

# 9405

## ARITHMETIC LOGIC REGISTER STACK

FAIRCHILD MACROLOGIC™ TTL

**DESCRIPTION** — The Arithmetic Logic Register Stack (ALRS) is designed to implement general registers in high performance programmable digital systems. The device contains a 4-bit arithmetic logic unit (ALU), an 8-word by 4-bit RAM, and associated control logic. The ALU implements eight arithmetic and logic functions where one 4-bit operand is supplied from an external source (input data bus) and the second 4-bit operand is supplied internally from one of the eight RAM words selected by the Address Inputs ( $A_0$ - $A_2$ ). The result of the operation is loaded into the same RAM location and simultaneously, is loaded into the output register making it available at the 3-state output data bus.

The 9405 operates on four bits of data but features are provided for expansion to longer word lengths. Carry Propagate and Carry Generate Outputs are provided for an external carry lookahead where maximum operating speed is required. In applications where high speed arithmetic is not needed, ripple expansion may also be implemented. The 9405 provides three status signals: Zero, Negative and Overflow. These qualify the result of an operation. The 9405 is a member of Fairchild's MACROLOGIC TTL family and is fully compatible with all TTL families.

- EIGHT GENERAL REGISTERS/ACCUMULATORS IN A SINGLE PACKAGE
- HIGH SPEED — 10 MHz MICROINSTRUCTION RATE
- EXPANDABLE IN MULTIPLES OF FOUR BITS
- PROVIDES FOR RIPPLE OR LOOKAHEAD CARRY
- IMPLEMENTS 64 MICROINSTRUCTIONS
- PROVIDES STATUS — ZERO, NEGATIVE, AND OVERFLOW
- 3-STATE OUTPUTS
- 24-PIN PACKAGE

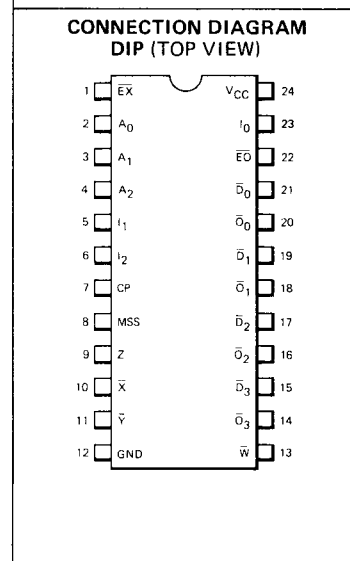
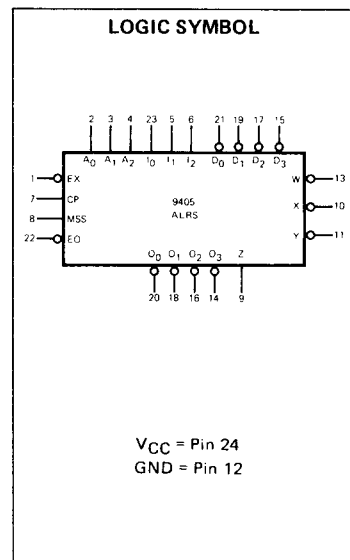
### PIN NAMES

$\bar{D}_0$ - $\bar{D}_3$	Data Inputs (Active LOW)
$A_0$ - $A_2$	Address Instruction Inputs
$I_0$ - $I_2$	ALU Instruction Inputs (Note b)
MSS	Most Significant Slice Input (Active HIGH)
CP	Clock Input
$\bar{E}\bar{O}$	Output Enable Input (Active LOW)
$\bar{E}\bar{X}$	Execute Input (Active LOW)
$Q_0$ - $Q_3$	Data Outputs (Active LOW)
$\bar{W}$	Ripple Carry Output (Active LOW) (Note c)
$\bar{X}$	Carry Propagate Output (Note d)
$\bar{Y}$	Carry Generate Output (Note e)
Z	Zero Status Output (Active HIGH, Open Collector) (Note f)

### NOTES:

- a) 1 Unit Load (U.L.) = 40  $\mu$ A HIGH, 1.6 mA LOW (0.5 V).
- b)  $I_0$  used also for Carry Input on lesser significant slices.
- c)  $\bar{W}$  Output also carries instruction information.
- d)  $\bar{X}$  Output provides Negative Status (active LOW) on most significant slice.
- e)  $\bar{Y}$  Output provides Overflow Status (active LOW) on most significant slice.
- f) An external pull-up resistor is required to supply HIGH level drive capability.

