# Am29C116/116-1/116-2

## 16-Bit CMOS Microprocessors

**Advanced Micro Devices**

## DISTINCTIVE CHARACTERISTICS

- **Am29C116**
  Supports up to 100-ns system cycle time. Less than 1-watt power dissipation and equivalent performance to the bipolar Am29116.
- **Am29C116-1**
  Faster, speed-select version of the Am29C116 (90 ns).
- **Am29C116-2**
  Can operate with 80-ns clock cycle.
- **Pin-Compatible and Functionally Equivalent to the Am29116**
  The architecture, instruction set, and pin-out are completely identical to the bipolar Am29116.

- **Optimized for High-Performance Controllers**
  The architecture is optimized for controllers providing an excellent solution for applications requiring bit-manipulation power.
- **Powerful Field Insertion/Extraction and Bit-Manipulation Instructions**
  Rotate-and-Merge, Rotate-and-Compare and bit-manipulation instructions provided for complex bit control.
- **Immediate Instruction Capability**
  May be used for storing constants in microcode or for configuring a second data port.
- **16-Bit Barrel Shifter**
- **32-Working Registers**

## GENERAL DESCRIPTION

The Am29C116 is a microprogrammable 16-bit CMOS microprocessor whose architecture and instruction set is optimized for high-performance peripheral controllers, like graphics controllers, disk controllers, communications controllers, front-end concentrators and modems. The device also performs well in microprogrammed processor applica-tions, especially when combined with the Am29C517A, 16 x 16 Multiplier. In addition to its complete arithmetic and logic instruction set, the Am29C116 instruction set contains functions particularly useful in controller applications; bit set, bit reset, bit test, rotate-and-merge, rotate-and-com-pare, and cyclic-redundancy-check (CRC) generation.

## SIMPLIFIED BLOCK DIAGRAM*
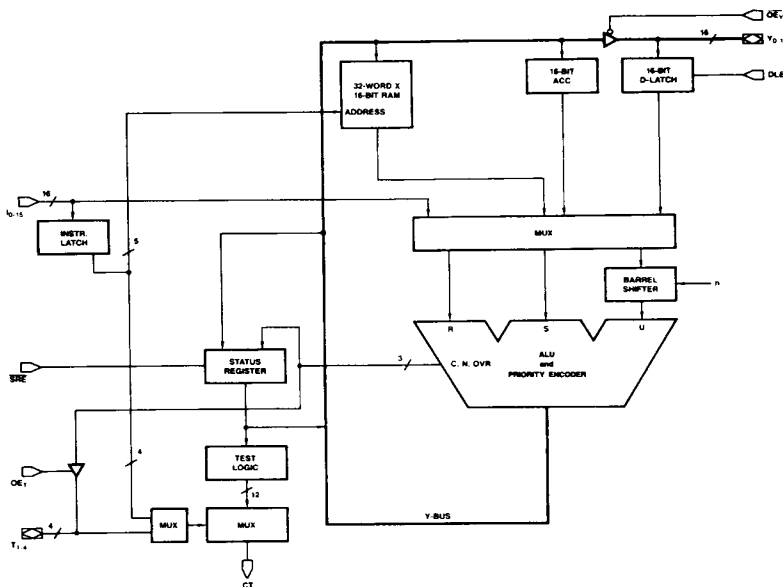


BD001963

*For a detailed block diagram, refer to Figure 2.

| Publication # | Rev. | Amendment |
|---|---|---|
| 07697 | C | /0 |
| Issue Date: **March 1988** | | |

## RELATED AMD PRODUCTS

| Part No. | Description |
|---|---|
| Am29C10A | CMOS 12-Bit Sequencer |
| Am29C111 | CMOS 16-Bit Microsequencer |
| Am29114 | 8-Level Real-Time Interrupt Controller |
| Am29117 | 2-Port 16-Bit Microprocessor |
| Am29C117 | CMOS Version of Am29117 |
| Am29118 | 8-Bit Am29C116 I/O Support |
| Am29130 | 16-Bit Barrel Shifter |
| Am29PL131 | 64 x 32 Field-Programmable Controller |
| Am29PL141 | 64 x 32 Field-Programmable Controller |
| Am29CPL141 | CMOS Version of Am29PL141 |
| Am29PL142 | 128 x 32 Field-Programmable Controller |
| Am29CPL144 | CMOS 512 x 32 Field-Programmable Controller |
| Am29C331 | CMOS 16-Bit Microsequencer |
| Am29C516A | CMOS 16 x 16 Multiplier |
| Am29C517A | CMOS 2-Port 16 x 16 Multiplier |

The following diagram (Figure 1) is a summary of devices within the Am29116 Family, showing performance versus power.
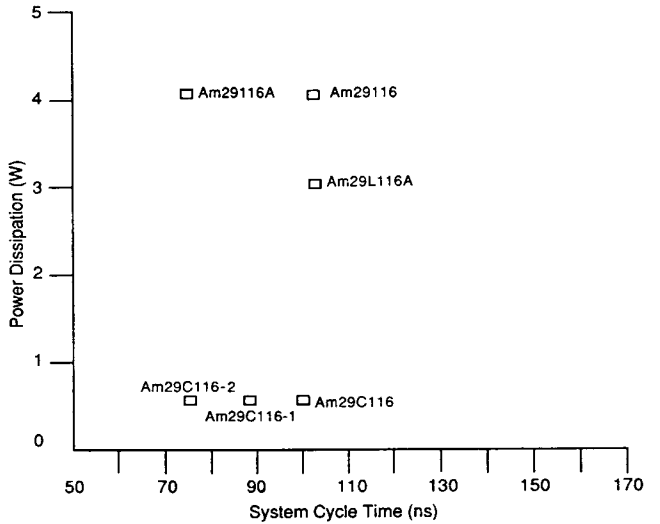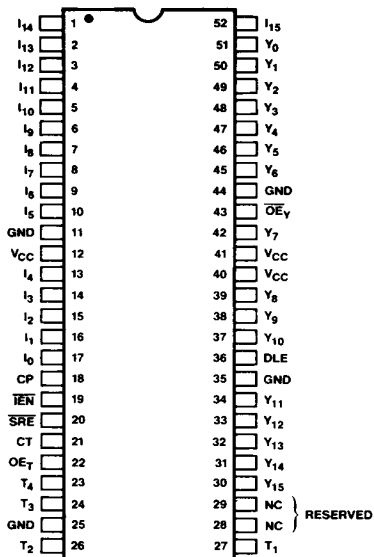


OP002292

**Figure 1. Am29116 Family**
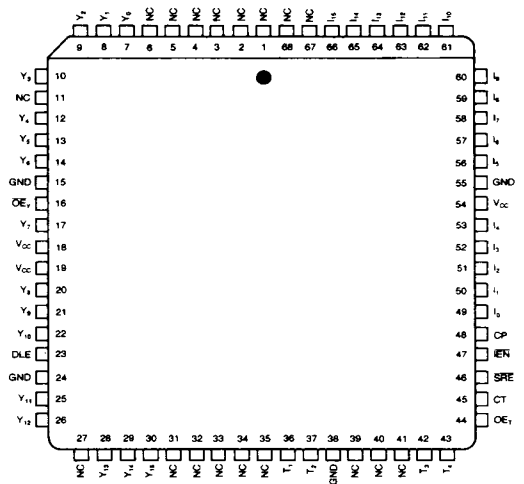**(Performance vs. Power)**

2

# CONNECTION DIAGRAMS
## Top View

### DIPs

| | | | | |
|---|---|---|---|---|
| $I_{14}$ | 1 | | 52 | $I_{15}$ |
| $I_{13}$ | 2 | | 51 | $Y_0$ |
| $I_{12}$ | 3 | | 50 | $Y_1$ |
| $I_{11}$ | 4 | | 49 | $Y_2$ |
| $I_{10}$ | 5 | | 48 | $Y_3$ |
| $I_9$ | 6 | | 47 | $Y_4$ |
| $I_8$ | 7 | | 46 | $Y_5$ |
| $I_7$ | 8 | | 45 | $Y_6$ |
| $I_6$ | 9 | | 44 | GND |
| $I_5$ | 10 | | 43 | $\overline{OE}_Y$ |
| GND | 11 | | 42 | $Y_7$ |
| $V_{CC}$ | 12 | | 41 | $V_{CC}$ |
| $I_4$ | 13 | | 40 | $V_{CC}$ |
| $I_3$ | 14 | | 39 | $Y_8$ |
| $I_2$ | 15 | | 38 | $Y_9$ |
| $I_1$ | 16 | | 37 | $Y_{10}$ |
| $I_0$ | 17 | | 36 | DLE |
| CP | 18 | | 35 | GND |
| $\overline{IEN}$ | 19 | | 34 | $Y_{11}$ |
| $\overline{SRE}$ | 20 | | 33 | $Y_{12}$ |
| CT | 21 | | 32 | $Y_{13}$ |
| $OE_T$ | 22 | | 31 | $Y_{14}$ |
| $T_4$ | 23 | | 30 | $Y_{15}$ |
| $T_3$ | 24 | | 29 | NC |
| GND | 25 | | 28 | NC |
| $T_2$ | 26 | | 27 | $T_1$ |

RESERVED

CD004190

### PLCC

CD009470

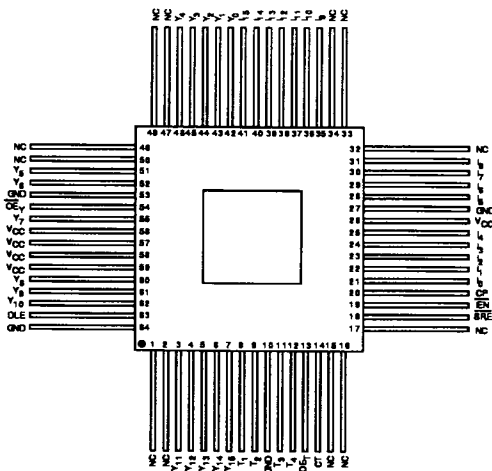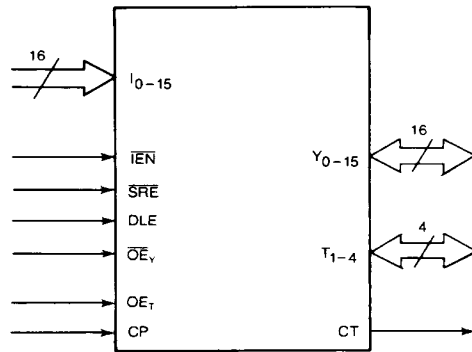### LCC
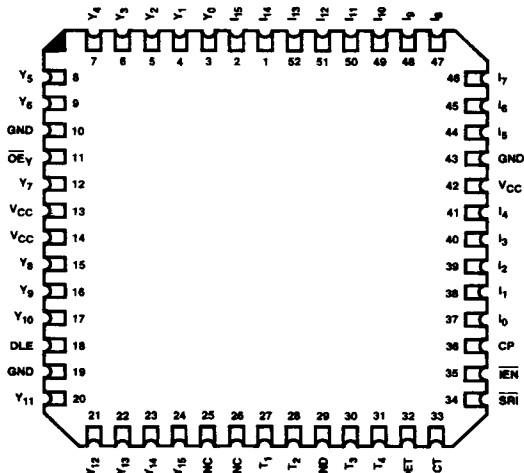
CD004180

### Flatpack

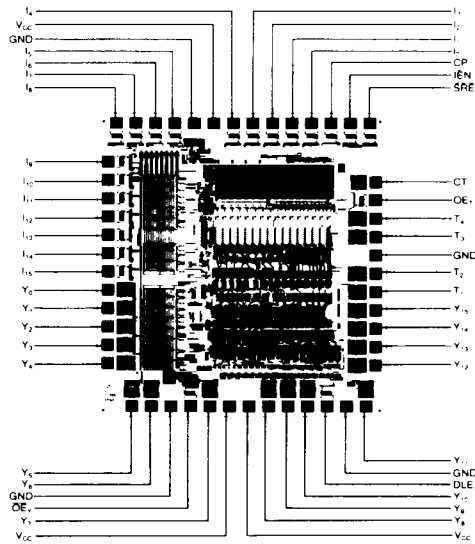Note: Pin 1 is marked for orientation.

3

# LOGIC SYMBOL


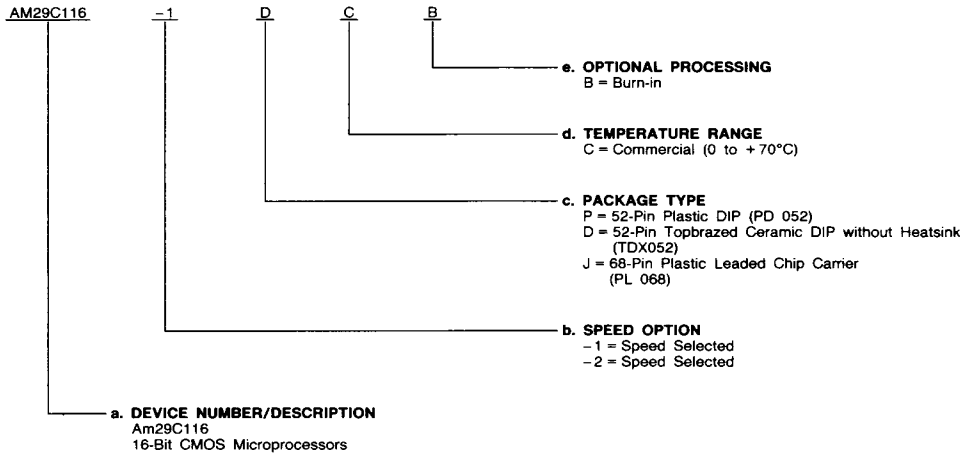
LS002254

# METALLIZATION AND PAD LAYOUT



Die Size: 0.169″ x 0.179″
Component Count: 14,000

4

# 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**

AM29C116     –1     D     C     B

**e. OPTIONAL PROCESSING**
B = Burn-in

**d. TEMPERATURE RANGE**
C = Commercial (0 to +70°C)

**c. PACKAGE TYPE**
P = 52-Pin Plastic DIP (PD 052)
D = 52-Pin Topbrazed Ceramic DIP without Heatsink (TDX052)
J = 68-Pin Plastic Leaded Chip Carrier (PL 068)

**b. SPEED OPTION**
–1 = Speed Selected
–2 = Speed Selected

**a. DEVICE NUMBER/DESCRIPTION**
Am29C116
16-Bit CMOS Microprocessors

### Valid Combinations

| Valid Combinations | |
|---|---|
| AM29C116 | |
| AM29C116-1 | PC, PCB, DC, DCB, JC |
| AM29C116-2 | |

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.
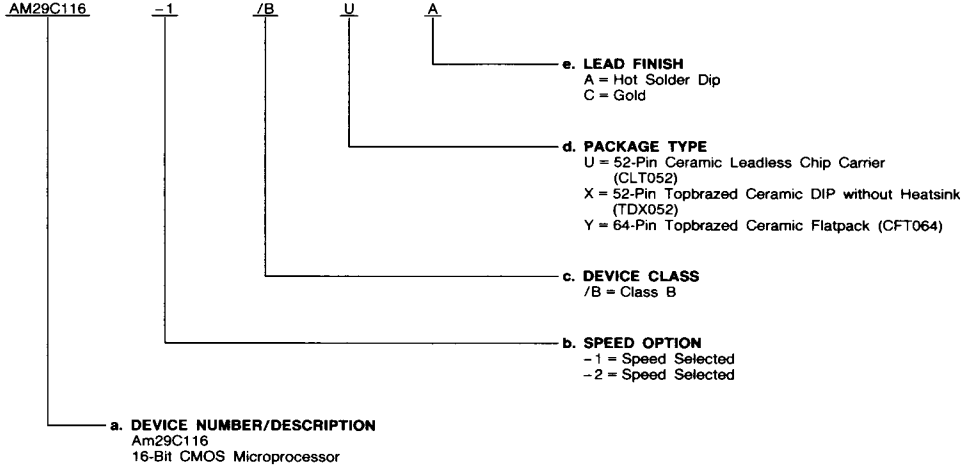
5

# 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. 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**

```
AM29C116        −1      /B      U       A
```

**e. LEAD FINISH**
A = Hot Solder Dip
C = Gold

**d. PACKAGE TYPE**
U = 52-Pin Ceramic Leadless Chip Carrier (CLT052)
X = 52-Pin Topbrazed Ceramic DIP without Heatsink (TDX052)
Y = 64-Pin Topbrazed Ceramic Flatpack (CFT064)

**c. DEVICE CLASS**
/B = Class B

**b. SPEED OPTION**
−1 = Speed Selected
−2 = Speed Selected

**a. DEVICE NUMBER/DESCRIPTION**
Am29C116
16-Bit CMOS Microprocessor

### 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.

| Valid Combinations | |
|---|---|
| AM29C116 | |
| AM29C116-1 | /BUA, /BXC, /BYC |
| AM29C116-2 | |

### Group A Tests

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

# PIN DESCRIPTION

**CP     Clock Pulse (Input)**

The clock input to the Am29C116. The RAM latch is transparent when the clock is HIGH. When the clock goes LOW, the RAM output is latched. Data is written into the RAM during the low period of the clock provided $\overline{IEN}$ is LOW and if the instruction being executed designates the RAM as the destination of operation. The Accumulator and Status Register will accept data on the LOW-HIGH transition of the clock if $\overline{IEN}$ is also LOW. The instruction latch becomes transparent when it exits an immediate instruction mode during a LOW-HIGH transition of the clock.

**CT     Conditional Test (Output)**

The condition code multiplexer selects one of the twelve condition code signals and places them on the CT output. A HIGH on the CT output indicates a passed condition and a LOW indicates a failed condition.

**DLE     Data Latch Enable (Input)**

When DLE is HIGH, the 16-bit data latch is transparent and is latched when DLE is LOW.

**$\overline{IEN}$     $\overline{\text{Instruction Enable}}$ (Input)**

With $\overline{IEN}$ LOW, data can be written into the RAM when the clock is LOW. The Accumulator can accept data during the LOW-HIGH transition of the clock. Having $\overline{IEN}$ LOW, the Status Register can be updated when $\overline{SRE}$ is LOW. With $\overline{IEN}$ HIGH, the conditional test output, CT, is disabled as a function of the instruction inputs. $\overline{IEN}$ should be LOW for the first half of the first cycle of an immediate instruction.

**$I_0 - I_{15}$     Instruction Inputs — 16 (Input)**

Used to select the operations to be performed in the Am29C116. Also used as data inputs while performing immediate instructions.

**$OE_T$     Output Enable (Input)**

When $OE_T$ is LOW, the 4-bit T outputs are disabled (high-impedance); when $OE_T$ is HIGH, the 4-bit T outputs are enabled (HIGH or LOW).

**$\overline{OE}_Y$     $\overline{\text{Output Enable}}$ (Input)**

When $\overline{OE}_Y$ is HIGH, the 16-bit Y outputs are disabled (high-impedance); when $\overline{OE}_Y$ is LOW, the 16-bit Y outputs are enabled (HIGH or LOW).

**$\overline{SRE}$     $\overline{\text{Status Register Enable}}$ (Input)**

When $\overline{SRE}$ and $\overline{IEN}$ are both LOW, the Status Register is updated at the end of all instructions with the exception of NO-OP, Save Status, and Test Status. Having either $\overline{SRE}$ or $\overline{IEN}$ HIGH will inhibit the Status Register from changing.

**$T_1 - T_4$     Input/Output Pins — 4 (Input/Output)**

Under the control of $OE_T$, the four lower status bits Z, C, N, OVR, become outputs on $T_1$-$T_4$, respectively, when $OE_T$ goes HIGH. When $OE_T$ is LOW, $T_1$-$T_4$ are used as inputs to generate the CT output.

**$Y_0 - Y_{15}$     Data I/O Lines — 16 (Input/Output)**

When $\overline{OE}_Y$ is HIGH, $Y_0$-$Y_{15}$ are used as external data inputs which allow data to be directly loaded into the 16-bit data latch. Having $\overline{OE}_Y$ LOW allows the ALU data to be output on $Y_0$-$Y_{15}$.

