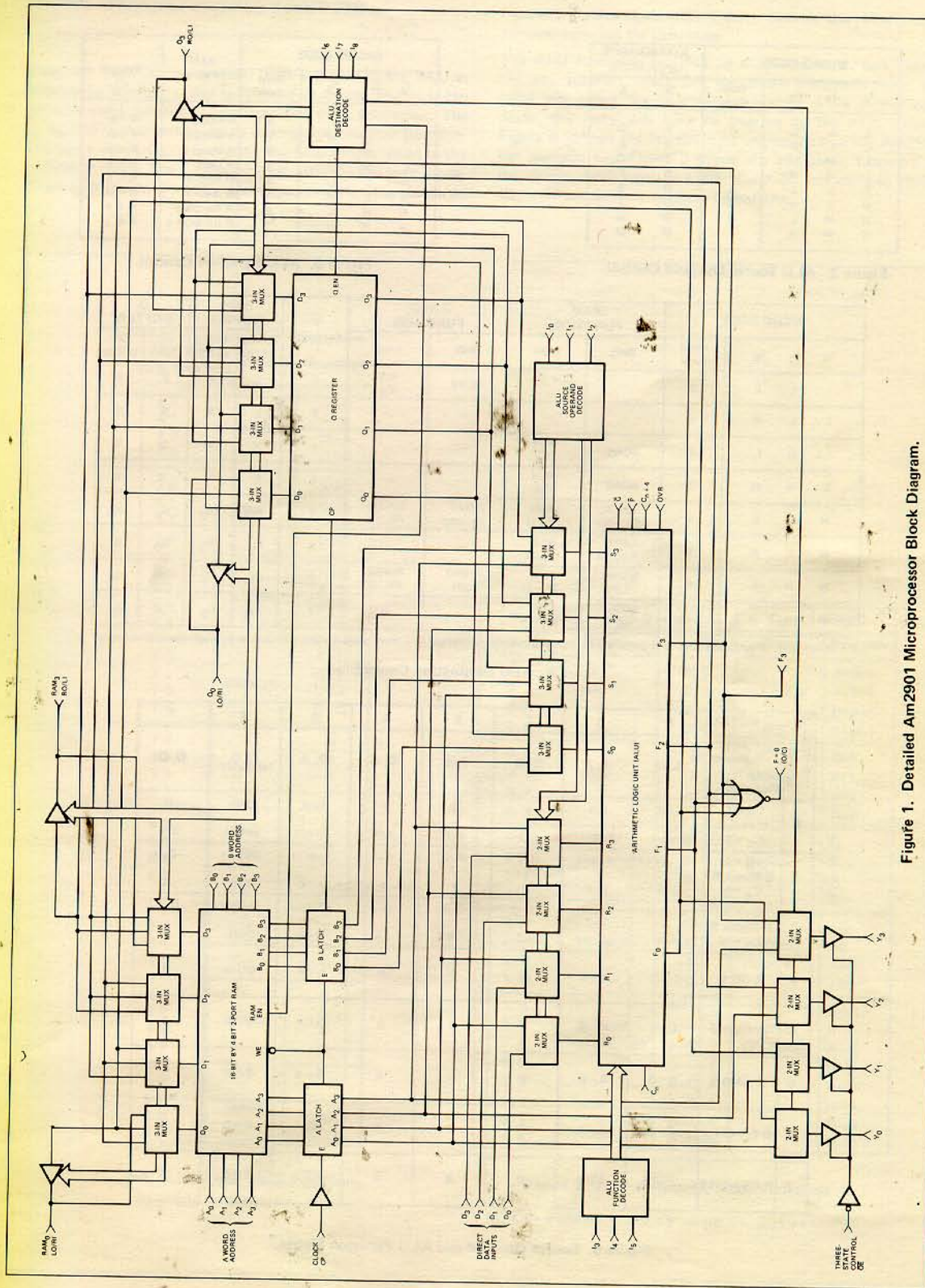


Figure 1. Detailed Am2901 Microprocessor Block Diagram.

use

MICRO CODE				ALU SOURCE OPERANDS	
I ₂	I ₁	I ₀	Octal Code	R	S
L	L	L	0	A	Q
L	L	H	1	A	B
L	H	L	2	O	Q
L	H	H	3	O	B
H	L	L	4	O	A
H	L	H	5	D	A
H	H	L	6	D	Q
H	H	H	7	D	O

Figure 2. ALU Source Operand Control.

use

MICRO CODE				ALU Function	Symbol
I ₅	I ₄	I ₃	Octal Code		
L	L	L	0	R Plus S	R + S
L	L	H	1	S Minus R	S - R
L	H	L	2	R Minus S	R - S
L	H	H	3	R OR S	R ∨ S
H	L	L	4	R AND S	R ∧ S
H	L	H	5	\bar{R} AND S	$\bar{R} \wedge S$
H	H	L	6	R EX-OR S	R ∨ S
H	H	H	7	R EX-NOR S	$\bar{R} \vee \bar{S}$

Figure 3. ALU Function Control.

No-Wire

MICRO CODE				RAM FUNCTION		Q-REG. FUNCTION		Y	RAM SHIFTER		Q SHIFTER	
I ₈	I ₇	I ₆	Octal Code	Shift	Load	Shift	Load	OUTPUT	RAM ₀ LO/RI	RAM ₃ LI/RO	Q ₀ LO/RI	Q ₃ LI/RO
L	L	L	0	—	—	NONE	ALU (F _i)	F	X	X	X	X
L	L	H	1	—	—	—	—	F	X	X	X	X
L	H	L	2	NONE	ALU (F _i)	—	—	A	X	X	X	X
L	H	H	3	NONE	ALU (F _i)	—	—	F	X	X	X	X
H	L	L	4	LEFT (DOWN)	ALU (F _{i+1})	LEFT (DOWN)	Q-REG (Q _{i+1})	F	F ₀	IN ₃	Q ₀	IN ₃
H	L	H	5	LEFT (DOWN)	ALU (F _{i+1})	—	—	F	F ₀	IN ₃	Q ₀	X
H	H	L	6	RIGHT (UP)	ALU (F _{i-1})	RIGHT (UP)	Q-REG (Q _{i-1})	F	IN ₀	F ₃	IN ₀	Q ₃
H	H	H	7	RIGHT (UP)	ALU (F _{i-1})	—	—	F	IN ₀	F ₃	X	Q ₃

X = Don't care. Electrically, the shift pin is a TTL input internally connected to a three-state output which is in the high-impedance state.

Figure 4. ALU Destination Control.

OCTAL I ₅ I ₄ I ₃	OCTAL I ₂ I ₁ I ₀	ALU Source	0	1	2	3	4	5	6	7
		ALU Function	A, Q	A, B	O, Q	O, B	O, A	D, A	D, Q	D, O
0	C _n = L R Plus S C _n = H	A+Q	A+B	Q	B	A	D+A	D+Q	D	
1	C _n = L S Minus R C _n = H	A+Q+1	A+B+1	Q+1	B+1	A+1	D+A+1	D+Q+1	D+1	
2	C _n = L R Minus S C _n = H	A-Q-1	A-B-1	-Q-1	-B-1	-A-1	D-A-1	D-Q-1	D-1	
3	R OR S	A ∨ Q	A ∨ B	Q	B	A	D ∨ A	D ∨ Q	D	
4	R AND S	A ∧ Q	A ∧ B	0	0	0	D ∧ A	D ∧ Q	0	
5	\bar{R} AND S	$\bar{A} \wedge Q$	$\bar{A} \wedge B$	Q	B	A	$\bar{D} \wedge A$	$\bar{D} \wedge Q$	0	
6	R EX-OR S	A ∨ Q	A ∨ B	Q	B	A	D ∨ A	D ∨ Q	D	
7	R EX-NOR S	$\bar{A} \vee \bar{Q}$	$\bar{A} \vee \bar{B}$	\bar{Q}	\bar{B}	\bar{A}	$\bar{D} \vee \bar{A}$	$\bar{D} \vee \bar{Q}$	\bar{D}	

+ = Plus; - = Minus; ∨ = OR; ∧ = AND; ∨ = EX-OR

Figure 5. Source Operand and ALU Function Matrix.

SOURCE OPERANDS AND ALU FUNCTIONS

There are eight source operand pairs available to the ALU as selected by the I_0 , I_1 , and I_2 instruction inputs. The ALU can perform eight functions; five logic and three arithmetic. The I_3 , I_4 , and I_5 instruction inputs control this function selection. The carry input, C_n , also affects the ALU results when in the arithmetic mode. The C_n input has no effect in the logic mode. When I_0 through I_5 and C_n are viewed together, the matrix of

Figure 5 results. This matrix fully defines the ALU/source operand function for each state.

The ALU functions can also be examined on a "task" basis, i.e., add, subtract, AND, OR, etc. In the arithmetic mode, the carry will affect the function performed while in the logic mode, the carry will have no bearing on the ALU output. Figure 6 defines the various logic operations that the Am2901 can perform and Figure 7 shows the arithmetic functions of the device. Both carry-in LOW ($C_n = 0$) and carry-in HIGH ($C_n = 1$) are defined in these operations.

Octal I543, I210	Group	Function
4 0	AND	$A \wedge Q$
4 1		$A \wedge B$
4 5		$D \wedge A$
4 6		$D \wedge Q$
3 0	OR	$A \vee Q$
3 1		$A \vee B$
3 5		$D \vee A$
3 6		$D \vee Q$
6 0	EX-OR	$A \vee Q$
6 1		$A \vee B$
6 5		$D \vee A$
6 6		$D \vee Q$
7 0	EX-NOR	$\overline{A \vee Q}$
7 1		$\overline{A \vee B}$
7 5		$\overline{D \vee A}$
7 6		$\overline{D \vee Q}$
7 2	INVERT	\overline{Q}
7 3		\overline{B}
7 4		\overline{A}
7 7		\overline{D}
6 2	PASS	Q
6 3		B
6 4		A
6 7		D
3 2	PASS	Q
3 3		B
3 4		A
3 7		D
4 2	"ZERO"	0
4 3		0
4 4		0
4 7		0
5 0	MASK	$\overline{A} \wedge Q$
5 1		$\overline{A} \wedge B$
5 5		$\overline{D} \wedge A$
5 6		$\overline{D} \wedge Q$

Figure 6. ALU Logic Mode Functions.
(C_n Irrelevant)

Octal I543, I210	$C_n = 0$ (Low)		$C_n = 1$ (High)	
	Group	Function	Group	Function
0 0	ADD	A+Q	ADD plus one	A+Q+1
0 1		A+B		A+B+1
0 5		D+A		D+A+1
0 6		D+Q		D+Q+1
0 2	PASS	Q	Increment	Q+1
0 3		B		B+1
0 4		A		A+1
0 7		D		D+1
1 2	Decrement	Q-1	PASS	Q
1 3		B-1		B
1 4		A-1		A
2 7		D-1		D
2 2	1's Comp.	$\overline{Q-1}$	2's Comp. (Negate)	\overline{Q}
2 3		$\overline{B-1}$		\overline{B}
2 4		$\overline{A-1}$		\overline{A}
1 7		$\overline{D-1}$		\overline{D}
1 0	Subtract (1's Comp)	$Q-A-1$	Subtract (2's Comp)	$Q-A$
1 1		$B-A-1$		$B-A$
1 5		$A-D-1$		$A-D$
1 6		$Q-D-1$		$Q-D$
2 0		$A-Q-1$		$A-Q$
2 1		$A-B-1$		$A-B$
2 5		$D-A-1$		$D-A$
2 6		$D-Q-1$		$D-Q$

Figure 7. ALU Arithmetic Mode Functions.

LOGIC FUNCTIONS FOR G, P, C_{n+4}, AND OVR

The four signals G, P, C_{n+4}, and OVR are designed to indicate carry and overflow conditions when the Am2901 is in the add or subtract mode. The table below indicates the logic equations for these four signals for each of the eight ALU functions. The R and S inputs are the two inputs selected according to Figure 2.