LOADING (Note a)	
HIGH	LOW
0.5 U.L.	0.23 U.L.
0.5 U.L.	0.23 U.L.
0.5 U.L.	0.23 U.L.
0.5 U.L.	0.23 U.L.
0.5 U.L.	0.23 U.L.
0.5 U.L.	0.23 U.L.
0.5 U.L.	0.23 U.L.
130 U.L.	10 U.L.
10 U.L.	5 U.L.
10 U.L.	5 U.L.
10 U.L.	10 U.L.
	5 U.L.



BLOCK DIAGRAM

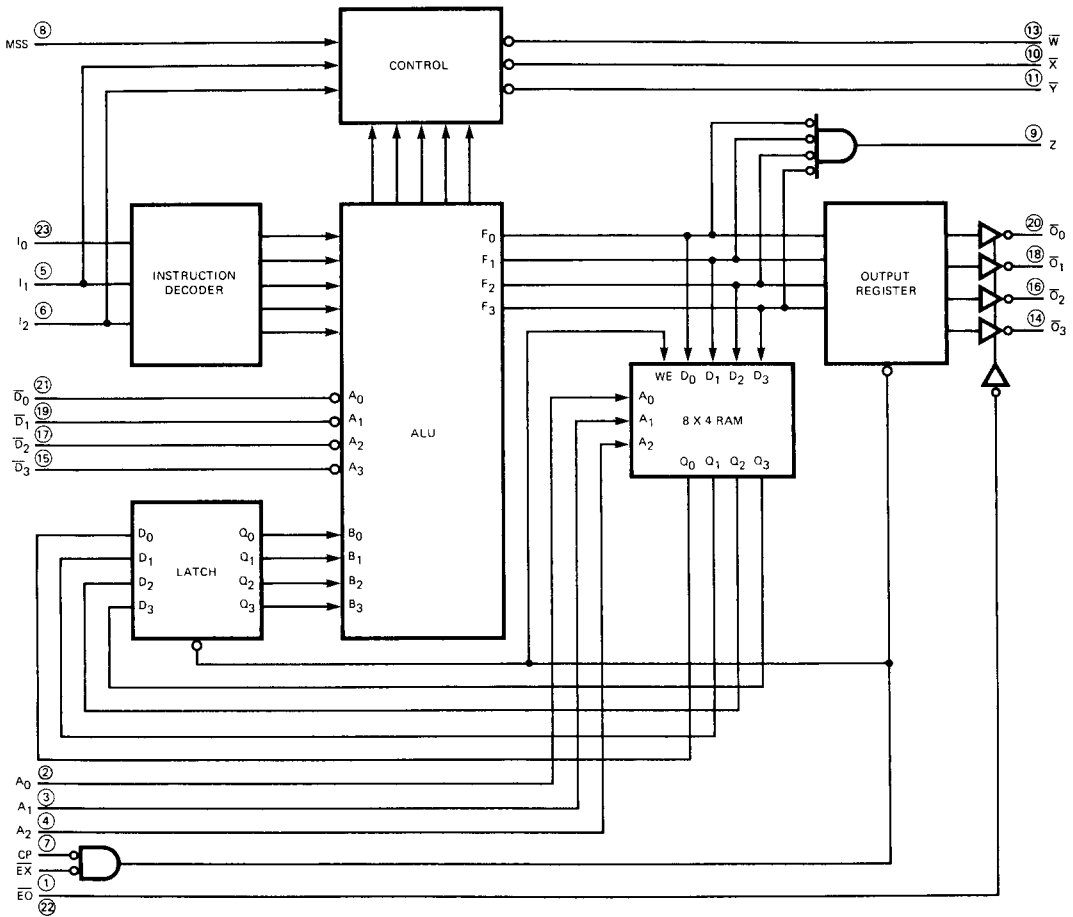


TABLE 1  
INSTRUCTION FIELD ASSIGNMENT

I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	INTERNAL OPERATION	
L L L	Rx plus D-Bus plus 1 → Rx	Accumulate
L L H	Rx plus D-Bus → Rx	Accumulate
L H L	Rx · D-Bus → Rx	Logic AND
L H H	D-Bus → Rx	Load
H L L	Rx → Output Register	Output
H L H	Rx + D-Bus →	Logic OR
H H L	Rx ⊕ D-Bus → Rx	Exclusive OR
H H H	D-Bus → Rx	Load Complement

NOTES:

1. Rx is the RAM location addressed by A<sub>0</sub>-A<sub>2</sub>.
2. The result of any operation is always loaded into the Output Register.

