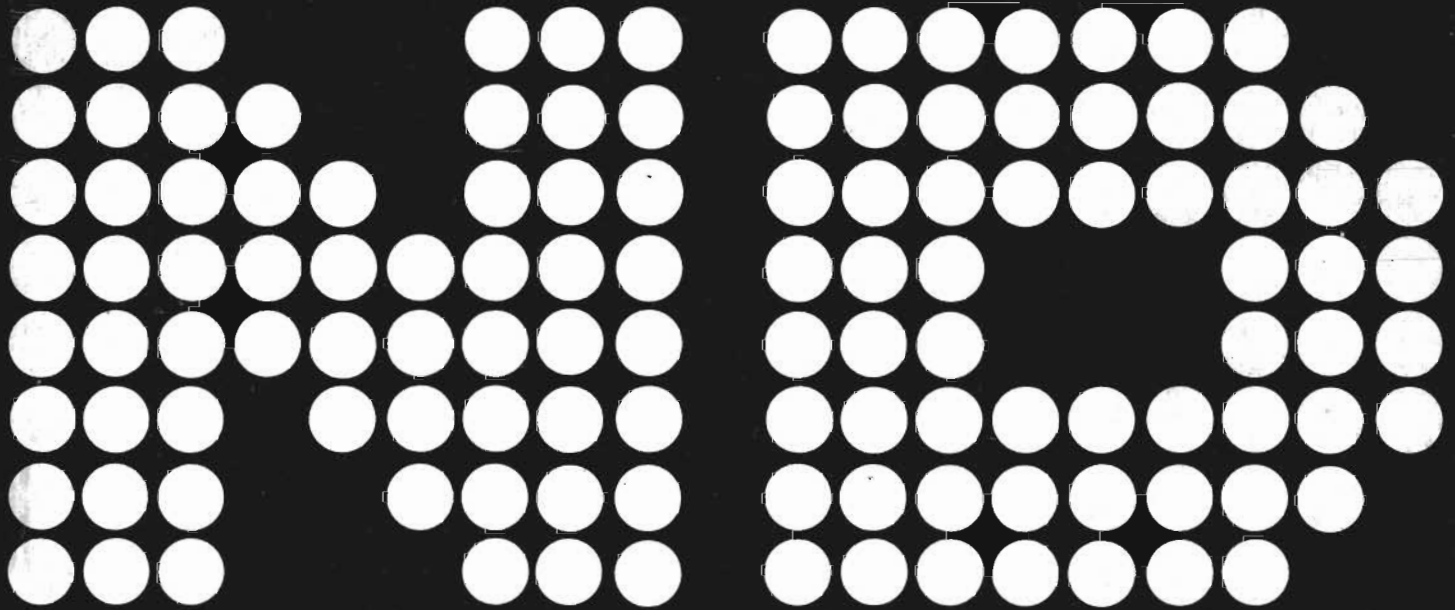
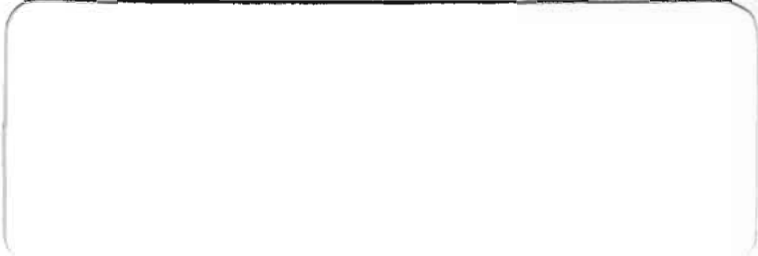


**NORD-100
Microprogramming
Description**

NORSK DATA A.S





NORD-100
Microprogramming
Description

NOTICE

The information in this document is subject to change without notice. Norsk Data A.S assumes no responsibility for any errors that may appear in this document. Norsk Data A.S assumes no responsibility for the use or reliability of its software on equipment that is not furnished or supported by Norsk Data A.S.

The information described in this document is protected by copyright. It may not be photocopied, reproduced or translated without the prior consent of Norsk Data A.S.

Copyright © 1979 by Norsk Data A.S.

Manuals can be updated in two ways, new versions and revisions. New versions consist of a complete new manual which replaces the old manual. New versions incorporate all revisions since the previous version. Revisions consist of one or more single pages to be merged into the manual by the user, each revised page being listed on the new printing record sent out with the revision. The old printing record should be replaced by the new one.

New versions and revisions are announced in the ND Bulletin and can be ordered as described below.

The reader's comments form at the back of this manual can be used both to report errors in the manual and to give an evaluation of the manual. Both detailed and general comments are welcome.

These forms, together with all types of inquiry and requests for documentation should be sent to the local ND office or (in Norway) to:

Documentation Department
Norsk Data A.S
P.O. Box 4, Lindeberg gård
Oslo 10

TO THE READER

The NORD-100 Microprogramming Description manual describes how to microprogram the NORD-100 in order to take full advantage of the writable control store feature. This manual should be of interest to all technical personnel writing their own microprogram. It will also be useful for those who want a more detailed understanding of the microprogrammed structure of the NORD-100 CPU.

PREREQUISITE KNOWLEDGE

A general knowledge of the NORD-100 computer system, together with some basic understanding of the NORD-100 CPU and some experience in digital techniques is recommended.

THE MANUAL

Chapter 1 is a short introduction to the writable control store option, while Chapter 2 gives an introduction to the CPU architecture. The microprogram word format is described in Chapter 3 and how to write the microprogram is found in Chapter 4. Chapter 5 shows the usage of the writable control store.

In the appendixes are some data sheets for the main building blocks in the CPU, together with the microprogram flow charts. The data sheets are taken from "AM2900 Bipolar Microprocessor Family" issued by Advanced Micro Devices, Inc. More information about the circuits will be found in this publication. Data sheets for the microprogram sequencer are found in "The TTL Data Book" issued by Texas Instruments.

TABLE OF CONTENTS

+ + +

<i>Section:</i>		<i>Page:</i>
1	INTRODUCTION	1-1
2	CPU ARCHITECTURE INTRODUCTION	2-1
3	MICROPROGRAM WORD FORMAT	3-1
4	WRITING THE MICROPROGRAM	4-1
5	WRITABLE CONTROL STORE USAGE	5-1
 <i>Appendix:</i>		
A	MNEMONIC LIST	A-1
B	DATA FOR AM 2901 BIT SLICE	B-1
C	DATA FOR AM 2914 INTERRUPT CONTROLLER	C-1
D	FUNCTIONAL BLOCK DIAGRAM FOR 74S482 SEQUENCER	D-1
E	MICROPROGRAM FLOW CHARTS	E-1
F	MAIN MEMORY/WCS CORRESPONDENCE	F-1

1 INTRODUCTION

The NORD-100 CPU with the Writable Control Store option, has its basic microprogram contained in the lower 2K of the microprogram addressing area (addresses 0 - 3777). The Writable Control Store (WCS) contains the addresses 4000 - 4377.

In order to write micro code for the WCS, the following steps must be undertaken:

1. Specification of the new machine instruction that will be implemented by the new microprogram.
2. Design and coding of the necessary microinstructions to perform the machine instructions.
3. Assembly of the symbolic microprogram into binary format with the NORD-100 WCS assembler.
4. Place the binary representation of the WCS contents in main memory and load it into WCS with the load writable control store machine instruction.
5. Use the Customer Specified Instructions to execute the microprogram now residing in the WCS.

2 CPU ARCHITECTURE INTRODUCTION

The CPU consists of elements built around the Internal Data Bus, IDB. IDB is shown hatched in Figure 2.1. A few comments to each of the elements in Figure 2.1 follows.

ALU performs all arithmetic and logic functions. Two 4 bit operand pointers, the A and B operands, select sources for operations to be performed. Nine ALU function bits select operations within the ALU. The data input and output ports are directly connected to IDB.

SHIFT LINKAGE controls the shift mode as well as the length of the shift operands.

STS 0 - 7 contains the lower byte of the STS register.

FLOATING ARITHMETIC CONTROL performs the necessary functions to speed multiply and divide operations.

GPR is a General Purpose Register used by the microprogram for various purposes. It will always be shifted the same way as an ALU register if shift is specified in the ALU control. GPR is especially valuable during multiply and divide operations. In multiply operations it holds one of the operands and in divide operations it gets the result shifted in from the right.

MMR is the communicating register against the memory management module.

BUS CONTROL controls the access to the system bus, which links the CPU and peripheral interfaces, memory modules, etc.

PES and *PEA* keeps information about system bus abnormalities.

WDA and *DBR* keeps data going between IDB and the system bus.

μPROGRAM CONTROL STORE holds the microprogram controlling the CPU.

μPROGRAM SEQUENCER generates the address of the next microinstruction to be executed. This sequencer can get the address from the microprogram (jump), from the current address + 1 (continue) or from an internal stack, which is 4 deep, and can be pushed or popped.

SEQUENCER CONTROL and *CONDITION SELECT* controls conditional branching in the microprogram sequence. A code indicating a test object is outputted to these circuits from the microprogram, together with a code indicating what to do if a later test gives a false result. When a test is enabled at a later state, the result of the former ALU operation is tested and may result in conditional branching.

VECTOR SELECT controls the source of the 4 lower address bits in a vectorized jump sequencing microinstruction. The 8 uppermost address bits are specified in the microinstruction (in the microinstruction the 4 lower bits must be 0). In this manner, a 16 way branch is effectively implemented.

MAP contains the entry points of all machine instructions. It is used to generate entry point addresses whenever a machine instruction is to be executed.

PIC contains the interrupt system from level 10 and upwards.

MPIDS can set interrupt bits in the PIC.

IR contains the 10 least significant bits of a machine instruction.

STS 8 - 15 contains the upper byte in the STS register.

LOOP COUNTER is a 6 bit register with the upper bit as sign bit. It has an auto increment/decrement facility which makes it count towards zero when counting. Zero inhibits further counting. The loop counter is used to repeat one (or a few) microinstructions a predetermined number of times, or to count, for example, the number of shifts needed to normalize a floating number.

BMG (Bit Mask Generator) generates a 1 bit among zeros. The position of the 1 bit is controlled by the A operand.

REG. FILE contains the register blocks of the interrupt levels not presently active. It also contains scratch registers for MOPC and for the interrupt system. It is addressed using the A operand to indicate the level, and the B operand to indicate the register number within that level.

Bits 55 - 63, together with bits 44 - 45, controls the behaviour of the ALU. The ALU is built of the bit slice component 2901, and the functional description of 2901 is included as Appendix B.

The effects of bits 55 - 63 are explained in Appendix B, with some additional information in the following pages.

Bits 44 and 45 modifies bits 55 - 63 before they reach the ALU.

ALUM0 = 0 (bit 44)
ALUM1 = 0 (bit 45): Micro Controlled Mode

The ALU behaviour is controlled entirely from the microprogram.

ALUM0 = 1 (bit 44)
ALUM1 = 0 (bit 45): Multiply Mode

The ALU behaviour is controlled from the microprogram, except that GPR bit 0 controls the addition of the A operand register or 0. This is only used in multiplication instructions.

ALUM0 = 0 (bit 44)
ALUM1 = 1 (bit 45): Divide Mode

The ALU behaviour is controlled from the microprogram except that GPR bit 0 controls whether or not the A operand register should be added or subtracted. This is only used in divide instructions.

ALUM0 = 1 (bit 44)
ALUM1 = 1 (bit 45): Shift Instruction (machine instruction is shift)

The shift direction and shift mode is controlled from the loop counter and the bits 9 - 10 of the machine instruction. The extra microinstruction executed after the loop counter reaches 0 will not shift the ALU result.

Bits 53 - 54 controls the behaviour of the 8 lower bits of the STS register.

- | | |
|----|---|
| 00 | will not affect the STS register |
| 01 | will update the C, O and Q flip-flops to the result of the current ALU instruction |
| 10 | will update the M flip-flop to the disappearing bit in a shift microinstruction. If there is ALUM IRSHIFT, M will be updated regardless of the setting of bits 53 - 54. |
| 11 | will load the 8 lower bits of the STS register from IDB |

Bits 50 - 52 controls the source of the A operand to the ALU.

- 000 The A operand is bits 12 - 15 of the microinstruction.
- 010 The A operand is the PIL register (used to address the current level register block in the register file).
- 100 The A operand is bits 3 - 6 of the current machine instruction. Used in bit operation instructions and in instructions specifying a level in the register file.
- 101 The A operand is bits 3 - 5 of the current machine instruction. This is used to address the source register in machine instructions.
- 110 The A operand is bits 0 - 3 of the loop counter.

Bits 48 - 49 controls the source of the B operand to the CPU.

- 00 The B operand is bits 16 - 19 of the microinstruction.
- 01 The B operand is bits 0 - 2 of the current machine instruction. Used to address the destination register in machine instructions.
- 10 The B operand is bits 3 - 5 of the current machine instruction. Used to address the source register in machine instructions.
- 11 The B operand is bits 0 - 3 of the loop counter. This is used to read the loop counter by using BARG as IDB source.

Bits 46 - 47 control the carry in line to the ALU.

- 00 Carry is 0
- 01 Carry is 1
- 10 Carry is the C flip-flop of the STS register
- 11 Carry is the GPR bit 0 (used in division)

Bits 42 - 43 control the shift type in shift microinstructions, if the machine instruction is not a shift instruction. They indicate which page table to use together with memory requests.

- 00 Arithmetic shift/ Fetch memory request
- 01 Rotational shift/ Request through PT
- 10 Zero-end-input shift/ Request through A PT
- 11 Link-end-input shift/ Request through the page table last specified

Bits 37 - 41 control which source should control the content of IDB (refer to Figure 2.1).

0	ALU	enables its result onto IDB
1	BMG	a 1 bit among zeros is enabled onto IDB. The position of the 1 bit is controlled by the A operand.
2	GPR	the general purpose register is enabled onto IDB.
22	GPR	with bits 8 - 15 equal to bit 7 is enabled onto IDB. This makes sign extension easy during address calculations and argument instructions, etc.
3	DBR	the system bus data read register is enabled onto IDB. This must be used with the WAIT for data ready timing specification.
4	ARG	The 16 lower bits of the microinstruction is enabled onto IDB. This makes it easy to generate 16 bit arguments within the microprogram.
5	REG	The register file address selected by the A and B operands is enabled onto IDB.
6	STS	The 16 bit STS register is enabled onto IDB.
7	MMR	The memory management read register is enabled onto IDB.
10	BARG	The B operand is regarded as an argument and enabled onto IDB. In this way arguments between 0 - 17 may be generated.
11	SWAP	A byte swap circuit is enabled onto IDB. The two bytes present on IDB during the last cycle have exchanged positions.
12	PEA	The inverse of the PEA register is enabled onto IDB.
13	PES	The inverse of the PES register is enabled onto IDB.
14	AARG	The A operand is regarded as an argument shifted 3 bits left, and enabled onto IDB. In this way arguments between 0 - 170 with 0 in the 3 lower bits may be generated.
15	PICS	The internal status of the PIC is enabled onto IDB. The RSTS command must be issued to the PIC simultaneously.
16	IOR	The register containing UART data and status is enabled onto IDB.
17	NONE	IDB is disabled. Read commands to the PIC may cause PIC to enable internal PIC registers onto IDB.

Bits 32 - 36 specifies a command to be performed by the microinstructions. The commands are:

0	NONE	No command executed.
1	LDPIL	The upper byte of the STS register is loaded with the upper byte of IDB. The new PIL will not be available as A operand before in the microinstruction following the next microinstruction.
2	LDGPR	IDB is loaded into GPR. If a shift is specified in the same microinstruction, GPR will not be shifted, only loaded.
3	EWRF	IDB is written into the register file word specified by the A and B operands.
4	CLIRQ	Inhibits the ability of interrupts to modify the microinstruction sequence at the time of a mapped jump. Used to avoid interrupts when the MAPJ bit is used at other times than FETCH, CLIRQ will also block PANEL interrupts from being included in the IRQ condition.
5	RSDA	Reset the data available signal from the UART on the CPU board.
6	TBSTR	Transmit Buffer Strobe. Transfers the lower IDB byte to the UART output part.
7	SIOC	Set I/O control. Transfers IDB to control several functions. The bits of IDB have these functions. Bit 0: PIN (prepare interrupt) signal for clock. Bit 1: PIN signal for terminal input. Bit 2: PIN signal for terminal output. Bit 3: RFT (ready for transfer) signal for clock. Bit 4: Green LED on CPU (OK). Bit 5: Red LED on CPU (ERROR). Gives master clear on system bus. Bit 6: When = 1, inhibits MOPC input when in continue. Bit 7: Resets the counters giving 50 Hz clock frequency. Used by IOX 11. Bit 10: Not used. Bit 11-12: 0 0 5 bits UART character length 0 1 6 bits UART character length 1 0 7 bits UART character length 1 1 8 bits UART character length Bit 13: When 0 — 1 stop bit; When 1 — 2 (1.5 at 5 bits) stop bits. Bit 14: An extra bit is added to give even parity. The latest IDB value outputted by a SIOC command will always be saved in a word named STATUS in the register file.
10	ESRP	A special flip-flop, the R = P flip-flop, is set to 1 if the ALU result is 0. Used to detect that the next location (where prefetch has already been performed) has been written into (STA * + 1).

11	EPIC	indicates that A operand should be a command to the PIC. The specifications of the PIC are included in Appendix C. If the PIC command is a read command, other than read status, the PIC will enable the register onto IDB regardless of the microprogram bits 37 - 41.
12	SMPID	sets bits in the micro PID register in the PIC. Bits can only be set to 1 by this command. (The PIC command MCLPID can reset bits to 0.)
13	START	resets the STOP flip-flop. The CPU will start executing a main memory program.
14	STOP	sets the STOP flip-flop. Whenever a FETCH has been performed, the microprogram is forced to microaddress 3760 or 3770 by the interrupt hardware. The stop routine is executed.
15	CLRTI	resets the clock flip-flop. Every 20 ms the microprogram is forced to microaddress 3761 or 3771, thereby entering the clock routine. This routine sends out the CLRTI command.
16	CLFF	clears flip-flops having special functions regarding the floating point rounding indicator (TG). The DZD (double zero detect used in FDV) and the OOD (one out detect used in FAD, FSB and FMU) flip-flops are cleared by CLFF.
17	LDLC	loads the loop counter with the 6 lower bits of IDB. The modified loop counter will not be available as A operand before in the microinstruction following the next microinstruction.
20	LDIRNO	select register on memory management module to be accessed later. IDB is transferred to the selector on MMM.
21	LDIR	(or XMM in symbolic language) will transfer IDB to the MMM register selected by a previous LDIRNO command. (To read a selected MMM register, only the IDB source designation MMR should be used.)
30	EXRQ or IRDRQ	differ according to the contents of bits 42 - 43: 00 EXRQ, examine memory request. The physical address must be on IDB. The segment number must already be loaded into the MMM segment register. 01 IRDRQ, PT, indirect address read request with mapping through the normal page table. 10 IRDRQ, APT, indirect address read request with mapping through the alternative page table. 11 IRDRQ, HOLD, indirect address read request with mapping through the last used page table.

31	DERQ	deposit memory request. The physical address must be on IDB, in the segment already loaded into the MMM segment register.
34	RDRQ	mapped read request has four variants, depending on bits 42 - 43: 00 Fetch request 01 Read request mapped through the normal page table. 10 Read request mapped through the alternative page table. 11 Read request mapped through the last used page table.
35	WRRQ	mapped write request has 3 variants, depending on bits 42 - 43: 01 Write request mapped through the normal page table 10 Write request mapped through the alternative page table 11 Write request mapped through the last used page table
36	IDENT	IDENT bus request. As with write requests, the write data must be enabled onto IDB in the next microinstruction, accompanied by a WAIT cycle. The following instruction must also be a WAIT cycle, with DBR as source to IDB.
37	IOX	IOX bus request. Data out and in will be present during the next 2 microinstructions. The timing, as seen from the microprogram, will be identical with IDENT.

Bits 28 - 31 control the microprogram sequencer. The sequencer is documented in Appendix D. Bits 30 - 31 control the next instruction selection and bits 28 - 29 control the internal stack. These bits will only be in control as long as bits 22 - 23 are high.

Bit 27 is active low, and will load IR from IDB in the next instruction. It will clear the R = P flip-flop. It sets the false sequence control to JUMP and POP, and the text object to DR = 2 (used by ROP and BOP instructions to enable test of the P register as destination). If the next microinstruction is a JMP, the address will be taken from the mapping PROM, or be generated by the interrupt hardware if CLIRQ command has not been given.

Bit 26 is active low, and will start the loading of the WCS from main memory. The next microinstruction must be a jump to microaddress 7400. Thereafter, the CPU hardware goes into a special state and moves the memory addresses 36000 - 37777 into WCS. The microprogram continues from microaddresses 0 when the loading of WCS is finished.

Bit 25 is active low and specifies vectorized jump. The 8 upper bits of the jump microaddress are normally taken from micro word bits 4 - 11. The lower 4 bits can be selected from 4 sources, controlled by bits 42 - 43.

- 00 IR 0 - 3 controls the vector, used in TRA and TRR.
- 01 IR 8 - 10 controls the vector, used for address calculation in memory reference instructions.
- 10 ALD 0 - 3, the ALD setting controls the vector.
- 11 The A operand bits control the vector.

Bit 24 is active low and sets condition to be tested on a later occasion. A 4 bit code indicating test condition and a 4 bit code indicating the sequence instructions in case of a false test, are outputted to a hold register.