Definitions (+ = OR)

$$\begin{aligned} P_0 &= R_0 + S_0 & G_0 &= R_0 S_0 \\ P_1 &= R_1 + S_1 & G_1 &= R_1 S_1 \\ P_2 &= R_2 + S_2 & G_2 &= R_2 S_2 \\ P_3 &= R_3 + S_3 & G_3 &= R_3 S_3 \end{aligned}$$

$$C_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_n$$

$$C_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_n$$

I ₅₄₃	Function	\bar{P}	\bar{G}	C _{n+4}	OVR
0	R + S	$\overline{P_3 P_2 P_1 P_0}$	$\overline{G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0}$	C ₄	C ₃ ∨ C ₄
1	S - R	← Same as R + S equations, but substitute \bar{R}_i for R _i in definitions →			
2	R - S	← Same as R + S equations, but substitute \bar{S}_i for S _i in definitions →			
3	R ∨ S	LOW	$P_3 P_2 P_1 P_0$	$\overline{P_3 P_2 P_1 P_0} + C_n$	$\overline{P_3 P_2 P_1 P_0} + C_n$
4	R ∧ S	LOW	$\overline{G_3 + G_2 + G_1 + G_0}$	$G_3 + G_2 + G_1 + G_0 + C_n$	$\overline{G_3 + G_2 + G_1 + G_0} + C_n$
5	$\bar{R} \wedge S$	LOW	← Same as R ∧ S equations, but substitute \bar{R}_i for R _i in definitions →		
6	R ∨ \bar{S}	← Same as R ∨ S, but substitute \bar{R}_i for R _i in definitions →			
7	$\bar{R} \vee \bar{S}$	$G_3 + G_2 + G_1 + G_0$	$P_3 G_3 + P_3 P_2 G_2 + P_3 P_2 P_1 G_1 + P_3 P_2 P_1 P_0$	$P_3 G_3 + P_3 P_2 G_2 + P_3 P_2 P_1 G_1 + P_3 P_2 P_1 P_0 (G_0 + C_n)$	Complement of C _{n+4} at left

+ = OR

Figure 8.

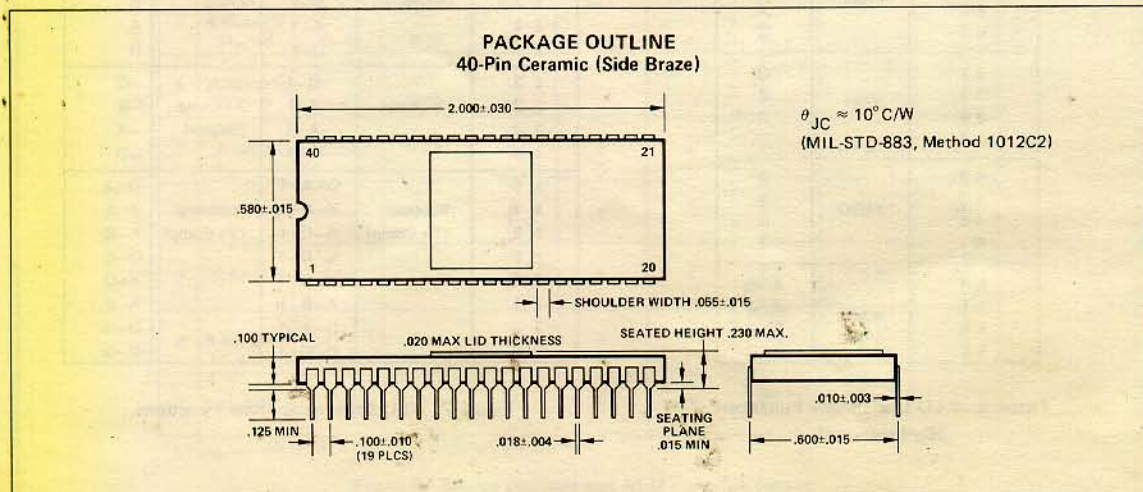
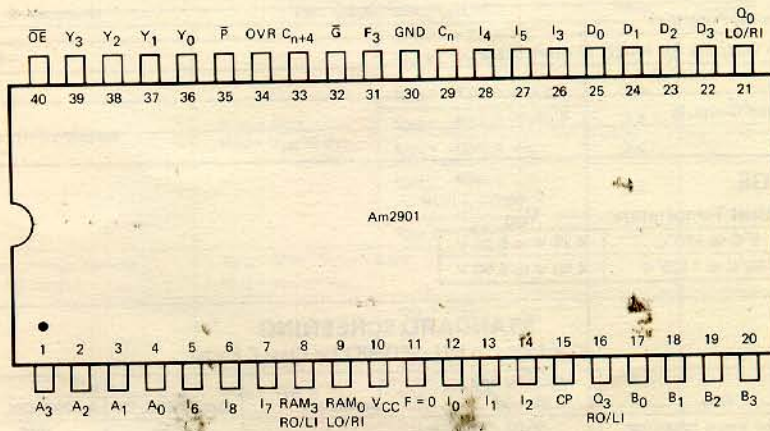


Figure 9.

CONNECTION DIAGRAM
Top View



Note: Pin 1 is marked for orientation.

Figure 10.

PIN DEFINITIONS

- A₀₋₃** The four address inputs to the register stack used to select one register whose contents are displayed through the A-port.
- B₀₋₃** The four address inputs to the register stack used to select one register whose contents are displayed through the B-port and into which new data can be written when the clock goes LOW.
- I₀₋₈** The nine instruction control lines to the Am2901, used to determine what data sources will be applied to the ALU (I₀₁₂), what function the ALU will perform (I₃₄₅), and what data is to be deposited in the Q-register or the register stack (I₆₇₈).
- RO/LI** A shift line at the MSB of the Q register (Q₃RO/LI) and the register stack (RAM₃RO/LI). Electrically these lines are three-state outputs connected to TTL inputs internal to the Am2901. When the destination code on I₆₇₈ indicates a right shift (octal 6 or 7) the three-state outputs are enabled and the MSB of the Q register is available on the Q₃RO/LI pin and the MSB of the ALU output is available on the RAM₃RO/LI pin. Otherwise, the three-state outputs are OFF (high-impedance) and the pins are electrically LS-TTL inputs. When the destination code calls for a down (left) shift, the pins are used as the data inputs to the MSB of the Q register (octal 4) and RAM (octal 4 or 5).
- LO/RI** Shift lines like RO/LI, but at the LSB of the Q-register and RAM. These pins are tied to the RO/LI pin of the adjacent device to transfer data between devices for left and right shifts of the Q register and ALU data.
- D₀₋₃** Direct data inputs. A four-bit data field which may be selected as one of the ALU data sources for entering data into the Am2901. D₀ is the LSB.
- Y₀₋₃** The four data outputs of the Am2901. These are three-state output lines. When enabled, they display either the four outputs of the ALU or the data on the A-port of the register stack, as determined by the destination code I₆₇₈.
- \overline{OE}** Output Enable. When \overline{OE} is HIGH, the Y outputs are OFF; when \overline{OE} is LOW, the Y outputs are active (HIGH or LOW).
- $\overline{P}, \overline{G}$** The carry generate and propagate outputs of the Am2901's ALU. These signals are used with the Am2902 for carry-lookahead. See Figure 8 for the logic equations.
- OVR** Overflow. This pin is logically the Exclusive-OR of the carry-in and carry-out of the MSB of the ALU. At the most significant end of the word, this pin indicates that the result of an arithmetic two's complement operation has overflowed into the sign-bit. See Figure 8 for logic equation.
- F = 0** This is an open collector output which goes HIGH (OFF) if the data on the four ALU outputs F₀₋₃ are all LOW. In positive logic, it indicates the result of an ALU operation is zero.
- C_n** The carry-in to the Am2901's ALU.
- C_{n+4}** The carry-out of the Am2901's ALU. See Figure 8 for equations.
- CP** The clock to the Am2901. The Q register and register stack outputs change on the clock LOW-to-HIGH transition. The clock LOW time is internally the write enable to the 16 x 4 RAM which comprises the "master" latches of the register stack. While the clock is LOW, the "slave" latches on the RAM outputs are closed, storing the data previously on the RAM outputs. This allows synchronous master-slave operation of the register stack.

MAXIMUM RATINGS (Above which the useful life may be impaired)

Storage Temperature	-65°C to +150°C
Temperature (Ambient) Under Bias	-55°C to +125°C
Supply Voltage to Ground Potential	-0.5 V to +6.3 V
DC Voltage Applied to Outputs for HIGH Output State	-0.5 V to +V _{CC} 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

OPERATING RANGE

P/N	Ambient Temperature	V _{CC}
Am2901PC, DC	0°C to +70°C	4.75 V to 5.25 V
Am2901DM, FM	-55°C to +125°C	4.50 V to 5.50 V

STANDARD SCREENING
(Conforms to MIL-STD-883 for Class C Parts)

Step	MIL-STD-883 Method	Conditions	Level	
			Am2901PC, DC	Am2901DM, FM
Pre-Seal Visual Inspection	2010	B	100%	100%
Stabilization Bake	1008	C 24-hour 150°C	100%	100%
Temperature Cycle	*1010	C -65°C to +150°C 10 cycles	100%	100%
Centrifuge	2001	B 10,000 G	100% *	100%
Fine Leak	1014	A 5 x 10 ⁻⁸ atm-cc/cm ³	100% *	100%
Gross Leak	1014	C2 Fluorocarbon	100% *	100%
Electrical Test Subgroups 1 and 7	5004	See below for definitions of subgroups	100%	100%
Insert Additional Screening here for Class B Parts				
Group A Sample Tests				
Subgroup 1	5005	See below for definitions of subgroups	LTPD = 5	LTPD = 5
Subgroup 2			LTPD = 7	LTPD = 7
Subgroup 3			LTPD = 7	LTPD = 7
Subgroup 7			LTPD = 7	LTPD = 7
Subgroup 8			LTPD = 7	LTPD = 7
Subgroup 9			LTPD = 7	LTPD = 7

*Not applicable for Am2901PC

ADDITIONAL SCREENING FOR CLASS B PARTS

Step	MIL-STD-883 Method	Conditions	Level
			Am2901DMB, FMB
Burn-In	1015	D 125°C 160 hours min.	100%
Electrical Test Subgroup 1 Subgroup 2 Subgroup 3 Subgroup 7 Subgroup 9	5004		100% 100% 100% 100% 100%
Return to Group A Tests in Standard Screening			

ORDERING INFORMATION

Package Type	Temperature Range	Order Number
Molded DIP	0°C to +70°C	AM2901PC
Hermetic DIP	0°C to +70°C	AM2901DC
Hermetic DIP	-55°C to +125°C	AM2901DM
Hermetic Flat Pack	-55°C to +125°C	AM2901FM
Dice	0°C to +70°C	AM2901XC

GROUP A SUBGROUPS

(as defined in MIL-STD-883, method 5005)

Subgroup	Parameter	Temperature
1	DC	25°C
2	DC	Maximum rated temperature
3	DC	Minimum rated temperature
7	Function	25°C
8	Function	Maximum and minimum rated temperature
9	Switching	25°C
10	Switching	Maximum Rated Temperature
11	Switching	Minimum Rated Temperature

ELECTRICAL CHARACTERISTICS OVER OPERATING RANGE (Unless Otherwise Noted)
 (Group A, Subgroups 1, 2 and 3)

Parameters	Description	Test Conditions (Note 1)	Min.	Typ. (Note 2)	Max.	Units
V _{OH}	Output HIGH Voltage	V _{CC} = MIN. V _{IN} = V _{IH} or V _{IL}	I _{OH} = -1.6mA Y ₀ , Y ₁ , Y ₂ , Y ₃	2.4		Volts
			I _{OH} = -1.0mA, C _{n+4}	2.4		
			I _{OH} = -800μA, OVR, P̄	2.4		
			I _{OH} = -600μA, F ₃	2.4		
			I _{OH} = -600μA All RO/LI, LO/RI	2.4		
I _{OH} = -1.6mA, Ḡ	2.4					
I _{CEX}	Output Leakage Current for F = 0 Output	V _{CC} = MIN., V _{OH} = 5.5V V _{IN} = V _{IH} or V _{IL}			250	μA
V _{OL}	Output LOW Voltage	V _{CC} = MIN., V _{IN} = V _{IH} or V _{IL}	I _{OL} = 16mA Y ₀ , Y ₁ , Y ₂ , Y ₃		0.5	Volts
			I _{OL} = 10mA, C _{n+4} , F = 0		0.5	
			I _{OL} = 8.0mA, OVR, P̄		0.5	
			I _{OL} = 6.0mA, F ₃ All RO/LI, LO/RI		0.5	
V _{IH}	Input HIGH Level	Guaranteed input logical HIGH voltage for all inputs	2.0			Volts
V _{IL}	Input LOW Level	Guaranteed input logical LOW voltage for all inputs	Military		0.7	Volts
			Commercial		0.8	
V _I	Input Clamp Voltage	V _{CC} = MIN., I _{IN} = -18mA			-1.5	Volts
I _{IL} (Note 3)	Input LOW Current	V _{CC} = MAX. V _{IN} = 0.5V	Clock, OE		-0.36	mA
			A ₀ , A ₁ , A ₂ , A ₃		-0.36	
			B ₀ , B ₁ , B ₂ , B ₃		-0.36	
			D ₀ , D ₁ , D ₂ , D ₃		-0.72	
			I ₀ , I ₁ , I ₂ , I ₆ , I ₈		-0.36	
			I ₃ , I ₄ , I ₅ , I ₇		-0.72	
			All LO/RI, RO/LI (Note 4)		-0.8	
			C _n		-3.6	
I _{IH} (Note 3)	Input HIGH Current	V _{CC} = MAX. V _{IN} = 2.7V	Clock, OE		20	μA
			A ₀ , A ₁ , A ₂ , A ₃		20	
			B ₀ , B ₁ , B ₂ , B ₃		20	
			D ₀ , D ₁ , D ₂ , D ₃		40	
			I ₀ , I ₁ , I ₂ , I ₆ , I ₈		20	
			I ₃ , I ₄ , I ₅ , I ₇		40	
			All LO/RI, RO/LI (Note 4)		100	
			C _n		200	
I _I	Input HIGH Current	V _{CC} = MAX., V _{IN} = 5.5V			1.0	mA
I _{OZ}	Off State (High Impedance) Output Current	V _{CC} = MAX.	Y ₀ , Y ₁ , Y ₂ , Y ₃	V _O = 2.4V	50	μA
				V _O = 0.5V	-50	
			All LO/RI, RO/LI	V _O = 2.4V (Note 5)	100	
				V _O = 0.5V (Note 5)	-800	
I _{SC}	Output Short Circuit Current (Note 4)		Y ₀ , Y ₁ , Y ₂ , Y ₃ , Ḡ		-6.0	mA
			C _{n+4}		-6.0	
			OVR, P̄		-6.0	
			F ₃		-6.0	
			All RO/LI, LO/RI		-6.0	
I _{CC}	Power Supply Current	V _{CC} = MAX.	Military	185	280	mA
			Commercial	185	280	

- Notes: 1. For conditions shown as MIN or MAX, use the appropriate value specified under Electrical Characteristics for the applicable device type.
 2. Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
 3. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
 4. LO/RI and RO/LI are three-state outputs internally connected to TTL inputs. Input characteristics are measured with I₆₇₈ in a state such that the three-state output is OFF.