# FUNCTIONAL DESCRIPTION

The following diagram (Figure 2) shows a detailed block diagram of the Am29C116.
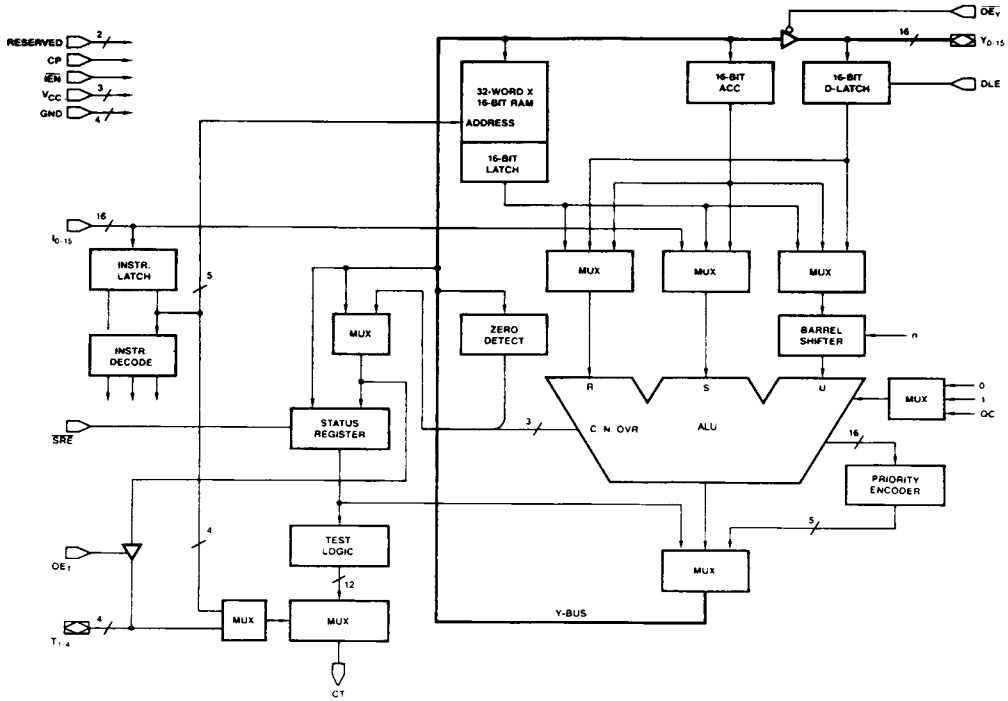


BD006540

**Figure 2. Detailed Am29C116 Block Diagram**

## Architecture of the Am29C116

The Am29C116 is a high-performance, microprogrammable 16-bit CMOS microprocessor.

As shown in the Block Diagrams, the device consists of the following elements interconnected with 16-bit data paths.

- 32-Word by 16-Bit RAM
- Accumulator
- Data Latch
- Barrel Shifter
- ALU
- Priority Encoder
- Status Register
- Condition-Code Generator/Multiplexer
- Three-State Output Buffers
- Instruction Latch and Decoder

### 32-Word by 16-Bit RAM

The 32-Word by 16-Bit RAM is a single-port RAM with a 16-bit latch at its output. The latches are transparent when the clock input (CP) is HIGH and latched when the clock input is LOW. Data is written into the RAM while the clock is LOW if the $\overline{IEN}$ input is also LOW and if the instruction being executed defines the RAM as the destination of the operation. For byte instructions, only the lower eight RAM bits are written into; for word instructions, all 16 bits are written into. With the use of an external multiplexer on five of the instruction inputs, it is possible to select separate read and write addresses for the same instruction. This two-address operation is not allowed for immediate instructions.

### Accumulator

The 16-bit Accumulator is an edge-triggered register. The Accumulator accepts data on the LOW-to-HIGH transition of the clock input if the $\overline{IEN}$ input is LOW and if the instruction being executed defines the Accumulator as the destination of the operation. For byte instructions, only the lower eight bits of the Accumulator are written into; for word instructions, all 16 bits are written into.

### Data Latch

The 16-bit Data Latch holds the data input to the Am29C116 on the bi-directional Y bus. The latch is transparent when the DLE input is HIGH and latched when the DLE input is LOW.

### Barrel Shifter

A 16-bit Barrel Shifter is used as one of the ALU inputs. This permits rotating data from either the RAM, the Accumulator or the Data Latch up to 15 positions. In the word mode, the Barrel Shifter rotates a 16-bit word; in the byte mode, it rotates only the lower eight bits.

### Arithmetic Logic Unit

The Am29C116 contains a 16-bit ALU with full carry lookahead across all 16 bits in the arithmetic mode. The ALU is capable of operating on either one, two or three operands, depending upon the instruction being executed. It has the ability to execute all conventional one and two operand operations, such as pass, complement, two's complement, add, subtract, AND, NAND, OR, NOR, EXOR, and EX-NOR. In addition, the ALU can also execute three-operand instructions such as rotate and merge and rotate and compare with mask. All ALU operations can be performed on either a word or byte basis, byte operations being performed on the lower eight bits only.

The ALU produces three status outputs, C (carry), N (negative) and OVR (overflow). The appropriate flags are generated at the byte or word level, depending upon whether the device is executing in the byte or word mode. The Z (zero) flag,

although not generated by the ALU, detects zero at both the byte and word level.

The carry input to the ALU is generated by the Carry Multiplexer which can select an input of zero, one, or the stored carry bit from the Status Register, QC. Using QC as the carry input allows execution of multiprecision addition and subtractions.

### Priority Encoder

The Priority Encoder produces a binary-weighted code to indicate the locations of the highest order ONE at its input. The input to the Priority Encoder is generated by the ALU which performs an AND operation on the operand to be prioritized and a mask. The mask determines which bit locations to eliminate from prioritization. In the word mode, if no bit is HIGH, the output is a binary zero. If bit 15 is HIGH, the output is a binary one. Bit 14 produces a binary two, etc. Finally, if only bit 0 is HIGH, a binary 16 is produced.

In the byte mode, bits 8 thru 15 do not participate. If none of bits 7 thru 0 are HIGH, the output is a binary zero. If bit 7 is HIGH a binary one is produced. Bit 6 produces a binary two, etc. Finally, if only bit 0 is HIGH, a binary 8 is produced.

### Status Register

The Status Register holds the 8-bit status word. With the Status-Register Enable, ($\overline{SRE}$) input LOW and the $\overline{IEN}$ input LOW, the Status Register is updated at the end of all instructions except NO-OP, Save-Status and Test-Status instructions. $\overline{SRE}$ going HIGH or $\overline{IEN}$ going HIGH inhibits the Status Register from changing.

The lower four bits of the Status Register contain the ALU status bits of Zero (Z), Carry (C), Negative (N), and Overflow (OVR). The upper four bits contain a Link bit and three user-definable status bits (Flag 1, Flag 2, Flag 3).

With $\overline{SRE}$ LOW and $\overline{IEN}$ LOW, the lower four status bits are updated after each instruction except those mentioned above, NO-OP, Save Status, Status Test and the Status Set/Reset instruction for the upper four bits. Under the same conditions, the upper four status bits are changed only during their respective Status Set/Reset instructions and during Status Load instructions in the word mode. The Link-Status bit is also updated after each shift instruction.

The Status Register can be loaded from the internal Y-bus, and can also be selected as a source for the internal Y-bus. When the Status Register is loaded in the word mode, all 8-bits are updated; in the byte mode, only the lower 4 bits (Z, C, N, OVR) are updated.

When the Status Register is selected as a source in the word mode, all eight bits are loaded into the lower byte of the destination; the upper byte of the destination is loaded with all zeros. In the byte mode, the Status Register again loads into the lower byte of the destination, but the upper byte remains unchanged. This Store and Load combination allows saving and restoring the Status Register for interrupt and subroutine processing. The four lower status bits (Z, C, N, OVR) can be read directly via the bidirectional T bus. These four bits are available as outputs on the $T_{1-4}$ outputs whenever $OE_T$ is HIGH.

### Condition-Code Generator/Multiplexer

The Condition-Code Generator/Multiplexer contains the logic necessary to develop the 12 condition-code test signals. The multiplexer portion can select one of these test signals and place it on the CT output for use by the microprogram sequence. The multiplexer may be addressed in two different ways. One way is through the Test Instruction. This instruction specifies the test condition to be placed in the CT output, but

does not allow an ALU operation at the same time. The second method uses the bidirectional T bus as an input. This requires extra bits in the microword, but provides the ability to simultaneously test and execute. The test instruction lines, $I_{0-4}$, have priority over $T_{1-4}$, for testing status.

### Three-State Output Buffers

There are two sets of Three-State Output Buffers in the Am29C116. One set controls the bidirectional, 16-bit Y bus. These outputs are enabled by placing a LOW on the $\overline{OE}$ input. A HIGH puts the Y outputs in the high-impedance state, allowing data to be input to the Data latch from an external source.

The second set of Three-State Output Buffers controls the bidirectional 4-bit T bus and is enabled by placing a HIGH on the $OE_T$ input. This allows storing the four internal ALU status bits (Z, C, N, OVR) externally. A LOW $OE_T$ input forces the T outputs into the high-impedance state. External devices can

then drive the T bus to select a test condition for the CT output.

### Instruction Latch and Decoder

The 16-bit Instruction Latch is normally transparent to allow decoding of the Instruction Inputs by the Instruction Decoder into the internal control signals for the Am29C116. All instructions except Immediate Instructions are executed in a single clock cycle.

Immediate instructions require two clock cycles for execution. During the first clock cycle, the Instruction Decoder recognizes that an Immediate Instruction is being specified and captures the data on the Instruction Inputs in the Instruction Latch. During the second clock cycle, the data on the Instruction Inputs is used as one of the operands for the function specified during the first clock cycle. At the end of the second clock cycle, the Instruction Latch is returned to its transparent state.

### Instruction Set

The instruction set of the Am29C116 is very powerful. In addition to the single and two operand logical and arithmetic instructions, the Am29C116 instruction set contains functions particularly useful in controller applications: bit set, bit reset, bit test, rotate and merge, rotate and compare, and cyclic-redundancy-check (CRC) generation. Complex instructions like rotate and merge, rotate and compare, and prioritize are executed in a single microcycle.

Three data types are supported by the Am29C116.

- Bit
- Byte
- Word (16-bit)

In the byte mode, data is written into the lower half of the word and the upper half is unchanged. The special case is when the status register is specified as the destination. In the byte mode, the LSH (OVR, N, C, Z) of the status register is updated and in the word mode all eight bits of the status register are updated. The status register does not change for save status and test status instructions. In the test status instructions, the CT output has the result and the Y-bus is undefined.

The Am29C116 Instruction Set can be divided into eleven types of instructions. These are:

- Single Operand
- Two Operand
- Single Bit Shift
- Rotate and Merge
- Bit Oriented
- Rotate by n Bits
- Rotate and Compare
- Prioritize
- Cyclic-Redundancy-Check
- Status
- No-Op

Each instruction type is arbitrarily divided into quadrants. Two of the sixteen instruction lines decode to four quadrants labelled from 0 to 3. The quadrants were defined mainly for convenience in classification of the instruction set and addressing modes and can be used together with the OP CODES to distinguish the instructions.

The following pages describe each of the instruction types in detail. Throughout the description $\overline{OE}_Y$ is assumed to be LOW allowing ALU outputs on the Y-bus.

Table 1 illustrates operand source-destination combinations for each instruction type.

# TABLE 1. OPERAND SOURCE DESTINATION COMBINATIONS

## Single Operand

| Instruction Type | Source (R/S) | Destination |
|---|---|---|
| Single Operand | RAM (Note 2) | RAM |
| | ACC | ACC |
| | D | Y Bus |
| | D(0E) | Status |
| | D(SE) | ACC and Status |
| | I | |
| | 0 | |

## Two Operand

| Instruction Type | Source (R) | Source (S) | Destination |
|---|---|---|---|
| Two Operand | RAM | ACC | RAM |
| | RAM | I | ACC |
| | D | RAM | Y Bus |
| | D | ACC | Status |
| | ACC | I | ACC and |
| | D | I | Status |

## Single Bit Shift

| Instruction Type | Source (U) | Destination |
|---|---|---|
| Single Bit Shift | RAM | RAM |
| | ACC | ACC |
| | ACC | Y Bus |
| | D | RAM |
| | D | ACC |
| | D | Y Bus |

## Rotate n Bits

| Instruction Type | Source (U) | Destination |
|---|---|---|
| Rotate n Bits | RAM | RAM |
| | ACC | ACC |
| | D | Y Bus |

## Bit Oriented

| Instruction Type | Source (R/S) | Destination |
|---|---|---|
| Bit Oriented | RAM | RAM |
| | ACC | ACC |
| | D | Y Bus |

## Rotate and Merge

| Instruction Type | Rotated Source (U) | Mask (S) | Non-Rotated Source/ Destination (R) |
|---|---|---|---|
| Rotate and Merge | D | I | ACC |
| | D | RAM | ACC |
| | D | I | RAM |
| | D | ACC | RAM |
| | ACC | I | RAM |
| | RAM | I | ACC |

## Rotate and Compare

| Instruction Type | Rotated Source (U) | Mask (S) | Non-Rotated Source/ Destination (R) |
|---|---|---|---|
| Rotate and Compare | D | I | |
| | D | I | RAM |
| | D | ACC | RAM |
| | RAM | I | ACC |

## Prioritize (Note 3)

| Instruction Type | Source (R) | Mask (S) | Destination |
|---|---|---|---|
| Prioritize (Note 3) | RAM | RAM | RAM |
| | ACC | ACC | ACC |
| | D | I | Y Bus |
| | | 0 | |

## Cyclic Redundancy Check

| Instruction Type | Data In | Destination | Polynomial |
|---|---|---|---|
| Cyclic Redundancy Check | QLINK | RAM | ACC |

## No Operation

| Instruction Type | |
|---|---|
| No Operation | – |

## Set Reset Status

| Instruction Type | Bits Affected |
|---|---|
| Set Reset Status | OVR, N, C, Z |
| | LINK |
| | Flag1 |
| | Flag2 |
| | Flag3 |

## Store Status

| Instruction Type | Source | Destination |
|---|---|---|
| Store Status | Status | RAM |
| | | ACC |
| | | Y Bus |

## Status Load

| Instruction Type | Source (R) | Source (S) | Destination |
|---|---|---|---|
| Status Load | D | ACC | Status |
| | ACC | I | Status and |
| | | | ACC |
| | D | I | |

## Test Status

| Instruction Type | Test Condition (CT) |
|---|---|
| Test Status | $(N \oplus OVR) + Z$ |
| | $N \oplus OVR$ |
| | Z |
| | OVR |
| | Low |
| | C |
| | $Z + \overline{C}$ |
| | N |
| | LINK |
| | Flag 1 |
| | Flag 2 |
| | Flag 3 |

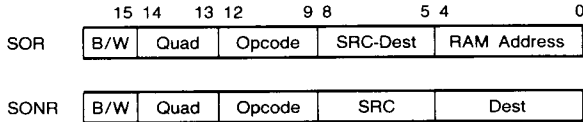Notes: 1. When there is no dividing line between the R&S OPERAND or SOURCE and DESTINATION, the two must be used as a given pair. But where there exists such a separation, any combination of them is possible.
2. In the SINGLE OPERAND INSTRUCTION, RAM cannot be used when both ACC and STATUS are designated as a DESTINATION.
3. In the PRIORITIZE INSTRUCTION, OPERAND and MASK must be different sources.

# SINGLE OPERAND INSTRUCTIONS

The Single Operand Instructions contain four indicators: byte or word mode, opcode, source and destination. They are further subdivided into two types. The first type uses RAM as a source or destination or both, and the second type does not use RAM as a source or destination. Both types have different instruction formats as shown below. Under the control of instruction inputs, the desired function is performed on the source and the result is either stored in the specified destination or placed on the Y-bus or both. For a special case where 8-bit to 16-bit conversion is needed, the Am29C116 is capable of extending sign bit (D(SE)) or binary zero (D(0E)) over 16-bits in the word mode. The least significant four bits of the Status Register (OVR, N, C, Z) are affected by the function performed in this category. The most significant bits of status register (FLAG1, FLAG2, FLAG3, LINK) are not affected. The only limitation in this type is that the RAM cannot be used as a source when both ACC and the Status Register are specified as a destination.

## SINGLE OPERAND FIELD DEFINITIONS

```
          15 14   13 12     9 8      5 4          0
SOR     | B/W | Quad | Opcode | SRC-Dest | RAM Address |

SONR    | B/W | Quad | Opcode |   SRC    |   Dest      |
```

## SINGLE OPERAND INSTRUCTION

| Instruction[1] | B/W[2] | Quad[3] | Opcode | | R/S[4] Dest[4] | | | RAM Address | |
|---|---|---|---|---|---|---|---|---|---|
| | | | I5 I4   I3 I2   9 | | 8      5 | | | 4          0 | |
| SOR | 0 = B<br>1 = W | 10 | 1100 MOVE SRC→Dest<br>1101 COMP S̄R̄C̄→Dest<br>1110 INC SRC + 1→Dest<br>1111 NEG S̄R̄C̄ + 1→Dest | | 0000 SORA RAM ACC<br>0010 SORY RAM Y Bus<br>0011 SORS RAM Status<br>0100 SOAR ACC RAM<br>0110 SODR D RAM<br>0111 SOIR I RAM<br>1000 SOZR 0 RAM<br>1001 SOZER D(0E) RAM<br>1010 SOSER D(SE) RAM<br>1011 SORR RAM RAM | | | 00000 R00 RAM Reg 00<br>. .    . .    . . . .<br>11111 R31 RAM Reg 31 | |
| **Instruction** | **B/W** | **Quad** | **Opcode** | | **R/S[4]** | | | **Destination** | |
| SONR | 0 = B<br>1 = W | 11 | 1100 MOVE SRC→Dest<br>1101 COMP S̄R̄C̄→Dest<br>1110 INC SRC + 1→Dest<br>1111 NEG S̄R̄C̄ + 1→Dest | | 0100 SOA ACC<br>0110 SOD D<br>0111 SOI I<br>1000 SOZ 0<br>1001 SOZE D(0E)<br>1010 SOSE D(SE) | | | 00000 NRY Y Bus<br>00001 NRA ACC<br>00100 NRS Status[5]<br>00101 NRAS ACC, Status[5] | |

Notes: 1. The instruction mnemonic designates different instruction formats used in the Am29C116. They are useful in microcode assembly.
2. B = Byte Mode, W = Word Mode.
3. See Instruction Set description.
4. R = Source; S = Source; Dest = Destination.
5. When status is destination,
Status i ← Yi i = 0 to 3 (Byte mode)
i = 0 to 7 (Word mode)

## Y BUS AND STATUS – SINGLE OPERAND INSTRUCTIONS

| Instruction | Opcode | Description | B/W | Y — Bus | Flag3 | Flag2 | Flag1 | LINK | OVR | N | C | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOR | MOVE | SRC→Dest | 0 = B | Y ← SRC | NC | NC | NC | NC | 0 | U | 0 | U |
| SONR | COMP | S̄R̄C̄→Dest | 1 = W | Y ← S̄R̄C̄ | NC | NC | NC | NC | 0 | U | 0 | U |
| | INC | SRC +1 → Dest | | Y ← SRC +1 | NC | NC | NC | NC | U | U | U | U |
| | NEG | S̄R̄C̄ +1→Dest | | Y ← S̄R̄C̄ +1 | NC | NC | NC | NC | U | U | U | U |

SRC = Source
U = Update
NC = No Change

0 = Reset
1 = Set
i = 0 to 15 when not specified

12

# TWO OPERAND INSTRUCTIONS

The Two Operand Instructions contain five indicators: byte or word mode, opcode, R source, S source, and destination. They are further subdivided into two types. The first type uses RAM as the source and/or destination and the second type does not use RAM as source or destination. The first type has two formats; the only difference is in the quadrant. Under the control of instruction inputs, the desired function is performed on the specified sources and the result is stored in the specified destination or placed on the Y-bus or both. The least significant four bits of the status register (OVR, N, C, Z) are affected by the arithmetic functions performed and only the N and Z bits are affected by the logical functions performed. The OVR and C bits of the status register are forced to ZERO for logical functions. Add with carry and Subtract with carry instructions are useful for Multiprecision Add or Subtract.

