



**Maintenance Library**

**5415**

**Processing Unit Models D25 and D26  
Theory-Maintenance Diagrams**

## PREFACE

This manual contains the theory of operations, maintenance diagrams, and service information for the IBM 5415 Processing Unit Models D25 and D26. All references to Model 15D mean Models 15 D25 and 15 D26.

The Models 15 D19 through 15 D24 have a maximum of 256K bytes of storage, while the Models 15 D25 and 15 D26 have 384K and 512K bytes of storage respectively. To handle the additional storage, the Models 15 D25 and 15 D26 have the following changes:

- Replacement of the > 64K and > 128K ADDR BITS toggle switches with an eight-position rotary switch to provide a > 256K selection.
- Addition of a > 256K PH latch and circuit.
- Addition of a 19th bit circuit to condition SAR bit E13.
- Replacement of the roller drum to include an I/O > 256K position.

The Model D processing unit includes fast I cycle processing, a maximum of 512K bytes of main storage, and 3340/3344 disk drives. Refer to SY31-0367 for information about 5415 Models A and B, SY31-0417 for information about 5415 Model C, or SY31-0464 for information about 5415 Models D19 through D24.

A 16-bit ALU was added to the Model D CPU for fast I cycle processing and is referred to in this manual as the auxiliary (aux) ALU. Wherever just ALU is used, it refers to the original Model 15 ALU.

This manual is intended for use by GSD customer engineers for use in the classroom and for recall when servicing in a free-lance mode.

The manual gives an explanation of the logical circuit functions and major objectives. With this information, the CE can interpret the operation of circuits illustrated in the companion diagrams.

Machine operations are presented in operational flowcharts, most of which are *two-level*. The general flow path indicated by the heavy line of the two-level charts shows the major objectives of an operation or instruction. Detailed flow paths of major objectives are to the right of the general flow path.

Positive-logic diagrams support the operational flowcharts. They show logical circuit operation without regard to signal levels. Most of the logic diagrams in this manual are not block for block representations of ALDs (automated logic diagrams). Rather, only blocks necessary for a logical understanding of the operation are shown.

For machine characteristics and installation instructions, refer to the *IBM System/3 Installation Manual—Physical Planning, GA21-9084*.

### First Edition (November 1977)

Changes are continually made to the specifications herein; any such change will be reported in subsequent revisions or technical newsletters.

Address comments concerning the contents of this publication to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901. Comments become the property of IBM.

## SAFETY

### Personal Safety

Read and follow the safety suggestions in the *CE Safety Practices Manual, S229-1264*, a pocket-sized card issued to all IBM customer engineers.

Remember:

- Loose clothing can become entangled in moving parts of the machine.
- Drive belts, because of their internal cable construction, can cause serious injury. DO NOT crank a machine by pulling on the drive belts.
- Heat sinks are at an electrical potential. DO NOT short heat sinks to each other or to the machine side frame.
- Always unplug machine power and wait ONE FULL MINUTE before attempting repairs or adjustments in the power supply area.
- Voltages developed in the resonant circuit of regulating power supplies are apt to be much greater than the line voltages.
- Follow the specific safety precautions that accompany many of the adjustment procedures in this manual.

Be aware that an I/O device motor and/or clutches can operate unexpectedly. Conditions that could cause this are:

- Program commands.
- Loss of dc voltage to a machine, gate, board or chassis, card, or pin.
- Removing or inserting a card or cable.
- Probing and accidentally shorting a pin.

### Equipment Safety

#### Electrical

Always replace blown fuses with fuses of the same type and rating. Using fuses of a different type or higher rating could result in component damage.

Remove power from the machine before replacing logic cards, magnets, or solenoids. Failure to do this could result in damage to the card being replaced or to other cards in the net.

#### Mechanical

Do not operate the machine under power with units disassembled, removed, or maladjusted. Keep tools clear of the mechanism when the machine is operating under power.

**CAUTION:** Do not use IBM cleaning fluid on plastic parts.

# CONTENTS

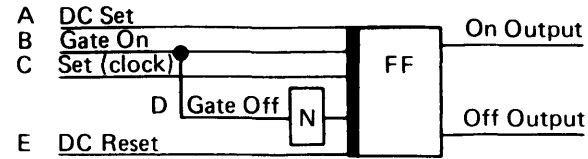
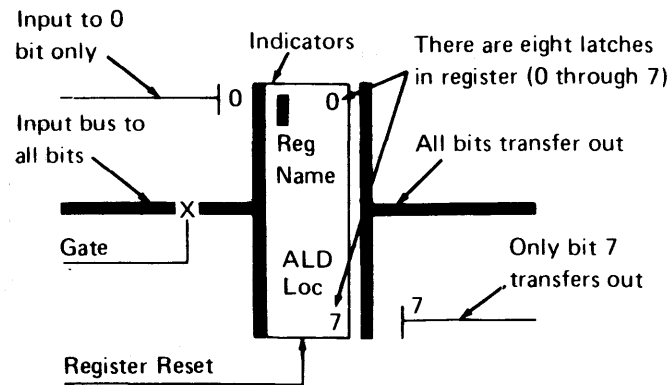
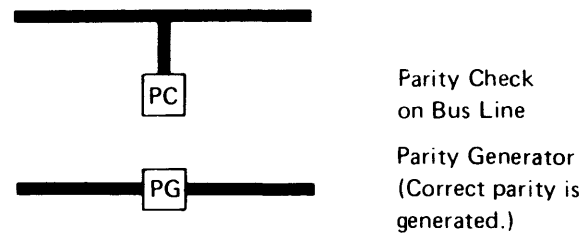
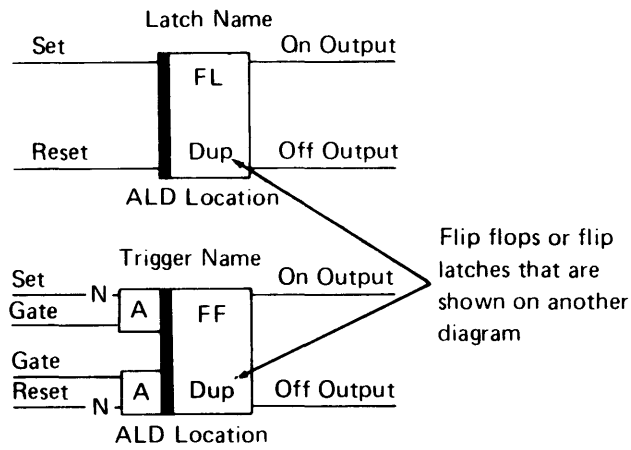
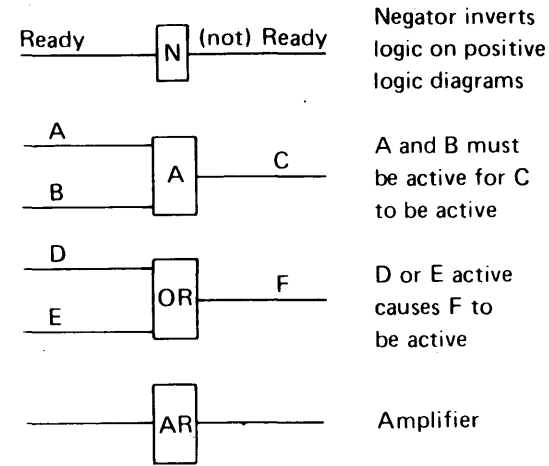
PREFACE . . . . .	ii	Number Systems . . . . .	1-9	AAR (A-Address Register) . . . . .	2-30	BSM READ OPERATION . . . . .	3-5
SAFETY . . . . .	iii	Decimal Number System . . . . .	1-9	BAR (B-Address Register) . . . . .	2-30	BYTE CONTROL AND CHECK BIT GENERATION	3-6
Personal Safety . . . . .	iii	Binary Number System . . . . .	1-9	Index Register 1 and Index Register 2 . . . . .	2-30	Store Data Parity Check . . . . .	3-6
Equipment Safety . . . . .	iii	Hexadecimal Number System . . . . .	1-9	ARR (Address Recall Register) . . . . .	2-30	BSM . . . . .	3-7
Electrical . . . . .	iii	Number System Conversions . . . . .	1-10	LCR (Length Count Register) . . . . .	2-30	Controls . . . . .	3-7
Mechanical . . . . .	iii	Hexadecimal to Decimal . . . . .	1-10	DRR (Data Recall Register) . . . . .	2-30	Addressing . . . . .	3-7
CONTENTS . . . . .	iv	Decimal to Hexadecimal . . . . .	1-10	PSR (Program Status Register) . . . . .	2-30	Timing . . . . .	3-7
LEGEND . . . . .	vii	Hexadecimal to Binary and Binary to Hexadecimal	1-10	MRDAR (MFCU Read Data Address Register) . . . . .	2-30	FETCH DATA REGISTER AND CONTROL . . . . .	3-9
ABBREVIATIONS . . . . .	viii	Data Formats . . . . .	1-10	MPCAR (MFCU Punch Data Address Register) . . . . .	2-30	ERROR CHECKING AND CORRECTION . . . . .	3-10
<b>INTRODUCTION . . . . .</b>	<b>1-1</b>	Instruction Formats . . . . .	1-11	MPTAR (MFCU Print Data Address Register) . . . . .	2-30	Fetch Check Bit Generator . . . . .	3-10
IBM SYSTEM/3 MODEL 15D . . . . .	1-1	Operation Code . . . . .	1-11	LPDAR (Line Printer Data Address Register) . . . . .	2-30	Syndrome Bit Generator . . . . .	3-10
System Control Programming . . . . .	1-2	Q-Code . . . . .	1-11	LPIAR (Line Printer Image Address Register) . . . . .	2-30	Syndrome Bit Decoder . . . . .	3-10
Multiprogramming . . . . .	1-2	Control Code . . . . .	1-11	Processing Unit LSRs . . . . .	2-31	Invert Data Switch . . . . .	3-11
I/O Devices . . . . .	1-2	Storage Addresses . . . . .	1-11	I/O LSRs . . . . .	2-32	Error Classification Logic . . . . .	3-11
3277 Display Station/Keyboard . . . . .	1-2	Addressing . . . . .	1-12	LSR Addressing . . . . .	2-33	Correctable Error . . . . .	3-11
IBM 3284 Printer . . . . .	1-2	Direct Addressing . . . . .	1-12	CPU LSR Select . . . . .	2-34	Uncorrectable Error . . . . .	3-11
IBM 2501 Card Reader . . . . .	1-2	Indexed Addressing . . . . .	1-12	LSR Select (I/O Channel Bank 1) . . . . .	2-35	Diagnostic Mode . . . . .	3-11
IBM 2560 Multi-Function Card Machine . . . . .	1-3	Instruction and Execute Cycle . . . . .	1-12	LSR Select (I/O Channel Bank 2) . . . . .	2-36	MEMORY OUTPUT (MOP) . . . . .	3-12
IBM 3340 Direct Access Storage Facility . . . . .	1-3	Sequential Instruction Execution . . . . .	1-12	LSR Select (I/O Channel Bank 3) . . . . .	2-37	SDBI/SDBO PROCESSOR CHECKS . . . . .	3-13
IBM 3741 Data Station Models 1 and 2 IBM 3741		Branching . . . . .	1-13	I/O LSR Address Bit Generation . . . . .	2-38	MAIN STORAGE MAINTENANCE . . . . .	3-14
Programmable Work Station Models 3 and 4 . . . . .	1-3	Interrupt Routines . . . . .	1-13	LSR Controls . . . . .	2-39	<b>I/O INTERFACE . . . . .</b>	<b>4-005</b>
IBM 3344 Direct Access Storage . . . . .	1-3	I/O Cycle Steal . . . . .	1-13	I/O LSR Select Check . . . . .	2-40	CHANNEL CABLING (CPU) . . . . .	4-010
IBM 5424 Multi-Function Card Unit . . . . .	1-4	<b>PROCESSOR FUNCTIONAL UNITS . . . . .</b>	<b>2-1</b>	OP REGISTER . . . . .	2-41	DBO TRANSLATOR (5424) . . . . .	4-095
IBM 1442 Card Read Punch . . . . .	1-4	CPU CLOCK . . . . .	2-1	PRIVILEGED MODE/MODEL 10 COMPATIBILITY	2-42	DBI TRANSLATOR (5424) . . . . .	4-095
IBM 3410 Magnetic Tape Unit . . . . .	1-4	CYCLE CONTROLS . . . . .	2-2	Privileged Mode . . . . .	2-42	I/O INTERFACE LINES . . . . .	4-100
IBM 1403 Printer Model 2, Model 5, and Model		RUN CONTROLS . . . . .	2-4	Model 10 Compatibility . . . . .	2-42	DBO AND TRANSLATOR . . . . .	4-107
N1 . . . . .	1-5	ALU . . . . .	2-8	Invalid Op Code and Privileged Op Check . . . . .	2-43	CONSOLE SWITCHES . . . . .	4-110
SIOC . . . . .	1-5	AND/OR and Test False . . . . .	2-8	Op and Q Register Parity . . . . .	2-45	DBI AND TRANSLATOR . . . . .	4-115
BSCA . . . . .	1-5	ALU DATA FLOW . . . . .	2-9	PROGRAM CHECK REGISTERS . . . . .	2-46	CHANNEL DBO PARITY, INVALID DEVICE	
Display Adapter . . . . .	1-5	Binary Subtraction . . . . .	2-10	PROGRAM CHECK ASSEMBLER, STORE DATA		ADDRESS . . . . .	4-120
LCA (Local Communications Adapter) . . . . .	1-5	Binary Addition . . . . .	2-11	ASSEMBLER . . . . .	2-48	DBO PARITY . . . . .	4-125
MLTA (RPQ) . . . . .	1-5	Decimal Subtraction . . . . .	2-11	INTERRUPTS . . . . .	2-49	DBI PARITY . . . . .	4-130
BSCC (Binary Synchronous Communications		ALU P Bit Generation . . . . .	2-12	Interrupt Mask . . . . .	2-49	<b>CPU INSTRUCTIONS . . . . .</b>	<b>5-1</b>
Controller) . . . . .	1-5	Decimal Addition . . . . .	2-13	Supervisor Program (Interrupt Level 0) . . . . .	2-50	INSTRUCTIONS AND SIGNAL REFERENCE . . . . .	5-1
Channel Terminate Feature . . . . .	1-5	Recomplement . . . . .	2-14	Op End Interrupt (Interrupt Level 5) . . . . .	2-50	INSTRUCTION CYCLES (FAST I-CYCLE) . . . . .	5-2
System Maintenance . . . . .	1-5	ALU CONTROLS . . . . .	2-15	Program Check Interrupt (Interrupt Level 7) . . . . .	2-50	TWO-ADDRESS INSTRUCTIONS . . . . .	5-3
Scheduled Maintenance . . . . .	1-5	CHECK ALU . . . . .	2-17	INTERVAL TIMER . . . . .	2-52	I-Cycles . . . . .	5-3
5415 PROCESSING UNIT (CPU) . . . . .	1-6	Parity Generation and Parity Check . . . . .	2-17	Load I/O . . . . .	2-52	I-Op Cycle . . . . .	5-3
Introduction . . . . .	1-6	Carry Check . . . . .	2-17	Start I/O . . . . .	2-52	I-Q Cycle . . . . .	5-3
Fast I-Cycles . . . . .	1-6	ALU Parity Checking . . . . .	2-18	Sense I/O . . . . .	2-52	I-Op and I-Q Cycles . . . . .	5-4
Data Flow . . . . .	1-7	Carry Check . . . . .	2-19	MAIN STORAGE ADDRESSING . . . . .	2-54	I-H and I-L Cycles . . . . .	5-7
I-Cycle Operation . . . . .	1-7	A-REGISTER AND CONTROLS . . . . .	2-20	I/O LSR 17th, 18th, and 19th Bits . . . . .	2-55	I-H2 and I-L2 Cycles . . . . .	5-7
E Cycle Operation . . . . .	1-7	A-Register . . . . .	2-20	Storage Protect . . . . .	2-56	Indexing (I-X Cycles) . . . . .	5-10
CPU Timing . . . . .	1-7	B REGISTER AND CONTROLS . . . . .	2-21	Address Parity Checks . . . . .	2-57	Execution Cycles . . . . .	5-14
Parity Checking and Generation . . . . .	1-7	A/B REGISTER PARITY CHECKING . . . . .	2-22	Invalid Address/Storage Protect Check . . . . .	2-58	A-Cycle . . . . .	5-14
Unit Check . . . . .	1-7	AUXILIARY ALU . . . . .	2-23	Unit Record Restart . . . . .	2-59	B-Cycle . . . . .	5-14
Errors . . . . .	1-7	AUXILIARY ALU CARD CHECK . . . . .	2-24	<b>MAIN STORAGE . . . . .</b>	<b>3-1</b>	Add Logical Characters—ALC . . . . .	5-15
Processor Data Flow . . . . .	1-8	AUXILIARY B REGISTER . . . . .	2-25	One Byte Fetch . . . . .	3-1	Subtract Logical Characters—SLC . . . . .	5-15
Machine Language . . . . .	1-9	CONDITION REGISTER (CR) . . . . .	2-26	One Byte Store . . . . .	3-1	Compare Logical Characters—CLC . . . . .	5-16
		Condition Register Settings . . . . .	2-26	Dual Byte Fetch . . . . .	3-1	Move Characters—MVC . . . . .	5-16
		PROGRAM MODE REGISTERS (PMR) . . . . .	2-28	Dual Byte Store . . . . .	3-1	Move Characters or Compare, Add, or Subtract	
		Byte 1 . . . . .	2-28	MAIN STORAGE LOCATIONS . . . . .	3-2	Logical Characters . . . . .	5-17
		Byte 2 . . . . .	2-28	DATA FLOW . . . . .	3-3	Add or Subtract Zoned Decimal—AZ—SZ . . . . .	5-20
		LOCAL STORAGE REGISTERS (LSR) . . . . .	2-30	BSM WRITE OPERATION . . . . .	3-4	Recomplement Cycles . . . . .	5-21
		IAR (Instruction Address Register) . . . . .	2-30				



Recomplementing . . . . .	5-21	Alter Storage . . . . .	6-5	CE CONTROLS . . . . .	7-7	SUPPLY NO. 3 (Lower Boards)	8-12
Zero and Add Zoned—ZAZ . . . . .	5-21	DISPLAY STORAGE . . . . .	6-8	ADDRESS/DATA Switches (Below Console		Voltage Adjustment . . . . .	8-12
Zero and Add Zoned and Add or Subtract Zoned		Objectives . . . . .	6-8	Display) (ALD PA111) . . . . .	7-7	End View of -4V Regulator Card . . . . .	8-12
Decimal . . . . .	5-22	ALTER ATT/PMR . . . . .	6-12	SYSTEM RESET Key (ALD PC101) . . . . .	7-7	+6 VOLT LOGIC SUPPLY . . . . .	8-13
Edit—ED . . . . .	5-26	Objectives . . . . .	6-12	CE Key Switch (ALD PC111) . . . . .	7-7	+6 Vdc Regulator . . . . .	8-13
Insert and Test Character—ITC . . . . .	5-30	ALTER SAR . . . . .	6-14	CHECK RESET Key (ALD PC101) . . . . .	7-7	+6V Expansion Bulk Supply (Feature) . . . . .	8-13
Move Hex Character—MVX . . . . .	5-34	Objectives . . . . .	6-14	BSCA STEP Key (ALD HE160) . . . . .	7-7	Voltage Adjustment . . . . .	8-13
ONE ADDRESS INSTRUCTIONS . . . . .	5-38	DISPLAY SAR/MSAR . . . . .	6-18	ADDRESS COMPARE Light (ALD PB121) . . . . .	7-7	End View of +6V Regulator Card . . . . .	8-13
I-Cycles . . . . .	5-38	<b>SYSTEM CONSOLE</b> . . . . .	<b>7-1</b>	I/O CHECK Light (ALD PB121) . . . . .	7-7	MAIN STORAGE POWER SUPPLY . . . . .	8-14
Move Logical Immediate—MVI . . . . .	5-38	OPERATOR CONTROLS . . . . .	7-1	LSR DISPLAY SELECTOR (ALD PC111) . . . . .	7-7	Voltage Adjustments . . . . .	8-14
Compare Logical Immediate—CLI . . . . .	5-38	The Emergency Power-Off (EPO) Pull Switch (ALD		CE MODE SELECTOR (ALD PA101) . . . . .	7-8	+3.4 Vdc . . . . .	8-14
Move Logical Immediate or Compare Logical		YA102) . . . . .	7-1	TEST . . . . .	7-8	+8.5 Vdc . . . . .	8-14
Immediate . . . . .	5-39	Usage Meter (ALD YA104) . . . . .	7-1	PROCESS . . . . .	7-8	Sequencing . . . . .	8-15
Set Bits On/Off Masked and Test Bits On/Off		File Control Panel . . . . .	7-2	STEP . . . . .	7-8	Overvoltage (OV)/Undervoltage (UV) Protection . . . . .	8-15
Masked . . . . .	5-42	PROGRAM LOAD SELECTOR Switch		CE Switches . . . . .	7-9	Sequence Card (A1) . . . . .	8-15
Set Bits On Masked—SBN . . . . .	5-42	without 3344 installed (ALD PC101, YD100) . . . . .	7-2	EXTENDED SAR ADDRESS BITS Switch . . . . .	7-9	Regulator Card (A2) . . . . .	8-15
Set Bits Off Masked—SBF . . . . .	5-42	PROGRAM LOAD SELECTOR Switch		> 64K ADDR BIT Switch (ALD PA101) . . . . .	7-9	+3.4 Vdc . . . . .	8-15
Test Bits On Masked—TBN . . . . .	5-42	with 3344 installed (ALD PC101, YD100) . . . . .	7-2	SAR/MSAR Display Switch (ALD 'sel MSAR		+8.5 Vdc . . . . .	8-15
Test Bits Off Masked—TBF . . . . .	5-42	PROCESSOR CHECK Light (ALD PC111) . . . . .	7-3	sw' PA101) . . . . .	7-9	Sense Relay . . . . .	8-15
Set Bits On Masked . . . . .	5-43	POWER ON/OFF Switch (ALD YA102, PC111) . . . . .	7-3	I/O OVERLAP Switch (ALD PC121) . . . . .	7-9	EXPANSION MAIN STORAGE POWER	
Set Bits Off Masked . . . . .	5-43	START Key (ALD PC111) . . . . .	7-3	DISPLAY CHK (Check) BITS Switch (ALD		SUPPLY . . . . .	8-16
Test Bits On Masked . . . . .	5-43	Message Display Unit (ALD PB131) . . . . .	7-3	PC121) . . . . .	7-9	Voltage Adjustments . . . . .	8-16
Test Bits Off Masked . . . . .	5-43	I/O ATTENTION Light (ALD PC111) . . . . .	7-3	BSCA/LCA LOCAL TEST (ALD HE160) . . . . .	7-9	+3.4 Vdc . . . . .	8-16
Store, Load, or Add to Register . . . . .	5-46	PROGRAM LOAD Key (ALD PC111) . . . . .	7-3	I/O CHECK Switch (ALD PA101) . . . . .	7-9	+8.5 Vdc . . . . .	8-16
Store Register—ST . . . . .	5-46	STOP Key/Light (ALD PC111) . . . . .	7-3	PARITY CHECK Switch (ALD PA101) . . . . .	7-9	Sequencing . . . . .	8-17
Load Register—L . . . . .	5-46	CONSOLE DISPLAY . . . . .	7-4	STORAGE TEST Switch (ALD PA101) . . . . .	7-9	Overvoltage (OV)/Undervoltage (UV) Protection . . . . .	8-17
Add to Register—A . . . . .	5-46	INT LEV (Interrupt Level) Light (ALD PB111) . . . . .	7-4	ADDR INCREM (Address Increment) Switch		Sequence Card (A1) . . . . .	8-17
Load Address—LA . . . . .	5-50	PWR CHK (Power Check) Light (ALD YA102) . . . . .	7-4	(ALD PA101) . . . . .	7-9	Regulator Card (A2) . . . . .	8-17
Branch On Condition—BC . . . . .	5-51	TH CHK (Thermal Check) Light (ALD YA102) . . . . .	7-4	ADDRESS COMPARE Switches (ALD PA101) . . . . .	7-10	+3.4 Vdc . . . . .	8-17
Store CPU (SCP) . . . . .	5-52	MACHINE CYCLE Indicators (ALD PB111) . . . . .	7-4	<b>POWER SUPPLY</b> . . . . .	<b>8-2</b>	+8.5 Vdc . . . . .	8-17
Load CPU (LCP) . . . . .	5-53	CLOCK Indicators (ALD PB121) . . . . .	7-4	INTRODUCTION . . . . .	8-2	Sense Relay . . . . .	8-17
Load/Store CPU . . . . .	5-54	LAMP TEST Key (ALD YA102) . . . . .	7-4	AC/DC Voltage . . . . .	8-2	±3 VOLT LOGIC SUPPLIES . . . . .	8-18
COMMAND INSTRUCTIONS . . . . .	5-63	Register Display Unit (ALD PB101, PC101) . . . . .	7-5	Basic Unit . . . . .	8-3	±3 Volt Regulator Card . . . . .	8-18
I R Cycle . . . . .	5-63	BSCA OPERATOR'S PANEL . . . . .	7-6	Input Power Requirements . . . . .	8-3	±3 Vdc Regulator Card . . . . .	8-18
Jump On Condition—JC . . . . .	5-65	BSCA Attention Light . . . . .	7-6	Parts Replacement . . . . .	8-3	±3 Volt Power Supply . . . . .	8-18
Halt Program Level (HPL) . . . . .	5-66	Unit Check Light . . . . .	7-6	Checks and Adjustments . . . . .	8-3	±3 Vdc Power Supply . . . . .	8-18
Command CPU (CCP) . . . . .	5-67	Data Terminal Ready Light . . . . .	7-6	POWER SUPPLIES AND COOLING . . . . .	8-4	Voltage Adjustments . . . . .	8-19
I/O INSTRUCTIONS . . . . .	5-68	Data Set Ready Light . . . . .	7-6	DC Bulk Supplies . . . . .	8-6	±3 Vdc Regulator Card . . . . .	8-19
Start I/O—SIO . . . . .	5-68	Clear To Send Light . . . . .	7-6	POWER SUPPLY REGULATORS . . . . .	8-7	+3 Vdc Level Adjustment . . . . .	8-19
I/O Instruction I-Cycles . . . . .	5-69	Receive Trigger Light . . . . .	7-6	Start-Up Control . . . . .	8-8	-3 Vdc Level Adjustment . . . . .	8-19
I/O Cycle . . . . .	5-70	Transmit Trigger Light . . . . .	7-6	Voltage Regulation . . . . .	8-8	±3 Vdc Power Supply . . . . .	8-19
Example 1: . . . . .	5-70	Receive Mode Light . . . . .	7-6	Overvoltage Protection . . . . .	8-9	-3 Vdc Level Adjustment . . . . .	8-19
Example 2: . . . . .	5-70	Transmit Mode Light . . . . .	7-6	Overcurrent Protection . . . . .	8-9	+3 Vdc Level Adjustment . . . . .	8-19
I/O Cycle Data Transfer . . . . .	5-71	Receive Initial Light . . . . .	7-6	Undervoltage Protection . . . . .	8-10	±5 VOLT FEATURE POWER SUPPLY . . . . .	8-20
Cycle Steal Priority (CSP) . . . . .	5-72	Busy Light . . . . .	7-6	DC Bulk Supplies . . . . .	8-6	±5 Vdc Feature Regulator (BSCC) . . . . .	8-20
Load I/O-LIO . . . . .	5-73	Character Phase Light . . . . .	7-6	POWER SUPPLY REGULATORS . . . . .	8-7	±12/-12 VOLT FEATURE POWER SUPPLIES . . . . .	8-21
Sense I/O—SNS . . . . .	5-76	Data Mode Light . . . . .	7-6	Start-Up Control . . . . .	8-8	±12 Vdc Feature Regulator (MLTA and BSCA) . . . . .	8-21
Test I/O and Branch—TIO . . . . .	5-79	Control Mode Light . . . . .	7-6	Voltage Regulation . . . . .	8-8	-12 Vdc Feature Supply (BSCA or BSCC, Without	
Advance Program Level—APL . . . . .	5-80	Digit Present Light . . . . .	7-6	Overvoltage Protection . . . . .	8-9	1200 bps Modem and not MLTA) . . . . .	8-21
<b>SYSTEM OPERATIONS</b> . . . . .	<b>6-1</b>	Auto Call Unit Power Off Light . . . . .	7-6	Overcurrent Protection . . . . .	8-9	+24 VOLT LOGIC SUPPLY . . . . .	8-23
INITIAL PROGRAM LOAD—IPL . . . . .	6-1	Call Request Light . . . . .	7-6	Undervoltage Protection . . . . .	8-10	POWER SEQUENCE . . . . .	8-24
SYSTEM RESET . . . . .	6-2	Data Line in Use Light . . . . .	7-6	DC Bulk Supplies . . . . .	8-6	Power Up Sequence . . . . .	8-24
Objectives . . . . .	6-2	Test Mode Light . . . . .	7-6	POWER SUPPLY REGULATORS . . . . .	8-7	Normal Power Down Sequence . . . . .	8-26
ALTER STORAGE/INITIAL MEMORY SCAN . . . . .	6-4	External Test Switch Light . . . . .	7-6	Start-Up Control . . . . .	8-8	Power Off Sequence . . . . .	8-26
Initial Memory Scan . . . . .	6-4	Rate Select Switch . . . . .	7-6	Voltage Regulation . . . . .	8-8	THERMAL AND POWER CHECKS . . . . .	8-27
				End View of -4V Regulator Card . . . . .	8-11	Abnormal Power Indications . . . . .	8-27
				B GATE -4V LOGIC SUPPLY NO. 2 (Upper			
				Boards) and B GATE -4V FEATURE LOGIC			

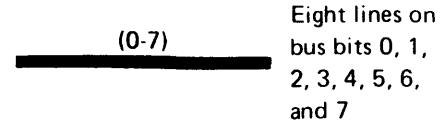
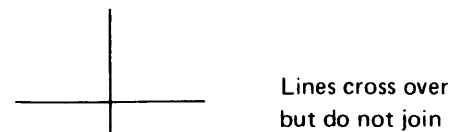
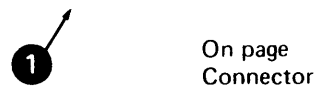
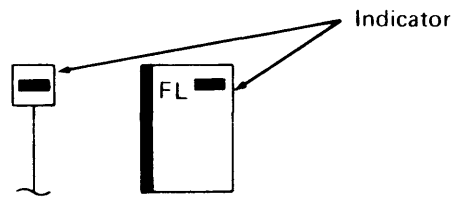
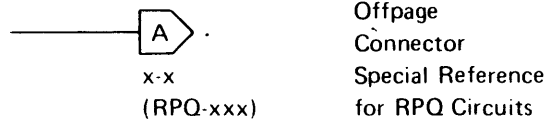
Power Check and Thermal Check Indicators . . . . .	8-28
Test Points (TPs) . . . . .	8-28
System Sequencing and Sensing Relays . . . . .	8-29
Power On Sequence . . . . .	8-29
Abnormal Power Off . . . . .	8-29
Overvoltage and Overcurrent Power Off Sequence . . . . .	8-29
Undervoltage Power Off Sequence . . . . .	8-29
Thermal Power Off Sequence . . . . .	8-29
Emergency Power Off . . . . .	8-29
COIL/RELAY FUNCTIONAL DESCRIPTIONS . . . . .	8-30
Convenience Outlet (YA102) . . . . .	8-30
-5V Sense . . . . .	8-30
Thermal Relay, Relay Panel (YA102) . . . . .	8-30
±12V Supply Sense, Located on Power Supply (MLTA/BSCA-YA140) . . . . .	8-30
AC Voltage to Bulk Power Supply (YA102) . . . . .	8-31
-4V No. 1 (A gate) Sense (YA102A) . . . . .	8-31
+6V (+5V for BSCC only) Sense (YA102A) . . . . .	8-31
+6V UV Detect (YA102B) . . . . .	8-31
Main Storage Supply Sense (YA102A) . . . . .	8-31
I/O AC Power (YA101 and YA102) . . . . .	8-31
I/O AC Power (YA101 and YA102) . . . . .	8-31
7.25 Vac and 41 Vac Distribution (Lamp and Meter – YA102) . . . . .	8-31
I/O Power Sense (YA102A) . . . . .	8-31
+60V Sense (YA102A) . . . . .	8-31
Power Sequence Complete . . . . .	8-32
-4V No. 1 (A gate) OV/OC Fault . . . . .	8-32
-4V No. 1 (A gate) UV or +6V (+5V for BSCC only) OV/OC Fault . . . . .	8-32
+6V (+5V for BSCC only) UV Fault . . . . .	8-32
-4V No. 2 OV/OC Fault . . . . .	8-32
-4V No. 2 Sense and -5V Sense . . . . .	8-32
-4V No. 3 OV/OC Fault . . . . .	8-32
-4V No. 3 Sense . . . . .	8-32
B Gate FET Protect . . . . .	8-32
B Gate FET Protect for No. 3 Feature Power Supply . . . . .	8-33
Expansion Main Storage Power Supply (Feature) . . . . .	8-33
-12V Supply Sense (BSCA/BSCC) . . . . .	8-33
<b>APPENDIX A. CROSS REFERENCE . . . . .</b>	<b>A-1</b>
FEALD – CARD CROSS-REFERENCE LIST . . . . .	A-1
LOGIC CARD REFERENCE . . . . .	A-2
<b>INDEX . . . . .</b>	<b>X-1</b>

**LEGEND**



Two ways to turn FF on:  
 1. B and C active  
 2. A active

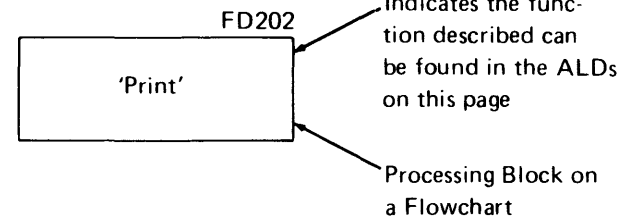
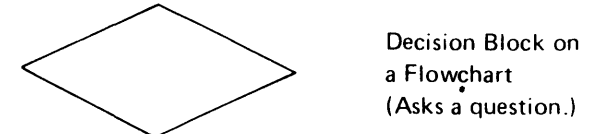
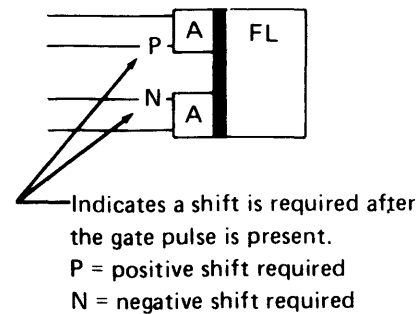
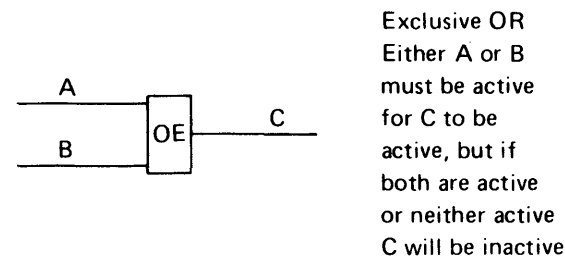
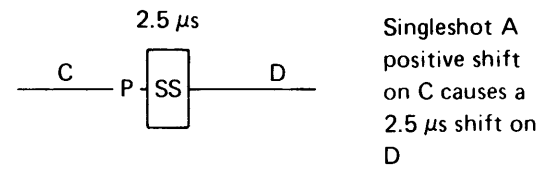
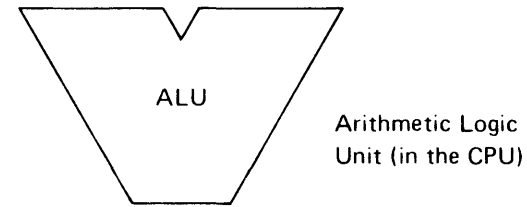
Two ways to turn FF off:  
 1. C and D active  
 2. E active



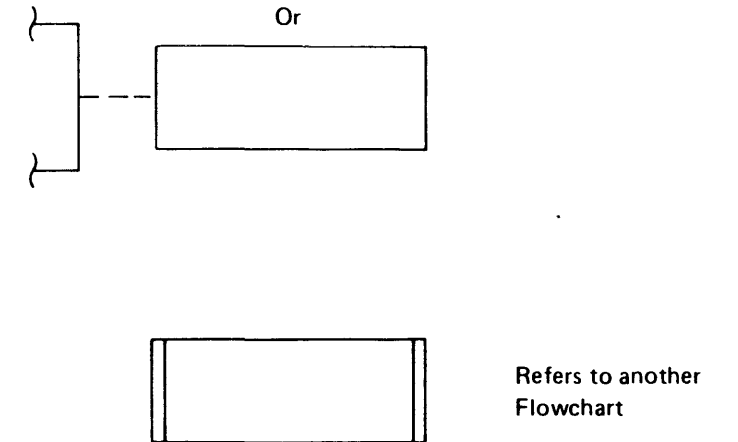
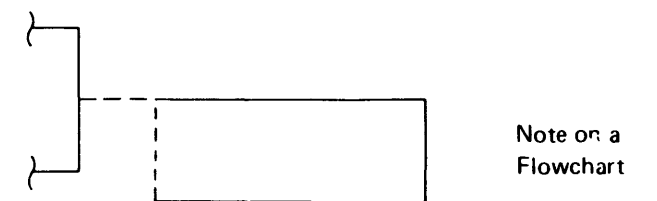
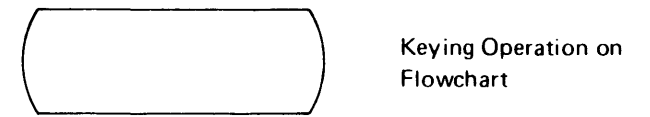
**Boolean Algebra Symbols**

• In a line name means AND  
 + In a line name means OR

(xx - - - x) Indicates a line name that does not exist as an actual ALD name, but used to better explain the function of a line or block



Single quotes in the block indicate a line name, flip latch, or flip flop name



## ABBREVIATIONS

A	Ampere	FDR	Fetch Data Register	OC	Overcurrent
AAR	A Address Register	FE	Field Engineering	Op	Operation; Operand
ac	Alternating Current	FEALD	Field Engineering Automated Logic Diagram	OV	Overvoltage
ALD	Automated Logic Diagram	hex	Hexadecimal	P	Parity
ALU	Arithmetic and Logic Unit	Hz	Hertz	PCB	Printer Control Board
Arith	Arithmetic	IAR	Instruction Address Register	PEB	Printer Electronic Board
ARR	Address Recall Register	Instr	Instruction	PG	Parity Generation
ASCII	American Standard Code for Information Interchange	I/O	Input/Output	PH	Polarity Hold
ATT	Address Translation Table	IPL	Initial Program Load	PMR	Program Mode Register
AUX	Auxiliary	K	K equal 1024	POT	Potentiometer
BAR	B Address Register	LCR	Length Count Register	POR	Power On Reset
BCD	Binary Coded Decimal	LCRR	Length Count Recall Register	PSR	Program Status Register
BSCA	Binary Synchronous Communications Adapter	LIO	Load Input/Output Register	PSS	Print Subscan
BSCC	Binary Synchronous Communications Controller	LPDAR	Line Printer Data Address Register	PTR	Printer
CB	Circuit Breaker	LPIAR	Line Printer Image Address Register	Rd	Read
CE	Customer Engineer	LSR	Local Storage Register	Reg	Register
Ckts	Circuits	MAP	Maintenance Analysis Procedures	RPO	Request Price Quotation
cm	Centimeter	M/C	Machine Cycle	SAR	Storage Address Register
CPU	Processing Unit	MFCM	Multi-Function Card Machine	SCR	Silicon Controlled Rectifier
CR	Condition Register	MFCU	Multi-Function Card Unit	SDBO	Storage Data Bus Out
CRR	Condition Recall Register	MLTA	Multiple Line Terminal Adapter	SDR	Storage Data Register
Ctrl	Control	MOSFET	Metal Oxide Semiconductor Field Effect Transistor	SIO	Start Input/Output
DA	Device Address	MPCAR	MFCU Punch Data Address Register	SIOC	Serial Input/Output Channel
DAR	Data Address Register	MPTAR	MFCU Print Data Address Register	SLD	Solid Logic Dense
DBI	Data Bus In	MRDAR	MFCU Read Data Address Register	SLT	Solid Logic Technology
DBO	Data Bus Out	MS	Main Storage	SMS	Standard Modular System
DDSA	Data-Phone* Digital Service Adapter	MST	Monolithic System Technology	SNS	Sense
DFCR	Data File Control Address Register	mV	Millivolt	SPT	Storage Protect Table
DFDR	Data File Data Address Register	ns	Nanosecond	SS	Single Shot
Disp	Display			Sync	Synchronize
DRR	Data Recall Register			S/Z	Sense/Inhibit
EBCDIC	Extended Binary Coded Decimal Interchange Code			TIO	Test Input/Output and Branch
EC	Engineering Change			TP	Test Point
ECC	Error Checking and Correction			TR Pac	Transformer/Rectifier Pack
EIA	Electronic Industries Association			UV	Under voltage
				V	Volts
				Vac	Volts Alternating Current
				Vdc	Volts Direct Current
				wr	write
				XR	Index Register
				Z	Inhibit
				=	Equals
				>	Greater than
				<	Less than

\*Trademark of the American Telephone &amp; Telegraph Co.

## INTRODUCTION

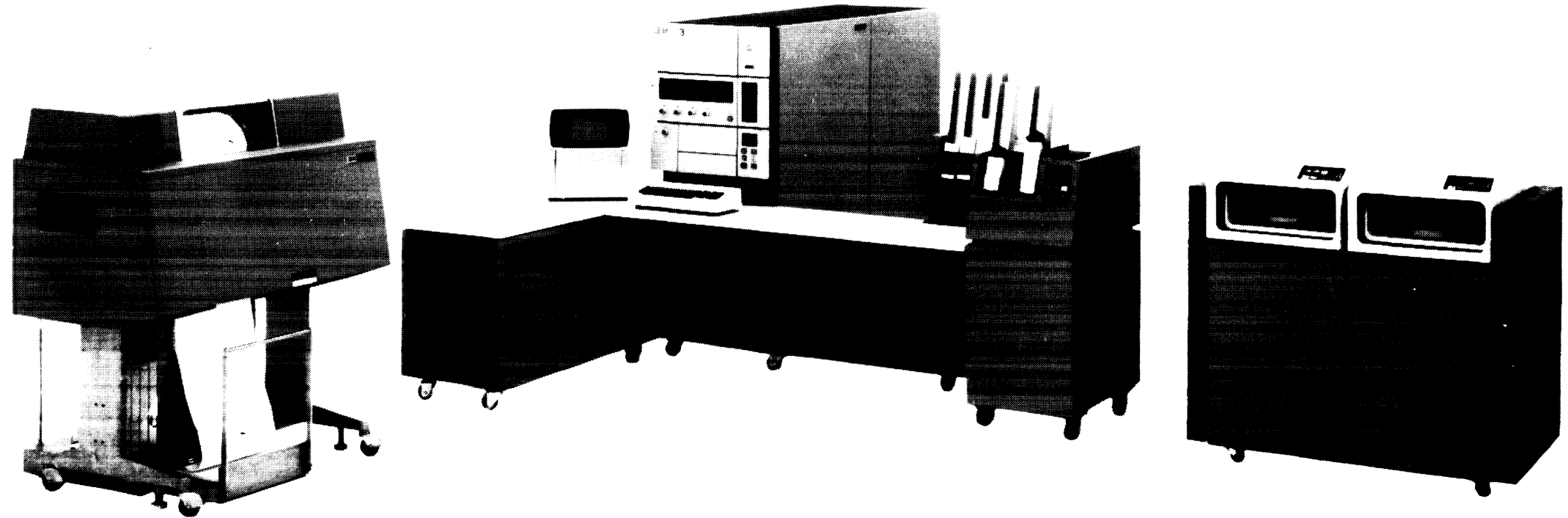
### IBM SYSTEM/3 MODEL 15D

The System/3 Model 15D provides a greater processing speed and greater disk storage than previous Model 15s. The processing speed is increased by accessing storage twice during instruction (I) cycles. The addition of the 3344 DSA provides the greater disk storage.

The various configurations of the System/3 Model 15D provide complete unit record type functions including card reading, punching, interpreting, collating, reproducing, summary punching, computing, and printing. In addition, magnetic tape and disk storage drive removable disks offer practically unlimited data storage growth.

A minimum configuration of Model 15D consists of a Fast I-cycle processing Unit (96K bytes of storage), a 3277 Display Station Model 1 with a keyboard (feature no. 4632), a 3340 Direct Access Storage Facility, a 1403 Printer, and one of the following: 5424 MFCU, 2560 MFCM, 1442 Card Read Punch, or 3741 Data Station (cardless systems).

The 3277 Display Station Model 1 with the keyboard (feature no. 4632) is called CRT/keyboard in this manual.



## IBM SYSTEM/3 MODEL 15D (continued)

The System/3 Model 15D operates in a multi-programming environment without the Model 10 Dual Program Feature. The CPU has the same basic instruction set, cycle time, and access time as other System/3 models.

### Features include:

- MOSFET (metal oxide semiconductor field effect transistor) monolithic storage with ECC (error correction and checking) for CPU main storage.
- Maximum of 512K bytes of storage available.
- Write/fetch storage protection (for CPU cycles only) in 2K-byte segments. This feature is used by the supervisor to keep user programs from interfering with each other or with the supervisor.
- Three additional instructions to control multi-programming and other new enhancements.
- Address translation capability to address main storage above 64K.
- Four additional interrupt levels.
- Operation end interrupts for all input/output devices.
- Provisions to mask off (inhibit) all interrupts except program check.
- Privileged mode of operation. Privileged instructions are executed only by the system control programming.
- Program check interrupt to prevent a *hard stop* for invalid storage or device address, invalid operation code, storage protection violation, and privileged mode operation error.

- Complete overlap of all I/O operations without data overrun errors.
- A two-byte disk data channel to reduce CPU burden (I/O cycle steals) during data transfer.
- An internal timer to provide time-of-day services.
- Unit record restart to allow program detection of an I/O device not-ready to ready transition.
- Fast I-cycle processing unit (Model 15D) increases execution rate of instruction cycles.

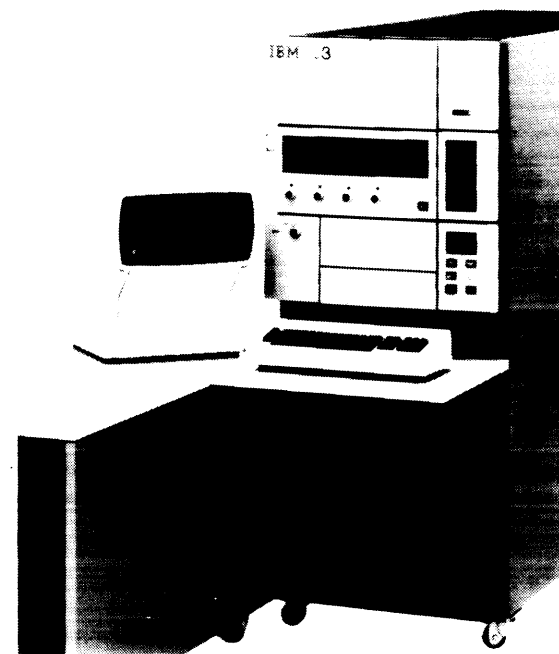
## System Control Programming

### Multiprogramming

One user program can reside in each of the System/3 Model 15 program partitions. The programs then share the CPU facilities, thus reducing the time that the system is in a wait state.

Program partition priority is controlled by the supervisor. All programs operate with interrupts enabled; when an interrupt occurs, the supervisor gains control, processes the interrupt, and gives control to the highest priority partition that is in a ready state. The high priority partition gives up control when it encounters a condition that prevents further processing. The supervisor takes control away from the low priority partition at the completion of an event for which the high priority partition is waiting.

## I/O Devices



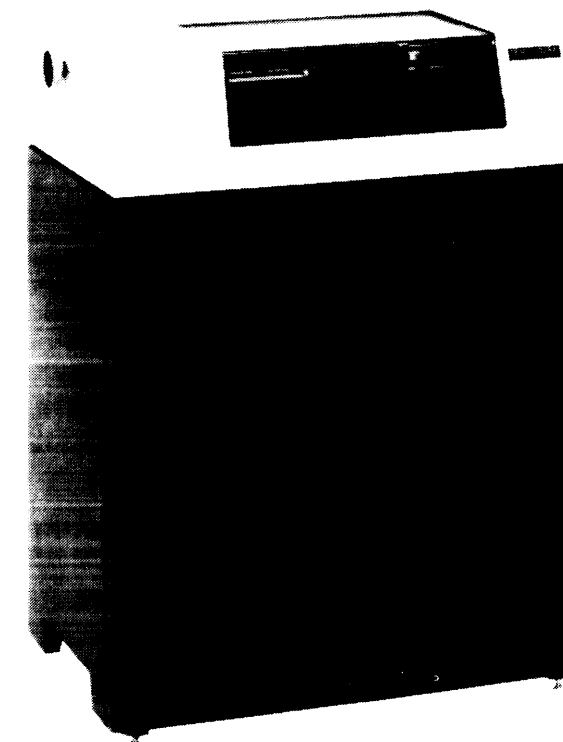
### 3277 Display Station/Keyboard

The CRT/keyboard used on System/3 Model 15 is composed of:

- An IBM 3277 Display Station Model 1 cathode ray tube (CRT) screen.
- A 78-key operator console keyboard.

The CRT/keyboard is the operator/system communication device for Model 15, and is required on every system. It attaches directly to the CPU and is on the system table top. The operator uses the CRT/keyboard for inquiry, secondary output, limited key entry of data, and operator/program interaction.

The CRT displays 480 characters (12 lines of 40 characters each) and supports a 64-character set (36 alphameric characters, 27 special characters, and a blank). The keyboard is a movable 78-key EBCDIC (Extended Binary Coded Decimal Interchange Code) keyboard that has 45 alphameric keys, 21 control keys and 12 program function keys. The CRT displays each character as it is keyed.

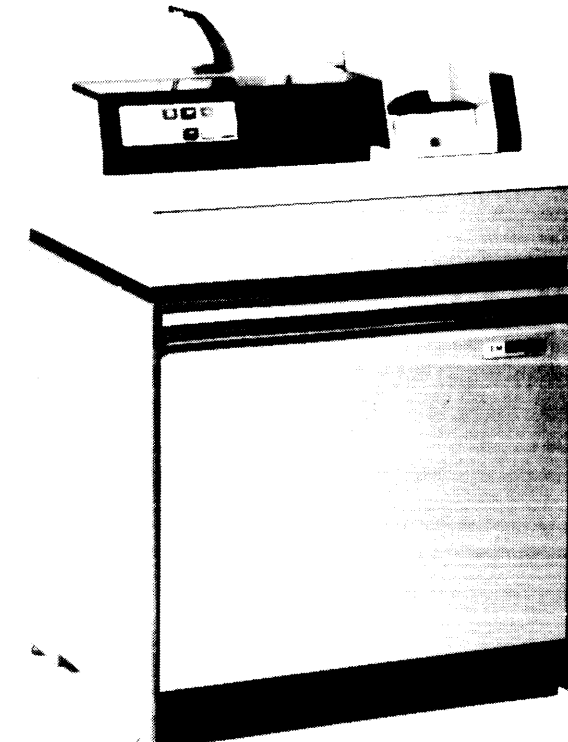


### IBM 3284 Printer

The 3284 Model 1 Printer can be attached to System/3 Model 15 as an auxiliary printer.

The primary use of the 3284 is to obtain hard copy output of system messages that appear on the CRT. The 3284 can also be used for low volume output of other programs.

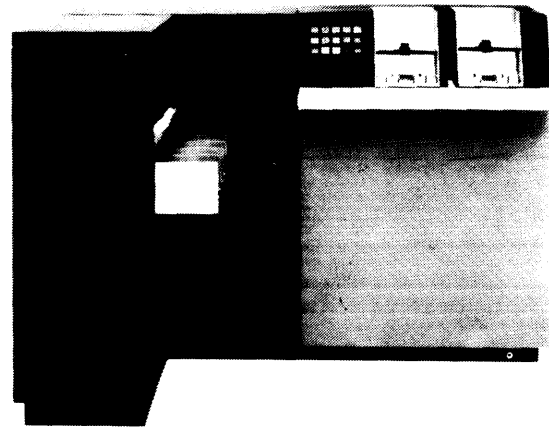
The 3284 Printer uses a matrix print head and pin feed platen, and prints 40 characters per second. It uses a character set of 64 EBCDIC characters and can produce print lines of 120, 126, or 132 print positions.



### IBM 2501 Card Reader

System/3 Model 15 supports the IBM 2501 Card Reader Models A1 and A2. Models A1 and A2 read 80-column cards at a maximum rate of 600 and 1000 cards per minute, respectively. The card image feature is available; however, it is supported only by the Basic Assembler.

**I/O Devices (continued)**

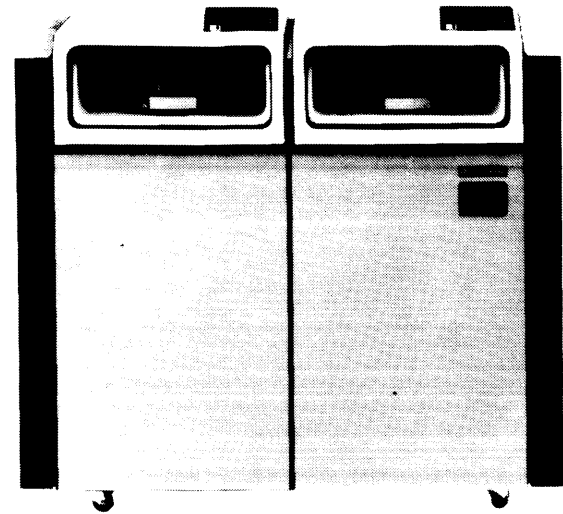


**IBM 2560 Multi-Function Card Machine**

The IBM 2560 Multi-Function Card Machine (MFCM) is an 80-column card device available in two models; Model A1 and Model A2. The MFCM can:

- Read cards from either of two hoppers. Model A1 reads 500 cards per minute; Model A2 reads 310 cards per minute.
- Punch cards at 160 columns per second (Model A1) or 120 columns per second (Model A2).
- Print on cards (Model A1 with optional print feature) at 140 positions per second.
- Stack cards in any of five stackers (Model A1) or four stackers (Model A2).

The MFCM can be used to match records within two files, merge two files, select records, or sort card files.



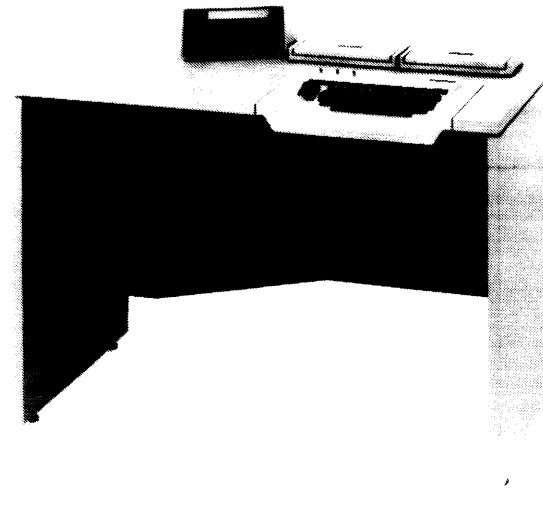
**IBM 3340 Direct Access Storage Facility**

The IBM 3340 Direct Access Storage Facility provides the IBM System/3 Model 15D with a maximum of 205 megabytes of direct access storage. The system requires at least one 3340 Model A2 and can also be equipped with either a 3340 Model B1 or B2.

Two, three, or four drives can be attached to a single system in the following configurations:

Configuration of Models	Total Number of Drives	Total Capacity in Data Bytes
1 3340 Model A2 only	2	102,924,288
1 3340 Model A2 and 1 3340 Model B1	3	154,386,432
1 3340 Model A2 and 1 3340 Model B2	4	205,848,576

Each 3340 drive contains the mechanical and electrical components needed to house, load, filter, and drive a 3348 Model 70 Data Module. The 3340 Model A2 also provides logic and power for all the 3340 drives.

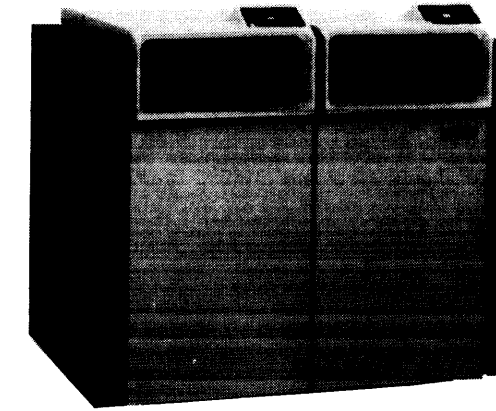


**IBM 3741 Data Station Models 1 and 2  
IBM 3741 Programmable Work Station Models 3 and 4**

The System/3 Model 15 supports all four models (Data Station Models 1 or 2, or the Programmable Work Station Models 3 or 4) of the 3741. The 3741 (all models) is a standalone device with one operator station, which has one keyboard unit, one display unit, one or two diskette units, and a control unit. Models 2 and 4 are also equipped with the binary synchronous communications adapter.

The 3741 can be used online as a diskette input/output device or offline to perform such 3741 functions as data entry, communications, and as a programmable work station (Models 3 and 4 only). In the online mode, the 3741 keyboard is inoperative except to take the 3741 offline. Data transfer between the 3741 and the system is always between the system and the diskette, never between the system and the keyboard/display screen.

A 3741 Model 1, 2, 3, or 4 is required on all 5415D cardless systems (channel terminate feature installed).

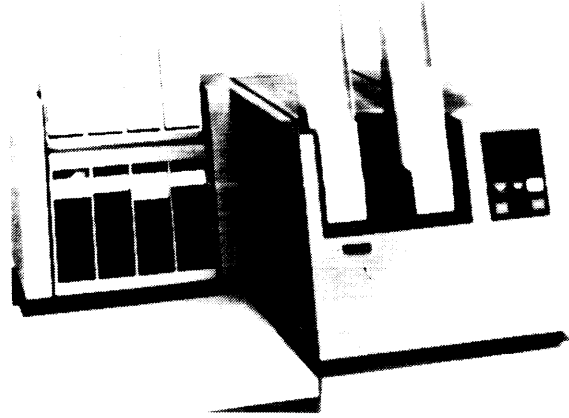


**IBM 3344 Direct Access Storage**

The 3344 Direct Access Storage Model B2 is a two-drive unit that attaches to the 3340 Direct Access Storage Facility Model A2 on the 5415 Model D. There is a 3344 data module permanently mounted within each drive. The disk surfaces and read/write heads are sealed within each 3344 data module.

The two drives of a 3344 provide approximately 407 million bytes of direct access storage.

## I/O Devices (continued)



### IBM 5424 Multi-Function Card Unit

The IBM 5424 Multi-Function Card Unit (MFCU) is a 96-column card device that:

- Reads or feeds cards from either of two hoppers.
- Punches cards.
- Prints on cards.
- Stacks cards in any of four stackers.

Two models of the MFCU are available. Model A1 reads 250 cards per minute and punches or prints 60 cards per minute. Model A2 reads 500 cards per minute and punches or prints 120 cards per minute.

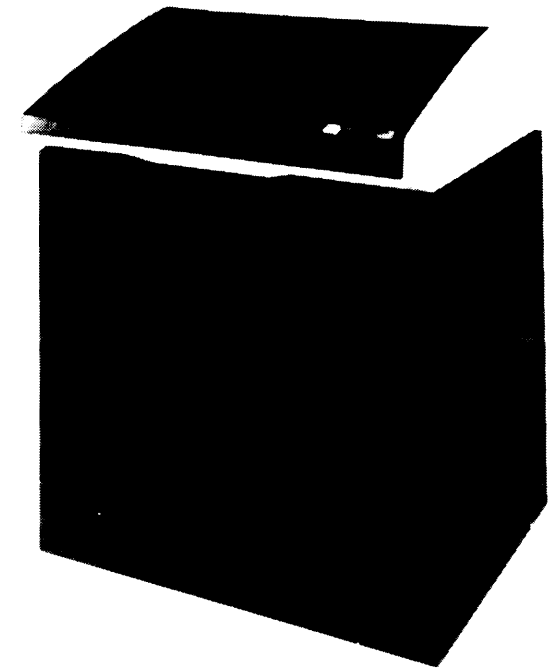
The MFCU is controlled by the 5424 attachment and the processing unit. The MFCU combines the functions of a card reader, collator, card punch, sorter, and interpreter. All multifunction operations, except sorting, can be performed in a single pass through the machine; sorting requires multiple passes.

### IBM 1442 Card Read Punch

An IBM 1442 Card Read Punch Model 6 or Model 7 can be attached to the System/3 Model 15 to provide 80-column card reading and punching. Model 6 reads 300 cards per minute and punches 80 columns per second; Model 7 reads 400 cards per minute and punches 160 columns per second.

The 1442 can perform the following operations:

- Read
- Punch with no feed
- Punch and feed
- Stack cards in either of two stackers
- Read column binary (card image)—supported only by the Basic Assembler program.



### IBM 3410 Magnetic Tape Unit

The 3410/3411 Magnetic Tape Units read and write half-inch magnetic tape. The 3410 is a tape unit only; the 3411 is a tape unit and a control unit in the same frame. From one to four tape units can be attached to System/3 Model 15.

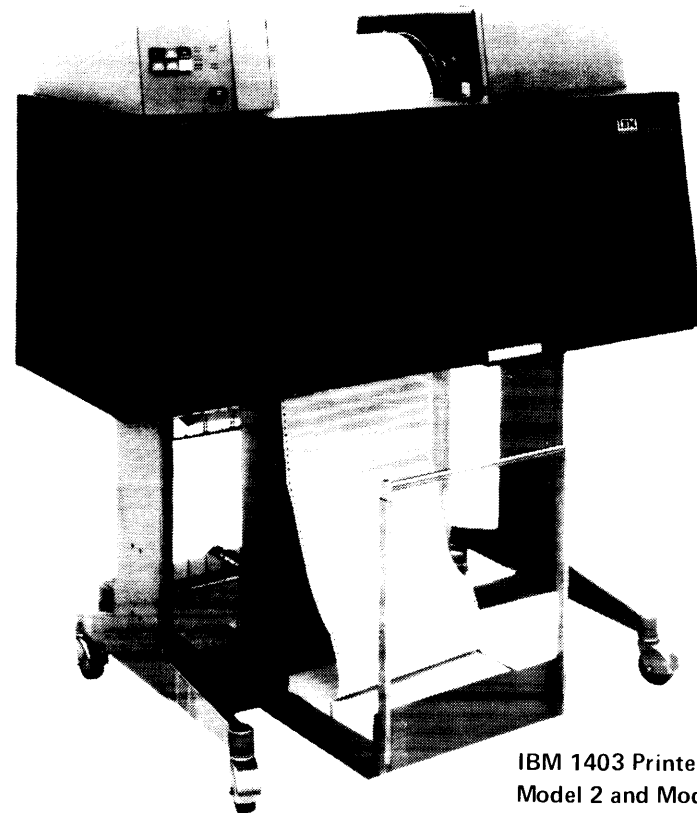
The 3410/3411 Models 1, 2, and 3 have data rates of 20, 40, and 80 kilobytes per second, respectively. All units in a magnetic tape system must be the same model. Recording density can be 800 or 1600 bits per inch (bpi) for nine-track tape, or 200, 556, or 800 bpi for seven-track tape.

Both the 3410 and 3411 are desk high units with tape reels mounted horizontally rather than vertically. A transparent sliding cover provides easy access to the tape reels.

Each 3410/3411 tape unit must be equipped with a special feature that specifies the read/write format desired. The features are single density, dual density, and seven-track. Dual-density and seven-track features cannot be installed on the same subsystem.



## I/O Devices (continued)



IBM 1403 Printer  
Model 2 and Model 5

### IBM 1403 Printer Model 2, Model 5, and Model N1

The IBM 1403 Printer (Model 2, 5, or N1) is required on every System/3 Model 15. It is attached via an IBM 5421 Printer Control Unit. Each model produces a 132-print-position line. The character set can be expanded from 48 characters (basic) to as many as 120 characters if the universal character set special feature is used.

*Note:* The Models 2 and 5 require an interchangeable chain cartridge adapter special feature for installation of the universal character set.

Various type fonts, styles, and character arrangements are available.

The 1403 printers use a type cartridge with 240 characters. The standard 48 character set is repeated five times on the cartridge to permit the following throughput (single line spacing):

Model 2 — 600 lines per minute  
Model 5 — 465 lines per minute  
Model N1 — 1100 lines per minute

### SIOC

The Serial Input/Output Channel (SIOC) allows attachment of the same devices as on the System/3 Model 10, such as 1231, 1255, 1419, etc.

### BSCA

System/3 Model 15 supports the same communications capabilities that are presently available on the System/3 Model 10. One or two binary synchronous communications adapters (BSCA) are available to allow synchronous communications transmission rates from 600 bps to 50,000 bps. Communication with other IBM processors as well as with batch and interactive terminals is possible with BSCA on both point-to-point and multipoint basis. The EIA Local feature permits attachment of local CRT clusters and other bisynchronous terminals without the addition of modems and data communication lines.

### Display Adapter

The display adapter provides control and I/O channel interfacing for the 3277 Display Station and 3284/86/88 auxiliary printers without the requirement of a 3271 Control Unit and a remote communications line. Each terminal connects directly to the display adapter via a single coaxial cable. The terminal may be located up to 2000 feet from the system.

The display adapter interacts with the program like a BSCA device (EBCDIC only, point-to-point nonswitched). Up to 30 devices, display stations, or printers can be used.

This feature cannot be used with BSCA-2. When installed, it uses the BSCA-2 channel address, interrupt level and cycle steal priority.

### LCA (Local Communications Adapter)

The LCA provides a directly connected binary synchronous communications adapter on System/3 Model 15 and allows the attachment of 3741 Model 2 without the use of modems. Also, a direct connection to 3271/3275 provides limited function local CRT attachment. The LCA feature is a simplified, reduced function BSCA similar to the existing BSCA-1 on the System/3 Model 10. Attached devices must have a BSCA feature with a point-to-point nonswitched network, EBCDIC transmission code, and a local EIA data set interface. A data rate of 2,400 bps is standard.

### MLTA (RPQ)

The multiple line terminal adapter (MLTA) provides attachment capability for IBM's low-speed start/stop terminals. MLTA allows attachment of one to eight communication lines with multiple terminals per line. Transmission speeds are 134.5 bps to 1200 bps. Terminals may be attached on an unlimited, limited, or in-house basis.

### BSCC (Binary Synchronous Communications Controller)

The BSCC permits the IBM System/3 to function as a point-to-point or a multipoint processor terminal. One or two BSCC lines are available and each line handles 127 3270-type devices. The BSCC lines allow synchronous communications transmission rates from 600 bps to 9,600 bps. The BSCC operates with EIA, 38LS, or DDSA modems; or the EIA local feature permits attachment of a cluster of local terminals without the use of modems or data communication lines.

### Channel Terminate Feature

The channel terminate feature terminates channel bank 1 on 5415D cardless systems. The card I/O board (01A-A3) is replaced by the channel terminate board which has two MST line terminator cards installed in locations U2 and U3.

A directly attached 3741 is required on all 5415D cardless systems. The 3741 becomes the APLD (alternate program load device).

Installation of the channel terminator feature does not prevent attachment of the 2501 as a card input device.

### System Maintenance

The primary sources available for failure detection and isolation are the diagnostics and Maintenance Analysis Procedures (MAPs) used in conjunction with one another. The System/3 also logs certain errors in storage to assist with failure analysis. All this is described in the diagnostic user's guide.

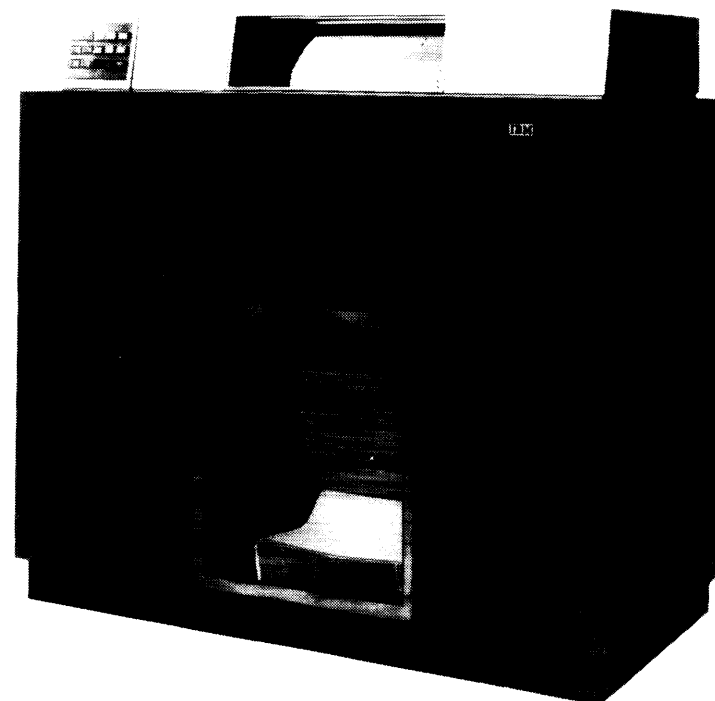
This manual provides theory, major operation objectives, data flow, and conceptual positive logic diagrams for learning the system, then as an aid to finding a machine failure using the MAPs.

### Scheduled Maintenance

Every 6 months:

1. Check filters visually and replace if necessary.
2. Check cooling fans for proper operation.

I/O scheduled maintenance procedures are in the maintenance manual for each I/O device.



IBM 1403 Printer Model N1

## 5415 PROCESSING UNIT (CPU)

### Introduction

The IBM 5415 CPU contains the facilities for addressing storage, arithmetic and logical processing of data, sequencing instructions, and controlling the transfer of data between main storage and attached input/output devices. The basic unit of information is the byte which represents one alphabetic, numeric, or special character. In arithmetic operations, a byte contains one numeric character and a zone. The low-order byte contains the sign in the zone portion. Bytes may be handled separately or grouped together to form fields.

The CPU main storage (MOSFET technology) is available in 96, 128, 160, 192, 224, 256, 384 and 512K byte capacity. Since the System/3 data path allows addressing only 64K of main storage, an address translation table (ATT) is used. Through use of the ATT, the 512K of main storage can be addressed.

The supervisor program uses storage protection to protect main storage in 2K-byte segments. Write protect prevents one user program from writing over another user program or the supervisor program. Fetch protect prevents an unauthorized user from accessing data.

The CPU main storage unit read and write cycle time is 480 ns each. There is calculation time between the read/write time providing a basic machine cycle (read/compute/write) time of 1.52  $\mu$ s.

Interrupts allow the system to make optimum use of the I/O devices, to handle program checks, and to execute data handling routines. An interrupt that originates at an I/O device signals operation end or requests special attention from the CPU. Generally, an interrupt means that the CPU must interrupt a current instruction sequence, perform an intervening instruction sequence, and return to the interrupted program.

The CPU has direct control over the I/O devices attached to it. Program instructions that determine which operation is performed (read, write, etc.) and which unit is to be used test and initiate I/O operations.

The CPU is available for processing during most of each I/O operation even though many devices may be functioning simultaneously. This overlap of I/O operations and CPU processing is made possible by a cycle steal capability by which an I/O device, while performing an I/O operation, breaks into the main program and uses enough cycles to transmit the bytes that are immediately available. For example, when reading a row of data from a card, that data is placed in the proper storage location during I/O cycles. The main program then continues processing until the next row of data is available. Thus, the cycle steal capability provides the benefit of a buffer without sacrificing storage capacity or requiring a special buffer. When the operation end interrupt occurs, the CPU executes the program for handling the data just read from the card.

Step-by-step data processing is controlled by registers (op register, Q register, and condition register) that contain the operation code for the instruction being performed and the additional information required to execute the instruction.

The arithmetic and logical unit (ALU) performs calculations within the CPU. The CPU routes all data to be processed through the ALU, which is capable of performing the desired action.

In the CPU, odd parity is provided for all bytes of data to provide a means of validity checking. As the data is transferred throughout the CPU, parity circuits check for correct data transfer. As data is changed within the ALU, the parity of the resultant byte is determined by a second ALU (check ALU) and the generated parity is then checked at the ALU latched output.

In addition to parity checking, as the program is executed, the CPU checks each operation code to ensure that it contains a valid instruction.

Only the supervisor program can execute certain CPU instructions (privileged mode) unless the CPU is in Model 10 mode. A program mode register (PMR) controls this Model 10 mode as well as other CPU functions. There are nine such registers, one to control each program level.

### Fast I-Cycles

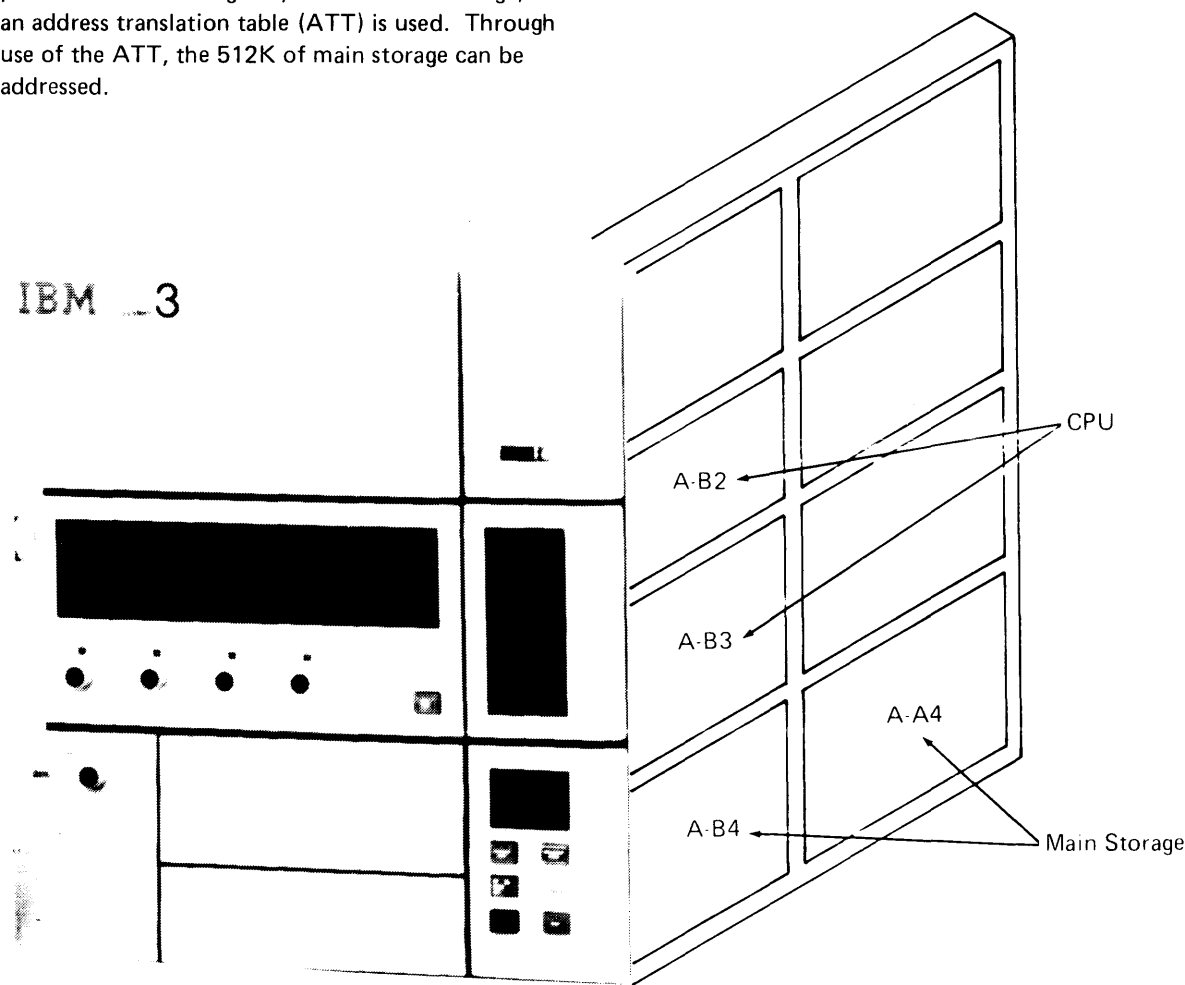
The execution rate of instruction (I) cycles is increased in the 5415D by accessing storage twice (two instruction bytes) during a 1.52 usec machine cycle. A four byte instruction (I-Op, I-Q, I-H1, and I-L1 cycle) previously taking four machine cycles to complete now requires only two. See simplified timing diagram **A** for relationship between previous I cycle processing and fast I cycle operation. To access storage twice during a single machine cycle, a two-byte IAR update in parallel with each storage fetch is required. This two-byte update is accomplished by adding a 16-bit auxiliary ALU. The aux ALU is also used to flush data from storage to the proper register on all fast I cycles and to perform base displacement operations during I-X cycles.

Certain CPU operations must be performed at normal processing speed. For those operations, the last half of each I cycle is followed by an idle or dummy half cycle. For example, a dummy half cycle is taken following the last I cycle of a 3 or 5 byte instruction. During this half cycle, the processor is idle with no storage fetches or IAR updates. One exception is the IR cycle of a command format instruction which extends over the entire machine cycle.

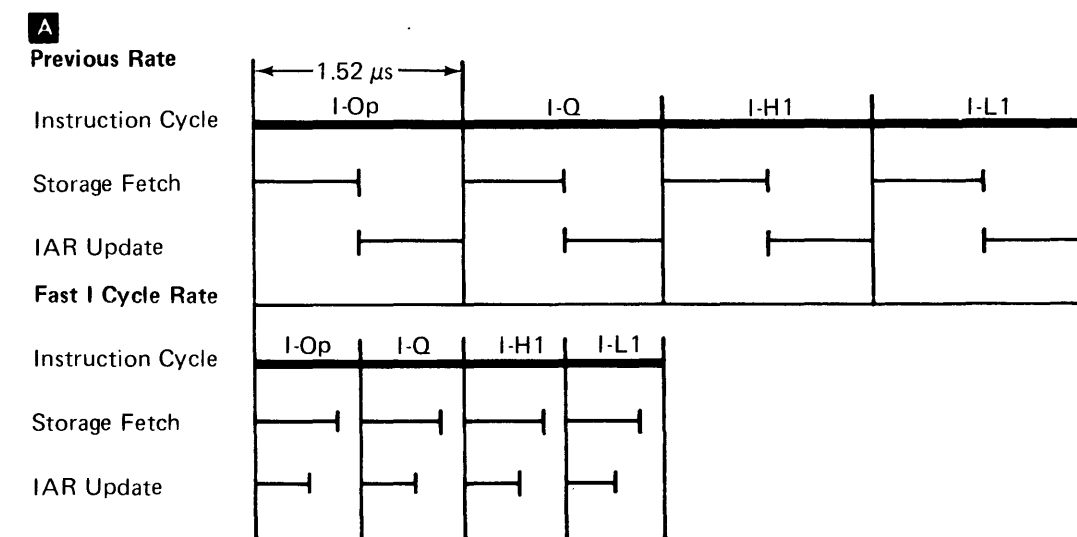
Those operations that require normal processing speed include:

- I/O and Halt instructions
- The last cycle of a 3 or 5 byte instruction
- Certain time dependent diagnostic programs
- Cycles during which program checks occur

Refer to the instruction cycle diagram on page 5-2 for the I cycle, dummy half cycle, and machine cycle relationship of all instruction types.



IBM 5415 Processing Unit Model D



## 5415 PROCESSING UNIT (Continued)

### Data Flow

Data flows serially through the machine in 8-bit bytes plus one parity bit through ALU and aux ALU. The output from ALU and aux ALU is gated and distributed to the remaining functional units of the machine.

### I-Cycle Operation

The aux ALU receives data from the local store registers (two-bytes wide) or the SDB0 (one-byte wide) and can either pass the inputs directly through aux ALU unchanged or combine them with an ADD or AND function. Data passed directly through can be incremented by +1 if the carry in signal is active.

The auxiliary ALU output is gated to the local store registers, condition register, Q register, and op register. Refer to section 5 for a detailed description of each I-cycle operation.

### E Cycle Operation

The ALU receives two bytes of data and combines them in parallel into one byte. ALU performs decimal add and subtract, binary add and subtract, and logical AND and OR operations. All data to the ALU comes from the A and B registers, and the output is the contents of B modified by the contents of A.

Output of the ALU is available to the I/O attachments on the data bus out (DBO) in one of two forms: either translated from EBCDIC to System/3 card code, or straight from the ALU. If data is transferred to or from a disk drive, two bytes of data can be transferred: one through ALU, the other through the extended channel. The ALU output is also available for entry into main storage, to the op and Q registers for instruction decode, to the condition register (CR), and to the local storage registers (LSR) for temporary storage.

### CPU Timing

The length of the CPU machine cycle is 1.52  $\mu$ s. Each machine cycle contains a storage read and an optional storage write time.

Each machine cycle is divided into nine clock times; clock 0 through clock 8. Clock 0 and clock 1 are each 200 ns and all of the remaining clock times are 160 ns. Each clock is divided into 40 ns phases. A clock 9 time, consisting of five phase pulses (200 ns), is taken each cycle during cycles of the following operations (1) system reset, (2) step mode, (3) alter SAR, (4) alter ATT/PMR, and (5) alter/display storage with the storage test switch in its step position.

### Parity Checking and Generation

The CPU checks for dropped or extra bits during data transfer by checking for an odd number of bits after the transfer. The parity checking (P) and parity generating (PG) points are shown in data flow diagram.

Correct ALU output parity is generated. After the data leaves the A and B registers, it can be altered by the decimal and binary complement circuits, the ALU, the decimal correct circuits, and the sign control circuits. The parity changes caused by all these make it necessary to generate parity for the ALU output.

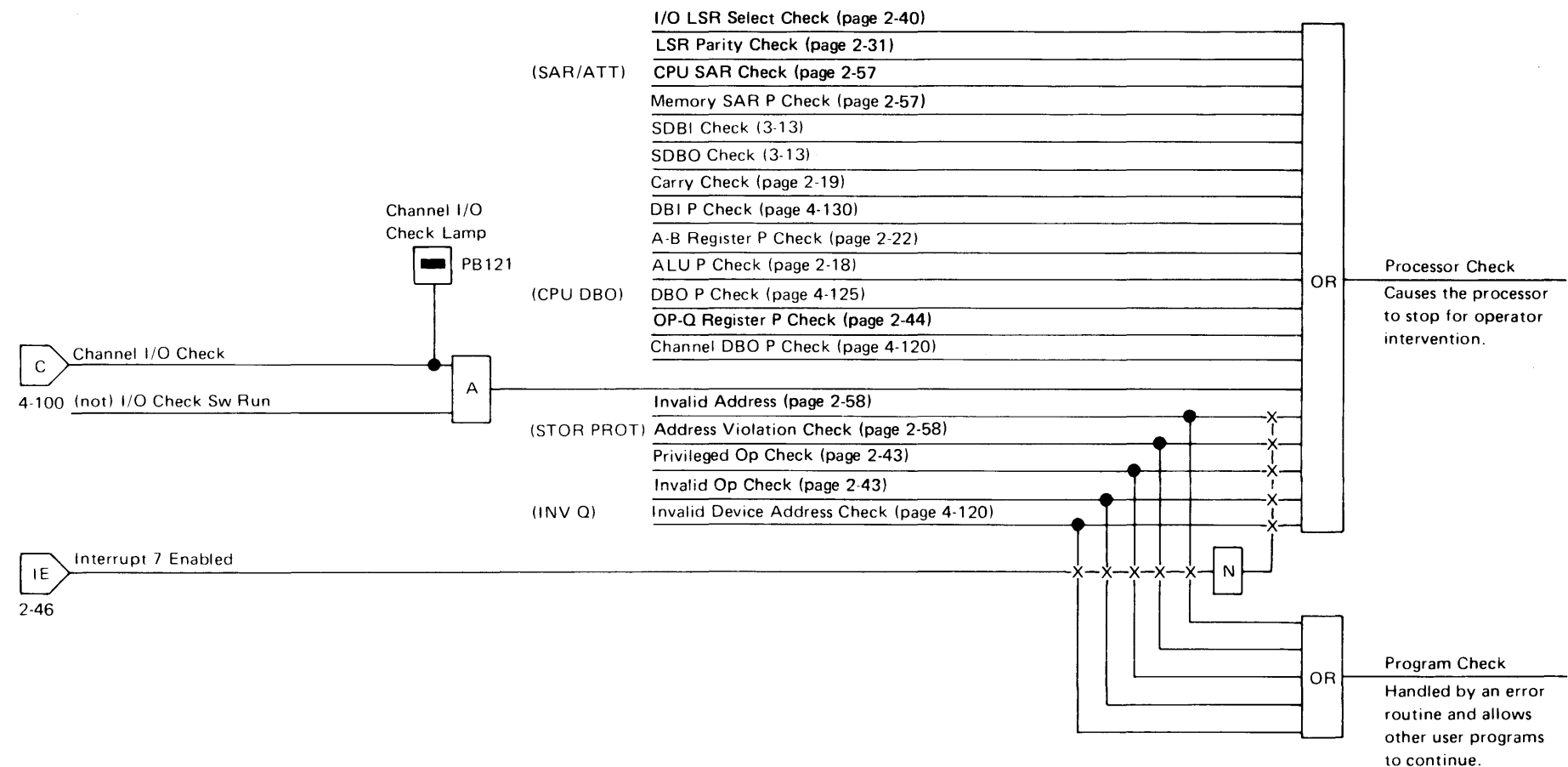
### Unit Check

Unit check handling of testable indicators is controlled by programs. Restart procedures are conveyed to the operator by programmed halt operations, halt identifiers displayed on the console, and recovery/restart procedure listings.

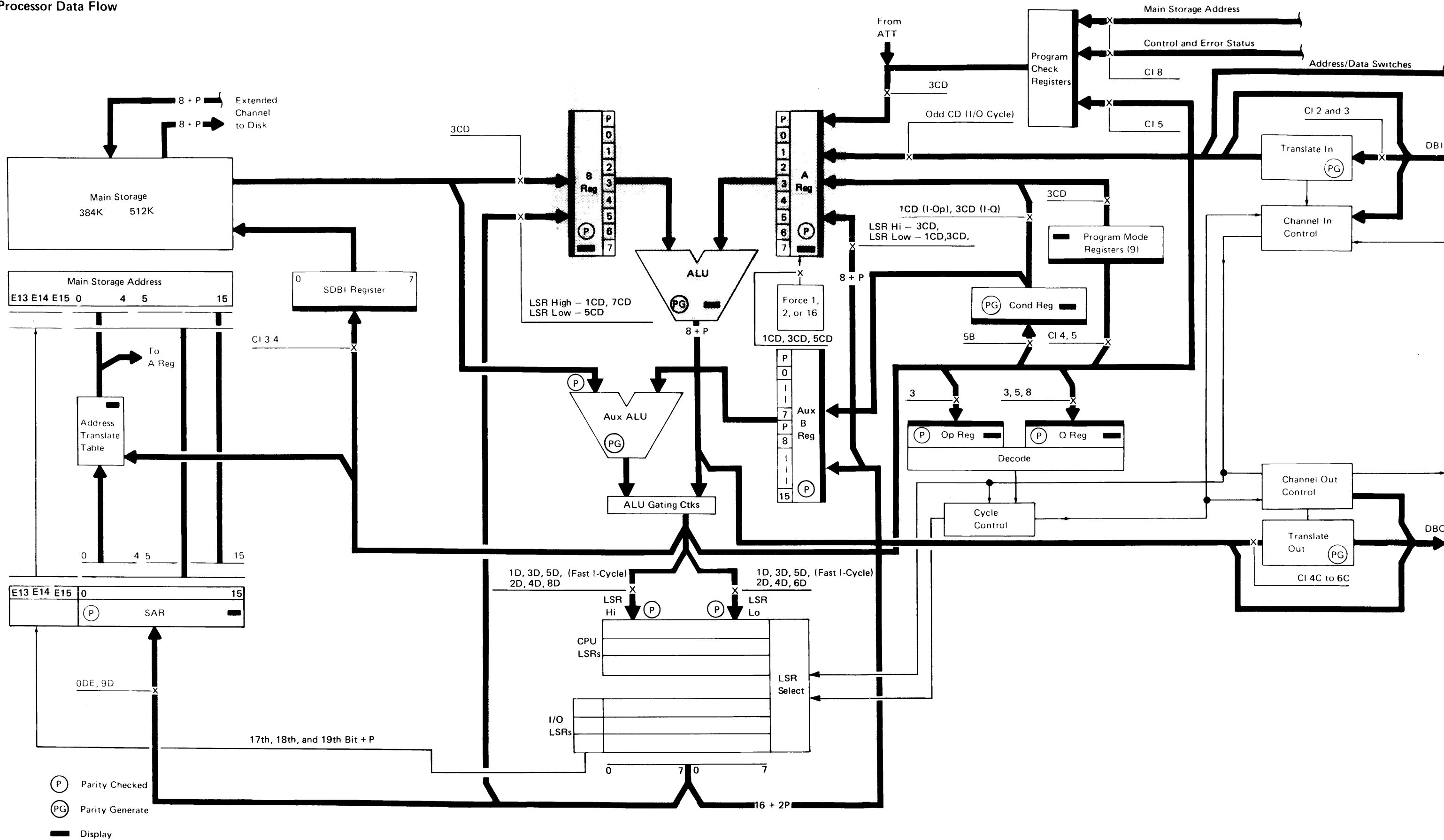
### Errors

The types of CPU errors are Processor checks and Program checks.

8	PROC CHK			I/O LSR	LSR	SAR ATT	MSAR	INV ADDR	STOR PROT	SDBI	SDBO	CAR	DBI	A/B	ALU	CPU DBO	OP/Q	PRIV OP	INV OP	CHAN DBO	INV Q	PROC CHK
---	-------------	--	--	------------	-----	------------	------	-------------	--------------	------	------	-----	-----	-----	-----	------------	------	------------	-----------	-------------	----------	-------------



Processor Data Flow



## Machine Language (Part 1 of 5)

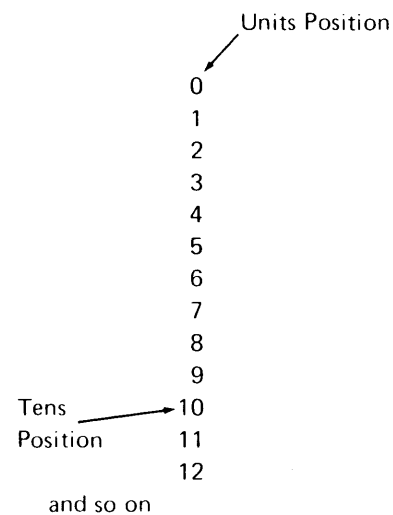
### Number Systems

To understand the operation of the CPU, it is necessary to understand the number systems and character codes used. Accordingly, the following topics discuss the decimal, binary, and hexadecimal number systems.

#### Decimal Number System

The decimal number system has ten symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. The base of the system is 10.

Counting starts in the units position with 0 and proceeds through the next nine symbols. When 9 is reached, there are no more symbols; therefore, a 1 is placed in the position to the left (tens position) and the count resumes with a 0 in the original position:



Continuing the count, it takes one hundred one numbers (count began with zero) before a third position (hundreds position) is required to express a 3-digit number. Similarly, it takes one thousand one numbers before a fourth position (thousands position) is required to express a 4-digit number. Because of the role that the powers of 10 play in the representation of a number, (10 unique symbols), 10 is said to be the base of the decimal system.

A number is made up of coefficients defined by the power of their position. Each coefficient is multiplied by a power of 10 and some number from 0 to 9. For example, the number 123 breaks down as follows:

$$(1 \times 10^2) + (2 \times 10^1) + (3 \times 10^0)$$

#### Binary Number System

The binary number system has two symbols: 0 and 1. The base of the system is 2.

Digital computers use binary circuits and binary mathematics. The binary, or base 2 system, uses two symbols, 0 and 1, to represent all quantities. Counting is started in the same manner as in the decimal system, with 0 for zero and 1 for one. The number 2 is expressed by placing a 1 in the next position to the left and starting again with 0 in the original position. Thus binary 10 is equivalent to 2 in the decimal system. Counting continues with a carry to the next higher order every time a 2 is reached instead of every time a 10 is reached. Counting in the binary system is as follows:

Binary	Decimal	Binary	Decimal
0	0	110	6
1	1	111	7
10	2	1000	8
11	3	1001	9
100	4	1010	10
101	5	1011	11

and so on

The 1's and 0's of a binary number represent the coefficients of the ascending powers of 2. To illustrate, assume the binary number 1111011; the number is expressed as:

$$(1 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$$

The various terms do not have the meanings of units, tens, hundreds, thousands, etc., as in the decimal system, but signify units, twos, fours, eights, sixteens, etc. Thus the binary number breaks down as follows (compared with decimal equivalent):

Binary	Decimal
1111011	123
1	3 units
1	20 tens
1	100 hundreds
1	123
0	
1	1 units
1	2 twos
1	0 fours
1	8 eights
1	16 sixteens
1	32 thirty-twos
1	64 sixty-fours
1	123

#### Hexadecimal Number System

- System has 16 symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.
- Base of system is 16.
- System is shorthand notation for binary numbers.
- Four binary bits are represented by one hexadecimal symbol.
- Byte is represented by two hexadecimal symbols.

Binary numbers have approximately 3.3 times as many coefficients as their decimal counterparts. This increased length presents a problem when talking or writing about binary numbers. A long string of 1s and 0s cannot be effectively spoken or read. A shorthand system is necessary, one that has a simple relationship to the binary system and that is compatible with the basic 8-bit byte used in the CPU. The hexadecimal number system meets these requirements.

Counting is performed as in the decimal and binary systems. When the last symbol (F) is reached, a 1 is placed in the next position to the left and counting resumes with a 0 in the original position, as follows:

0	10	20	A0
1	11	21	A1
2	12	22	A2
3	13	23	
4	14		
5	15		
6	16		
7	17		
8	18		
9	19		
A	1A	9A	
B	1B	9B	
C	1C	9C	
D	1D	9D	
E	1E	9E	
F	1F	9F	

and so on

One hexadecimal symbol can represent four binary bits. Hence byte, can be represented by two hexadecimal symbols. The relationship between the hexadecimal, binary, and decimal systems is as follows:

Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

It is important to remember that four binary positions are equivalent to one hexadecimal position.

Hexadecimal numbers are represented in the same manner as decimal and binary numbers, except that the base is 16. The terms of the number represent the coefficients of the ascending powers of 16 as in the following example of the hexadecimal number 257 (decimal equivalent equals 599):

$$\begin{aligned} 257 &= (2 \times 16^2) + (5 \times 16^1) + (7 \times 16^0) \\ &= (2 \times 256) + (5 \times 16) + (7 \times 1) \\ &= 512 + 80 + 7 \\ &= 599 \end{aligned}$$

## Machine Language (Part 2 of 5)

### Number System Conversions

In the preceding examples, numbering systems derive decimal equivalents by multiplying the coefficients of the ascending powers of the base. Large numbers are difficult to convert with this method. Simpler methods for converting hexadecimal to decimal and back, and hexadecimal to binary and back are as follows.

#### Hexadecimal to Decimal

To convert a hexadecimal number to decimal:

1. Convert any term represented by a letter symbol to its decimal equivalent.
2. Multiply the high order term by 16.
3. Add the next lower order term to the product obtained in step 2.
4. Multiply the result obtained in step 3 by 16.
5. Add the next lower order term to the product obtained in step 4.
6. Continue multiplying and adding until the low order term has been added.

As an example, convert the hexadecimal number 273 to its decimal equivalent.

$$\begin{array}{r}
 \phantom{x} 2 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 \times 16 \\
 \hline
 \phantom{x} 32 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 + \phantom{x} 7 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 \hline
 \phantom{x} 39 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 \times 16 \\
 \hline
 \phantom{x} 234 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 + \phantom{x} 39 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 \hline
 \phantom{x} 624 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 + \phantom{x} 3 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 \hline
 627
 \end{array}$$

As a second example, convert the hexadecimal number A7B. Converting letter symbols to decimal equivalents yields 10 7 11.

$$\begin{array}{r}
 \phantom{x} 10 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 \times 16 \\
 \hline
 \phantom{x} 160 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 + \phantom{x} 7 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 \hline
 \phantom{x} 167 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 \times 16 \\
 \hline
 \phantom{x} 1002 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 + \phantom{x} 11 \phantom{000} \phantom{00} \phantom{00} \phantom{00} \\
 \hline
 2683
 \end{array}$$

#### Decimal to Hexadecimal

To convert a decimal number to hexadecimal:

1. Divide the decimal number by 16; the remainder of this first division becomes the low order term of the final answer.
2. Divide the quotient (obtained from the first division) by 16; again the remainder becomes a part of the final answer (next higher order term).
3. Repeat steps 1 and 2 until the quotient is less than 16. This final quotient is the high-order term of the final answer.
4. Convert any term between 10 and 15 to its hexadecimal letter-symbol equivalent.

For example, convert the decimal number, 471 to hexadecimal:

$$\begin{array}{r}
 29 \phantom{00} \phantom{00} \phantom{00} \\
 16 \overline{) 471} \\
 \underline{32} \phantom{00} \phantom{00} \phantom{00} \\
 151 \phantom{00} \phantom{00} \phantom{00} \\
 \underline{144} \phantom{00} \phantom{00} \phantom{00} \\
 7
 \end{array}
 \qquad
 \begin{array}{r}
 1 \phantom{00} \phantom{00} \phantom{00} \\
 16 \overline{) 29} \\
 \underline{16} \phantom{00} \phantom{00} \phantom{00} \\
 13
 \end{array}
 \qquad
 \begin{array}{r}
 0 \phantom{00} \phantom{00} \phantom{00} \\
 16 \overline{) 1} \rightarrow 1 \text{ D} \\
 \underline{16} \phantom{00} \phantom{00} \phantom{00} \\
 7
 \end{array}$$

answer = 1D7

#### Hexadecimal to Binary and Binary to Hexadecimal

Hexadecimal 0 through F, which have the decimal values of 0 through 15, respectively, are represented in the binary system by four binary bits. To convert a hexadecimal number to its binary equivalent, express each term in its equivalent four-bit binary group. To convert binary numbers to hexadecimal numbers, reverse the process.

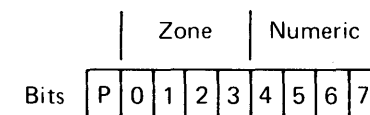
For Example:

Hexadecimal: 3 7 B 1  
Binary: 0011 0111 1011 0001

Hexadecimal: A 6 5 F  
Binary: 1010 0110 0101 1111

#### Data Formats

The basic unit of information in the CPU is the byte. Each byte is eight bits or two hexadecimal characters long. An additional bit (P bit) is added to each 8-bit byte to maintain odd parity. The bit structure of a byte is:



Each main storage address location contains one byte of information. Therefore, each time main storage is addressed a full byte is read from storage.

Each byte is divided into two parts. Bits 4 to 7 represent the numeric portion of a character and bits 0 to 3 represent the zone portion. Therefore, a byte can represent numeric, alphabetic, or special characters. These characters are expressed in EBCDIC (Extended Binary Coded Decimal Interchange Code).

When used as a numeric quantity, each byte contains one numeric digit in bits 4 to 7 with the sign of the entire field contained in the zone portion of the low-order byte. The zone portion of the rest of the bytes in the field contain the EBCDIC for a numeric digit (hex F). The EBCDIC for plus is hex F and for minus hex D. Internally, the CPU also recognizes the ASCII-8 (American National Standard Code for Information Interchange) for minus (hex B) but changes it to EBCDIC in the result field of a decimal operation. The CPU considers any other zone combination to be plus (BA and A).

A B-zone punched over the units position of a field indicates a minus field to the CPU. A plus field contains no zone punch. This chart shows conversion for EBCDIC and card code.

EBCDIC Bits	0123	1100	1101	1110	1111
4567	Zone Digit Pch Punch	BA	B	A	
0000					*φ
0001	1	A	J		1
0010	2	B	K	S	2
0011	21	C	L	T	3
0100	4	D	M	U	4
0101	4 1	E	N	V	5
0110	42	F	O	W	6
0111	421	G	P	X	7
1000	8	H	Q	Y	8
1001	8 1	I	R	Z	9

\*Card Code for Numeric Zero is A Only

The maximum length of a source field is 16 digits and the maximum length of a result field is 31 digits.

## Machine Language (Part 3 of 5)

### Instruction Formats

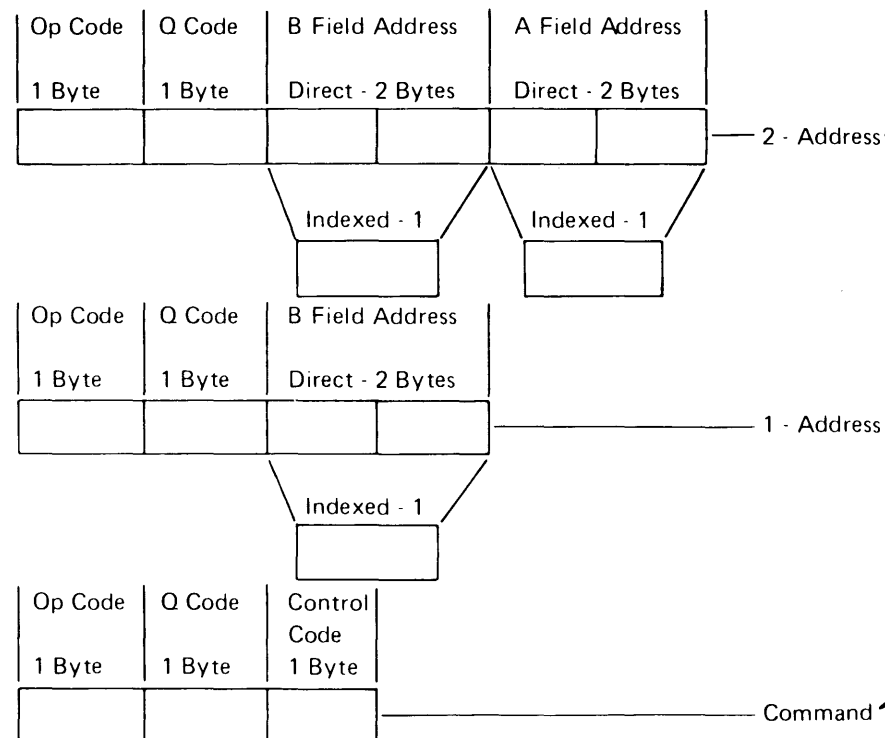
The instruction length is three to six bytes. Bits 0-3 of the operation code determine the type of instruction and addressing.

The CPU executes three types of instructions. They are:

- One address instructions.
- Two address instructions.
- Command instructions.

One address instructions address only one field within main storage and therefore contain one address. (The load address instruction contains the needed data rather than an address.) Two-address instructions are those instructions that address two separate fields within main storage and therefore contain two addresses. Command instructions are those instructions that do not address main storage and therefore contain no addresses.

Each instruction has an operation code and a Q code. Either a control code, or one or two addresses follow them. Thus, the length of the instruction varies from three to six bytes depending upon the type of instruction and the type of addressing specified (direct or indexed).



### Operation Code

The first byte of each instruction is the operation code (op code). This op code specifies the kind of instruction, the method of addressing and the operation to be performed.

The first half byte (bits 0-3) specifies the kind of the instruction and the method of addressing to be used.

Op Code Bits	Number of Bytes in Address	B Field Address	A Field Address
00 00	2	Direct	Direct
01 01	1	Indexed XR1	Indexed XR1
10 10	1	Indexed XR2	Indexed XR2
11 11		No address	No address

If all four bits of the first half byte are on (all ones), the instruction is a command instruction. The bits are grouped in pairs (bits 0-1 and bits 2-3). If both bits in either group are on, the instruction is a one-address instruction; if neither group has both bits on, the instruction is a two-address instruction. If a bit is on in either of the groups in a two-address instruction or if three bits are on in a one-address instruction, the address is indexed. Both bits off in either pair indicates direct addressing.

The second half byte (bits 4-7) of the op code determines the actual operation to be performed. The complete instruction set executed by the CPU is shown in the following chart:

Mnemonic and Operation			Q Code Use <b>A</b>	Control Code Use
0100	ZAZ	Zero and Add Zoned	Field Length	
0110	AZ	Add Zoned Decimal		
0111	SZ	Subtract Zoned Decimal		
1100	MVC	Move Characters		
1110	ALC	Add Logical Characters		
1111	SLC	Subtract Logical Characters		
1101	CLC	Compare Logical Characters		
1010	ED	Edit		
1011	ITC	Insert and Test Characters		
1000	MVX	Move Hex Character		
1100	MVI	Move Logical Immediate	Immediate Data	
1101	CLI	Compare Logical Immediate		
1010	SBN	Set Bits On Masked	Bit Selection	
1011	SBF	Set Bits Off Masked		
1000	TBN	Test Bits On Masked	Register Selection	
1001	TBF	Test Bits Off Masked		
0100	ST	Store Register	Branch	
0101	L	Load Register		
0110	A	Add to Register	Condition	
0010	LA	Load Address		
0000	BC	Branch on Condition	Device	
0001	TIO	Test I/O and Branch		
0000	SNS	Sense I/O	Address and Data Selection	
0001	LIO	Load I/O		
1110	SCP	Store CPU	Register Selection	
1111	LCP	Load CPU		
0000	HPL	Halt Program Level	Halt Identifier (tens)	Halt Identifier (units)
0001	APL	Advance Program Level		
0010	JC	Jump on Condition		
0011	SIO	Start I/O		
0100	CCP	Command CPU		
				Address Modifier
				Stacker Select, Spacing, etc.
				Action to be Performed

### Q-Code

The second byte of each instruction is the Q-code. Depending upon the operation specified, the Q-code is used to further define the instruction **A**.

Refer to the description of each instruction for details of its Q-code.

### Control Code

The control code is the third byte used in the five command instructions **B** and contains additional data pertaining to the command being executed.

### Storage Addresses

Instructions in a one- or two-address format use the third and following bytes for main or register storage addressing.

## Machine Language (Part 4 of 5)

### Addressing

When executing instructions, the CPU uses two types of addressing; direct addressing and indexed addressing.

#### Direct Addressing

Direct addressing requires a two-byte address for each field or location used by the instruction. For one-address instructions, the first two-byte address that follows the Q-code is the address of the result or destination field (B field). For two-address instructions, the first address that follows the Q code is the address of the result or destination field (B field) and the second two-byte address is the source field (A field). The BAR (B address register) maintains the B field address and the AAR (A address register) maintains the A field address.

Most addresses given in the instructions are for the location of the low-order or rightmost digit of the field. Therefore, the CPU executes the instruction, processes each digit position, and decrements the BAR and ARR to address storage in descending order. An exception is the insert and test character instruction. The CPU executes this instruction from high-order to low-order digits and increments the BAR.

#### Indexed Addressing

Indexed addressing provides the programmer with a means of changing addresses within a program without changing the instruction. An indexed address is a single byte within the instruction. This single byte is added to the contents of a two-byte index register to form a new address. This new address is then loaded into the BAR or AAR depending upon the address being indexed.

Indexing is used to: (1) execute an instruction with an indexed address, (2) add a constant to the index register, and (3) branch to an address to execute the instruction at a different main storage location. Thus it is possible to execute an instruction or series of instructions many times without wasting main storage.

Two index registers (XR1 or XR2) are available for indexing. The recognition of indexed addresses and the selection of each index register is covered under *Instruction Formats*.

### Instruction and Execute Cycle

There are two types of machine cycles used in the internal operation of the CPU. These are instruction cycles (I-cycles) and execute cycles (A-cycles and B-cycles).

- I-cycles read out instructions from storage.
- A-cycles and B-cycles execute the instruction.

I-cycles move the instructions from storage to the various registers used to execute the instruction. If the instruction does not require additional use of main storage after the completion of I-cycles, (such as a branch instruction), the operation is complete without execute cycles. However, most operations require the use of data from one or two main storage fields. Execute cycles manipulate this data to perform the operation.

The CPU uses two types of execute cycles; A-cycles to address main storage source fields, and B-cycles to address the main storage result field. If only one field is involved with the instruction, B-cycles address main storage.

### Sequential Instruction Execution

- Instructions are located in consecutive, ascending main storage locations.
- Instruction address register (IAR) is incremented by one each instruction cycle.

If computations are performed manually, several steps can be combined into a single step. For example, in a payroll operation, all deductions can be added together in one step. However, the CPU performs computations in a step-by-step procedure and must add each deduction into the deduction total in a separate operation.

Example:

Manual		CPU	
Federal Tax	17.50	Federal Tax	17.50
State Tax	6.30	State Tax	6.30
Charity	1.50	Total	23.80
Total		Charity	1.50
Deductions	25.30	Total	
		Deductions	25.30

In the example shown the CPU requires three steps. Since it would be desirable to retain the federal tax figure, the total deductions are calculated in a separate location. Therefore, the first step is to reset the total deduction field to zeros and add the federal tax into the total deduction field (zero and add zoned). Then, in separate steps, the state tax and charity fields are added to the total (add zoned decimal).

Because of this sequential method of processing the instructions are in ascending storage locations in the CPU. Instruction sequence is maintained by retaining the address of the storage location in the IAR. The IAR is a two-byte LSR (local storage register) and is incremented each cycle so the next ascending storage location can be addressed.

In the sample program the IAR contains the storage location 1000 which is addressed during an I-cycle. The operation code is read from storage and loaded into the operation register. The IAR is incremented by one so storage location 1001 can be addressed during the next cycle. This process continues until the storage location 1005 has been addressed. The CPU then executes the instruction. After the instruction execution is complete, the IAR again addresses storage location 1006 (operation code of next instruction).

Zero and Add Zoned Operation Code	Q Code (Length of Two Fields)	Address of Total Deductions Field (B Field)	Address of Federal Tax Field (A Field)
00000100			
Storage Location 1000	1001	1002	1003
			1004
			1005

Add Zoned Decimal Operation Code	Q Code (Length of Two Fields)	Address of Total Deductions Field (B Field)	Address of State Tax Field (A Field)
00000110			
Storage Location 1006	1007	1008	1009
			1010
			1011

Add Zoned Decimal Operation Code	Q Code (Length of Two Fields)	Address of Total Deductions Field (B Field)	Address of Charity Field (A Field)
00000110			
Storage Location 1012	1013	1014	1015
			1016
			1017

It is possible to address over 64K positions of storage with the two-byte address by using the ATT (address translate table). Refer to page 2-54 for a complete description of the ATT.

A branch instruction, an I/O cycle, or an interrupt routine alters the sequential instruction addressing previously discussed. An I/O cycle or interrupt routine always returns control to the interrupted program and resumes its program execution where it stopped; a branch instruction does not.



## Machine Language (Part 5 of 5)

### Branching

The CPU can use branching to alter the instruction sequence. Branching provides flexibility within a given CPU program. When the program branches to a different storage location and skips certain steps, it alters the results of the stored program. A sample program that contains a branching operation is shown. In this example, the company has a stock option plan that permits employees who earn \$80.00 or more to purchase stock. If they earn less than \$80.00, the stock deduction is bypassed.

During the subtract zoned decimal operation, the CR (condition register) is set to high, low, or equal depending upon the result. The condition register is set after all arithmetic and compare operations but the CR setting varies with the operation performed. For the subtract zoned decimal operation, the CR is set to low if the result is negative. If \$80.00 is subtracted from the salary field, any salary of less than \$80.00 results in a minus total.

The Q-code bit structure determines the branch condition for a branch on condition instruction. In the following example, the branching condition is a CR low setting. If the CR is set at high or equal, the next sequential instruction (storage location 1028) is executed. However, if the CR which is set at low, the branch-to address (storage location 1034) replaces the next sequential address 1028.

Subtract Zoned Decimal Operation	Q Code (Length of 2 Fields)	Address of Salary Field (B Field)		Address of Minimum Earnings (\$80.00) Field (A Field)	
00000111					

Storage Location      1018      1019      1020      1021      1022      1023

Branch On Condition Operation Code	Q Code (Branch On CR Low)	Branch to Address (Storage Location 1034) (B Field)	
11000000	00000010	00010000	00110100

Storage Location      1024      1025      1026      1027

Add Zoned Decimal Operation Code	Q Code (Length of 2 Fields)	Address of Total Deduction Field (B Field)		Address of Stock Deduction Field (A Field)	
00000110					

Storage Location      1028      1029      1030      1031      1032      1033

### Interrupt Routines

The 5415 CPU has eight interrupt levels. Five of these levels are devoted to I/O devices and three levels support CPU functions.

The supervisor program, the interval timer, and the program check interrupt are all assigned an interrupt level which supports the CPU operation.

An interrupt routine interrupts the main program with a separate program routine. For this reason, all interrupts except program check interrupts can occur only at the completion of an instruction. Program check interrupts occur in the CPU cycle following the program check.

Interrupts are assigned priorities; the highest interrupt level takes precedence over lower level interrupts which means it is possible for one interrupt routine to interrupt another routine of lower priority. The assigned priority from high to low is:

Program level 7	Program check
6	Interval timer/Unit record restart
4	SIOC
3	MLTA/BSCC
2	BSCA
5	Op-end
1	CRT
0	Supervisor program

### I/O Cycle Steal

The CPU issues a start I/O (SIO) instruction which starts the I/O device. Whenever the I/O device reaches a point in its mechanical operation where it needs data from storage (write, print, punch) or has data to send to storage (read), the device requests an I/O cycle. The CPU uses two methods for transferring data to and from the various I/O devices: cycle steal and sense/load during an interrupt routine.

The 5424 uses the cycle steal method of transferring its data. An I/O cycle request can occur during any cycle and is granted before the next CPU cycle. The attachmnet then has complete control of data flow, LSR selection, and storage during that cycle.

Since more than one device attachment may request a cycle at a time, each device is assigned a cycle steal priority.

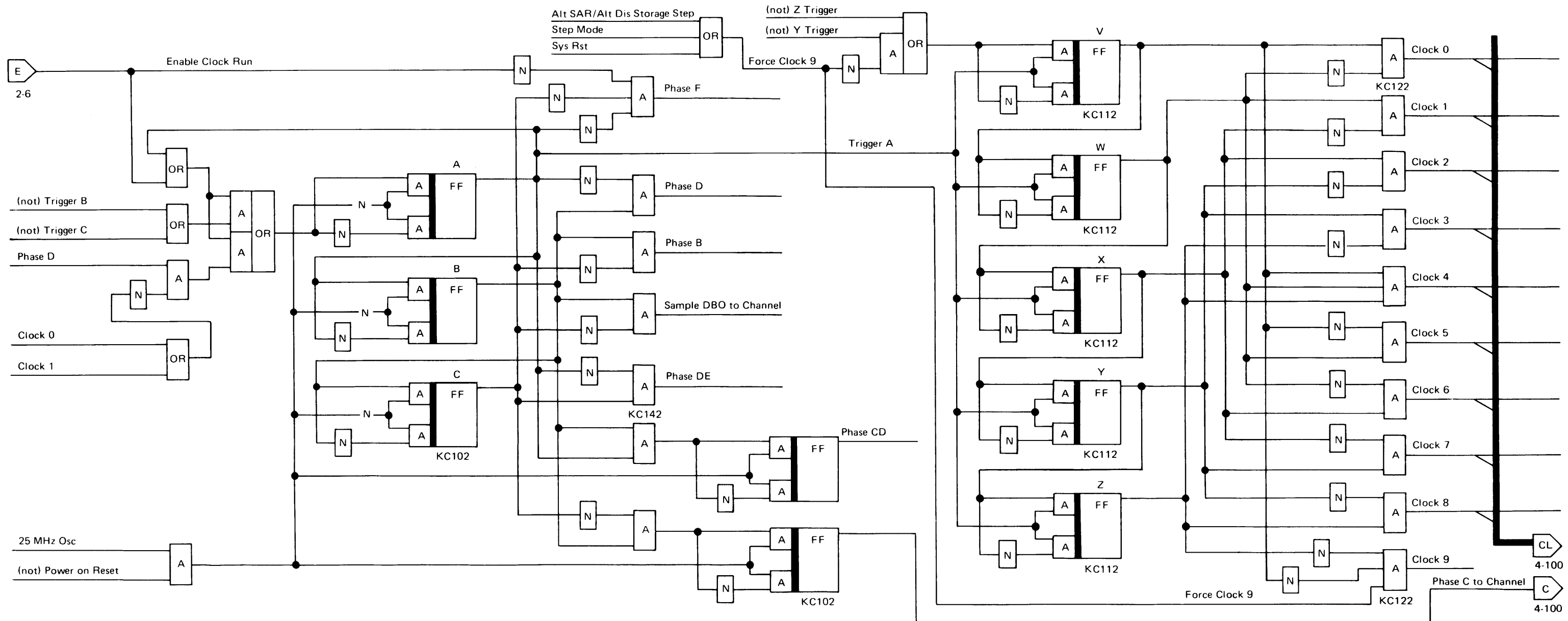
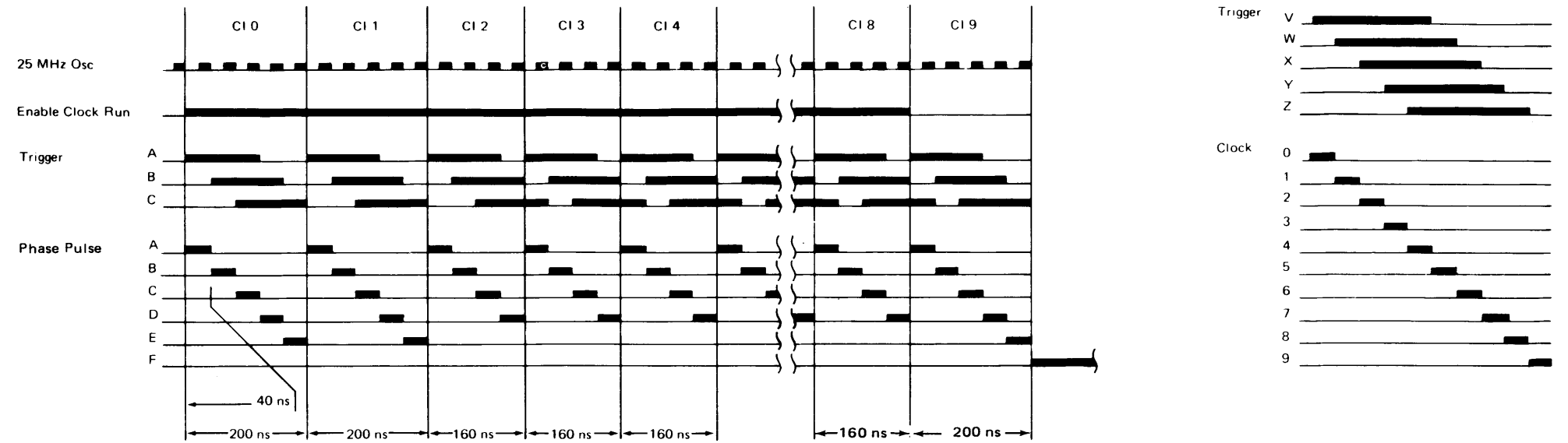
Priority	Device
1-2	Spare
3	Magnetic Tape
4	SIOC
5	Spare
6	BSCA-1
7	MFCU Read, Punch/MFCM Read, Punch, Print
8	2501 Card Reader
9	MLTA/BSCC
10	BSCA-2/Display Adapter
11	1442 Card Read Punch
12	MFCU Print
13-14	Spare
15	3340/3344
16	3741
17	1403 Printer
18	3277 Display Station/Keyboard
19-20	Spare

This page intentionally left blank.

# PROCESSOR FUNCTIONAL UNITS

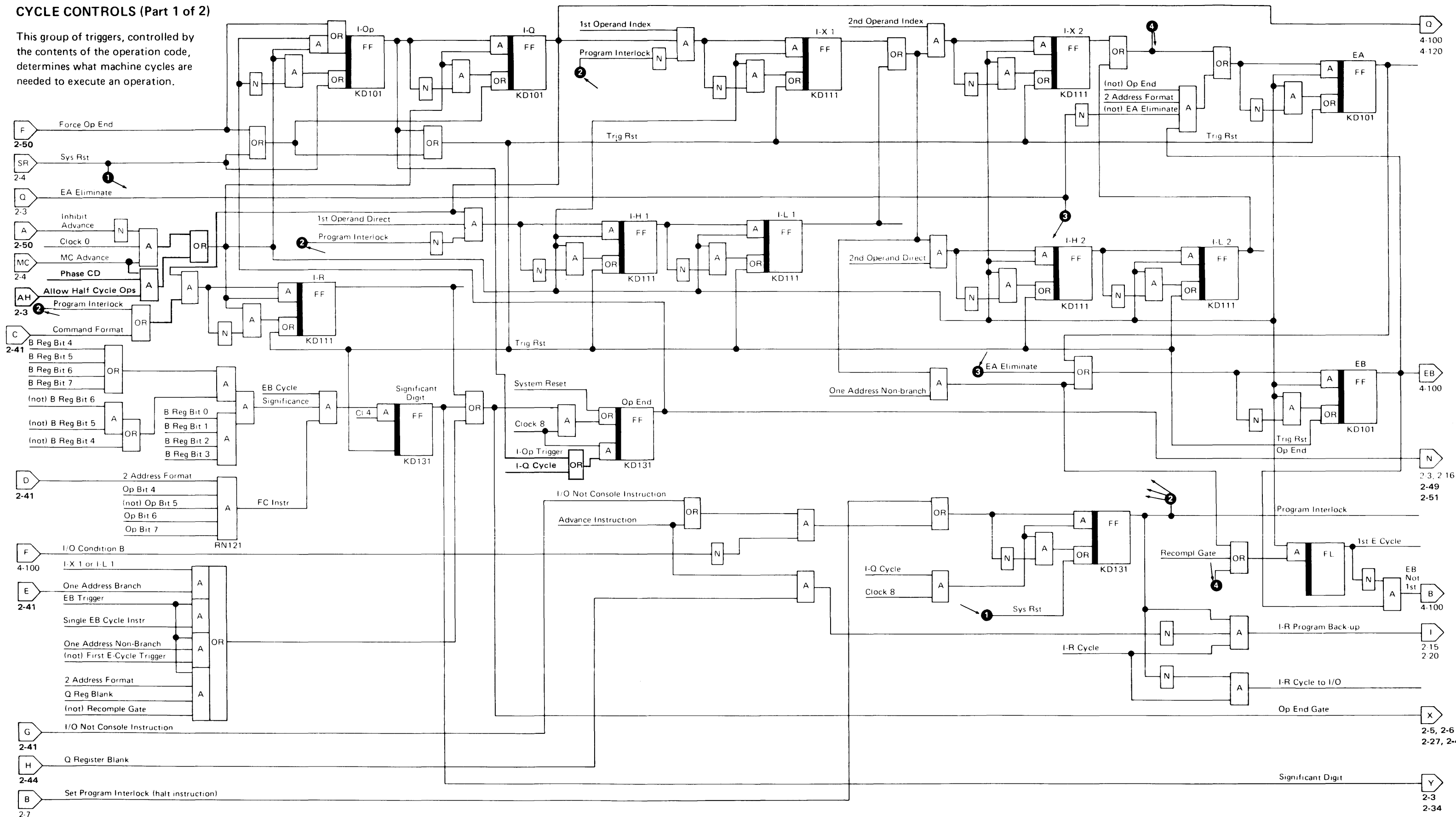
## CPU CLOCK

A 25 MHz oscillator drives this group of binary triggers which supply the basic timing pulses for both the CPU internal operations and the I/O attachments. There are nine (0-8) basic clock times. (The CE can force a clock nine for diagnostics.) Clock times 0, 1 and 9 are each 200 ns long while the remaining clock times are each 160 ns long. All clock times are divided into 40 ns phase pulses. Clock times 0, 1 and 9 have five phases (A through E); the other clock times have only four (A through D).



### CYCLE CONTROLS (Part 1 of 2)

This group of triggers, controlled by the contents of the operation code, determines what machine cycles are needed to execute an operation.