GUARANTEED OPERATING CONDITIONS

Tables I, II, and III below define the timing requirements of the Am2901 in a system. The Am2901 is guaranteed to function correctly over the operating range when used within the delay and set-up time constraints of these tables for the appropriate device type. The tables are divided into three types of parameters; clock characteristics, combinational delays from inputs to outputs, and set-up and hold time requirements. The latter table defines the time prior to the end of the cycle (i.e., clock LOW-to-HIGH transition) that each input must be stable to guarantee that the correct data is written into one of the internal registers.

The performance of the Am2901 within the limits of these tables is guaranteed by the testing defined as "Group A, Subgroup 9" Electrical Testing. For a copy of the tests and limits used for subgroup 9, contact Advanced Micro Devices' Product Marketing.

TABLE I

CYCLE TIME AND CLOCK CHARACTERISTICS

TIME	Am2901DC	Am2901DM
Minimum Read-Modify-Write Cycle (time from selection of A, B registers to end of cycle)	105 ns	120 ns
Maximum Clock Frequency to Shift Q Register (50% duty cycle)	9.5 MHz	8.3 MHz
Minimum Clock LOW Time	30 ns	30 ns
Minimum Clock HIGH Time	30 ns	30 ns
Minimum Clock Period	105 ns	120 ns

TABLE II

MAXIMUM COMBINATIONAL PROPAGATION DELAYS (all in ns, $C_L \leq 15\text{pF}$)

From Input \ To Output	Am2901DC								Am2901DM							
	Y	F ₃	C _{n+4}	\bar{G}, \bar{P}	F=0 R _L = 470	OVR	RO, LO		Y	F ₃	C _{n+4}	\bar{G}, \bar{P}	F=0 R _L = 470	OVR	RO, LO	
							RAM	Q							RAM	Q
Clock	115	85	100	100	110	95	105	60	125	95	110	110	120	105	115	65
A, B	110	85	80	80	110	75	110	—	120	95	90	90	120	85	120	—
D	100	70	70	70	100	60	60	—	110	80	75	75	110	65	65	—
C _n	55	35	30	—	50	40	55	—	60	40	30	—	55	45	60	—
I ₀₁₂	85	65	65	65	80	65	80	—	90	70	70	70	85	70	85	—
I ₃₄₅	70	55	60	60	70	60	65	—	75	60	65	65	75	65	70	—
I ₆₇₈	55	—	—	—	—	—	45	45	60	—	—	—	—	—	50	50
OE Enable/Disable	40/25	—	—	—	—	—	—	—	40/25	—	—	—	—	—	—	—
A bypassing ALU (I = 2xx)	60	—	—	—	—	—	—	—	65	—	—	—	—	—	—	—

SET-UP AND HOLD TIMES (minimum cycles from each input)

Set-up and hold times are defined relative to the clock LOW-to-HIGH edge. Inputs must be steady at all times from the set-up

time prior to the clock until the hold time after the clock. The set-up times allow sufficient time to perform the correct operation on the correct data so that the correct ALU data can be written into one of the registers.

TABLE III

Set-Up and Hold Times (all in ns) (Note 1)

From Input	Notes	Am2901DC		Am2901DM	
		Set-Up Time	Hold Time	Set-Up Time	Hold Time
A, B Source	2, 3, 4	105 $t_{pwL} + 30$	0	120 $t_{pwL} + 30$	0
B Dest.	2, 4	$t_{pwL} + 15$	0	$t_{pwL} + 15$	0
D		100	0	110	0
C _n		55	0	60	0
I ₀₁₂		85	0	90	0
I ₃₄₅		70	0	75	0
I ₆₇₈	4	$t_{pwL} + 15$	0	$t_{pwL} + 15$	0
RI, LI (RAM or Q)		30	0	30	0

Notes: 1. See Figure 11 and 12.

2. If the B address is used as a source operand, allow for the "A, B source" set-up time; if it is used only for the destination address, use the "B dest." set-up time.

3. Where two numbers are shown, both must be met.

4. " t_{pwL} " is the clock LOW time.

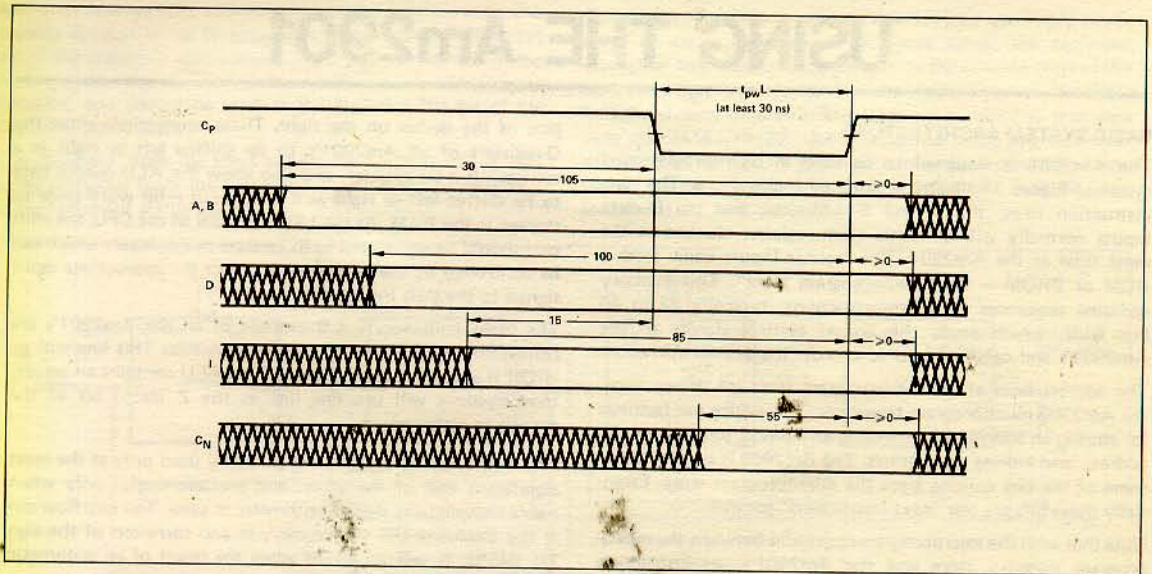
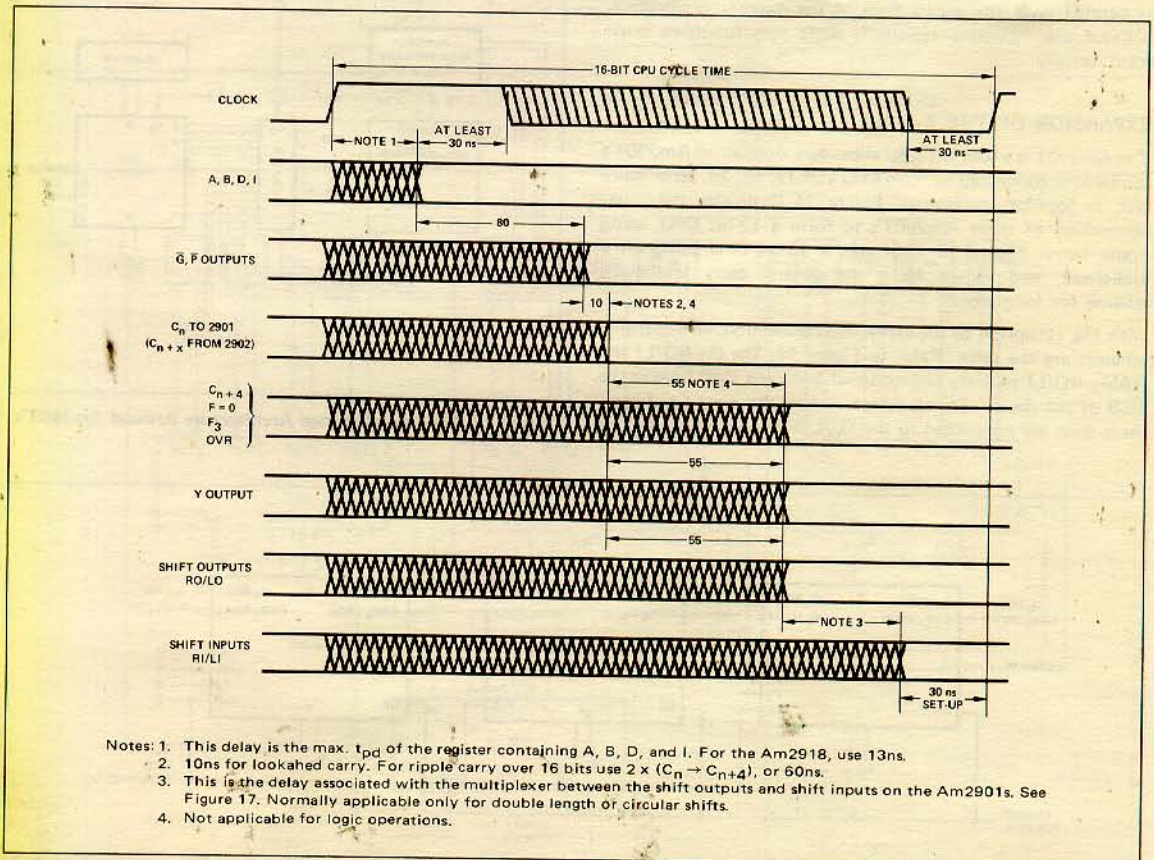


Figure 11. Minimum Cycle Times from Inputs. Numbers Shown are Minimum Data Stable Times for Am2901 DC, in ns.



- Notes: 1. This delay is the max. t_{pd} of the register containing A, B, D, and I. For the Am2918, use 13ns.
 2. 10ns for lookahed carry. For ripple carry over 16 bits use $2 \times (C_n \rightarrow C_{n+4})$, or 60ns.
 3. This is the delay associated with the multiplexer between the shift outputs and shift inputs on the Am2901s. See Figure 17. Normally applicable only for double length or circular shifts.
 4. Not applicable for logic operations.

Figure 12. Switching Waveforms for 16-Bit System Assuming A, B, D and I are all Driven from Registers with the same Propagation Delay, Clocked by the Am2901 Clock.

USING THE Am2901

BASIC SYSTEM ARCHITECTURE

The Am2901 is designed to be used in microprogrammed systems. Figure 13 illustrates such an architecture. The nine instruction lines, the A and B addresses, and the D data inputs normally will all come from registers clocked at the same time as the Am2901. The register inputs come from a ROM or PROM — the "microprogram store". This memory contains sequences of microinstructions, typically 28 to 40 bits wide, which apply the proper control signals to the Am2901's and other circuits to execute the desired operation.

The address lines of the microprogram store are driven from the Am2909 microprogram sequencer. This device has facilities for storing an address, incrementing an address, jumping to any address, and linking subroutines. The Am2909 is controlled by some of the bits coming from the microprogram store. Essentially these bits are the "next instruction" control.