## TWO OPERAND FIELD DEFINITIONS

```
          15 14  13 12      9 8      5 4              0
TOR1    │ B/W │ Quad │ SRC-SRC │ Opcode │ RAM Address │
                      │  -Dest  │

TOR2    │ B/W │ Quad │ SRC-SRC │ Opcode │ RAM Address │
                      │  -Dest  │

TONR    │ B/W │ Quad │ SRC-SRC │ Opcode │    Dest     │
```

## TWO OPERAND INSTRUCTIONS

| Instruction | B/W | Quad | | | R[1] | S[1] | Dest[1] | | Opcode | | | RAM Address | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 = B | 00 | 0000 | TORAA | RAM | ACC | ACC | 0000 | SUBR | S minus R | 00000 | R00 | RAM Reg 00 |
| | 1 = W | | 0010 | TORIA | RAM | I | ACC | 0001 | SUBRC[2] | S minus R | . . | . . | . . . . |
| | | | 0011 | TODRA | D | RAM | ACC | | | with carry | 11111 | R31 | RAM Reg 31 |
| | | | 1000 | TORAY | RAM | ACC | Y Bus | 0010 | SUBS | R minus S | | | |
| | | | 1010 | TORIY | RAM | I | Y Bus | 0011 | SUBSC[2] | R minus S | | | |
| | | | 1011 | TODRY | D | RAM | Y Bus | | | with carry | | | |
| TOR1 | | | 1100 | TORAR | RAM | ACC | RAM | 0100 | ADD | R plus S | | | |
| | | | 1110 | TORIR | RAM | I | RAM | 0101 | ADDC | R plus S | | | |
| | | | 1111 | TODRR | D | RAM | RAM | | | with carry | | | |
| | | | | | | | | 0110 | AND | R • S | | | |
| | | | | | | | | 0111 | NAND | $\overline{R \cdot S}$ | | | |
| | | | | | | | | 1000 | EXOR | $R \oplus S$ | | | |
| | | | | | | | | 1001 | NOR | $\overline{R + S}$ | | | |
| | | | | | | | | 1010 | OR | R + S | | | |
| | | | | | | | | 1011 | EXNOR | $\overline{R \oplus S}$ | | | |

| Instruction | B/W | Quad | | | R[1] | S[1] | Dest[1] | | Opcode | | | RAM Address | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 = B | 10 | 0001 | TODAR | D | ACC | RAM | 0000 | SUBR | S minus R | 00000 | R00 | RAM Reg 00 |
| | 1 = W | | 0010 | TOAIR | ACC | I | RAM | 0001 | SUBRC[2] | S minus R | . . | . . | . . . . |
| | | | 0101 | TODIR | D | I | RAM | | | with carry | 11111 | R31 | RAM Reg 31 |
| | | | | | | | | 0010 | SUBS | R minus S | | | |
| | | | | | | | | 0011 | SUBSC[2] | R minus S | | | |
| | | | | | | | | | | with carry | | | |
| TOR2 | | | | | | | | 0100 | ADD | R plus S | | | |
| | | | | | | | | 0101 | ADDC | R plus S | | | |
| | | | | | | | | | | with carry | | | |
| | | | | | | | | 0110 | AND | R • S | | | |
| | | | | | | | | 0111 | NAND | $\overline{R \cdot S}$ | | | |
| | | | | | | | | 1000 | EXOR | $R \oplus S$ | | | |
| | | | | | | | | 1001 | NOR | $\overline{R + S}$ | | | |
| | | | | | | | | 1010 | OR | R + S | | | |
| | | | | | | | | 1011 | EXNOR | $\overline{R \oplus S}$ | | | |

Note 1: R = Source
S = Source
Dest = Destination
Note 2: During subtraction the carry is interpreted as $\overline{\text{borrow}}$.

## TWO OPERAND INSTRUCTIONS

| Instruction | B/W | Quad | R$^1$ | S$^1$ | Opcode | | Destination | |
|---|---|---|---|---|---|---|---|---|
| TONR | 0 = B<br>1 = W | 11 | 0001 TODA D<br>0010 TOAI ACC<br>0101 TODI D | ACC<br>I<br>I | 0000 SUBR<br>0001 SUBRC<br><br>0010 SUBS<br>0011 SUBSC<br><br>0100 ADD<br>0101 ADDC<br><br>0110 AND<br>0111 NAND<br>1000 EXOR<br>1001 NOR<br>1010 OR<br>1011 EXNOR | S minus R<br>S minus R with carry<br>R minus S<br>R minus S with carry<br>R plus S<br>R plus S with carry<br>R•S<br>$\overline{R•S}$<br>R⊕S<br>$\overline{R + S}$<br>R + S<br>$\overline{R⊕S}$ | 00000 NRY<br>00001 NRA<br>00100 NRS<br>00101 NRAS | Y Bus<br>ACC<br>Status$^2$<br>ACC, Status$^2$ |

Notes 1: R = Source
      S = Source
   2: When status is destination,
      Status $i \leftarrow Y_i$ i = 0 to 3 (Byte mode)
                  i = 0 to 7 (Word mode)
   3: During subtraction the carry is interpreted as $\overline{borrow}$.
   4: OVR = $C_8 \oplus C_7$ (Byte mode)
      OVR = $C_{16} \oplus C_{15}$ (Word mode)

## Y BUS AND STATUS CONTENTS – TWO OPERAND INSTRUCTIONS

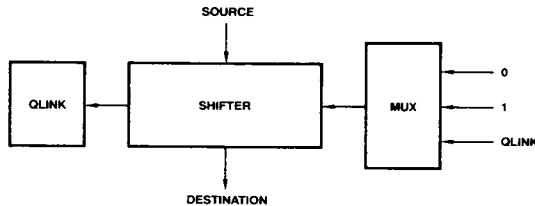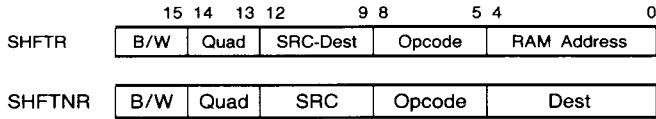| Instruction | Opcode | Description | B/W | Y – Bus | Flag3 | Flag2 | Flag1 | LINK | OVR | N | C | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TOR1<br>TOR2<br>TONR | SUBR | S minus R | 0 = B | $Y \leftarrow S + \overline{R} + 1$ | NC | NC | NC | NC | U | U | U | U |
| | SUBRC | S minus R with carry | 1 = w | $Y \leftarrow S + \overline{R} + QC$ | NC | NC | NC | NC | U | U | U | U |
| | SUBS | R minus S | | $Y \leftarrow R + \overline{S} + 1$ | NC | NC | NC | NC | U | U | U | U |
| | SUBSC | R minus S with carry | | $Y \leftarrow R + \overline{S} + QC$ | NC | NC | NC | NC | U | U | U | U |
| | ADD | R plus S | | $Y \leftarrow R + S$ | NC | NC | NC | NC | U | U | U | U |
| | ADDC | R plus S with carry | | $Y \leftarrow R + S + QC$ | NC | NC | NC | NC | U | U | U | U |
| | AND | R·S | | $Y \leftarrow R_i$ AND $S_i$ | NC | NC | NC | NC | 0 | U | 0 | U |
| | NAND | $\overline{R·S}$ | | $Y_i \leftarrow R_i$ NAND $S_i$ | NC | NC | NC | NC | 0 | U | 0 | U |
| | EXOR | R⊕S | | $Y_i \leftarrow R_i$ EXOR $S_i$ | NC | NC | NC | NC | 0 | U | 0 | U |
| | NOR | $\overline{R + S}$ | | $Y_i \leftarrow R_i$ NOR $S_i$ | NC | NC | NC | NC | 0 | U | 0 | U |
| | OR | R + S | | $Y_i \leftarrow R_i$ OR $S_i$ | NC | NC | NC | NC | 0 | U | 0 | U |
| | EXNOR | $\overline{R⊕S}$ | | $Y_i \leftarrow R_i$ EXNOR $S_i$ | NC | NC | NC | NC | 0 | 0 | 0 | U |

U = Update
NC = No Change
0 = Reset
1 = Set
i = 0 to 15 when not specified

# SINGLE BIT SHIFT INSTRUCTIONS

The Single Bit Shift Instructions contain four indicators: byte or word mode, direction and shift linkage, source and destination. They are further subdivided into two types. The first type uses RAM as the source and/or destination and the second type does not use RAM as source or destination. Under the control of the instruction inputs, the desired shift function is performed on the specified source and the result is stored in the specified destination or placed on the Y-bus or both. The direction and shift linkage indicator defines the direction of the shift (up or down) as well as what will be shifted into the vacant bit. On a shift-up instruction, the LSB may be loaded with ZERO, ONE, or the Link-Status bit (QLINK). The MSB is loaded into the Link-Status bit as shown in Figure 3. On a shift-down instruction, the MSB may be loaded with ZERO, ONE, the contents of the Status Carry flip-flop, (QC), the Exclusive-OR of the Negative-Status bit and the Overflow-Status bit (QN ⊕ QOVR) or the Link-Status bit. The LSB is loaded into the Link-Status bit as shown in Figure 4. The N and Z bits of the Status register are affected but the OVR and C bits are forced to ZERO. The Shift-Down with QN ⊕ QOVR is useful for Two's Complement multiplication.

## SINGLE BIT SHIFT FIELD DEFINITIONS:

|  | 15 14 | 13 12 | 9 8 | 5 4 | 0 |
|---|---|---|---|---|---|
| SHFTR | B/W | Quad | SRC-Dest | Opcode | RAM Address |

|  | 15 14 | 13 12 | 9 8 | 5 4 | 0 |
|---|---|---|---|---|---|
| SHFTNR | B/W | Quad | SRC | Opcode | Dest |



PF000360

**Figure 3. Shift Up Function**
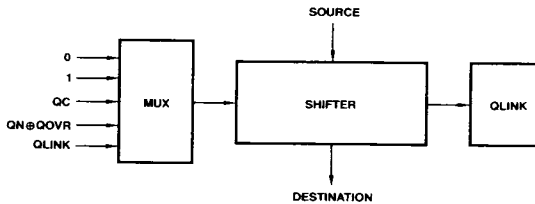


PF000350

**Figure 4. Shift Down Function**

15

# SINGLE BIT SHIFT INSTRUCTIONS

## SINGLE BIT SHIFT

| Instruction | B/W | Quad | U¹ | | | Dest¹ | Opcode | | | | RAM Address | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SHFTR | 0 = B<br>1 = W | 10 | 0110<br>0111 | SHRR<br>SHDR | RAM<br>D | RAM<br>RAM | 0000<br>0001<br>0010<br>0100<br>0101<br>0110<br>0111<br>1000 | SHUPZ<br>SHUP1<br>SHUPL<br>SHDNZ<br>SHDN1<br>SHDNL<br>SHDNC<br>SHDNOV | Up<br>Up<br>Up<br>Down<br>Down<br>Down<br>Down<br>Down | 0<br>1<br>QLINK<br>0<br>1<br>QLINK<br>QC<br>QN⊕QOVR | 00000<br>. .<br>11111 | R00<br>. .<br>R31 | RAM Reg 00<br>. . . .<br>RAM Reg 31 |

| Instruction | B/W | Quad | U¹ | | | Opcode | | | | Destination | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SHFTNR | 0 = B<br>1 = W | 11 | 0110<br>0111 | SHA<br>SHD | ACC<br>D | 0000<br>0001<br>0010<br>0100<br>0101<br>0110<br>0111<br>1000 | SHUPZ<br>SHUP1<br>SHUPL<br>SHDNZ<br>SHDN1<br>SHDNL<br>SHDNC<br>SHDNOV | Up<br>Up<br>Up<br>Down<br>Down<br>Down<br>Down<br>Down | 0<br>1<br>QLINK<br>0<br>1<br>QLINK<br>QC<br>QN⊕QOVR | 00000<br>00001 | NRY<br>NRA | Y Bus<br>ACC |

Note 1.    U = Source
           Dest = Destination

## Y BUS AND STATUS – SINGLE BIT SHIFT INSTRUCTIONS

| Instruction | Opcode | Description | | B/W | Y – Bus | Flag3 | Flag2 | Flag1 | LINK | OVR | N | C | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SHR<br>SHNR | SHUPZ<br>SHUP1<br>SHUPL | Up<br>Up<br>Up | 0<br>1<br>QLINK | 1 = W | $Y_i \leftarrow SRC_{i-1}$, i = 1 to 15;<br>$Y_0 \leftarrow$ Shift Input | NC | NC | NC | $SRC_{15}*$ | 0 | $SRC_{14}$ | 0 | U |
| | | | | 0 = B | $Y_i \leftarrow SRC_{i-1}$, i = 1 to 7;<br>$Y_0 \leftarrow$ Shift Input;<br>$Y_8 \leftarrow SRC_7$, $Y_i \leftarrow SRC_{i-9}$<br>for i = 9 to 15 | NC | NC | NC | $SRC_7*$ | 0 | $SRC_6$ | 0 | U |
| | SHDNZ<br>SHDN1<br>SHDNL<br>SHDNC<br>SHCNOV | Down<br>Down<br>Down<br>Down<br>Down | 0<br>1<br>QLINK<br>QC<br>QN⊕QOVR | 1 = W | $Y_i \leftarrow SRC_{i+1}$, i = 0 to 14;<br>$Y_{15} \leftarrow$ Shift Input | NC | NC | NC | $SRC_0*$ | 0 | Shift<br>Input | 0 | U |
| | | | | 0 = B | $Y_i \leftarrow SRC_{i+1}$, i = 0 to 6;<br>$Y_i \leftarrow SRC_{i-7}$, i = 8 to 14;<br>$Y_{7,15} \leftarrow$ Shift Input | NC | NC | NC | $SRC_0*$ | 0 | Shift<br>Input | 0 | U |

SRC = Source
U = Update
NC = No Change
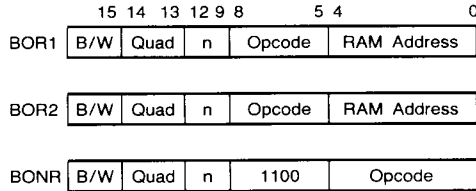0 = Reset
1 = Set
i = 0 to 15 when not specified

*Shifted Output is loaded into the QLINK.

# BIT ORIENTED INSTRUCTIONS

The Bit Oriented Instructions contain four indicators: byte or word mode, operation, source/destination, and the bit position of the bit to be operated on (Bit 0 is the least significant bit). They are further subdivided into two types. The first type uses the RAM as both source and destination and has two kinds of formats which differ only by quadrant. The second type does not use the RAM as a source or a destination. Under the control of the instruction inputs, the desired function is performed on the specified source and the result is stored in the specified destination or placed on the Y-bus or both. The operations which can be performed are: Set Bit n which forces the $n^{th}$ bit to a ONE leaving other bits unchanged; Reset Bit n which forces the $n^{th}$ bit to ZERO leaving the other bits unchanged; Test Bit n, which sets the ZERO Status Bit depending on the state of bit n leaving all the bits unchanged; Load $2^n$, which loads ONE in Bit position n and ZERO in all other bit positions; Load $\overline{2^n}$ which loads ZERO in bit position n and ONE in all other bit positions; increment by $2^n$, which adds $2^n$ to the operand; and decrement by $2^n$ which subtracts $2^n$ from the operand. For all the Load, Set, Reset and Test instructions, the N and Z bits are affected and OVR and C bit of the Status register are forced to ZERO. For all arithmetic instructions the LSH (OVR, C, N, Z bits) of the Status register is affected.

## BIT ORIENTED FIELD DEFINITIONS

|  | 15 14 | 13 12 9 | 8 | 5 4 | 0 |
|---|---|---|---|---|---|
| BOR1 | B/W | Quad | n | Opcode | RAM Address |

| | | | | | |
|---|---|---|---|---|---|
| BOR2 | B/W | Quad | n | Opcode | RAM Address |

| | | | | | |
|---|---|---|---|---|---|
| BONR | B/W | Quad | n | 1100 | Opcode |

## BIT ORIENTED INSTRUCTIONS

| Instruction | B/W | Quad | n | Opcode | | | RAM Address | | |
|---|---|---|---|---|---|---|---|---|---|
| BOR1 | 0 = B 1 = W | 11 | 0 to 15 | 1101 1110 1111 | SETNR RSTNR TSTNR | Set RAM, bit n Reset RAM, bit n Test RAM, bit n | 00000 .. 11111 | R00 .. R31 | RAM Reg 00 .... RAM Reg 31 |

| Instruction | B/W | Quad | n | Opcode | | | RAM Address | | |
|---|---|---|---|---|---|---|---|---|---|
| BOR2 | 0 = B 1 = W | 10 | 0 to 15 | 1100 1101 1110 1111 | LD2NR LDC2NR A2NR S2NR | $2^n \rightarrow$ RAM $\overline{2^n} \rightarrow$ RAM RAM plus $2^n \rightarrow$ RAM RAM minus $2^n \rightarrow$ RAM | 00000 .. 11111 | R00 .. R31 | RAM Reg 00 .... RAM Reg 31 |

| Instruction | B/W | Quad | n | | Opcode | | |
|---|---|---|---|---|---|---|---|
| BONR | 0 = B 1 = W | 11 | 0 to 15 | 1100 | 00000 00001 00010 00100 00101 00110 00111 10000 10001 10010 10100 10101 10110 10111 | TSTNA RSTNA SETNA A2NA S2NA LD2NA LDC2NA TSTND RSTND SETND A2NDY S2NDY LS2NY LDC2NY | Test ACC, bit n Reset ACC, bit n Set ACC, bit n ACC plus $2^n \rightarrow$ ACC ACC minus $2^n \rightarrow$ ACC $2^n \rightarrow$ ACC $\overline{2^n} \rightarrow$ ACC Test D, bit n Reset D, bit n Set D, bit n D plus $2^n \rightarrow$ Y BUS D minus $2^n \rightarrow$ Y Bus $2^n \rightarrow$ Y Bus $\overline{2^n} \rightarrow$ Y Bus |

# BIT ORIENTED INSTRUCTIONS

## Y BUS AND STATUS – BIT ORIENTED INSTRUCTIONS

| Instruction | Opcode | Description | B/W | Y – Bus | Flag3 | Flag2 | Flag1 | LINK | OVR | N | C | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BOR1 | SETNR | Set RAM Bit n | 0 = B | $Y_i \leftarrow RAM_i$ for $i \neq n$; $Y_n \leftarrow 1$ | NC | NC | NC | NC | 0 | U | 0 | 0 |
| | RSTNR | Reset RAM, Bit n | 1 = W | $Y_i \leftarrow RAM_i$ for $i \neq n$; $Y_n \leftarrow 0$ | NC | NC | NC | NC | 0 | U | 0 | U |
| | TSTNR | Test Ram, Bit n | | $Y_i \leftarrow 0$ for $i \neq n$; $Y_n \leftarrow SRC_n$ | NC | NC | NC | NC | 0 | U | 0 | U |
| BOR2 | LD2NR | $2^n \rightarrow RAM$ | | $Y_i \leftarrow 0$ for $i \neq n$; $Y_n \leftarrow 1$ | NC | NC | NC | NC | 0 | U | 0 | 0 |
| | LDC2NR | $\overline{2^n} \rightarrow RAM$ | | $Y_i \leftarrow 1$ for $i \neq n$; $Y_n \leftarrow 0$ | NC | NC | NC | NC | 0 | U | 0 | 0 |
| | A2NR | $RAM + 2^n \rightarrow RAM$ | | $Y \leftarrow RAM + 2^n$ | NC | NC | NC | NC | U | U | U | U |
| | S2NR | $RAM - 2^n \rightarrow RAM$ | | $Y \leftarrow RAM - 2^n$ | NC | NC | NC | NC | U | U | U | U |
| BONR | TSTNA | Test ACC, Bit n | | $Y_i \leftarrow 0$ for $i \neq n$; $Y_n \leftarrow ACC_n$ | NC | NC | NC | NC | 0 | U | 0 | U |
| | RSTNA | Reset ACC, Bit n | | $Y_i \leftarrow ACC_i$ for $i \neq n$; $Y_n \leftarrow 0$ | NC | NC | NC | NC | 0 | U | 0 | U |
| | SETNA | Set ACC, Bit n | | $Y_i \leftarrow ACC_i$ for $i \neq n$; $Y_n \leftarrow 1$ | NC | NC | NC | NC | 0 | U | 0 | 0 |
| | A2NA | $ACC + 2^n \rightarrow ACC$ | | $Y \leftarrow ACC + 2^n$ | NC | NC | NC | NC | U | U | U | U |
| | S2NA | $ACC - 2^n \rightarrow ACC$ | | $Y \leftarrow ACC - 2^n$ | NC | NC | NC | NC | U | U | U | U |
| | LD2NA | $2^n \rightarrow ACC$ | | $Y_i \leftarrow 0$ for $i \neq n$; $Y_n \leftarrow 1$ | NC | NC | NC | NC | 0 | U | 0 | 0 |
| | LDC2NA | $\overline{2^n} \rightarrow ACC$ | | $Y_i \leftarrow 1$ for $i \neq n$; $Y_n \leftarrow 0$ | NC | NC | NC | NC | 0 | U | 0 | 0 |
| | TSTND | Test D, Bit n | | $Y_i \leftarrow 0$ for $i \neq n$; $Y_n \leftarrow D_n$ | NC | NC | NC | NC | 0 | U | 0 | U |
| | RSTND | Reset D, Bit n* | | $Y_i \leftarrow D_i$ for $i \neq n$; $Y_n \leftarrow 0$ | NC | NC | NC | NC | 0 | U | 0 | U |
| | SETND | Set D, Bit n* | | $Y_i \leftarrow D_i$ for $i \neq n$; $Y_n \leftarrow 1$ | NC | NC | NC | NC | 0 | U | 0 | 0 |
| | A2NDY | $D + 2^n \rightarrow Y$ Bus | | $Y \leftarrow D + 2^n$ | NC | NC | NC | NC | U | U | U | U |
| | S2NDY | $D - 2^n \rightarrow Y$ Bus | | $Y \leftarrow D - 2^n$ | NC | NC | NC | NC | U | U | U | U |
| | LD2NY | $2^n \rightarrow Y$ Bus | | $Y_i \leftarrow 0$ for $i \neq n$; $Y_n \leftarrow 1$ | NC | NC | NC | NC | 0 | U | 0 | 0 |
| | LDC2NY | $\overline{2^n} \rightarrow Y$ Bus | | $Y_i \leftarrow 1$ for $i \neq n$; $Y_n \leftarrow 0$ | NC | NC | NC | NC | 0 | U | 0 | 0 |

SRC = Source
U = Update
NC = No Change
0 = Reset
1 = Set
i = 0 to 15 when not specified

*Destination is not D Latch but Y Bus.