Bit 23 is active low and enables testing (against the previously set test condition) of the result of the micro instruction before the micro instruction containing this test. If the test is true, the sequencer executes the bits 28 - 31 to find the next microinstruction. If the test is false, the sequencer executes the false sequence instruction previously outputted to the hold register by bit 24.

Bit 22 is active low and increments or decrements the loop counter (always towards 0). If the selected condition is false, RETURN, HOLD is executed by the sequencer. If it is true, bits 28 - 31 are executed. The new value of the loop counter will not be available as A operand or test object before in the microinstruction following the next microinstruction.

Bits 20 - 21 controls the duration of each microinstruction. It also controls the synchronization between the CPU and the system bus control.

- 00 specifies fast instruction, approx. 150 ns
- 01 specifies intermediate instruction, approx. 170 ns
- 10 specifies slow instruction, approx. 190 ns
- 11 specifies wait for bus control synchronization

After a read memory request, one WAIT cycle must occur to finish the bus request. If the result is needed in the CPU, WAIT must be accompanied by DBR as IDB source.

After a write memory request, the data to write must be enabled onto IDB in the next microinstruction, accompanied with a WAIT cycle. Thereafter, another WAIT cycle must be encountered before a new request is sent out.

As long as the above rules are obeyed, WAIT cycles may be used freely, with no effect.

In NORD-100 all microinstructions are either "slow" or "wait".

In NORD-100, with fast option, most microinstructions are "fast", while the same are "wait" as in the ordinary NORD-100. A few microinstructions have to be "slow" because of component timing characteristics. This applies to:

1. All microinstructions handling the UART (SIOC, RSDA, TBSTR).
2. Microinstructions where the IDB source is used by the ALU as one of the operands in a arithmetic operation, and the result will be tested by the F=0 condition in the next microinstruction.
3. All microinstructions that involve data exchange with the memory management module (microinstructions containing IDBS, MMR or COMM, LDIR or COMM, XMM).

Bits 16 - 19 control the B operand as long as bits 48 - 49 are zero.

Bits 12 - 15 control the A operand as long as bits 50 - 52 are zero. The A operand controls BMG and acts as instruction to the PIC (see Appendix C).

Bits 0 - 11 contains the jump address in jump microinstructions. The jump addresses may also be generated by the mapping PROM or the interrupt hardware, and partly by the vectoized jump mechanism. When these sources generate jump addresses, the corresponding microword bits must be 0.

Bits 0 - 3 selects the false sequence to be outputted by bit 24 to the hold mechanism. It specifies the sequencer instruction in case a later test (bit 23) gives a false result.

Bits 4 - 7 selects the test object to be outputted by bit 24 to the hold mechanism. If a test object changes state, one microinstruction will be executed before the change is detected. The cause of this is the pipelining of ALU and sequencing calculations.

The 16 test objects are:

0	DR = 2	destination register is P. Used to discard prefetch if P is destination in ROP and BOP.
1	LCZ	loop counter is 0. Used to go through a loop a predetermined number of times.
2	IRQ	used to test the interrupt request flip-flop, to see if interrupts are pending. If the test is preceded by the CLIRQ command, only interrupts from AM 2914 are tested and not panel interrupts such as STOP, RTC, etc.
3	RESTR	used to test if the current program is running in restricted mode. Instructions that are privileged should not execute when this condition is true. Privileged WCS instructions must test this condition, other privileged instructions are mapped different whether RESTR is on or off.
4	FETCH	indicates whether the last memory access was a fetch of new instruction. Used to generate correct P in case of page fault.
5	OOD	the one out detect flip-flop is used to set TG correctly in FAD, FSB and FMU instructions.
6	DZD	the double zero detect flip-flop is used to set TG correctly in FDV instructions.
7	COND	This condition will latch the true/false result of the last microinstruction so that it can be tested later if that is more convenient than testing it now. It will postpone the test as long as desired.
8	R = P or ZF	Used to detect if single or multiple store instruction has affected * + 1. If true, the prefetched instruction must be discarded.

9	OPCTT1	is used to test whether device number 300 - 307 should use the MOPC terminal. If a strap on the CPU card is removed, this line will never be true. If the strap is intact, the line will be true when device numbers 300 - 307 are encountered, indicating that this device is not on the system bus but directly micro-program controlled.
10	OVF	overflow from ALU. Indicates overflow during arithmetic operations.
11	CRY	carry out from ALU.
12	F11	bit 11 from ALU.
13	F15	bit 15 (sign bit) from ALU
14	ZF	Result = 0 from ALU.
15	STP	indicates that the STOP flip-flop is 1.

Bits 0 - 15 will be enabled onto IDB as an argument if ARG is IDB source.

4 WRITING THE MICROPROGRAM

Assembly Language Considerations

At this point, it is necessary to go carefully through the mnemonic list given in Appendix A and compare the octal values corresponding to the mnemonics with Figure 2.1. This gives an understanding of which mnemonic performs what function in the microprogram. It is important to be aware that bits 22 - 27 are inverted after the word is assembled.

All the mnemonics in the list are translated by the NORD-100 WCS assembler to its corresponding octal value. All the octal values of the mnemonics in one microinstruction are "OR"ed together. If an octal number is encountered, it is placed in bits 0 - 15 and regarded as an argument. If a symbolic name is encountered which is not found in the mnemonic list, it is regarded as a label, which is or will be defined. When there is no more input to the WCS assembler, all labels are replaced by their values, which are filled into bits 0 - 11 in the correct microaddresses. The characters accepted by the NORD-100 WCS assembler are "space" and "tab". These are used as delimiters between mnemonics, labels and numbers. "%" is used to indicate that the rest of the line is a comment. Any other characters are used to form character groups, which may be mnemonics, labels and commands. If only numbers between 0 - 7 are found in a character group, it is an octal number.

If "/" is found in a character group, the current location counter (CLC) is updated to the octal number in front of /.

If ":" is found in a character group, the rest of the character group is regarded as a label.

The character ";" terminates each microinstruction and CLC is then incremented. Each microinstruction may contain an unspecified number of lines.

Microprogram Architecture Considerations

Whenever the microprogram jumps to execute one machine instruction, the jump address is taken from the mapping PROM. The addresses for the customer specified instructions are found in WCS in the addresses 4001 to 4015 (refer to the NORD-100 Reference manual). After WCS has been loaded, execution of one of the customer specified instructions will cause the microprogram to branch to one of the microaddresses 4001 - 4015. When this point is entered, some of the CPU scratch registers have defined value:

1. GPR contains the machine instruction.
2. False sequence is JMP and POP.
3. R = P flip-flop is cleared.
4. Test object is DR = 2.
5. IR contains the lower 11 bits of the machine instruction.
6. A FETCH request is sent out for the next machine instruction.
7. The P register points to the next machine instruction.
8. The top word of the sequencer stack contains the microaddress PMODX.
9. The next to the top word of the sequencer stack contains the microaddress USDBR.

When the microinstructions in WCS are finished doing their task, return to the PROM resident part of the microprogram should take place in an orderly manner. There are five natural return points:

USDBR:

The system bus must not have been used by the WCS instructions. No WAIT cycle or DBR reading must have been done. The fetch request sent along before the jump to WCS will give the next machine instruction. The sequence stack must be popped once, to bring USDBR to the top word.

USGPR:

The WCS instructions must have read the DBR onto IDB, with an accompanying WAIT cycle, and saved it in GPR. Thereafter the WCS routines may have used the system bus for other purposes. The next machine instruction to execute will be the one requested before the WCS instruction was entered. The sequencer stack must be popped once to bring USDBR to the top word.

INIT:

The WCS instructions must have had a WAIT cycle to end the fetch bus cycle that was set up before WCS was entered. The prefetched instruction will not be used and INIT will send out a new fetch request. This return from a WCS instruction should be used after jump instructions. No special contents of the sequencer stack are necessary.

PMODX:

The WCS instruction need not have had a WAIT cycle. Except for that, PMODX is equal to INIT. One can easily return to PMODX by using the top word of the stack as return address.

STEND:

This return address should be used from instructions storing words in main memory, if one wishes to use the prefetched word if that is not modified. The WCS instruction must have read the DBR onto IDB with an accompanying WAIT cycle, and saved it in GPR. The ALU register R1 must contain the address written into. If multiple stores are performed by the WCS routine, a microprogram subroutine, WRRQ, can be used to send out new requests and to handle the R = P flip-flop (see STF as an example). If * + 1 is modified, a new fetch is performed, if it is not the prefetched instruction saved in GPR is used. The sequence stack must be popped once before the jump to STEND to bring USDBR to the top word.

Note that it is always safest to end a machine instruction with a jump to PMODX but that it will save memory accesses and time if one of the other return points is used. To find examples, the reader is advised to study the microprogram listing and the flow charts in Appendix E carefully.

Example of a machine instruction "set argument to register":

1	1 0 0	0 0 0	0 1 1	x x x	x x x
1	4	0	3		DR

The argument (0 - 7) is moved into the destination register, which can be any of the registers D, P, B, L, A, T, or X. Since the code for the instruction is 1403xx, which is CUST2, its entry address is 4001. The code for the machine instruction is two microwords, including the test for P as destination. If P is the destination, the next instruction must be fetched again and the instruction returns to PMODX. If P is not the destination, return is to USDBR. The stack is then popped. The displacement is found by using the B operand equal to IR bits 3 - 5 and using BARG as IDB source.

4001/ B, SRCE IDBS, BARG 4016;	ALUF, PASSD T, JMP	ALUD, Q T, HOLD	CYCL, FAST
4016/ B, DEST IDBS, ALU USDBR	ALUF, PASSQ T, RETURN CONDENABL;	ALUD, B T, HOLD	

5 WRITABLE CONTROL STORE USAGE

The symbolic microprogram will most easily be prepared using the editor QED, with SINTRAN III. It should be saved on an ordinary file.

Thereafter, the NORD-100 WCS assembler is loaded, by typing N-100-WCS-ASSEMB to SINTRAN III. The assembly process is started by typing DO 10 to the assembler. The user is asked to give the names of the input file, list file and object output file. The assembly goes on until the input file is exhausted, and an image of the WCS is then saved on the object file. This file will contain 2066 bytes, of which the first 16 and the last 2 are header and trailers, and not WCS microprogram code. The remaining 2048 bytes are the microprogram to be loaded into WCS.

In order to load this microprogram into WCS, the bytes must be placed in physical addresses 36000 to 37777. If the 2048 bytes on the files are numbered from 1 to 2048, the correspondence between main memory addresses and bytes is as follows:

36000	7	8
36001	5	6
36002	3	4
36003	1	2
36004	15	16
36005	13	14
36006	11	12
36007	9	10
37774	2047	2048
37775	2045	2046
37776	2043	2044
37777	2041	2042

The correspondence between main memory and WCS is indicated in Appendix F.

When all the 2048 bytes have been placed according to this scheme, the machine instruction 143500 (LWCS) must be executed. Thereafter, customer specified instructions may be executed. If they are executed before LWCS is performed, a ROM-out-of-range internal interrupt will be generated, and the WCS will not be used. After power up, or after pushing the Master Clear button of the computer, WCS will be unaccessible and LWCS must be executed in order to allow customer specified instructions to be executed.

APPENDIX A
MNEMONIC LIST

61 B,R5 000000 000000 000015 000000 B-OPERAND IS 15. USED TO ADDRESS REGISTER 15, CONTAINING SCRATCH
62 B,R6 000000 000000 000016 000000 B-OPERAND IS 16. USED TO ADDRESS REGISTER 16, CONTAINING SCRATCH
63 B,R7 000000 000000 000017 000000 B-OPERAND IS 17. USED TO ADDRESS REGISTER 17, CONTAINING SCRATCH
64 B,DEST 000001 000000 000000 000000 B-OPERAND IS THE VALUE OF 'INSTRUCTION BITS 0-2' LOADED BY LAST 'T,MAPJ'
65 B,SRC 000002 000000 000000 000000 IS THE VALUE OF 'INSTRUCTION BITS 3-5' LOADED BY LAST 'T,MAPJ'
66 B,LC 000003 000000 000000 000000 B-OPERAND IS THE VALUE IF THE 'LOOP COUNTER' TWO INSTRUCTIONS AGO
67 B,0 000000 000000 000000 000000 B-OPERAND IS 0
68 B,1 000000 000000 000001 000000 B-OPERAND IS 1
69 B,2 000000 000000 000002 000000 B-OPERAND IS 2
70 B,3 000000 000000 000003 000000 B-OPERAND IS 3
71 B,4 000000 000000 000004 000000 B-OPERAND IS 4
72 B,5 000000 000000 000005 000000 B-OPERAND IS 5
73 B,6 000000 000000 000006 000000 B-OPERAND IS 6
74 B,7 000000 000000 000007 000000 B-OPERAND IS 7
75 B,10 000000 000000 000010 000000 B-OPERAND IS 10
76 B,11 000000 000000 000011 000000 B-OPERAND IS 11
77 B,12 000000 000000 000012 000000 B-OPERAND IS 12
78 B,13 000000 000000 000013 000000 B-OPERAND IS 13
79 B,14 000000 000000 000014 000000 B-OPERAND IS 14
80 B,15 000000 000000 000015 000000 B-OPERAND IS 15
81 B,16 000000 000000 000016 000000 B-OPERAND IS 16
82 B,17 000000 000000 000017 000000 B-OPERAND IS 17
83 ALUF,ANDAQ 010000 000000 000000 000000 A ^ Q -> F
84 ALUF,ANDAB 010200 000000 000000 000000 A ^ B -> F
85 ALUF,ANDDA 011200 000000 000000 000000 D ^ A -> F
86 ALUF,ANDDQ 011400 000000 000000 000000 D ^ Q -> F
87 ALUF,ORAQ 006000 000000 000000 000000 A V Q -> F
88 ALUF,ORAB 006200 000000 000000 000000 A V B -> F
89 ALUF,ORDA 007200 000000 000000 000000 D V A -> F
90 ALUF,ORDQ 007400 000000 000000 000000 D V Q -> F
91 ALUF,XORAB 014000 000000 000000 000000 A XOR B -> F
92 ALUF,XORAB 014200 000000 000000 000000 A XOR B -> F
93 ALUF,XORDA 015200 000000 000000 000000 D XOR A -> F
94 ALUF,XORDQ 015400 000000 000000 000000 D XOR Q -> F
95 ALUF,XNORAB 016000 000000 000000 000000 NOT(A XOR B) -> F
96 ALUF,XNORAB 016200 000000 000000 000000 NOT(A XOR B) -> F
97 ALUF,XNORDA 017200 000000 000000 000000 NOT(D XOR A) -> F
98 ALUF,XNORDQ 017400 000000 000000 000000 NOT(D XOR Q) -> F
99 ALUF,ZERO 010400 000000 000000 000000 0 -> F
100 ALUF,INVQ 016400 000000 000000 000000 NOT(Q) -> F
101 ALUF,INVB 016600 000000 000000 000000 NOT(B) -> F
102 ALUF,INVA 017000 000000 000000 000000 NOT(A) -> F
103 ALUF,INVD 017600 000000 000000 000000 NOT(D) -> F
104 ALUF,PASSQ 014400 000000 000000 000000 Q -> F
105 ALUF,PASSB 014600 000000 000000 000000 B -> F
106 ALUF,PASSA 015000 000000 000000 000000 A -> F
107 ALUF,PASSD 015600 000000 000000 000000 D -> F
108 ALUF,MASKAQ 012000 000000 000000 000000 NOT(A) ^ Q -> F
109 ALUF,MASKAB 012200 000000 000000 000000 NOT(A) ^ B -> F
110 ALUF,MASKDA 013200 000000 000000 000000 NOT(D) ^ A -> F
111 ALUF,MASKDQ 013400 000000 000000 000000 NOT(D) ^ Q -> F
112 ALUF,A+Q 000000 000000 000000 000000 A + Q -> F
113 ALUF,A+B 000200 000000 000000 000000 A + B -> F
114 ALUF,D+A 001200 000000 000000 000000 D + A -> F
115 ALUF,D+Q 001400 000000 000000 000000 D + Q -> F
116 ALUF,A+Q+1 000000 040000 000000 000000 A + Q + 1 -> F
117 ALUF,A+B+1 000200 040000 000000 000000 A + B + 1 -> F
118 ALUF,D+A+1 001200 040000 000000 000000 D + A + 1 -> F
119 ALUF,D+Q+1 001400 040000 000000 000000 D + Q + 1 -> F
120 ALUF,-Q-1 004400 000000 000000 000000 -Q - 1 -> F


```

181 CYCL,NORM 000000 000000 000020 000000 SPECIFIES 'NORMAL' MICRO INSTRUCTION CYCLE
182 CYCL,SLOW 000000 000000 000040 000000 SPECIFIES 'SLOW' MICRO INSTRUCTION CYCLE
183 CYCL,WAIT 000000 000000 000060 000000 SPECIFIES 'WAIT-FOR-BUS-SYNCHRONIZATION' MICRO CYCLE
184 LGOUNT 000000 000000 000100 000000 COUNT LOOP-COUNTER ; IF FALSE 'RETURN' & 'HOLD' ; IF TRUE USE TRUE SPECIFICATIONS
185 GMDENABL 000000 000000 000200 000000 ENABLE CONDITIONAL SEQUENCING, USE 'FALSE' SPECS IF CONDITION FALSE
186 SLWCS 000000 000000 000200 000000 START LOAD WRITABLE CONTROL STORE, AVOID 'ROM-OUT-OF-RANGE'
187 IDBS,ALU 000000 000000 000200 000000 ARITHMETIC-LOGIC-UNIT -> IDB
188 IDBS,EMG 000000 000000 000300 000000 BIT-MASK-GENERATOR -> IDB
189 IDBS,GPR 000000 000100 000000 000000 GENERAL-PURPOSE-REGISTER -> IDB
190 IDBS,DIR 000000 000140 000000 000000 DATA-BUS-REGISTER -> IDB
191 IDBS,ARG 000000 000200 000000 000000 ARGUMENT (MICRO-INSTRUCTION-BITS 0 - 15) -> IDB
192 IDBS,REG 000000 000240 000000 000000 REGISTER-FILE -> IDB
193 IDBS,STS 000000 000300 000000 000000 STATUS -> IDB
194 IDBS,MMR 000000 000340 000000 000000 MEMORY-MANAGEMENT-REGISTER -> IDB
195 IDBS,BARG 000000 000400 000000 000000 B-OPERAND-ARGUMENT (0-17) -> IDB
196 IDBS,SWAP 000000 000440 000000 000000 BYTE-SWAP OF LAST IDB -> IDB
197 IDBS,FEA 000000 000500 000000 000000 PEA-REGISTER -> IDB
198 IDBS,PES 000000 000540 000000 000000 PES-REGISTER -> IDB
199 IDBS,AAARG 000000 000600 000000 000000 A-OPERAND-ARGUMENT*10 (0-170) -> IDB
200 IDBS,PTC 000000 000640 000000 000000 PRIORITY-INTERRUPT-CONTROL STATUS REGISTER BUS -> IDB
201 IDBS,IOR 000000 000700 000000 000000 UART-(UNIVERSAL ASYNCHRONOUS RECEIVE TRANSMIT) DATA AND STATUS -> IDB
202 IDBS,DSABL 000000 000740 000000 000000 DISABLE SOURCES TO IDB (USED TO READ 'PIC'-INFO, EXCEPT PIC-STATUS)
203 IDBS,GPR_SEXT 000000 001100 000000 000000 GPR-BITS 0-7 (WITH BITS 8-15 EQUAL TO BIT 7) -> IDB
204 COM4,ESRP 000000 000010 000000 000000 SET 'R'-P'-FLIP-FLOP IF 'F'-0' IS TRUE
205 COM4,EPIC 000000 000011 000000 000000 A-OPERAND IS AN INSTRUCTION TO 'PIC' (PRIORITY INTERRUPT CONTROLLER)
206 COM4,SMPID 000000 000012 000000 000000 SET MICRO-PID. PID-POSITIONS WHERE IDB HAS A '1' IS FORCED HIGH
207 COM4,START 000000 000013 000000 000000 RESET THE 'STOP'-FLIP-FLOP
208 COM4,SSTOP 000000 000014 000000 000000 SET THE 'STOP'-FLIP-FLOP
209 COM4,CLRTC 000000 000015 000000 000000 CLEAR THE 20 MS CLOCK FLIP-FLOP
210 COM4,CLFF 000000 000016 000000 000000 CLEAR THE 'LOOP' - AND THE 'DZD'-FLIP-FLOP
211 COM4,LDLC 000000 000017 000000 000000 LOAD THE 'LOOP COUNTER' WITH THE 6 LOWER IDB-BITS
212 COM4,LDPL 000000 000001 000000 000000 IDB-BITS 8-15 -> STATUS-BITS 8-15
213 COM4,LDGPR 000000 000002 000000 000000 IDB -> GENERAL-PURPOSE-REGISTER
214 COM4,EWRF 000000 000003 000000 000000 IDB -> REGISTER FILE WORD ADDRESSED BY A-OPERAND AND B-OPERAND
215 COM4,CLIRQ 000000 000004 000000 000000 PREVENT JUMP TO THE INTERRUPT VECTOR, REMOVE 'PANEL'-EFFECT ON 'IRQ'-TEST
216 COM4,RSDA 000000 000005 000000 000000 RESET DATA AVAILABLE IN 'UART'
217 COM4,IBSTR 000000 000006 000000 000000 'TRANSIT BUFFER STROBE' TO 'UART'. LOWER IDB-BITS LOADED INTO 'UART'
218 COM4,SIOC 000000 000007 000000 000000 LOAD 'I/O-CONTROL' REGISTER, CONTROLLING TERMINAL 1 AND CLOCK INTERRUPTS ETC.
219 COM4,RDRQ 000000 000034 000000 000000 READ MEMORY REQUEST. ADDRESS PRESENT ON IDB
220 COM4,WRRQ 000000 000035 000000 000000 WRITE MEMORY REQUEST. ADDRESS PRESENT ON IDB
221 COM4,IDENT 000000 000036 000000 000000 START 'IDENT'-CYCLE ON BUS. ADDRESS PRESENT ON IDB
222 COM4,IOX 000000 000037 000000 000000 START 'IOX'-CYCLE ON BUS. ADDRESS PRESENT ON BUS
223 COM4,IRDRQ 000000 000030 000000 000000 INDIRECT ADDRESS MEMORY REQUEST. ADDRESS ON IDB
224 COM4,DERQ 000000 000031 000000 000000 DEPOSIT MEMORY REQUEST. ADDRESS ON IDB
225 COM4,EXRQ 000000 000030 000000 000000 EXAMINE MEMORY REQUEST. ADDRESS ON IDB
226 COM4,LDIR 000000 000020 000000 000000 LOAD INT. REG. NO. INTO MMM
227 COM4,XMM 000000 000021 000000 000000 READ/WRITE MMM REGISTER
228 T,JMP 000000 000000 000000 000000 TRUE SEQUENCE IS JMP
229 T,JMPO-3 000000 000000 001000 000000 TRUE SEQUENCE IS JMP. IR-BITS 0-3 CONTROLS THE FOUR LOWER JUMP ADDRESS BITS
230 T,JMP8-10 000000 002000 001000 000000 TRUE SEQUENCE IS JMP. IR-BITS 8-10 CONTROLS THE 3 LOWER JUMP ADDRESS BITS
231 T,JMPALD 000000 004000 001000 000000 TRUE SEQUENCE IS JMP. 'ALD' CONTROLS THE 4 LOWER JUMP ADDRESS BITS
232 T,JMPOPR 000000 006000 001000 000000 TRUE SEQUENCE IS JMP. THE A-OPERAND CONTROLS THE 4 LOWER JUMP ADDRESS BITS
233 T,MAPJ 000000 000000 004000 000000 IF NEXT INSTRUCTION IS JMP, THE JMP WILL BE MAPPED BY THE MAPPING HARDWARE
234 T,RETURN 000000 000000 000000 000000 TRUE SEQUENCE IS RETURN
235 T,NEXT 000000 000000 100000 000000 TRUE SEQUENCE IS NEXT
236 T,HOLD 000000 000000 000000 000000 TRUE STACK OPERATION IS HOLD
237 T,POP 000000 000000 010000 000000 TRUE STACK OPERATION IS POP
238 T,LOAD 000000 000000 020000 000000 TRUE STACK OPERATION IS LOAD
239 T,PUSH 000000 000000 030000 000000 TRUE STACK OPERATION IS PUSH
240 F,JMP 000000 000000 000000 000000 FALSE SEQUENCE IS JMP. USED TOGETHER WITH A CONDITION SETTING ('COND',...)

