THE
Am29116

A 16 - BIT BIPOLAR MICROPROCESSOR

AMD CUSTOMER EDUCATION

THE Am29116

by

William J. Harmon, Jr.
Donnamaie E. White, Ph.D.
Warren K. Miller
Deepak Mithani
Sunil Joshi
Jenny Yee

Publication date: August 28, 1980

2nd Edition

This brochure contains all currently releasable information on the Am29116

901 Thompson Place, P.O. Box 453, Sunnyvale, California 94086
(408)732-2400 TWX: 910-339-9280 TELEX: 34-6306

*** .DEF file is in preparation by the Customer Education Center for use with AMDASM on the AmSYS29

HIGH SPEED MICROPROGRAMMED CONTROLLERS

HIGH SPEED CONTROLLERS

A "TYPICAL" SYSTEM IS SHOWN ON THE FOLLOWING PAGE

IT CONSISTS OF:

- A HIGH SPEED PROCESSOR

  (WITH INDICATIONS THAT IT IS A MICROPROGRAMMED CPU - A "TYPICAL" CPU AS TAUGHT IN ED2900A/B)
- AN INTELLIGENT CONTROLLER

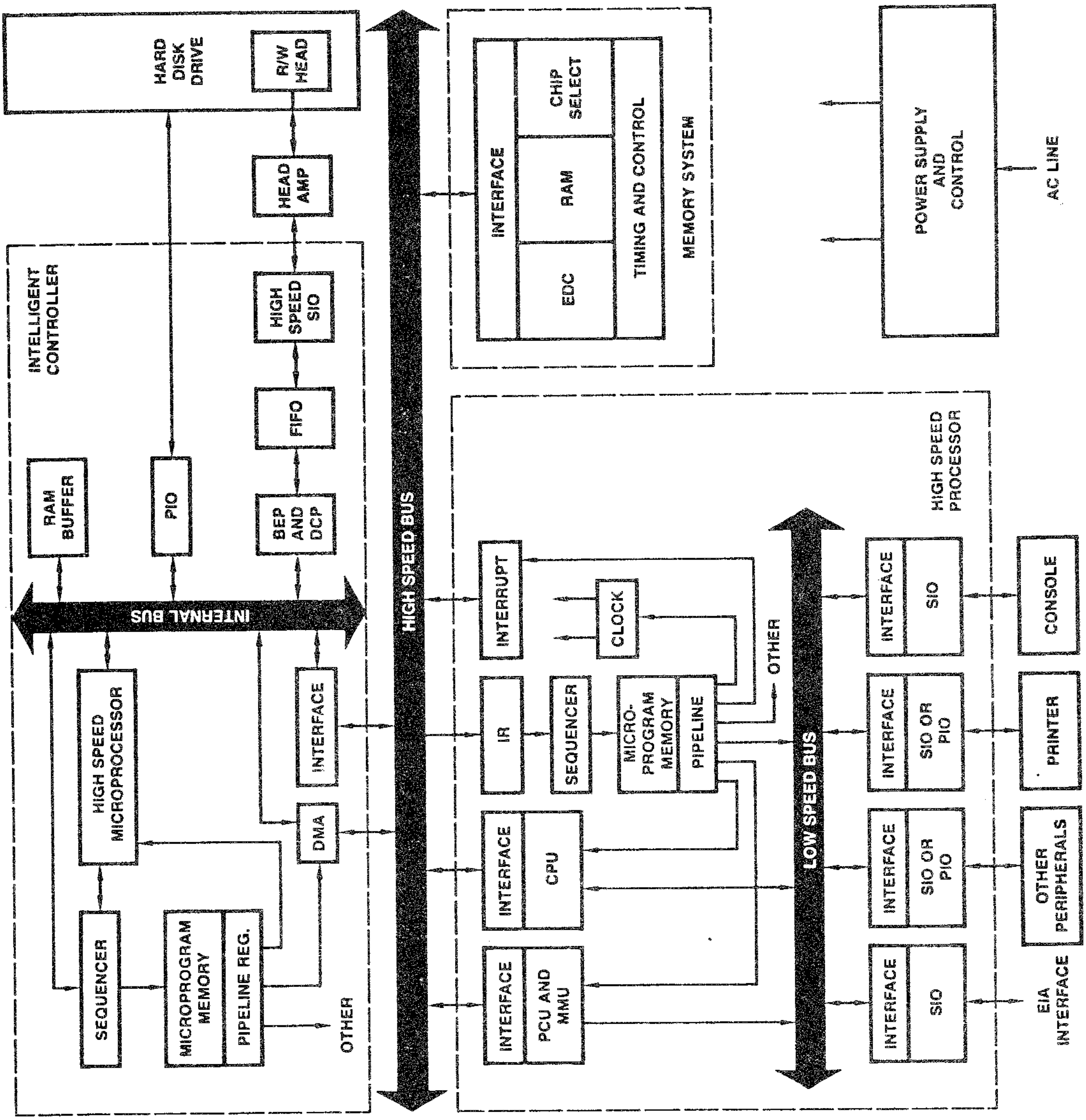
  A MICROPROGRAMMED HIGH-SPEED MICROPROCESSOR WITH THE NECESSARY INTERFACES
- A MEMORY SYSTEM

NOTE THAT THE HIGH SPEED PROCESSOR COULD BE FORMED FROM

THE Am29203 AND Am2910

OR

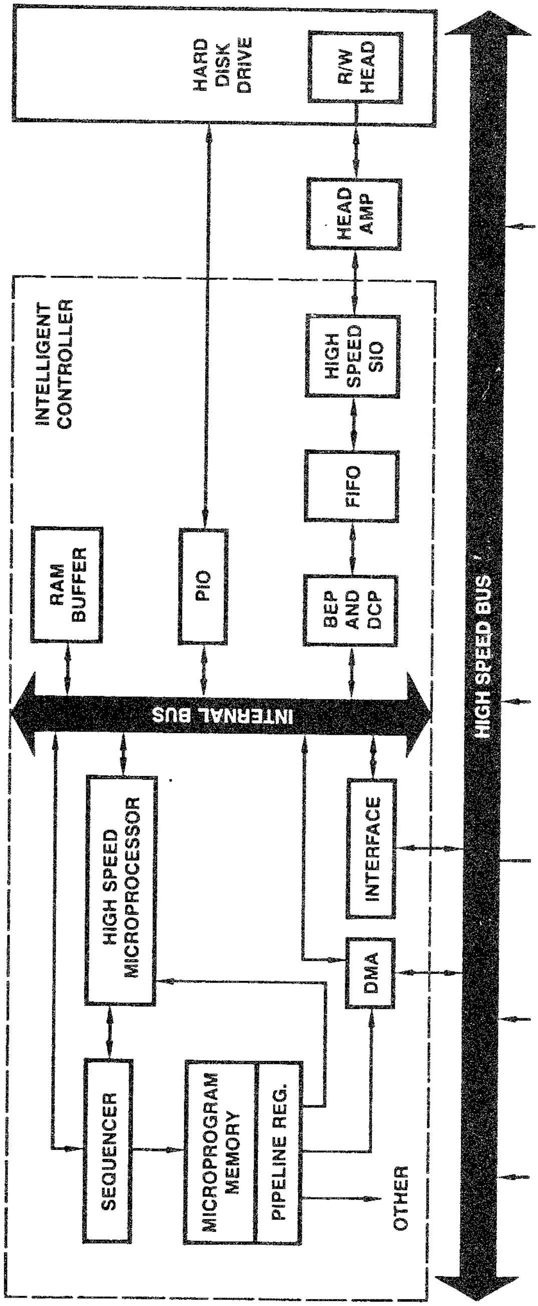
THE AmZ8002 (OR 8001)


HARD DISK DRIVE
R/W HEAD
HEAD AMP
INTELLIGENT CONTROLLER
HIGH SPEED SIO
FIFO
BEP AND DCP
PIO
RAM BUFFER
INTERNAL BUS
HIGH SPEED MICROPROCESSOR
INTERFACE
DMA
SEQUENCER
MICROPROGRAM MEMORY
PIPELINE REG.
OTHER
HIGH SPEED BUS
MEMORY SYSTEM
INTERFACE
CHIP SELECT
RAM
EDC
TIMING AND CONTROL
POWER SUPPLY AND CONTROL
AC LINE
HIGH SPEED PROCESSOR
INTERRUPT
CLOCK
IR
SEQUENCER
MICRO-PROGRAM MEMORY
PIPELINE
OTHER
INTERFACE
CPU
INTERFACE
PCU AND MMU
LOW SPEED BUS
INTERFACE
SIO
CONSOLE
INTERFACE
SIO OR PIO
PRINTER
INTERFACE
SIO OR PIO
OTHER PERIPHERALS
INTERFACE
SIO
EIA INTERFACE

WE WILL BE CONCERNED WITH THE CONSTRUCTION OF THE INTELLIGENT CONTROLLER - DETAILED ON THE FOLLOWING PAGE

THE CHOICES THAT EXIST FOR THE IMPLEMENTATION OF A CONTROLLER ARE:

MSI	BECOMING OBSOLETE
MOS MICROPROCESSOR	LOW PERFORMANCE
2909/11A or 2910; 2901	MEDIUM PERFORMANCE
2910-29116	BETTER
29112-29116	HIGH PERFORMANCE


HARD DISK DRIVE
R/W HEAD
HEAD AMP
INTELLIGENT CONTROLLER
HIGH SPEED SIO
FIFO
RAM BUFFER
PIO
BEP AND DCP
INTERNAL BUS
HIGH SPEED MICROPROCESSOR
INTERFACE
DMA
SEQUENCER
MICROPROGRAM MEMORY
PIPELINE REG.
OTHER
HIGH SPEED BUS

CONTROLLER FAMILY

29116	16-BIT BIPOLAR MICROPROCESSOR
29112	INTERRUPTABLE SEQUENCER
29150	SERIAL I/O PORT
2950/51	PARALLEL I/O PORT
2940/42	DMA ADDRESS GENERATOR
29153	FIFO
9520	BURST ERROR PROCESSOR
29XXX	ASPARAGUS

CONTROLLER IMPLEMENTATION CRITERIA

- DATA TRANSFER RATE
  - 600MB TYPICAL MAXIMUM CAPACITY
  - 6MB/SEC DATA TRANSFER RATE
- COMPACT
- ERROR DETECTION/CORRECTION
  - CRC - CYCLIC REDUNDANCY CHECK
  - BEP - BURST ERROR PROCESSING
- INTELLIGENCE
  - PROGRAMMABLE
- FILE SEARCH/SORT/MERGE
- GENERALIZED INTERFACE
- DUAL/MULTIPLE PORTING
- MULTIPLE DRIVER/CONTROLLER
- DIAGNOSTICS

THE CONSTRUCTION OF THIS CONTROLLER WILL BE DONE WITH EXISTING AND SOON-TO-EXIST 2900 FAMILY MEMBERS

THIS PUBLICATION DEALS WITH THE Am29116

CONTROLLER REQUIREMENTS

FUNCTIONAL OPERATIONS

- MOVE DATA FROM PORT TO PORT
- TEST INCOMING SIGNALS (BITS)
  - STATUS
  - COMMANDS
- GENERATE OUTGOING SIGNALS (BITS)
  - STATUS
  - COMMANDS
  - TIMING SIGNALS

THE DESIGN GOALS OF THE Am29116 ARE DETAILED ON THE FOLLOWING PAGE

THE OBJECTIVE OF 100ns IS VERY NEARLY MET

THE RECOMMENDED CYCLE IS 120ns (INCLUDES MSI)

BASED ON SIMULATION

EXTENDED CYCLE LENGTH PERMITS THREE ADDRESS INSTRUCTIONS

WHICH IN THIS CASE MEANS ALTERING THE RAM ADDRESS

ALL OTHER STATED GOALS ARE MET

DESIGN GOALS:

MICROPROGRAMMED MACHINES
16 BITS
100ns MICROCYCLE
52-PIN DIP
+5V ONLY
TTL COMPATIBLE I/O

Am29116 OVERVIEW

Am29116

OUTSTANDING FEATURES

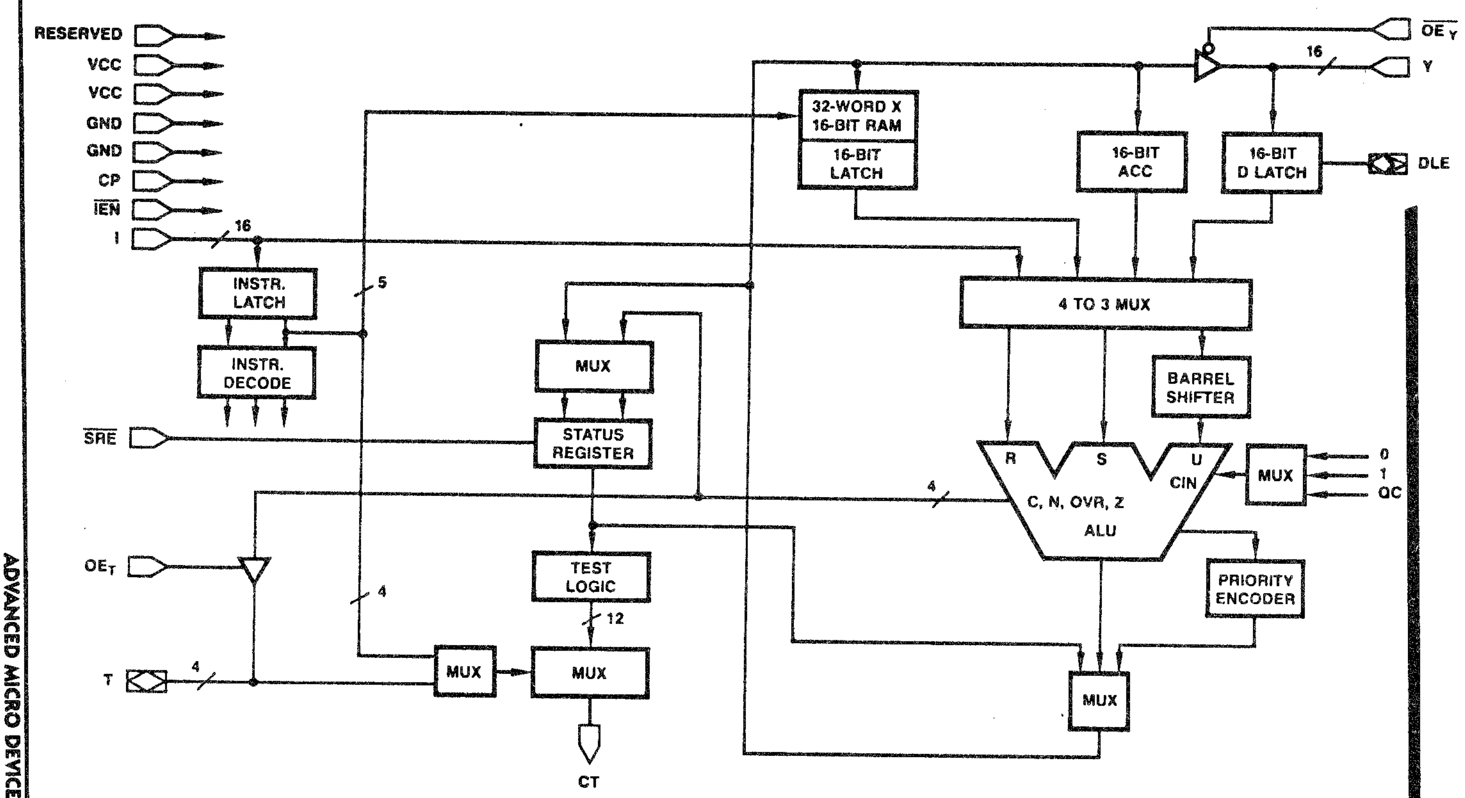
- 16-BIT DATA PATH
    - 16-BIT ALU
    - FULL CARRY LOOK-AHEAD
    - CAN OPERATE IN 16-BIT WORD MODE
    - CAN OPERATE IN 8-BIT BYTE MODE
- 32x16-BIT RAM SCRATCHPAD ON-BOARD
    - SINGLE PORT
    - WITH EXTERNAL MULTIPLEXER ADDED MAY SELECT DIFFERENT SOURCE AND DESTINATION ADDRESS FOR SAME INSTRUCTION (REQUIRES TIMING ADJUST)
- 16-BIT ACC
- 16-BIT DATA LATCH
- 16-BIT BARREL SHIFTER
    - BYTE OR WORD MODE
    - ROTATES 1 TO 15 BITS UP IN ONE CYCLE