**FUNCTIONAL DESCRIPTION** — As shown in the Block Diagram the 9405 Arithmetic Logic Register Stack (ALRS) consists of a 4-bit ALU, an 8-word by 4-bit RAM with output latches, an instruction decode network, control logic and a 4-bit Output Register.

The ALU receives the active LOW input data ( $\overline{D}_0\text{--}\overline{D}_3$ ) as one operand while the RAM provides the second operand through latches. The ALU output is stored in both the RAM and output register. The active LOW Output Data Bus ( $\overline{O}_0\text{--}\overline{O}_3$ ) is obtained from the output register through 3-state buffers. An active LOW Output Enable ( $\overline{EO}$ ) input controls these buffers; a HIGH level  $\overline{EO}$  disables the buffers (high impedance state).

The instruction bus for the 9405 consists of two fields, A and I;  $A_0\text{--}A_2$  specify the desired location of the RAM and  $I_0\text{--}I_2$  specify the desired function to be performed. Table 1 lists Instruction Field Code assignments. Thus, the 9405 provides eight registers ( $R_0\text{--}R_7$ ) and eight different operations may be performed on any of these registers. The  $I_0\text{--}I_2$  Inputs are decoded by the instruction decode network to generate necessary control signals for the ALU. The ALU also generates and transmits to the control logic the following signals: Carry Out, Carry Propagate, Carry Generate, Negative Status and Overflow Status. The control logic manipulates the status signals as a function of  $I_0\text{--}I_2$  and a control input MSS. A HIGH on the MSS Input declares the most significant slice in a 9405 array (the diode-input on MSS allows it to be tied directly to  $V_{CC}$ ). All devices, except the most significant 9405 should have a LOW level (ground) on the MSS Input. The control logic generates three device outputs,  $\overline{W}$ ,  $\overline{X}$  and  $\overline{Y}$  for arrayed operation. An all zero result from the ALU is decoded and presented at the open collector Zero Status (Z) Output.

The  $I_0$  input serves a dual purpose. For arithmetic instructions, it is used as the carry input and for non-arithmetic instructions it serves as an instruction input. This is possible because only two arithmetic instructions require carry. The dual purpose use of  $I_0$  plays an important role in 9405 expansion schemes.

**OPERATION** — The 9405 operates on a single clock. CP and  $\overline{EX}$  are inputs to a 2-input active LOW AND gate. A microcycle starts as the clock goes HIGH. For normal operation the Execute ( $\overline{EX}$ ) is LOW. Data is read from the RAM through enabled latches and applied as one operand to the ALU. Data inputs ( $\overline{D}_0\text{--}\overline{D}_3$ ) are applied to the ALU as the other operand and the operation as determined by instruction lines  $I_0\text{--}I_2$  is executed. When CP is LOW, the latches are disabled and the result of the operation is written back into the RAM provided that  $\overline{EX}$  is LOW. The A lines must obviously be held stable during this time. On the LOW-to-HIGH CP transition, the result of the operation is loaded into the output register and a new microcycle can start. If  $\overline{EX}$  is held HIGH, the operation selected by the I and A Inputs is performed, but the result is not written back into the RAM and is not clocked into the output register.

**9405 ARRAYS** — The 9405 is organized to operate on a 4-bit wide data bus but can easily be expanded for longer words. Expansion requires that carries from lesser significant slices be propagated towards the most significant slice. The 9405 provides full lookahead capability for high speed arithmetic. Appropriate Carry Generate ( $\overline{Y}$ ) and Carry Propagate ( $\overline{X}$ ) outputs are provided so that only one external carry lookahead generator is needed for every four 9405s. When speed is not a prime consideration, it is possible to implement ripple carry expansion.