F 2-50 Force Op End

SR 2-4 Sys Rst

Q 2-3 EA Eliminate

A 2-50 Inhibit Advance

MC 2-4 Clock 0

A 2-4 MC Advance

AH 2-3 Allow Half Cycle Ops

C 2-41 Command Format

B Reg Bit 4  
B Reg Bit 5  
B Reg Bit 6  
B Reg Bit 7  
(not) B Reg Bit 6  
(not) B Reg Bit 5  
(not) B Reg Bit 4

D 2-41 2 Address Format  
Op Bit 4  
(not) Op Bit 5  
Op Bit 6  
Op Bit 7

F 4-100 I/O Condition B

E 2-41 One Address Branch  
EB Trigger  
Single EB Cycle Instr  
One Address Non-Branch  
(not) First E-Cycle Trigger

G 2-41 I/O Not Console Instruction

H 2-44 Q Register Blank

B 2-7 Set Program Interlock (halt instruction)

Q 4-100  
4-120

EB 4-100

N 2-3, 2-16  
2-49  
2-51

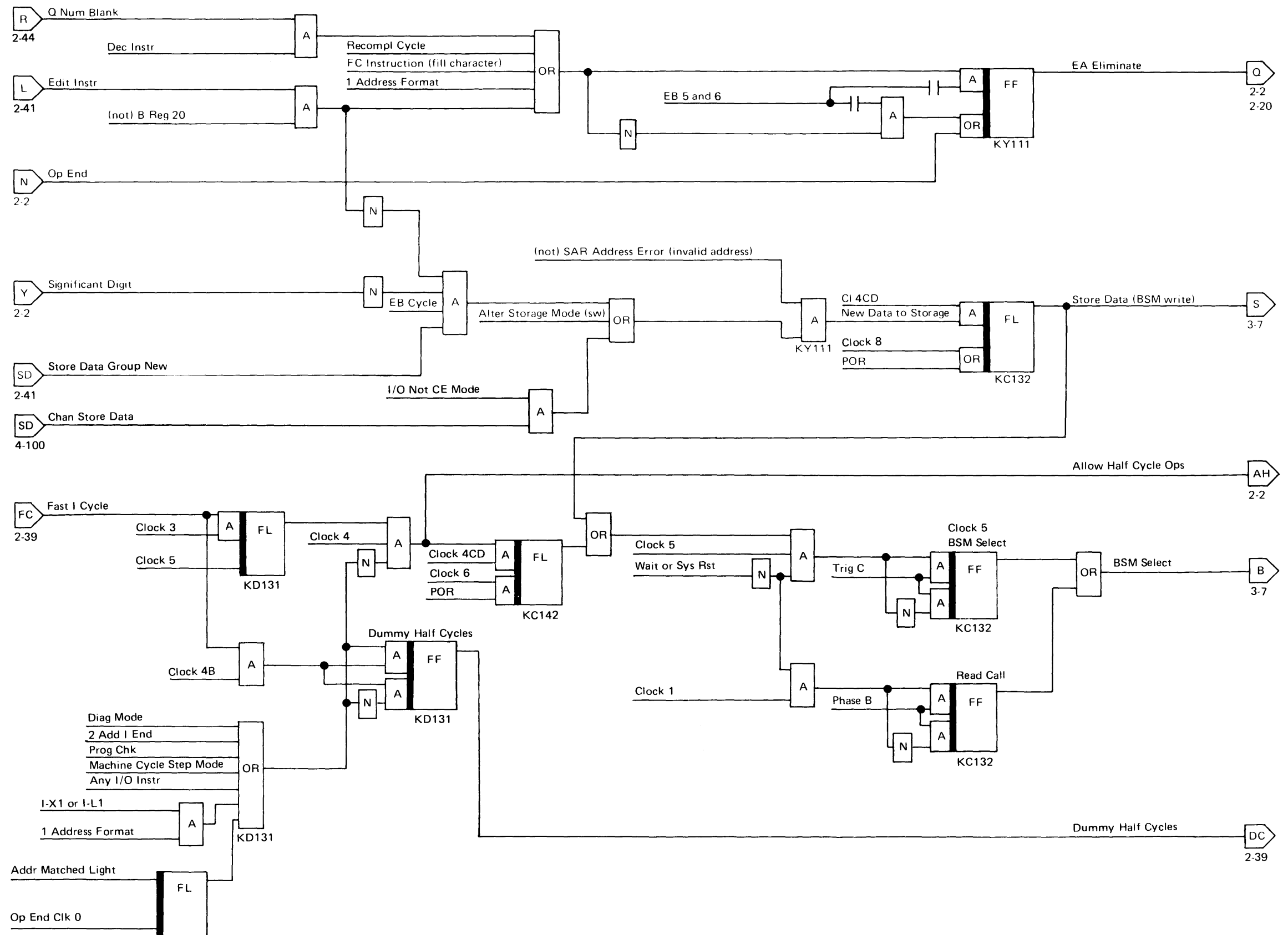
B 4-100

I 2-15  
2-20

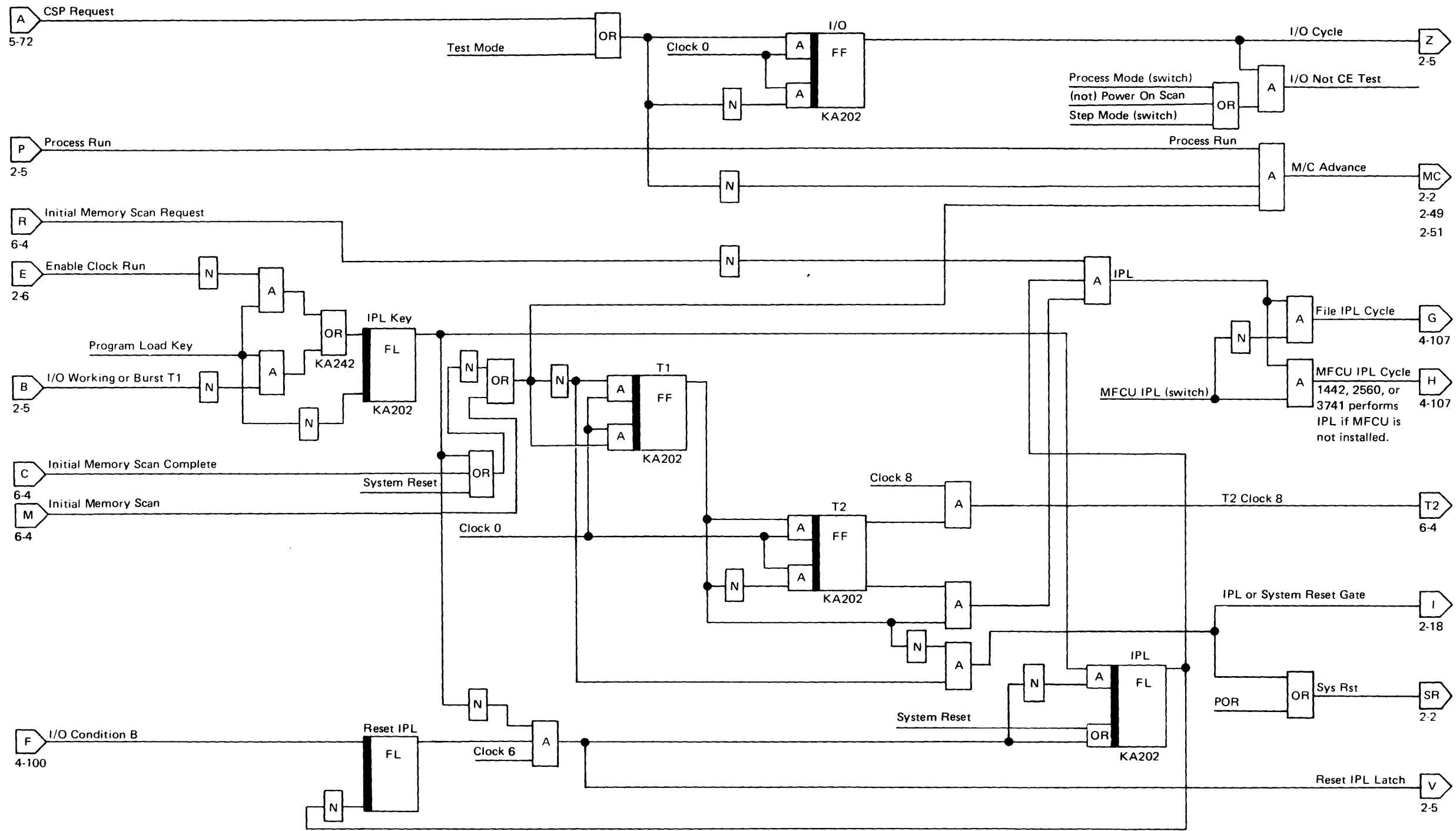
X 2-5, 2-6  
2-27, 2-4

Y 2-3  
2-34

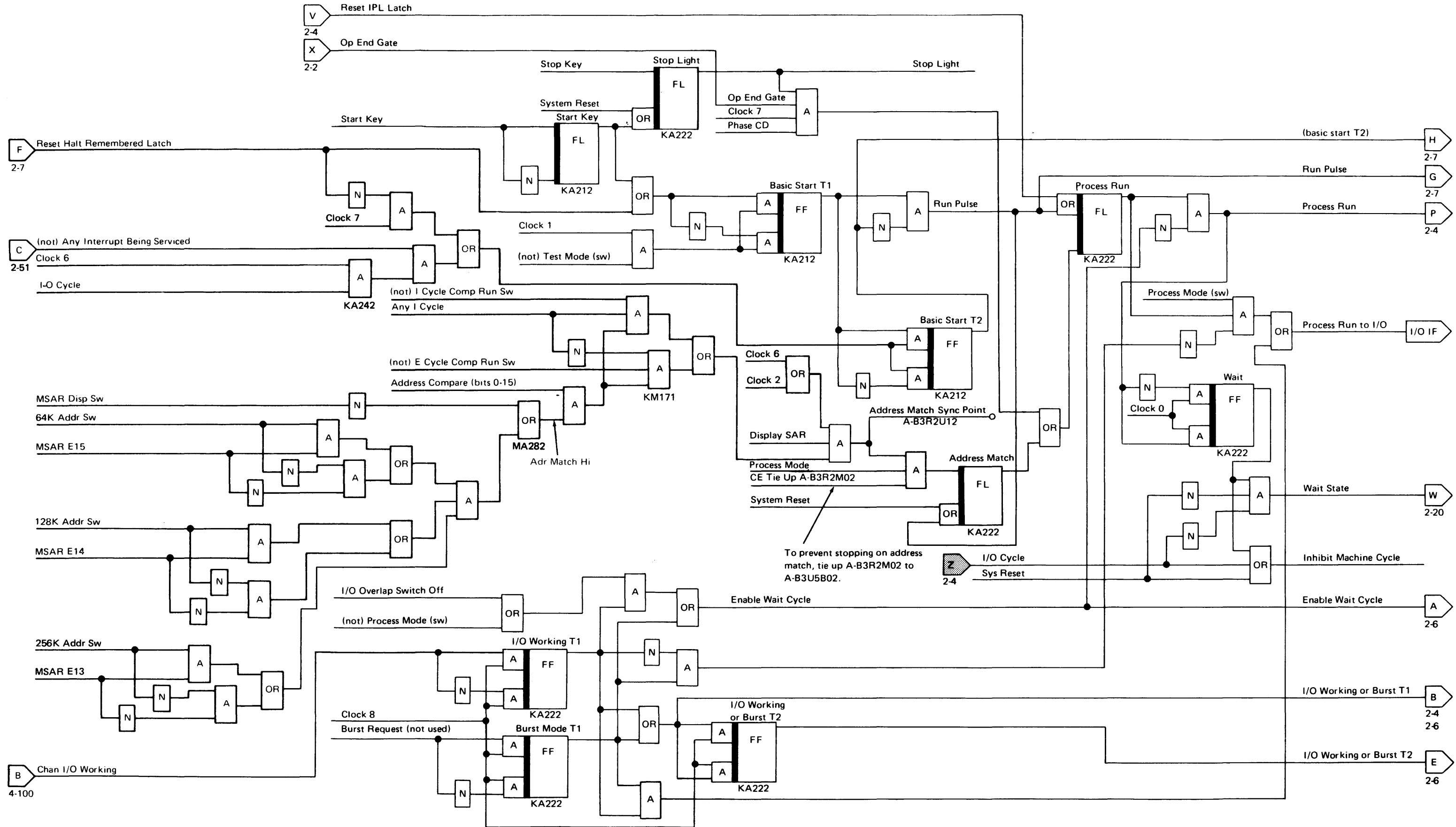
CYCLE CONTROLS (Part 2 of 2)



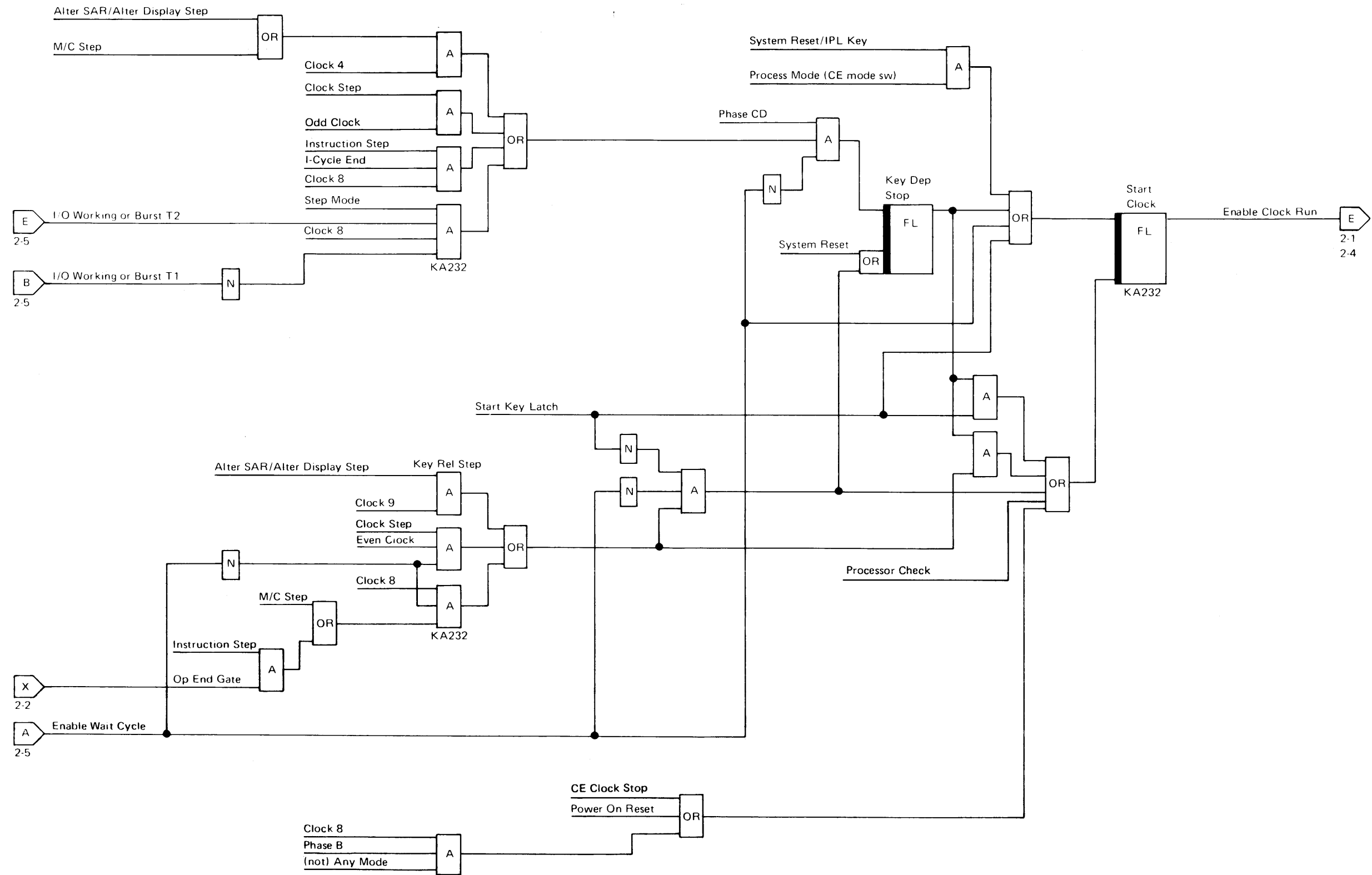
RUN CONTROLS (Part 1 of 4)



RUN CONTROLS (Part 2 of 4)

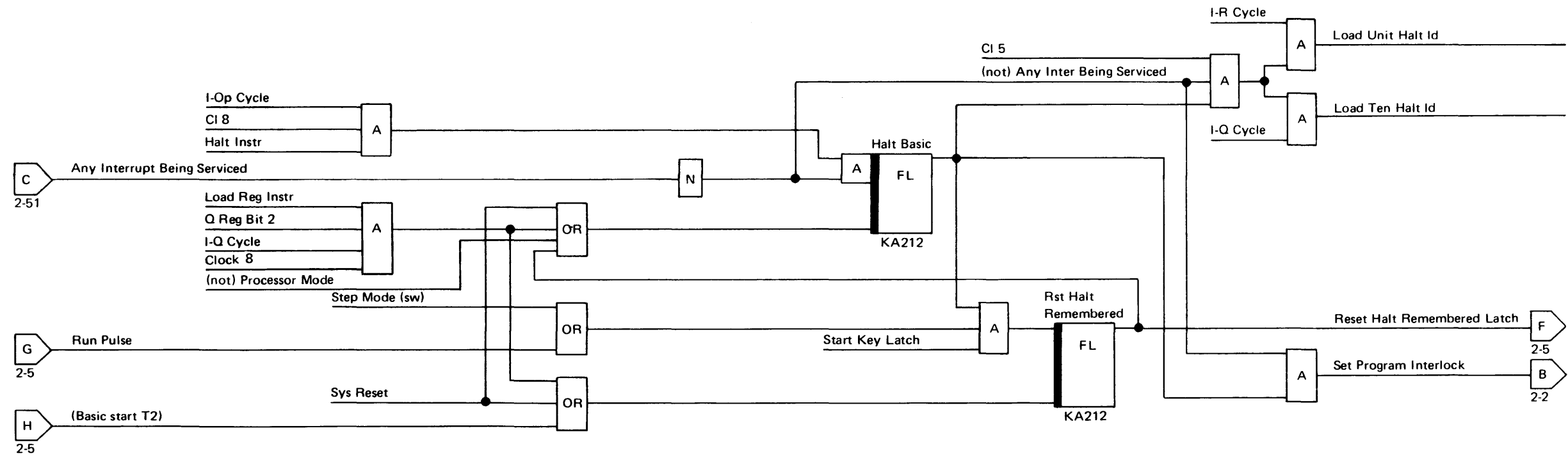


RUN CONTROLS (Part 3 of 4)





RUN CONTROLS (Part 4 of 4)



## ALU

The ALU (arithmetic and logic unit) is a multiple function unit that receives information from the A and B registers and performs the following functions with the A and B register data:

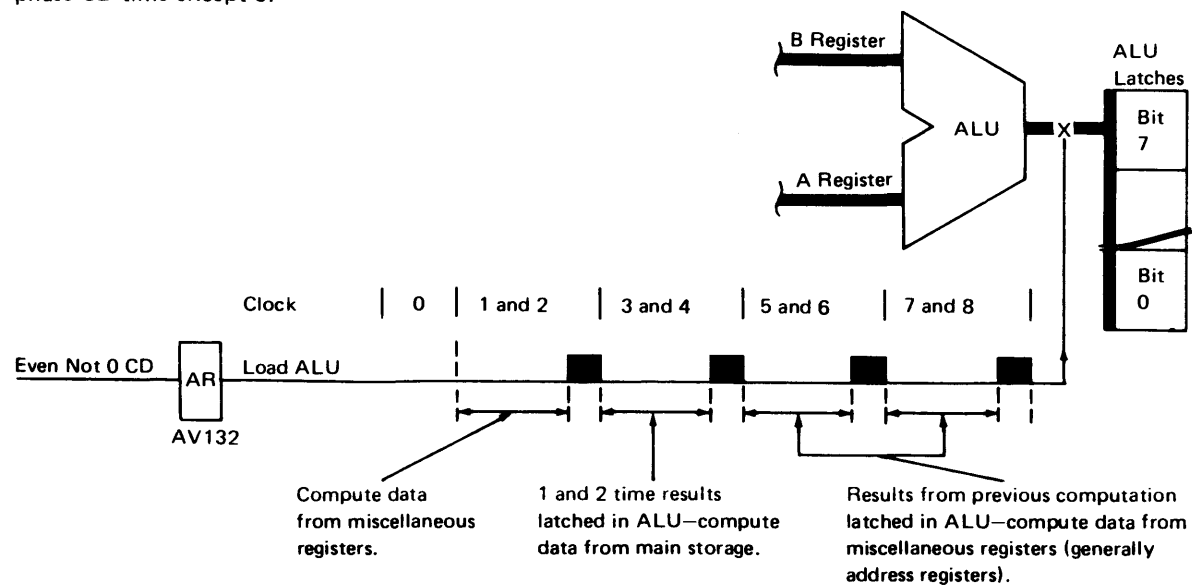
- Logical OR
- Logical AND
- Tests for presence or absence of bits
- Pass B register through
- Pass A register through
- Binary subtract
- Binary add
- Decimal subtract
- Decimal add

The results of each computation are available in the ALU latches while ALU circuits are starting the next computation.

Data from the A and B registers, except the P bits, enters the ALU in parallel form and results in a single byte output. Each ALU position (bit 7 through bit 0) consists of a group of AND, OR, and exclusive OR blocks which give the correct output for the desired function. Because the output parity is not necessarily the same as the input, correct parity is generated for the results.

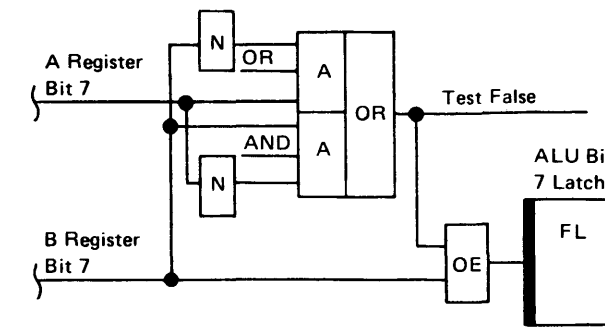
The output from the ALU is sent to main storage, local storage registers, operation register, Q register, condition register, and data bus out. The use of this output is discussed with individual operation descriptions.

The ALU processes a full byte of information at a time. Register A or B can supply the full byte or depending upon the operation only one-half of the byte. The ALU is used four times during each machine cycle and is loaded each even clock phase CD time except 0.

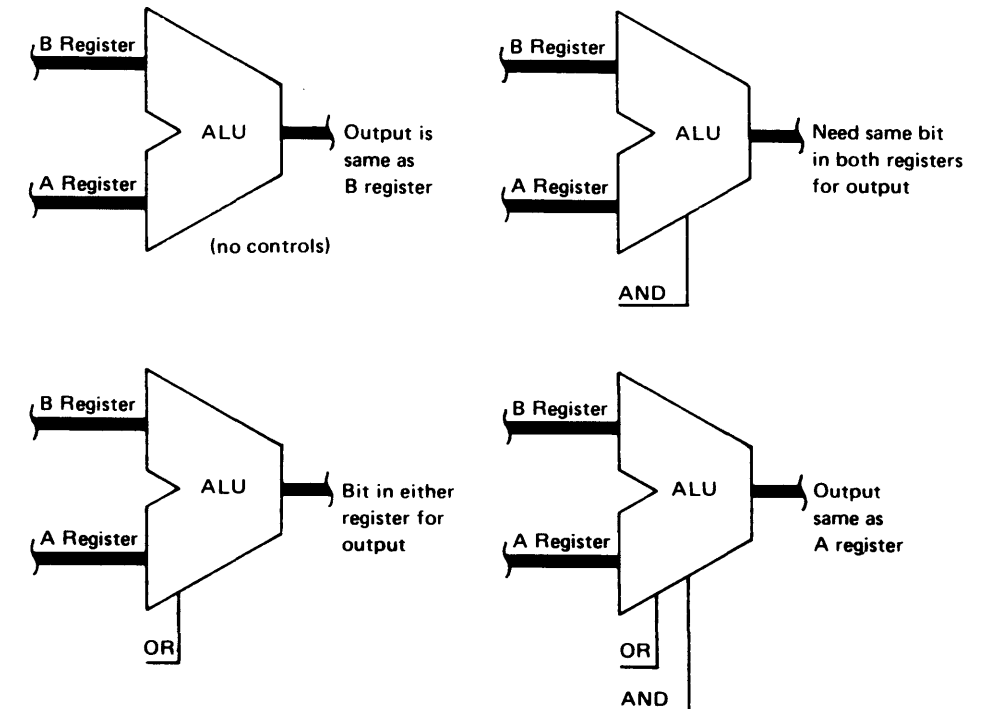


## AND/OR and Test False

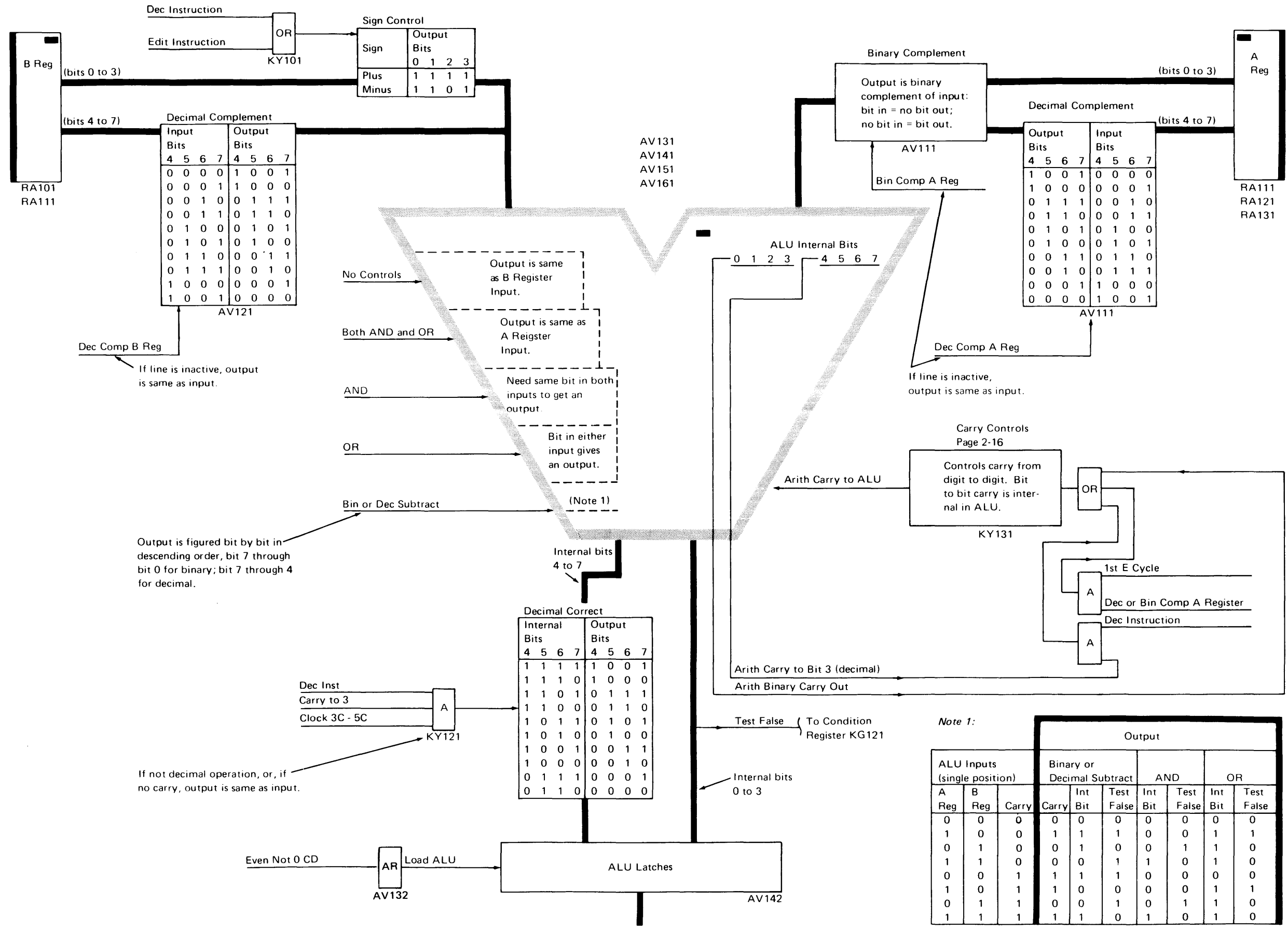
The following figure illustrates the AND/OR and test false functions for a single bit position. Output from the test false lines sets the CR test false latch. The output depends upon the presence or absence of bits in the A and B registers and the active control line (AND or OR). Use of test false output is discussed with individual operation descriptions.



The AND function is a bit-by-bit comparison of the two registers. If the same bit is on in each register the output is active. The OR function output is active if a bit is present in either register. Outputs available through the use of the AND and OR control lines are:



# ALU DATA FLOW

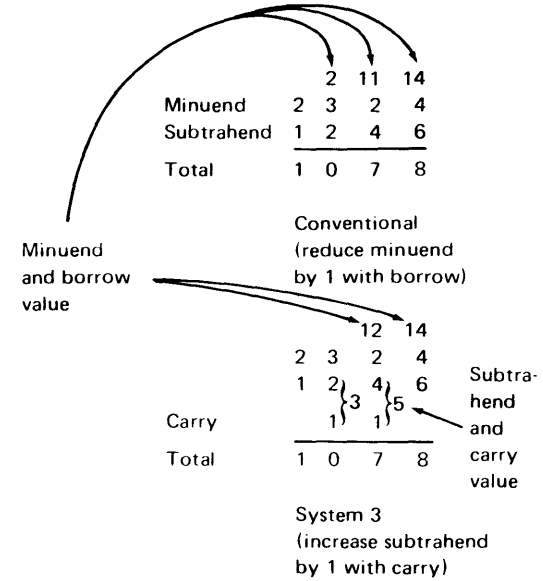


Note 1:

ALU Inputs (single position)			Binary or Decimal Subtract			AND		OR	
A Reg	B Reg	Carry	Carry	Int Bit	Test False	Int Bit	Test False	Int Bit	Test False
0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	1	1
0	1	0	0	1	0	0	1	1	0
1	1	0	0	0	1	1	0	1	0
0	0	1	1	1	1	0	0	0	0
1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	0	1	1	0
1	1	1	1	1	0	1	0	1	0

### Binary Subtraction

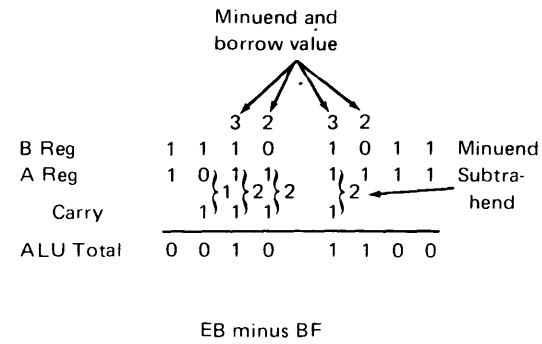
The following figure shows a decimal comparison between the conventional method of subtraction and the type of subtraction used by the System/3.



Under the conventional method, whenever it is necessary to borrow from the next position, the minuend is increased by 10 in the position where the subtraction is taking place and decreased by 1 in the position that is borrowed from. For instance, when subtracting the 6 from the 4 in the units positions, after borrowing from the tens position, the units position of the minuend becomes 14. Because of the borrow, the tens position is reduced to 1. In this example, another borrow is necessary so, after the borrow the tens positions of the minuend is 11; this method continues to the end of the problem.

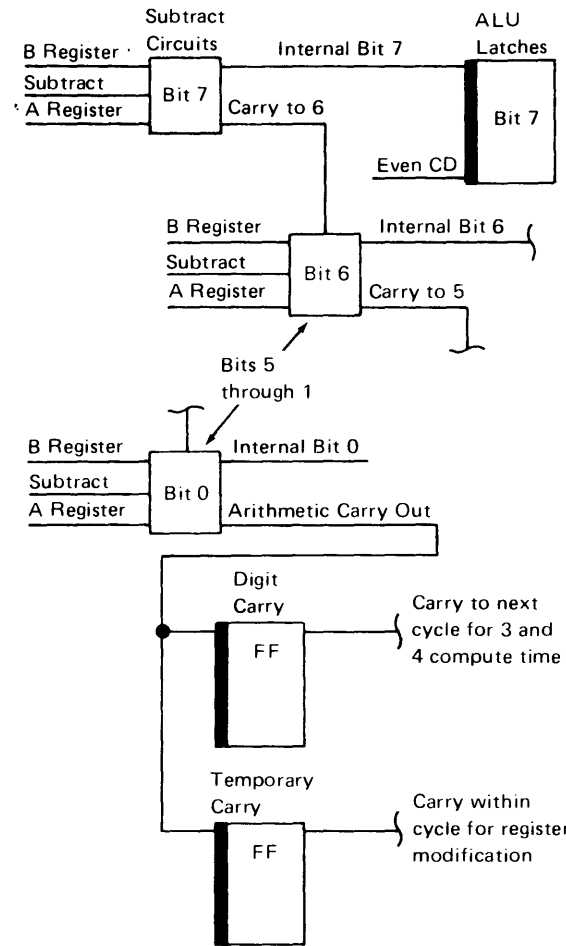
The same result is reached if, instead of reducing the minuend by 1 after borrowing, the subtrahend is increased by 1 with a carry. Thus, in the tens position of the preceding example, the carry method subtracts 5 from 12 instead of 4 from 11 as in the conventional method.

Binary subtracting is done in the same way except for the value of the borrows. Because decimal numbers have ascending powers of 10, a borrow has value of 10. Similarly, since binary numbers have ascending powers of 2, a borrow has a value of 2. The following figure illustrates this by subtracting the hexadecimal value BF from the hexadecimal value EB.

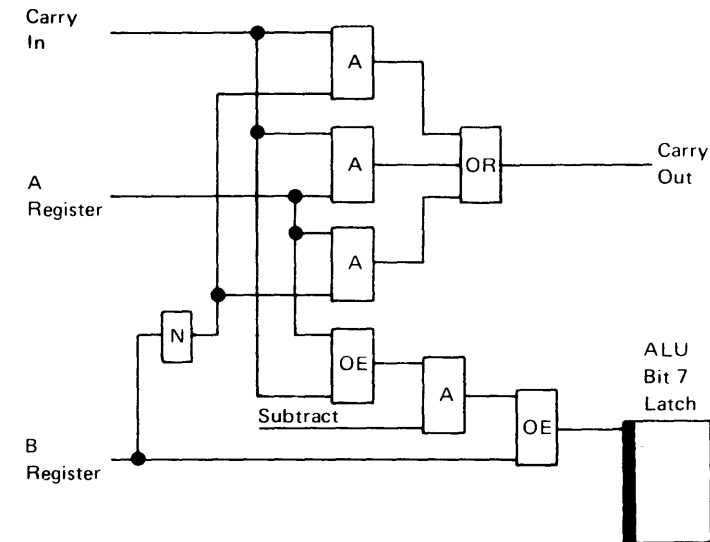


After subtracting the first two positions, it becomes necessary to borrow in order to subtract the third position. This borrow has a value of 2 and the result of the subtraction is 1. Using the carry method of subtraction, a carry to the fourth position gives the subtrahend a value of 2. This forces a borrow from the next position which, when added to the minuend, gives a value of 3. This method continues to the end of the problem.

The minuend enters the ALU from the B register and the subtrahend enters from the A register.



The subtrahend is subtracted bit-by-bit, starting with bit 7 and continuing through bit 0. Carries from bit to bit are internal but if there is a carry from bit 0 it is held in the carry triggers until it is needed in bit 7. This figure illustrates the subtract function for a single bit position.



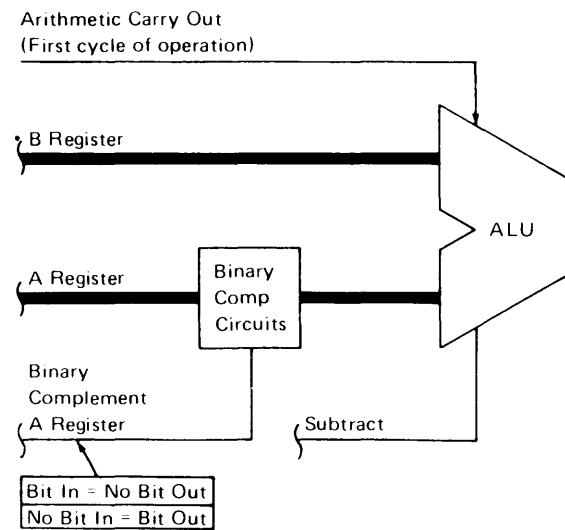
## Binary Addition

Binary addition is accomplished by complementing the A register and subtracting the result from the B register. The A register is complemented when a bit is replaced by a no-bit and a no-bit is replaced by a bit. In order to get a true complement, it is necessary to force a carry into the low-order bit of the first character. For example, to add the hexadecimal values 8F 83 and 3F 93:

Borrowed Amount	32	222 22	
Minuend (true)	1000 1111	1000 0011	
Subtrahend (complement)	1100 0000	0110 1100	
Carry	1	1111 1 1	Forced Carry
ALU Total	1100 1111	0001 0110	

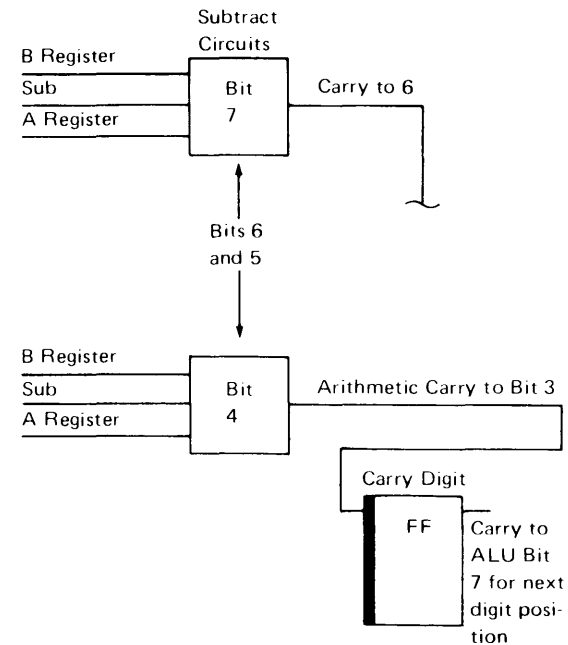
8F 83 plus 3F 93

The ALU controls and circuits for binary addition are similar to binary subtract. The exceptions are: the A register input to ALU is complemented; and a carry is forced into bit 7 during the first cycle.



## Decimal Subtraction

The ALU subtracts the binary numbers in bytes. However, because a single decimal number only uses one-half of each byte, the ALU must subtract each half separately. Thus, a carry from bit 4 to bit 3 is used to set the digit carry trigger.



Subtraction is done the same way as binary subtraction as long as the minuend (B register) is larger than the subtrahend (A register). But, because the ALU is capable of handling digits up to 15 (hex F) and in this case the digits have a maximum value of 9, the ALU reaches an incorrect decimal result when there is a borrow from the next digit (A register larger than B register). This difference of 6 must be subtracted from the result in order to reach a correct result.

B Larger Than A	A Larger than B
8 - 3	3 - 8
B register 222	B register 2
A register 1000	A register 0011
Carry 0011	Carry 1000
Total 111	Total 1011
0101	1011

Note: Subtract and complement functions affect only the digit portions. The zone portion is not affected.

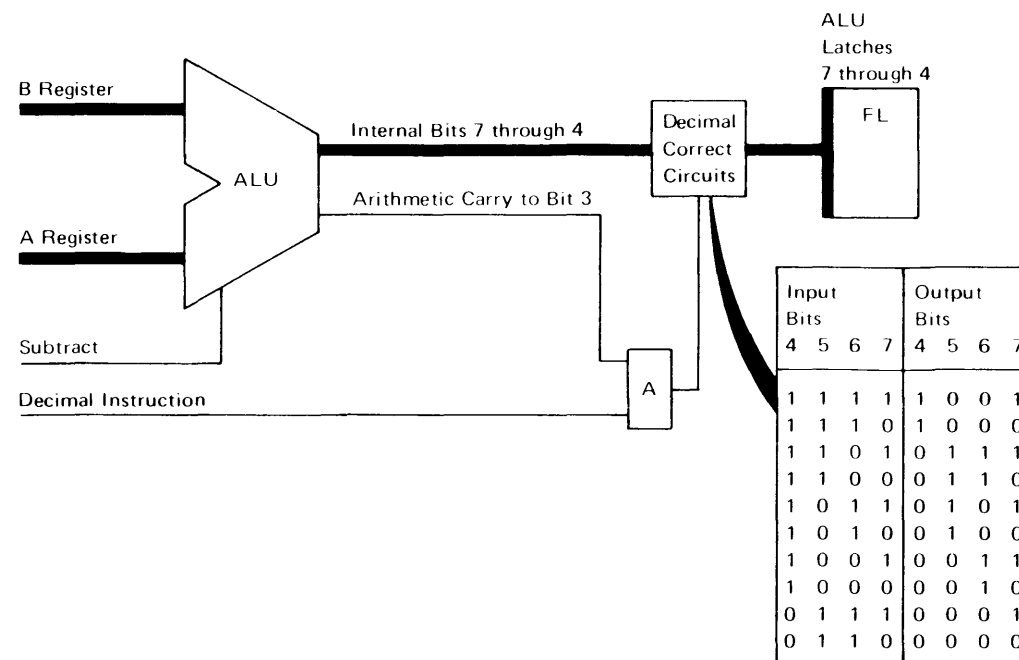
	2
Total	1011
Decimal Correct	0110
Carry*	1
Corrected Total	0101

\*Mathematical carry; not done by carry circuit

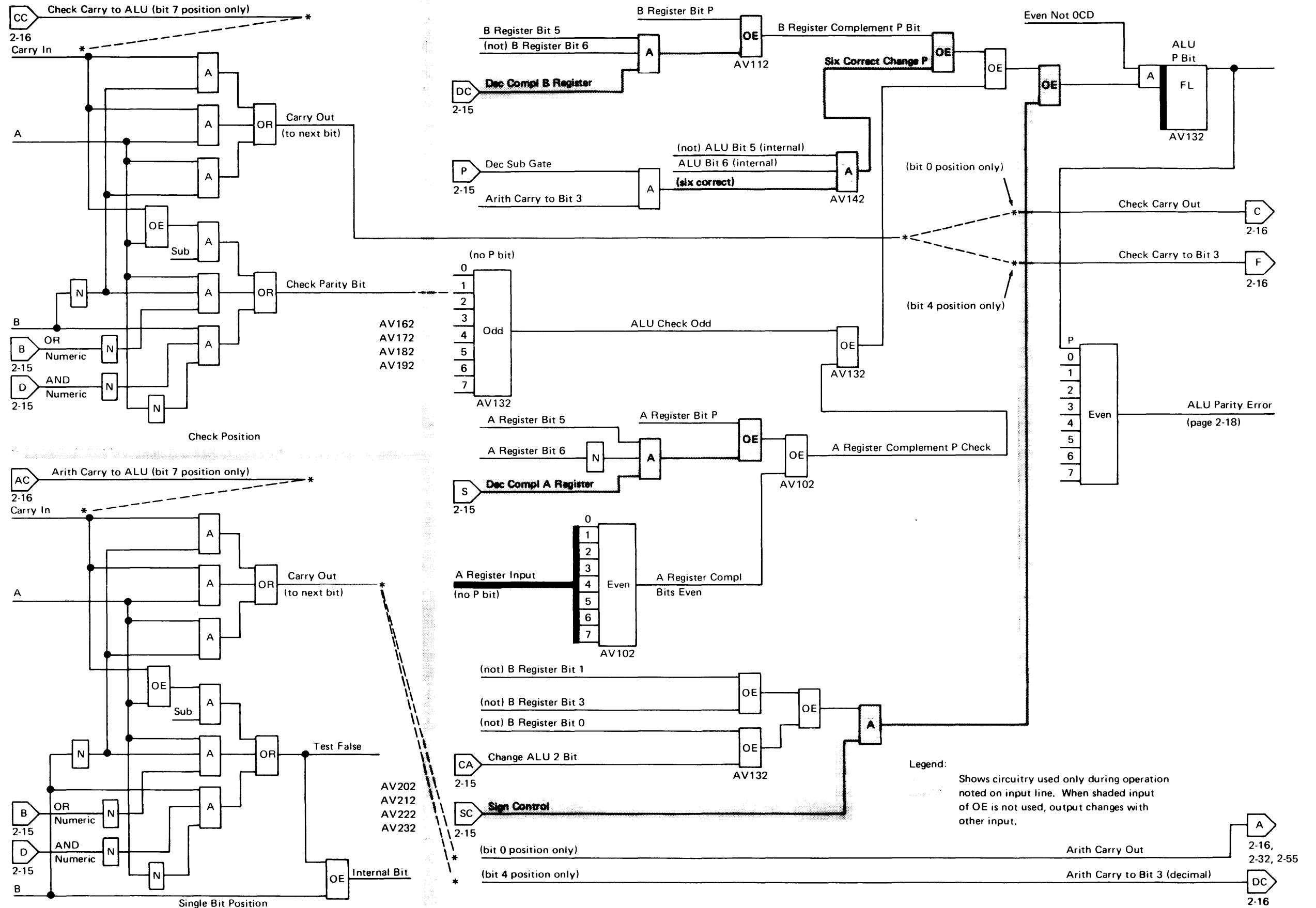
The decimal correct circuits are activated by the carry from the bit 4 position.

The following figure shows the data flow for decimal subtraction and contains a table of the bit correction for the legitimate decimal characters.

Bits 0 to 3 of the low-order decimal character contain the sign of the field. The ALU output for bits 0 to 3 is determined by the sign control circuits and is discussed with the individual operation descriptions.



ALU P Bit Generation



## Decimal Addition

To add decimal digits, it is necessary to complement the A register and subtract it from the B register. However, since the characters being complemented are decimal, the binary equivalent of the 9s complement is used.

B Register	2233	
	0011	
A Register Complemented	0111	
Carry*	1111	← Forced
Total	1011	

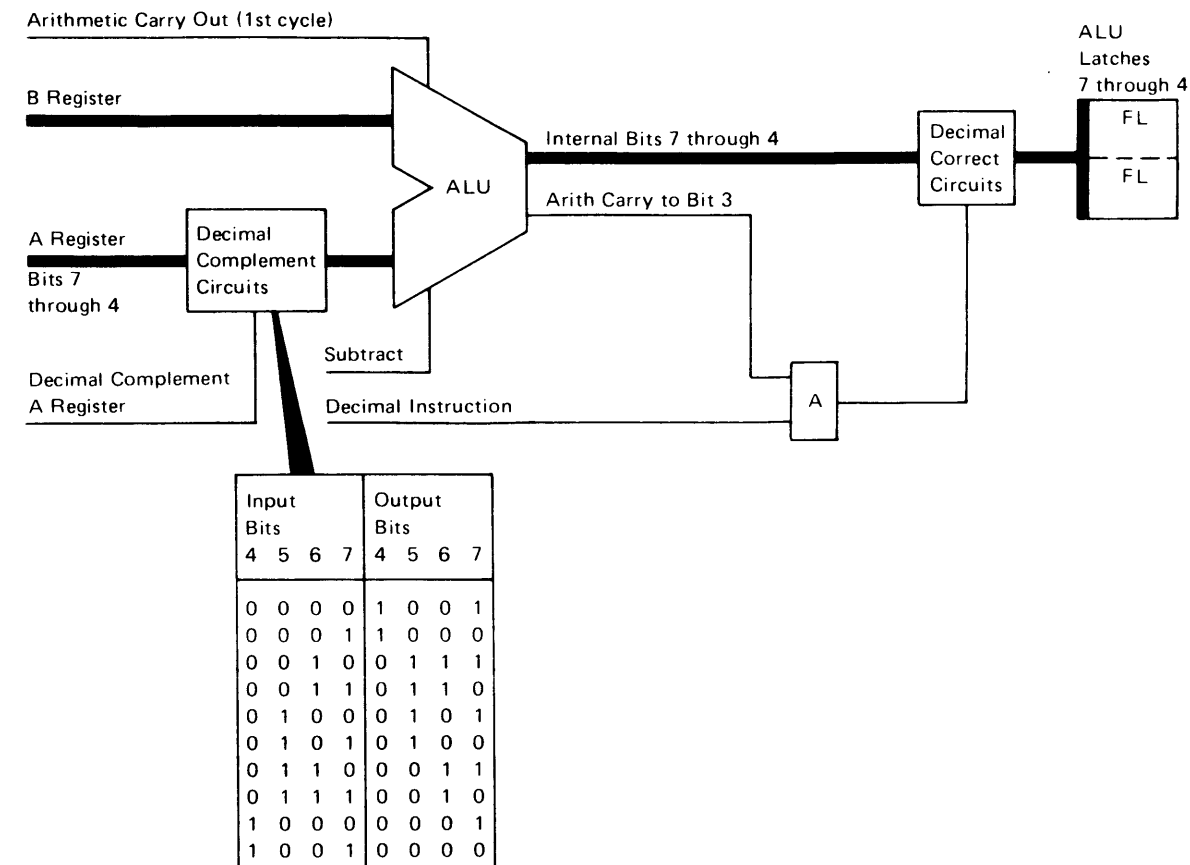
Total	2
Decimal Correct	1011
Carry*	1
Corrected Total	0101

B Register (3) plus A Register (2)

\*Mathematical carry; not done by carry circuits

As in subtraction, if the complemented A register digit is larger than the B register, the result must be corrected.

The following figure shows the decimal add data flow with a table for the 9s complement of the legitimate decimal digits.

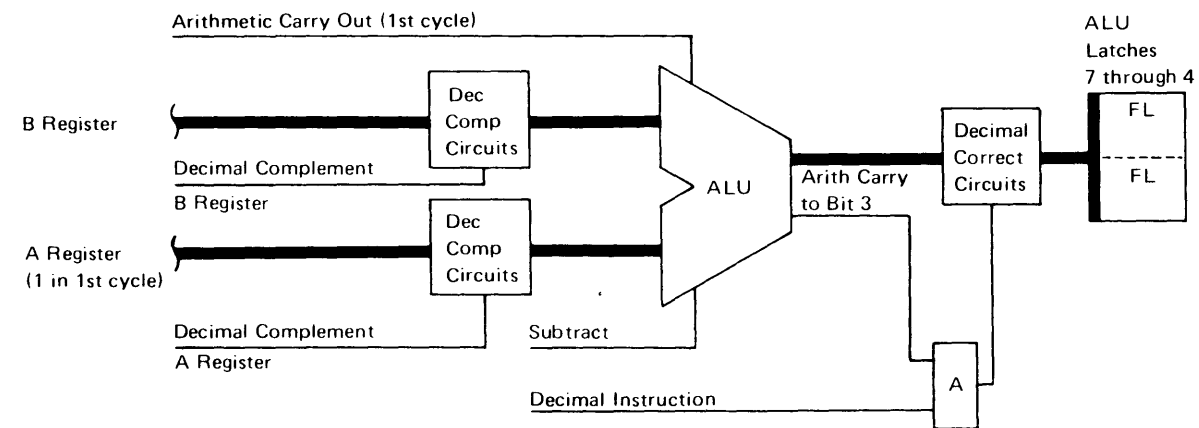


### Recomplement

Decimal adding or subtracting under certain conditions produces a result that is a complement form of the correct result. This requires a recomplement operation whereby the result is fed through the ALU a second time to change it to its true form. A recomplement is necessary if the operation is:

- An add operation with unlike signs for the two fields and the A-field is larger than the B-field.
- A subtract operation with like signs for the two fields and the A-field is larger than the B-field.
- A result is minus zero.

The need for a recomplement cycle is signaled by a carry from the high order position of the field. In the case of a minus zero result, the recomplement is signaled by the condition register and is covered under the decimal operations. This figure contains the data flow for recomplementing.



The following figure illustrates the method used to recomplement. After subtracting 52 from 27, the ALU result is 75. To recomplement, the result is fed back through the B register and is decimal complemented before entering the ALU. The A register is set to 1 on the first cycle and is left blank each remaining cycle. The decimal complement of the A register is subtracted from the B register complement. A carry is forced into the first digit just as in regular complementing. The final result is the 10's complement of the original result.

				Subtract Cycles			
Borrowed Amount	22	2					
B Register (9s Complement)	0010	0111					
A Register (9s Complement of 1)	0101	0010					
Carry	1	1					
Total	1101	0101	→	Borrowed Amount	32		
				Total	1101	0101	
				Decimal Correct	0110		
				Carry*	11		
				Total	0111	0101	

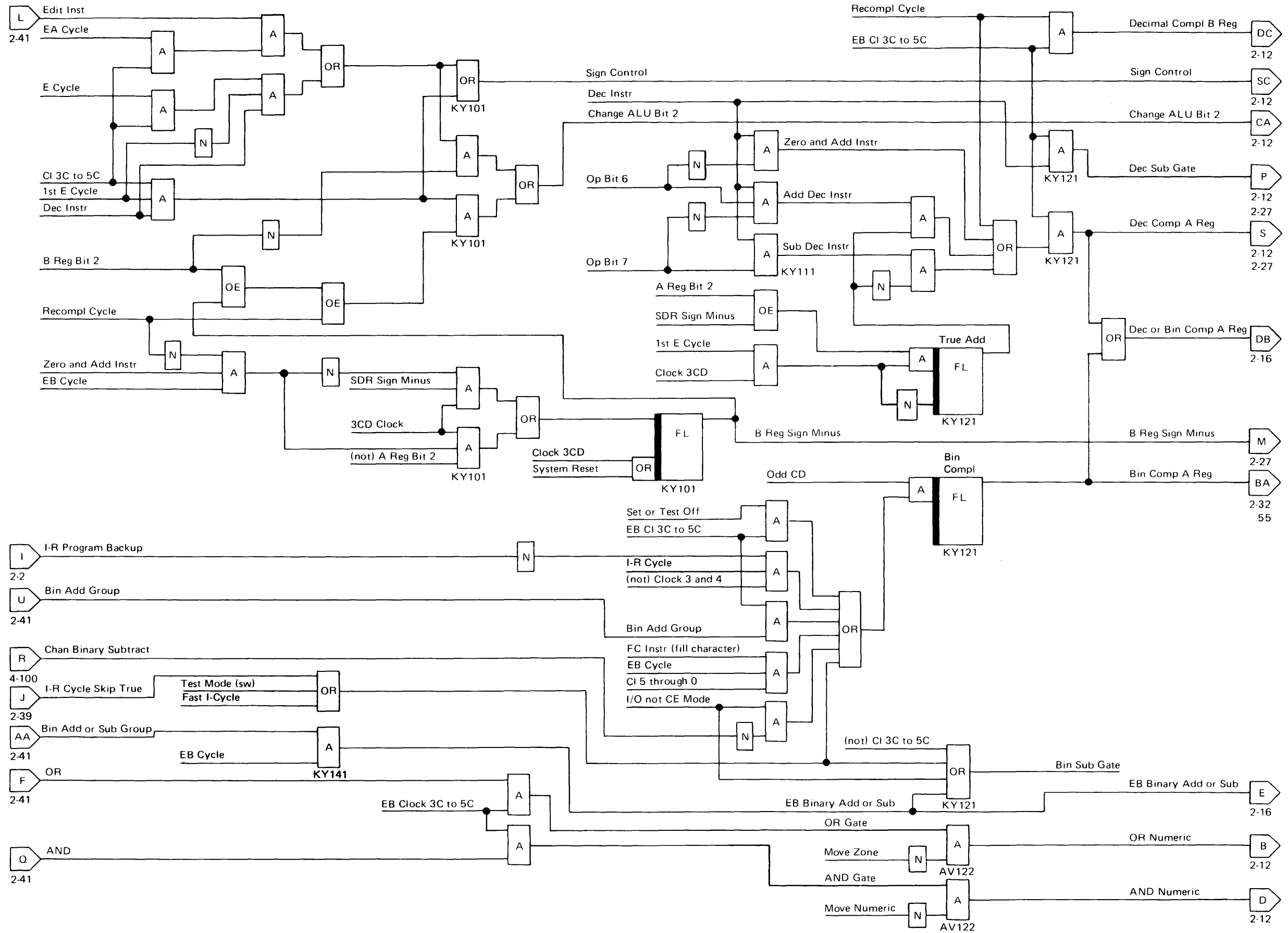
  

				Recomplement Cycles			
Borrowed Amount	2	2	2	22			
B Register (9s Complement)	0010	0100					
A Register (9s Complement of 1)	1001	1000	← Forced				
Carry	11	111					
Total	1000	1011	→	Borrowed Amount	22	2	
				Total	1000	1011	
				Decimal Correct	0110	0110	
				Carry*	11	1	
				Total	0010	0101 (minus)	

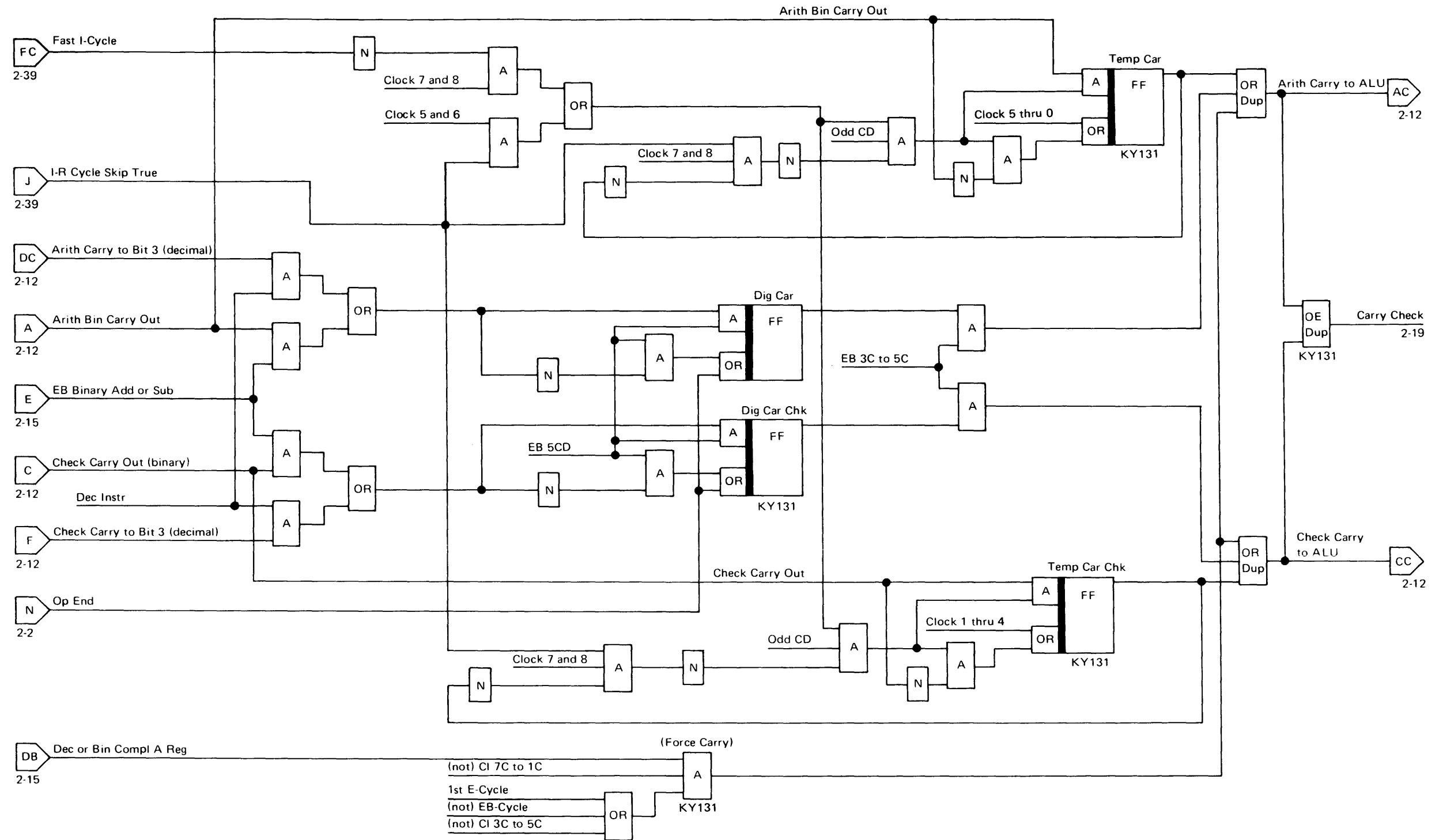
\*Mathematical carry; not done by carry circuits



ALU CONTROLS (Part 1 of 2)



ALU CONTROLS (Part 2 of 2)



## CHECK ALU

### Parity Generation and Parity Check

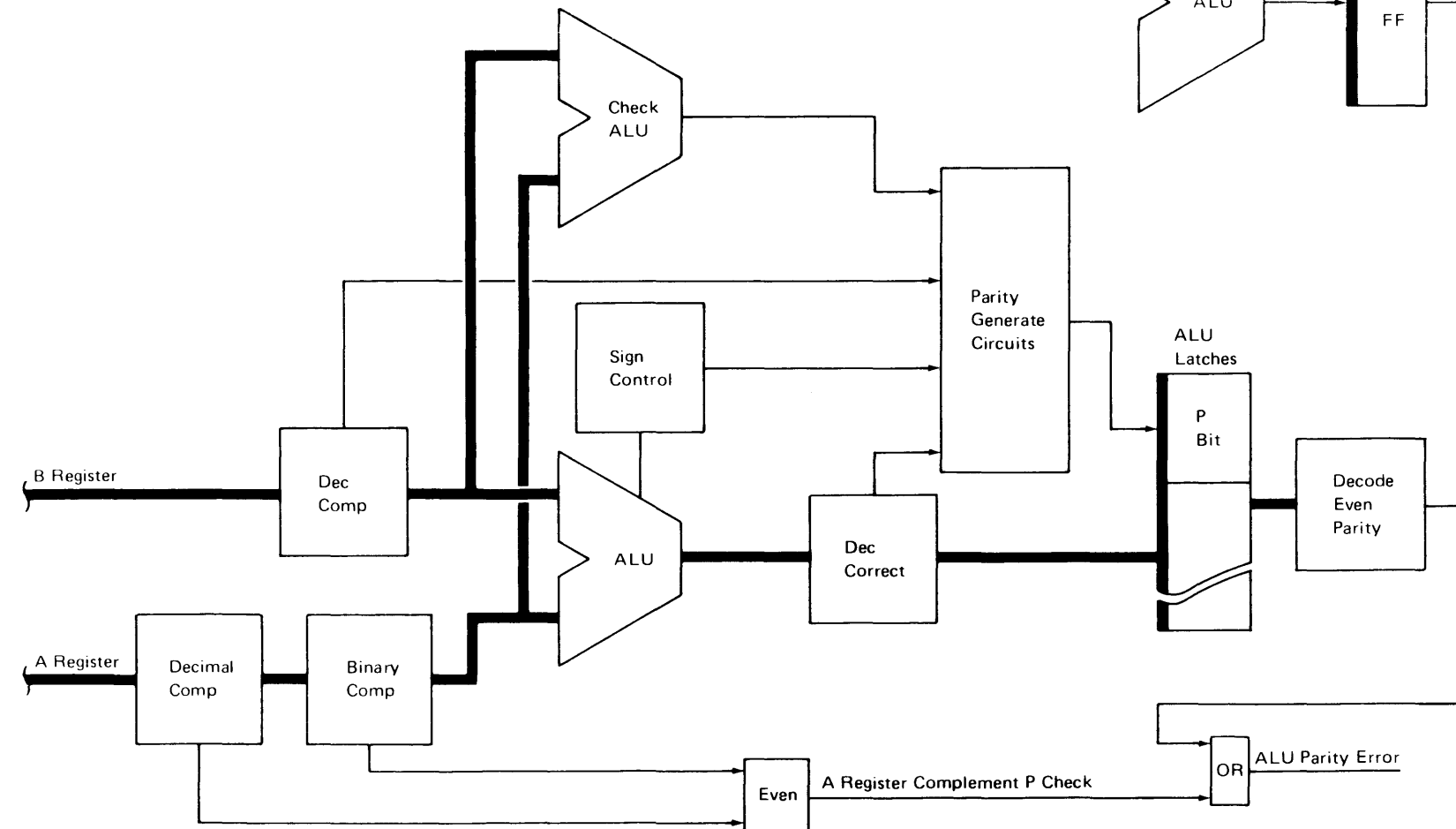
Correct parity is generated for the ALU output. After the data leaves the A and B registers, it can be altered by the decimal and binary complement circuits, the ALU, the decimal correct circuits, and the sign control circuits. The parity changes caused by the ALU and these circuits make it necessary to generate correct parity for the ALU output.

A second ALU, or check ALU, is provided to determine the parity changes which take place within the ALU.

The check ALU does not have a latched output, decimal correction, or sign control, but otherwise performs similarly to the ALU. The output of the check ALU is a group of exclusive ORs that count the number of changes made to the B register complement after it enters the ALU.

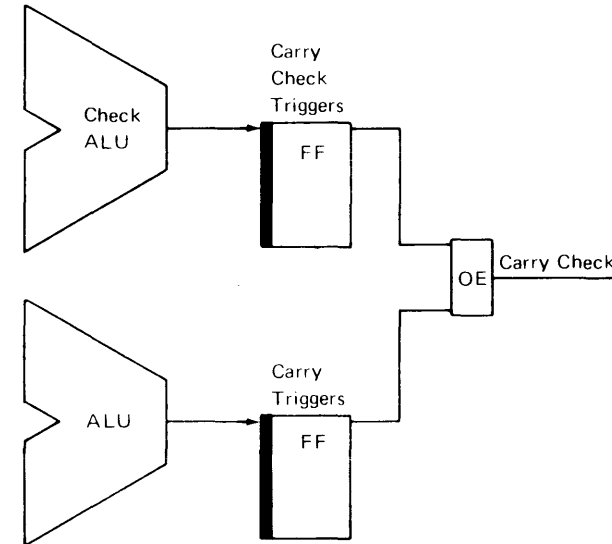
The check ALU output is then added to the changes caused by the B register complement, sign control, and the decimal correct circuits to determine if a P bit is required for the ALU. The output of the ALU is then checked to ensure that it has odd parity.

The A register complement circuits are checked separately. Incorrect parity from either the ALU or the A register complement circuits cause an ALU parity error.

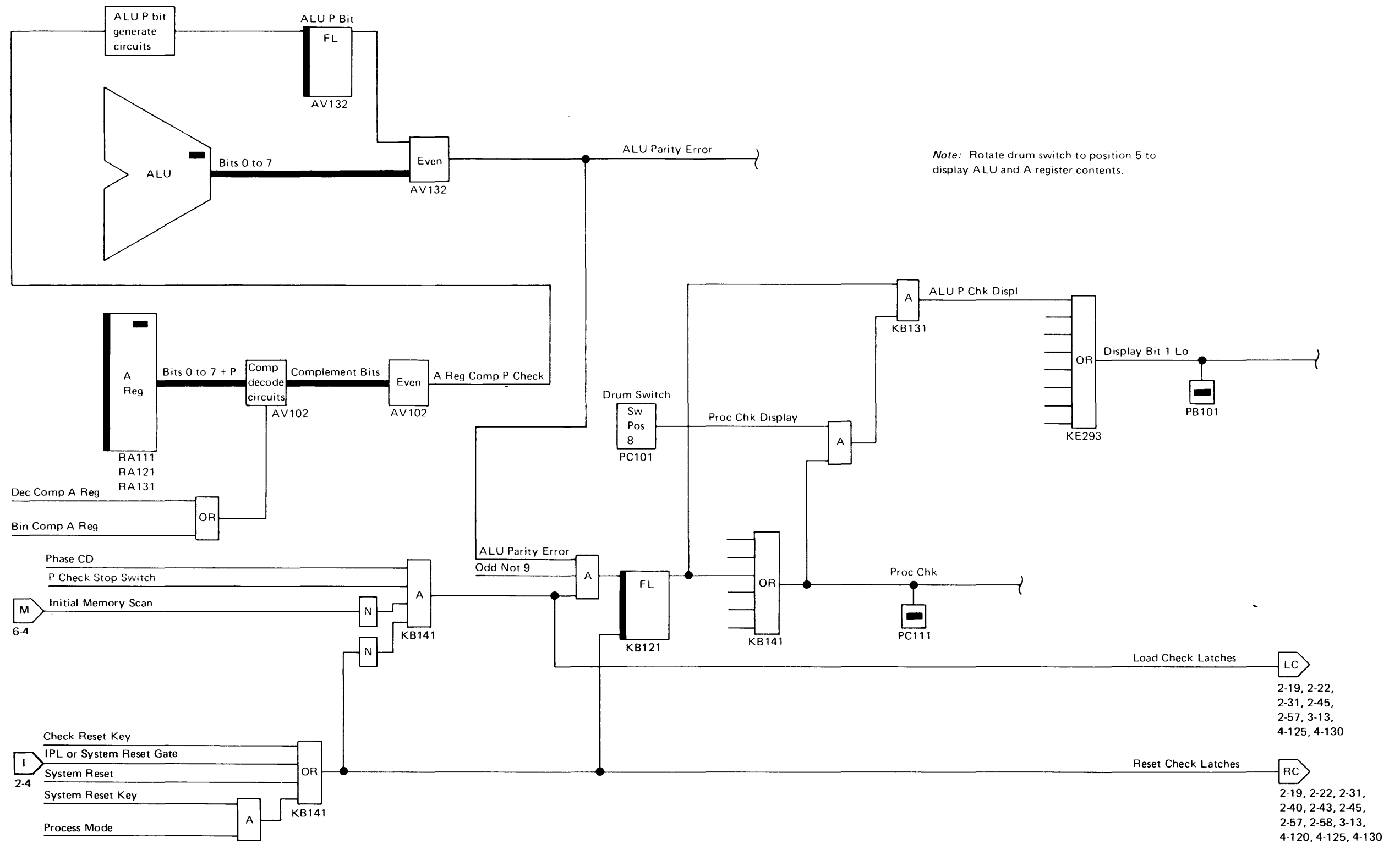


### Carry Check

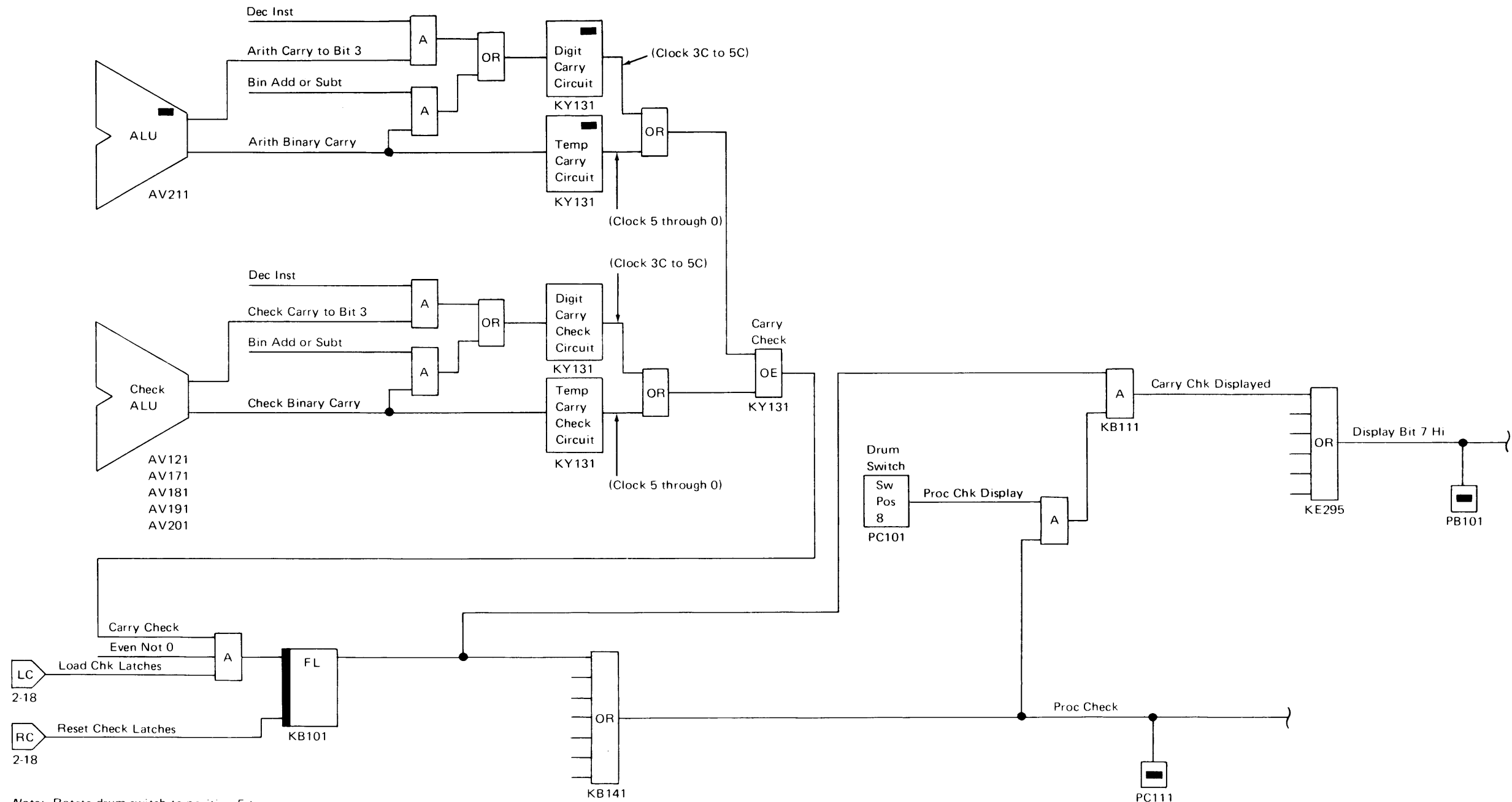
An additional set of carry triggers is used to control carries from the high order bit of the check ALU. The triggers function identically to the digit carry trigger and the temporary carry trigger used for the ALU. The outputs of the two carry control groups are then compared to check for the correct number of carries for the ALU.



ALU Parity Checking



# Carry Check

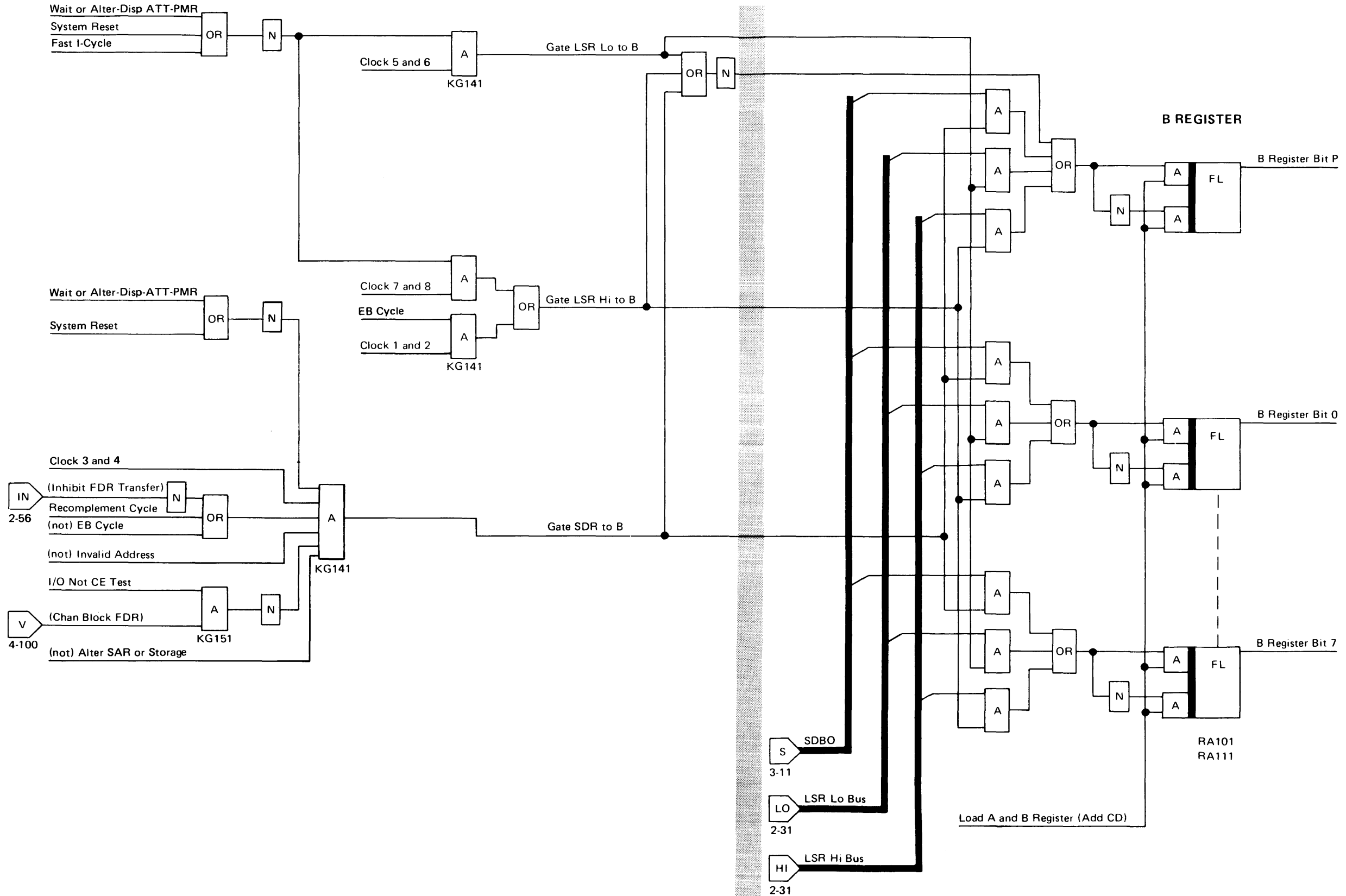


*Note:* Rotate drum switch to position 5 to display ALU output, to position 4 to display ALU controls and carries.

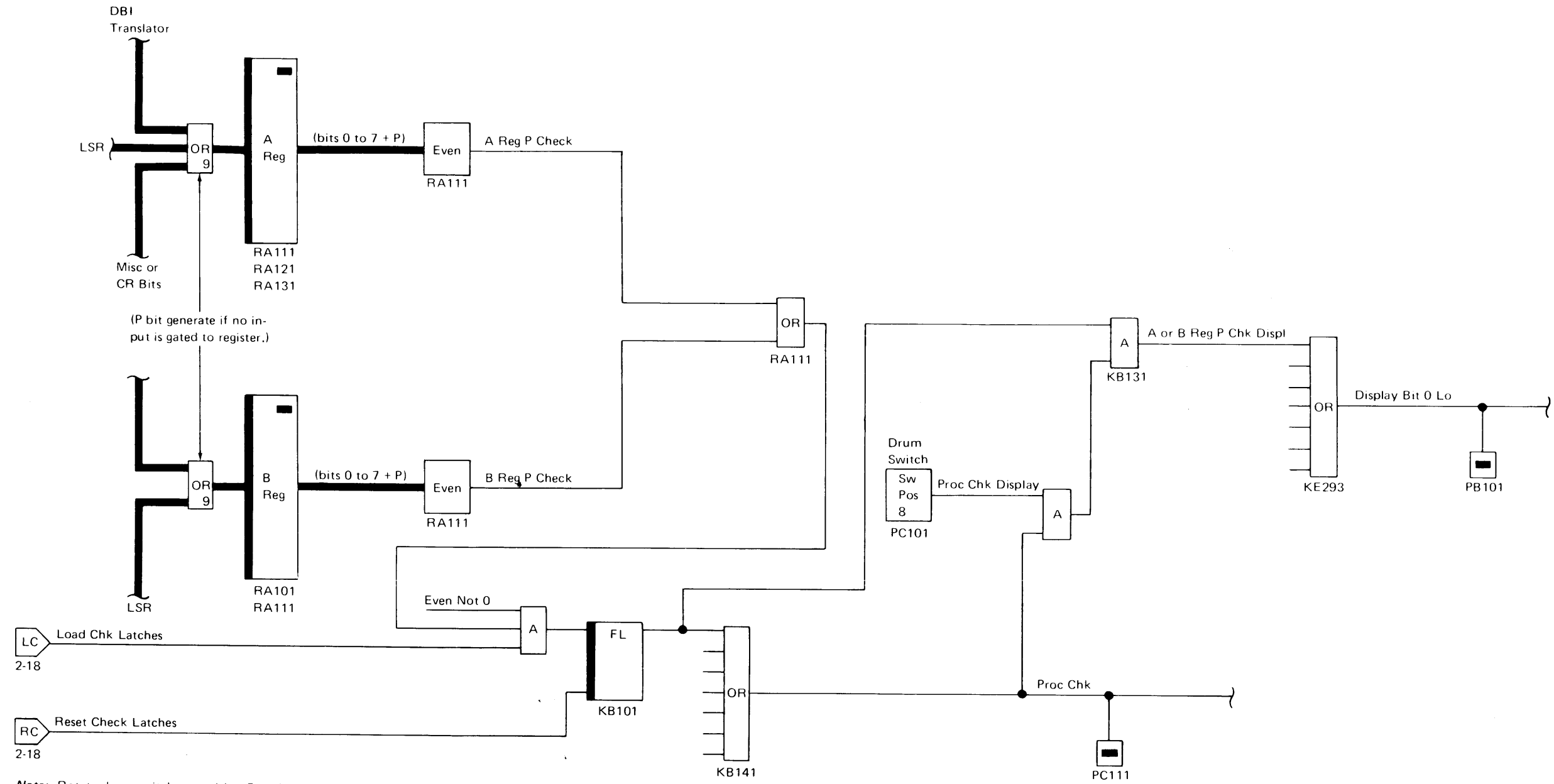


## B REGISTER AND CONTROLS

The B register is a one-byte data buffer. The B register contents are gated to the ALU where the data can be modified by the A register. The data in the FDR (fetch data register) is normally gated into the B register every machine cycle, but can be inhibited if the operation requires. During an I/O cycle, the attachment has control of gating the data to the B register.



A/B REGISTER PARITY CHECKING



**Note:** Rotate drum switch to position 5 to display A register contents, position 4 to display B register contents.



## AUXILIARY ALU

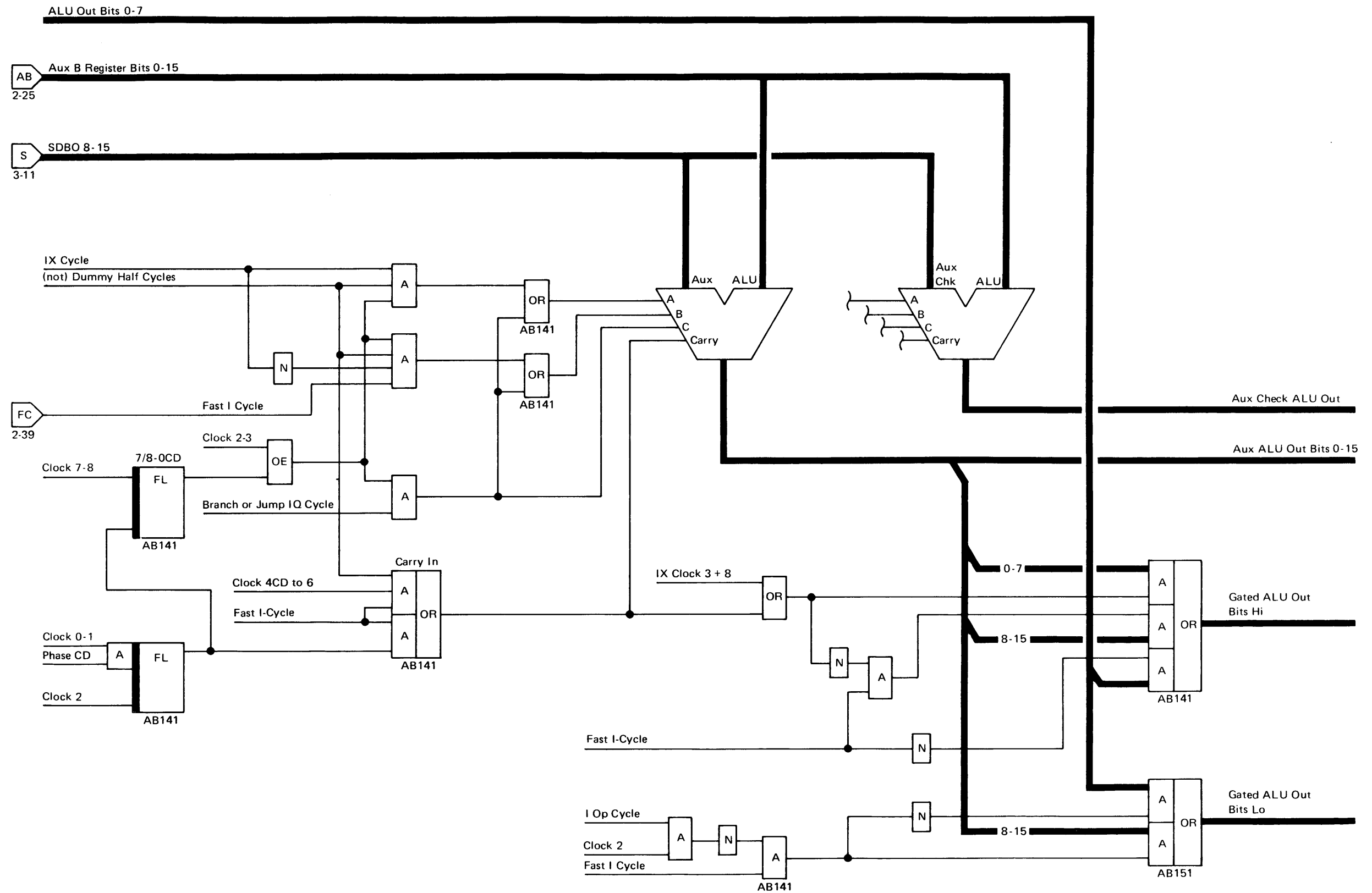
The auxiliary ALU is a 16 bit multiple function unit that receives data from the auxiliary B register and the storage data bus out (SDBO) and performs the following three functions with that data:

- Pass SDBO and auxiliary B register data through
- AND
- ADD

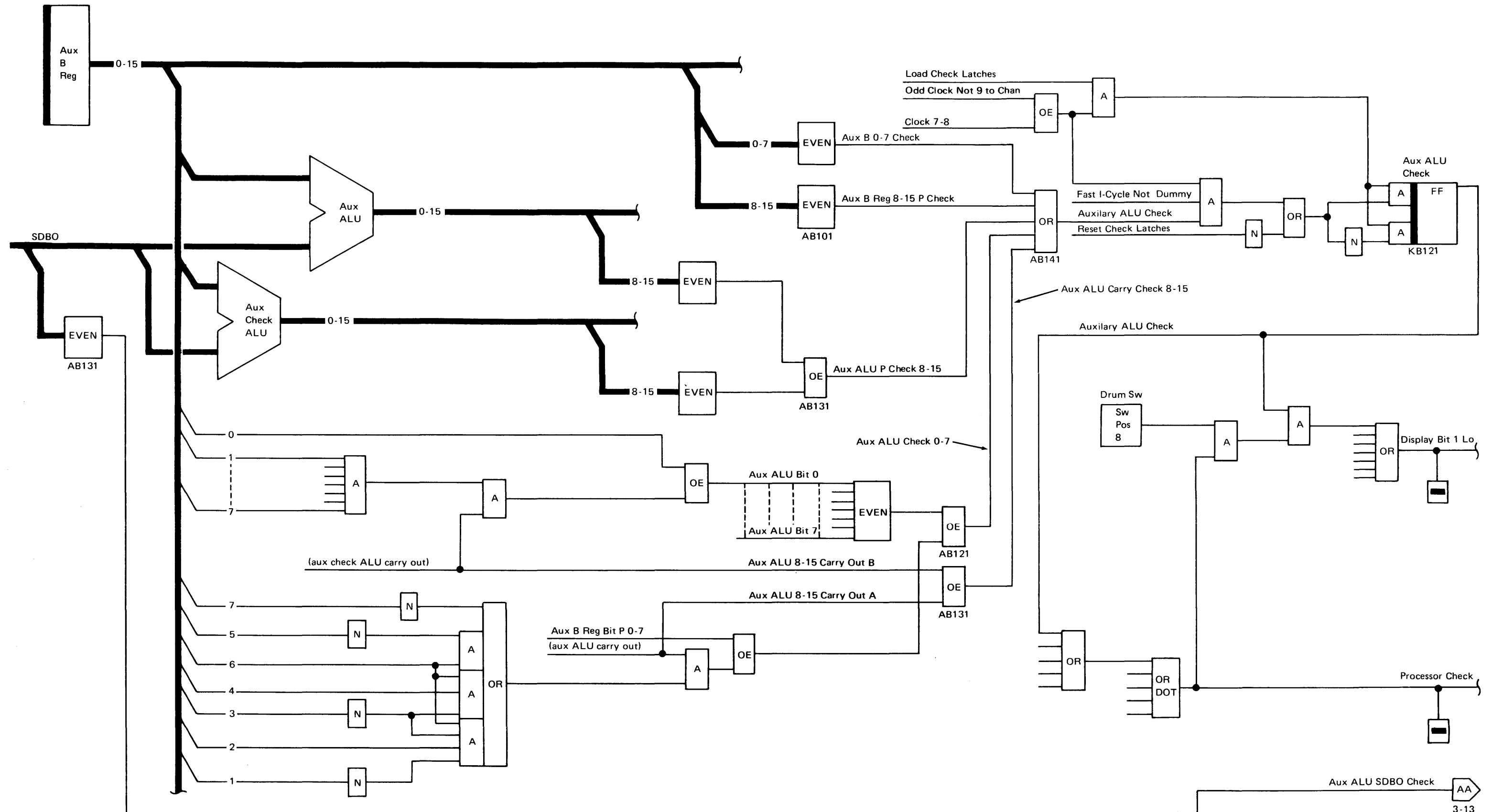
The aux ALU is used to update the two byte IAR and to pass data from storage to the proper register during all fast I cycle operations. It is also used to update the base displacement during I-X cycles.

Data can be passed directly through the aux ALU or combined by the AND and ADD functions. Data passed directly through the aux ALU can be incremented by +1 if the carry in signal is active.

Correct parity is generated for the aux ALU output and the output gated to the local store registers, condition register, Q register, and op register. The use of this output is covered under the individual operation.

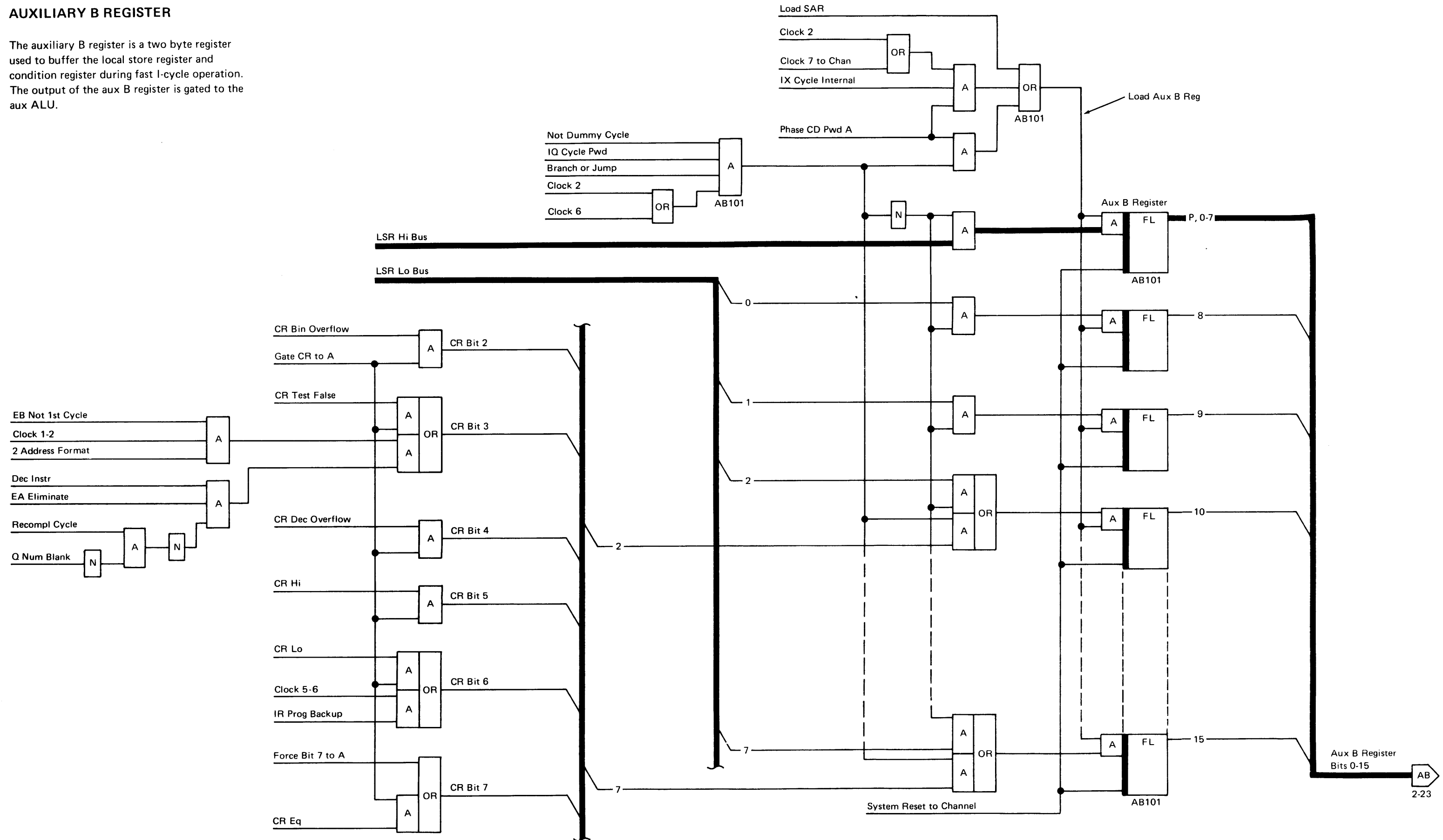


AUXILIARY ALU CARD CHECK



## AUXILIARY B REGISTER

The auxiliary B register is a two byte register used to buffer the local store register and condition register during fast I-cycle operation. The output of the aux B register is gated to the aux ALU.



**CONDITION REGISTER (CR) (Part 1 of 2)**

The condition register is a six-bit register that contains the six conditions the system may test as a result of instruction execution. These six bits are tested as follows:

2	3	4	5	6	7	
x	x	x	x	x	1	Equal condition
x	x	x	x	1	x	Low condition
x	x	x	1	x	x	High condition
x	x	1	x	x	x	Decimal overflow condition
x	1	x	x	x	x	Test false condition
1	x	x	x	x	x	Binary overflow condition

The equal, low, high, and binary overflow conditions reflect the result of executing the last instruction that affected them (one of the following):

- Add zoned decimal
- Zero and add zoned
- Subtract zoned decimal
- Edit
- Compare logical characters
- Add logical characters
- Subtract logical characters
- Add to register
- Compare logical immediate

The decimal overflow or the test false condition can be reset by:

- Branch on condition.
- Jump on condition.
- Load register (PSR) instruction.
- System reset.

System reset initializes the condition register to:

- Equal condition.
- Not overflow condition.
- Not test false condition.

The condition register may be loaded from the ALU output, but normally the bits are set individually in the latches by the CPU logic as a result of instruction execution. When the CR contents are needed for program testing, its output is fed to the A register and into the ALU.

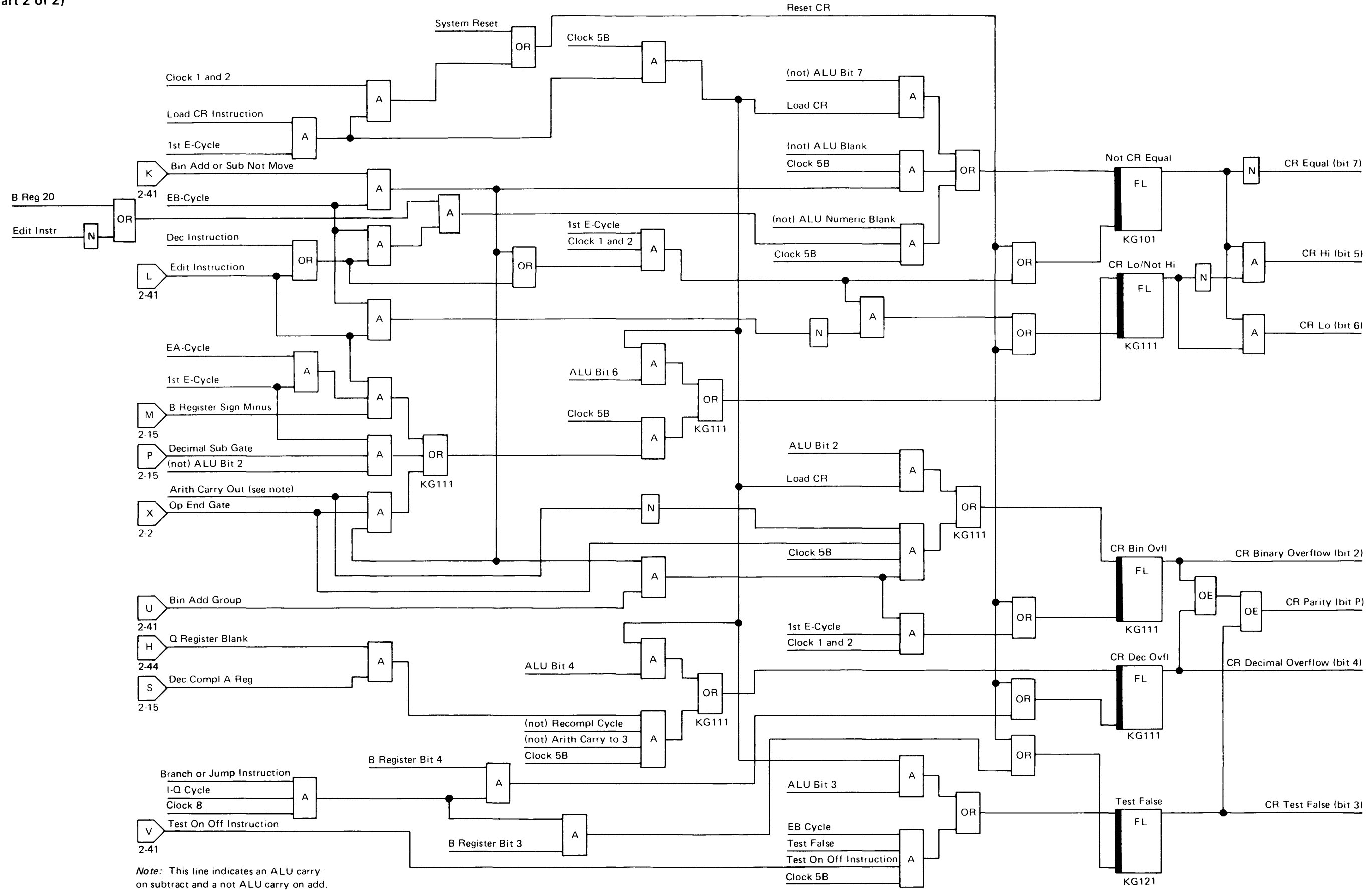
The lower six bits of the PSR (program status register) contain the image of the condition register for the specific program level. PSR is used to save and to initialize condition register settings of the individual program levels.

The conditions of the I/O attachment logic are stored in registers in the attachment and do not affect the state of the condition register.

**Condition Register Settings**

Operation							
Condition	Bit	Decimal Arithmetic	Logical Compare Sub Logical	Add Logical Add to Register	Edit	Test Bits	Branch or Jump on Condition
Equal	7	Result is Zero	First Operand is Equal to Second	Result is Zero	Source is Zero		
Low	6	Negative	First Operand is Lower than Second	No Carry and Non-Zero Result	Negative		
High	5	Positive	First Operand is Higher than Second	Carry and Non-Zero Result	Positive		
Decimal Overflow	4	Overflow					Overflow Reset if Tested
Test False	3					Test False	Test False Reset if Tested
Binary Overflow	2			Overflow			

CONDITION REGISTER (Part 2 of 2)



## PROGRAM MODE REGISTERS (PMR) (Part 1 of 2)

The nine program mode registers (PMR) control the Model 15 CPU functions and are the interface between the programmer and the CPU hardware.

Each of the nine PMRs is uniquely associated with one of the eight interrupt levels plus the program level. Since each machine level has its own PMR, the interrupt routines and the main program can independently use the enhanced functions of the CPU.

As interrupts occur and take control from the main program or lower level interrupts, the PMR for that interrupt is selected. When interrupts are reset, system control falls back to its previous level. The PMR that was in control previously is selected and processing resumes as if the interrupt never occurred.

When a program changes the contents of the current PMR, the new contents are effective at NSI (next sequential instruction).

The contents of the PMRs are indeterminate after power up. The controlling set of polarity holds (active PMR) are, however, forced to privileged mode active and all others are inactive. The system is then referred to as being in Model 10 mode. The CPU remains in this mode until the first load PMR instruction is executed.

The program level PMR must first be loaded, then the interrupt level PMRs are loaded. This must be done before any interrupts occur to prevent possible checks caused by using the indeterminate contents of the PMRs after power up.

The ATT control bits (1, 2, and 3) and the storage protect bit (6) must not be turned on until the contents of the ATT/SPT have been set by the program. To do so would result in checks because the ATT/SPT contents are also indeterminate after power up.

A description of each PMR bit is presented below (see **A** on next page).

### Byte 1

Bit 0 I/O 18th bit

This bit is used in conjunction with LIO and SNS commands to load and sense the 18th address bit associated with each I/O device LSR.

If this bit is on and an LIO is issued to a device LSR, the 18th bit in the selected LSR is turned on. If this bit is off and an LIO is issued to a device LSR, the 18th bit in the selected LSR is turned off. When a SNS command is issued to a device LSR, the 18th bit in the PMR (bit 0) is set to the state of the 18th bit in the selected device LSR. The PMR may be stored in memory and bit 0 inspected to determine the contents of the device LSR. Since a two instruction sequence (LCP, LIO or SNS, SCP) is required to set or inspect the 18th bit in the I/O LSR, an interrupt may occur between these instructions. Each interrupt level, however, has its own PMR and one level will not destroy the results of another. It is not necessary to set on the mask interrupt bit while executing this two-instruction sequence.

Bit 1 Address translation EB cycles

When this bit is on, the output of the ATT is used to develop a 18-bit real address during EB cycles only. EB cycles are used to fetch, modify, and store the contents of the B field (operand 1). If the B field is above 64K of storage, this bit must be on and the ATT must be appropriately loaded. If the B field is below 64K of storage, the ATT may be used but is not required. This bit must not be turned on until the ATT contents are set after power on. This bit should be off while the ATT contents are changed.

Bit 2 Address translation EA cycles

When this bit is on, the output of the ATT is used to develop a 18-bit real address during EA cycles only. EA cycles are used to fetch the contents of the A field (operand 2). If the A field is above 64K, this bit must be on and the ATT must be appropriately loaded. If the A field is below 64K of storage, the ATT may be used but is not required. This bit must not be set on until the ATT contents are set after power on.

Bit 3 Address translation I cycles

When this bit is on, the output of the ATT is used to develop a 18-bit read address during I cycles. I cycles are used to fetch the instruction to be executed from main storage. If the instruction to be executed is above 64K of storage, this bit must be on and the ATT must be appropriately loaded. If the instruction to be executed is below 64K of storage, the ATT may be used but is not required. This bit must not be set on until the ATT contents are set after power on.

Bit 4 Privileged mode

When this bit is on, the system is in privileged mode and all instructions can be used. If the bit is off, the following instructions cannot be used:

LIO  
SNS  
SIO  
TIO  
APL  
HPL  
CCP except when Q equals 10  
LCP  
SCP  
L, ST, A except when Q equals 01, 02, 04, 08, 10, 20, or 40

If an attempt is made to execute these instructions, a program check interrupt results (a processor check if interrupt level 7 is not enabled). This bit has no significance for interrupt level 0. The system is automatically in privileged mode when the CPU is in interrupt level 0.

Bit 5 I/O 17th bit

This bit is used in conjunction with LIO and SNS commands to load and sense the 17th address bit associated with each I/O device LSR.

If this bit is on and a LIO is issued to a device LSR, the 17th bit in the selected LSR will be set on. If this bit is off and a LIO is issued to a device LSR, the 17th bit in the LSR will be set off. When a SNS command is issued to a device LSR, the PMR 17th bit (bit 5) will be set to the state of the 17th bit in the selected LSR. The PMR may be stored in memory and bit 5 inspected to determine the contents of the device LSR. Since a two-instruction sequence (LCP, LIO or SNS, SCP) is required to set or inspect the 17th bit in the I/O LSR, an interrupt may occur between these instructions. Each interrupt level, however, has its own PMR and one level will not destroy the results of another. It is not necessary to set on the mask interrupt bit while executing this two instruction sequence.

Bit 6 Storage protect

When this bit is on, the storage protect keys in the SPT (bits 0 and 1) are inspected for each memory cycle except I/O cycles. A violation causes a program check interrupt or, if interrupt level 7 is not enabled, a processor check occurs. When this bit is off, the protect keys are ignored and all locations are available to the program. This bit must not be set until the SPT has been properly loaded. Furthermore, to prevent an error indication, this bit must be off whenever the SPT contents are changed.

Bit 7 Mask interrupts

When this bit is on, all interrupt requests (except program check) remain pending and the CPU remains in its present interrupt level.

This bit may be set in the program level or any interrupt level. Once set, the CPU cannot change levels until the bit is set off. This bit must be set off before any interrupt is reset. Failure to do so causes the CPU to remain in that interrupt level. Caution should be exercised when using this bit because it could cause an overrun in those devices whose interrupts must be serviced within a certain period of time.

Bits 1 through 6 exist for each machine level, but bit 7 is common to all levels. Once bit 7 is set, the machine cannot pass to another level. Independent control of the mask bit is not required.

The mask bit is always effective immediately to prevent or allow interrupts that could occur before the next instruction.

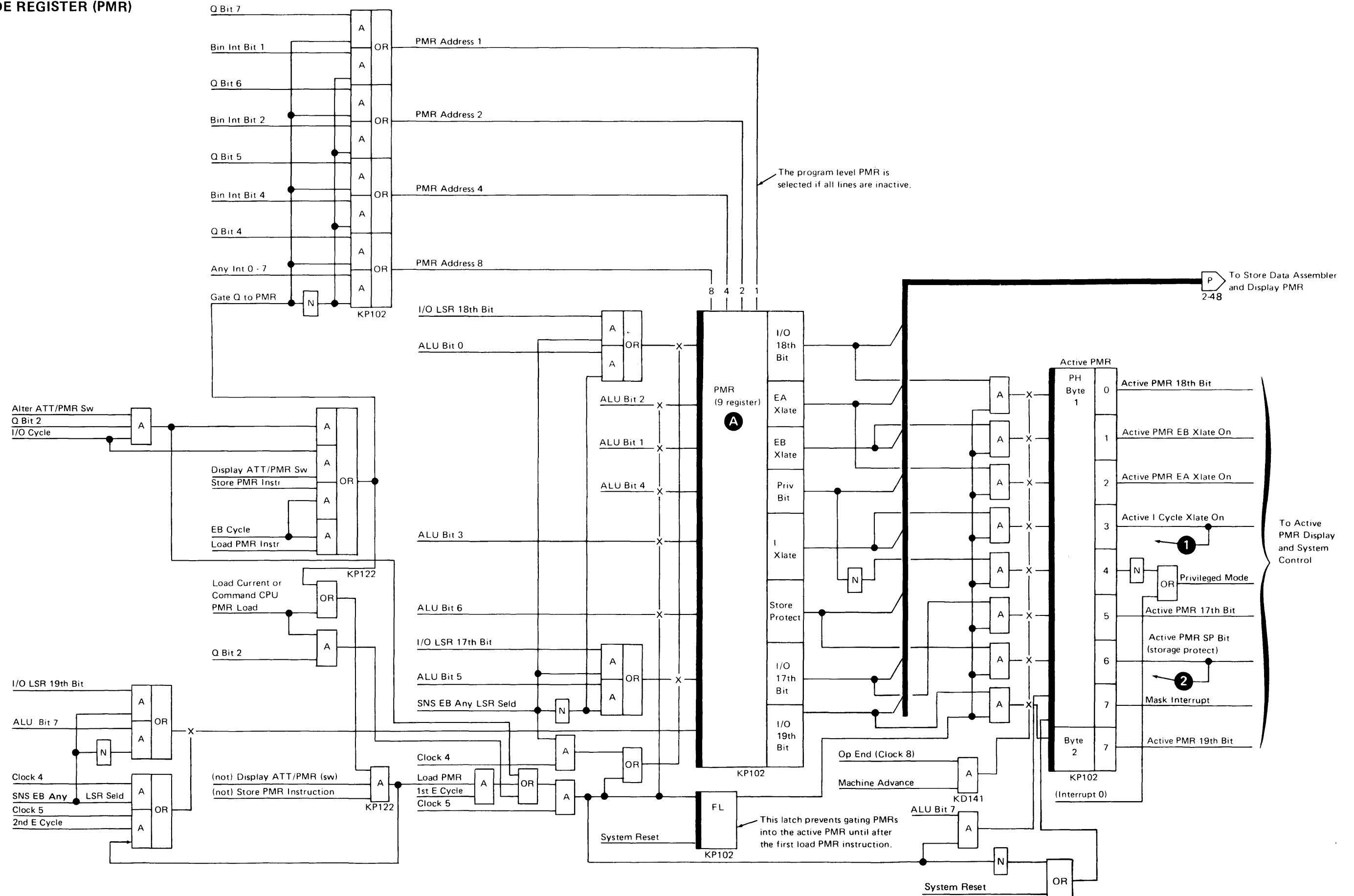
### Byte 2

Bit 7 I/O 19th bit

This bit is used in conjunction with LIO and SNS commands to load and sense the 19th address bit associated with each I/O device LSR.

If this bit is on and an LIO is issued to a device LSR, the 19th bit in the selected LSR will be set on. If this bit is off and an LIO is issued to a device LSR, the 19th bit in the LSR will be set off. When an SNS command is issued to a device LSR, the PMR 19th bit (bit 7) will be set to the state of the 19th bit in the selected LSR. The PMR may be stored in memory and bit 7 inspected to determine the contents of the device LSR. Since a two-instruction sequence (LCP, LIO or SNS, SCP) is required to set or inspect the 19th bit in the I/O LSR, an interrupt may occur between these instructions. Each interrupt level, however, has its own PMR and one level will not destroy the results of another. It is not necessary to set on the mask interrupt bit while executing this two-instruction sequence.

**PROGRAM MODE REGISTER (PMR)  
(Part 2 of 2)**



## LOCAL STORAGE REGISTERS (LSR)

The LSRs consist of two addressable HDB (high density buffer) units that maintain:

- Sequential instruction addresses.
- Current operand addresses during instruction execution.
- I/O data area addresses.

In addition, LSRs contain:

- Index registers for modification of operand addresses.
- Interim storage for data, length count, and program condition status referred to as *scratch pad* type of storage.

Page 2-33 list the LSRs for the system and available features. To read out data, only the select lines are needed. To write, 'data', 'write hi or write lo', and the 'LSR select' lines are needed. Following are summaries of the functions of the base system LSRs:

### IAR (Instruction Address Register)

The IAR contains the location of the next sequential instruction byte that is read out of storage. At the beginning of each I-cycle, the address in the IAR is gated into the SAR so that the address can be decoded. During each I-cycle, the contents of the IAR is incremented by 1 in preparation for the next I-cycle.

### AAR (A-Address Register)

The AAR contains the storage address of the next byte that is addressed in the A-field. During I-cycles, the A-field address is taken from the instruction and loaded into the AAR. At the beginning of each A-cycle, the address in the AAR is gated into the SAR. During each A-cycle, the contents of the AAR is decremented by 1 in preparation for the next A-cycle.

### BAR (B-Address Register)

The BAR contains the storage address of the next byte that is addressed in the B-field. During I-cycles, the B-field address is taken from the instruction and loaded into the BAR. At the

beginning of each B-cycle, the address in the BAR is gated into the SAR. During each B-cycle, the contents of the BAR is normally decremented by 1 in preparation for the next B-cycle.

### Index Register 1 and Index Register 2

These registers can each store a two-byte address to be used in indexing operations. During an indexing operation, the CPU automatically adds the single byte displacement from the instruction to the contents of XR1 or XR2 to obtain the actual B or A-field address. The contents of the index registers are not changed as a result of the addition. The resulting address is placed in the BAR or the AAR.

### ARR (Address Recall Register)

On a branch instruction, the ARR contains the 'branch to' address. On a decimal instruction, the ARR retains the starting address of the B-field in the event recomplementing is required. On an insert and test characters instruction, the ARR contains the address of the first significant digit encountered.

### LCR (Length Count Register)

The LCR is a one byte register that contains the length count of the B and A-fields. It is decremented by 1 on each B-cycle except the first one.

### DRR (Data Recall Register)

The DRR is a one byte register that provides temporary storage for the data character read out of storage during each A-cycle. It is also used to store the Q code of single address instructions.

### PSR (Program Status Register)

The high byte of the PSR is used as the LCRR (length count recall register). The LCRR is used only during a recomplement operation. It stores the length of the data fields and is decremented on each recomplement cycle except the first. The low byte of the PSR is used as the condition recall register (CRR). The CRR is an image of the condition register and is used to store the contents of the condition register.

### MRDAR (MFCU Read Data Address Register)\*

The MRDAR contains the storage position that is to be addressed next while reading data from a card into the card read area in core storage.

### MPCAR (MFCU Punch Data Address Register)\*

The MPCAR contains the storage position in the MFCU punch data area that is to be addressed next during a punch operation.

### MPTAR (MFCU Print Data Address Register)\*

The MPTAR contains the storage position in the MFCU print data area that is to be addressed next during an MFCU print operation.

### LPDAR (Line Printer Data Address Register)

The LPDAR contains the storage position in the line printer data area that is to be addressed next during a print operation.

### LPIAR (Line Printer Image Address Register)

The LPIAR contains the storage position in the chain image area that is to be addressed next during a print operation of the line printer.

### Interrupt Level 0 - 7 Instruction Address Registers—

These registers contain the address of the next sequential instruction byte to be read out of storage during an interrupt level operation.

### Interrupt Level 0 - 7 Address Recall Registers—

These registers have the same function as the P-ARR, but are active only during that interrupt level operation.

The registers are paired to give an LSR Hi (the high-order byte) and an LSR Lo (the low-order byte). Only one byte can be written into an LSR at a time. To write into an LSR, it is necessary to activate the select line for a pair of registers and the 'LSR write Hi' or 'LSR write Lo'. All 18 bits for the LSR selected are available at the output of the LSR array. These bits can be gated to the SAR (18 bits), the B register (9 bits), and the A register (9 bits) for modification of the addresses in these registers.

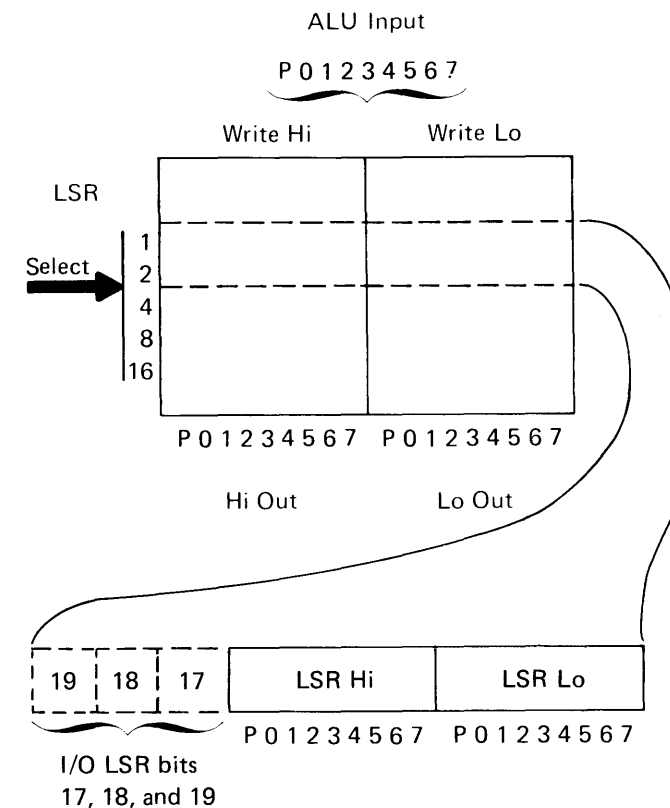
\* These registers are used for the card I/O device that is installed. The base system has one of the following devices installed: 5424, 2560 or 1442.

LSR select and write lines are normally controlled by the CPU. However, during an I/O cycle, the I/O attachment can control the select lines for the LSR assigned to it.

A description of the CPU LSRs and LSR parity checking may be found on page 2-31.

A description of the I/O LSRs may be found on page 2-32.

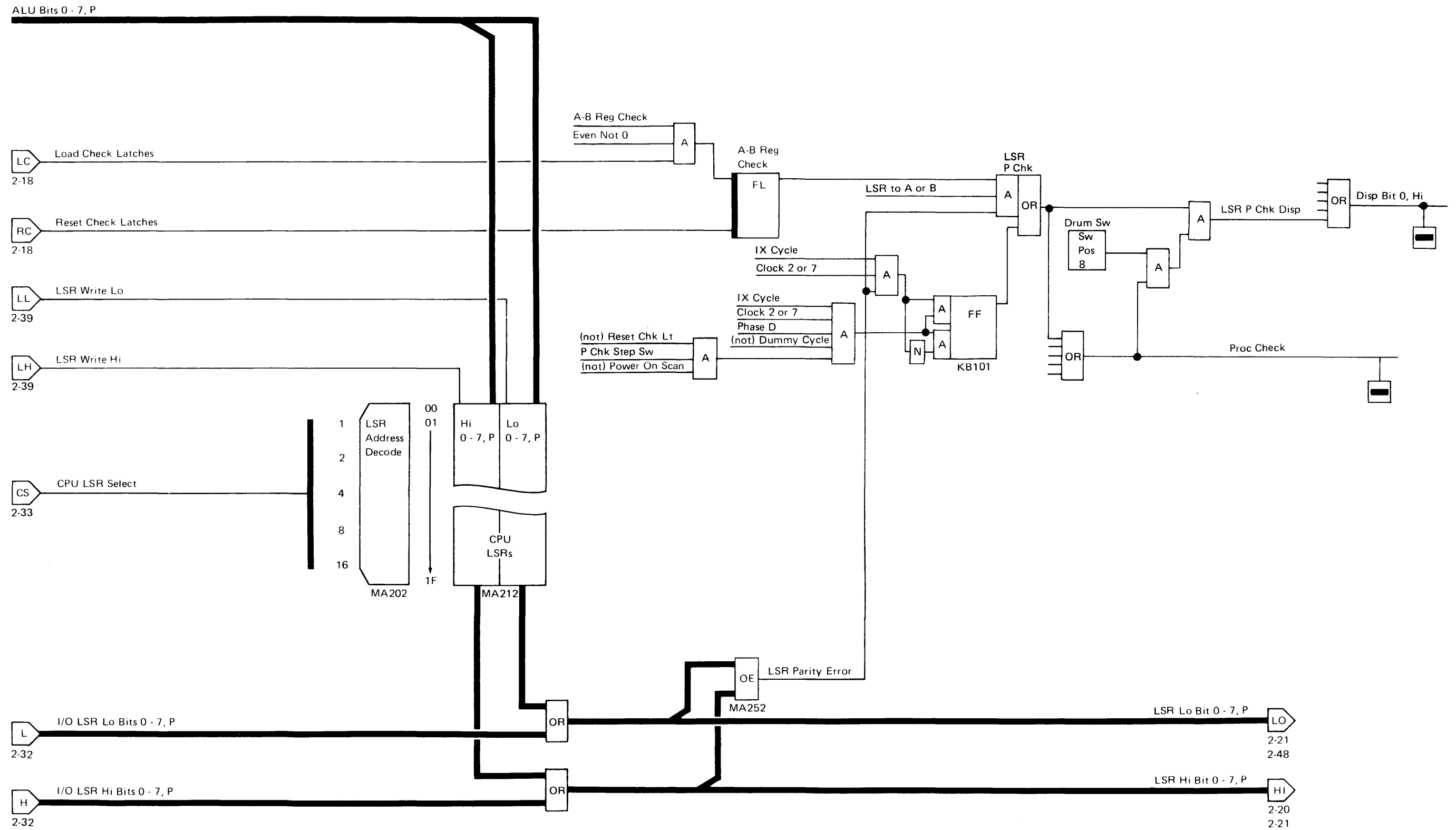
A description of the I/O LSR select checking may be found on page 2-40.



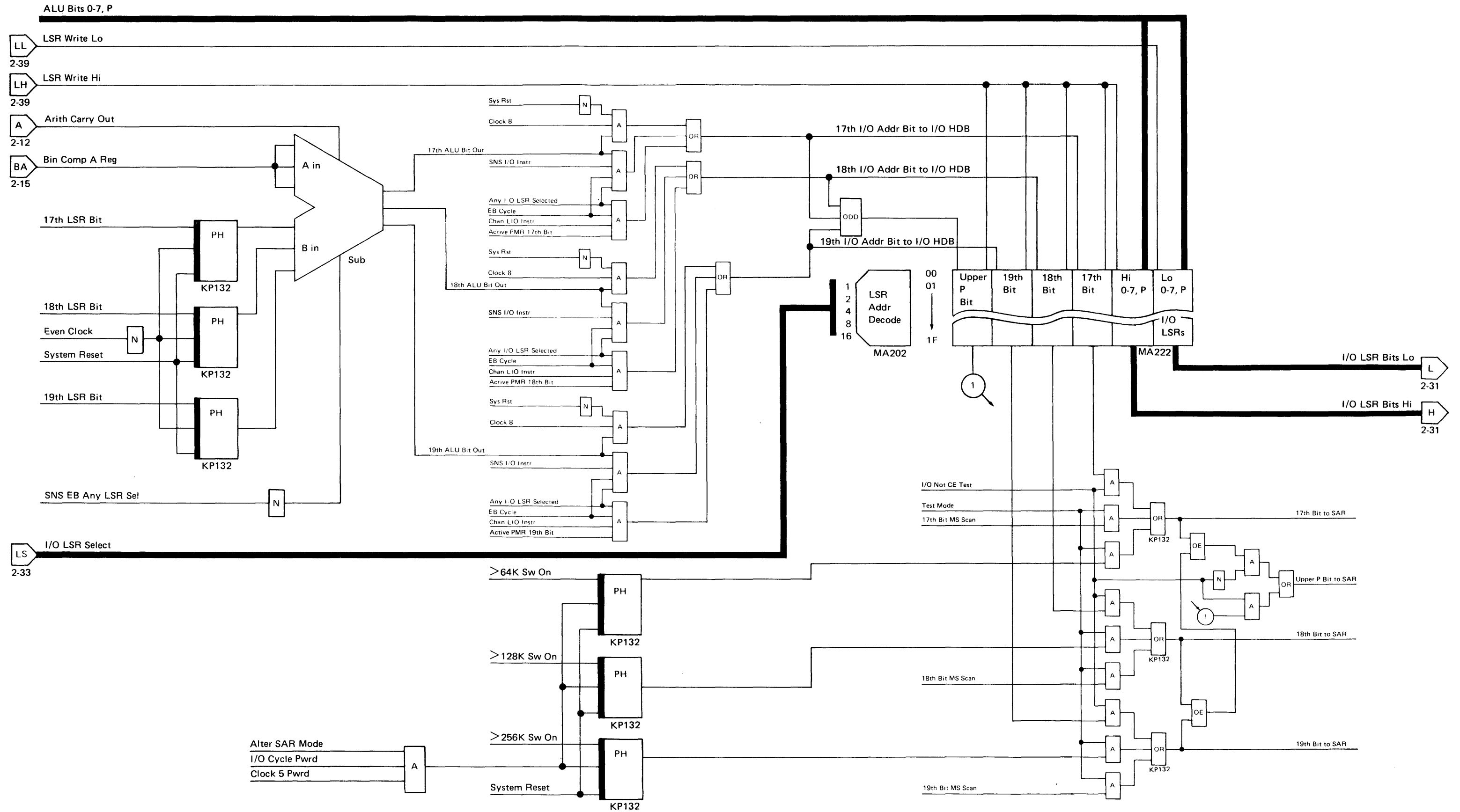
I/O LSRs have a 17th, 18th and 19th bit position. These bit positions are turned on by an ALU overflow during an I/O LSR update, or during an LIO with PMR bit 5 (17th bit) of byte 1 bit 0 (18th bit) of byte 1, or bit 7 (19th bit) of byte 2 on. The circuit is shown on page 2-32.



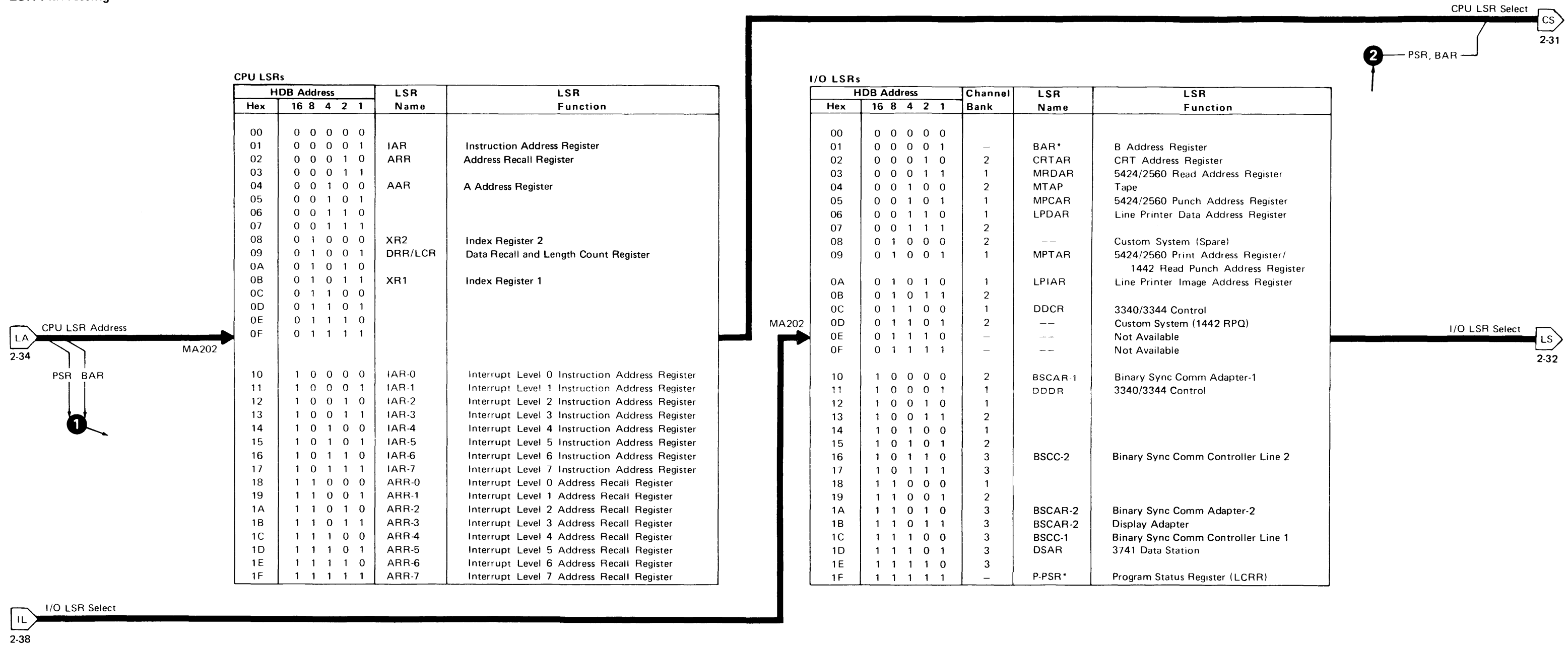
Processing Unit LSRs



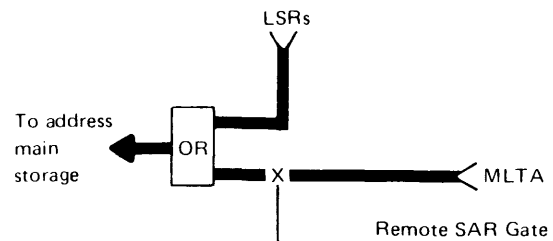
# I/O LSRs



# LSR Addressing

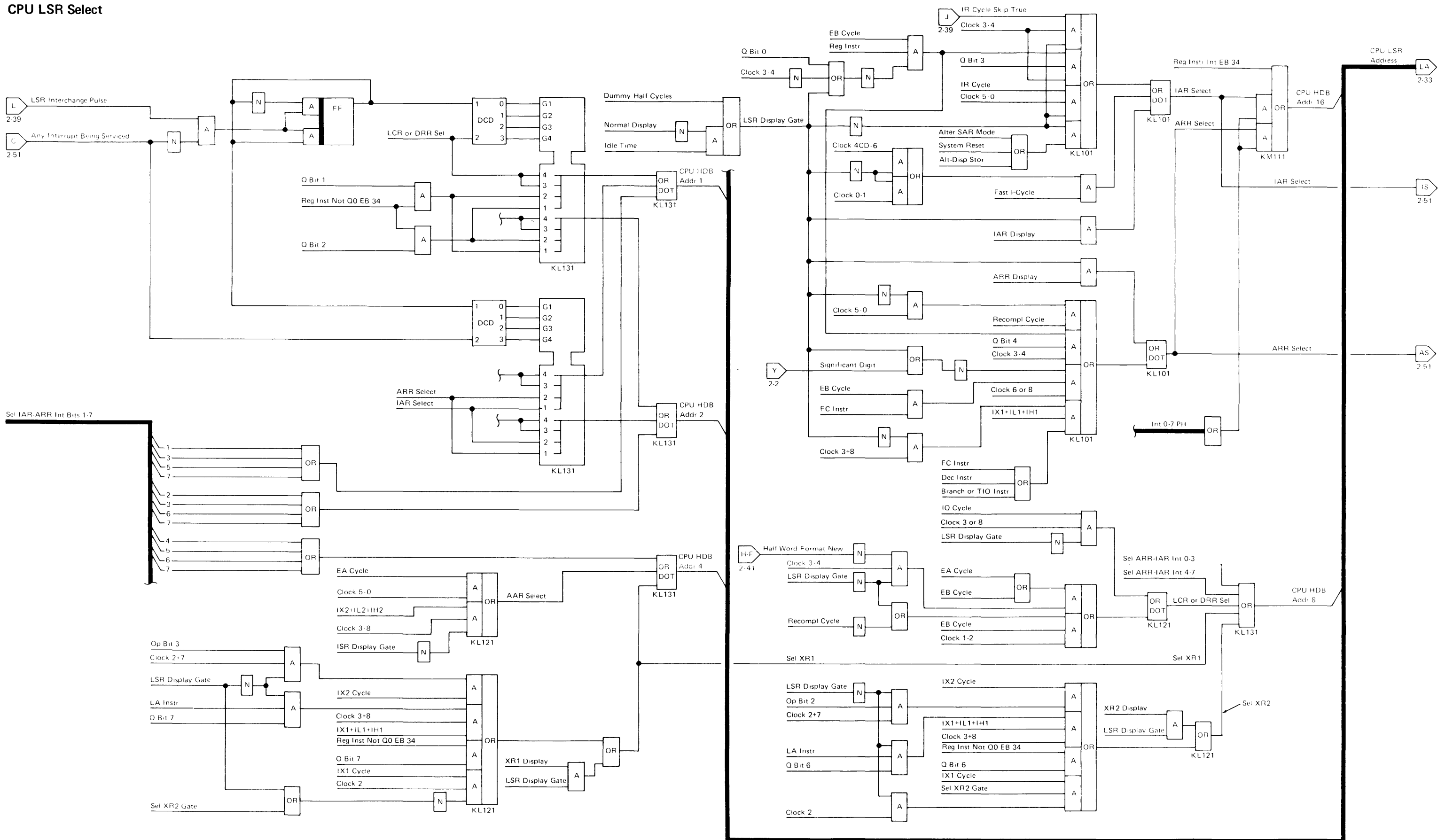


Note: The MLTA addresses main storage by using the remote SAR lines and activating 'remote SAR gate'.

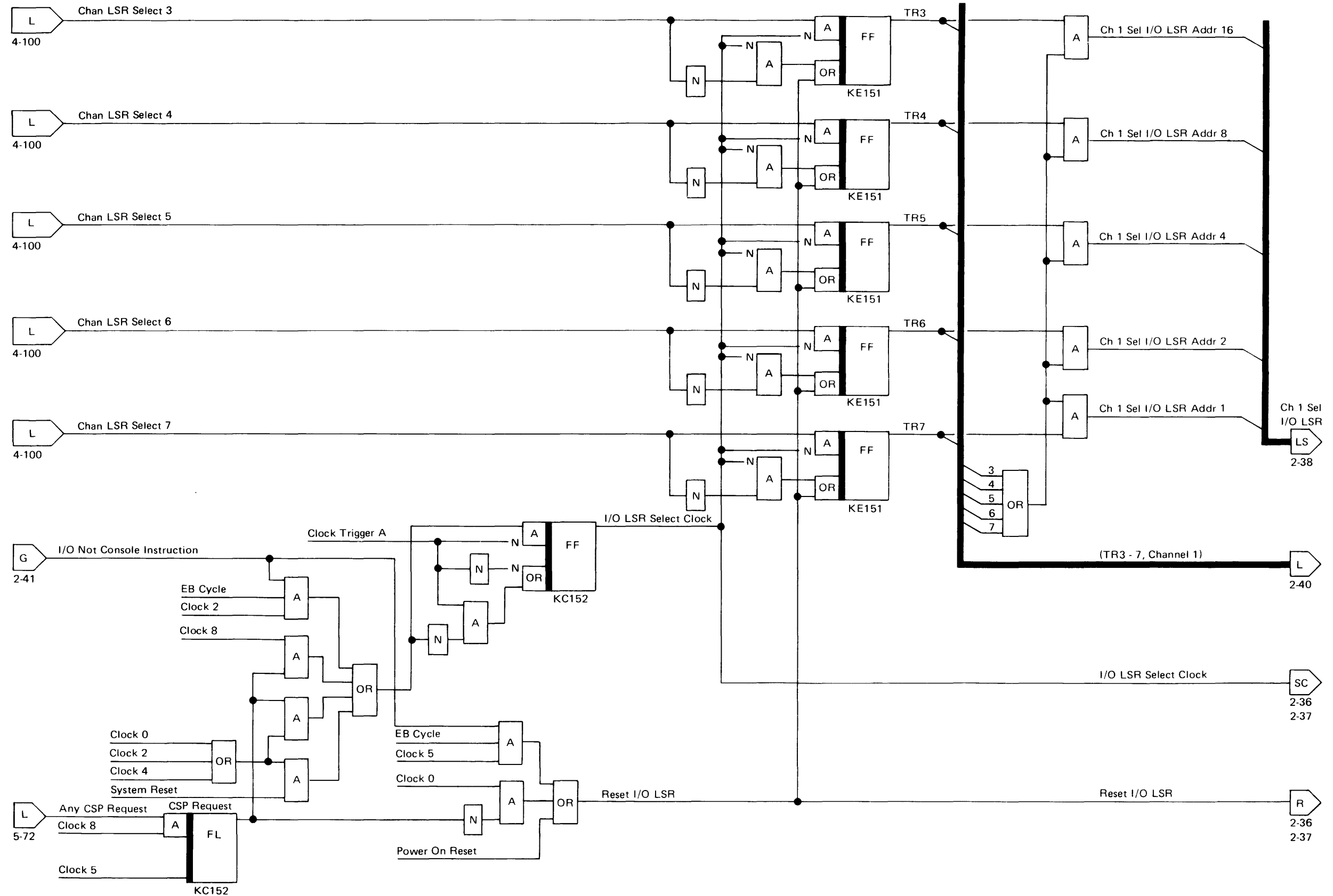


\*Only one LSR can be selected in an HDB module at a time. Most CPU operations require selecting only one LSR. However, the following operations require selecting two LSRs at the same time: A system reset operation selects the P-PSR and IAR (refer to page 6-2); A branch, TIO, or decimal instruction selects the BAR and the ARR during the IH cycles. Refer to page 5-8. Therefore, the BAR and the P-PSR are located in the I/O LSR module but function as CPU LSRs.