- 8-BIT STATUS REGISTER
- CONDITION CODE GENERATOR/MULTIPLEXER
  - 12 DIFFERENT TEST CONDITIONS
- IMMEDIATE INSTRUCTION CAPABILITY
  - FIRST MICROCYCLE - INSTRUCTION LATCHED
  - SECOND MICROCYCLE - IMMEDIATE DATA AVAILABLE
  - BOTH ON I_i LINES
- CRC GENERATION
  - ANY CRC POLYNOMIAL OF 16-BITS OR LESS
  - 80% OF CRC APPLICATIONS ARE 16-BIT POLYNOMIALS
- POWERFUL INSTRUCTION SET
  - NOT YET RELEASED
- NOT EXPANDABLE

THE FOLLOWING PAGE GIVES A BLOCK DIAGRAM OF THE Am29116

- NOTE THAT THE THREE INPUTS TO THE OPERAND MUXS ARE RESTRICTED AS TO WHICH CAN GO TO WHICH ALU PORT

BLOCK DIAGRAM



32x16 RAM

- SINGLE PORT
- 16-BIT LATCH AT OUTPUT
    - TRANSPARENT WHEN CLOCK = HIGH
    - LATCHED WHEN CLOCK = LOW
- DATA WRITTEN
    - WHEN CLOCK LOW
        - IF $\overline{\text{IEN}}$ = LOW
        - IF RAM IS INSTRUCTION DESTINATION
- BYTE INSTRUCTIONS
    - USE LOWER 8 BITS OF RAM WORDS
- WORD INSTRUCTIONS
    - ALL 16 BITS WRITTEN
- WITH EXTENDED TIMING, EXTERNAL MUX
    - CAN CHANGE RAM ADDRESS DURING THE INSTRUCTION

ACC

- 16-BIT EDGE-TRIGGERED REGISTER
- ACC ACCEPTS DATA ON RISING EDGE OF CLOCK
  - IF $\overline{\text{IEN}}$ = LOW
  - IF ACC IS INSTRUCTION DESTINATION

DATA LATCH

- 16-BITS
- HOLDS DATA INPUT FROM BIDIRECTIONAL Y-BUS
- TRANSPARENT WHEN DLE INPUT = HIGH
- LATCHED WHEN DLE INPUT = LOW
  - LATCHES ALL 16 BITS AT ONCE
  - CANNOT LATCH BYTE → ONLY WORD

BARREL SHIFTER

- ONE ALU INPUT
- ROTATE DATA FROM
  - RAM
  - ACC
  - DATA LATCH (D)
- 1 TO 15 BIT ROTATE IN ONE MICROCYCLE (ONE MICROINSTRUCTION) [WORD MODE]
- 1 TO 7 BIT ROTATE IN ONE MICROCYCLE (ONE MICROINSTRUCTION) [BYTE MODE]
  - LOWER BYTE ONLY IS INVOLVED

ALU

- HIGH-SPEED ALU
- 16-BITS WIDE
- ONE, TWO, OR THREE OPERAND INSTRUCTIONS
- DOES ALL THE USUAL ONE AND TWO OPERAND FUNCTIONS
    - PASS
    - SUB
    - OR
    - NOR
    - ADD
    - AND
    - EXOR
    - EX-NOR
    - NAND
    - COMPLEMENT
    - TWO'S COMPLEMENT
- PLUS - IT HAS THREE OPERAND INSTRUCTIONS
    - ROTATE AND MERGE
    - ROTATE AND COMPARE WITH MASK

ALU - con't

- THREE STATUS OUTPUTS

  C N OVR

- Z STATUS FROM ZERO DETECT LOGIC

- CARRY-IN MULTIPLEXER
  - SELECT ZERO, ONE, QC

NOTE: Qi IS USED TO REFER TO STATUS REGISTER BITS

PRIORITY ENCODER

- PRODUCES BINARY WEIGHTED CODE TO INDICATE THE LOCATION OF THE HIGHEST ORDER ONE AT ITS INPUT

- INPUT FROM ALU
  - OPERAND .AND. $\overline{\text{MASK}}$

(MORE DETAILS LATER IN LECTURE)

STATUS REGISTER

- 8-BIT STATUS WORD

  $Q_i \quad i = 0,\ldots,7$

F	F	F	L	OV	N	C	Z

- $(\overline{SRE} \wedge \overline{IEN} = 0) \iff$ UPDATE ALU BITS OF STATUS REGISTER
  - EXCEPT FOR
    - NOP
    - SAVE STATUS
    - TEST STATUS

- UPPER 4 BITS CHANGED ONLY FOR
    - STATUS SET-RESET
    - STATUS LOAD - WORD MODE

- LINK STATUS UPDATED AFTER EACH SHIFT
- LOAD FROM INTERNAL Y-BUS
- LOAD TO INTERNAL Y-BUS

STATUS WORD

MSB 7	6	5	4	3	2	1	LSB 0
FLAG 3	FLAG 2	FLAG 1	LINK	OVR	N	C	Z

STATUS INSTRUCTIONS

SET/RESET	LOAD	SAVE
WORD	WORD	WORD
Z, C, N, OVR	Z, C, N, OVR	
LINK		
FLAG 1		
FLAG 2		
FLAG 3		

- STATUS REGISTER MAY BE SOURCE
  - LOAD WORD

0000 0000	SSSS SSSS	ZERO FILL HIGH BYTE

16 BIT [illegible]

  - LOAD BYTE

UNCHANGED	SSSS SSSS	UNALTERED HIGH BYTE

[illegible]

* [illegible]

- LOWER 4 STATUS BITS (ALU BITS) MAY BE READ TO T BUS
  - AVAILABLE WHEN OE_T = HIGH
  - NOTE ACTIVE HIGH ENABLE!

TRI-STATE BUFFERS

- BIDIRECTIONAL 16-BIT Y BUS
  - ENABLE WITH $\overline{OE}_Y$ = LOW
- BIDIRECTIONAL 4-BIT T BUS
  - ENABLE WITH OE_T = HIGH

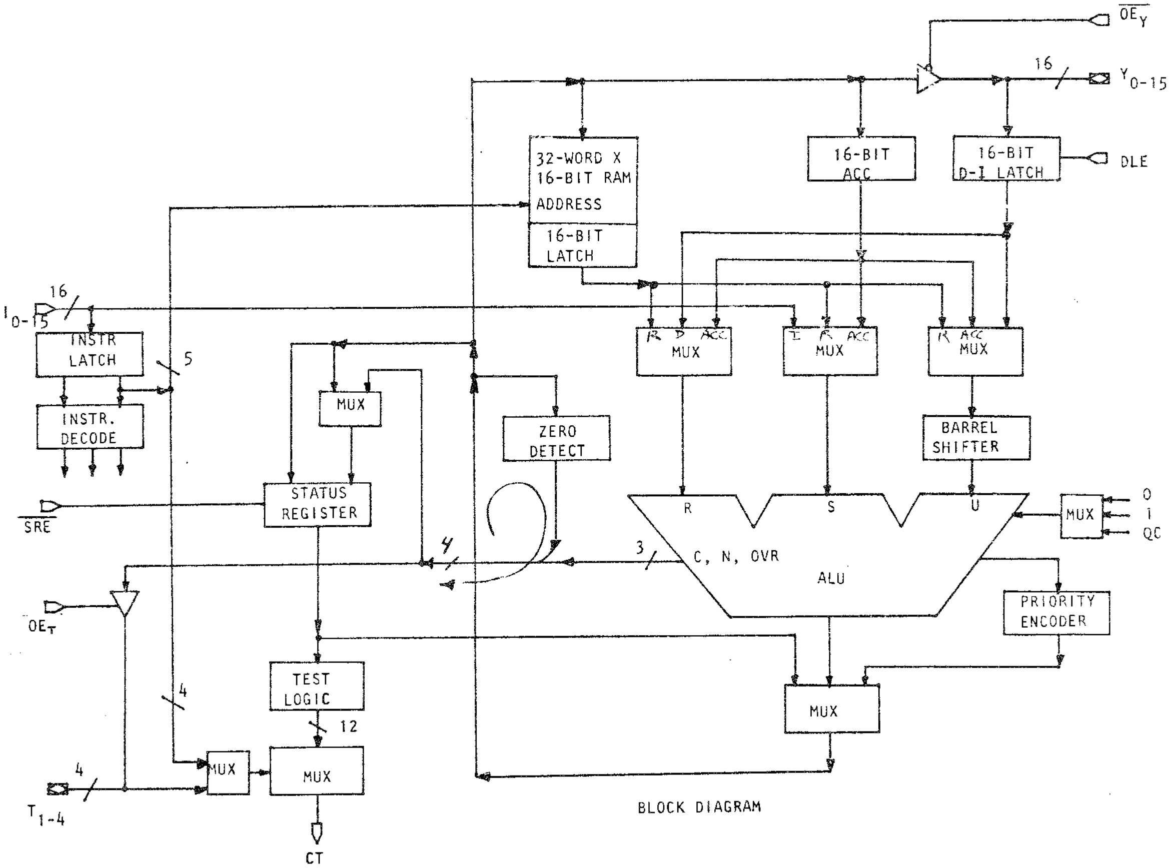
    ALU STATUS OUTPUT
  - DISABLE WITH OE_T = LOW

    T_i SELECTS TEST CONDITION

INSTRUCTION LATCH AND DECODER

- NORMALLY TRANSPARENT
  - FOR ALL INSTRUCTIONS EXCEPT IMMEDIATE

    INSTRUCTIONS EXECUTED IN <u>ONE</u> CLOCK CYCLE
- ON RECEIVING IMMEDIATE INSTRUCTION
  - LATCH INSTRUCTION FROM LINES I_i
  - NEXT CYCLE - I_i USED AS DATA



BLOCK DIAGRAM

Am29116 INSTRUCTIONS

OVERVIEW

Am 29116 INSTRUCTIONS

- SINGLE OPERAND
- TWO OPERAND
- SHIFT
- ROTATE
- BIT ORIENTED
- PRIORITIZE
- ROTATE & MERGE
- ROTATE & COMPARE
- CRC
- STATUS

ALU SOURCES

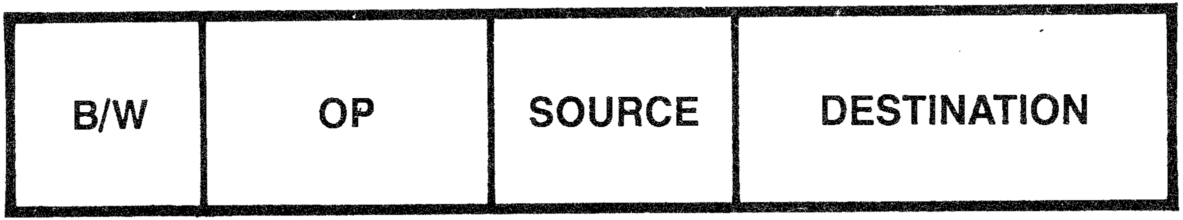
RAM
ACC
D LATCH
INSTRUCTION

ALU DESTINATIONS

(ALWAYS AT Y)

RAM
ACC
NONE

SINGLE OPERAND INSTRUCTIONS



B/W	OP	SOURCE	DESTINATION

OP

PASS
COMPLEMENT
INCREMENT
TWO'S COMPLEMENT

SINGLE OPERAND INSTRUCTIONS

	CHOOSE ONE	CHOOSE ONE
FUNCTION	**SOURCE (R)**	**DESTINATION**
$R \rightarrow$ DEST.	RAM	RAM
$\overline{R} \rightarrow$ DEST.	ACC	ACC
$R + 1 \rightarrow$ DEST.	D	NONE
$\overline{R} + 1 \rightarrow$ DEST.	D (OE)	(ALWAYS AT Y IF $\overline{OE}_Y$ ACTIVE)
	D (SE)	
	I	
	0	