```

```

241 F, RETURN 000000 000000 000000 000004 FALSE SEQUENCE IS RETURN. USED TOGETHER WITH A CONDITION SETTING
242 F, NEXT 000000 000000 000000 000010 FALSE SEQUENCE IS NEXT. USED TOGETHER WITH A CONDITION SETTING
243 F, HOLD 000000 000000 000000 000000 FALSE STACK OPERATION IS HOLD. USED TOGETHER WITH A CONDITION SETTING
244 F, POP 000000 000000 000000 000001 FALSE STACK OPERATION IS POP. USED TOGETHER WITH A CONDITION SETTING
245 F, LOAD 000000 000000 000000 000002 FALSE STACK OPERATION IS LOAD. USED TOGETHER WITH A CONDITION SETTING
246 F, PUSH 000000 000000 000000 000003 FALSE STACK OPERATION IS PUSH. USED TOGETHER WITH A CONDITION SETTING
247 COND, STP 000000 000000 000400 000360 CONDITION FOR TESTING IS 'STOP'
248 COND, OPC1T1 000000 000000 000400 000220 CONDITION FOR TESTING IS IF 'T1' IS ON CPU-BOARD
249 COND, OVF 000000 000000 000400 000240 CONDITION FOR TESTING IS 'OVERFLOW' FROM ALU-BIT-SLICES
250 COND, CNY 000000 000000 000400 000260 CONDITION FOR TESTING IS 'CARRY' FROM ALU-BIT-SLICES
251 COND, F11 000000 000000 000400 000300 CONDITION FOR TESTING IS BIT 11 FROM ALU-BIT-SLICES
252 COND, F15 000000 000000 000400 000320 CONDITION FOR TESTING IS BIT 15 FROM ALU-BIT-SLICES
253 COND, F=0 000000 000000 000400 000340 CONDITION FOR TESTING IS 'ALL ZEROS' FROM ALU-BIT-SLICES
254 COND, PR=2 000000 000000 000400 000000 CONDITION FOR TESTING IS 'R-BITS 0-2' = '010' (DESTINATION FIELD IS 'DP')
255 COND, LC=0 000000 000000 000400 000020 CONDITION FOR TESTING IS 'LOOP-COUNTER'-CONTENT = 0
256 COND, FETCH 000000 000000 000400 000100 CONDITION FOR TESTING IS LAST MEM.REQ. WAS FETCH
257 COND, IRQ 000000 000000 000400 000040 CONDITION FOR TESTING IS 'IRQ'. CHECKS LEV 10-15 IF 'COMM, CLRQ' HAS BEEN GIVEN
258 COND, RESTR 000000 000000 000400 000050 CONDITION FOR TESTING IS 'RESTRICTED-MODE'. TRUE IF RING 0-1
259 COND, R=P 000000 000000 000400 000200 CONDITION FOR TESTING IS THE R=P-FLIP-FLOP. TRUE IF '(R=P)' OR '(F=0)' IS TRUE
260 COND, SOB 000000 000000 000400 000120 CONDITION FOR TESTING IS THE 'ONE-OUT-DETECT'-FLIP-FLOP
261 COND, DZD 000000 000000 000400 000140 CONDITION FOR TESTING IS THE 'DOUBLE-ZERO-DETECT'-FLIP-FLOP
262 COND, COND 000000 000000 000400 000160 CONDITION FOR TESTING IS THE OUTCOME OF THE LATEST TEST. (USED TO DELAY TESTS)
263 AB, CDIGI 000000 000000 000011 170000 COUNTER CONTROLLING THE NUMBER OF DIGITS IN AN OCTAL NUMBER
264 AB, UPWR 000000 000000 000012 130000 SCRATCH WORD KEEPING THE UPPER ADDRESS LIMIT IN A MOPC DUMP COMMAND
265 AB, CURRN 000000 000000 000013 130000 SCRATCH WORD KEEPING THE CURRENT ADDRESS IN A MOPC DUMP COMMAND
266 AB, PRCHR 000000 000000 000015 160000 SCRATCH WORD CONTAINING THE NEXT CHARACTER TO BE WRITTEN BY MOPC
267 AB, TX11 000000 000000 000014 160000 SCRATCH WORD CONTAINING DISPLAY TEXT
268 AB, TX12 000000 000000 000013 160000 SCRATCH WORD CONTAINING DISPLAY TEXT
269 AB, TX12 000000 000000 000013 160000 SCRATCH WORD CONTAINING DISPLAY TEXT
270 AB, SDRAM 000000 000000 000015 140000 SCRATCH WORD CONTAINING SCRAMBELED REPRESENTATION OF LETTERS IN MOPC-INPUT
271 AB, OCTNR 000000 000000 000015 150000 SCRATCH WORD CONTAINING OCTAL NUMBER ASSEMBLED FROM MOPC-INPUT
272 AB, DISPL 000000 000000 000015 160000 TYPE OF RUNNING DISPLAY
273 AB, OCTAB 000000 000000 000013 150000 ADDRESS OF RUNNING DISPLAY
274 AB, OCTA2 000000 000000 000012 160000 WORD TO EXTEND ADDRESS IN OCTAD TO 24 BITS
275 AB, DEPOS 000000 000000 000011 150000 SOME OCTAL DIGIT HAS BEEN WRITTEN SINCE LAST COMMAND WAS TERMINATED
276 AB, DUMPF 000000 000000 000016 150000 SCRATCH WORD INDICATING THAT A DUMP IS IN PROGRESS
277 AB, ROMLY 000000 000000 000014 150000 THE EXAMINED REGISTER IS READ-ONLY
278 AB, WRTYP 000000 000000 000012 150000 TYPE OF VARIABLE IN CASE OF DEPOSIT
279 AB, WRADR 000000 000000 000011 150000 ADDRESS OF VARIABLE IN CASE OF DEPOSIT
280 AB, ACTLV 000000 000000 000017 140000 SCRATCH WORD HOLDING 'ACTIVE LEVELS'
281 AB, PVL 000000 000000 000015 170000 SCRATCH WORD HOLDING 'PREVIOUS LEVEL'
282 AB, ILE 000000 000000 000012 170000 SCRATCH WORD HOLDING A 'PIC'-REPRESENTATION OF THE LAST 'IIE'-SETTING
283 AB, PID 000000 000000 000014 170000 SCRATCH WORD HOLDING THE 'MICROPROGRAM-KNOWN BITS' OF THE 'PID'-REGISTER
284 AB, PIE 000000 000000 000013 170000 SCRATCH WORD HOLDING THE 'PIE'-REGISTER
285 AB, STATUS 000000 000000 000017 150000 SCRATCH WORD HOLDING THE LATEST 'COMM, SIOC'-INFORMATION
286 AB, OCTN2 000000 000000 000015 130000 SCRATCH WORD EXPANDING THE 'AB, OCTNR' TO 24 BITS
287 AB, NUMBER 000000 000000 000012 120000 SCRATCH WORD HOLDING AN OCTAL NUMBER BEING PRINTED BY MOPC
288 AB, OPR 000000 000000 000011 160000 SCRATCH WORD HOLDING THE 'OPR'-REGISTER VALUE
289 AB, BRAPT 000000 000000 000016 140000 SCRATCH WORD HOLDING BREAKPOINT ADDRESS
290 AB, SINGL 000000 000000 000014 140000 SCRATCH WORD COUNTING SINGLE-INSTRUCTION
291 AB, BPFLG 000000 000000 000013 140000 SCRATCH WORD INDICATING THAT BREAKPOINT IS ON
292 AB, MACL 000000 000000 000011 140000 SCRATCH WORD HOLDING RETRY COUNTER FOR LOAD AFTER MACL
293 AB, LMP 000000 000000 000012 140000 SCRATCH WORD HOLDING THE 'LMP'-REGISTER VALUE
294 AB, EXNOD 000000 000000 000011 130000 SCRATCH WORD HOLDING THE 'EXM'-REGISTER
295 AB, MANIR 000000 000000 000014 130000 SCRATCH WORD HOLDING FLAG FOR MANUAL IR
296 AB, SSAVE 000000 000000 000015 130000 SCRATCH WORD HOLDING STS DURING DECIMAL INSTRUCTIONS
297 REMOVED 000000 000000 000000 000000
298 REMOVED 000000 000000 000000 000000
299 REMOVED 000000 000000 000000 000000
300 REMOVED 000000 000000 000000 000000

```

APPENDIX B

DATA FOR AM 2901 BIT SLICE

Taken from "AM2900 Bipolar Microprocessor Family" issued by Advanced Micro Devices, Inc. (to be contacted for more information).



<p>DISTINCTIVE CHARACTERISTICS</p> <ul style="list-style-type: none"> • Two-address architecture – Independent simultaneous access to two working registers saves machine cycles. • Eight-function ALU – Performs addition, two subtraction operations, and five logic functions on two source operands. • Flexible data source selection – ALU data is selected from five source ports for a total of 203 source operand pairs for every ALU function. • Left/right shift independent of ALU – Add and shift operations take only one cycle. • Four status flags – Carry, overflow, zero, and negative. • Expandable – Connect any number of Am2901's together for longer word lengths. • Microprogrammable – Three groups of three bits each for source operand, ALU function, and destination control. 	<p>GENERAL DESCRIPTION</p> <p>The four-bit bipolar microprocessor slice is designed as a high-speed cascadable element intended for use in CPU's, peripheral controllers, programmable microprocessors and numerous other applications. The microinstruction flexibility of the Am2901 will allow efficient emulation of almost any digital computing machine.</p> <p>The device, as shown in the block diagram below, consists of a 16-word by 4-bit two-port RAM, a high-speed ALU, and the associated shifting, decoding and multiplexing circuitry. The nine-bit microinstruction word is organized into three groups of three bits each and selects the ALU source operands, the ALU function, and the ALU destination register. The microprocessor is cascadable with full look-ahead or with ripple carry, has three-state outputs, and provides various status flag outputs from the ALU. Advanced low-power Schottky processing is used to fabricate this 40-lead LSI chip.</p>
<p>TABLE OF CONTENTS</p> <p>Block Diagram 4</p> <p>Function Tables 5</p> <p>Package Outline 7</p> <p>Connection Diagram 8</p> <p>Pin Definitions 8</p> <p>Screening 9</p> <p>Order Codes 9</p> <p>DC Characteristics 10</p> <p>AC Characteristics 11</p> <p>Switching Waveforms 12</p> <p>Applications 13</p>	<p>MICROPROCESSOR SLICE BLOCK DIAGRAM</p> <p>The diagram illustrates the internal architecture of the Am2901 slice. At the top, a 9-bit microinstruction word is divided into three 3-bit fields: Destination Control (bits 8, 7, 6), ALU Function (bits 5, 4, 3), and ALU Source (bits 2, 1, 0). These fields control various internal blocks: a 16-word by 4-bit RAM with RAM Shift and RAM Enable inputs; two 4-bit Addressable Registers (A and B) with their own shift and enable controls; a 4-bit G Register; an ALU Data Source Selector that routes data from the registers or external sources to the 8-function ALU; and an Output Data Selector that routes the ALU result to the DATA OUT pin. Status flags (CARRY, Z, N, O) are also shown as outputs from the ALU.</p>

Copyright © 1976 Advanced Micro Devices, Inc.
Reproduced with permission of copyright owner.

ARCHITECTURE

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

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

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

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

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

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

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

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

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

ALU function. The definition of these inputs is shown in Figure 3. The octal code is also shown for reference. The normal technique for cascading the ALU of several devices is in a look-ahead carry mode. Carry generate, G, and carry propagate, P, are outputs of the device for use with a carry-look-ahead generator such as the Am2902 (1B2). A carry-out, C_{n+4}, is also generated and is available as an output for use as the carry flag in a status register. Both carry-in (C_n) and carry-out (C_{n+4}) are active HIGH.

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

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

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

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

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

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

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

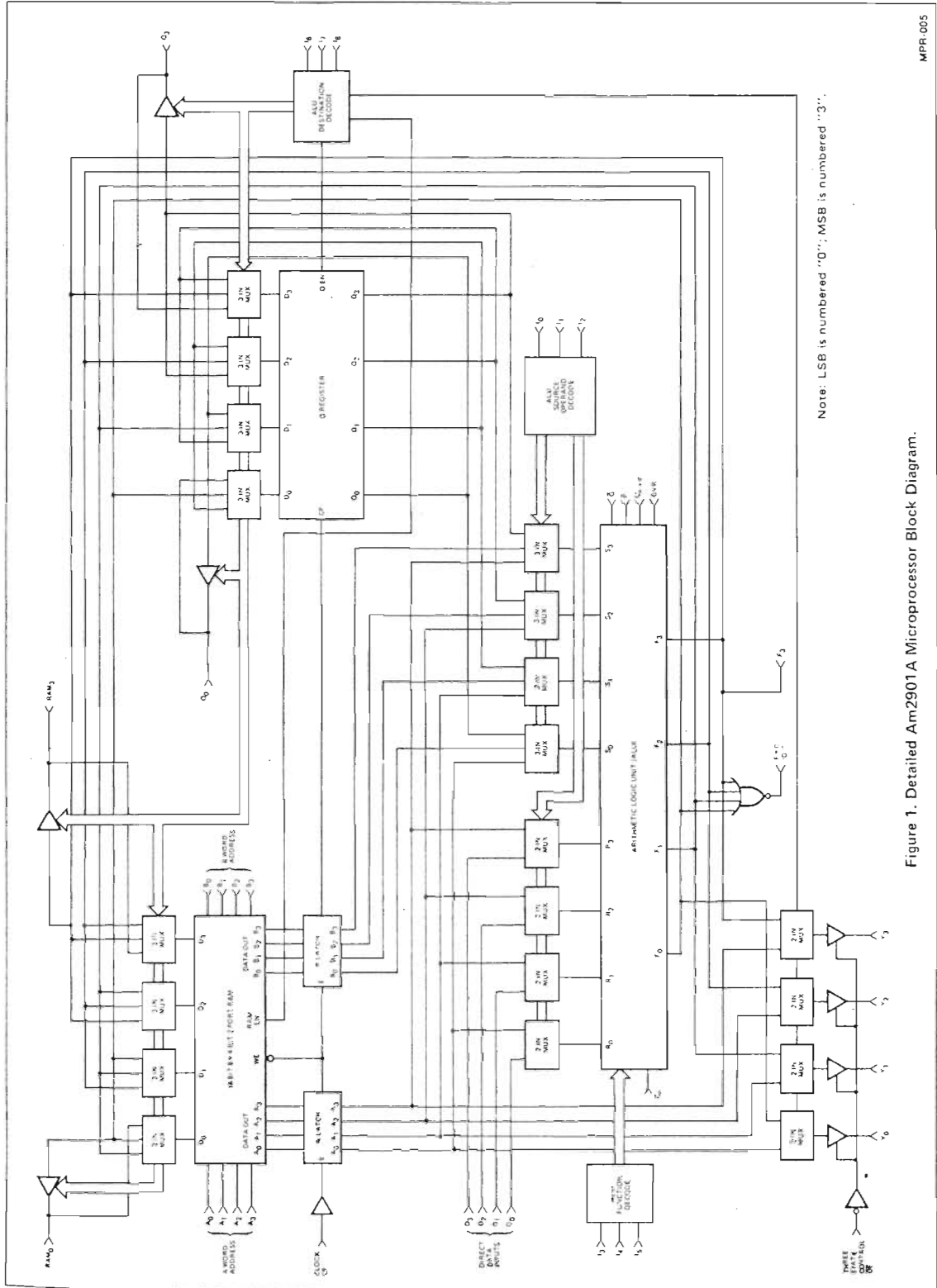


Figure 1. Detailed Am2901A Microprocessor Block Diagram.

Copyright © 1976 Advanced Micro Devices, Inc.
Reproduced with permission of copyright owner.

ND-06.018.01

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

Figure 2. ALU Source Operand Control.

MICRO CODE				ALU Function	Symbol
I ₅	I ₄	I ₃	Octal Code		
L	L	L	0	R Plus S	R + S
L	L	H	1	S Minus R	S - R
L	H	L	2	R Minus S	R - S
L	H	H	3	R OR S	R ∨ S
H	L	L	4	R AND S	R ∧ S
H	L	H	5	R AND S	R ∧ S
H	H	L	6	R EX-OR S	R ⊕ S
H	H	H	7	R EX-NOR S	R ⊙ S

Figure 3. ALU Function Control.

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

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

Figure 4. ALU Destination Control.

OCTAL ALU Function	OCTAL ALU Source	0	1	2	3	4	5	6	7
		C _n = L R Plus S C _n = H	A + Q	A + B	Q	B	A	D + A	D + Q
C _n = L S Minus R C _n = H	A - Q + 1	A + B + 1	Q + 1	B + 1	A + 1	D + A + 1	D + Q + 1	D + 1	
C _n = L R Minus S C _n = H	Q - A - 1	B - A - 1	Q - 1	B - 1	A - 1	A - D - 1	Q - D - 1	-D - 1	
C _n = L R Minus S C _n = H	Q - A	B - A	Q	B	A	A - D	Q - D	-D	
C _n = L R Minus S C _n = H	A - Q - 1	A - B - 1	-Q - 1	-B - 1	-A - 1	D - A - 1	D - Q - 1	D - 1	
C _n = L R Minus S C _n = H	A - Q	A - B	-Q	-B	-A	D - A	D - Q	D	
R OR S	A ∨ Q	A ∨ B	Q	B	A	D ∨ A	D ∨ Q	D	
R AND S	A ∧ Q	A ∧ B	Q	B	A	D ∧ A	D ∧ Q	D	
R AND S	A ∧ Q	A ∧ B	Q	B	A	D ∧ A	D ∧ Q	D	
R EX-OR S	A ⊕ Q	A ⊕ B	Q	B	A	D ⊕ A	D ⊕ Q	D	
R EX-NOR S	A ⊙ Q	A ⊙ B	Q	B	A	D ⊙ A	D ⊙ Q	D	

+ = Plus, - = Minus, ∨ = OR, ∧ = AND, ⊕ = EX-OR

Figure 5. Source Operand and ALU Function Matrix.

SOURCE OPERANDS AND ALU FUNCTIONS

There are eight source operand pairs available to the ALU as selected by the I₀, I₁, and I₂ instruction inputs. The ALU can perform eight functions; five logic and three arithmetic. The I₃, I₄, and I₅ instruction inputs control this function selection. The carry input, C_n, also affects the ALU results when in the arithmetic mode. The C_n input has no effect in the logic mode. When I₀ through I₅ and C_n are viewed together, the matrix of

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

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

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

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

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

Figure 7. ALU Arithmetic Mode Functions.