Note that with the microprogram register in-between the microprogram memory store and the Am2901's, an instruction accessed on one cycle is executed on the next cycle. As one instruction is executed, the next instruction is being read from microprogram memory. In this configuration, system speed is improved because the execution time in the Am2901's occurs in parallel with the access time of the microprogram store. Without the "pipeline register", these two functions must occur serially.

EXPANSION OF THE Am2901

The Am2901 is a four-bit CPU slice. Any number of Am2901's can be interconnected to form CPU's of 12, 16, 24, 36 or more bits, in four-bit increments. Figure 14 illustrates the interconnection of three Am2901's to form a 12-bit CPU, using ripple carry. Figure 15 illustrates a 16-bit CPU using carry lookahead, and Figure 16 is the general carry lookahead scheme for long words.

With the exception of the carry interconnection, all expansion schemes are the same. Refer to Figure 14. The Q_3 -RO/LI and RAM_3 -RO/LI pins are bidirectional left/right shift lines at the MSB of the device. For all devices except the most significant, these lines are connected to the Q_0 -LO/RI and RAM_0 -LO/RI

pins of the device on the right. These connections allow the Q-registers of all Am2901's to be shifted left or right as a contiguous n-bit register, and also allow the ALU output data to be shifted left or right as a contiguous n-bit word prior to storage in the RAM. At the LSB and MSB of the CPU, the shift pins should be connected to three-state multiplexers which can be controlled by the microcode to select the appropriate input signals to the shift inputs. (See Figure 17)

The open collector $F = 0$ outputs of all the Am2901's are connected together and to a pull-up resistor. This line will go HIGH if and only if the output of the ALU contains all zeroes. Most systems will use this line, as the Z (zero) bit of the processor status word.

The overflow and F_3 pins are generally used only at the most significant end of the array, and are meaningful only when two's complement signed arithmetic is used. The overflow pin is the Exclusive-OR of the carry-in and carry-out of the sign bit (MSB). It will go HIGH when the result of an arithmetic

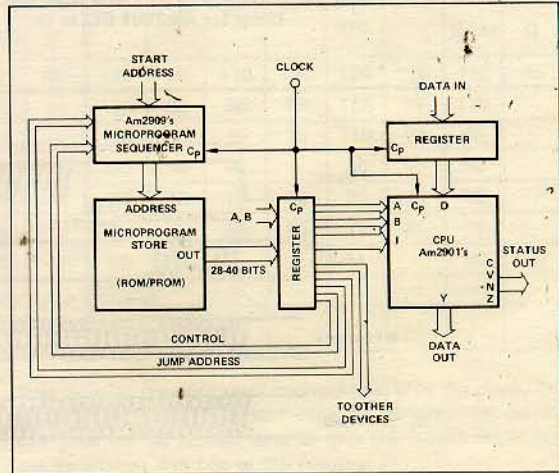


Figure 13. Microprogrammed Architecture Around Am2901's.

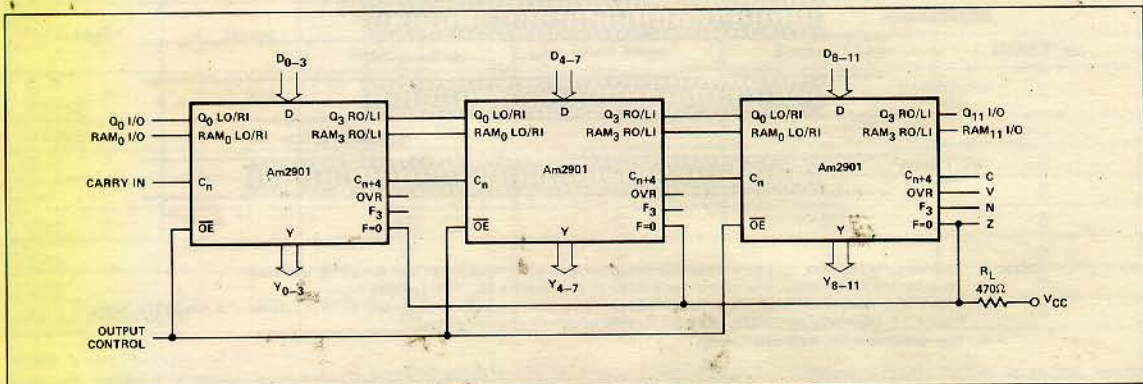


Figure 14. Three Am2901's used to Construct 12-Bit CPU with Ripple Carry. Corresponding A, B, and I Pins on all Devices are Connected Together.

operation is a number requiring more bits than are available, causing the sign bit to be erroneous. This is the overflow (V) bit of the processor status word. The F₃ pin is the MSB of the ALU output. It is the sign of the result in two's complement notation, and should be used as the Negative (N) bit of the processor status word.

The carry-out from the most significant Am2901 (C_{n+4} pin) is the carry-out from the array, and is used as the carry (C) bit of the processor status word.

Carry interconnections between devices may use either ripple carry or carry lookahead. For ripple carry, the carry-out (C_{n+4}) of each device is connected to the carry-in (C_n) of the next more significant device. Carry lookahead uses the Am2902 lookahead carry generator. The scheme is identical to that used with the 74181/74182. Users unfamiliar with this technique should refer to AMD's application note on Arithmetic Logic Units. Figures 15 and 16 illustrate single and multiple level lookahead.

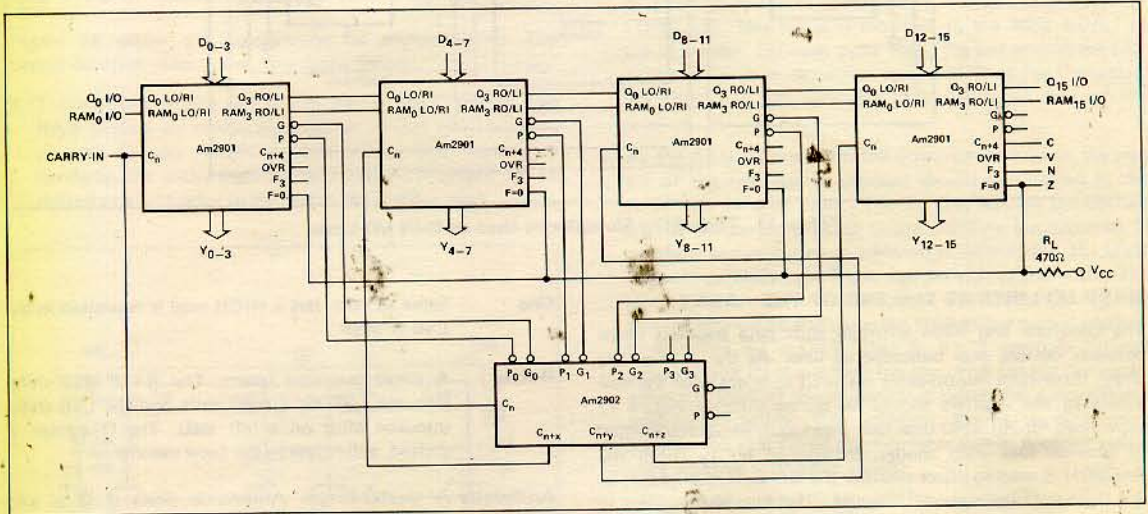


Figure 15. Four Am2901's in a 16-Bit CPU using the Am2902 for Carry Lookahead.

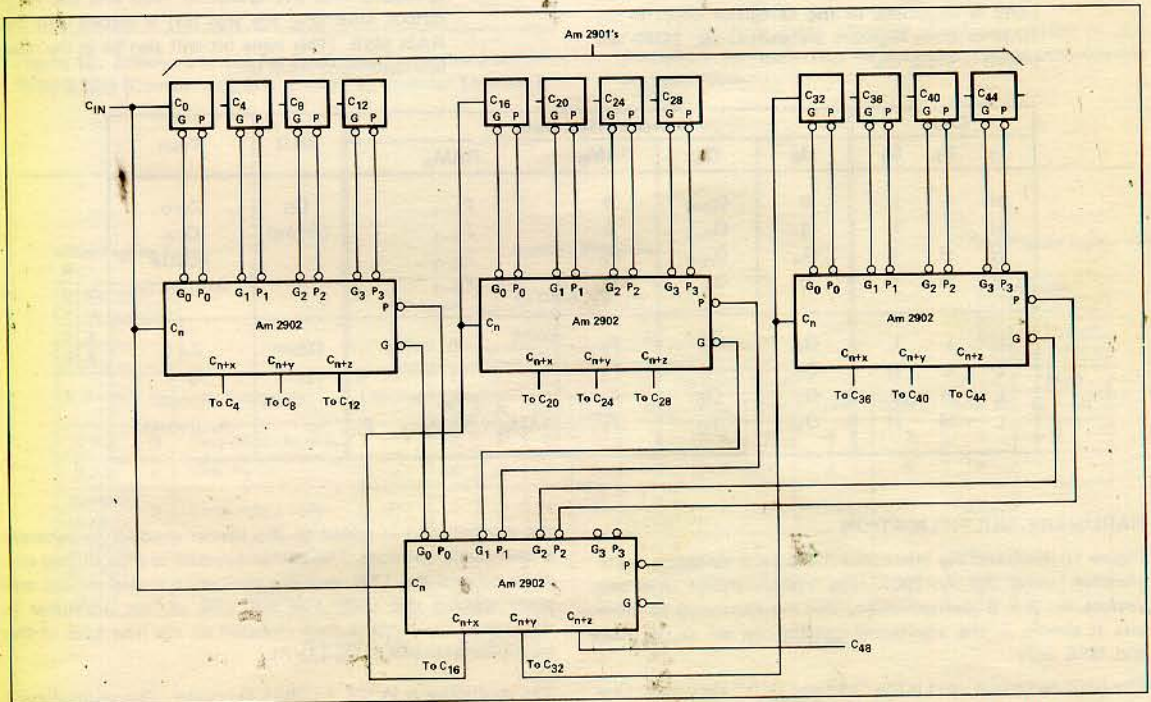


Figure 16. Carry Lookahead Scheme for 48-Bit CPU using 12 Am2901's. The Carry-Out Flag (C₄₈) Should be Taken from the Lower Am2902 Rather than the Right-Most Am2901 for Higher Speed.

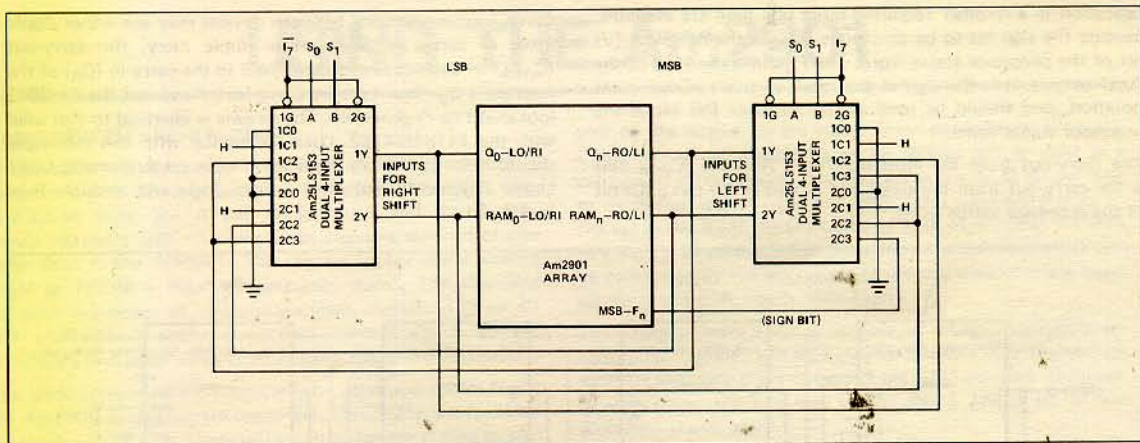


Figure 17. Three-State Multiplexers Used on Shift I/O Lines.

SHIFT I/O LINES AT THE END OF THE ARRAY

The Q-register and RAM left/right shift data transfers occur between devices over bidirectional lines. At the ends of the array, three-state multiplexers are used to select what the new inputs to the registers should be during shifting. Figure 17 shows two Am25LS253 dual four-input multiplexers connected to provide four shift modes. Instruction bit I_7 (from the Am2901) is used to select whether the left-shift multiplexer or the right-shift multiplexer is active. The four shift modes in this example are:

Zero A LOW is shifted into the MSB of the RAM on a down shift. If the Q-register is also shifted, then a LOW is deposited in the Q-register MSB. If the RAM or both registers are shifted up, LOWs are placed in the LSBs.

One Same as zero, but a HIGH level is deposited in the LSB or MSB.

Rotate A single precision rotate. The RAM MSB shifts into the LSB on a right shift and the LSB shifts into the MSB on a left shift. The Q-register, if shifted, will rotate in the same manner.