D_{SE} - D WITH SIGN EXTEND - FOR TWO'S COMPLEMENT ARITHMETIC

$\overline{R}$ - DEST FOR ONE'S COMPLEMENT ARITHMETIC

$\overline{R}$ + 1 - DEST FOR TWO'S COMPLEMENT ARITHMETIC

TWO OPERAND INSTRUCTIONS

B/W	OP	OPERAND ← OPERAND (R) (S)	DESTINATION

OP

ARITHMETIC	LOGICAL
ADD	AND
ADD WITH CARRY	NAND
SUB (R-S, S-R)	OR
SUB WITH CARRY	NOR
	EX-OR
	EX-NOR

TWO OPERAND INSTRUCTIONS

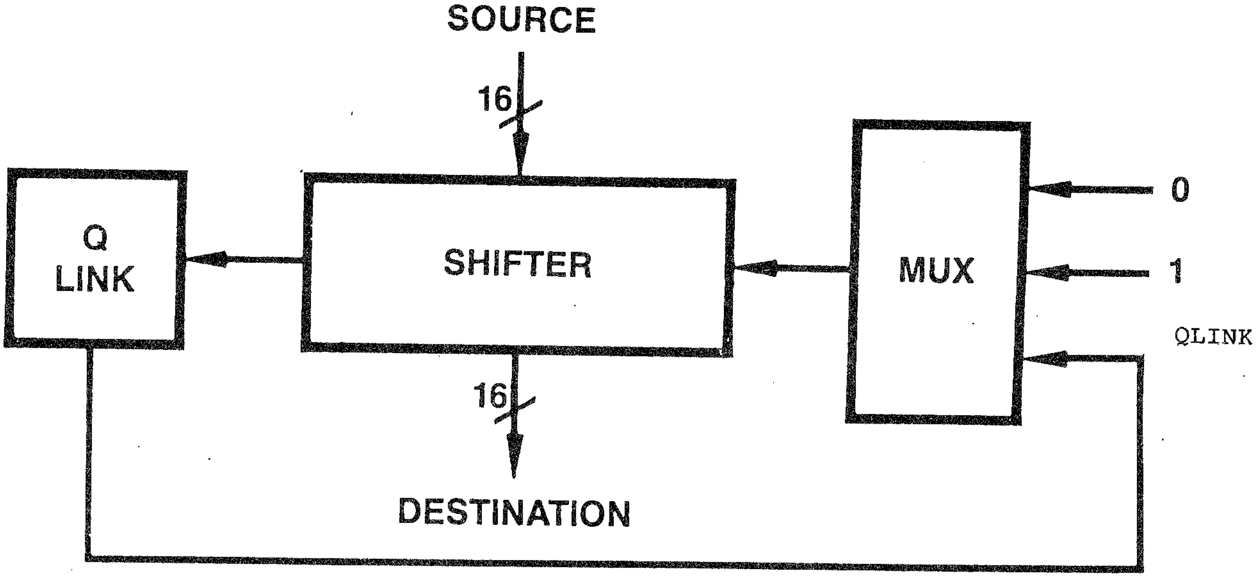
CHOOSE PAIR

FUNCTION	OPERANDS		DESTINATION
	R	S	
S MINUS R	RAM	ACC	RAM
S MINUS R WITH CARRY	RAM	I	ACC
R MINUS S	D	RAM	NONE
R MINUS S WITH CARRY	D	ACC	
R PLUS S	ACC	I	
R PLUS S WITH CARRY	D	I	
R AND S			
R NAND S			
R OR S			
R NOR S			
R EX-OR S			
R EX-NOR S			

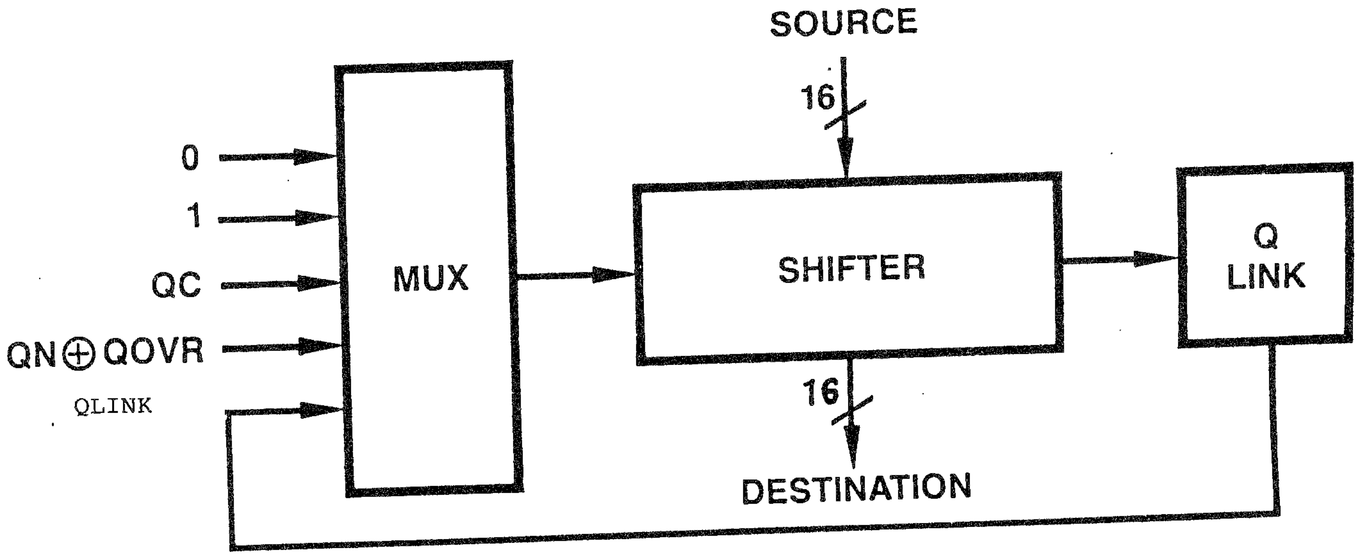
SHIFT INSTRUCTIONS

- BYTE OR WORD MODE
- DIRECTION AND SHIFT LINKAGE
    - UP OR DOWN
    - SPECIFY VACANT BIT FILLER
- SOURCE
- DESTINATION

SHIFT UP FUNCTION



SHIFT DOWN FUNCTION



SINGLE BIT SHIFT INSTRUCTIONS

CHOOSE ONE PAIR

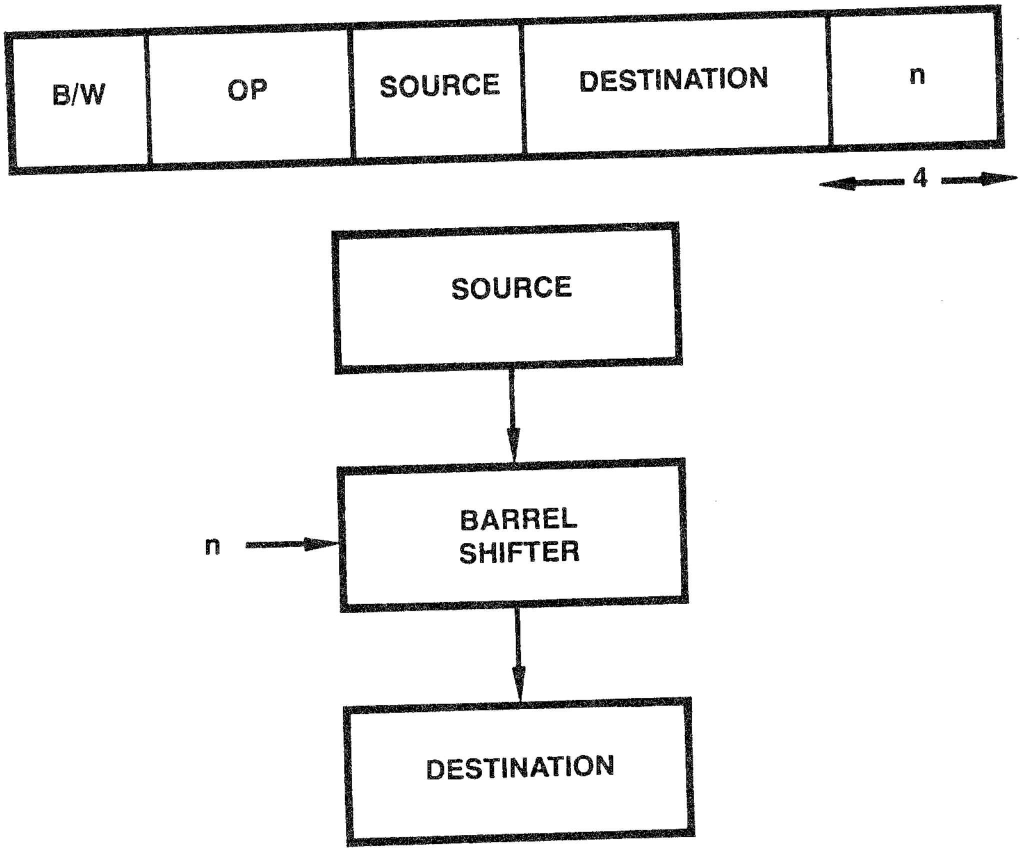
DIRECTION	FILLER INPUT	SOURCE	DESTINATION
UP	0	RAM	RAM
UP	1	ACC	ACC
UP	QLINK	ACC	NONE
DOWN	0	D	RAM
DOWN	1	D	ACC
DOWN	QLINK	D	NONE
DOWN	QC		
DOWN	QN $\oplus$ QOVR		

REMEMBER THAT THE DESTINATION IS ALWAYS AT Y IF $\overline{OE}_Y$ IS ACTIVE

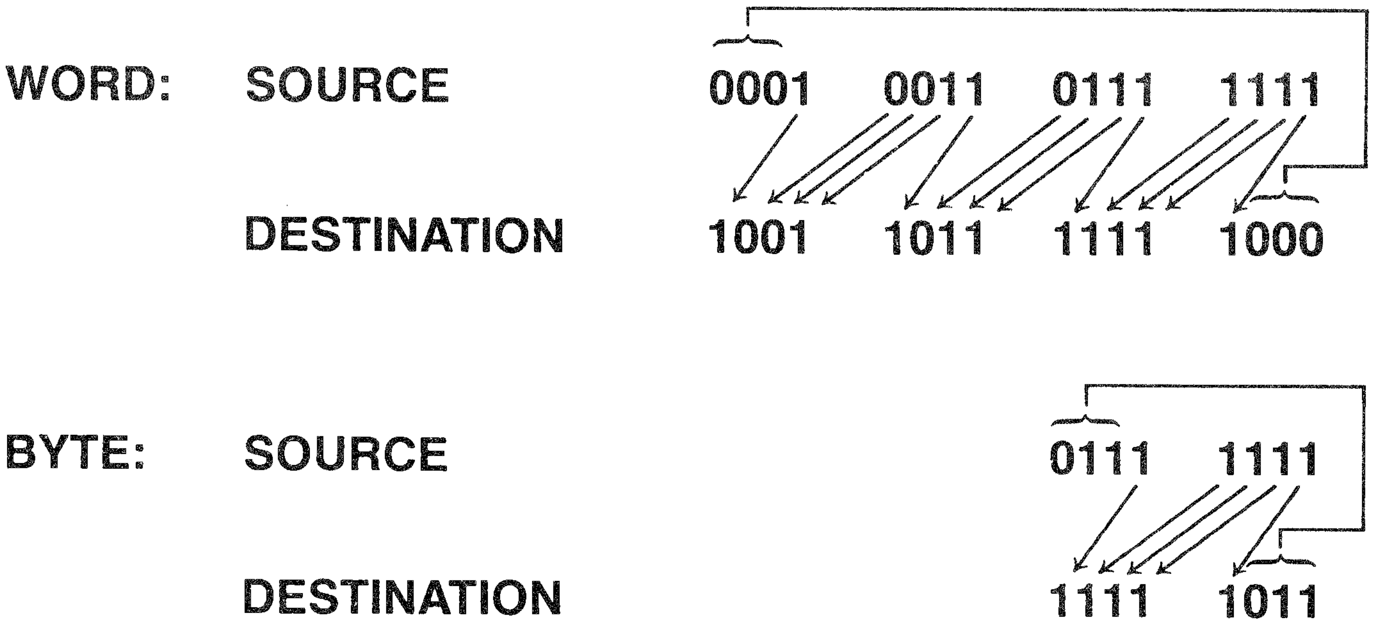
ROTATE INSTRUCTIONS

- BYTE OR WORD MODE
- n IS BIT POSITIONS
- $0 \leq n \leq 15$ WORD MODE
- $0 \leq n \leq 7$ BYTE MODE
- $<R_3>$ ROTATE TO $<R_3>$
- $<R_3>$ ROTATE TO $<R_7>$

ROTATE INSTRUCTIONS



ROTATE EXAMPLE: n = 3



WORD: SOURCE 0001 0011 0111 1111

DESTINATION 1001 1011 1111 1000

BYTE: SOURCE 0111 1111

DESTINATION 1111 1011

ROTATE INSTRUCTIONS

SOURCE (U)	DESTINATION
RAM	RAM
ACC	ACC
D	NONE

BIT INSTRUCTIONS

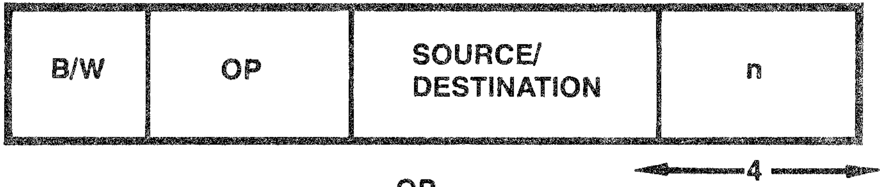
- BYTE OR WORD MODE
- ONE OPERAND IS SOURCE AND DESTINATION
- n = ADDRESS OF BIT
- OPERATIONS:

SET BIT N	Nth BIT <-- 1
RESET BIT N	Nth BIT <-- 0
TEST BIT N	SET ZERO STATUS FROM BIT N ONLY
LOAD 2^N	1 --> BIT N; 0 --> ALL OTHER BITS
LOAD $\bar{2}^N$	0 --> BIT N; 1 --> ALL OTHER BITS
INCR BY 2^N	ADD 2^N TO OPERAND
DECR BY 2^N	SUBTRACTS 2^N FROM OPERAND

```
15 14 13 12 10 9 8 7 6 5 4 3 2 1 0
          ^
N = 12 ---|
```

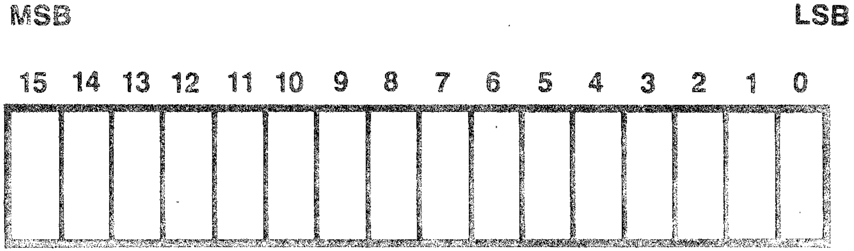
2^{12} = 0001 0000 0000 0000

BIT ORIENTED INSTRUCTIONS



OP

SET BIT n
RESET BIT n
TEST BIT n
INCREMENT BY 2^n
DECREMENT BY 2^n
LOAD 2^n
LOAD $\overline{2^n}$



BIT ORIENTED INSTRUCTIONS

CHOOSE ONE PAIR

FUNCTION	SOURCE	DESTINATION
SET BITn	RAM	RAM
RESET BITn	ACC	ACC
TEST BITn	D	NONE
$2^n \rightarrow$ DEST.		
$\overline{2^n} \rightarrow$ DEST.		
SOURCE + $2^n \rightarrow$ DEST.		
SOURCE – $2^n \rightarrow$ DEST.		