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

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

Definitions (+ = OR)

$$\begin{aligned}
 P_0 &= R_0 + S_0 & G_0 &= R_0 S_0 \\
 P_1 &= R_1 + S_1 & G_1 &= R_1 S_1 \\
 P_2 &= R_2 + S_2 & G_2 &= R_2 S_2 \\
 P_3 &= R_3 + S_3 & G_3 &= R_3 S_3 \\
 C_4 &= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_n \\
 C_3 &= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_n
 \end{aligned}$$

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

+ = OR

Figure 8.

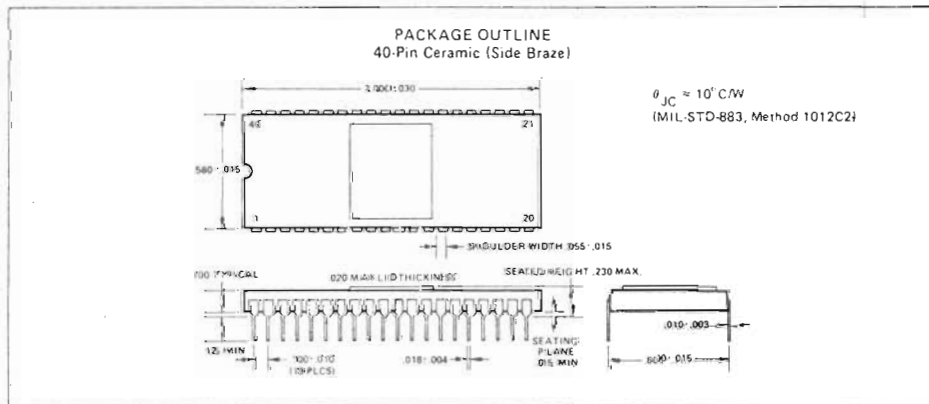


Figure 9.

Copyright © 1976 Advanced Micro Devices, Inc.
Reproduced with permission of copyright owner.

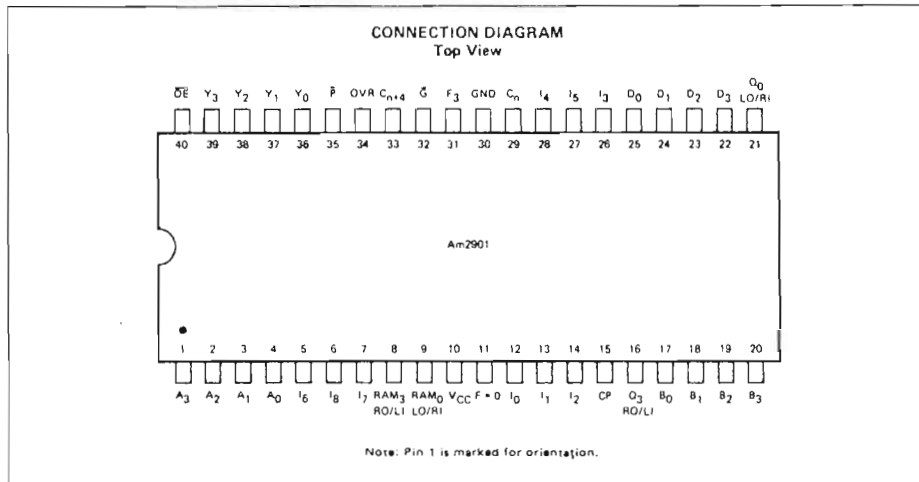


Figure 10.

PIN DEFINITIONS

A₀₋₃	The four address inputs to the register stack used to select one register whose contents are displayed through the A-port.	Y₀₋₃	The four data outputs of the Am2901. These are three-state output lines. When enabled, they display either the four outputs of the ALU or the data on the A-port of the register stack, as determined by the destination code I ₆₇₈ .
B₀₋₃	The four address inputs to the register stack used to select one register whose contents are displayed through the B-port and into which new data can be written when the clock goes LOW.	\overline{OE}	Output Enable. When \overline{OE} is HIGH, the Y outputs are OFF; when \overline{OE} is LOW, the Y outputs are active (HIGH or LOW).
I₀₋₈	The nine instruction control lines to the Am2901, used to determine what data sources will be applied to the ALU (I ₀₁₂), what function the ALU will perform (I ₃₄₅), and what data is to be deposited in the Q-register or the register stack (I ₆₇₈).	$\overline{P}, \overline{G}$	The carry generate and propagate outputs of the Am2901's ALU. These signals are used with the Am2902 for carry-lookahead. See Figure 8 for the logic equations.
RO/LI	A shift line at the MSB of the Q register (Q ₃ RO/LI) and the register stack (RAM ₃ RO/LI). Electrically these lines are three-state outputs connected to TTL inputs internal to the Am2901. When the destination code on I ₆₇₈ indicates a right shift (octal 6 or 7) the three-state outputs are enabled and the MSB of the Q register is available on the Q ₃ RO/LI pin and the MSB of the ALU output is available on the RAM ₃ RO/LI pin. Otherwise, the three-state outputs are OFF (high-impedance) and the pins are electrically LS-TTL inputs. When the destination code calls for a down (left) shift, the pins are used as the data inputs to the MSB of the Q register (octal 4) and RAM (octal 4 or 5).	OVR	Overflow. This pin is logically the Exclusive-OR of the carry-in and carry-out of the MSB of the ALU. At the most significant end of the word, this pin indicates that the result of an arithmetic two's complement operation has overflowed into the sign-bit. See Figure 8 for logic equation.
LO/RI	Shift lines like RO/LI, but at the LSB of the Q-register and RAM. These pins are tied to the RO/LI pin of the adjacent device to transfer data between devices for left and right shifts of the Q register and ALU data.	F = 0	This is an open collector output which goes HIGH (OFF) if the data on the four ALU outputs F ₀₋₃ are all LOW. In positive logic, it indicates the result of an ALU operation is zero.
D₀₋₃	Direct data inputs. A four-bit data field which may be selected as one of the ALU data sources for entering data into the Am2901. D ₀ is the LSB.	C_n	The carry-in to the Am2901's ALU.
		C_{n+4}	The carry-out of the Am2901's ALU. See Figure 8 for equations.
		CP	The clock to the Am2901. The Q register and register stack outputs change on the clock LOW-to-HIGH transition. The clock LOW time is internally the write enable to the 16 x 4 RAM which comprises the "master" latches of the register stack. While the clock is LOW, the "slave" latches on the RAM outputs are closed, storing the data previously on the RAM outputs. This allows synchronous master-slave operation of the register stack.

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

Storage Temperature	-65°C to +150°C
Temperature (Ambient) Under Bias	-55°C to +125°C
Supply Voltage to Ground Potential	-0.5 V to +6.3 V
DC Voltage Applied to Outputs for HIGH Output State	-0.5 V to +V _{CC} max.
DC Input Voltage	-0.5 V to +5.5 V
DC Output Current, Into Outputs	30 mA
DC Input Current	-30 mA to +5.0 mA

OPERATING RANGE

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

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

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

*Not applicable for Am2901PC

ADDITIONAL SCREENING FOR CLASS B PARTS

Step	MIL-STD-883 Method	Conditions	Level
			Am2901DM, FM
Burn-In	1015	D 125°C 168 hours min.	100%
Electrical Test Subgroup 1	5004		100%
Subgroup 2			100%
Subgroup 3			100%
Subgroup 7			100%
Subgroup 8			100%
Return to Group A Tests in Standard Screening			

ORDERING INFORMATION

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

GROUP A SUBGROUPS

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

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

Copyright © 1976 Advanced Micro Devices, Inc.
Reproduced with permission of copyright owner.

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

Parameters	Description	Test Conditions (Note 1)	Min.	Typ. (Note 2)	Max.	Units	
V _{OH}	Output HIGH Voltage	V _{CC} = MIN., V _{IN} = V _{IH} or V _{IL}	I _{OH} = -1.6mA Y ₀ , Y ₁ , Y ₂ , Y ₃	2.4			Volts
			I _{OH} = -1.0mA, C _{n+4}	2.4			
			I _{OH} = -800μA, OVR, \bar{P}	2.4			
			I _{OH} = -600μA, F ₃	2.4			
			I _{OH} = -600μA All RO/LI, LO/RI	2.4			
I _{CEX}	Output Leakage Current for F = 0 Output	V _{CC} = MIN., V _{OH} = 5.5V V _{IN} = V _{IH} or V _{IL}			250	μA	
V _{OL}	Output LOW Voltage	V _{CC} = MIN., V _{IN} = V _{IH} or V _{IL}	I _{OL} = 16mA Y ₀ , Y ₁ , Y ₂ , Y ₃			0.5	Volts
			I _{OL} = 10mA, C _{n+4} , F=0			0.5	
			I _{OL} = 8.0mA, OVR, \bar{P}			0.5	
			I _{OL} = 6.0mA, F ₃ All RO/LI, LO/RI			0.5	
V _{IH}	Input HIGH Level	Guaranteed input logical HIGH voltage for all inputs	2.0			Volts	
V _{IL}	Input LOW Level	Guaranteed input logical LOW voltage for all inputs	Military			0.7	Volts
			Commercial			0.8	
V _I	Input Clamp Voltage	V _{CC} = MIN., I _{IN} = -18mA				-1.5	Volts
I _{IL} (Note 3)	Input LOW Current	V _{CC} = MAX., V _{IN} = 0.5V	Clock, \bar{OE}			-0.36	mA
			A ₀ , A ₁ , A ₂ , A ₃			-0.36	
			B ₀ , B ₁ , B ₂ , B ₃			-0.36	
			D ₀ , D ₁ , D ₂ , D ₃			-0.72	
			I ₀ , I ₁ , I ₂ , I ₆ , I ₈			-0.36	
			I ₃ , I ₄ , I ₅ , I ₇			-0.72	
			All LO/RI, RO/LI (Note 4)			-0.8	
			C _n			-3.8	
I _{IH} (Note 3)	Input HIGH Current	V _{CC} = MAX., V _{IN} = 2.7V	Clock, \bar{OE}			20	μA
			A ₀ , A ₁ , A ₂ , A ₃			20	
			B ₀ , B ₁ , B ₂ , B ₃			20	
			D ₀ , D ₁ , D ₂ , D ₃			40	
			I ₀ , I ₁ , I ₂ , I ₆ , I ₈			20	
			I ₃ , I ₄ , I ₅ , I ₇			40	
			All LO/RI, RO/LI (Note 4)			100	
			C _n			200	
I _I	Input HIGH Current	V _{CC} = MAX., V _{IN} = 5.5V				1.0	mA
I _{OZ}	Off State (High Impedance) Output Current	V _{CC} = MAX.	Y ₀ , Y ₁ , Y ₂ , Y ₃	V _O = 2.4V		50	μA
				V _O = 0.5V		-50	
			All LO/RI, RO/LI	V _O = 2.4V (Note 5)		100	
				V _O = 0.5V (Note 5)		-800	
I _{SC}	Output Short Circuit Current (Note 4)		Y ₀ , Y ₁ , Y ₂ , Y ₃ , \bar{G}	-6.0		-40	mA
			C _{n+4}	-6.0		-40	
			OVR, \bar{P}	-6.0		-40	
			F ₃	-6.0		-40	
			All RO/LI, LO/RI	-6.0		-40	
I _{CC}	Power Supply Current	V _{CC} = MAX.	Military		185	280	mA
			Commercial		185	280	

Notes: 1. For conditions shown as MIN or MAX, use the appropriate value specified under Electrical Characteristics for the applicable device type.
2. Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
3. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
4. LO/RI and RO/LI are three-state outputs internally connected to TTL inputs. Input characteristics are measured with I_{67g} in a state such that the three-state output is OFF.

GUARANTEED OPERATING CONDITIONS

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

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

TABLE I

CYCLE TIME AND CLOCK CHARACTERISTICS

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

TABLE II

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

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

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

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

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

TABLE III

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

From Input	Notes	Am2901DC		Am2901DM	
		Set-Up Time	Hold Time	Set-Up Time	Hold Time
A, B Source	2, 3, 4	105 $t_{pwL} + 30$	0	120 $t_{pwL} + 30$	0
B Dest.	2, 4	$t_{pwL} + 15$	0	$t_{pwL} + 15$	0
D		100	0	110	0
C _n		55	0	60	0
I ₀₁₂		85	0	90	0
I ₃₄₅		70	0	75	0
I ₆₇₈	4	$t_{pwL} + 15$	0	$t_{pwL} + 15$	0
R1, L1 (RAM or Q)		30	0	30	0

- Notes: 1. See Figure 11 and 12.
 2. If the B address is used as a source operand, allow for the "A, B source" set-up time. If it is used only for the destination address, use the "B dest." set-up time.
 3. Where two numbers are shown, both must be met.
 4. " t_{pwL} " is the clock LOW time.

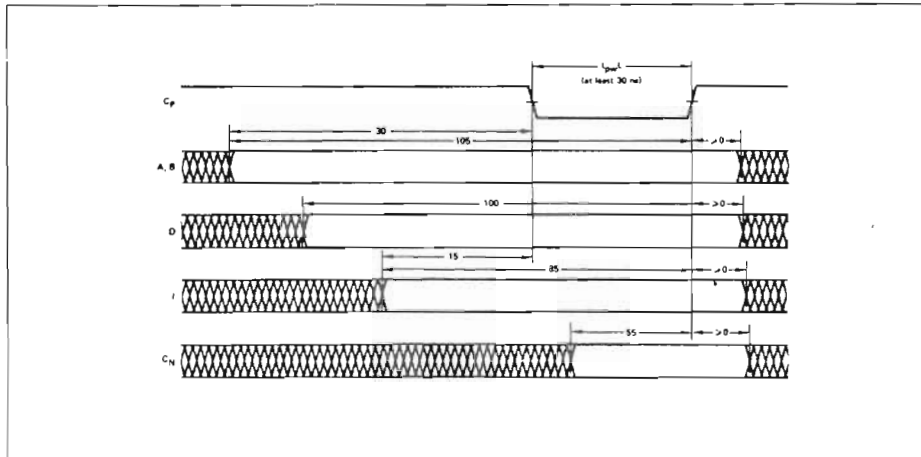
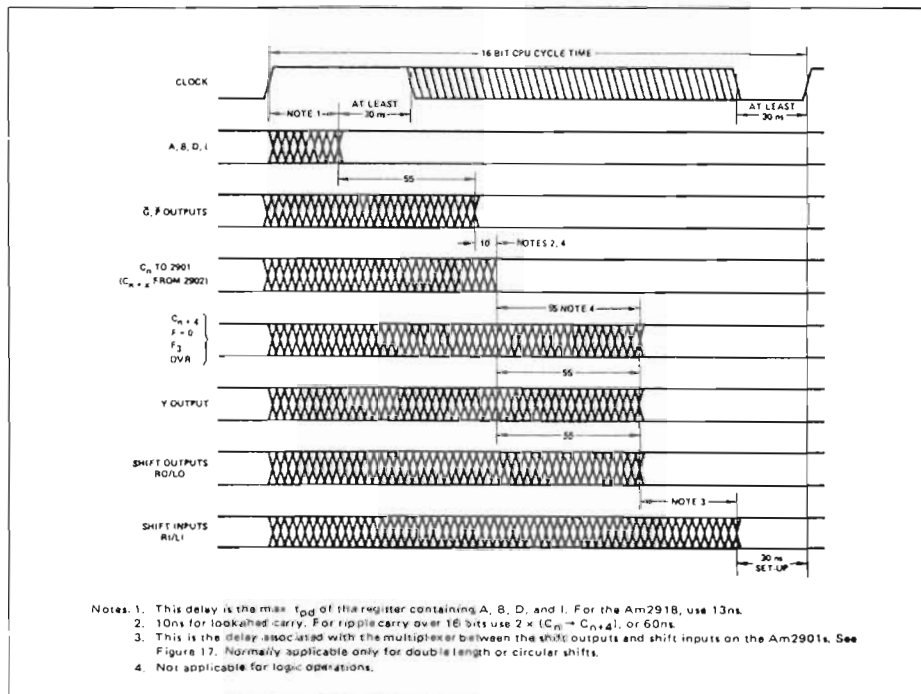


Figure 11. Minimum Cycle Times from Inputs. Numbers Shown are Minimum Data Stable Times for Am2901 DC, in ns.



- Notes: 1. This delay is the max. t_{od} of the register containing A, B, D, and I. For the Am2918, use 13ns.
 2. 10ns for lookahead carry. For ripple carry over 16 bits use $2 \times (C_{m1} \rightarrow C_{n+1})$, or 60ns.
 3. This is the delay associated with the multiplexer between the shift outputs and shift inputs on the Am2901s. See Figure 17. Normally applicable only for double length or circular shifts.
 4. Not applicable for logic operations.

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

Copyright © 1976 Advanced Micro Devices, Inc.
 Reproduced with permission of copyright owner.

ND-06.018.01

APPENDIX C

DATA FOR AM 2914 INTERRUPT CONTROLLER

Taken from "AM2900 Bipolar Microprocessor Family" issued by Advanced Micro Devices, Inc. (to be contacted for more information).

Am2914

Vectored Priority Interrupt Controller

DISTINCTIVE CHARACTERISTICS

- Accepts 8 interrupt inputs
- Interrupts may be pulses or levels and are stored internally
- Built-in mask register
- Six different operations can be performed on mask register
- Built-in status register
- Status register holds code for lowest allowed interrupt
- Vectored output
- Output is binary code for highest priority unmasked interrupt
- Expandable
- Any number of Am2914's may be stacked for large interrupt systems
- Microprogrammable
- Executes 16 different microinstructions
- Instruction enable pin aids in vertical microprogramming
- High speed operation
- Delay from an interrupt clocked into the interrupt register to interrupt request output is typically 60 ns

FUNCTIONAL DESCRIPTION

The Am2914 is a high-speed, eight-bit priority interrupt unit that is cascadeable to handle any number of priority interrupt requests. The high-speed of the Am2914 makes it ideal for use in Am2900 family microcomputer designs, but it can also be used with the Am5600A MOS microprocessor.

The Am2914 receives interrupt requests on 8 interrupt input lines (IP0-7). A LOW level is a request. An internal latch may be used to catch pulses on these lines, or the latch may be bypassed so the request lines drive the edge-triggered interrupt register directly. An 8-bit mask register is used to mask individual interrupts. Considerable flexibility is provided for controlling the mask register. Requests in the interrupt register are ANDed with the corresponding bits in the mask register and the results are sent to an 8-input priority encoder, which produces a three-bit encoded vector representing the highest numbered input which is not masked.

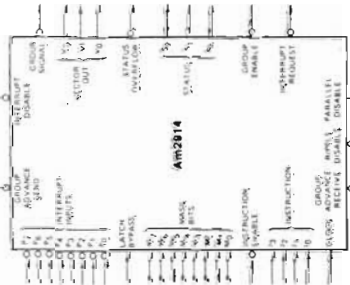
An interrupt status register is used to point to the lowest priority at which an interrupt will be accepted. The contents of the status register are compared with the output of the priority encoder, and an interrupt request output will occur if the vector is greater than or equal to status. Whenever a vector is read from the Am2914 the status register is automatically updated to point to one level higher than the vector read. (The status register can be loaded externally or read out at any time using the S pins.) Signals are provided for moving the status upward across devices (Group Advance Send and Group Advance Receive) and for inhibiting lower priorities from higher order devices (Ripple Disable, Parallel Disable, and Interrupt Disable). A status overflow output indicates that an interrupt has been read at the highest priority.

The Am2914 is controlled by a 4-bit instruction field (I0-I3). The command on the instruction lines is executed if IE is LOW and is ignored if IE is HIGH, allowing the 4 bits to be shared with other devices.

TABLE OF CONTENTS

Block Diagram	2-107
Connection Diagrams	2-108
Instructions	2-108
Ordering Information	2-109
DC Characteristics	2-110
AC Characteristics	2-111
Burn-in Circuit	2-113
Applications	2-114
Detailed Logic Description	2-130

LOGIC SYMBOL



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

Step	MIL-STD-883 Method	Conditions	Level
Pre-Seal Visual Inspection	2010	B	Am2914PC, DC 100% Am2914DM, FM 100%
Stabilization Bake	1008	C 24-hour C 150°C	100% 100%
Temperature Cycle	1010	C -65°C to +150°C C 10 cycles	100%
Centrifuge	2001	B 10,000 G	100% *
Fine Leak	1014	A 5 x 10 ⁻⁸ atm-cc/sec	100% *
Gross Leak	1014	C2 Fluorocarbon	100% *
Electrical Test	5004	See below for definitions of subgroups	100%
Insert Additional Screening here for Class B Parts			
Group A Sample Tests			
Subgroup 1			LTPD = 5
Subgroup 2			LTPD = 7
Subgroup 3			LTPD = 7
Subgroup 7	5005	See below for definitions of subgroups	LTPD = 7
Subgroup 8		Maximum accept number is 3	LTPD = 7
Subgroup 9			LTPD = 7

*Not applicable for Am2914PC.