18

# ROTATE BY n BITS INSTRUCTIONS

The Rotate by n Bits Instructions contain four indicators: byte or word mode, source, destination and the number of places the source is to be rotated. They are further subdivided into two types. The first type uses RAM as a source and/or a destination and the second type does not use RAM as a source or destination. The first type has two different formats and the only difference is in the quadrant. The second type has only one format as shown in the table. Under the control of instruction inputs, the n indicator specifies the number of bit positions the source is to be rotated up (0 to 15), and the result is either stored in the specified destination or placed on the Y bus or both. An example of this instruction is given in Figure 5. In the Word mode, all 16-bits are rotated up; while in the Byte mode, only the lower 8-bits (0-7) are rotated up. In the Word mode, a rotate up by n bits is equivalent to a rotate down by (16-n) bits. Similarly, in the Byte mode a rotate up by n bits is equivalent to a rotate down by (8-n) bits. The N and Z bits of the Status Register are affected and OVR and C bits are forced to ZERO.

### EXAMPLE: n = 4, Word Mode

| | | | | |
|---|---|---|---|---|
| Source | 0001 | 0011 | 0111 | 1111 |
| Destination | 0011 | 0111 | 1111 | 0001 |

### EXAMPLE: n = 4, Byte Mode

| | | | | |
|---|---|---|---|---|
| Source | 0001 | 0011 | 0111 | 1111 |
| Destination | 0001 | 0011 | 1111 | 0111 |

**Figure 5. Rotate by n Example**

### ROTATE BY n BITS FIELD DEFINITIONS

|   | 15 | 14 13 | 12 9 | 8 | 5 4 | 0 |
|---|---|---|---|---|---|---|
| ROTR1 | B/W | Quad | n | SRC-Dest | RAM Address | |
| ROTR2 | B/W | Quad | n | SRC-Dest | RAM Address | |
| ROTNR | B/W | Quad | n | 1100 | SRC-Dest | |

## ROTATE BY n BITS INSTRUCTIONS

| Instruction | B/W | Quad | n | | U[1] | Dest[1] | | RAM Address | |
|---|---|---|---|---|---|---|---|---|---|
| ROTR1 | 0 = B<br>1 = W | 00 | 0 to 15 | 1100<br>1110<br>1111 | RTRA<br>RTRY<br>RTRR | RAM<br>RAM<br>RAM | ACC<br>Y Bus<br>RAM | 00000<br>. .<br>11111 | R00<br>. .<br>R31 | RAM Reg 00<br>. . . .<br>RAM Reg 31 |

| Instruction | B/W | Quad | n | | U[1] | Dest[1] | | RAM Address | |
|---|---|---|---|---|---|---|---|---|---|
| ROTR2 | 0 = B<br>1 = W | 01 | 0 to 15 | 0000<br>0001 | RTAR<br>RTDR | ACC<br>D | RAM<br>RAM | 00000<br>. . . .<br>11111 | R00<br>. .<br>R31 | RAM Reg 00<br>. . .<br>RAM Reg 31 |

| Instruction | B/W | Quad | n | | | | U[1] | Dest[1] | |
|---|---|---|---|---|---|---|---|---|---|
| ROTNR | 0 = B<br>1 = W | 11 | 0 to 15 | 1100 | | | 11000<br>11001<br>11100<br>11101 | RTDY<br>RTDA<br>RTAY<br>RTAA | D<br>D<br>ACC<br>ACC | Y Bus<br>ACC<br>Y Bus<br>ACC |

Note 1: U = Source
Dest = Destination

## Y BUS AND STATUS – ROTATE BY n BITS INSTRUCTIONS

| Instruction | Op-code | B/W | Y – Bus | Flag3 | Flag2 | Flag1 | LINK | OVR | N | C | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ROTR1<br>ROTR2<br>ROTNR | | 1 = W | $Y_i \leftarrow SRC_{(i-n)mod16}$ | NC | NC | NC | NC | 0 | $SRC_{15-n}$ | 0 | U |
| | | 0 = B | $Y_i \leftarrow SRC_{i+8} = SRC_{(i-n)mod8}$<br>for $i = 0$ to 7 | NC | NC | NC | NC | 0 | $SRC_{8-n}$ | 0 | U |

SRC = Source
U = No Change
0 = Reset
1 = Set
i = 0 to 15 when not specified

19

# ROTATE AND MERGE INSTRUCTION

The Rotate and Merge Instructions contain five indicators: byte or word mode, rotated source, non-rotated source/destination, mask and the number of bit positions a source is to be rotated. The function performed by the Rotate and Merge instruction is illustrated in Figure 6. The rotated source, U, is rotated up by the Barrel Shifter n places. The mask input then selects, on a bit by bit basis, the rotated U input or R input. A ZERO in bit i of the mask will select the $i^{th}$ bit of the R input as the $i^{th}$ output bit, while ONE in bit i will select the $i^{th}$ rotated U input as the output bit. The output word is stored in the non-rotated operand location. The N and Z bits are affected. The OVR and C bits of the Status register are forced to ZERO. An example of this instruction is given in Figure 7.



PF000630

**Figure 6. Rotate and Merge Function**

## ROTATE AND MERGE FIELD DEFINITIONS:

| | 15 14 13 | 12 9 | 8 | | 5 4 | 0 |
|---|---|---|---|---|---|---|
| ROTM | B/W | Quad | n | ROT SRC-Non ROT SRC-Mask | RAM Address | |

### EXAMPLE: n = 4, Word Mode

| | | | | |
|---|---|---|---|---|
| U | 0011 | 0001 | 0101 | 0110 |
| Rotated U | 0001 | 0101 | 0110 | 0011 |
| R | 1010 | 1010 | 1010 | 1010 |
| Mask (S) | 0000 | 1111 | 0000 | 1111 |
| Destination | 1010 | 0101 | 1010 | 0011 |

**Figure 7. Rotate and Merge Example**

## ROTATE AND MERGE INSTRUCTION

| Instruction | B/W | Quad | n | | | U[1] | R/Dest[1] | S[1] | RAM Address | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ROTM | 0 = B<br>1 = W | 01 | 0 to 15 | 0111<br>1000<br>1001<br>1010<br>1100<br>1110 | MDAI<br>MDAR<br>MDRI<br>MDRA<br>MARI<br>MRAI | D<br>D<br>D<br>D<br>ACC<br>RAM | ACC<br>ACC<br>RAM<br>RAM<br>RAM<br>ACC | I<br>RAM<br>I<br>ACC<br>RAM<br>I | 00000<br>. .<br>11111 | R00<br>. .<br>R31 | RAM Reg 00<br>. . . .<br>RAM Reg 31 |

Note 1.  U = Rotated Source
R/Dest = Non-Rotated Source and Destination
S = Mask

## Y BUS AND STATUS – ROTATED MERGE

| Instruction | Opcode | B/W | Y – Bus | Flag3 | Flag2 | Flag1 | LINK | OVR | N | C | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ROTM | | 1=W | $Y_i \leftarrow$ (Non Rot Op)$_i \cdot \overline{(mask)}_i$ + (Rot Op)$_{(i-n) mod\ 16} \cdot (mask)_i$ | NC | NC | NC | NC | 0 | U | 0 | U |
| | | 0 = B | $Y_i \leftarrow$ (Non Rot Op)$_i \cdot \overline{(mask)}_i$ + (Rot Op)$_{(i-n) mod\ 8} \cdot (mask)_i$ | NC | NC | NC | NC | 0 | U | 0 | U |

U = Update
NC = No Change
0 = Reset
1 = Set
i = 0 to 15 when not specified

# ROTATE AND COMPARE INSTRUCTIONS

The Rotate and Compare Instructions contain five indicators: byte or word mode, rotated source, non-rotated source, mask, and the number of bit positions the rotated source is to be rotated up. Under the control of instruction inputs, the function performed by the Rotate and Compare instruction is illustrated in Figure 8. The rotated operand is rotated by the Barrel Shifter n places. The mask is invert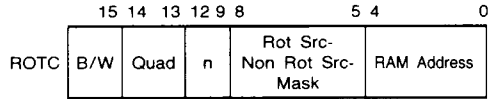ed and ANDed on a bit-by-bit basis with the output of the Barrel Shifter and R input. Thus, a ONE in the mask input eliminates that bit from the comparison. A ZERO allows the comparison. If the comparison passes, the Zero flag is set. If it fails, the Zero flag is reset. The N and Z bit are affected. The OVR and C bits of the Status register are forced to ZERO. An example of this instruction is given in Figure 9.
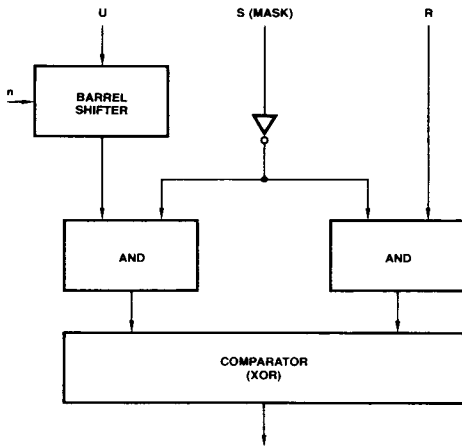


PF000650

**Figure 8. Rotate and Compare Function**

## ROTATE AND COMPARE FIELD DEFINITIONS

| | 15 14 | 13 12 9 | 8 | 5 4 | 0 |
|---|---|---|---|---|---|
| ROTC | B/W | Quad | n | Rot Src-<br>Non Rot Src-<br>Mask | RAM Address |

**EXAMPLE: n = 4, Word Mode**

| | | | | |
|---|---|---|---|---|
| U | 0011 | 0001 | 0101 | 0110 |
| U Rotated | 0001 | 0101 | 0110 | 0011 |
| R | 0001 | 0101 | 1111 | 0000 |
| Mask (S) | 0000 | 0000 | 1111 | 1111 |
| Z (status) = 1 | | | | |

**Figure 9. Rotate and Compare Examples**

## ROTATE AND COMPARE INSTRUCTIONS

| Instruction | B/W | Quad | n | | U[1] | R[1] | S[1] | RAM Address | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ROTC | 0=B<br>1=W | 01 | 0 to 15 | 0010 CDAI<br>0011 CDRI<br>0100 CDRA<br>0101 CRAI | D<br>D<br>D<br>RAM | ACC<br>RAM<br>RAM<br>ACC | I<br>I<br>ACC<br>I | 00000<br>. .<br>11111 | R00<br>. .<br>R31 | RAM Reg 00<br>. . .<br>RAM Reg 31 |

Note 1.　U = Rotated Source
　　　　　R = Non-Rotated Source
　　　　　S = Mask

## Y BUS AND STATUS – ROTATE AND COMPARE

| Instruction | Opcode | B/W | Y – Bus | Flag3 | Flag2 | Flag1 | LINK | OVR | N | C | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ROTC | | 1 = W | $Y_i \leftarrow (\text{Non Rot Op})_i \cdot \overline{(\text{mask})}_i \oplus$ <br> $(\text{Rot Op})_{(i-n)\bmod 16} \cdot \overline{(\text{mask})}_i$ | NC | NC | NC | NC | 0 | U | 0 | U |
| | | 0 = B | $Y_i \leftarrow (\text{Non Rot Op})_i \cdot \overline{(\text{mask})}_i \oplus$ <br> $(\text{Rot Op})_{(i-n)\bmod 8} \cdot \overline{(\text{mask})}_i$ | NC | NC | NC | NC | 0 | U | 0 | U |

U = Update
NC = No Change
0 = Reset
1 = Set
i = 0 to 15 when not specified

# PRIORITIZE INSTRUCTION

The Prioritize Instructions contain four indicators: byte or word mode, operand source (R), mask source (S) and destination. They are further subdivided into two types. The function performed by the Prioritize instruction is shown in Figure 10. The R operand is ANDed with the complement of the Mask operand. A ZERO in the Mask operand allows the corresponding bit in the R operand to participate in the priority encoding function. A ONE in the Mask operand forces the corresponding bit in the R operand to a ZERO, eliminating it from participation in the priority encoding function.

The priority encoder accepts a 16-bit input and produces a 5-bit binary-weighted code indicating the bit position of the highest priority active bit. If none of the inputs are active, the output is ZERO. In the Word mode, if input bit 15 is active, the output is 1, etc. Figure 11 lists the output as a function of the highest-priority active-bit position in both the Word and Byte mode. The N and Z bits are affected and the OVR and C bits of the status register are forced to ZERO. The only limitation in this instruction is that the operand and the mask must be different sources.
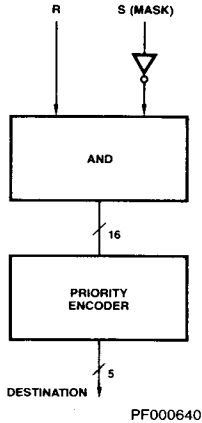


PF000640

**Figure 10. Prioritize Function**

## PRIORITIZE INSTRUCTION FIELD DEFINITIONS

| 15 | 14 13 | 12        9 | 8      5 | 4        0 |
|----|-------|-------------|----------|------------|
| B/W | Quad | Destination | Source (R) | RAM Address/Mask (S) |

| B/W | Quad | Mask (S) | Destination | RAM Address/Source (R) |
|-----|------|----------|-------------|------------------------|

| B/W | Quad | Mask (S) | Source (R) | RAM Address/Destination |
|-----|------|----------|------------|-------------------------|

| B/W | Quad | Mask (S) | Source (R) | Destination |
|-----|------|----------|------------|-------------|

|  | WORD MODE | | BYTE MODE* | |
|---|---|---|---|---|
| **Highest Priority Active Bit** | **Encoder Output** | **Highest Priority Active Bit** | **Encoder Output** |
| None | 0 | None | 0 |
| 15 | 1 | 7 | 1 |
| 14 | 2 | 6 | 2 |
| . | . | . | . |
| . | . | . | . |
| . | . | . | . |
| 1 | 15 | 1 | 7 |
| 0 | 16 | 0 | 8 |

*Bits 8 through 15 do not participate.

**Figure 11.**

## PRIORITIZE INSTRUCTION

| Instruction | B/W | Quad | Destination | | | Source (R) | | | RAM Address/Mask (S) | | |
|-------------|-----|------|------|------|------|------|------|------|------|------|------|
| PRT1 | 0 = B<br>1 = W | 10 | 1000<br>1010<br>1011 | PRIA<br>PR1Y<br>PR1R | ACC<br>Y Bus<br>RAM | 0111<br>1001 | RPT1A<br>PR1D | ACC<br>D | 00000<br>. .<br>11111 | R00<br>. .<br>R31 | RAM Reg 00<br>. . . .<br>RAM Reg 31 |

| Instruction | B/W | Quad | Mask (S) | | | Destination | | | RAM Address/Source (R) | | |
|-------------|-----|------|------|------|------|------|------|------|------|------|------|
| PRT2 | 0 = B<br>1 = W | 10 | 1000<br>1010<br>1011 | PRA<br>PRZ<br>PRI | Acc<br>0<br>I | 0000<br>0010 | PR2A<br>PR2Y | ACC<br>Y Bus | 00000<br>. .<br>11111 | R00<br>. .<br>R31 | RAM Reg 00<br>. . . .<br>RAM Reg 31 |

| Instruction | B/W | Quad | Mask (S) | | | Source (R) | | | RAM Address/Dest | | |
|-------------|-----|------|------|------|------|------|------|------|------|------|------|
| PRT3 | 0 = B<br>1 = W | 10 | 1000<br>1010<br>1011 | PRA<br>PRZ<br>PRI | ACC<br>0<br>I | 0011<br>0100<br>0110 | PR3R<br>PR3A<br>PR3D | RAM<br>ACC<br>D | 00000<br>. .<br>11111 | R00<br>. .<br>R31 | RAM Reg 00<br>. . . .<br>RAM Reg 31 |

| Instruction | B/W | Quad | Mask (S) | | | Source (R) | | | Destination | | |
|-------------|-----|------|------|------|------|------|------|------|------|------|------|
| PRTNR | 0 = B<br>1 = W | 11 | 1000<br>1010<br>1011 | PRA<br>PRZ<br>PRI | ACC<br>0<br>I | 0100<br>0110 | PRTA<br>PRTD | ACC<br>D | 00000<br>00001 | NRY<br>NRA | Y Bus<br>ACC |

## Y BUS AND STATUS - PRIORITIZE INSTRUCTION

| Instruction | Opcode | B/W | Y – Bus | Flag3 | Flag2 | Flag1 | LINK | OVR | N | C | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PRT1<br>PRT2 | | 1 = W | $Y_i \leftarrow$ CODE $(SCR_n \cdot \overline{mask_n})$;<br>$Y_m \leftarrow 0$; i = 0 to 4 and n = 0 to 15<br>m = 5 to 15 | NC | NC | NC | NC | 0 | U | 0 | U |
| PRT3<br>PRTNR | | 0 = B | $Y_i \leftarrow$ CODE $(SCR_n \cdot \overline{mask_n})$;<br>$Y_m \leftarrow 0$; i = 0 to 3 and n = 0 to 7<br>m = 4 to 15 | NC | NC | NC | NC | 0 | U | 0 | U |

| | | |
|---|---|---|
| SRC = Source | NC = No Change | 1 = Set |
| U = Update | 0 = Reset | i = 0 to 15 when not specified |

## CRC INSTRUCTION

The CRC (Cyclic-Redundancy-Check) Instructions contain one indicator: address of a RAM register to use as the check sum register. The CRC instruction provides a method for generation of the check bits in a CRC calculation. Two CRC instructions are provided – CRC Forward and CRC Reverse. The reason for providing two instructions is that CRC standards do not specify which data bit is to be transmitted first, the LSB or the MSB, but they do specify which check bit must be transmitted first. Figure 12 illustrates the method used to generate these check bits for the CRC Forward function and

Figure 13 illustrates method used for the 2CRC Reverse function. The ACC serves as a polynomial mask to define the generating polynomial while the RAM register holds the partial result and eventually the calculated check sum. The LINK-bit is used as the serial input. The serial input combines with the MSB of the check-sum register, according to the polynomial defined by the polynomial mask register. When the last input bit has been processed, the check-sum register contains the CRC check bits. The LINK, N and Z bits are affected and the OVR and C bits of the Status register are forced to ZERO.