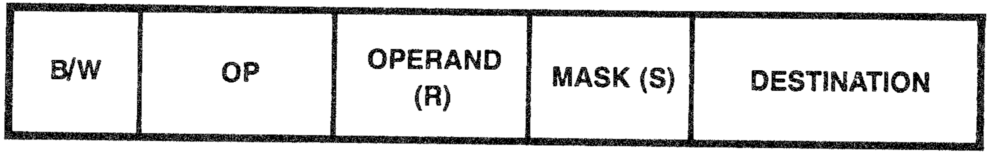
PRIORITIZE INSTRUCTIONS

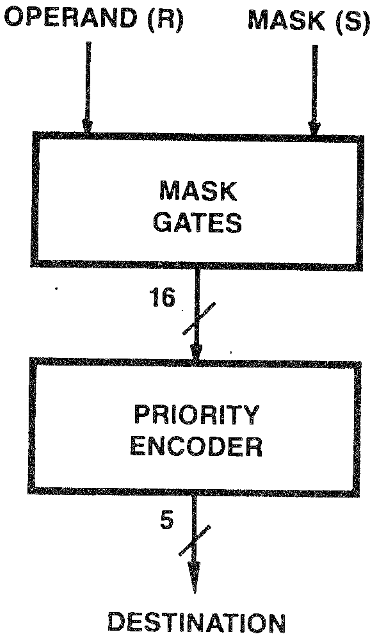
- OPERAND TO R-PORT
- MASK TO S-PORT
- OPERATION IS

  OPERAND .AND. $\overline{\text{MASK}}$

  - MASK BIT = 0 PASSES OPERAND BIT
  - MASK BIT = 1 BLOCKS OPERAND BIT

PRIORITIZE INSTRUCTIONS





PRIORTIZE INSTRUCTION

WORD MODE

HIGHEST PRIORITY ACTIVE BIT	ENCODER OUTPUT
NONE	0
15	1
14	2
•	•
•	•
•	•
1	15
0	16

EXAMPLE:

OPERAND	0001	0010	0010	1010
MASK	1111	0000	1111	0000
HIGHEST PRIORITY		↑		
DESTINATION		0007 (HEX)		

WORD EXAMPLE

MSB LSB

OPERAND: 0001 0010 0010 1010

MASK: 1111 0000 1111 0000

RESULT: 0000 0010 0000 1010

↑

HIGHEST BIT = BIT 9

ENCODED PRIORITY = 16 - 9 = 7

DESTINATION: 0 0 0 7 (HEX)

PRIORITIZE INSTRUCTION

BYTE MODE*

HIGHEST PRIORITY ACTIVE BIT	ENCODER OUTPUT
NONE	0
7	1
6	2
• • •	• • •
1	7
0	8

*BITS 8-15 DO NOT PARTICIPATE

EXAMPLE:	OPERAND	0001	0010	0010	1010
	MASK	1111	0000	1111	0000
	HIGHEST PRIORITY				↑
	DESTINATION			0005 (HEX)	

BYTE EXAMPLE

	MSB			LSB
OPERAND:	0001	0010	0010	1010
MASK:	1111	0000	1111	0000
RESULT:	NOT INVOLVED		0000	1010
				↑

HIGHEST BIT = BIT 3

ENCODED PRIORITY = 8 - 3 = 5

DESTINATION:	0	0	0	5 (HEX)

PRIORITIZE INSTRUCTIONS

OPERAND (R)	MASK (S)	DESTINATION
RAM	RAM	RAM
ACC	ACC	ACC
D	I	NONE
	NONE	

NOTE: OPERAND AND MASK MUST BE DIFFERENT SOURCES.

ROTATE AND MERGE INSTRUCTIONS

- UNROTATED OPERAND R
- ROTATE OPERAND U
- USE MASK M TO SELECT

DESTINATION $D_i = U_i$ IF $M_i = 1$

$D_i = R_i$ IF $M_i = 0$

EXAMPLES

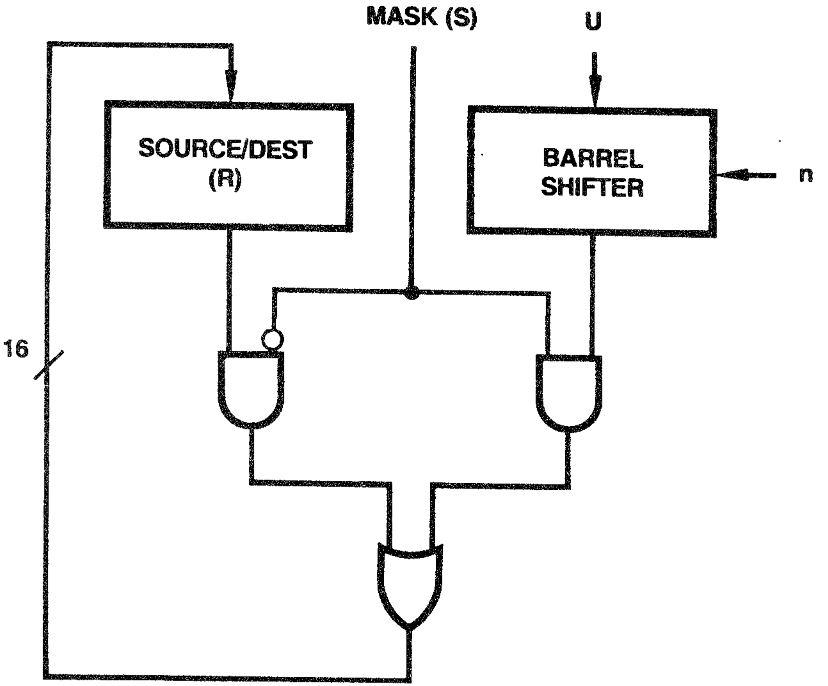
N = 4 WORD MODE

```
U:            0011  0001  0101  0110

ROTATED U:    0001  0101  0110  0011

R:            1010  1010  1010  1010

MASK S:       0000  1111  0000  1111 = RRRR UUUU RRRR UUUU

DEST:         1010  0101  1010  0011


MASK S:       0101  1010  0110  1001 = RURU URUR RUUR URRU

DEST:         1011  0000  1110  0011
```

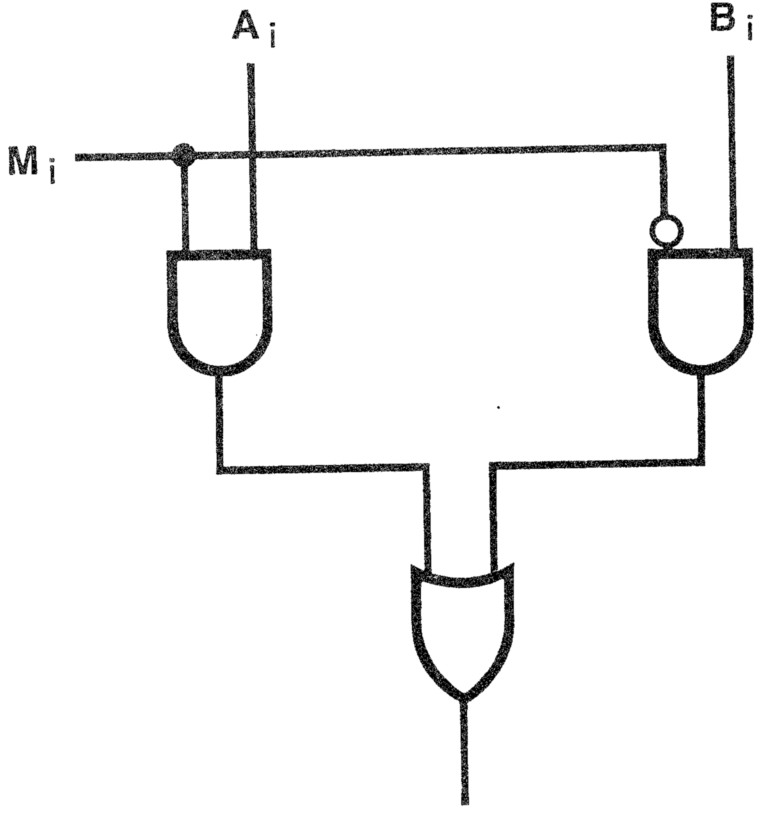
ROTATE AND MERGE INSTRUCTIONS

CHOOSE ONE TRIPLE

B/W	OP	NON-ROTATED OPERAND/DEST (R) – MASK (S) – ROTATED OPERAND (U)	n
			← 4 →



MERGE DEFINITION



$M_i = 1$ SELECT A

$M_i = 0$ SELECT B

$Y_i = M_i A_i + \overline{M}_i B_i$

ROTATE AND MERGE INSTRUCTIONS

TRIPLES

NON-ROTATED SOURCE AND DEST. (R)	ROTATED SOURCE (U)	MASK (S)
ACC	D	I
ACC	D	RAM
RAM	D	I
RAM	D	ACC
RAM	ACC	I
ACC	RAM	I

ROTATE AND COMPARE INSTRUCTIONS

- UNROTATED OPERAND R
- ROTATED OPERAND U
- .AND. U WITH $\overline{\text{MASK}}$ S AND R INPUT

$(U\ \bar{S}\ R = 0)$ <==> $(U = R)$

$(U\ \bar{S}\ R = 1)$ <==> $(U \neq R)$

EXAMPLES

n = 4 WORD MODE

U:	0011	0001	0101	0110
ROTATED U:	0001	0101	0110	0011
R:	0001	0101	1111	0000
MASK S:	0000	0000	1111	1111
RESULT:	0001	0101	0000	0000

Z STATUS = 1

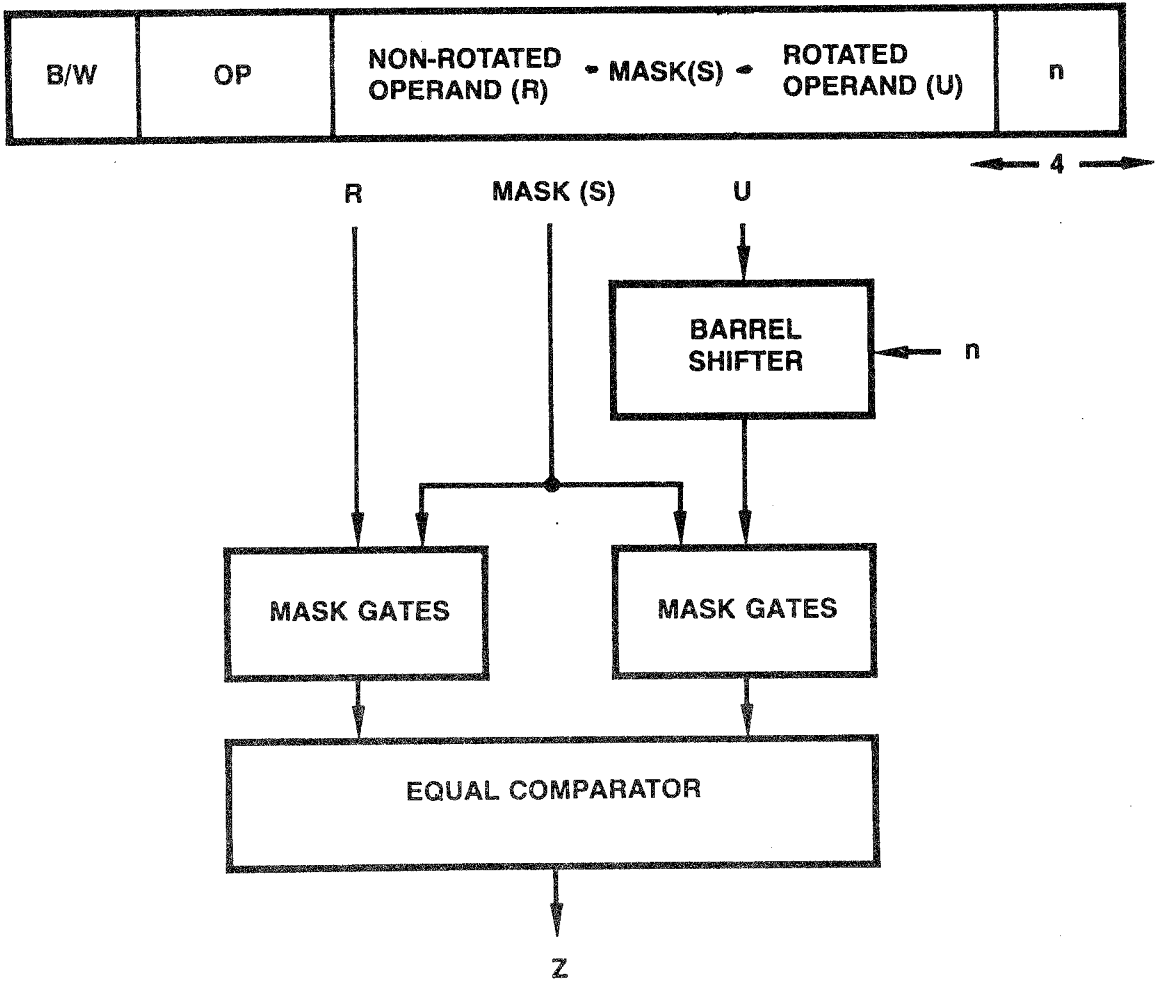
MASK S:	1111	1111	0000	0000
RESULT:	0000	0000	0110	0000

Z STATUS = 1

MASK S:	1111	1111	0110	0000
RESULT:	0000	0000	0000	0000

Z STATUS = 0

ROTATE AND COMPARE INSTRUCTIONS



ROTATE AND COMPARE INSTRUCTIONS

CHOOSE ONE TRIPLE

NON-ROTATED SOURCE (R)	ROTATED SOURCE (U)	MASK (S)
ACC	D	I
RAM	D	I
RAM	D	ACC
ACC	RAM	I

CRC INSTRUCTIONS

CRC THEORY WILL NOT BE TAUGHT IN THIS CLASS
[TAUGHT IN ED29116]