In arrayed operation, it is common to bus the  $\overline{EX}$ , CP and  $\overline{EO}$  Inputs of all devices. The Z Output is open collector and is normally OR-tied with the other devices and to an external load resistor so that a HIGH level indicates a zero result from an operation in the array.

Figure 1 shows a ripple carry 16-bit wide array using four 9405s. The MSS input is tied to  $V_{CC}$  on the most significant slice (ALRS 4); the MSS inputs of the other devices are tied to ground. The instruction bus of this array consists of A-Field and I-Field. A-Field is obtained by connecting corresponding A inputs of all four devices. The  $I_0$  input of device 1 (i.e., least significant slice) in conjunction with the bussed  $I_1$ ,  $I_2$  Inputs forms the I-Field for the array. The  $I_0$  Inputs of devices 2, 3 and 4 are connected to the  $\overline{W}$  Outputs of devices 1, 2 and 3 respectively. The ALU network generates the carry propagate output. The control logic operates on this signal as a function of  $I_1$  and  $I_2$  to generate the  $\overline{W}$  Output. If both  $I_1$  and  $I_2$  are LOW (i.e., an arithmetic instruction), the  $\overline{W}$  Output is the carry output of that slice. In case of non-arithmetic instructions, it assumes the state of the  $I_0$  input. Thus, in Figure 1, if an arithmetic instruction is specified, carry propagates through the  $\overline{W}$  Output to  $I_0$  Input of the next higher significant slice. On the other hand, non-arithmetic instructions effectively connect all  $I_0$  Inputs together to form the I-Field for the array. The  $\overline{W}$  Output of device 4 is the carry output from the array. The control logic also generates  $\overline{X}$  and  $\overline{Y}$  Outputs which participate in expansion when full carry lookahead is required. These outputs are normally ignored in ripple expansion except for the most significant slice. In the most significant slice,  $\overline{X}$  and  $\overline{Y}$  correspond to Negative and Overflow status signals.

Thus,  $\overline{X}$  Output of device 4 is LOW, if the result of an operation has its most significant bit as "1" (i.e., negative result). Similarly a LOW level on  $\overline{Y}$  output of device 4 indicates that arithmetic overflow has occurred. If the two operands have the same sign and the result has opposite sign, then it is assumed that an overflow has occurred. It should be noted that  $\overline{W}$ ,  $\overline{X}$  and  $\overline{Y}$  are not controlled by EX or CP. Figure 2 shows a 16-bit array with full carry lookahead expansion. Implementation of the lookahead scheme requires the use of an external 93S42/74S182 in addition to the four 9405s in the array. Since device 1 is the least significant and device 4 is the most significant slice, the MSS Inputs of the first three devices are connected to ground while device 4 has a HIGH at this input. The A-Field for the array instruction bus is obtained by connecting corresponding A Inputs of all four devices. Bussed  $I_1$  and  $I_2$  Inputs together with the  $I_0$  Input of device 1 form the I-Field for the array. The  $I_0$  Inputs for devices 2, 3 and 4 are obtained from the 93S42/74S182 Carry Outputs ( $C_{n+x}$ ,  $C_{n+y}$ , and  $C_{n+z}$

respectively). Also the  $\bar{P}$  and  $\bar{G}$  Inputs of 93S42/74S182 are connected to  $\bar{X}$  and  $\bar{Y}$  Outputs of the 9405s as shown. The control logic in the 9405 (see Block Diagram) generates  $\bar{X}$  and  $\bar{Y}$  Outputs as a function of  $I_1, I_2$  and MSS Inputs as well as the Carry Generate and Carry Propagate Outputs of the ALU. If the MSS Input of a slice is LOW and an arithmetic instruction is specified, its  $\bar{X}$  Output reflects Carry Propagate and  $\bar{Y}$  reflects Carry Generate Outputs from that slice. For an arithmetic instruction the  $I_0$  Input is treated as carry-in into a slice irrespective of MSS. Thus, whenever  $I_1$  and  $I_2$  are LOW, the array behaves as an adder with full carry lookahead. The  $\bar{W}$  Outputs still reflect carry output, which is ignored for devices 1, 2 and 3. The  $\bar{W}$  Output of device 4 is the carry output from the array. Also, note that the  $I_0$  Input of device 1 is not only an instruction input but also provides the carry input to the array so the  $I_0$  Input of device 1 must be connected to the appropriate 93S42/74S182 input as shown.