Arithmetic A double-length Arithmetic Shift if Q is also shifted. On an up shift a zero is loaded into the Q-register LSB and the Q-register MSB is loaded into the RAM LSB. On a down shift, the RAM LSB is loaded into the Q-register MSB and the ALU output MSB (F_n , the sign bit) is loaded into the RAM MSB. (This same bit will also be in the next less significant RAM bit.)

Code			Source of New Data				Shift	Type
I_7	S_1	S_0	Q_0	Q_n	RAM_0	RAM_n		
H	L	L	0	Q_{n-1}	0	F_{n-1}	Up (Right)	Zero One Rotate Arithmetic
H	L	H	1	Q_{n-1}	1	F_{n-1}		
H	H	L	Q_n	Q_{n-1}	F_n	F_{n-1}		
H	H	H	0	Q_{n-1}	Q_n	F_{n-1}		
L	L	L	Q_1	0	F_1	0	Down (Left)	Zero One Rotate Arithmetic
L	L	H	Q_1	1	F_1	1		
L	H	L	Q_1	Q_0	F_1	F_0		
L	H	H	Q_1	F_0	F_1	$RAM_n = RAM_{n-1} = F_n$		

HARDWARE MULTIPLICATION

Figure 18 illustrates the interconnections for a hardware multiplication using the Am2901. The system shown uses two devices for 8×8 multiplication, but the expansion to more bits is simple — the significant connections are at the LSB and MSB only.

The basic technique used is the "add and shift" algorithm. One clock cycle is required for each bit of the multiplier. On each cycle, the LSB of the multiplier is examined; if it is a "1", then

the multiplicand is added to the partial product to generate a new partial product. The partial product is then shifted one place toward the LSB, and the multiplier is also shifted one place toward the LSB. The old LSB of the multiplier is discarded. The cycle is then repeated on the new LSB of the multiplier available at Q_0 -LO/RI.

The multiplier is in the Am2901 Q-register. The multiplicand is in one of the registers in the register stack, R_a . The product will be developed in another of the registers in the stack, R_b .

The A address inputs are used to address the multiplicand in R_a , and the B address inputs are used to address the partial product in R_b . On each cycle, R_a is conditionally added to R_b , depending on the LSB of Q as read from the Q_0 -LO/RI output, and both Q and the ALU output are shifted left one place. The instruction lines to the Am2901 on every cycle will be:

$I_{876} = 4$ (shift register stack input and Q register left)
 $I_{543} = 0$ (Add)
 $I_{210} = 1$ or 3 (select A, B or 0, B as ALU sources)

Figure 18 shows the connections for multiplication. The circled numbers refer to the paragraphs below.

1. The adjacent RO/LI and LO/RI pins of the Q-register and RAM shifters are connected together so that the Q-registers of both (or all) Am2901's shift left or right as a unit. Similarly, the entire eight-bit (or more) ALU output can be shifted as a unit prior to storage in the register stack.

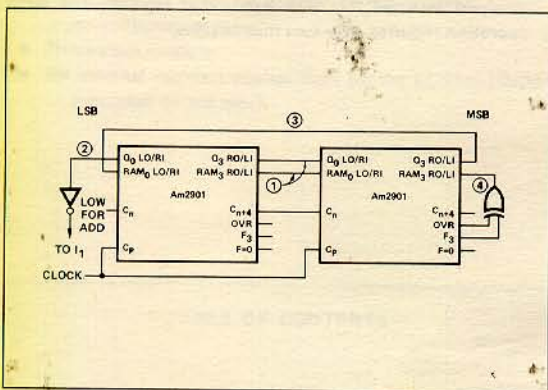


Figure 18. Interconnection for Dedicated Multiplication (8 by 8 Bit) (Corresponding A, B and I Connected Together).

2. The shift output at the LSB of the Q-register determines whether the ALU source operands will be A and B (add multiplicand to partial product) or 0 and B (add nothing to partial product). Instruction bit I_1 can select between A, B or 0, B as the source operands; it can be driven directly from the complement of the LSB of the multiplier.

3. As the new partial product appears at the input to the register stack, it is shifted left by the RAM shifter. The new LSB of the partial product, which is complete and will not be affected by future operations, is available on the RAM_0 -LO/RI pin. This signal is returned to the MSB RO/LI of the Q-register. On each cycle then, the just-completed LSB of the product is deposited in the MSB of the Q-register; the Q-register fills with the least significant half of the product.

4. As the ALU output is shifted down on each cycle, the sign bit of the new partial product should be inserted in the RAM left in shift input. The F_3 flag will be the correct sign of the partial product unless overflow has occurred. If overflow occurs during an addition or subtraction, the OVR flag will go HIGH and F_3 is not the sign of the result. The sign of the result must then be the complement of F_3 . The correct sign bit to shift into the MSB of the partial product is therefore $\overline{F_3} \oplus OVR$; that is, F_3 if overflow has not occurred and $\overline{F_3}$ if overflow has occurred. On the last cycle, when the MSB of the multiplier is examined, a conditional subtraction rather than addition should be performed, because the sign bit of the multiplier carries negative rather than positive arithmetic weight

$$(Y = -Y_i 2^i + Y_{i-1} 2^{i-1} + \dots + Y_0 2^0).$$

This scheme will produce a correct two's complement product for all multiplicands and multipliers in two's complement notation.

Figure 19 is a table showing the input states of the Am2901's for each step of a signed, two's complement multiplication.

Initial Register States			Am2901 Microcode											Final Register States		
R			Program _____ 2's Comp. Multiply											R		
0	Multiplier		Date _____ 8/5/75 _____ By _____ J. S.											0	Multiplier	
1	Multiplicand													1	Multiplicand	
2	X													2	LSH Product	
3	X													3	MSH Product	
S, F →	D	Description	Repeat	Pin States (Octal)										Jump		
				A	B	I_{876}	I_{543}	I_{210}	C_n	Q LI	Q RI	RAM LI	RAM RI	To	If	
$O \vee A$	Q	Move Multiplier to Q	—	0	X	0	3	4	X	X	X	X	X			
$O \wedge B$	B	Clear R_3	—	X	3	2	4	3	X	X	X	X	X			
$(Q+B)/2$ $(A+B)/2$	B	Cond. Add & Shift	$n-1$	1	3	4	0	$\frac{1 \text{ or } 3}{I_1 = Q_0 LO}$	0	—	$RAM_0 LO$	—	$N(F_3)$			
$(B-O)/2$ $(B-A)/2$	B	Cond. Subt. & Shift	—	1	3	4	1	$\frac{1 \text{ or } 3}{I_1 = Q_0 LO}$	1	—	$RAM_0 LO$	—	$N(F_3)$			
$O \vee Q$	B	Move LSH Prod. to R_2	—	X	2	2	3	2	X	X	X	X	X			

X = Don't Care S = Source F = Function D = Destination

Figure 19.

EXAMPLES OF SOME OTHER OPERATIONS

1. Byte Swapping

Occasionally the two halves of a 16-bit word must be swapped. D_{0-7} is interchanged with D_{8-15} . The quickest way to perform this operation is to rotate the word in RAM, shifting two bits at a time. Only four shift cycles are required. The same register is selected on both the A and B ports; the two are added together with carry-in connected to carry-out, producing a right shift of one place; then the ALU is shifted right one more place prior to storage.

Byte Swap of R_0

$A = B = 0$ $I = 701$ $RAM_0 RI = RAM_{15} RO$ $C_{IN} = C_{OUT}$

Repeat 4 times

2. Instruction Fetch Cycle

Execution of a macroinstruction generally begins with an instruction fetch cycle. The current contents of the PC (in one of the registers) is the address of the macroinstruction to be fetched, and must be read out to the

memory address register. Then the PC is incremented to point to the next macroinstruction. The macroinstruction obtained from memory is then loaded into the Am2909 microprogram sequencer to cause a jump to the microcode for executing the instruction.

The PC can be read out and incremented in one cycle by using the Am2901 destination code 2, and addressing the PC with both the A and B addresses. The current value of PC will appear on the Y outputs, and PC+1 will be returned to the register. If the PC is in register 15, then:

$A = B = 15$, $I = 203$, $Carry-in = 1$

The PC will be on the Y outputs via the RAM A-port. On the clock LOW-to-HIGH transition, the program counter is incremented and the value on the Y outputs is loaded into the memory address register. During the following cycle, the memory is read and, on the next clock LOW-to-HIGH transition the instruction from the memory is dropped into the Am2909 instruction register. The fetch operation requires only two microcycles.

Am2909

Microprogram Sequencer
Advanced Micro Devices
Bipolar Microprocessor Circuits



DISTINCTIVE CHARACTERISTICS

- 4-bit slice cascadable to any number of microwords
- Internal address register
- Branch input for N-way branches
- Cascadable 4-bit microprogram counter
- 4 x 4 file with stack pointer and push/pop control for nesting microsubroutines.
- Zero input for returning to the zero microcode word
- Individual OR input for each bit for branching to higher microinstructions
- Three-state outputs
- All internal registers change state on the LOW-to-HIGH transition of the clock

GENERAL DESCRIPTION

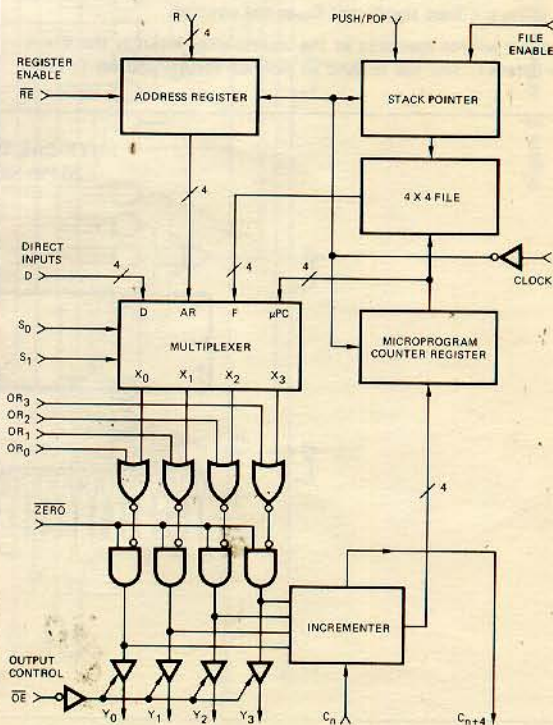
The Am2909 is a four-bit wide address controller intended for sequencing through a series of microinstructions contained in a ROM or PROM. Two Am2901's may be interconnected to generate an eight-bit address (256 words), and three may be used to generate a twelve-bit address (4K words).

The Am2909 can select an address from any of four sources. They are: 1) a set of external direct inputs (D); 2) external data from the R inputs, stored in an internal register; 3) a four-word deep push/pop stack; or 4) a program counter register (which usually contains the last address plus one). The push/pop stack includes certain control lines so that it can efficiently execute nested subroutine linkages. Each of the four outputs can be OR'ed with an external input for conditional skip or branch instructions, and a separate line forces the outputs to all zeroes. The outputs are three-state.

TABLE OF CONTENTS

Block Diagram	20
Pin Definitions	21
Connection Diagram	21
Physical Dimensions	19
Function Tables	22
Subroutining	23
Screening	24
Order Codes	24
DC Characteristics	25
AC Characteristics	26

MICROPROGRAM SEQUENCER BLOCK DIAGRAM



ARCHITECTURE OF THE Am2909

The Am2909 is a bipolar microprogram sequencer intended for use in high-speed microprocessor applications. The device is a cascadable 4-bit slice such that two devices allow addressing of up to 256-words of microprogram and three devices allow addressing of up to 4K words of microprogram. A detailed logic diagram is shown in Figure 2.

The device contains a four-input multiplexer that is used to select either the address register, direct inputs, microprogram counter, or file as the source of the next microinstruction address. This multiplexer is controlled by the S_0 and S_1 inputs.

The address register consists of four D-type, edge-triggered flip-flops with a common clock enable. When the address register enable is LOW, new data is entered into the register on the clock LOW-to-HIGH transition. The address register is available at the multiplexer as a source for the next microinstruction address. The direct input is a four-bit field of inputs to the multiplexer and can be selected as the next microinstruction address. This allows an N-way branch where N is any word in the microcode.

The Am2909 contains a microprogram counter (μ PC) that is composed of a 4-bit incrementer followed by a 4-bit register. The incrementer has carry-in (C_n) and carry-out (C_{n+4}) such that cascading to larger word lengths is straight-forward. The μ PC can be used in either of two ways. When the least significant carry-in to the incrementer is HIGH, the microprogram register is loaded on the next clock cycle with the current Y output word plus one ($Y+1 \rightarrow \mu$ PC.) Thus sequential microinstructions can be executed. If this least significant C_n is LOW, the incrementer passes the Y output word unmodified and the microprogram register is loaded with the same Y word on the next clock cycle ($Y \rightarrow \mu$ PC.) Thus, the same microinstruction can be executed any number of times by using the least significant C_n as the control.