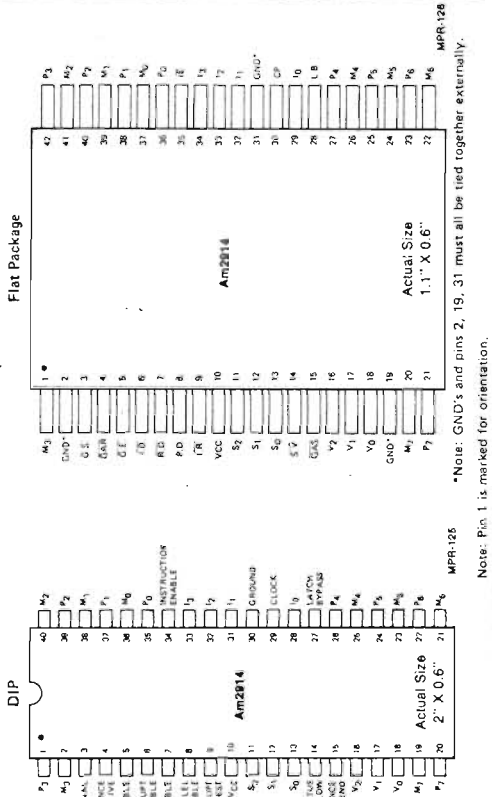
ORDERING INFORMATION

Order the part number according to the table below to obtain the desired package, temperature range, and screening level.

Order Number	Package Type (Note 1)	Temperature Range (Note 2)	Screening Level (Note 3)
AM2914PC	P-40	C	C-1
AM2914DC	D-40	C	C-1
AM2914DC-B	D-40	C	B-1
AM2914DM	D-40	M	C-3
AM2914DM-B	D-40	M	B-3
AM2914FM	F-42	M	C-3
AM2914FM-B	F-42	M	B-3
AM2914XC	Dic8	C	Visual Inspection to MIL-STD-883 Method 2010B.
AM2914XM	Dic8	M	

- Notes: 1. P = Molded DIP, D = Hermetic DIP, F = Flat Pak. Number following letter is number of leads. See Appendix B for detailed outline. Where Appendix B contains several dash numbers, any of the variations of the package may be used unless otherwise specified.
2. C = 0°C to +70°C, M = -55°C to +125°C.
3. See Appendix A for details of screening. Levels C-1 and C-3 conform to MIL-STD-883, Class C. Level B-3 conforms to MIL-STD-883, Class B.

CONNECTION DIAGRAMS — Top Views



Metalization and Pad Layout

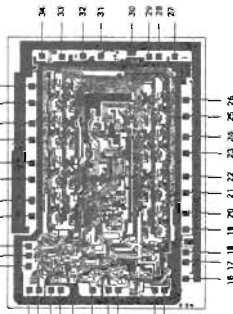


TABLE I
MICROINSTRUCTION SET FOR Am2914 PRIORITY INTERRUPT CIRCUIT

Decimal 5 ₁ 2 ₁ 1 ₀	Mnemonic	Instructions	Instruction
14	LDM	Load mask register from M bus	Mask Register Functions Load mask register from M bus Read mask register from M bus Set mask register (initials all interrupts) Clear mask register from M bus Bit set mask register from M bus
17	RDW	Read mask register from M bus	
18	SETM	Set mask register (initials all interrupts)	
19	CLRM	Clear mask register from M bus	
10	BCLR	Bit clear mask register from M bus	Priority Interrupt Register Clear Clear interrupts from mask register (uses the M bus). Clear the individual interrupt associated with the last vector read
11	BSETM	Bit set mask register from M bus	
9	LDSTA	Status Register Functions Read status register from S bus and LGE Rip-flop from GE input	Master Clear Clear all interrupts, clear mask register, clear status register, clear LGE flip-flop, enable interrupt request.
6	RDSTA	Read status register to S bus	
15	ENIN	Interrupt Request Control Enable interrupt request	Vectorized Output Read vector output to V outputs, load V+1 into status register, load V into vector hold register, and set vector clear enable flip-flop.
13	DISIN	Disable interrupt request	

SWITCHING CHARACTERISTICS AT 25°C AND 5.0 VOLTS
 Note: Guaranteed limits at 25°C and 5.0V are group A, subgroup 9 tests
 All outputs fully loaded, $C_L = 50pF$. Measurements made at 1.5V with
 input levels of 0V and 3.0V. All numbers are in ns.
 For interrupt request output, $R_L = 470\Omega$

TABLE I. CLOCK AND INTERRUPT INPUT PULSE WIDTHS (ns)

Time	GUARANTEED
Minimum Clock LOW Time	30
Minimum Clock HIGH Time	30
Minimum Interrupt Input (PG-PY) LOW Time for Guaranteed Acceptance (Pulse Mode)	25
Maximum Interrupt Input (PG-PY) LOW Time for Guaranteed Rejection (Pulse Mode)	10

TABLE II. COMBINATIONAL PROPAGATION DELAYS (ns)

To Output	TYPICAL						GUARANTEED					
	M Bus	S Bus	V012	Irpt Req	Ripple Disable	Group Advance Send	M Bus	S Bus	V012	Irpt Req	Ripple Disable	Group Advance Send
From Input												
IE	38	40	40	-	-	30	48	55	55	-	-	47
I0123	36	40	40	-	-	30	48	55	55	-	-	47
Irpt. Disable	-	-	25	35	8	19	-	-	37	42	13	23

$R_L = 2.0k\Omega, C_L = 15\mu F$

TABLE III. DELAYS FROM CLOCK TO OUTPUTS (ns)

Clock Path	TYPICAL						GUARANTEED					
	To Irpt Req	To Irpt Req	To PD	To RD	To GAS	To Status O'flow	To Irpt Req	To Irpt Req	To PD	To RD	To GAS	To Status O'flow
Irpt Latches and Register	55	65	37	39	47	-	67	82	57	57	66	-
Mask Register	55	65	37	39	47	-	67	82	57	57	66	-
Status Register	45	55	28	31	37	-	59	74	57	57	58	-
Lowest Group Enabled Flip Flop	-	-	22	25	-	-	17	-	42	45	-	32
Irpt Request Enable Flip Flop	-	-	40	-	-	-	-	56	-	-	-	-
Status Overflow Flip Flop	-	-	-	-	-	17	-	-	-	-	-	25

TABLE IV. SET-UP AND HOLD TIME REQUIREMENTS (ns)
 (All relative to clock. LOW-to-HIGH transition)

From Input	GUARANTEED	
	Set-up Time	Hold Time
S-Bus	11	8
M-Bus	11	8
PG-PY	11	6
Latch Bypass	16	0
IE	46	0
I0123 (See Note)	1pwL + 29	0
GE	11	11
GAR	11	11
Irpt. Disable	35	0
PG-PY Hold Time Relative to LB	-	16

Note: 1pwL is the Clock LOW Time. Both Set-up Times must be met.

Parameter	Temperature	Vcc
Ah2914PC, DE	0°C to +70°C	4.75V to 5.25V
Ah2914DM, FM	-55°C to +125°C	4.50V to 5.50V
Storage Temperature		-65°C to +150°C
Temperature (Ambient) Under Bias		-65°C to +125°C
Supply Voltage to Ground Potential		-0.5V to +7.0V
DC Voltage Applied to Outputs for High Output State		+0.5V to +Vcc max.
DC Input Voltage		-0.5V to 5.5V
DC Output Current, Into Outputs		30mA
DC Input Current		-30mA to +5.0mA

OPERATING RANGE

Parameter	Temperature	Vcc
Ah2914PC, DE	0°C to +70°C	4.75V to 5.25V
Ah2914DM, FM	-55°C to +125°C	4.50V to 5.50V

ELECTRICAL CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (Unless Otherwise Noted)

Parameter	Description	Test Conditions (Note 1)	Min.	Typ. (Note 2)	Max.	Units
V _{OH}	Output High Voltage	V _{CC} = MIN., V _{IN} = VIH or VIL, I _{OL} = -1.0mA, I _{OH} = -2.8mA	2.4	2.4		Volts
I _{CEX}	Output Leakage Current for IR Output	V _{CC} = MIN., V _O = 5.5V			250	μA
V _{OL}	Output LOW Voltage	V _{CC} = MIN., V _{IN} = VIH or VIL, I _{OL} = 4.0mA, I _{OL} = 8.0mA, I _{OL} = 12mA			0.4 0.45 0.5	Volts
V _{IH}	Input HIGH Level	Guaranteed input logical HIGH voltage for all inputs	2.0			Volts
V _{IL}	Input LOW Level	Guaranteed input logical LOW voltage for all inputs			0.8	Volts
V _I	Input Clamp Voltage	V _{CC} = MIN., I _{IN} = -18mA			-1.5 -0.15	Volts
I _{IL}	Input LOW Current	V _{CC} = MAX., V _{IN} = 0.4V			-0.1 -0.4	mA
I _{IH}	Input HIGH Current	V _{CC} = MAX., V _{IN} = 5.5V			-0.8 -0.8	mA
I _I	Input HIGH Current	V _{CC} = MAX., V _{IN} = 5.5V			1.0 1.0	mA
I _O	Off-State Output Current	V _{OUT} = 0.5V, V _{CC} = MAX.			-150 -100	μA
I _{CC}	Power-Supply Current	V _{OUT} = 2.4V, V _{CC} = 5.0V, 25°C		170		mA
I _{SC}	Output Short Circuit Current (Note 3)	V _{CC} = MAX.	-40			mA

Note 1: For conditions shown as MIN. or MAX., use the appropriate values specified under Electrical Characteristics for the applicable device type.
 2: Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
 3: Test margin from 0ns output should be allowed at 8 times; Duration of the short circuit test should not exceed one second.

SWITCHING CHARACTERISTICS OVER OPERATING VOLTAGE AND TEMPERATURE RANGE

(Group A, subgroup 10 and 11 tests and limits)

All outputs fully loaded, $C_L = 50pF$. Measurements made at 1.5V with input levels of 0V and 3.0V. For Interrupt Request Output, $R_L = 470\Omega$.

TABLE V. CLOCK AND INTERRUPT INPUT PULSE WIDTHS (ns)

Time	Am2914PC, DC, XC		Am2914DM, FM, XM	
	$T_A = 0^\circ C$ to $+70^\circ C$, $5V \pm 5\%$	$T_C = -55^\circ C$ to $+125^\circ C$, $5V \pm 10\%$	$T_A = 0^\circ C$ to $+70^\circ C$, $5V \pm 5\%$	$T_C = -55^\circ C$ to $+125^\circ C$, $5V \pm 10\%$
Minimum Clock LOW Time	30		30	
Minimum Clock HIGH Time	30		30	
Minimum Interrupt Input (P ₀ -P ₇) LOW Time for Guaranteed Acceptance (Pulse Width)	40		40	
Maximum Interrupt Input (P ₀ -P ₇) LOW Time for Guaranteed Rejection (Pulse Width)	8		8	
Minimum Clock Period, IE = H on current cycle and previous cycle	50		55	
Minimum Clock Period, IE = L on current cycle or previous cycle	100		110	

TABLE VI. MAXIMUM COMBINATIONAL PROPAGATION DELAYS (ns)

To Output	Am2914PC, DC, XC				Am2914DM, FM, XM			
	$T_A = 0^\circ C$ to $+70^\circ C$, $5V \pm 5\%$	$T_C = -55^\circ C$ to $+125^\circ C$, $5V \pm 10\%$	M Bus	S Bus	M Bus	S Bus	V012	Group Advance Send
From Input								
IE	52	60	60	68	56	60	68	62
I0123	52	60	60	68	56	60	68	62
Irpt. Disable	-	40	52	14	27	-	46	60
								15
								33

$R_L = 2.0k\Omega$, $C_L = 15pF$

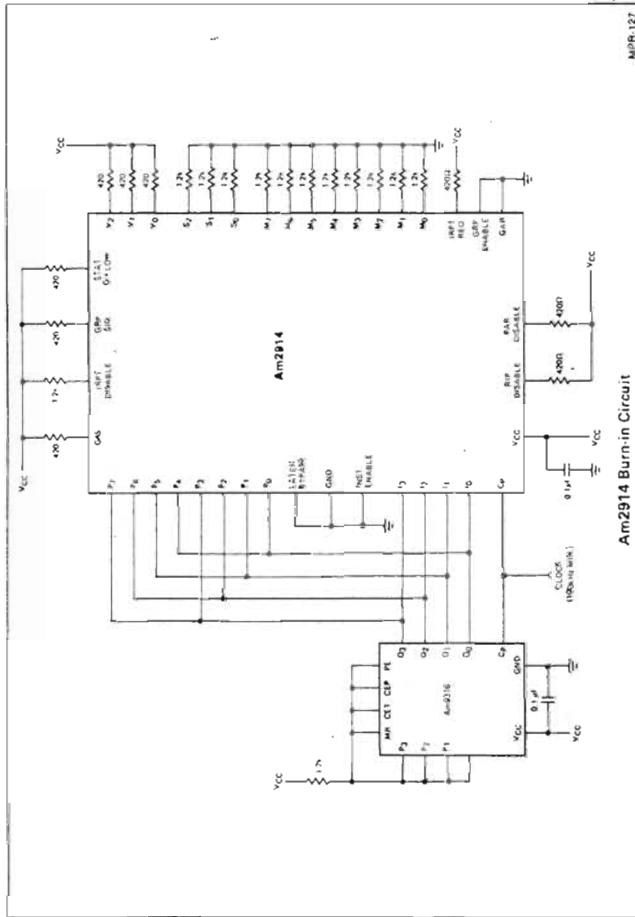
TABLE VII. MAXIMUM DELAYS FROM CLOCK TO OUTPUTS (ns)

Clock Path	Am2914PC, DC, XC				Am2914DM, FM, XM			
	$T_A = 0^\circ C$ to $+70^\circ C$, $5V \pm 5\%$	$T_C = -55^\circ C$ to $+125^\circ C$, $5V \pm 10\%$	To V012	To Status	To V012	To Status	To RD	To GS
Irpt Latches and Register	76	97	67	80	82	106	75	85
Mask Register	76	97	67	80	82	106	75	85
Status Register	67	88	63	70	73	96	66	76
Lowest Group Enabled Flip-Flop	-	48	52	-	38	-	54	58
Irpt Request Enable Flip-Flop	-	62	-	-	66	-	-	-
Status Overflow Flip-Flop	-	-	-	30	-	-	-	35

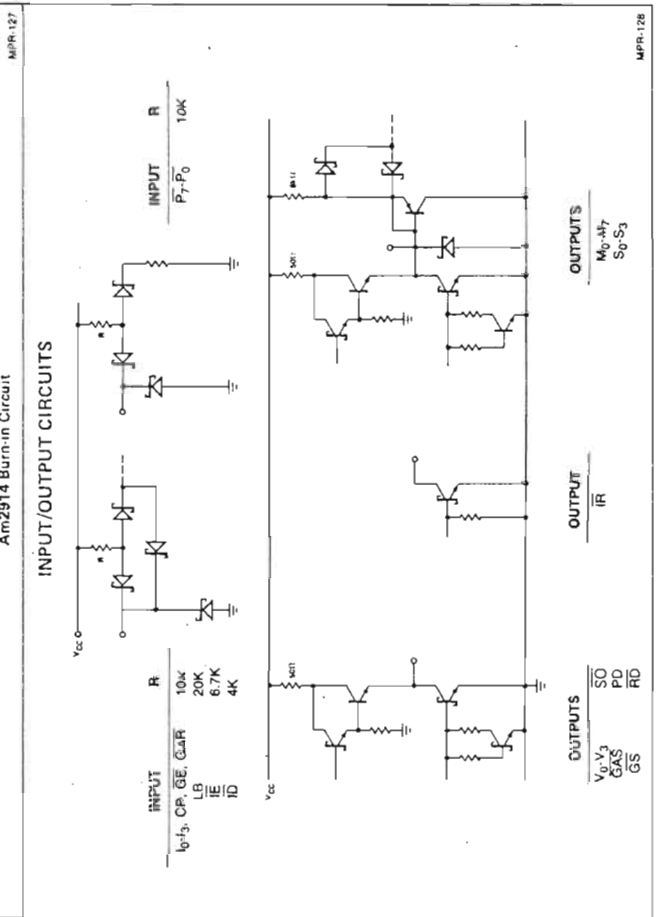
TABLE VIII. SET-UP AND HOLD TIME REQUIREMENTS (ns)

From Input	Am2914PC, DC, XC		Am2914DM, FM, XM	
	$T_A = 0^\circ C$ to $+70^\circ C$, $5V \pm 5\%$	$T_C = -55^\circ C$ to $+125^\circ C$, $5V \pm 10\%$	Set-Up Time	Hold Time
S-Bus	15	10	15	10
M-Bus	15	10	15	10
P0-P7	15	8	15	8
Latch Bypass	20	0	20	0
IE	55	0	55	0
I0123 (See Note)	$t_{pwL} + 33$	0	$t_{pwL} + 40$	0
GE	15	13	15	13
GAR	15	13	15	13
Irpt Disable	42	0	42	0
P0-P7 Hold Time Relative to LB	-	20	-	20

Note: t_{pwL} is the Clock LOW Time. Both Set-up times must be met



Am2914 Burn-in Circuit



INPUT/OUTPUT CIRCUITS

A MICROPROGRAMMABLE, BIPOLAR, LSI INTERRUPT STRUCTURE USING THE Am2914

INTRODUCTION

Advanced Micro Devices' introduction of the Am2914 Vector Priority Interrupt Controller now makes possible the structuring of a microprogrammable bipolar LSI interrupt system. The design engineer may use the Am2914 to simplify his design process, dramatically reduce the system cost, size and package count, and increase the speed, capability and reliability of his interrupt system.

The Am2914 is a modular, low cost, standard LSI component that may be microprogrammed to meet the requirements of specific applications. Today's engineer may utilize the Am2914 microprogrammability to provide functional flexibility and ease of engineering change, while taking advantage of its modularity to provide hardware regularity and future expansion capability.

THE INTERRUPT CONCEPT

In any state machine, a requirement exists for the efficient synchronization and response to asynchronous events such as power failure, machine malfunctions, control panel service requests, external timer signals, supervisory calls, program errors, and input/output device service requests. The merit of such an "asynchronous event handler" may be measured in terms of response time, system throughput, real time overhead, hardware cost and memory space required.

The simplest approach to asynchronous event handling is the poll approach. A status indicator is associated with each possible asynchronous event. The processor tests each indicator in sequence and, in effect, "asks" if service is required. This program-driven method is inefficient for a number of reasons. Much time is consumed polling when no service is required; programs must have frequent test points to poll indicators; and since indicators are polled in sequence, considerable time may elapse before the processor responds to an event. Thus, system throughput is low; real time overhead and response time are high, and a large memory space is required.

The interrupt method is a much more efficient way of servicing asynchronous requests. An asynchronous event requiring service generates an interrupt request signal to the processor. When the processor receives the interrupt request, it may suspend the program it is currently executing, execute an interrupt service routine which services the asynchronous request, then resume the execution of the suspended program. In this system, the execution of the service routine is initiated by an interrupt request; thus, the system is interrupt driven and service routines are executed only when service is requested. Although hardware cost may be higher in this type of system, it is more efficient since system throughput is higher, response time is faster, real time overhead is lower and less memory space is required.

INTERRUPT SYSTEM FUNCTIONAL DEFINITION

A complete and clear functional definition is key to the design of a good interrupt system. The following features are useful.

Hardware Modularity: Modular interrupt system hardware is beneficial in two ways. First, hardware modularity provides expansion capability. Additional modules may be added as the need to service additional requests arises. Secondly, hardware modularity provides a structural regularity which simplifies the system structure and also reduces the number of hardware part numbers.

Fast Interrupt System Response Time: Quick interrupt system response provides more efficient system operation. Fast response reduces real time overhead and increases overall system throughput.

INTERRUPT SYSTEM IMPLEMENTATION USING THE Am2914

The Am2914 provides all of the foregoing features on a single LSI chip. The Am2914 is a high-speed, eight-bit priority interrupt unit that is cascadeable to handle any number of priority interrupt request levels. The Am2914's high speed is ideal for use in Am2900 Family microcomputer designs, but it can also be used with the Am9080A MOS microprocessor.

The Am2914 receives interrupt requests on eight interrupt input lines (P₀-P₇). A LOW level is a request. An internal latch may be used to catch pulses (HIGH-LOW-HIGH) on these lines, or the latch may be bypassed so that the request lines drive the D inputs to the edge-triggered Interrupt Register directly. An eight-bit Mask Register is used to mask individual interrupts. Considerable flexibility is provided for controlling the Mask Register. Requests in the Interrupt Register (P₀-P₇) are ANDed with the corresponding bits in the mask register (M₀-M₇) and the results are sent to an eight-input priority encoder, which produces a three-bit encoded vector representing the highest priority input which is not masked.

An internal Status Register is used to point to the lowest priority at which an interrupt will be accepted. The contents of the Status Register are compared with the output of the

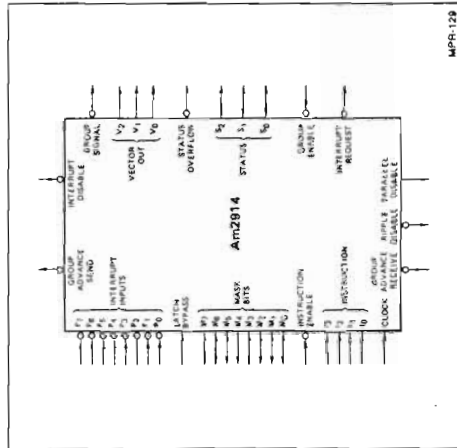


Figure 1. Am2914 Logic Symbol.

priority encoder, and an Interrupt Request output will occur if the vector is greater than or equal to the contents of the Status Register. Whenever a vector is read from the Am2914, the Status Register is automatically updated to point to one level higher than the vector read. (The Status Register can be loaded externally or read out at any time using the S-Bus.) Signals are provided for moving the status upward across devices (Group Advance Send and Group Advance Receive) and for inhibiting lower priorities from higher order devices (Ripple Disable, Parallel Disable, and Interrupt Disable). A Status Overflow output indicates that an interrupt has been read at the highest priority.