When a non-arithmetic instruction is specified to the array, the control logic of the 9405 forces a LOW on  $\bar{X}$  and a HIGH on  $\bar{Y}$  Outputs on all except the most significant slice. An examination of the 93S42/74S182 logic reveals that whenever  $\bar{P}$  is LOW and  $\bar{G}$  is HIGH the associated carry output is the same as the carry input. Thus, in Figure 2 devices 2, 3, and 4 will assume the logic level as that presented to the  $I_0$  Input of device 1 during non-arithmetic instructions effectively bussing  $I_0$  through all four devices. As in the case of ripple expansion  $\bar{X}$  and  $\bar{Y}$  Outputs of device 4 represent Negative and Overflow from the array.

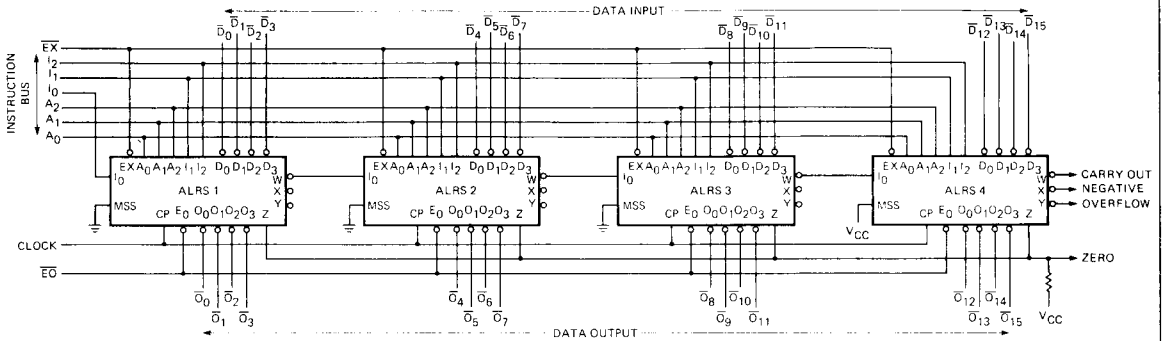


Fig. 1 RIPPLE CARRY EXPANSION

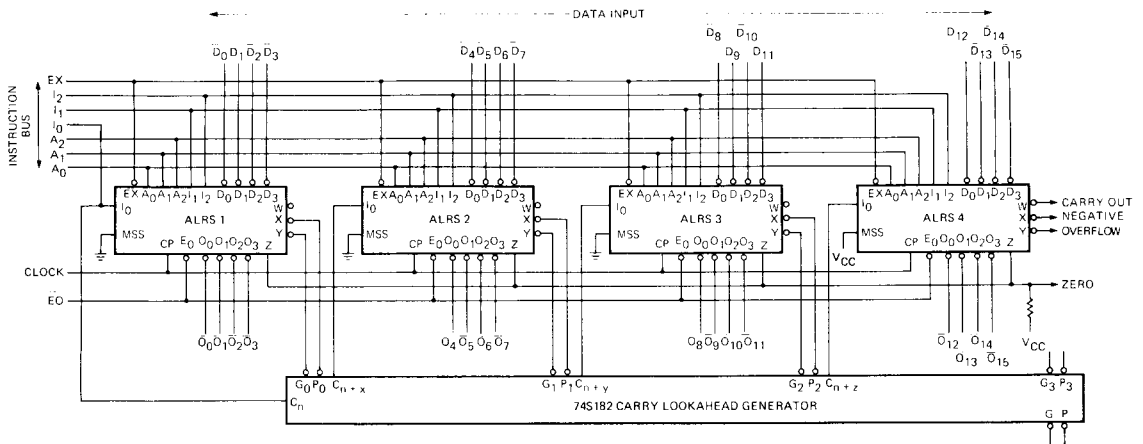


Fig. 2. CARRY LOOKAHEAD EXPANSION

**DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE** (unless otherwise noted)

SYMBOL	PARAMETER		LIMITS			UNITS	TEST CONDITIONS (Note 1)
			MIN	TYP	MAX		
V <sub>IH</sub>	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage
V <sub>IL</sub>	Input LOW Voltage	XM			0.7	V	Guaranteed Input LOW Voltage
		XC			0.8		
V <sub>CD</sub>	Input Clamp Diode Voltage			-0.9	-1.5	V	V <sub>CC</sub> = MIN, I <sub>IN</sub> = -18 mA
V <sub>OH</sub>	Output HIGH Voltage W, X Outputs	XM	2.4	3.4		V	V <sub>CC</sub> = MIN, I <sub>OH</sub> = -400 μA
		XC	2.4	3.4			