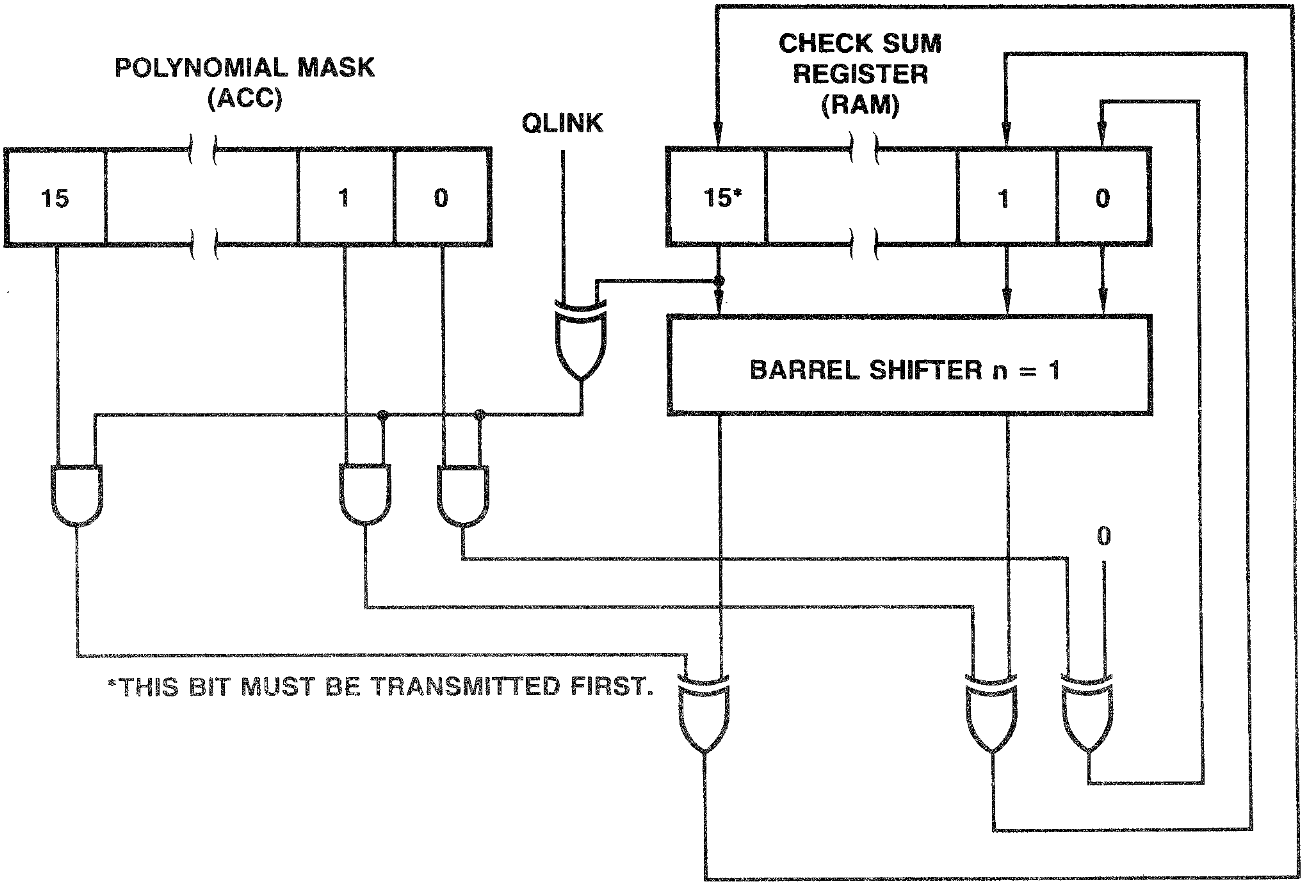
- CRC POLYNOMIALS ARE MERELY POLYNOMIAL VERSIONS OF HAMMING CODE

- Am29116 GENERATES CHECK BITS FOR CRC
  - ANY POLYNOMIAL OF 16 BITS OR LESS
    (80 TO 95% OF CRC CALCULATIONS USE 16 BIT POLYNOMIALS)

- TWO FUNCTIONS AVAILABLE
  - CRC FORWARD - CHECKSUM BIT 15 FIRST
  - CRC REVERSE - CHECKSUM BIT 0 FIRST

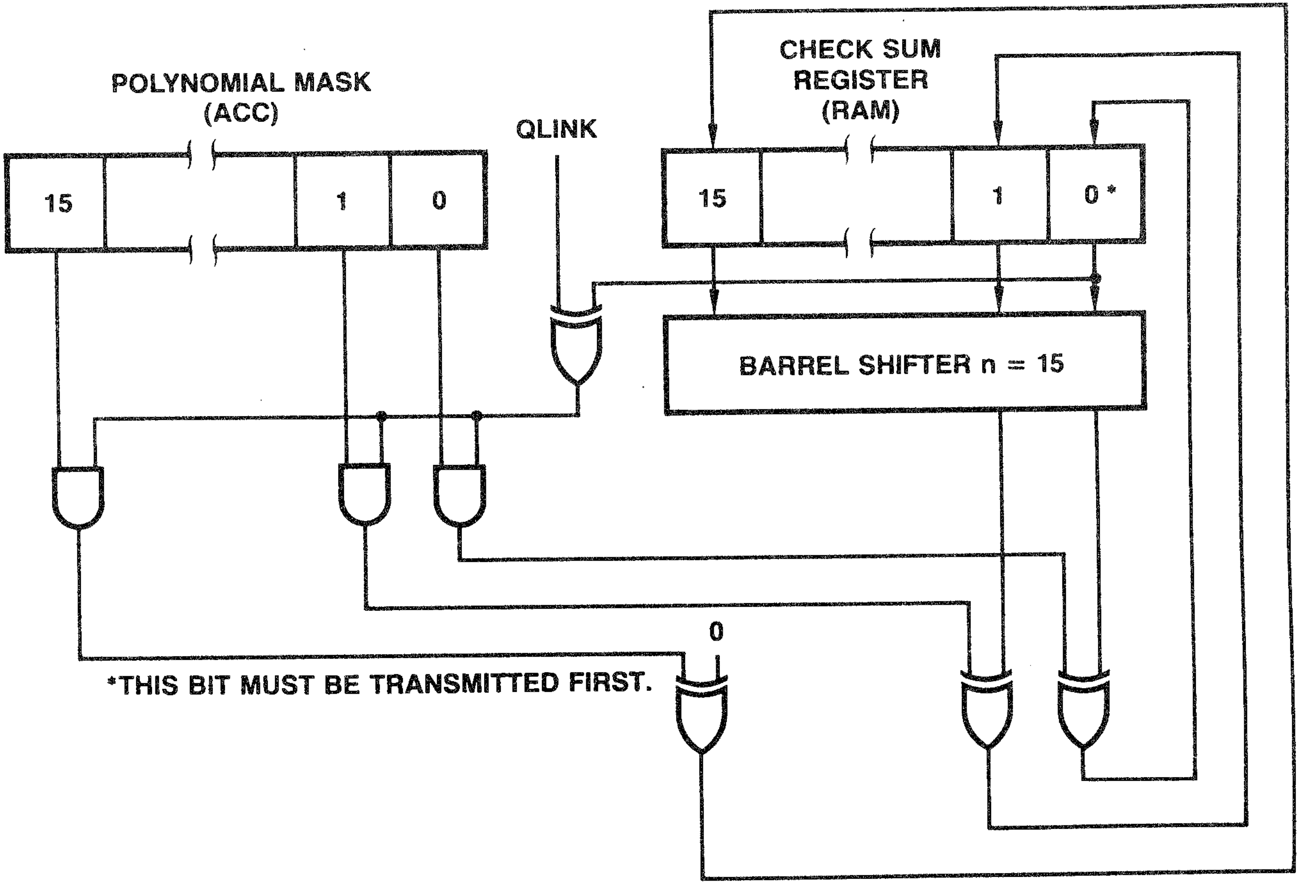
- CRC CALCULATIONS CANNOT BE DONE IN BYTE MODE; WORD MODE ONLY

CRC FORWARD FUNCTION



*THIS BIT MUST BE TRANSMITTED FIRST.

CRC REVERSE FUNCTION



*THIS BIT MUST BE TRANSMITTED FIRST.

STATUS INSTRUCTIONS

- SET STATUS

  SET ALL ALU BITS (Z C N OVR)

  SET LINK

  SET FLAG1

  SET FLAG2

  SET FLAG3

- RESET - SAME
- STORE STATUS TO DESTINATION
- LOAD STATUS

  BYTE MODE - LOAD 4 ALU BITS ONLY

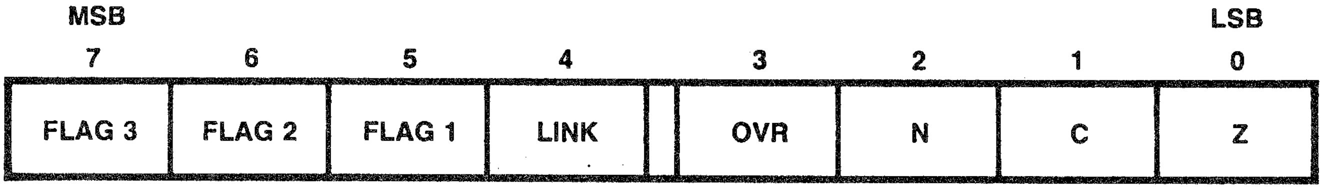
  WORD MODE - LOAD ALL 8 BITS

- ALU STATUS LOADED AFTER ALL INSTRUCTIONS EXCEPT

  NOP, STATUS INSTRUCTION (AS DETAILED EARLIER)

- IF STATUS REGISTER ENABLE IS ACTIVE, NO OPERATION IS PERFORMED ON THE STATUS REGISTER
- QLINK IS UPDATED AFTER EACH SHIFT
- FLAG1, FLAG2, FLAG3 ARE CHANGED USING SET/RESET

STATUS WORD



STATUS INSTRUCTIONS

SET/RESET	LOAD	SAVE
WORD	WORD	WORD
Z, C, N, OVR	Z, C, N, OVR	
LINK		
FLAG 1		
FLAG 2		
FLAG 3		

STATUS SET/RESET INSTRUCTIONS

FUNCTION	BIT(S) AFFECTED
SET	Z, C, N, OVR
CLEAR	LINK
	FLAG 1
	FLAG 2
	FLAG 3

CAN SET OR RESET ENTIRE WORD

STATUS STORE INSTRUCTIONS

SOURCE	DESTINATION
STATUS	RAM ACC NONE

STATUS LOAD INSTRUCTIONS
SINGLE OPERAND

FUNCTION	SOURCE (R)	DESTINATION
R → DEST.	RAM*	STATUS
$\overline{R}$ → DEST.	ACC	STATUS, ACC
R + 1 → DEST.	D	
$\overline{R}$ + 1 → DEST.	D (OE)	
	D (SE)	
	I	
	0	

*SEE SPECIAL CONDITIONS FOR RAM.

STATUS LOAD INSTRUCTIONS
TWO OPERAND

	OPERANDS		
FUNCTION	R	S	DESTINATION
S MINUS R	D	ACC	STATUS
S MINUS R WITH CARRY	ACC	I	STATUS, ACC
R MINUS S	D	I	
R MINUS S WITH CARRY			
R PLUS S			
R PLUS S WITH CARRY			
R AND S			
R NAND S			
R OR S			
R NOR S			
R EX-OR S			
R EX-NOR S			

TEST STATUS

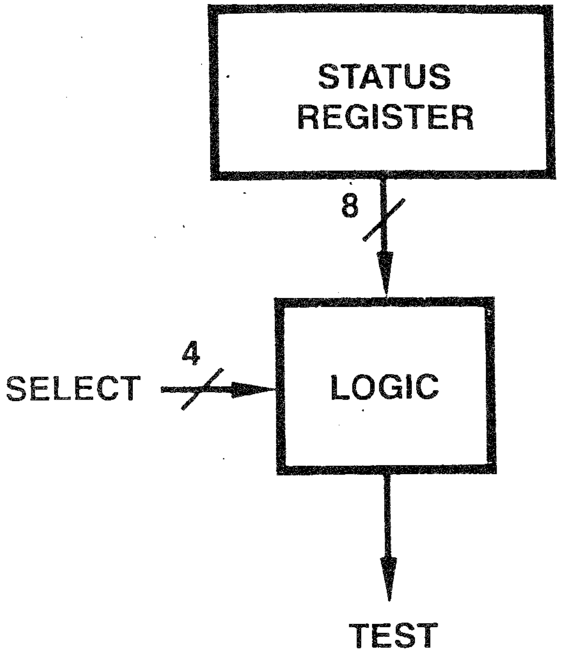
- ALL TESTING IS PERFORMED ON THE VALUES STORED (Qi) IN THE STATUS REGISTER

- 12 CONDITIONAL TESTS

QN	QC	QZ
QOVR	QLINK	QFLAG1
QFLAG2	QFLAG3	LOW
$QN \oplus QOVR$	$QZ + \overline{Q}\overline{C}$	$QN \oplus QOVR + QZ$

- TESTING CAN BE PERFORMED DURING ANOTHER INSTRUCTION'S EXECUTION BY USING THE Ti LINES AS TEST SELECT INPUT

STATUS



ALU TEST		OTHER TEST
Z	(=)	LINK
C	(≥)	FLAG 1
N		FLAG 2
OVR		FLAG 3
$N \oplus OVR$	(<)	LOW
$[N \oplus OVR] + Z$	(≤)	
$Z + \overline{C}$	(≤)	

INSTRUCTION SUMMARY

INSTRUCTION TYPE	OPERAND COMBINATIONS (NOTE 1)		
	SOURCE R/S		DESTINATION
SINGLE OPERAND	RAM (NOTE 2)		RAM
	ACC		ACC
	D		Y BUS
	D(0E)		STATUS
	D(SE)		ACC & STATUS
	I		
	0		
	R	S	DESTINATION
	RAM	ACC	
TWO-OPERAND	RAM	I	RAM
	D	RAM	ACC
	D	ACC	Y BUS
	ACC	I	
	D	I	
	SOURCE (U)		DESTINATION
	RAM		RAM
SINGLE BIT SHIFT	ACC		ACC
	ACC		Y BUS
	D		RAM
	D		ACC
	D		Y BUS

INSTRUCTION TYPE	OPERAND COMBINATIONS (NOTE 1)	
	SOURCE (U)	DESTINATION
ROTATE	RAM	RAM
	ACC	ACC
	D	Y BUS
BIT ORIENTED	RAM	RAM
	ACC	ACC
	D	Y BUS

FUNCTION			
	R	S	DESTINATION
PRIORITIZE (NOTE 3)	RAM	RAM	RAM
	ACC	ACC	ACC
	D	I	Y BUS
		0	
	S		DESTINATION
STORE STATUS	STATUS		RAM
			ACC
			Y BUS
	R	S	DESTINATION
STATUS LOAD	D	ACC	STATUS
	ACC	I	STATUS & ACC
	D	I	
		BITS AFFECTED	
SET RESET		Z, C, N, OVR	
		LINK	
		FLAG 1	
		FLAG 2	
		FLAG 3	
	ROTATED OPERAND (U)	MASK (S)	NON-ROTATED OPERAND/ DESTINATION (R)
ROTATE & MERGE	D	I	ACC
	D	RAM	ACC
	D	I	RAM
	D	ACC	RAM
	ACC	I	RAM
	RAM	I	ACC

	ROTATED OPERAND (U)	MASK (S)	NON-ROTATED OPERAND/ DESTINATION (R)
ROTATE & COMPARE	D	I	ACC
	D	I	RAM
	D	ACC	RAM
	RAM	I	ACC

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 INSTRUCTIONS, OPERAND and MASK must be different sources.

THE WESCON PAPER

A HIGH-PERFORMANCE 16-BIT BIPOLAR MICROPROCESSOR--

THE Am29116

WESCON PAPER (1979)

William J. Harmon, Jr.
Warren K. Miller

Advanced Micro Devices
901 Thompson Place
Sunnyvale, CA 94086

INTRODUCTION

The Am29116 is a high-performance 16-bit bipolar microprocessor intended for use in microprogrammed systems, particularly peripheral controllers, although it is also suitable for use in communication controllers, industrial controllers and digital modems. The chip can also be used in microprogrammed processor applications. In addition to its complete arithmetic and logic instruction set, the Am29116 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.