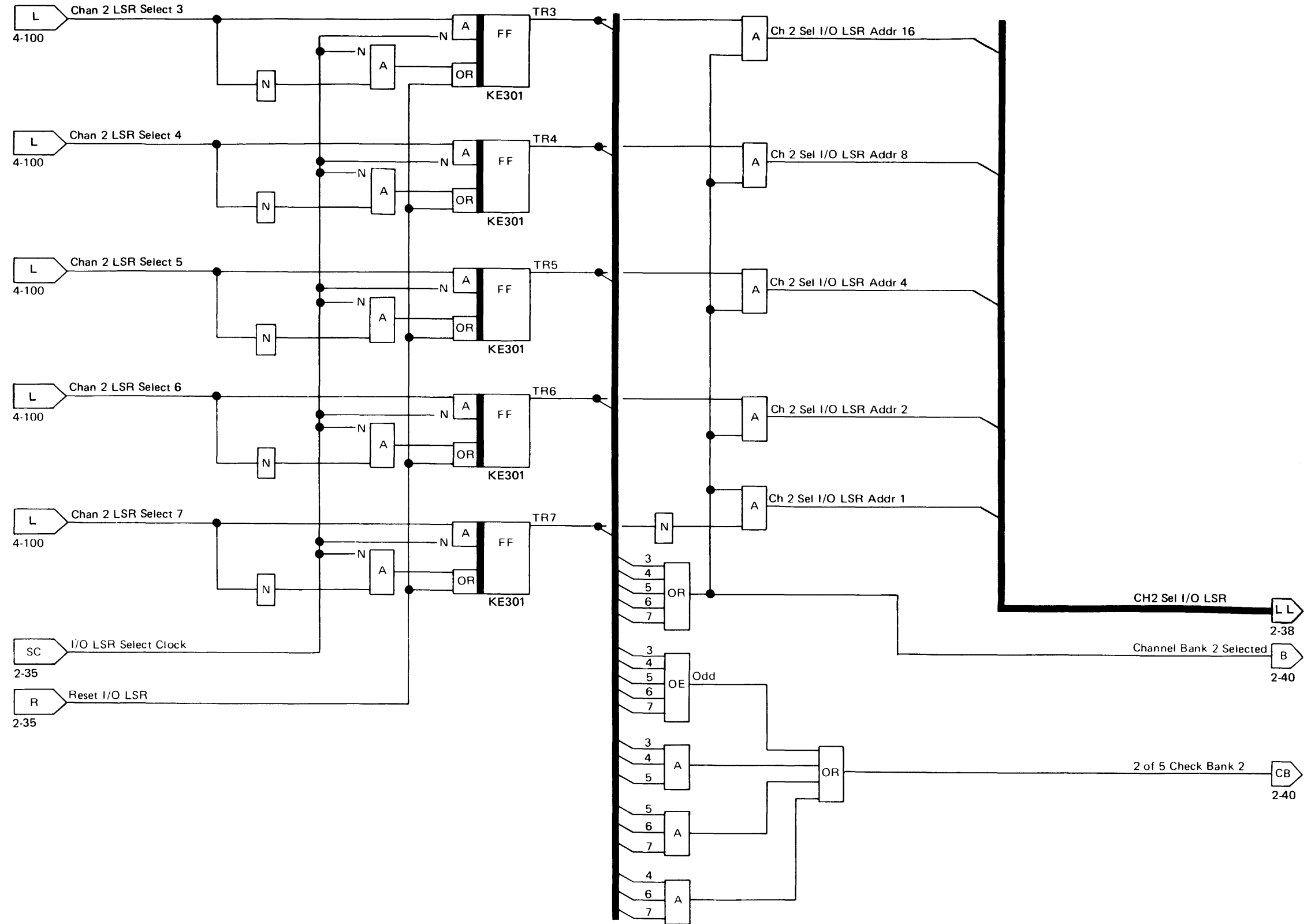
### CPU LSR Select



LSR Select (I/O Channel Bank 1)

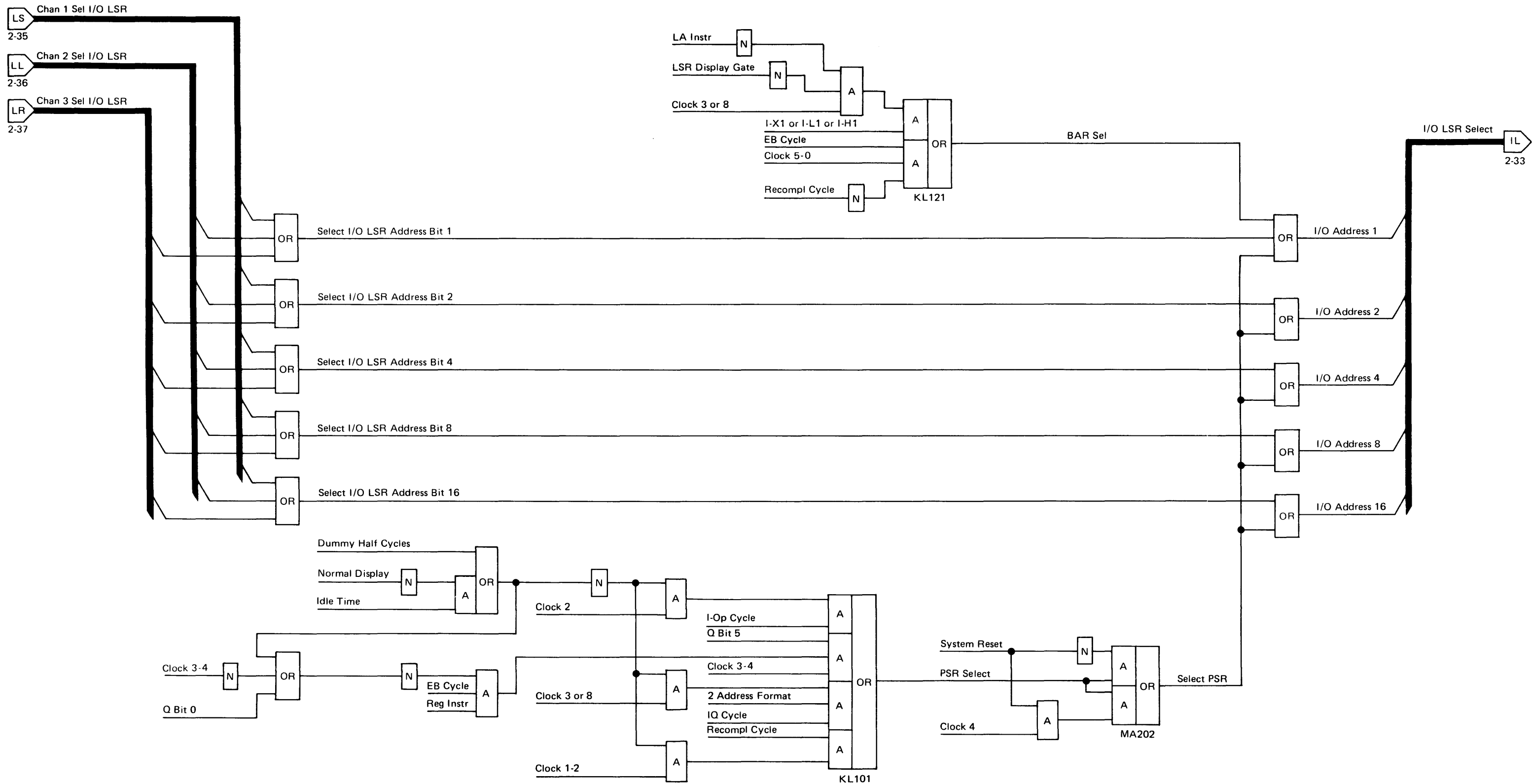


LSR Select (I/O Channel Bank 2)



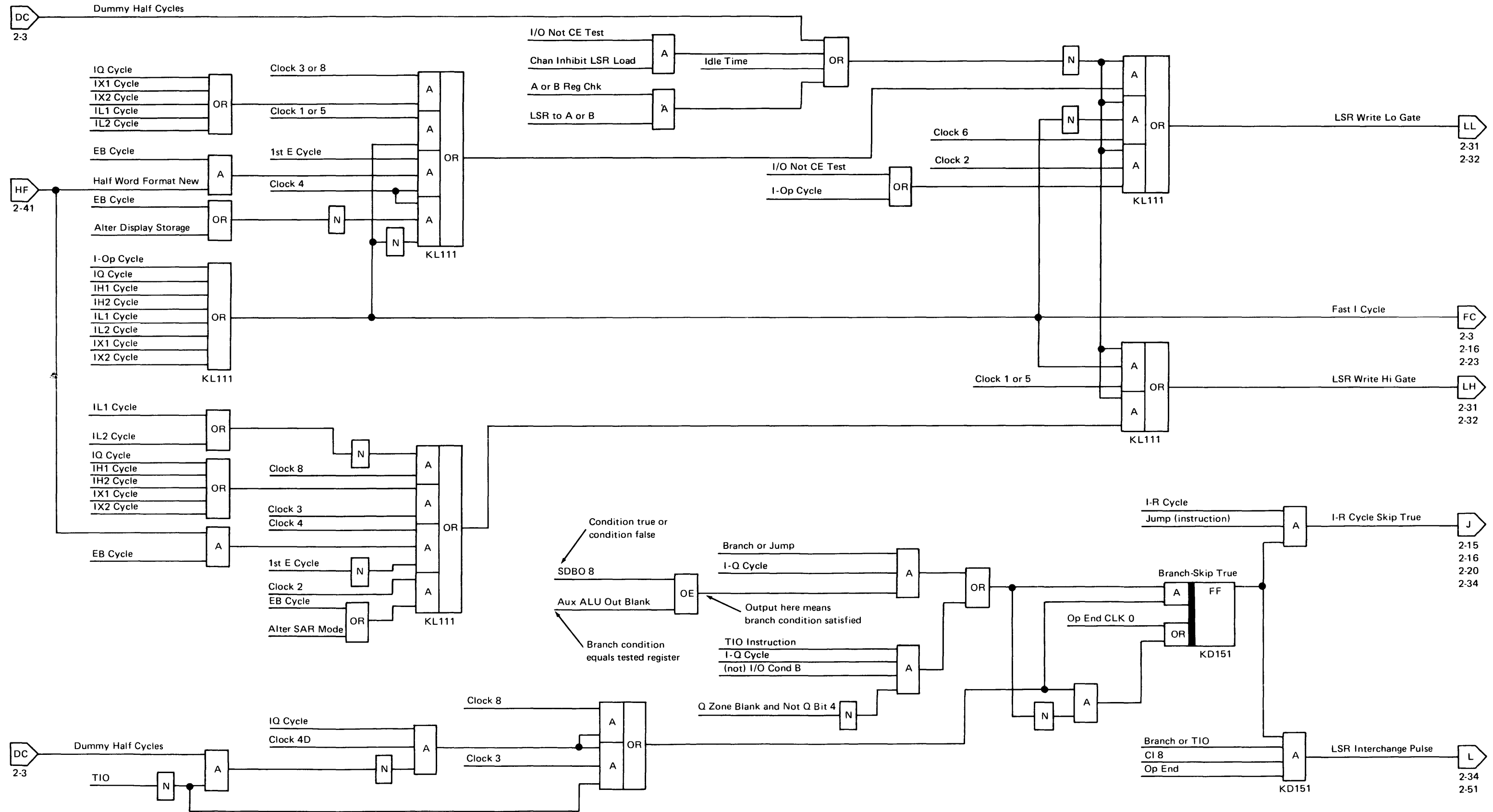


### I/O LSR Address Bit Generation

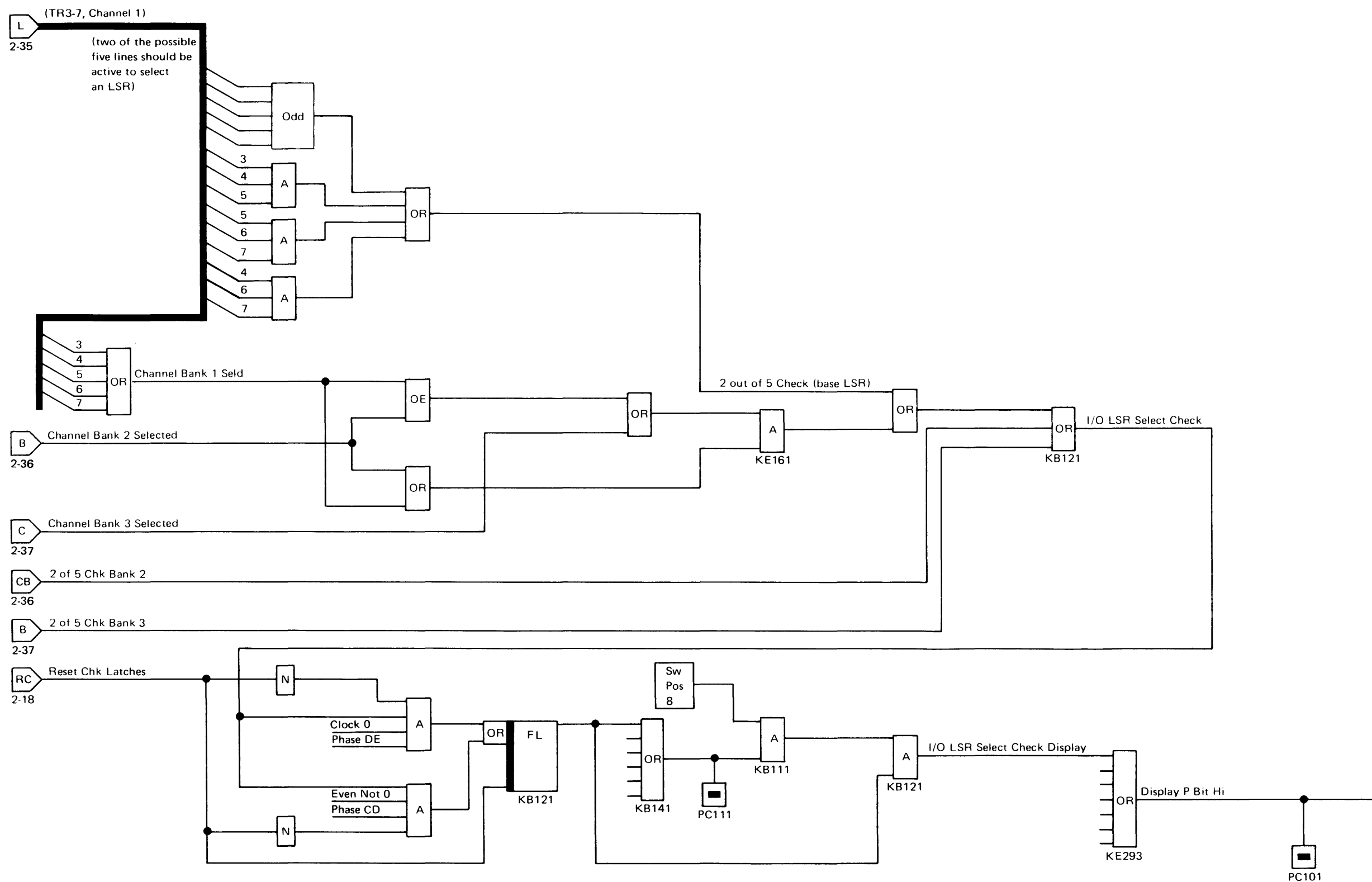




# LSR Controls

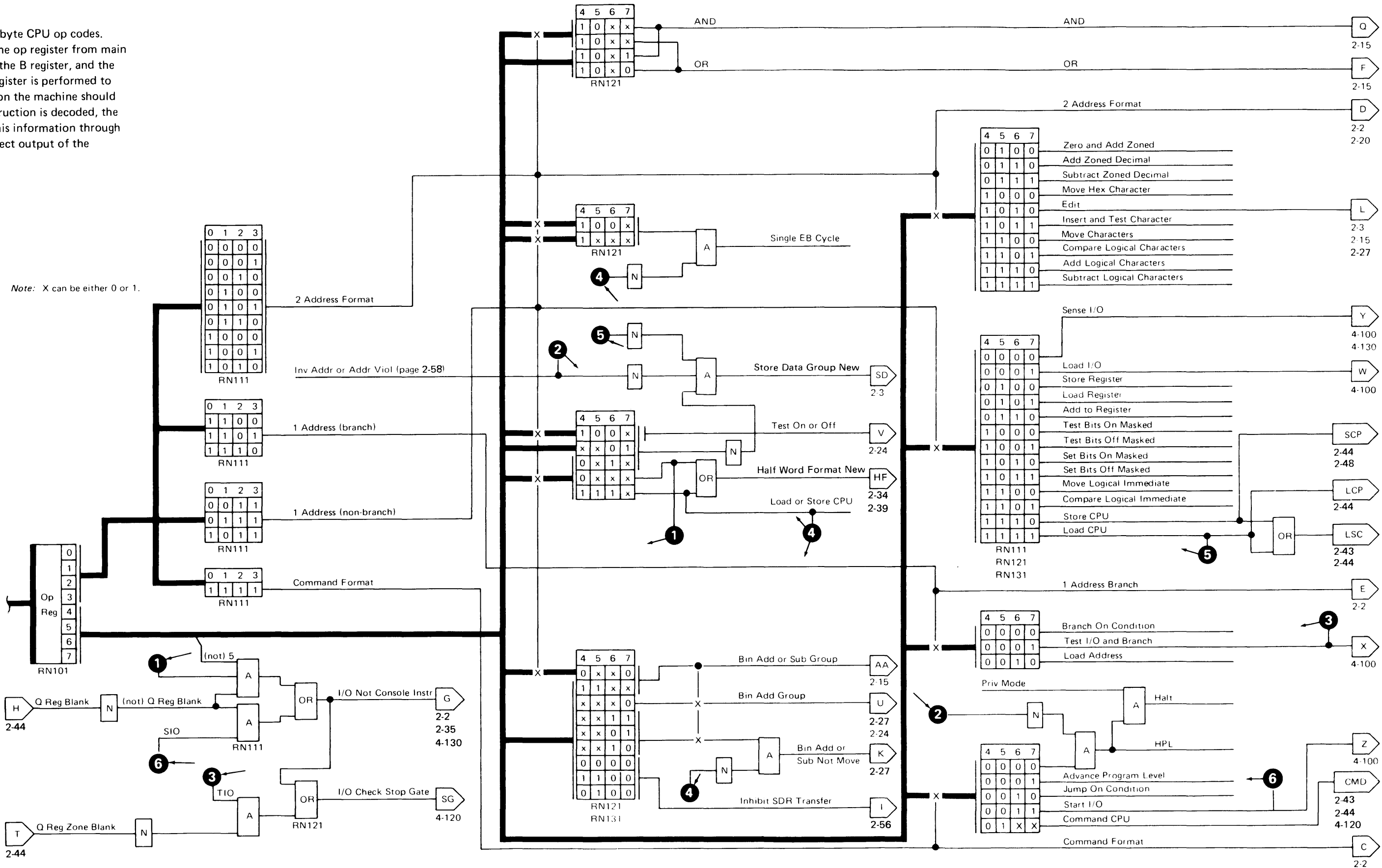


I/O LSR Select Check



# OP REGISTER

The op register stores one-byte CPU op codes. The op codes are sent to the op register from main storage through the SDR, the B register, and the ALU. Decode of the op register is performed to determine which instruction the machine should execute. When an I/O instruction is decoded, the I/O attachments receive this information through signal lines which are a direct output of the decode logic.



## PRIVILEGED MODE/MODEL 10 COMPATIBILITY

### Privileged Mode

Privileged mode prevents user programs from issuing I/O commands that could destroy efficient multiprogramming or destroy another user's program. Privileged instructions are issued only by the supervisor when the CPU is in privileged mode.

The privileged commands include all I/O commands, APL, HPL, store CPU, load CPU, command CPU (except Q = 10), and register commands (L, ST, A) that handle interrupt IARs. These privileged commands can only be executed when the system is in privileged mode. Attempts to execute these commands when the system is not in a privileged state causes a program check interrupt or a processor check if program check interrupt is disabled (see page 2-43).

Privileged mode is controlled by bit 4 of the PMR except when the system is forced to privileged mode because the CPU is in interrupt level 0. In all other levels, the program controls privileged mode through the PMR.

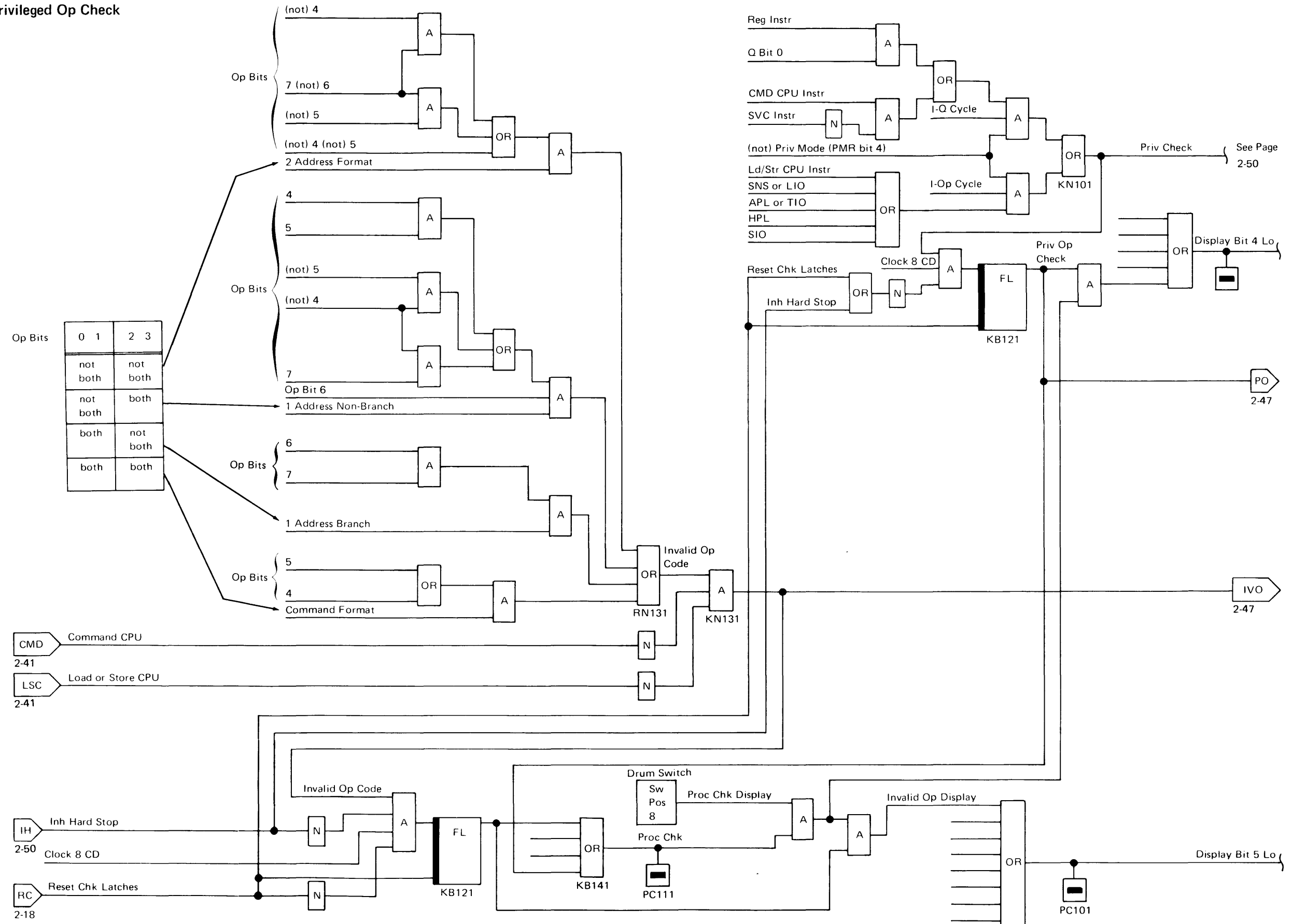
### Model 10 Compatibility

The CPU is forced to Model 10 mode by system reset, IPL, or power on. The Model 15 enhancements are disabled and privileged mode is on (all instructions are executable). The system remains in this mode regardless of the contents of any PMR, until the first instruction to load any PMR is encountered. At that point, the PMR contents determine the system mode.

While in Model 10 mode, the Model 15 will execute Model 10 programs with the following restrictions:

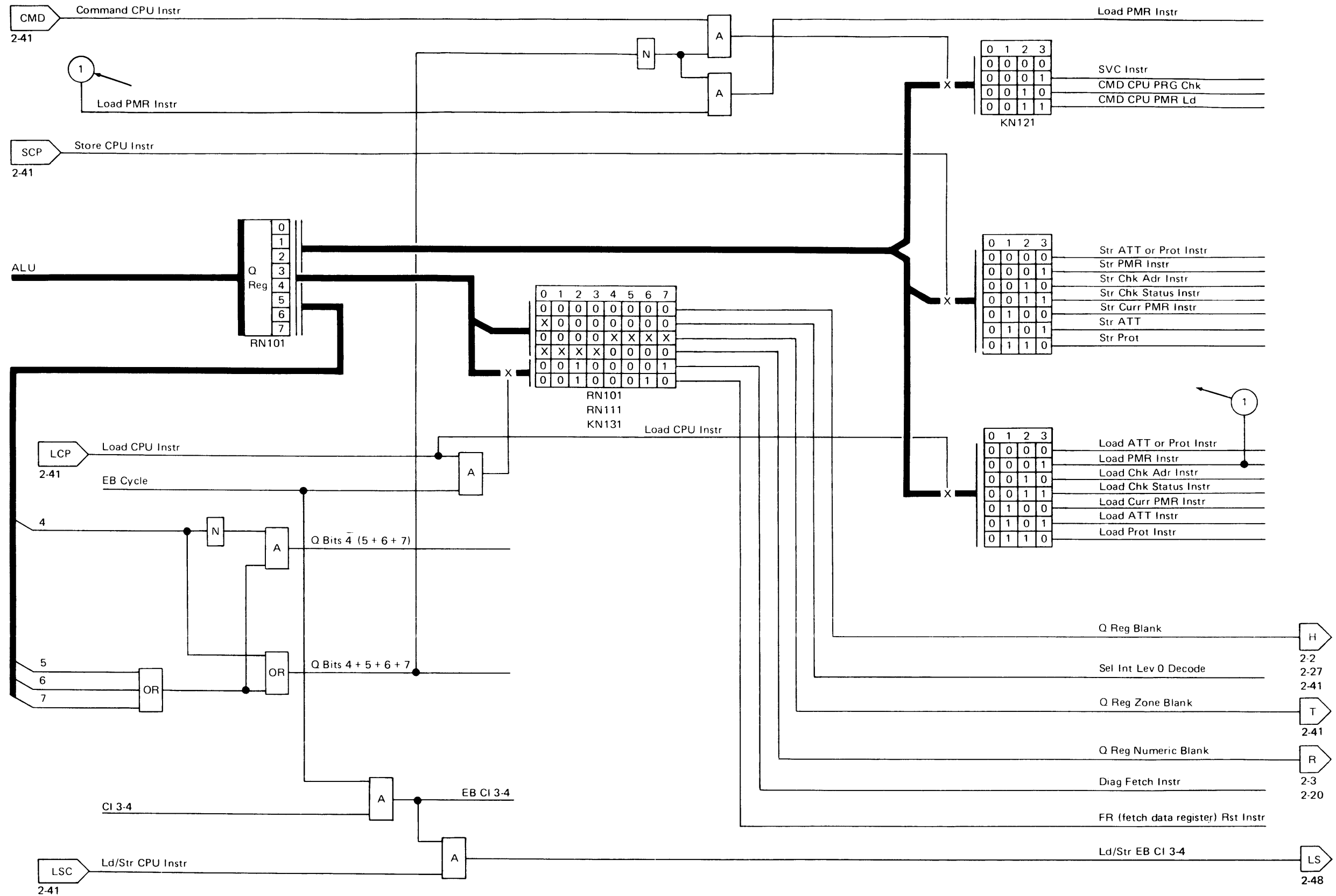
1. Programs that fail on a Model 10 may fail in a different manner on the Model 15. The instructions that were added to the Model 15 are invalid on the Model 10. Furthermore, I/O devices have N codes defined for op-end interrupt control that are invalid on the Model 10.
2. Dual programming does not exist on the Model 15. Therefore, TIO, SIO, and APL instructions do not select the P2 IAR. Instead, the Q code selects the P1 ARR.
3. TIO, SIO, and APL instructions with a device address of 0 (Model 10 DPF) are treated as a no-op.

Invalid Op Code and Privileged Op Check

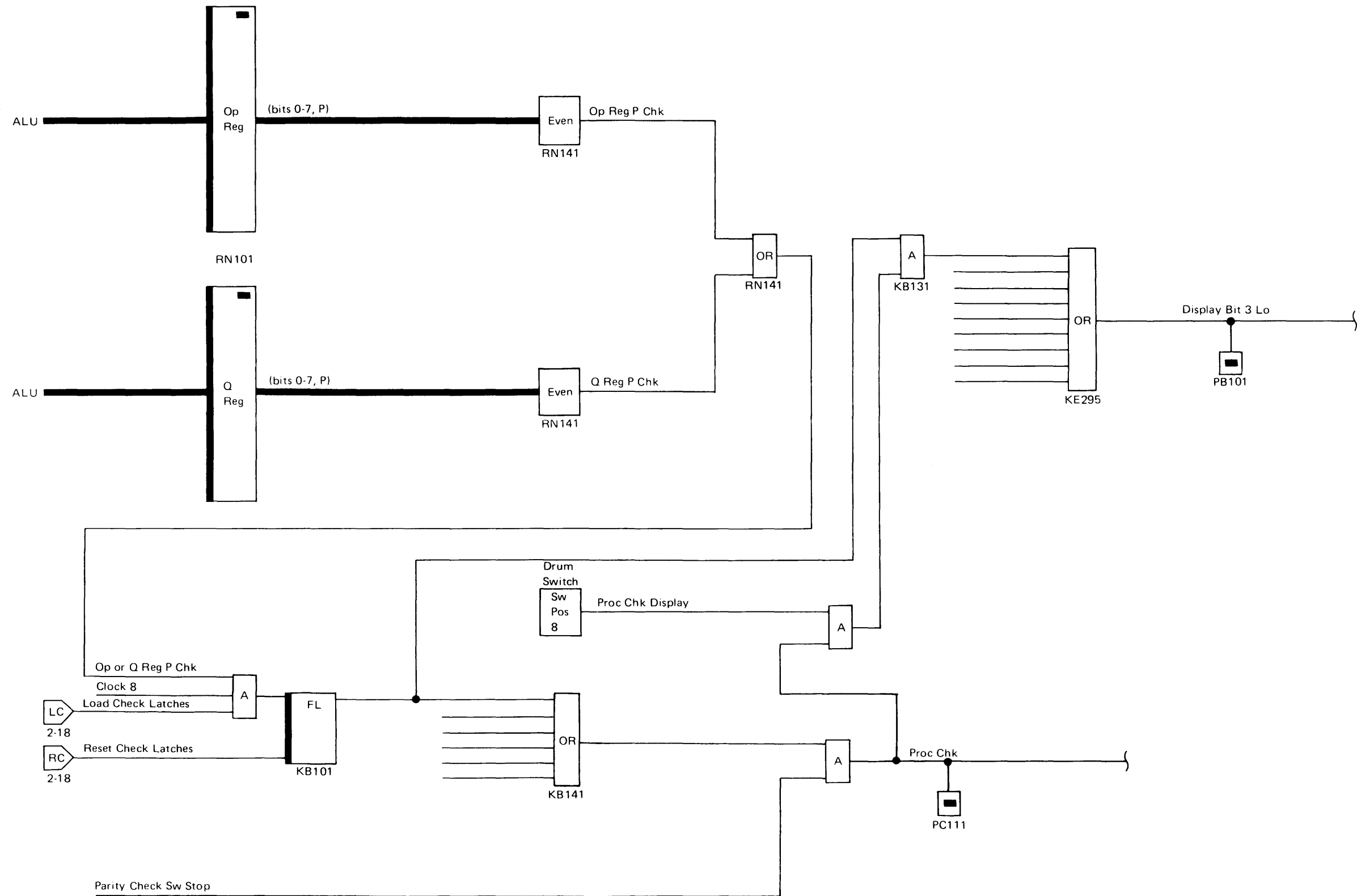


### Q REGISTER

All instructions include a Q byte that serves to extend or modify the op code. During the execution of an instruction, the Q byte is stored in the Q register. This register is loaded through the same data path as the op register and its output is decoded to determine the necessary information for the CPU to execute the instructions. During the execution of I/O instructions, the Q register stores the Q byte, but each I/O device also receives the Q byte from the ALU output on DBO at the same time. The CPU does not use the Q byte of the I/O instruction even though it is stored in the Q register.



Op and Q Register Parity



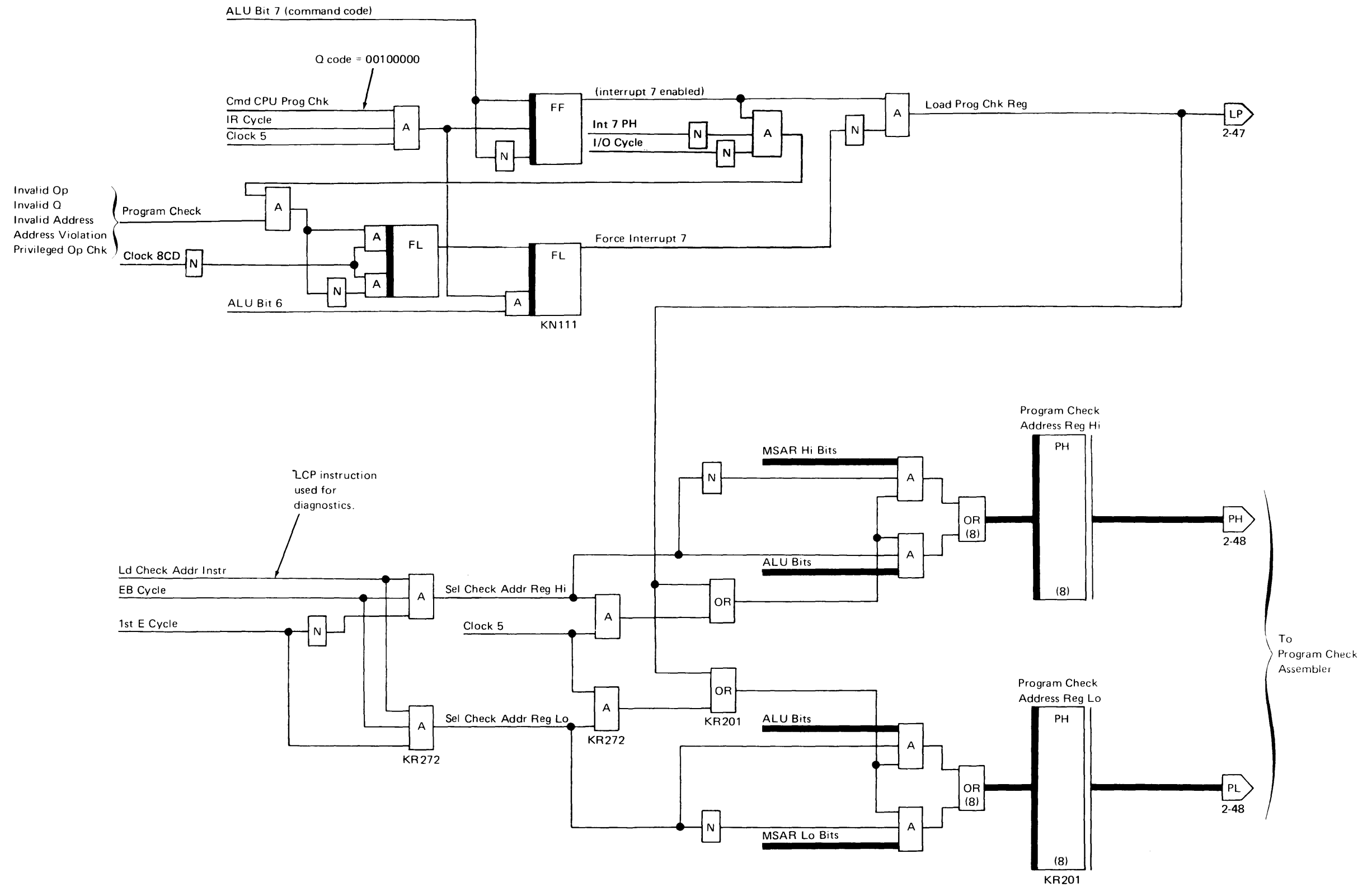
### PROGRAM CHECK REGISTERS (Part 1 of 2)

If multiprogramming is to be efficient errors caused by one user must not stop the system and deprive another user of processing time.

Through the use of the program check interrupt and the program check registers, the CPU program can (1) analyze the conditions that caused the check, (2) print out an error message, and (3) transfer CPU control to another user.

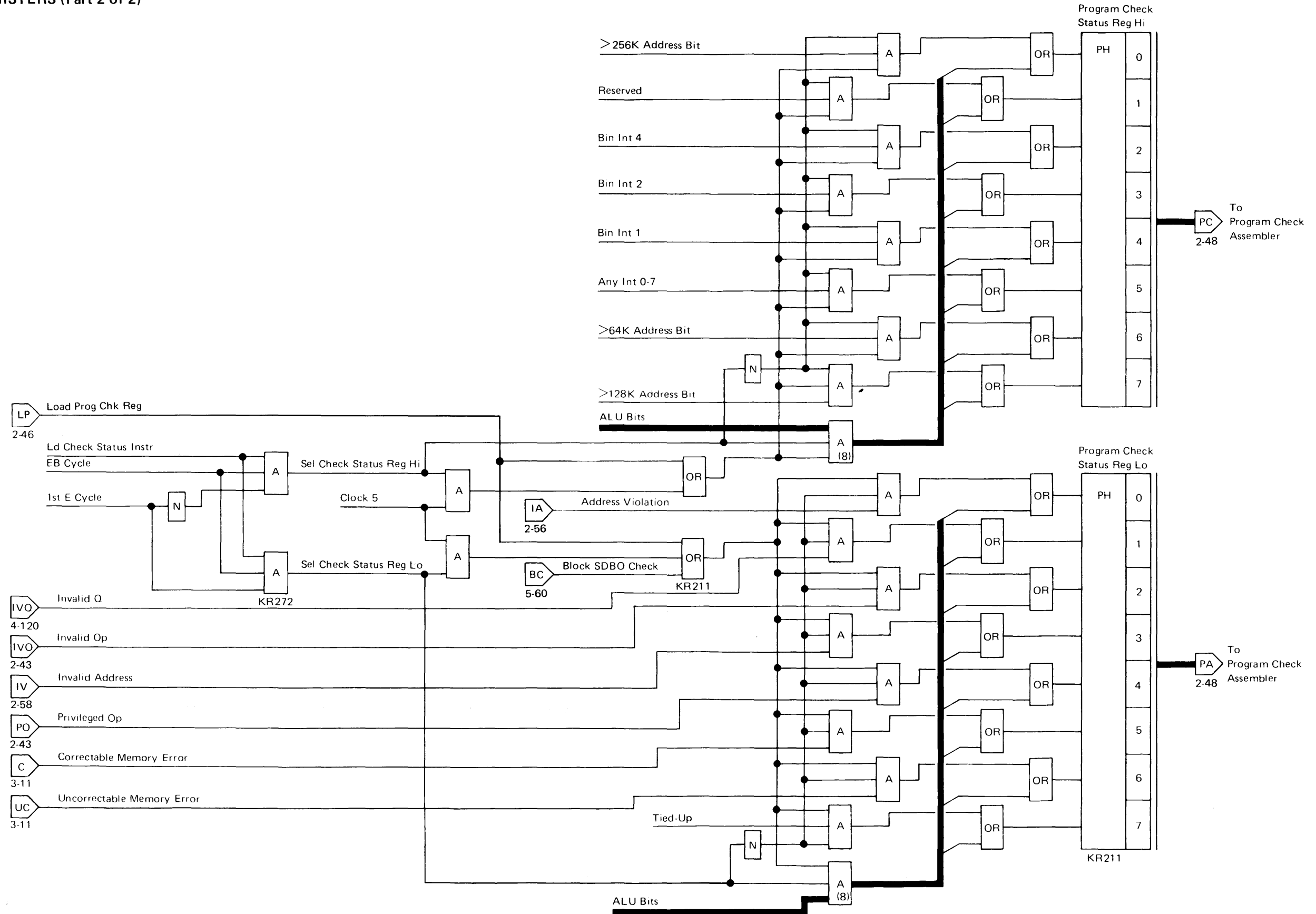
The program check registers contain information saved from the cycle during which a check last occurred. If interrupt 7 (program check) is enabled, these registers are set to reflect the system condition when a program check occurs. The register information is reset when interrupt level 7 is reset.

The program check registers can also be loaded with a load CPU instruction (LCP) or their contents can be moved to storage with a store CPU instruction (SCP).





PROGRAM CHECK REGISTERS (Part 2 of 2)





## INTERRUPTS (Part 1 of 3)

The CPU performs its computations in a step-by-step procedure until altered by a branch (refer to page 5-51) or an interrupt.

All interrupts follow the same general outline; (1) interrupt the program in progress (always at op-end), (2) execute the requested program, (3) return control to the interrupted program.

The 5415 processing unit has eight interrupt levels. Five of these levels are devoted to I/O devices and three support CPU functions. The interrupts listed below are from highest priority to lowest:

Interrupt Level	Priority	Function Performed
7*	1	Program check — Handles soft errors.
6	2	Interval timer SIOC MLTA/BSCC BSCA } I/O control and data transfer
4	3	
3	4	
2	5	
5*	6	Device op-end — Notifies the CPU that the I/O device has reached end of operation.
1	7	CRT display — I/O control and data transfer
0*	8	Supervisor program — transfers control from a problem program to the supervisor program.
None		Main program level

\*Description on facing page.

The interrupt routine being performed is established by the interrupt priority latches. As in cycle steal, the highest interrupt level device takes precedence over lower level devices. Thus, it is possible for an interrupt routine to interrupt a routine of a lower priority device. However, each device maintains its interrupt request until it is satisfied, so the lower priority device finishes its routine upon completion of the higher level routine.

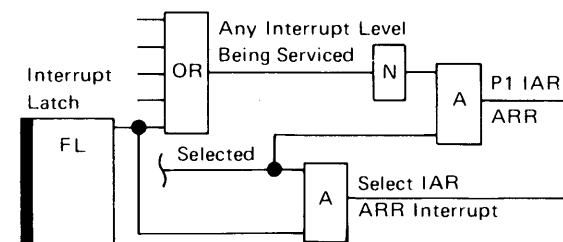
Each interrupt level has a separate IAR, ARR and PMR in the CPU so these registers for the main program are not disturbed. Any other registers (CR or index registers) used during the interrupt must be stored at the beginning and re-established at the end of each interrupt routine.

The stored program controls the capability of a device to interrupt by enabling and disabling the device through SIO, LIO, or SVC instructions. Once an interrupt has occurred, the interrupt routine is also ended by the same instruction.

During the I-Q cycle, device selection occurs in the same manner as any SIO instruction. Then at clock 5 of the I-R cycle, the control code is sent to the device attachment on DBO. The control code is decoded by the device attachment to turn on the 'interrupt enable' latch. This latch remains on until a disable control code is sent in another instruction.

If a device has a need to interrupt, the 'interrupt request' latch is turned on. At the end of the operation being performed in the CPU, an interrupt poll is sent to the device. This activates the 'DBI bit' line to turn on the interrupt latch for the device in the CPU. If more than one 'DBI bit' line is active, only the highest priority interrupt latch is turned on (see diagram on page 2-51).

With any interrupt latch on, the selection of the normal IAR/ARR is blocked and the IAR/ARR for the active interrupt level latch is selected. The interrupt request latch in the device attachment stays on until an instruction with the proper control code resets it.

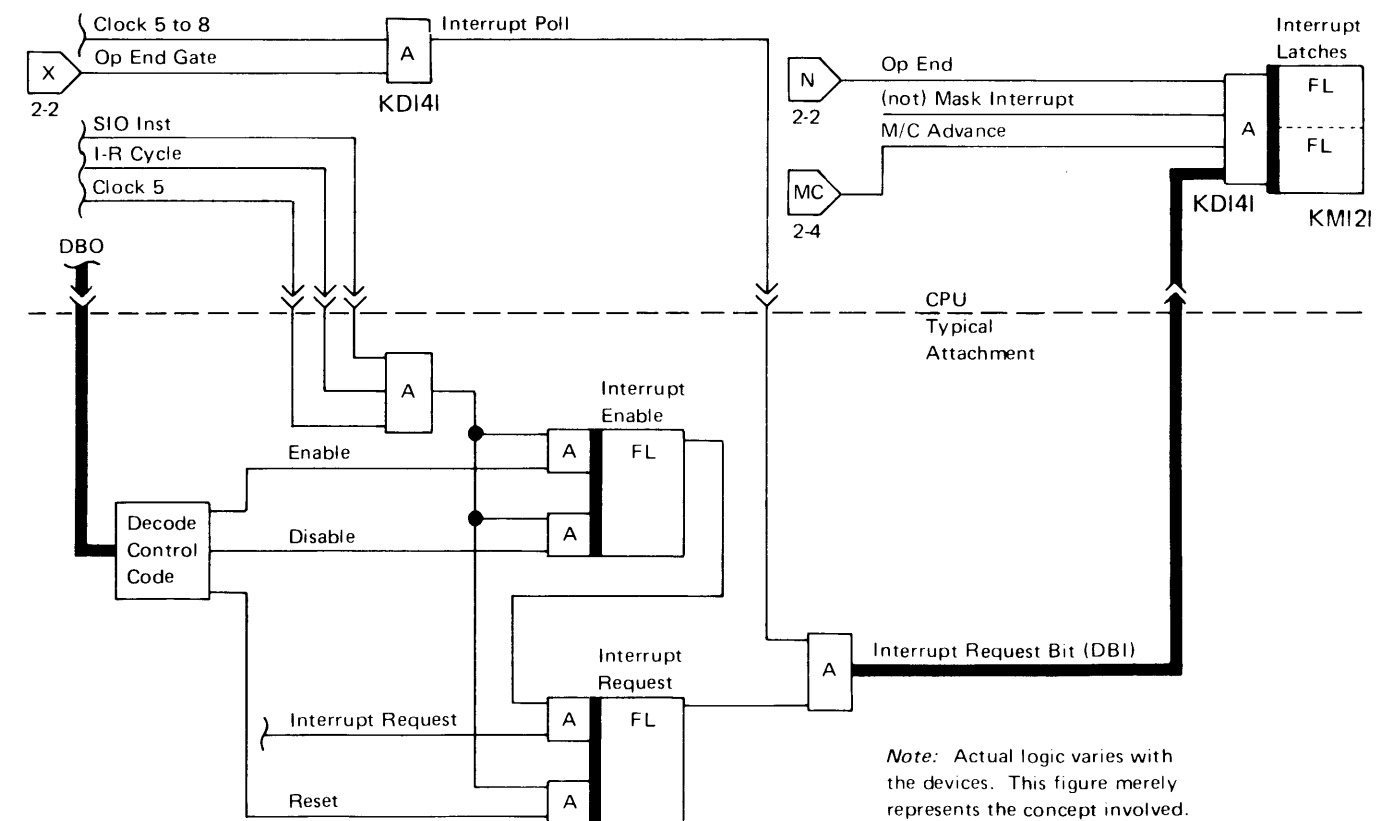


## Interrupt Mask

A mask function is provided to simplify interrupt processing. This function gives the programmer the ability to complete a routine before it is interrupted by a higher priority program.

The mask interrupt function is controlled by a bit in the PMR. When this bit is on, any higher priority interrupt request remains pending until the mask is set off. The exception is interrupt level 7 (program check interrupt). It is not affected by the mask.

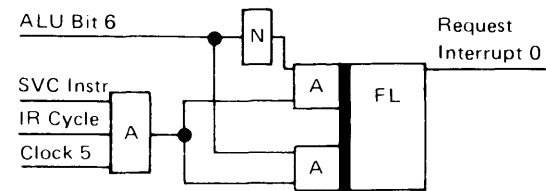
The interrupt mask bit in the PMR must be set off before an interrupt is reset. Failure to do so will cause the CPU to remain in that interrupt.



**INTERRUPTS (Part 2 of 3)**

**Supervisor Program (Interrupt Level 0)**

A user program passes CPU control to the supervisor by initiating an interrupt on level 0. This is accomplished by using the Command CPU instruction with a Q code of 10 (supervisor call). Refer to Command CPU instruction description on page 5-67.



**Op End Interrupt (Interrupt Level 5)**

When an I/O device has completed the operation in progress, and 'op end interrupt' is enabled in the device, 'op end interrupt request' is activated. At the end of the CPU operation being performed, 'interrupt poll' is activated. 'Interrupt poll' and 'interrupt request' combine to turn on the 'interrupt level 5' latch in the CPU.

Once CPU control is transferred to the interrupt level 5 program, that program determines which device initiated the op end interrupt. This is done through TIO and SNS instructions. The determination of which device requested the interrupt and what priority each device holds, is a programming function.

**Program Check Interrupt (Interrupt Level 7)**

For the Model 15, program check interrupt allows the CPU to enter an interrupt routine rather than a processor check hard stop condition. Errors causing a program check interrupt are:

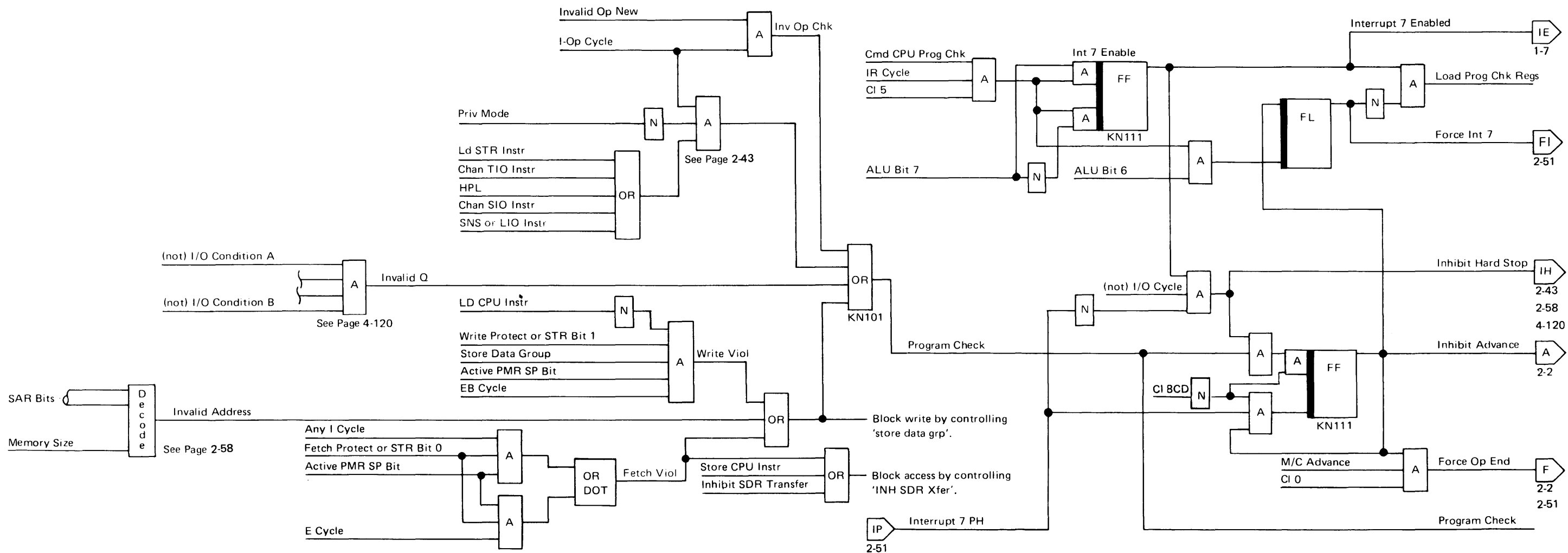
- Invalid address
- Invalid Q
- Invalid Op
- Privileged Op
- Storage violation

The occurrence of any of these errors during the program check interrupt routine causes a processor check hard stop. Invalid address during I/O cycles also causes a processor check hard stop.

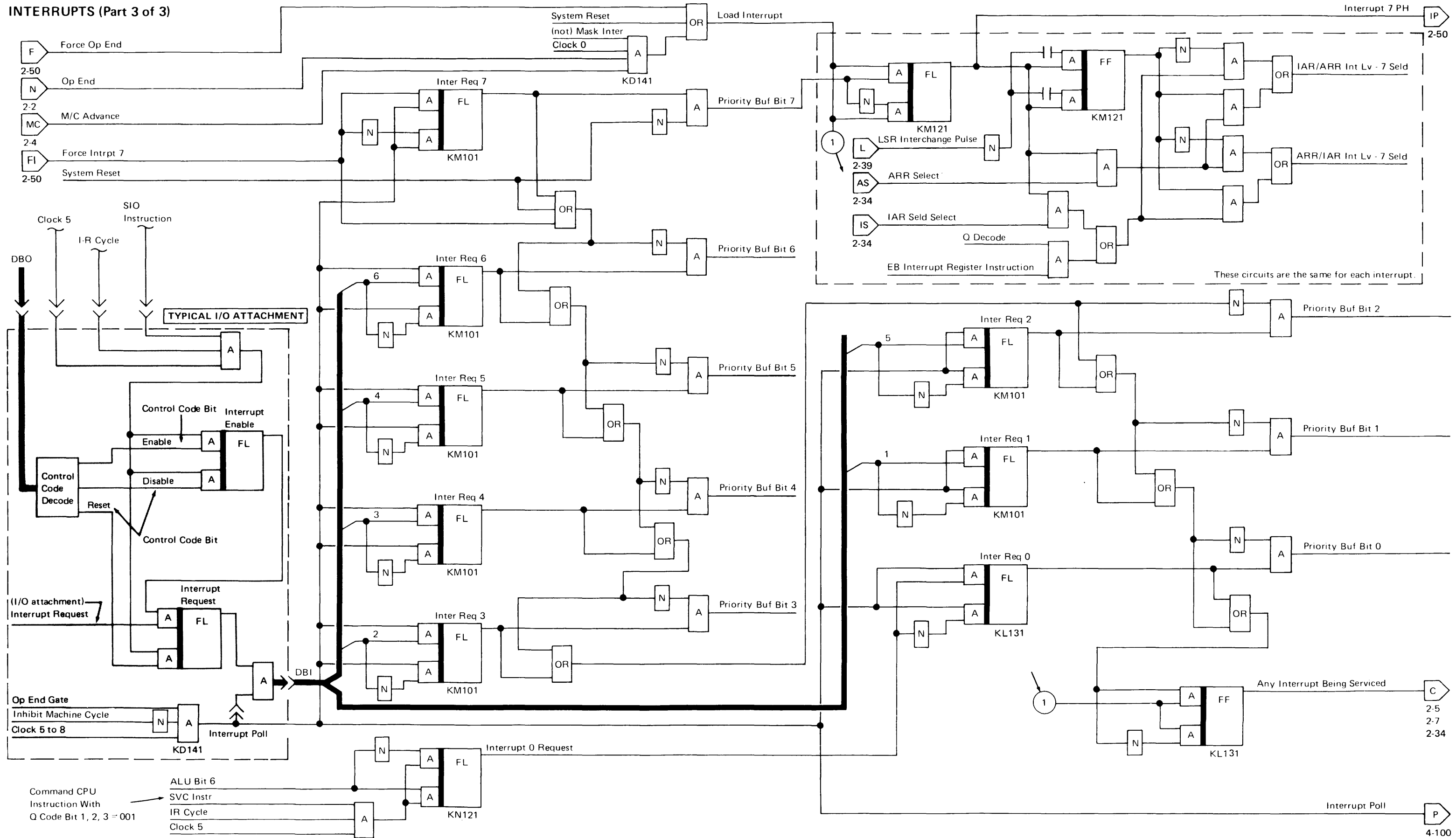
The program check interrupt routine can analyze the cause of the error from status provided by the CPU hardware, prepare a message for the user causing the error, and then transfer CPU control to another user thereby making maximum use of the CPU time.

The program check interrupt is assigned to level 7 which is the highest priority interrupt. The program check function must be enabled by a command CPU instruction. If the function is not enabled, the CPU performs like a Model 10 in that the checks mentioned cause a processor check hard stop. The command CPU instruction is also used to disable and reset interrupt level 7. The status required to analyze the error source is provided in registers that may be stored using the store CPU instruction. This status includes the specific check, the physical main storage address at the time of error, and the active interrupt level, if any, at the time of error.

A description of the program check registers can be found on page 2-46. A description of the program mode registers can be found on page 2-28.



**INTERRUPTS (Part 3 of 3)**



### INTERVAL TIMER (Part 1 of 2)

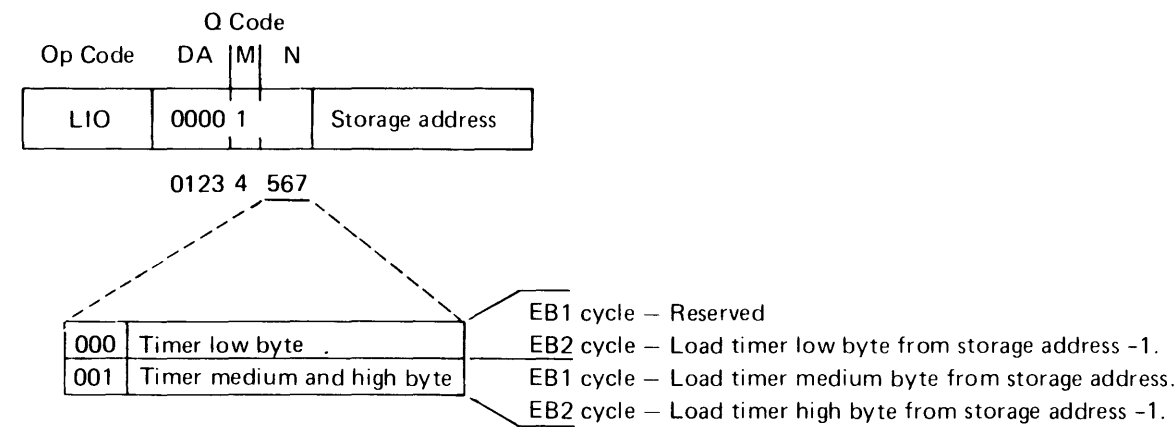
The interval timer is a 24 bit (3 byte) binary counter that is decremented every 3.3 ms. The CPU treats the timer like an I/O device. LIO, SIO, and SNS instructions are used for control. I/O cycles are not used, however, so the timer does not affect the system I/O burden.

A value is set into the counter by an LIO instruction and the contents sensed by using an SNS instruction. SIO instructions are used to enable, disable, and start the counter. Once started, the counter is decremented by one every 3.33 ms. ('adv int timer'). The counter overflows when the count goes from zero to a negative value and an interrupt request is generated (level 6). If a value of FF FF FF is loaded and the timer started, an interrupt occurs approximately 15.5 hours later.

The timer uses a separate oscillator and clock that continues to decrement the timer during halts, stops, and interrupts. Also, the timer does not stop when an overflow occurs. It continues to run until stopped by an LIO, a stop timer command (SIO), a system reset, or by stopping the system clock. Once stopped, the timer does not resume counting until an SIO is issued.

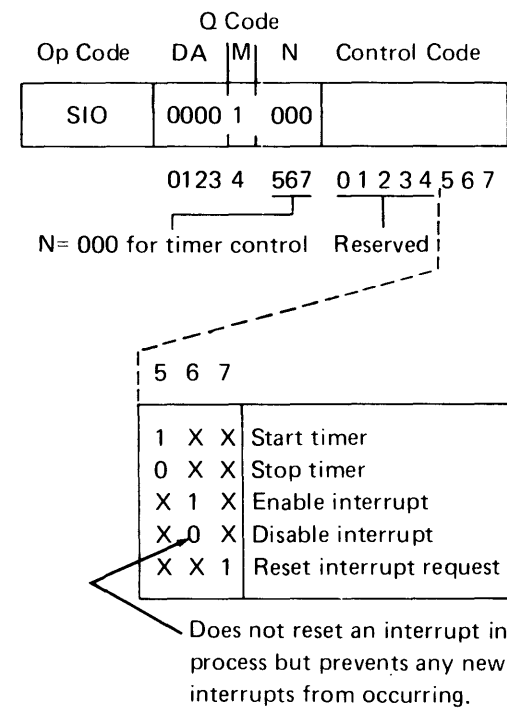
I/O instruction format is shown in the following illustrations.

#### Load I/O



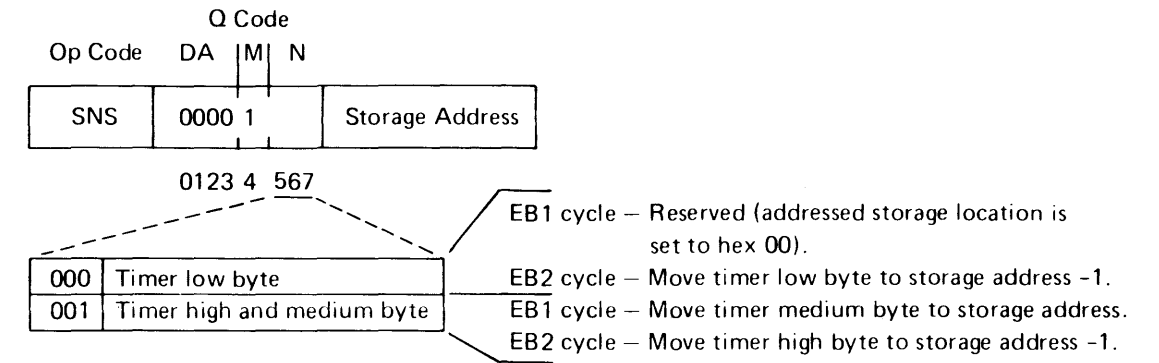
An LIO instruction to the timer causes decrementing to stop. When the timer value is completely loaded (two LIO instructions), the timer is started by an SIO (timer control) instruction.

#### Start I/O



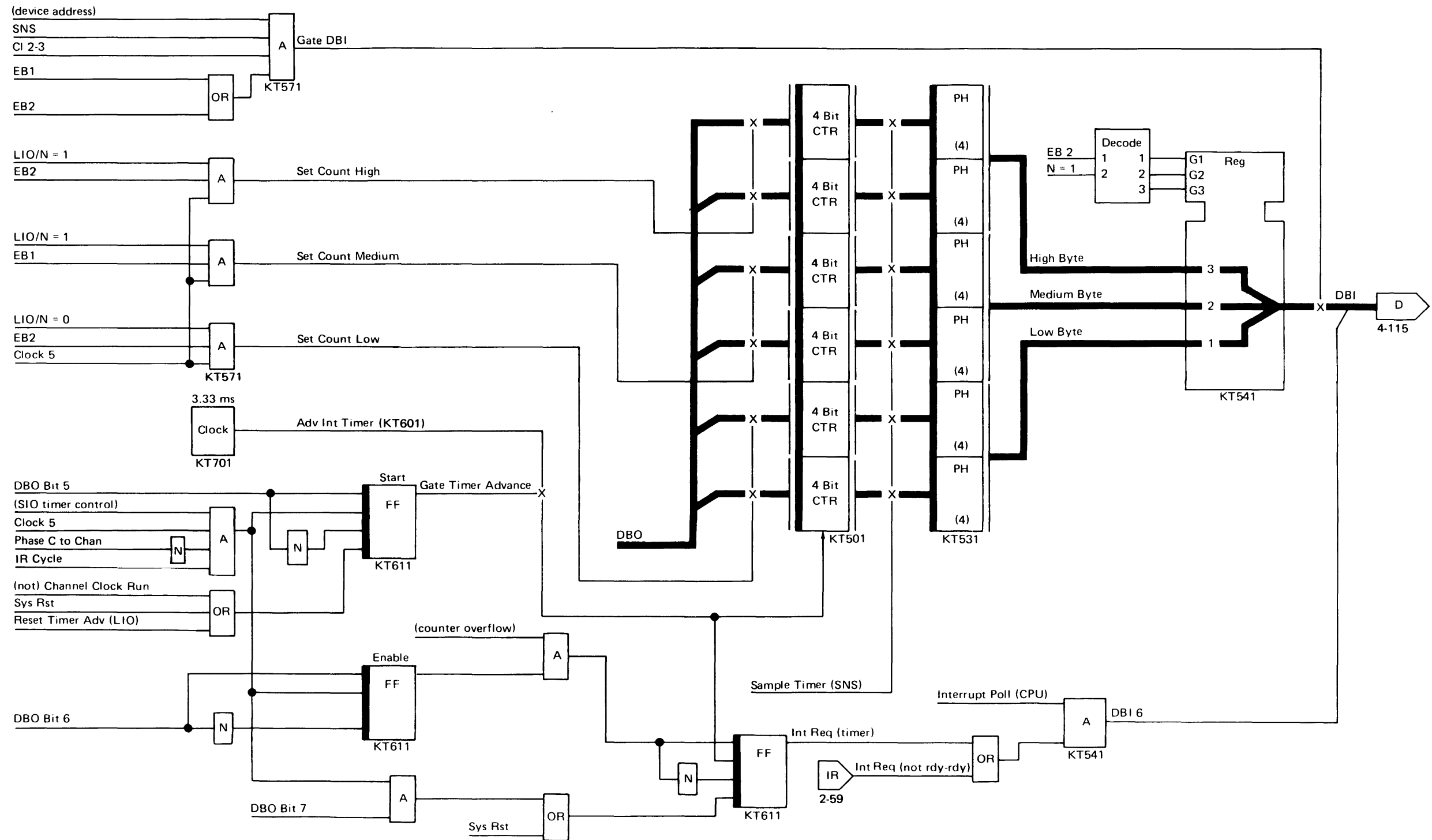
A start I/O command is always accepted.

#### Sense I/O



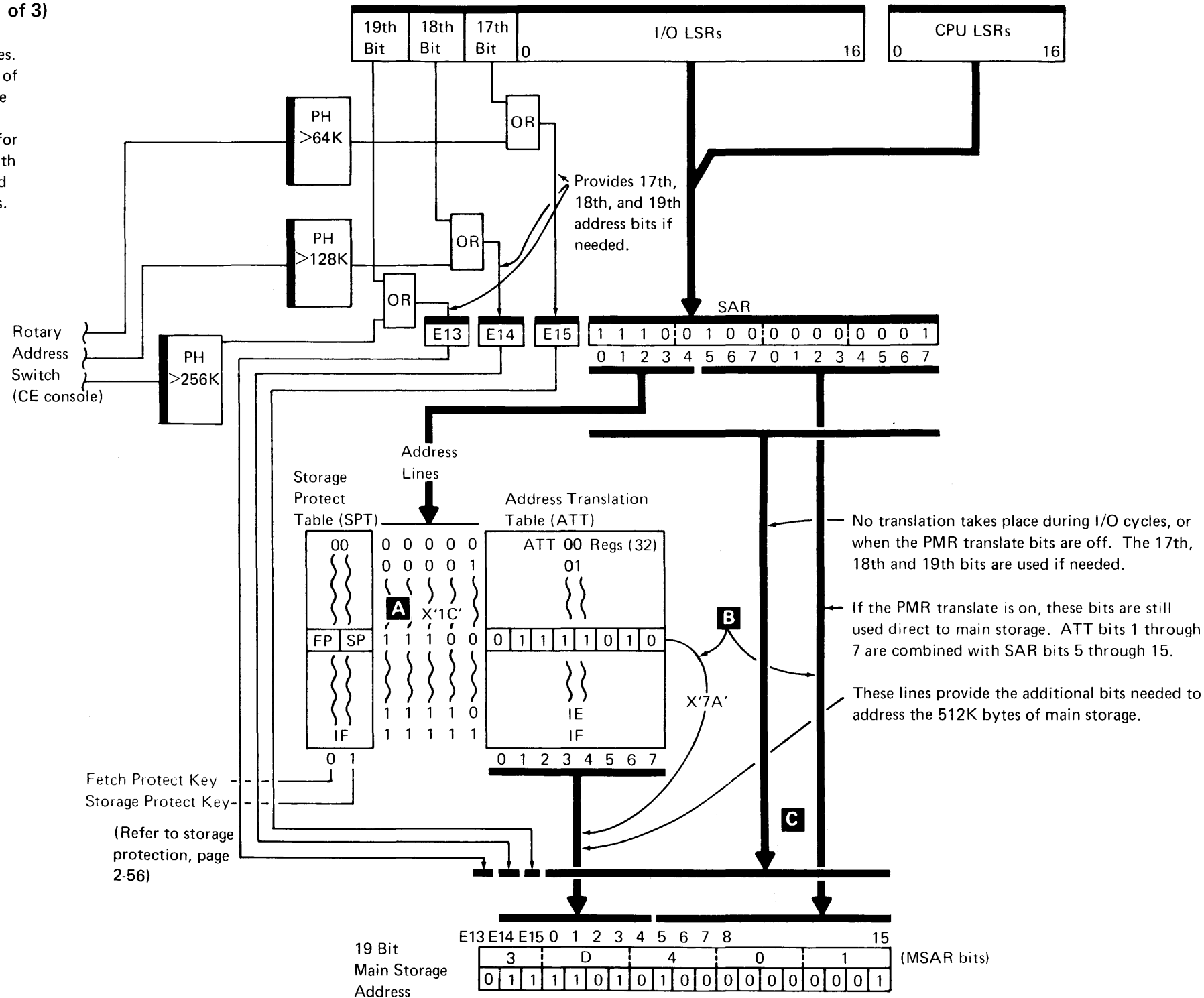
When moving the timer contents to main storage, the low-order byte must be stored first. When timer circuits detect an SNS byte with the N code equal to 0, the three-byte counter is transferred to a three-byte register. This prevents the possibility of presenting erroneous timer data if the timer should be decremented between the two SNS instructions that are necessary to store the entire timer contents.

INTERVAL TIMER (Part 2 of 2)



### MAIN STORAGE ADDRESSING (Part 1 of 3)

System/3 data flow uses 2 byte (16 bit) addresses. Sixteen bits control addressing up to 64K bytes of storage. To address 512K bytes of storage, three more address bits are required. These bits are provided by an Address Translate Table (ATT) for CPU operations, an I/O LSR 17th, 18th, and 19th bit for I/O operations, and a >64K, >128K, and >256K CE rotary switch for console operations.



### Address Translate Table

The address translate table consists of 32 registers (00-1F) that can be loaded by the program. These registers provide the high order six bits of the main storage address when the translator is being used. The PMR controls the translator with 3 bits:

- Bit 1 on = translate on B cycles
- Bit 2 on = translate on A cycles
- Bit 3 on = translate on I cycles

To use the ATT, the programmer must first load the ATT registers with the desired bits using LCP commands (ATT is not power on reset) and then set the PMR bits to control the translator as described previously.

Example:

Assume the following instruction is ready to be executed:

Op	Q	B field	A field
MVC	length	operand 1	operand 2
0C	00	0501	E401

- The PMR is set up to translate on A cycles only.
- ATT register 1C (hex) has previously been loaded with a value of 7A.

The A field address is E401 and is considered a *logical* address since it is to be translated into a *real* address by the ATT. The real memory address for this example is 3D401.

During the A cycle, address translation occurs:

- Five high order bits from SAR address the ATT **A**.
- Seven ATT bits and 11 SAR bits form the MSAR address 3D401 **B**. During the B cycle, address translation is inactive. Sixteen SAR bits form the MSAR address **C**.

The byte of data from MS address 3D401 is written into address 0501.



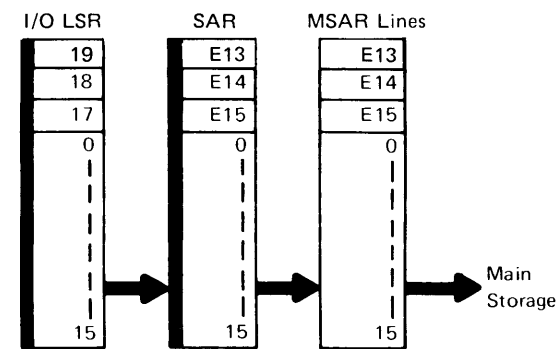
## MAIN STORAGE ADDRESSING (Part 2 of 3)

### I/O LSR 17th, 18th, and 19th Bits

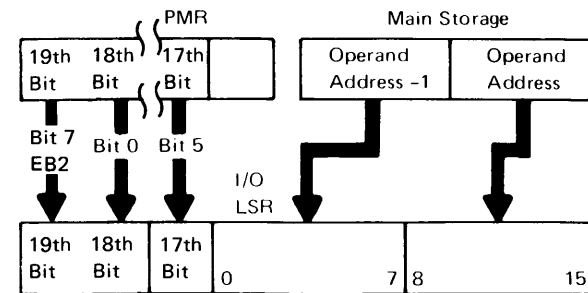
Because the ATT is not used on I/O cycles, each I/O LSR must contain the proper number of bits to address storage. The number of bits in each I/O LSR varies with main storage size (the maximum being 19 bits for 512K of storage). The following discussion assumes a main storage size of greater than 128K bytes.

I/O LSR bits 17, 18, and 19 are gated into the SAR E15, SAR E14, and SAR E13 positions during I/O cycles.

The SAR then addresses storage untranslated.

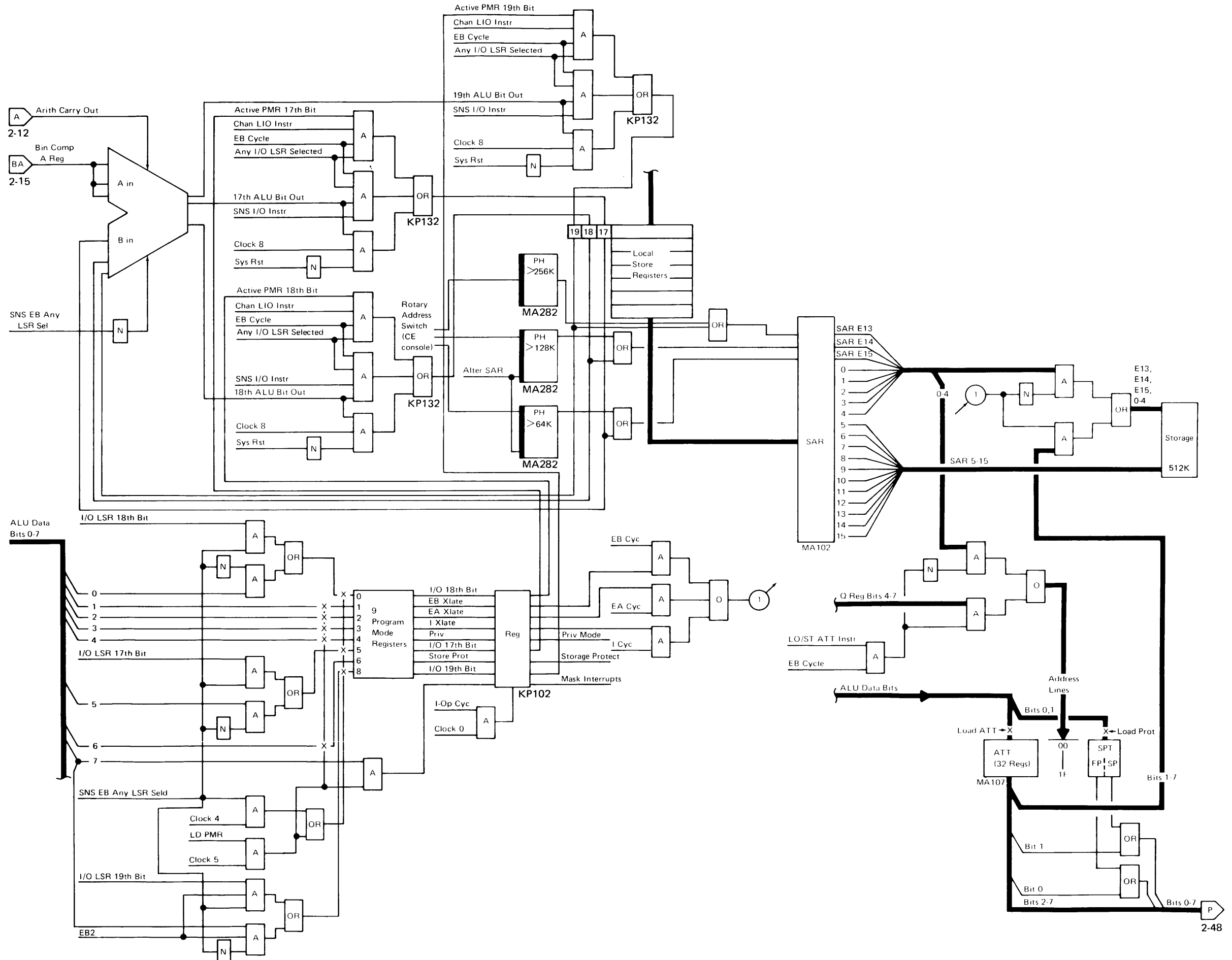


I/O LSRs are loaded by LIO instructions. The low-order 16 bits are loaded from storage during two consecutive B cycles; the 17th, 18th, and 19th bits are conditioned on the same LIO by the PMR 17th bit (bit 5), PMR 18th bit (bit 0), and PMR 19th bit (bit 7 of byte 2) respectively.



The 17th, 18th, and 19th I/O LSR bits are modified during I/O cycles each time a 64K boundary is crossed.

When an SNS instruction is issued to an I/O LSR, the 16 low-order bits are transferred to the specified main storage locations and the 17th bit is gated into the PMR 17th bit (bit 5) position. The 18th bit is gated into the PMR 18th bit (bit 0) position. The 19th bit is gated into the PMR 19th bit (bit 7 of byte 2).



## MAIN STORAGE ADDRESSING (Part 3 of 3)

### Storage Protect

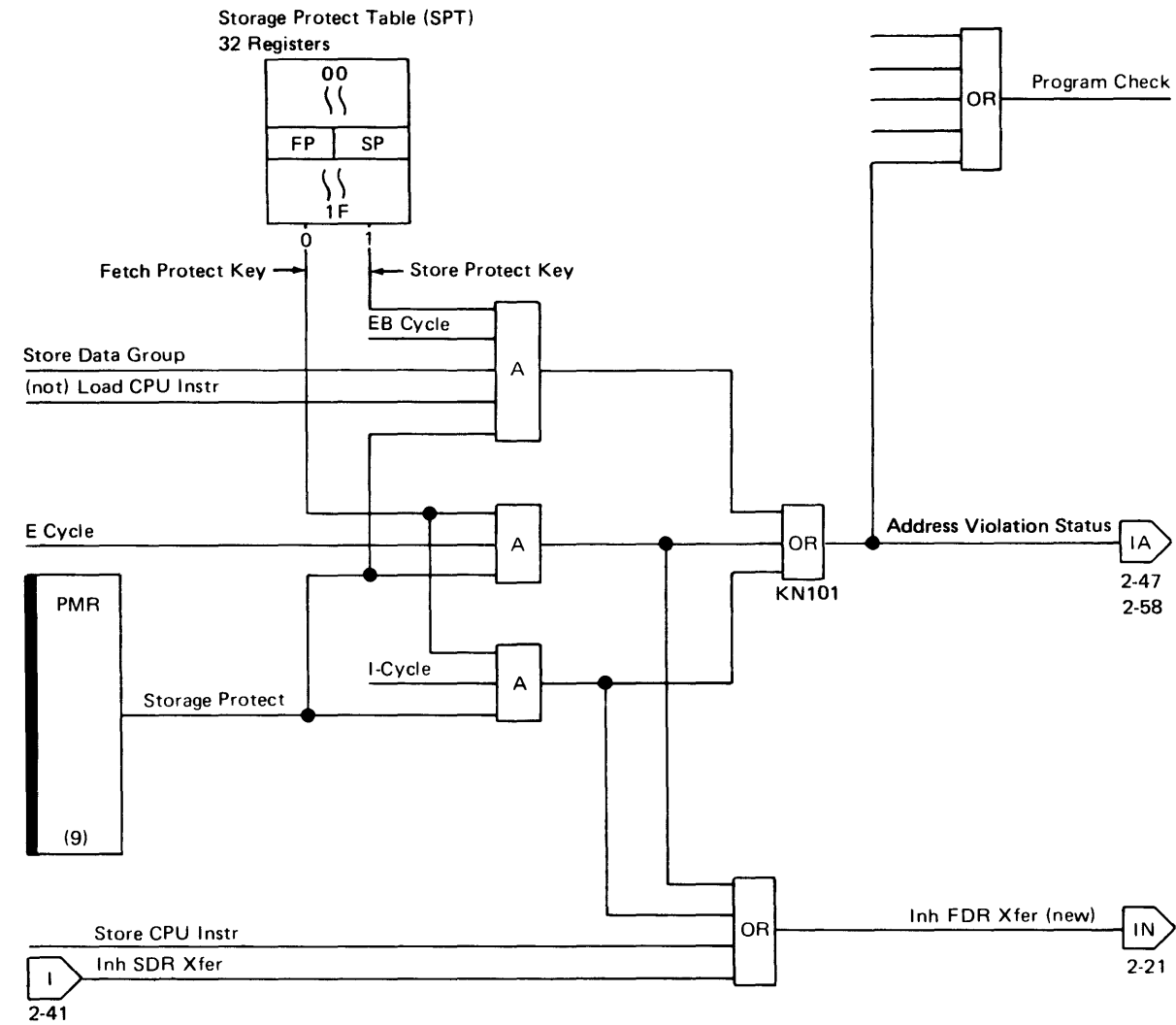
Storage protect provides the ability to protect main storage from unauthorized or accidental fetch and store operations. This is accomplished with two bits in each of the 32 registers of the storage protect table (SPT, protect keys). Bit 0 is the fetch protect key; bit 1 is the write protect key.

The storage protect function is activated by the storage protect bit in the PMR. When this bit is on, the protect keys in the SPT are inspected for every main storage cycle except I/O cycles. If the bit is off, the protect keys in the SPT are not inspected and any available location can be accessed or written into.

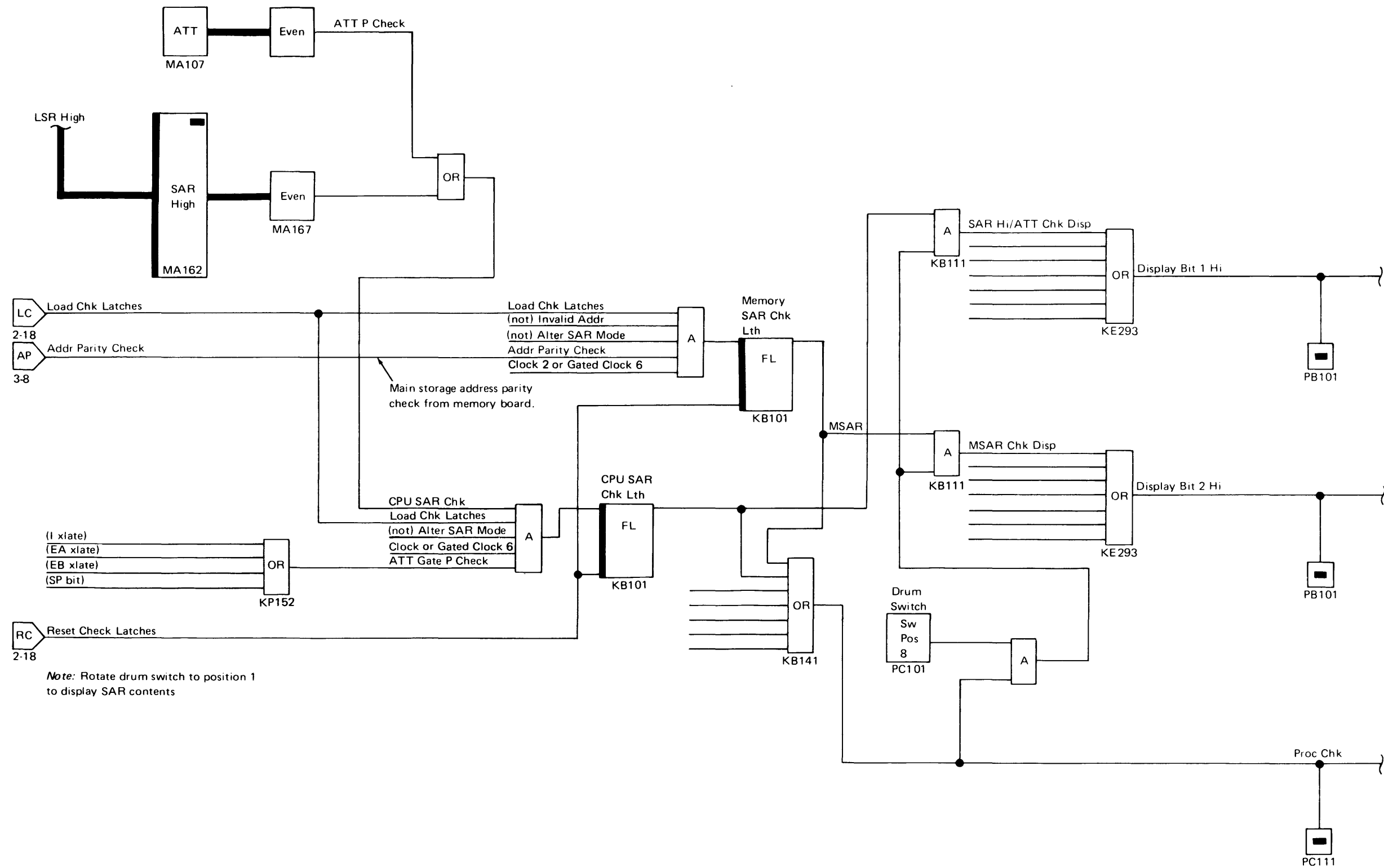
The storage protect hardware does not function on I/O cycles. The supervisor program must verify that I/O buffers reside in a user's allocated storage.

If attempts are made to fetch or write protected locations, a storage violation occurs and the CPU forces a program check interrupt (a processor check if the program check interrupt is not enabled).

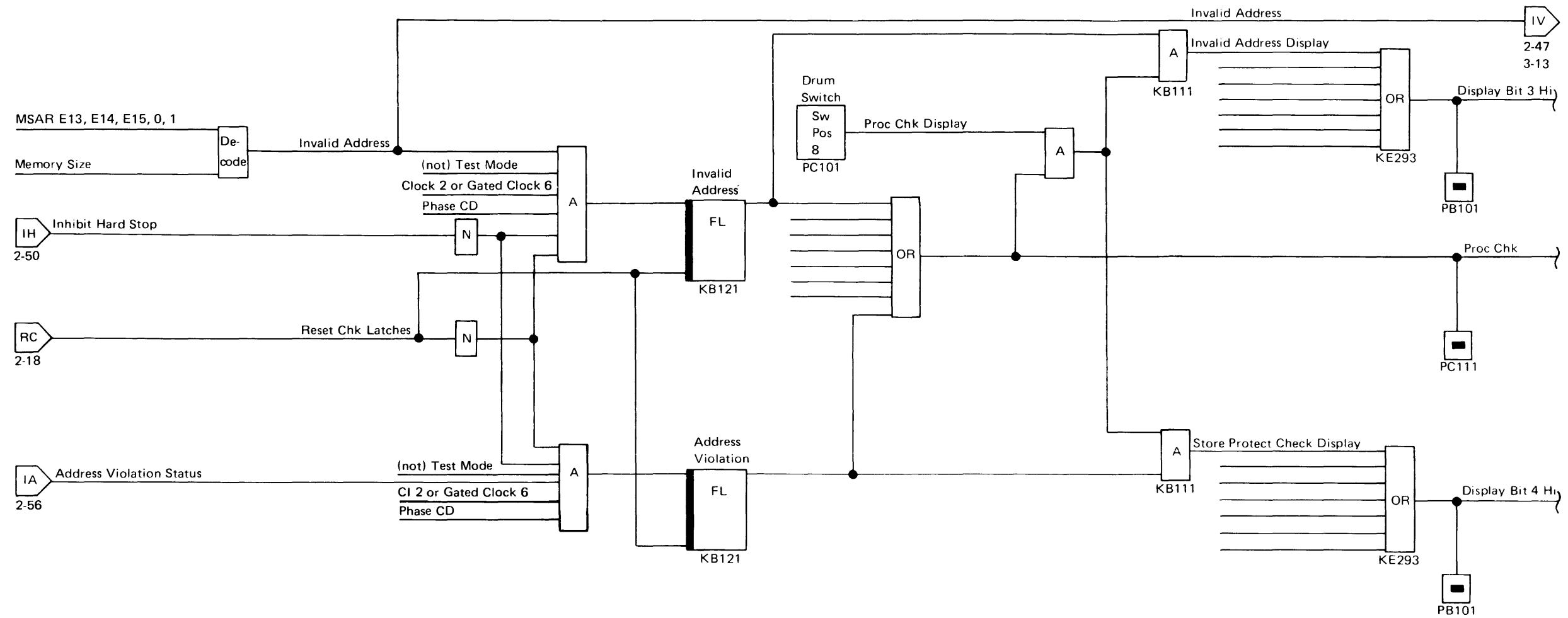
The protect keys are changed or saved with CPU instructions LCP/SCP. The contents of the protect keys cannot be predicted after the power up sequence and must be loaded by the user. The store protect bit in the PMR is set off during the power up sequence, so protection is not used until it is activated by the program.



# Address Parity Checks



Invalid Address/Storage Protect Check



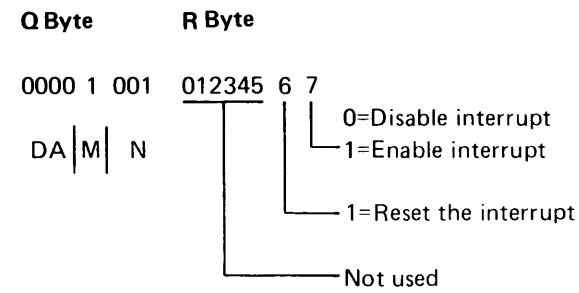
## Unit Record Restart

The unit record restart circuits detect a device not-ready to ready transition and interrupts the CPU program. Interrupt level 6 is shared with the interval timer. The interrupt routine determines which function interrupted on level 6 then which I/O device changed from not-ready to ready.

The CPU can enable, disable, and reset the interrupt using a SIO instruction and test for interrupt request pending using a test I/O instruction.

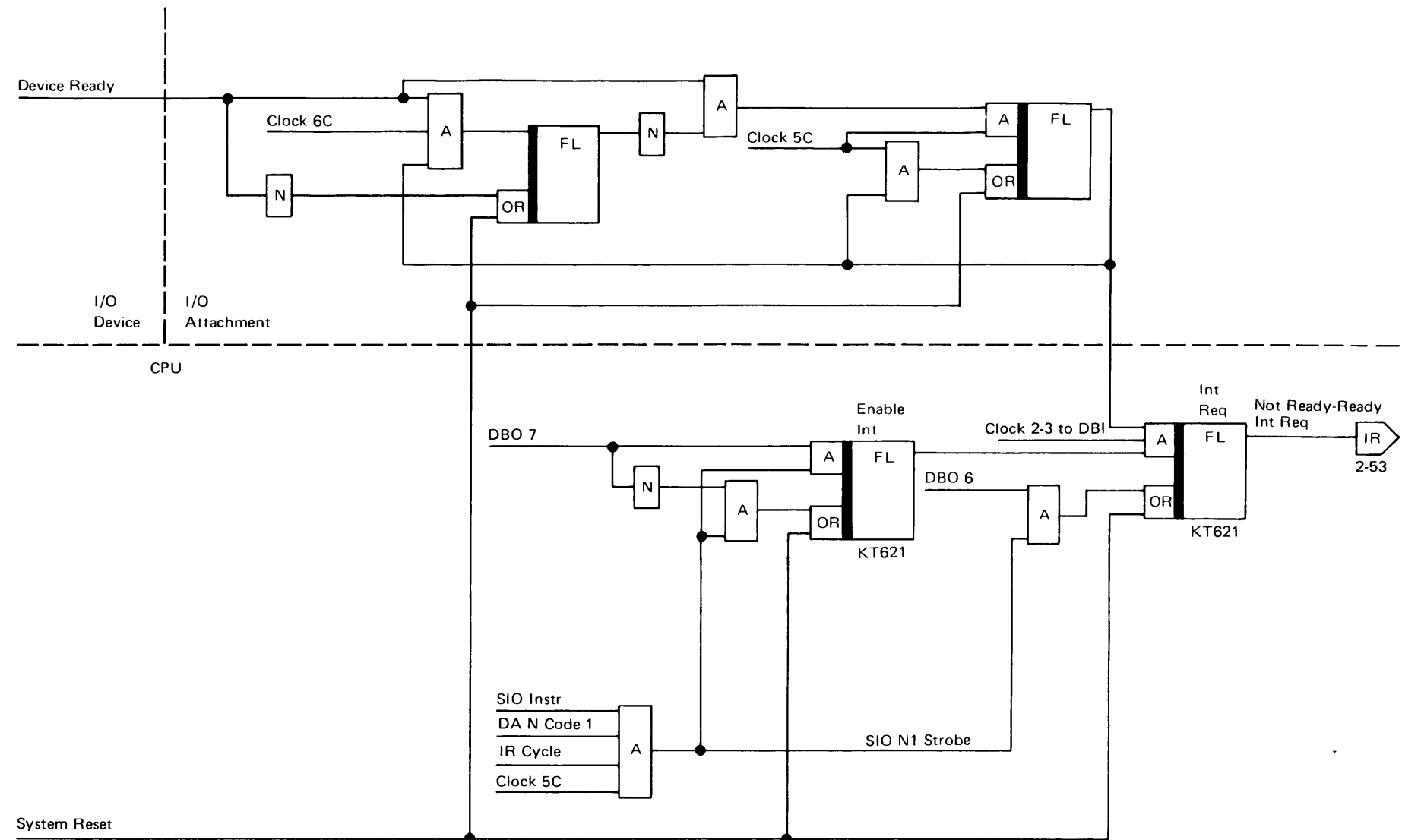
### Instruction Format

#### SIO Not-Ready to Ready Interrupt Control



#### TIO Not-Ready to Ready Interrupt Request

Q Byte	Condition Tested
0000 1 001	Not-ready/ready
DA   M   N	Interrupt request



This page intentionally left blank.

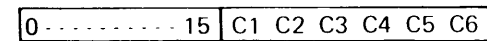
# MAIN STORAGE

The Model D provides the customer with a maximum of 512K bytes of main storage. Seven storage capacities are available (96, 128, 160, 192, 224, 384, and 512K). On the Models 15 D25 and 15 D26, the lower 256K (0K through 256K) of main storage is mounted on circuit board A-B4 and the upper 256K (384K through 512K) on circuit board A-A4.

The upper 256K of main storage contains its own timing, addressing, ECC (error check and correction) generation, fetch data register, ECC logic, and byte control circuitry. Each of these functions operates identically to, but independently of, the lower 256K.

MOSFET circuitry is used in this main storage. A read operation is nondestructive and, therefore, is not always followed by a write operation. A write cycle is required only when it is desired to change data at the addressed location.

Within the BSM (basic storage module), two bytes of data plus six check bits are accessed each storage cycle.



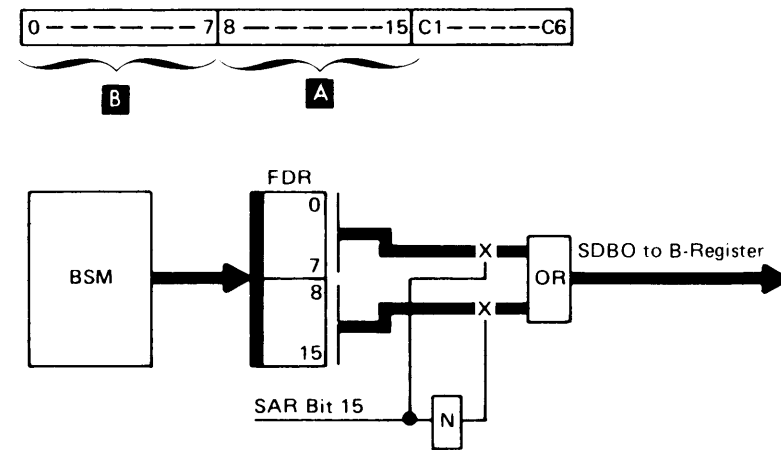
Thus, 96K of main storage is divided into 48K addressable storage positions. The 15th address bit (SAR 15) is not used to access these storage positions and the same two bytes are accessed if SAR 15 = 0 or SAR 15 = 1. The two byte storage position always begins at an even address and SAR 15 then selects one of the 2 bytes later on in the data path.

During a normal store/fetch operation, only one byte of the addressed storage position is used (the CPU data path can move only one byte at a time). SAR 15 is used to select one of the two bytes. A disk operation, however, uses both bytes; one is gated through the CPU data path, the other is gated to/from the extended channel. This dual byte operation allows the disks to transfer two bytes of data during one CPU (disk I/O) cycle. The disk must activate dual byte mode and the storage address must be an even number.

The six check bits allow the ECC circuits to detect and correct any single bit errors. Errors involving more than one bit are detected but not corrected (causing an SDBO processor check). ECC is described later in this section.

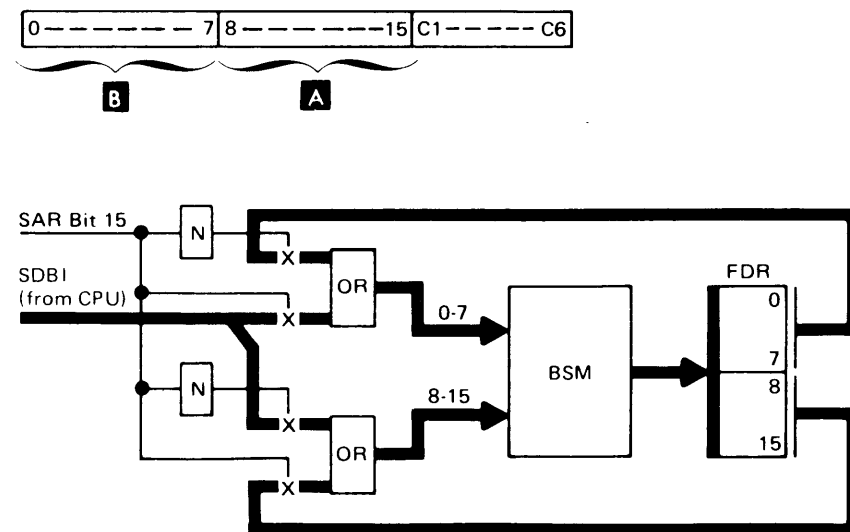
## One Byte Fetch

The BSM executes a fetch cycle to read out two bytes of data (plus 6 check bits). SAR bit 15 determines which byte is gated to the SDBO (store data bus out); if SAR bit 15 equals 0, SDBO bits 8-15 are gated **A**; if SAR 15 equals 1, SDBO bits 0-7 are gated to the B register **B**.



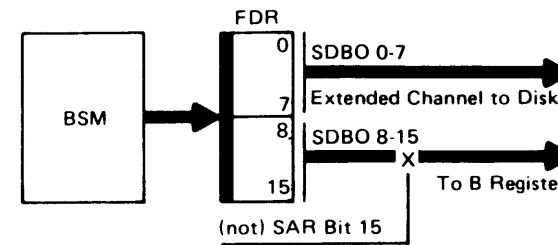
## One Byte Store

A fetch operation precedes each store operation. If SAR bit 15 equals 0, the contents of the SDBI (store data bus in) are gated to replace main storage bits 8-15 **A**. If SAR bit 15 equals 1, the contents of SDBI are gated to replace main storage bits 0-7 **B**. Because two bytes are written on one cycle, the other byte that was read from storage during the fetch operation is automatically restored.



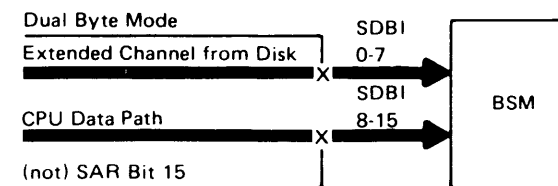
## Dual Byte Fetch

The BSM executes a fetch cycle to read out two bytes of data. These two bytes always start on an even address. The even address byte is gated to the CPU data path; the odd address byte is gated to the extended channel. To accomplish this two byte transfer, SAR bit 15 must be zero, and the disk must activate 'dual byte mode'.

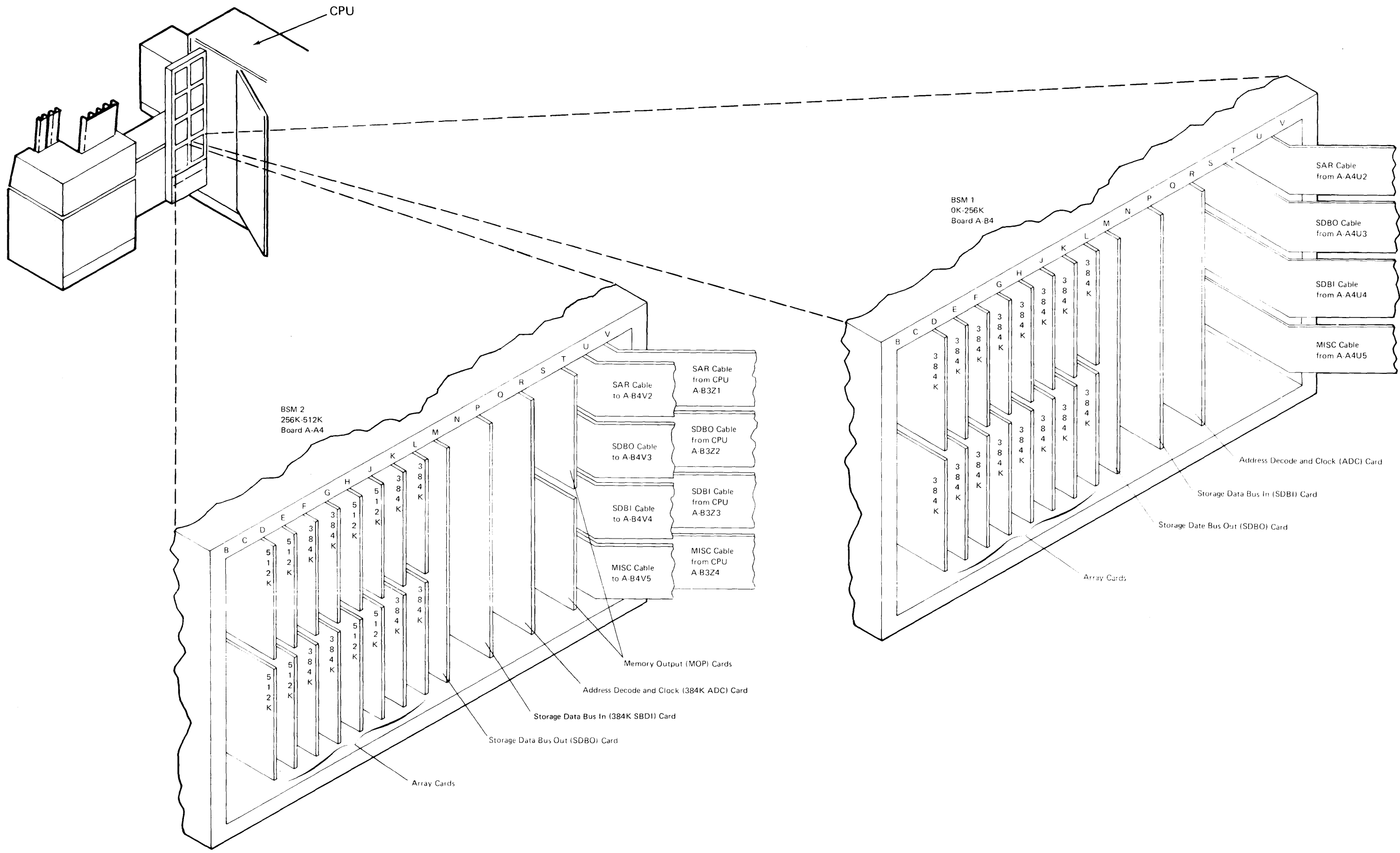


## Dual Byte Store

After the BSM executes a fetch cycle, the byte from the extended channel is stored as main storage bits 0-7 (odd address), and the byte from the CPU data path is stored as main storage bits 8-15 (even address). To accomplish this two byte write operation, SAR bit 15 must be zero and the disk must activate 'dual byte mode'.

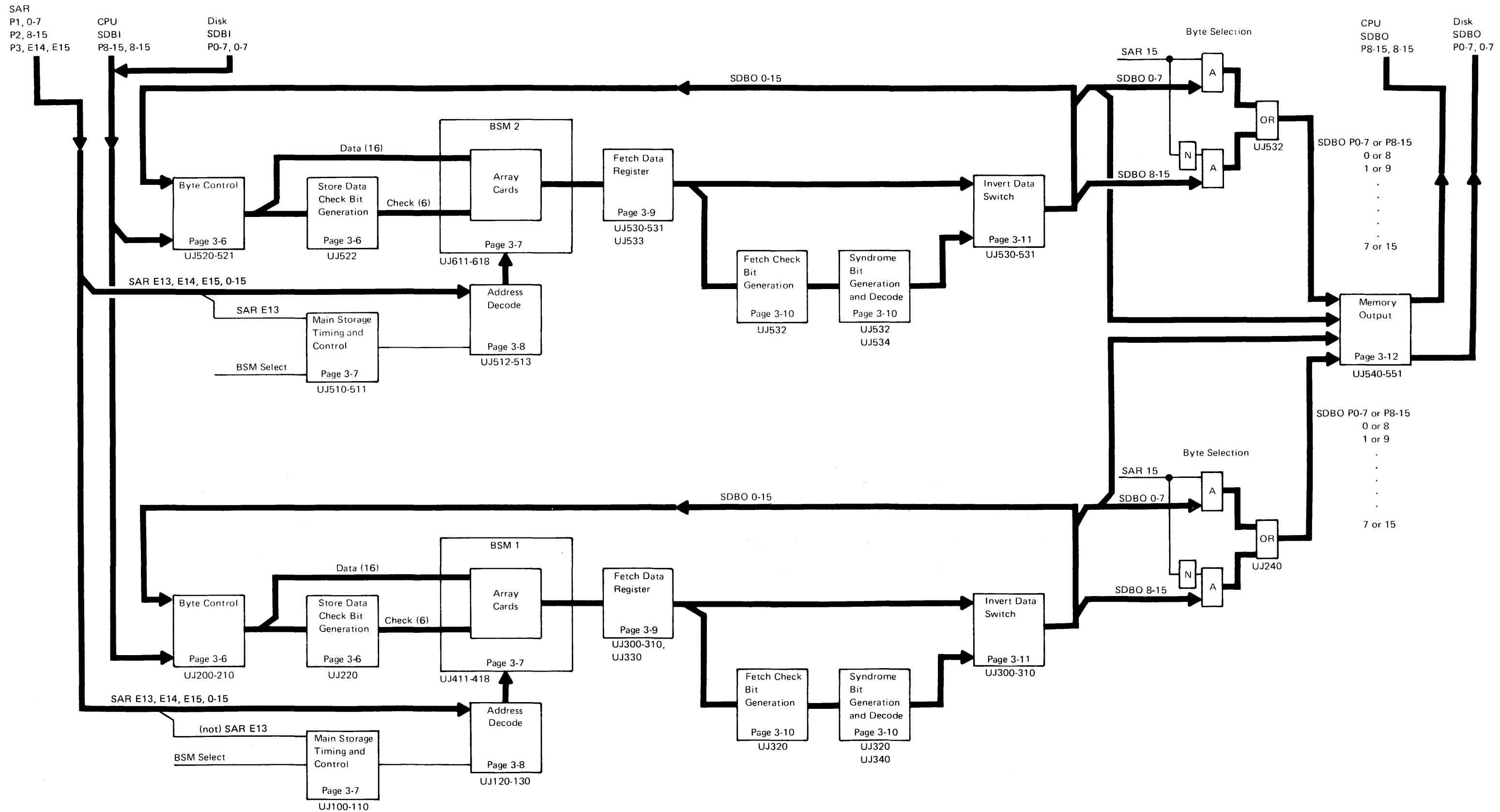


MAIN STORAGE LOCATIONS

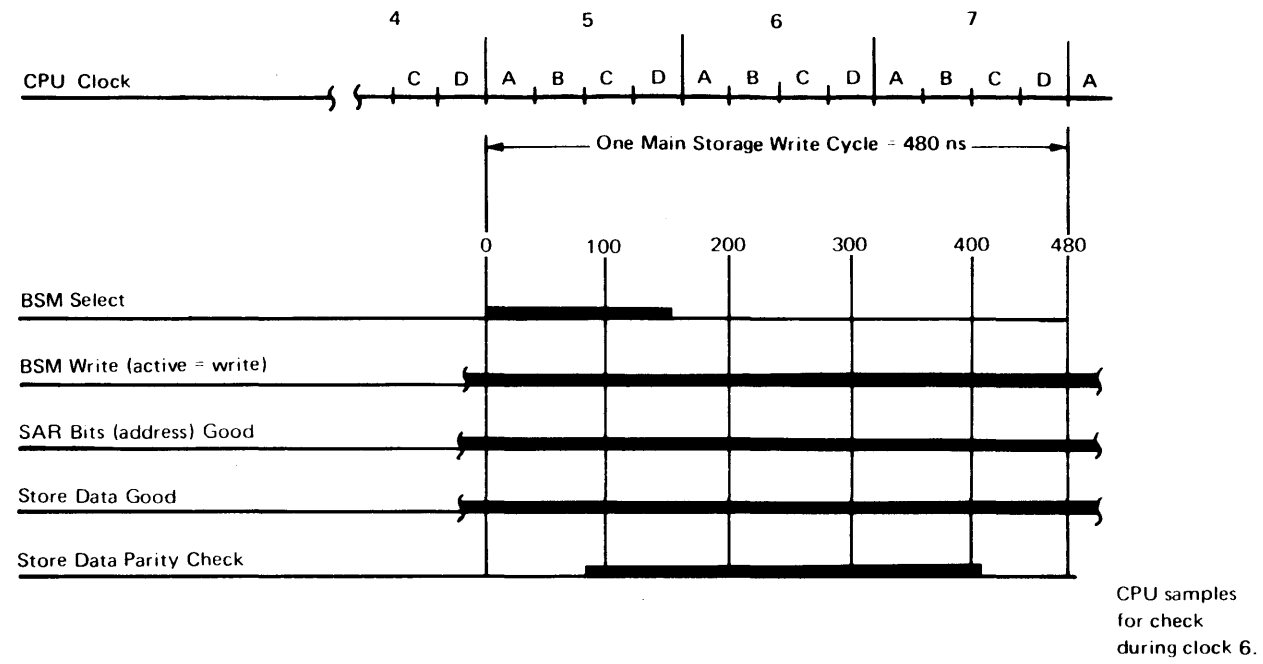
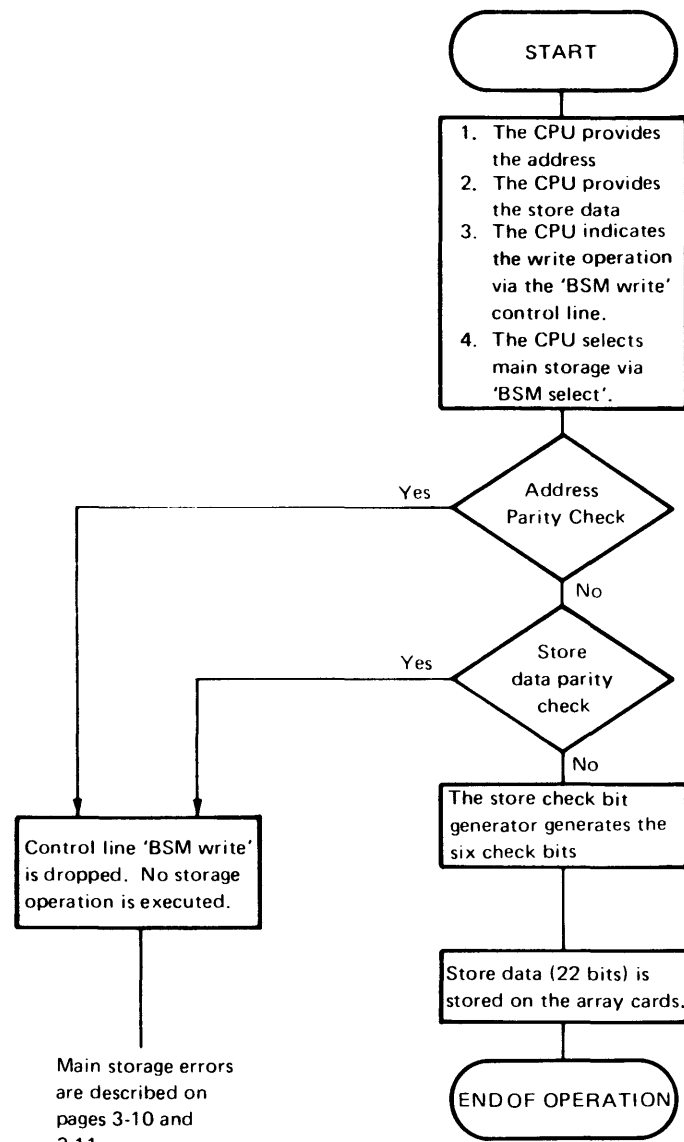




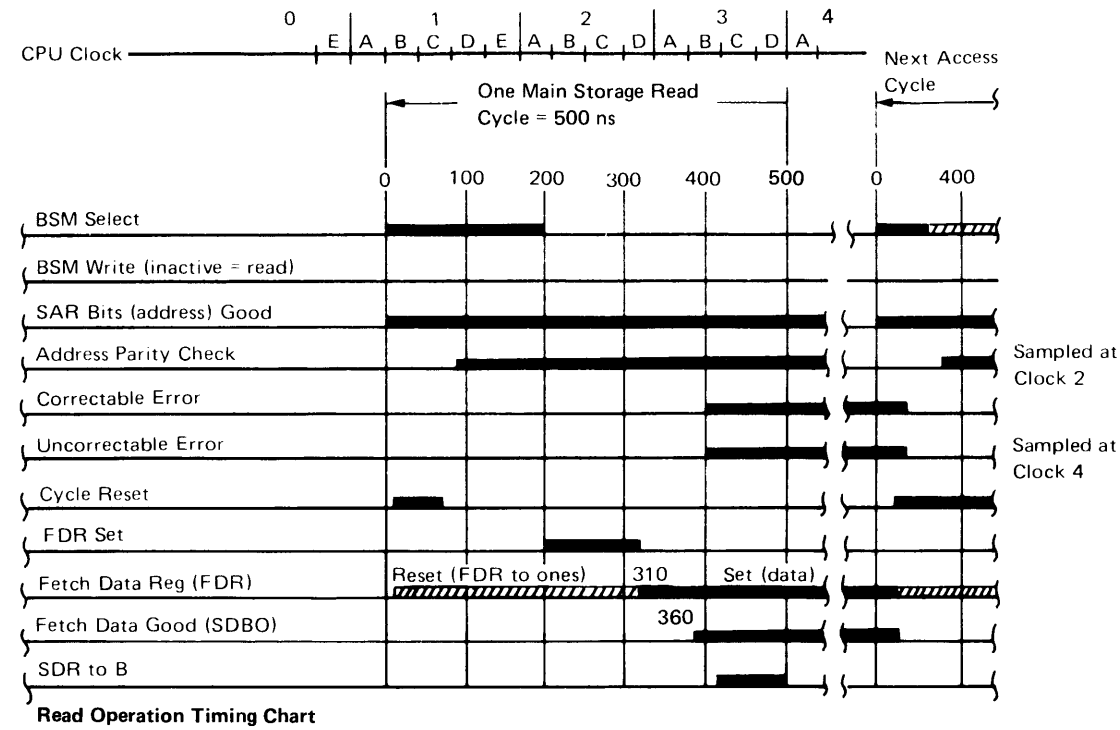
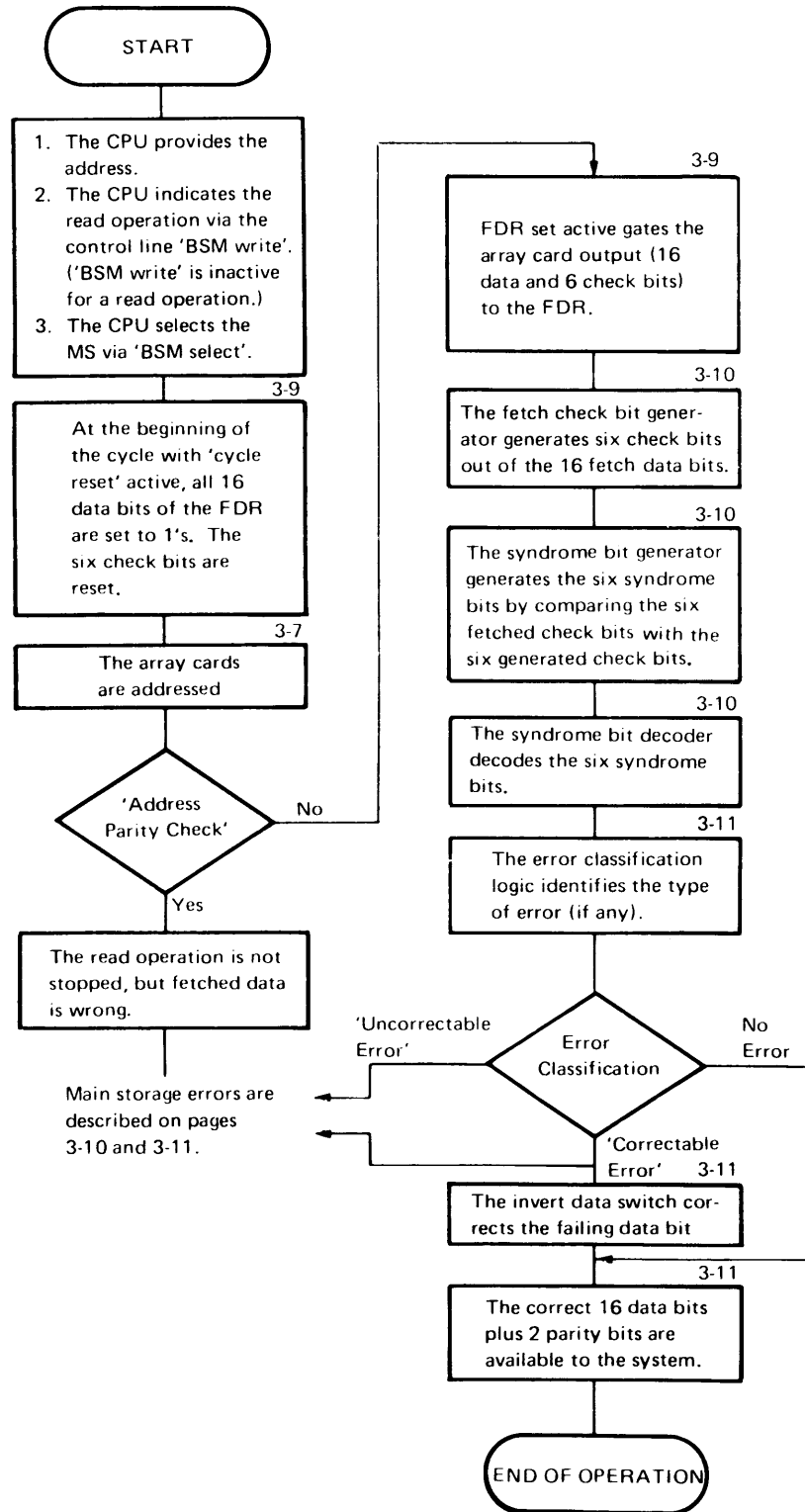
DATA FLOW



BSM WRITE OPERATION



# BSM READ OPERATION



Note: Diagnostic read operation (3) is covered by the LCP description on page 5-53.

## BYTE CONTROL AND CHECK BIT GENERATION

### Store Check Bit Generator

- Generates a unique group of check bits determined by the number of logical 1's in the 16 data bits to be stored.
- Each data bit is connected to three even blocks.
- Each even block is fed by eight data bit lines.
- An even block generates a check bit if an even number of eight input lines (unshaded squares) are active.
- Resulting 22 bits have even parity.

Example:

Storage Location	Data
0000	F2
0001	F1

Keep a count of the number of unshaded squares in the C-bit line that correspond to a 1 in the data line. When all of the unshaded squares for a particular C-bit line have been checked, determine whether the count is odd or even. If the count is even, that C bit is active; if odd, it is off. In this example, an even number (6) of C1 bits (unshaded squares) are active; therefore, C1 is active.

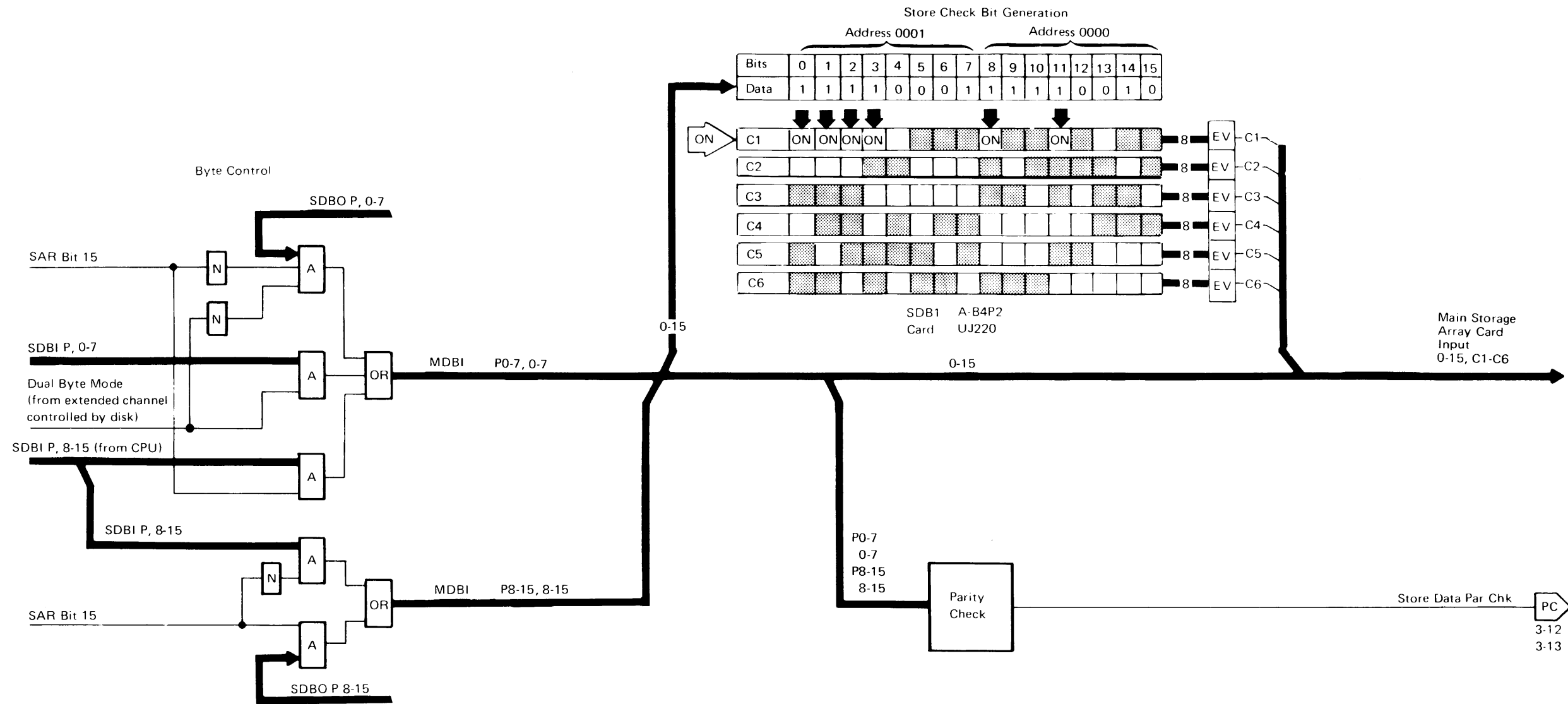
This check is repeated for each of the C-bit lines. Four of the six check bits are active (C1, C2, C4, and C6) in this example.

Ten data bits (in the example) plus the four check bits equal 14 bits active, or even parity.

### Store Data Parity Check

The store data parity check indicates wrong parity on the store data bus. It prevents the store operation.

This check causes an SDBI processor check.



Cards and ALD pages for byte control

Byte Control	ALD	Card (SDBI)
BSM 1	UJ200-210	A-B4P2
BSM 2	UJ520-521	A-A4P2

Cards and ALD pages for store check bit generation

Generation	ALD	Card (SDBI)
BSM 1	UJ220	A-B4P2
BSM 2	UJ522	A-A4P2

## BSM

### Controls

The BSM controls, which move data into and out of main storage, consist of the addressing scheme for locating words in the storage unit, read/write control signals, and a clock for timing operations within the storage cycle. To start a storage cycle, the CPU sends an address read (not BSM write) or write (BSM write), and a select pulse to the BSM. The cycle proceeds under control of the storage unit clock and data is gated into or out of the BSM.

### Addressing

Addressing is a method of locating information in the BSM. The main storage address coming from the CPU consists of 19 bits plus three parity bits **A**.

SAR bits E14, E15, and 0 through 14 are decoded **B** and provide the address of the selected double byte storage word. Since two bytes are read out of storage at a time, SAR bit 15 is not used for actual addressing, but determines which of the two bytes **C** is gated to the SDBO.

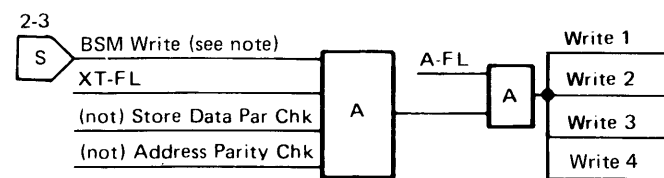
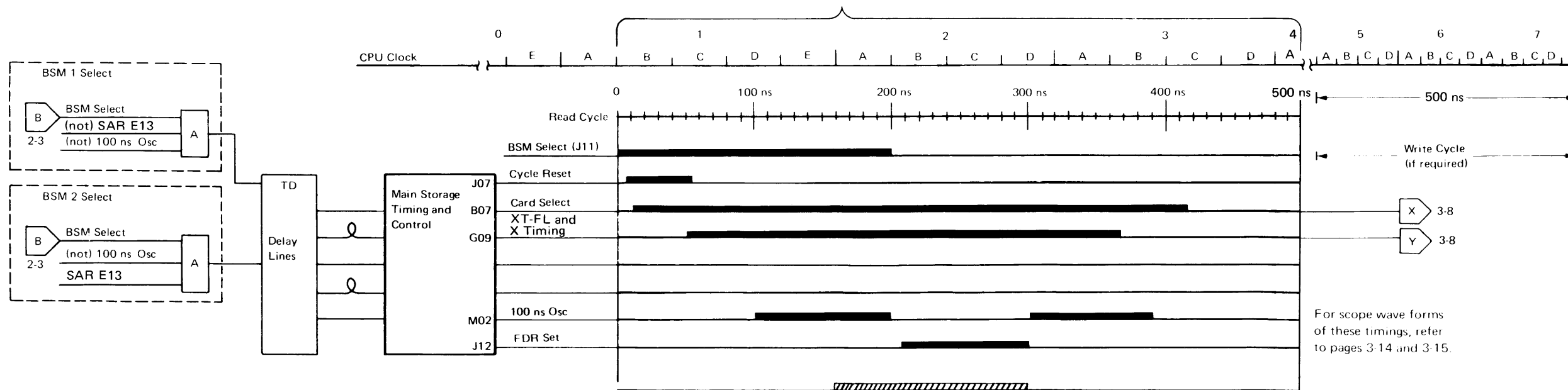
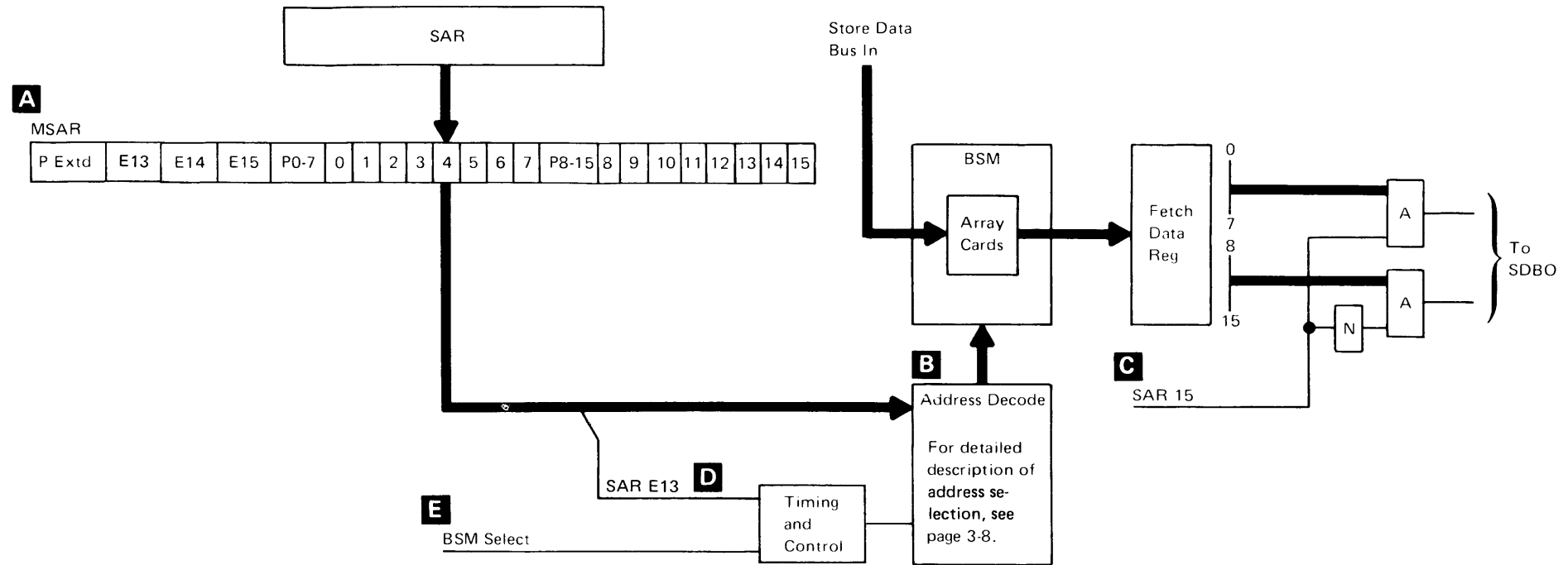
BSM 1 and BSM 2 selection and output gating (MOP) is controlled by SAR bit E13 (board select) **D**.