The last source available at the multiplexer input is the 4 x 4 file (stack). The file is used to provide return address linkage

when executing microsubroutines. The file contains a built-in stack pointer (SP) which always points to the last file word written. This allows stack reference operations (looping) to be performed without a push or pop.

The stack pointer operates as an up/down counter with separate push/pop and file enable inputs. When the file enable input is LOW and the push/pop input is HIGH, the PUSH operation is enabled. This causes the stack pointer to increment and the file to be written with the required return linkage — the next microinstruction address following the subroutine jump which initiated the PUSH.

If the file enable input is LOW and the push/pop control is LOW, a POP operation occurs. This implies the usage of the return linkage during this cycle and thus a return from subroutine. The next LOW-to-HIGH clock transition causes the stack pointer to decrement. If the file enable is HIGH, no action is taken by the stack pointer regardless of any other input.

The stack pointer linkage is such that any combination of pushes, pops or stack references can be achieved. One microinstruction subroutines can be performed. Since the stack is 4 words deep, up to four microsubroutines can be nested.

The ZERO input is used to force the four outputs to the binary zero state. When the ZERO input is LOW, all Y outputs are LOW regardless of any other inputs (except \overline{OE}). Each Y output bit also has a separate OR input such that a conditional logic one can be forced at each Y output. This allows jumping to different microinstructions on programmed conditions.

The Am2909 features three-state Y outputs. These can be particularly useful in military designs requiring external Ground Support Equipment (GSE) to provide automatic checkout of the microprocessor. The internal control can be placed in the high-impedance state, and preprogrammed sequences of microinstructions can be executed via external access to the control ROM/PROM.

PHYSICAL DIMENSIONS 28-Pin Side Brazed

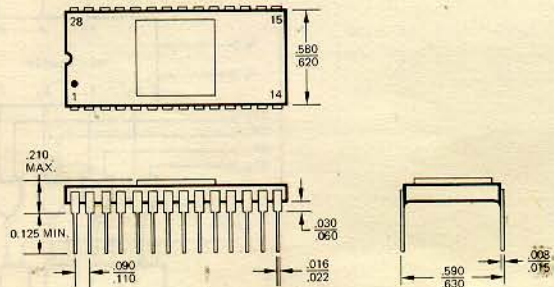


Figure 1.

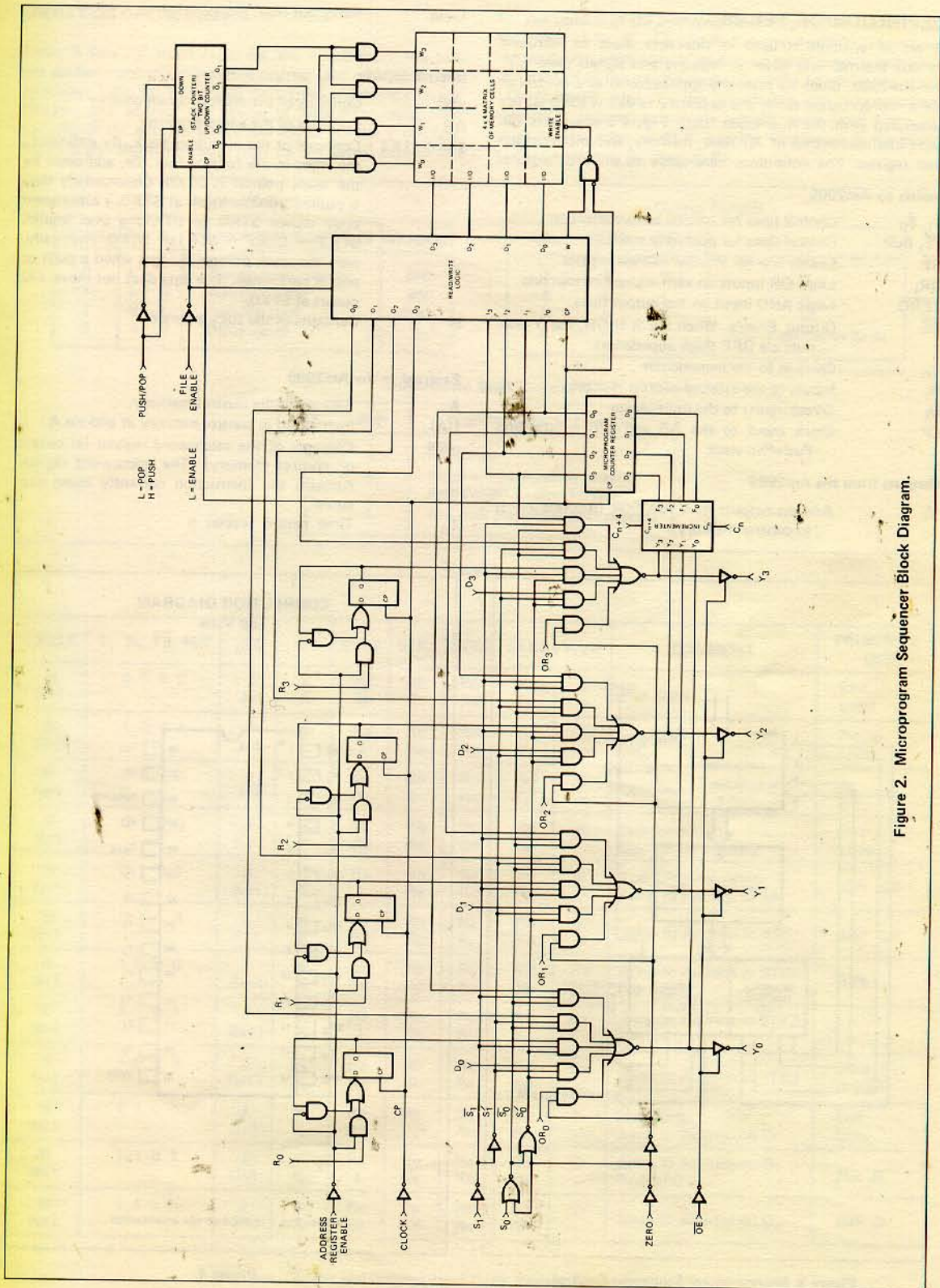


Figure 2. Microprogram Sequencer Block Diagram.

DEFINITION OF TERMS

A set of symbols is used in this data sheet to represent various internal and external registers and signals used with the Am2909. Since its principle application is as a controller for a microprogram store, it is necessary to define some signals associated with the microcode itself. Figure 3 illustrates the basic interconnection of Am2909, memory, and microinstruction register. The definitions here apply to this architecture.

Inputs to Am2909

S_1, S_0	Control lines for address source selection
\overline{FE}, PUP	Control lines for push/pop stack
\overline{RE}	Enable line for internal address register
OR_i	Logic OR inputs on each address output line
\overline{ZERO}	Logic AND input on the output lines
\overline{OE}	Output Enable. When \overline{OE} is HIGH, the Y outputs are OFF (high impedance)
C_n	Carry-in to the incrementer
R_i	Inputs to the internal address register
D_i	Direct inputs to the multiplexer
CP	Clock input to the AR and μPC register and Push-Pop stack

Outputs from the Am2909

Y_i	Address outputs from Am2909. (Address inputs to control memory.)
-------	--

C_{n+4} Carry out from the incrementer

Internal Signals

μPC	Contents of the microprogram counter
AR	Contents of the address register
$STK0-STK3$	Contents of the push/pop stack. By definition, the word in the four-by-four file, addressed by the stack pointer is $STK0$. Conceptually data is pushed into the stack at $STK0$; a subsequent push moves $STK0$ to $STK1$; a pop implies $STK3 \rightarrow STK2 \rightarrow STK1 \rightarrow STK0$. Physically, only the stack pointer changes when a push or pop is performed. The data does not move. I/O occurs at $STK0$.
SP	Contents of the stack pointer

External to the Am2909

A	Address to the control memory
$I(A)$	Instruction in control memory at address A
μWR	Contents of the microword register (at output of control memory). The microword register contains the instruction currently being executed.
T_n	Time period (cycle) n

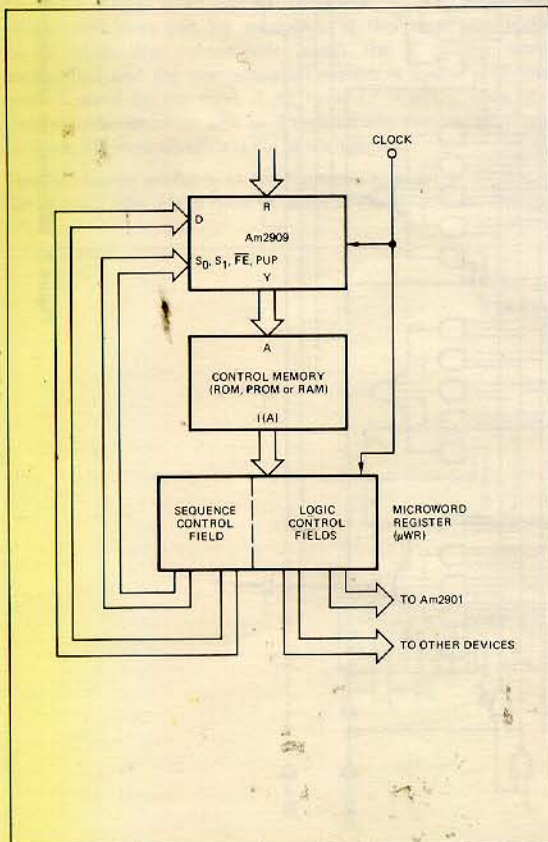


Figure 3. Microprogram Sequencer Control.

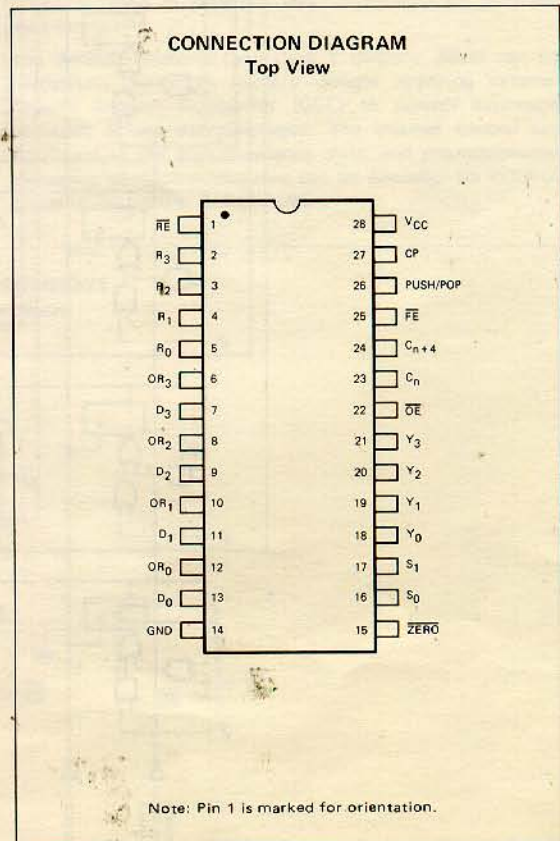


Figure 4.

OPERATION OF THE Am2909

Figure 5 lists the select codes for the multiplexer. The two bits applied from the microword register (and additional combinational logic for branching) determine which data source contains the address for the next microinstruction. The contents of the selected source will appear on the Y outputs. Figure 5 also shows the truth table for the output control and

for the control of the push/pop stack. Figure 6 shows in detail the effect of S_0 , S_1 , \overline{FE} and PUP on the Am2909. These four signals define what address appears on the Y outputs and what the state of all the internal registers will be following the clock LOW-to-HIGH edge. In this illustration, the microprogram counter is assumed to contain initially some word J, the address register some word K, and the four words in the push/pop stack contain R_a through R_d .

Address Selection				Output Control				
OCTAL	S_1	S_0	SOURCE FOR Y OUTPUTS	SYMBOL	OR_i	ZERO	OE	Y_i
0	L	L	Microprogram Counter	μPC	X	X	H	Z
1	L	H	Address register	AR	X	L	L	L
2	H	L	Push-Pop stack	STK0	H	H	L	H
3	H	H	Direct inputs	D_i	L	H	L	Source selected by $S_0 S_1$

Z = High Impedance

Synchronous Stack Control		
\overline{FE}	PUP	PUSH-POP STACK CHANGE
H	X	No change
L	H	Increment stack pointer, then push current PC onto STK0
L	L	Pop stack (decrement stack pointer)

Figure 5.