OUTSTANDING FEATURES

16-Bit Data Path - The Am29116 contains a 16-bit data path with full carry lookahead over all 16 bits in the ALU during arithmetic operation. In order to facilitate interfacing the device to other circuits, the Am29116 has the ability to execute all instructions in either the 16-bit word or 8-bit byte mode.

32 Working Registers - In order to provide adequate on-chip storage, the Am29116 contains 32 working registers arranged in a single port RAM architecture. With the use of an external multiplexer, it is possible to select separate read and write addresses for the same instruction. The device also contains a 16-bit Accumulator and a

16-Bit Barrel Shifter - A 16-bit Barrel Shifter which can rotate an input up to 15 positions is also included in the device. Like the ALU, the barrel shifter can work in either the word or byte mode.

Status Register and Condition-Code Generator/Multiplexer - The Am29116 contains an 8-bit Status Register and a Condition-Code Generator/Multiplexer. The Status Register stores the four ALU status outputs, Z, C, N, OVR, as well as a Link bit for shifting and three user-definable Flag bits. The Condition-Code Generator/Multiplexer allows testing of 12 different test conditions. The output of the Condition-Code Generator/Multiplexer can be connected directly to the conditional-test input of a microprogram sequencer.

Immediate Instruction Capability - Immediate instructions can be executed by the Am29116. These are two-microcycle instructions. The first instruction contains information necessary to perform the instruction. The second instruction contains immediate data, which is entered via the 16 Instruction Inputs.

CRC Generation - The Am29116 has instructions which perform CRC, (Cyclic-Redundancy Check), calculations for any CRC polynomial of 16 bits or less.

Powerful Instruction Set - The instruction set of the Am29116 is very powerful. In addition to the normal single- and two-operand logical and arithmetic instructions, the Am29116 can also execute the following instructions in a single microcycle: rotate and merge, rotate and compare, and prioritize.

ARCHITECTURE OF THE Am29116

The Am29116 is a high-performance, microprogrammable 16-bit bipolar microprocessor. This 48-pin device is designed internally with ECL (emitter-coupled logic) circuitry and has TTL to ECL and ECL to TTL converters on all inputs and outputs. The design goal is to execute all microinstructions in 100 nanoseconds over the commercial operating range.

All data paths within the device are 16-bits wide. As shown in the Block Diagram, Figure 1, the device consists of the following:

- 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 $\overline{\text{IEN}}$ 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.

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 $\overline{\text{IEN}}$ 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 Am29116 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 Am29116 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 subtraction.

Priority Encoder - The Priority Encoder produces a binary-weighted code to indicate the location 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.

With the Status-Register Enable, $\overline{\text{SRE}}$, input LOW and the $\overline{\text{IEN}}$ input LOW, the Status Register is updated at the end of all instructions except NO-OP, Save-Status and Test-Status instructions. $\overline{\text{SRE}}$ going HIGH or $\overline{\text{IEN}}$ 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 $\overline{\text{SRE}}$ LOW and $\overline{\text{IEN}}$ 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 microcode, but provides the ability to simultaneously test and execute.

Three-State Output Buffers - There are two sets of Three-State Output Buffers in the Am29116. 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 control 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.

Bit Instructions - The Bit Instructions contain four indicators: byte or word mode, operation, source/destination, and the address of the bit to be operated on.

The operations which can be performed are: set bit N which forces the N^{th} bit to a ONE; reset bit N, which forces the N^{th} bit to ZERO; test bit N, which sets the ZERO Status Bit depending on the state of bit N; 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.

Prioritize Instruction - The Prioritize Instructions contain four indicators: byte or word mode, R operand, Mask operand (S), and destination.

The function of the Prioritize Instructions 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.

Single-Operand Instructions - Single-Operand Instructions contain four indicators: byte or word mode, operation, source and destination.

The operations which can be performed are Pass, Complement, Increment and Two's Complement.

Two-Operand Instructions - Two-Operand Instructions contain five indicators: byte or word mode, operation, R operand, S operand, and destination.

The possible operations are R plus S, R plus S plus Carry, R minus S, S minus R, R minus S with Carry, S minus R with Carry, R AND S, R NAND S, R OR S, R NOR S, R EX-OR S, R EX-NOR S.

Shift Instructions - Shift Instructions contain four indicators: byte or word mode, direction and shift linkage, source and destination.

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. 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 ($QN \oplus QOVR$) or the Link-Status bit. The LSB is loaded into the Link-Status bit.

Bit Instructions - The Bit Instructions contain four indicators: byte or word mode, operation, source/destination, and the address of the bit to be operated on.

The operations which can be performed are: set bit N which forces the N^{th} bit to a ONE; reset bit N, which forces the N^{th} bit to ZERO; test bit N, which sets the ZERO Status Bit depending on the state of bit N; 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.

Prioritize Instruction - The Prioritize Instructions contain four indicators: byte or word mode, R operand, Mask operand (S), and destination.

The function of the Prioritize Instructions 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.

indicators: byte or word mode, source, destination and the number of places the source is to be rotated.

The N indicator specifies the number of bit positions the source is to be rotated up (0 to 15). In the word mode, all 16-bits are rotated up while in the byte mode, only the lower 8-bits (0 to 7) are rotated up.

Rotate and Merge Instructions - The Rotate and Merge Instructions contain five indicators: byte and word mode, rotated operand, non-rotated operand/destination, mask and number of bit positions the rotated operand is to be rotated.

The function performed by the Rotate and Merge Instruction

The rotated operand, U, is rotated 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 rotated U input as the i^{th} output bit, while a ONE in bit i will select the i^{th} R input as the output bit. The output word is stored in the non-rotated operand location.

Rotate and Compare Instructions - The Rotate and Compare Instructions contain five indicators: byte or word mode, rotated operand, non-rotated operand, mask, and number of bit positions the rotated operand, is to be rotated.

The function performed by the rotate and compare instruction

The rotated operand is rotated by the Barrel Shifter N places. The Mask is inverted 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.

CRC Instructions - The CRC (cyclic redundancy check) Instructions provide a method for generation of the check bits in a CRC calculation.

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.

Two CRC instructions are provided-CRC Forward and CRC Reverse The difference in these two instructions arises because 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.

NO-OP Instruction - The No-op Instruction does not change any internal registers in the Am29116. It preserves the status register, RAM registers and the ACC register.

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

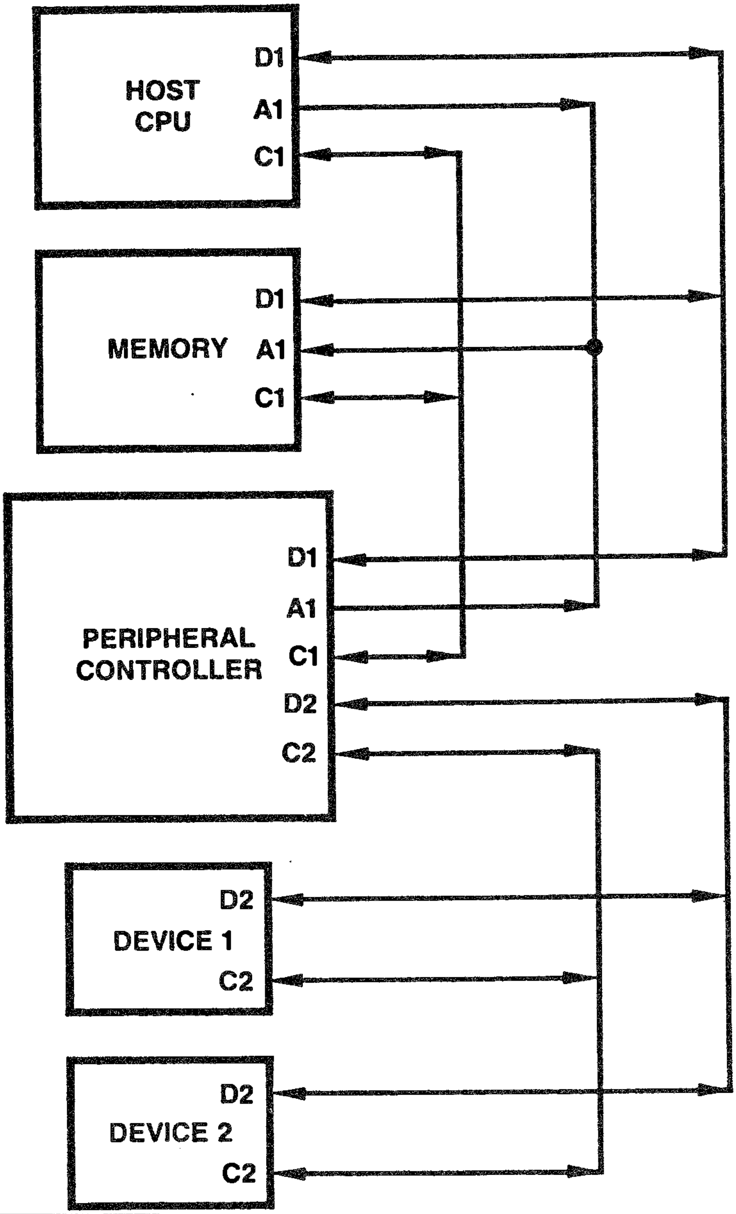
The Reset Status Instruction contains a single indicator. This indicator specifies which bit or group of bits, contained in the status register are to be reset (forced to ZERO).

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

The Load Status Instruction contains two indicators. The indicators specify the byte or word mode and the source for the Status Register. In the byte mode only, the lower 4-bits (QC, QN, QZ, QOVR) are loaded from the source. In the word mode, all 8-bits of the Status Register are loaded from the source.

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, QFLAG2 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 of 2's complement and unsigned numbers.
The Status Register may also be tested via the bidirectional T bus. See the discussion on the Status Register for a full description.

TYPICAL SYSTEM CONFIGURATION



Am29116 APPLICATIONS

The intended primary applications for the Am29116 are high-performance peripheral controllers. Figure 14 shows a typical system configuration for a Host Computer, Memory and Peripheral Controller. The interface between the three units is via three buses; the data bus (D1), the address bus (A1) and the control bus (C1). The interface between the Peripheral Controller and the Peripheral Devices is via a data bus (D2) which may be either serial or parallel, and a control bus (C2). Information on the control buses consists of status, command and timing signals.

A typical implementation of the Peripheral Controller is shown in Figure 15. The bidirectional interface to the D1 data bus is via two Am2950 8-Bit Parallel I/O Ports; two Am2940 8-bit DMA Address Generators drive the A1 bus and another Am2950 interfaces to the bidirectional C1 bus. The interface to the serial D2 bus is via a parallel-to-serial and a serial-to-parallel converter, and the bidirectional interface to the C2 bus is via two Am2950s. The interface between these bus-interface units and the Am29116 is a 16-bit bidirectional bus which connects to the Y 0-15 outputs of the 29116. Also connected to this bus is a 256-word RAM for temporary data storage and a 12-bit interface (1-1/2 Am2920s) to the D_{0-11} inputs of the Am2910 Microprogram Sequencer. The bus-control and clock-enable signals for these devices are generated by the Pipeline Register at the output of the Microprogram Memory.

The Am29116, Am2910 and the Microprogram Memory perform the data manipulation and routing; command and status testing and generation; and timing-signal generation functions. The implementation illustrated in Figure 15 minimizes the amount of hardware necessary to implement a controller. This is accomplished by A) sharing the Instruction-Inputs to the Am29116 with the D_{0-11} inputs to the Am2910, B)generating all necessary test conditions within the 29116 which allow connecting the CT output of the 29116 directly to the $\overline{CC}$ inputs of the Am2910, C) by generating the CT output via the Instruction Inputs, D) performing all the necessary status manipulations within the Am29116, E) using the same RAM address for reading and writing and F) running the controller at a fixed clock rate.

PERIPHERAL CONTROLLER, MINIMUM PARTS CONFIGURATION

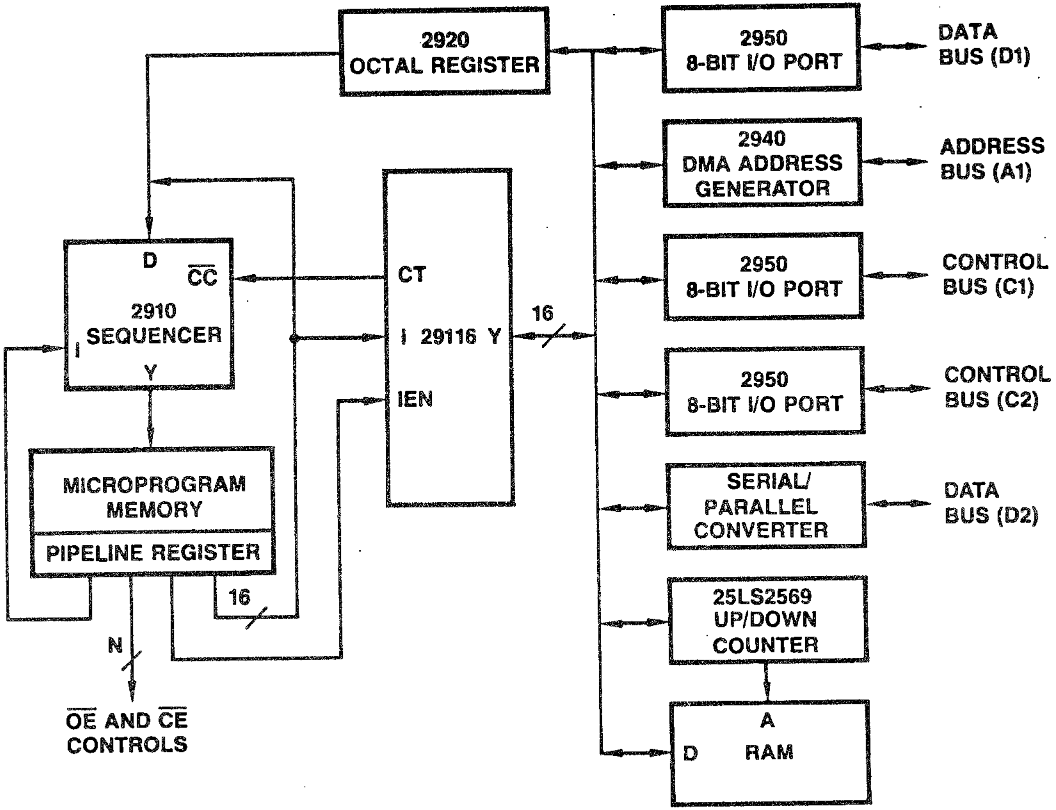


Figure 15

Maximizing Throughput

Although the implementation shown in Figure 15 minimizes the amount of required hardware, it does limit the throughput of the controller. The architecture shown in Figure 16 uses the same bus interface circuits but maximizes the throughput of the controller at the expense of additional hardware. In this implementation,

the Instruction Inputs of the Am29116 and the D_{0-11} inputs of the Am2910 are driven from separate microcode bits; this allows simultaneous instruction execution in the Am29116 and direct jumping in the Am2910. The multiplexer at the $\overline{CC}$ input of the Am2910 allows testing of conditions without loading the signals into the Am29116. Four additional bits of microcode drive the T_{1-4} inputs of the Am29116; this allows simultaneous conditional testing and execution of an instruction in the Am29116. The Am2904 can be loaded with the four ALU arithmetic status bits (Z, C, N, OVR). The flexibility of the Am2904, such as selective loading of status bits, reduces the number of cycles necessary to perform status manipulation. By adding five additional microcode bits and a multiplexer at the I_{0-4} inputs of the Am29116, separate RAM source and destination addresses can be used in the same microcycle; for example, the contents of RAM address 3 can be added to the contents of the Accumulator and the results can be stored in RAM Address 27. The Am2925 System-Clock Generator and Driver, in addition to providing the basic oscillator and clock driver functions, provides the ability to dynamically alter the length of the microcycle; this facilitates interfacing the Am29116 to slower bus interface and peripheral circuits.