### CYCLIC-REDUNDANCY-CHECK DEFINITIONS:

```
      15 14 13 12    9 8    5 4            0
CRCF  [ 1 | Quad | 0110 | 0011 | RAM Address ]

CRCR  [ 1 | Quad | 0110 | 1001 | RAM Address ]
```



*This bit must be transmitted first.
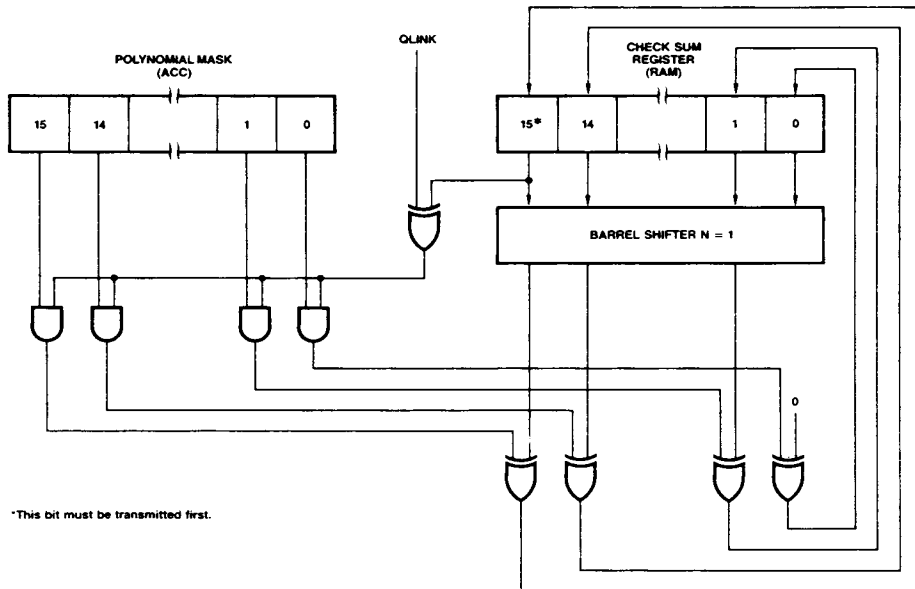
PF000330

**Figure 12. CRC Forward Function**

23

# CRC INSTRUCTION



*This bit must be transmitted first.
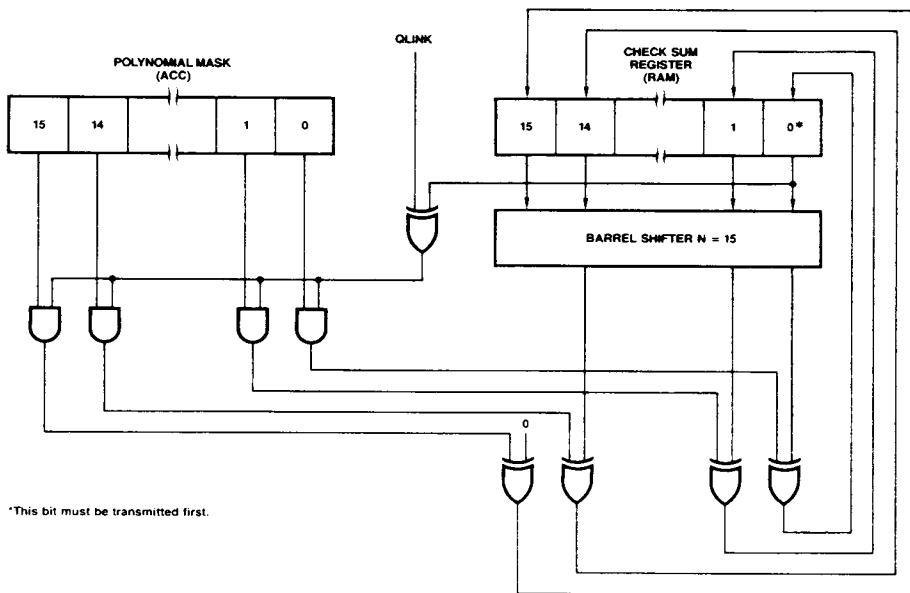
PF000320

**Figure 13. CRC Reverse Function**

## CYCLIC REDUNDANCY CHECK

| Instruction | B/W | Quad | | | RAM Address | | |
|---|---|---|---|---|---|---|---|
| CRCF | 1 | 10 | 0110 | 0011 | 00000<br>. .<br>11111 | R00<br>. .<br>R31 | RAM Reg 00<br>. . . .<br>RAM Reg 31 |

| Instruction | B/W | Quad | | | RAM Address | | |
|---|---|---|---|---|---|---|---|
| CRCR | 1 | 10 | 0110 | 1001 | 00000<br>. .<br>11111 | R00<br>. .<br>R31 | RAM Reg 00<br>. . . .<br>RAM Reg 31 |

## Y BUS AND STATUS – CYCLIC REDUNDANCY CHECK

| Instruction | Opcode | B/W | Y – Bus | Flag3 | Flag2 | Flag1 | LINK | OVR | N | C | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CRCF | | 1 = W | $Y_i \leftarrow [(QLINK \oplus RAM_{15}) \cdot ACC_i]$<br>$\oplus RAM_{i-1}$ for i = 15 to 1<br>$Y_0 \leftarrow [(QLINK \oplus RAM_{15}) \cdot ACC_0] \oplus 0$ | NC | NC | NC | $RAM_{15}{}^*$ | 0 | U | 0 | U |
| CRCR | | 1 = W | $Y_i \leftarrow [(QLINK \oplus RAM_0) \cdot ACC_i]$<br>$\oplus RAM_{i+1}$ for i = 14 to 0<br>$Y_{15} \leftarrow [(QLINK \oplus RAM_0) \cdot ACC_{15}] \oplus 0$ | NC | NC | NC | $RAM_0{}^*$ | 0 | U | 0 | U |

*QLINK is loaded with the shifted out bit from the checksum register.

U = Update
NC = No Change
0 = Reset
1 = Set
i = 0 to 15 when not specified

24

# STATUS INSTRUCTIONS

Status Instructions – The Set Status Instruction contains a single indicator. This indicator specifies which bit or group of bits, contained in the status register (Figure 14), are to be set (forced to a ONE).

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Flag3 | Flag2 | Flag1 | LINK | OVR | N | C | Z |
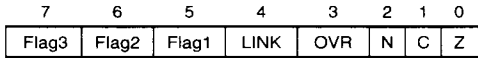
MPR-775

### Figure 14. Status Byte

The Reset Status Instruction contains a single indicator. This indicator specifies which bit or group of bits, contained in the status register, are to be reset (forced to ZERO).

The Store Status Instruction contains two indicators; byte/word and a second indicator that specifies the destination of the status register. The Store Status Instruction allows the status of the processor to be saved and restored later, which is an especially useful function for interrupt handling.

The status register is always stored in the lower byte of the RAM or the ACC register. Depending upon byte or word mode the upper byte is unchanged or loaded with all ZEROs respectively.

The Load Status instructions are included in the single operand and two operand instruction types.

**The Test Status Instructions** contain a single indicator which specifies which one of the 12 possible test conditions are to be placed on the Conditional-Test output. Besides the eight bits in the Status register (QZ, QC, QN, QOVR, QLINK, QFlag1, QFlag 2, and QFlag3), four logical functions (QN $\oplus$ QOVR), (QN $\oplus$ QOVR) + QZ, QZ + $\overline{QC}$ and LOW may also be selected. These functions are useful in testing results of Two's Complement and unsigned number arithmetic operations. The status register may also be tested via the bidirectional T bus. The code to test the status register via T bus is similar to the code used by instruction lines $I_1$ to $I_4$ as shown below. Instruction lines $I_{0-4}$ have priority over T bus for testing the status register on CT output. See the discussion on the status register for a full description.

| $T_4$ $I_4$ | $T_3$ $I_3$ | $T_2$ $I_2$ | $T_1$ $I_1$ | CT |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | (N $\oplus$ OVR) + Z |
| 0 | 0 | 0 | 1 | N $\oplus$ OVR |
| 0 | 0 | 1 | 0 | Z |
| 0 | 0 | 1 | 1 | OVR |
| 0 | 1 | 0 | 0 | LOW* |
| 0 | 1 | 0 | 1 | C |
| 0 | 1 | 1 | 0 | Z + $\overline{C}$ |
| 0 | 1 | 1 | 1 | N |
| 1 | 0 | 0 | 0 | LINK |
| 1 | 0 | 0 | 1 | Flag1 |
| 1 | 0 | 1 | 0 | Flag2 |
| 1 | 0 | 1 | 1 | Flag3 |

*LOW = CT is forced LOW

### STATUS

```
           15 14 13 12  9 8   5 4           0
SETST     0  Quad 1011 1010   Opcode

RSTST     0  Quad 1010 1010   Opcode

SVSTR    B/W Quad 0111 1010   RAM Address/Dest

SVSTNR   B/W Quad 0111 1010   Destination
```

## STATUS INSTRUCTIONS

| Instruction | B/W | Quad | | | Opcode | | |
|---|---|---|---|---|---|---|---|
| SETST | 0 | 11 | 1011 | 1010 | 00011 | SONCZ | Set OVR, N, C, Z |
| | | | | | 00101 | SL | Set LINK |
| | | | | | 00110 | SF1 | Set Flag1 |
| | | | | | 01001 | SF2 | Set Flag2 |
| | | | | | 01010 | SF3 | Set Flag3 |

| Instruction | B/W | Quad | | | Opcode | | |
|---|---|---|---|---|---|---|---|
| RSTST | 0 | 11 | 1010 | 1010 | 00011 | RONCZ | Reset OVR, N, C, Z |
| | | | | | 00101 | RL | Reset LINK |
| | | | | | 00110 | RF1 | Reset Flag1 |
| | | | | | 01001 | RF2 | Reset Flag2 |
| | | | | | 01010 | RF3 | Reset Flag3 |

| Instruction | B/W | Quad | | | RAM Address/Dest | | |
|---|---|---|---|---|---|---|---|
| SVSTR | 0 = B  1 = W | 10 | 0111 | 1010 | 00000 | R00 | RAM Reg 00 |
| | | | | | .. | .. | .... |
| | | | | | 11111 | R31 | RAM Reg 31 |

| | | | | | Destination | | |
|---|---|---|---|---|---|---|---|
| SVSTNR | 0 = B  1 = W | 11 | 0111 | 1010 | 00000 | NRY | Y Bus |
| | | | | | 00001 | NRA | ACC |

## STATUS INSTRUCTIONS

| Instruction | B/W | Quad | | | Opcode (CT) | | |
|---|---|---|---|---|---|---|---|
| Test | 0 | 11 | 1001 | 1010 | 00000 | TNOZ | Test (N⊕OVR) + Z |
| | | | | | 00010 | TNO | Test N⊕OVR |
| | | | | | 00100 | TZ | Test Z |
| | | | | | 00110 | TOVR | Test OVR |
| | | | | | 01000 | TLOW | Test LOW |
| | | | | | 01010 | TC | Test C |
| | | | | | 01100 | TZC | Test Z + $\overline{C}$ |
| | | | | | 01110 | TN | Test N |
| | | | | | 10000 | TL | Test LINK |
| | | | | | 10010 | TF1 | Test Flag1 |
| | | | | | 10100 | TF2 | Test Flag2 |
| | | | | | 10110 | TF3 | Test Flag3 |

Note: $\overline{IEN}$ · test status instruction has priority over $T_{1-4}$ instruction.

## Y BUS AND STATUS – FOR STATUS INSTRUCTIONS

| Instruction | Opcode | Description | B/W | Y – Bus | Flag3 | Flag2 | Flag1 | LINK | OVR | N | C | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SETST | SONCZ | Set OVR, N, C, Z | 0 = B | $Y_i \leftarrow 1$ for i = 0 to 15 | NC | NC | NC | NC | 1 | 1 | 1 | 1 |
| | SL | Set LINK | | | NC | NC | NC | 1 | NC | NC | NC | NC |
| | SF1 | Set Flag1 | | | NC | NC | 1 | NC | NC | NC | NC | NC |
| | SF2 | Set Flag2 | | | NC | 1 | NC | NC | NC | NC | NC | NC |
| | SF3 | Set Flag3 | | | 1 | NC | NC | NC | NC | NC | NC | NC |
| RSTST | RONCZ | Reset OVR, N, C, Z | 0 = B | $Y_i \leftarrow 0$ for i = 0 to 15 | NC | NC | NC | NC | 0 | 0 | 0 | 0 |
| | RL | Reset LINK | | | NC | NC | NC | 0 | NC | NC | NC | NC |
| | RF1 | Reset Flag1 | | | NC | NC | 0 | NC | NC | NC | NC | NC |
| | RF2 | Reset Flag2 | | | NC | 0 | NC | NC | NC | NC | NC | NC |
| | RF3 | Reset Flag3 | | | 0 | NC | NC | NC | NC | NC | NC | NC |
| SVSTR SVSTNR | | Save Status* | 0 = B, 1 = W | $Y_i \leftarrow$ Status for i-0 to 7; $Y_i \leftarrow 0$ for i = 8 to 15 | NC | NC | NC | NC | NC | NC | NC | NC |
| Test | TNOZ | Test (N⊕OVR) + Z | 0 = B | ** | NC | NC | NC | NC | NC | NC | NC | NC |
| | TNO | Test N⊕OVR | | | NC | NC | NC | NC | NC | NC | NC | NC |
| | TZ | Test Z | | | NC | NC | NC | NC | NC | NC | NC | NC |
| | TOVR | Test OVR | | | NC | NC | NC | NC | NC | NC | NC | NC |
| | TLOW | Test LOW | | | NC | NC | NC | NC | NC | NC | NC | NC |
| | TC | Test C | | | NC | NC | NC | NC | NC | NC | NC | NC |
| | TZC | Test Z + $\overline{C}$ | | | NC | NC | NC | NC | NC | NC | NC | NC |
| | TN | Test N | | | NC | NC | NC | NC | NC | NC | NC | NC |
| | TL | Test LINK | | | NC | NC | NC | NC | NC | NC | NC | NC |
| | TF1 | Test Flag1 | | | NC | NC | NC | NC | NC | NC | NC | NC |
| | TF2 | Test Flag2 | | | NC | NC | NC | NC | NC | NC | NC | NC |
| | TF3 | Test Flag3 | | | NC | NC | NC | NC | NC | NC | NC | NC |

U = Update
NC = No Change
0 = Reset
1 = Set
i = 0 to 15 when not specified

*In byte mode only the lower byte from the Y bus is loaded into the RAM or ACC and in word mode all 16-bits from the Y bus are loaded into the RAM or ACC.

**Y-Bus is Undefined.

26

# NO-OP INSTRUCTION

The NO-OP Instruction has a fixed 16-bit code. This instruction does not change any internal registers in the Am29C116. It preserves the status register, RAM register and the ACC register.

### NO OPERATION FIELD DEFINITION

| | 15 14 | 13 12 | 9 8 | 5 4 | 0 |
|---|---|---|---|---|---|
| NOOP | 0 | 11 | 1000 | 1010 | 00000 |

## NO-OP INSTRUCTION

| Instruction | B/W | Quad | | | |
|---|---|---|---|---|---|
| NOOP | 0 | 11 | 1000 | 1010 | 00000 |

## Y BUS AND STATUS – NO-OP INSTRUCTION

| Instruction | Opcode | B/W | Y – Bus | Flag3 | Flag2 | Flag1 | LINK | OVR | N | C | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|
| NOOP | | 0 = B | * | NC | NC | NC | NC | NC | NC | NC | NC |

SRC = Source
U = Update
NC = No Change
0 = Reset
1 = Set
i = 0 to 15 when not specified
*Y-Bus is undefined.

# SUMMARY OF MNEMONICS

## Instruction Type

| | |
|---|---|
| SOR | Single Operand RAM |
| SONR | Single Operand Non-RAM |
| TOR1 | Two Operand RAM (Quad 0) |
| TOR2 | Two Operand RAM (Quad 2) |
| TONR | Two Operand Non-RAM |
| SHFTR | Single Bit Shift RAM |
| SHFTNR | Single Bit Shift Non-RAM |
| ROTR1 | Rotate n Bits RAM (Quad 0) |
| ROTR2 | Rotate n Bits RAM (Quad 1) |
| ROTNR | Rotate n Bits Non-RAM |
| BOR1 | Bit Oriented RAM (Quad 3) |
| BOR2 | Bit Oriented RAM (Quad 2) |
| BONR | Bit Oriented Non-RAM |
| ROTM | Rotate and Merge |
| ROTC | Rotate and Compare |
| PRT1 | Prioritize RAM; Type 1 |
| PRT2 | Prioritize RAM; Type 2 |
| PRT3 | Prioritize RAM; Type 3 |
| PRTNR | Prioritize Non-RAM |
| CRCF | Cyclic Redundancy Check Forward |
| CRCR | Cyclic Redundancy Check Reverse |
| NOOP | No Operation |
| SETST | Set Status |
| RSTST | Reset Status |
| SVSTR | Save Status RAM |
| SVSTNR | Save Status Non-RAM |
| TEST | Test Status |

## SOURCE AND DESTINATION

### Single Operand

| | |
|---|---|
| SORA | Single Operand RAM to ACC |
| SORY | Single Operand RAM to Y Bus |
| SORS | Single Operand RAM to Status |
| SOAR | Single Operand ACC to RAM |
| SODR | Single Operand D to RAM |
| SOIR | Single Operand I to RAM |
| SOZR | Single Operand 0 to RAM |
| SOZER | Single Operand D(0E) to RAM |
| SOSER | Single Operand D(SE) to RAM |
| SORR | Single Operand RAM to RAM |
| SOA | Single Operand ACC |
| SOD | Single Operand D |
| SOI | Single Operand I |
| SOZ | Single Operand 0 |
| SOZE | Single Operand D(0E) |
| SOSE | Single Operand D(SE) |
| NRY | Non-RAM Y Bus |
| NRA | Non-RAM ACC |
| NRS | Non-RAM Status |
| NRAS | Non-RAM ACC, Status |

## Two Operand

| | |
|---|---|
| TORAA | Two Operand RAM, ACC to ACC |
| TORIA | Two Operand RAM, I to ACC |
| TODRA | Two Operand D, RAM to ACC |
| TORAY | Two Operand RAM, ACC to Y Bus |
| TORIY | Two Operand RAM, I to Y Bus |
| TODRY | Two Operand D, RAM to Y Bus |
| TORAR | Two Operand RAM, ACC to RAM |
| TORIR | Two Operand RAM, I to RAM |
| TODRR | Two Operand D, RAM to RAM |
| TODAR | Two Operand D, ACC to RAM |
| TOAIR | Two Operand ACC, I to RAM |
| TODIR | Two Operand D, I to RAM |
| TODA | Two Operand D, ACC |
| TOAI | Two Operand ACC, I |
| TODI | Two Operand D, I |

## Single Bit Shift

| | |
|---|---|
| SHRR | Shift RAM, Store in RAM |
| SHDR | Shift D, Store in RAM |
| SHA | Shift ACC |
| SHD | Shift D |

## Rotate n Bits

| | |
|---|---|
| RTRA | Rotate RAM, Store in ACC |
| RTRY | Rotate RAM, Place on Y Bus |
| RTRR | Rotate RAM, Store in RAM |
| RTAR | Rotate ACC, Store in RAM |
| RTDR | Rotate D, Store in RAM |
| RTDY | Rotate D, Place on Y Bus |
| RTDA | Rotate D, Store in ACC |
| RTAY | Rotate ACC, Place on Y Bus |
| RTAA | Rotate ACC, Store in ACC |

## Rotate and Merge

| | |
|---|---|
| MDAI | Merge Disjoint Bits of D and ACC Using I as Mask and Store in ACC |
| MDAR | Merge Disjoint Bits of D and ACC Using RAM as Mask and Store in ACC |
| MDRI | Merge Disjoint Bits of D and RAM Using I as Mask and Store in RAM |
| MDRA | Merge Disjoint Bits of D and RAM Using ACC as Mask and Store in RAM |
| MARI | Merge Disjoint Bits of ACC and RAM Using I as Mask and Store in RAM |
| MRAI | Merge Disjoint Bits of RAM and ACC Using I as Mask and Store in ACC |

## Rotate and Compare

| | |
|---|---|
| CDAI | Compare Unmasked Bits of D and ACC Using I as Mask |

Mnemonics copyright © 1980
Advanced Micro Devices, Inc.