V <sub>OH</sub>	Output HIGH Voltage O <sub>0</sub> , O <sub>1</sub> , O <sub>2</sub> , O <sub>3</sub>	XM	2.4	3.4		V	I <sub>OH</sub> = -2.0 mA
		XC	2.4	3.1			I <sub>OH</sub> = -5.7 mA
I <sub>OH</sub>	Output HIGH Current Z Output				100	μA	V <sub>CC</sub> = MIN, V <sub>OH</sub> = 5.5 V
V <sub>OL</sub>	Output LOW Voltage W, X, Z		0.3	0.4		V	V <sub>CC</sub> = MIN, I <sub>OL</sub> = 4.0 mA
			0.4	0.5		V	V <sub>CC</sub> = MIN, I <sub>OL</sub> = 8.0 mA
V <sub>OL</sub>	Output LOW Voltage O <sub>0</sub> , O <sub>1</sub> , O <sub>2</sub> , O <sub>3</sub> , Y		0.3	0.4		V	V <sub>CC</sub> = MIN, I <sub>OL</sub> = 8.0 mA
			0.4	0.5		V	V <sub>CC</sub> = MIN, I <sub>OL</sub> = 16 mA
I <sub>OZH</sub>	Output Off Current HIGH			100		μA	V <sub>CC</sub> = MAX, V <sub>OUT</sub> = 2.4 V, V <sub>E</sub> = 2 V
I <sub>OZL</sub>	Output Off Current LOW			-100		μA	V <sub>CC</sub> = MAX, V <sub>OUT</sub> = 0.5 V, V <sub>E</sub> = 2 V
I <sub>IH</sub>	Input HIGH Current			1.0	40	μA	V <sub>CC</sub> = MAX, V <sub>IN</sub> = 2.7 V
					1.0		mA
I <sub>IL</sub>	Input LOW Current				-0.36	mA	V <sub>CC</sub> = MAX, V <sub>IN</sub> = 0.4 V
I <sub>OS</sub>	Output Short Circuit Current		-30		-100	mA	V <sub>CC</sub> = MAX, V <sub>OUT</sub> = 0 V (Note 3)
I <sub>CC</sub>	Supply Current			100	160	mA	V <sub>CC</sub> = MAX, Inputs Open

**NOTES:**

- For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.
- Typical limits are at V<sub>CC</sub> = 5.0 V, T<sub>A</sub> = 25°C.
- Not more than one output should be shorted at a time.