Figures 15 and 16 are intended to show the two extremes of minimizing hardware versus maximizing throughput.

PERIPHERAL CONTROLLER,
MAXIMUM PERFORMANCE CONFIGURATION

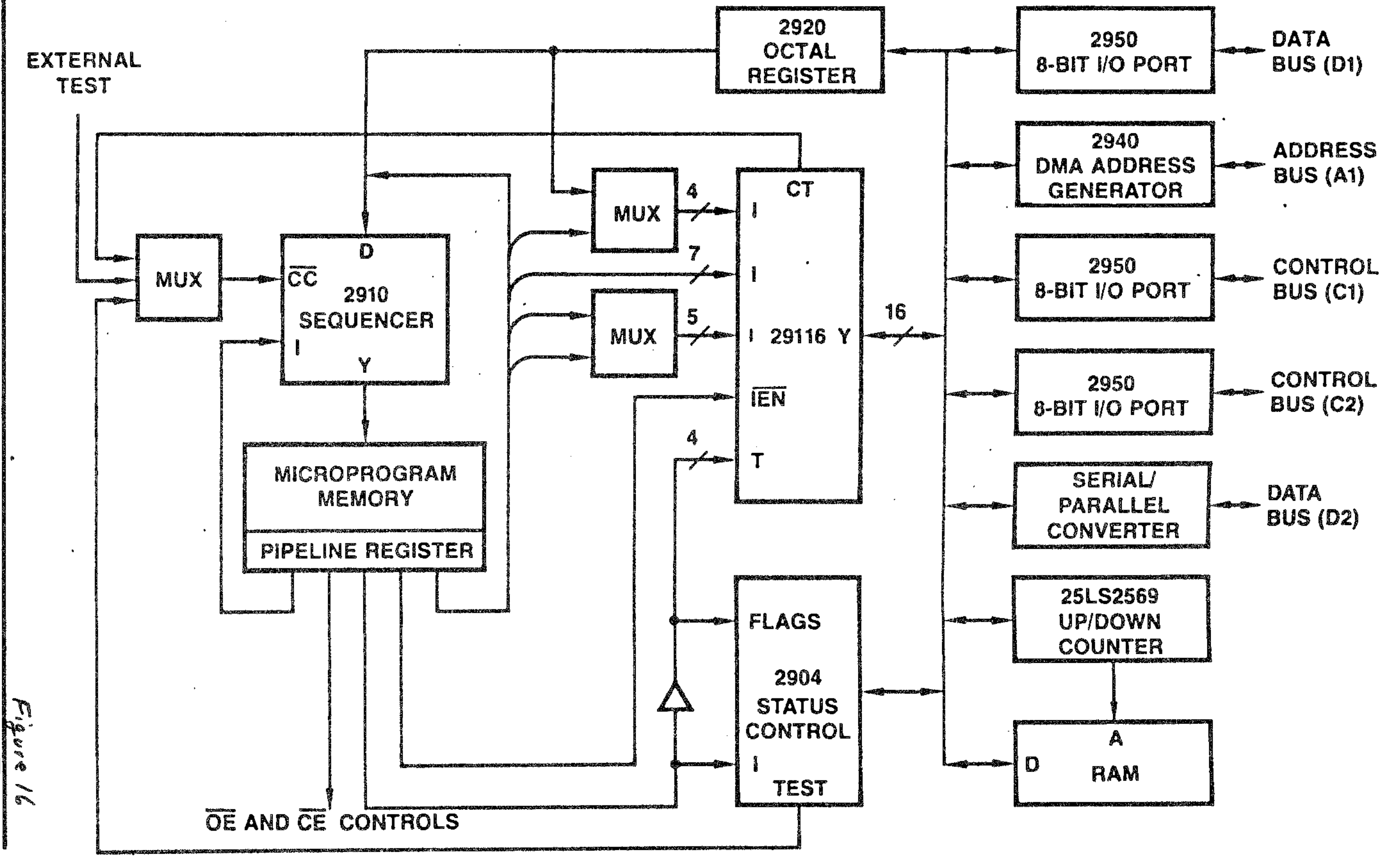


Figure 16

MICROWORD FORMAT

Am29116 MICROWORD FORMAT

IF IT IS ASSUMED THAT THE Am2910 IS USED AS THE SEQUENCER FOR THE Am29116 THEN THE MICROWORD MIGHT LOOK AS FOLLOWS:

2910 INSTR (4)	COND MUX (3)	BRANCH ADDR /COUNTER (12)	29116 INSTRUCTION/ IMMEDIATE DATA (16)	$\overline{I}_{EN}$ (1)	$\overline{OE}_{Y}$ (1)	DLE (1)	$\overline{SRE}$ (1)	OE_{T} (1)	TEST INSTR (4)	● ● ●

THE CONDITIONAL MULTIPLEXER MAY VERY WELL BE REPLACED BY AN Am2904

THE TEST INSTRUCTION FIELD IS OPTIONAL (ONLY USED WHEN A TEST IS TO BE PERFORMED DURING ANOTHER INSTRUCTION'S EXECUTION)

THE Am29116 INSTRUCTION FIELD IS OVERLAYED BY THE IMMEDIATE DATA FOR THE Am29116 (THE INSTRUCTION IS LATCHED ON-BOARD THE Am29116)

.DEF 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
PRTR1	PRIORITIZE RAM; TYPE 1
PRTR2	PRIORITIZE RAM; TYPE 2
PRTR3	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
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
SHDN2	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

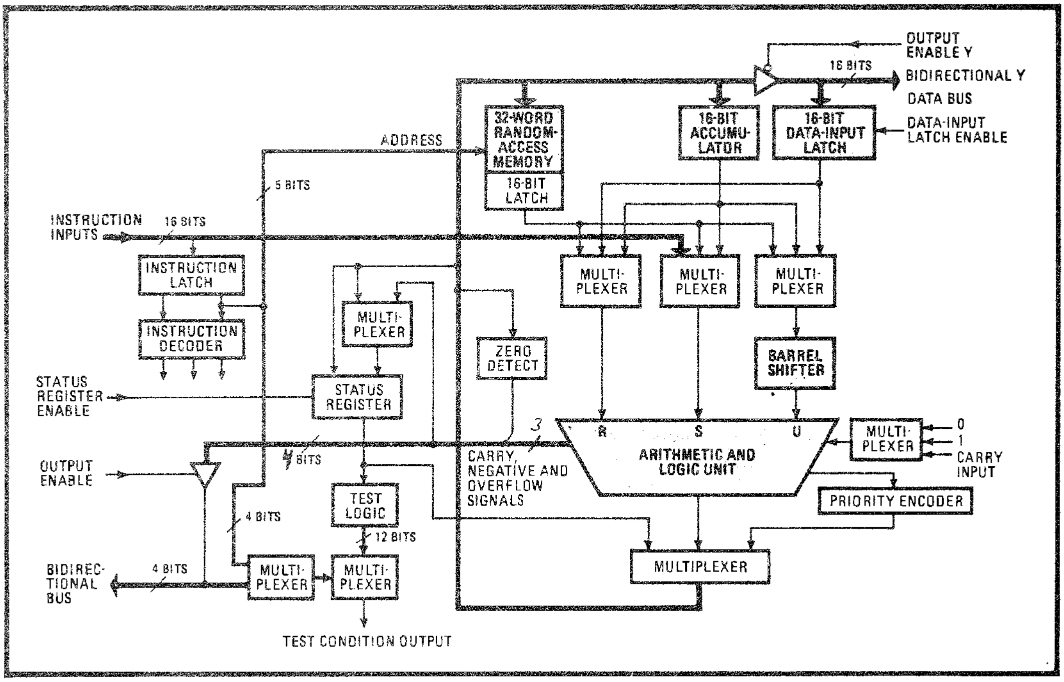
MOVE	MOVE AND UPDATE STATUS
COMP	COMPLEMENT (1's COMPLEMENT)
INC	INCREMENT
NEG	TWO's COMPLEMENT

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

Z8000 - 2900 - vrs. others

Am29116



Comparison of execution times

Processor type	Bit test	Bit set	16-bit add	Rotate by N	Rotate by N and merge
Am8048	2* 5000 ns**	1 2500 ns	6 15,000 ns	12 382,500 ns	28 422,500 ns
Am9080	5 4250 ns	2 1750 ns	1 2500 ns	17 200,000 ns	37 225,000 ns
Am29116	1 100 ns	1 100 ns	1 100 ns	1 100 ns	1 100 ns

* Number of instruction words ** Execution time

PERFORMANCE ANALYSIS

PROCESSOR TYPE	INSTRUCTION TYPE	
	BIT TEST	BIT SET
8048	2 5,000ns	1 2,500ns
9080	5 4,250ns	2 1,750ns
Z8000	1 1,000ns	1 1,000ns
2901B (4) 2904 2902A 25LS2538 (2)	1 100ns	1 100ns
2901B (4) 2904 2902A 25LS2538 (2) 25S10 (8)	1 100ns	1 100ns
29116	1 100ns	1 100ns

OF INSTRUCTION WORDS / EXECUTION TIME

PERFORMANCE ANALYSIS

PROCESSOR TYPE	INSTRUCTION TYPE	
	ROTATE BY N	ROTATE AND MERGE
8048	28 212,500ns	44 252,500ns
9080	32 113,000ns	42 138,000ns
Z8000	8 11,250ns	12 15,250ns
2901B (4) 2904 2902A 25LS2538 (2)	9 1,025ns	12 1,325ns
2901B (4) 2904 2902A 25LS2538 (2) 25S10 (8)	1 160ns	4 460ns
29116	1 100ns	1 100ns