The Am2914 is controlled by a four-bit microinstruction field I₀-I₃. The microinstruction is executed if IE (Instruction Enable) is LOW and is ignored if IE is HIGH, allowing the four I bits to be shared with other functions. Sixteen different microinstructions are executed. Figure 2 shows the microinstructions and the microinstruction codes.

MICROINSTRUCTION DESCRIPTION	MICROINSTRUCTION CODE
MASTER CLEAR	0000
CLEAR ALL INTERRUPTS	0001
CLEAR INTERRUPTS FROM M-BUS	0010
CLEAR INTERRUPTS FROM MASK REGISTER	0011
CLEAR INTERRUPT, LAST VECTOR READ	0100
READ VECTOR	0101
READ STATUS REGISTER	0110
READ MASK REGISTER	0111
SET MASK REGISTER	1000
LOAD STATUS REGISTER	1001
BIT CLEAR MASK REGISTER	1010
BIT SET MASK REGISTER	1011
CLEAR MASK REGISTER	1100
DISABLE INTERRUPT REQUEST	1101
LOAD MASK REGISTER	1110
ENABLE INTERRUPT REQUEST	1111

Figure 2. Am2914 Microinstruction Set.

In this microinstruction set, the Master Clear microinstruction is selected as binary zero so that during a power up sequence, the microinstruction register in the microprogram control unit of the central processor can be cleared to all zeros. Thus, on the next clock cycle, the Am2914 will execute the Master Clear function. This includes clearing the Interrupt Latches and Register as well as the Mask Register and Status Register. The LGE flip-flop of the least significant group is set LOW because the Group Advance Receive input is tied LOW. All other Group Advance Receive inputs are tied to Group Advance Send outputs and these are forced HIGH during this instruction. This clear instruction also sets the Interrupt Request Enable flip-flop so that a fully interrupt driven system can be easily initiated from any interrupt.

Am2914 BLOCK DIAGRAM DESCRIPTION

The Am2914 block diagram is shown in Figure 3. The Microinstruction Decode circuitry decodes the Interrupt Microinstructions and generates required control signals for the chip. The Interrupt Register holds the Interrupt Inputs and is an eight-bit, edge triggered register which is set on the rising edge of the CP Clock signal if the Interrupt Input is LOW.

The Interrupt latches are set/reset latches. When the Latch Bypass signal is LOW, the latches are enabled and act as negative pulse catchers on the inputs to the Interrupt Register. When the Latch Bypass signal is HIGH, the Interrupt latches are transparent.

The Mask Register holds the eight mask bits associated with the eight interrupt levels. The register may be loaded from or read to the M-Bus. Also, the entire register or individual mask bits may be set or cleared.

The Interrupt Detect circuitry detects the presence of any unmasked interrupt input. The eight-input Priority Encoder determines the highest priority, non-masked interrupt input and forms a binary coded interrupt vector. Following a Vector Read, the three-bit Vector Hold Register holds the binary coded interrupt vector. This stored vector can be used later for clearing interrupts.

The three-bit Status Register holds the status bits and may be loaded from or read to the S-Bus. During a Vector Read, the incrementer increments the interrupt vector by one, and the result is clocked into the Status Register. Thus, the Status Register points to a level one greater than the vector just read. The three-bit Comparator compares the Interrupt Vector with the contents of the Status Register and indicates if the interrupt Vector is greater than or equal to the contents of the Status Register.

The Lowest Group Enabled Flip-Flop is used when a number of Am2914's are cascaded. In a cascaded system, only one Lowest Group Enabled Flip-Flop is LOW at a time. It indicates the eight interrupt group, which contains the lowest priority interrupt level which will be accepted and is used to form the higher order status bits.

The Interrupt Request and Group Enable logic contain various gating to generate the Interrupt Request, Parallel Disable, Ripple Disable, and Group Advance Send signals.

The Status Overflow signal is used to disable all interrupts. It indicates the highest priority interrupt vector has been read and the Status Register has overflowed.

The Clear Control logic generates the eight individual clear signals for the bits in the Interrupt Latches and Register. The Vector Clear Enable Flip-Flop indicates if the last vector read was from this chip. When it is set it enables the Clear Control Logic.

The CP clock signal is used to clock the Interrupt Register, Mask Register, Status Register, Vector Hold Register, and the Lowest Group Enabled, Vector Clear Enable and Status Overflow Flip-Flops, all on the clock LOW-to-HIGH transition.

The Am2914 can be microprogrammed in many different ways. Figure 4 shows an example interrupt sequence. The Read Vector microinstruction is necessary in order to read the interrupt priority level. Since vector plus one is automatically loaded into the Status Register when a Read Vector microinstruction is executed, the Status Register possibly will overflow and disable all interrupts. For this reason, the Status

Register outputs are enabled onto the Status Bus (S0-S7). The Status Bus is a three-bit, bi-directional, three-state bus. The Load Mask Register microinstruction loads data from the three-state, bi-directional M-Bus into the Mask Register. The Read Mask Register microinstruction enables the Mask Register outputs onto the bi-directional, three-state M-Bus.

The Set Mask Register microinstruction sets all the bits in the Mask Register to one. This results in all interrupts being inhibited.

The entire Mask Register is cleared by the Clear Mask Register microinstruction. This enables all interrupts subject to the Interrupt Enable flip-flop and the Status Register.

The Bit Clear Mask Register microinstruction may be used to selectively clear individual Mask Register bits. This microinstruction clears those Mask Register bits which have corresponding M-Bus bits equal to one. Mask Register bits with corresponding M-Bus bits equal to zero are not affected.

The Bit Set Mask Register microinstruction sets those Mask Register bits which have corresponding M-Bus bits equal to one. Other Mask Register bits are not affected.

All Interrupt Requests may be disabled by execution of the Disable Interrupt Request microinstruction. This microinstruction resets an Interrupt Request Enable flip-flop on the chip.

The Enable Interrupt Request microinstruction sets the Interrupt Enable flip-flop. Thus, Interrupt Requests are enabled subject to the contents of the Mask and Status Registers.

The Clear All Interrupts microinstruction clears the Interrupt Latches and Register.

The Clear Interrupts from Mask Register microinstruction clears those Interrupt Latches and Register bits which have corresponding Mask Register bits set equal to one. The M-Bus is used by the Am2914 during the execution of this microinstruction and must be floating.

The Clear Interrupts from M-Bus microinstruction clears those Interrupt Latches and Register bits which have corresponding M-Bus bits set equal to one.

The Clear Interrupt, Last Vector Read microinstruction clears the Interrupt Latch and Register bit associated with the last vector read.

The Read Vector microinstruction is used to read the vector value of the highest priority request causing the interrupt. The vector outputs are three-state drivers that are enabled onto the V0-V2 bus during this instruction. This microinstruction also automatically loads the value "vector plus one" into the Status Register. In addition, this instruction sets the Vector Clear Enable flip-flop and loads the current vector value into the Vector Hold Register so that this value can be used by the Clear Interrupt, Last Vector Read microinstruction. This allows the user to read the vector associated with the interrupt, and at some later time clear the Interrupt Latch and Register bit associated with the vector read.

The Load Status Register microinstruction loads S-Bus data into the Status Register and also loads the LGE flip-flop from the Group Enable input.

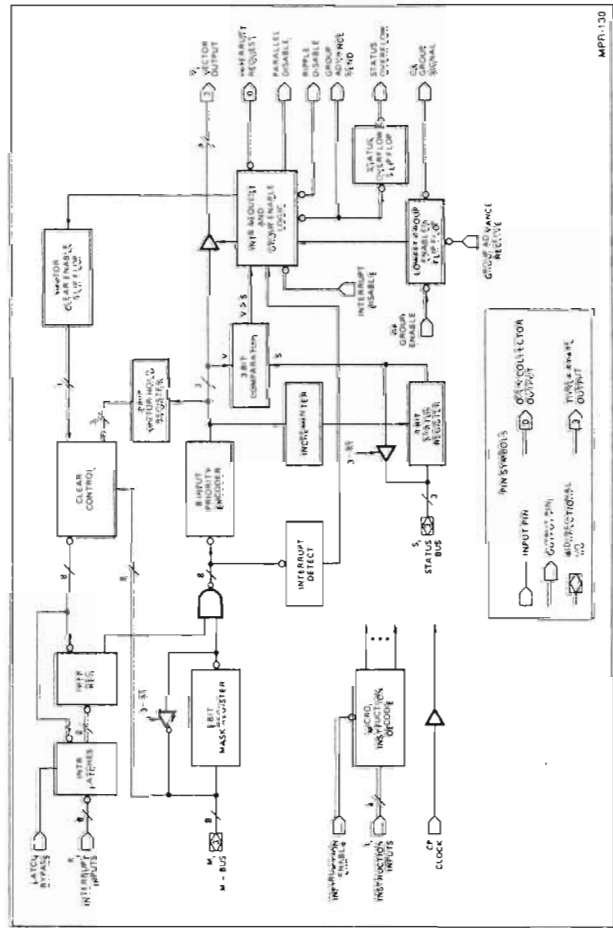


Figure 3. Am2914 Block Diagram.

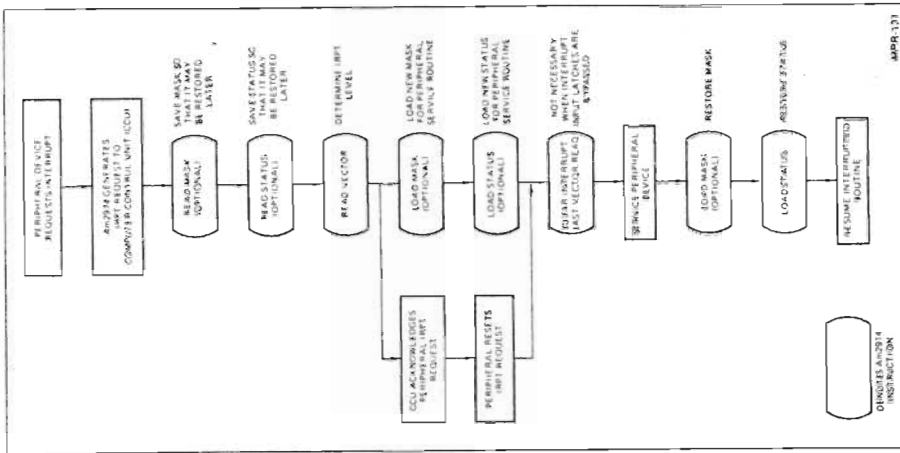


Figure 4. Example Interrupt Sequence.

Register must be reloaded periodically. The other Am2914 microinstructions are optional.

CASCADING THE Am2914

A number of input/output signals are provided for cascading the Am2914 Vectors Priority Interrupt Encoder. A definition of these I/O signals and their required connections follows:

Group Signal (GS) — This signal is the output of the Lowest Group Enabled Flip-flop and during a Read Status microinstruction is used to generate the high order bits of the Status word.

Group Enable (GE) — This signal is one of the inputs to the Lowest Group Enable flip-flop and is used to load the flip-flop during the Load Status microinstruction.

Group Advance Send (GAS) – During a Read Vector micro-instruction, this output signal is LOW when the highest priority vector (vector seven) of the group is being read. In a cascaded system Group Advance Send must be tied to the Group Advance Receive input of the next higher group in order to transfer status information.

Group Advance Receive (GAR) – During a Master Clear or Read Vector microinstruction, this input signal is used with other internal signals to load the Lowest Group Enabled flip-flop. The Group Advance Receive input of the lowest priority group must be tied to ground.

Status Overflow (SV) – This output signal becomes LOW after the highest priority vector (vector seven) of the group has been read and indicates the Status Register has overflowed. It stays LOW until a Master Clear or Load Status microinstruction is executed. The Status Overflow output of the highest priority group should be connected to the Interrupt Disable input of the same group and serves to disable all interrupts until new status is loaded or the system is master cleared. The Status Overflow outputs of lower priority groups should be left open (see Figure 7).

Interrupt Disable (ID) – When LOW, this input signal inhibits the Interrupt Request output from the chip and also generates a Ripple Disable output.

Ripple Disable (RD) – This output signal is used only in the Ripple Cascade Mode (see below). The Ripple Disable output is LOW when the Interrupt Disable input is LOW, the Lowest Group Enabled flip-flop is LOW, or an Interrupt Request is generated in the group. In the ripple cascade mode, the

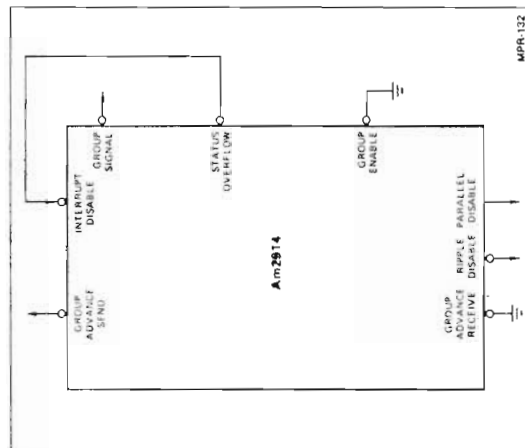


Figure 5. Cascade Lines Connection for Single Chip System.

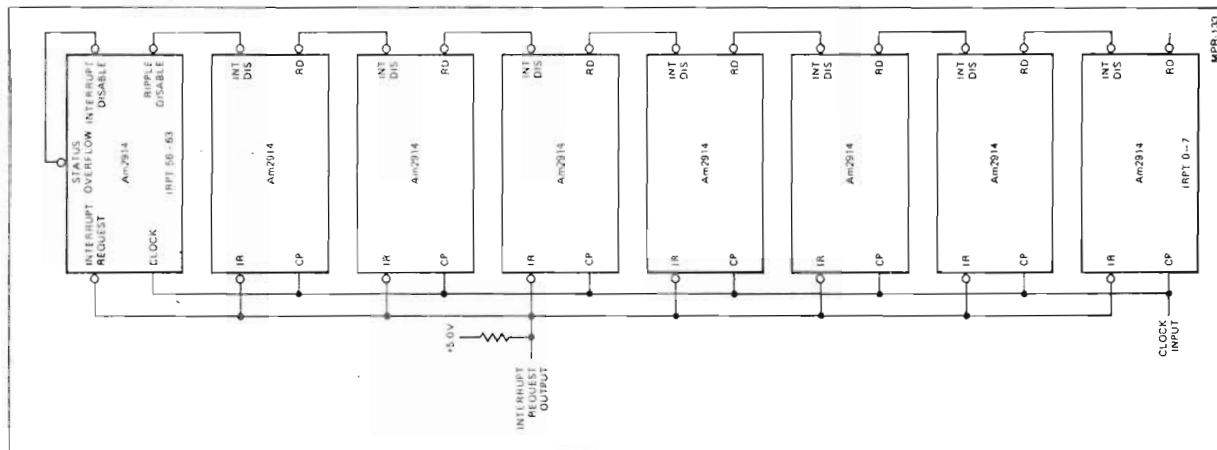


Figure 6. Interrupt Disable Connections for Ripple Cascade Mode.

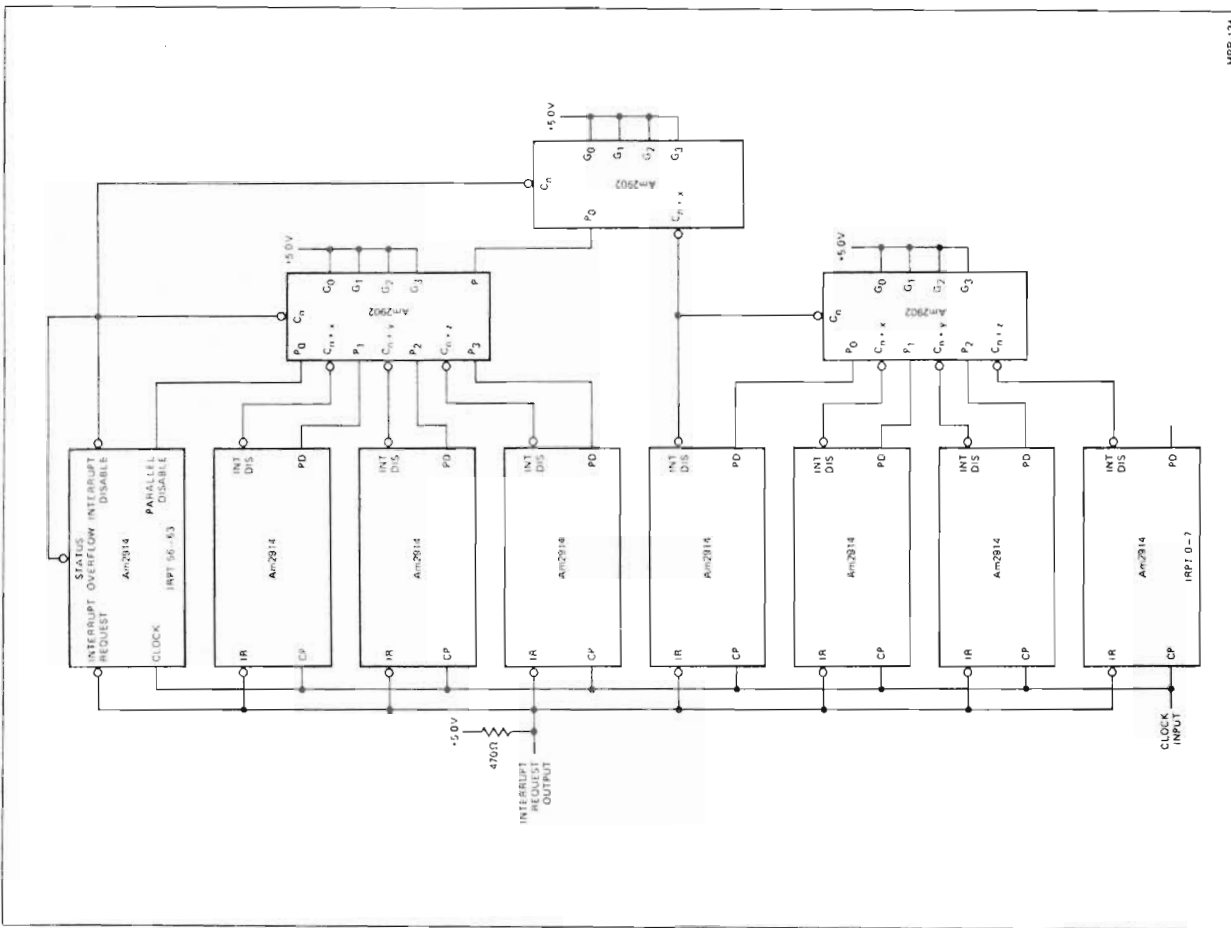


Figure 7. Interrupt Disable Connections for Parallel Cascade Mode.

Ripple Disable output is tied to the Interrupt Disable input of the next lower priority group (see Figure 6).

Parallel Disable (PD) — This output is used only in the parallel cascade mode (see below). It is HIGH when the Lowest Group Enabled flip-flop is LOW or an Interrupt Request is generated in the group. It is not affected by the Interrupt Disable input.

A single Am2914 chip may be used to prioritize and encode up to eight interrupt inputs. Figure 5 shows how the above cascade lines should be connected in such a single chip system. The Group Advance Receive and Group Enable inputs should be connected to ground so that the Lowest Group Enabled flip-flop is forced LOW during a Master Clear or Load Status microinstruction. Status Overflow should be connected to Interrupt Disable in order to disable interrupts when vector seven is read. The Group Advance Send, Ripple Disable, Group Signal and Parallel Disable pins should be left open.

The Am2914 may be cascaded in either a Ripple Cascade Mode or a Parallel Cascade Mode. In the Ripple Cascade Mode, the Interrupt Disable signal, which disables lower priority interrupts, is allowed to ripple through lower priority groups. Figures 6, 9 and 11 show the cascade connections required for a ripple cascade 64 input interrupt system.

In the parallel cascade mode, a parallel lookahead scheme is employed using the high-speed Am2902 Lookahead Carry Generator. Figures 7, 9 and 10 show the cascade connections required for a parallel cascade 64-input interrupt system. For this application, the Am2902 is used as a lookahead interrupt disable generator. A Parallel Disable output from any group results in the disabling of all lower priority groups in parallel. Figure 8 shows the Am2902 logic diagram and equations.

In Figures 9 and 10, the Am2913 Priority Interrupt Expander is shown forming the high order bits of the vector and status, respectively. The Am2913 is an eight-line to three-line priority encoder with three-state outputs which are enabled by the five output control signals G1, G2, G3, G4, and G5. In Figure 9, the Am2913 is connected so that its outputs are enabled during a Read Vector instruction, and in Figure 10 the Am2913 is connected so that its outputs are enabled during a Read Status instruction. The Am2913 logic diagram and truth table are shown in Figure 11.

The Am25LS138 three-line to eight-line Decoder also is shown in Figure 10. It is used to decode the three high order status bits during a Load Status instruction. The Am25LS138 logic diagram and truth table are shown in Figure 12.