28

| | | |
|---|---|---|
| CDRI | Compare Unmasked Bits of D and RAM Using I as Mask | |
| CDRA | Compare Unmasked Bits of D and RAM Using ACC as Mask | |
| CRAI | Compare Unmasked Bits of RAM and ACC Using I as Mask | |

**Prioritize**

| | |
|---|---|
| PR1A | ACC as Destination for Prioritize Type 1 |
| PR1Y | Y Bus as Destination for Prioritize Type 1 |
| PR1R | RAM as Destination for Prioritize Type 1 |
| PRT1A | ACC as Source for Prioritize Type 1 |
| PR1D | D as Source for Prioritize Type 1 |
| PR2A | ACC as Destination for Prioritize Type 2 |
| PR2Y | Y Bus as Destination for Prioritize Type 2 |
| PR3R | RAM as Source for Prioritize Type 3 |
| PR3A | ACC as Source for Prioritize Type 3 |
| PR3D | D as Source for Prioritize Type 3 |
| PRTA | ACC as source for Prioritize Type Non-RAM |
| PRTD | D as Source for Prioritize Type Non-RAM |
| PRA | ACC as Mask for Prioritize Type 2, 3, and Non-RAM |
| PRZ | Mask Equal to Zero for Prioritize Type 2, 3, and Non-RAM |
| PRI | I as Mask for Prioritize Type 2, 3, and Non-RAM |

## OPCODE

**Addition**

| | |
|---|---|
| ADD | Add without Carry |
| ADDC | Add with Carry |
| A2NA | Add $2^n$ to ACC |
| A2NR | Add $2^n$ to RAM |
| A2NDY | Add $2^n$ to D, Place on Y Bus |

**Subtraction**

| | |
|---|---|
| SUBR | Subtract R from S without Carry |
| SUBRC | Subtract R from S with Carry |
| SUBS | Subtract S from R without Carry |
| SUBSC | Subtract S from R with Carry |
| S2NR | Subtract $2^n$ from RAM |
| S2NA | Subtract $2^n$ from ACC |
| S2NDY | Subtract $2^n$ from D, Place on Y Bus |

**Logical Operations**

| | |
|---|---|
| AND | Boolean AND |
| NAND | Boolean NAND |
| EXOR | Boolean EXOR |
| NOR | Boolean NOR |
| OR | Boolean OR |
| EXNOR | Boolean EXNOR |

**SHIFTS**

| | |
|---|---|
| SHUPZ | Shift Up Towards MSB with 0 Insert |
| SHUP1 | Shift Up Towards MSB with 1 Insert |
| SHUPL | Shift Up Towards MSB with LINK Insert |

| | | |
|---|---|---|
| SHDNZ | Shift Down Towards LSB with 0 Insert | |
| SHDN1 | Shift Down Towards LSB with 1 Insert | |
| SHDNL | Shift Down Towards LSB with LINK Insert | |
| SHDNC | Shift Down Towards LSB with Carry Insert | |
| SHDNOV | Shift Down Towards LSB with Sign EXOR Overflow Insert | |

**Loads**

| | |
|---|---|
| LD2NR | Load $2^n$ into RAM |
| LDC2NR | Load $\overline{2^n}$ into RAM |
| LD2NA | Load $2^n$ into ACC |
| LDC2NA | Load $\overline{2^n}$ into ACC |
| LD2NY | Place $2^n$ on Y Bus |
| LDC2NY | Place $\overline{2^n}$ on Y Bus |

**Bit Oriented**

| | |
|---|---|
| SETNR | Set RAM, Bit n |
| SETNA | Set ACC, Bit n |
| SETND | Set D, Bit n |
| SONCZ | Set OVR, N, C, Z, in Status Register |
| SL | Set LINK Bit in Status Register |
| SF1 | Set Flag1 Bit in Status Register |
| SF2 | Set Flag2 Bit in Status Register |
| SF3 | Set Flag3 Bit in Status Register |
| RSTNR | Reset RAM, Bit n |
| RSTNA | Reset ACC, Bit n |
| RSTND | Reset D, Bit n |
| RONCZ | Reset OVR, N, C, Z, in Status Register |
| RL | Reset LINK Bit in Status Register |
| RF1 | Reset Flag1 Bit in Status Register |
| RF2 | Reset Flag2 Bit in Status Register |
| RF3 | Reset Flag3 Bit in Status Register |
| TSTNR | Test RAM, Bit n |
| TSTNA | Test ACC, Bit n |
| TSTND | Test D, Bit n |

**Arithmetic Operations**

| | |
|---|---|
| MOVE | Move and Update Status |
| COMP | Complement (1's Complement) |
| INC | Increment |
| NEG | Two's Complement |

**Conditional Test**

| | |
|---|---|
| TNOZ | Test (N $\oplus$ OVR) + Z |
| TNO | Test N $\oplus$ OVR |
| TZ | Test Zero Bit |
| TOVR | Test Overflow Bit |
| TLOW | Test for LOW |
| TC | Test Carry Bit |
| TZC | Test Z + $\overline{C}$ |
| TN | Test Negative Bit |
| TL | Test LINK Bit |
| TF1 | Test Flag1 Bit |
| TF2 | Test Flag2 Bit |
| TF3 | Test Flag3 Bit |

Mnemonics copyright © 1980
Advanced Micro Devices, Inc.

29

# APPLICATIONS

**Minimum System Cycle Time Calculations for the Am29C116**
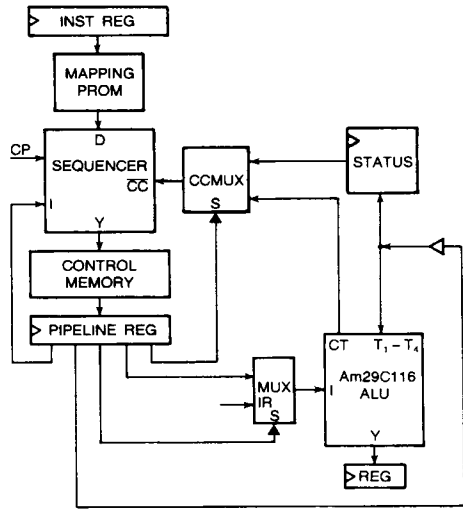


BD005953

**Figure 15. System Block Diagram**

## DATA PATH TIMING ANALYSIS

| | | | | 29C116 | 29C116-1 | 29C116-2 |
|---|---|---|---|---|---|---|
| I. | **Without Any External Logic** | | | | | |
| | a. | Pipeline Register | CP–Q | 12 ns | 12 ns | 12 ns |
| | | RALU (29C116) | I–T | 84 | 73 | 60 |
| | | Status Register | Setup | 4 | 4 | 4 |
| | | **Cycle Time:** | | **100 ns** | **89 ns** | **76 ns** |
| | b. | Pipeline Register | CP–Q | 12 ns | 12 ns | 12 ns |
| | | RALU (29C116) | I–Y | 79 | 65 | 57 |
| | | Data Register | Setup | 4 | 4 | 4 |
| | | **Cycle Time:** | | **95 ns** | **81 ns** | **73 ns** |
| II. | **With Multiplexers for Address, N–Count, etc.** | | | | | |
| | a. | Pipeline Register | CP–Q | 12 ns | 12 ns | 12 ns |
| | | Multiplexer | Sel–Y | 15 | 15 | 15 |
| | | RALU (29C116) | I–T | 84 | 73 | 60 |
| | | Status Register | Setup | 4 | 4 | 4 |
| | | **Cycle Time:** | | **115 ns** | **104 ns** | **91 ns** |
| | b. | Pipeline Register | CP–Q | 12 ns | 12 ns | 12 ns |
| | | Multiplexer | Sel–Y | 15 | 15 | 15 |
| | | RALU (29C116) | I–Y | 79 | 65 | 57 |
| | | Data Register | Setup | 4 | 4 | 4 |
| | | **Cycle Time:** | | **110 ns** | **96 ns** | **88 ns** |

## The Use of an External Status Register in Reducing Microcycle Length

The standard connection of the CT pin of the Am29C116 and microcycle length calculation arising from that connection are shown below:

### CRITICAL PATH TIMING (FIGURE 16-1)

| Part Number | Path | Maximum Commercial Delay (ns) |
|---|---|---|
| Pipeline Register | CP–Q | 12 |
| Am29C116-2 | I, T–CT | 25 |
| CC–MUX | D–W | 7 |
| Am29C10A-1 | CC–Y | 26 |
| Control Memory | $t_{AA}$ | 40 |
| Pipeline Register | Setup | 4 |
| | | 114 |

While 114-ns cycle time is quite fast, it can be improved by using an external register for status testing.

### CRITICAL PATH TIMING (FIGURE 16-2)

| Part Number | Path | Maximum Commercial Delay (ns) |
|---|---|---|
| Status Reg | CP–Y | 12 |
| CC–MUX | Sel–W | 15 |
| Am29C10A-1 | CC–Y | 26 |
| Control Memory | $t_{AA}$ | 40 |
| Pipeline Register | Setup | 4 |
| | | 97 |

The cycle time has been reduced from 114 ns to 97 ns.
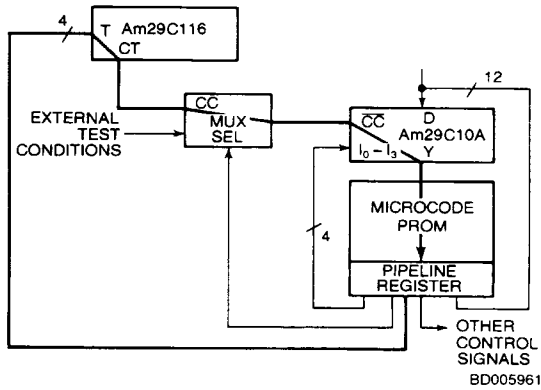


Figure 16-1.
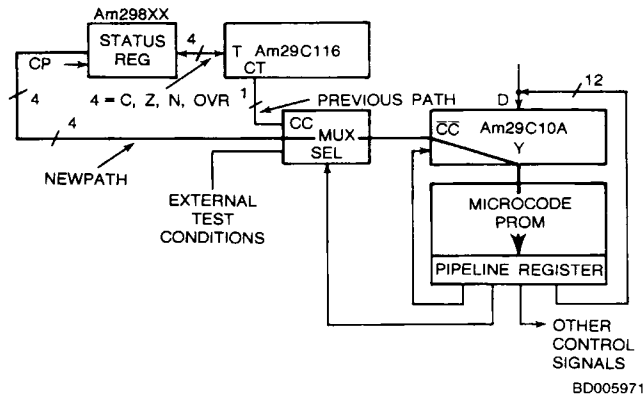


Figure 16-2.

## ABSOLUTE MAXIMUM RATINGS

Storage Temperature ............................. −65 to +150°C
(Case) Temperature Under Bias .............. −55 to +125°C
Supply Voltage to
  Ground Potential Continuous ............ −0.3 V to +7.0 V
DC Voltage Applied to Outputs For
  High Output State ................. −0.3 V to $V_{CC}$ +0.3 V
DC Input Voltage ..................... −0.3 V to $V_{CC}$ +0.3 V
DC Output Current, Into LOW Outputs ................ 30 mA
DC Input Current ........................... −10 mA to +10 mA

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

## OPERATING RANGES

Commercial (C) Devices
  Ambient Temperature ($T_A$) ...................... 0 to +70°C
  Supply Voltage ($V_{CC}$) ................... +4.5 V to +5.5 V

Military* (M) Devices
  Ambient Temperature ($T_A$) ................ −55 to +125°C
  Supply Voltage ($V_{CC}$) ................... +4.5 V to +5.5 V

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

*Military Product 100% tested at $T_A$ = +25°C, +125°C, and −55°C.

## DC CHARACTERISTICS over operating ranges unless otherwise specified (for APL Products, Group A, Subgroups 1, 2, 3 are tested unless otherwise noted)

| Parameter Symbol | Parameter Description | Test Conditions (Note 1) | | | Min. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| $V_{OH}$ | Output HIGH Voltage | $V_{CC}$ = Min. $V_{IN}$ = $V_{IH}$ or $V_{IL}$ | $I_{OH}$ = −1.6 mA/−1.2 mA (COM'L/MIL) | | 2.4 | | V |
| $V_{OL}$ | Output LOW Voltage | $V_{CC}$ = Min. $V_{IN}$ = $V_{IH}$ or $V_{IL}$ | $I_{OL}$ = 16 mA/12 mA (COM'L/MIL) | | | 0.5 | V |
| $V_{IH}$ | Guaranteed Input Logical HIGH Voltage (Note 2) | | | | 2.0 | | V |
| $V_{IL}$ | Guaranteed Input Logical LOW Voltage (Note 2) | | | | | 0.8 | V |
| $I_{IL}$ | Input LOW Current | $V_{CC}$ = Max. $V_{IN}$ = 0.5 Volts | | | | −10 | μA |
| $I_{IH}$ | Input HIGH Current | $V_{CC}$ = Max. $V_{IN}$ = $V_{CC}$ − 0.5 V | | | | 10 | μA |
| $I_{OZH}$ | Off State (HIGH Impedance) Output Current | $V_{CC}$ = Max. $V_O$ = 2.4 Volts | | | | 10 | μA |
| $I_{OZL}$ | Off State (HIGH Impedance) Output Current | $V_{CC}$ = Max. $V_O$ = 0.5 Volts | | | | −10 | μA |
| $I_{CC}$ | Static Power Supply Current (Note 3) | $V_{CC}$ = Max. $I_O$ = 0 μA | COM'L $T_A$ = 0 to +70°C | (Note 4) CMOS $V_{IN}$ = $V_{CC}$ or GND | | 120 | mA |
| | | | | (Note 4) TTL $V_{IN}$ = 0.5 V or 2.4 V | | 170 | |
| | | | MIL $T_A$ = −55 to +125°C | (Note 4) CMOS $V_{IN}$ = $V_{CC}$ or GND | | 145 | |
| | | | | (Note 4) TTL $V_{IN}$ = 0.5 V or 2.4 V | | 200 | |
| $C_{PD}$ | Power Dissipation Capacitance (Note 5) | $V_{CC}$ = 5.0 V $T_A$ = 25°C No Load | | | 850 pF Typical | | |

Notes: 1. $V_{CC}$ conditions shown as Min. or Max. refer to ±10% $V_{CC}$ limits.
2. These input levels provide zero-noise immunity and should only be statically tested in a noise-free environment (not functionally tested).
3. Worst-case $I_{CC}$ is measured at the lowest temperature in the specified operating range.
4. Use CMOS $I_{CC}$ when the device is driven by CMOS circuits and TTL $I_{CC}$ when the device is driven by TTL circuits.
5. $C_{PD}$ determines the dynamic current consumption:

$I_{CC}$ (Total) = $I_{CC}$ (Static) + ($C_{PD}$ + $nC_L$)$\frac{f}{2}$, where f is the clock frequency, $C_L$ output load capacitance, and n number of loads.

## 29C116 $I_{CC}$ vs Cycle Time
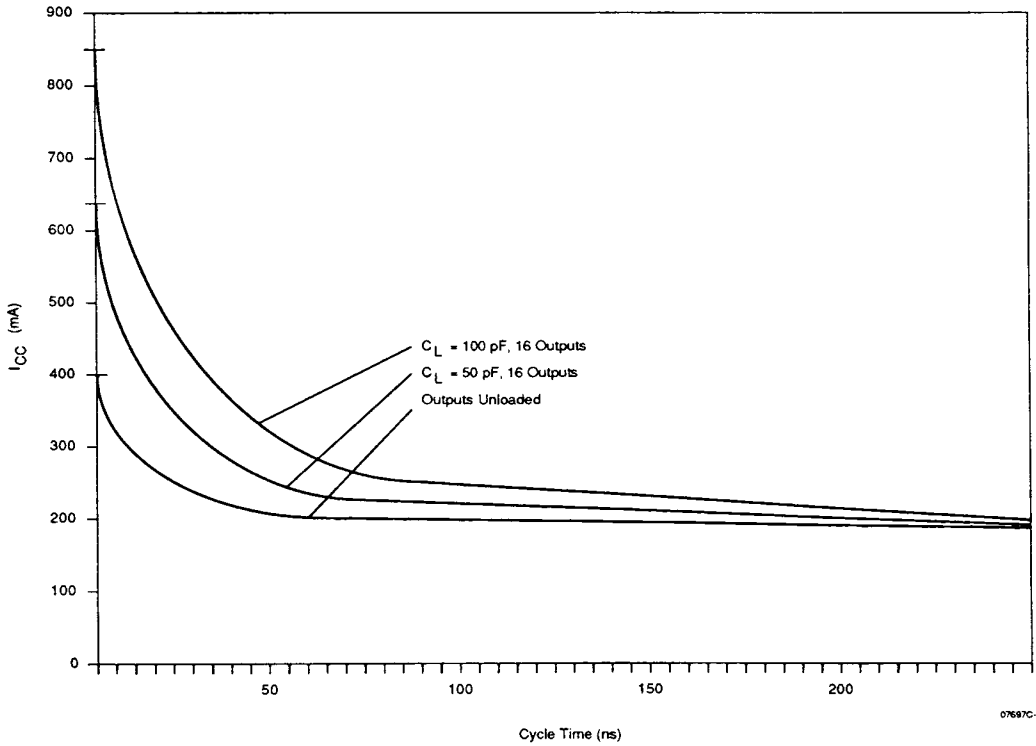


Note: Values are calculated as typical total $I_{CC}$ at $V_{CC} = 5.5$ V.

OP002660

# SWITCHING CHARACTERISTICS over COMMERCIAL operating range unless otherwise specified
($T_A$ = 0 to +70°C, $V_{CC}$ = 4.5 to 5.5 V, $C_L$ = 50 pF)

## Am29C116

### A. Combinational Delays (nsec)

| Input | Outputs | | |
|---|---|---|---|
| | $Y_{0-15}$ | $T_{1-4}$ | CT |
| $I_{0-4}$ (ADDR) | 79 | 84 | – |
| $I_{0-15}$ (DATA) | 79 | 84 | – |
| $I_{0-15}$ (INSTR) | 79 | 84 | 48 |
| DLE | 58* | 60 | – |
| $T_{1-4}$ | – | – | 39 |
| CP | 63 | 66 | 40 |
| $Y_{0-15}$ | 62* | 64 | – |
| $\overline{IEN}$ | – | – | 43 |

$Y_{0-15}$ must be stored in the Data Latch and its source disabled before the delay to $Y_{0-15}$ as an output can be measured.
*Guaranteed indirectly by other tests.

### B. Enable/Disable Times (nsec)
(Disable: $C_L$ = 5 pF, 0.5-V Change on Outputs)

| From Input | To Output | Enable | | Disable | |
|---|---|---|---|---|---|
| | | $t_{PZH}$ | $t_{PZL}$ | $t_{PHZ}$ | $t_{PLZ}$ |
| $\overline{OE}_Y$ | $Y_{0-15}$ | 22 | 22 | 22 | 22 |
| $OE_T$ | $T_{1-4}$ | 25 | 25 | 25 | 25 |

### C. Clock and Pulse Requirements (nsec)

| Input | Min. LOW Time | Min. HIGH Time |
|---|---|---|
| CP | 20 | 30 |
| DLE | – | 15 |
| $\overline{IEN}$ | 22 | – |

### D. Setup and Hold Times (nsec)

| Input | With Respect to | HIGH-to-LOW Transition | | LOW-to-HIGH Transition | | Comment | |
|---|---|---|---|---|---|---|---|
| | | Setup | Hold | Setup | Hold | | |
| $I_{0-4}$ (RAM ADDR) | CP | ($t_{s1}$) 24 | ($t_{h1}$) 0 | – | – | Single ADDR (Source) | |
| $I_{0-4}$ (RAM ADDR) | CP and $\overline{IEN}$ both LOW | ($t_{s2}$) 5 | – | – | ($t_{h7}$) 2 | Two ADDR (Destination) | |
| $I_{0-15}$ (DATA) | CP | – | – | ($t_{s8}$) 65 | ($t_{h8}$) 0 | | |
| $I_{0-15}$ (INSTR) | CP | ($t_{s3}$) 38 | ($t_{h3}$) 17 | ($t_{s9}$) 65 | ($t_{h9}$) 2 | | |
| $\overline{IEN}$ HIGH | CP | ($t_{s4}$) 10 | – | – | ($t_{h10}$) 2 | Disable | |
| $\overline{IEN}$ LOW | CP | – | ($t_{s5}$) 20 | – | ($t_{h5}$) 1 | ($t_{s11}$) 22 | – | ($t_{h11}$) 1** | – | Enable | Immediate first cycle |
| $\overline{SRE}$ | CP | – | – | ($t_{s12}$) 17 | ($t_{h12}$) 0 | | |
| Y | CP | – | – | ($t_{s13}$) 44 | ($t_{h13}$) 0 | | |
| Y | DLE | ($t_{s6}$) 10 | ($t_{h6}$) 6 | – | – | | |
| DLE | CP | – | – | ($t_{s14}$) 45 | ($t_{h14}$) 0 | | |

** Status register and accumulator destination only.

## Am29C116-1

### A. Combinational Delays (nsec)

| Input | Outputs | | |
|---|---|---|---|
| | $Y_{0-15}$ | $T_{1-4}$ | CT |
| $I_{0-4}$ (ADDR) | 65 | 73 | – |
| $I_{0-15}$ (DATA) | 65 | 73 | – |
| $I_{0-15}$ (INSTR) | 65 | 73 | 30 |
| DLE | 55* | 55 | – |
| $T_{1-4}$ | – | – | 27 |
| CP | 60 | 66 | 37 |
| $Y_{0-15}$ | 53* | 53 | – |
| $\overline{IEN}$ | – | – | 25 |

$Y_{0-15}$ must be stored in the Data Latch and its source disabled before the delay to $Y_{0-15}$ as an output can be measured.
*Guaranteed indirectly by other tests.