CYCLE	$S_1, S_0, \overline{FE}, PUP$	μPC	AR	STK0	STK1	STK2	STK3	Y_{OUT}	COMMENT	PRINCIPLE USE
N N+1	0 0 0 0 —	J J+1	K K	R_a R_b	R_b R_c	R_c R_d	R_d R_a	J —	Pop Stack	End Loop
N N+1	0 0 0 1 —	J J+1	K K	R_a J	R_b R_a	R_c R_b	R_d R_c	J —	Push μPC	Set-up Loop
N N+1	0 0 1 X —	J J+1	K K	R_a R_a	R_b R_b	R_c R_c	R_d R_d	J —	Continue	Continue
N N+1	0 1 0 0 —	J K+1	K K	R_a R_b	R_b R_c	R_c R_d	R_d R_a	K —	Pop Stack; Use AR for Address	End Loop
N N+1	0 1 0 1 —	J K+1	K K	R_a J	R_b R_a	R_c R_b	R_d R_c	K —	Push μPC ; Jump to Address in AR	JSR AR
N N+1	0 1 1 X —	J K+1	K K	R_a R_a	R_b R_b	R_c R_c	R_d R_d	K —	Jump to Address in AR	JMP AR
N N+1	1 0 0 0 —	J R_a+1	K K	R_a R_b	R_b R_c	R_c R_d	R_d R_a	R_a —	Jump to Address in STK0; Pop Stack	RTS
N N+1	1 0 0 1 —	J R_a+1	K K	R_a J	R_b R_a	R_c R_b	R_d R_c	R_a —	Jump to Address in STK0; Push μPC	
N N+1	1 0 1 X —	J R_a+1	K K	R_a R_a	R_b R_b	R_c R_c	R_d R_d	R_a —	Jump to Address in STK0	Stack Ref (Loop)
N N+1	1 1 0 0 —	J D+1	K K	R_a R_b	R_b R_c	R_c R_d	R_d R_a	D —	Pop Stack; Jump to Address on D	End Loop
N N+1	1 1 0 1 —	J D+1	K K	R_a J	R_b R_a	R_c R_b	R_d R_c	D —	Jump to Address on D; Push μPC	JSR D
N N+1	1 1 1 X —	J D+1	K K	R_a R_a	R_b R_b	R_c R_c	R_d R_d	D —	Jump to Address on D	JMP D

X = Don't Care, 0 = LOW, 1 = HIGH, Assume C_n = HIGH

Figure 6. Output and Internal Next-Cycle Register States for Am2909.

Figure 7 illustrates the execution of a subroutine using the Am2909. The configuration of Figure 3 is assumed. The instruction being executed at any given time is the one contained in the microword register (μ WR). The contents of the μ WR also controls (indirectly, perhaps) the four signals S_0 , S_1 , FE , and PUP. The starting address of the subroutine is applied to the D inputs of the Am2909 at the appropriate time.

In the columns on the left is the sequence of microinstructions to be executed. At address J+2, the sequence control portion of the microinstruction contains the command "Jump to sub-

routine at A". At the time T_2 , this instruction is in the μ WR, and the Am2909 inputs are set-up to execute the jump and save the return address. The subroutine address A is applied to the D inputs from the μ WR and appears on the Y outputs. The first instruction of the subroutine, I(A), is accessed and is at the inputs of the μ WR. On the next clock transition, I(A) is loaded into the μ WR for execution, and the return address J+3 is pushed onto the stack. The return instruction is executed at T_5 . Figure 8 is a similar timing chart showing one subroutine linking to a second, the latter consisting of only one microinstruction.

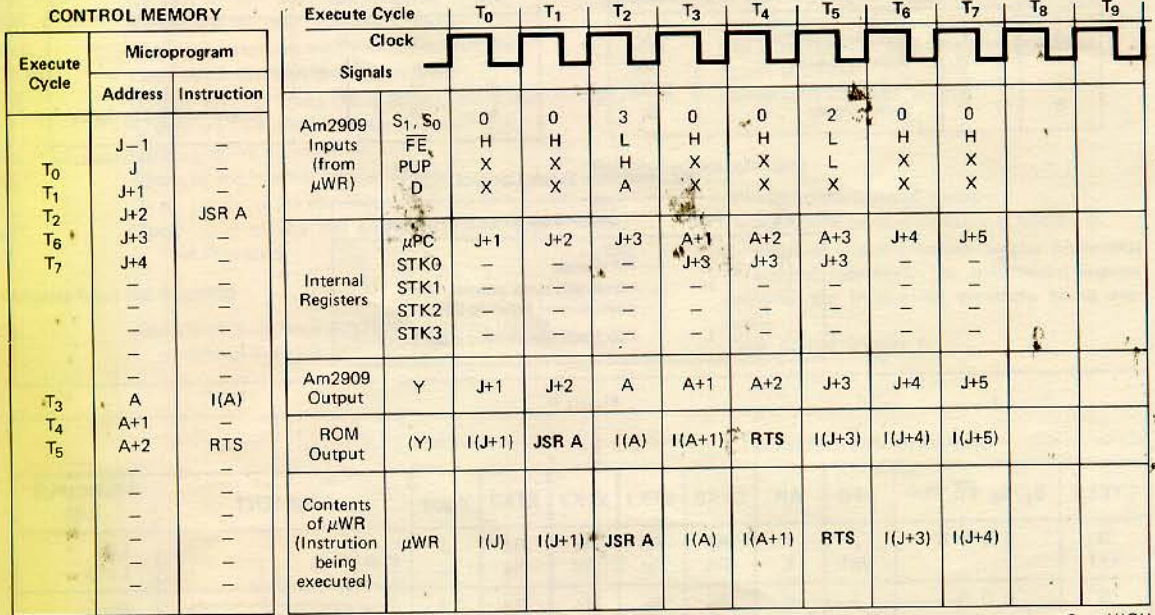


Figure 7. Subroutine Execution.

C_n = HIGH

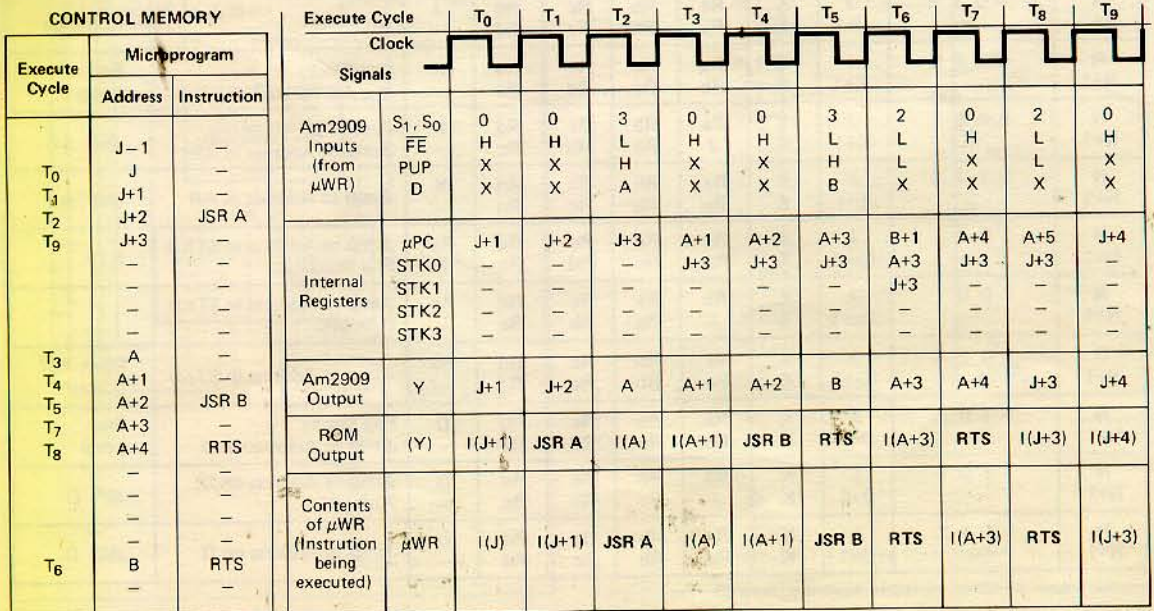


Figure 8. Two Nested Subroutines. Routine B is Only One Instruction.

C_n = HIGH

MAXIMUM RATINGS (Above which the useful life may be impaired)

Storage Temperature	-65°C to +150°C
Temperature (Ambient) Under Bias	-55°C to +125°C
Supply Voltage to Ground Potential	-0.5 V to +7.0 V
DC Voltage Applied to Outputs for HIGH Output State	-0.5 V to +V _{CC} max.
DC Input Voltage	-0.5 V to +7.0 V
DC Output Current, Into Outputs	30 mA
DC Input Current	-30 mA to +5.0 mA

OPERATING RANGE

P/N	Ambient Temperature	V _{CC}
Am2909PC, DC	0°C to +70°C	4.75 V to 5.25 V
Am2909DM, FM	-55°C to +125°C	4.50 V to 5.50 V

STANDARD SCREENING
 (Conforms to MIL-STD-883 for Class C Parts)

Step	MIL-STD-883 Method	Conditions	Level	
			Am2909PC, DC	Am2909DM, FM
Pre-Seal Visual Inspection	2010	B	100%	100%
Stabilization Bake	1008	C 24-hour 150°C	100%	100%
Temperature Cycle	1010	C -65°C to +150°C 10 cycles	100%	100%
Centrifuge	2001	B 10,000 G	100% *	100%
Fine Leak	1014	A 5 x 10 ⁻⁸ atm-cc/cm ³	100% *	100%
Gross Leak	1014	C2 Fluorocarbon	100% *	100%
Electrical Test Subgroups 1 and 7	5004	See below for definitions of subgroups	100%	100%
Insert Additional Screening here for Class B Parts				
Group A Sample Tests				
Subgroup 1	5005	See below for definitions of subgroups	LTPD = 5	LTPD = 5
Subgroup 2			LTPD = 7	LTPD = 7
Subgroup 3			LTPD = 7	LTPD = 7
Subgroup 7			LTPD = 7	LTPD = 7
Subgroup 8			LTPD = 7	LTPD = 7
Subgroup 9			LTPD = 7	LTPD = 7

* Not applicable for Am2909PC

ADDITIONAL SCREENING FOR CLASS B PARTS

Step	MIL-STD-883 Method	Conditions	Level
			Am2909DMB, FMB
Burn-In	1015	D 125°C 160 hours min.	100%
Electrical Test Subgroup 1 Subgroup 2 Subgroup 3 Subgroup 7 Subgroup 9	5004		100% 100% 100% 100% 100%
Return to Group A Tests in Standard Screening			

ORDERING INFORMATION

Package Type	Temperature Range	Order Number
Molded DIP	0°C to +70°C	AM2909PC
Hermetic DIP	0°C to +70°C	AM2909DC
Hermetic DIP	-55°C to +125°C	AM2909DM
Hermetic Flat Pack	-55°C to +125°C	AM2909FM
Dice	0°C to +70°C	AM2909XC

GROUP A SUBGROUPS

(as defined in MIL-STD-883, method 5005)

Subgroup	Parameter	Temperature
1	DC	25°C
2	DC	Maximum rated temperature
3	DC	Minimum rated temperature
7	Function	25°C
8	Function	Maximum and minimum rated temperature
9	Switching	25°C
10	Switching	Maximum Rated Temperature
11	Switching	Minimum Rated Temperature

ELECTRICAL CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (Unless Otherwise Noted)

Am2909XC $T_A = 0^\circ\text{C to } +70^\circ\text{C}$ $V_{CC} = 5.0\text{V} \pm 5\%$ (COM'L) MIN. = 4.75V MAX. = 5.25V
 Am2909XM $T_A = -55^\circ\text{C to } +125^\circ\text{C}$ $V_{CC} = 5.0\text{V} \pm 10\%$ (MIL) MIN. = 4.50V MAX. = 5.50V

Parameters	Description	Test Conditions (Note 1)		Min.	Typ. (Note 2)	Max.	Units	
V _{OH}	Output HIGH Voltage	V _{CC} = MIN., V _{IN} = V _{IH} or V _{IL}	MIL	I _{OH} = -1.0mA	2.4		Volts	
			COM'L	I _{OH} = -2.6mA	2.4			
V _{OL}	Output LOW Voltage	V _{CC} = MIN., V _{IN} = V _{IH} or V _{IL}	I _{OL} = 4.0mA			0.4	Volts	
			I _{OL} = 8.0mA			0.45		
			I _{OL} = 12mA (Note 5)			0.5		
V _{IH}	Input HIGH Level	Guaranteed input logical HIGH voltage for all inputs			2.0		Volts	
V _{IL}	Input LOW Level	Guaranteed input logical LOW voltage for all inputs		MIL		0.7	Volts	
				COM'L		0.8		
V _I	Input Clamp Voltage	V _{CC} = MIN., I _{IN} = -18mA				-1.5	Volts	
I _{IL}	Input LOW Current	V _{CC} = MAX., V _{IN} = 0.4V	C _n			-1.08	mA	
			Push/Pop, OE			-0.72		
			Others			-0.36		
I _{IH}	Input HIGH Current	V _{CC} = MAX., V _{IN} = 2.7V	C _n			40	μA	
			Push/Pop			40		
			Others			20		
I _I	Input HIGH Current	V _{CC} = MAX., V _{IN} = 7.0V	C _n , Push/Pop			0.2	mA	
			Others			0.1		
I _{SC}	Output Short Circuit Current (Note 3)	V _{CC} = MAX.			-30	-85	mA	
I _{CC}	Power Supply Current	V _{CC} = MAX. (Note 4)				80	130	mA
I _{OZ}	Output OFF Current	V _{CC} = MAX., OE = 2.7V	V _{OUT} = 0.4V			-20	μA	
			V _{OUT} = 2.7V			20		

- Notes: 1. For conditions shown as MIN. or MAX., use the appropriate value specified under Electrical Characteristics for the applicable device type.
 2. Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
 3. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
 4. Apply GND to C_n, R₀, R₁, R₂, R₃, OR₀, OR₁, OR₂, OR₃, D₀, D₁, D₂, and D₃. Other inputs open. All outputs open. Measured after a LOW-to-HIGH clock transition.
 5. The 12mA output applies only to Y₀, Y₁, Y₂ and Y₃.