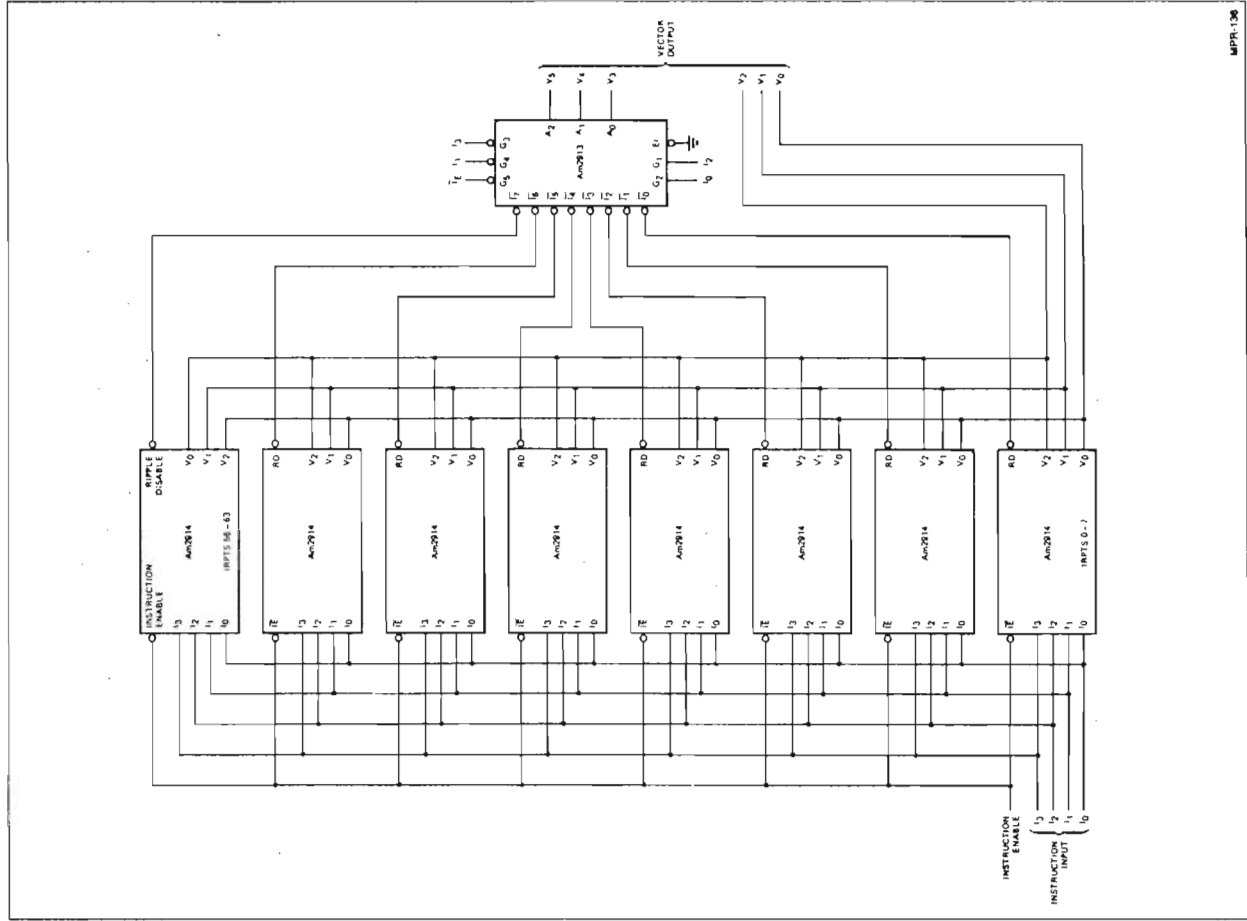


Figure 9. Vector Connections for both the Parallel and Ripple Cascade Modes.

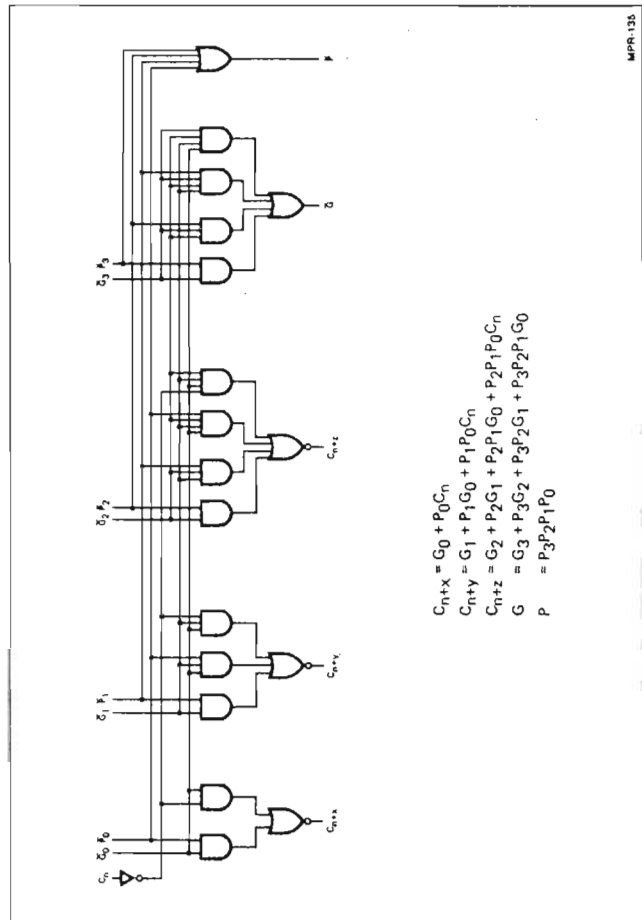


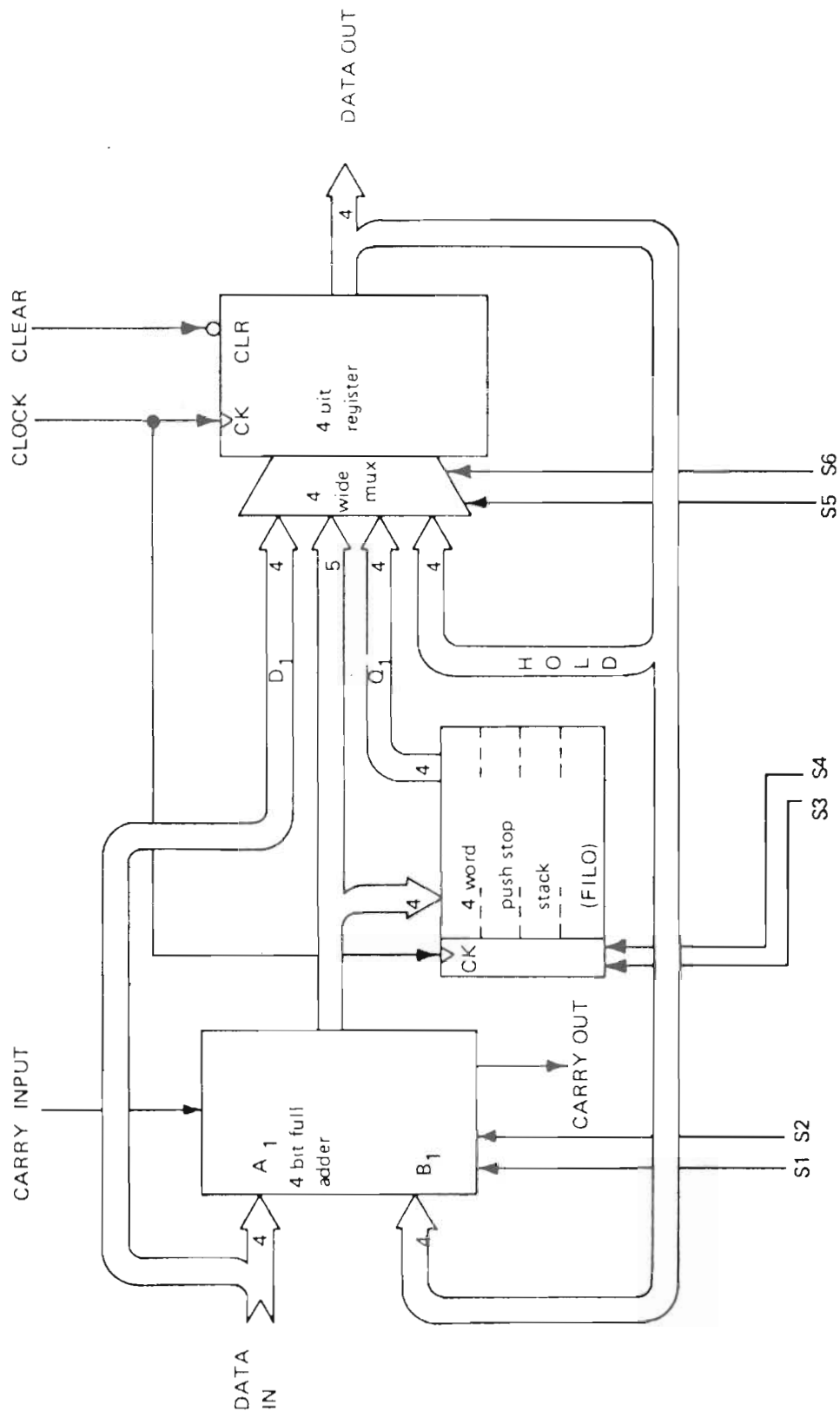
Figure 8. Am2902 Carry Look-Ahead Generator Logic Diagram and Equations.

APPENDIX D

FUNCTIONAL BLOCK DIAGRAM FOR 74S482
SEQUENCER

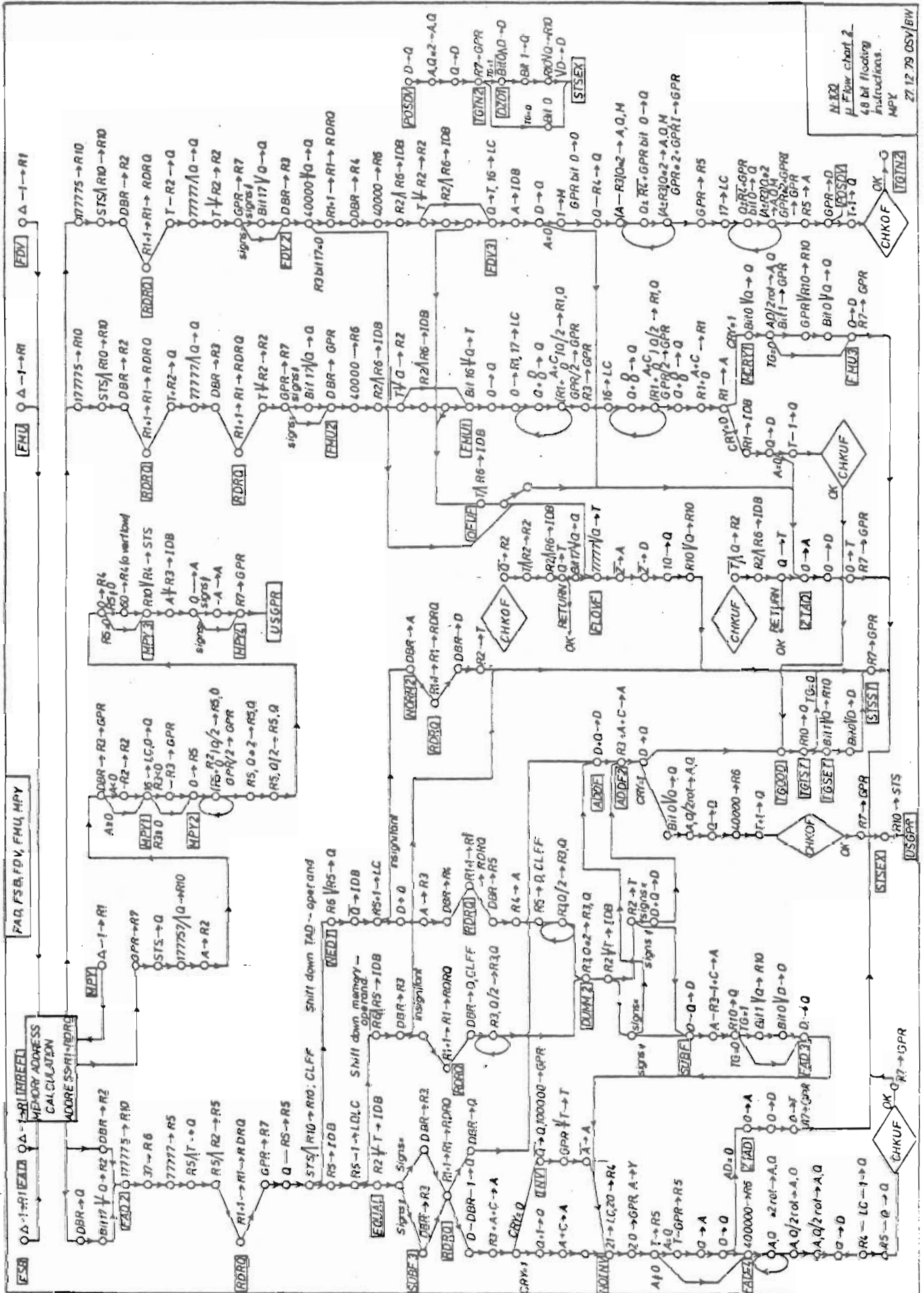
This appendix shows the functional block diagram for the 74S482 sequencer from Texas Instruments, Inc. More details will be found in "The TTL Data Book" issued by the same company.

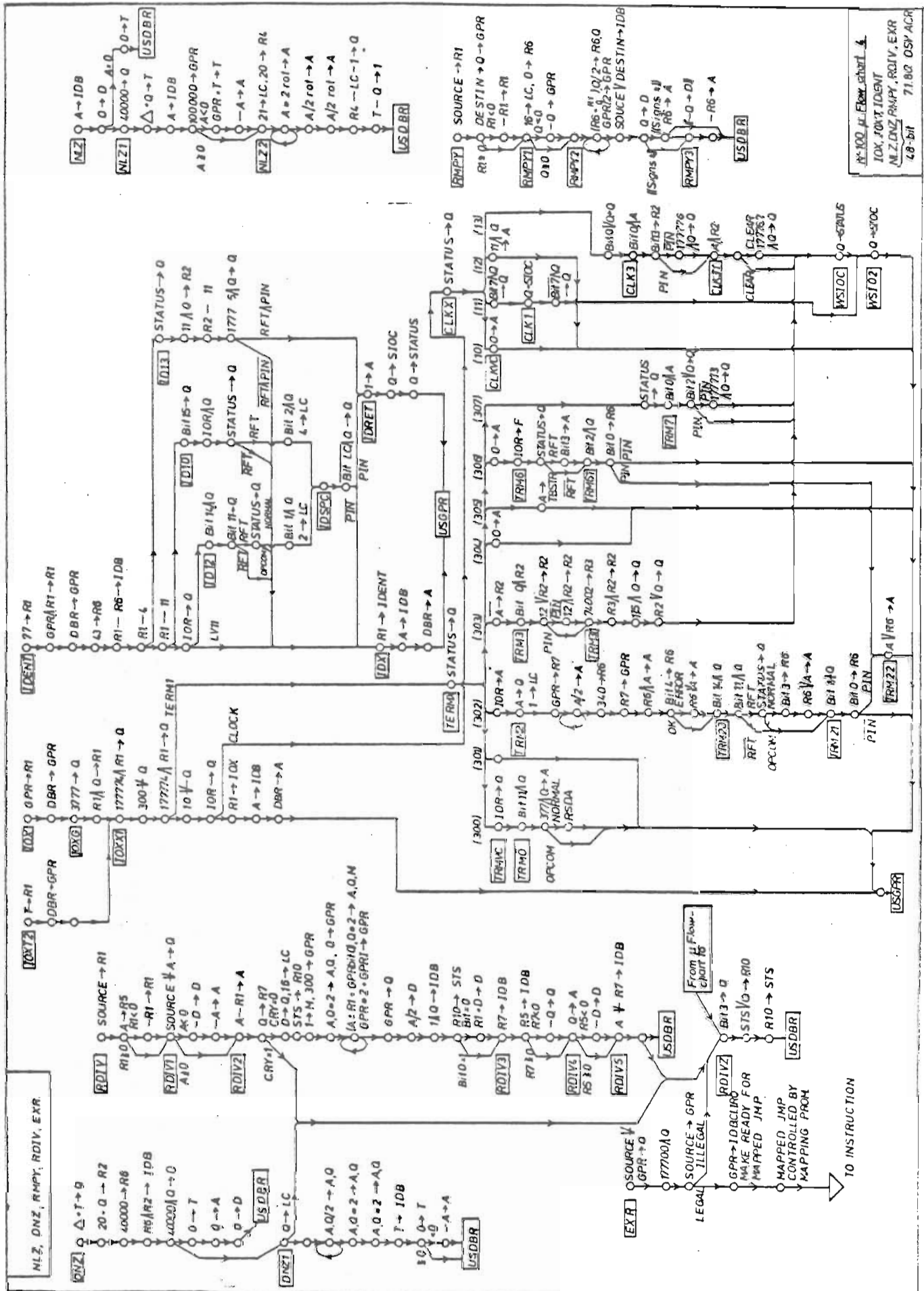
FUNCTIONAL BLOCK DIAGRAM



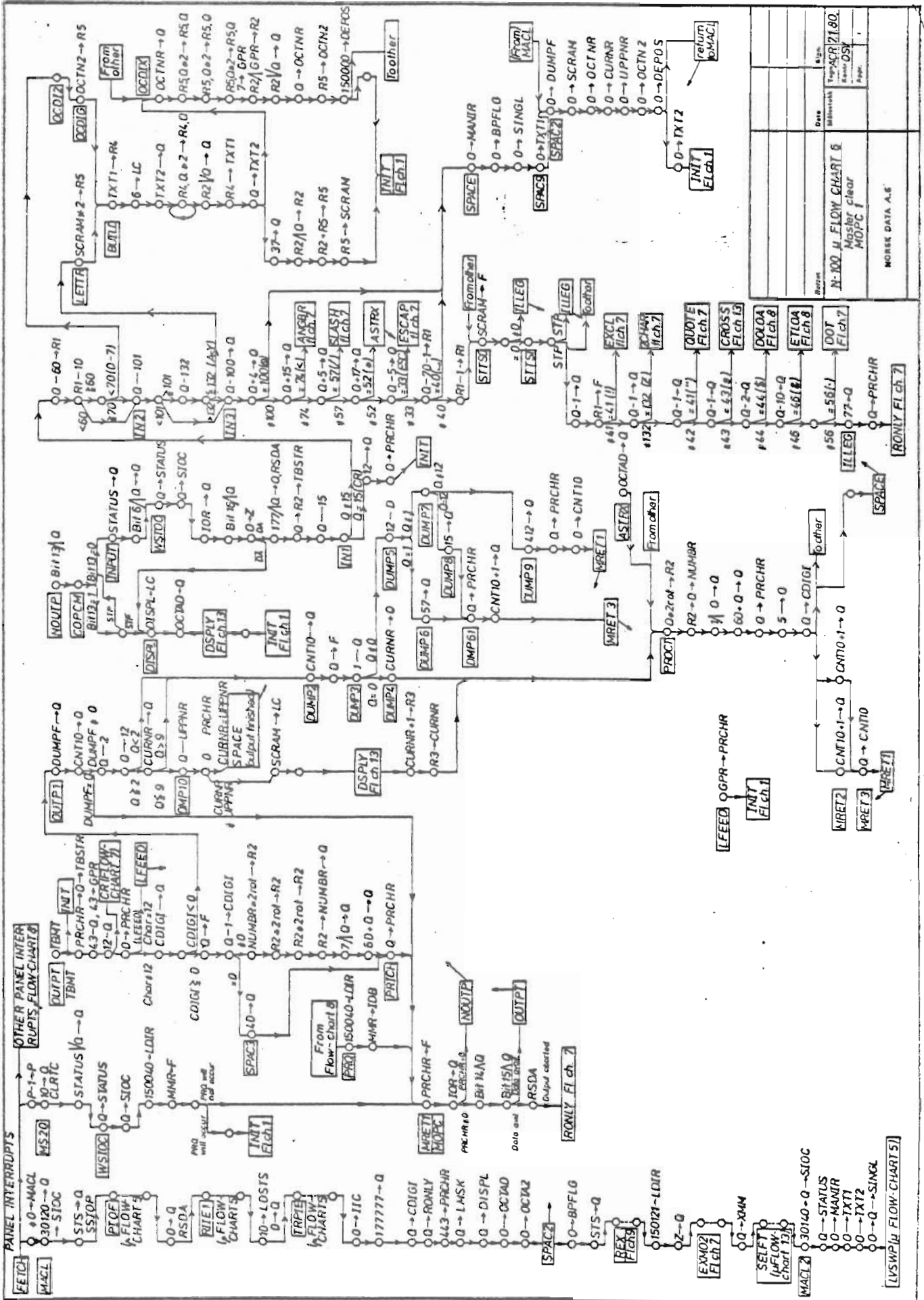
APPENDIX E

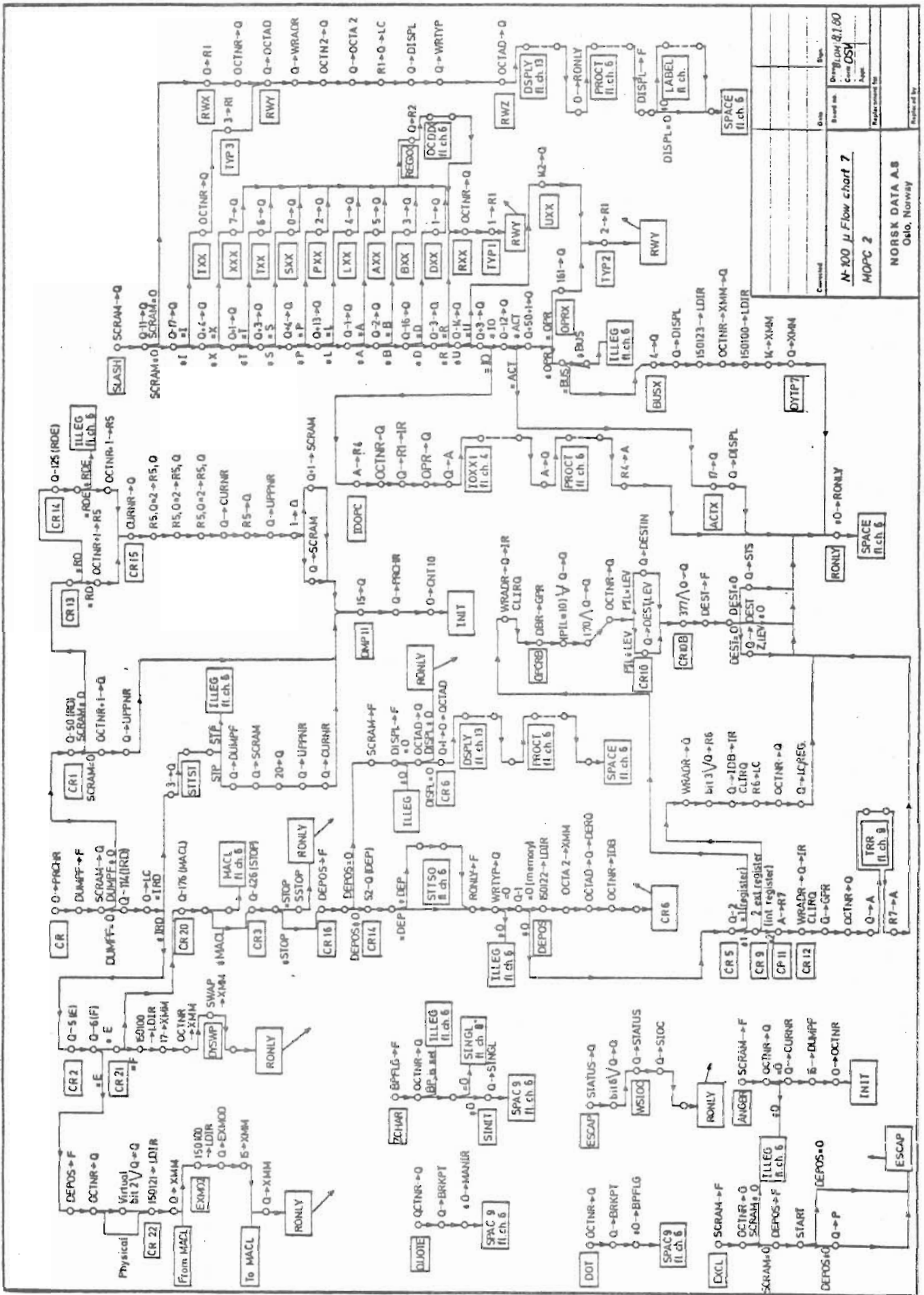
MICROPROGRAM FLOW CHARTS





R=100 μ s Flow chart 3
 IOX, JOX, IDENT
 NLZ, DNZ, RMPY, RDIV, EXR
 48-bit 71.80 OSYACR