OF INSTRUCTION WORDS / EXECUTION TIME

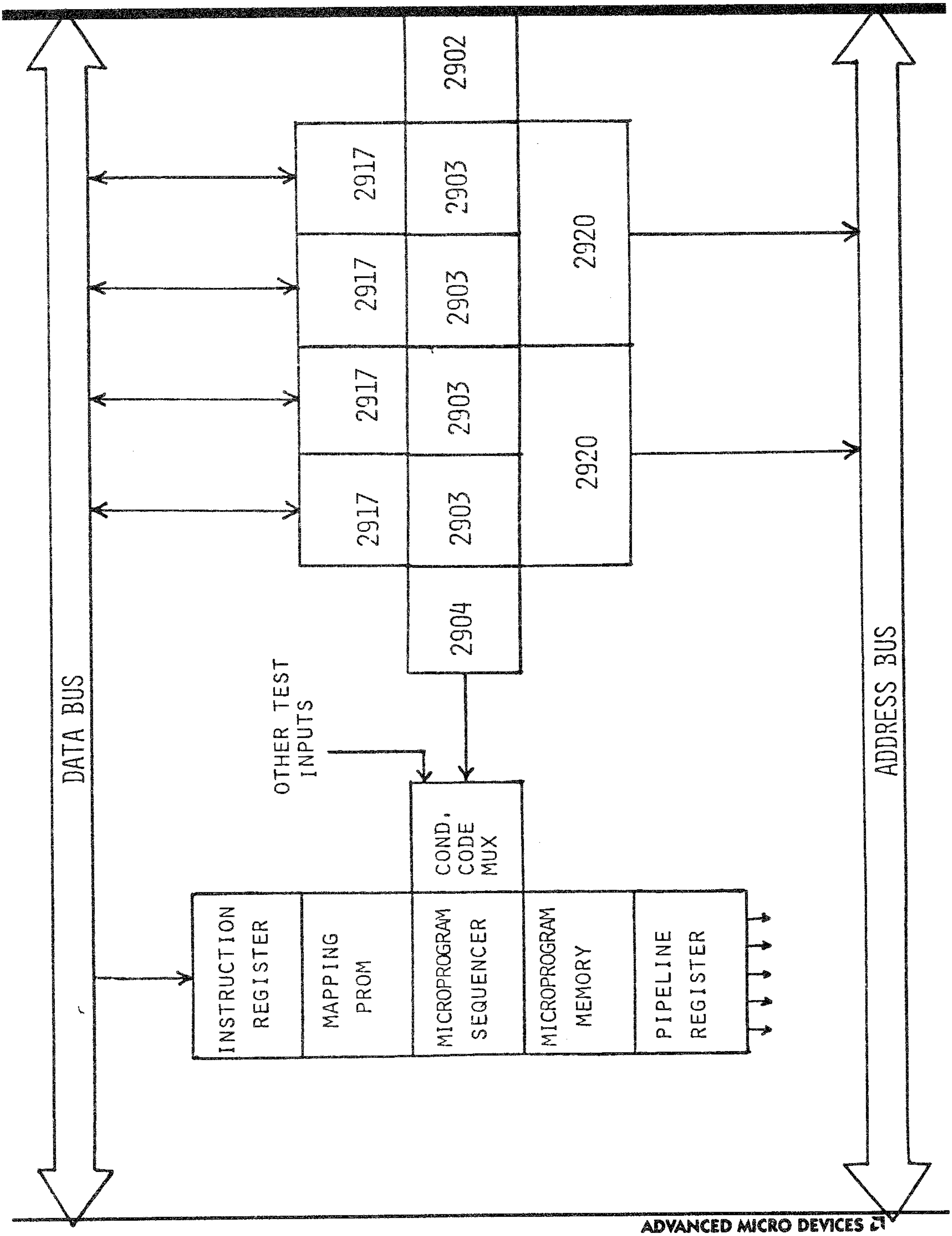
PERFORMANCE ANALYSIS

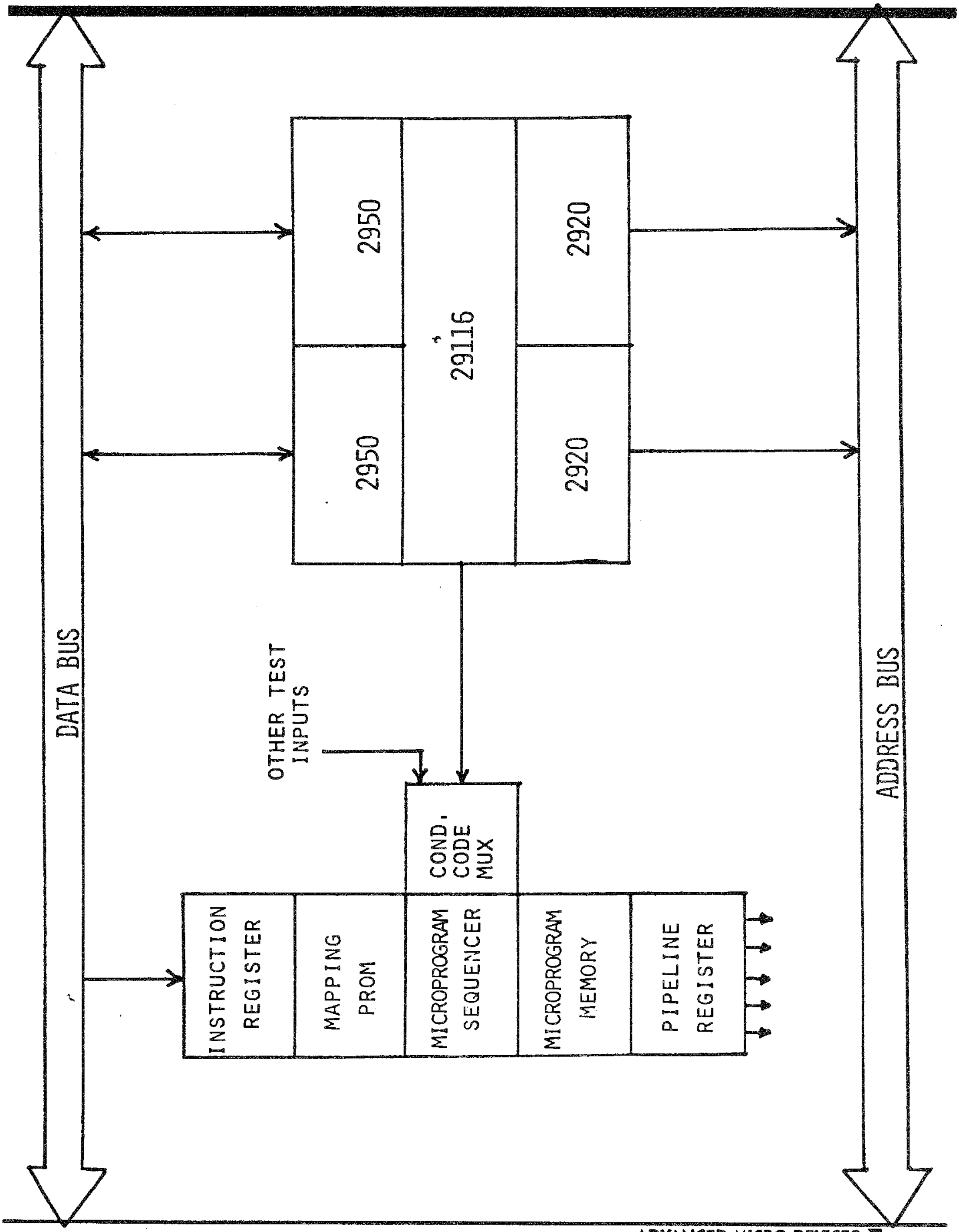
PROCESSOR TYPE	INSTRUCTION TYPE	
	16-BIT ADD	16-BIT MULTIPLY
8048	6 15,000ns	54 810,000ns
9080	1 2,500ns	40 509,000ns
Z8000	1 1,000ns	1 17,500ns
2901B (4) 2904 2902A 25LS2538 (2)	1 115ns	2 2,000ns
2901B (4) 2904 2902A 25LS2538 (2) 25S10 (8)	1 115ns	2 2,000ns
29116	1 100ns	12 9,600ns

OF
INSTRUCTION
WORDS

EXECUTION
TIME

THE Am29116 AS A CPU


DATA BUS
ADDRESS BUS
2902
2917
2917
2917
2917
2903
2903
2903
2903
2920
2920
2904
OTHER TEST INPUTS
COND. CODE MUX
INSTRUCTION REGISTER
MAPPING PROM
MICROPROGRAM SEQUENCER
MICROPROGRAM MEMORY
PIPELINE REGISTER


DATA BUS
2950
2950
29116
2920
2920
OTHER TEST INPUTS
COND. CODE MUX
INSTRUCTION REGISTER
MAPPING PROM
MICROPROGRAM SEQUENCER
MICROPROGRAM MEMORY
PIPELINE REGISTER
ADDRESS BUS

EXERCISES

EXERCISES - Am29116

True or False:

1. The Am29116 is a TTL compatible, ECL internal device.

2. The Am29116 is expandable (two can be hooked together).

3. The Am29116 is for 8 bit or 16 bit intelligent controllers.

4. The Am29116 can perform conditional testing on its status register.

5. The barrel shifter 'shifts' (rotates) 1 to 15 bits up or down in one microcycle.

6. The Am29116 must be used with an Am2904.

7. The Am29116 can perform immediate operations.

8. The Am29116 has a choice of four input sources to the MUX which in turn provides three ALU inputs, R, S, U.

9. The Am29116 can perform three address instructions.

10. The Am2922 (MUX) and the Am2904 (GLUE) both have polarity control on the conditional inputs.

11. Fast clock speed is synonomous with high throughput.

12. The Am29116 can generate CRC polynomials up to 16 bits long.

13. The Am29116 always has its ALU output at Y_i.

14. The ALU destinations are RAM, ACC, D Latch.

15. Single operand instructions are PASS, COMPLEMENT, INCREMENT, and TWO'S COMPLEMENT.

16. D_{OE} (D with zero extend) is used for Two's complement arithmetic.

17. $\bar{R}$ --> DEST is One's complement, $\bar{R}$ + 1 --> DEST is Two's complement.

18. The Am29116 can perform NAND, NOR, EXOR in one microcycle.

19. Shift up can use 0, 1, or the QLINK bit as input to the LSB.

20. Shift down uses 0, 1, or the QLINK bit as the only input choices to the MSB.

21. Rotate can work in byte or word mode.

22. Rotate uses the U ALU input.

23. LOAD 2^n causes a mask (1 in a field of 0s) to be generated and used for loading RAM, ACC.

24. Read bus, change bit, output to bus is possible in one microcycle with the Am29116.

25. If you bit change the ACC, the destination is the ACC or the RAM.

26. There are 17 choices for priority encoding.

27. Byte mode prioritize does not use bits 8 - 15.

28. The Am29116 can perform operations on up to and including 16 bit CRC polynomials.

29. 95% of CRC calculations use 16 bit polynomials.

30. The CRC calculations can be done forward or reverse (transmit bit 0 first or transmit bit 15 first).

31. CRC can be done in byte or word mode.

32. The status word can be loaded from D, RAM, ACC.

33. The Z, C, N, OVR status bits can be loaded without affecting LINK or FLAGs.

34. You can set or reset the entire status word on the Am29116.

35. You can set or reset the arithmetic flags individually on the Am29116.

36. On the Am29116, you can load the status and the ACC registers both in the same microcycle if the RAM is the source.

37. If the status register enable is on, the status register is frozen (any operation on the status register is "killed").

38. All conditional testing is performed on the stored values in the status register.

FILL IN OR ANSWER:

39. How many bits in the Am29116 status register?

40. How many conditional tests can be made?

41. Can you perform a conditional test during another instruction? If so, how?

42. Can byte operations be performed in the upper or lower byte of the 16-bit Am29116?

43. Can the Am29116 support a 100ns microcycle time?

44. List three possible sources for single operand instructions.

45. Can you load the D (data) latch one byte at a time?

46. List three possible source pairs for two operand instructions.

47. If n = 2, show the pattern in the 16-bit word after a word rotate given:

1	1	1	1	1	1										
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
1	1	0	0	1	0	1	0	0	0	1	1	0	1	0	1

48. Does the Am29116 support $\langle R_3 \rangle$ .ROTATE. --> $\langle R_3 \rangle$?

49. Does the Am29116 support $\langle R_3 \rangle$ .ROTATE. --> $\langle R_7 \rangle$?

50. If there is no priority request, what is produced by the priority encoder (word mode)?

51. If bit 15 is active, what is produced (word mode)?

52. If bit 15 is active, what is produced (byte mode)?

53. The Am29116 is an order of magnitude faster than the Am2901 which is an order of magnitude faster than the AmZ8000 for specific controller oriented operations. What can you say about program area?

54. Can the Am29116 be used to do multiply?

55. Can the Am29116 be used for bit operations?

56. Can the Am29116 be used for rotate operations?

57. Does the Am29116 have an ALU?

58. Can the Am29116 be used to build a CPU?

59. If the mask bit i is zero in a ROTATE-MERGE instruction, which operand's bit i is passed to the destination?

60. If U = 0011 0001 0101 0110
R = 1010 1010 1010 1010
MASK = 0101 1010 0110 1001
and n = 4
what bit pattern is produced by a word mode ROTATE-MERGE instruction?

61. If the highest bit position with a one (1) is bit position 7, and the mask is 1010_{HEX}, what is produced by a word prioritize instruction?

62. Repeat for byte mode prioritize.

63. For LOAD 2^N complement, ($\bar{2}^N$), what happens?

64. Suppose you want to do a word rotate down five bit positions. How do you do this on an Am29116?

65. What is QLINK?

66. Can you set/reset the ALU status bits one by one as with the Am2904 Machine status register?

67. Can you set/reset the LINK and FLAGi status bits one at a time?

68. In byte mode, are the lower 8 bits of the status register loaded?

69. Which instructions do NOT cause the ALU status bits to be updated?

70. When are the upper four status bits (LINK, FLAG1, FLAG2, FLAG3) changed?

SOLUTIONS

1. True

2. False

3. True - byte or word operation all almost every instruction

4. True - using either the instruction lines, I_i, or the T_i lines

5. False - the barrel shifter rotates only 1 to 15 bits UP. An effective down shift is achieved by choosing the appropriate number of up rotates to be equivalent to the down rotate desired.

6. False - use an Am2904 for emulations (bit settable status registers)

7. True - requires two microwords (two microcycles if 1 microinstruction executes in 1 microcycle)

8. True - I_i, ACC, D Latch, or RAM

9. True - control timing and use an external MUX similiar to the procedure used by the Am2903/Am29203

10. True

11. False! - remember the branch on previous - NOP two step required by the double pipelined Am2910 - Am2901/2903/29203 "typical" CPU (ED2900B)

12. True

13. True IF $\overline{OE}_Y$ is enabled else false

14. True with reservations - the D latch can be used but requires some tricky timing, you could generate a race condition if the D latch is also a source for the operation. The D latch is not intended to be a destination, and its use as such is not recommended. Normal destinations: RAM, ACC or NONE.

15. True

16. False - D_{SE} (D sign extend with bit 7 extended) is used for Two's complement arithmetic

17. True

18. True

19. True Shift uses the 0, 1 or QLINK. The inputs to the carry MUX should not be confused with the shift inputs.

20. False - also Q_C, $C_N \oplus Q_{OVR}$ (Q_i is used to indicate the status register contents in the Am29116 literature; not to be confused with the Q register of the RALUs)

21. True

22. True

23. True - also goes to Y_i

24. True - watch timing

25. False - ACC only, instruction has common source/destination field

26. True - none and 0 - 15

27. True

28. True

29. True - AMD survey

30. True

31. False!

32. True - The status can also be loaded from the Instruction lines as well

33. True

34. True

35. False - if you need this use an Am2904

36. False - this is the one source which disallows loading the ACC

37. True

38. True

ANSWERS TO FILL IN OR ANSWER SECTION:

39. Z, C, N, OVR, LINK, FLAG1, FLAG2, FLAG3

40. There are 12 condition code test signals:
the 8 status bits themselves (one at a time)
plus: $N \oplus OVR$ $[N \oplus OVR] + Z$
$Z + \bar{C}$ LOW

41. Yes - by using the T_i lines as input. This requires a wider microword so that both the T_i instruction lines and the I_i instruction lines are present.

42. lower *only*

43. Yes - *very carefully*! Requires a register between the microcontroller and the control storage (usually a PROM memory) as well as the pipeline register at the control storage output. 125ns is more easily achieved. Timing studies will be more detailed when the part is released (end of 1980).

44. RAM, ACC, D, I, ZERO

45. NO

46. RAM-ACC, RAM-I, D-RAM, D-ACC, ACC-I, D-I

47. Rotate is UP only:

1100 1010 0011 0101 n=2 becomes 0010 1000 1101 $01\underline{11}$

48. Yes

49. Yes - watch your timing.

50. 0000

51. 0001

52. Bit 15 does not participate in the byte mode.

53. The program space is larger for the faster Am29116.

54. Yes - but it was not intended for this application.

55. Yes

56. Yes - the barrel shifter works in the byte or word mode.

57. Yes - 16-bit ALU

58. Yes - but it was not intended for this application.

59. If the mask bit i is zero in a ROTATE-MERGE instruction, the ith bit of the U operand is passed to the destination.

60. If
    U = 0011 0001 0101 0110
    R = 1010 1010 1010 1010
    MASK = 0101 1010 0110 1001
    and n = 4
    the bit pattern that is produced by a word mode ROTATE-MERGE instruction is 1011 0000 1110 0011.

61. If the highest bit position with a one (1) is bit position 7, and the mask is 1010_{HEX}, 0009_{HEX} is produced by a word prioritize instruction.

62. Repeating for byte mode prioritize, ~~same~~ the answer ~~as~~ is ~~61~~ 15.

63. For LOAD $\overline{2}^{N}$, ZERO --> bit N, ONE --> all other bits in destination.

64. To do a word rotate down five bit positions, do a rotate up of n = 11 [16 - 5 = 11].

    Example: 0000 1111 0101 1010
    1101 0000 0111 1010

65. QLINK is the linkage bit for shift operations; also for CRC instructions (serial input).

66. You CANNOT set/reset the ALU status bits one by one as with the Am2904 Machine status register. They function more like the microstatus register.

67. You CAN set/reset the LINK and FLAGi status bits one at a time.

68. In byte mode, the lower 4 bits of the status register are loaded.

69. The instructions which do <u>NOT</u> cause the ALU status bits to be updated are: NOP, status register instructions, save status, test status, or any instruction if either

    $\overline{SRE}$ or $\overline{IEN}$

    are not LOW.

70. The upper four status bits (LINK, FLAG1, FLAG2, FLAG3) are changed during status set/reset; status load (word mode only); plus QLINK is updated after each shift.