When SAR E13 is not active, BSM 1 is selected. When SAR E13 is active, then BSM 2 is selected and the write signal to the first board is degated. On a fetch operation the SAR bit E13 controls which output data (BSM 1 or BSM 2) is gated to the CPU. See page 3-12.

*Note:* BSM write is active on write operations and inactive on read operations.

## Timing

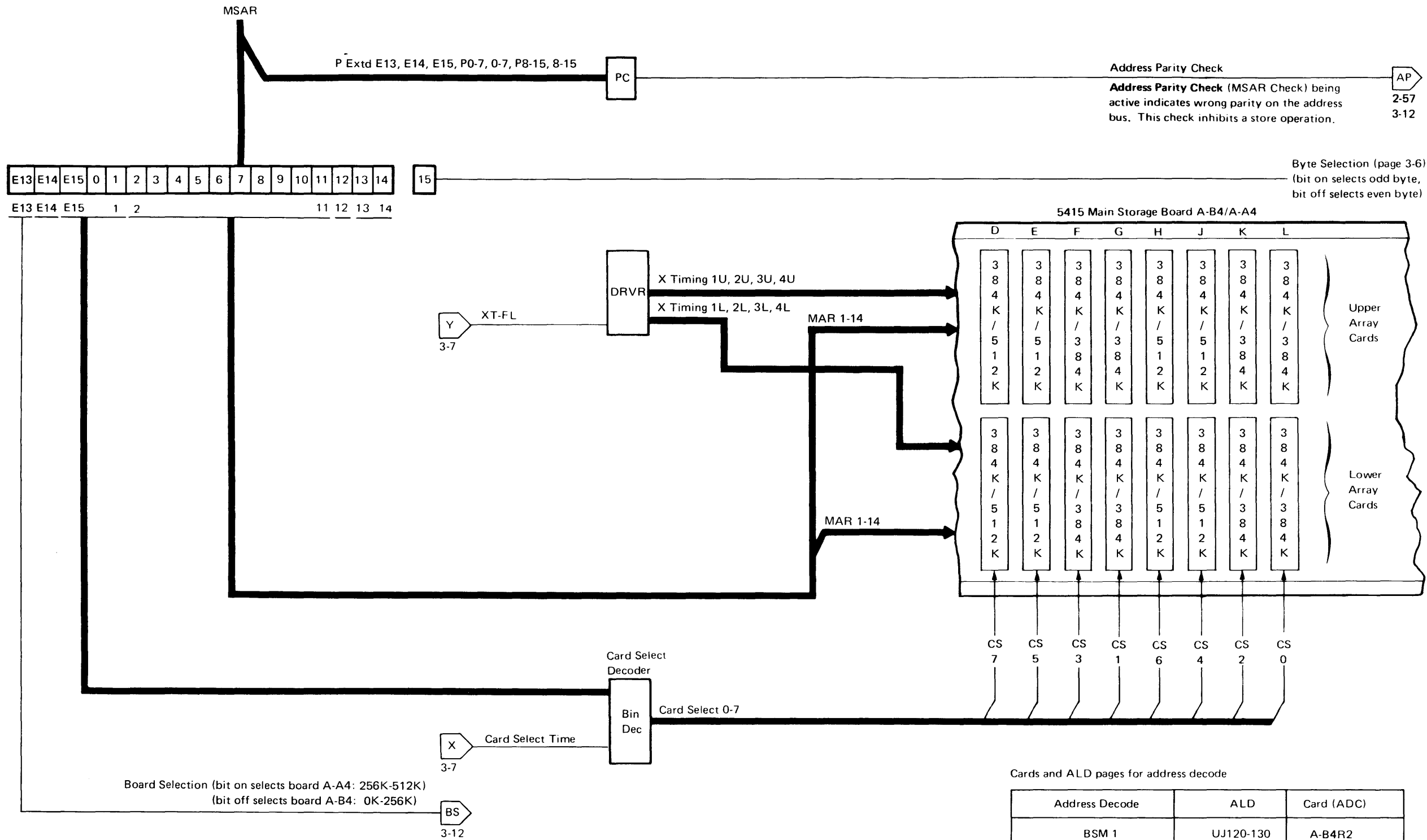
A delay line clock and timing latches make up the timing mechanism. The delay line clock is started by the BSM select pulse **E**. As the pulse travels through the delay line it sets and resets the timing latches.



Cards and ALD pages for main storage timing and control

Timing and Control	ALD	Card (ADC)
BSM 1	UJ100-110	A-B4R2
BSM 2	UJ510-511	A-A4R2

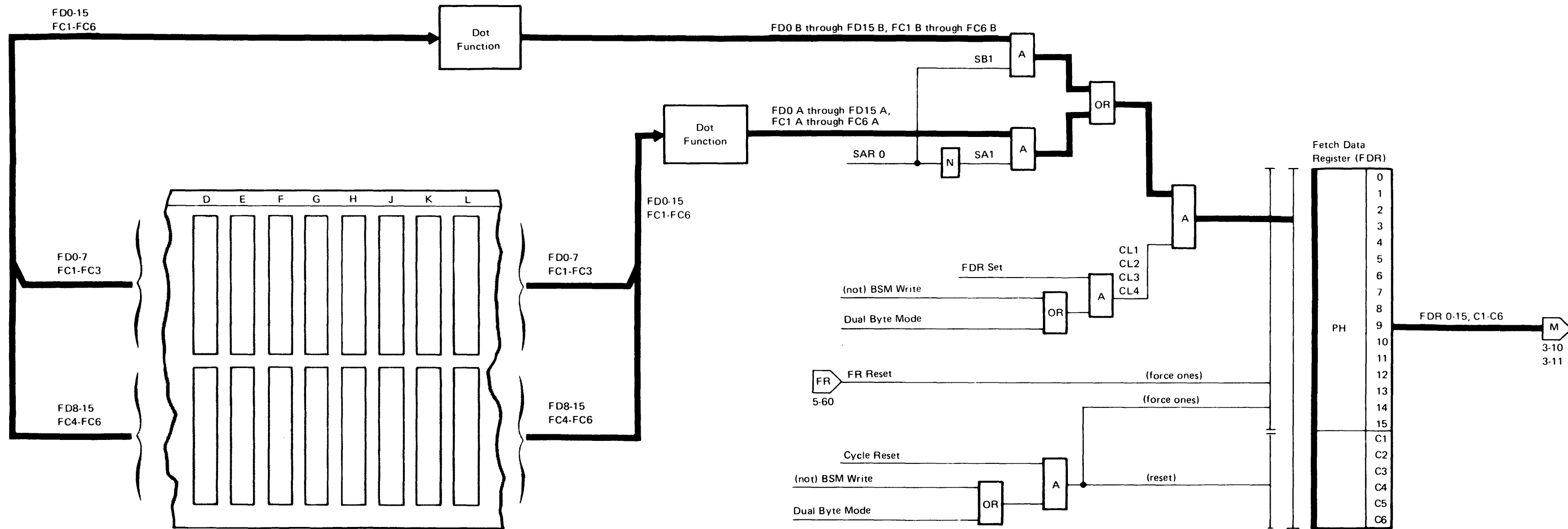
BSM



Cards and ALD pages for address decode

Address Decode	ALD	Card (ADC)
BSM 1	UJ120-130	A-B4R2
BSM 2	UJ512-513	A-A4R2

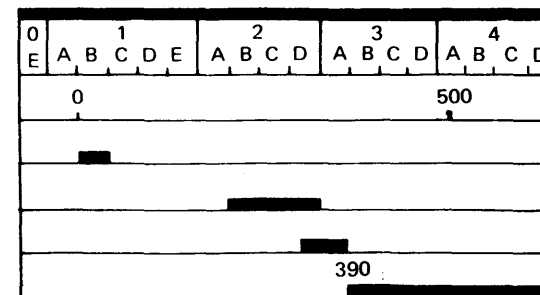
# FETCH DATA REGISTER AND CONTROL



Cards and ALD pages for main storage

Main Storage	ALD	Cards (array)
BSM 1	UJ411-418	See page 3-2 for array card locations.
BSM 2	UJ611-618	See page 3-2 for array card locations.

CPU Time  
 BSM Time  
 Gates Reset (reset to ones)  
 FDR Set (data)  
 ECC Time  
 Fetch Data Good on SDBO



Cards and ALD pages for fetch data register (FDR)

FDR	ALD	Card (SDBO)
BSM 1	UJ300-310, UJ330	A-B4M2
BSM 2	UJ530-531, UJ533	A-A4M2

**ERROR CHECKING AND CORRECTION**

The data transferred to main storage during a write operation consists of 16 data bits plus the corresponding two parity bits. The store check bit generator generates six check bits to add to the data bits. If parity is correct, the two parity bits are dropped at this point. The 16 data bits plus the six generated check bits are then stored as a 22 bit word on the array cards.

The 22 bits coming from the BSM during a read operation are stored in the fetch data register (FDR). If there is an error in the 22 bits, the ECC logic detects and corrects (single bit errors) the error and indicates the type of error to the system. Using the 16 correct data bits, two parity bits are generated (one for each byte). These two bytes of data are then available to the system.

**Fetch Check Bit Generator A**

The fetch check bit generator generates six check bits (C1-C6) from the 16 fetch data bits coming from the FDR. These checks are generated in the same manner as in the store check bit generator.

**Syndrome Bit Generator B**

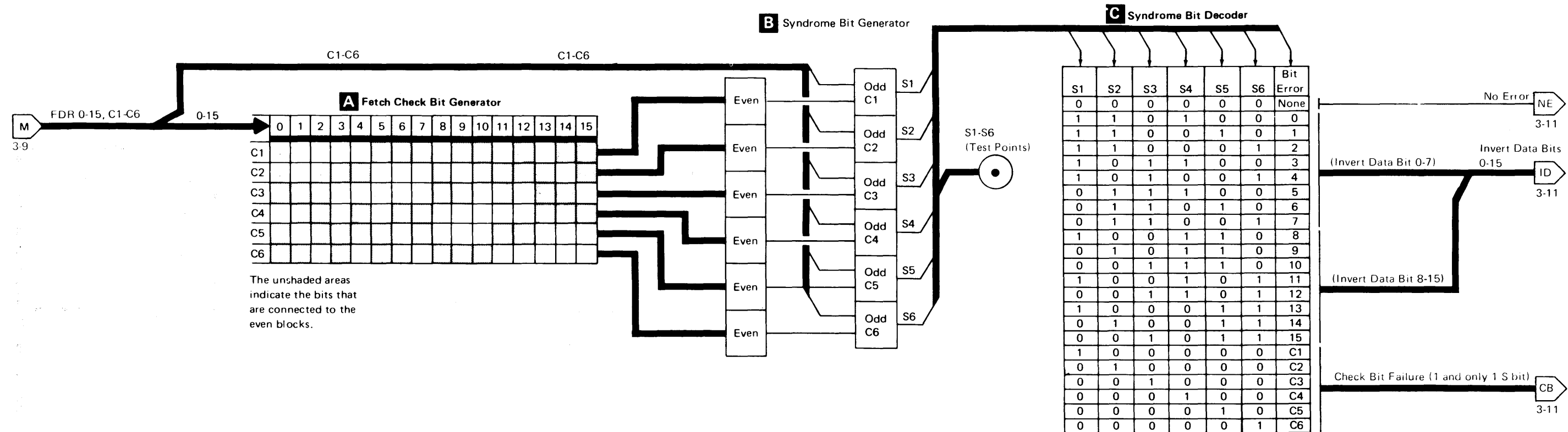
The syndrome bit generator generates six syndrome bits by comparing, bit by bit, the six generated check bits (C1-C6) with the six check bits from storage. If all check bits compare, all six syndrome bits are zero. Each unequal compare generates a one for the syndrome bit.

For diagnostic purposes, the six syndrome bits are connected to test points on the SDBO card.

**Syndrome Bit Decoder C**

This circuit decodes the six syndrome bits and generates the 'invert data' lines that identify the failing data bit in case of a single bit error. It also identifies multiple errors or no errors.

Only one syndrome bit decoder output line is active at a time.



Cards and ALD pages for fetch check and syndrome bit generators

Generators	ALD	Card (SDBO)
BSM 1	UJ320	A-B4M2
BSM 2	UJ532	A-A4M2

Cards and ALD pages for syndrome bit decoder

Decoder	ALD	Card (SDBO)
BSM 1	UJ340	A-B4M2
BSM 2	UJ534	A-A4M2



## ERROR CHECKING AND CORRECTION

### Invert Data Switch **A**

The invert data switch corrects a single bit error by inverting the failing bit. It also corrects the parity of the byte in which the failing bit is corrected.

### Error Classification Logic **B**

This circuit identifies the type of error and generates the error lines 'correctable error' and 'uncorrectable error.'

The number of active syndrome bits relates directly to the kind of error. (See syndrome bit decoder on page 3-10.)

Syndrome Bits	Type of Error
all 6 bits = 0	No error
3 bits = 0 3 bits = 1	Single data bit error
5 bits = 0 1 bit = 1	Single check bit error
All other bit combinations	Multiple bit error

The error classification logic also generates an 'invert parity' line for each single data bit error.

#### Correctable Error

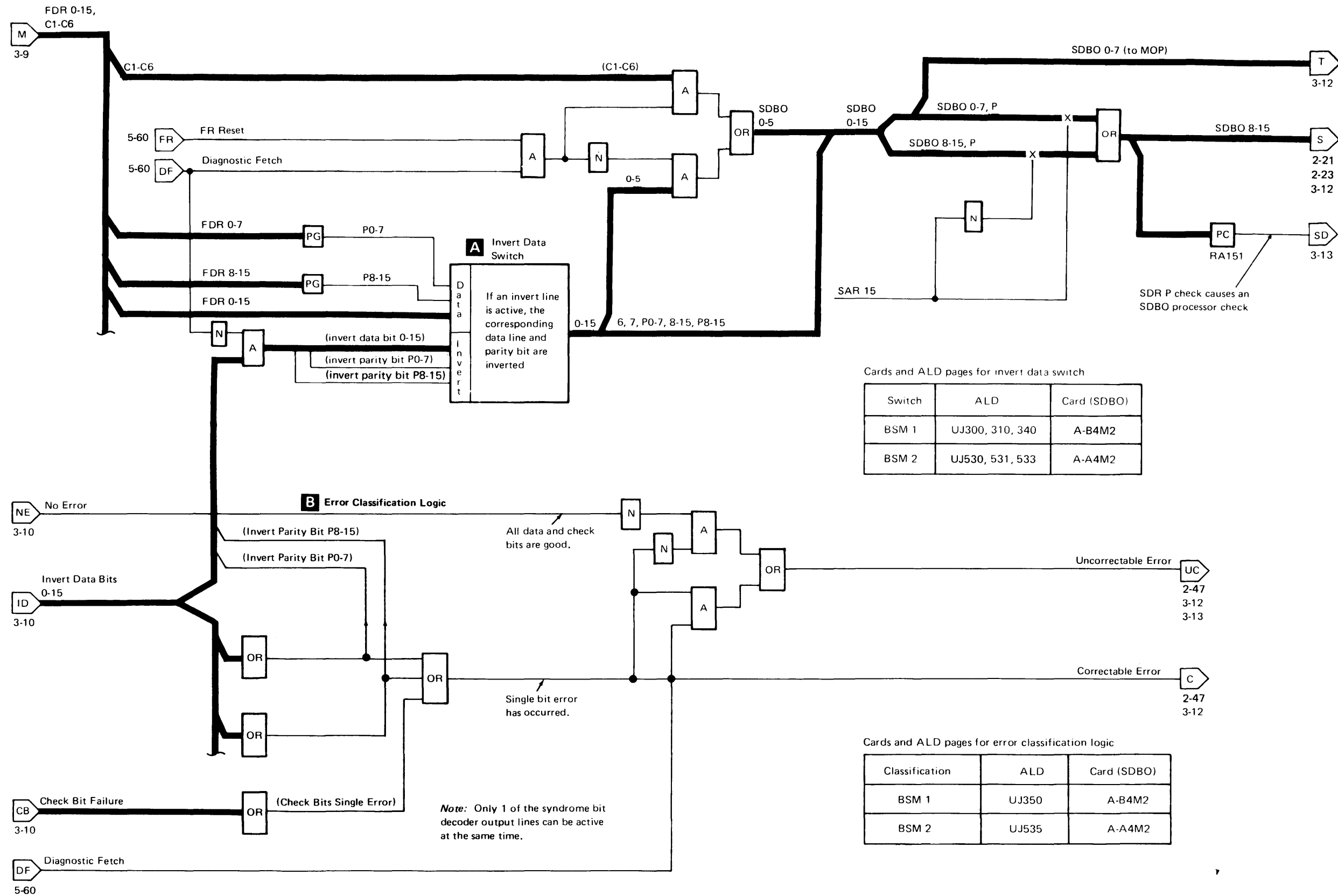
The correctable error signal indicates a correctable error has occurred during a fetch operation (a single bit error has occurred).

#### Uncorrectable Error

When a multiple bit error occurs, the syndrome bit combination does not allow an output from the decoder. Main storage circuits cannot correct this type of error. The result is an uncorrectable error indication and an SDBO processor check.

#### Diagnostic Mode

This line inhibits error correction of the fetched data. The current contents of the fetch data register are available at the SDBO bus.



Cards and ALD pages for invert data switch

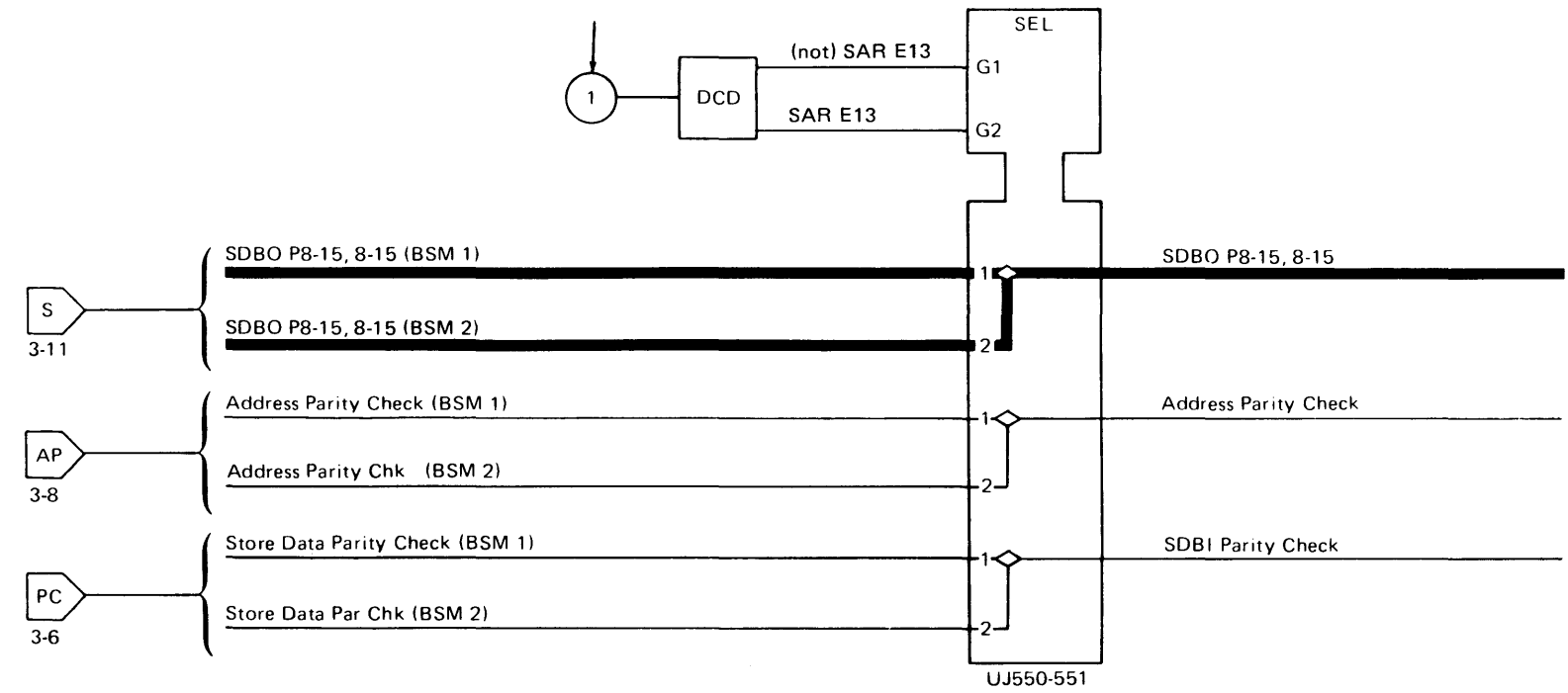
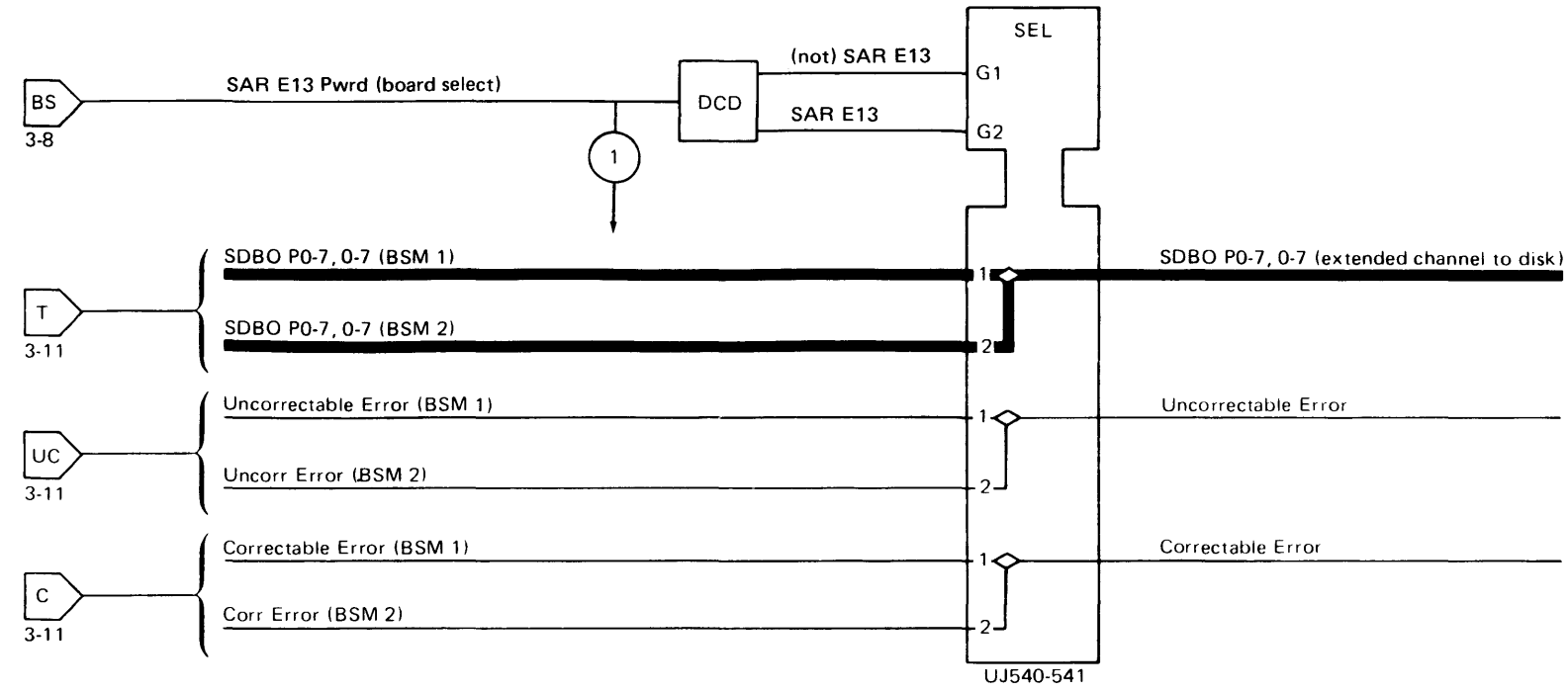
Switch	ALD	Card (SDBO)
BSM 1	UJ300, 310, 340	A-B4M2
BSM 2	UJ530, 531, 533	A-A4M2

Cards and ALD pages for error classification logic

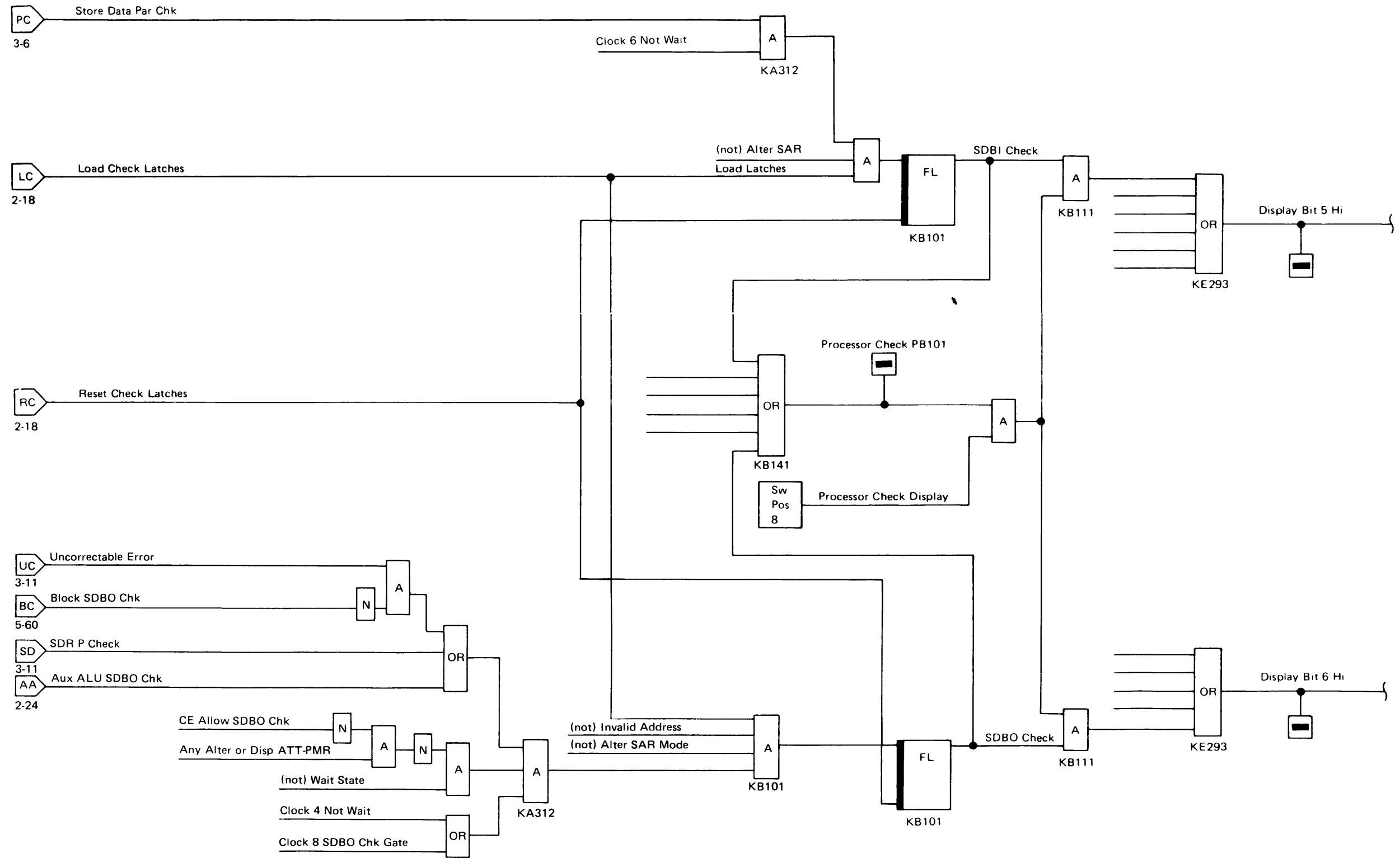
Classification	ALD	Card (SDBO)
BSM 1	UJ350	A-B4M2
BSM 2	UJ535	A-A4M2

**MEMORY OUTPUT (MOP)**

On a fetch operation, the fetched data (SDBO bits) enters the memory output circuitry where SAR bit E13 controls which BSM output is gated to the CPU. The memory output cards select the active bus only to repower the bus lines to the CPU.



SDBI/SDBO PROCESSOR CHECKS



### MAIN STORAGE MAINTENANCE

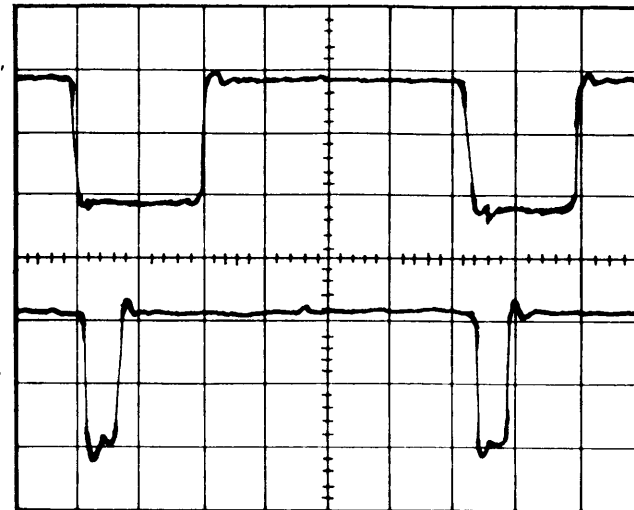
The following scope wave forms are provided to help you service the 5415 main storage unit.

Operation: Alter storage — Loop all positions of main storage (description is on page 6-4).

Tektronix\*\*  
 453 Scope: Sync Ext Minus +Clk 0  
 A-B3T2S09 (KC122)  
 Ch 1 and Ch 2 50 mV/div  
 Time Base 0.1 μs/div  
 All Probes X10

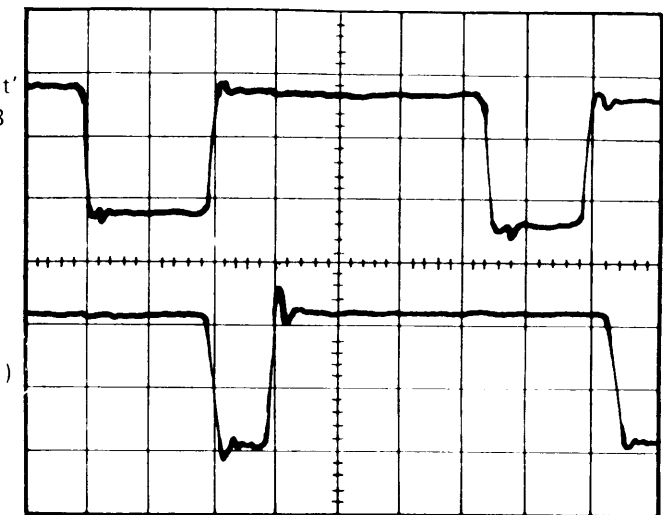
Note: Ground probes at panel to avoid excessive ringing.

'-BSM Select'  
 A-B4V5D13 (UJ012)



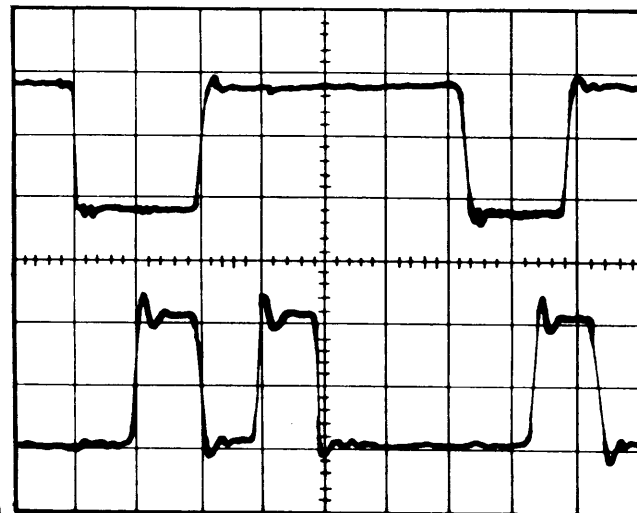
'-Cycle Reset'  
 BSM 1 - A-B4R2J07 (UJ110)  
 BSM 2 - A-A4R2J07\* (UJ511)

'-BSM Select'  
 A-B4V5D13 (UJ012)



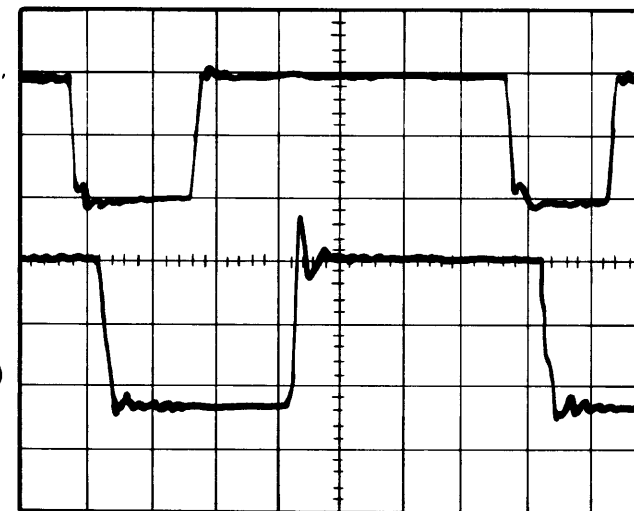
'-FDR Set'  
 BSM 1 - A-B4R2J12 (UJ110)  
 BSM 2 - A-A4R2J12\* (UJ511)

'-BSM Select'  
 A-B4V5D13 (UJ012)



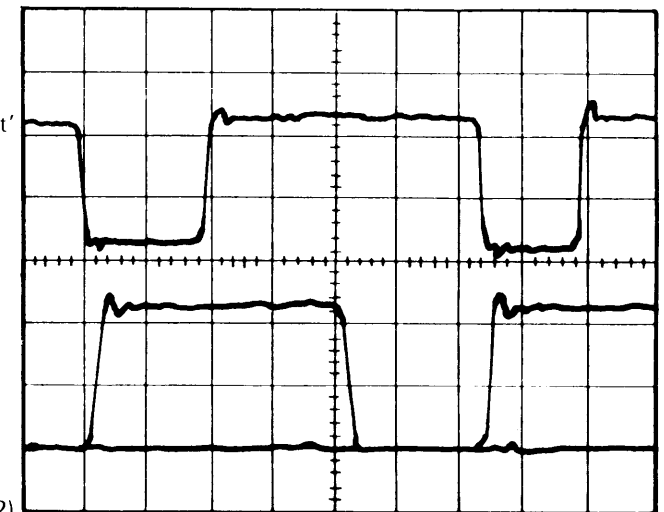
'+100 ns Osc'  
 BSM 1 - A-B4R2M02 (UJ100)  
 BSM 2 - A-A4R2M02\* (UJ510)

'-BSM Select'  
 A-B4V5D13 (UJ012)



'-X Timing'  
 BSM 1 - A-B4R2G11 (UJ120)  
 BSM 2 - A-A4R2G11\* (UJ512)

'-BSM Select'  
 A-B4V5D13 (UJ012)

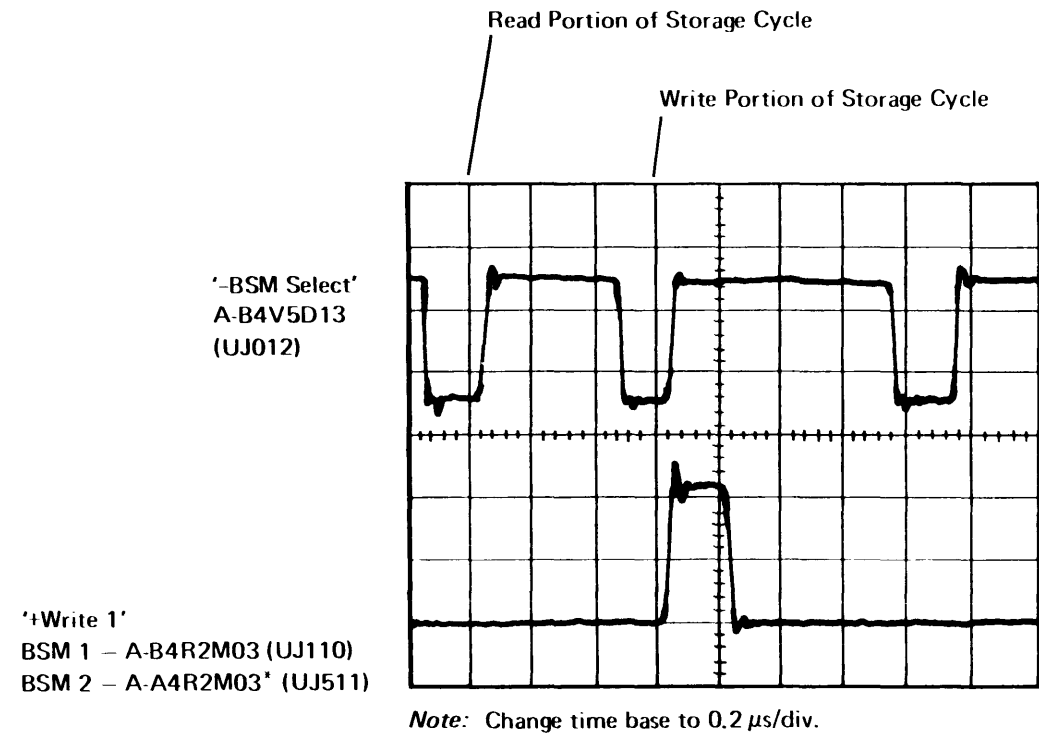


'+Card Select 0'  
 BSM 1 - A-B4R2B07 (UJ120)  
 BSM 2 - A-A4R2B07\* (UJ512)

\*Use this test point when altering upper storage (> 256K).

\*\*Trademark of Tektronix, Incorporated.

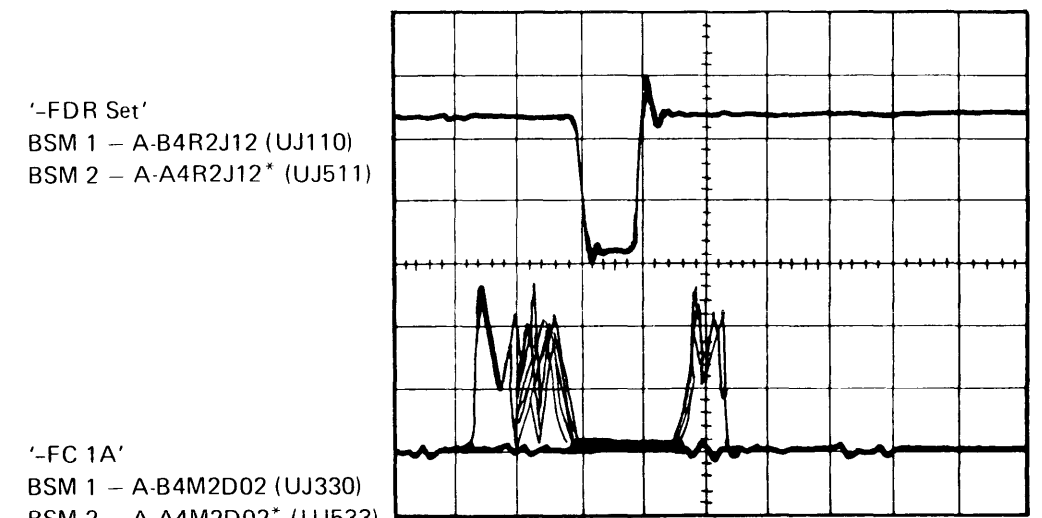
**MAIN STORAGE MAINTENANCE**



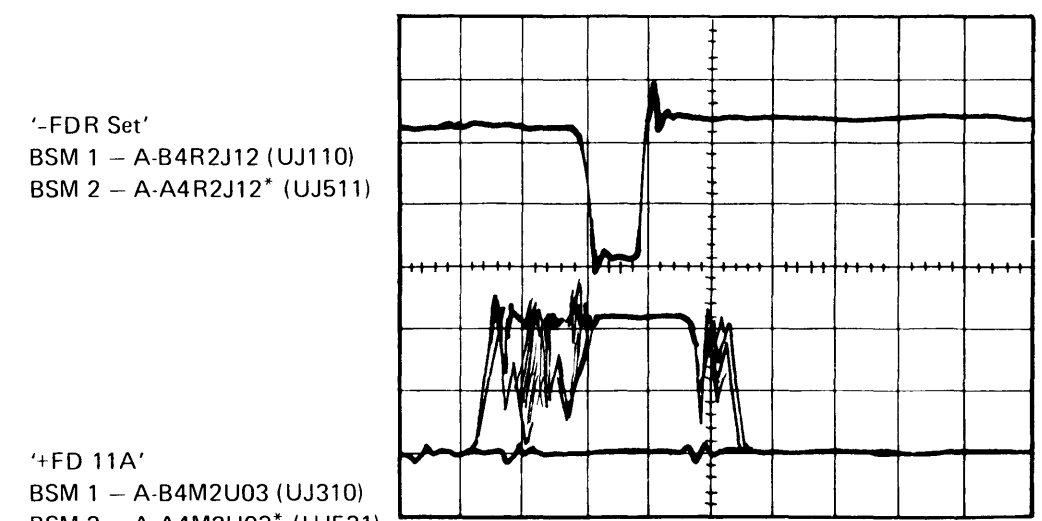
Operation: Display Storage - Loop all positions of main storage data in storage = Hex 'FF' (1111 1111) (description is on page 6-8)

Tektronix  
453 Scope: Sync Ext Minus '+Clk 0'  
A-B3T2S09  
(KC122)  
Ch 1 and Ch 2 50 mV/div  
Time Base 0.1  $\mu$ s/div

*Note:* Ground probe at panel to avoid excessive ringing.



*Note:* Lower trace could be any of the six check bits.



*Note:* Lower trace could be any of the 16 data bits.

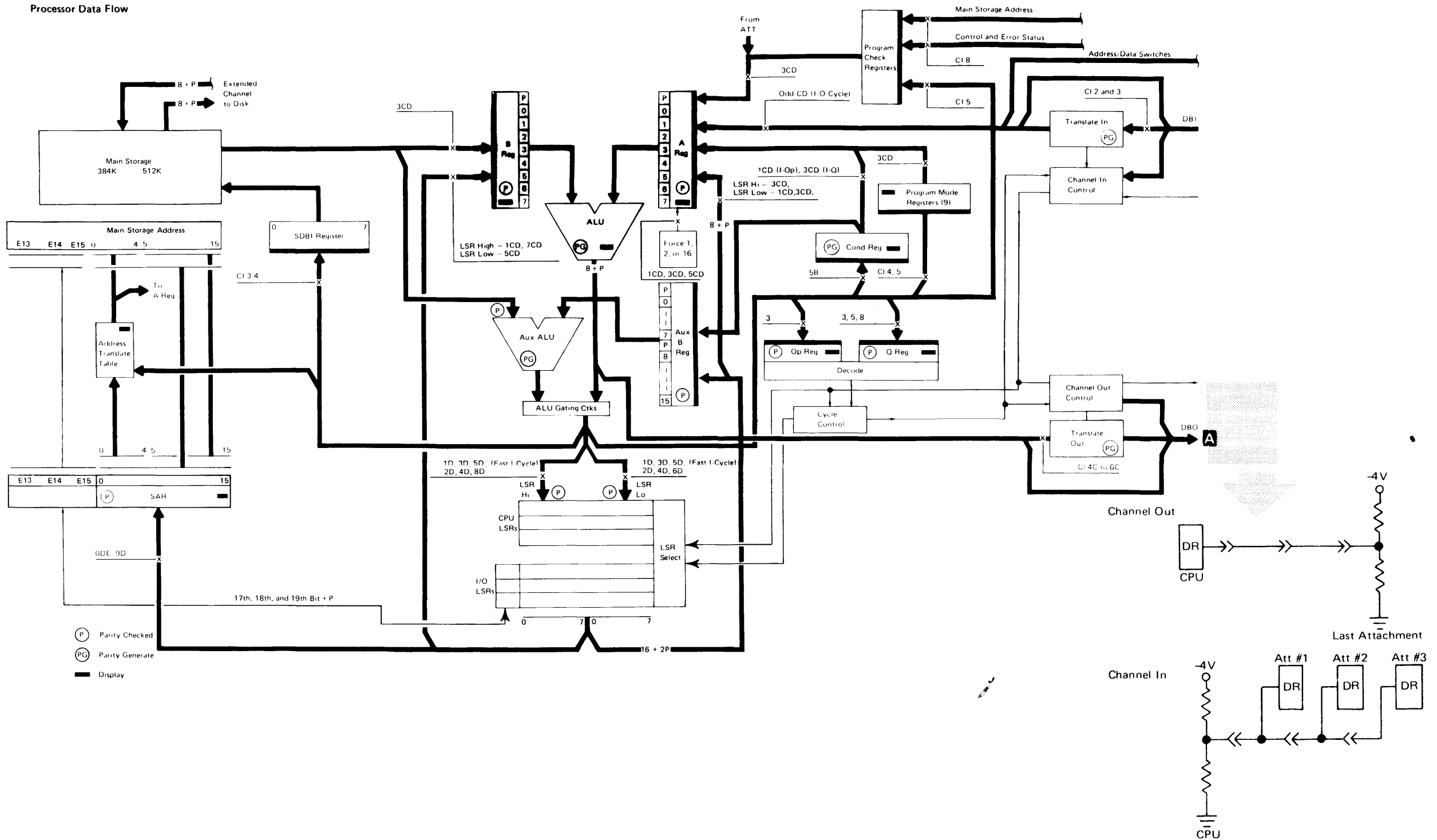
\*Use this test point when displaying upper storage (>256K).

This page intentionally left blank.

# I/O INTERFACE

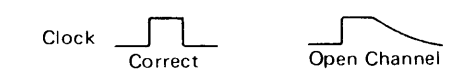
All I/O devices communicate with the CPU over the I/O channel. This attachment interface consists of several major functions:

1. DBI, DBO drivers and terminators **A**.
2. Extended DBI, DBO for disk only **B**.
3. DBI and DBO translators **C** (description is on page 4-095).
4. Cycle steal hardware **D** (description is on page 5-72).
5. Input lines for console switches **E** (description is on page 4-110).



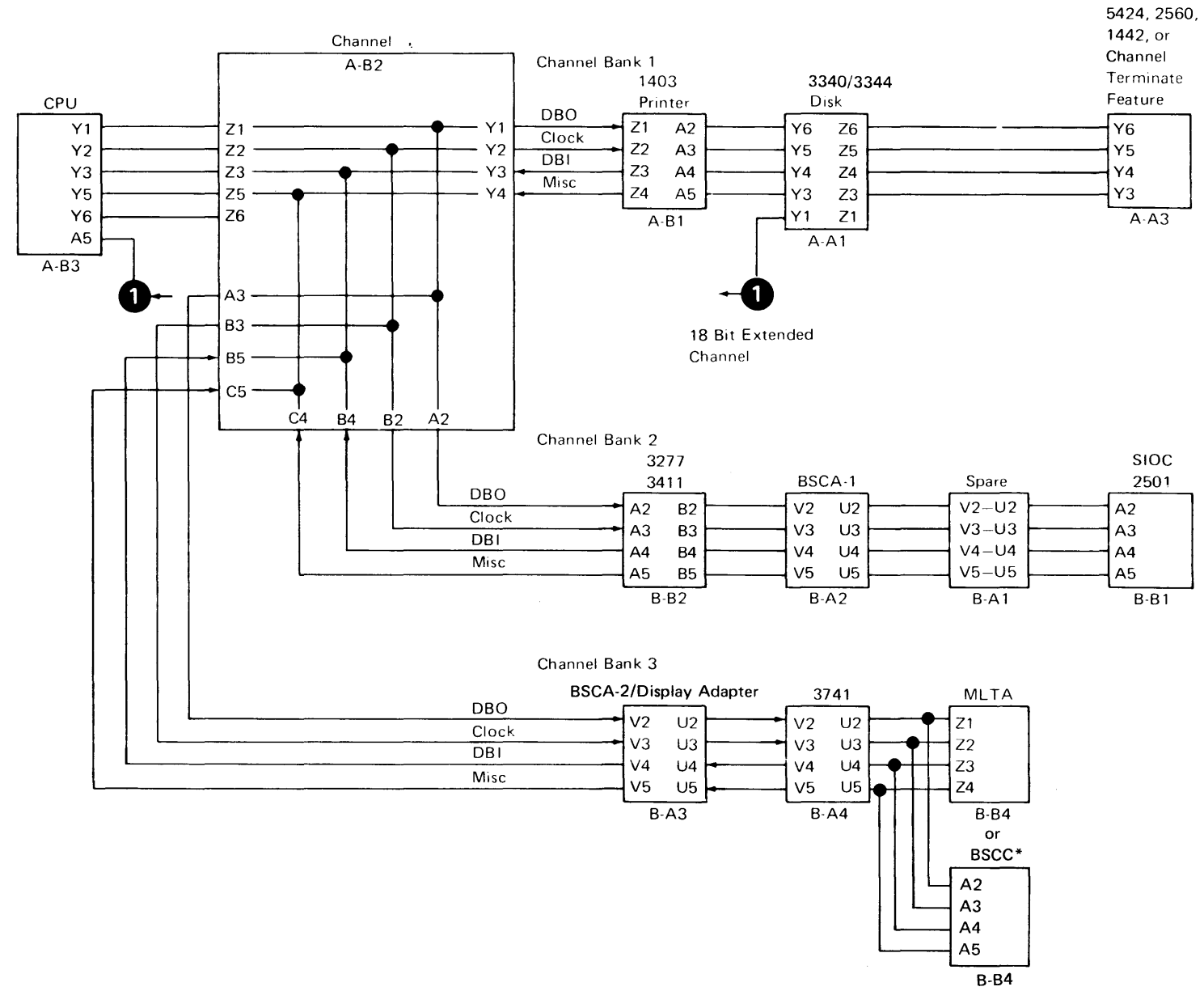
*Note:* An open in channel wiring between driver and voltage divider can cause incorrect wave shape.

Example:



### CHANNEL CABLING (CPU)

This diagram shows the channel cable connections within the CPU. If any feature board is not installed, then the cables run between the existing boards. If, for instance, the A1 board on B gate is not installed, the cables would run from the B-A2 board to the B-B1 board.

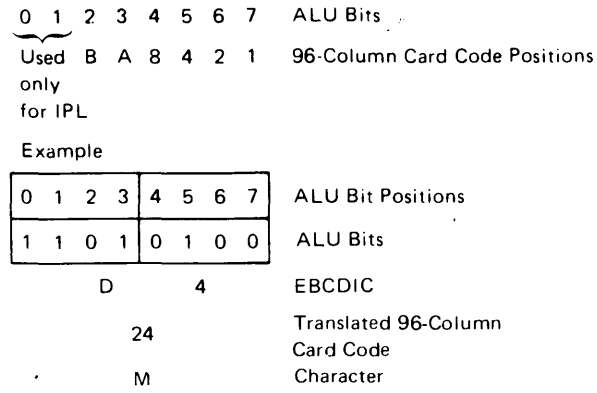


\*MLTA and BSCC are mutually exclusive features (cannot have both features on same machine).



### DBO TRANSLATOR (5424)

The DBO translator is used during clock 4 and 5 time to translate ALU data (EBCDIC) to 96-column card code. The DBO translator is not used during every I/O cycle and if the 'translate out' line is inactive, the data is transferred to the I/O attachment unchanged. The figure below shows a 'translate out' conversion table.

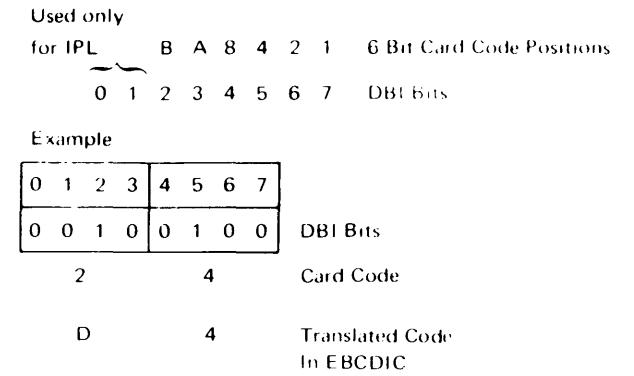


See page 4-107 for DBO and translator circuits.

ALU BITS 0,1,2,3 4,5,6,7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	40	5A	60	D0	00 SPACE	1A &	20 -	90	C0	70	E0	50	80	30 }	A0	10 0
1	F1	E1	51	C1	B1	A1	11 /	81	71	61	D1	41	31 A	21 J	91	01 1
2	F2	E2	D2	C2	B2	A2	92	82	72	62	52	42	32 B	22 K	12 S	02 2
3	F3	E3	D3	C3	B3	A3	93	83	73	63	53	43	33 C	23 L	13 T	03 3
4	F4	E4	D4	C4	B4	A4	94	84	74	64	54	44	34 D	24 M	14 U	04 4
5	F5	E5	D5	C5	B5	A5	95	85	75	65	55	45	35 E	25 N	15 V	05 5
6	F6	E6	D6	C6	B6	A6	96	86	76	66	56	46	36 F	26 O	16 W	06 6
7	F7	E7	D7	C7	B7	A7	97	87	77	67	57	47	37 G	27 P	17 X	07 7
8	F8	E8	D8	C8	B8	A8	98	88	78	68	58	48	38 H	28 Q	18 Y	08 8
9	F9	E9	D9	C9	B9	A9	99	89	79	69	59	49	39 I	29 R	19 Z	09 9
A	7A	6A	FO	4A	3A c	2A !	BO :	0A	FA	EA	DA	CA	BA	AA	9A	8A
B	7B	6B	5B	4B	3B .	2B \$	1B #	0B #	FB	EB	DB	CB	BB	AB	9B	8B
C	7C	6C	5C	4C	3C <	2C *	1C %	0C @	FC	EC	DC	CC	BC	AC	9C	8C
D	7D	6D	5D	4D	3D (	2D )	1D -	0D ,	FD	ED	DD	CD	BD	AD	9D	8D
E	7E	6E	5E	4E	3E +	2E ;	1E >	0E =	FE	EE	DE	CE	BE	AE	9E	8E
F	7F	6F	5F	4F	3F 	2F	1F ?	0F "	FF	EF	DF	CF	BF	AF	9F	8F

### DBI TRANSLATOR (5424)

The DBI translator is used during clock 2 and 3 time to translate the 96-column card code into EBCDIC. The translator is not used in every I/O cycle and if the 'translate in' line is inactive, the I/O data is transferred to the A register unchanged. The figure below shows a 'translate in' conversion table.

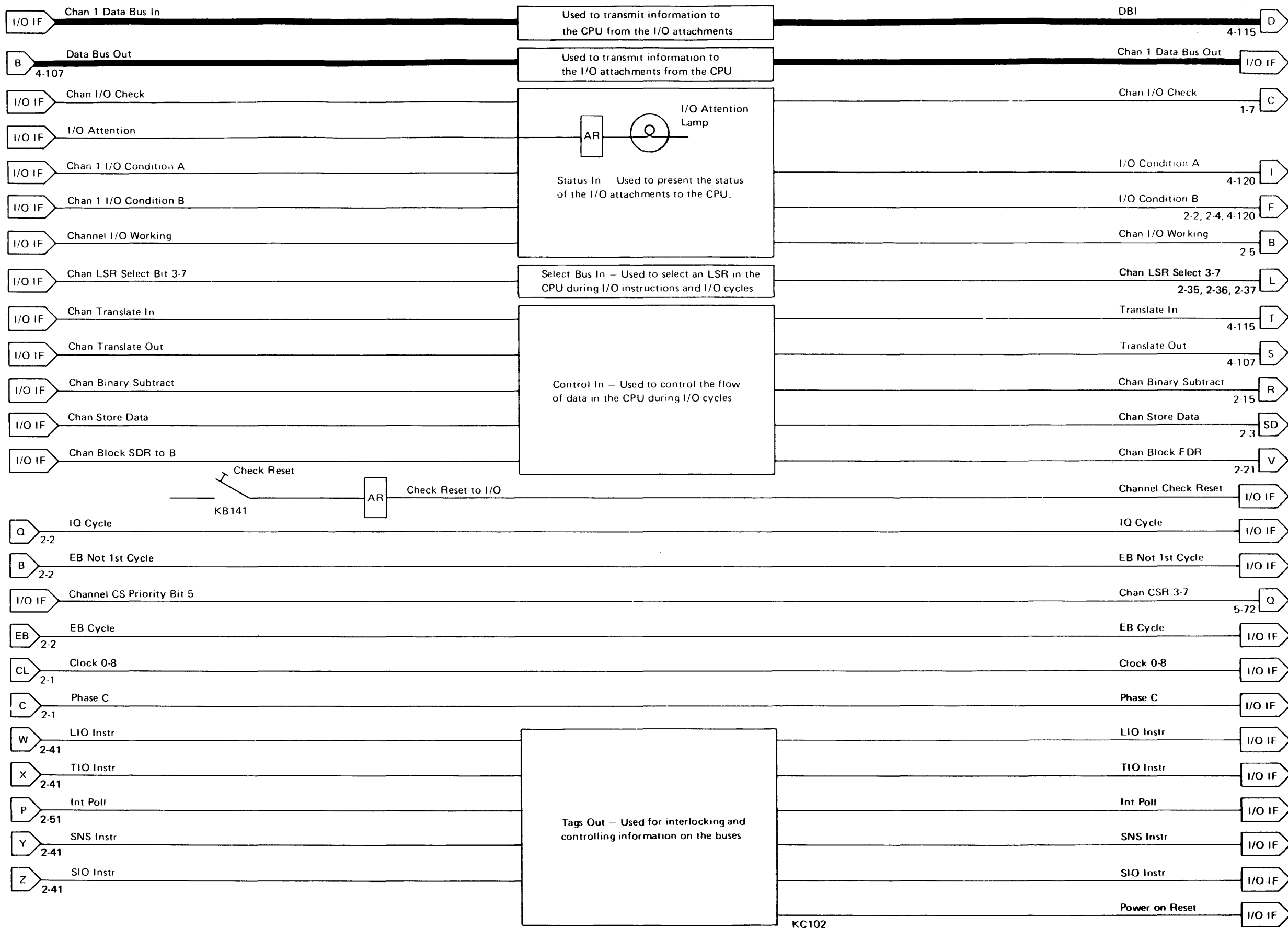


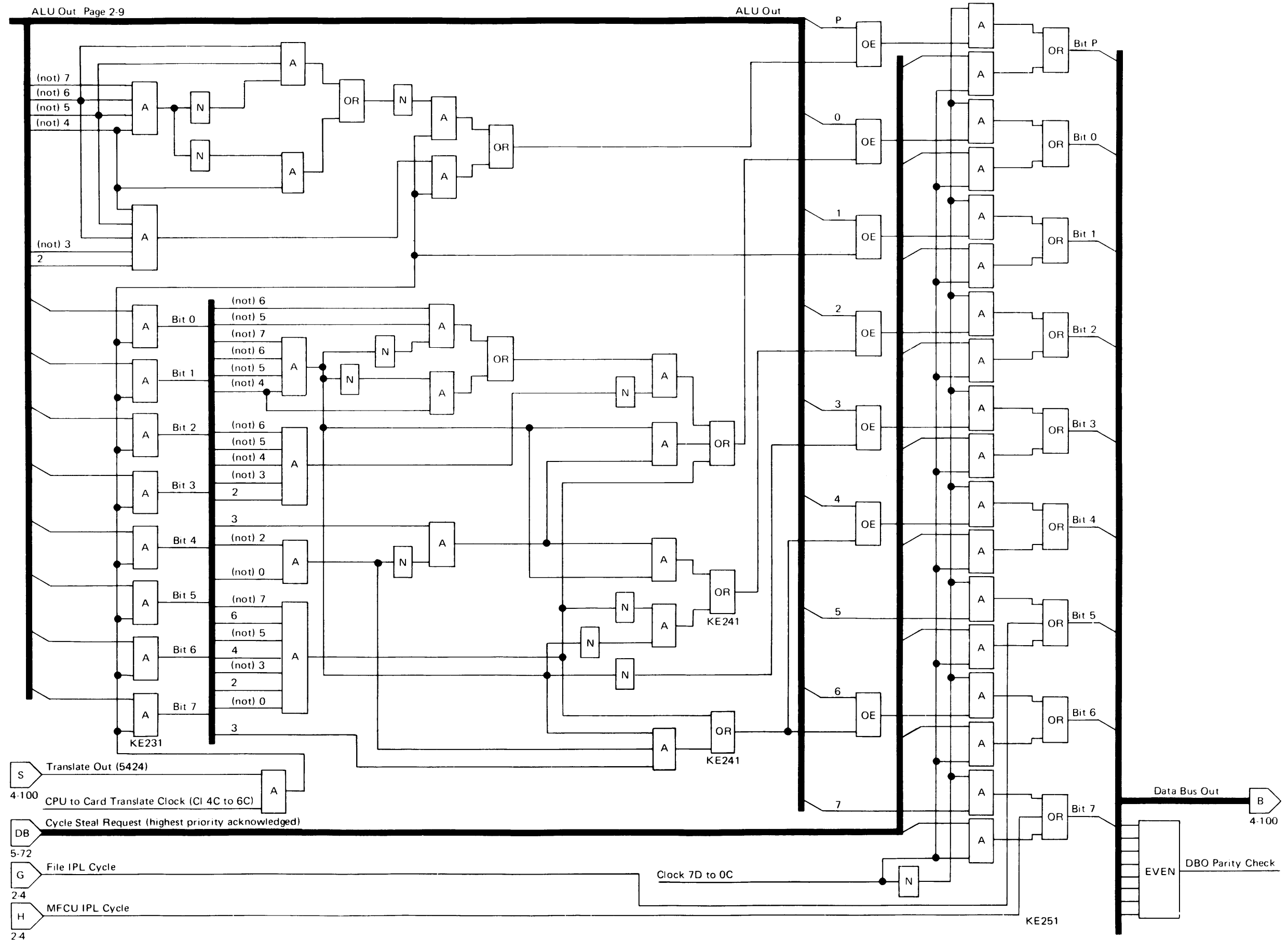
See page 4-115 for DBI and translator circuits.

DBI BITS 0,1,2,3 4,5,6,7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	40 SPACE	F0 φ	60	D0	00	B0	20	90	C0	70	E0	6A	80	30	A0	2A
1	F1 1	61 /	D1 J	C1 A	B1	21	91	81	71	E1	51	41	31	A1	11	01
2	F2 2	E2 S	D2 K	C2 B	B2	A2	92	82	72	62	52	42	32	22	12	02
3	F3 3	E3 T	D3 L	C3 C	B3	A3	93	83	73	63	53	43	33	23	13	03
4	F4 4	E4 U	D4 M	C4 D	B4	A4	94	84	74	64	54	44	34	24	14	04
5	F5 5	E5 V	D5 N	C5 E	B5	A5	95	85	75	65	55	45	35	25	15	05
6	F6 6	E6 W	D6 O	C6 F	B6	A6	96	86	76	66	56	46	36	26	16	06
7	F7 7	E7 X	D7 P	C7 G	B7	A7	97	87	77	67	57	47	37	27	17	07
8	F8 8	E8 Y	D8 Q	C8 H	B8	A8	98	88	78	68	58	48	38	28	18	08
9	F9 9	E9 Z	D9 R	C9 I	B9	A9	99	89	79	69	59	49	39	29	19	09
A	7A :	6A &	5A !	4A c	3A	2A	1A	0A	FA	EA	DA	CA	BA	AA	9A	8A
B	7B #	6B .	5B \$	4B .	3B	2B	1B	0B	FB	EB	DB	CB	BB	AB	9B	8B
C	7C (	6C %	5C *	4C <	3C	2C	1C	0C	FC	EC	DC	CC	BC	AC	9C	8C
D	7D ,	6D )	5D -	4D (	3D	2D	1D	0D	FD	ED	DD	CD	BD	AD	9D	8D
E	7E	6E >	5E ;	4E +	3E	2E	1E	0E	FE	EE	DE	CE	BE	AE	9E	8E
F	7F "	6F ?	5F 	4F 	3F	2F	1F	0F	FF	EF	DF	CF	BF	AF	9F	8F

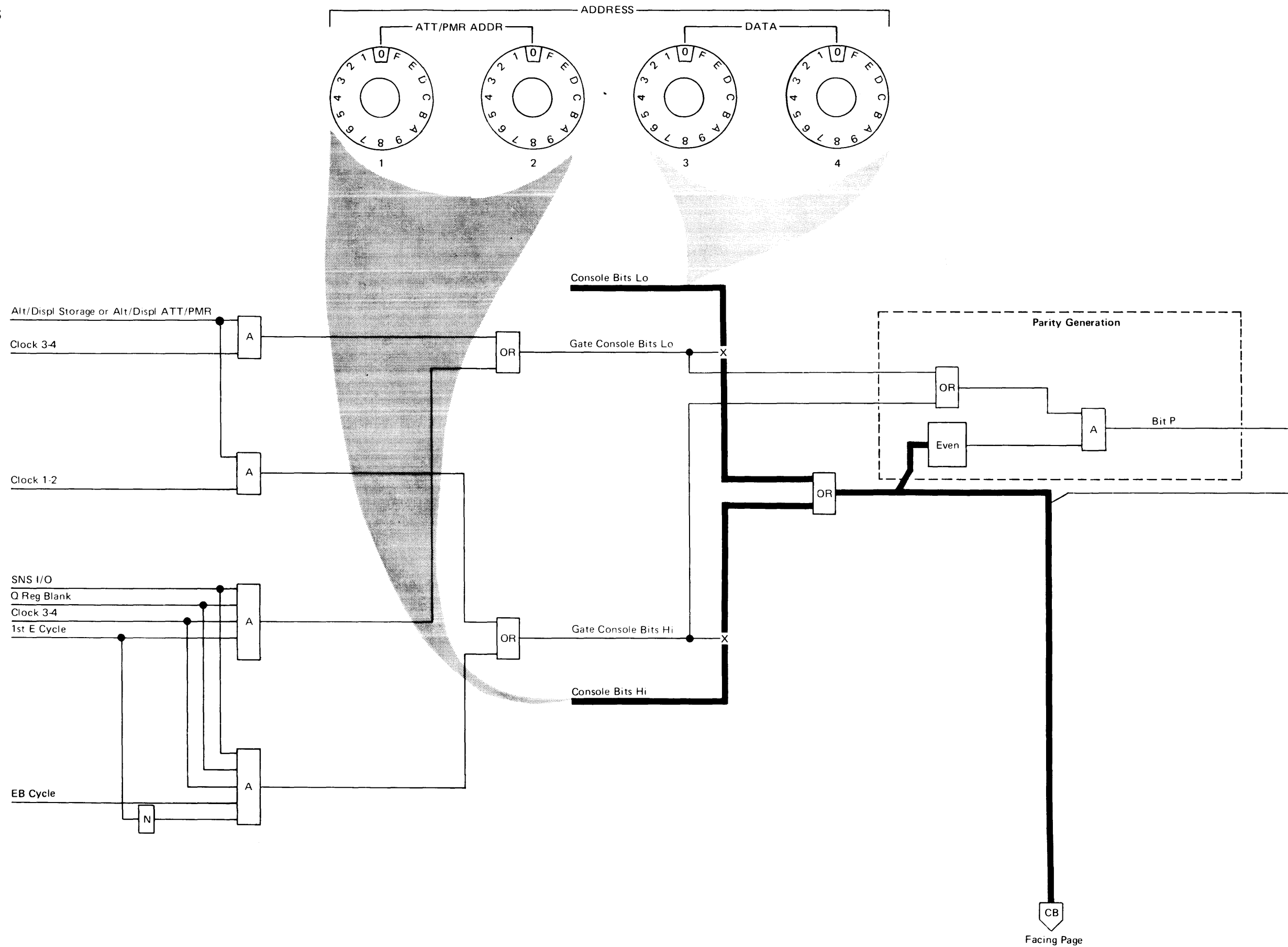
I/O INTERFACE LINES

This page provides a CPU entry point from I/O attachment manuals. Only lines shown to exit those manuals are found here.

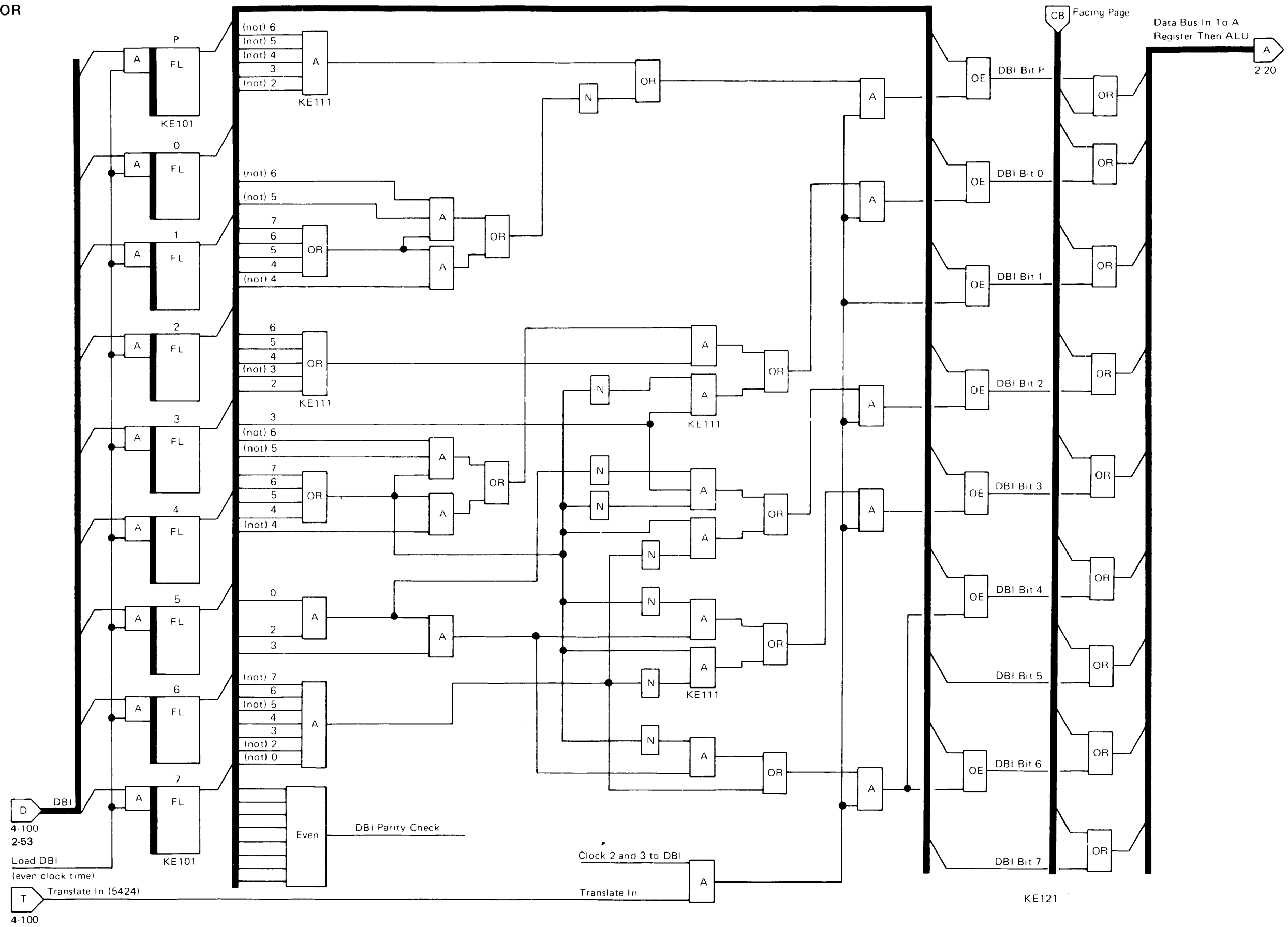




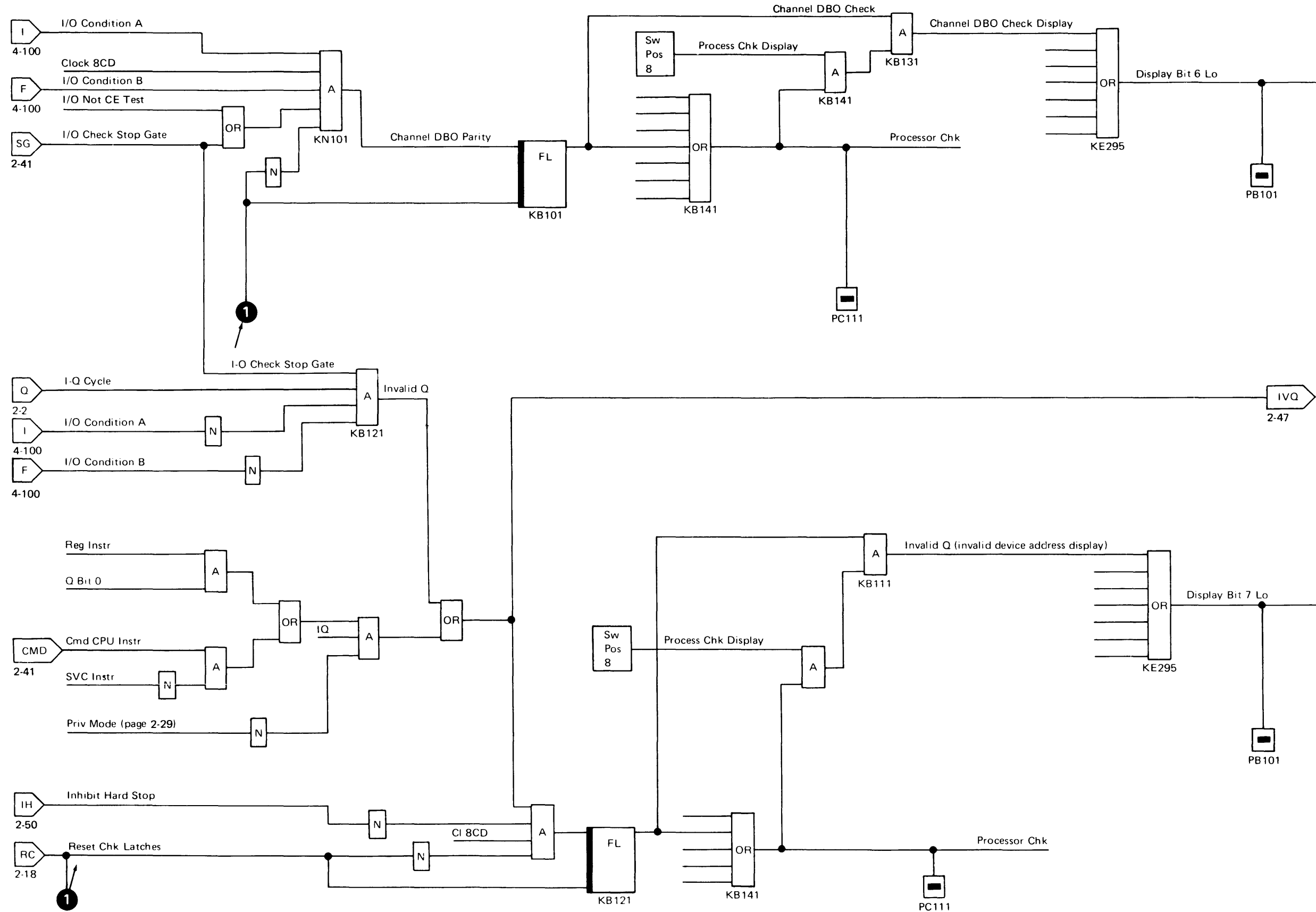
CONSOLE SWITCHES



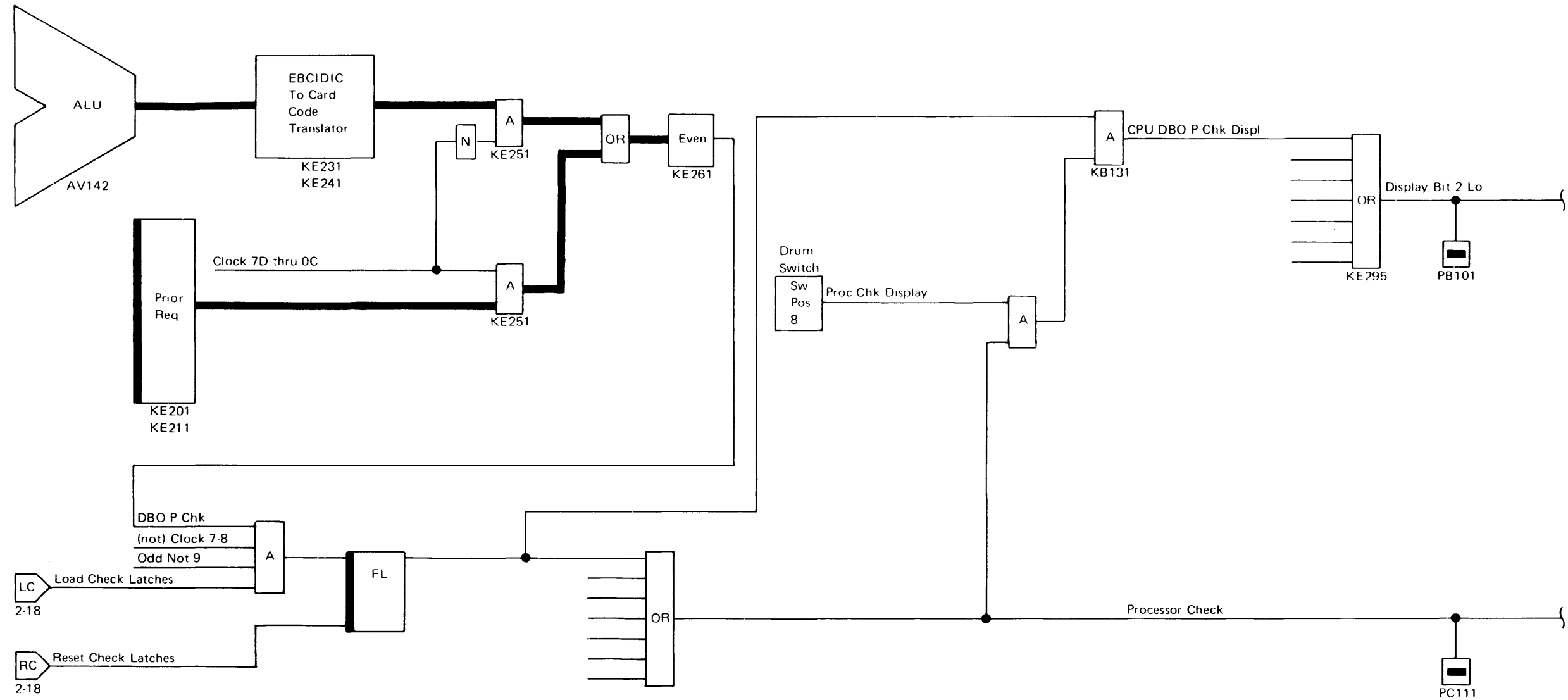
DBI AND TRANSLATOR



### CHANNEL DBO PARITY, INVALID DEVICE ADDRESS

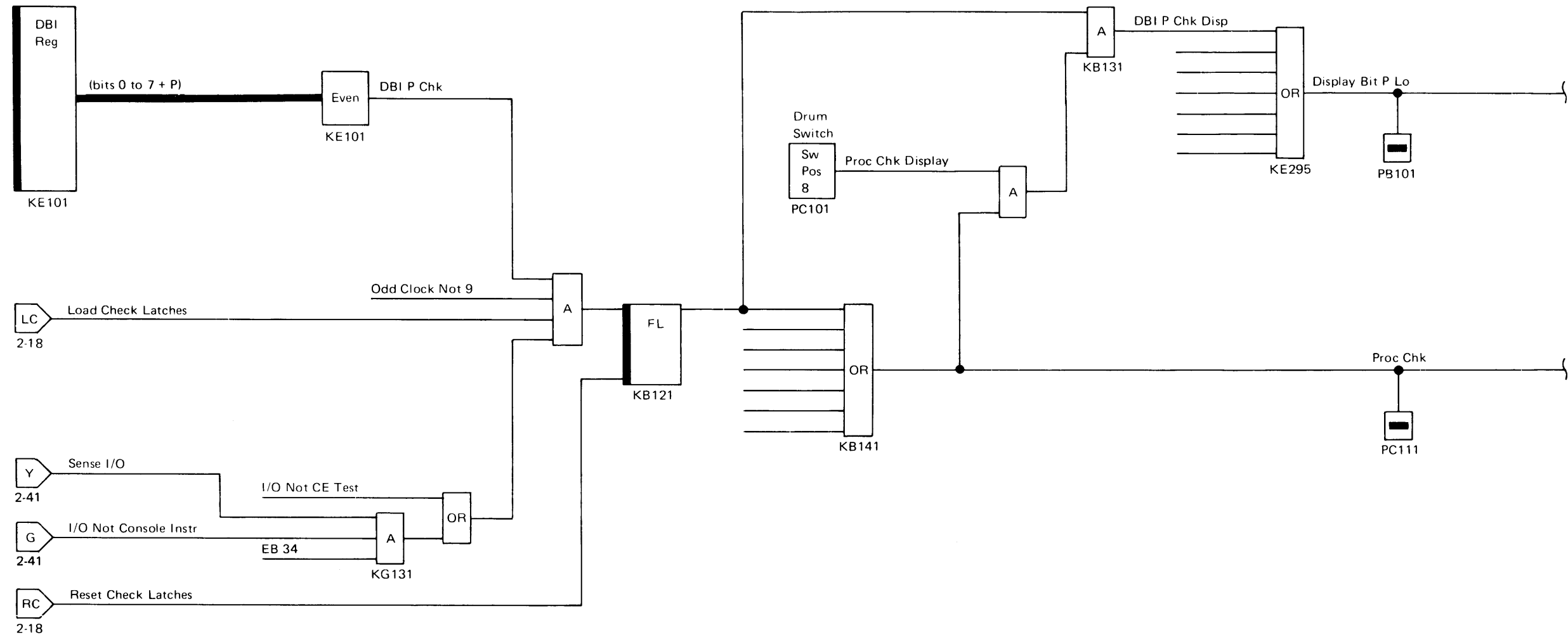


DBO PARITY



*Note:* Rotate drum switch to position 7 to display CS priority bits, position 5 to display ALU output

DBI PARITY



Note: Rotate drum switch to position 5 to display A-register contents. Rotate drum switch to position 1 to display SAR (contains address of failing column) to determine bit structure.





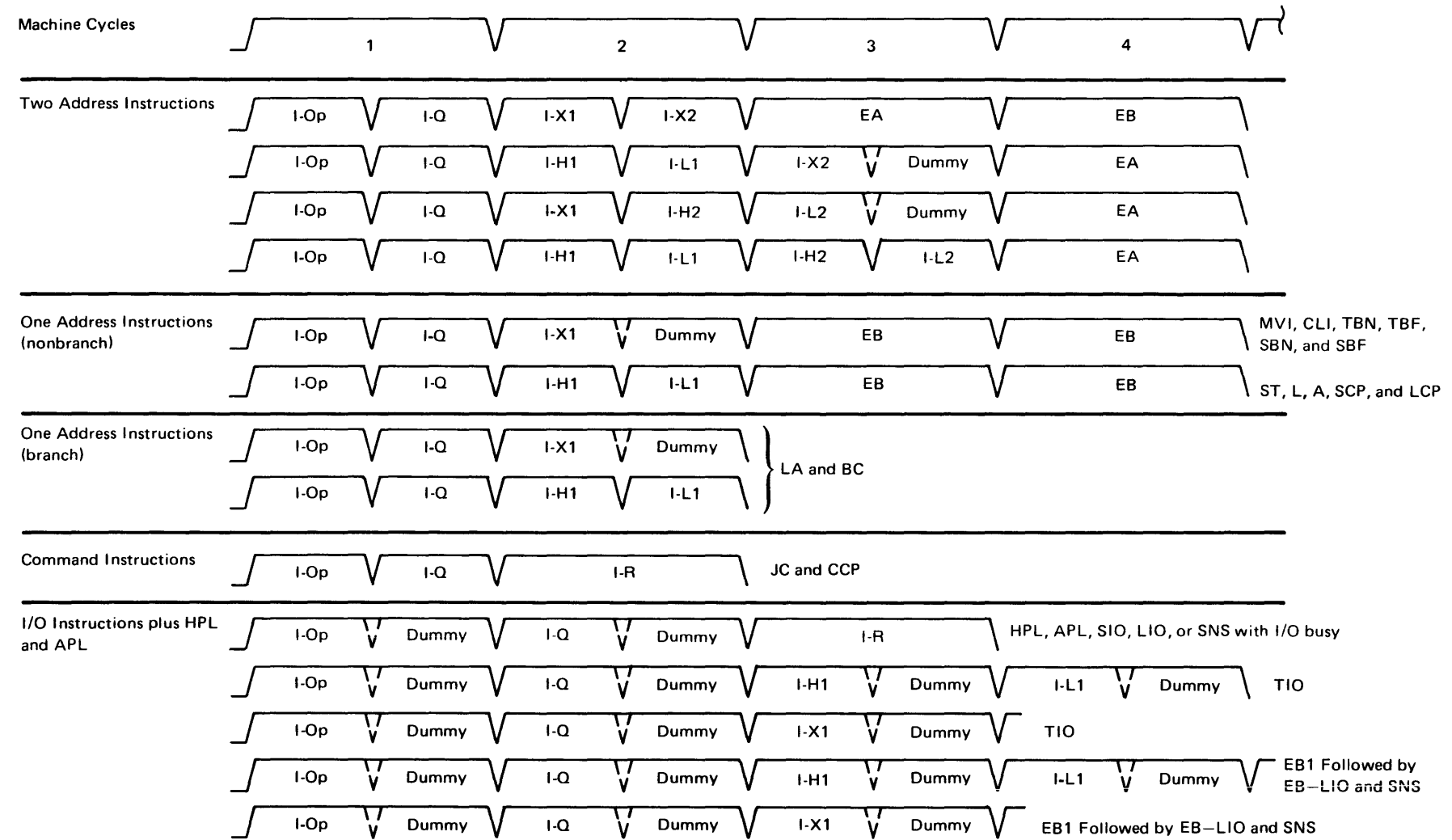
### INSTRUCTION CYCLES (FAST I-CYCLE)

Not all instructions are executed at fast I-cycle speed. Certain instruction types require normal processing speed.

These operations that require normal processing speed include:

- I/O and Halt instructions
- The last cycle of a 3- or 5-byte instruction
- Certain time dependent diagnostic programs
- Cycles during which program checks occur

The following instruction cycle diagram shows the I-cycle, dummy half cycle, and machine cycle relationship for all instruction types.



## TWO-ADDRESS INSTRUCTIONS

### I-Cycles

- Load operation code into op register. **A**
- Load Q code into Q register, LCR, and LCRR. **B**
- Load B field address into BAR. **C**
- Load B field address into ARR for decimal instructions. **C**
- Load A field address into AAR. **D**

To perform two-address instructions, you must know:

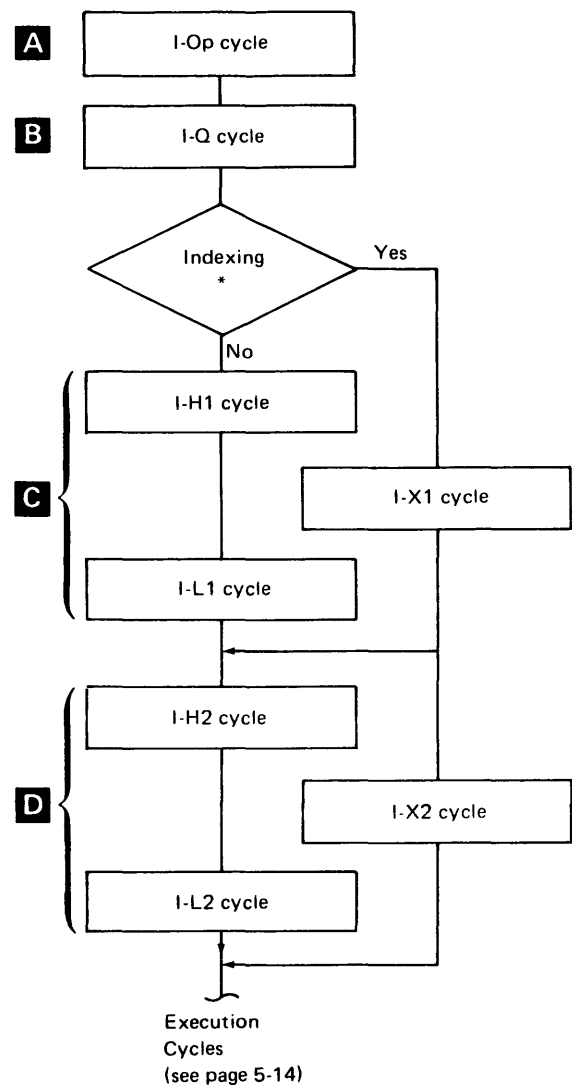
1. What operation.
2. Location of the fields.
3. Length of the fields and any special consideration that must be given them.

I-cycles are used to load the various controlling registers with this information.

First, an I-op cycle transfers the operation code from main storage to the op register. For two-address instructions, the Q code generally contains the length count for the fields. The only exception to this is the move hex character operation in which the Q code controls the data flow. Difference in Q code use are covered under the individual operations. However, in all cases an I-Q cycle loads the Q code into the Q register, the LCR, and the LCRR.

I-H1 and I-L1 cycles load the B field address into the BAR. For decimal instructions, the B field address is also loaded into the ARR. If indexing is used, a single I-X1 cycle replaces the I-H1 and I-L1 cycles.

I-H2 and I-L2 cycles load the A field address into the AAR. The A field address can also be indexed by replacing the I-H2 and I-L2 cycles with a single I-X2 cycle.

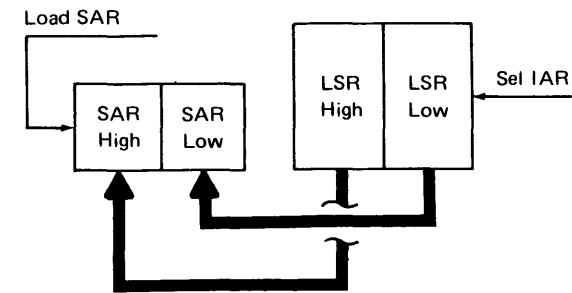


\*Indexing is described on page 5-10.

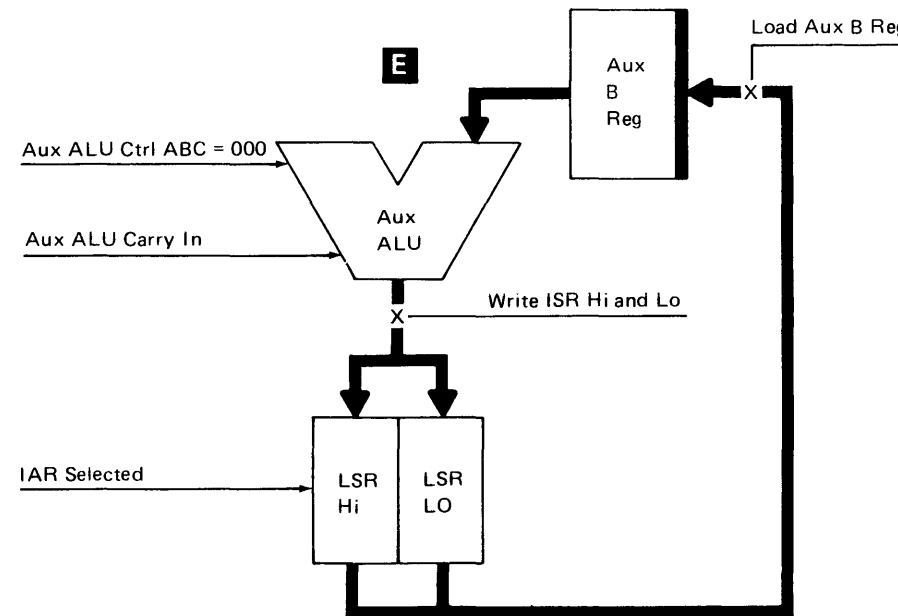
### I-Op Cycle

- Load op code into op register
- Increment instruction address register
- Load CR to CRR

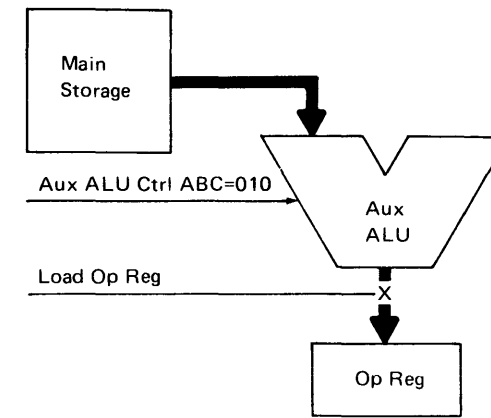
The first step in an I-op cycle, as in all cycles, is to address the storage location to be used during that cycle. The contents of the IAR are transferred to SAR and a storage fetch cycle started.



The two-byte IAR is incremented in parallel with the storage fetch operation. The IAR is transferred through the aux B register and incremented by +1 in aux ALU (aux ALU carry in active) **E**.



At clock 3 time the operation register byte is read from storage and transferred through aux ALU and stored in the op register.

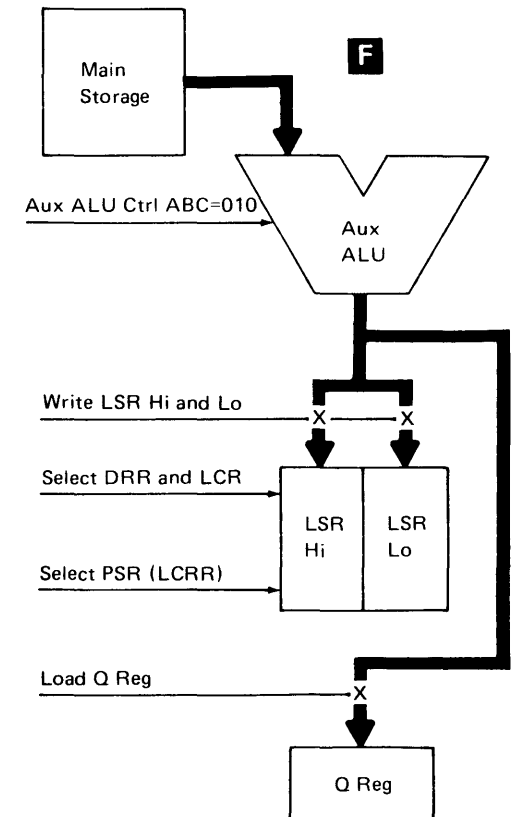


### I-Q Cycle

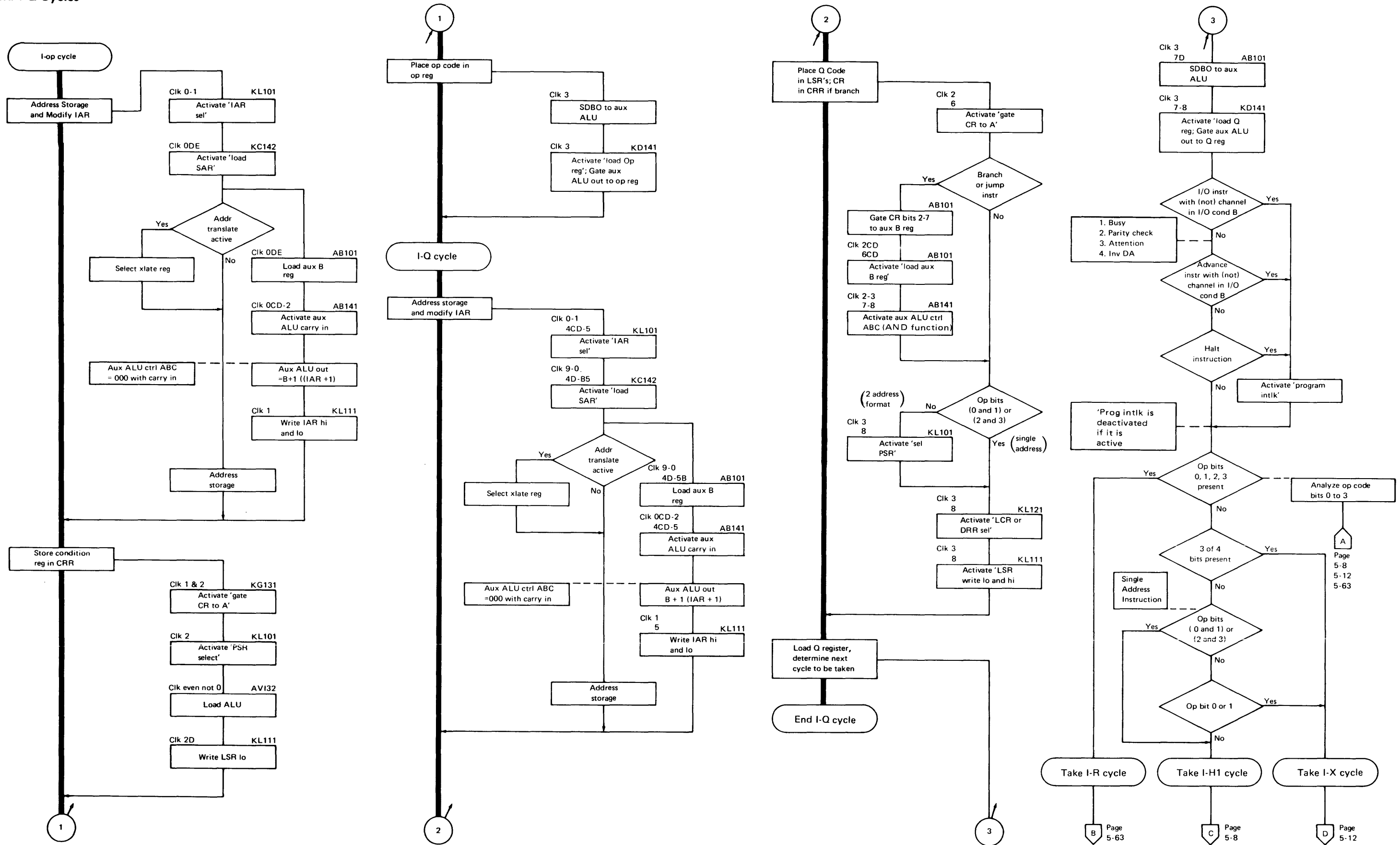
- Load Q code into Q register and: (1) length count register (LCR) and length count recall register (LCRR), or (2) data recall register
- Increment instruction address register

The I-Q cycle is the same as an I-op cycle, except that the Q code byte is stored in the LCR, the LCRR, and the Q register. **F**

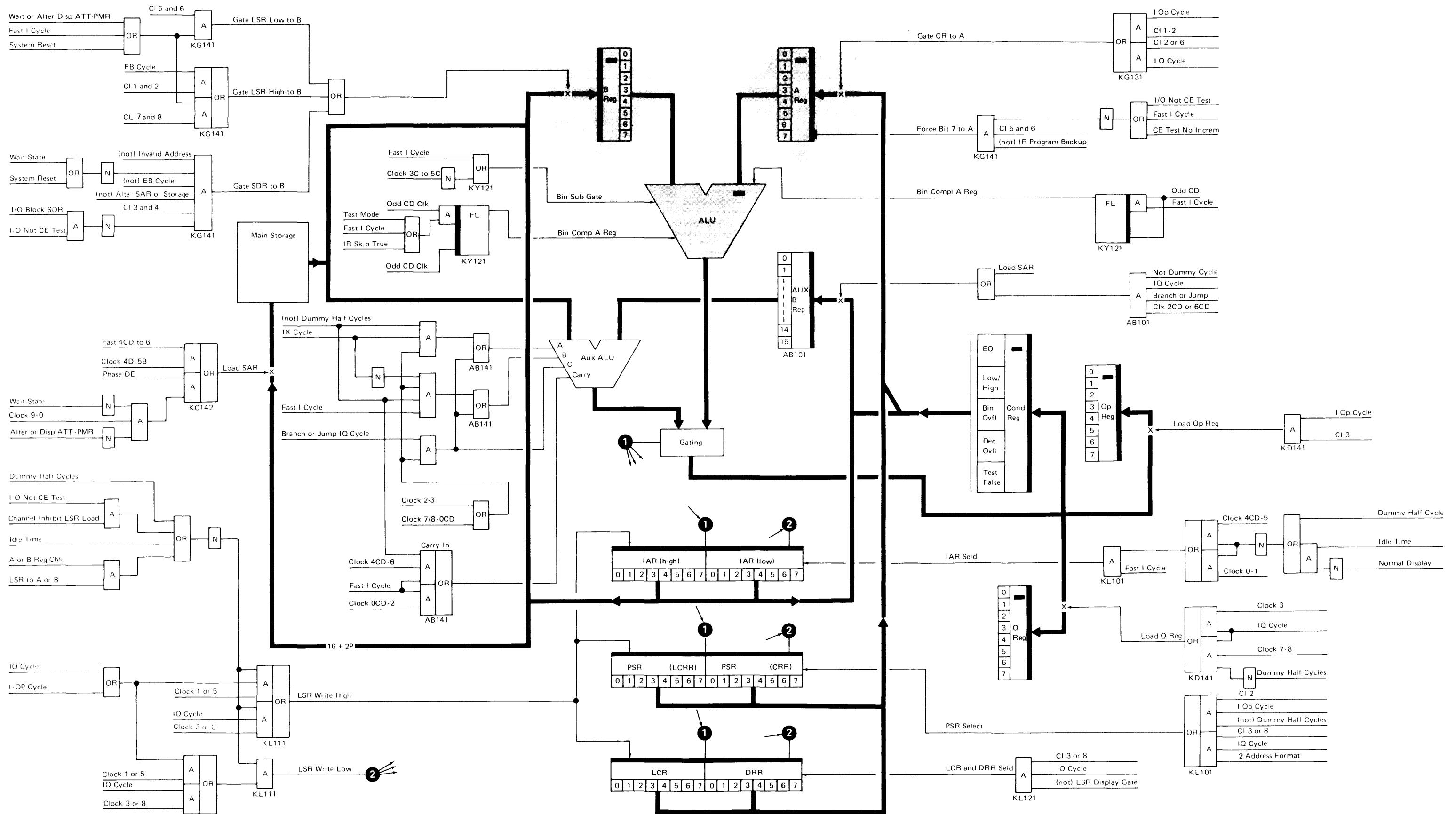
The IAR is incremented in the same manner as for an I-op cycle. **E**



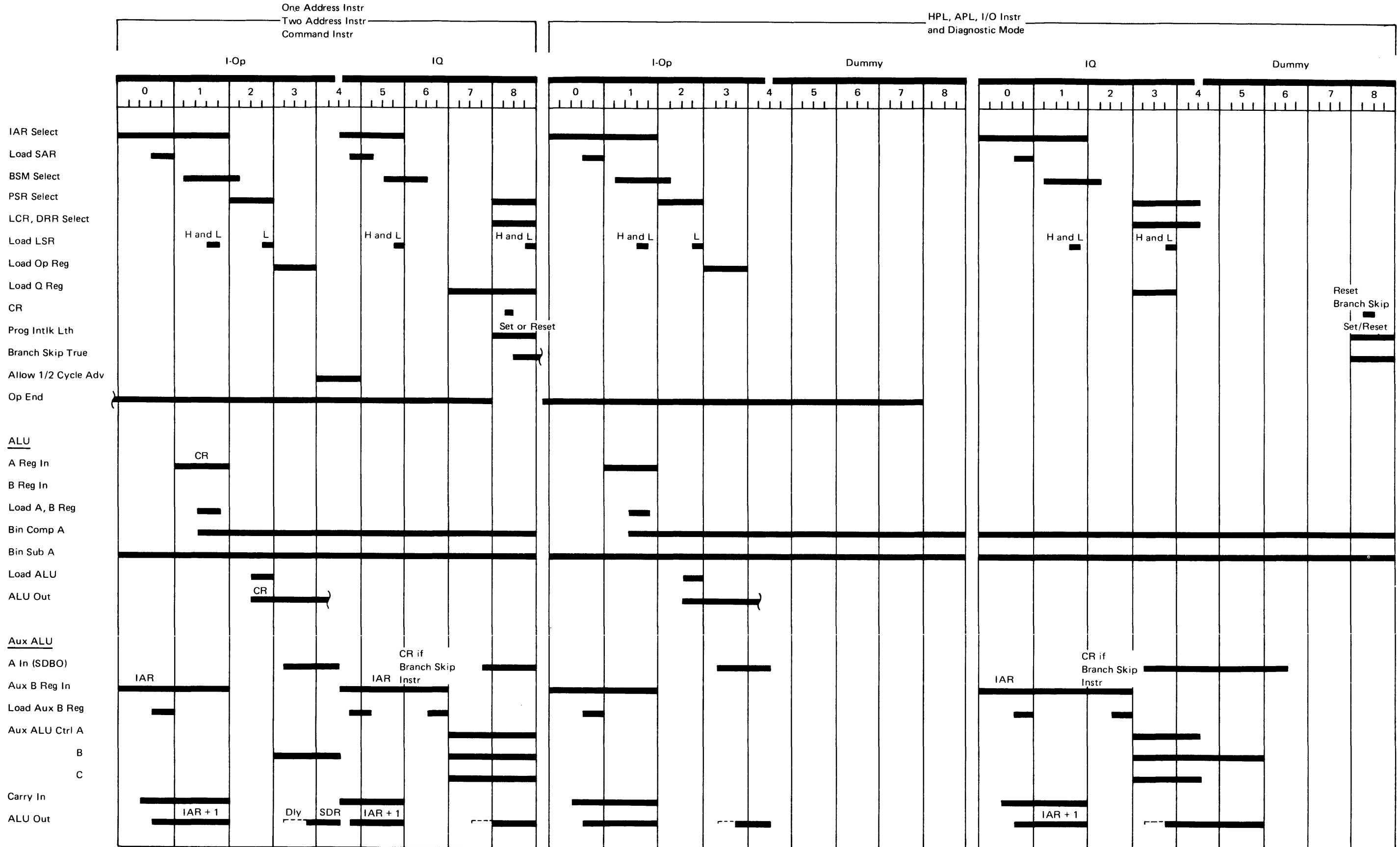
I-Op and I-Q Cycles



# I-Op and I-Q Cycles



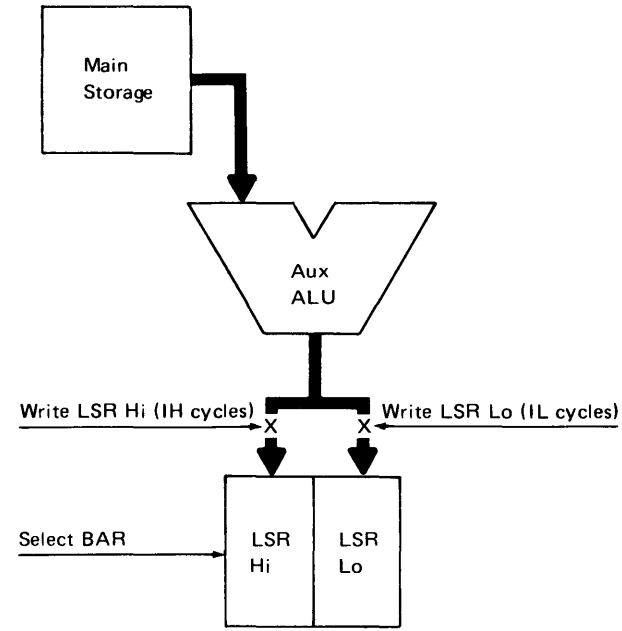
I-Op and I-Q Cycles



### I-H and I-L Cycles

- Load B address register except during load address instruction.
- Load selected index register for load address instruction.
- Load A address register for second address.
- Load address recall register for branch or decimal instructions.

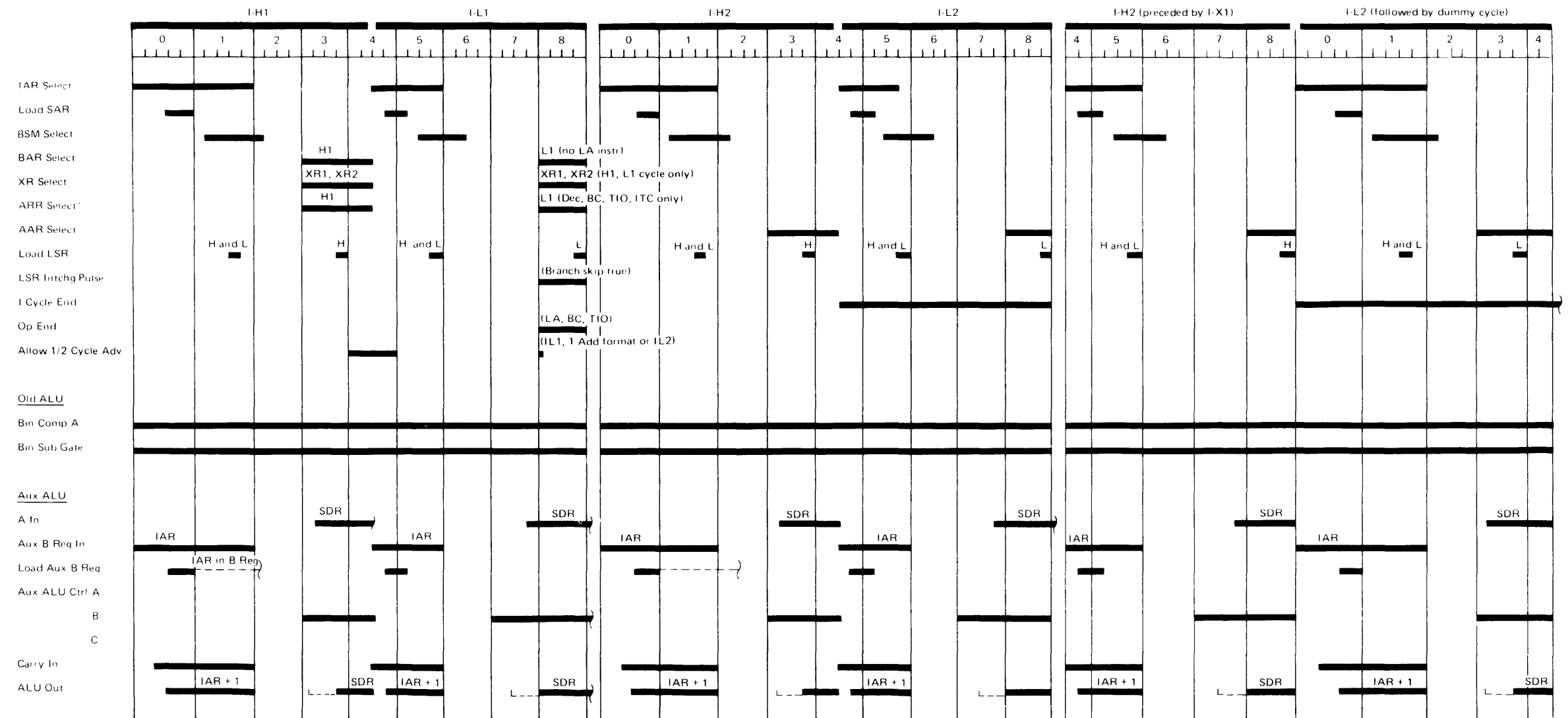
I-H1 and I-L1 cycles are the same as the I-op and I-Q cycles except that the data bytes, B field address, are stored in the BAR. During the I-H1 cycle, the first byte is stored in the high order position of the BAR.



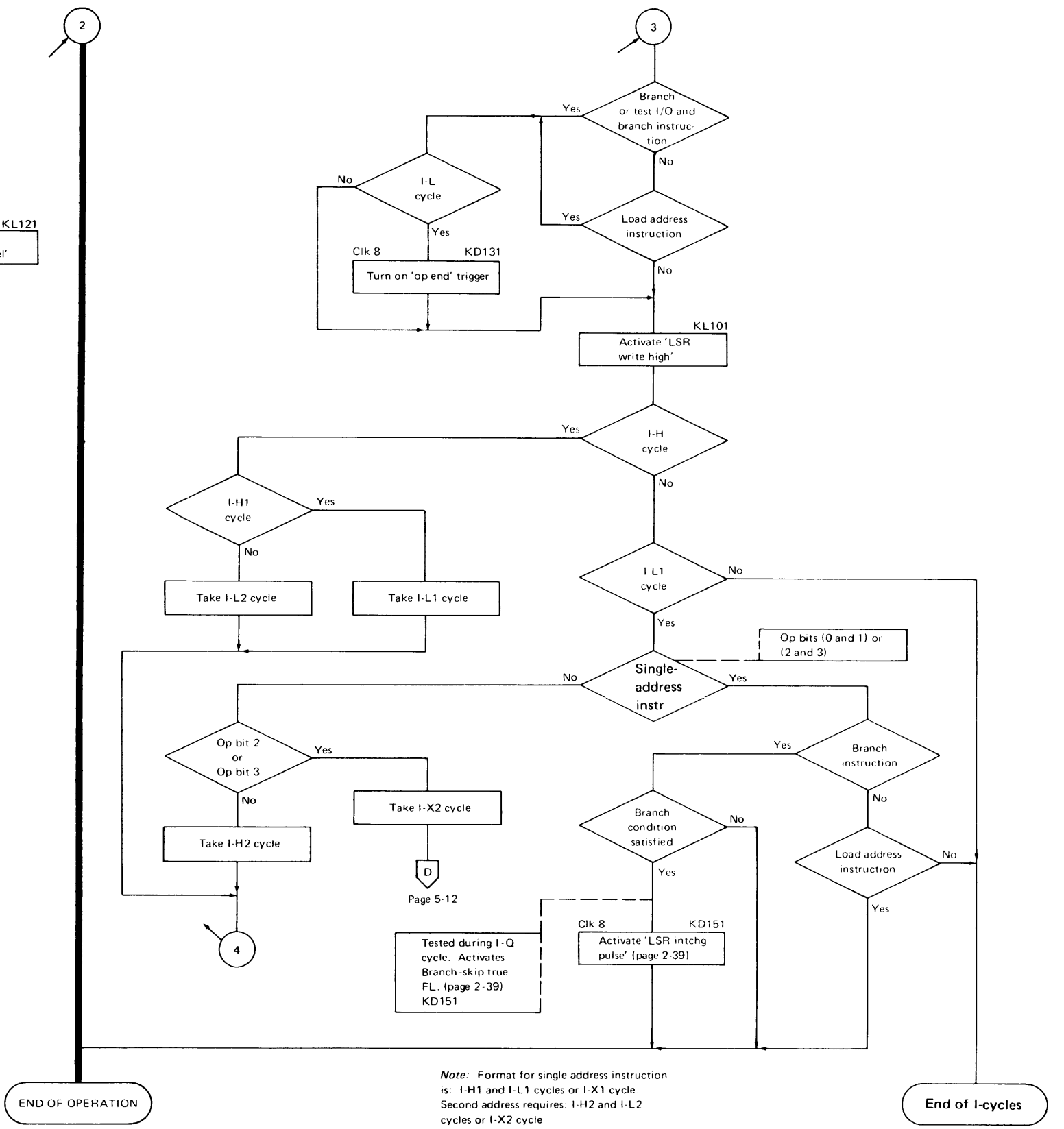
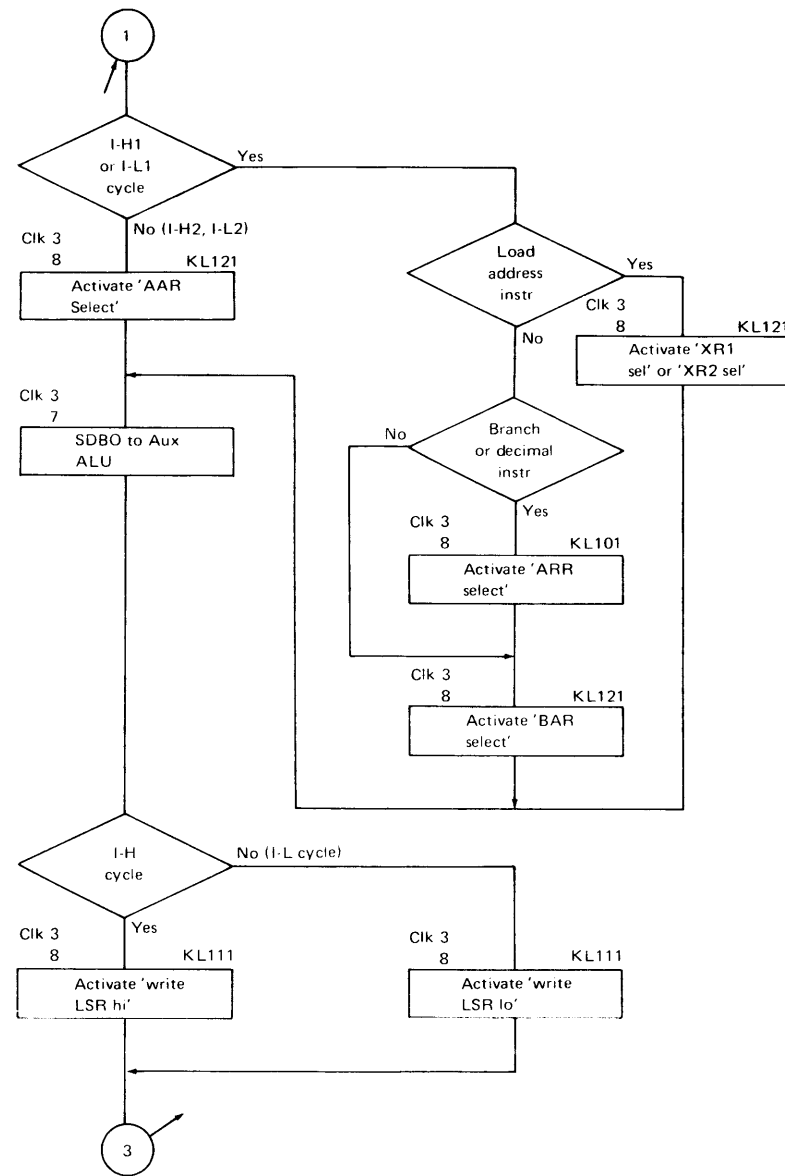
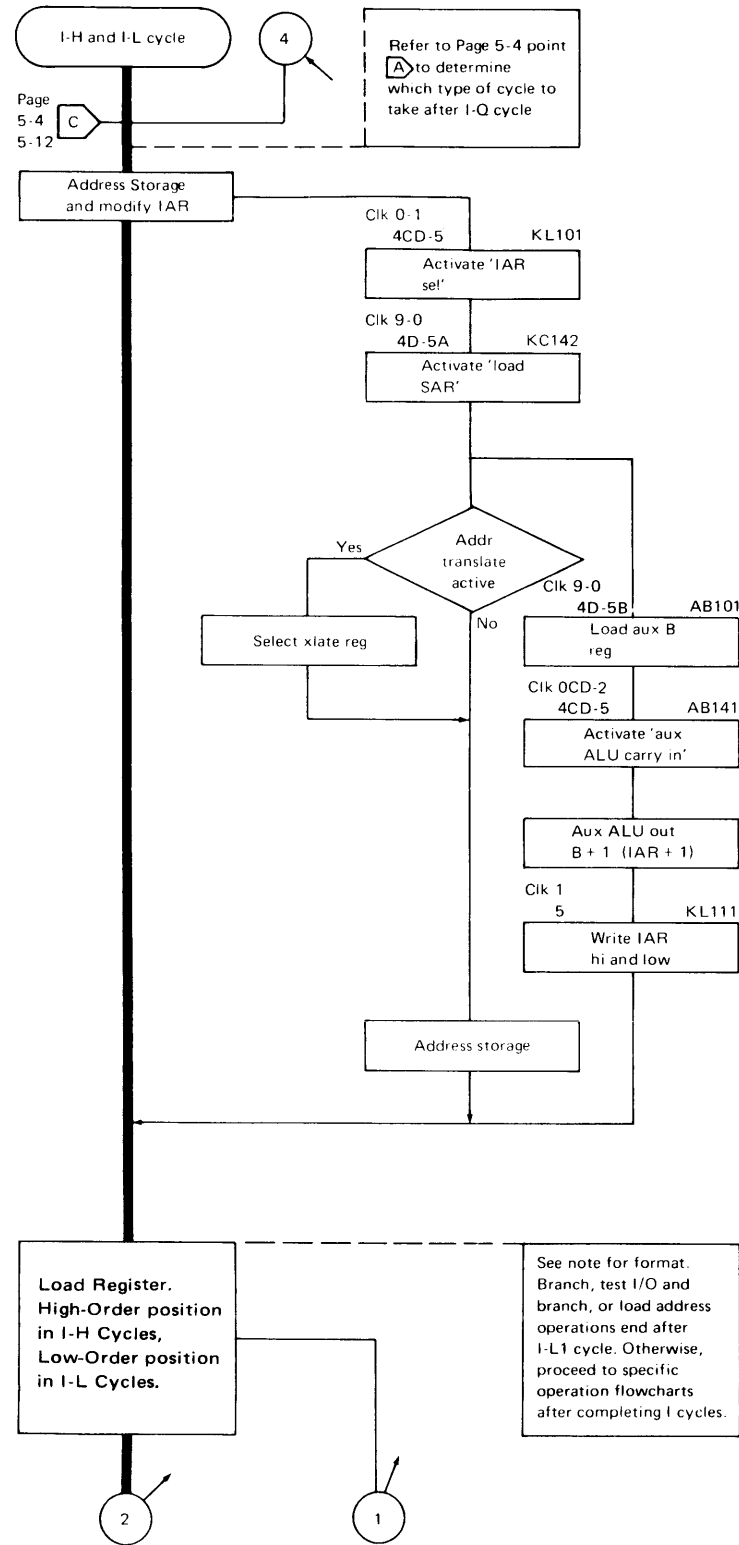
The following cycle, I-L1, stores the second byte in the low-order position of the BAR. For decimal instructions, the bytes are also stored in the ARR.

### I-H2 and I-L2 Cycles

I-H2 and I-L2 cycles are the same as I-H1 and I-L1 cycles except the address bytes are stored in the AAR.

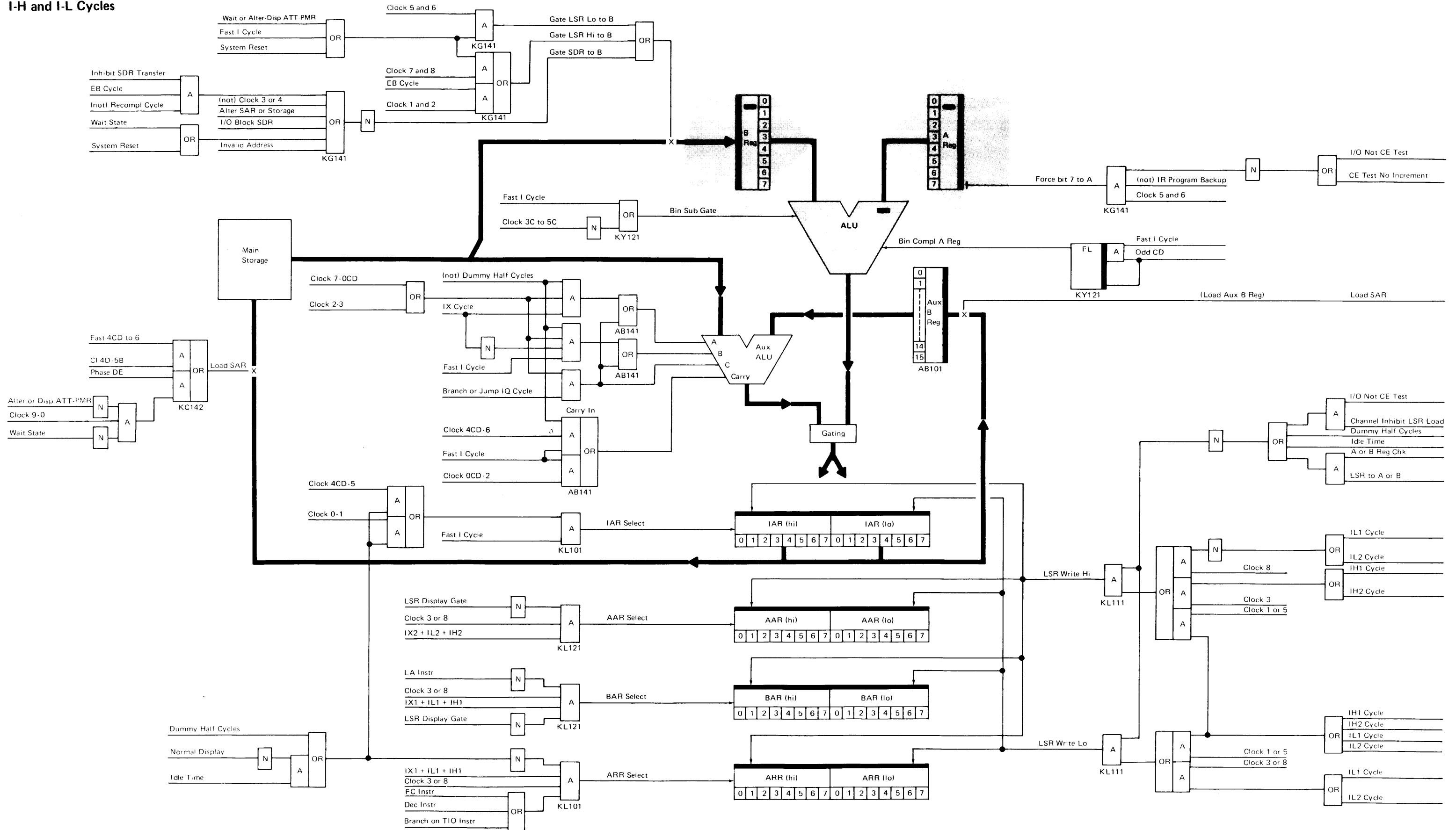


### I-H and I-L Cycles





# I-H and I-L Cycles



**Indexing (I-X Cycles Part 1 of 4)**

- A single byte from the instruction is added to two bytes from an index register to create a new storage address.
- This new address is loaded into the B Address Register.
- This index cycle does not change the instruction byte or the address in the index register. If either is to be changed, an additional CPU instruction is required.

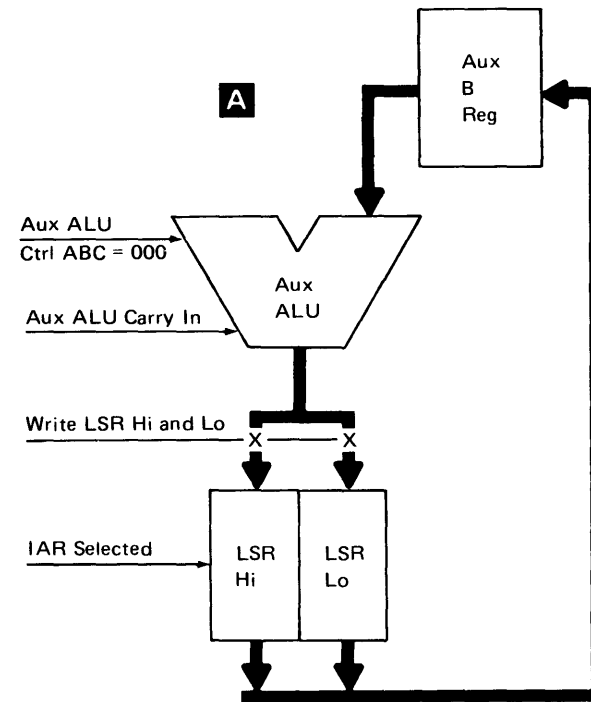
The need for I-X cycles is determined by the bit structure of bits 0 through 3 of the operation register. An I-X1 cycle results from the presence of either bit 0 or bit 1, but not both; an I-X2 cycle from either bit 2 or 3, but not both. The bit that is present also determines the index register used.