### B. Enable/Disable Times (nsec)
(Disable: $C_L = 5$ pF, 0.5-V Change on Outputs)

| From Input | To Output | Enable | | Disable | |
|---|---|---|---|---|---|
| | | $t_{PZH}$ | $t_{PZL}$ | $t_{PHZ}$ | $t_{PLZ}$ |
| $\overline{OE}_Y$ | $Y_{0-15}$ | 22 | 22 | 22 | 22 |
| $OE_T$ | $T_{1-4}$ | 22 | 22 | 22 | 22 |

### C. Clock and Pulse Requirements (nsec)

| Input | Min. LOW Time | Min. HIGH Time |
|---|---|---|
| CP | 20 | 30 |
| DLE | – | 15 |
| $\overline{IEN}$ | 20 | – |

### D. Setup and Hold Times (nsec)

| Input | With Respect to | HIGH-to-LOW Transition | | LOW-to-HIGH Transition | | Comment | |
|---|---|---|---|---|---|---|---|
| | | Setup | Hold | Setup | Hold | | |
| $I_{0-4}$ (RAM ADDR) | CP | $(t_{s1})$ 13 | $(t_{h1})$ 0 | – | – | Single ADDR (Source) | |
| $I_{0-4}$ (RAM ADDR) | CP and $\overline{IEN}$ both LOW | $(t_{s2})$ 5 | – | – | $(t_{h7})$ 0 | Two ADDR (Destination) | |
| $I_{0-15}$ (DATA) | CP | – | – | $(t_{s8})$ 60 | $(t_{h8})$ 0 | | |
| $I_{0-15}$ (INSTR) | CP | $(t_{s3})$ 24 | $(t_{h3})$ 12 | $(t_{s9})$ 60 | $(t_{h9})$ 0 | | |
| $\overline{IEN}$ HIGH | CP | $(t_{s4})$ 5 | – | – | $(t_{h10})$ 2 | Disable | |
| $\overline{IEN}$ LOW | CP | – $(t_{s5})$ 5 | – $(t_{h5})$ 1 | $(t_{s11})$ 10 – | $(t_{h11})$ 1** – | Enable | Immediate first cycle |
| $\overline{SRE}$ | CP | – | – | $(t_{s12})$ 12 | $(t_{h12})$ 0 | | |
| Y | CP | – | – | $(t_{s13})$ 42 | $(t_{h13})$ 0 | | |
| Y | DLE | $(t_{s6})$ 6 | $(t_{h6})$ 5 | – | – | | |
| DLE | CP | – | – | $(t_{s14})$ 43 | $(t_{h14})$ 0 | | |

** Status register and accumulator destination only.

# SWITCHING CHARACTERISTICS over COMMERCIAL operating range

## Am29C116-2

### A. Combinational Delays (nsec)

| Input | Outputs | | |
|---|---|---|---|
| | $Y_{0-15}$ | $T_{1-4}$ | CT |
| $I_{0-4}$ (ADDR) | 57 | 60 | – |
| $I_{0-15}$ (DATA) | 57 | 60 | – |
| $I_{0-15}$ (INSTR) | 57 | 60 | 25 |
| DLE | 45* | 45 | – |
| $T_{1-4}$ | – | – | 25 |
| CP | 50 | 55 | 29 |
| $Y_{0-15}$ | 45* | 45 | – |
| $\overline{IEN}$ | – | – | 25 |

$Y_{0-15}$ must be stored in the Data Latch and its source disabled before the delay to $Y_{0-15}$ as an output can be measured.
*Guaranteed indirectly by other tests.

### B. Enable/Disable Times (nsec)
(Disable: $C_L$ = 5 pF, 0.5-V Change on Outputs)

| From Input | To Output | Enable | | Disable | |
|---|---|---|---|---|---|
| | | $t_{PZH}$ | $t_{PZL}$ | $t_{PHZ}$ | $t_{PLZ}$ |
| $\overline{OE}_Y$ | $Y_{0-15}$ | 22 | 22 | 22 | 22 |
| $OE_T$ | $T_{1-4}$ | 22 | 22 | 22 | 22 |

### C. Clock and Pulse Requirements (nsec)

| Input | Min. LOW Time | Min. HIGH Time |
|---|---|---|
| CP | 20 | 30 |
| DLE | – | 15 |
| $\overline{IEN}$ | 20 | – |

### D. Setup and Hold Times (nsec)

| Input | With Respect to | HIGH-to-LOW Transition | | LOW-to-HIGH Transition | | Comment | |
|---|---|---|---|---|---|---|---|
| | | Setup | Hold | Setup | Hold | | |
| $I_{0-4}$ (RAM ADDR) | CP | $(t_{s1})$ 13 | $(t_{h1})$ 0 | – | – | Single ADDR (Source) | |
| $I_{0-4}$ (RAM ADDR) | CP and $\overline{IEN}$ both LOW | $(t_{s2})$ 5 | – | – | $(t_{h7})$ 0 | Two ADDR (Destination) | |
| $I_{0-15}$ (DATA) | CP | – | – | $(t_{s8})$ 50 | $(t_{h8})$ 0 | | |
| $I_{0-15}$ (INSTR) | CP | $(t_{s3})$ 22 | $(t_{h3})$ 10 | $(t_{s9})$ 50 | $(t_{h9})$ 0 | | |
| $\overline{IEN}$ HIGH | CP | $(t_{s4})$ 5 | – | – | $(t_{h10})$ 2 | Disable | |
| $\overline{IEN}$ LOW | CP | – $(t_{s5})$ 5 | – $(t_{h5})$ 1 | $(t_{s11})$ 10 – | $(t_{h11})$ 1** – | Enable | Immediate first cycle |
| $\overline{SRE}$ | CP | – | – | $(t_{s12})$ 12 | $(t_{h12})$ 0 | | |
| Y | CP | – | – | $(t_{s13})$ 39 | $(t_{h13})$ 0 | | |
| Y | DLE | $(t_{s6})$ 6 | $(t_{h6})$ 5 | – | – | | |
| DLE | CP | – | – | $(t_{s14})$ 43 | $(t_{h14})$ 0 | | |

** Status register and accumulator destination only.

**SWITCHING CHARACTERISTICS** over **MILITARY** operating range (for APL Products, Group A, Subgroups 9, 10, 11 are tested unless otherwise noted) ($T_A = -55$ to $+125°C$, $V_{CC} = 4.5$ to $5.5$ V, $C_L = 50$ pF).

### Am29C116

#### A. Combinational Delays (nsec)

| Input | Outputs | | |
|---|---|---|---|
| | $Y_{0-15}$ | $T_{1-4}$ | CT |
| $I_{0-4}$ (ADDR) | 100 | 103 | – |
| $I_{0-15}$ (DATA) | 100 | 103 | – |
| $I_{0-15}$ (INSTR) | 100 | 103 | 50 |
| DLE | 68*† | 70 | – |
| $T_{1-4}$ | – | – | 46 |
| CP | 76 | 83 | 48 |
| $Y_{0-15}$ | 70*† | 72 | – |
| $\overline{IEN}$ | – | – | 50 |

$Y_{0-15}$ must be stored in the Data Latch and its source disabled before the delay to $Y_{0-15}$ as an output can be measured.
*Guaranteed indirectly by other tests.

#### B. Enable/Disable Times (nsec)
(Disable: $C_L = 5$ pF, 0.5-V Change on Outputs)

| From Input | To Output | Enable | | Disable | |
|---|---|---|---|---|---|
| | | $t_{PZH}$ | $t_{PZL}$ | $t_{PHZ}$ | $t_{PLZ}$ |
| $\overline{OE}_Y$ | $Y_{0-15}$ | 25 | 25 | 25 | 25 |
| $OE_T$ | $T_{1-4}$ | 30 | 30 | 30 | 30 |

#### C. Clock and Pulse Requirements (nsec)

| Input | Min. LOW Time | Min. HIGH Time |
|---|---|---|
| CP | 33 | 50 |
| DLE | – | 20 |
| $\overline{IEN}$ | 33 | – |

#### D. Setup and Hold Times (nsec)

| Input | With Respect to | HIGH-to-LOW Transition | | LOW-to-HIGH Transition | | Comment | |
|---|---|---|---|---|---|---|---|
| | | Setup | Hold | Setup | Hold | | |
| $I_{0-4}$ (RAM ADDR) | CP | $(t_{s1})$ 24 | $(t_{h1})$ 1 | – | – | Single ADDR (Source) | |
| $I_{0-4}$ (RAM ADDR) | CP and $\overline{IEN}$ both LOW | $(t_{s2})$ 10 | – | – | $(t_{h7})$ 3 | Two ADDR (Destination) | |
| $I_{0-15}$ (DATA) | CP | – | – | $(t_{s8})$ 78 | $(t_{h8})$ 3 | | |
| $I_{0-15}$ (INSTR) | CP | $(t_{s3})$ 57 | $(t_{h3})$ 17 | $(t_{s9})$ 78 | $(t_{h9})$ 3 | | |
| $\overline{IEN}$ HIGH | CP | $(t_{s4})$ 10 | – | – | – | Disable | |
| $\overline{IEN}$ LOW | CP | – | $(t_{s5})$ 20 | – | $(t_{h5})$ 3 | $(t_{s11})$ 28 | – | $(t_{h11})$ 3** | – | Enable | Immediate first cycle |
| $\overline{SRE}$ | CP | – | – | $(t_{s12})$ 19 | $(t_{h12})$ 1 | | |
| Y | CP | – | – | $(t_{s13})$ 53 | $(t_{h13})$ 2 | | |
| Y | DLE | $(t_{s6})$ 11 | $(t_{h6})$ 7 | – | – | | |
| DLE | CP | – | – | $(t_{s14})$ 54 | $(t_{h14})$ 0 | | |

**Status register and accumulator destination only.

† Not included in Group A tests.

## Am29C116-1

### A. Combinational Delays (nsec)

| Input | Outputs | | |
|---|---|---|---|
| | $Y_{0-15}$ | $T_{1-4}$ | CT |
| $I_{0-4}$ (ADDR) | 75 | 75 | – |
| $I_{0-15}$ (DATA) | 75 | 75 | – |
| $I_{0-15}$ (INSTR) | 75 | 75 | 29 |
| DLE | 62*† | 62 | – |
| $T_{1-4}$ | – | – | 29 |
| CP | 67 | 75 | 39 |
| $Y_{0-15}$ | 60*† | 60 | – |
| $\overline{IEN}$ | – | – | 29 |

$Y_{0-15}$ must be stored in the Data Latch and its source disabled before the delay to $Y_{0-15}$ as an output can be measured.
*Guaranteed indirectly by other tests.

### B. Enable/Disable Times (nsec)
(Disable: $C_L$ = 5 pF, 0.5-V Change on Outputs)

| From Input | To Output | Enable | | Disable | |
|---|---|---|---|---|---|
| | | $t_{PZH}$ | $t_{PZL}$ | $t_{PHZ}$ | $t_{PLZ}$ |
| $\overline{OE}_Y$ | $Y_{0-15}$ | 25 | 25 | 20 | 20 |
| $\overline{OE}_T$ | $T_{1-4}$ | 25 | 25 | 16 | 16 |

### C. Clock and Pulse Requirements (nsec)

| Input | Min. LOW Time | Min. HIGH Time |
|---|---|---|
| CP | 25 | 15 |
| DLE | – | 15 |
| $\overline{IEN}$ | 15 | – |

### D. Setup and Hold Times (nsec)

| Input | With Respect to | HIGH-to-LOW Transition | | LOW-to-HIGH Transition | | Comment | |
|---|---|---|---|---|---|---|---|
| | | Setup | Hold | Setup | Hold | | |
| $I_{0-4}$ (RAM ADDR) | CP | $(t_{s1})$ 12 | $(t_{h1})$ 1 | – | – | Single ADDR (Source) | |
| $I_{0-4}$ (RAM ADDR) | CP and $\overline{IEN}$ both LOW | $(t_{s2})$ 7 | – | – | $(t_{h7})$ 0 | Two ADDR (Destination) | |
| $I_{0-15}$ (DATA) | CP | – | – | $(t_{s8})$ 65 | $(t_{h8})$ 0 | | |
| $I_{0-15}$ (INSTR) | CP | $(t_{s3})$ 27 | $(t_{h3})$ 12 | $(t_{s9})$ 65 | $(t_{h9})$ 2 | | |
| $\overline{IEN}$ HIGH | CP | $(t_{s4})$ 5 | – | – | $(t_{h10})$ 2 | Disable | |
| $\overline{IEN}$ LOW | CP | – $(t_{s5})$ 7 | – $(t_{h5})$ 3 | $(t_{s11})$ 12 – | $(t_{h11})$ 3** – | Enable | Immediate first cycle |
| $\overline{SRE}$ | CP | – | – | $(t_{s12})$ 12 | $(t_{h12})$ 1 | | |
| Y | CP | – | – | $(t_{s13})$ 53 | $(t_{h13})$ 0 | | |
| Y | DLE | $(t_{s6})$ 7 | $(t_{h6})$ 3 | – | – | | |
| DLE | CP | – | – | $(t_{s14})$ 54 | $(t_{h14})$ 0 | | |

**Status register and accumulator destination only.

† Not included in Group A tests.

38

## Am29C116-2

### A. Combinational Delays (nsec)

| Input | Outputs | | |
|---|---|---|---|
| | $Y_{0-15}$ | $T_{1-4}$ | CT |
| $I_{0-4}$ (ADDR) | 65 | 65 | – |
| $I_{0-15}$ (DATA) | 65 | 65 | – |
| $I_{0-15}$ (INSTR) | 65 | 65 | 26 |
| DLE | 52*† | 52 | – |
| $T_{1-4}$ | – | – | 26 |
| CP | 57 | 65 | 33 |
| $Y_{0-15}$ | 52*† | 52 | – |
| $\overline{IEN}$ | – | – | 26 |

$Y_{0-15}$ must be stored in the Data Latch and its source disabled before the delay to $Y_{0-15}$ as an output can be measured.
*Guaranteed indirectly by other tests.

### B. Enable/Disable Times (nsec)
(Disable: $C_L$ = 5 pF, 0.5-V Change on Outputs)

| From Input | To Output | Enable | | Disable | |
|---|---|---|---|---|---|
| | | $t_{PZH}$ | $t_{PZL}$ | $t_{PHZ}$ | $t_{PLZ}$ |
| $\overline{OE}_Y$ | $Y_{0-15}$ | 22 | 22 | 18 | 18 |
| $OE_T$ | $T_{1-4}$ | 22 | 22 | 15 | 15 |

### C. Clock and Pulse Requirements (nsec)

| Input | Min. LOW Time | Min. HIGH Time |
|---|---|---|
| CP | 20 | 15 |
| DLE | – | 15 |
| $\overline{IEN}$ | 15 | – |

### D. Setup and Hold Times (nsec)

| Input | With Respect to | HIGH-to-LOW Transition Setup | HIGH-to-LOW Transition Hold | LOW-to-HIGH Transition Setup | LOW-to-HIGH Transition Hold | Comment | |
|---|---|---|---|---|---|---|---|
| $I_{0-4}$ (RAM ADDR) | CP | $(t_{s1})$ 12 | $(t_{h1})$ 1 | – | – | Single ADDR (Source) | |
| $I_{0-4}$ (RAM ADDR) | CP and $\overline{IEN}$ both LOW | $(t_{s2})$ 7 | – | – | $(t_{h7})$ 0 | Two ADDR (Destination) | |
| $I_{0-15}$ (DATA) | CP | – | – | $(t_{s8})$ 56 | $(t_{h8})$ 0 | | |
| $I_{0-15}$ (INSTR) | CP | $(t_{s3})$ 25 | $(t_{h3})$ 12 | $(t_{s9})$ 56 | $(t_{h9})$ 2 | | |
| $\overline{IEN}$ HIGH | CP | $(t_{s4})$ 5 | – | – | $(t_{h10})$ 2 | Disable | |
| $\overline{IEN}$ LOW | CP | – | $(t_{s5})$ 7 | – | $(t_{h5})$ 3 | $(t_{s11})$ 10 | – | $(t_{h11})$ 3** | – | Enable | Immediate first cycle |
| $\overline{SRE}$ | CP | – | – | $(t_{s12})$ 10 | $(t_{h12})$ 1 | | |
| Y | CP | – | – | $(t_{s13})$ 45 | $(t_{h13})$ 0 | | |
| Y | DLE | $(t_{s6})$ 7 | $(t_{h6})$ 3 | – | – | | |
| DLE | CP | – | – | $(t_{s14})$ 46 | $(t_{h14})$ 0 | | |

**Status register and accumulator destination only.

† Not included in Group A tests.

39

# Test Philosophy and Methods

The following points give the general philosophy that we apply to tests that must be properly engineered if they are to be implemented in an automatic environment. The specifics of what philosophies applied to which test are shown in the data sheet.

1. Ensure 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 output 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 hundreds of millivolts momentarily. Current level may vary from product to product.

4. Use extreme care in defining input levels for AC tests. Many inputs may be changed at once, so there will be significant noise at the device pins that may not actually reach $V_{IL}$ or $V_{IH}$ until the noise has settled. AMD recommends using $V_{IL} \leqslant 0$ V and $V_{IH} \geqslant 3.0$ V for AC tests.

5. To simplify failure analysis, programs should be designed to perform DC, Function, and AC tests as three distinct groups of tests.

6. Capacitive Loading for AC Testing

   Automatic testers and their associated hardware have stray capacitance that varies from one type of tester to another but is generally around 50 pF. This makes it impossible to make direct measurements of parameters that call for a smaller capacitive load than the associated stray capacitance. Typical examples of this are the so-called "float delays," which measure the propagation delays into the high-impedance state and are usually specified at a load capacitance of 5.0 pF. In these cases, the test is performed at the higher load capacitance (typically 50 pF), and engineering correlations based on data taken with a bench set up are used to predict the result at the lower capacitance.

Similarly, a product may be specified at more than one capacitive load. Since the typical automatic tester is not capable of switching loads in mid-test, it is impossible to make measurements at <u>both</u> capacitances even though they may both be greater than the stray capacitance. In these cases, a measurement is made at one of the two capacitances. The result at the other capacitance is predicted from engineering correlations based on data taken with a bench setup and the knowledge that certain DC measurements ($I_{OH}$, $I_{OL}$, for example) have already been taken and are within specification. In some cases, special DC tests are performed in order to facilitate this correlation.

7. Threshold Testing

   The noise associated with automatic testing (due to the long inductive cables), and the high gain of the tested device when in the vicinity of the actual device threshold, frequently give rise to oscillations when testing high-speed circuits. These oscillations are not indicative of a reject device, but instead, of an overtaxed test system. To minimize this problem, thresholds are tested at least once for <u>each</u> input pin. Thereafter, "hard" high and low levels are used for other tests. Generally this means that function and AC testing are performed at "hard" input levels rather than at $V_{IL}$ Max. and $V_{IH}$ Min.