**AC SET-UP REQUIREMENTS:** V<sub>CC</sub> = 5.0 V, T<sub>A</sub> = 25°C, C<sub>L</sub> = 15 pF, See Fig. 3

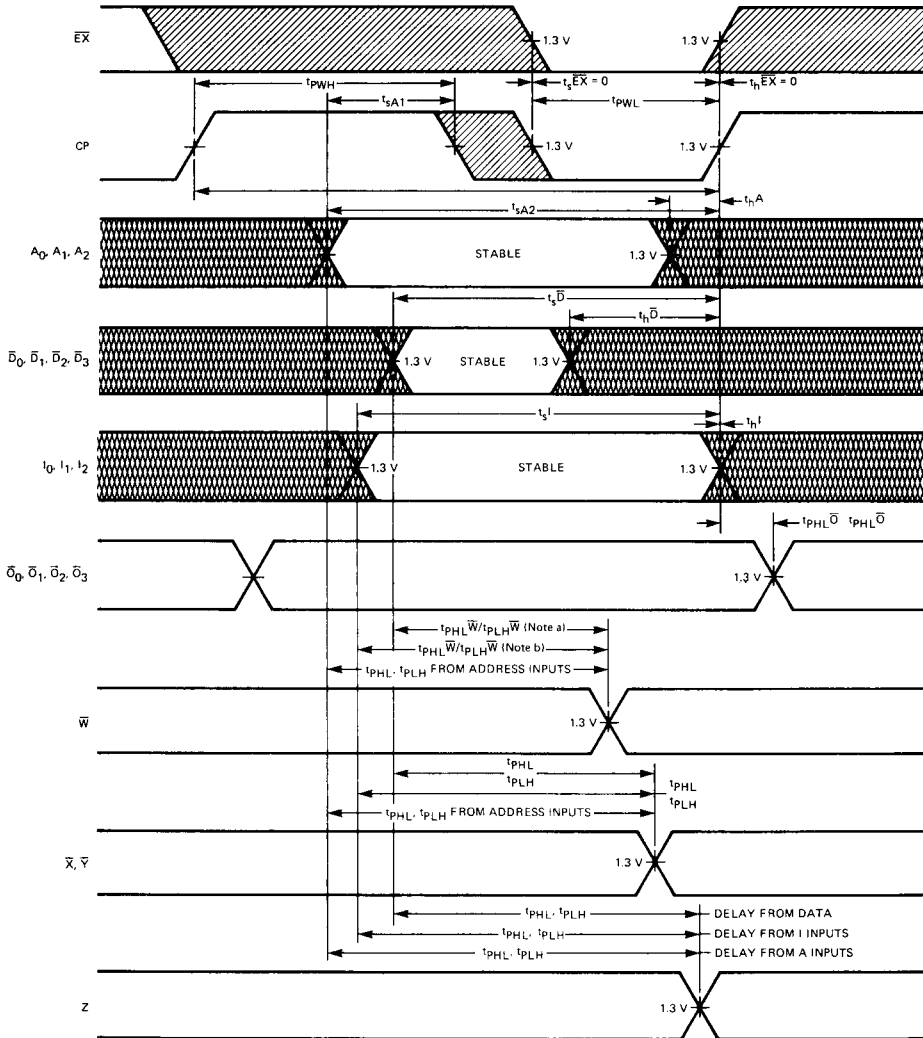
SYMBOL	PARAMETER		LIMITS			UNITS	TEST CONDITIONS
			MIN	TYP	MAX		
t <sub>CW</sub>	Clock Period			75		ns	
t <sub>PWH</sub>	Clock Pulse Width (HIGH)			30			
t <sub>PWL</sub>	Clock Pulse Width (LOW)			20			
t <sub>sEX</sub>	Set-Up Time, EX to CP			0			
t <sub>hEX</sub>	Hold Time, EX to CP			0			
t <sub>sA1</sub>	Set-Up Time, A <sub>0</sub> , A <sub>1</sub> , A <sub>2</sub> to Negative Going CP (Note 1)			25		ns	
t <sub>sA2</sub>	Set-Up Time, A <sub>0</sub> , A <sub>1</sub> , A <sub>2</sub> to Positive Going CP (Note 1)			70			
t <sub>hA</sub>	Hold Time, A <sub>0</sub> , A <sub>1</sub> , A <sub>2</sub> to Positive Going CP			5		ns	EX LOW
t <sub>sD</sub>	Set-Up Time, D <sub>0</sub> , D <sub>1</sub> , D <sub>2</sub> , D <sub>3</sub> to Positive Going CP			45			
t <sub>hD</sub>	Hold Time, D <sub>0</sub> , D <sub>1</sub> , D <sub>2</sub> , D <sub>3</sub> to Positive Going Clock		-20				
t <sub>sI</sub>	Set-Up Time, I <sub>0</sub> , I <sub>1</sub> , I <sub>2</sub> to Positive Going Clock			50			
t <sub>hI</sub>	Hold Time, I <sub>0</sub> , I <sub>1</sub> , I <sub>2</sub> to Positive Going Clock			0			

**NOTE:**

- Both set-up times must be met simultaneously.