Operation Register Bit	Index Register Selected	Cycle
1	XR1	I-X1
0	XR2	
3	XR1	I-X2
2	XR2	

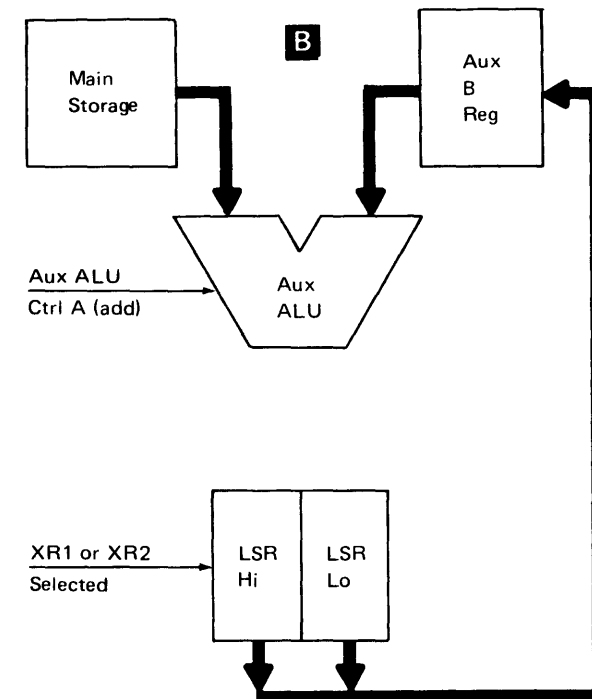
Instruction Format	Op Bits				B Address Register Contains:
	0	1	2	3	
One Address (nonbranch)	0	0	1	1	Two bytes from storage unchanged
	0	1	1	1	One byte from storage added to address from XR1
	1	0	1	1	One byte from storage added to address from XR2
One Address (branch)	1	1	0	0	Two bytes from storage unchanged
	1	1	0	1	One byte from storage added to address from XR1
	1	1	1	0	One byte from storage added to address from XR2
	1	1	1	1	Command

During I-X cycles the IAR is selected and loaded into the SAR in the same manner as for other instruction cycles. A storage fetch cycle is started and the B field address fetched from storage.

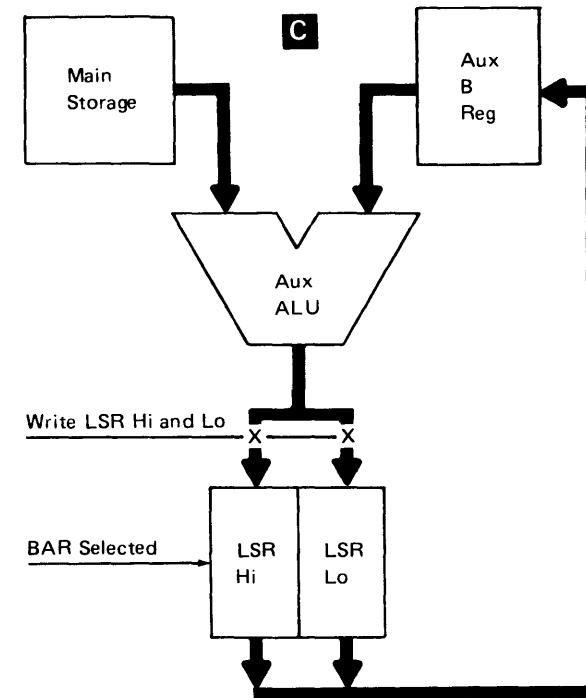
The two-byte IAR is incremented in parallel with the storage fetch operation. When the IAR is gated to SAR it is also transferred through the aux B register and incremented by +1 in the aux ALU (aux ALU carry in active) **A**.



The selected index register (XR1 or XR2) is transferred to the aux B register (right side of aux ALU). The B field address is read from storage and enters the left side of the aux ALU. The index register and the B field address are then added in the aux ALU **B**.



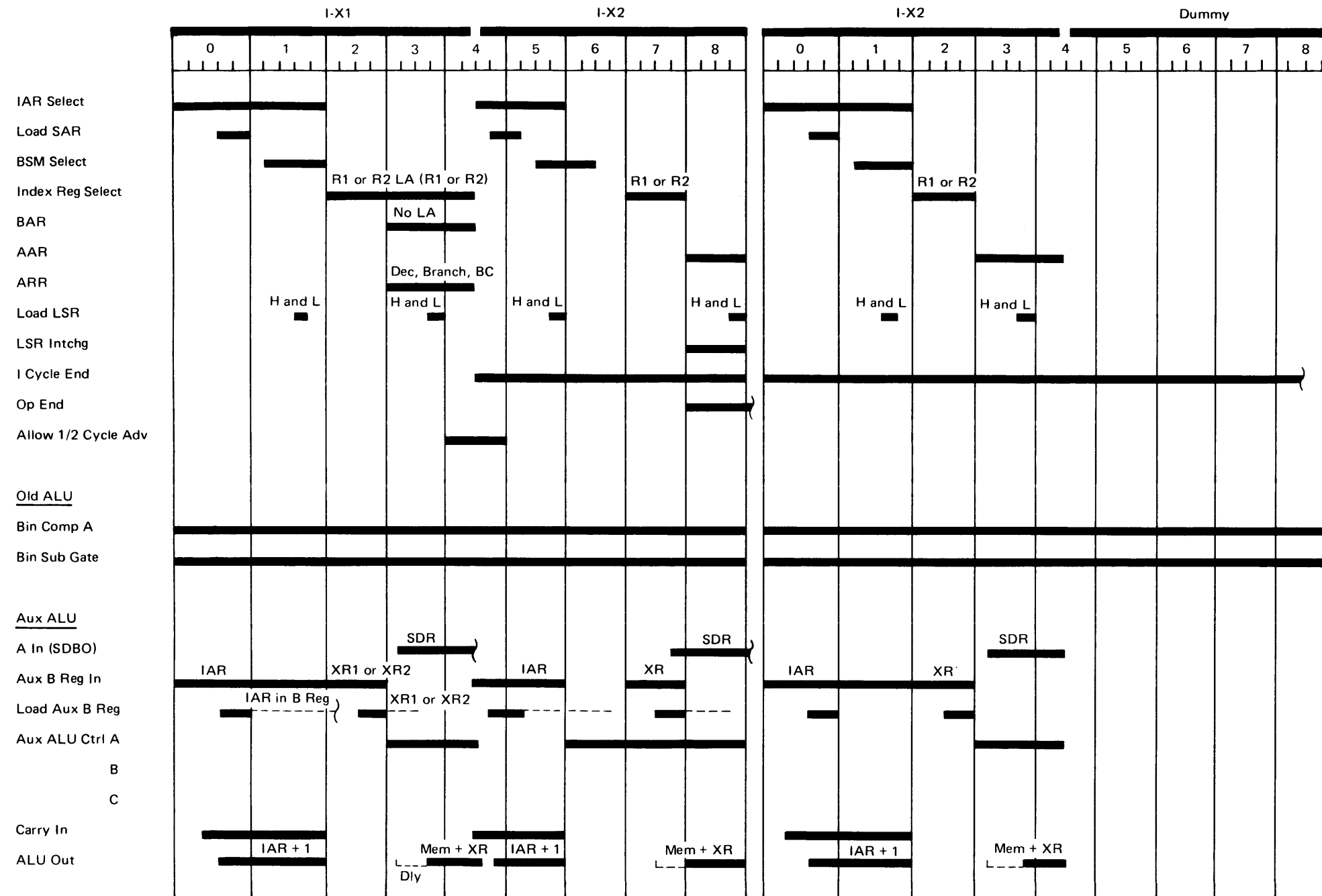
The aux ALU output is written into the low- and high-order positions of the BAR **C**.



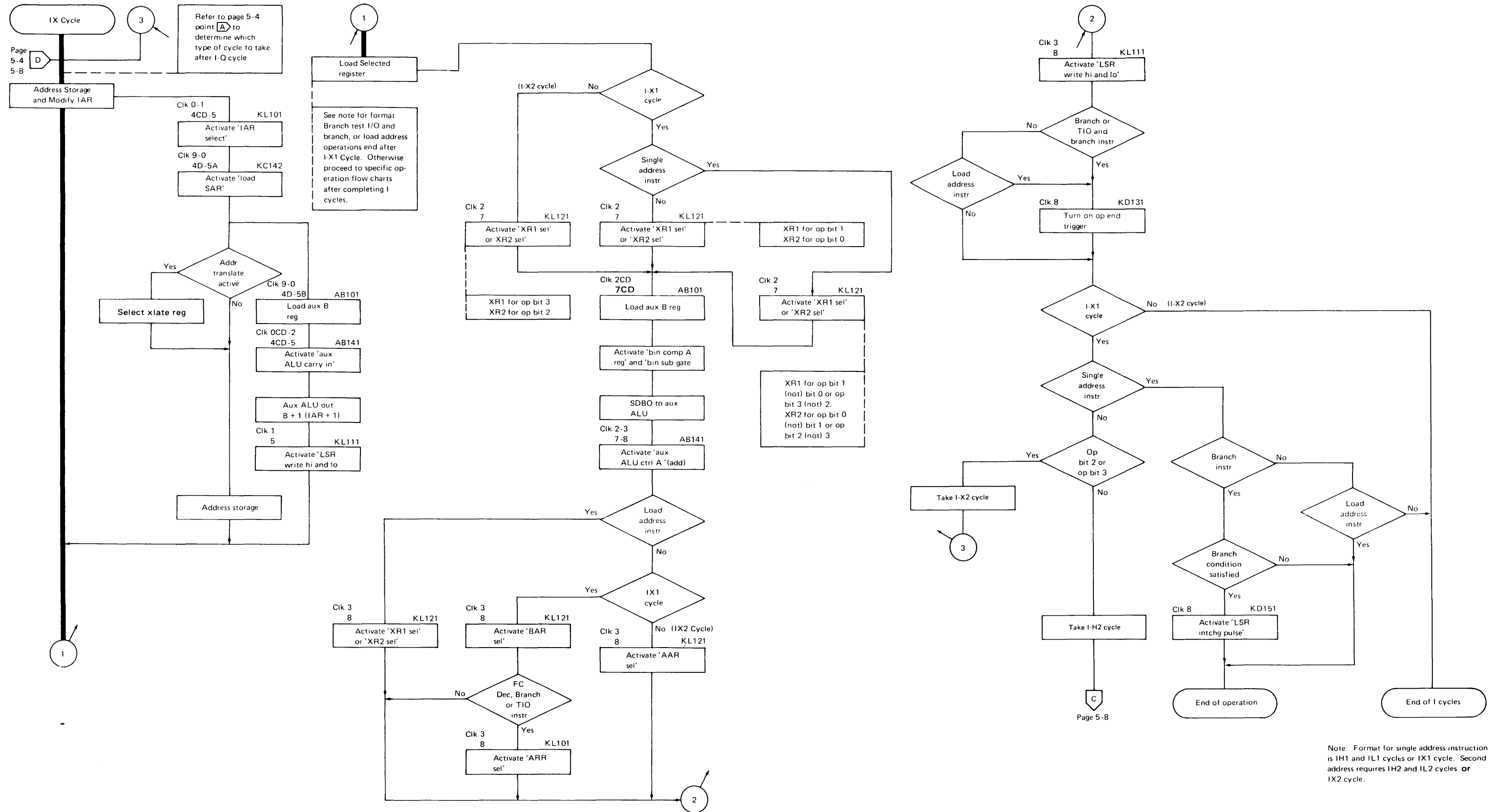
If the operation is a decimal or branch operation, the results are also stored in the ARR. An I-X2 cycle operation is the same as a I-X1 except the results are written in the AAR.

Instruction Format	Op Bits				B Address Register Contains:	A Address Register Contains:
	0	1	2	3		
Two Address	0	0	0	0	Two bytes from storage unchanged	Two bytes from storage unchanged
	0	0	0	1		One byte from storage added to address from XR1
	0	0	1	0		One byte from storage added to address from XR2
	0	1	0	0	One byte from storage added to address from XR1	Two bytes from storage unchanged
	0	1	0	1		One byte from storage added to address from XR1
	0	1	1	0		One byte from storage added to address from XR2
	1	0	0	0	One byte from storage added to address from XR2	Two bytes from storage unchanged
	1	0	0	1		One byte from storage added to address from XR1
	1	0	1	0		One byte from storage added to address from XR2

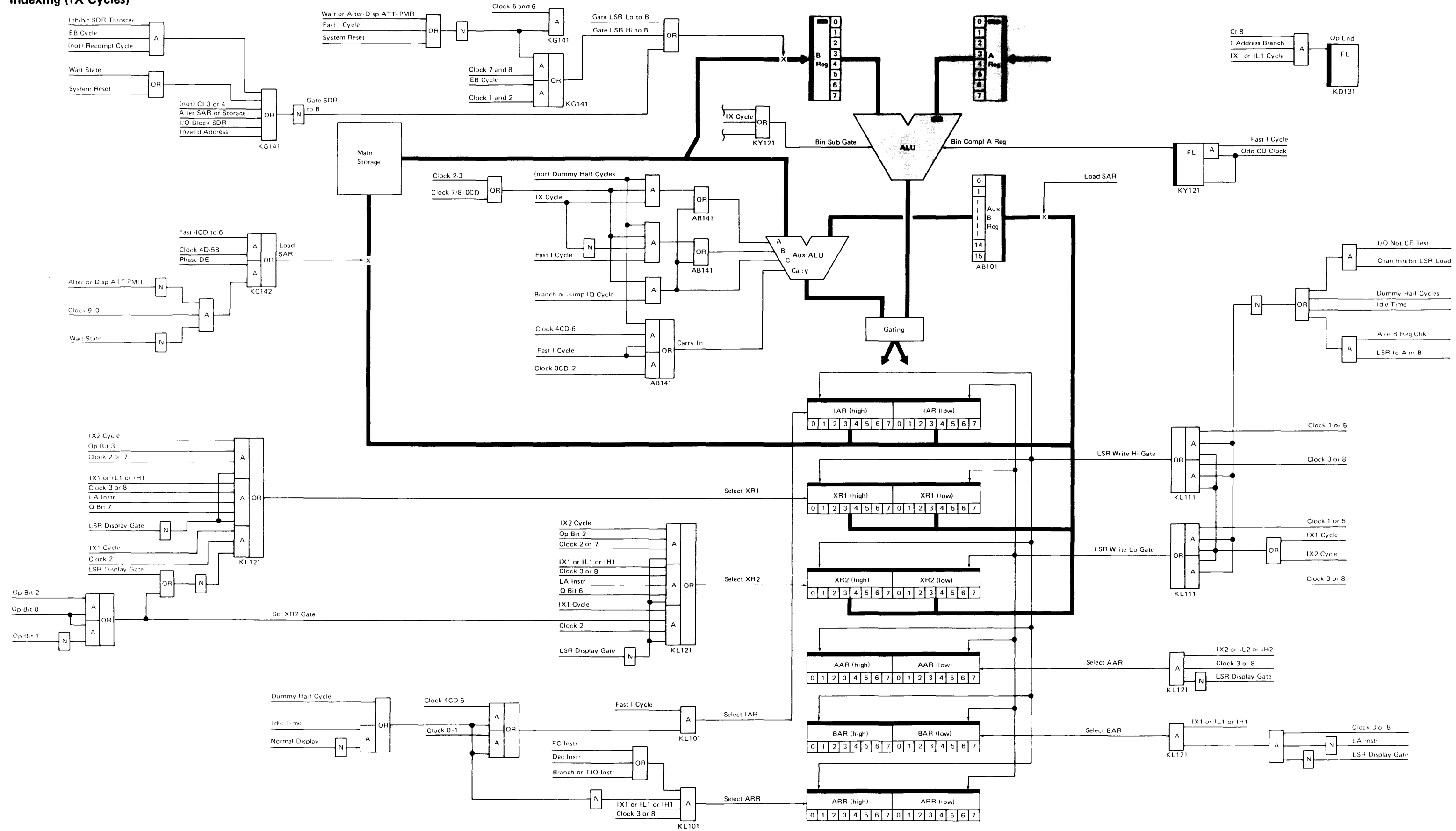
**Indexing (I-X Cycles)**



### Indexing (IX Cycles)



# Indexing (IX Cycles)



**Execution Cycles**

Because only one byte at a time can be removed from or placed into main storage, two cycles per byte are required when controlling data between two different storage locations. During the A-cycle, the A field byte is removed from storage and retained. The B-cycle is then used to remove the B field byte from storage and, depending upon the particular operation, to determine what to do with each byte.

**A-Cycle**

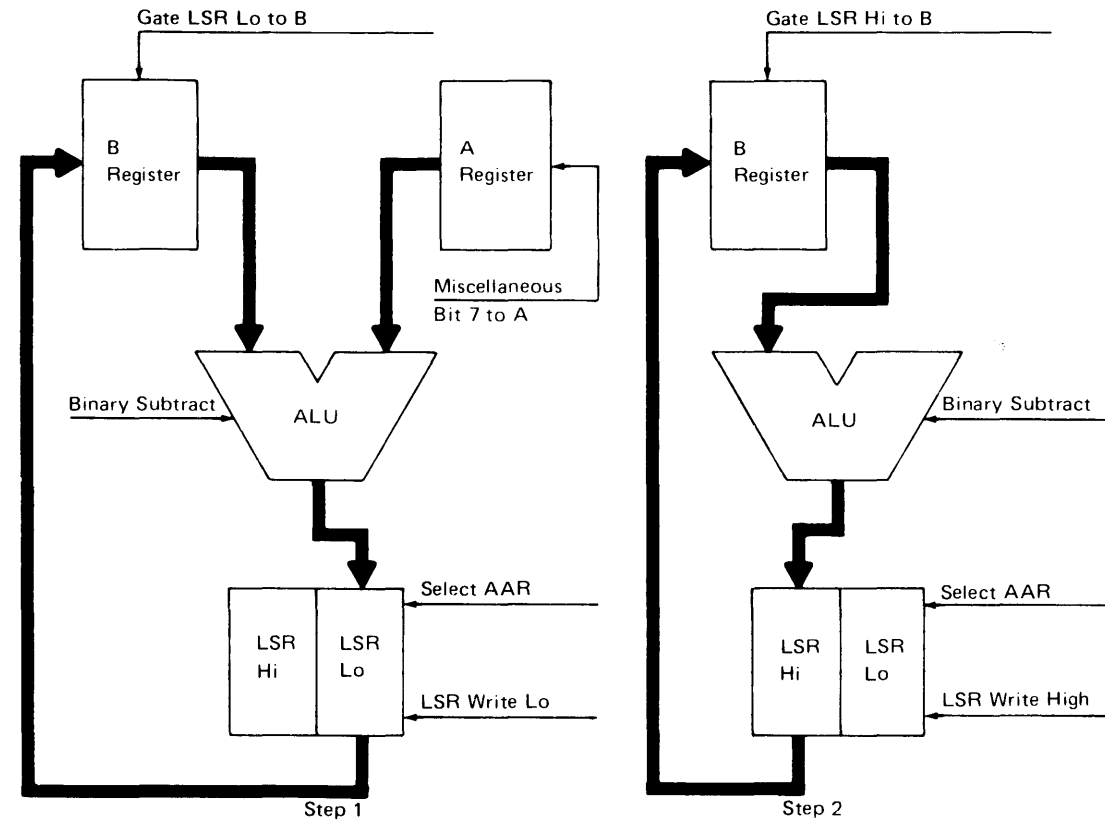
Because the A-cycle data flow is the same, regardless of the operation, it can be covered as a separate topic. Some operations require the condition register to be reset to equal during the first A-cycle and some require the use of sign control for the A field character, but the basic data flow remains the same.

- Store A field byte in DRR.

The first step in an A-cycle, as in all cycles, is to address the storage location to be used during that cycle. At clock 0 time, the contents of the AAR are transferred to the SAR in the same manner that the IAR was transferred.

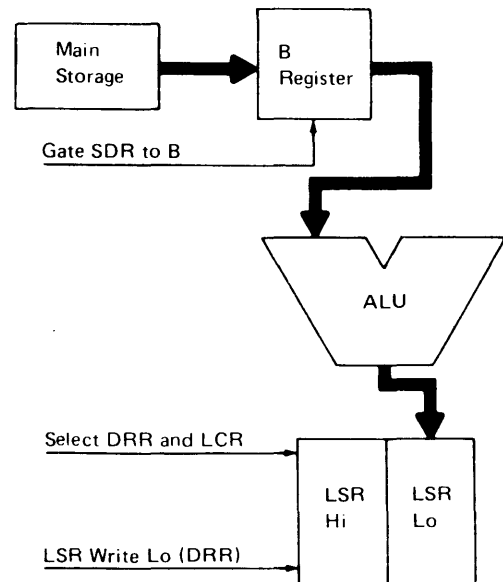
No data is transferred during clock 1 and 2 times as the CPU waits for the data to be read from storage and enter the FDR. During clock 3 and 4 times, the byte is transferred through the B register and ALU and stored in the DRR.

The rest of the cycle is then used to decrement the AAR so that the next position of the A field can be addressed if necessary. The example below shows that a 1 is subtracted from the AAR. Two steps are required because of the possibility of a carry from the low-order to the high-order position



**B-Cycle**

The B-cycle operation is covered under the individual operations descriptions that follow.



### Add Logical Characters—ALC

Bits 0 1 2 3 4 5 6 7  
Op Code x x x x 1 1 1 0

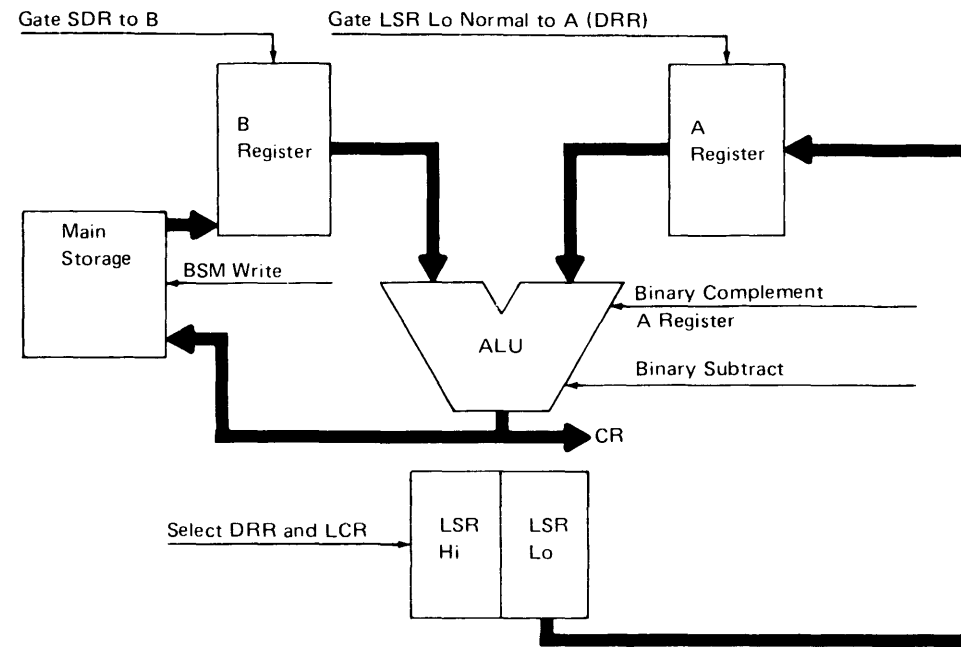
- Binary add A field to B field data.
- A and B fields are the same length (Q code plus 1).

The add logical characters operation adds the A field data, one byte at a time, to the B field data. The entire A field byte (bits 7 through 0) is binary added to the B field byte. The operation begins with the low order position of each field and continues until the high order position is reached. Both fields are the same length, which is 1 more than the Q code.

The CPU performs the add logical characters operation with a series of A- and B-cycles. First an A-cycle removes the first A field byte from storage and retains it in the DRR. Then a B-cycle removes the first B field byte from storage, adds it to the A field byte, and stores the result in the B field units location. The next A field byte is then added to the second B field byte through the same process, and so on to the end of the field.

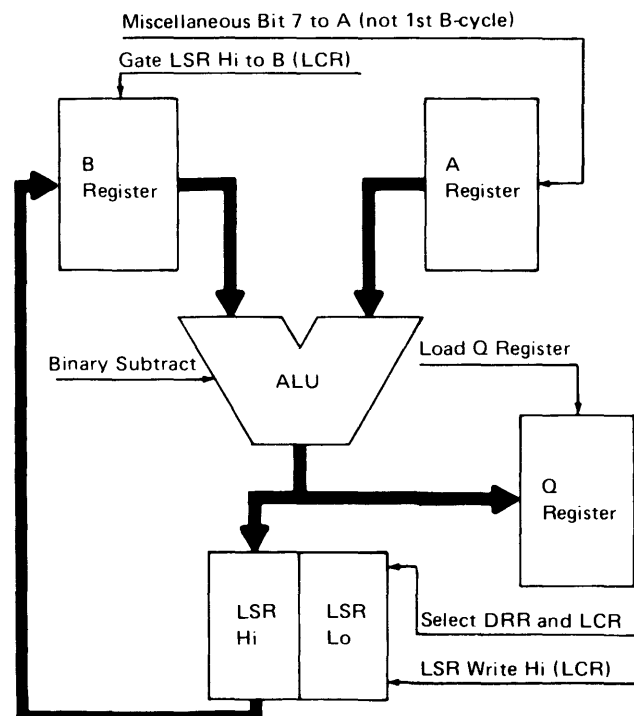
After the first A field byte has been stored in the DRR and the AAR has been decremented (refer to A-cycle), the CPU enters a B-cycle. The BAR is selected and loaded into the SAR in the same manner that the IAR was transferred.

The B field units byte is read from storage and is loaded into the B register. The A field byte is transferred from the DRR to the A register and the two bytes are binary added in the ALU. The result is then stored into the B field units storage location.



The BAR is then decremented in the same manner that the AAR was decremented. The Q register is tested to see if the end of the field has been reached (blank Q register). If the Q register is not blank, the CPU takes another A-cycle and another B-cycle to add the next characters; if the Q register is blank, the 'op-end' trigger is turned on and the operation ends.

During clock 1 and 2 of each B-cycle, except for the first, the LCR is decremented.



The LCR contains the field length which was stored there during the I-Q cycle. The result, which is latched into the ALU at clock 2CD time, is loaded into the Q register at clock 3 time. By not decrementing the LCR on the first B-cycle, the field length becomes 1 more than the Q code.

An additional function of the add logical characters operation is to set the condition register.

	Equal	Low	High	Binary Overflow
Result	is zero	No Carry and non-zero result	Carry and non-zero result	Result too large for field (no high order carry)

During clock 1 and 2 of the first B-cycle of the operation, the condition register is reset to equal. During each B-cycle, after computing the A and B field data at clock 3 and 4 time, the ALU output is sampled. If the ALU output is all zeros, the condition register remains set to equal. However, if an ALU output occurs during any B-cycle the result can no longer be equal and the equal condition is reset.

Once the equal condition has been reset the final high or low setting of the condition register is not determined until the last B-cycle of the operation. In the meantime, because of the machine circuitry, a CR high condition will be indicated. During the last B-cycle (Q register all zeros) if a carry results from the computation, the CR is set to low; if no carry occurs the condition register remains set to high.

If there is no carry from the high order position, the CR binary overflow condition is also set. This is an indication that the result is too large to be contained in the B field.

The condition register is set as shown:

Equal	Low	High
A and B fields are equal	B field is lower than A field	B field is higher than A field

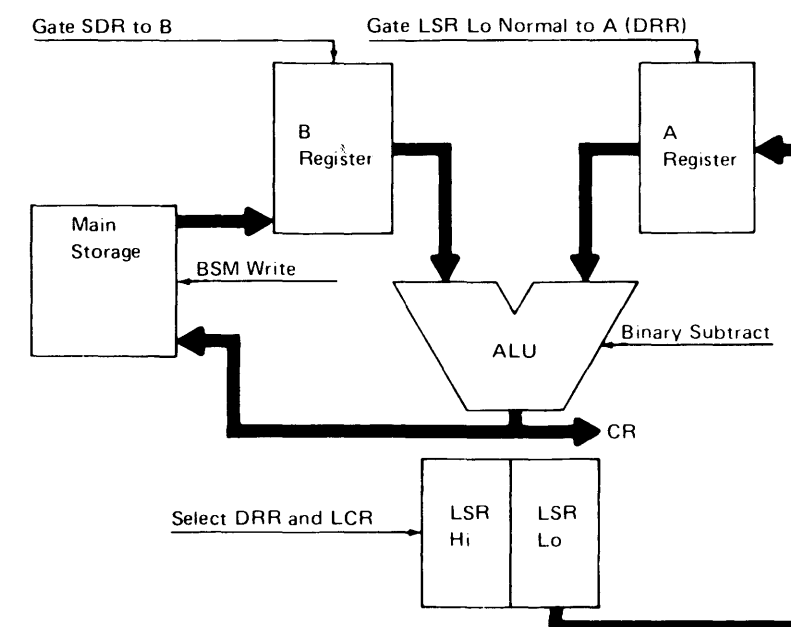
Although the settings have a different significance, the CR is set in the same manner as in the add logical characters operation, except the binary overflow is not set on.

### Subtract Logical Characters—SLC

Bits 0 1 2 3 4 5 6 7  
Op Code x x x x 1 1 1 1

- Binary subtract A field characters from B field characters.
- A and B fields are the same length (Q code plus 1).

The subtract logical characters operation is the same as the add logical characters operation except that the A field data is subtracted from the B field data.

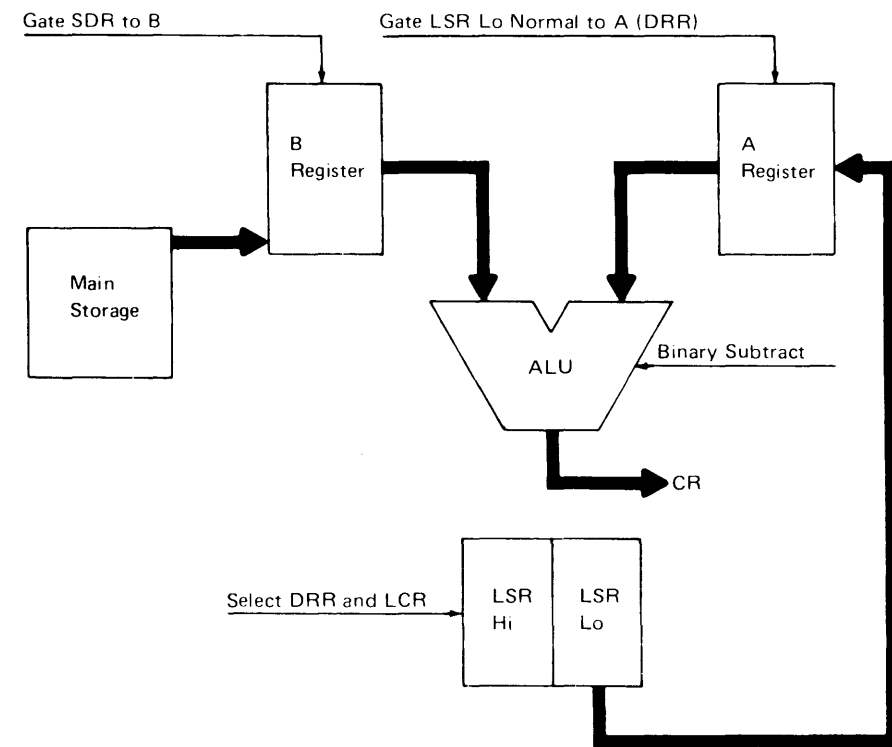


**Compare Logical Characters—CLC**

Bits      0 1 2 3 4 5 6 7  
 Op Code  x x x x 1 1 0 1

- Compare A field data to B field data.
- A and B fields are the same length (Q code plus 1).

During the compare logical characters operation, the CPU compares the two fields by binary subtracting the A field data from the B field data. The operation is the same as a subtract logical operation except that the results are not entered into storage.



The A and B fields remain unchanged by the operation, and the ALU results are used merely to set the condition register:

Equal	Low	High
A and B fields are equal	B field is lower than A field	B field is higher than A field

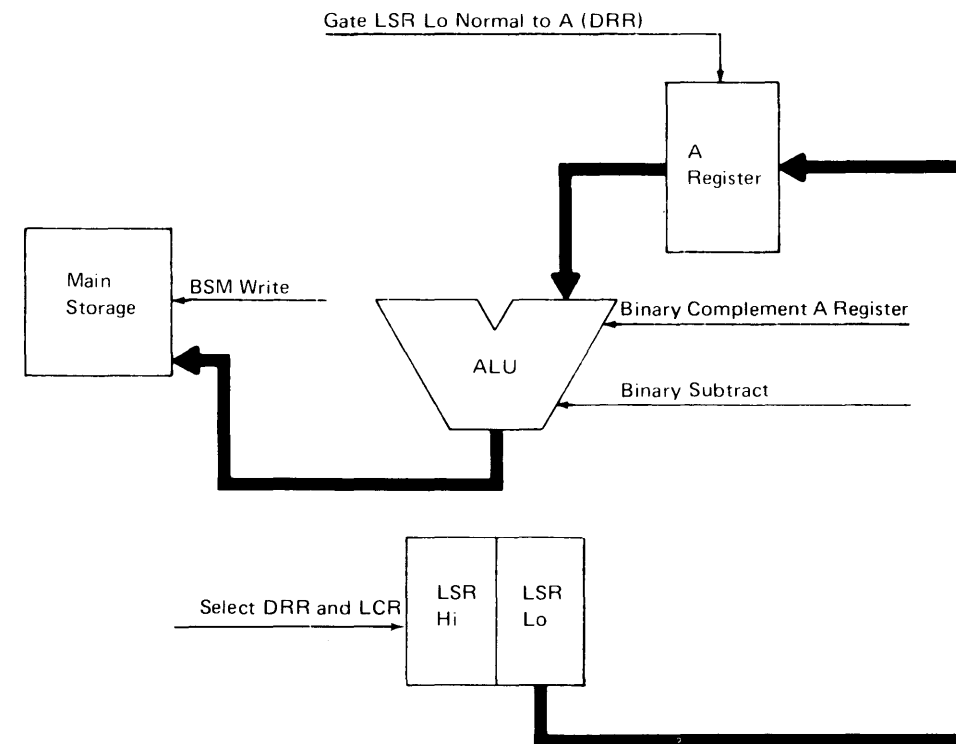
**Move Characters—MVC**

Bits      0 1 2 3 4 5 6 7  
 Op Code  x x x x 1 1 0 0

- Move A field characters to B field location.
- A and B fields are the same length (Q code plus 1).

The move characters operation moves the A field data, one byte at a time, into the B field location. The operation begins with the low-order position of each field and continues through the high order position.

The operation is the same as add logical characters (previous page), except that the B field character is not loaded into the B register.



When the B register is blank, the operation is the same as adding the A field to zero.

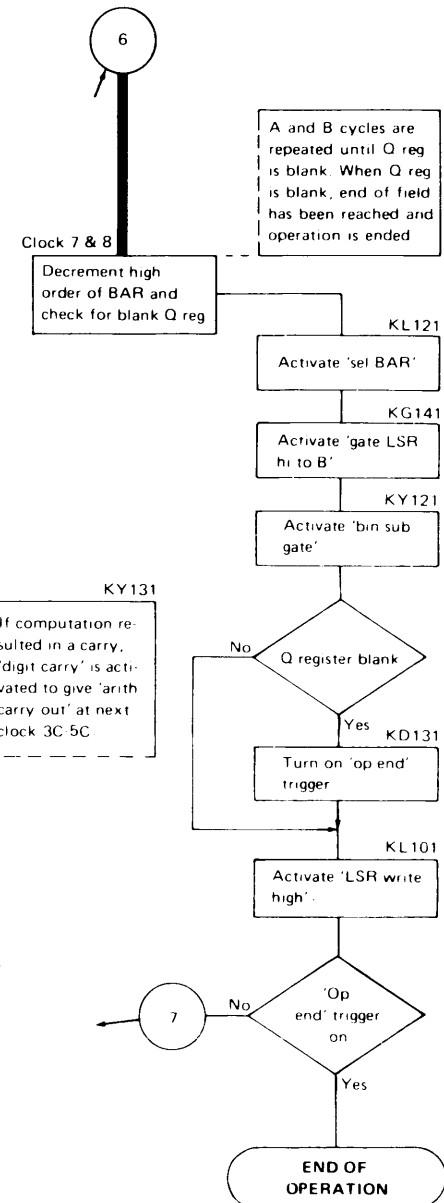
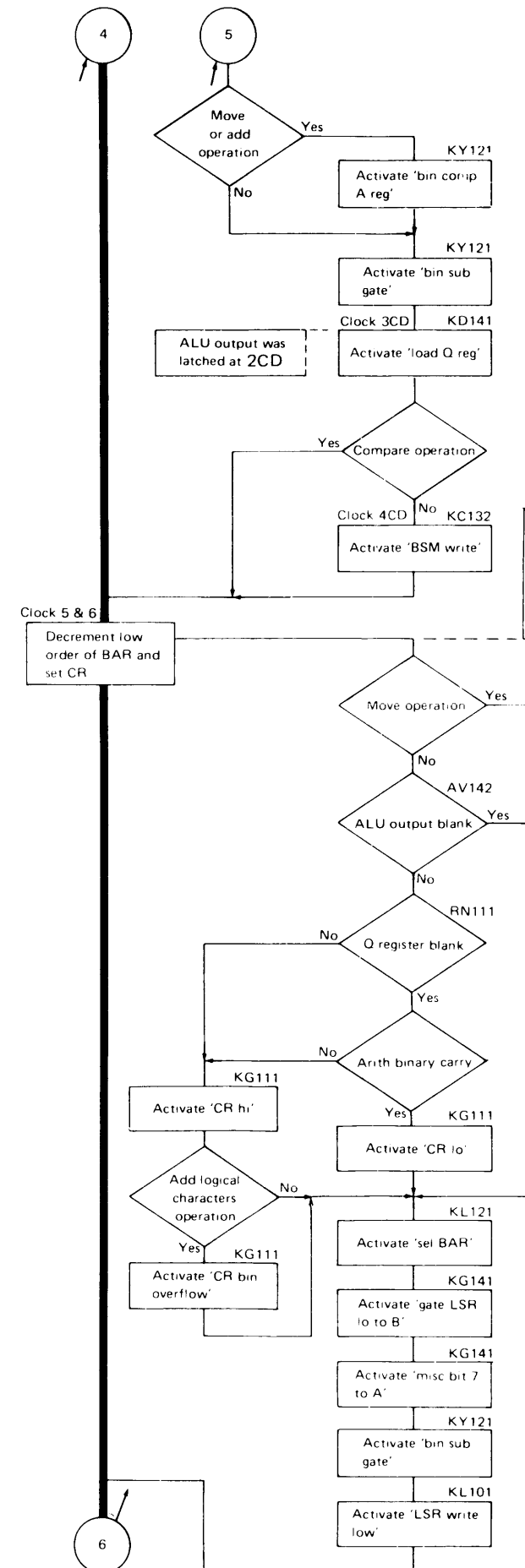
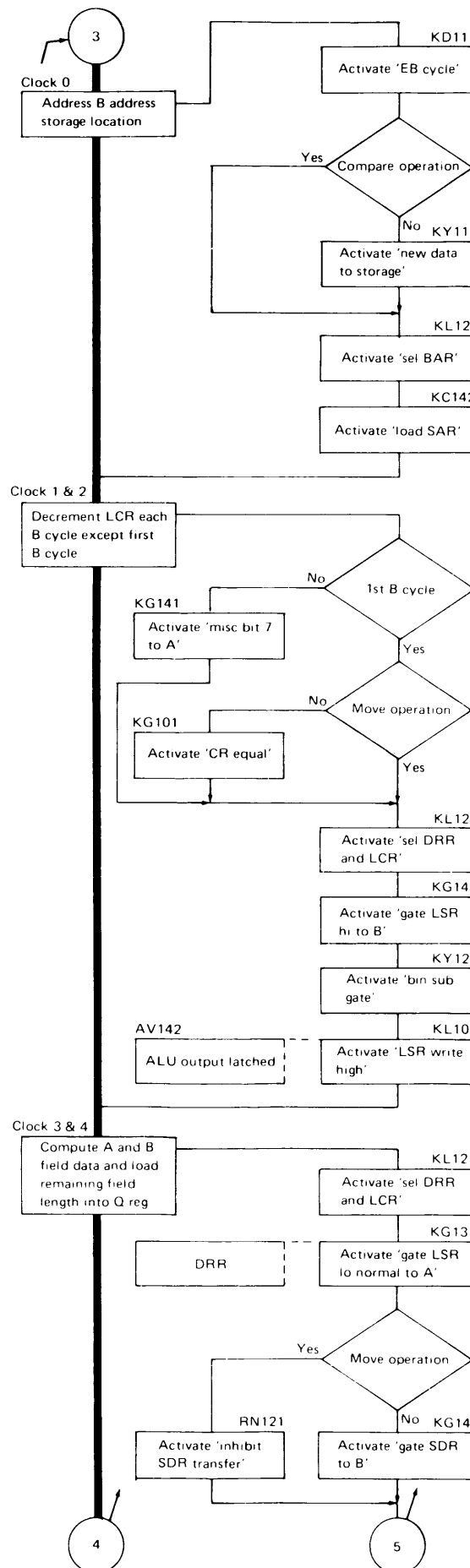
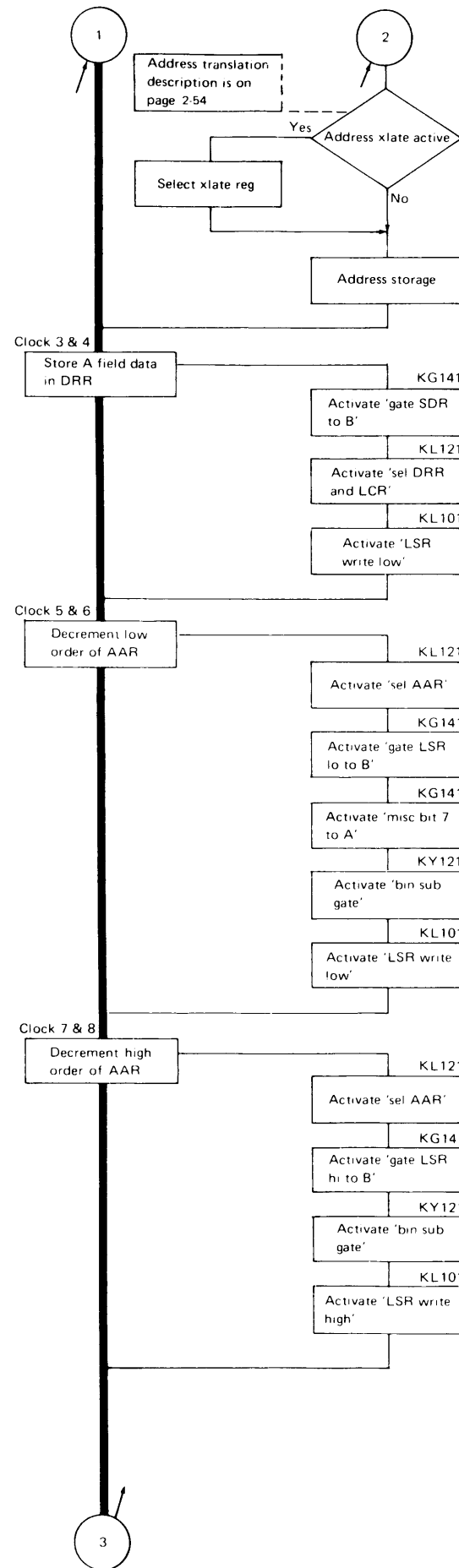
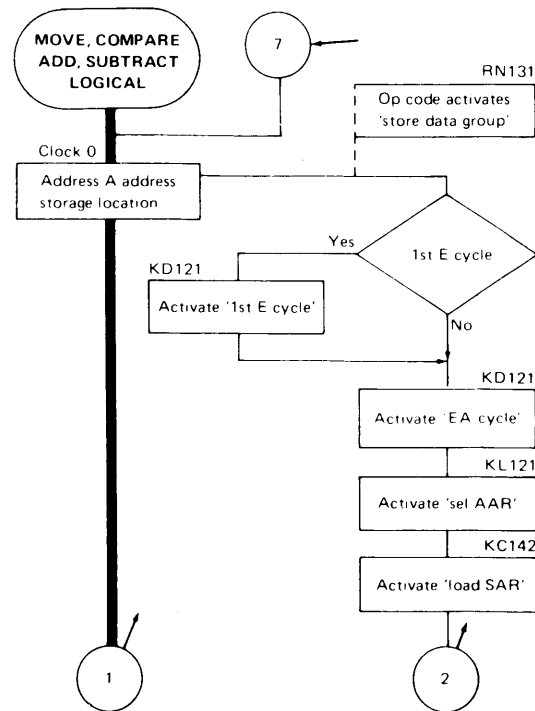
The AAR, the BAR, and the LCR are decremented the same way and the operation ends in the same manner (Q register all zeros). However, the condition register setting is not changed.



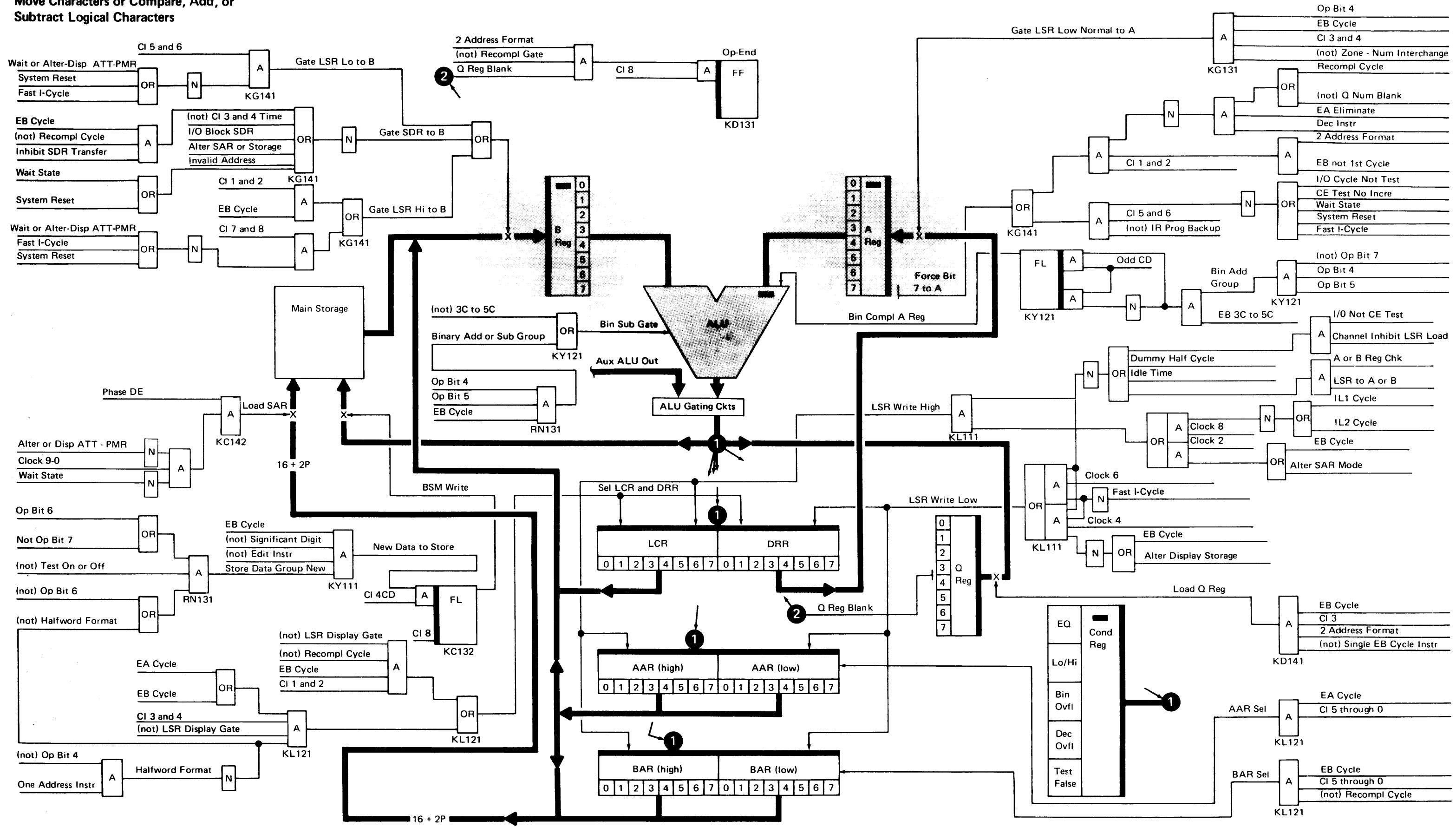
# Move Characters or Compare, Add, or Subtract Logical Characters

## Objectives:

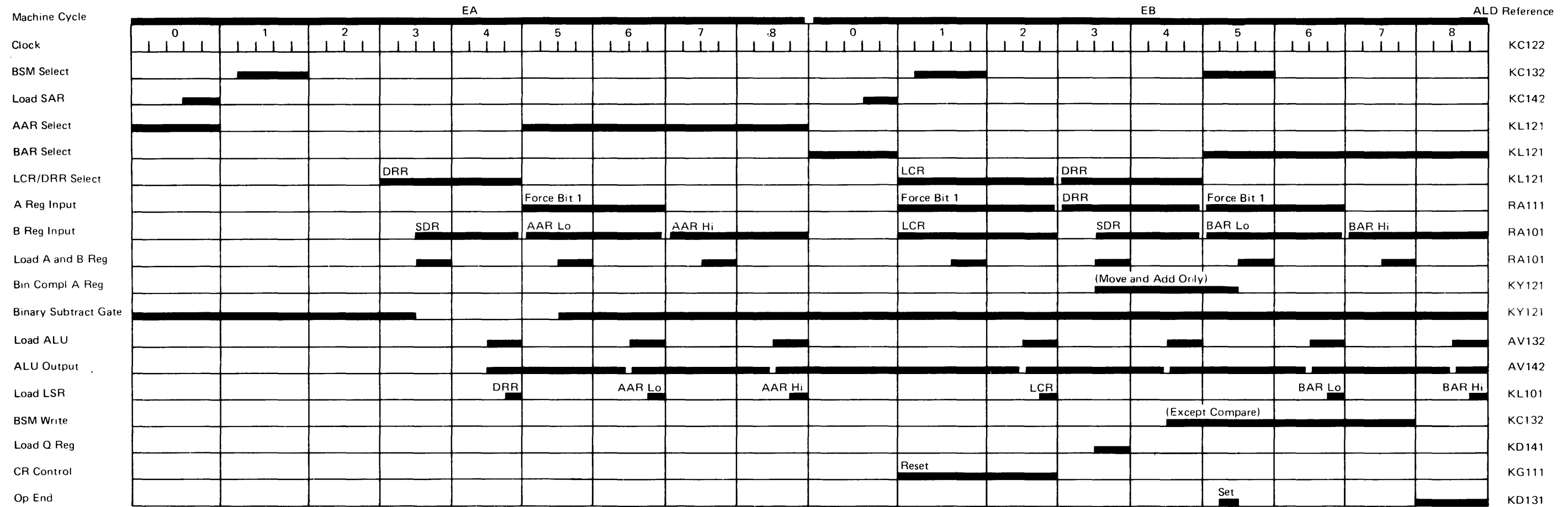
- Move Characters**  
 Bits 0 1 2 3 4 5 6 7  
 Op Code x x x x 1 1 0 0  
  - Move the contents of the A field to the B field.
- Compare Logical Characters**  
 Bits 0 1 2 3 4 5 6 7  
 Op Code x x x x 1 1 0 1  
  - Compare the A field data with the B field data.
- Add Logical Characters**  
 Bits 0 1 2 3 4 5 6 7  
 Op Code x x x x 1 1 1 0  
  - Binary add the A field data to the B field data and store results in B field location.
- Subtract Logical Characters**  
 Bits 0 1 2 3 4 5 6 7  
 Op Code x x x x 1 1 1 1  
  - Binary subtract the A field data from the B field data and store results in B field location.
- All Operations**  
  - A and B fields are same length.
  - Length of field is Q code plus 1.
  - Set condition register except during move operation.



### Move Characters or Compare, Add, or Subtract Logical Characters



**Move Characters or Compare, Add, or Subtract Logical Characters**



Condition Register				
Operation	Equal	Low	High	Binary Overflow
Move				
Add	If result is zero	If result not zero and a high order carry	If result not zero and no high order carry	Result too large for field (no high order carry)
Subtract	If A field equals B field	If B field is lower than A field	If B field is higher than A field	
Compare	If A field equals B field	If B field is lower than A field	If B field is higher than A field	

**Add or Subtract Zoned Decimal—AZ—SZ**

- Decimal add A field data to B field data for add instruction with like signs and subtract instruction with unlike signs.
- Decimal subtract A field data from B field data for add instruction with unlike signs and subtract instruction with like signs.
- Signs are in zone portion of low order bytes.
- Length of A field is a numeric portion of Q code plus 1.
- B field is longer than A field by amount in zone portion of Q code.

Although the add zoned decimal and subtract zoned decimal operations have different operation codes, the execution of each operation is the same depending upon the signs of the two fields. For instance, an add instruction with unlike signs for the two fields (1 minus and 1 plus) actually subtracts the A field from the B field. Likewise, a subtract instruction with unlike signs actually adds the A field to the B field. In either instance, the operations are the same except for the add or subtract function of the ALU.

The operations begin with the low order position of each field and continue until the high order position of the B field is reached. First an A-cycle removes the first A field byte from storage and retains it in the DRR. Then a B-cycle removes the first B field byte from storage, transfers it to the B register, and adds or subtracts the registers in the ALU. This process continues until the end of the A field, which can be shorter than the B field, and then B-cycles continue to the end of the B field.

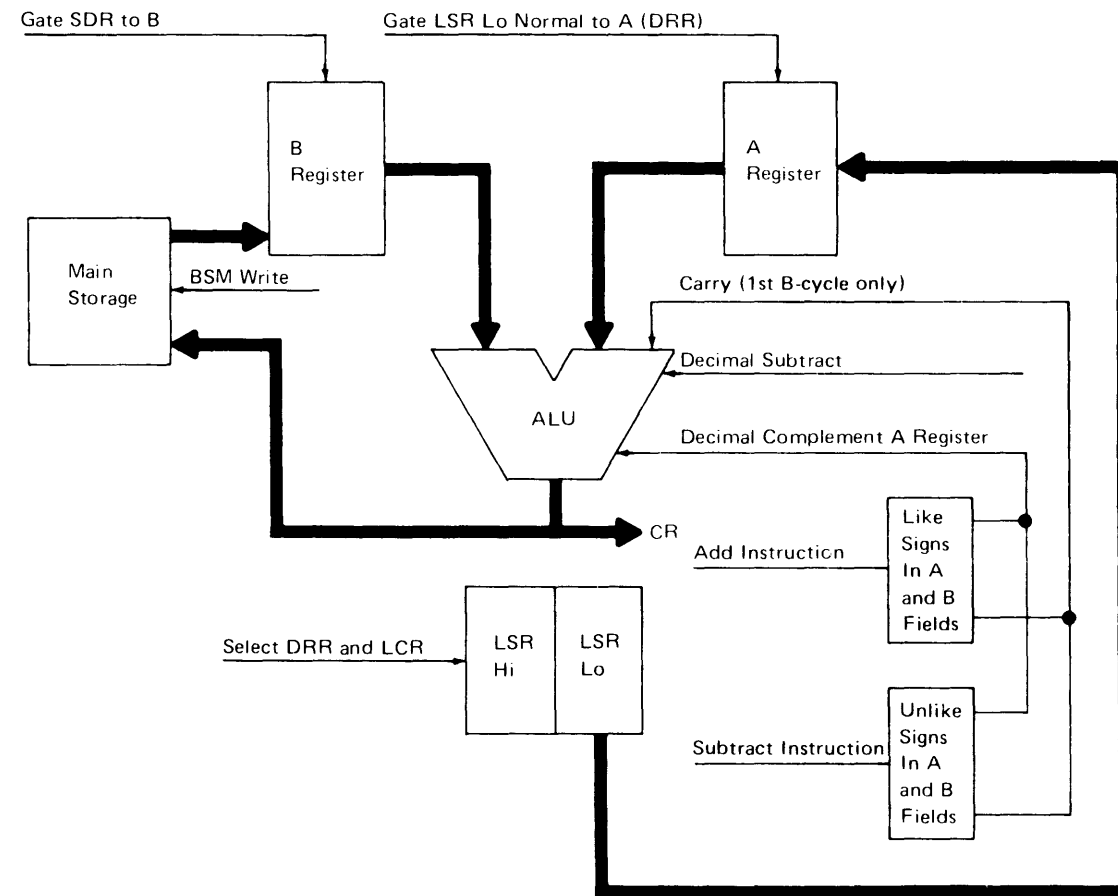
If the result is in true form, nothing more is done with it. If the result is in complement form, the result must be recomplemented. Results are in complement form when:

- Operation has a subtract function (no A register complement) and the A field is larger than the B field.
- Result is minus zero.

Recomplement begins with the low order position of the B field and continues to the high order position. Continuous B-cycles remove the bytes from storage and recomplement them in the ALU.

The numeric portion of the Q code plus 1 is the length of the A field. The B field is longer than the A field by the amount in the zone portion of the Q code.

After an A-cycle has stored the first A field byte in the DRR and the AAR has been decremented the CPU enters into a B-cycle. During the B-cycle, the first B field byte is read from storage and is loaded into the B register. The A field byte is transferred from the DRR to the A register and, depending upon the operation code and the signs of the two fields, the two bytes are either decimal added or subtracted in the ALU.



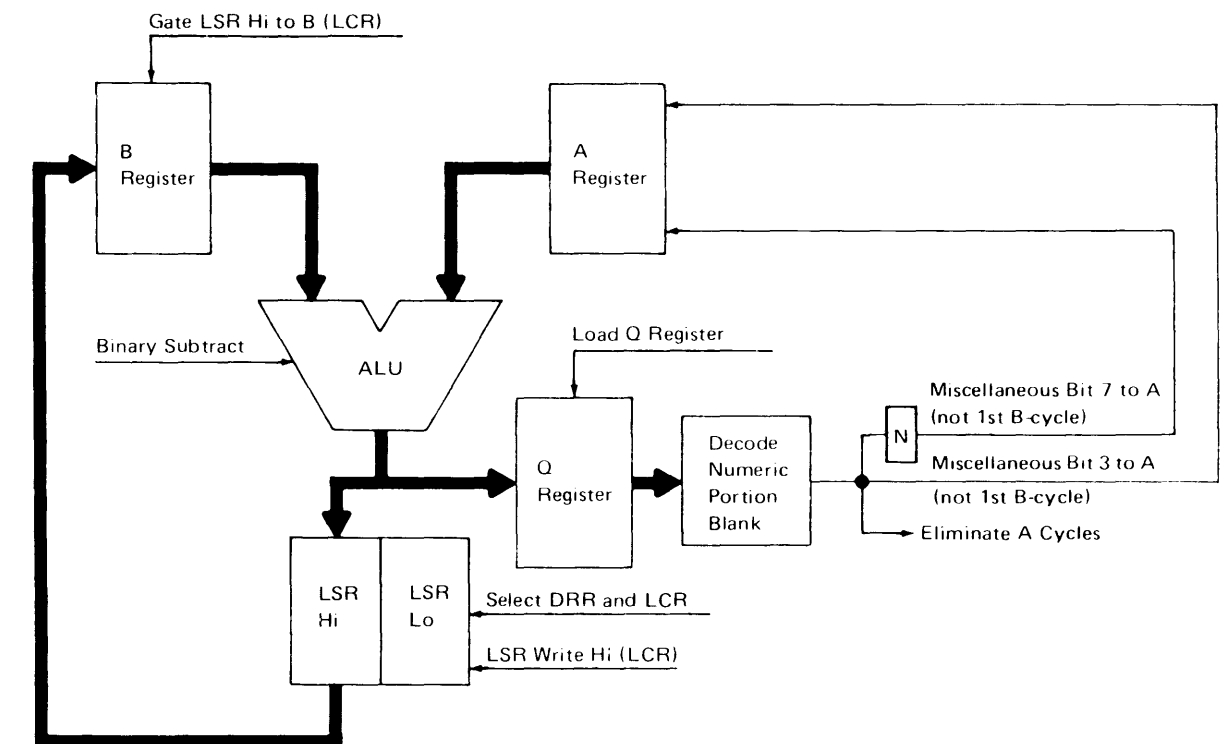
**Add or Subtract Zoned Decimal—Add or Subtract Characters**

The result is then stored in the B field storage location.

The BAR is decremented and the Q register is tested to see if the end of the A field (Q register numeric portion all zeros) or if the end of the B field (Q register all zeros) has been reached. If the numeric portion of the Q register is not all zeros, the CPU takes another A-cycle and another B-cycle to add or subtract the next characters. If the numeric portion is all zeros but the zone portion still contains bits, 'EA eliminate' is activated to block A-cycles and the CPU takes a B-cycle.

If the Q register is all zeros, a check is made to determine if the total is in true or complement form. If the operation function is decimal subtract (A register not complemented) and a carry occurs from the high order byte, 'recomplement cycle' is activated to start recomplementing. Under all other conditions, except a minus zero result, the 'op-end' trigger is turned on and the operation ends. A minus zero result, which is also recomplemented, is determined by the CR setting and is discussed later.

During clock 1 and 2 of each B-cycle, except for the first, the LCR is decremented. The LCR contains the length count that was stored there during the I-Q cycle. The result, which is latched into the ALU at 2CD time, is loaded into the Q register at clock 3 time. Until the Q register numeric portion is all zeros, the LCR is decremented with a 7 bit; after the numeric portion is all zeros, decrementing is done with a 3 bit.



**Decrementing Length Count—Unequal Length Fields**

### Add or Subtract Zoned Decimal (Continued)

During each A- and B-cycle, 'sign control' is activated to provide the EBCDIC code for the sign of each field. The sign is in bit 0-3 of the first byte of each field. EBCDIC sign for minus is 1101 and for plus is 1111. The CPU also recognizes the ASCII-8 code for minus (1011) but 'sign control' changes this to EBCDIC. After the first byte of each field, all zone bits (1111) are provided for each character. During the first B-cycle, the sign of the B-field is entered into storage.

During clock 1 and 2 of the first A-cycle, the condition register is reset to equal. Then in the first B-cycle, the 'CR lo/hi' latch is set by the result sign (lo for minus, hi for plus). However, if no numeric output occurs from the ALU (all zeros), the condition register remains set to equal. If, during any B-cycle, a non-zero ALU output occurs, the result can no longer be equal and the equal condition is reset. The setting of the 'CR lo/hi' latch is then used to determine the CR setting.

If the CR equal condition has not been reset before the last B-cycle and the 'CR lo/hi' latch is set to lo (minus) the result is minus zero. All zero results are considered plus and 'recomplement cycle' is activated to recomplement the results.

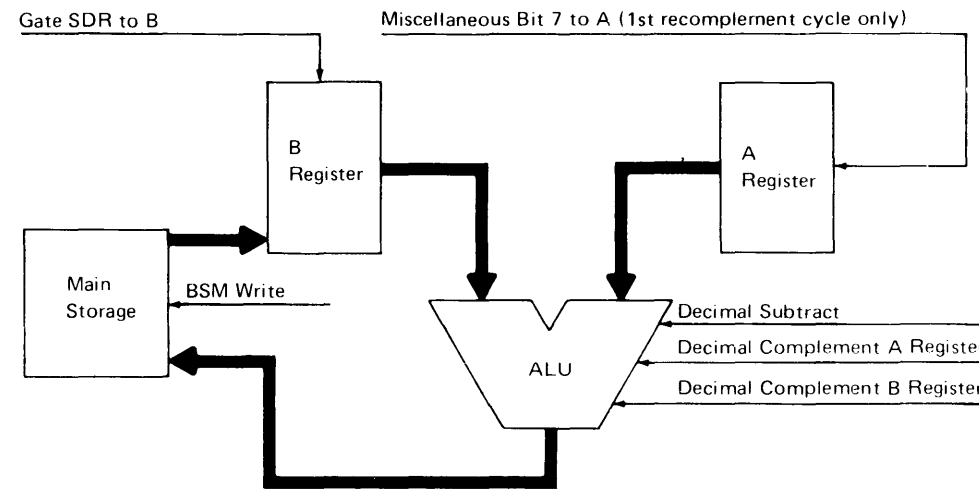
If the operation is an add function (decimal complement A register) and no carry occurs from the high order position, the CR decimal overflow condition is also set. This is an indication that the result is too large to be contained in the B field. The following chart shows the significance of all the CR settings.

Con- dition Register	Equal	Low	High	Decimal Over- flow
ALU result	Result is zero	Result is minus	Result is plus	Result too large for field

### Recomplement Cycles

Addressing for recomplement cycles is controlled by the ARR which contains the low order address of the B field (refer to I-H1 and I-L1 cycles). The ARR is decremented each recomplement cycle in the same manner that the AAR and BAR are decremented in other operations.

'EA eliminate' is active through the entire recomplement operation causing continuous B-cycles. Each byte is read from storage and loaded into the B register.



### Recomplementing

The A register has a 1 forced into it on the first recomplement cycle and is left with all zeros for the remainder of the cycles. Both the A and B registers are decimal complemented.

The length of the field is determined by the LCRR which was loaded during the I-Q cycle. Decrementing of the LCRR is the same as for the LCR in a decimal add or subtract operation.

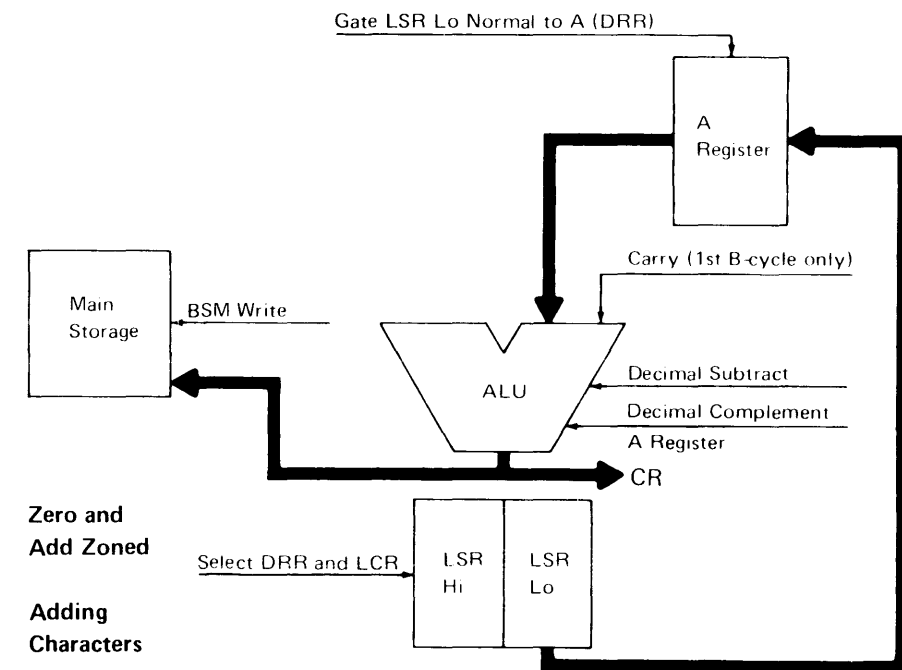
The Q register is tested each cycle to see if the end of the field has been reached (all zeros in the Q register). When the Q register is all zeros, the 'op-end' trigger is turned on and the operation ends.

Because recomplement is necessary only when the A field is larger than the B field or the result is minus zero, 'sign control' is activated to reverse the sign of the result. The 'CR lo/hi' latch is reset on the first cycle and the CR setting is determined in the same way as during decimal add or subtract.

### Zero and Add Zoned-ZAZ

- Decimal add A field data to zeros and place result in B field.
- B field is longer than A field by amount in zone portion of the Q code.

The zero and add zoned operation is similar to an add zoned decimal operation except the function is always add, without consideration of the fields signs. Another difference is that the B field characters are not loaded into the B register.



### Zero and Add Zoned

With the B register all 0's, the operation is the same as adding the A field to zero. The only other significant difference is that the A field sign is entered as the sign of the result instead of the B field sign.

The address registers and the LCR are decremented in the same way. The operation ends in the same manner. Recomplementing is not necessary unless the result is minus zero. The CR settings are shown in the chart below.

Con- dition Register	Equal	Low	High	Decimal Over- flow
ALU result	Result is zero	Result is minus	Result is plus	Result too large for field

### Zero and Add Zoned and Add or Subtract Zoned Decimal

**Objectives:**

1. Zero and Add Zoned

Bits	0	1	2	3	4	5	6	7
Op Code	x	x	x	x	0	1	0	0

- Decimally add A field data to zeros and place results in B field location.

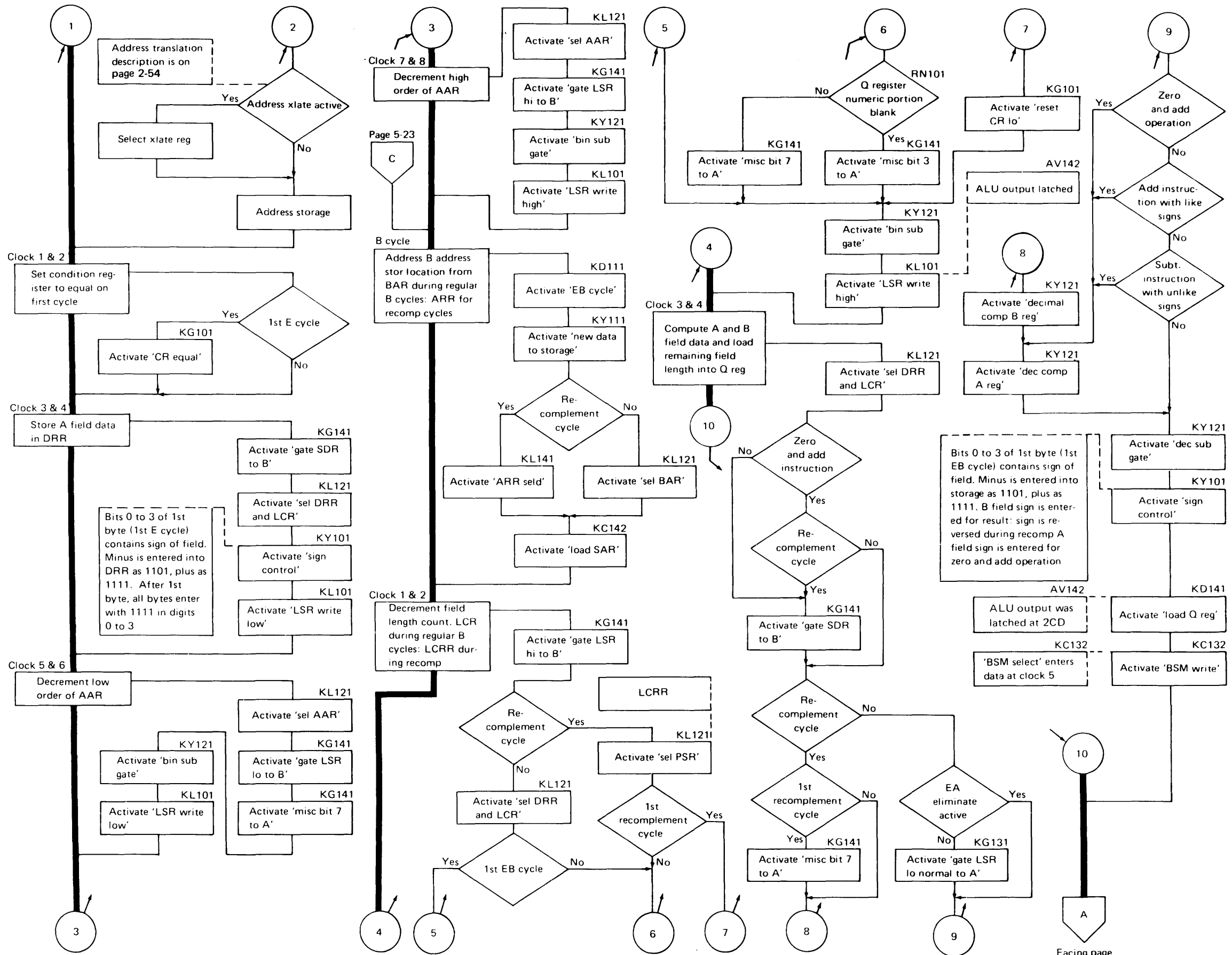
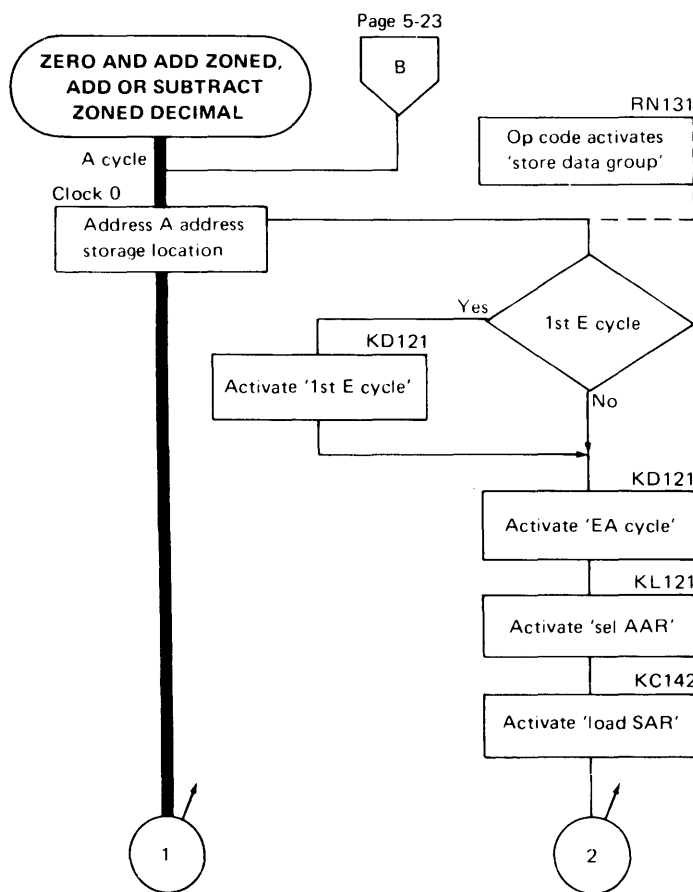
2. Add Zoned Decimal  
Subtract Zoned Decimal

Bits	0	1	2	3	4	5	6	7
Op Code	x	x	x	x	0	1	1	0
	x	x	x	x	0	1	1	1

- Decimally add A field data to B field or subtract A field data from B field.
- Instruction and signs of fields determine add or subtract function.

3. All Operations

- Length of A field is numeric portion of Q code + 1.
- B field is longer than A field by amount in zone portion of Q code.



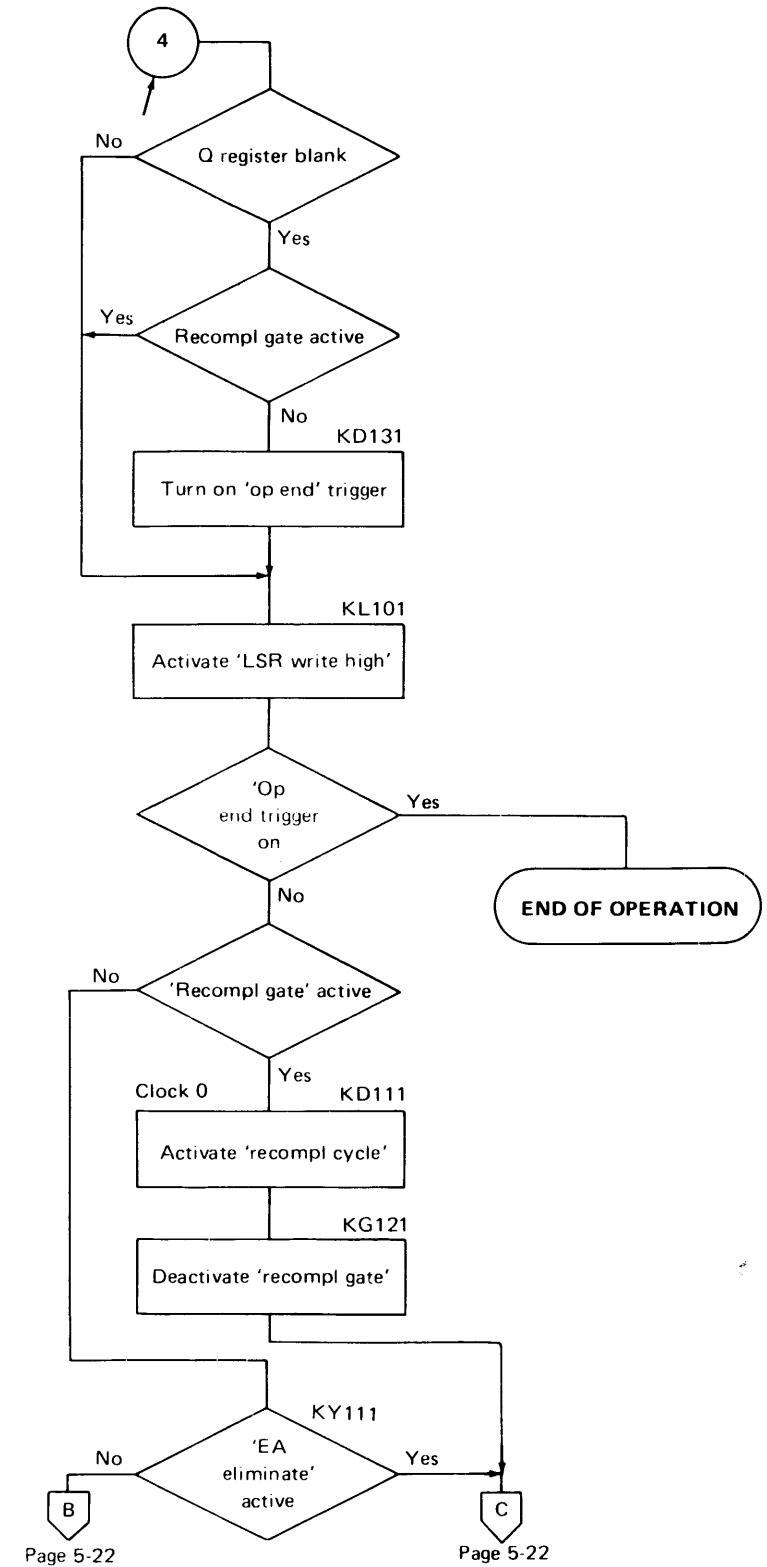
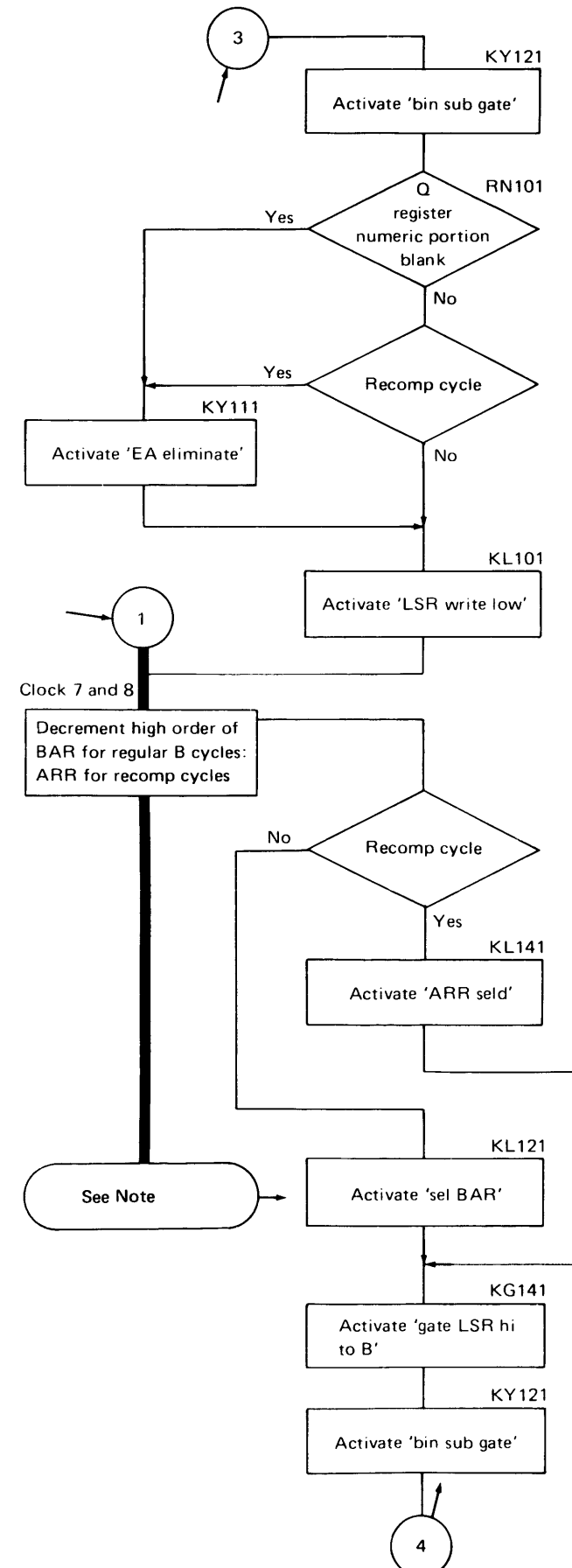
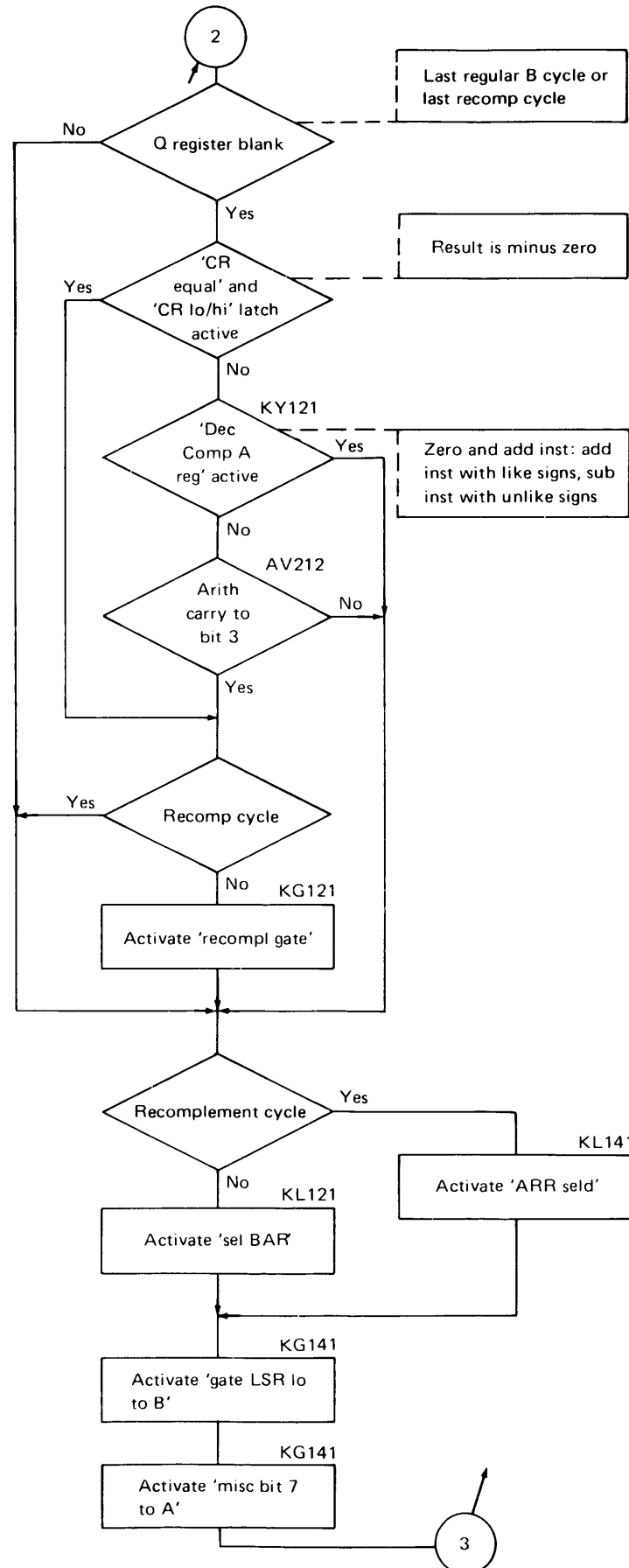
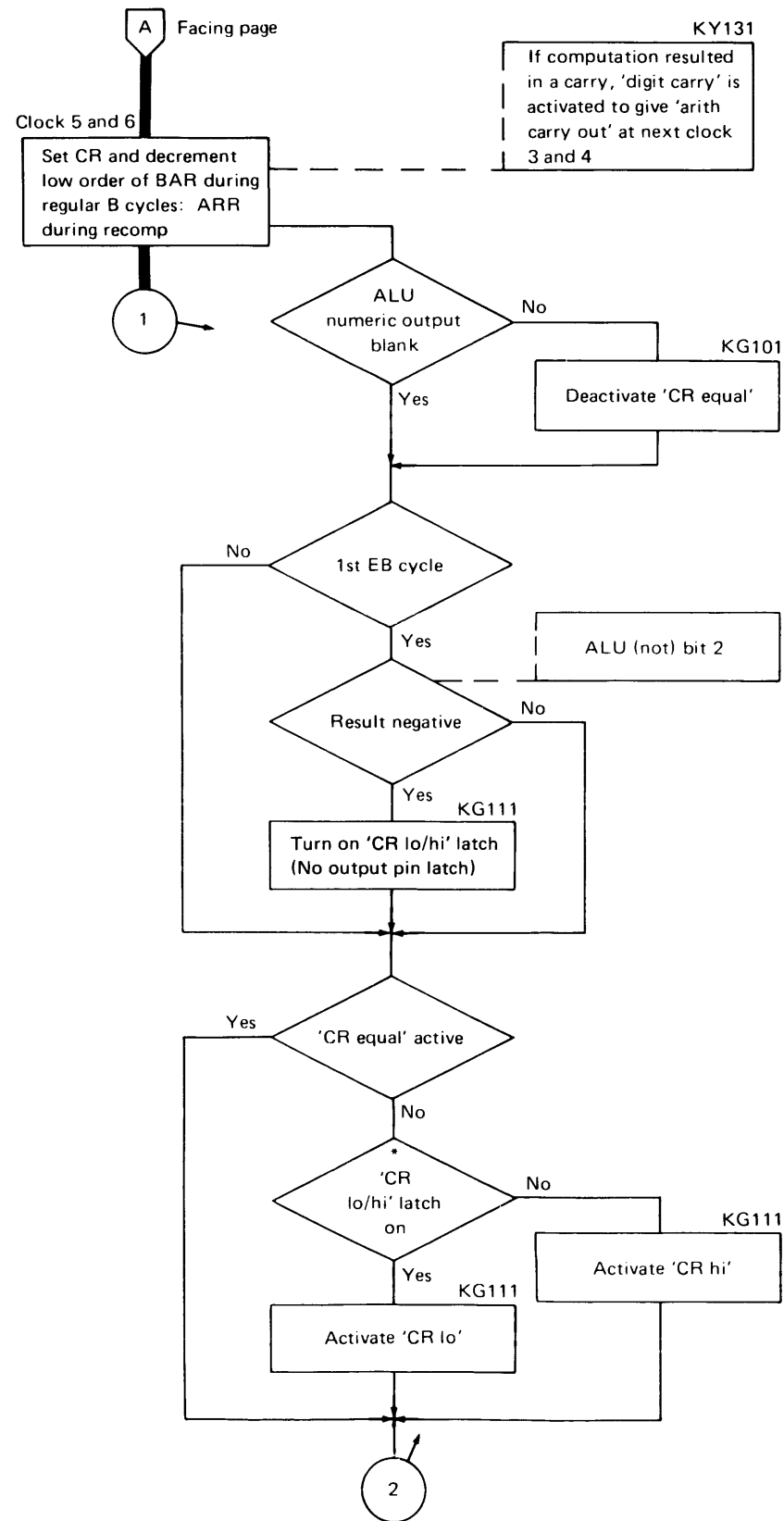
Bits 0 to 3 of 1st byte (1st EB cycle) contains sign of field. Minus is entered into storage as 1101, plus as 1111. After 1st byte, all bytes enter with 1111 in digits 0 to 3.

Bits 0 to 3 of 1st byte (1st EB cycle) contains sign of field. Minus is entered into storage as 1101, plus as 1111. B field sign is entered for result: sign is reversed during recomp A field sign is entered for zero and add operation.

ALU output was latched at 2CD

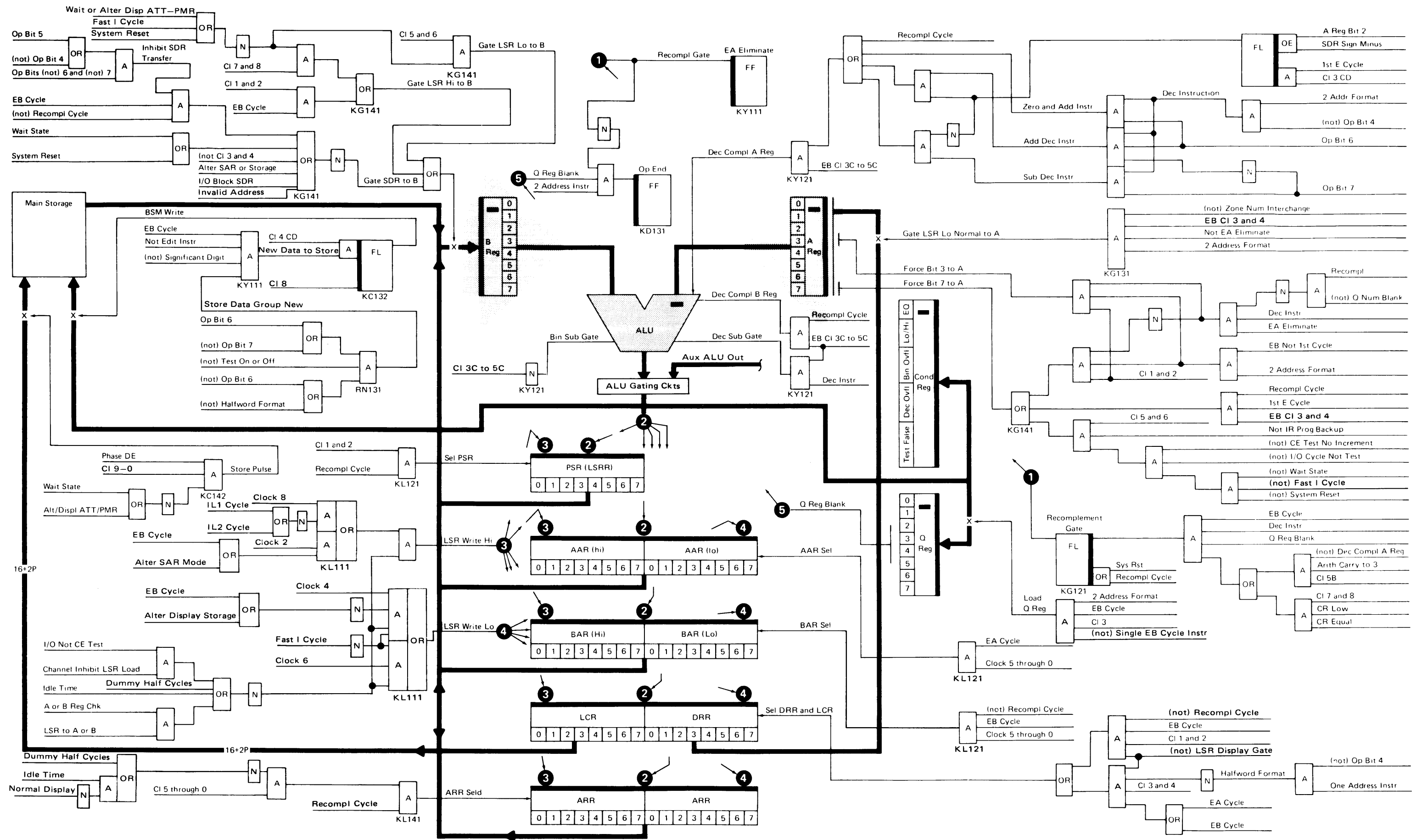
'BSM select' enters data at clock 5

**Zero and Add Zoned and Add or Subtract Zoned Decimal**



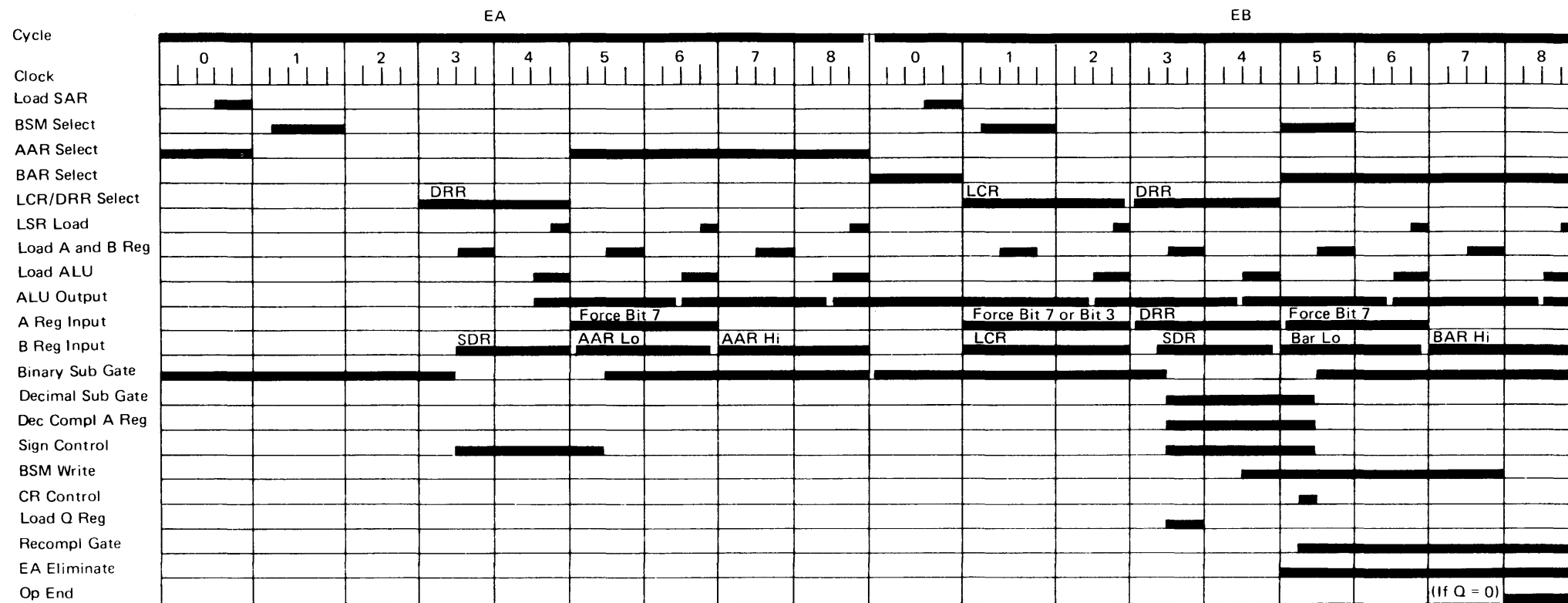
*Note:* Operation repeats A and B cycles until end of A field (Q register numeric portion blank). 'EA eliminate' then allows B cycles until end of B field (Q register blank). Operation ends unless result is in complement form. To recompute, the low order of the B field is established by the ARR and the length of the field is established by the LCRR. 'EA eliminate' then allows B cycles until end of the field (Q register blank). Operation ends.

### Zero and Add Zoned and Add or Subtract Zoned Decimal





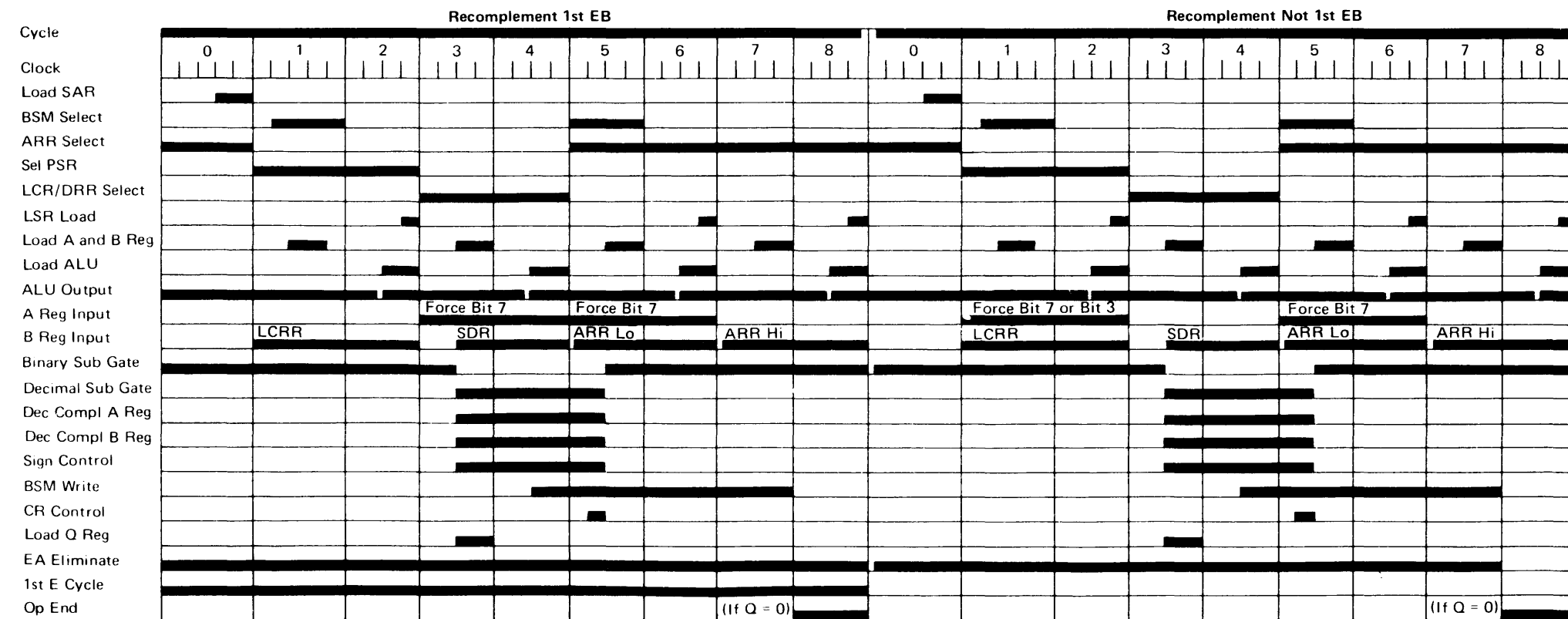
Zero and Add Zoned and Add or Subtract Zoned Decimal



ALD References

- KC122
- KC142
- KC132
- KL121
- KL121
- KL121
- KL101
- RA101
- AV132
- AV142
- RA111
- RA101
- KY121
- KY121
- KY101
- KC132
- KG111
- KD141
- KG121
- KY111
- KD131

Condition Register				
Operation	Equal	Low	High	Decimal Overflow
Zero and Add Zoned	Result is zero	Result is minus	Result is plus	_____
Add or Subtract Zoned	Result is zero	Result is minus	Result is plus	Result is too large for field

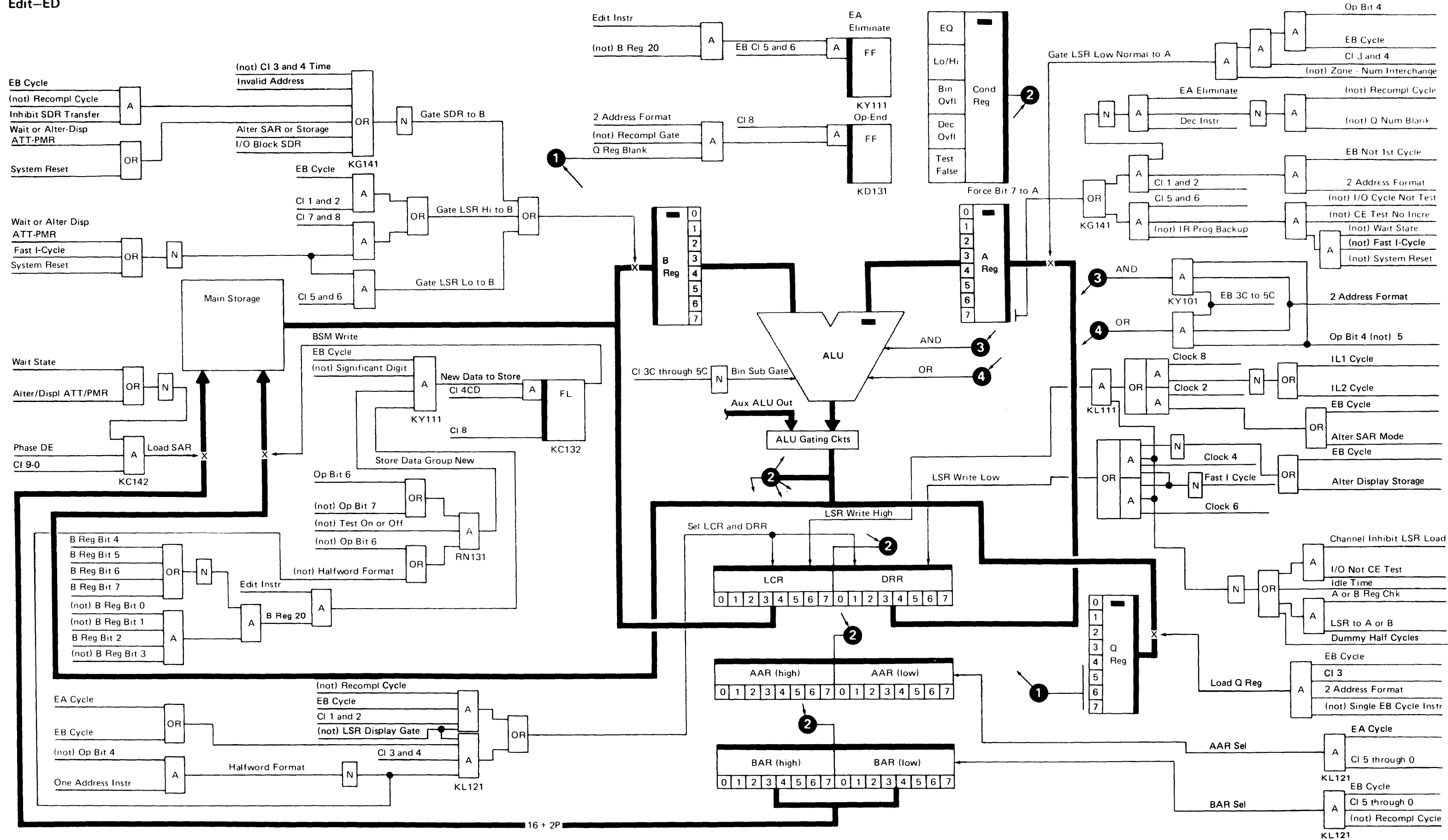


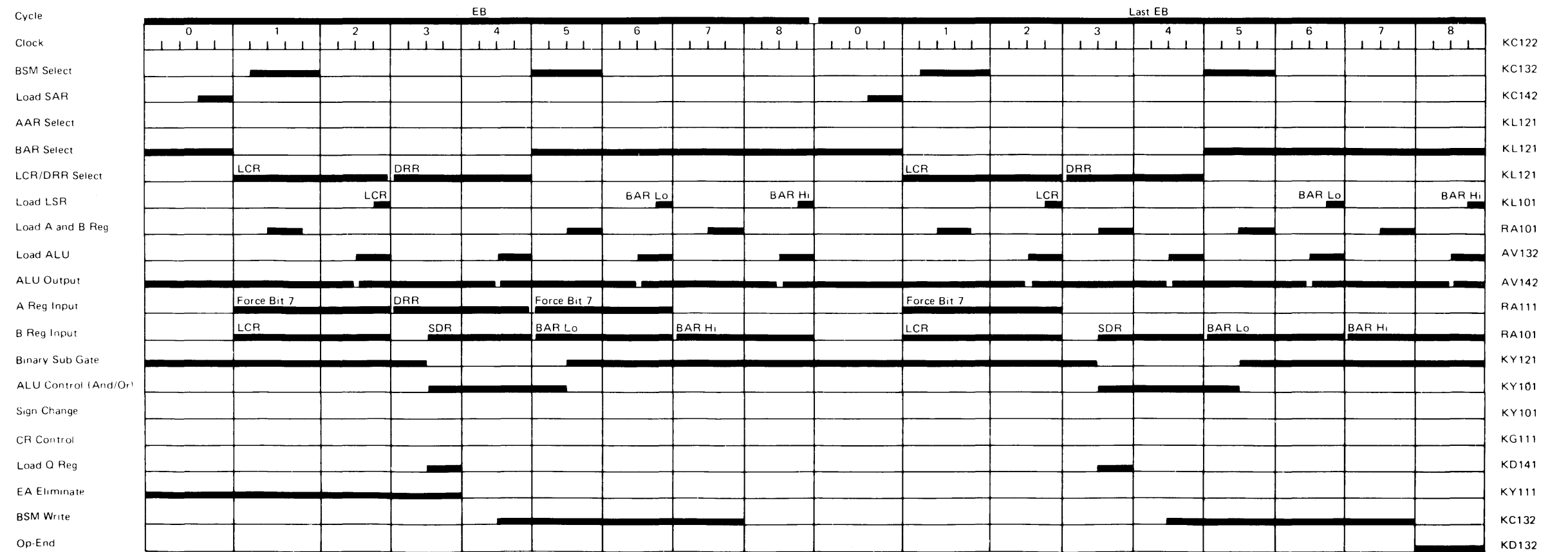
- KC122
- KC142
- KC132
- KL141
- KL121
- KL121
- KL101
- RA101
- AV132
- AV142
- RA111
- RA101
- KY121
- KY121
- KY121
- KY121
- KY101
- KC132
- KG111
- KD141
- KY111
- KD111
- KD131





Edit-ED





**Insert and Test Character—ITC**

Bits 0 1 2 3 4 5 6 7  
Op Code x x x x 1 0 1 1

- Replace all characters to left of first significant digit in B field with A field character.
- Only numeric characters 1 through 9 are considered significant digits.

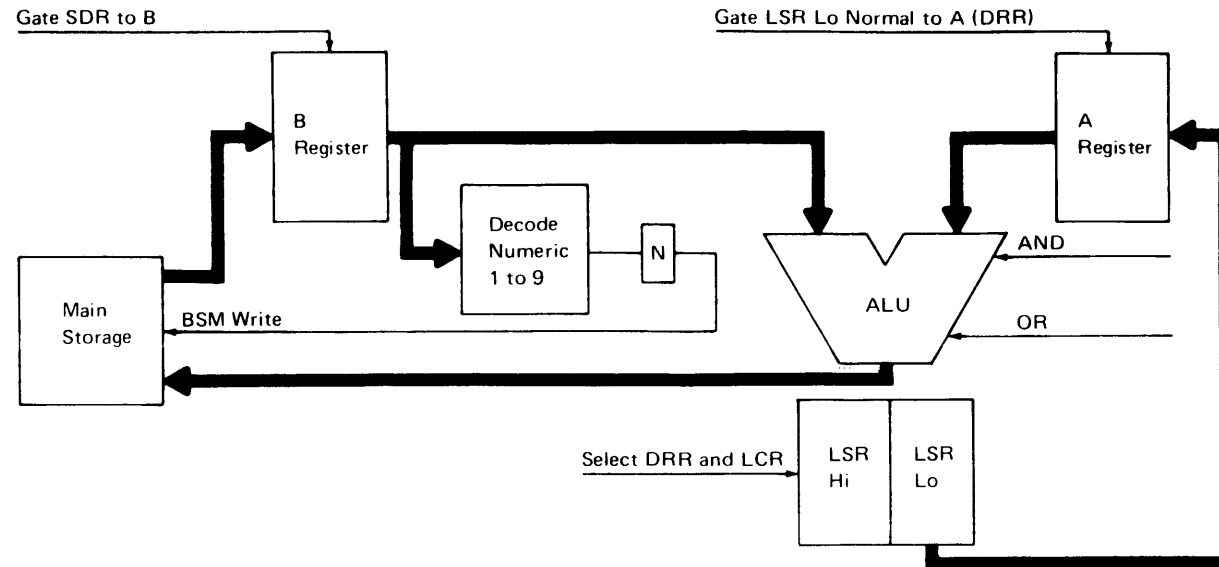
The insert and test character operation inserts a single A field character into each B field position to the left high order significant digit. Only numeric characters 1 through 9 are considered to be significant digits.

An example of an insert and test character operation is:

A field character \*  
B field before operation \$001.98  
B field after operation \$\*\*1.98

The B field starting address is the high order position and the operation continues until either, the low order position of the field is reached, or a significant digit is found. The B field length is one more than the Q code.

After the A field byte has been stored in the DRR the CPU enters into a B-cycle. The B field high order byte is read from storage and is loaded into the B register. The A field byte is transferred from the DRR to the A register and the 'AND' and 'OR' ALU control lines are activated to move the A field byte through the ALU.



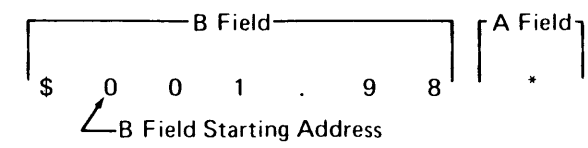
If the B register contains a character other than numeric 1 through 9, the A field byte is stored in the B field high order location.

The BAR is then incremented in the same manner that the IAR is incremented during I-cycles. 'EA eliminate' prevents the CPU from taking another A-cycle. Another B-cycle reads the next descending B field position from storage and it is checked in the same manner.

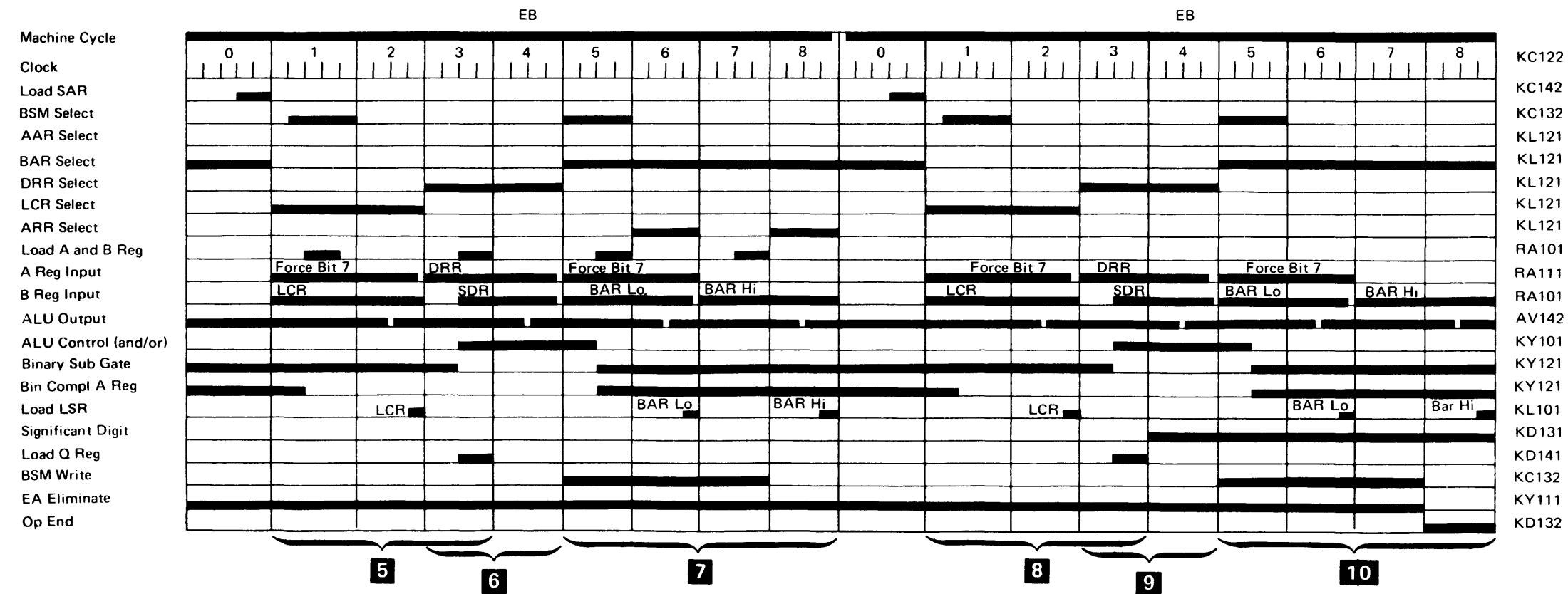
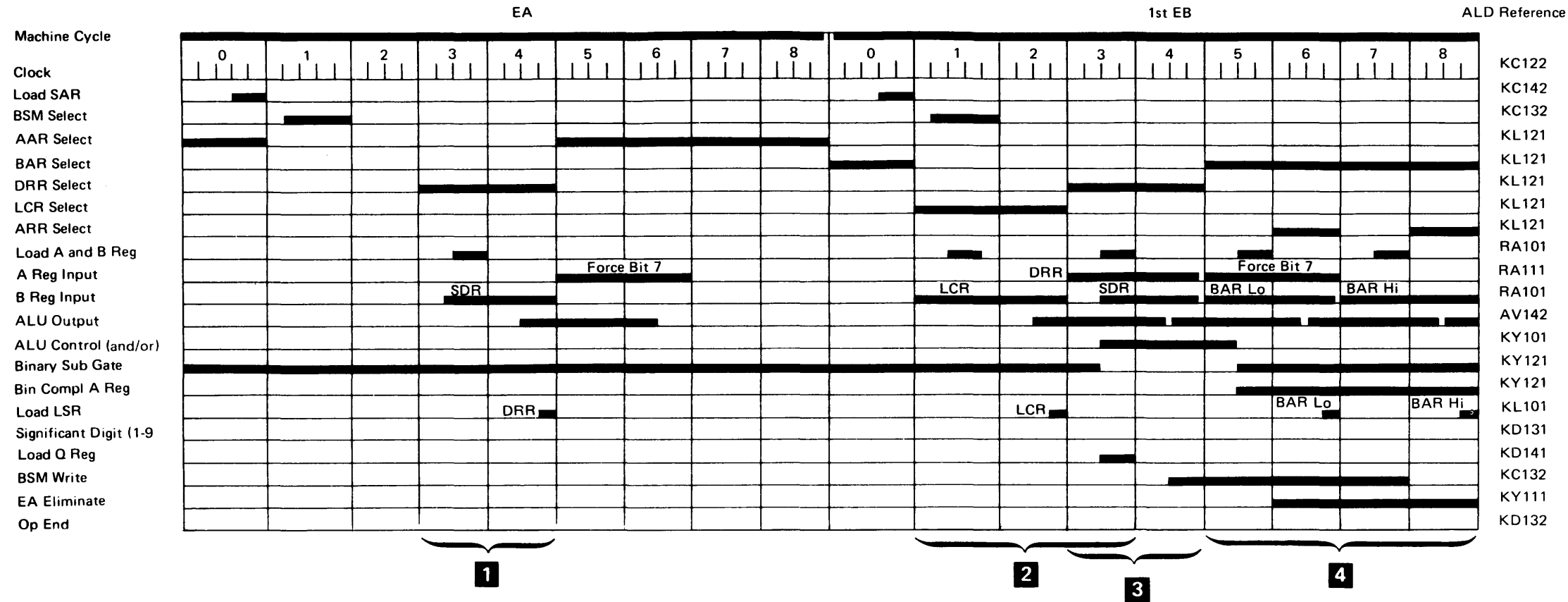
If the B field byte contains a significant digit the B field byte is regenerated back into storage and the 'op-end' trigger is turned on and the operation ends. Meanwhile, the LCR is decremented each B-cycle, except the first B-cycle, and is transferred to the Q register. If no significant digit is found before the length count is reduced to zero, the all zero Q register ends the operation.

Each B-cycle in which no significant digit is found, the address of the next B field byte is stored in the ARR for programming purposes. At the end of the operation, the ARR will contain the address of the first significant digit. If no significant digit is encountered, the ARR will contain the address of the byte to the right of the B field.

Action taken	Reference to timing chart on facing page
* to DRR-----	<b>1</b>
Length Count to LCR un- changed, also to Q Register	<b>2</b>
* to B Field (\$*01.98)-----	<b>3</b>
Increment BAR-----	<b>4</b>
Decrement length count and store in LCR and Q Register	<b>5</b>
* B Field (\$**1.98)-----	<b>6</b>
Increment BAR-----	<b>7</b>
Decrement length count and store in LCR and Q Register	<b>8</b>
No change in B Field due to significant digit (\$**1.98)	<b>9</b>
Increment BAR-----	<b>10</b>



# Insert and Test Characters – ITC



### Insert and Test Character—ITC

#### Objectives

#### Insert and Test Character

Bits 0 1 2 3 4 5 6 7  
Op Code x x x x 1 0 1 1

- Replace all characters to left of first significant digit in B field with A field character.
- Only numeric characters 1 to 9 are considered significant digits.
- Length of B field is Q code + 1.
- A field is only 1 character in length.

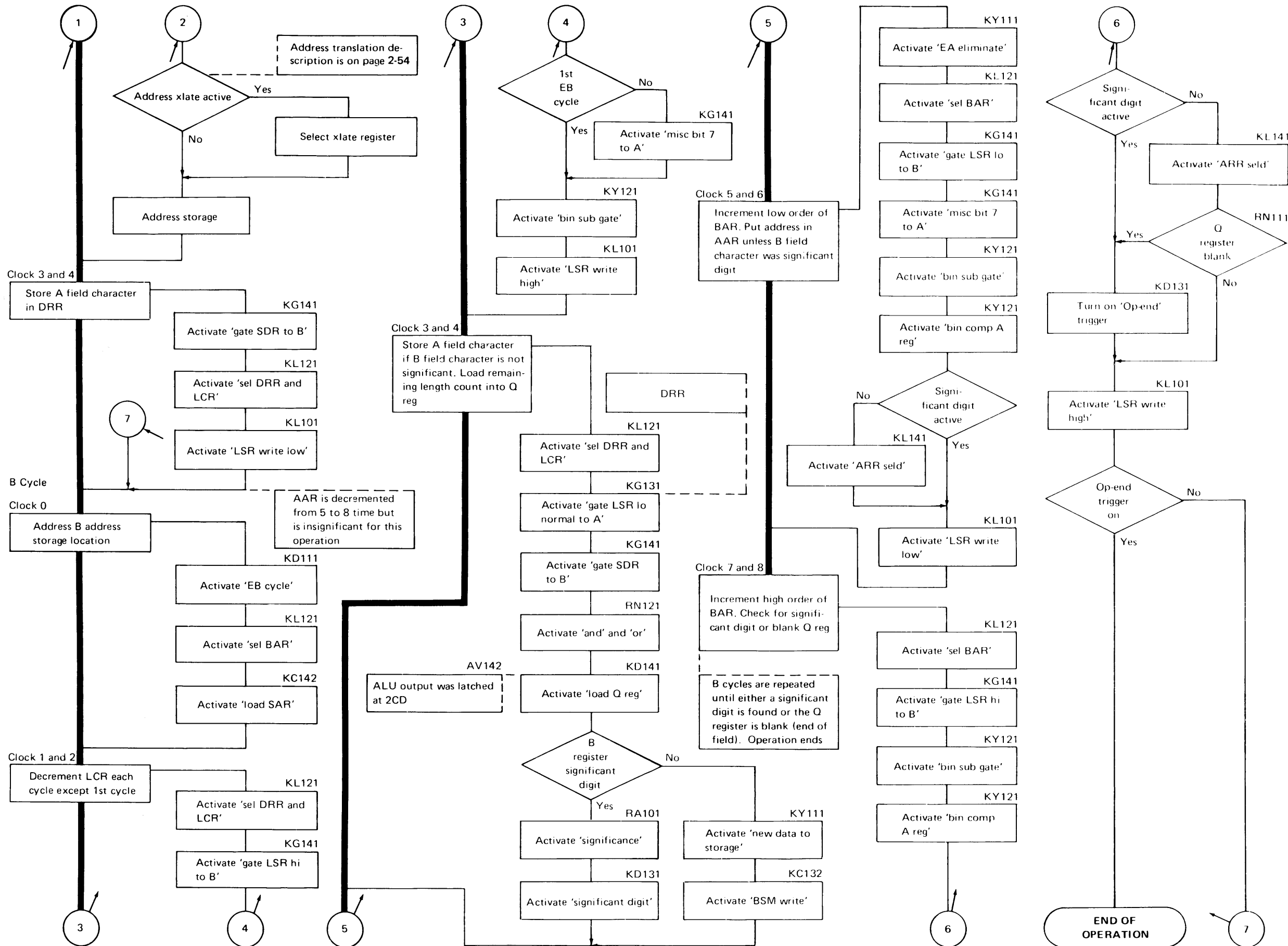
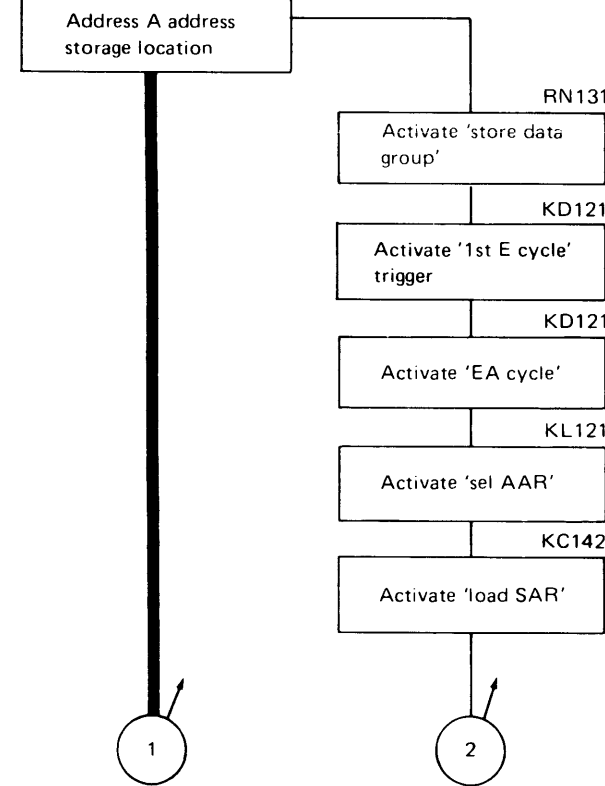
#### Example:

Edited field before operation \$001.98  
A field character \*  
Edited field after operation \$\*\*1.98  
B field starting address

#### INSERT AND TEST CHARACTERS

A Cycle

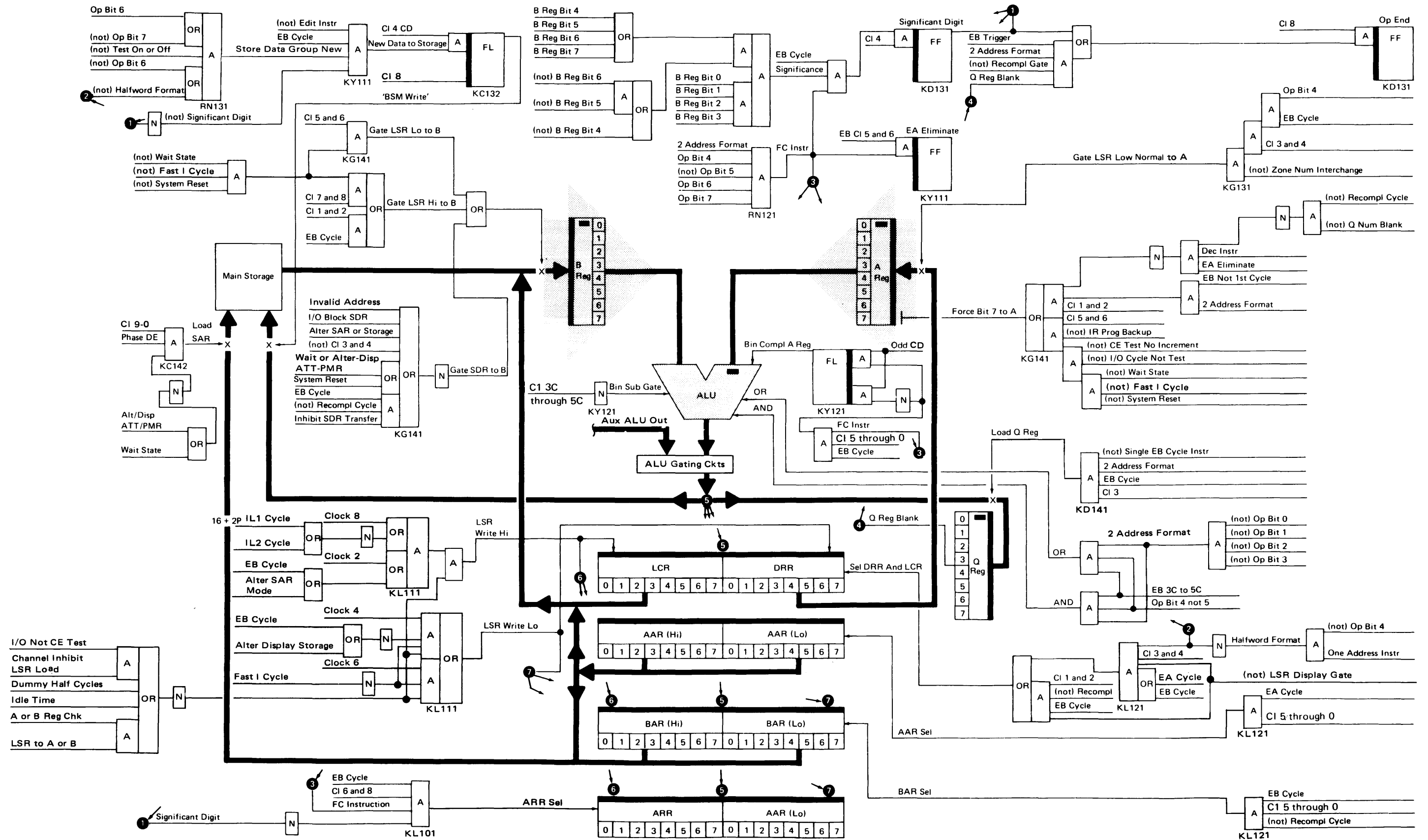
Clock 0



#### END OF OPERATION



Insert and Test Characters – ITC



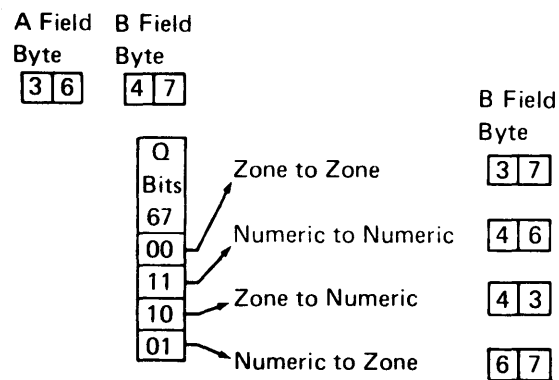
**Move Hex Character – MVX**

Bits 0 1 2 3 4 5 6 7  
Op Code x x x x 1 0 0 0

- Move half (zone or numeric) of A address byte to the numeric or zone portion of the B address byte.
- Do not change the other half of the B address byte.
- Bits 6 and 7 of Q code specify the portion of each byte used.

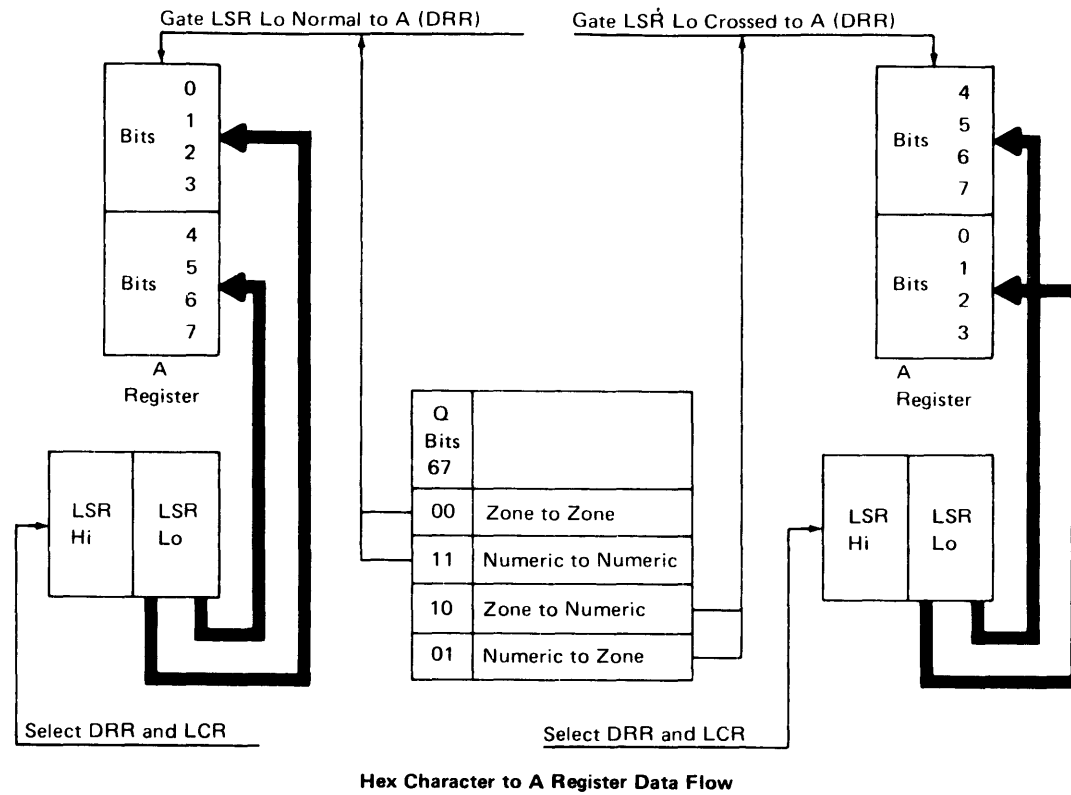
The move hex character operation moves a half byte of information from one main storage location to another. Either half of the A field byte (zone or numeric) may be placed in either portion of the B field byte (zone or numeric) without changing the other half of the B field byte. The type of move is determined by bits 6 and 7 of the Q code. The following example shows the four types of moves possible and gives the Q code bit structure for each type of move. Since each field is only one character in length, one A cycle and one B cycle are all that are required to complete the operation.