8. AC Testing

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

   In some cases, certain AC tests are redundant since they can be shown to be predicted by other tests that have already been performed. In these cases, the redundant tests are not performed.

# 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
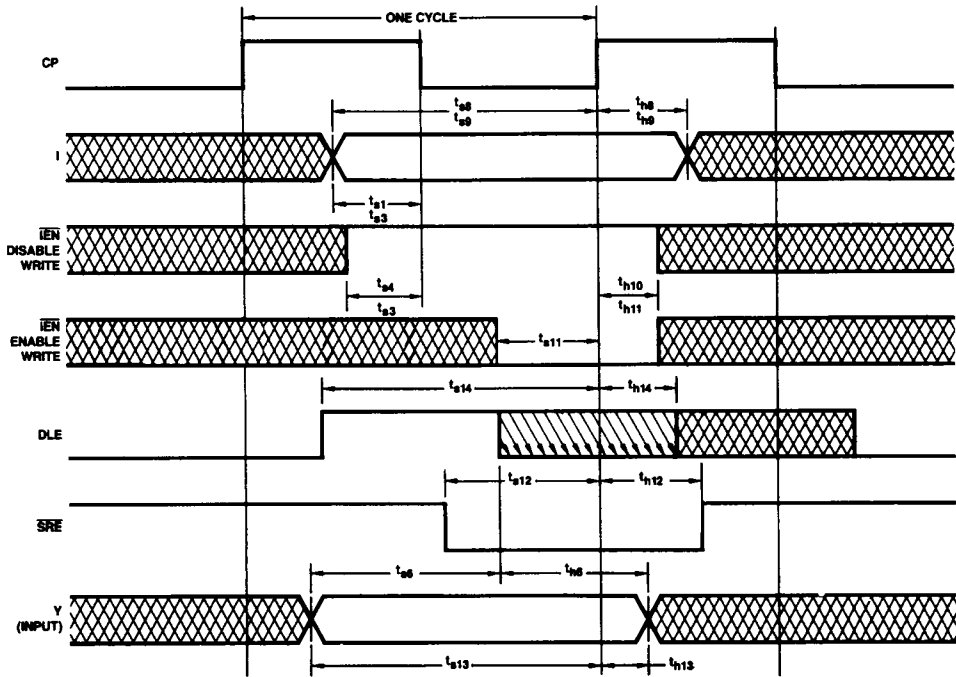


WF002563

### Single-Address Access Timing
If $t_{h6}$ is satisfied, $t_{h13}$ need not be satisfied.

41

# SWITCHING WAVEFORMS (Cont'd.)
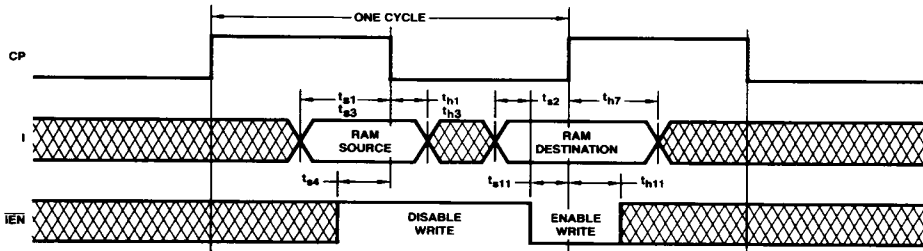


WF002545

## Double Address Access Timing
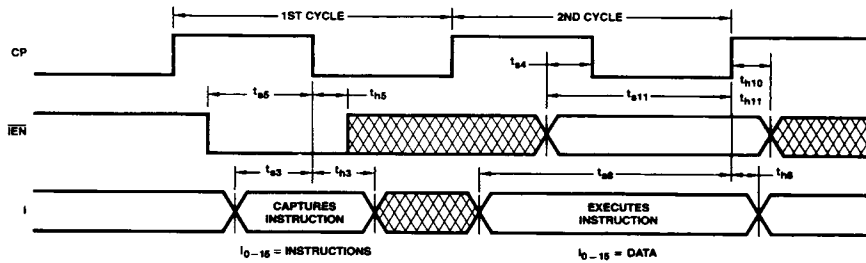


WF002550

## Immediate Instruction Cycle Timing

## SWITCHING TEST CIRCUITS
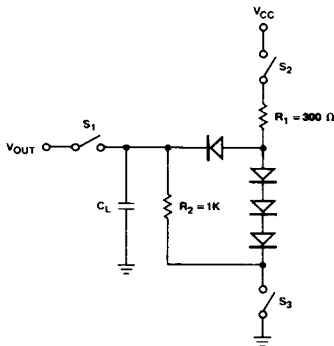


TCR01331
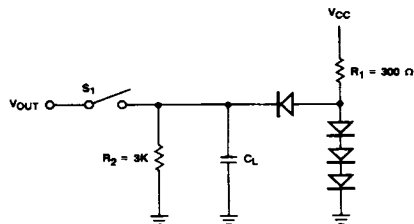
TC000421

### A. Three-State Outputs

### B. Normal Outputs

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 tests and all AC tests except output enable tests.

3. $S_1$ and $S_3$ are closed while $S_2$ is open for $t_{PZH}$ test.
   $S_1$ and $S_2$ are closed while $S_3$ is open for $t_{PZL}$ test.

4. $C_L$ = 5.0 pF for output disable tests.

42

# SWITCHING TEST WAVEFORMS
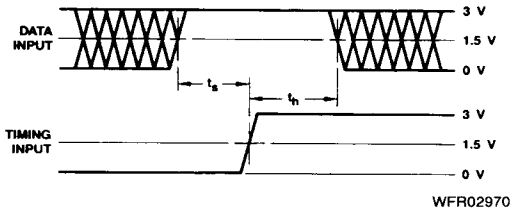
WFR02970

### Setup, Hold, and Release Times
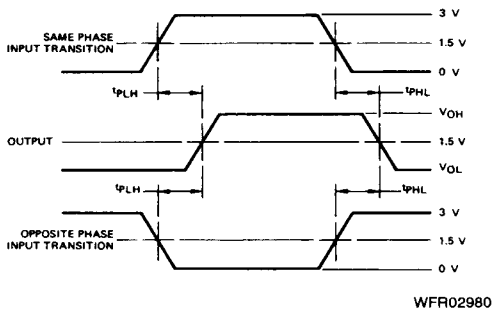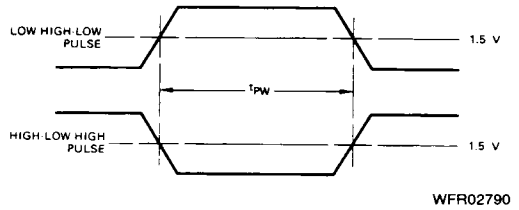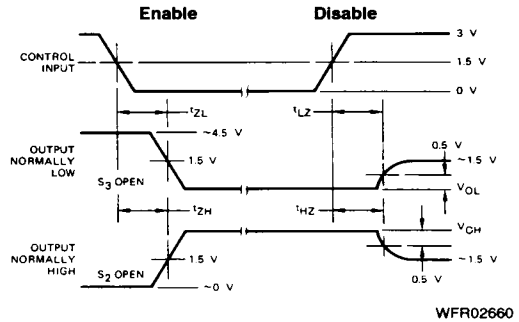
WFR02790

### Pulse Width

WFR02980

### Propagation Delay

WFR02660

### Enable and Disable Times

---

# INPUT/OUTPUT CIRCUIT DIAGRAMS

DRIVEN INPUT

TC004150

$C_I \approx 5.0$ pF, all inputs

OUTPUT

TC004160
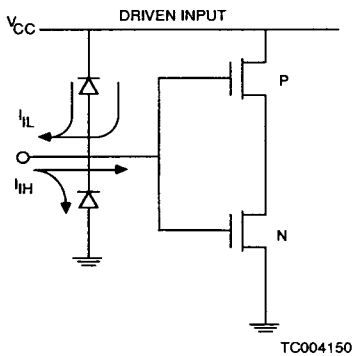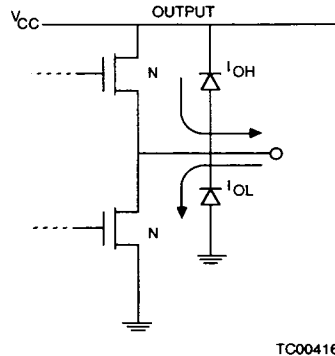
$C_O \approx 5.0$ pF, all outputs

# PHYSICAL DIMENSIONS*

## PD 052
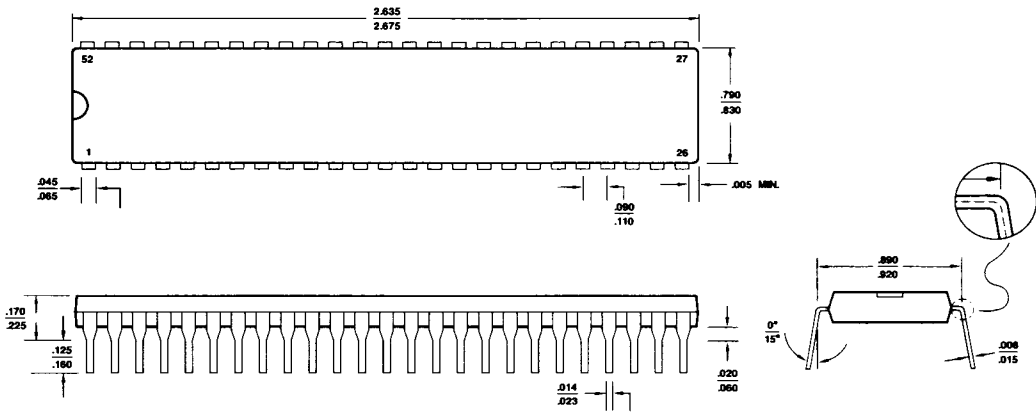


## TDX052



*For reference only.

**PL 068**
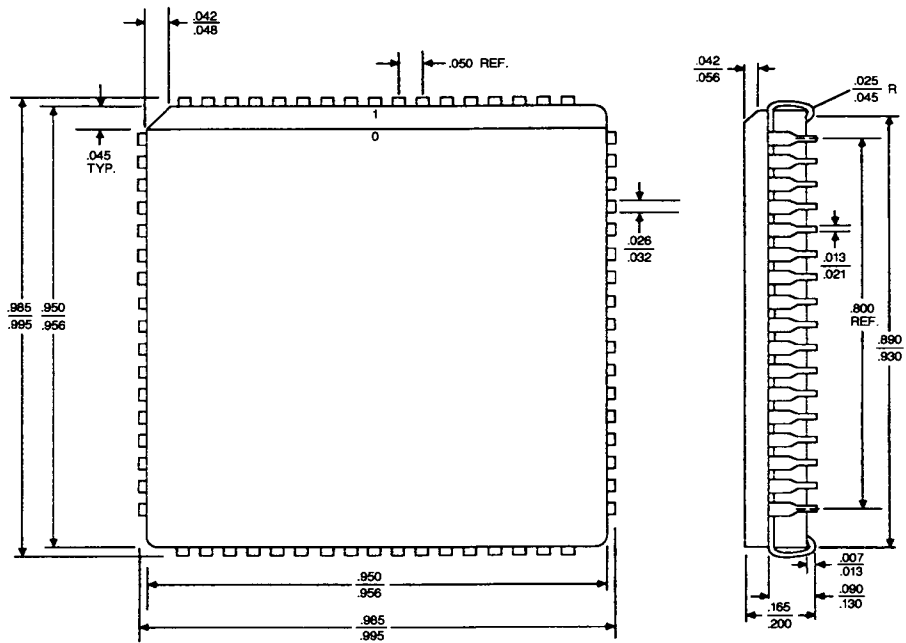
## CLT052
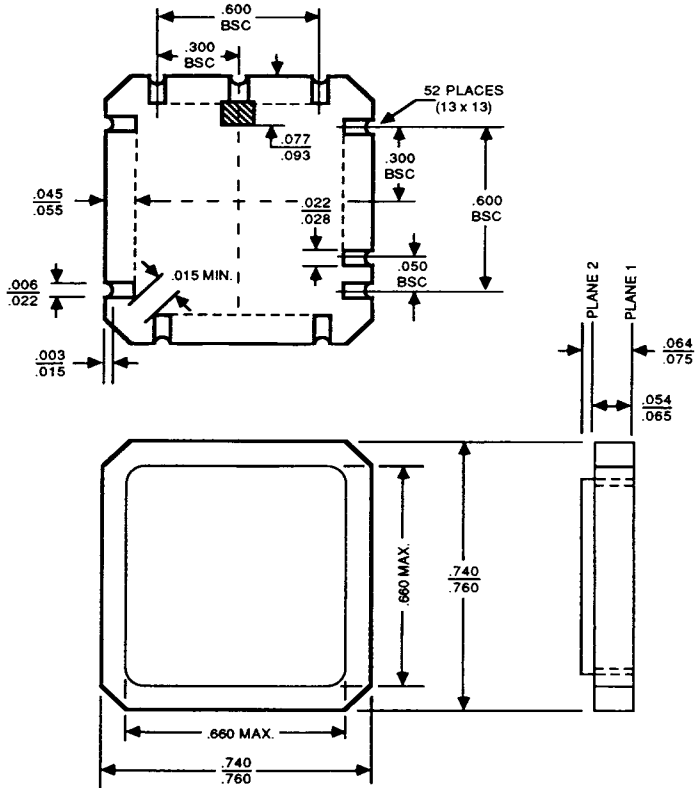
# ADVANCED MICRO DEVICES' NORTH AMERICAN SALES OFFICES

| | |
|---|---|
| ALABAMA | (205) 882-9122 |
| ARIZONA | (602) 242-4400 |
| CALIFORNIA, | |
| Culver City | (213) 645-1524 |
| Newport Beach | (714) 752-6262 |
| San Diego | (619) 560-7030 |
| San Jose | (408) 249-7766 |
| Santa Clara | (408) 727-3270 |
| Woodland Hills | (818) 992-4155 |
| CANADA, Ontario, | |
| Kanata | (613) 592-0060 |
| Willowdale | (416) 224-5193 |
| COLORADO | (303) 741-2900 |
| CONNECTICUT | (203) 264-7800 |
| FLORIDA, | |
| Clearwater | (813) 530-9971 |
| Ft Lauderdale | (305) 776-2001 |
| Melbourne | (305) 729-0496 |
| Orlando | (305) 859-0831 |
| GEORGIA | (404) 449-7920 |
| ILLINOIS, | |
| Chicago | (312) 773-4422 |
| Naperville | (312) 505-9517 |
| INDIANA | (317) 244-7207 |

| | |
|---|---|
| KANSAS | (913) 451-3115 |
| MARYLAND | (301) 796-9310 |
| MASSACHUSETTS | (617) 273-3970 |
| MINNESOTA | (612) 938-0001 |
| MISSOURI | (913) 451-3115 |
| NEW JERSEY | (201) 299-0002 |
| NEW YORK, | |
| Liverpool | (315) 457-5400 |
| Poughkeepsie | (914) 471-8180 |
| Woodbury | (516) 364-8020 |
| NORTH CAROLINA | (919) 878-8111 |
| OHIO | (614) 891-6455 |
| Columbus | (614) 891-6455 |
| Dayton | (513) 439-0470 |
| OREGON | (503) 245-0080 |
| PENNSYLVANIA, | |
| Allentown | (215) 398-8006 |
| Willow Grove | (215) 657-3101 |
| TEXAS, | |
| Austin | (512) 346-7830 |
| Dallas | (214) 934-9099 |
| Houston | (713) 785-9001 |
| WASHINGTON | (206) 455-3600 |
| WISCONSIN | (414) 792-0590 |

# ADVANCED MICRO DEVICES' INTERNATIONAL SALES OFFICES

| | | |
|---|---|---|
| BELGIUM, | | |
| Bruxelles | TEL | (02) 771 91 42 |
| | FAX | (02) 762 37 12 |
| | TLX | 61028 |
| FRANCE, | | |
| Paris | TEL | (1) 49-75-10-10 |
| | FAX | (1) 49-75-10-13 |
| | TLX | 263282 |
| WEST GERMANY, | | |
| Hannover area | TEL | (05143) 50 55 |
| | FAX | (05143) 55 53 |
| | TLX | 925287 |
| München | TEL | (089) 41 14-0 |
| | FAX | (089) 406490 |
| | TLX | 523883 |
| Stuttgart | TEL | (0711) 62 33 77 |
| | FAX | (0711) 625187 |
| | TLX | 721882 |
| HONG KONG, | | |
| Kowloon | TEL | 852-3-695377 |
| | FAX | 852-123-4276 |
| | TLX | 504260AMDAPHX |
| ITALY, Milano | TEL | (02) 3390541 |
| | | (02) 3533241 |
| | FAX | (02) 3498000 |
| | TLX | 315286 |
| JAPAN, | | |
| Kanagawa | TEL | 462-47-2911 |
| | FAX | 462-47-1729 |
| Tokyo | TEL | (03) 345-8241 |
| | FAX | (03) 342-5196 |
| | TLX | J24064AMDTKOJ |
| Osaka | TEL | 06-243-3250 |
| | FAX | 06-243-3253 |

| | | |
|---|---|---|
| KOREA, Seoul | TEL | 82-2-784-7598 |
| | FAX | 82-2-784-8014 |
| LATIN AMERICA, | | |
| Ft. Lauderdale | TEL | (305) 484-8600 |
| | FAX | (305) 485-9736 |
| | TLX | 5109554261 AMDFTL |
| NORWAY, | | |
| Hovik | TEL | (02) 537810 |
| | FAX | (02) 591959 |
| | TLX | 79079 |
| SINGAPORE | TEL | 65-2257544 |
| | FAX | 2246113 |
| | TLX | RS55650 MMI RS |
| SWEDEN, Stockholm | TEL | (08) 733 03 50 |
| | FAX | (08) 733 22 85 |
| | TLX | 11602 |
| TAIWAN | TLX | 886-2-7122066 |
| | FAX | 886-2-7122017 |
| UNITED KINGDOM, | | |
| Manchester area | TEL | (0925) 828008 |
| | FAX | (0925) 827693 |
| | TLX | 628524 |
| London area | TEL | (04862) 22121 |
| | FAX | (0483) 756196 |
| | TLX | 859103 |

# NORTH AMERICAN REPRESENTATIVES

| | |
|---|---|
| CALIFORNIA | |
| I² INC | OEM (408) 988-3400 |
| | DISTI (408) 498-6868 |
| CANADA | |
| Burnaby, B.C. | |
| DAVETEK MARKETING | (604) 430-3680 |
| Calgary, Alberta | |
| VITEL ELECTRONICS | (403) 278-5833 |
| Kanata, Ontario | |
| VITEL ELECTRONICS | (613) 592-0090 |
| Mississauga, Ontario | |
| VITAL ELECTRONICS | (416) 676-9720 |
| Quebec | |
| VITEL ELECTRONICS | (514) 636-5951 |
| IDAHO | |
| INTERMOUNTAIN TECH MKGT | (208) 888-6071 |
| INDIANA | |
| ELECTRONIC MARKETING | |
| CONSULTANTS, INC. | (317) 253-1668 |
| IOWA | |
| LORENZ SALES | (319) 377-4666 |
| KANSAS | |
| LORENZ SALES | (913) 384-6556 |

| | |
|---|---|
| KENTUCKY | |
| ELECTRONIC MARKETING | |
| CONSULTANTS, INC. | (317) 253-1668 |
| MICHIGAN | |
| SAI MARKETING CORP | (313) 750-1922 |
| MISSOURI | |
| LORENZ SALES | (314) 997-4558 |
| NEBRASKA | |
| LORENZ SALES | (402) 475-4660 |
| NEW MEXICO | |
| THORSON DESERT STATES | (505) 293-8555 |
| NEW YORK | |
| NYCOM, INC | (315) 437-8343 |
| OHIO | |
| Centerville | |
| DOLFUSS ROOT & CO | (513) 433-6776 |
| Columbus | |
| DOLFUSS ROOT & CO | (614) 885-4844 |
| Strongsville | |
| DOLFUSS ROOT & CO | (216) 238-0300 |
| PENNSYLVANIA | |
| DOLFUSS ROOT & CO | (412) 221-4420 |
| UTAH | |
| R² MARKETING | (801) 595-0631 |

---

008598 √ _ R