TYPICAL Am2909 AC CHARACTERISTICS (All in ns)

TABLE I
CLOCK REQUIREMENTS

t_{pwL}	30
t_{pwH}	30

TABLE II
COMBINATORIAL
PROPAGATION DELAYS

OUTPUTS INPUTS	Y_i	C_{n+4}
\overline{OE}	15	—
\overline{ZERO}	20	30
OR_i	10	20
S_0, S_1	20	30
D_i	10	20
C_n	—	10

TABLE III
CLOCK CONTROLLED
PROPAGATION DELAYS

FUNCTIONAL PATH	CLOCK TO Y_i	CLOCK TO C_{n+4}
VIA INSTRUCTION REG	25	30
VIA μ PROGRAM COUNTER	25	30
VIA FILE-PUSH MODE	35	40
VIA FILE-POP MODE	40	45

TABLE IV
SET-UP AND HOLD TIME
REQUIREMENTS

EXTERNAL INPUTS	t_s	t_h
\overline{RE}	15	0
R_i	10	0
PUSH/POP	10	0
\overline{FE}	10	0
C_n	10	0
D_i	15	0
OR_i	15	0
S_0, S_1	25	0
\overline{ZERO}	25	0

$T_A = 25^\circ C$ $R_L = 2k\Omega$ $C_L = 15pF$

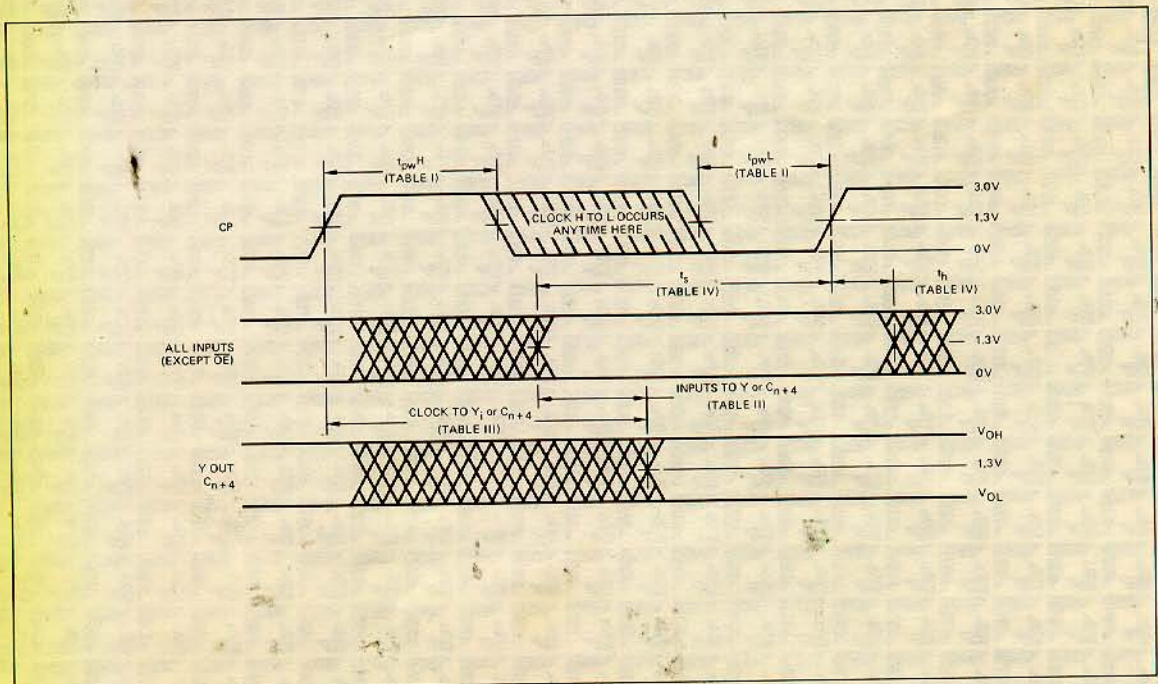
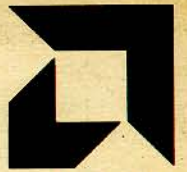


Figure 12. Switching Waveforms. See Tables for Specific Values.

Julia Springer

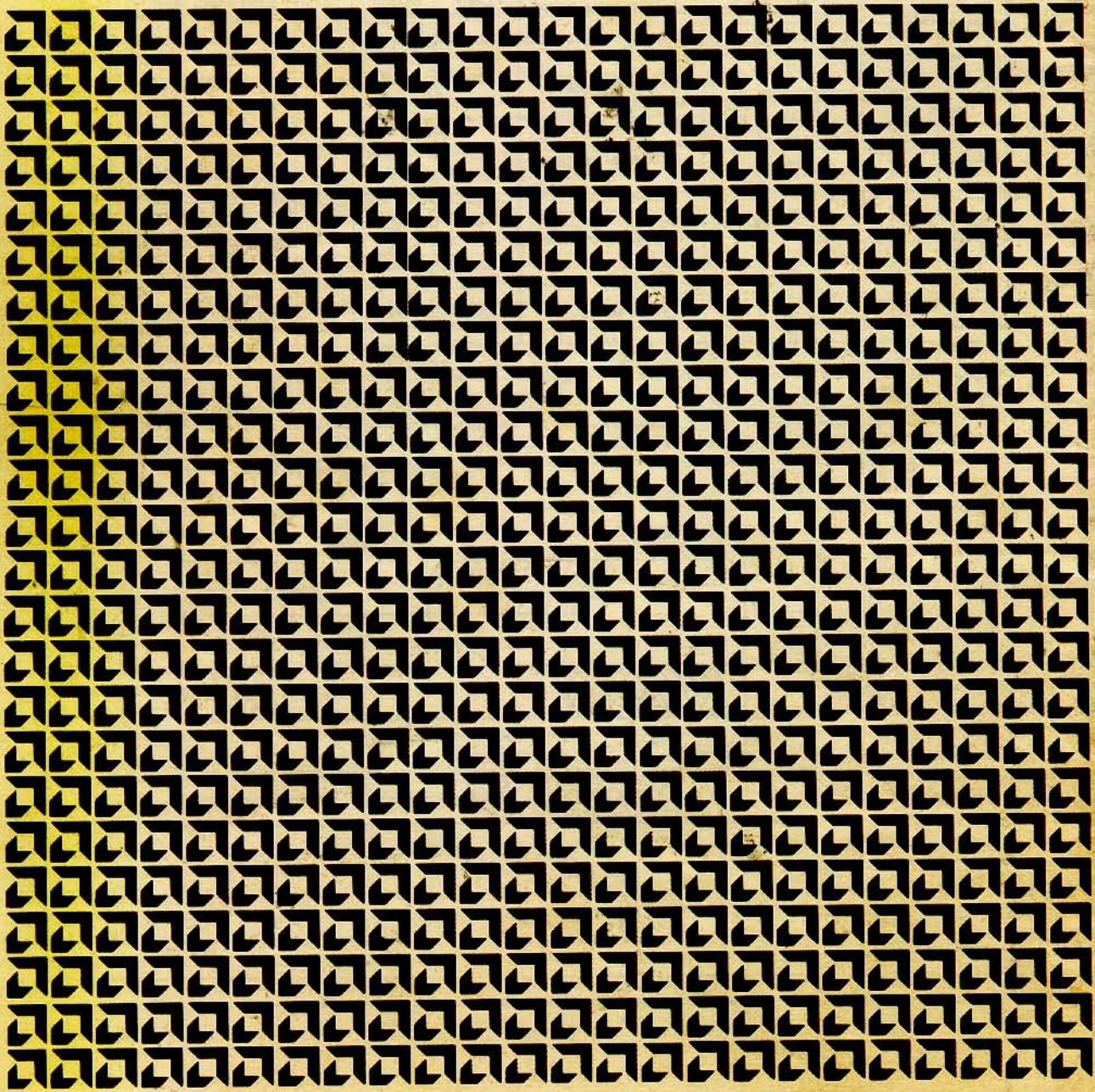


**ADVANCED
MICRO
DEVICES INC.**

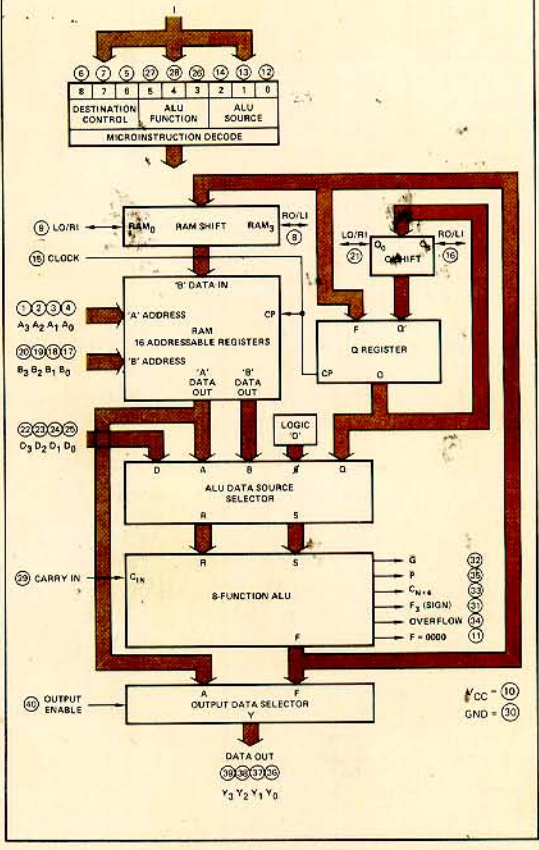
901 Thompson Place
Sunnyvale

California 94086
(408) 732-2400

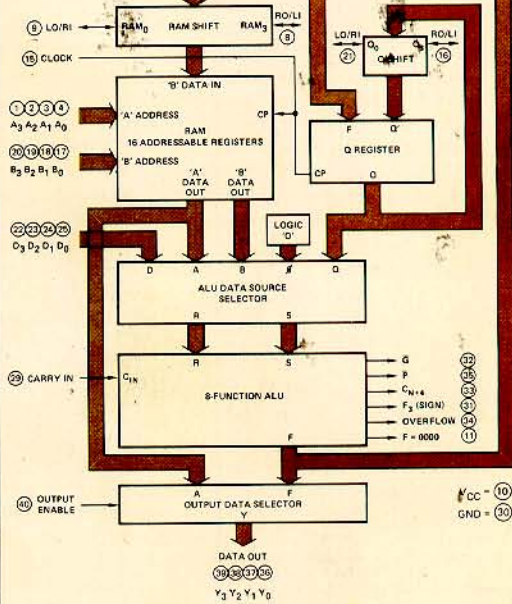
TWX: 910-339-9280
TELEX: 34-6306



Am2901 PIN CONNECTIONS



8	7	6	5	4	3	2	1	0
DESTINATION CONTROL			ALU FUNCTION			ALU SOURCE		
MICROINSTRUCTION DECODE								



DATA OUT
 Y₃ Y₂ Y₁ Y₀

Am2901 INSTRUCTION CODES

I ₈ I ₇ I ₆			I ₅ I ₄ I ₃			I ₂ I ₁ I ₀		
Destination			Function			Source		
LOAD	Y=		F			R	S	
0	F→Q	F	0	R+S*		0	A	Q
1	NOTHING	F	1	S-R		1	A	B
2	F→B	A	2	R-S		2	O	Q
3	F→B	F	3	RVS		3	O	B
4	F/2→B Q/2→Q	F	4	R^S		4	O	A
5	F/2→B	F	5	R^S		5	D	A
6	2F→B 2Q→Q	F	6	RVS		6	D	Q
7	2F→B	F	7	RVS		7	D	O