**Before Operation**                      **After Operation**

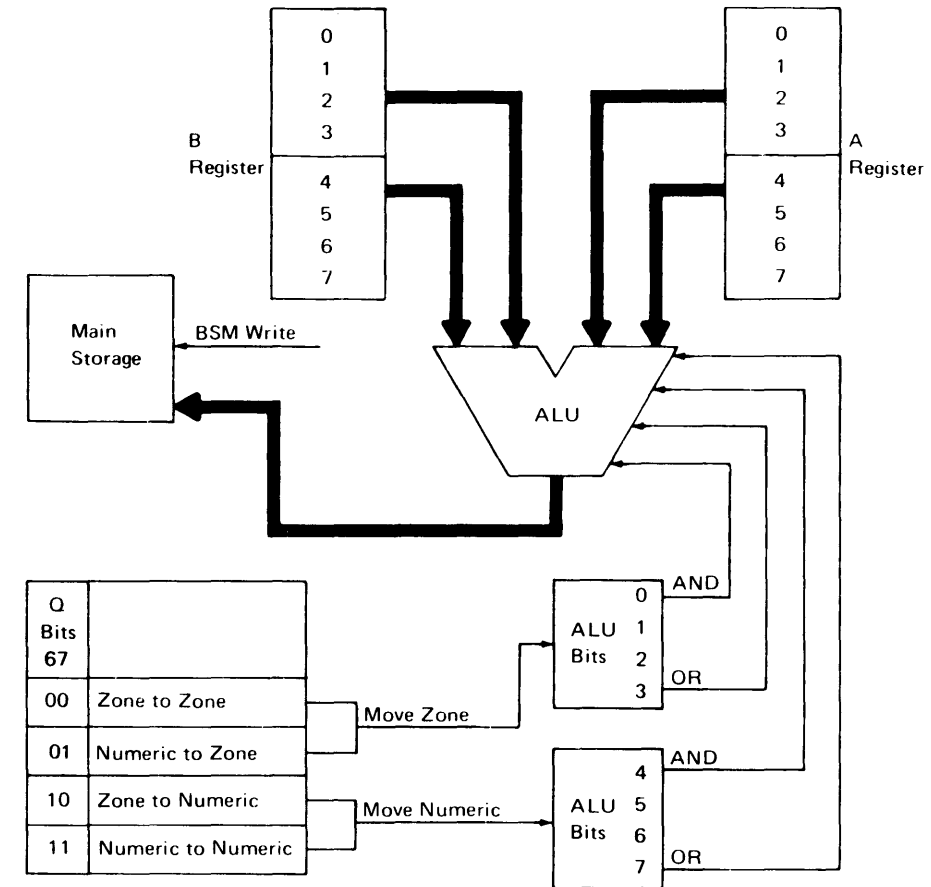


After an A cycle has stored the A field byte in the DRR, the CPU enters into a B cycle. During the B cycle, bits 6 and 7 of the Q code control the data flow.

The following figure shows that if the half byte is to be moved to the same relative position in the B field byte, the bits are transferred from the DRR to the A register in their normal positions. However, if the move is to the opposite portion of the B field byte, the A field byte enters the A register with the zone and numeric portions crossed. For example, a 3 bit enters the A register as a 7 bit, a 2 bit enters as a 6 bit, and so forth.



The portion of the B register byte that is to be changed by the A field half byte determines the ALU controls. The 'AND' and 'OR' ALU control lines are activated only for that portion that is to be changed.



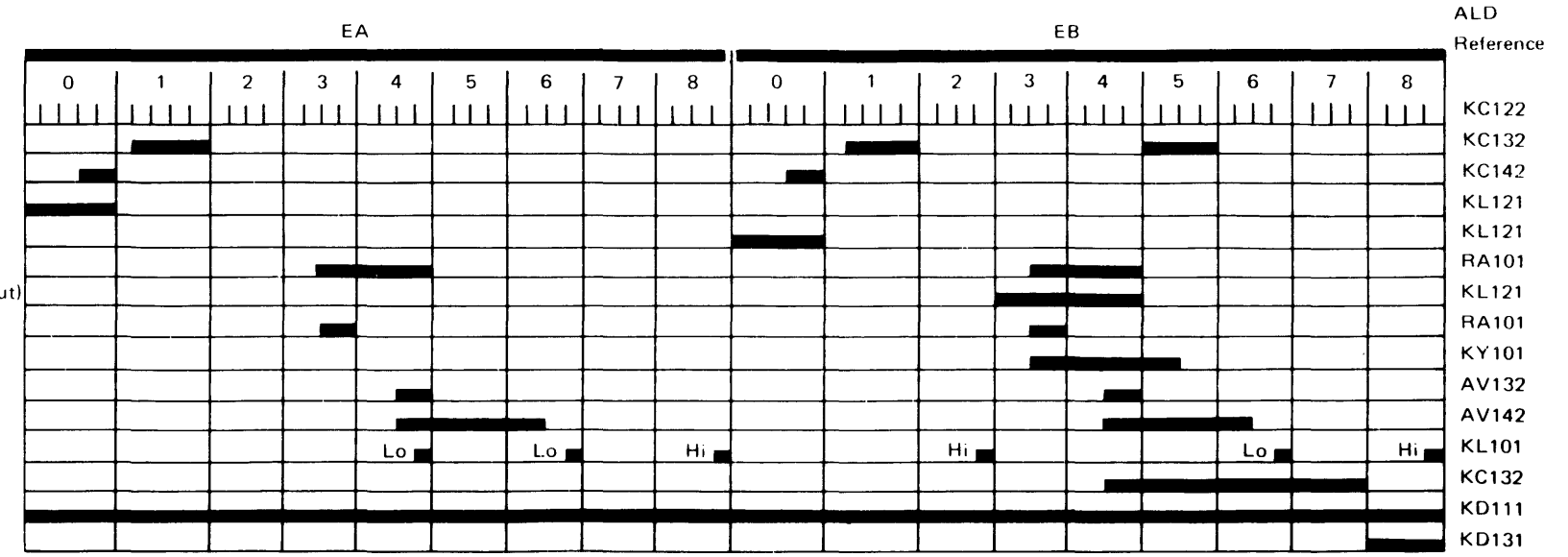
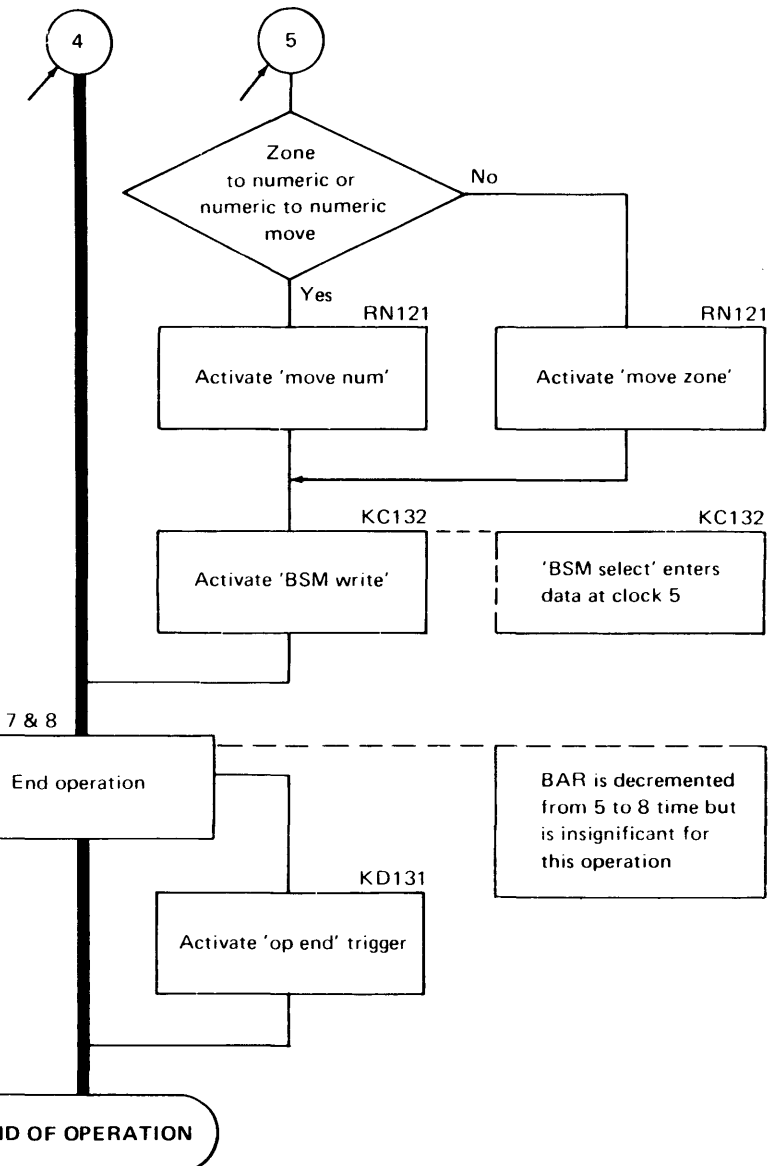
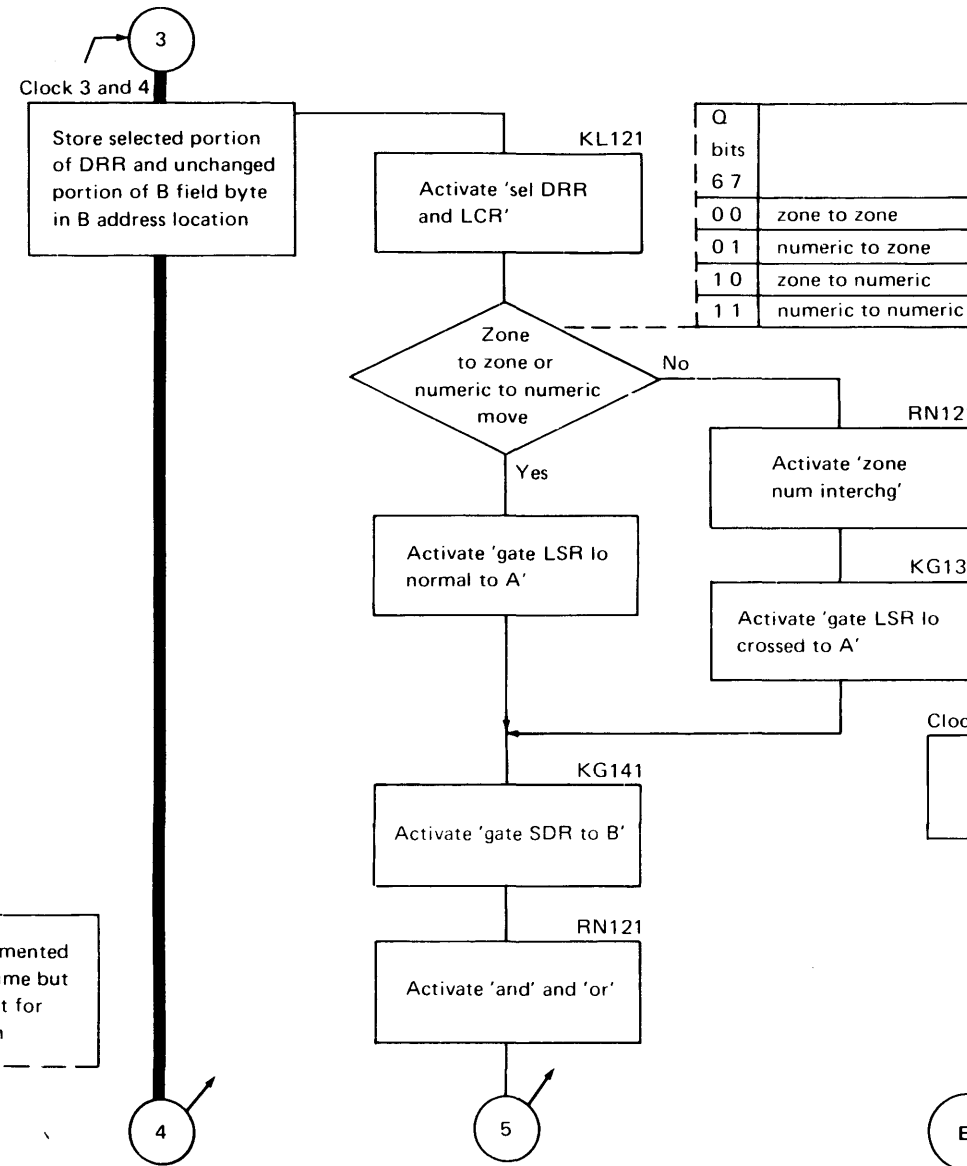
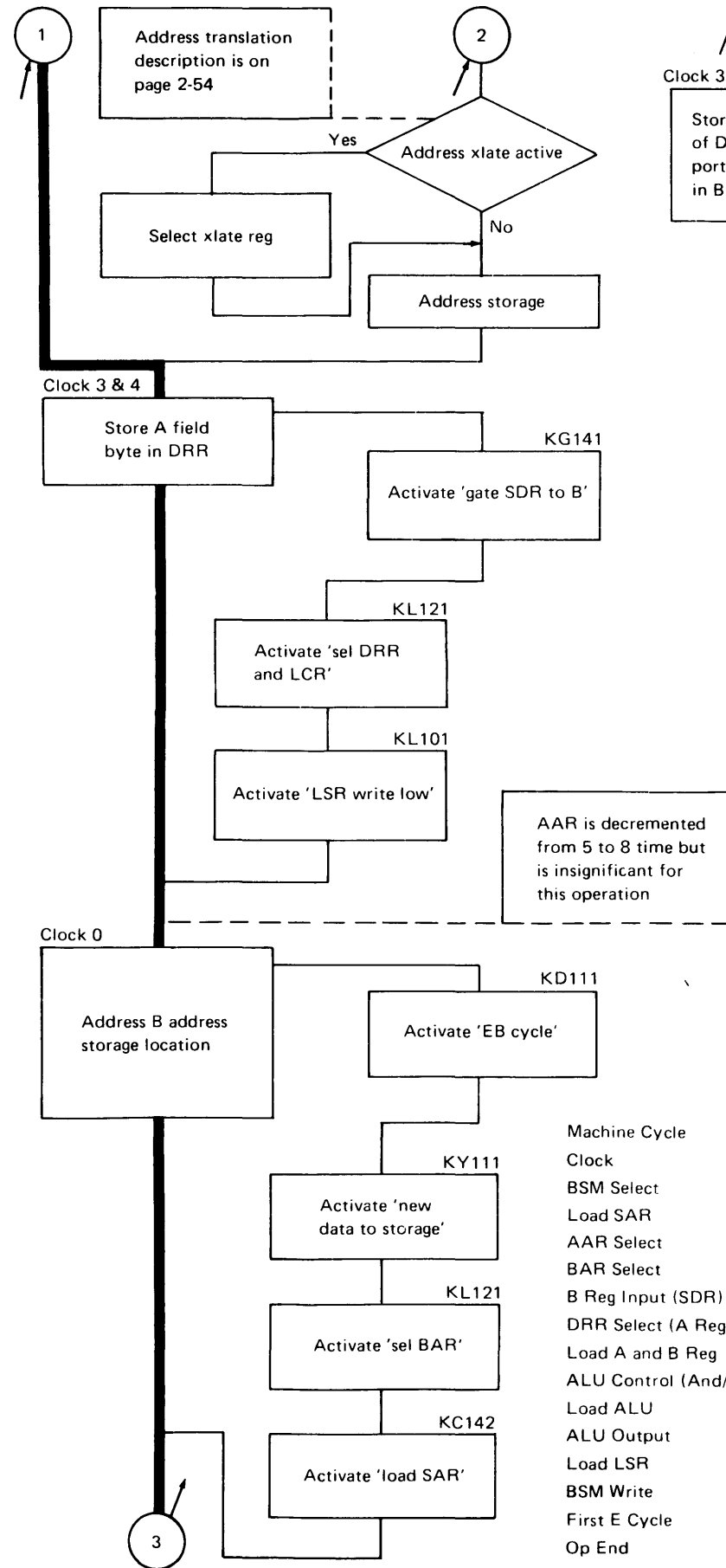
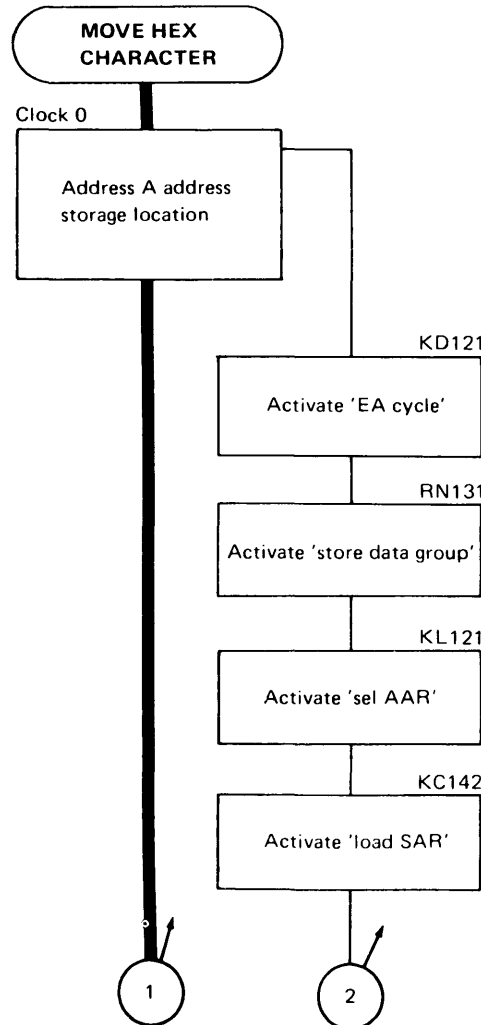
The other half of the B register byte passes through the ALU without any ALU controls and the new byte is entered into storage. At clock 8 time the 'op end' trigger is turned on and the operation ends.

# Move Hex Character—MVX

Bits 0 1 2 3 4 5 6 7  
Op Code x x x x 1 0 0 0

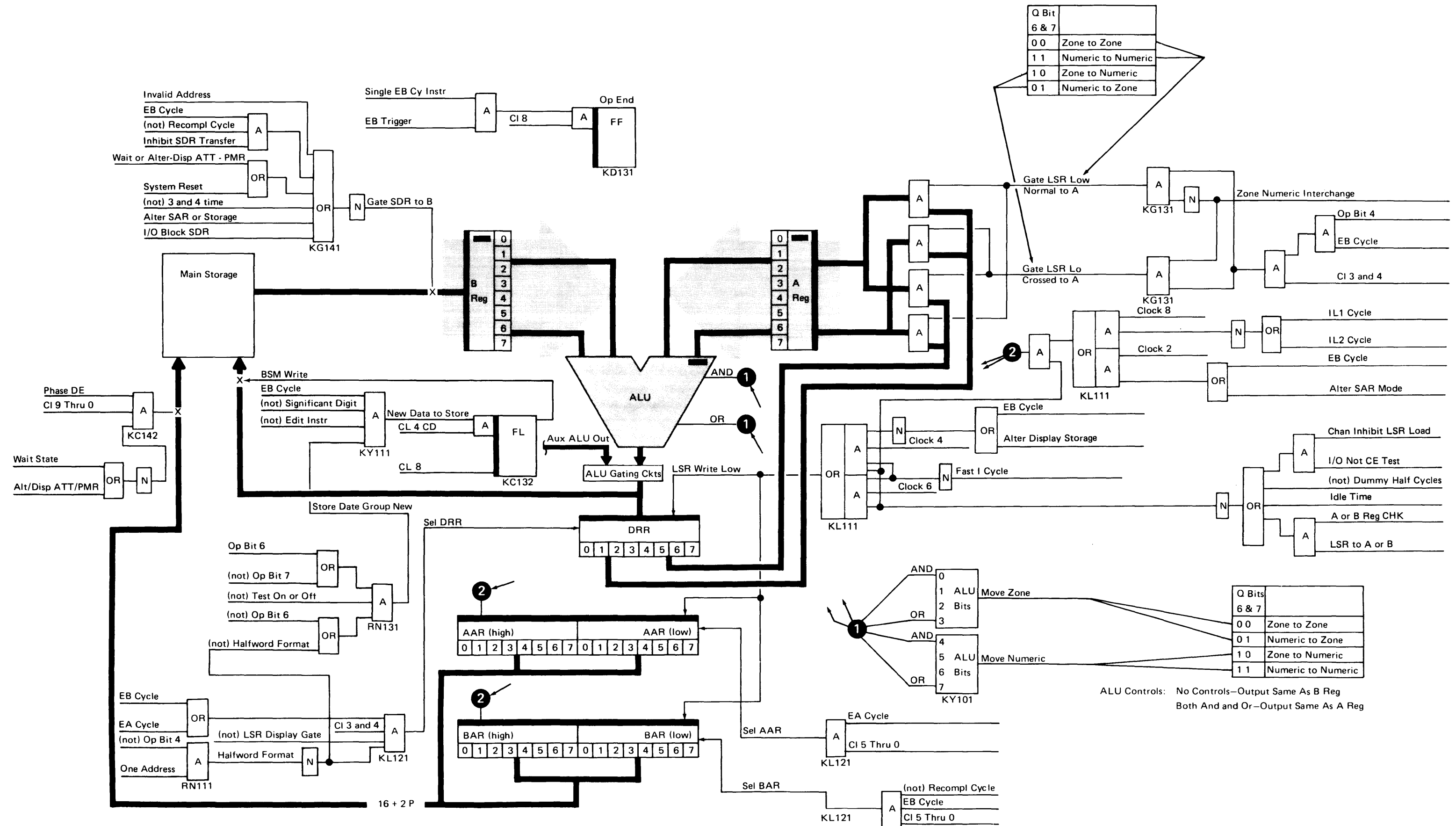
## Objectives:

- Move numeric (lower four bits) or zone (upper four bits) portion of the byte in the location specified by the A address register, to the numeric or zone portion of the byte in the location specified by the B address register.
- Do not change the other half of the B address location byte.
- Do not change the A address location byte.
- Bits 6 and 7 of Q code specify portion used.



This page intentionally left blank

Move Hex Character—MVX



### ONE ADDRESS INSTRUCTIONS

#### I-Cycles

- Load operation code into op register.
- Load Q code into Q register and DRR.
- Load B field address into BAR except for load address instruction.
- Load index register for load address instruction.
- Load B field address into ARR for branch instruction.

Single address instructions require a maximum B field length of two bytes. Therefore, it is not necessary to maintain a field length count in the Q code for them. Because it is not necessary to maintain a field length count in the Q code, the Q code can be used for controlling the functions necessary to execute the single address instructions. Use of the Q code is discussed with the individual instruction descriptions.

I-cycles for single address instructions are either three or four cycles in length. First, an I-op cycle transfers the operation code from main storage to the op register. Second, an I-Q cycle transfers the Q code into the Q register and the DRR.

Two cycles, an I-H1 and an I-L1 cycle, are then used to load the B field address into the BAR. For branch instructions, the B field address is also loaded into the ARR. For a load address instruction, and index register is loaded instead of the BAR. This is discussed with *Load Address*.

If indexing is used, a single I-X1 cycle replaces the I-H1 and I-L1 cycles.

I-op and I-Q cycles are the same as in 2 address instructions except the Q code is stored in the DRR instead of the LCR and LCRR. I-H1 and I-L1 cycles are the same unless the instruction is a load address instruction (refer to *Load Address*).

The need for an I-X1 cycles is determined by (1) either op bit 0 or 1, but not both, or (2) bit 2 or 3, but not both. The I-X1 cycle description and the index register selected are the same as in two address instructions.

Any additional considerations made during the I-cycles are discussed with the individual operation descriptions.

The I-cycle description begins on page 5-3.

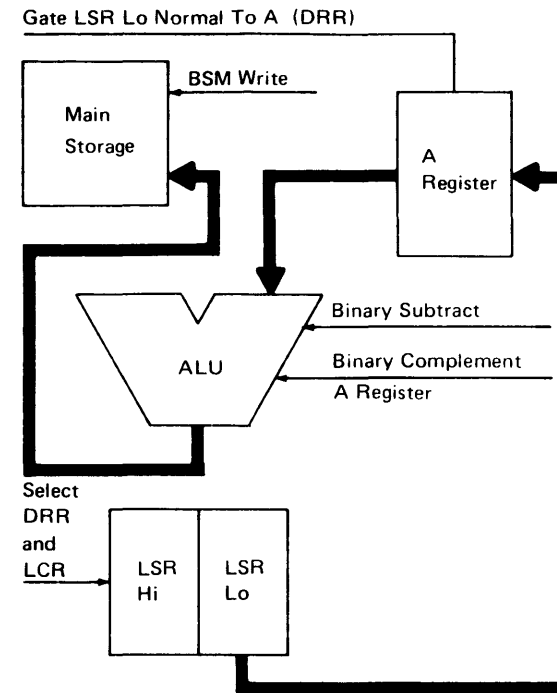
#### Move Logical Immediate—MVI

Bits 0 1 2 3 4 5 6 7  
Op Code x x 1 1 1 1 0 0

- Move the Q code byte to the B address storage location.

The move logical immediate operation moves the byte of data which is contained in the Q code portion of the instruction to the B address storage location. Since only one byte is being moved, the operation is executed with a single B-cycle.

During the B-cycle, the storage location is addressed by the BAR, and at clock 3 and 4 time the DRR is transferred to the A register.



The DRR contains the Q code that was stored in it during the I-Q cycle. The data in storage, if any is present, is not transferred to the B register and the A register is binary added to the zeros in the B register. The result is written into the B address location. The op-end trigger is then turned on and the operation ends.

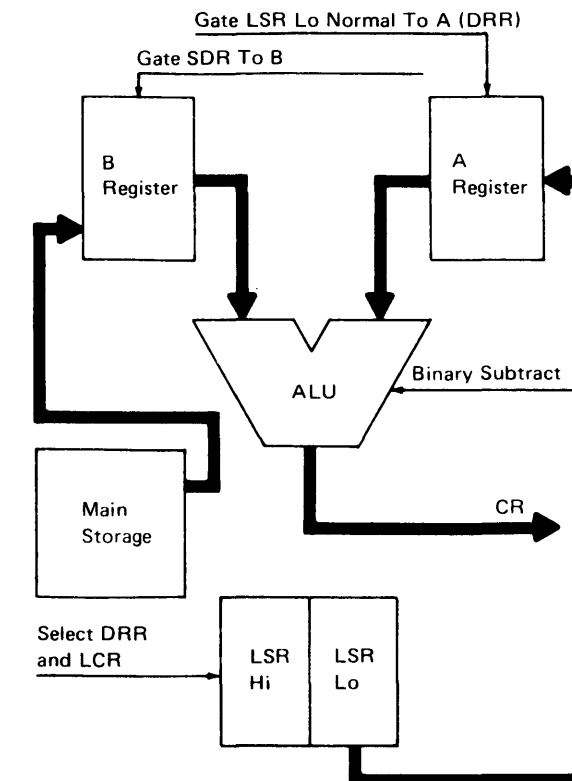
#### Compare Logical Immediate—CLI

Bits 0 1 2 3 4 5 6 7  
Op Code x x 1 1 1 1 0 1

- Compare Q code with byte in B address storage location.

The compare logical immediate operation compares the byte of data that is contained in the Q code portion of the instruction with the B address storage location byte. Since only one storage position is involved, the operation requires a single B-cycle.

The operation is similar to a move logical operation except the B address byte is loaded into the B register and the Q code byte is binary subtracted from it.



The results are not entered into storage but are used to set the condition register.

Equal	Low	High
B address and Q bytes are equal	B address byte is lower than Q byte	B address byte is higher than Q byte

# Move Logical Immediate or Compare Logical Immediate

## Objectives

### Move Logical Immediate

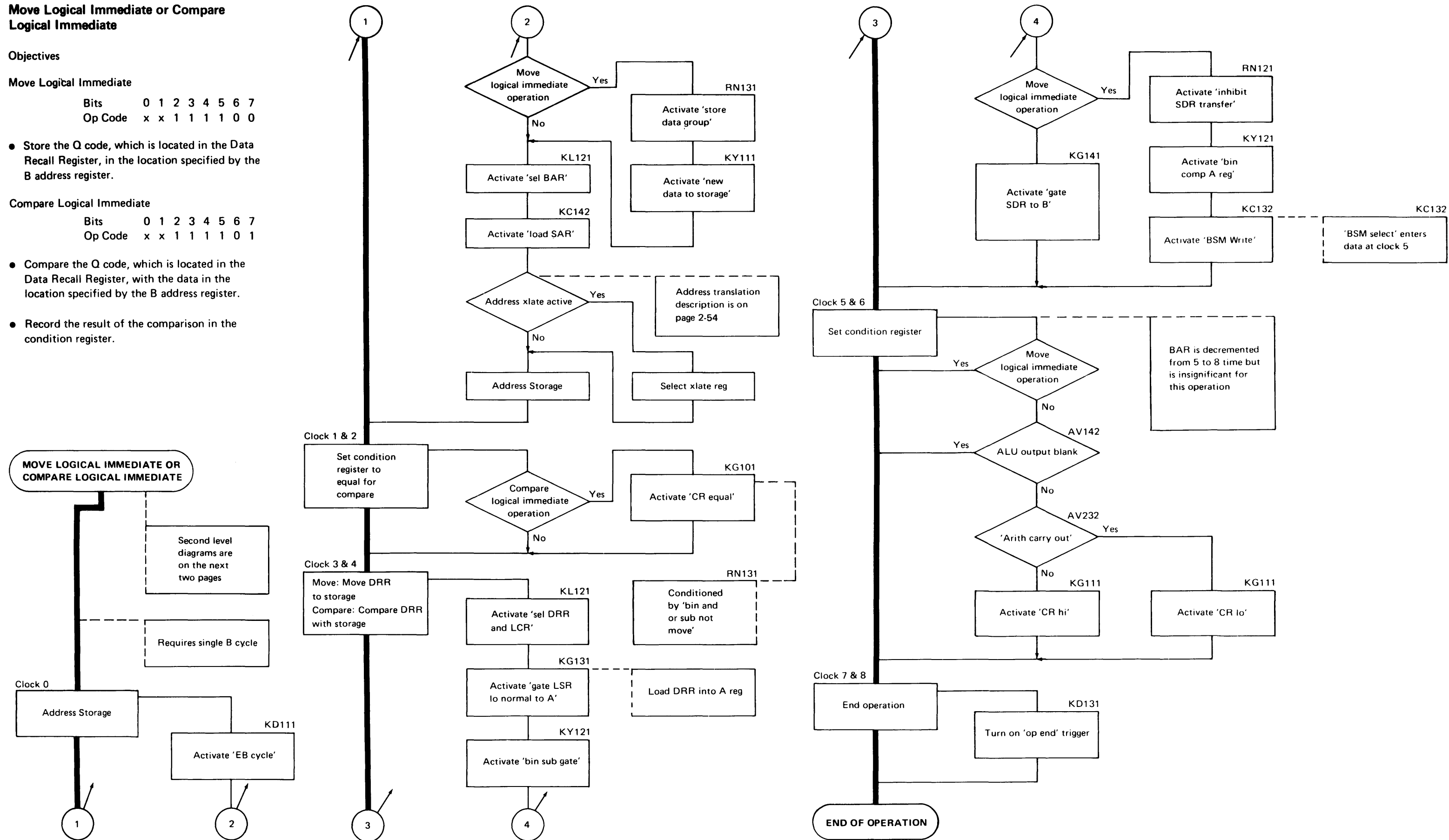
Bits 0 1 2 3 4 5 6 7  
Op Code x x 1 1 1 1 0 0

- Store the Q code, which is located in the Data Recall Register, in the location specified by the B address register.

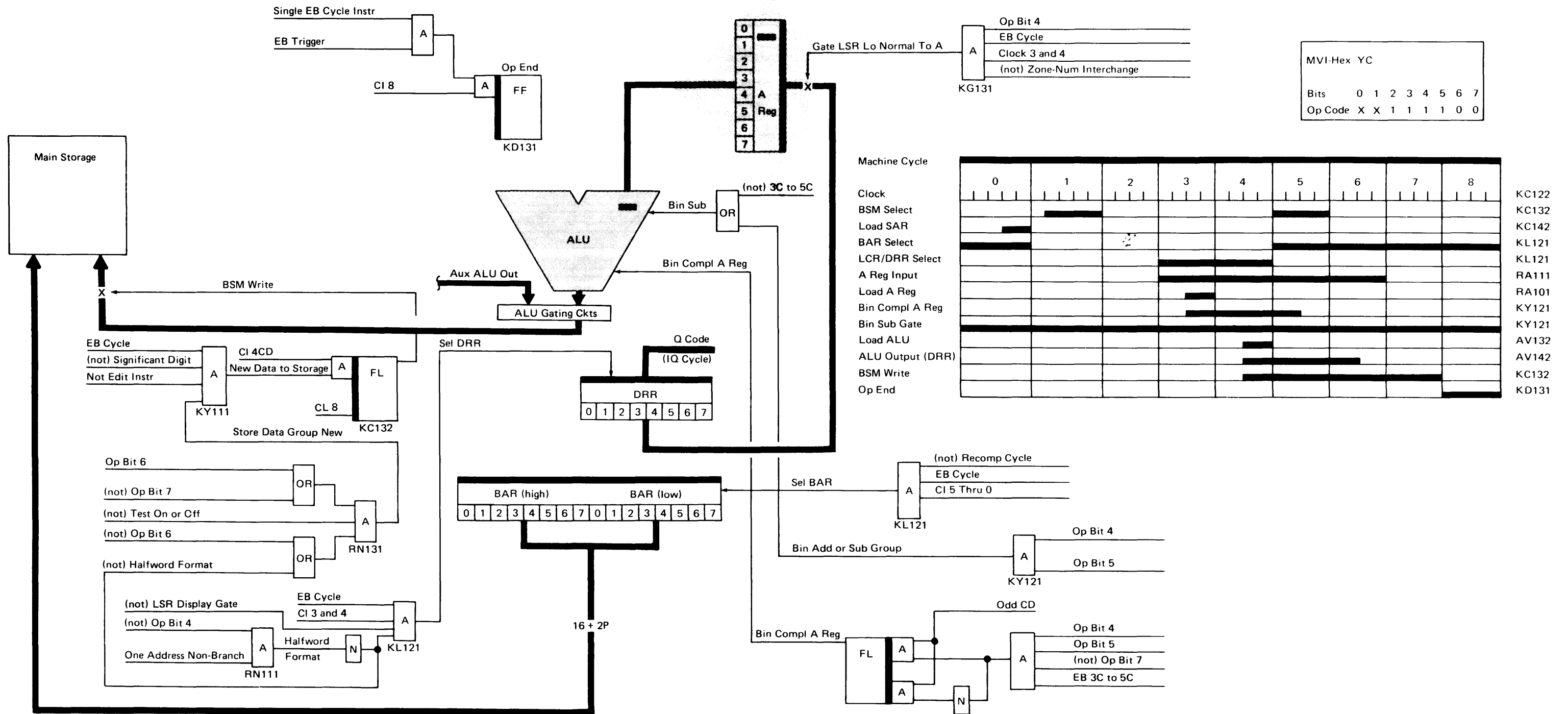
### Compare Logical Immediate

Bits 0 1 2 3 4 5 6 7  
Op Code x x 1 1 1 1 0 1

- Compare the Q code, which is located in the Data Recall Register, with the data in the location specified by the B address register.
- Record the result of the comparison in the condition register.

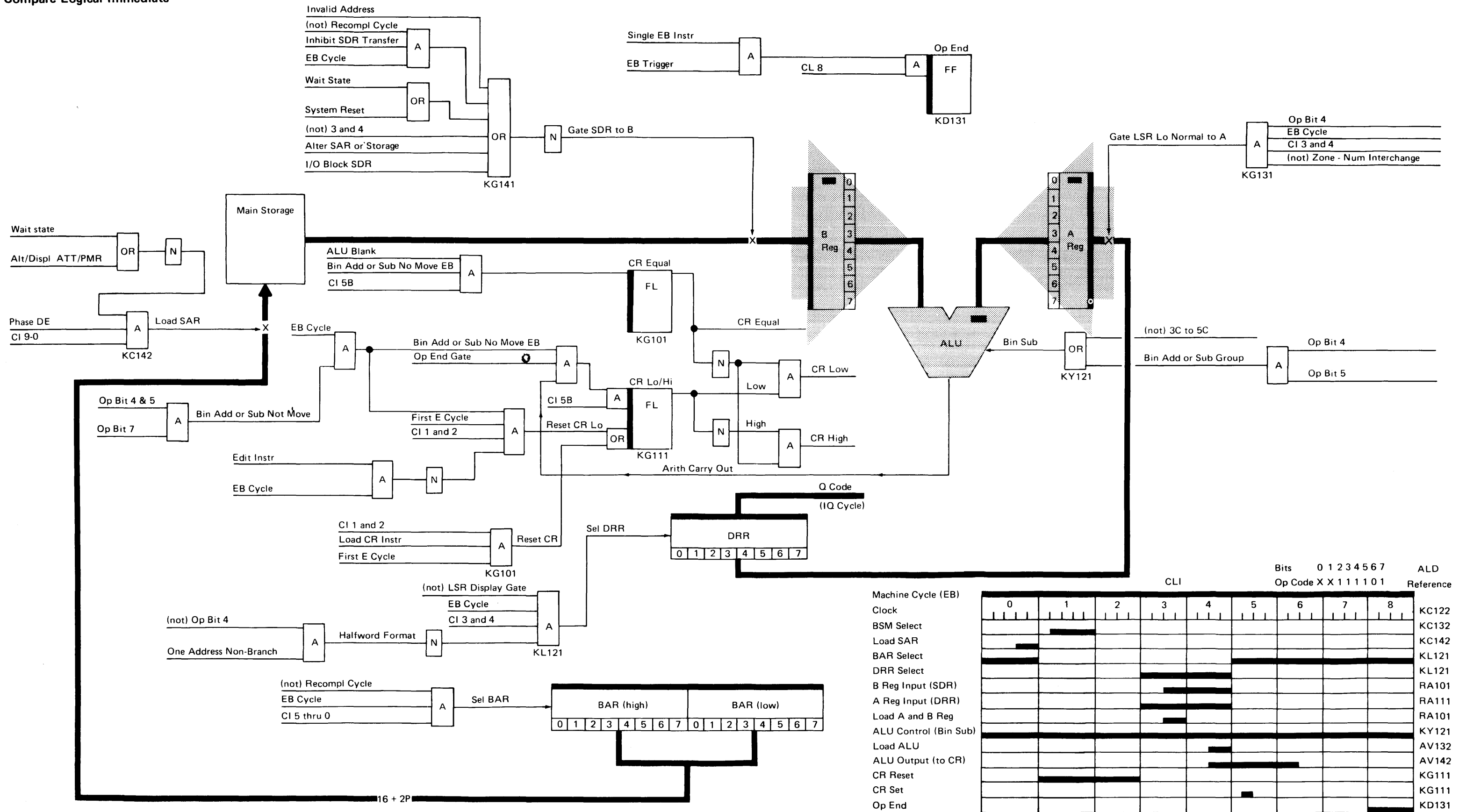


Move Logical Immediate





# Compare Logical Immediate



Machine Cycle (EB)	0	1	2	3	4	5	6	7	8	Reference
Clock										KC122
BSM Select										KC132
Load SAR										KC142
BAR Select										KL121
DRR Select										KL121
B Reg Input (SDR)										RA101
A Reg Input (DRR)										RA111
Load A and B Reg										RA101
ALU Control (Bin Sub)										KY121
Load ALU										AV132
ALU Output (to CR)										AV142
CR Reset										KG111
CR Set										KG111
Op End										KD131

**Set Bits On/Off Masked and Test Bits On/Off Masked (Part 1 of 3)**

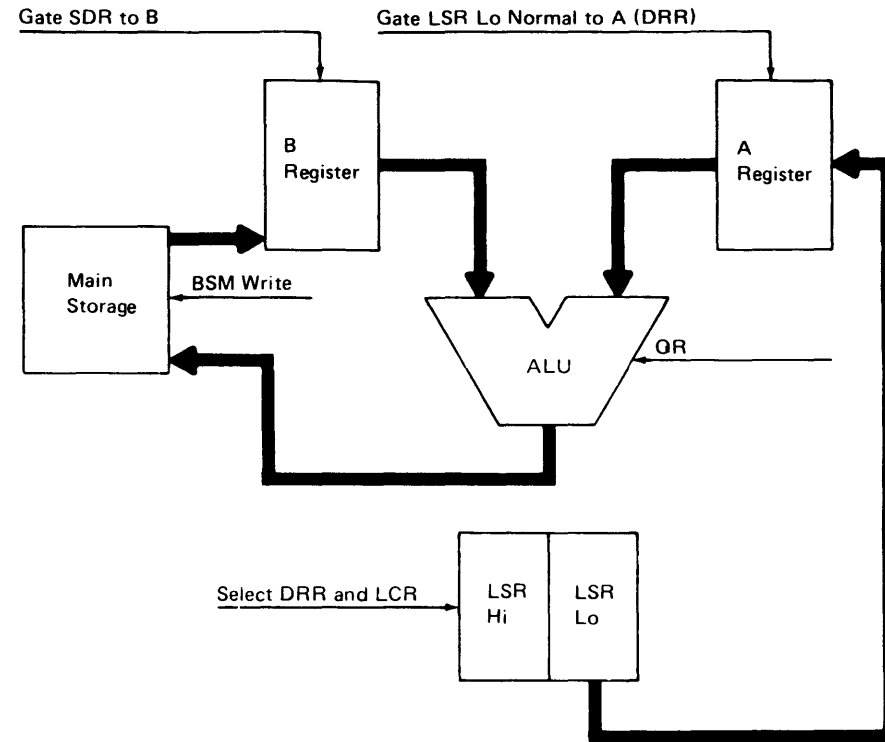
**Set Bits On Masked—SBN**

- Place bits that are present in the Q code into the B address storage location.
- Do not change the remainder of B address byte.

The set bits on masked operation, turns on the bits in the B address storage location that correspond to the Q code bits. Bits that were already on in the B address byte are left on.

Example: Q Code Byte 11000111  
 B Address Byte 10010010  
 New B Address Byte 11010111

The operation requires a single B-cycle and ORs the Q code byte, which was stored in the DRR during the I-Q cycle, with the B address byte. The result is written into storage. The op-end trigger is turned on and the operation ends.

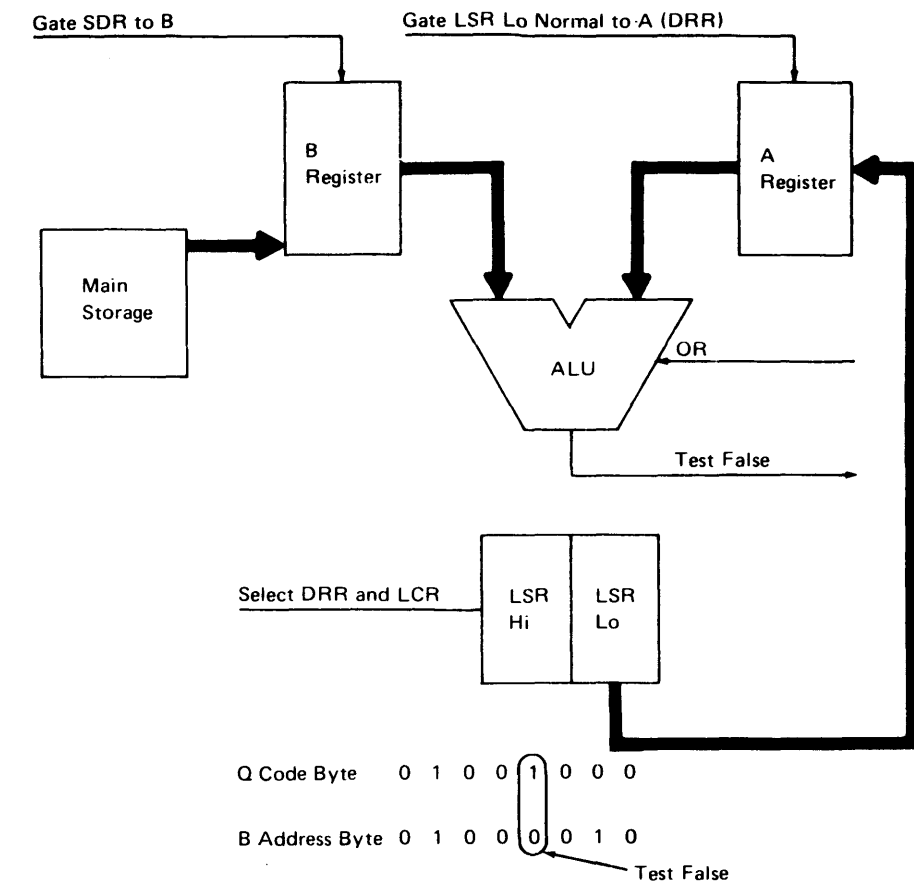


**Test Bits On Masked—TBN**

- Activate 'CR test false' if bits present in the Q code are not all present in the B address storage location.

The test bits on masked operation tests to determine if all bits present in the Q code are also present in the B address storage location. If they are not, the 'CR test false' latch is turned on.

The operation requires a single B-cycle and uses the 'OR' control line in the ALU. The Q code is transferred from the DRR to the A register and the B-field byte is loaded into the B register. Any bit in the A register that is not present in the B register gives a 'test false' output. The results are not written into storage but are used merely to set the condition register.



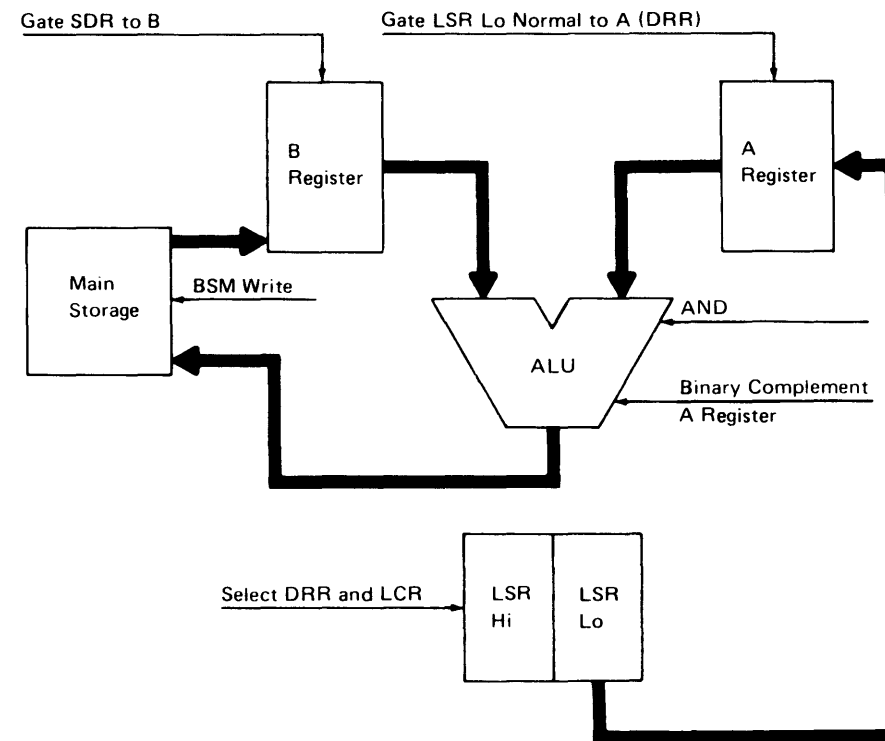
**Set Bits Off Masked—SBF**

- Remove bits that are present in the Q code from the B address storage location.
- Do not change the remainder of the B address byte.

The set bits off masked operation turns off the bits in the B address storage location that correspond to the Q code bits. The rest of the bits in the B address byte are left unchanged.

Example: Q code byte 11000111  
 B address byte 10010010  
 New B address byte 00010000

The operation requires a single B-cycle and ANDs the B address byte with the binary complement of the Q code. The result is written into storage and the op-end trigger is turned on to end the operation. The Q code was stored in the DRR during the I-Q cycle of the operation.

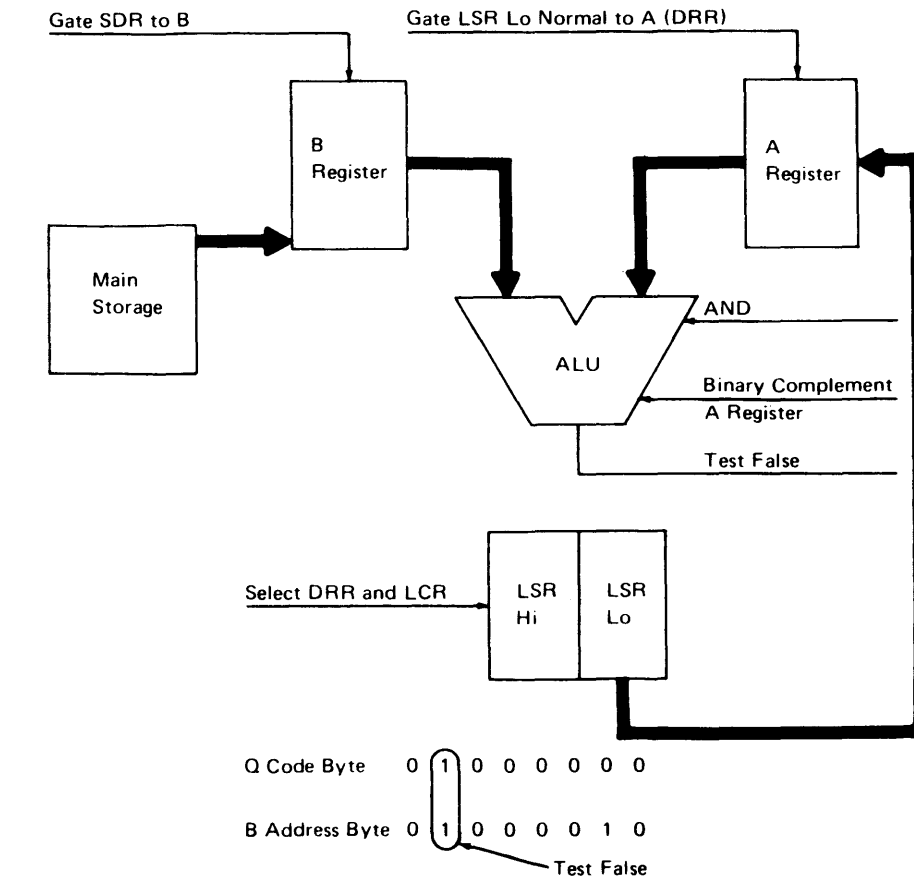


**Test Bits Off Masked—TBF**

- Activate 'CR test false' latch if any bits present in the Q code are also present in the B address storage location.

The test bits off masked operation tests to determine if all bits present in the Q code are absent from the B address storage location. If they are not, the 'CR test false' latch is turned on.

The operation requires a single B-cycle. The Q code is transferred from the DRR to the A register. The A register is binary complemented and the AND control line in the ALU is used to give a 'test false' output for any bit in the Q code which has a corresponding bit in the B address byte. The results are not written into storage but are used merely to set the condition register.



### Set Bits On/Off Masked and Test Bits On/Off Masked (Part 2 of 3)

Objectives:

#### Set Bits On Masked

Bits 0 1 2 3 4 5 6 7  
Op Code x x 1 1 1 0 1 0

- If a bit is present in the Q code, turn on the corresponding bit in the storage location specified by the B address register.
- Do not change bits which correspond with bits not present in the Q code.

#### Set Bits Off Masked

Bits 0 1 2 3 4 5 6 7  
Op Code x x 1 1 1 0 1 1

- If a bit is present in the Q code, turn off the corresponding bit in the storage location specified by the B address register.
- Do not change bits which correspond with bits not present in the Q code.

#### Test Bits On Masked

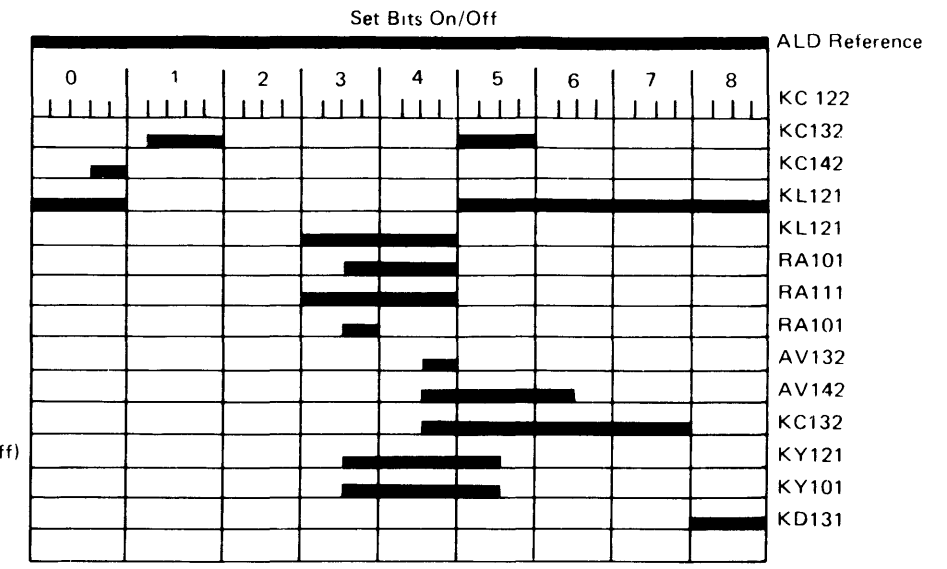
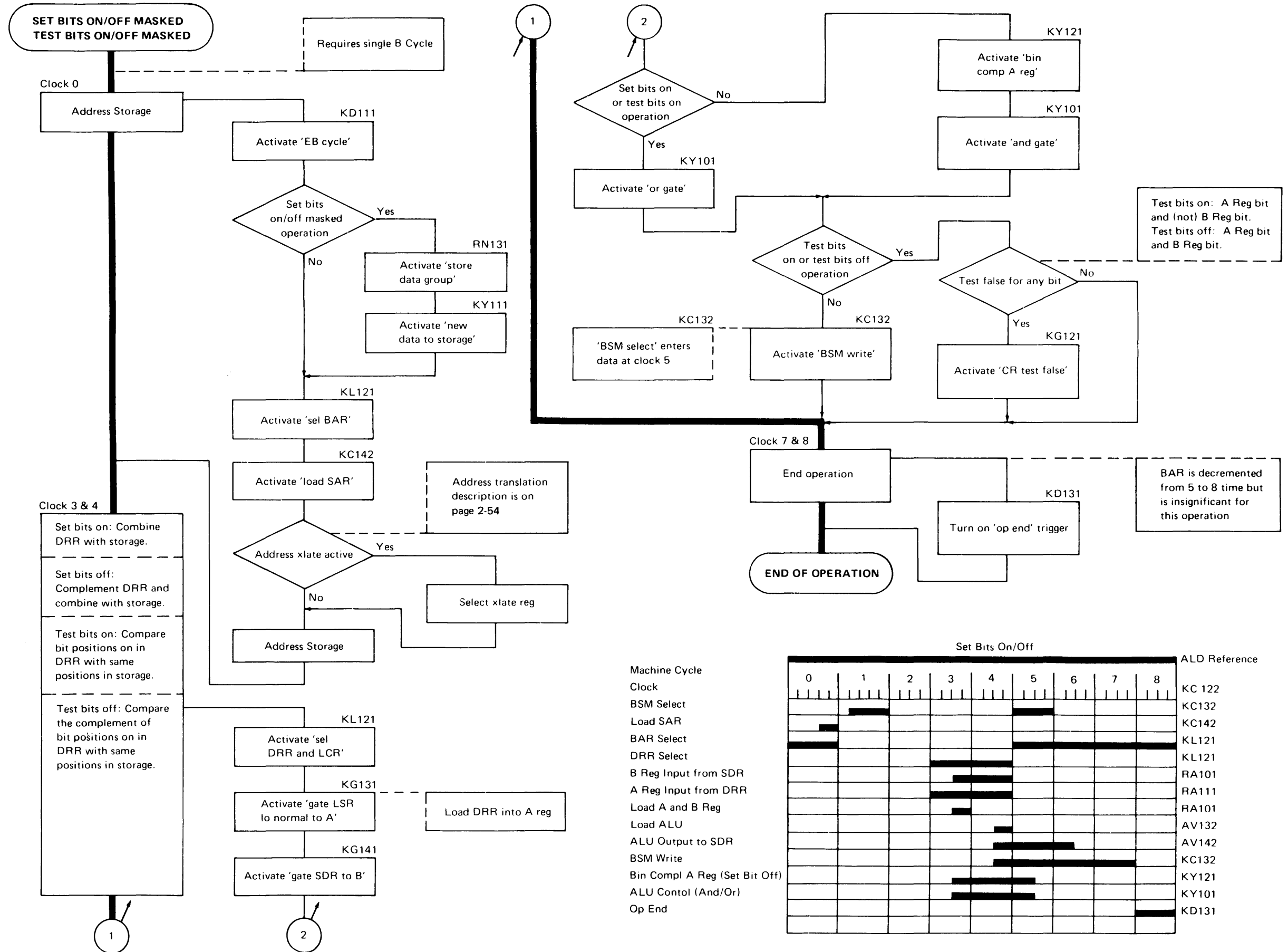
Bits 0 1 2 3 4 5 6 7  
Op Code x x 1 1 1 0 0 0

- If a bit is present in the Q code, test to see if the corresponding bit in the storage location specified by the B address register is on.
- Ignore bits that correspond with bits not present in the Q code.
- Turn on 'test false' latch if selected bits are not all on.

#### Test Bits Off Masked

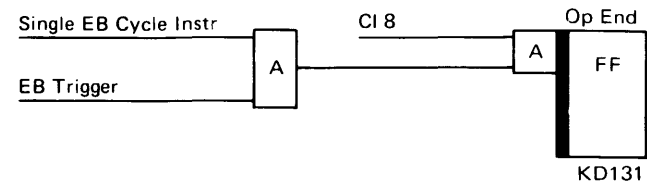
Bits 0 1 2 3 4 5 6 7  
Op Code x x 1 1 1 0 0 1

- If a bit is present in the Q code, test to see if the corresponding bit in the storage location specified by the B address register is off.
- Ignore bits that correspond with bits not present in the Q code.
- Turn on 'test false' latch if selected bits are not all off.



This page intentionally left blank

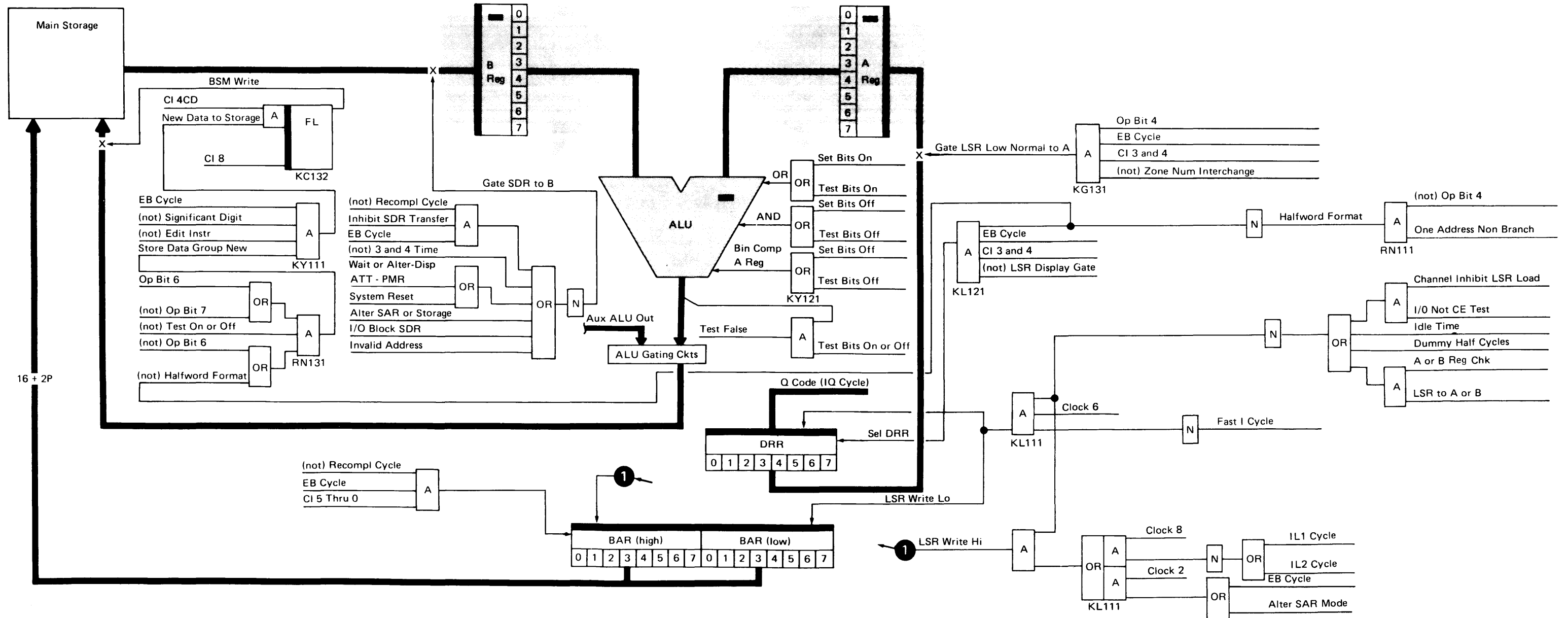
**Set Bits On/Off Masked and Test Bits On/Off Masked (Part 3 of 3)**



Machine Cycle

- Clock
- BSM Select
- Load SAR
- DRR Select
- BAR Select
- B Reg Input (SDR)
- A Reg Input (DRR)
- Load A and B Reg
- Load ALU
- Bin Compl A Reg (test bit off)
- ALU Control (and/or)
- ALU Output (test false)
- CR Control
- Op End

Test Bits On/Off									
0	1	2	3	4	5	6	7	8	ALD Ref
									KC122
									KC132
									KC142
									KL121
									KL121
									RA101
									RA111
									RA101
									AV132
									KY121
									KY101
									AV142
									KG101
									KD131



**Store, Load, or Add to Register (Part 1 of 4)**

**Store Register – ST**

Bits      0 1 2 3 4 5 6 7  
Op Code   x x 1 1 0 1 0 0

- Store the registers that are selected by the Q code into the location specified by the BAR.

The store register instruction stores an LSR in the B field storage locations. LSR selection is divided into two different groups depending upon the presence or absence of Q bit 0.

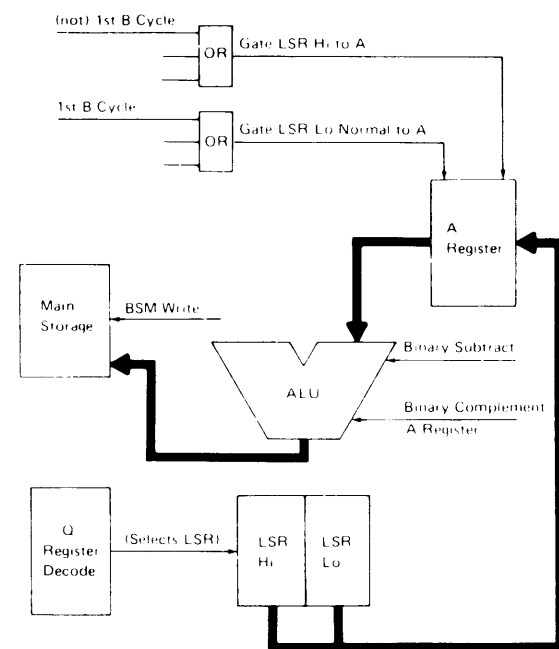
Q Code Bits	Register Selected	
	When Q Bit 0 = 1	When Q Bit 0 = 0
1	Interrupt 1-IAR	P-ARR
2	Interrupt 2-IAR	P-IAR
3	Interrupt 3-IAR	IAR*
4	Interrupt 4-IAR	ARR*
5	Interrupt 5-IAR	PSR
6	Interrupt 6-IAR	XR2
7	Interrupt 7-IAR	XR1

**Note:** When Q bit 0 = 1 and all other Q bits = 0, Interrupt 0-IAR is selected.

\* Current registers in use. Can be program level or any interrupt level registers.

Since the LSRs are two bytes long, the store register instruction requires two B cycles. During the first B cycle, the Q register selects the LSR and the low order position is transferred to the A register. The A register is binary added to the B register (all zeros). This moves the LSR byte through ALU and into storage.

The BAR is decremented and in the second B cycle, the high order byte of the LSR is moved. The 'op end' trigger is then turned on and the operation ends.



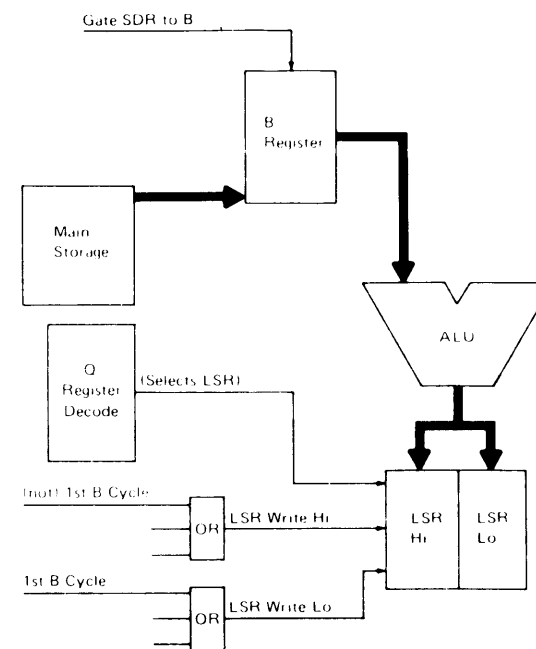
**Load Register – L**

Bits      0 1 2 3 4 5 6 7  
Op Code   x x 1 1 0 1 0 1

- Load the registers that are selected by the Q code with data from the location specified by the BAR.

The load register operation loads an LSR with the contents of the B field storage locations. LSR selection is the same as for a store register operation.

During the first B cycle, the Q register selects the LSR and the first B field byte is passed through the ALU without any ALU controls. The ALU output is then written into the low order position of the LSR.



The BAR is decremented and in the second B cycle, the next byte is written into the high order position of the selected LSR. The 'op-end' trigger is turned on to end the operation.

If the LSR selected by the Q code is the PSR (Q bit 5 and not bit 0), an additional function is performed. Since the PSR low order position is used as the CRR, the CR is also set by the ALU output during the first B cycle.

**CR setting for Load PSR**

ALU Output Bits	CR Results
7	Equal
6 not 7	Low
not 6 not 7	High
2	Binary Overflow
3	Test False
4	Decimal Overflow

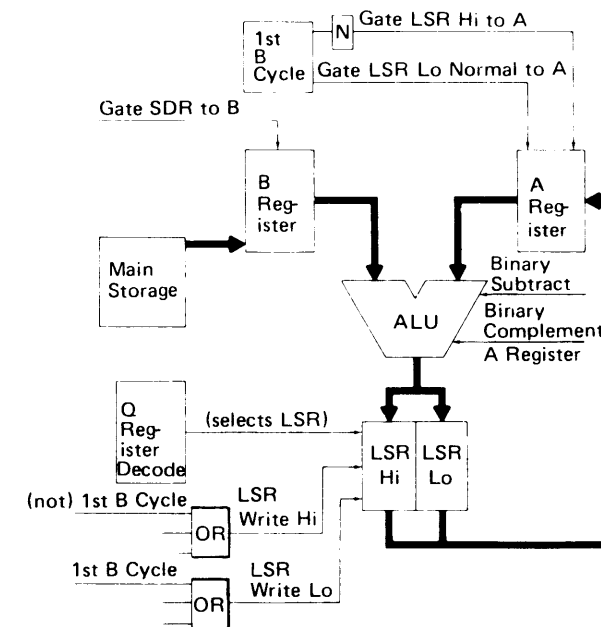
**Add to Register – A**

Bits      0 1 2 3 4 5 6 7  
Op Code   x x 1 1 0 1 1 0

- Add the data from the location specified by the BAR to the contents of the registers that are selected by the Q code.
- Load the results into the selected registers.

The add to register operation adds the B field to an LSR and loads the result into the LSR. LSR selection is the same as for a store register operation.

During the first B cycle, the Q register selects the LSR and the low order position of the LSR is transferred to the A register.



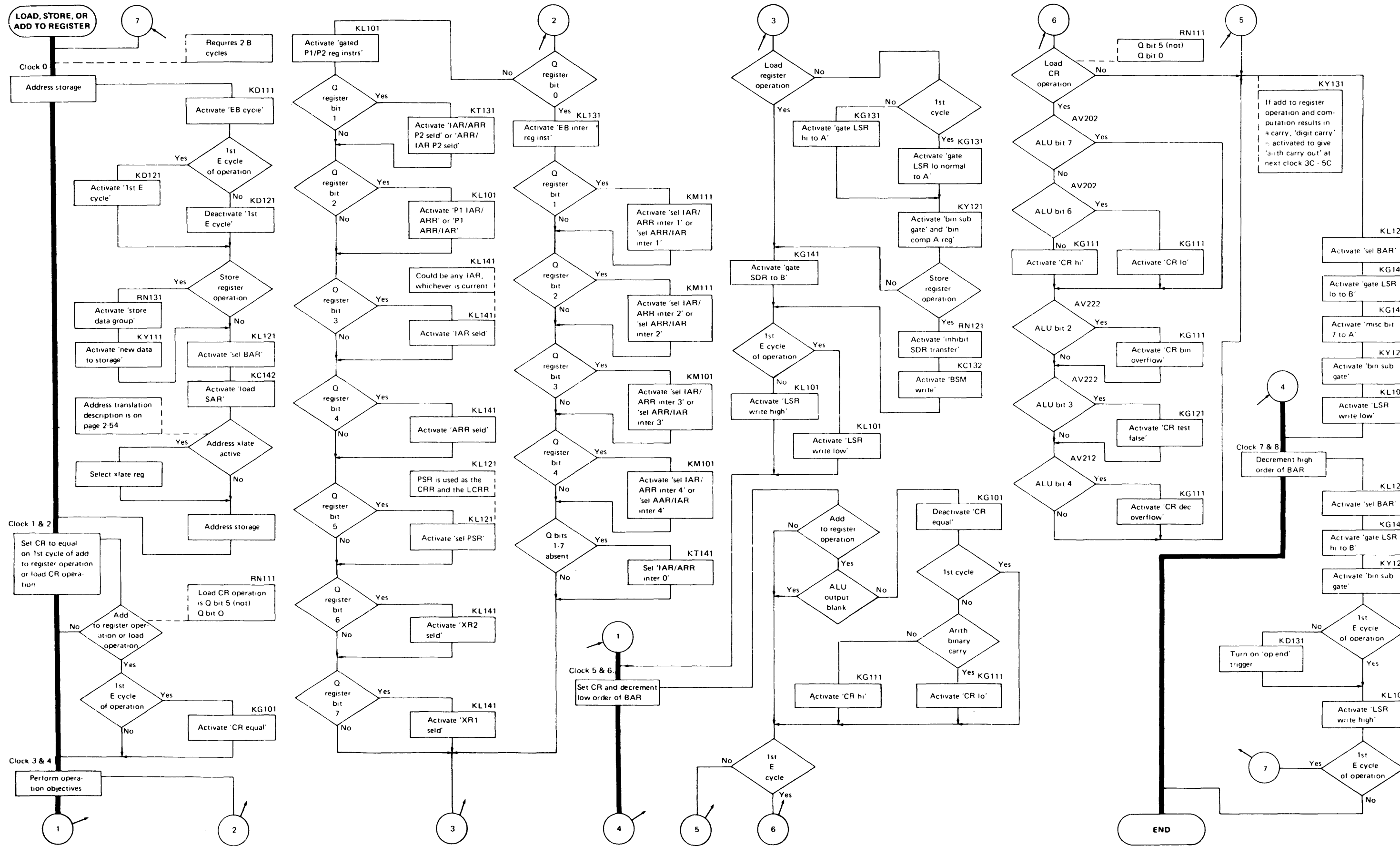
The first B field byte is loaded into the B register and binary added to the A register. The results are written into the low order position of the LSR.

The BAR is decremented and the process is repeated for the high order position of the LSR. The 'op-end' trigger is turned on to end the operation.

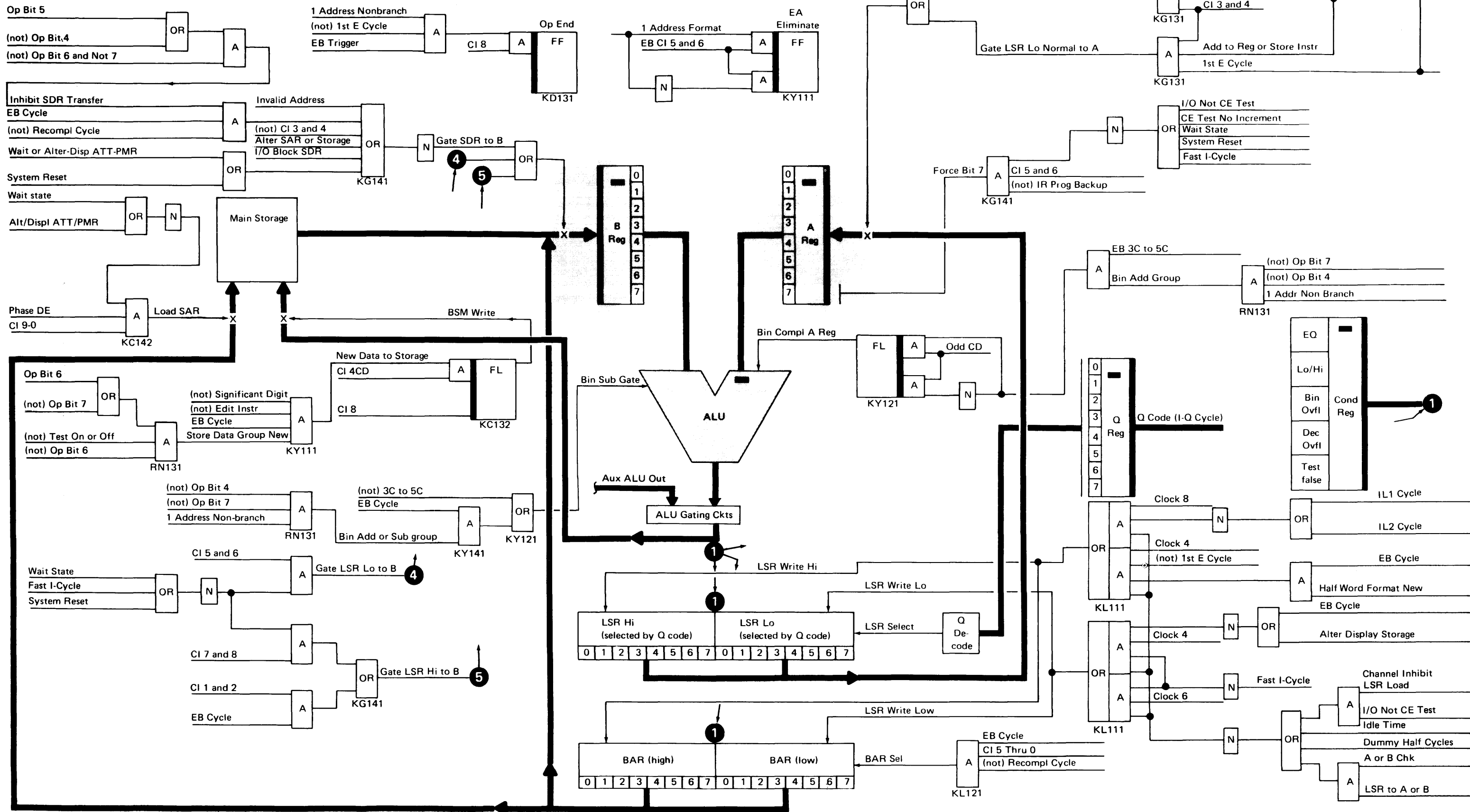
The results of the addition are also used to set the condition register.

Equal	Low	High	Binary Overflow
Result is zero	No Carry and non-zero result	Carry and non-zero result	Result too large for register (no high order carry)

Store, Load, or Add to Register (Part 2 of 4)

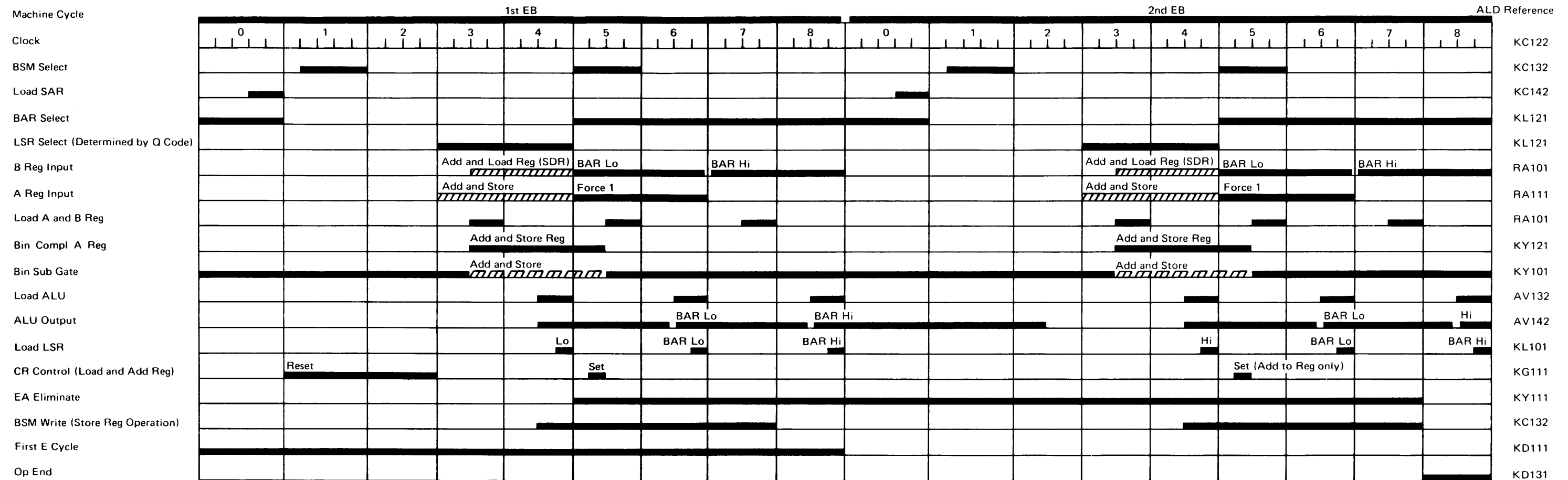


### Store, Load, or Add to Register (Part 3 of 4)





**Store, Load, or Add to Register  
(Part 4 of 4)**



Condition Register						
Operation	Equal	Low	High	Binary Overflow	Test False	Decimal Overflow
Load PSR	If ALU bit 7	If ALU bit 6, not 7	If ALU not bit 6 or 7	If ALU bit 2	If ALU bit 3	If ALU bit 4
Add to Register	If Result is zero	If Result is not zero and a high order carry	If Result is not zero and no high order carry	If Result is too large for Register (no high order carry)		

LSR Selection		
Q Code bits	Q bit 0=1	With No Q bit 0=0
1	Interrupt 1-IAR	P-IAR
2	Interrupt 2-IAR	P-ARR
3	Interrupt 3-IAR	Current IAR
4	Interrupt 4-IAR	Current ARR
5	Interrupt 5-IAR	PSR
6	Interrupt 6-IAR	XR2
7	Interrupt 7-IAR	XR1
No other bits	Interrupt 0 IAR	

**Load Address—LA**

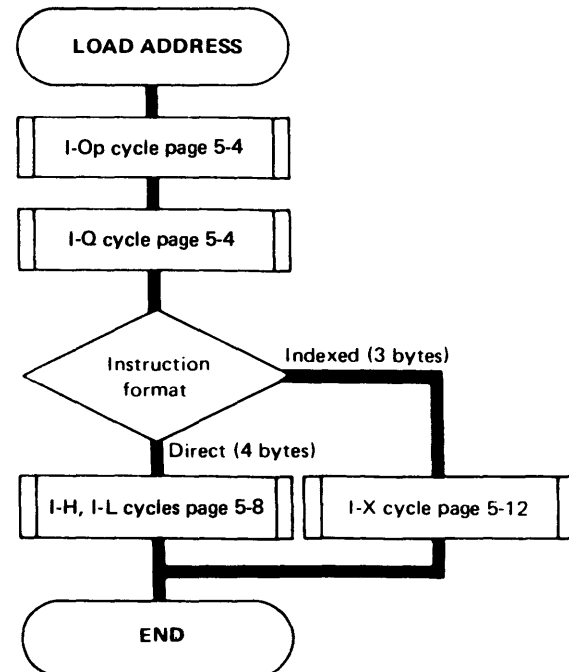
Bits 0 1 2 3 4 5 6 7  
 Op Code 1 1 x x 0 0 1 0

- Load one or two bytes from storage into one of the two index registers.
- If instruction format is four bytes, load two-byte address into index register selected by Q code bits 6 and 7.
- If instruction format is three bytes, add last instruction byte to index register selected by op code bits 2 and 3.
- Load result into index register selected by Q code bits 6 and 7.

**Index Register Selection for Load Address**

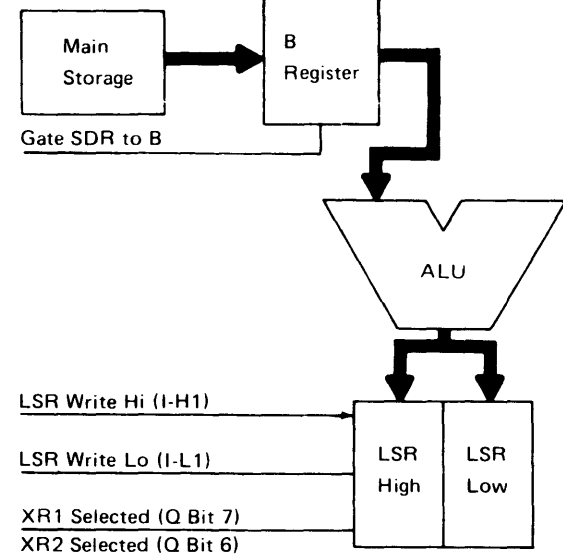
Q Code Bit	Register Selected
6	XR2
7	XR1

- Take I-H and I-L cycles (four byte format)  
 See Page 5-8
- Take I-X cycle (three byte format)  
 See Page 5-12

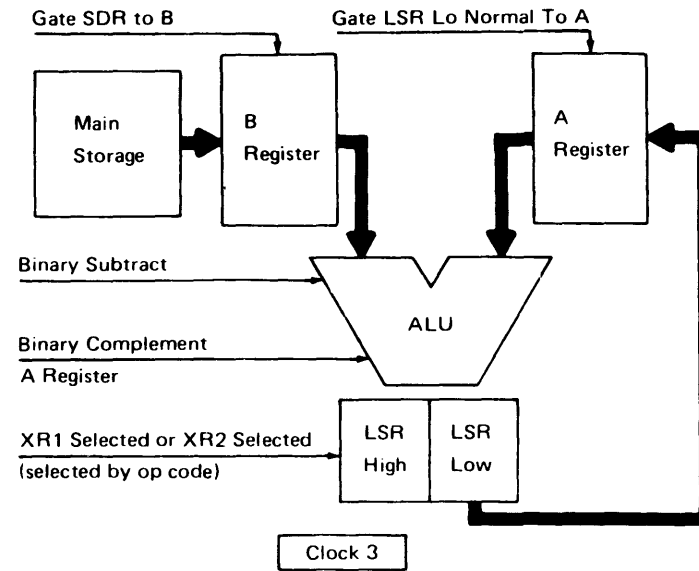


The load address instruction performs one of two possible operations, depending on the instruction length. If the instruction is four bytes long, the last two bytes of the instruction are taken from storage and loaded into the index register selected by bits 6 and 7 of the Q code; if the instruction is three bytes long, the last byte of the instruction is taken from storage, added to the contents of the index register selected by bits 0-3 of the op code, and then loaded into the index register selected by bits 6 and 7 of the Q code.

**Load Address Data Flow—Not Indexed**

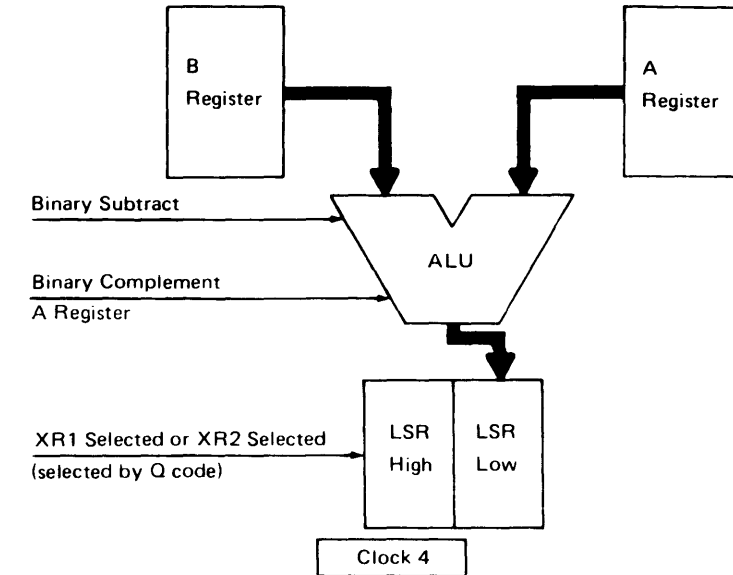


**Load Address Data Flow—Indexed**



A four byte format requires one I-H1 cycle and one I-L1 cycle. During the I-H1 cycle bits 6 and 7 of the Q code select one of the two index registers. Bit 7 only being on selects XR1, bit 6 only being on selects XR2. Data is transferred from the storage position addressed by the IAR to the B register, through the ALU, and into the high order position of the selected index register. The IAR is incremented and during the I-L1 cycle the process is repeated for the low order position of the index register.

A three-byte format requires going through only one I-X cycle. Data is transferred from the storage location address by the IAR to the B register and at clock 3 time is added to the contents of the selected index register (bits 0 through 3 of the op code). Bits 6 or 7 of the Q code selects one of the index registers at clock 4 time and the sum in the ALU is loaded into that register. The IAR is incremented for the next operation.



### Branch On Condition—BC

Bits      0 1 2 3 4 5 6 7  
Op Code 1 1 x x 0 0 0 0

- Condition register is tested for the condition specified in Q code.
- Branch to address is placed in ARR.
- Branch to address becomes next sequential instruction.

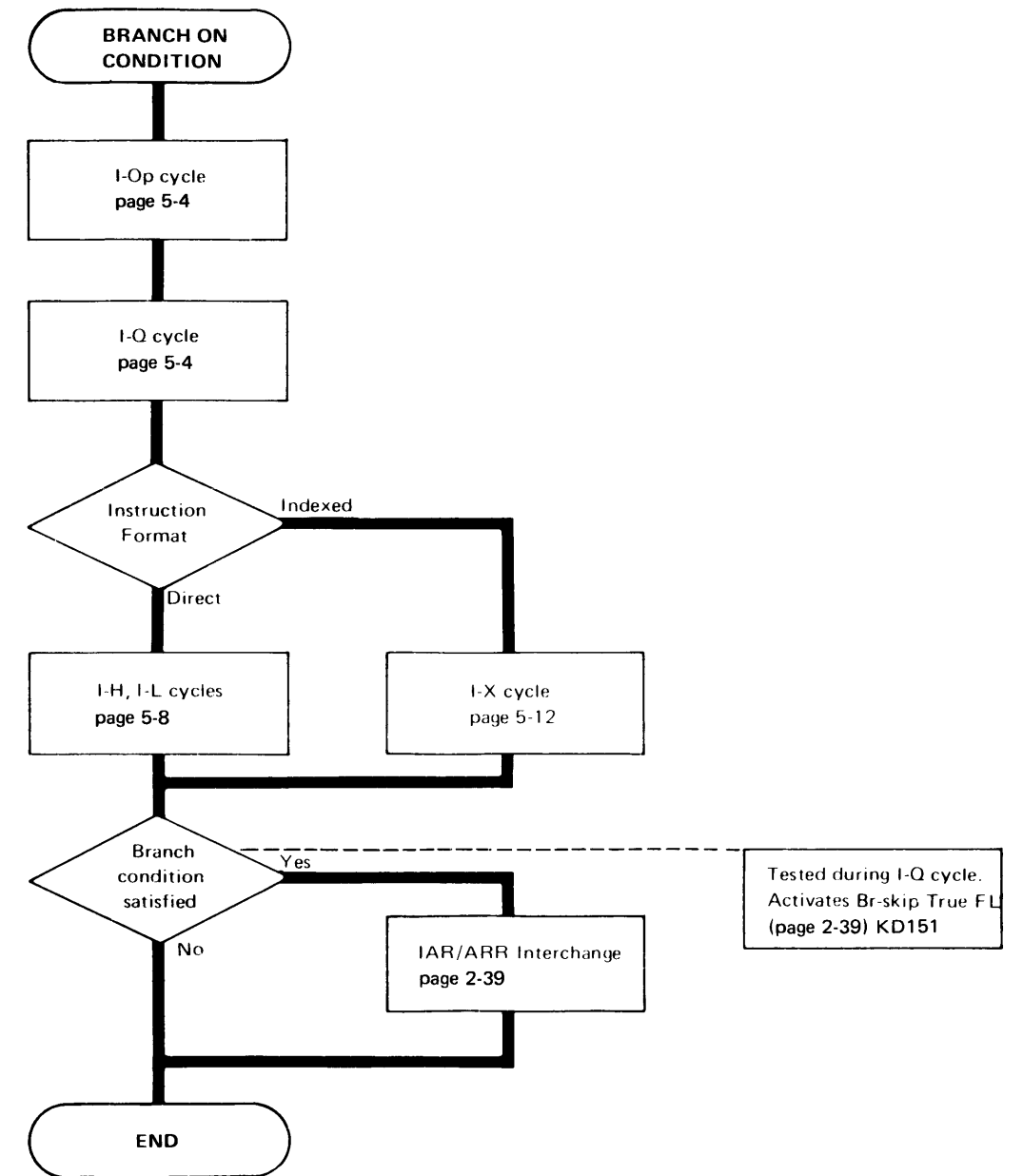
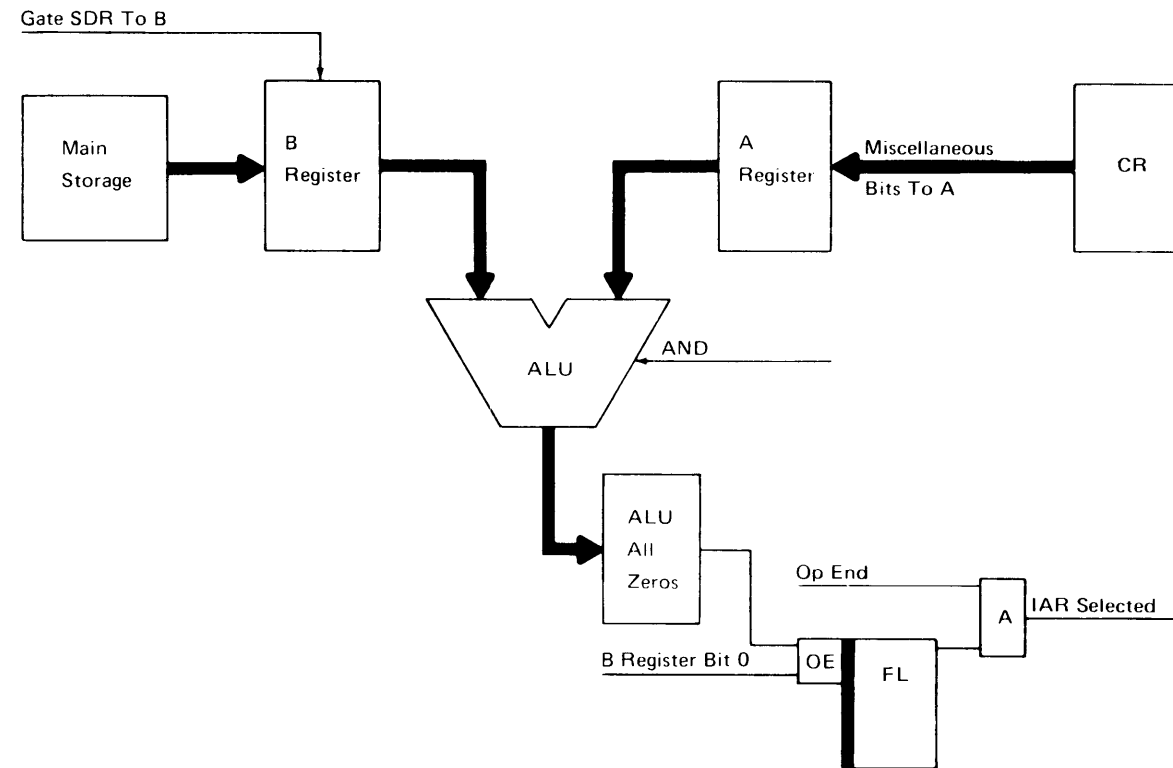
The branch on condition operation loads the two byte branch to address into the ARR. If the condition specified in bits 2 through 7 of the Q code is satisfied, an IAR/ARR interchange occurs at op end. The ARR is then used as the IAR.

Bit 0 of the Q code is used to specify if the branch is to be performed on condition true or condition false. If bit 0 is on and at least one of the conditions specified by the Q code is present, the branch is performed. If bit 0 is off and all conditions specified by the Q code are missing, the branch is performed.

During the I-Q cycle, the Q code data is transferred from storage, through the B register, and into the ALU. The contents of the condition register are decoded and enter the ALU through the A register. An ALU AND function is performed (both input bits must be the same to get an output), and the output is checked for nonzero. The result is placed in the Q register. B register bit 0 indicates whether or not an ALU sum of zero or not-zero is needed to satisfy the branch condition. The following list shows the function of each Q code bit when testing the condition register.

Q Bit	Condition Tested
0	Presence of Condition
Not 0	Absence of Condition
7	Equal
6	Low
5	High
4	Decimal Overflow
3	Test False
2	Binary Overflow

During the I-H1 cycle, the ARR is selected and the high order position of the branch to address is transferred from storage through the B register, ALU and into ARR high. The IAR is incremented and the process is repeated for the low order position.



**Store CPU (SCP)**

Bits      0 1 2 3 4 5 6 7  
 Op Code  x x 1 1 1 1 1 0

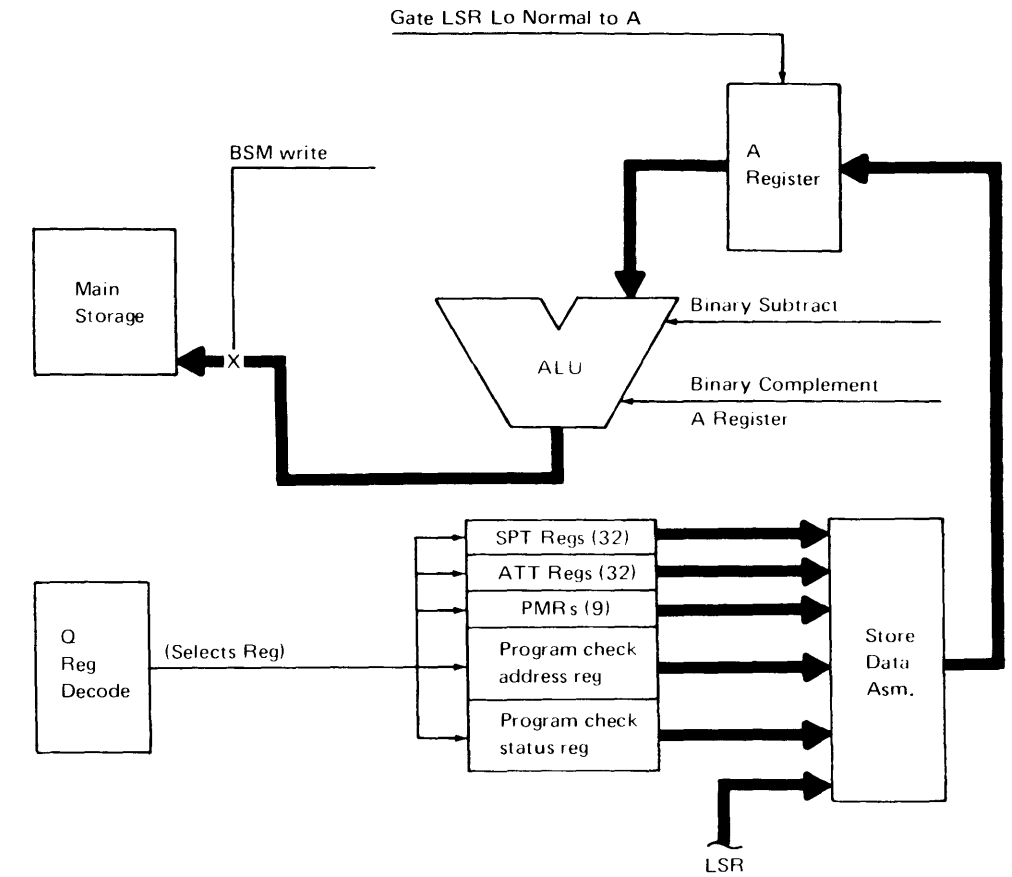
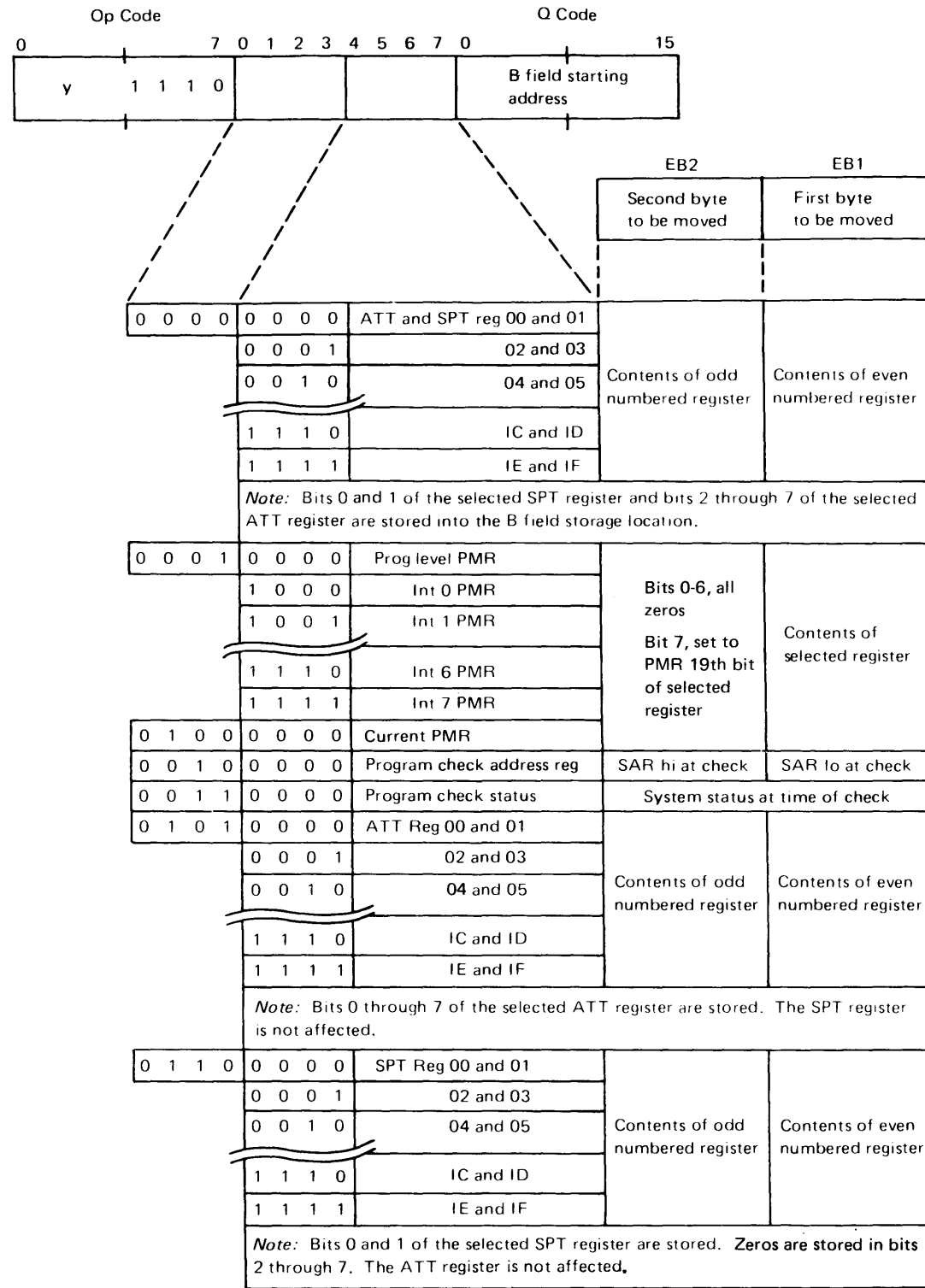
- Place the contents of the register selected by the Q code into the B field storage location.
- This is a privileged instruction.

The store CPU instruction is very similar to the store register instruction. It stores the register specified into the B field storage location. Registers selected by this instruction are divided into four groups depending on Q bits 1, 2, and 3:

1. Address translation/storage protect registers
2. Program mode registers
3. Program check address register
4. Program check status register

The store CPU instruction requires two B-cycles. During the first B-cycle the Q register bits are decoded to select the required register. The first byte is transferred to the A register and the B register is left all zeros so the A register is binary added to zero to move the byte through ALU and into storage.

The BAR is decremented and in the second B-cycle, the high order byte is moved. The 'op-end' trigger is then turned on and the operation ends. If the instruction is to store PMR, a second EB cycle is taken and data is transferred. The result is bits 0 through 6 are all zeros and bit 7 is set to PMR 19th bit.



## LOAD CPU (LCP)

Bits 0 1 2 3 4 5 6 7  
Op Code x x 1 1 1 1 1 1

- Place the contents of the B field (two bytes) into the register selected by the Q code.
- This is a privileged instruction.

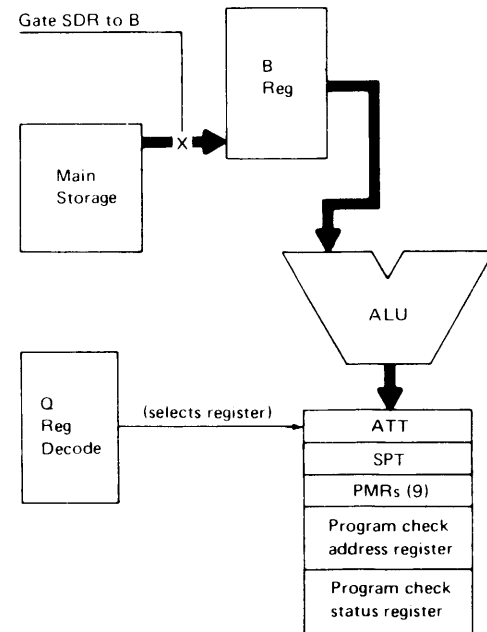
The load CPU instruction is very similar to the load register instruction. It loads a register with the contents of the two byte B field storage location.

The Q codes used in the Store CPU instruction are also used for this instruction. In addition, three other Q codes are defined only for a Load CPU instruction **A**.

During the first B cycle the Q register selects the register to be loaded. The first B field byte passes through ALU without any controls. The ALU output is then written into the low order register position.

The BAR is decremented and in the second B cycle, the next byte is written into the high order position of the selected register. The op-end trigger is turned on to end the operation. If this instruction is to load PMR, a second EB cycle is taken and data is transferred. Bit 7 is used to control the PMR 19th bit of the selected register. Bits 0-6 have no effect.

The following pages further describe the Load/Store CPU operations.



Op Code		Q Code			
0	7	0	7	0	7
Y 1 1 1 1		B field starting address			
				EB2	EB1
				Second byte moved to	First byte moved to
0 0 0 0	0 0 0 0	ATT and SPT Reg 00 and 01		Contents of odd numbered register	Contents of even numbered register
	0 0 0 1	02 and 03			
	0 0 1 0	04 and 05			
	1 1 1 0	IC and ID			
	1 1 1 1	IE and IF			
<i>Notes:</i>					
1. Operand bits 2 through 7 are loaded into the selected ATT register. Bits 0 and 1 are loaded with zeros.					
2. Operand bits 0 and 1 are loaded into the selected SPT register.					
0 0 0 1	0 0 0 0	Prog level PMR		Bit 7 is used to control PMR 19th bit	Selected register contents
	1 0 0 0	Int 0 PMR			
	1 0 0 1	Int 1 PMR			
	1 1 1 0	Int 6 PMR			
	1 1 1 1	Int 7 PMR			
0 1 0 0	0 0 0 0	Current PMR			
0 0 1 0	0 0 0 0	Program check addr reg		Chk addr reg hi	Chk addr reg lo
0 0 1 1	0 0 0 0	Program check status reg		Chk status reg hi	Chk status reg lo
0 1 0 1	0 0 0 0	ATT Reg 00 and 01		Contents of odd numbered register	Contents of even numbered register
	0 0 0 1	02 and 03			
	0 0 1 0	04 and 05			
	1 1 1 0	IC and ID			
	1 1 1 1	IE and IF			
<i>Notes:</i>					
1. Operand bits 0 through 7 are loaded into the selected ATT register.					
2. The SPT register is not affected.					
0 1 1 0	0 0 0 0	SPT Reg 00 and 01		Contents of odd numbered register	Contents of even numbered register
	0 0 0 1	02 and 03			
	0 0 1 0	04 and 05			
	1 1 1 0	IC and ID			
	1 1 1 1	IE and IF			
<i>Notes:</i>					
1. Operand bits 0 and 1 are loaded into the selected SPT register.					
2. The ATT register is not affected.					

## A

### 1. Memory Diagnostic Fetch (Q Code X'21')

The data in the main storage location addressed is loaded into the program check address register. The storage location must begin with an odd address so that the proper two bytes are fetched.

During the fetch, ECC is disabled so the program check register receives the exact contents of main storage. Because one bit errors now appear as uncorrectable errors, a status bit (bit 6) in the program check status register is set on.

The first main storage byte (bits 0-7) is loaded into byte 1 of the program check address register; the second byte (bits 8-15) is loaded into byte 2.

### 2. Memory FDR Reset (Q Code X'22')

The fetch data register is reset to hex FFFF then loaded into the program check address register.

The check bits are not affected and the ECC is not disabled.

The storage location must begin with an odd address. Then the first main storage byte (bits 0-7) is loaded into byte 1 of the program check address register; the second byte (bits 8-15) is loaded into byte 2.

### 3. Memory FDR Reset and Diagnostic Fetch (Q Code X'23')

The memory check bits are loaded into bits 0-5 of byte 1 of the program check address register. Bits 6 and 7 of byte 1 and all of byte 2 will contain 1's.

The storage location must begin with an odd address so that the proper two bytes will be fetched.

**Load/Store CPU (Part 1 of 9)**

**Store CPU**

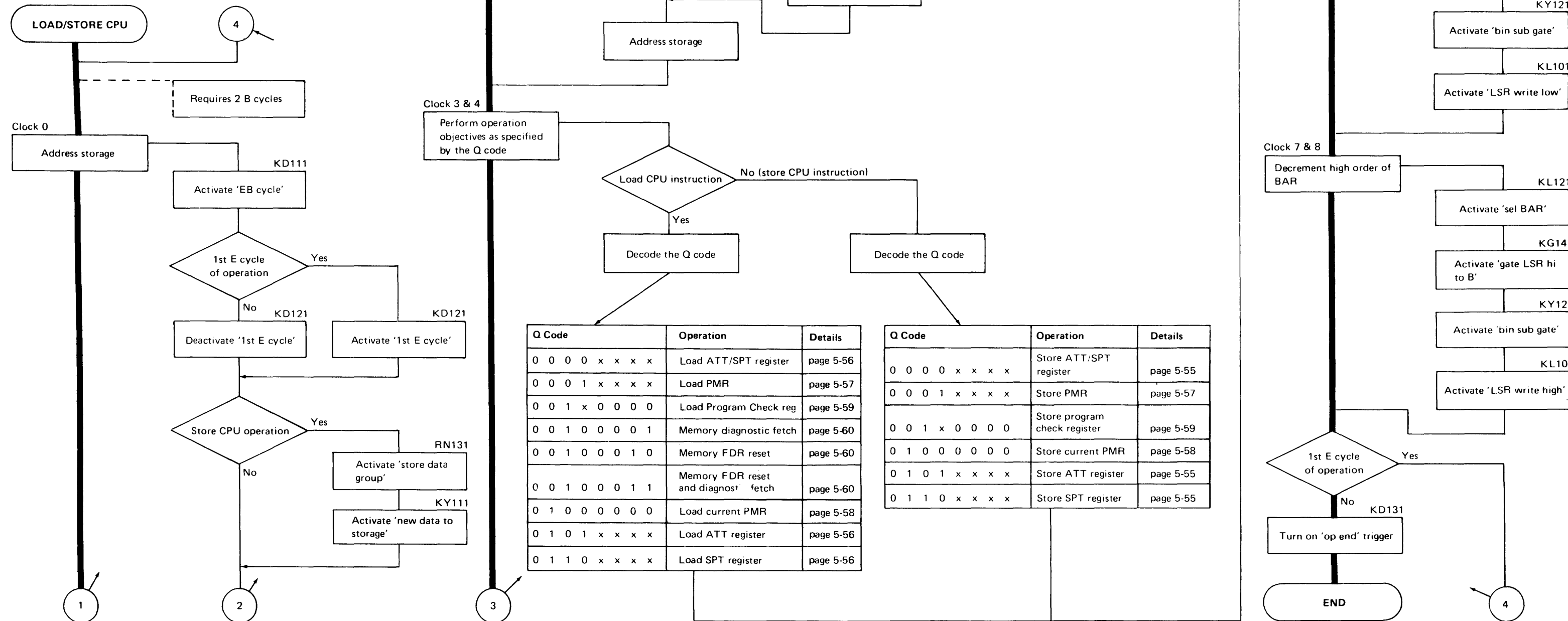
Bits 0 1 2 3 4 5 6 7  
Op Code x x 1 1 1 1 1 0

- Store the registers that are selected by the Q code into the location specified by the BAR.

**Load CPU**

Bits 0 1 2 3 4 5 6 7  
Op Code x x 1 1 1 1 1 1

- Load the registers that are selected by the Q code with data from the location specified by the BAR.

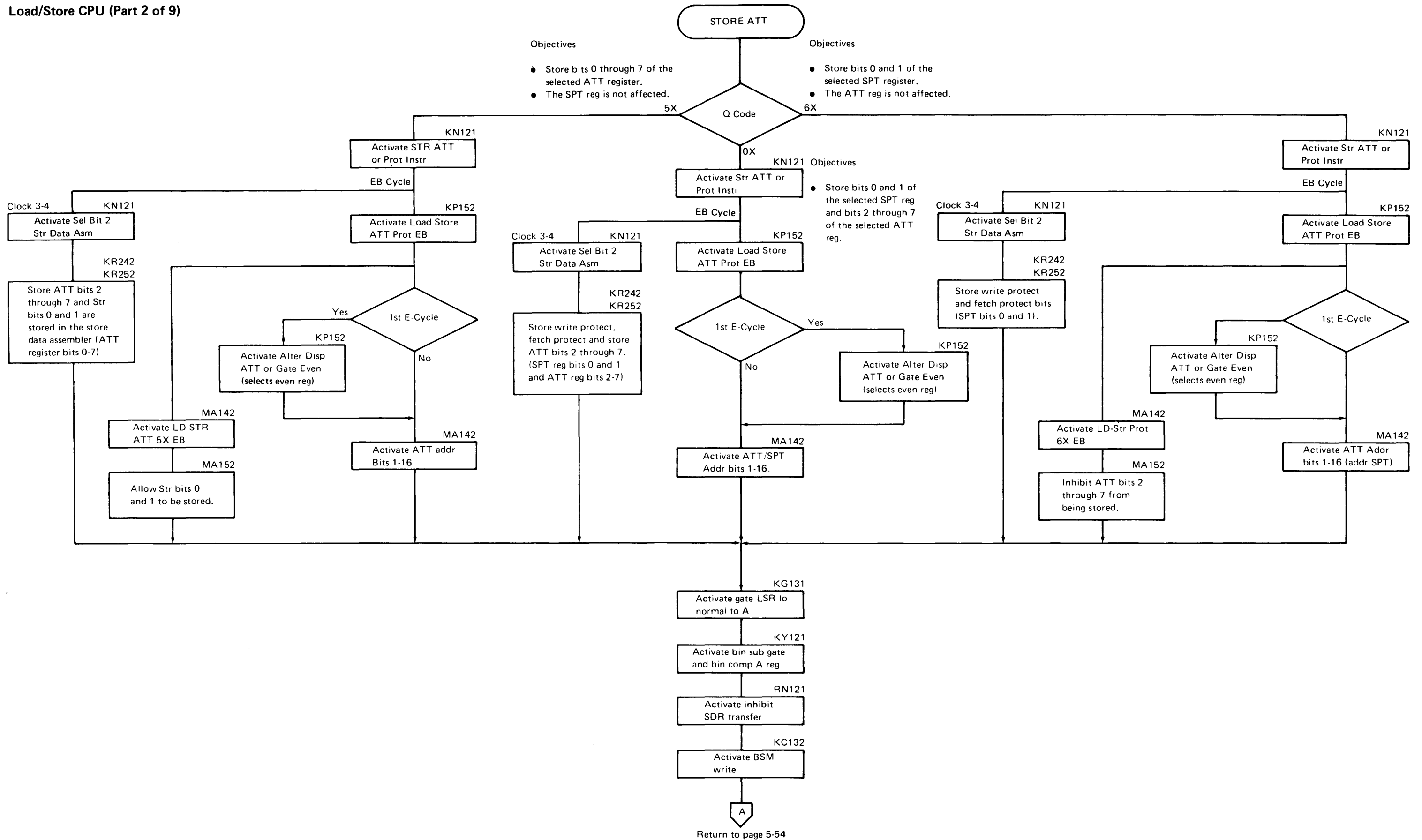


Q Code	Operation	Details
0 0 0 0 x x x x	Load ATT/SPT register	page 5-56
0 0 0 1 x x x x	Load PMR	page 5-57
0 0 1 x 0 0 0 0	Load Program Check reg	page 5-59
0 0 1 0 0 0 0 1	Memory diagnostic fetch	page 5-60
0 0 1 0 0 0 1 0	Memory FDR reset	page 5-60
0 0 1 0 0 0 1 1	Memory FDR reset and diagnos* fetch	page 5-60
0 1 0 0 0 0 0 0	Load current PMR	page 5-58
0 1 0 1 x x x x	Load ATT register	page 5-56
0 1 1 0 x x x x	Load SPT register	page 5-56

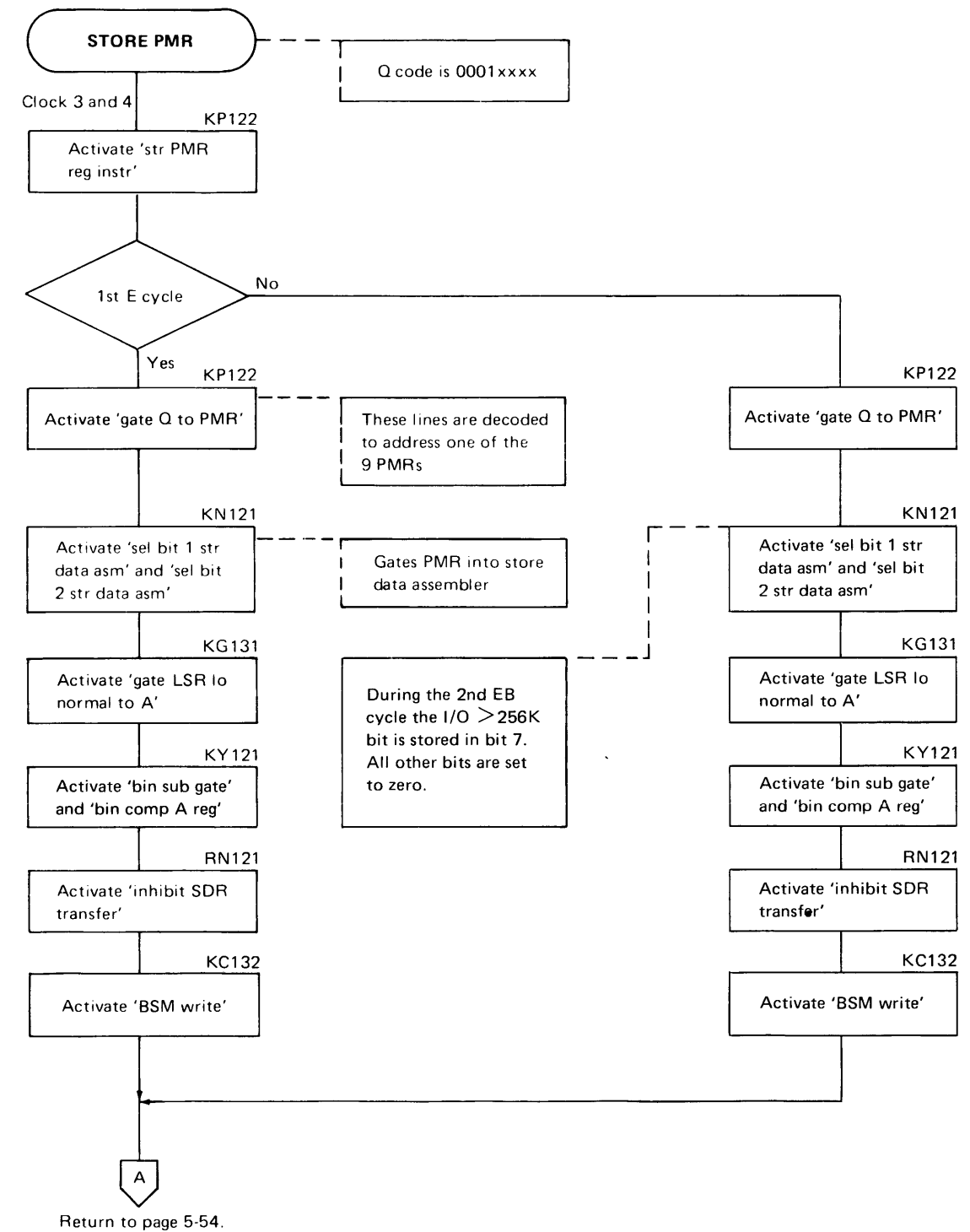
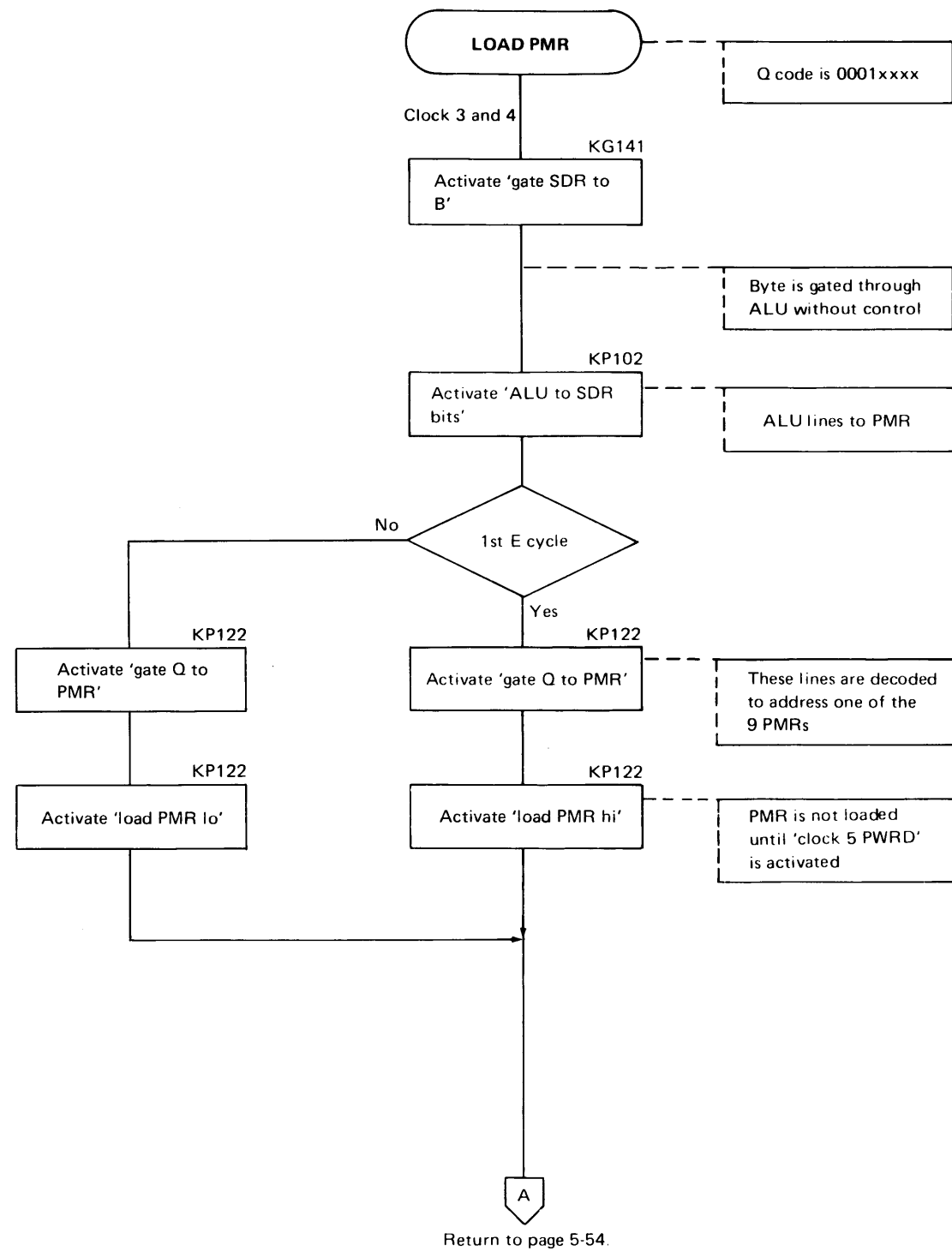
Q Code	Operation	Details
0 0 0 0 x x x x	Store ATT/SPT register	page 5-55
0 0 0 1 x x x x	Store PMR	page 5-57
0 0 1 x 0 0 0 0	Store program check register	page 5-59
0 1 0 0 0 0 0 0	Store current PMR	page 5-58
0 1 0 1 x x x x	Store ATT register	page 5-55
0 1 1 0 x x x x	Store SPT register	page 5-55

Address translation description is on page 2-54

Load/Store CPU (Part 2 of 9)

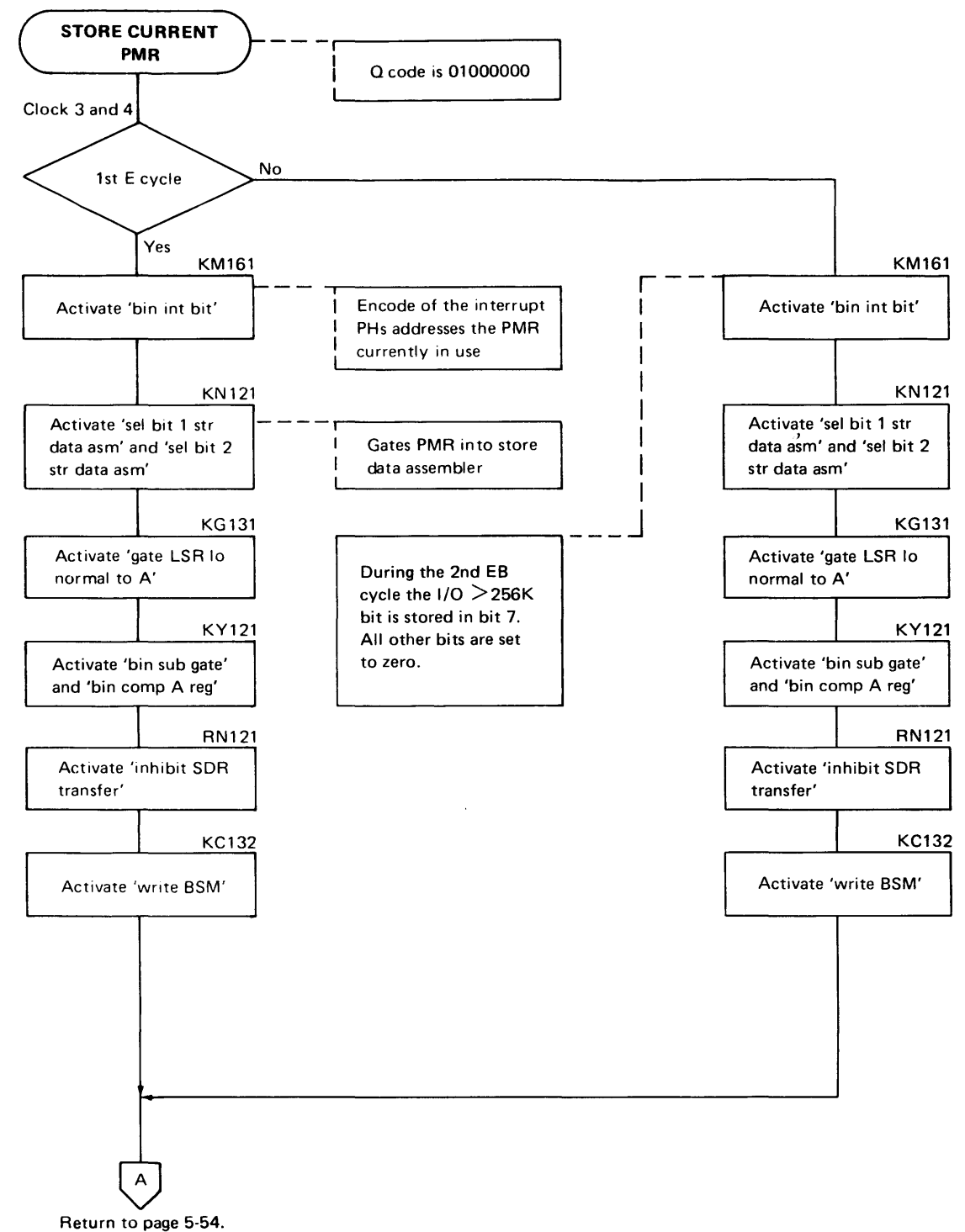
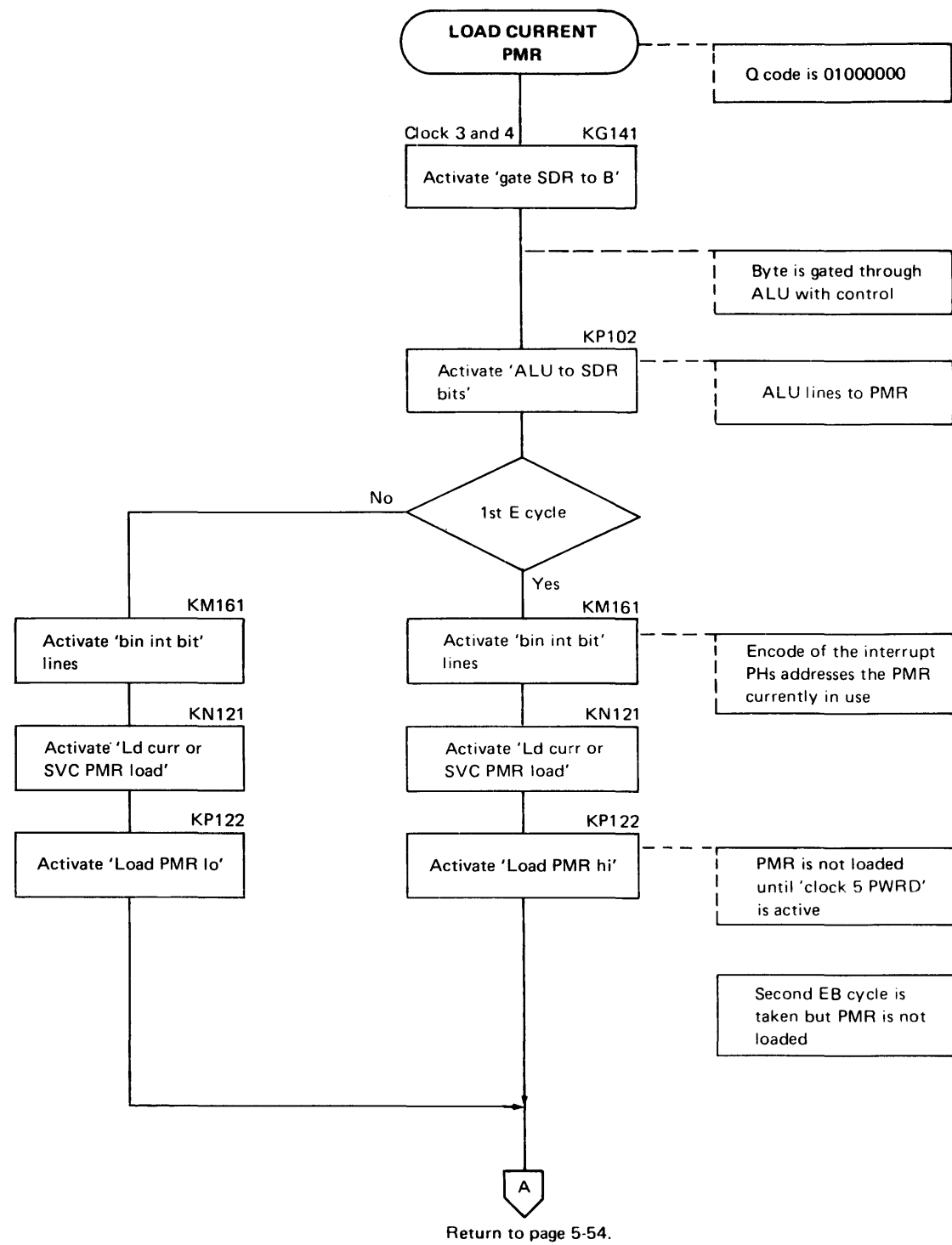