AC CHARACTERISTICS:  $V_{CC} = 5.0 \text{ V}$ ,  $T_A = 25^\circ\text{C}$ ,  $C_L = 15 \text{ pF}$ , See Fig. 3

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
$t_{PLH}$ $t_{PHL}$	Propagation Delay, Positive Going CP to $\bar{O}_0, \bar{O}_1, \bar{O}_2, \bar{O}_3$		25		ns	$\bar{E}O, \bar{E}X$ LOW
$t_{PLH}$ $t_{PHL}$	Propagation Delay, $I_0$ to $\bar{W}$		15		ns	$I_1$ or $I_2$ HIGH
$t_{PLH}$ $t_{PHL}$	Propagation Delay, Data ( $\bar{D}_0, \bar{D}_1, \bar{D}_2, \bar{D}_3$ ) to $\bar{W}$		35		ns	$I_1, I_2$ LOW
$t_{PLH}$ $t_{PHL}$	Propagation Delay, Data ( $\bar{D}_0, \bar{D}_1, \bar{D}_2, \bar{D}_3$ ) to $\bar{X}, \bar{Y}$		50		ns	MSS HIGH
				25		ns
$t_{PLH}$ $t_{PHL}$	Propagation Delay, $I_1, I_2$ to $\bar{X}, \bar{Y}$		22		ns	MSS LOW
$t_{PLH}$ $t_{PHL}$	Propagation Delay, Data ( $\bar{D}_0, \bar{D}_1, \bar{D}_2, \bar{D}_3$ ) to Z		55		ns	1 k $\Omega$ External Load Resistor to $V_{CC}$
$t_{PLH}$ $t_{PHL}$	Propagation Delay, $I_0$ to $\bar{W}$		40		ns	$I_1, I_2$ LOW
$t_{PLH}$ $t_{PHL}$	Propagation Delay, $I_1, I_2$ to $\bar{W}$		15		ns	$I_1, I_2$ LOW
				40		
$t_{PLH}$ $t_{PHL}$	Propagation Delay, $\bar{D}_3$ to $\bar{X}$		50		ns	$I_1, I_2$ HIGH MSS HIGH
$t_{PLH}$ $t_{PHL}$	Propagation Delay, Address ( $A_0, A_1, A_2$ ) to $\bar{X}, \bar{Y}$		55		ns	$I_1, I_2$ LOW MSS LOW
$t_{PLH}$ $t_{PHL}$	Propagation Delay, Address ( $A_0, A_1, A_2$ ) to $\bar{X}, \bar{Y}$		70		ns	$I_1, I_2$ LOW MSS HIGH
$t_{PLH}$ $t_{PHL}$	Propagation Delay, Address ( $A_0, A_1, A_2$ ) to $\bar{X}$		70		ns	$I_1, I_2$ HIGH MSS HIGH
$t_{PLH}$ $t_{PHL}$	Propagation Delay, Address ( $A_0, A_1, A_2$ ) to $\bar{W}$		55		ns	$I_1, I_2$ LOW
$t_{PLH}$ $t_{PHL}$	Propagation Delay, Address ( $A_0, A_1, A_2$ ) to Z		70		ns	$I_1, I_2$ LOW
				20		
$t_{PLH}$ $t_{PHL}$	Propagation Delay, $I_1, I_2$ to $\bar{X}, \bar{Y}$		45		ns	$I_1, I_2$ LOW MSS HIGH
				50		
$t_{PLH}$ $t_{PHL}$	Propagation Delay, $I_0$ to $\bar{X}, \bar{Y}$		50		ns	$I_1, I_2$ LOW MSS HIGH
$t_{PLH}$ $t_{PHL}$	Propagation Delay, $I_1, I_2$ to Z		42		ns	$I_1, I_2$ LOW
$t_{PLH}$ $t_{PHL}$	Propagation Delay, $I_0$ to Z		25		ns	$I_1, I_2$ LOW
$t_{PZH}$ $t_{PZL}$	Enable Delay, $\bar{E}O$ to Outputs $\bar{O}_0, \bar{O}_1, \bar{O}_2, \bar{O}_3$		12		ns	
$t_{PLZ}$ $t_{PHZ}$	Disable Delay, $\bar{E}O$ to $\bar{O}_0, \bar{O}_1, \bar{O}_2, \bar{O}_3$		10		ns	



NOTES:

- a) Delay for logical operation ( $I_1$  or  $I_2$  HIGH)
- b) Delay for arithmetic operation ( $I_1 = I_2 = \text{LOW}$ )

Fig. 3 ALRS TIMING DIAGRAM