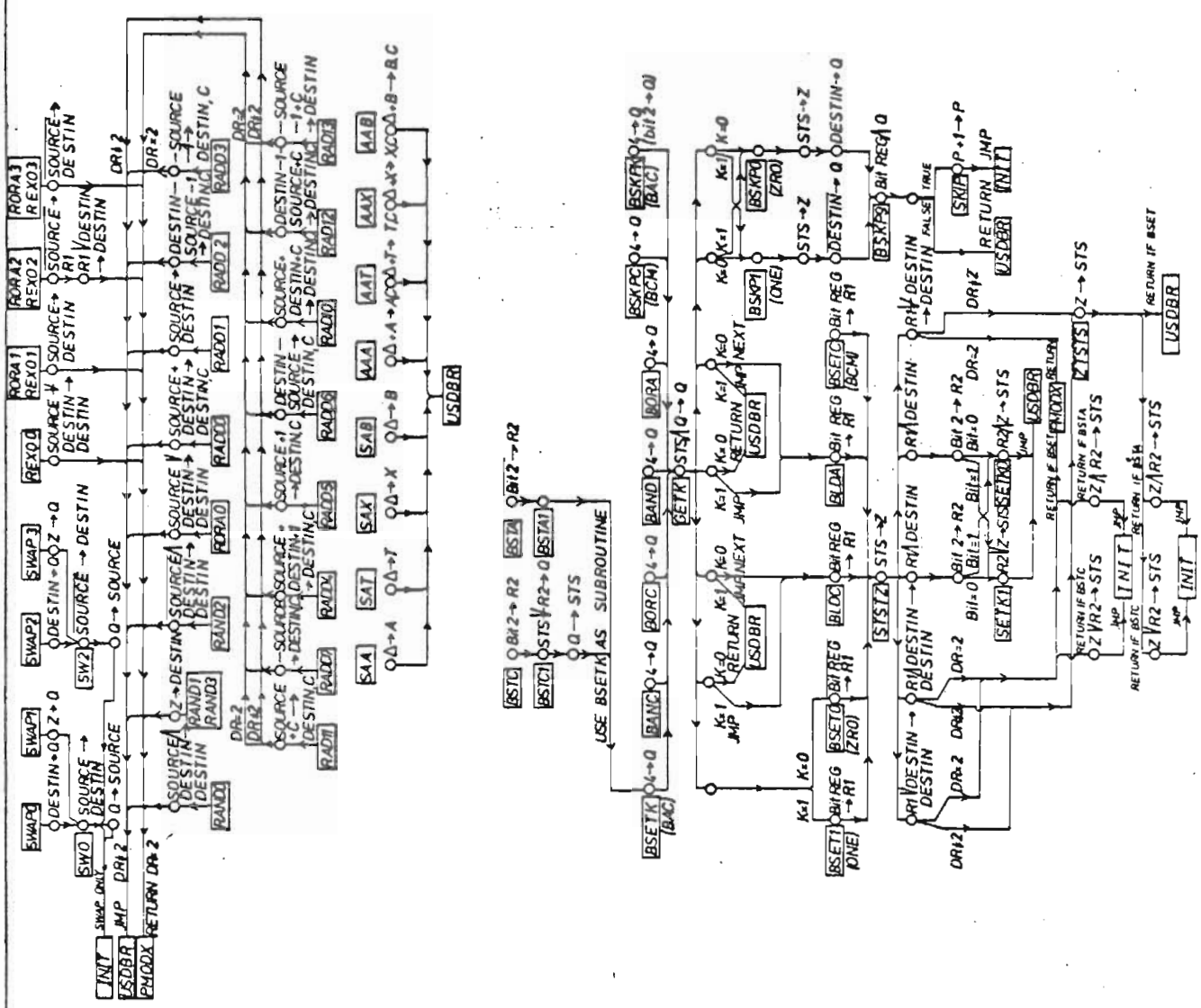


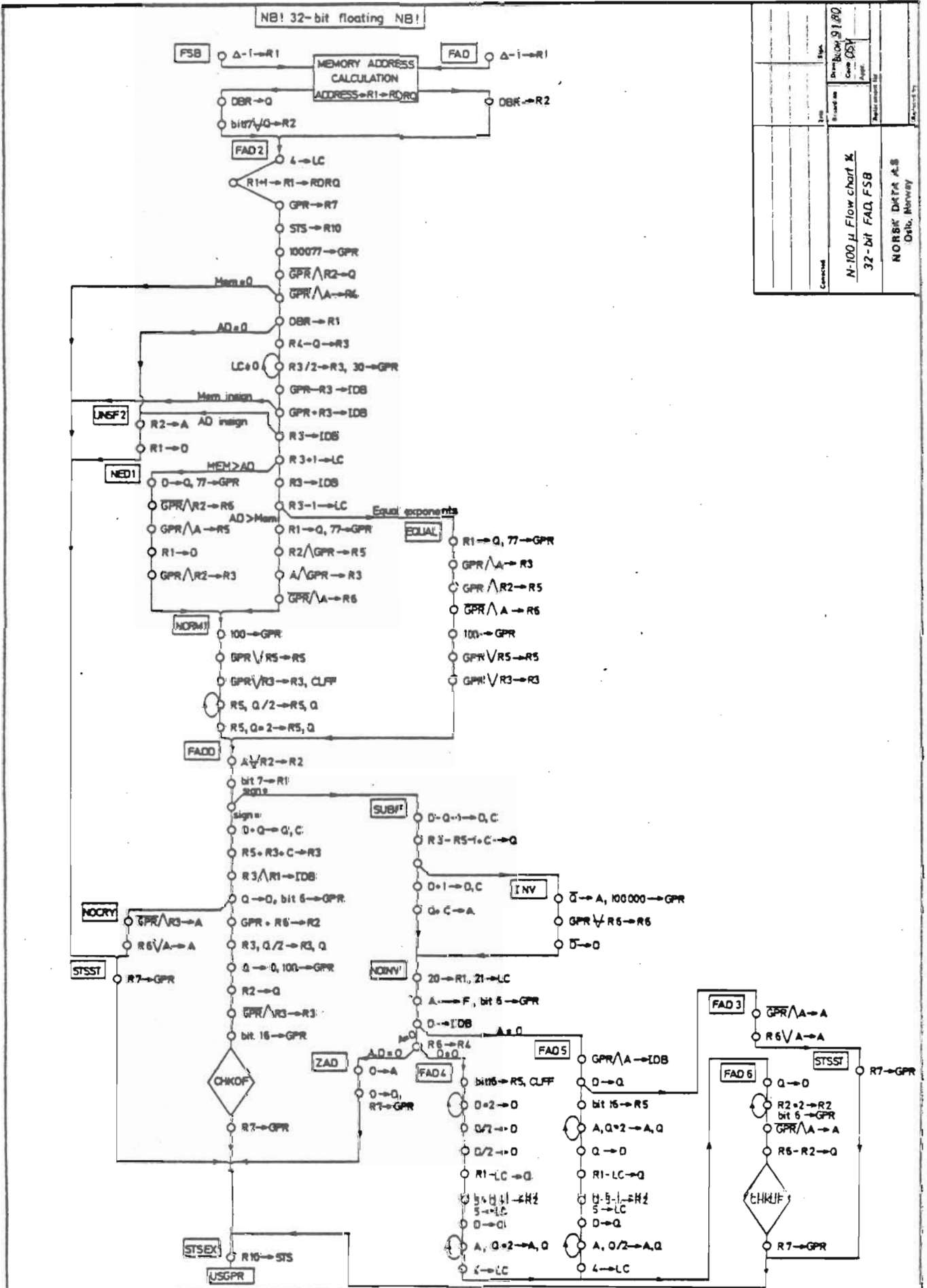


N-100 μ Flow chart 12

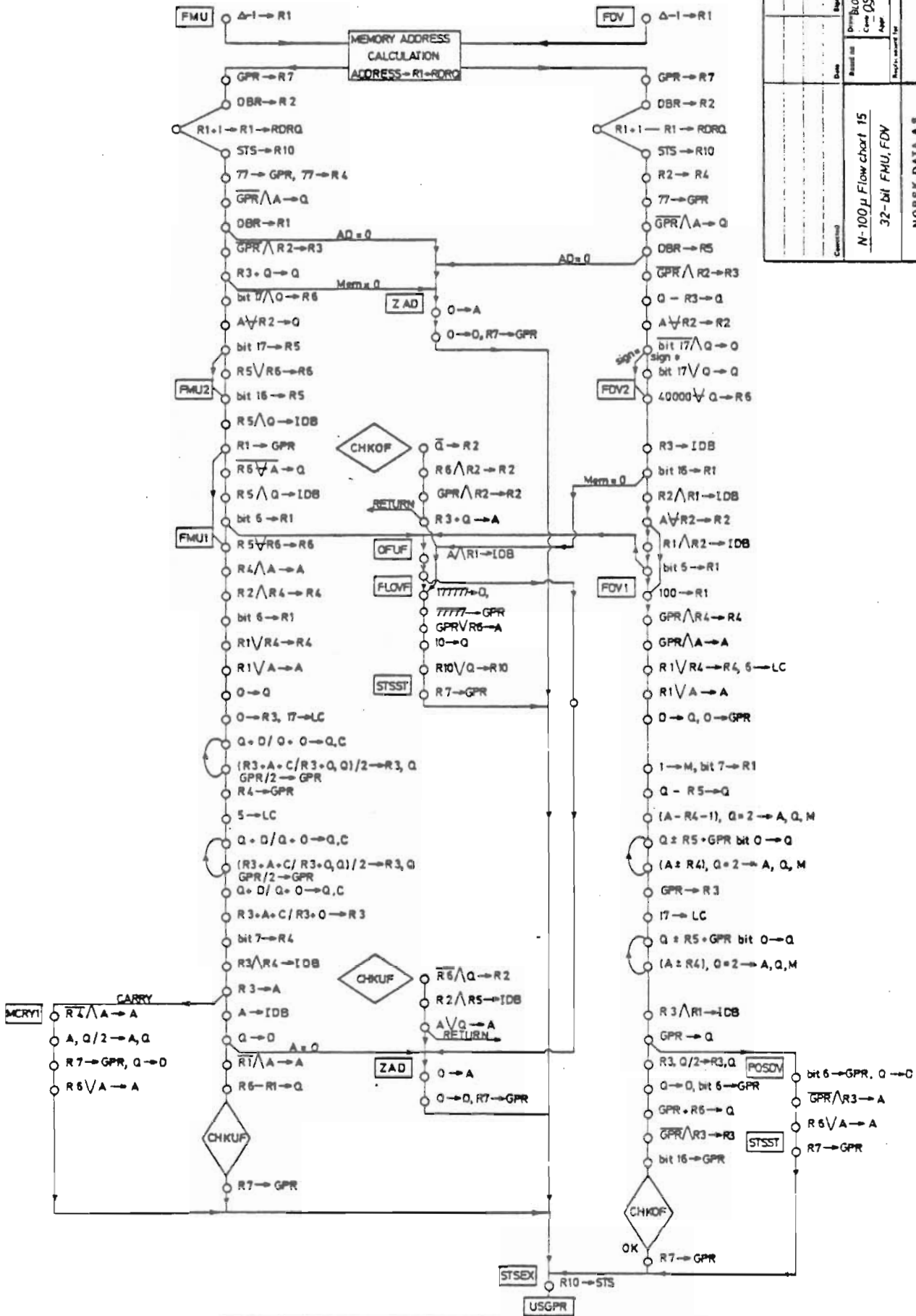
ROP instruction
ARG instruction
BOP instruction

Fig. 21.12.79 OSV/ACR.BW

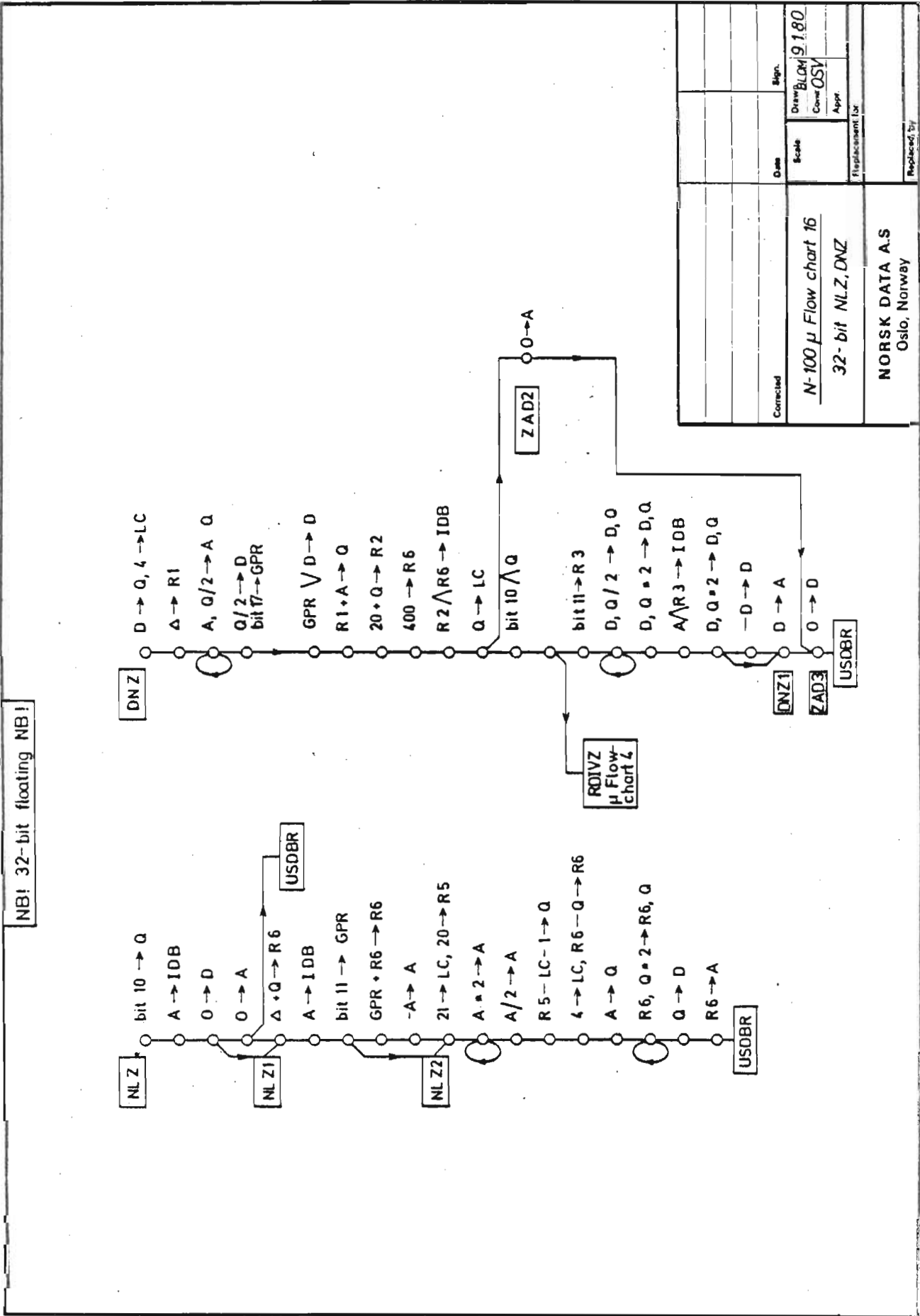




NB! 32-bit floating NB!



Page	15
Date	
Revised in	
Checked in	
Drawn	BLOM 9180
Case	OSY
Appr.	
Approved for	
Product No.	
General	N-100 μ Flow chart 15
	32-bit FMU, FDV
	NORSK DATA A.S.
	Ostlo, Norway

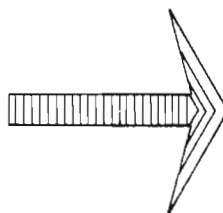
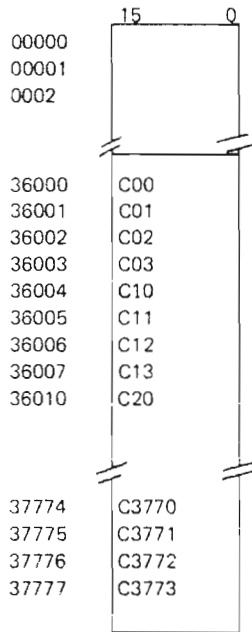


Corrected	Date	Sign.
	Scale	Drawn: BLON 9180
		Cover: OSY
		Appr.
Replacement for		
N-100 μ Flow chart 16		
32-bit NLZ, DNZ		
NORSK DATA A.S		
Oslo, Norway		
Replaced by		

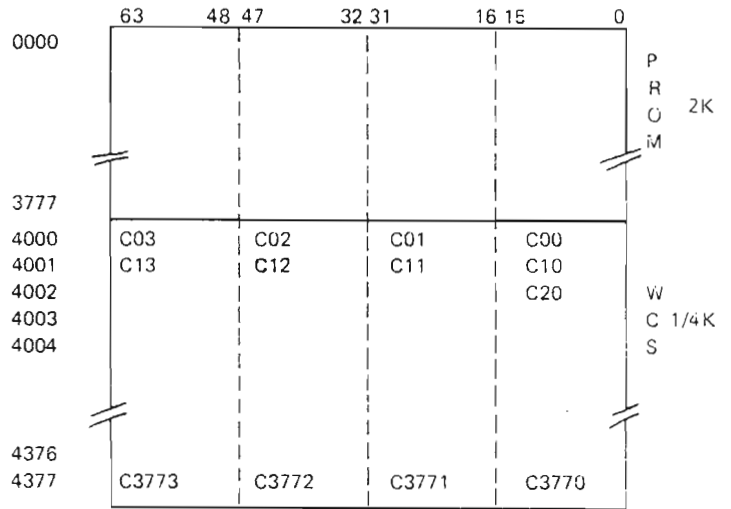
APPENDIX F

MAIN MEMORY/WCS CORRESPONDENCE

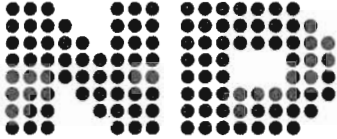
MAIN MEMORY (PHYSICAL)



WRITEABLE CONTROL STORE



6
3
r
.



NORSK DATA A.S
P.O. Box 4, Lindeberg gård
Oslo 10, Norway

COMMENT AND EVALUATION SHEET

NORD-100 Microprogramming Description
January 1980

ND-06.018.01

In order for this manual to develop to the point where it best suits your needs, we must have your comments, corrections, suggestions for additions, etc. Please write down your comments on this preaddressed form and mail it. Please be specific wherever possible.

FROM

.....

.....

– we make bits for the future

NORSK DATA A.S BOX 4 LINDEBERG GÅRD OSLO 10 NORWAY PHONE: 39 16 01 TELEX: 18661