Return to page 5-54

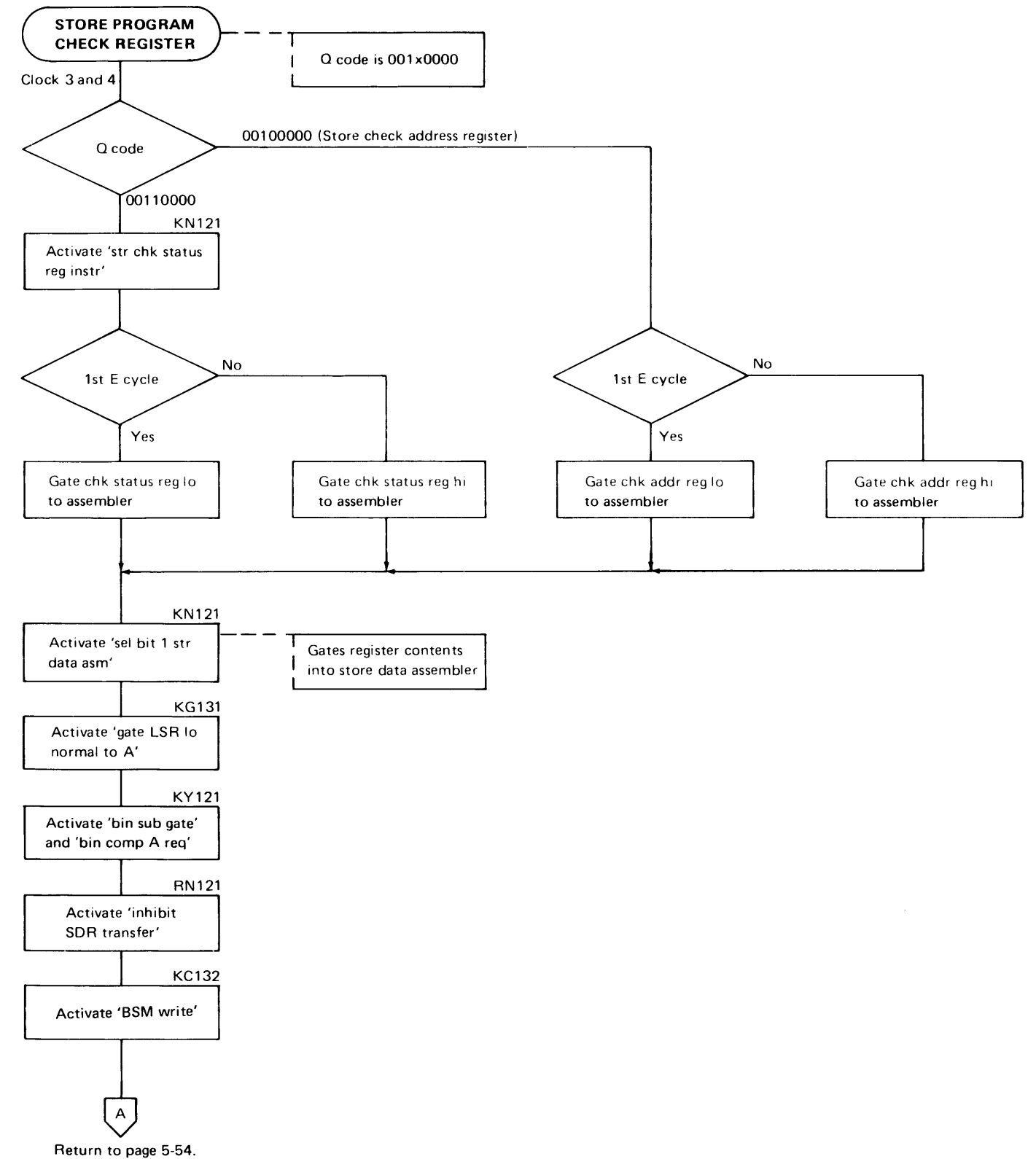
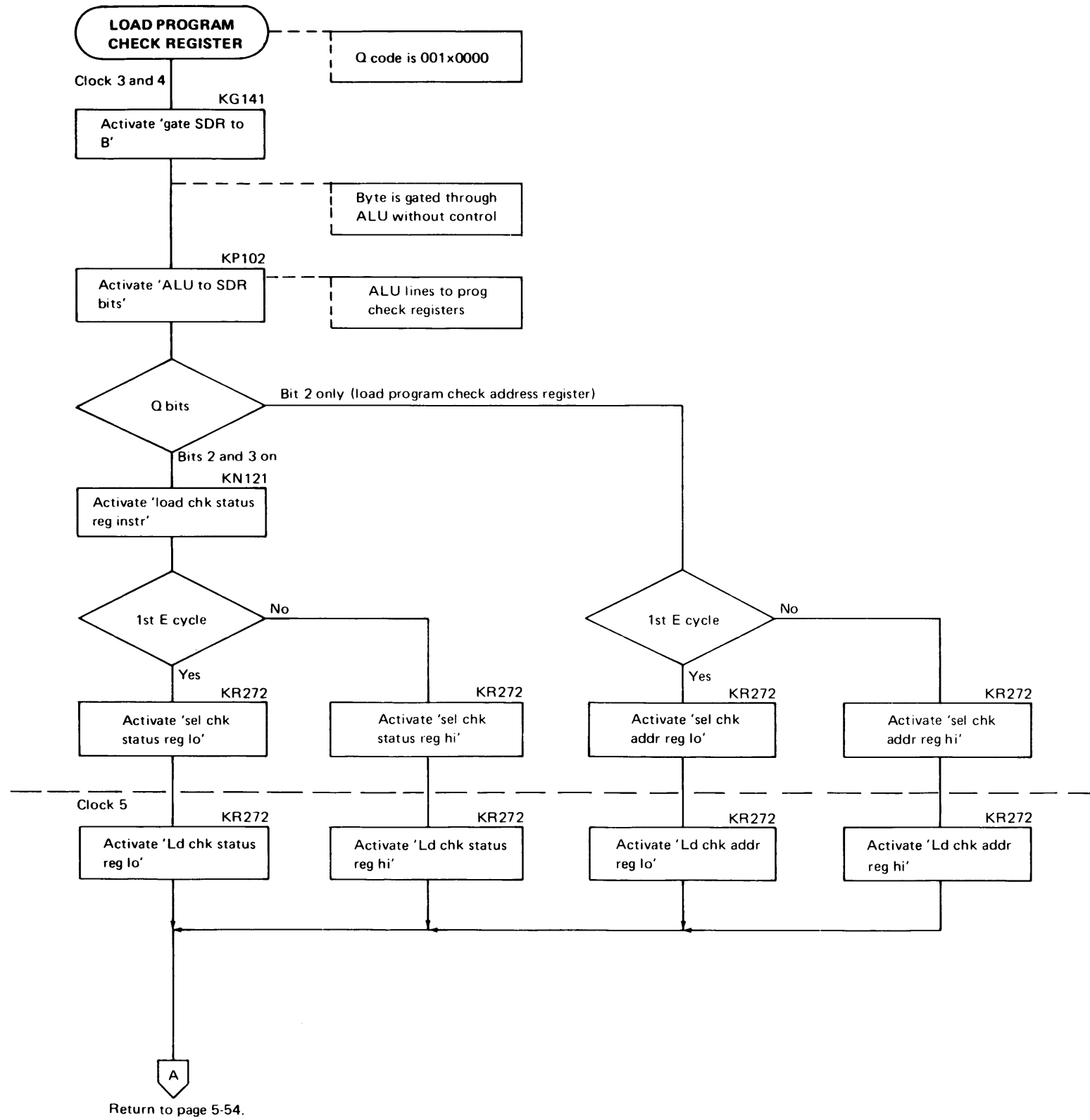




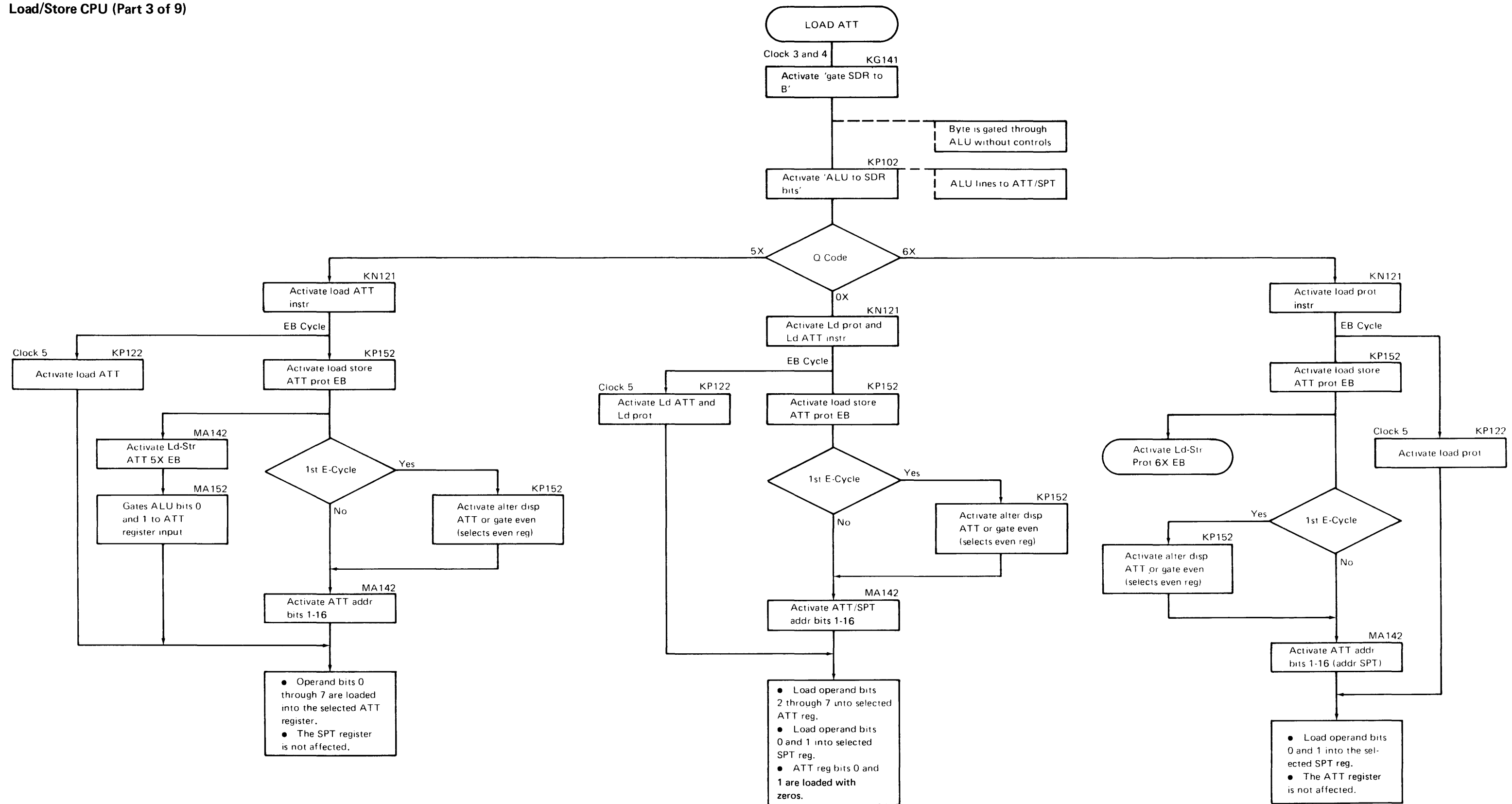
Load/Store CPU (Part 5 of 9)



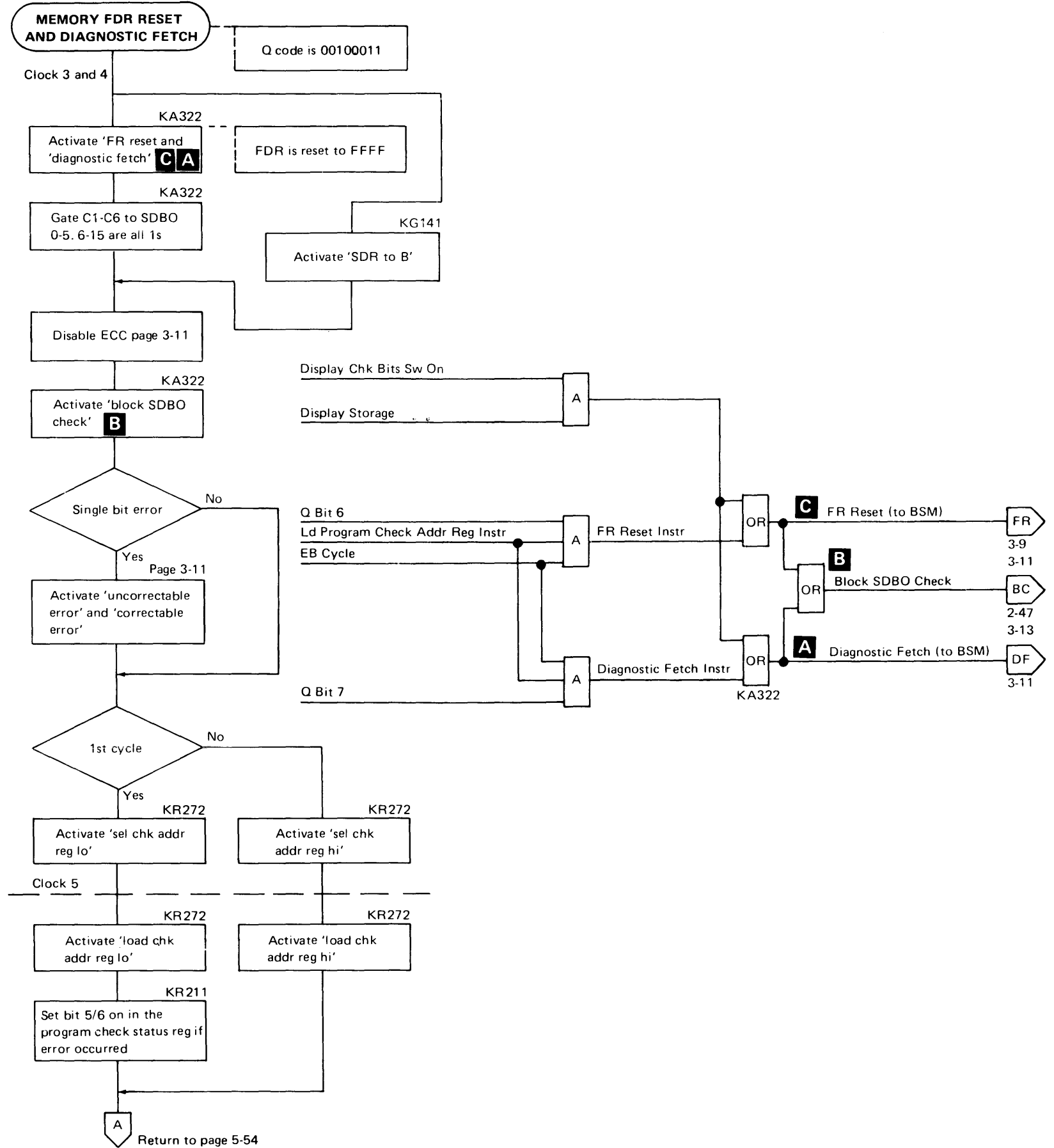
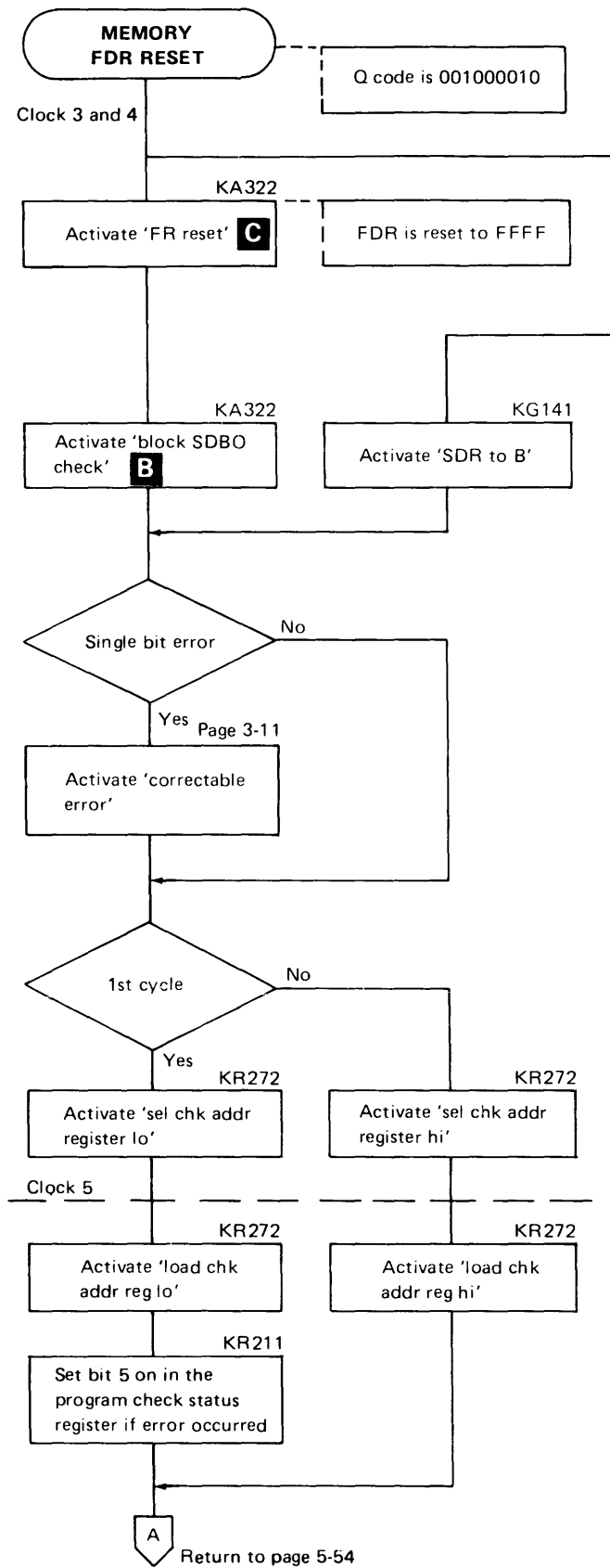
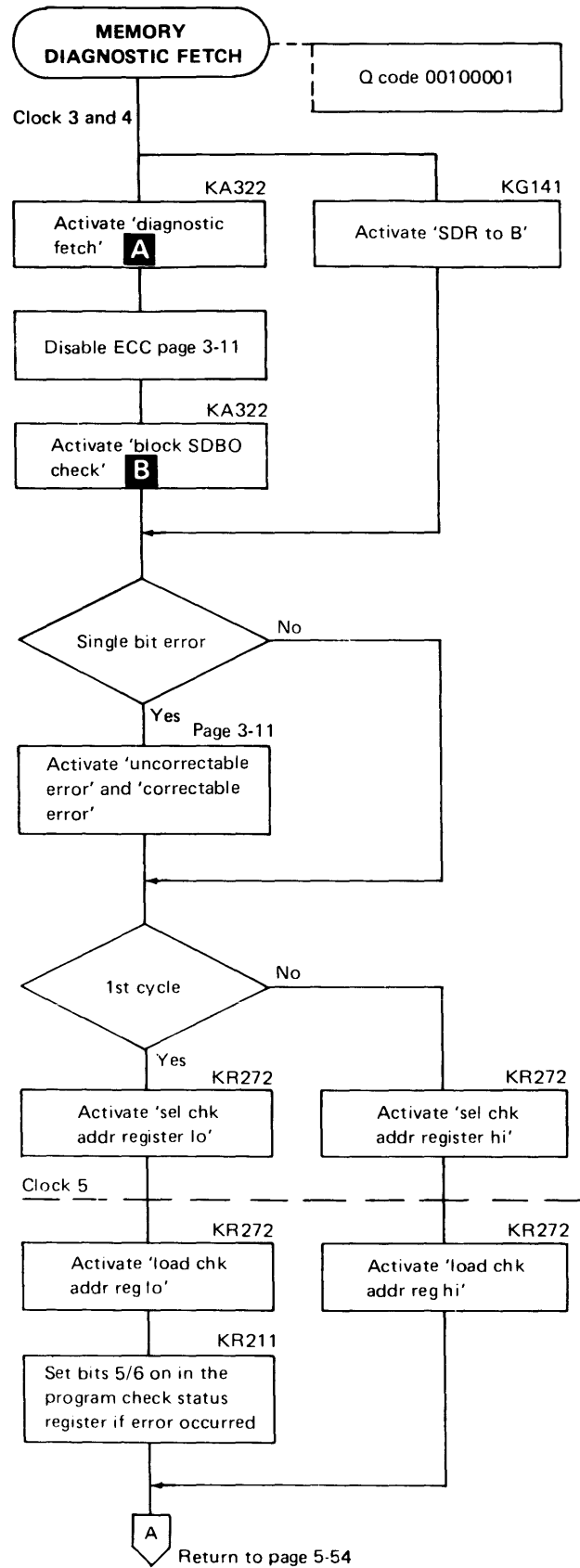
Load/Store CPU (Part 6 of 9)



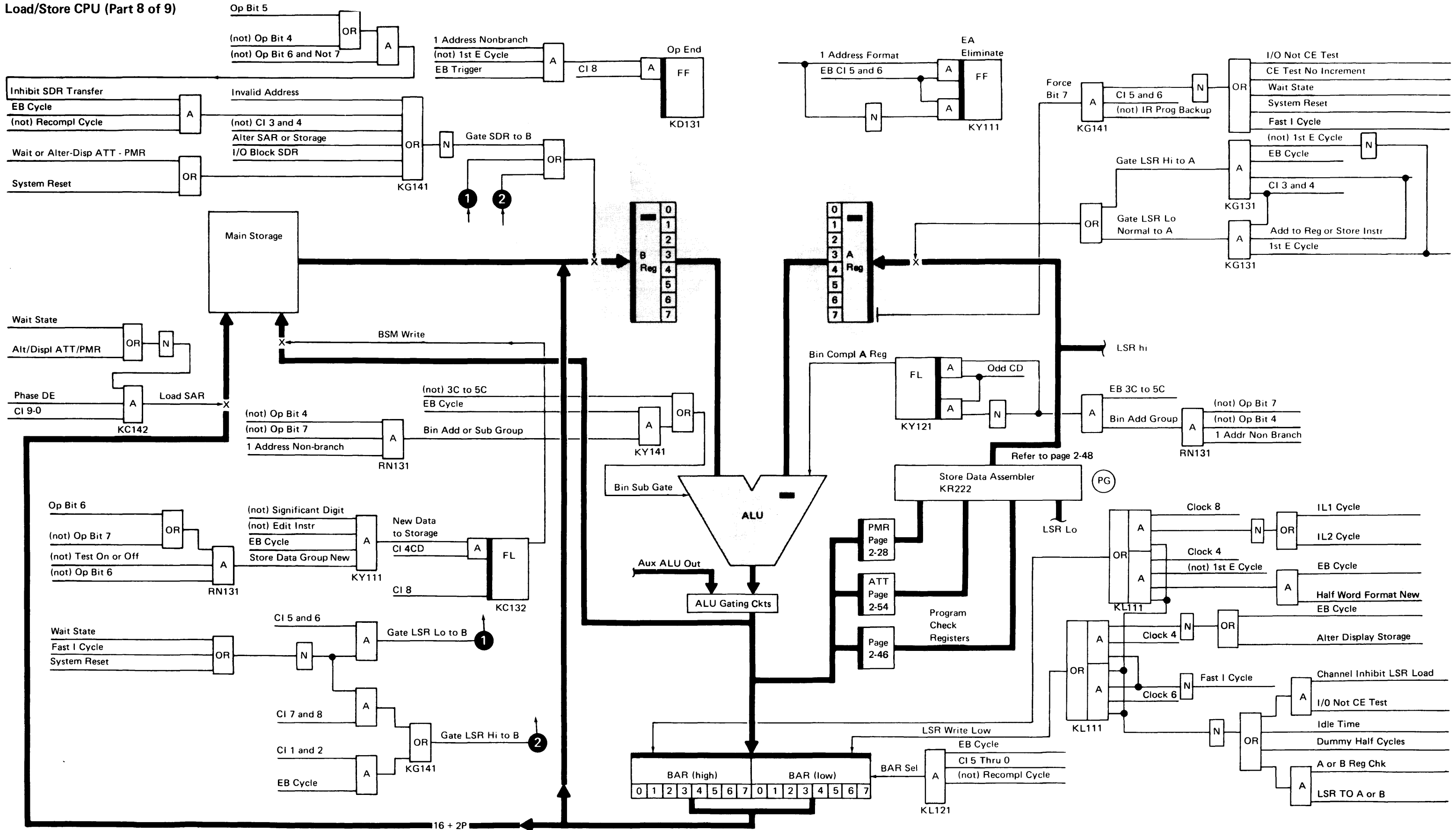
Load/Store CPU (Part 3 of 9)



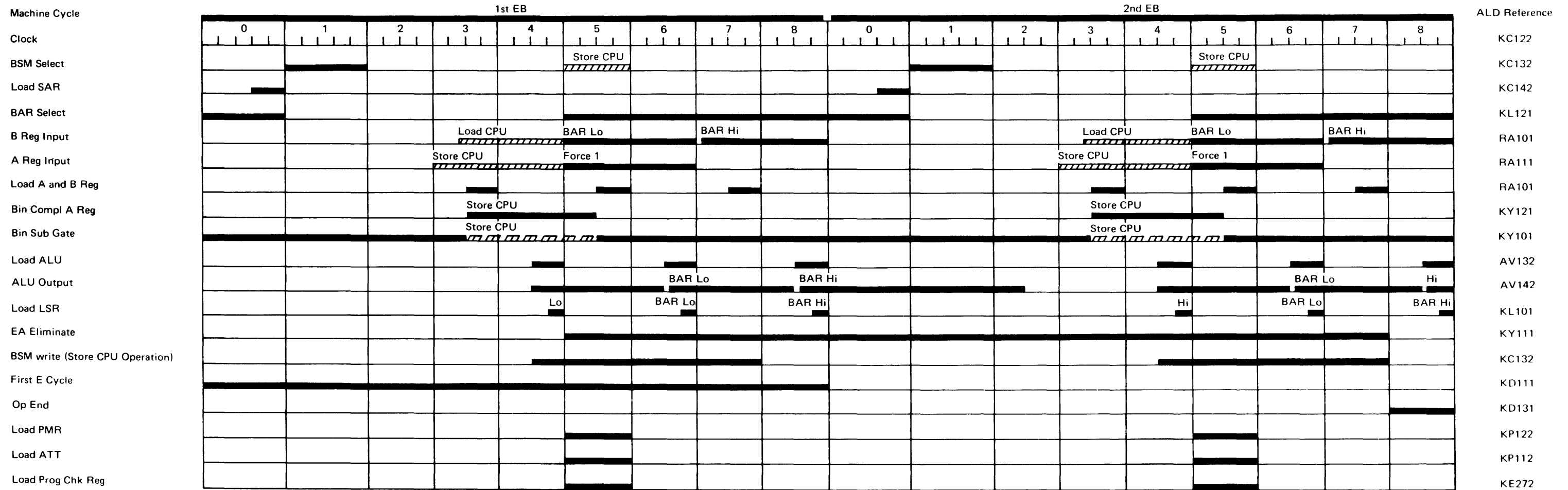
Load/Store CPU (Part 7 of 9)



Load/Store CPU (Part 8 of 9)



Load/Store CPU (Part 9 of 9)



## COMMAND INSTRUCTIONS

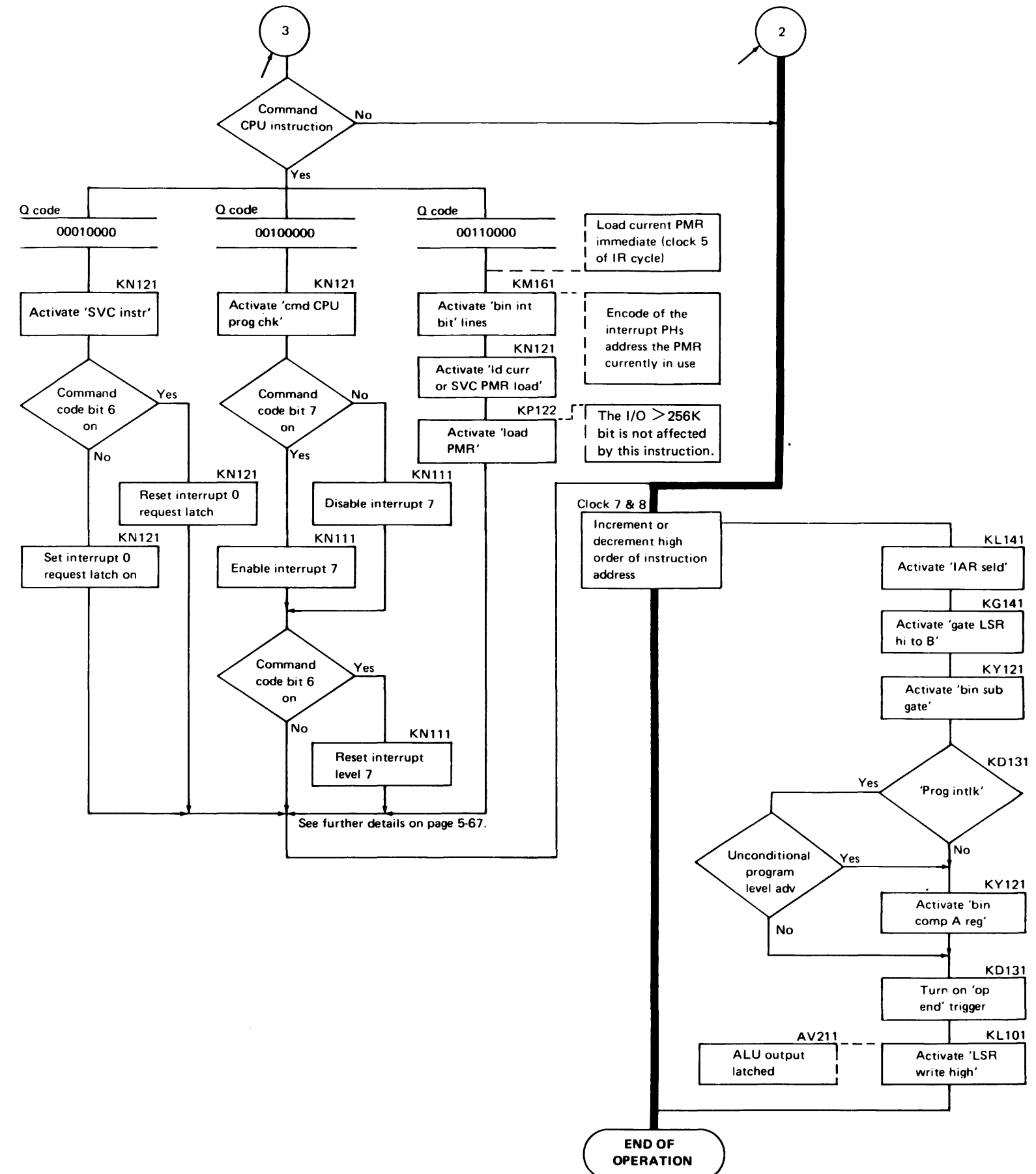
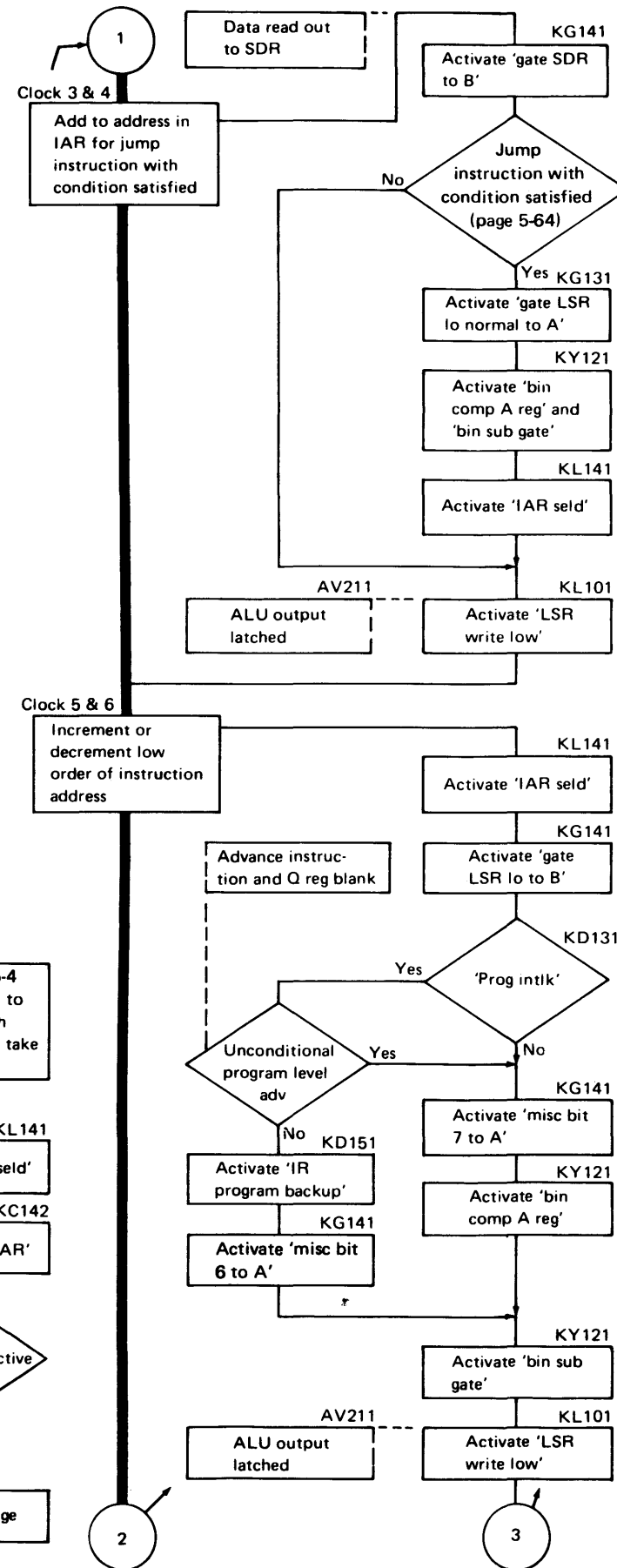
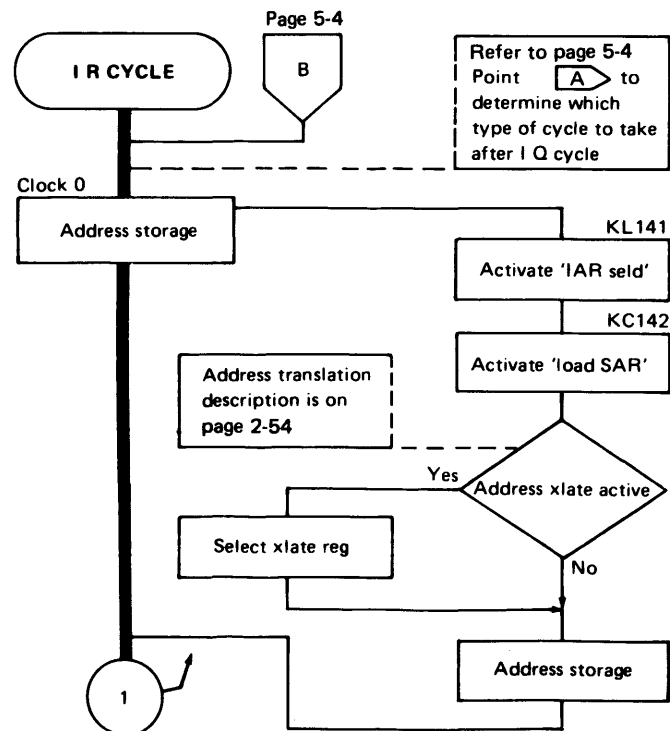
- Command instructions are: halt program level, advance program level, jump on condition, start I/O, and command CPU.
- Load operation code into op register.
- Q code used to define command.
- Control code is third byte of instruction and contains additional information pertaining to the command.

I-cycles for command operations are three cycles in length; first, an I-op cycle transfers the operation code from main storage to the op register. Second, an I Q cycle transfers the Q-code into the Q-register and DRR. If the operation is a branch or jump, the condition register is also tested for true or false. Third, an I R cycle is then used to transfer from storage the control code needed to execute the command. The details for use of the control code are discussed with specific operation descriptions.

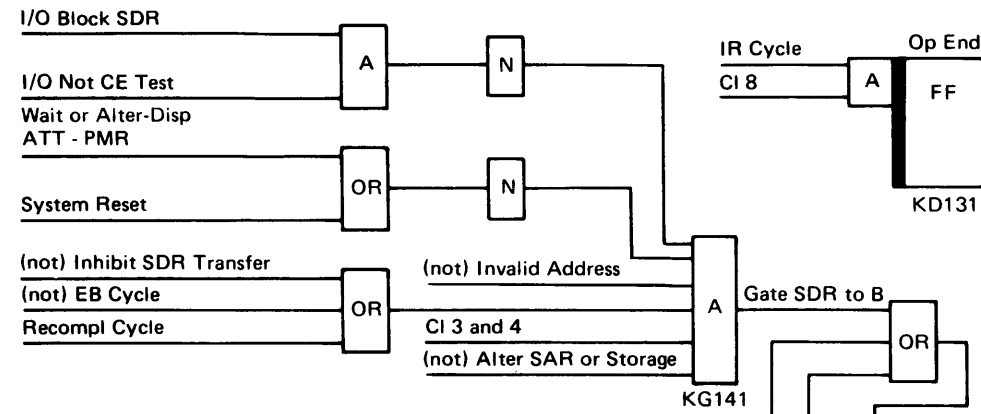
### I R Cycle (Part 1 of 2)

#### Objectives:

- Execute command instruction, unless program interlocked.
- Increment or decrement instruction address register.

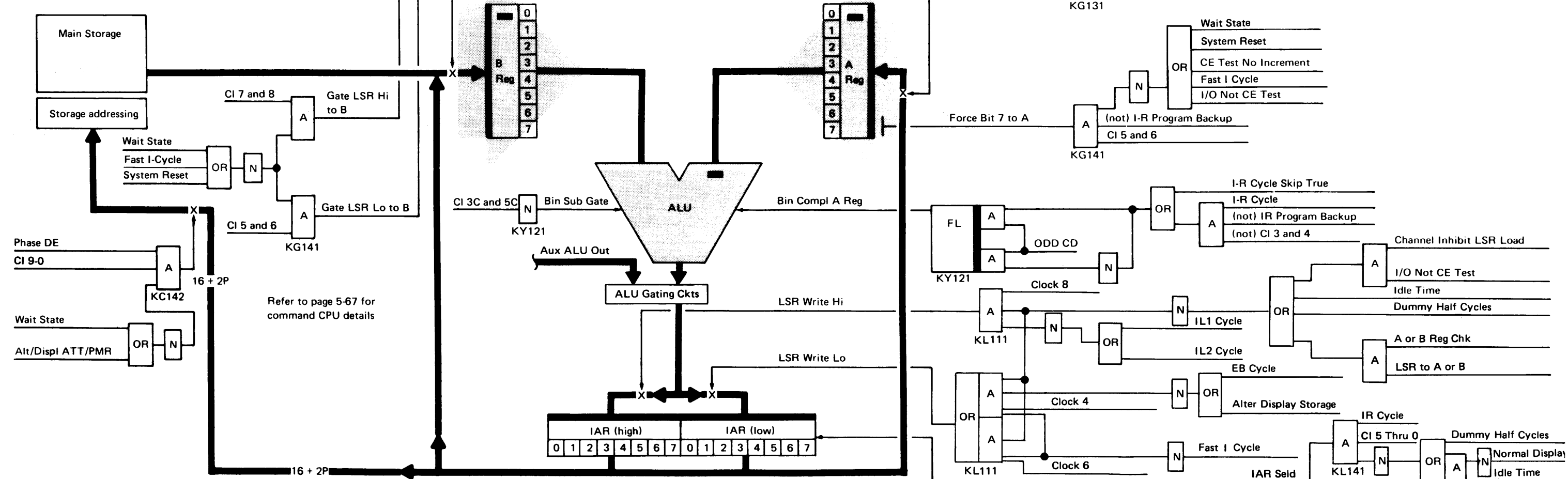


I-R Cycle (Part 2 of 2)



Cycle	0	1	2	3	4	5	6	7	8	ALD Reference
Clock	[Clock signal]									KC122
Load SAR	[Pulse at cycle 0]									KC142
BSM Select	[Pulse at cycle 1]									KC132
IAR Select	[Pulse at cycle 3]									KL141
LSR Load	[Pulse at cycle 4]									KL101
Load A and B Reg	[Pulse at cycle 1]									RA101
A Reg input	[Signal: *IAR]									RA111
B Reg input	[Signal: SDR]									RA101
Load ALU	[Pulse at cycle 2]									AV132
ALU Output	[Signal: Force bit 7]									AV142
bin Compl A Reg	[Signal: IAR Lo]									KY121
bin Sub Gate	[Signal: IAR Hi]									KY121
**Cmd CPU SVC	[Signal]									KN121
**Cmd CPU Prog chk	[Signal]									KN111
**Cmd CPU Ld PMR	[Signal]									KP102
Op End	[Signal]									KD131

\* Jump Instruction only  
 \*\* Determined by Q decode  
 Cmd CPU circuits are on page 5-67





## Jump On Condition—JC

Bits        0 1 2 3 4 5 6 7  
Op Code  1 1 1 1 0 0 1 0

- Condition register is tested for condition specified in Q code.
- If tested condition is satisfied, control code is added to IAR for next sequential instruction.
- Q code bit 0 is used to specify if jump is performed on condition true or condition false.
- Take I-R cycle.

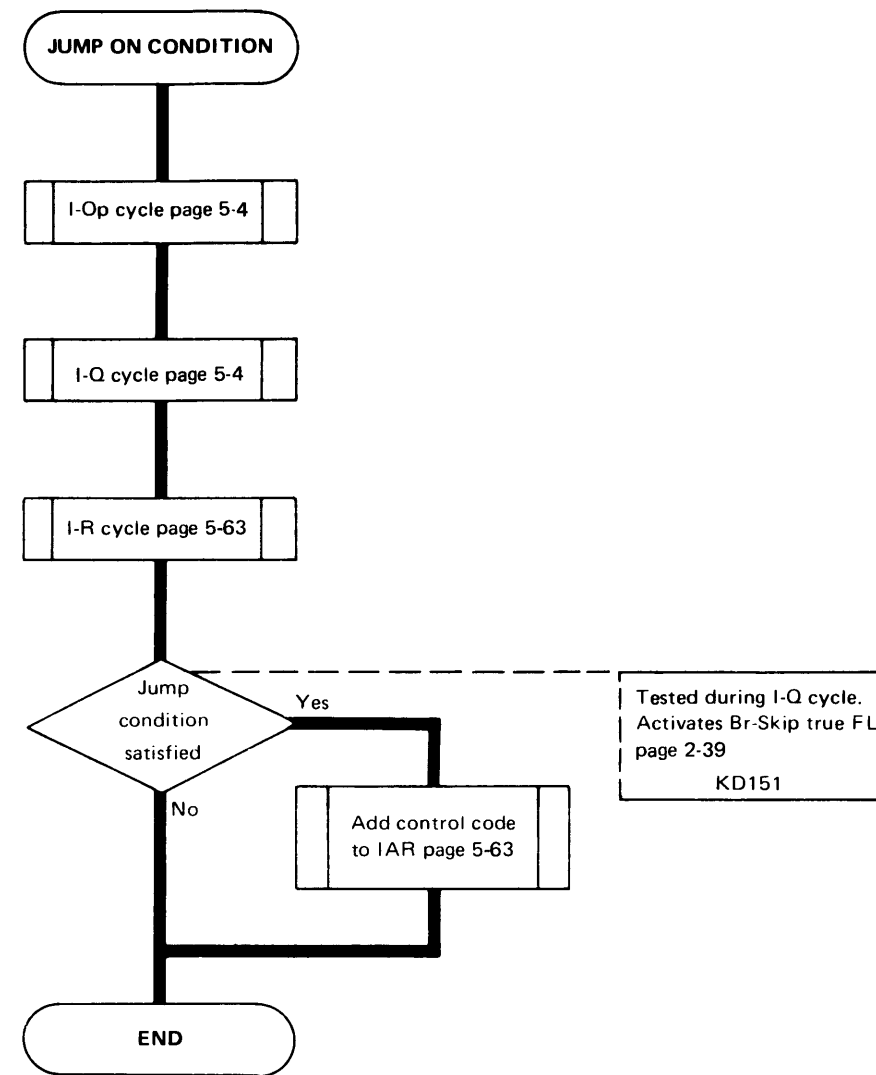
The jump on condition operation is similar to branch on condition except for the instruction address modification. If the condition register contents satisfy the condition specified in the Q code, the control code byte is added to the IAR.

Q Bit	Condition Tested
0	Presence of condition
Not 0	Absence of condition
7	Equal
6	Low
5	High
4	Decimal overflow
3	Test false
2	Binary overflow

During the I-Q cycle, the Q code is transferred from storage, through the B register and into the ALU. The condition register is decoded into the A register and enters the ALU. An ALU AND function is performed (both input bits must be the same to get an output) and the output is checked for non-zero.

Bit 0 of the Q register is used to specify if the jump is to be performed on condition true or condition false. If bit 0 is on and any one of the conditions specified by the Q code is present, the jump is performed. If bit 0 is off and all conditions specified by the Q code are missing, the jump is performed.

During the IR cycle, if the jump condition was satisfied, the IAR is selected and the control code is added to that register.



**Halt Program Level (HPL)**

Bits 0 1 2 3 4 5 6 7  
Op Code 1 1 1 1 0 0 0 0

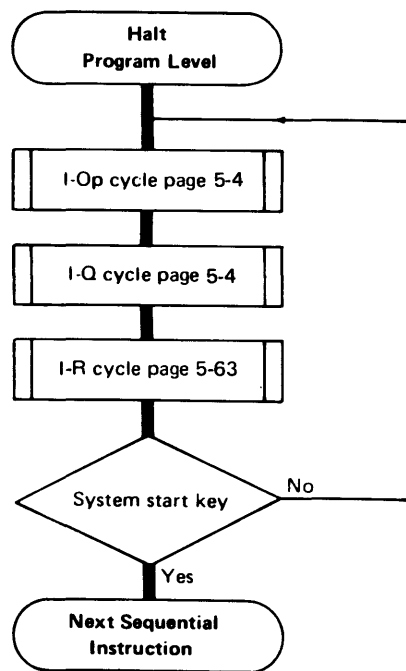
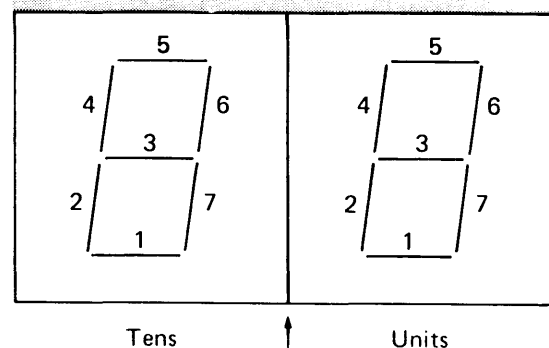
- Prevents execution of the next sequential instruction.
- Loops on halt instruction until the start key is pressed.
- Instruction format bytes two and three displayed on the console.

The halt program level instruction prevents execution of the next sequential instruction. During the I-Q cycle, the tens position of the halt identifier (instruction byte two) is displayed in the console display and program interlock is forced. During the I-R cycle the units position of the halt identifier (instruction byte three) is displayed. I-R cycle and program interlock activates I-R program back-up. Program back-up decrements the IAR by two ('force bit 6 to A') and this loop continues until the system Start key is pressed. Pressing the Start key eliminates the program interlock and the next sequential instruction is executed.

If a halt is executed during an interrupt level, program interlock is blocked and the IAR is advanced in the normal manner.



		Halt Identifiers			
F0		Tens		Units	
Char-acter	Hex Code	Display	Char-acter	Hex Code	Display
None	00		A	3F	A
1	03	1	b	79	b
2	76	2	C	6C	c
3	57	3	d	73	d
4	1B	4	E	7C	e
5	5D	5	F	3C	f
6	7D	6	H	3B	H
7	07	7	J	63	J
8	7F	8	L	68	L
9	5F	9	P	3E	P
0	6F	0	U	6B	U



Example:

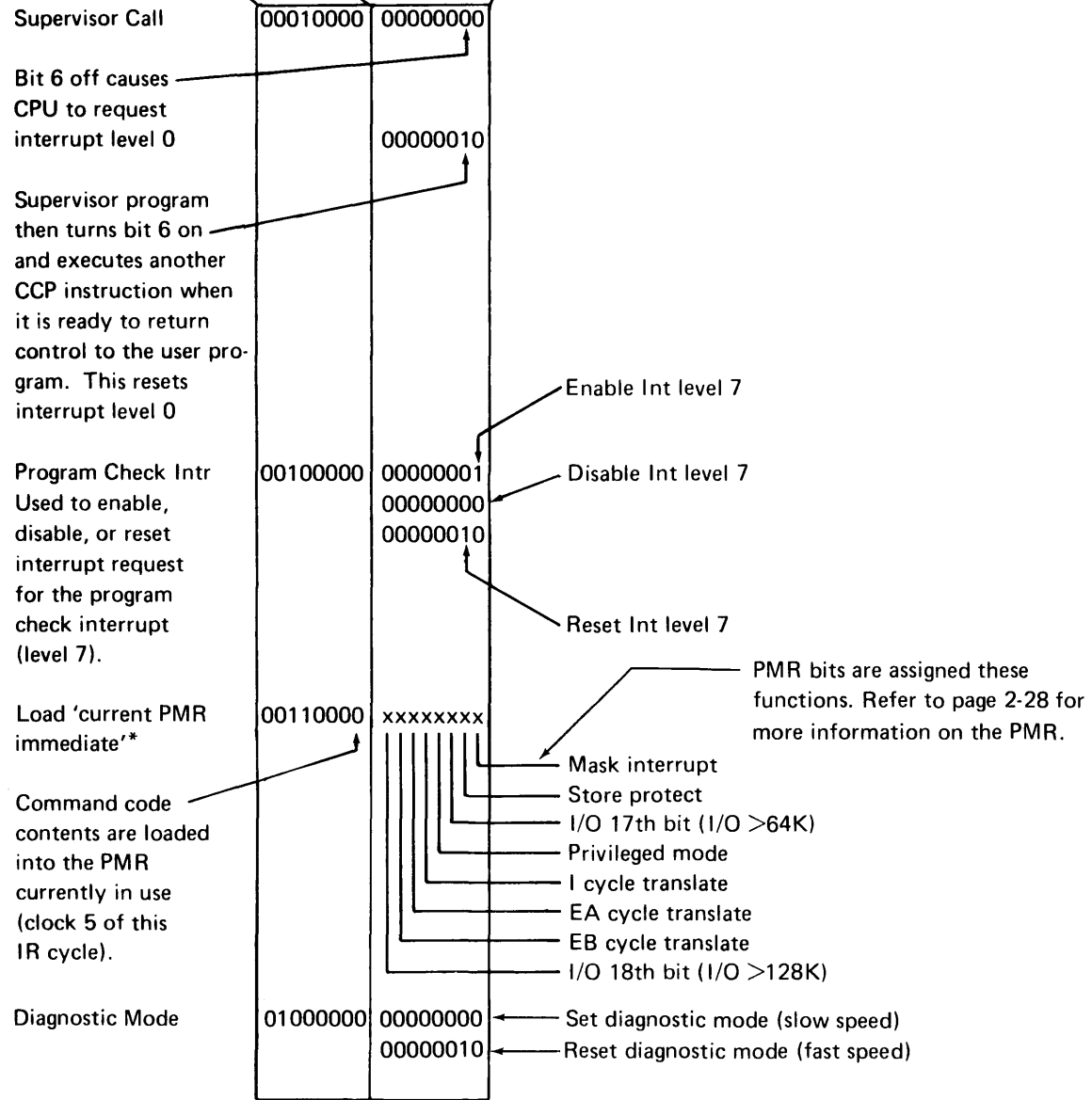
$\begin{matrix} \boxed{7} \\ \boxed{6} \end{matrix} = \begin{matrix} \text{Bits} & - & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \text{State} & - & x & 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{matrix}$   
 Hex Codes 7 6

### Command CPU (CCP)

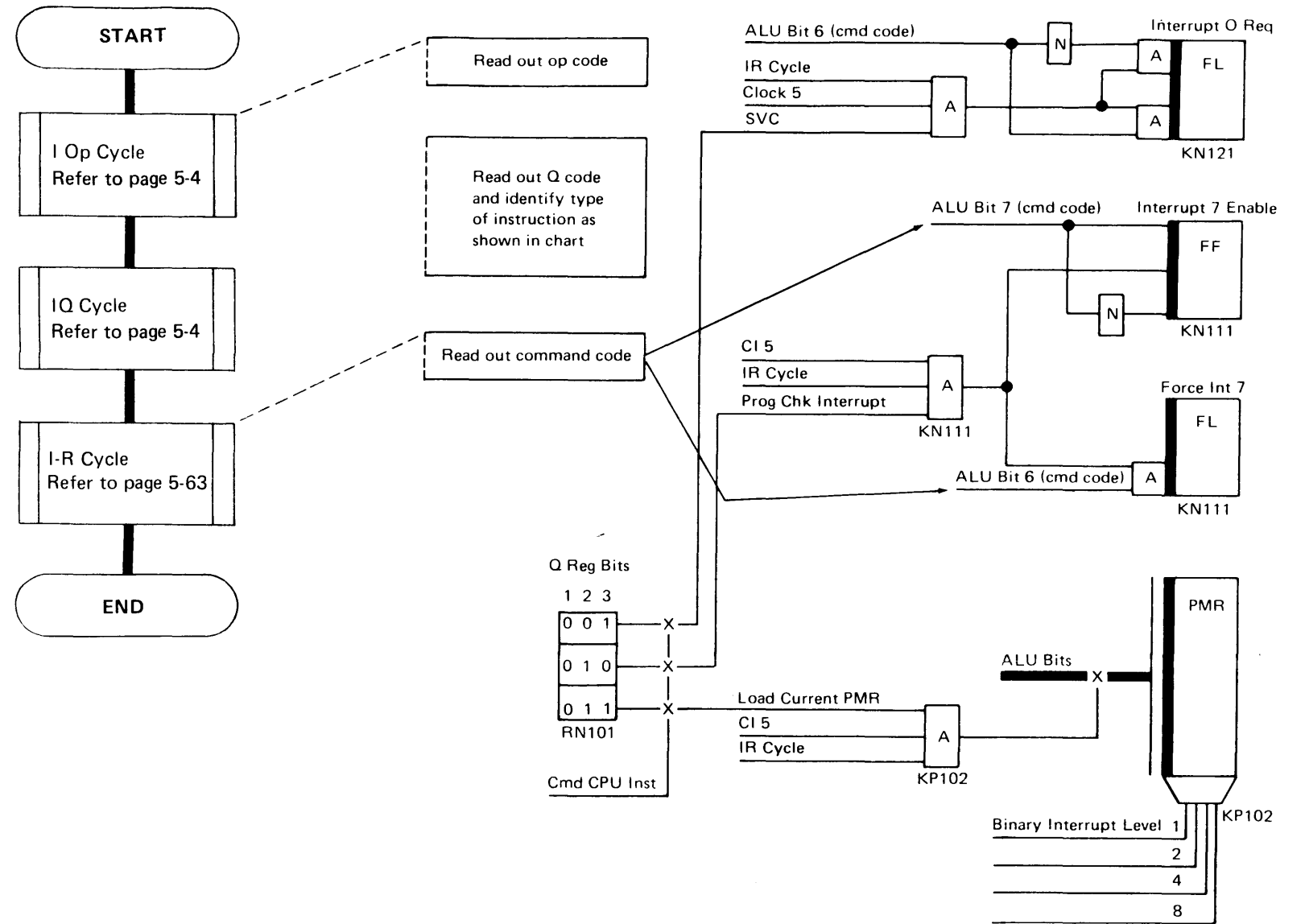
This instruction is used to control certain functions of the CPU as defined by the Q code. This page as well as the others referenced provide the instruction description.

**Op Code    Q Code    Command Code**

F	4	
---	---	--



\*This instruction will not alter the I/O > 256K bit (use the LCP instruction).



**I/O INSTRUCTIONS**

I/O instructions are of two types: one address instructions and command instructions. The Q code contains the address of the I/O device, the code for the primary or secondary unit involved, and the function to be performed (read, write, control). Where applicable, the control code contains additional information for the device (space, stacker selection, etc.).

I cycles follow the same cycle pattern as in other 1 address or command instructions. The I cycle link with the I/O attachments is discussed with the individual operation descriptions.

**Start I/O – SIO**

Bits      0 1 2 3 4 5 6 7  
Op Code  1 1 1 1 0 0 1 1

- Start an I/O device.
- Q code contains device address and function to be performed (read, punch, print).
- Control code contains additional instructions for device (space, stacker selections, etc.).
- Enable/disable op-end interrupt.

The start I/O instruction starts the mechanical function of any I/O device. The particular device selected and the function to be performed are determined by the Q code of the instruction. The device address (DA) is contained in bits 0-3 of the Q code. Bit 4 of the Q code contains the modifier (M) bit that selects the primary or secondary unit of the device **A**. Bits 5-7 contain the N field which is the function to be performed by the device. The following three N codes are common to all devices, but the remainder are assigned by the individual devices.

Bits      N Field  
5 6 7      Function

0 0 0    Control or Equivalent  
0 0 1    Read or Equivalent  
0 1 0    Write or Equivalent  
\*       

\*Additional codes determined by the individual devices.

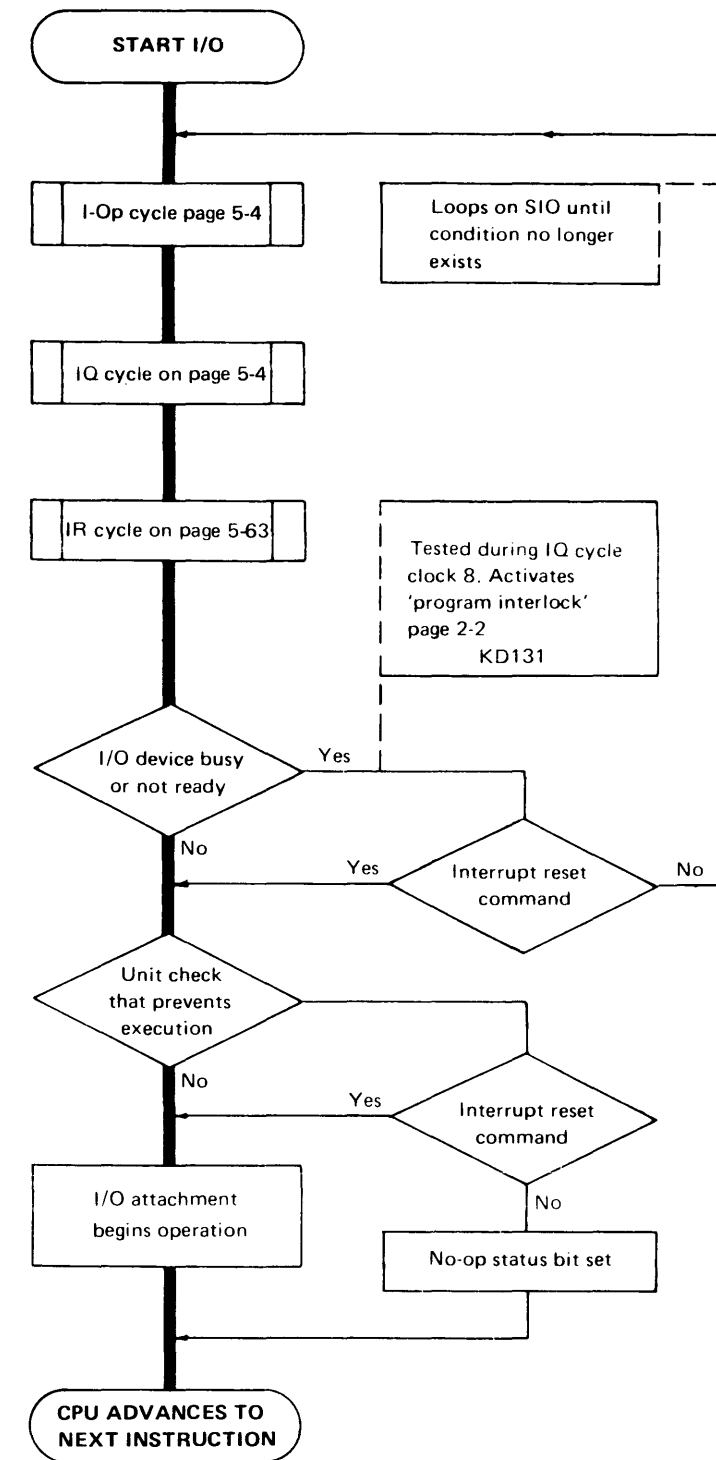
The control code byte of the instruction further defines the device function. For instance, it may define the stacker pocket selected in the MFCU or it may define the type of spacing for the line printer. Use of the control code varies with the individual devices. Refer to the manual for the various I/O attachments.

CPU control of the operation ends with the IR cycle of the instruction. If the addressed device is not busy or does not need attention and the Q byte and control byte reach the device without error the CPU continues with the next sequential instruction. If the device cannot execute the command, the program loops on the SIO instruction until the device is no longer busy or until the operator has corrected the attention condition.

Once the device has accepted the instruction, the operation is performed by the attachment circuitry. Whenever the device needs data from storage (write, print, punch) or has data to send to storage (read) the attachment breaks into the program to use the number of cycles it requires.

**A**

Device Address	M Bit	Assigned Device
Bits 0 1 2 3	Bit 4	
0000	0	CPU console
0000	1	Interval timer/Unit record restart
0001	0	} 3277 Display station and keyboard
0001	1	
0010	0	MLTA individual line/BSCC
0010	1	MLTA general adapter
0011	0	SIOC
0011	1	2501 Card Reader
0100	0	3741 Data Station
0100	1	Display Adapter
0101	0	1442
0101	1	Display Adapter
0110	0	Magnetic tape drive 0
0110	1	Magnetic tape drive 1
0111	0	Magnetic tape drive 2
0111	1	Magnetic tape drive 3
1000	0	BSCA1
1000	1	BSCA2/Display Adapter
1001	-	Spare
1010	-	Spare
1010	-	Spare
1011	-	Spare
1011	-	Spare
1100	0	3340 disk drive 1
1100	1	3340 disk drive 2
1101	0	3340/3344 disk drive 3
1101	1	3340/3344 disk drive 4
1110	0	1403 printer
1110	1	1403 printer - diagnostics
1111	0	5424 MFCU/2560 MFCM
1111	1	5424 MFCU/2560 MFCM

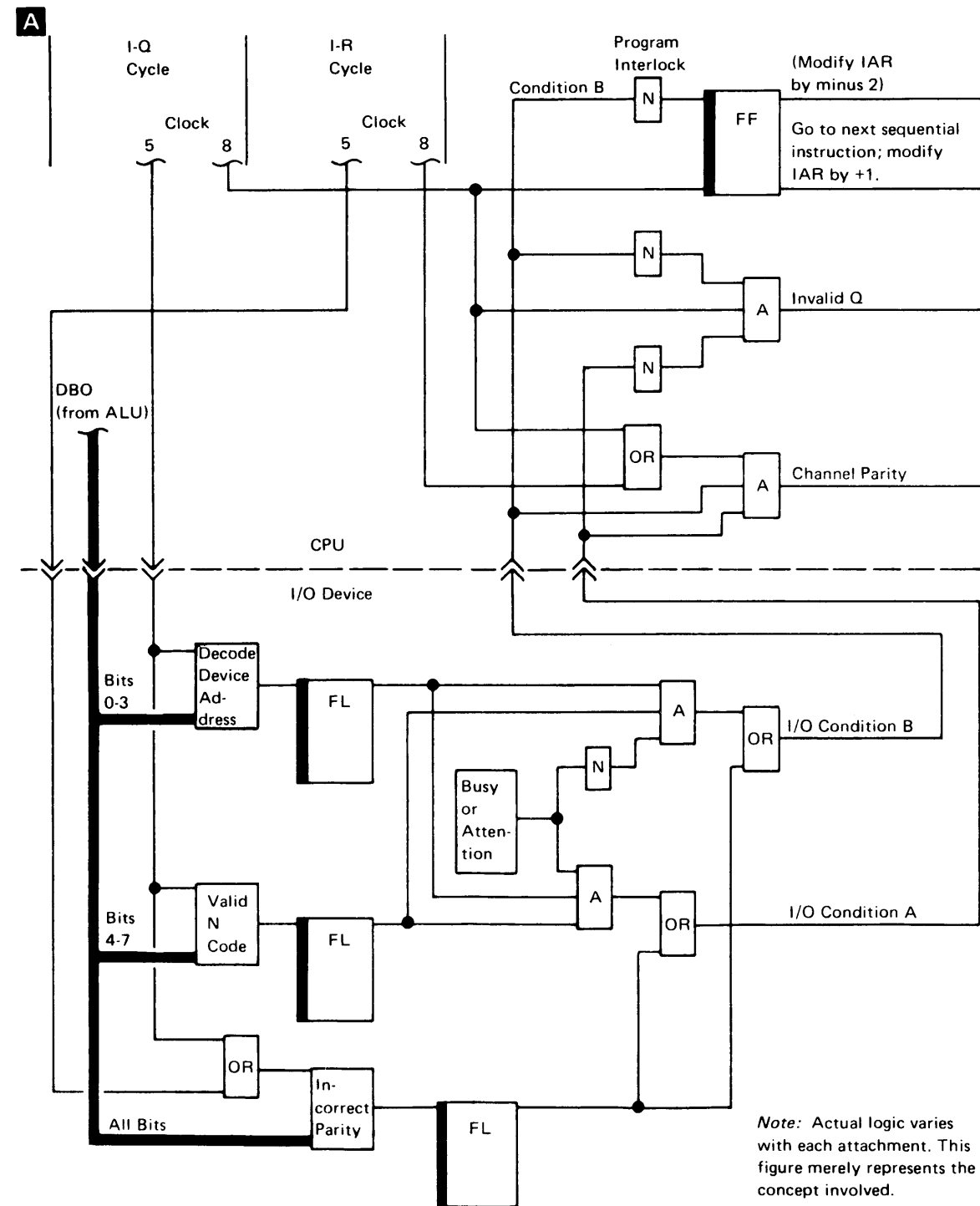


If unit check exists which will not prevent execution of SIO, instruction is executed and unit check is reset.

## I/O Instruction I-Cycles

At clock 5 time of the I-Q cycle, when the Q code is latched in the ALU latches, the ALU output is sent to the devices on the DBO. This Q byte by-passes the DBO translator.

Each device attempts to decode the address. If any device recognizes the address and if the Q byte contains a valid N code, the device activates either I/O condition A or I/O condition B. **A**



Note: Actual logic varies with each attachment. This figure merely represents the concept involved.

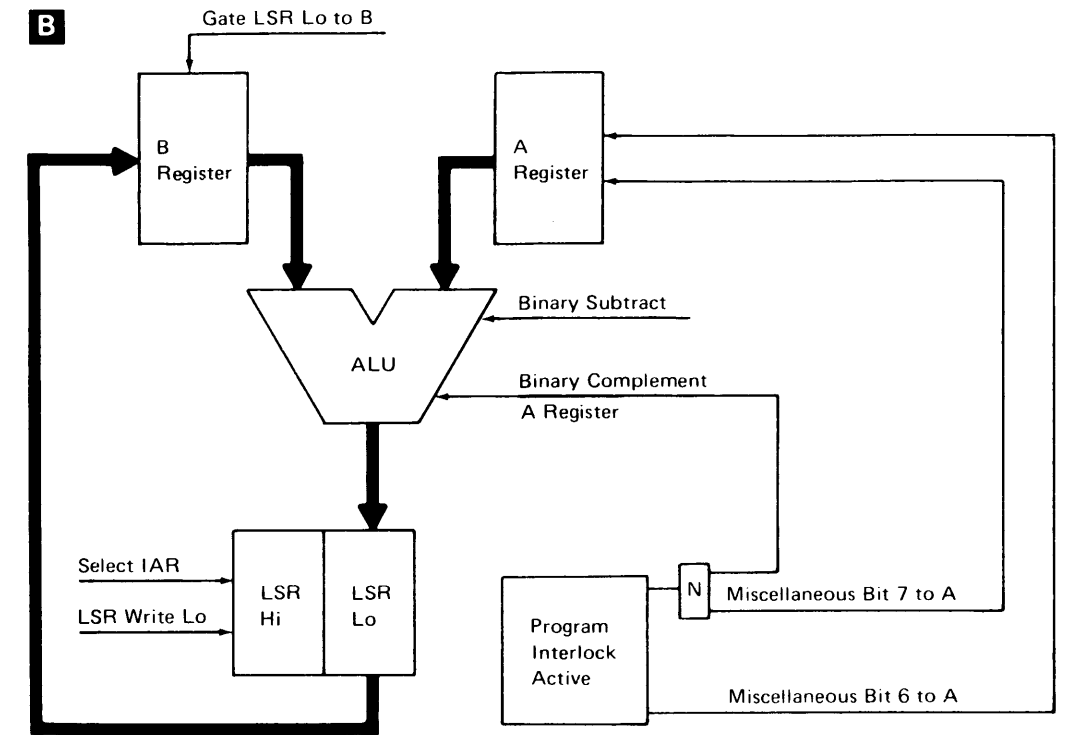
If the device is 'busy' or 'not ready and no errors', 'I/O condition A' only is activated and at clock 8 time of the I-Q cycle, the CPU activates 'program interlock'. If the device is able to execute the instruction, I/O condition B only is activated blocking the gate to the 'program interlock' trigger. If device is 'not ready with errors', I/O condition B is activated and device will no-op instruction.

IAR modification is dependent upon the 'program interlock' trigger. During I-R cycle clock 5 and 6 time, if program interlock is inactive, 1 is added to the IAR to increment it in the normal manner **B**. If, however, 'program interlock' is active, the IAR is decremented by 2 to retry the instruction. During clock 7 and 8 the ALU controls remain the same as clock 5 and 6 to modify the high order position of the IAR.

During the I-Q and I-R cycles, the DBO is also parity checked. If a parity check occurs the processor is signaled by activating both I/O condition A and I/O condition B. **A**

The following table shows the significance of all settings for the 'I/O condition A' and 'I/O condition B' lines.

Line Activated By Any Device	Significance
'I/O condition B' only	Correct address, valid N code, device not busy and does not need attention—instruction accepted.
'I/O condition A' only	Correct address, valid N code, device busy or needs attention—instruction rejected.
Both lines	Incorrect parity—causes processor check and DBO parity check.
Neither line	Invalid address or N code—causes processor check and invalid device address.



**I/O Cycle**

Device controls data flow and functional unit control lines within CPU.

I/O cycles follow the same general data manipulation procedure as CPU cycles. That is:

- Clock 0 – address storage
- Clocks 1 and 2 – Miscellaneous (generally LSR alteration)
- Clocks 3 and 4 – compute (data manipulation between CPU and I/O device)
- Clocks 5 through 8 – address register modification.

If an I/O cycle does not require all of these functions, the I/O device blocks them by controlling the CPU data flow control lines. The following examples represent a method of transferring data between the CPU and I/O devices. The actual method used depends upon the result desired by the individual attachments.

**Example 1:**

A device is given the cycle steal assignment at clock 7D prior to the actual I/O cycle. In order to have SAR loaded at clock 0 of the I/O cycle, the assigned device must raise the proper LSR select lines at clock 8 prior to the I/O cycle. The LSR select buffer will select the LSR during clock 0. The device can select LSRs at four times during the I/O cycle. It may select the same LSR, a different LSR, or no LSR.

The A register is loaded at odd CD clocks with the information that the device puts on DBI at even clocks. This data may be translated from card code to hexadecimal during clocks 2 and 3.

The B register is loaded at odd CD clocks with the following data:

Clock	B Register Data
1CD	00
3CD	Contents of SDR or 00 Controlled by attachment
5CD	Selected LSR low-order byte or 00 if no LSR is selected
7CD	Selected LSR high-order byte or 00 if no LSR is selected

ALU output during the I/O cycle is the contents of the B register minus the contents of the A register or the contents of the B register plus the contents of the A register. This is determined by the device through 'chan bin sub'. ALU output is the following data at the following clock times:

Clocks	ALU Output Data
2D to 4C	± DBI
4D to 6C	B Register ± DBI
6D to 8C	LSR low ± DBI
8D to 2C of next cycle	LSR high ± DBI

DBO equals the ALU outputs as latched except during clocks 7D to 0C. This allows the device to use ALU out data.

If the device needs data from storage (MFCU punch) the device does not block the 'gate SDR to B' line and the data is transferred from storage to the B register (next page). No data is entered into the A register from DBI so the B register is binary added to the blank A register to effectively move the B register through the ALU. The data is latched into the ALU latches and is available to the I/O device on DBO. The device activates the 'chan in transl out' line if the byte needs to be translated to card code as in an MFCU punch operation.

Not all I/O cycles move the data unchanged through the ALU. The following example shows the function of each cycle taken by the line printer to print a character. The objectives of the three cycles in the example are:

1. Remove the data byte to be printed from storage and retain it.
2. Remove the chain image character from storage and compare it with the data byte to see if the chain is in the correct location to print the character.
3. Place the value 4/0 into the data byte location as an indication that the character has been printed.

**Example 2:**

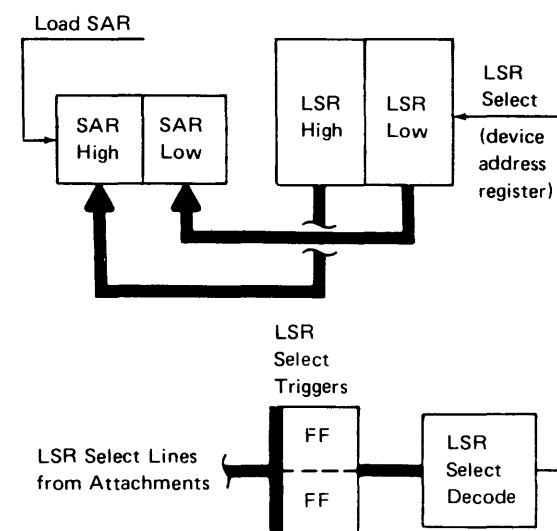
In the first print cycle steal, the LPDAR (line printer data address register) addresses storage. The byte of data is added to zeros in the A register to move the byte through the ALU and is sent to the printer attachment where it is retained.

In the second cycle, the LPIAR (line printer image address register) addresses storage and the chain image character is read out and placed in the B register. The byte of data transferred to the attachment during cycle 1 is sent back to the CPU from the printer attachment and enters the A register from DBI. The printer attachment activates the 'chan in bin sub' line to subtract the A register from the B register. The result is sent to the printer attachment. If the result is zero, the two characters are the same and the printer prints the character.

In the third print cycle, the LPDAR again addresses storage and the printer attachment activates the 'I/O block SDR' line to prevent the character from entering the B register.

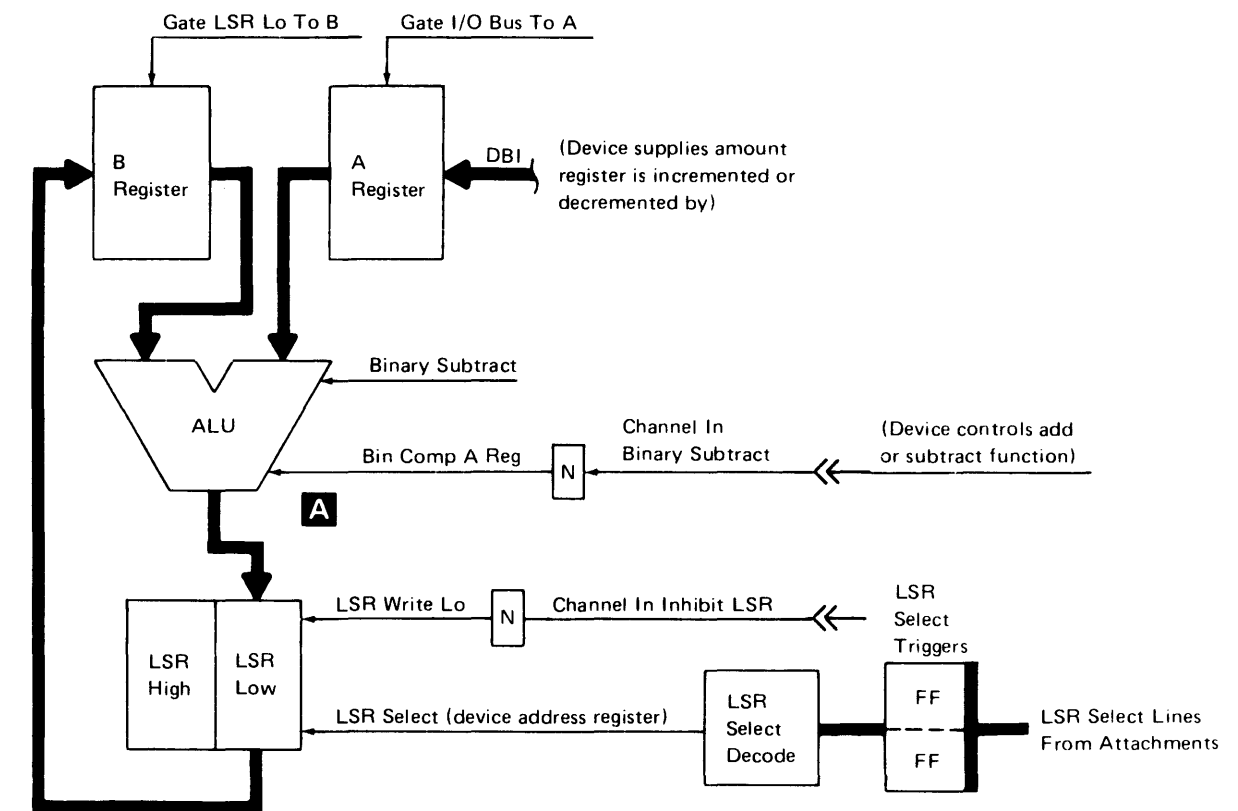
The printer attachment sends the hex value 4/0 to the CPU on DBI. This value is added to the zeros in the B register to move the 4/0 through the ALU. The printer attachment activates store new to enter the 4/0 into storage.

Address modification is the same as in a CPU cycle except that register selection, ALU controls, and the modification amount are all controlled by the I/O device. During clock 5 and 6, the LSR is selected in the same manner as during clock 0.

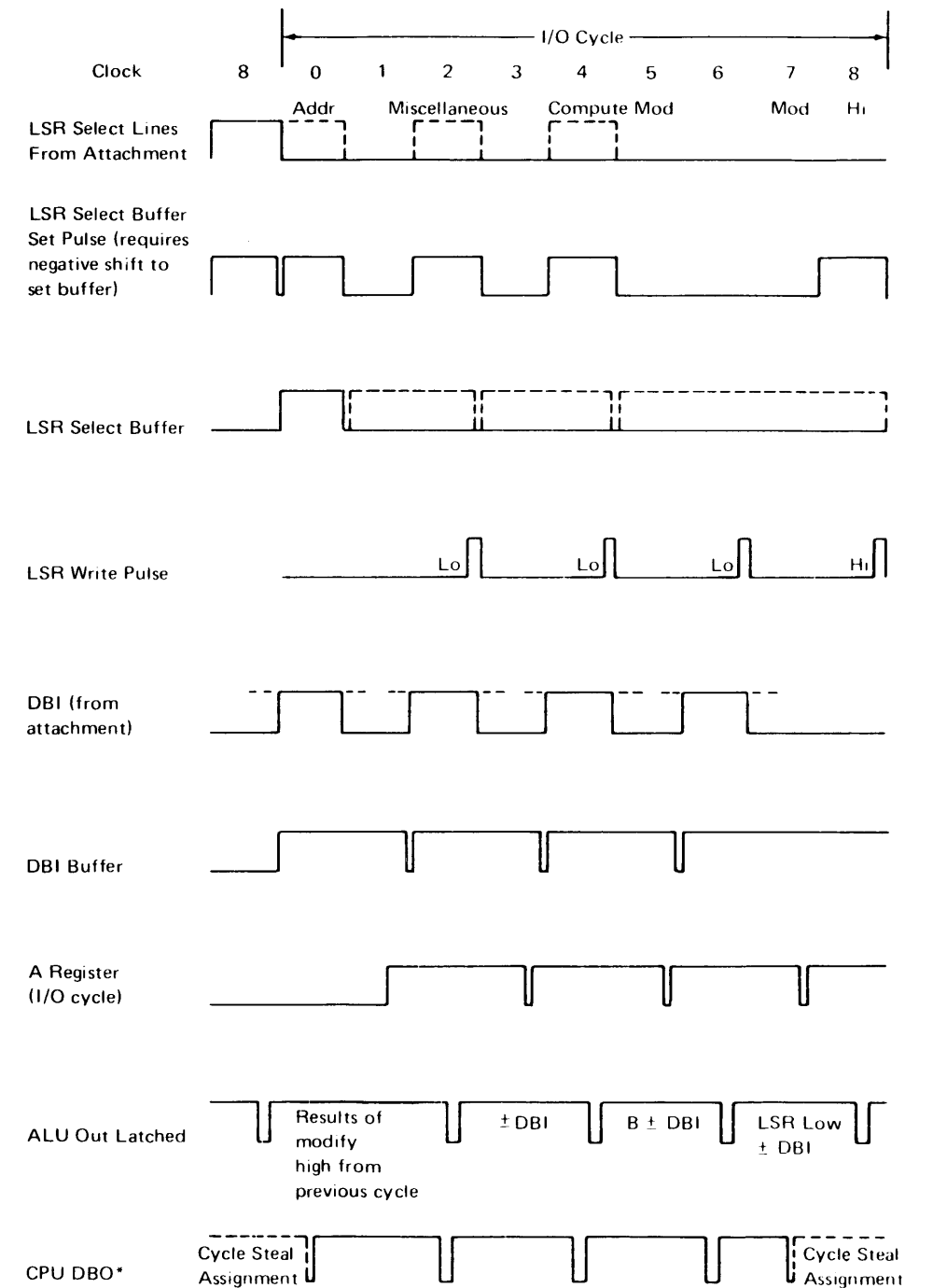
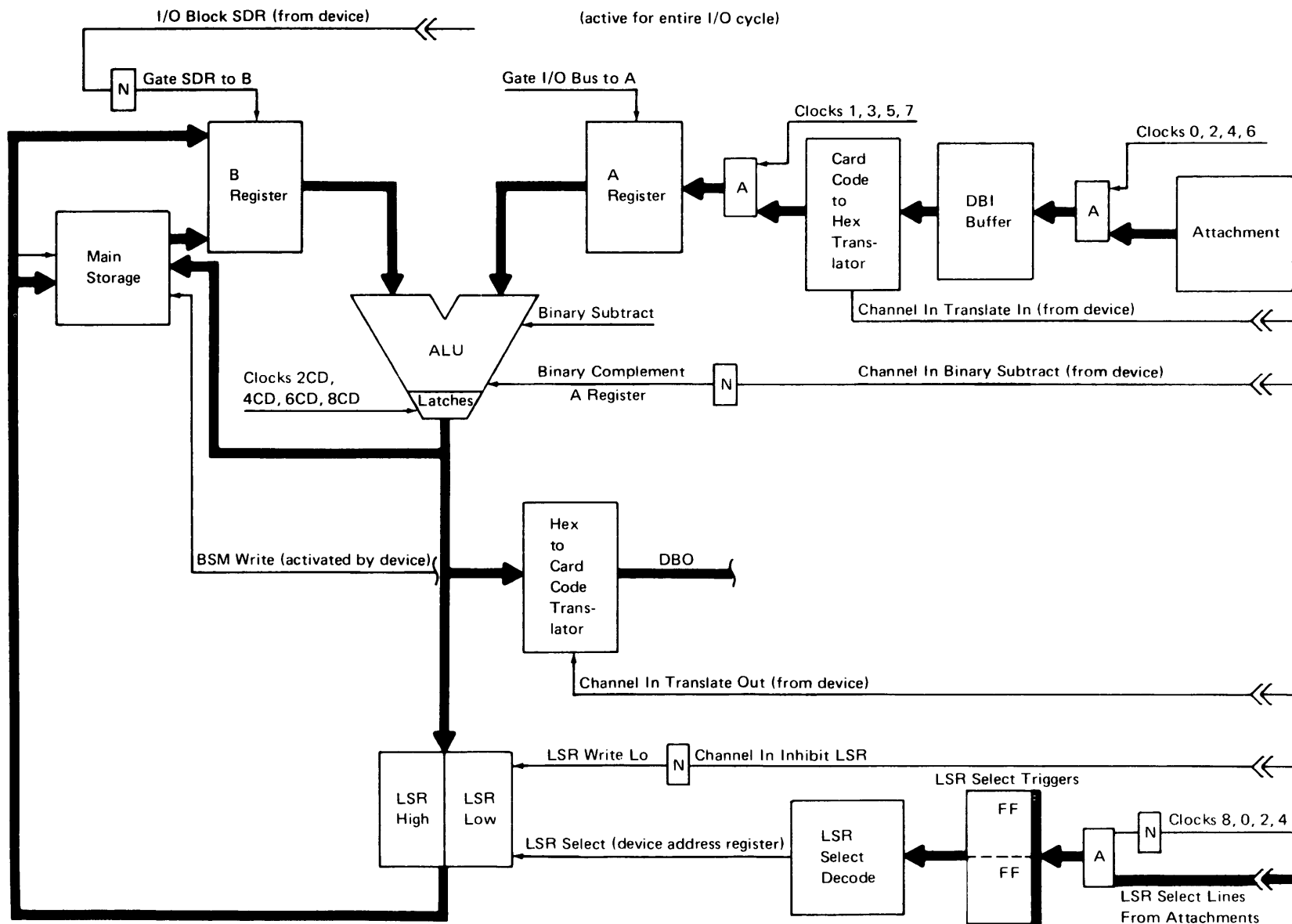


The amount the register is to be increased or decreased is entered into the A register from DBI. Incrementing or decrementing is determined by controlling the 'bin comp A reg' line **A**.

Address modification does not take place in all I/O cycles. For instance, during the third print cycle taken to print a character the printer attachment must address the same storage location as during the first print cycle. In all three print cycles, the printer attachment sends the value 12 to the CPU on DBI. This 12 is added to the LPDAR in all three cycles. However, in the first two cycles, incrementing is blocked by activating the 'chan in inh LSR' line to prevent the results from being written into the LPDAR.



# I/O Cycle Data Transfer



\*DBO during a cycle steal equals ALU Out except during 7D to 0C when the cycle steal assignment is gated on DBO.

**Cycle Steal Priority (CSP)**

- Device requests cycle.
- CPU assigns cycle by device priority.
- I/O cycle can occur between any two CPU cycles.

Whenever an I/O device needs data from storage (write, print, punch) or has data to send to storage (read), the device requests an I/O cycle. An I/O cycle request can occur during any cycle and is always granted by the CPU. More than one I/O device may request an I/O cycle at the same time so each device is assigned a particular cycle steal priority.

Cycle steal requests are generated by the attachments at even clock times. Because of the internal circuit delays, these lines are not sampled until the next clock pulse.

During CPU odd clock times, requests for cycle steal enter the CPU from the attachments on the 'priority request' lines. These requests are entered into the 'priority request' latches and triggers. If more than one device requests an I/O cycle during the same clock time, the bit triggers with the highest priority prevent the lower priority triggers from being turned on. A request at a later clock time resets the triggers and latches for any previous request.

At clock 7D time, the bit structure for the highest priority device among those requesting a cycle is sent to the devices on the DBO, bypassing the DBO translator. 'Any CSP request' blocks the 'machine advance' pulse preventing the CPU from advancing to the next CPU cycle.

Odd parity is maintained for the priority bits on DBO. A P bit is available on DBO when the parity latch is off. Thus, if no request is received, a P bit is available on DBO at clock 7D time. For clocks 1, 3, and 5, each device turns on 2 bit positions to be sent on DBO. The parity latch is off and provides the needed P bit. During clock 7, only one bit position is turned on by the requesting device, and requires the P bit latch be turned on to eliminate unneeded parity bit.

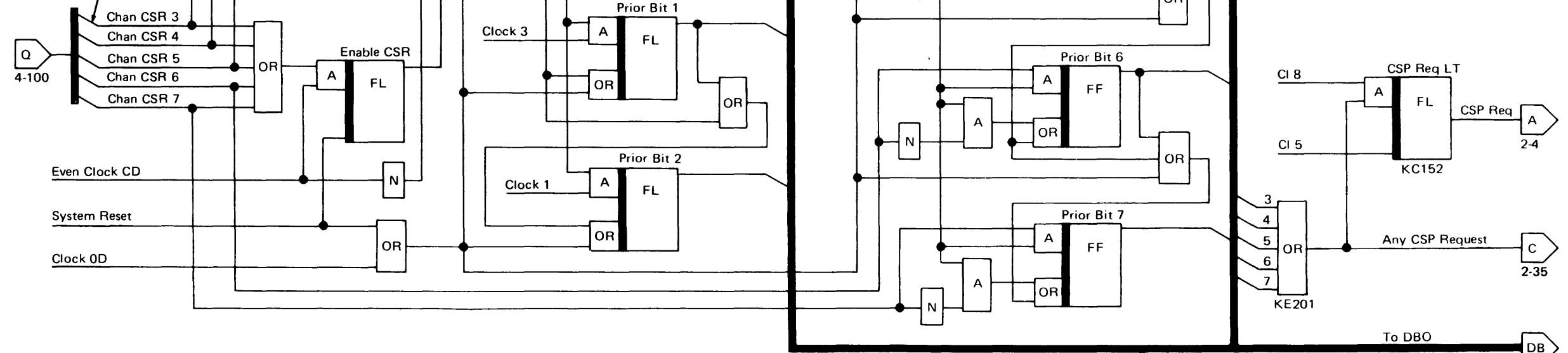
The following priority is assigned to the devices using cycle steal:

Priority Request	Cycle Steal Request Line	CPU Clock	Attachment	DBO Lines
1	—	—	Spare	—
2	—	—	Spare	—
3	5	7	3410 Tape Unit	0 0000 0100
4	6	7	SIOC	0 0000 0010
5	—	—	Spare	—
6	8	5	BSCA-1	1 1001 0000
7	4	5	5424 Read/Punch	1 1000 1000
7	4	5	2560 Read/Punch/Print	1 1000 1000
8	5	5	2501 Card Reader	1 1000 0100
9	6	5	MLTA/BSCC	1 1000 0010
10	7	5	BSCA-2/Display Adapter	1 1000 0001
11	3	3	1442 Card Read Punch	1 0101 0000
12	4	3	5424 Print	1 0100 1000
13-14	—	—	Spare	—
15	7	3	3340/3344 Disk Storage Facility	1 0100 0001
16	3	1	3741 Data Station	1 0011 0000
17	4	1	1403 Printer	1 0010 1000
18	5	1	3277 Display Station/Keyboard	1 0010 0100
19	6	1	Spare	—
20	—	—	Spare	—

This priority assignment makes it possible for the interface to operate on a time sharing basis without the need for I/O buffers. Once a cycle steal request has been granted, the attachment has complete control of the CPU for that cycle.

Page 4-100 shows the I/O interface lines.

*Note:* Because of internal circuit delays, these lines are generated during even clock times.





### Load I/O-LIO (Part 1 of 3)

Bits      0 1 2 3 4 5 6 7  
Op Code  x x 1 1 0 0 0 1

- Moves two bytes from storage to register selected by I/O attachment.
- Follows command format if device is busy or needs attention.
- A Q code of 0/0 results in a no op condition.
- Provides op-end interrupt control on some devices.

The load I/O instruction is a single address instruction that cannot be executed if the addressed I/O device is busy and needs attention. If the instruction cannot be executed because the addressed device is busy and needs attention it follows a command format and loops on the instruction in the same manner as an SIO instruction.

When it can be executed, the load I/O instruction removes two bytes from storage and loads them into a register selected by the device attachment. **A**

The register may be located in the attachment or may be an LSR in the CPU. In either case, two B-cycles are required to remove the bytes from storage.

The device address and M fields are located in bits 0-4 of the Q byte just as in any I/O instruction. The N field (bits 5-7) contains the register to be selected by the device attachment or op-end interrupt control for some I/O devices. Refer to the manual for the individual attachments for a description of the N field codes.

During the I-Q cycles, device selection and 'I/O condition A' and 'I/O condition B' line control shown in the following chart:

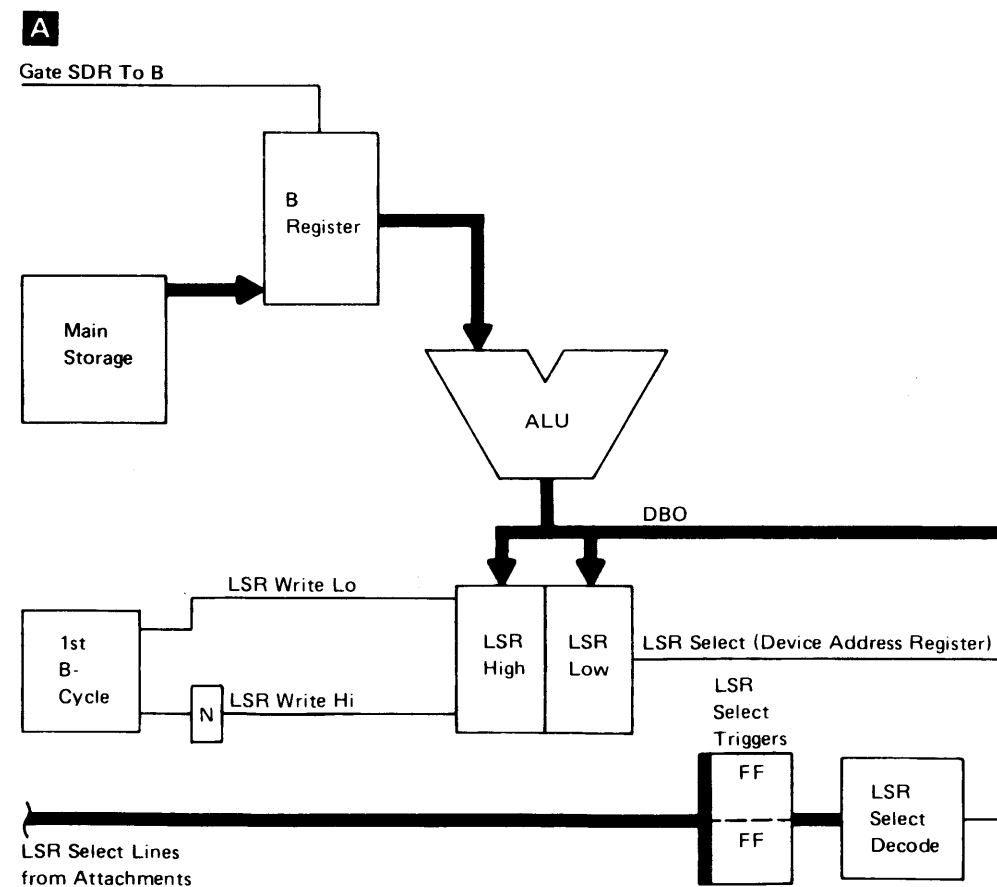
Line Activated By Any Device	Significance
'I/O condition B' only	Correct address, valid N code, device not busy and does not need attention—instruction accepted.
'I/O condition A' only	Correct address, valid N code, device busy or needs attention—instruction rejected.
Both lines	Incorrect parity—causes processor check and DBO parity check.
Neither line	Invalid address or N code—causes processor check and invalid device address.

If program interlock is activated, an I-R cycle is forced. During the I-R cycle, because program interlock is active, I-R program back-up is activated to loop the instruction in the same manner as in an SIO instruction.

If the device attachment can execute the instruction, the B field address is loaded into the BAR with either I-H1 and I-L1 cycles or an I-X1 cycle. After the I-cycles, the CPU enters into the first of two B-cycles.

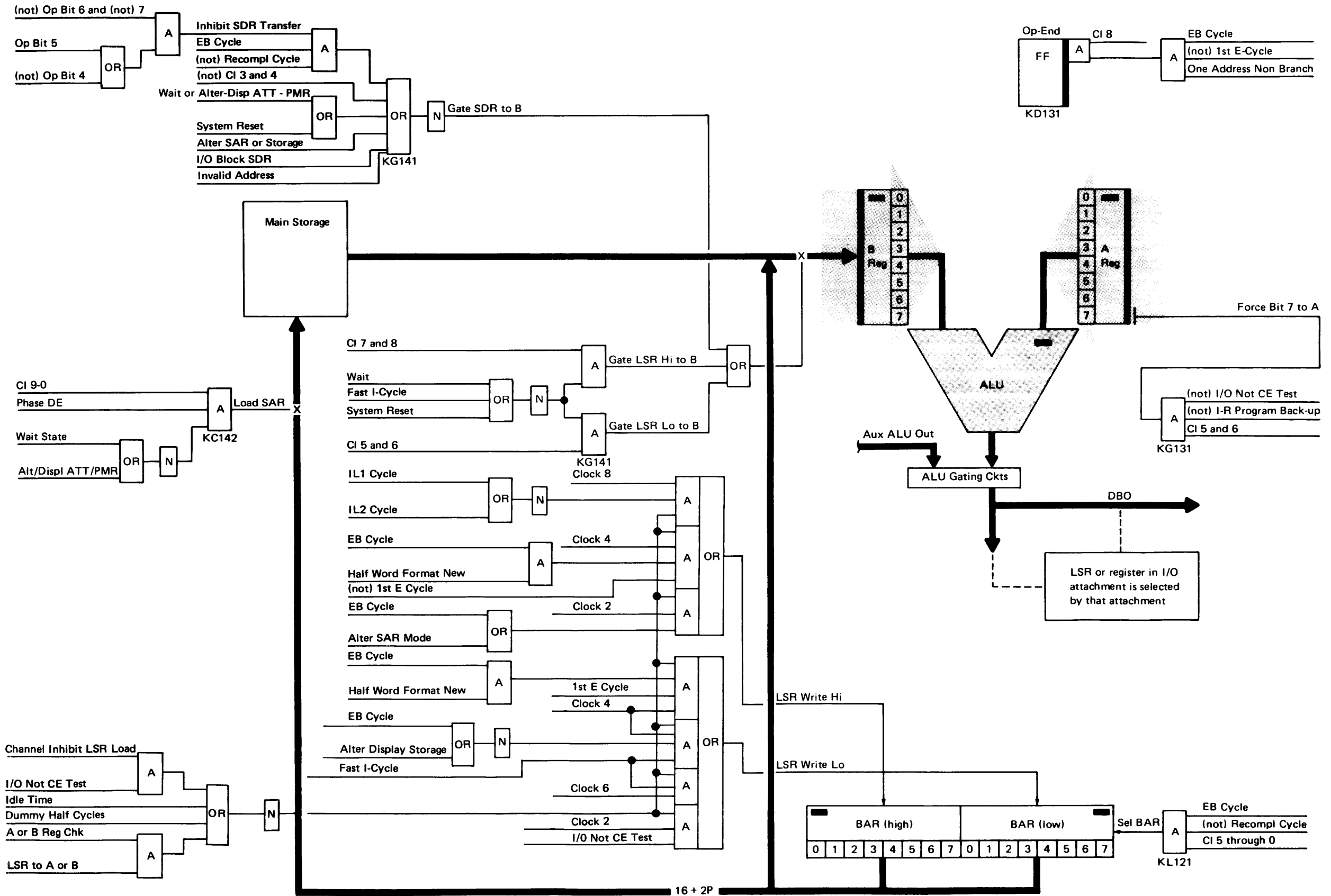
During the first B-cycle, the first byte is loaded into the B register and is passed through the ALU with no ALU controls. If the device attachment selects an LSR, the ALU output is written into the low order position of the selected LSR. If no LSR is selected, the ALU output is available on DBO to be entered into a register selected in the device attachment.

The BAR is decremented during clocks 5 to 8 times and in the second B-cycle, the data byte is entered into the high order position of the selected register. The op-end trigger is turned on and the operation ends.





Load I/O (Part 3 of 3)



**Sense I/O—SNS (Part 1 of 3)**

Bits     0 1 2 3 4 5 6 7  
Op Code x x 1 1 0 0 0 0

- Moves two bytes from the register selected by the I/O attachment to storage.
- Instruction executed even if device is busy or needs attention.
- Q code of hex 0/0 senses console address/data switches.

The sense I/O instruction moves two bytes of data from a register selected by the device attachment to main storage. The register may be located in the attachment or may be an LSR in the CPU. The instruction is executed regardless of whether the device is busy or not.

The device address and M fields are located in bits 0-4 of the Q byte just as in any I/O instruction. The N field (bits 5-7) contains the code for the register to be selected by the device attachment. Refer to the manual for the individual attachments for the register selection codes and the information contained in those registers.

If the Q code contains a hex 0/0, the two bytes of data set up in the console address/data switches are moved to the address specified in the B field address and that address minus one. The first B-cycle moves the data set up in the two right-most switches, and the second B-cycle moves the data set up in the two left-most switches.

During the I-Q cycle, device selection is the same as in a start I/O instruction except that the device attachment activates the 'I/O condition B' line whether it is busy or not. Valid address checking and valid N code checking remain the same.

Line Activated By Any Device	Significance
'I/O condition B' only	Correct address, valid N code, device not busy and does not need attention—instruction accepted.
'I/O condition A' only	Correct address, valid N code, device busy or needs attention—instruction rejected.
Both lines	Incorrect parity—causes processor check and DBO parity check.
Neither line	Invalid address or N code—causes processor check and invalid device address.

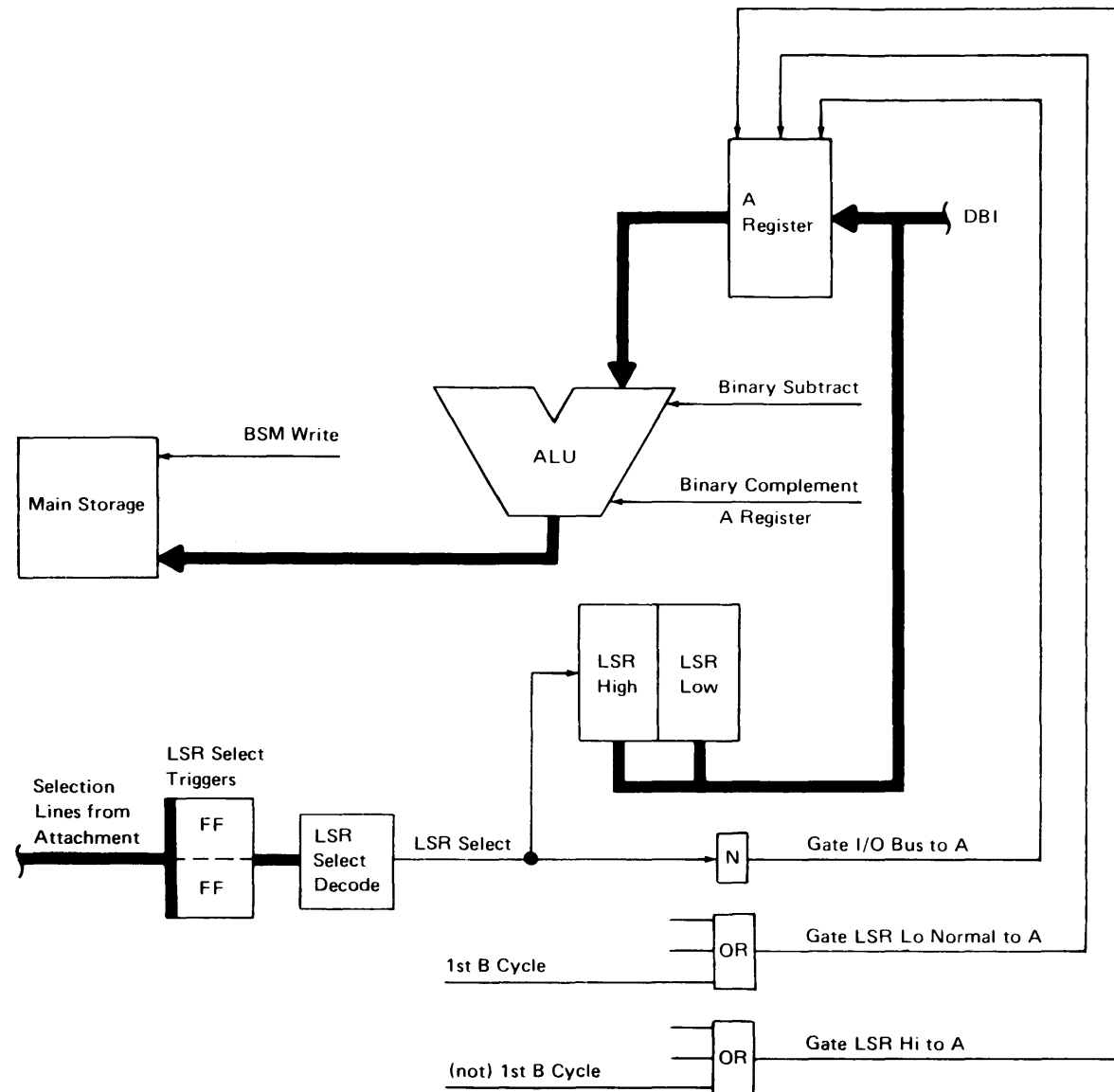
The B field address is loaded into the BAR in the same manner as in all one-address instructions. After the I-cycles, the CPU takes two B-cycles to store the registers in the B field storage location.

If the register selected by the attachment is an LSR, the CPU activates 'gate LSR low normal to A' during clock 3 and 4 of the first B-cycle.

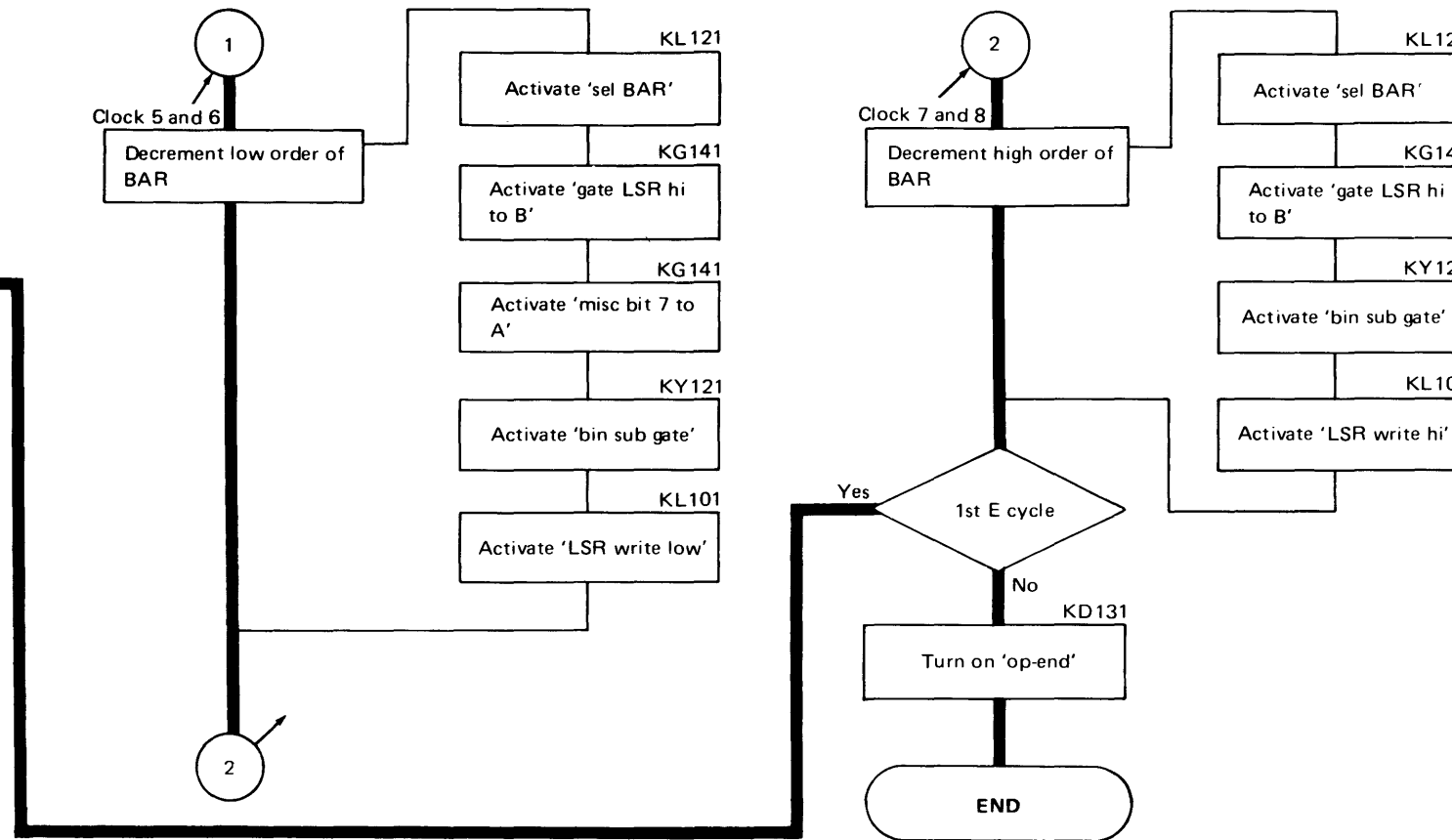
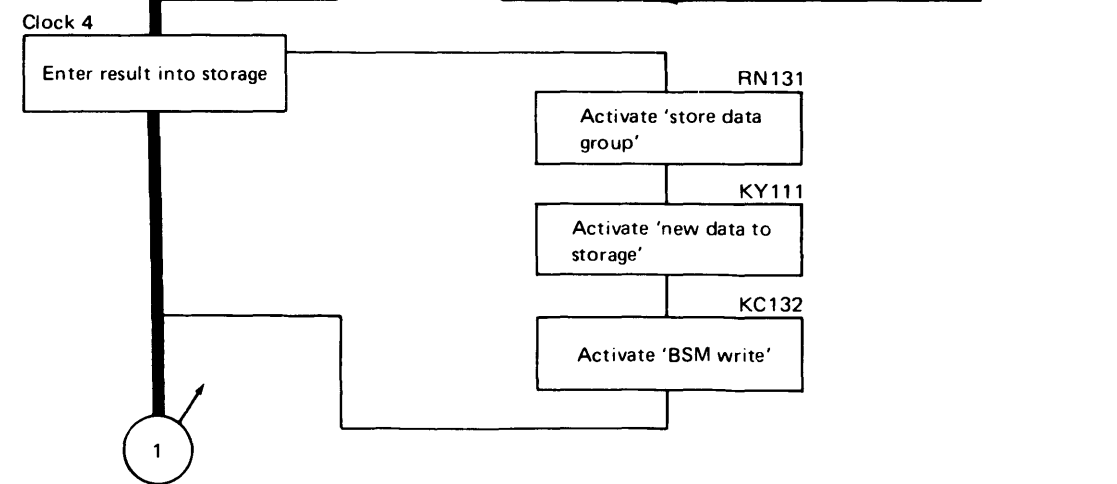
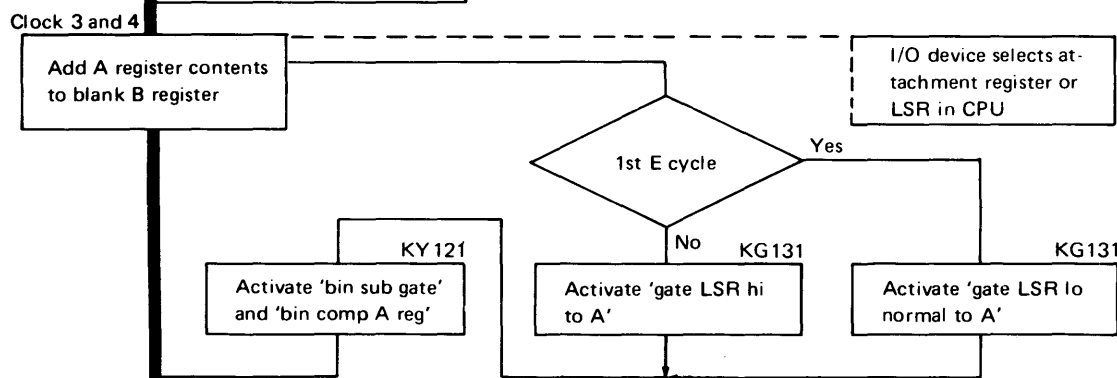
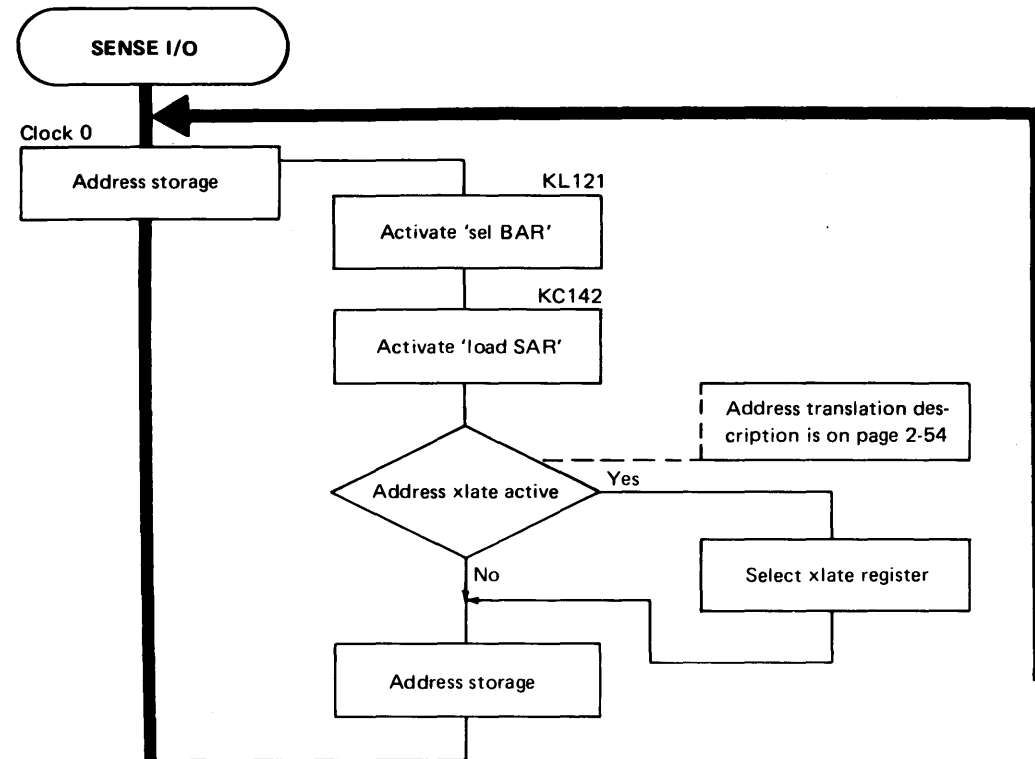
The low order of the selected LSR is then loaded into the A register where it is binary added to the B register and written into storage. Since the B register contains all 0's, the result is the same as the LSR low order byte.

The BAR is decremented and the LSR high order byte is transferred in the second B cycle. The 'op-end' trigger is turned on and the operation ends.

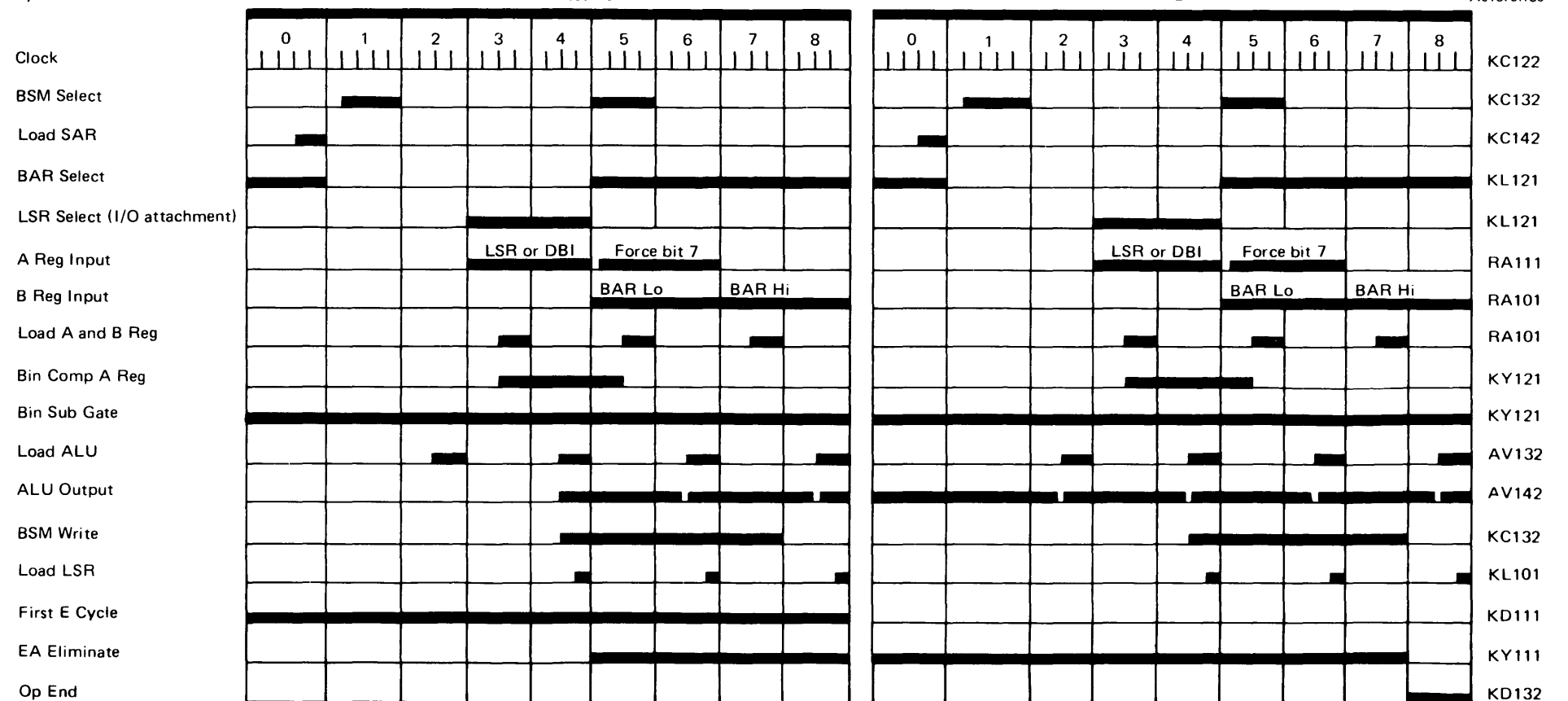
If the register selected by the attachment is not an LSR, gate I/O bus to A is activated in both B-cycles and the bytes enter the A register from DBI.



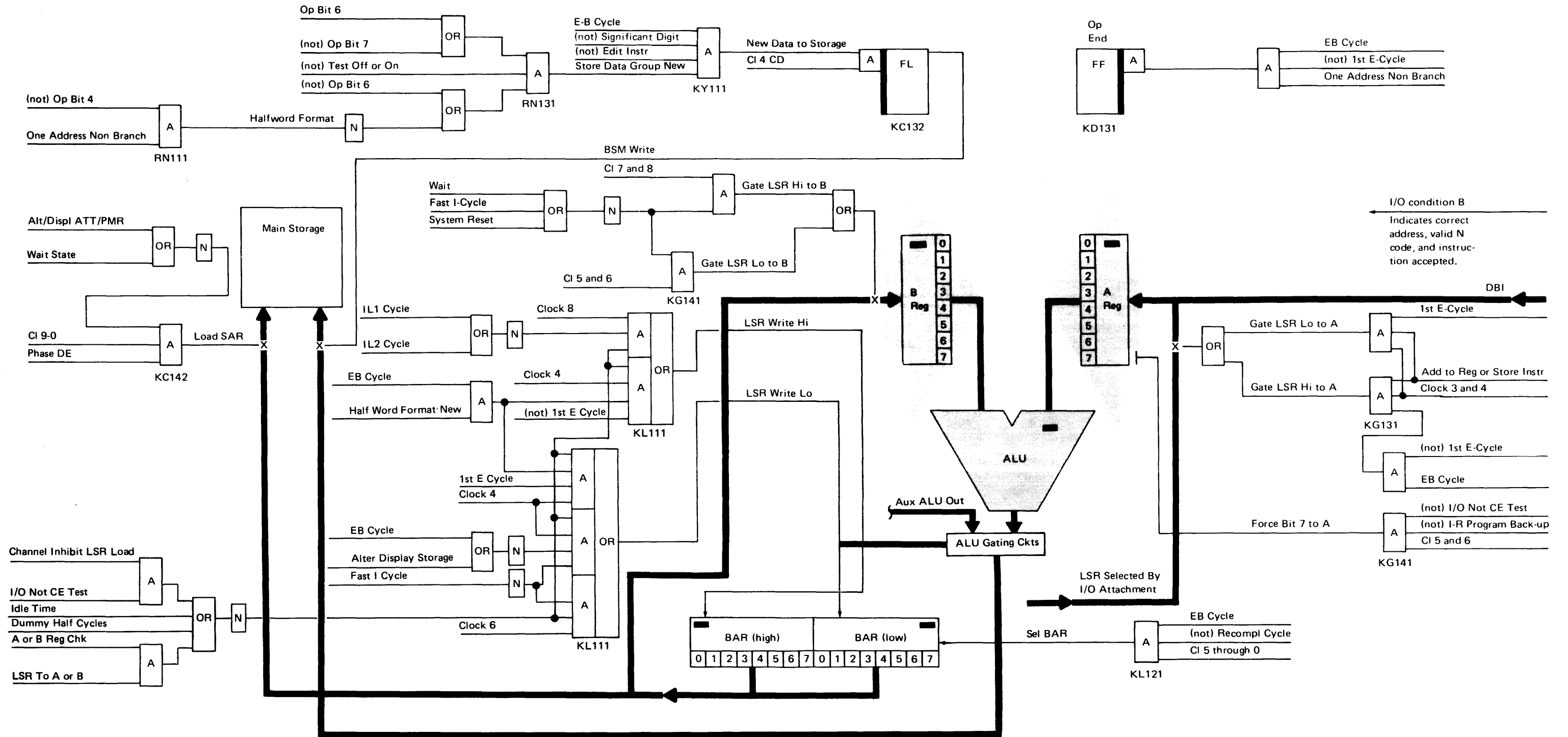
Sense I/O (Part 2 of 3)



Cycle



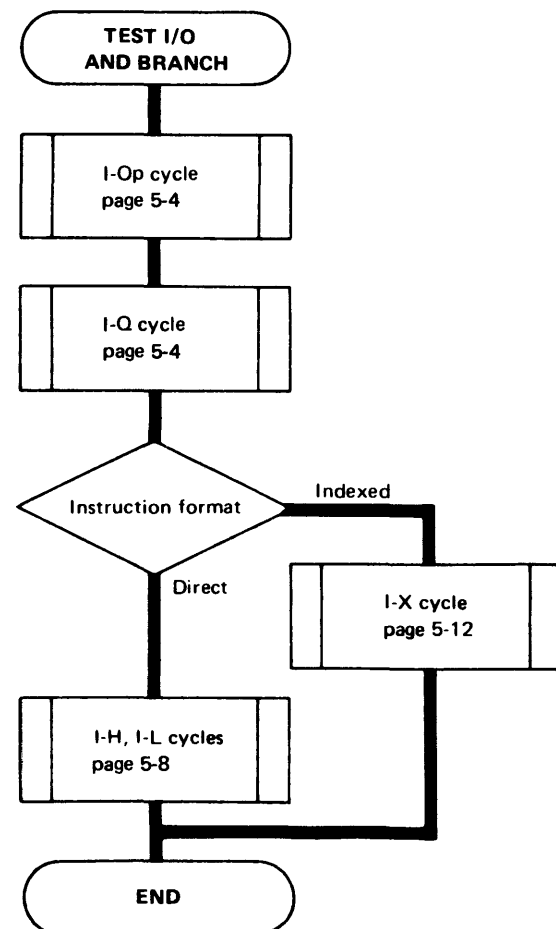
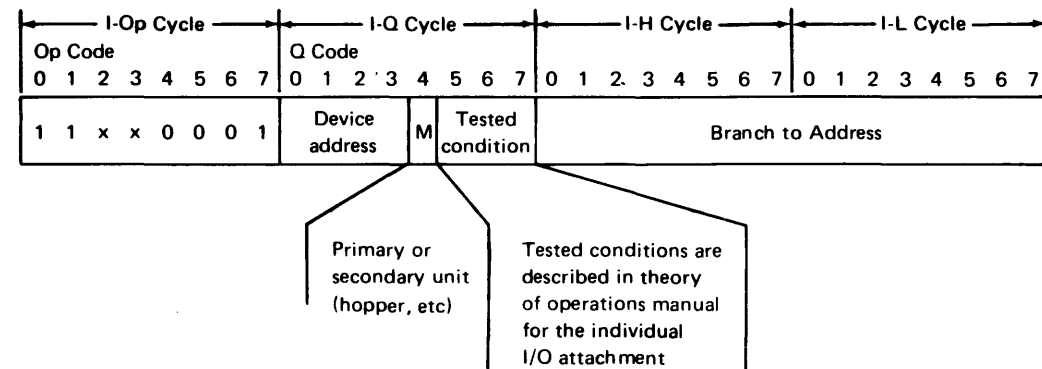
Sense I/O (Part 3 of 3)



### Test I/O and Branch—TIO

Bits 0 1 2 3 4 5 6 7  
 Op Code 1 1 x x 0 0 0 1

- Test for I/O condition specified in Q code N field.
- Branch to address is loaded into ARR.
- Use branch-to address for next instruction if tested condition exists. (IAR/ARR interchange occurs.)



The test I/O and branch instruction is a one address instruction that tests an I/O device for a specified condition and branches if that condition exists. The instruction is the same as a normal branch except that the I/O device is tested instead of the CR.

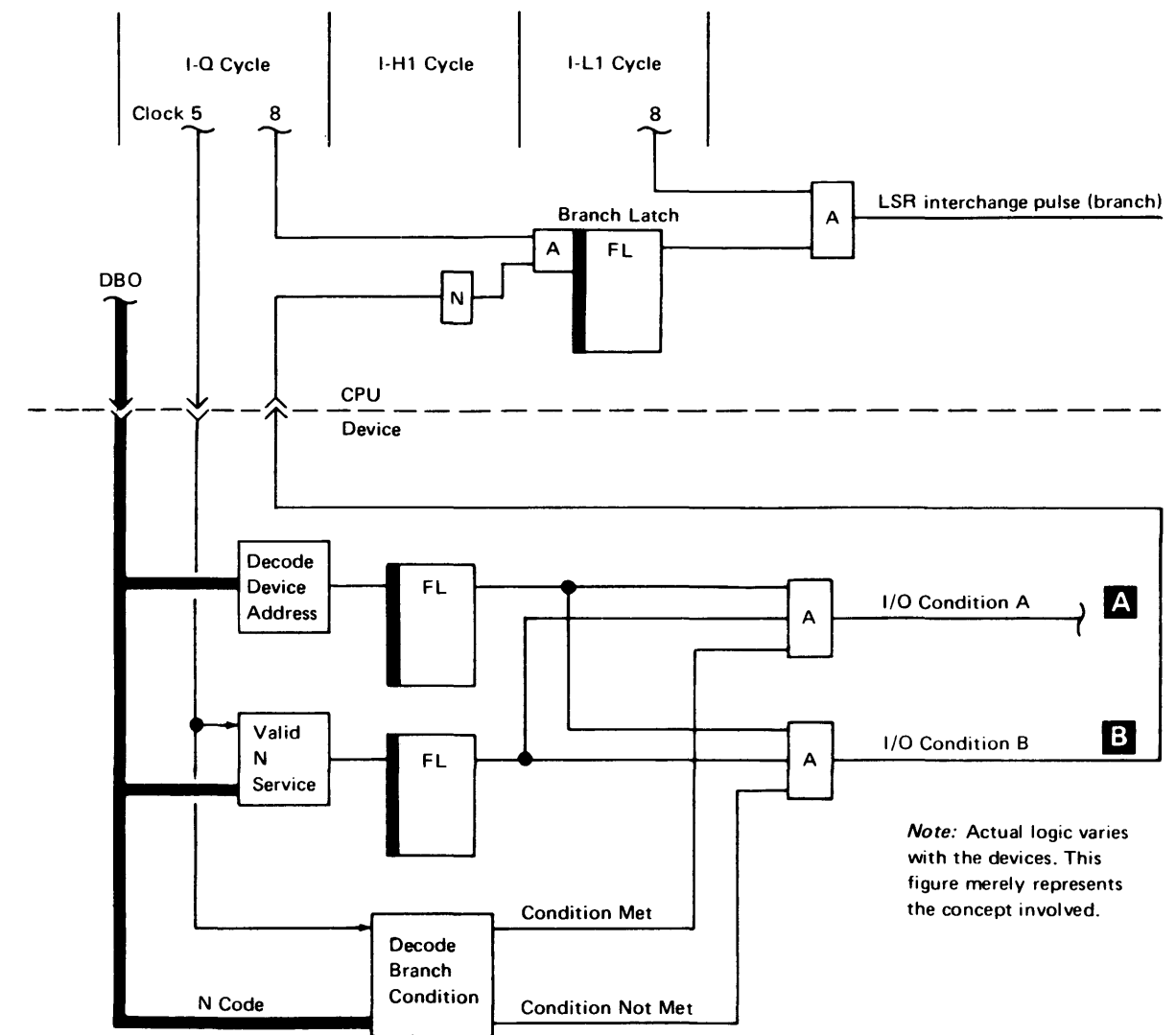
The device address and M fields are located in bits 0-4 of the Q byte just as in any I/O instruction. The N field (bits 5-7) contains the code for the condition being tested. The conditions vary with each attachment. Refer to the manual for the individual attachments for the condition codes.

Branching is determined by the device control of the 'I/O condition A' and 'I/O condition B' lines. The following chart shows the significance of the line settings.

Line Activated By Any Device	Significance
'I/O condition B' only <b>B</b>	Correct address, valid N code, condition for branching not met—proceed with next sequential instruction.
'I/O condition A' only <b>A</b>	Correct address, valid N code, condition for branching met—branch to new address.
Both lines	Incorrect parity—causes processor check and DBO parity check.
Neither line	Invalid address or N code—causes processor check and invalid device address.

During the I/O cycle, device selection is the same as in other I/O operations. The device attachment also tests for the condition specified in the N field. If the condition is met, the attachment activates the 'I/O condition A' line. With the 'I/O condition B' line inactive at clock 8 time, the branch condition is latched in the CPU.

The ARR is loaded with I-H1 and I-L1 cycles or an I-X1 cycle the same as in a normal branch operation. The 'LSR intchg pulse' is then activated to switch the IAR and ARR. Refer to *Branch to Condition*, page 5-51, for the IAR/ARR interchange explanation.



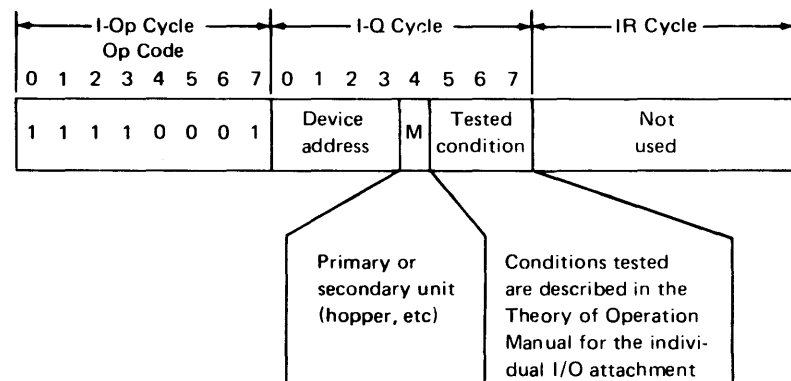
Note: Actual logic varies with the devices. This figure merely represents the concept involved.

**Advance Program Level—APL**

Bits 0 1 2 3 4 5 6 7  
Op Code 1 1 1 1 0 0 0 1

The 5415 does not have the dual program feature. Therefore, an APL instruction is executed just like the basic 5410. The APL instruction exists in the 5415 to allow Model 10 users to run their program unchanged.

- Test for I/O condition specified in Q code N field.
- Loop on APL instruction until condition tested for no longer exists.
- Q code N field of all zeros causes automatic advance to next sequential instruction.

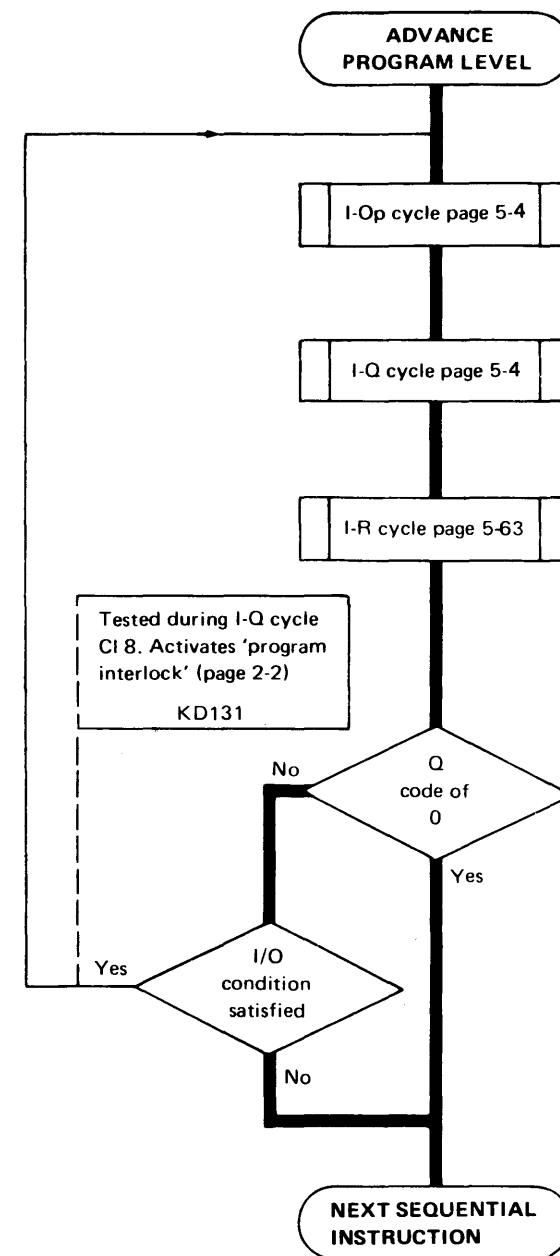


The advance program level instruction tests for the same conditions as a test I/O and branch operation. The N code is the same as in TIO and varies with each attachment. If the advance condition is met, the attachment activates the 'I/O condition A' line just as in the TIO.

With the 'I/O condition B' line inactive, program interlock is activated in the I-Q cycle the same as in a start I/O instruction. During the I-R cycle, with program interlock active, 'IR prog back-up' decrements the IAR by 2 just as in the SIO instruction. When the advance condition is not met, and 'I/O condition B' is activated by the attachment, the IAR is incremented in the normal manner.

An all zero Q code is recognized as an automatic advance instruction. With this condition the 'IR program back-up' line is blocked and the IAR is incremented in the normal manner. Thus, an automatic advance instruction is equivalent to a no-op operation.

Line Activated By Any Device	Significance
'I/O condition B' only	Correct address, valid N code, device not busy and doesn't need attention—instruction accepted.
'I/O condition A' only	Correct address, valid N code, device busy or needs attention—instruction rejected.
Both lines	Incorrect parity—causes processor check and DBO parity check.
Neither line	Invalid address or N code—causes processor check and invalid device address.





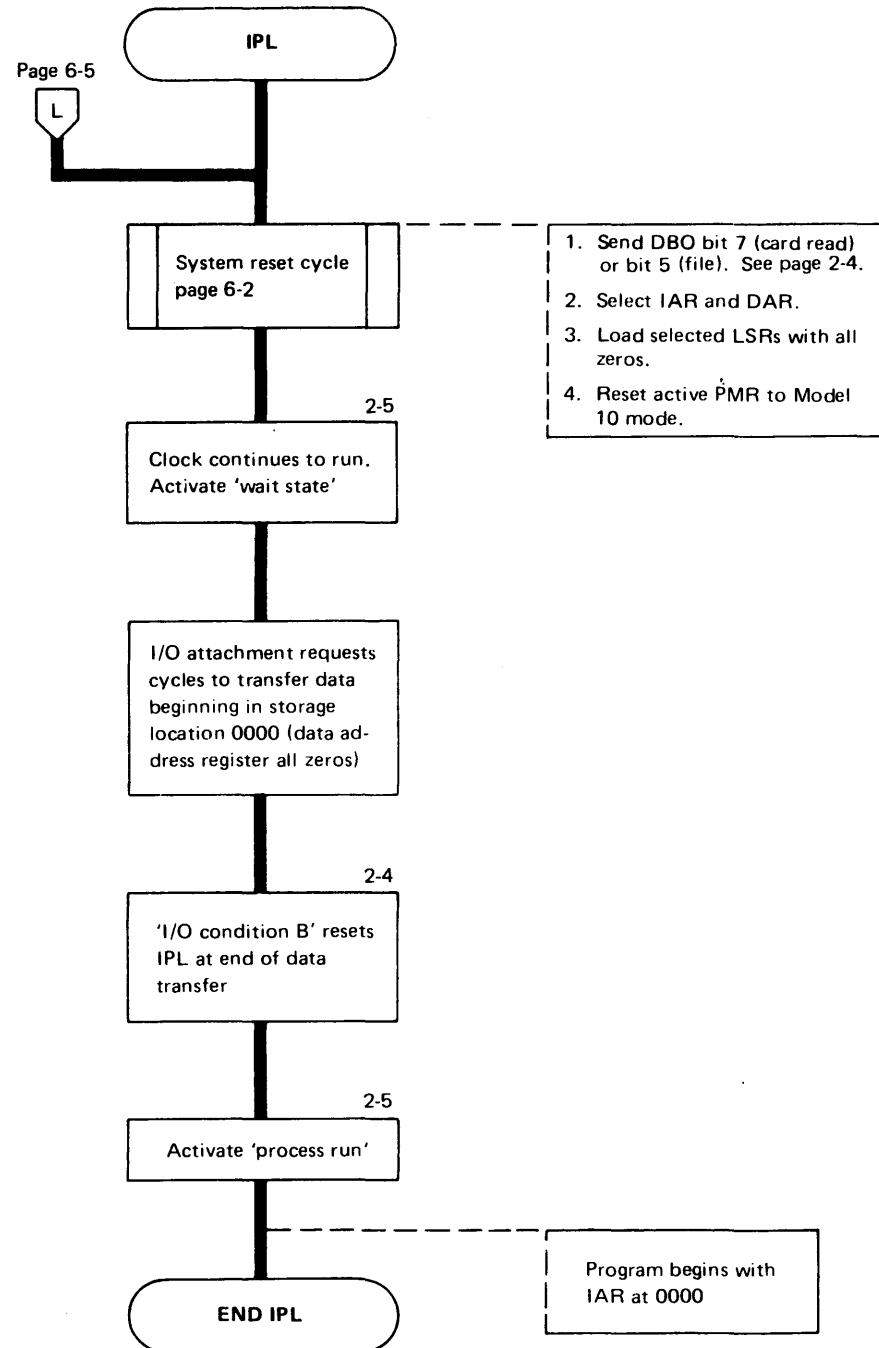
# SYSTEM OPERATIONS

## INITIAL PROGRAM LOAD – IPL

- Program load key initiates a system reset cycle.
- If this is first IPL or system reset since power on, perform an initial main storage scan.
- IPL activates data bus out 7 for the alternate IPL device (5424, 2560, 1442, or 3741).
- IPL activates data bus out 5 for disk read.

The initial program load operation is started by pressing the program load key. Pressing this key initiates a system reset cycle and selects the card device or file by activating a Data Bus Out line. DBO bit 7 selects the card read device while DBO bit 5 selects the disk. The input device selected is determined by the program load selector switch on the operators console (see page 2-4 for circuit).

After the input device is selected, it activates the select line to add all zeros into its DAR and the IAR. The Data Bus Out line causes the device to load one card or block into main storage beginning at address 00 by using the normal cycle steal sequence. After one record or card has been transferred to storage, the IPL line is reset with I/O condition B giving a machine cycle advance. Since the IAR was reset to zero, the first op code is in location 0000.

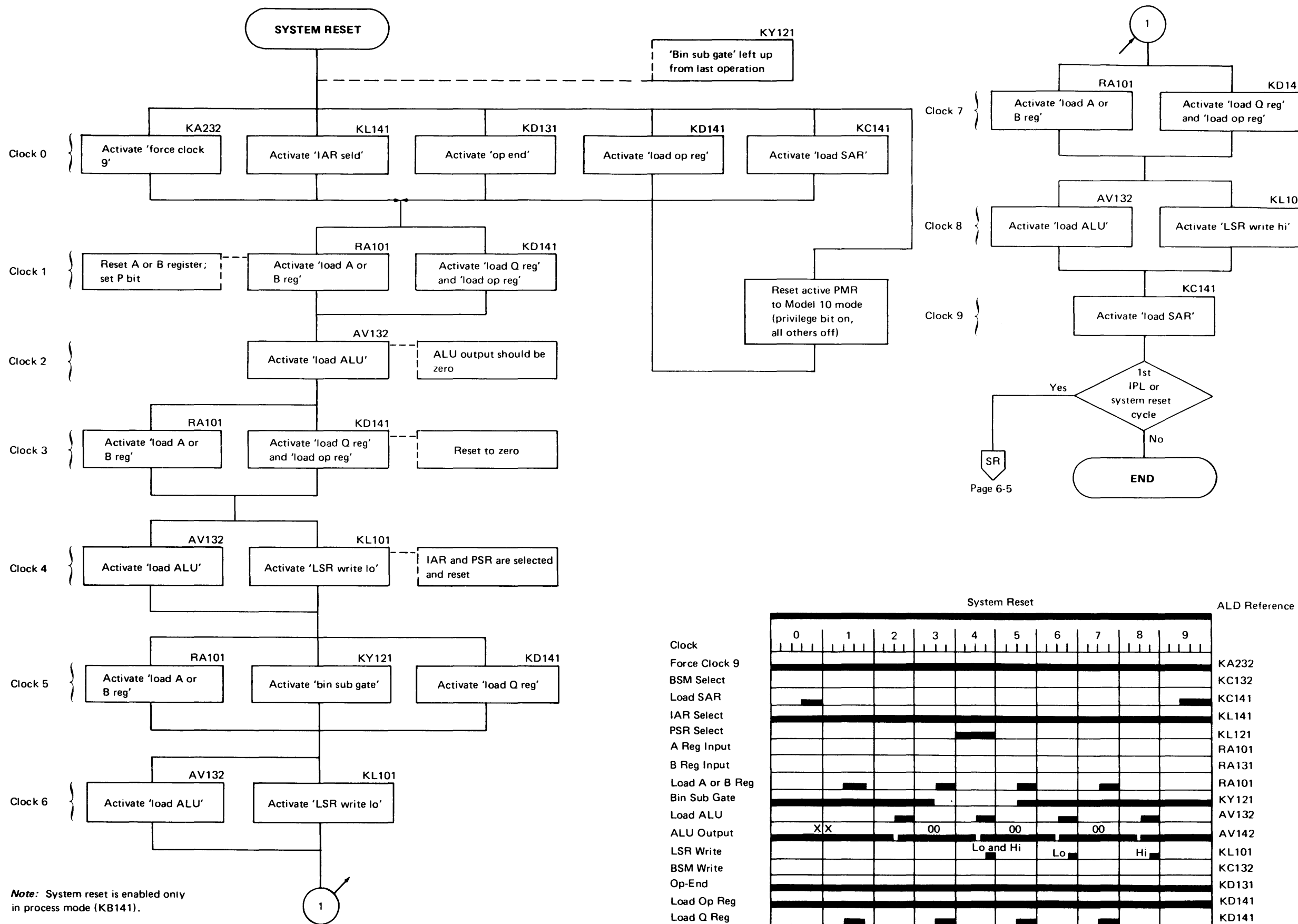


### SYSTEM RESET

#### Objectives

- Enter an immediate idle state.
- CPU registers, controls, and status indicators are reset.
- Perform initial main storage scan if this is the first IPL or system reset since power on.

**Note:** Before attempting a CE operation immediately following power on, press SYSTEM RESET to activate 'initial memory scan'. Main storage check bits are not reset with a power on reset.



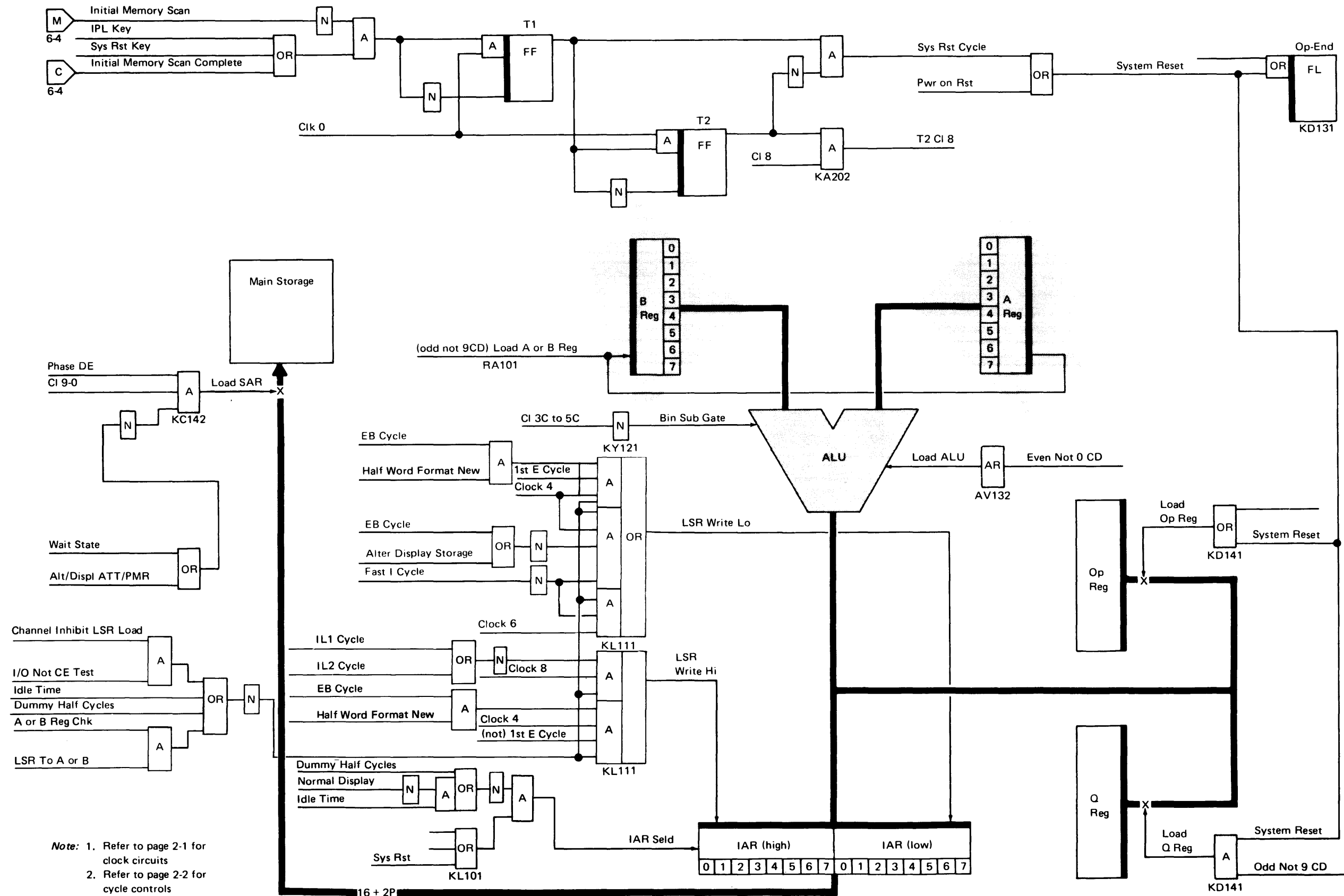
**Note:** System reset is enabled only in process mode (KB141).

Page 6-5

System Reset										ALD Reference	
Clock	0	1	2	3	4	5	6	7	8	9	
Force Clock 9	[Active]										KA232
BSM Select	[Active]										KC132
Load SAR	[Active]										KC141
IAR Select	[Active]										KL141
PSR Select	[Active]										KL121
A Reg Input	[Active]										RA101
B Reg Input	[Active]										RA131
Load A or B Reg	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	RA101
Bin Sub Gate	[Active]										KY121
Load ALU	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	AV132
ALU Output	XX		00		00		00				AV142
LSR Write	[Active]										KL101
BSM Write	[Active]										KC132
Op-End	[Active]										KD131
Load Op Reg	[Active]										KD141
Load Q Reg	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	[Active]	KD141

**Note:** Parity checking is disabled during system reset.

**SYSTEM RESET (Part 2 of 2)**



### ALTER STORAGE/INITIAL MEMORY SCAN (Part 1 of 4)

#### Initial Memory Scan

- Operation is the same as alter storage operation with the STORAGE TEST switch set to RUN.
- Transfer data from the two right-most console ADDRESS/DATA switches (one byte) into Q register and storage location designated by SAR.
- Increment SAR one address each cycle, and repeat data transfer.
- Increment SAR E15, SAR E14, and SAR E13 to alter the remaining BSMs.
- Continue operation until ALU carry occurs eight times.

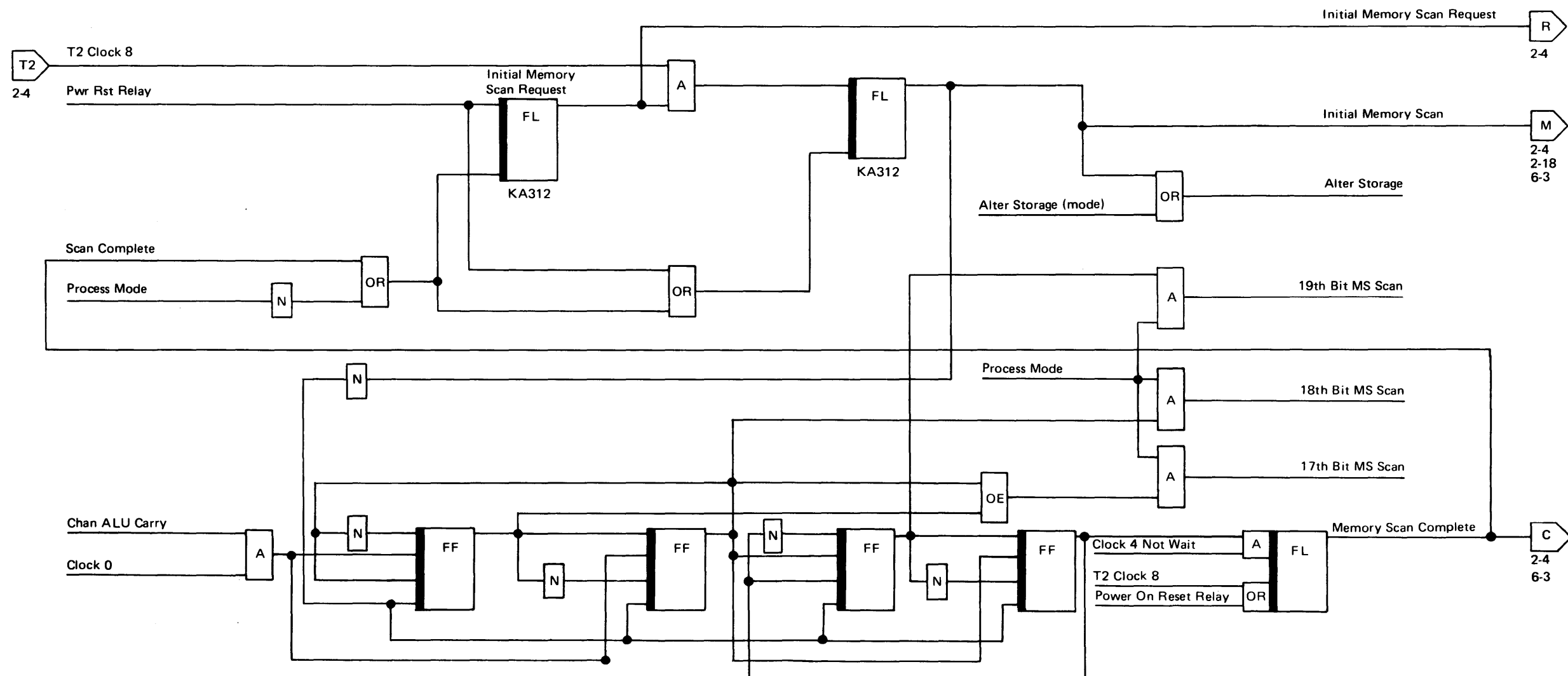
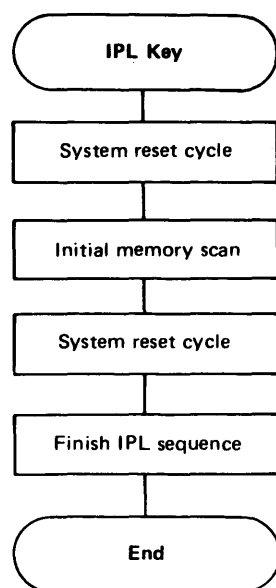
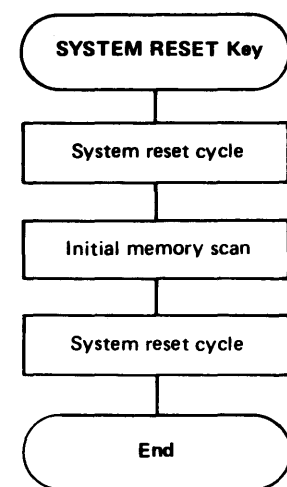
The Model 15 main storage check bits cannot be predicted after power on. Therefore, the contents of storage must be altered to generate check bits that will not cause an error. This is accomplished by storing the contents of the console data switches in every main storage location.

The initial memory scan cycles are taken the first time SYSTEM RESET or IPL is pressed after a power on.

'Initial memory scan request' is activated with the POR relay. 'T2 cl 8' of the first system reset activates 'initial memory scan'. 'Initial memory scan' then starts an alter storage operation. The alter storage operation is the same except the STORAGE TEST switch and START key functions are bypassed.

The contents of the console switches are stored sequentially in every storage address until an ALU carry occurs eight times (IAR update resulted in an overflow when the address for 64K was passed). The IAR overflow occurs eight times to make sure all main storage positions are in correct parity.

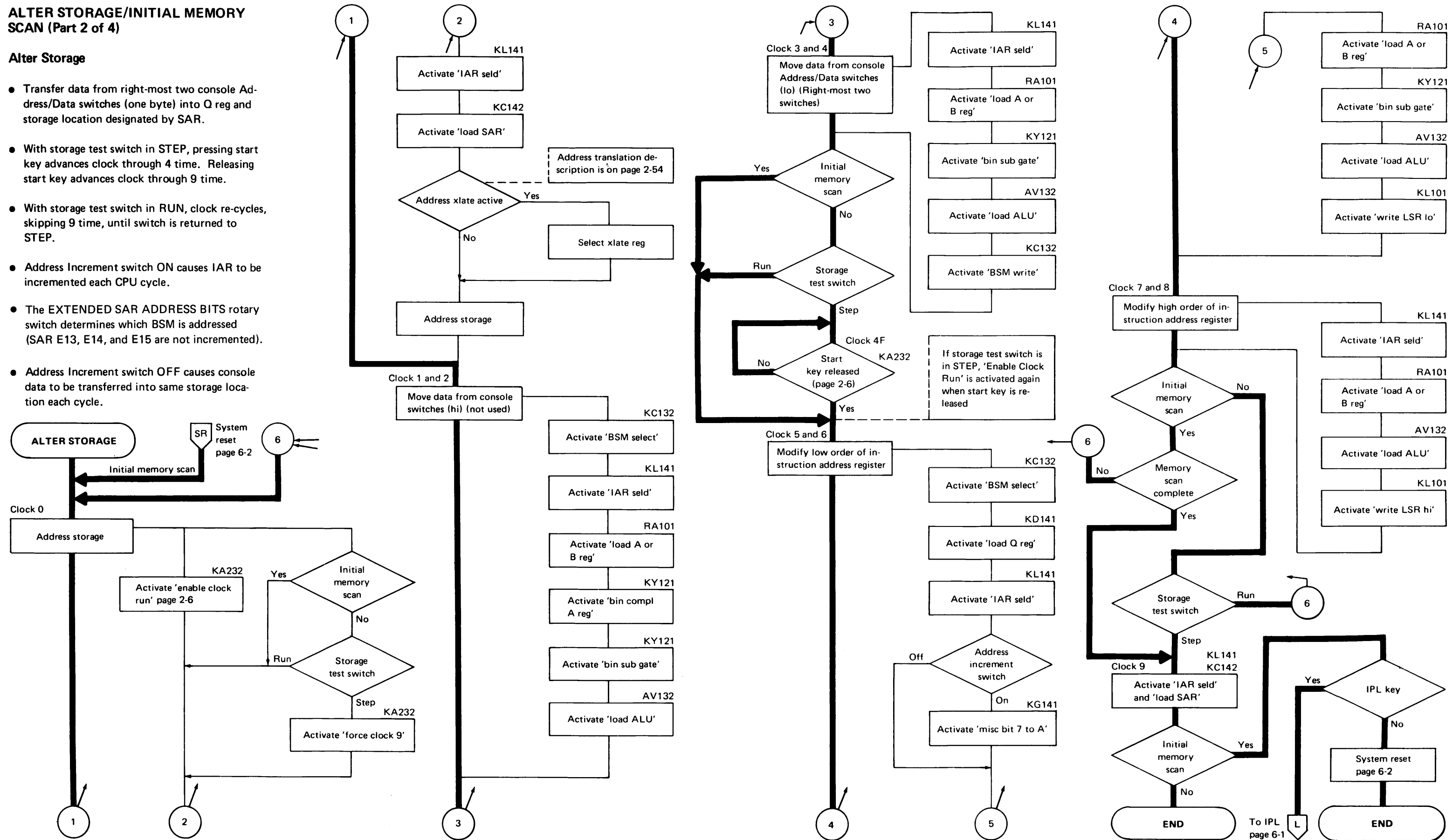
Main storage size is not considered because an invalid address check is prevented by inhibiting 'load check latches' and 'reset check latches' (see page 2-18 or ALD KB141).



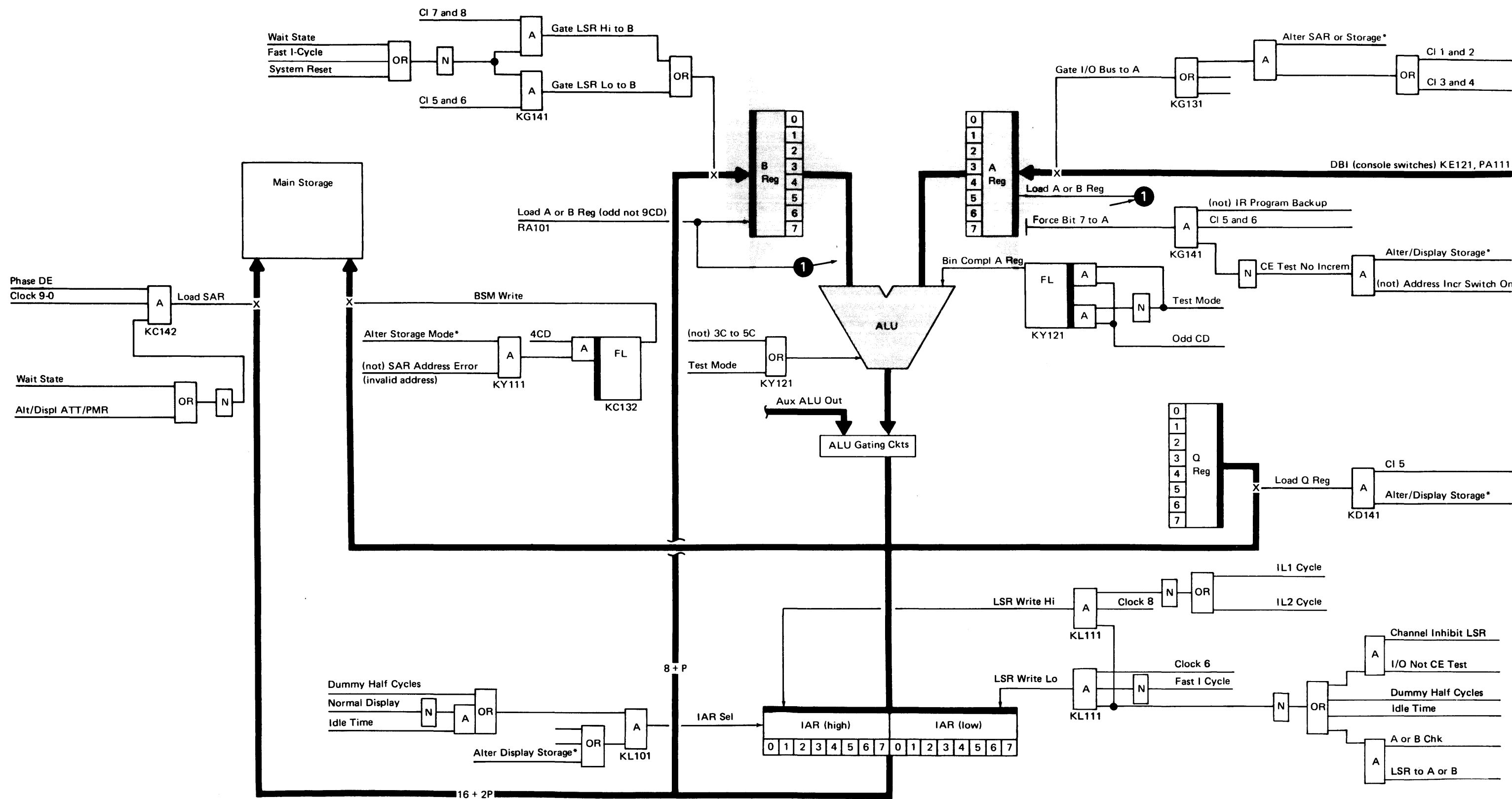
# ALTER STORAGE/INITIAL MEMORY SCAN (Part 2 of 4)

## Alter Storage

- Transfer data from right-most two console Address/Data switches (one byte) into Q reg and storage location designated by SAR.
- With storage test switch in STEP, pressing start key advances clock through 4 time. Releasing start key advances clock through 9 time.
- With storage test switch in RUN, clock re-cycles, skipping 9 time, until switch is returned to STEP.
- Address Increment switch ON causes IAR to be incremented each CPU cycle.
- The EXTENDED SAR ADDRESS BITS rotary switch determines which BSM is addressed (SAR E13, E14, and E15 are not incremented).
- Address Increment switch OFF causes console data to be transferred into same storage location each cycle.

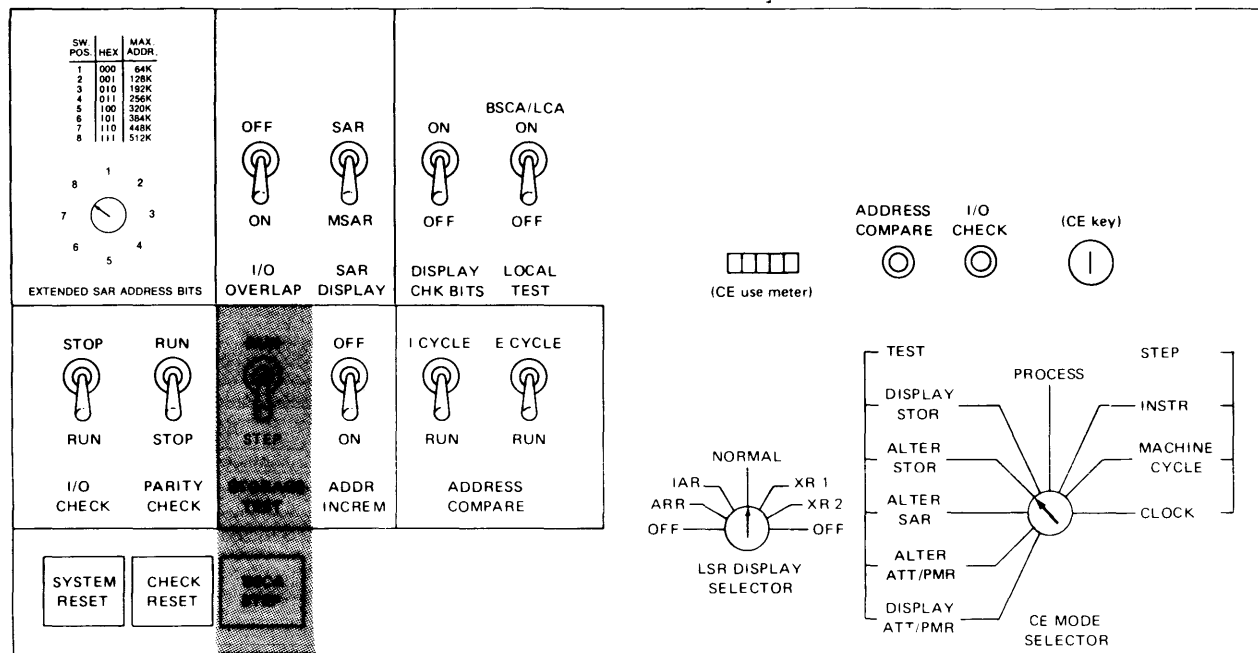


**ALTER STORAGE/INITIAL MEMORY SCAN (Part 3 of 4)**



\* Also active during initial memory scan.

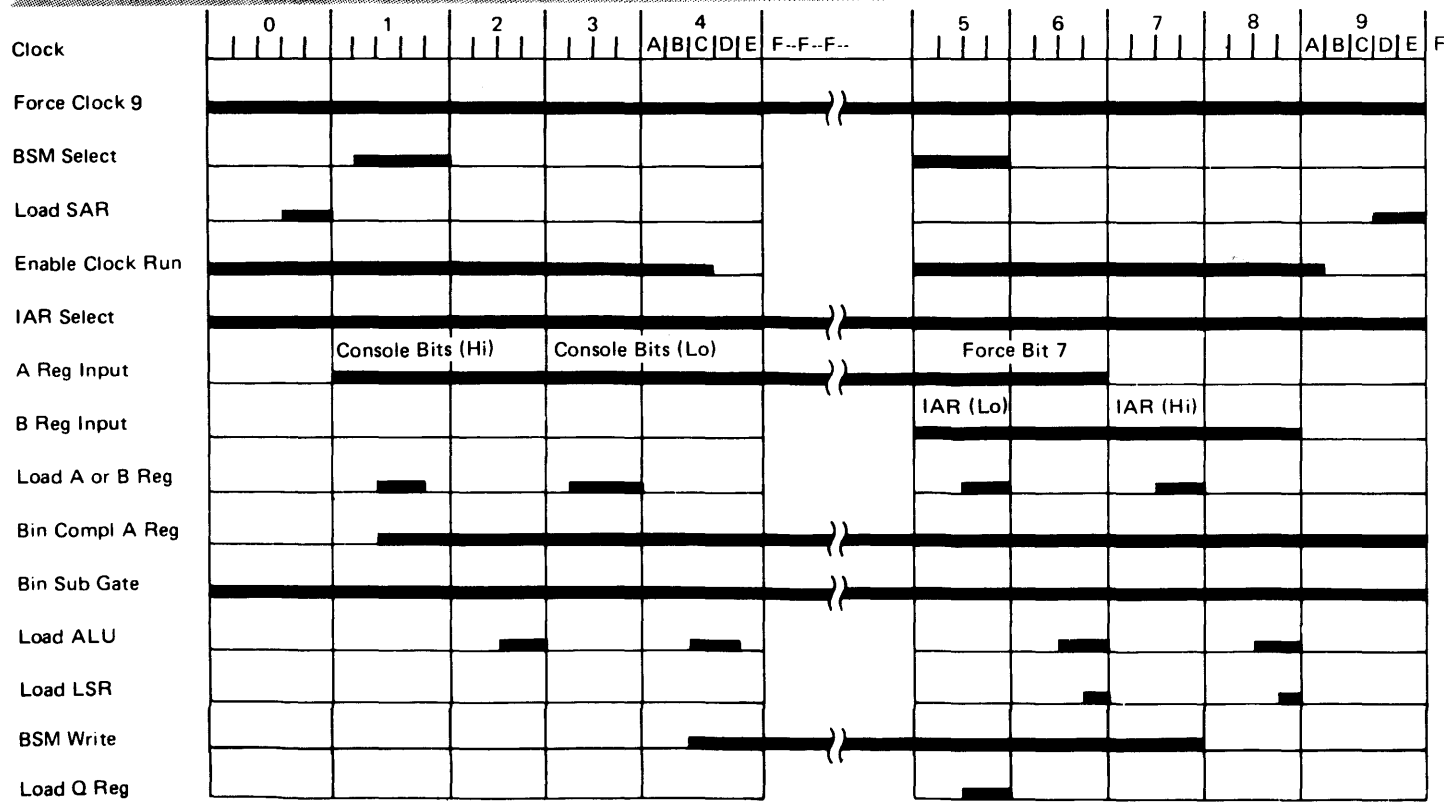
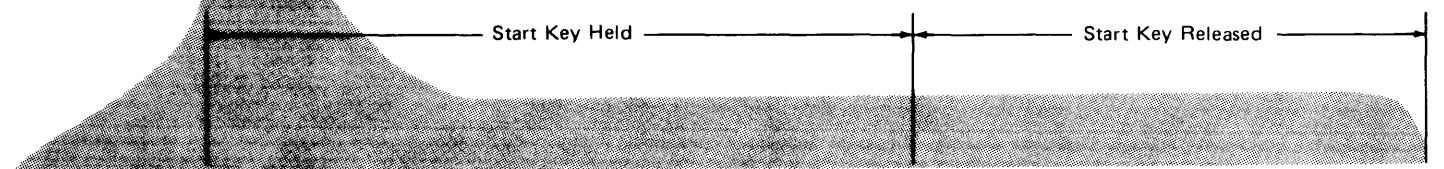
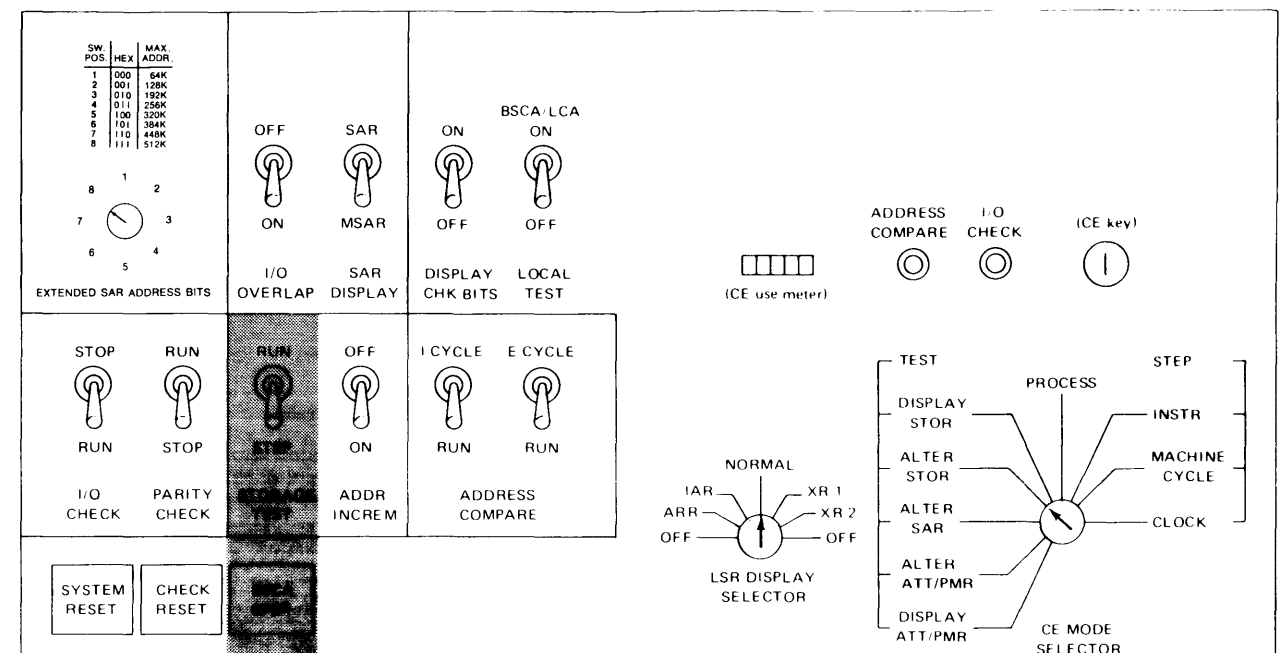
# ALTER STORAGE/INITIAL MEMORY SCAN (Part 4 of 4)



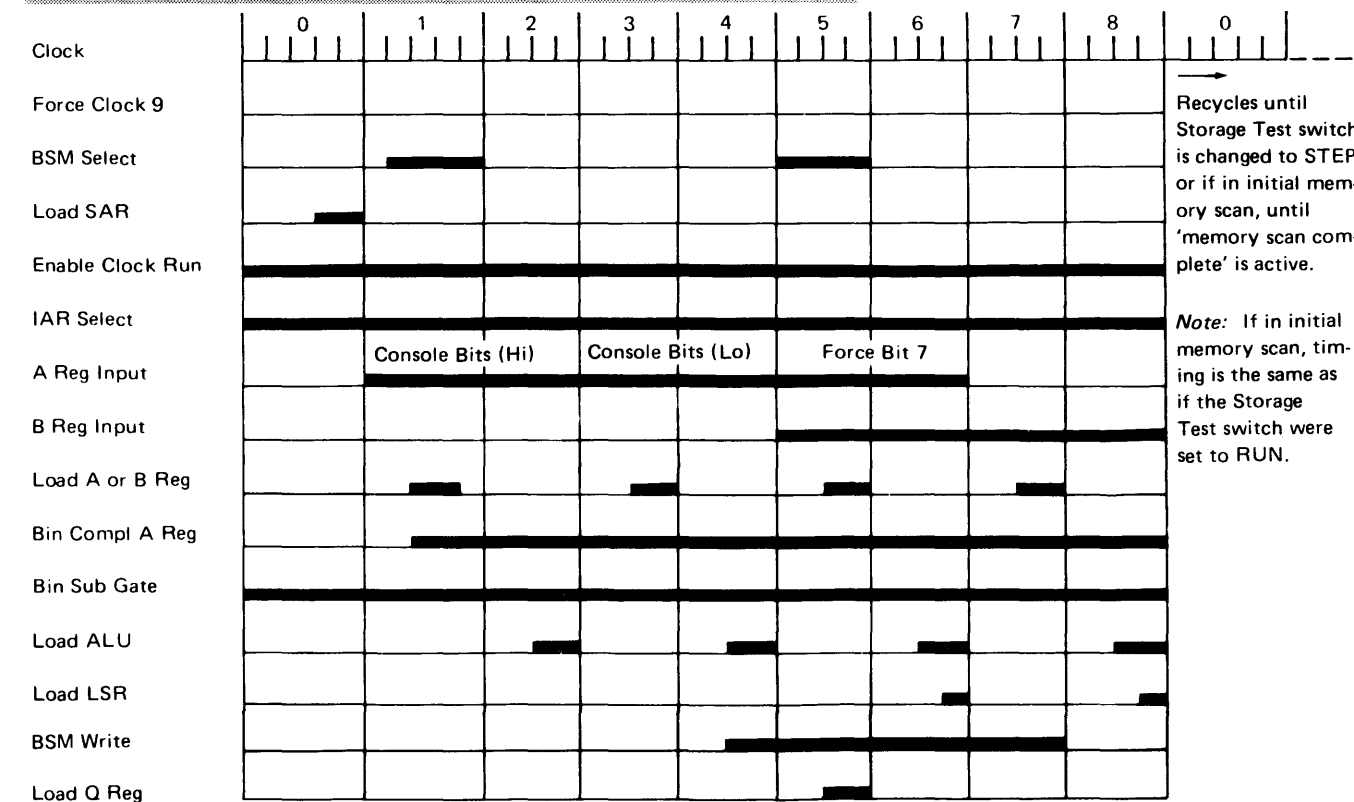
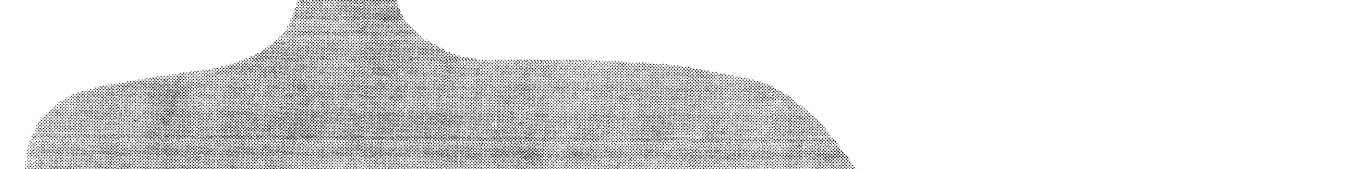
**ADDR INCREM Switch:**

**ON** - IAR incremented by one each CPU cycle.

**OFF** - IAR addresses same storage position each CPU cycle.



- ALD Reference**
- KC122
  - KA232
  - KC132
  - KC142
  - KA232
  - KL141
  - RA111
  - RA101
  - RA101
  - KY121
  - KY121
  - AV132
  - KL101
  - KC132
  - KD141



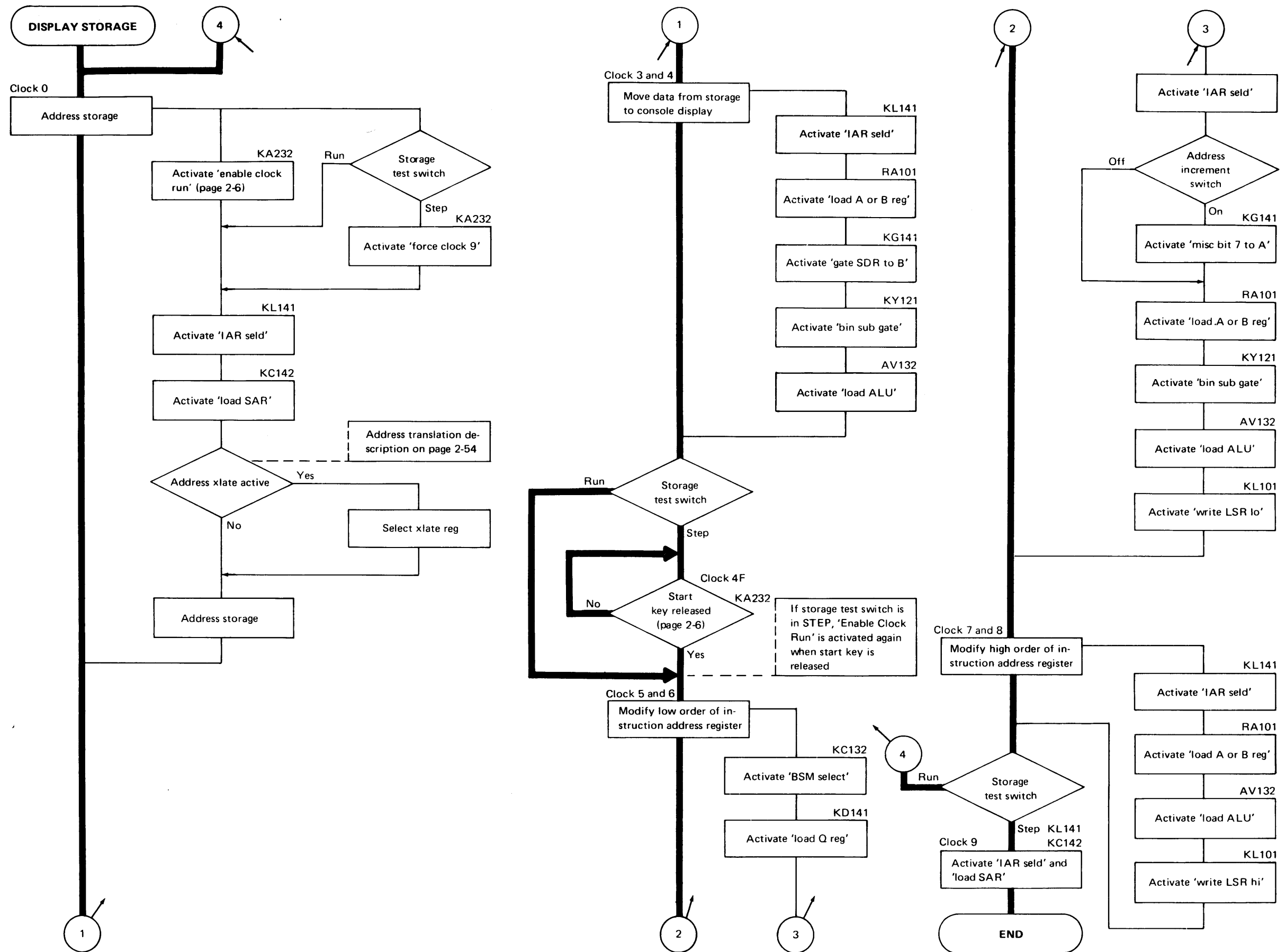
Recycles until Storage Test switch is changed to STEP or if in initial memory scan, until 'memory scan complete' is active.

Note: If in initial memory scan, timing is the same as if the Storage Test switch were set to RUN.

### DISPLAY STORAGE (Part 1 of 3)

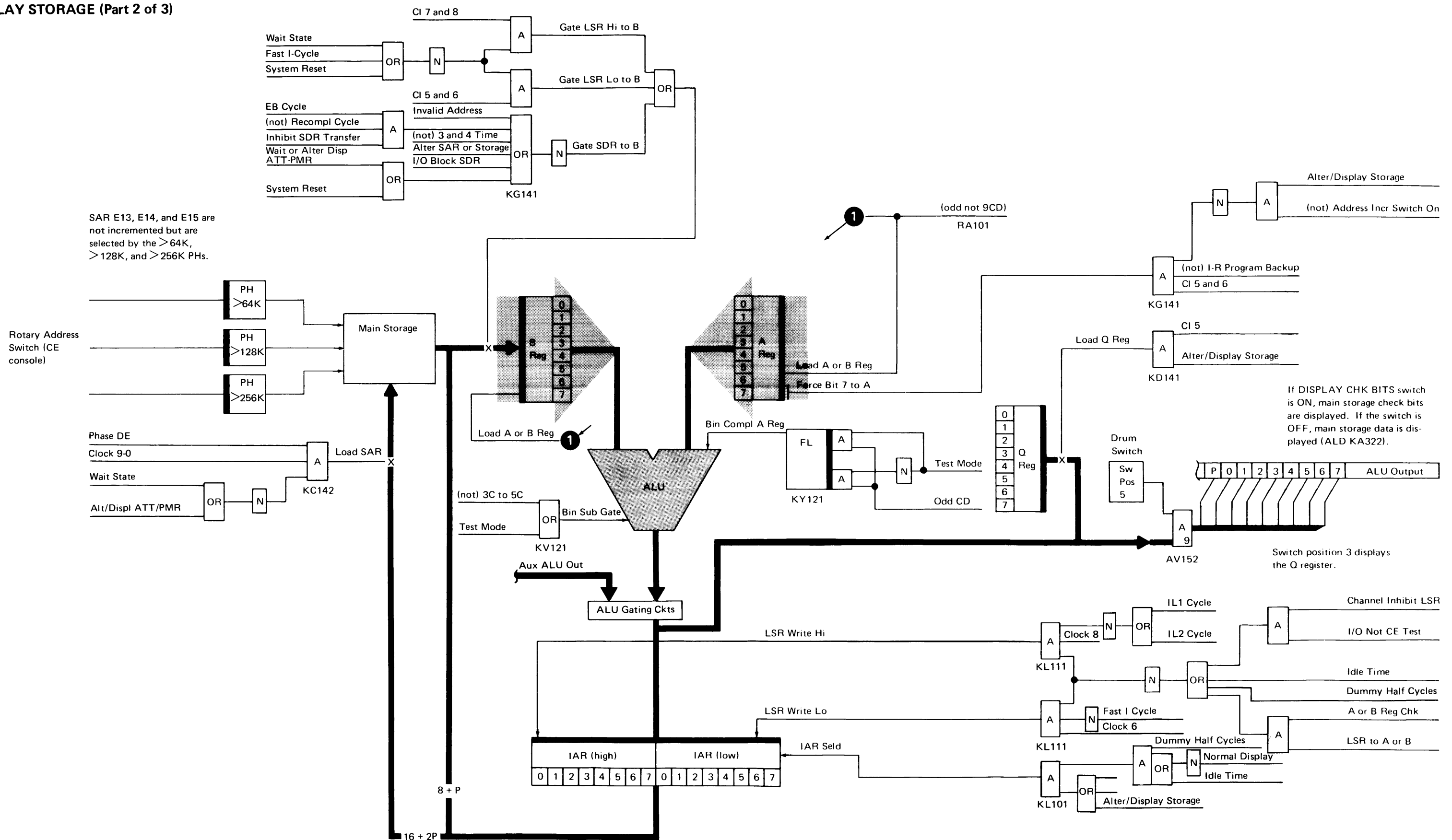
#### Objectives

- Transfer data from storage position addressed by SAR into Q register and display in console lights. Drum switch (roller display) is in position 5 for ALU output, position 3 for Q register output.
- DISPLAY CHK BITS switch set on determines if display is data or check bits (ALD KA322).
- With STORAGE TEST switch in STEP; pressing start key advances clock through 4 time. Releasing start key advances clock through 9 time.
- With STORAGE TEST switch in RUN; clock recycles, skipping 9 time, until switch is returned to STEP.
- ADDR INCREM switch ON causes IAR to be incremented each CPU cycle.
- EXTENDED SAR ADDRESS BITS rotary switch selects BSM (SAR E15, E14, and E13 are not incremented).
- ADDR INCREM switch OFF causes data to be transferred from the same storage location each cycle.
- If storage address is displayed SAR/MSAR switch should be on MSAR (see page 6-18).



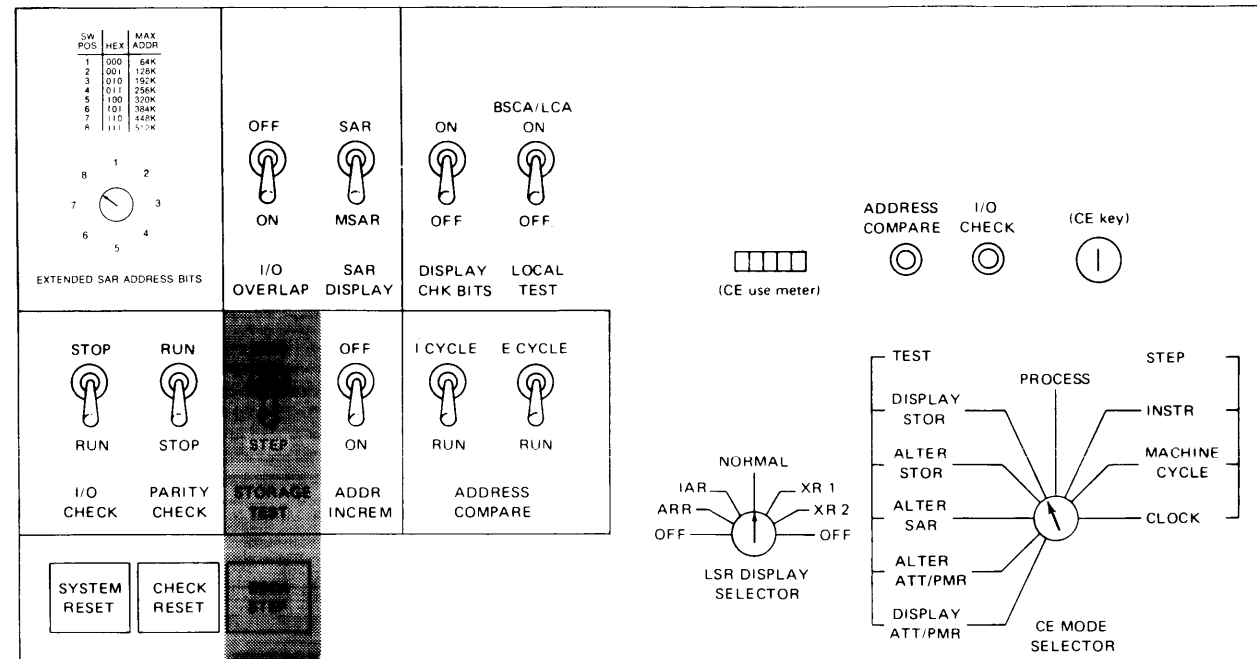


**DISPLAY STORAGE (Part 2 of 3)**



This page intentionally left blank

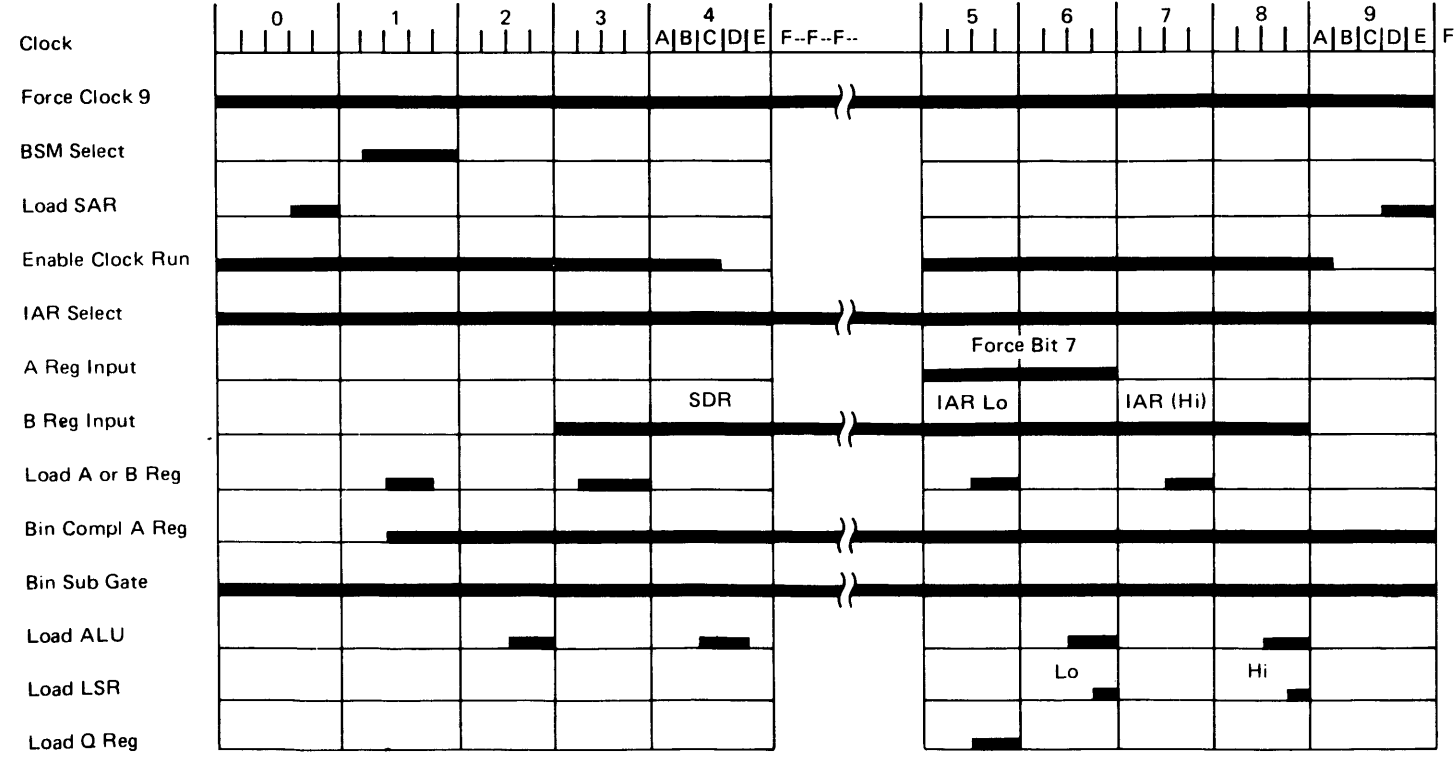
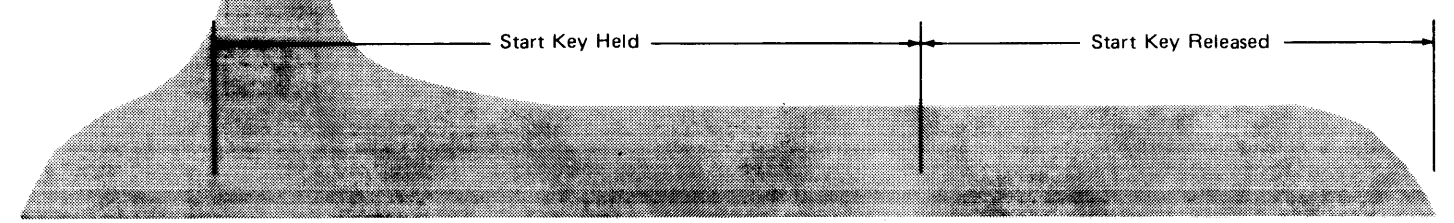
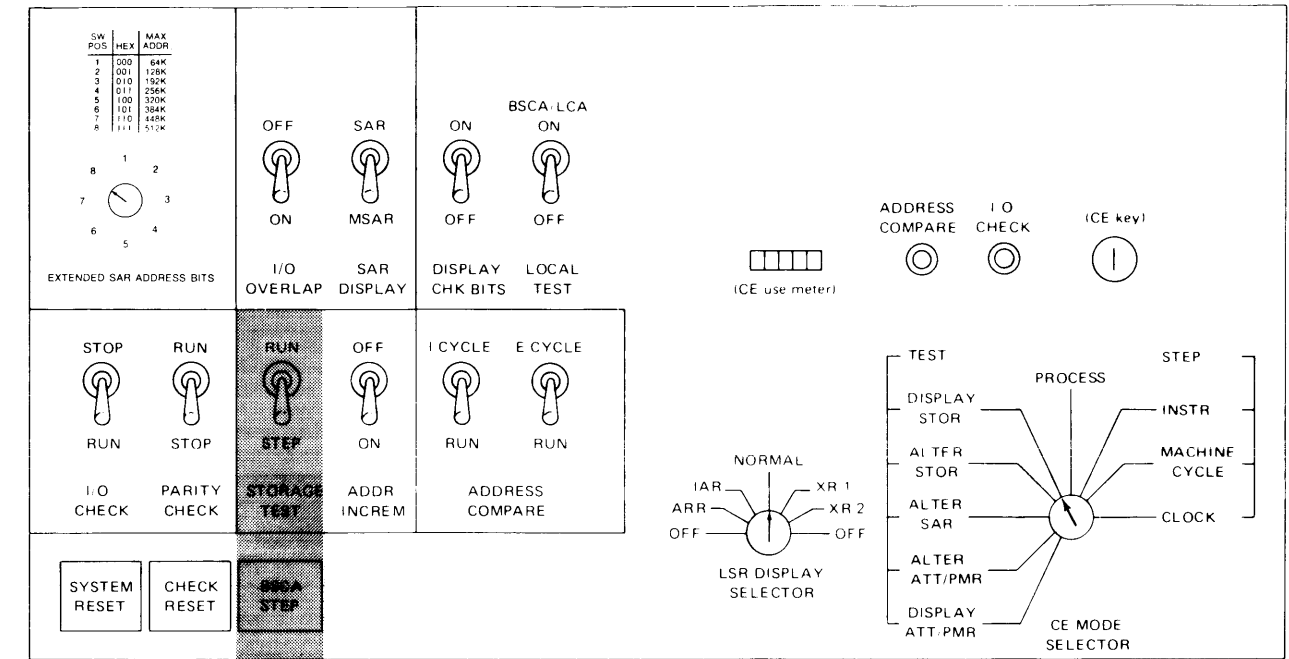
### DISPLAY STORAGE (Part 3 of 3)



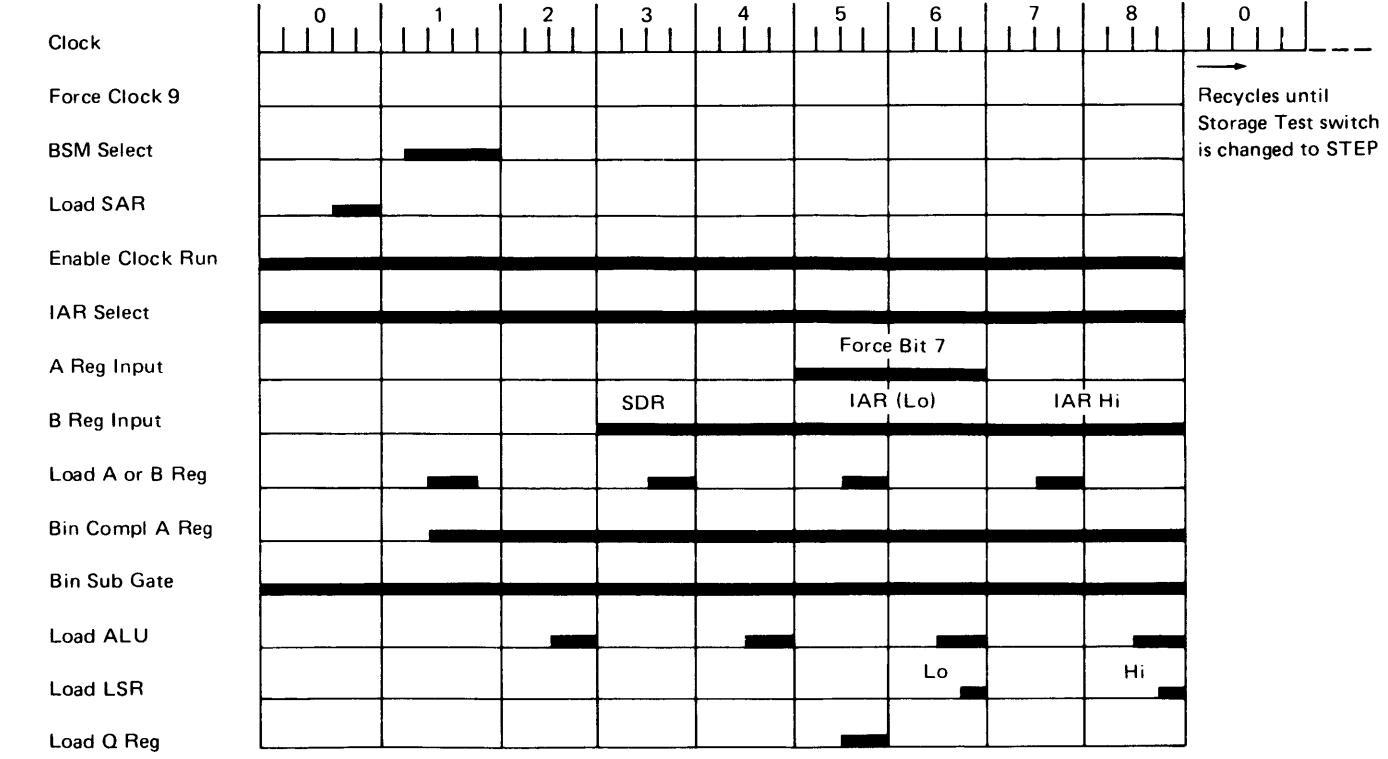
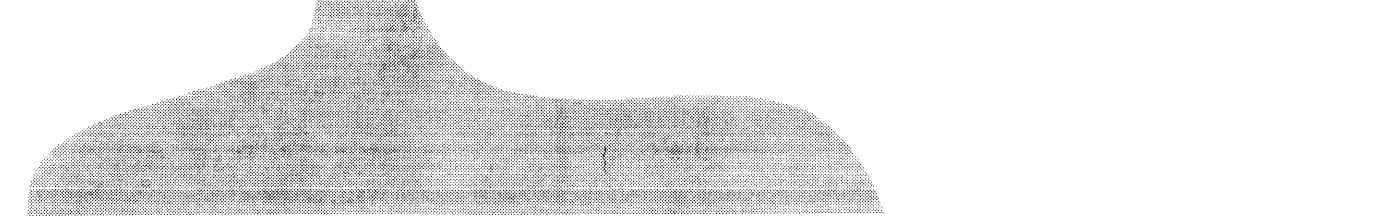
**ADDR INCREM Switch:**

ON — IAR incremented by one each CPU cycle.

OFF — IAR addresses same storage position each CPU cycle.



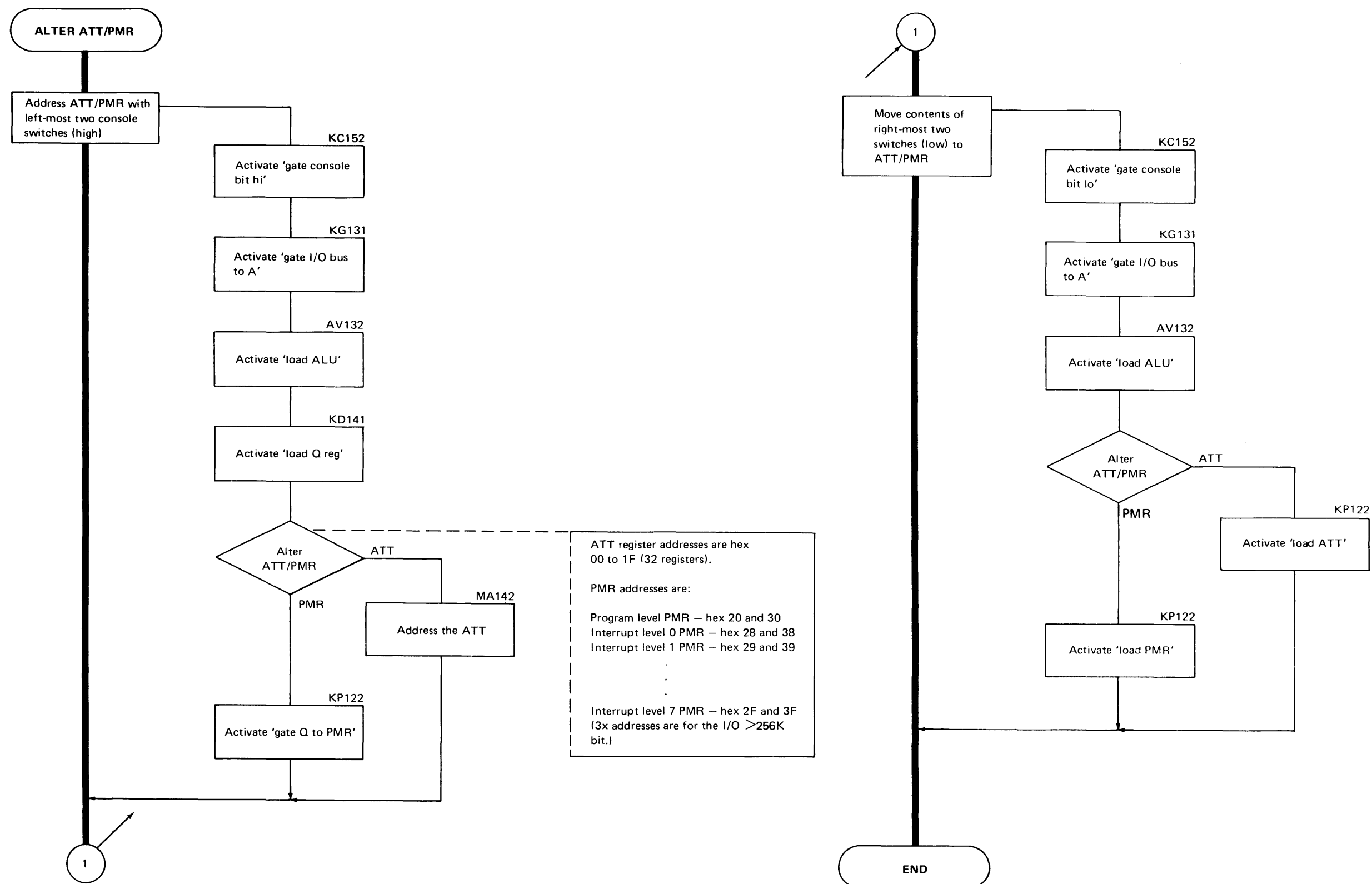
- ALD Reference
- KC122
  - KA232
  - KC132
  - KC142
  - KA232
  - KL141
  - RA111
  - RA101
  - RA101
  - KY121
  - KY121
  - AV132
  - KL101
  - KD141



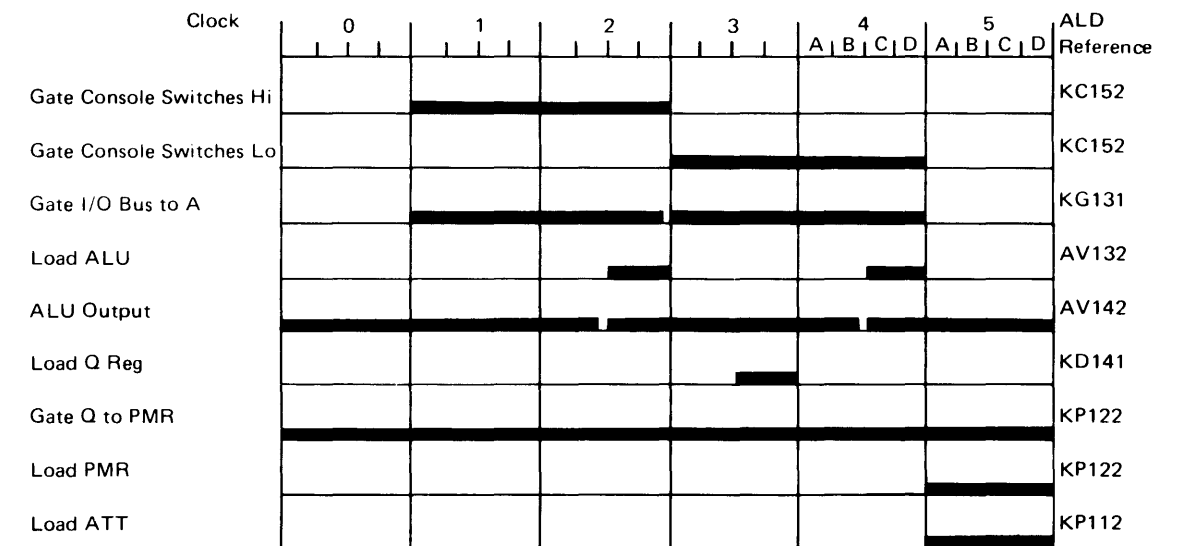
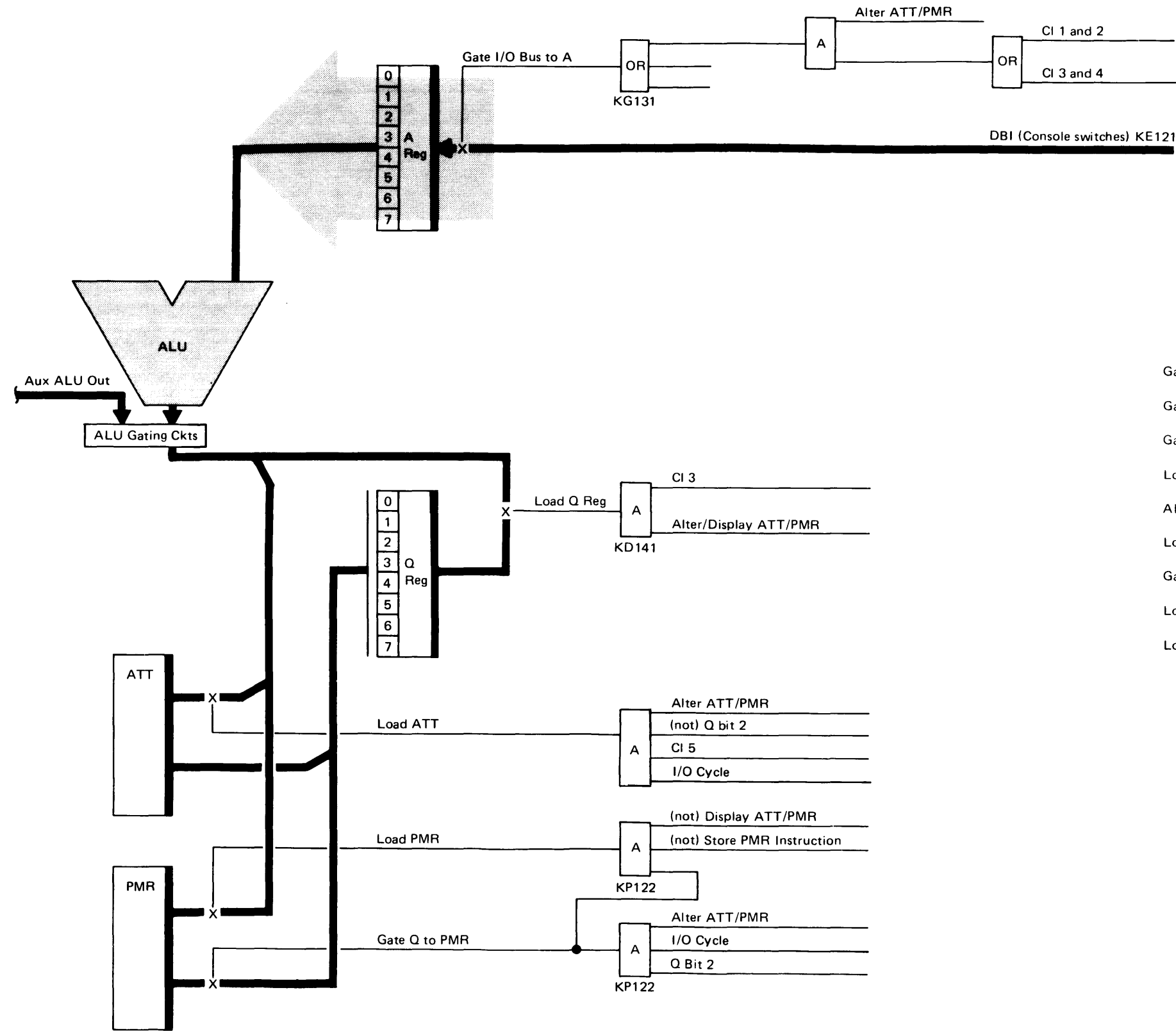
**ALTER ATT/PMR (Part 1 of 2)**

**Objectives**

- Move address from left-most two console ADDRESS/DATA switches into Q register.
- Address ATT/PMR.
- Move data from right-most two console switches into ATT or PMR register addressed.



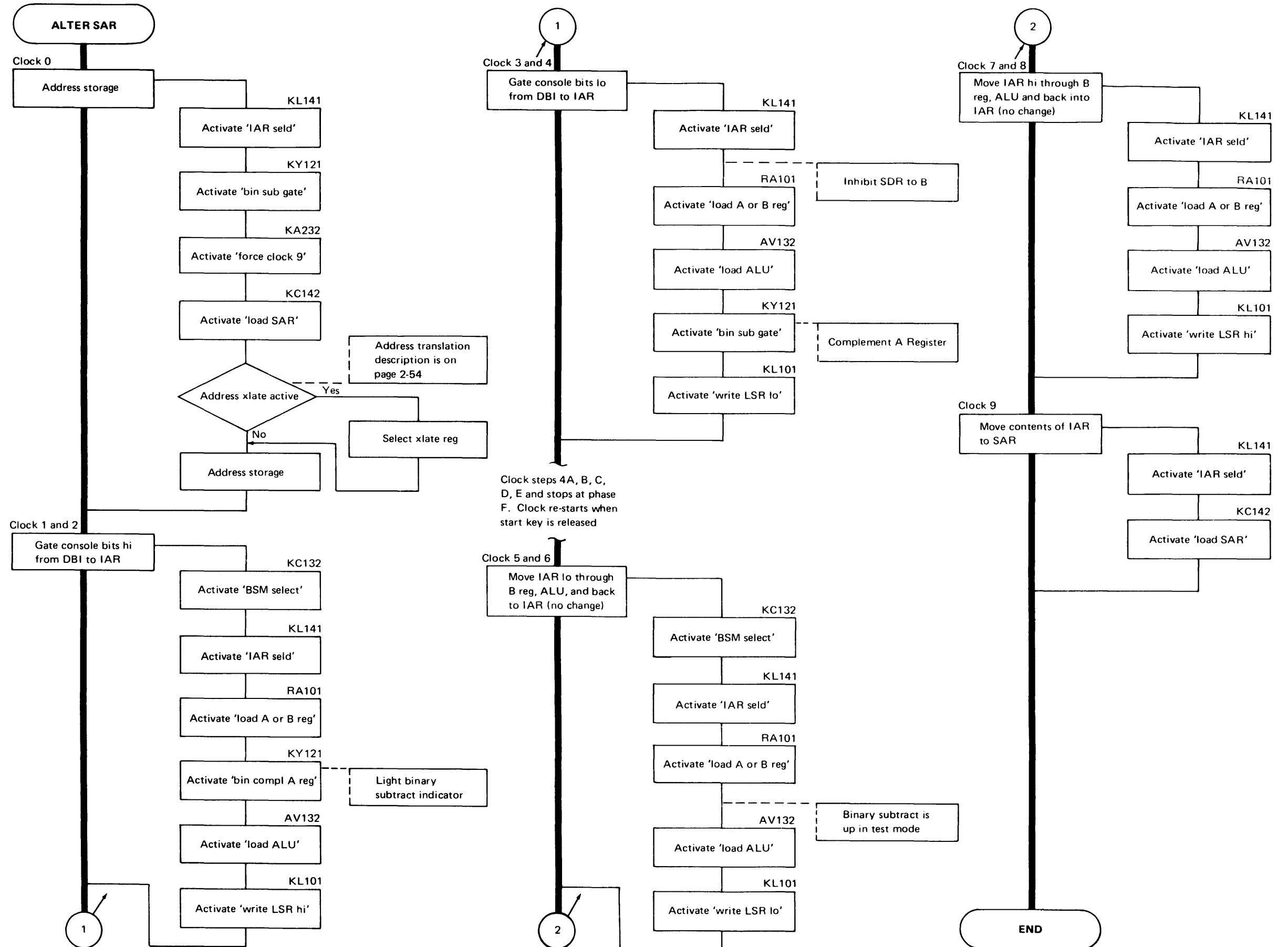
ALTER ATT/PMR (Part 2 of 2)



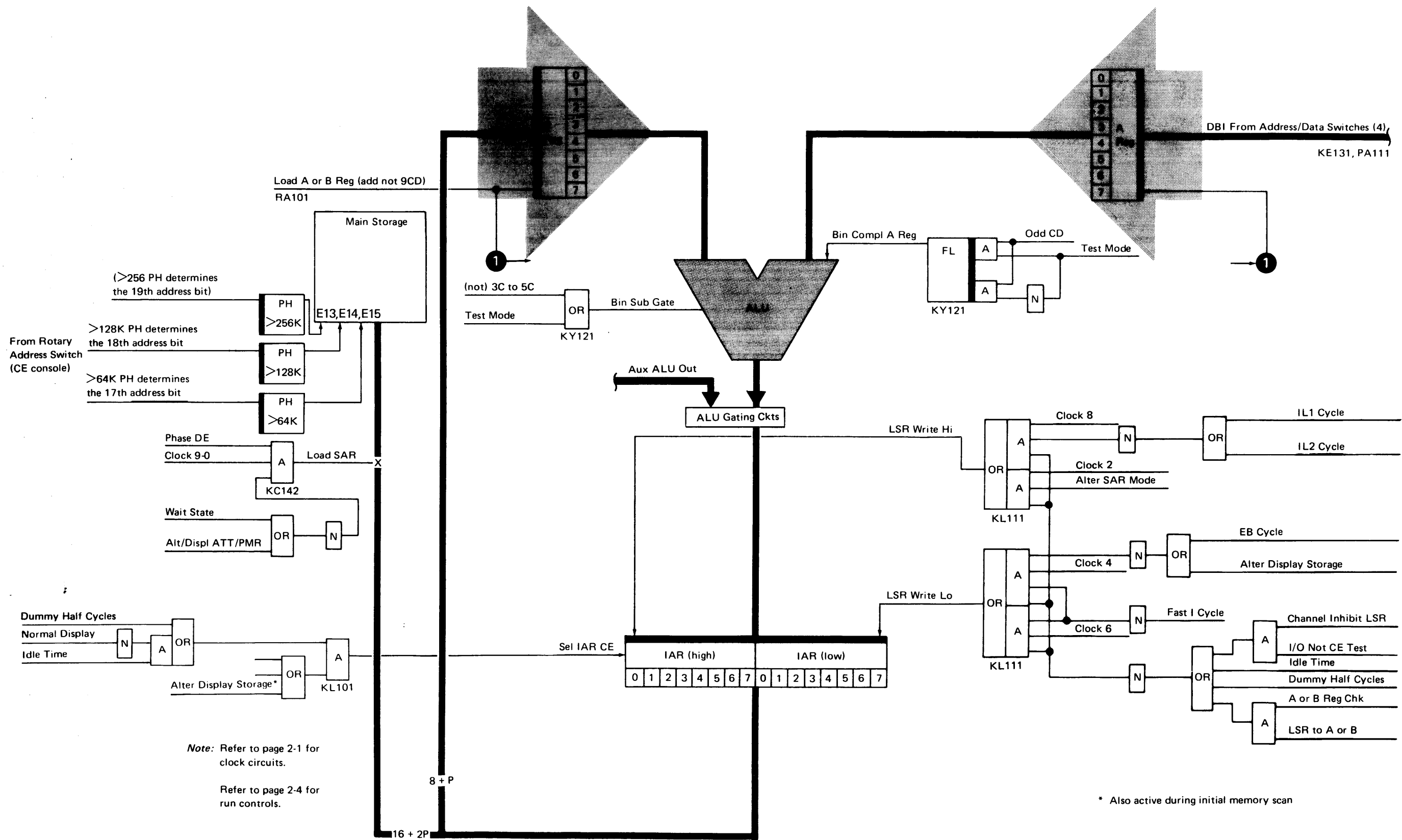
### ALTER SAR (Part 1 of 3)

#### Objectives

- Load contents of four console Address/Data switches into the Instruction Address Register (IAR) by way of Data Bus In, A register and ALU.
- Load SAR from the IAR and the >64K, >128K, and >256K PHs (ALD KP132) at clock 9 time.
- When start key is pressed, the clock runs 0 through 4. When start key is released, clock runs from 5 through 9.
- SAR/MSAR switch should be on MSAR (see page 6-18).



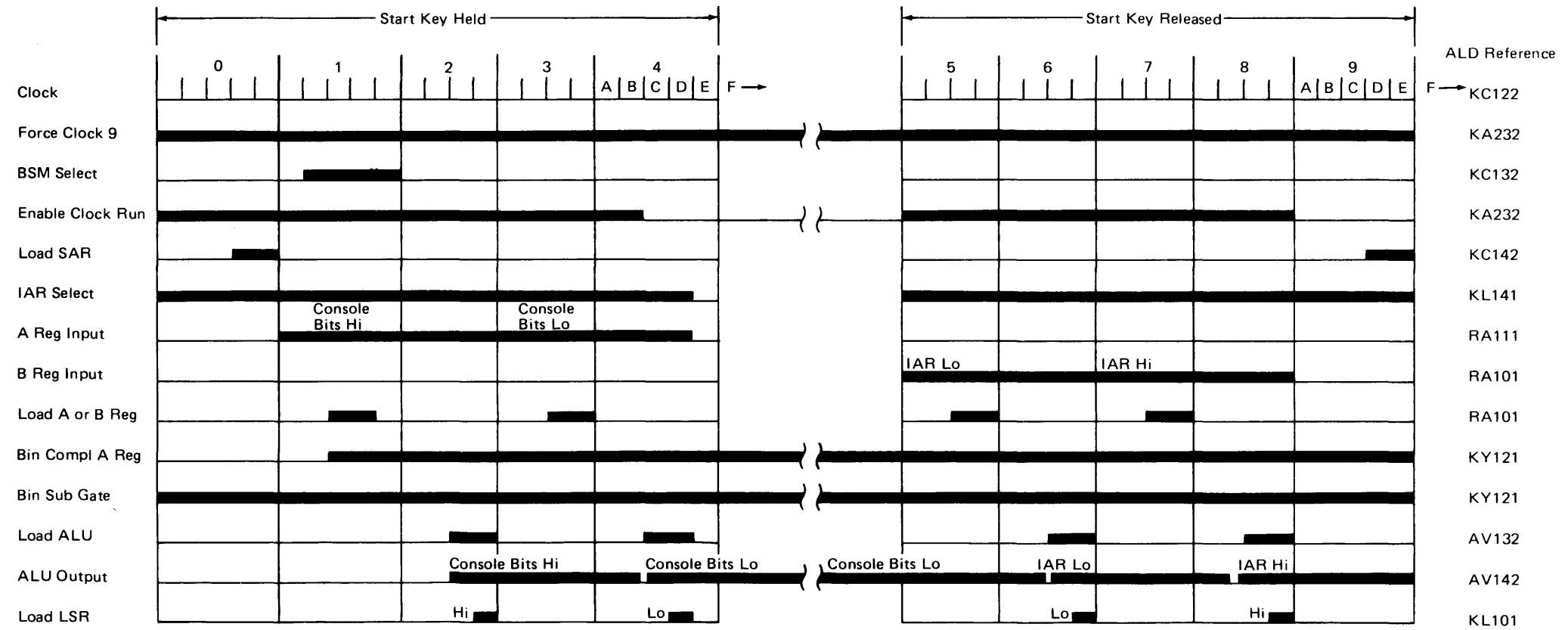
ALTER SAR (Part 2 of 3)



This page intentionally left blank



ALTER SAR (Part 3 of 3)



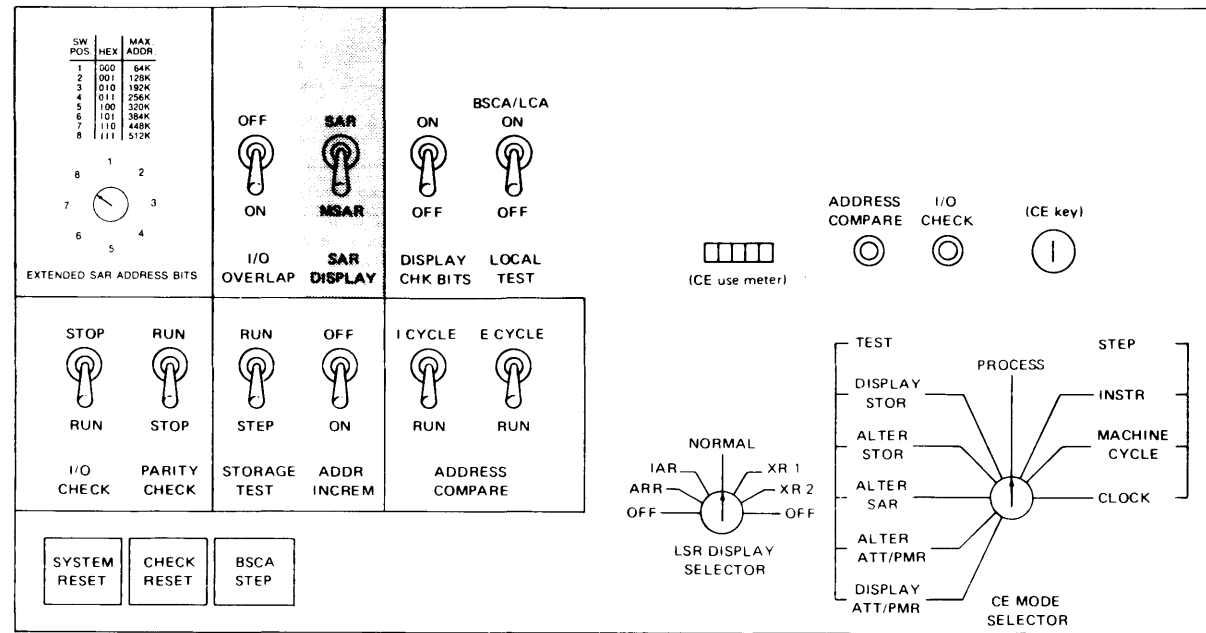
**DISPLAY SAR/MSAR**

SAR DISPLAY switch determines what is displayed in roller position 1.

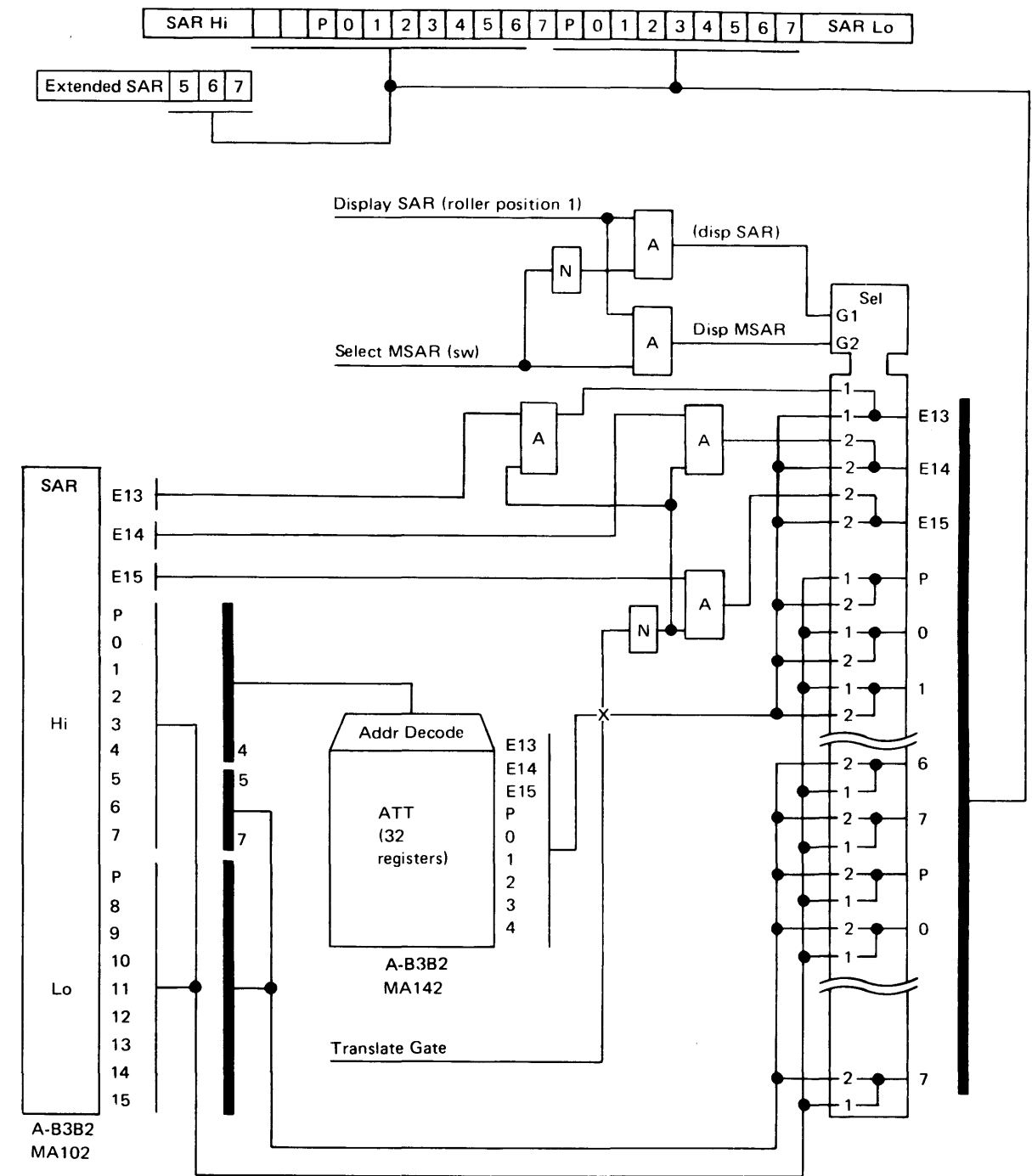
**Switch Setting**

**Displays**

- SAR** All SAR positions except E13, E14, and E15
- MSAR (translate active)** ATT contents plus SAR positions 5-15
- MSAR (translate inactive)** All SAR positions including E13, E14, and E15



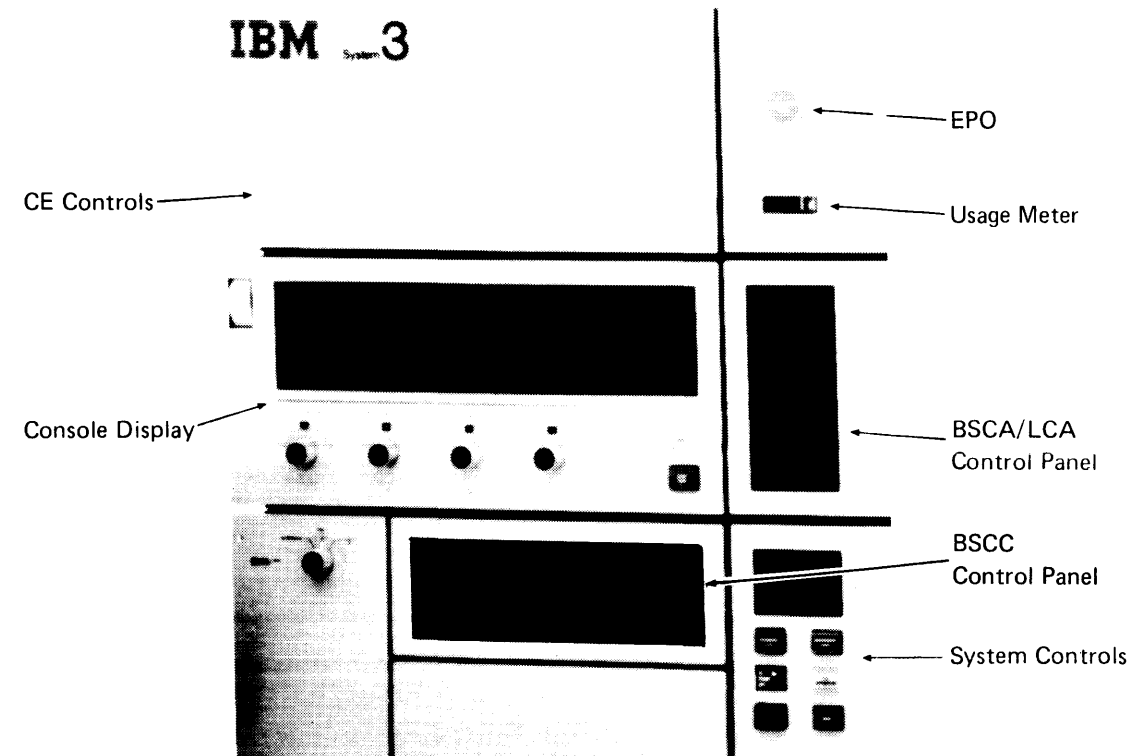
**Roller Position 1**



## SYSTEM CONSOLE

The system control panel contains the lights and switches required to operate and control System/3.

System controls include the: operator controls, console display, and customer engineering (CE) controls. The operator section contains the controls required for normal operation. The console display panel provides the operator and the CE with a visual record of the contents of the various registers in the CPU and the status of the major CPU controls. The CE controls serve as diagnostic aids in locating malfunctions.



## OPERATOR CONTROLS

### The Emergency Power-Off (EPO) Pull Switch (ALD YA102)

Pulling this switch turns off the power beyond the power-input terminal on every unit that is part of the system. The switch latches when you pull it out.

When the emergency pull switch is out, the POWER ON/OFF switch is ineffective.

### Usage Meter (ALD YA104)

The customer usage meter records system operating time. The meter begins recording time when you press START or PROGRAM LOAD and ends when the job is completed.

However, when operating in one of the step modes, the meter runs for 400  $\mu$ s each time START is pressed.

Time is not recorded if one of the following conditions exists:

- Manual or programmed halts stop the processing unit. (However, time is recorded when I/O operations are being performed during a programmed halt.)
- A processor check occurs.
- Power is lost.
- The CE key is turned off for system servicing.

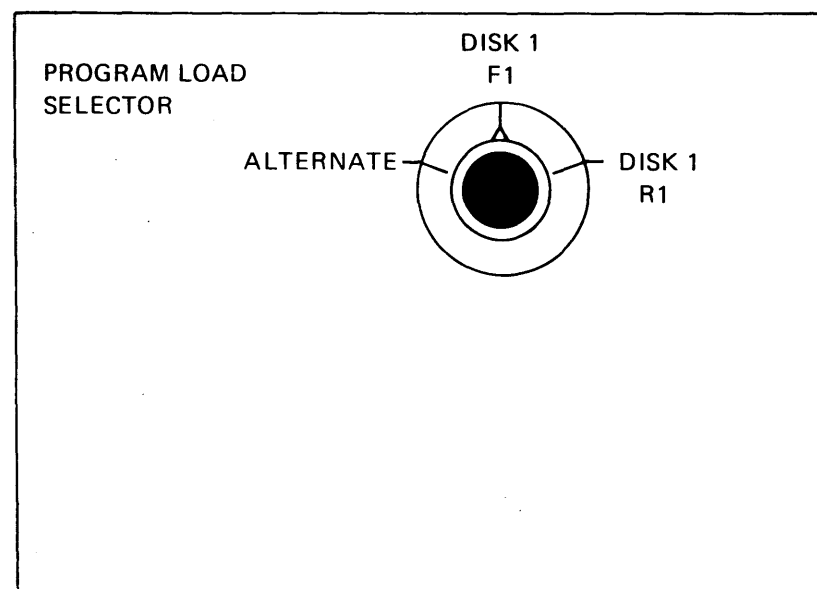
## File Control Panel

### PROGRAM LOAD SELECTOR Switch without 3344 installed (ALD PC101, YD100)

This switch is used to select 3340 disk drive 1. The 3340 F1 and R1 switch positions refer to 5444 simulation areas on the 3348 data module.

The alternate position refers to one of the card I/O devices designated as an IPL device. These are the 1442 or 2560 for reading 80-column cards or 5424 for reading 96-column cards.

*Note:* When the channel terminate feature (card-less system) is installed, the 3741 becomes the alternate program load device.

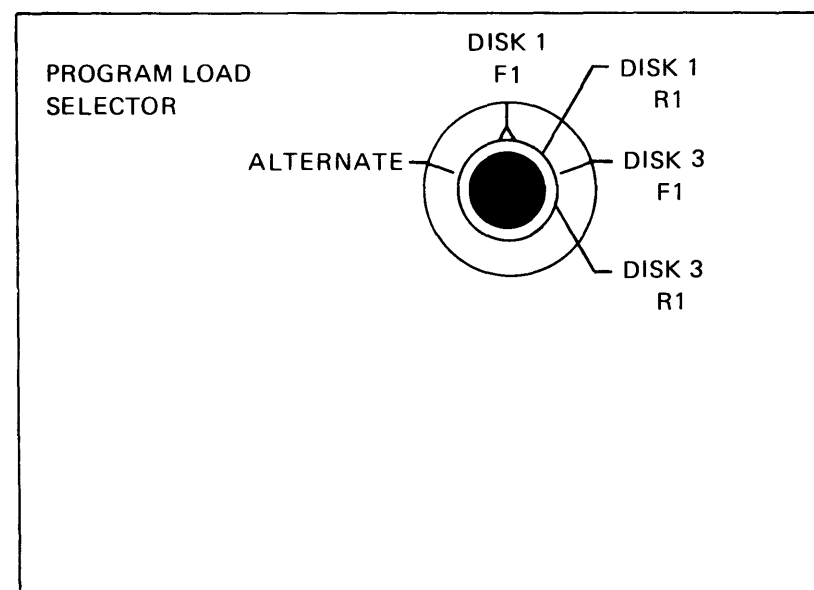


### PROGRAM LOAD SELECTOR Switch with 3344 installed (ALD PC101, YD100)

This switch is used to select disk drive 1 (3340) or disk drive 3 (3344). The DISK 1 F1 and R1 switch positions refer to 5444 simulation areas on the 3348 data module. The DISK 3 F1 and R1 switch positions refer to 5444 simulation areas on the 3344 fixed media storage.

The alternate position refers to one of the card I/O devices designated as an IPL device. These I/O devices are the 1442 or 2560 for reading 80-column cards, or 5424 for reading 96-column cards.

*Note:* When the channel terminate feature (card-less system) is installed, the 3741 becomes the alternate program load device.



### PROCESSOR CHECK Light (ALD PC111)

This light being on indicates one of the following:

- An invalid operation code, an invalid address, or a parity error is detected in the CPU.
- The device address (including the M field) and the N field of an I/O instruction is not recognized.
- The I/O device recognizes a parity error on data bus out at the I/O attachment.
- The immediate I/O error stop is on and an I/O error occurs.

This light is turned off when a system reset occurs or when the CE check-reset key is activated.

The processor stops on any of the above errors, and I/O data may be lost. The console display shows the error. Following a processor check, use the program load procedure for a normal restart.

### POWER ON/OFF Switch (ALD YA102, PC111)

This switch initiates the power on/off sequence of the system. As part of the power on/off sequence, a system reset is performed so that no I/O operations take place until they are specifically directed. The contents of main storage are not guaranteed after power on/off sequence.

*Note:* A power check occurs if the power on/off switch is turned on before a normal system power off sequence is completed.

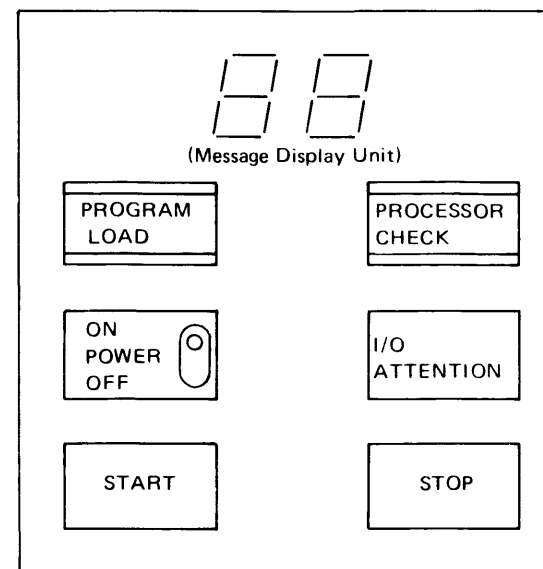
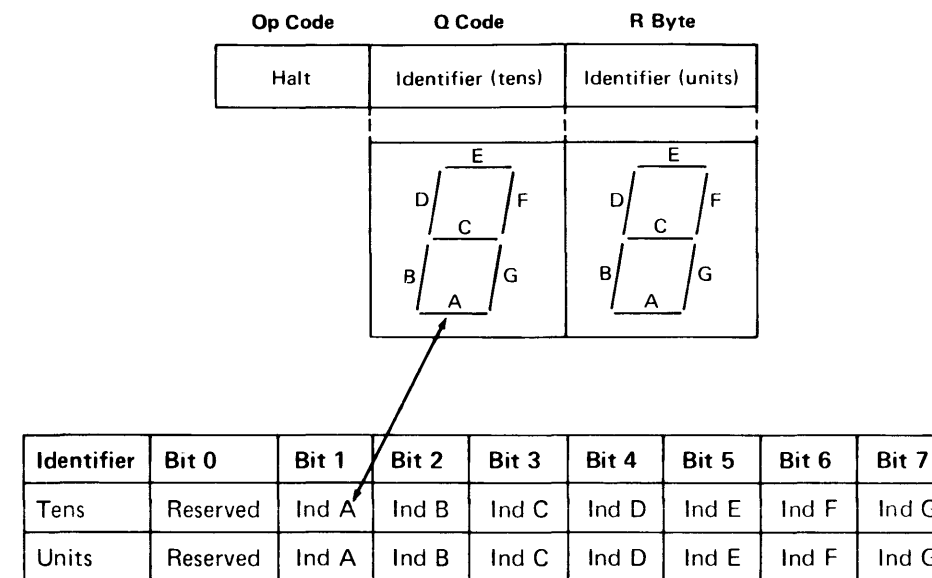
### START Key (ALD PC111)

Pressing the start key takes the processor out of its stopped state, turns off the stop light, and allows the processor to continue. In the CE mode of operation the start key is also used to start the processor clock and then sequentially advance it.

### Message Display Unit (ALD PB131)

This two-position message display unit keeps a running display of the halt identifier portion of a halt instruction.

Refer to page 5-66 for halt instruction description.



### I/O ATTENTION Light (ALD PC111)

I/O ATTENTION lights when an addressed I/O device requires normal operator intervention. Normal operator intervention includes:

- Printer – forms out, cover interlock.
- MFCU – hopper empty, stacker full, chip box full, cover interlock.

The light goes off when the operator has intervened and returned the device to the ready state. I/O attention does not stop normal CPU processing. However, most start I/O or load I/O instructions are not accepted. The exception is: Interrupt enable/disable instructions are always accepted.

### PROGRAM LOAD Key (ALD PC111)

This key is pressed to start initial program loading from the IPL device. The I/O device is selected with the program load selector switch. A system reset is performed as part of the program load sequence.

Pressing PROGRAM LOAD allows the first record or card from the disk file or card device to be read and stored in main storage, beginning at location 0000. When the key is released, the CPU proceeds to execute the instruction sequence starting at location 0000. Normal program load from the MFCU is executed through the primary hopper.

Should the I/O device selected be not ready, I/O ATTENTION lights when PROGRAM LOAD is pressed. Normally, to complete the program load function, it is only necessary to ready the device.

### STOP Key/Light (ALD PC111)

Pressing this key stops the processor at the end of the operation being performed. I/O transfers are completed without losing information. STOP lights to indicate processor stop. The processor may be restarted without loss of information by pressing START.

**CONSOLE DISPLAY**

**INT LEV (Interrupt Level) Light (ALD PB111) **A****

A single indicator lamp is used to monitor whether any interrupt level is being serviced. The INT 1, INT 2, and INT 4 lights indicate which interrupt level (binary encoded) is being serviced.

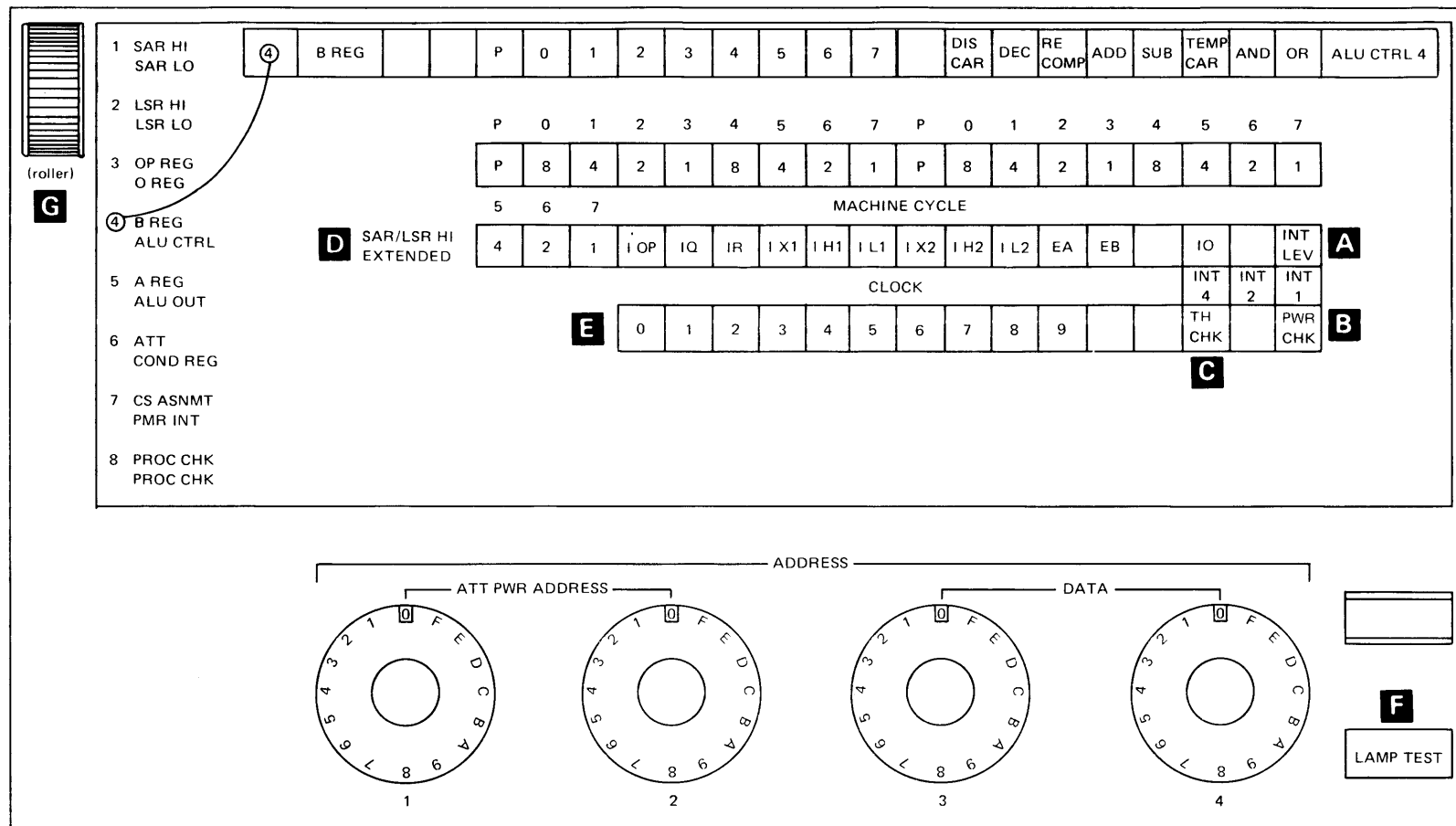
**PWR CHK (Power Check) Light (ALD YA102) **B****

This light comes on when a machine power supply malfunctions or when a thermal condition exists. This light also comes on during a power on sequence and remains on until the sequence is completed.

For additional information refer to the chart below.

**TH CHK (Thermal Check) Light (ALD YA102) **C****

The thermal check light and the power check light come on when overheating occurs in the CPU main frame. Turning OFF the POWER switch turns off the PWR CHK light. The TH CHK light remains on until the thermal condition is corrected. Then the normal power on sequence can be performed.



Console Display

**MACHINE CYCLE Indicators (ALD PB111) **D****

Twelve indicator lamps represent the twelve machine cycles. They identify the processor cycle just completed in all modes except the CE clock step mode; in the CE clock step mode, they indicate the cycle in progress. Three indicator lamps indicate the SAR/LSR HI EXTENDED condition.

**CLOCK Indicators (ALD PB121) **E****

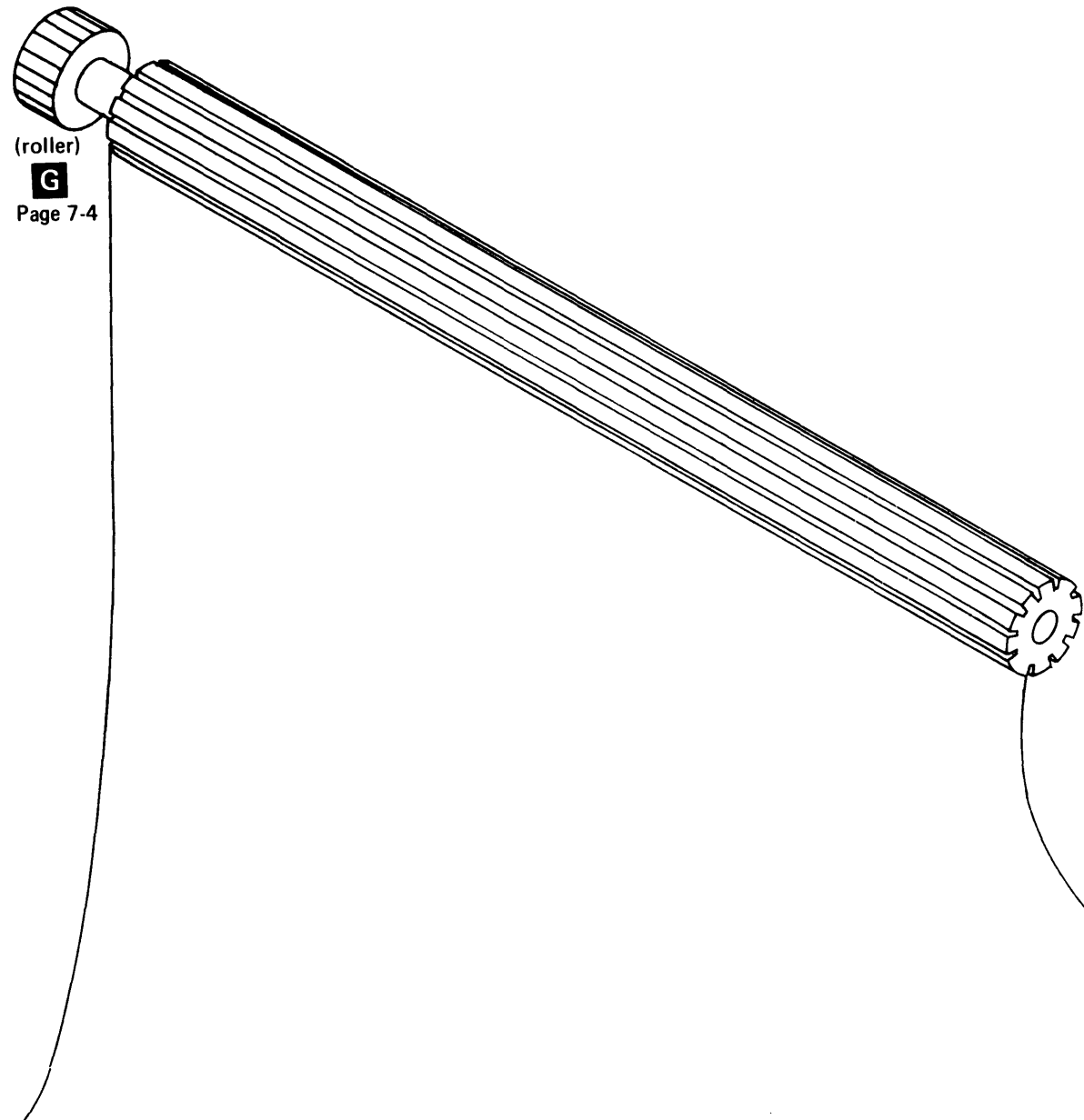
Ten indicator lamps represent clocks 0 through 9 which can be stepped through in the CE clock step mode. In the normal process mode, a machine cycle consists of clocks 0 through 8. Clock 9 is used with the CE step and test modes.

**LAMP TEST Key (ALD YA102) **F****

This key turns on all system lights so that you can check for burned out lamps.

Fault	Power Check/Thermal Indications			Action
	POWER ON/ OFF Switch	Indicators		
		PWR CHK	TH CHK	
Internal power supply malfunction	On	On	Off	1. Turn POWER OFF 2. Correct the problem 3. Press CHECK RESET 4. Turn POWER ON
Thermal condition	On	On	On	1. Turn POWER OFF 2. PWR CHK goes off 3. TH CHK stays on until condition is corrected
Customer power source loss	On	On	On	1. Turn POWER OFF 2. All indicators turn off 3. Turn POWER ON and continue operation
Emergency power off (EPO) switch pulled out	On	Off	Off	1. Turn POWER OFF 2. Correct the problem 3. Push in EPO switch 4. Turn POWER ON

CONSOLE DISPLAY (Continued)



Register Display Unit (ALD PB101, PC101)

**G** The register display unit consists of a row of twenty lights and eight legend strips mounted on an eight-position roller. At any one time, only one of the eight strips is visible through a cutout in the console above the row of lights. The legend strip and the corresponding register displayed by the row of lights are selected with the eight position switch.

Each legend strip, by number, is as follows:

1. **SAR HI/SAR LO**—Contents of storage address register (high and low) or MSAR depending on SAR display toggle switch. The SAR/LSR HI EXTENDED lights are also used.
2. **LSR HI/LSR LO**—Contents of LSR selected by the LSR display selector. The SAR/LSR HI EXTENDED lights are also used.
3. **OP REG**—Contents of the op register.  
**Q REG**—Contents of the Q register.
4. **B REG**—Contents of the B register.  
**ALU CTL**—The state of the following ALU controls are displayed as follows:  
 DIG CAR (digital carry)  
 DEC (decimal instruction)  
 RE COMP (recomplement)  
 ADD (addition)  
 SUB (subtraction)  
 TEMP CAR (temporary carry)  
 AND  
 OR

5. **A REG**—Contents of the A register.  
**ALU OUT**—Contents of the output of the ALU.
6. **ATT**—Contents of the address translate table registers. The ATT displayed will be the active ATT register unless the alter/display ATT function is being used. In this case, the addressed ATT register will be displayed.  
**COND REG**—The contents of the condition register is displayed as follows:  
 BIN OVF (binary overflow)  
 TF (test false)  
 DEC OVF (decimal overflow)  
 HI (high)  
 LO (low)  
 EQ (equal)
7. **CS ASNMT**—Cycle steal assignment is displayed as it is presented to the I/O devices on the I/O interface.  
**PMR**—The PMR displayed is the active PMR unless the alter/display PMR function is being used. In this case, the addressed PMR will be displayed.
8. **PROC CHK**—The processor checks are displayed as follows:  
 I/O LSR indicates selection of an LSR by an I/O device was not performed correctly.  
 LSR indicates parity is incorrect on the output of the LSR (see CE LSR Display Selector).  
 SAR/ATT indicates parity is incorrect in the storage address register or in the ATT register.

- MSAR indicates parity is incorrect at the memory end of the storage address register output lines.
- INV ADDR indicates that the SAR contains an invalid address (storage address exceeds the system storage size).
- STOR PROT indicates an attempt was made to read from or write into a protected address in main storage.
- SDBI indicates parity is incorrect at the input to main storage.
- SDBO indicates an uncorrectable main storage data error occurred.
- CAR indicates the carry out of the ALU is incorrect.
- DBI indicates parity is incorrect on the CPU end of the data bus in coming from the I/O devices.
- A/B indicates parity is incorrect in the A register or B register.
- ALU indicates parity is incorrect at the output of the ALU.
- CPU DBO indicates parity is incorrect on the CPU end of the data bus out to the I/O devices.
- OP/Q indicates parity is incorrect in the op register or Q register.
- PRIV OP indicates an attempt was made to execute a privileged instruction when the system was not in privileged mode.
- INV OP indicates an invalid op code in the op register.
- CHAN DBO indicates parity is incorrect on the I/O device end of the data bus out from the CPU.
- INV Q indicates an invalid Q byte is present in an I/O instruction.

1	SAR HI			P	0	1	2	3	4	5	6	7	P	0	1	2	3	4	5	6	7	SAR LO
2	LSR HI			P	0	1	2	3	4	5	6	7	P	0	1	2	3	4	5	6	7	LSR LO
3	OP REG			P	0	1	2	3	4	5	6	7	P	0	1	2	3	4	5	6	7	Q REG
4	B REG			P	0	1	2	3	4	5	6	7		DIG CAR	DEC	RE COMP	ADD	SUB	TEMP CAR	AND	OR	ALU CTL
5	A REG			P	0	1	2	3	4	5	6	7	P	0	1	2	3	4	5	6	7	ALU OUT
6	ATT			P	0	1	2	3	4	5	6	7	P			BIN OVF	TF	DEC OVF	HI	LO	EQ	COND REG
7	CS ASNMT			P	0	1	2	3	4	5	6	7	I/O >256K	I/O >128K	EB ATT	EA ATT	1 ATT	PRIV	I/O >64K	STOR PROT	MASK INT	PMR
8	PROC CHK			I/O LSR	LSR	SAR ATT	MSAR	INV ADDR	STOR PROT	SDBI	SDBO	CAR	DBI	A/B	ALU	CPU DBO	OP/Q	PRIV OP	INV OP	CHAN DBO	INV Q	PROC CHK

**BSCA OPERATOR'S PANEL**

**BSCA Attention Light**

The following table shows the conditions indicated by this light:

Instruction	Condition Indicated
Any receive or transmit and receive or (on non-switched and multipoint networks only) receive initial.	Data set is not ready.
Auto call or receive initial on switched network.	Auto call unit power is off or data line is being used.
Any SIO except control SIO.	Either (1) the BSCA is disabled, or (2) the external test switch is on and BSCA is not in test mode.
None.	Data set is not ready.

**Unit Check Light**

This light turns on when any bit in status byte 2 is on. Also, when an SNS transition or SNS stop register instruction is executed, it is possible for an LSR, S register, or DBI register parity check to occur, resulting in a unit check condition with the unit check light on. Under such a condition, the status byte 2 bits may all be zero.

The unit check indicator signifies that the BSCA program should enter an error recovery procedure.

**Data Terminal Ready Light**

This light indicates that the BSCA is enabled and that the data terminal is ready for use.

**Data Set Ready Light**

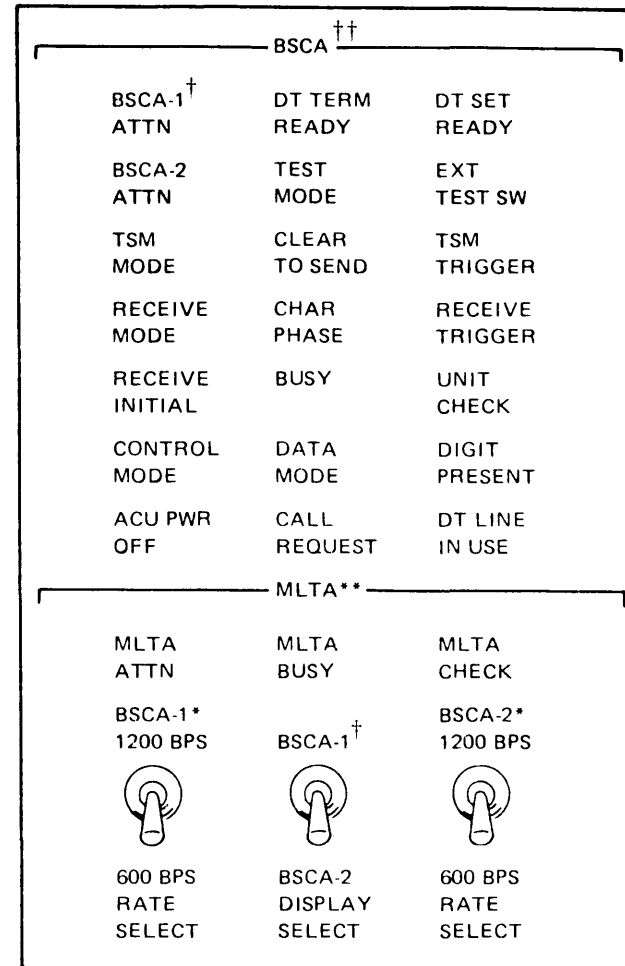
The DT SET READY light indicates that the data set ready line from the data set is on and that the data set is ready for use.

**Clear To Send Light**

This light indicates that the clear to send line from the data set is on and that the adapter may now transmit.

**Receive Trigger Light**

This light indicates the status of the receive trigger. The light is on when the trigger is at a binary 0 state.



\* Rate select switch is for machines used outside the United States. If the rate selection feature is specified on either of the BSCAs, it will be made available to both.  
 \*\* MLTA is available by RPQ only.  
 † This reads LCA on machines equipped with the local communications adapter (LCA) feature.  
 †† This reads LCA/BSCA on machines equipped with the LCA feature.

**Transmit Trigger Light**

The TSM TRIGGER light indicates the status of the transmit trigger. The light is on when the trigger is at a binary 0 state.

**Receive Mode Light**

This light indicates that the adapter has been instructed to perform a receive operation.

**Transmit Mode Light**

The TSM MODE light indicates that the adapter has been instructed to perform a transmit operation.

**Receive Initial Light**

This light is turned on by an SIO receive initial instruction. It is turned off at the end of the receive initial operation.

**Busy Light**

This light indicates that the communication adapter is executing a receive initial, transmit and receive, auto call, receive or loop test instruction.

**Character Phase Light**

The CHAR PHASE light indicates that the adapter has established character synchronism with the transmitting station. This light is turned on at the end of receive operations and whenever character synchronism is lost.

**Data Mode Light**

This light is turned on by the decoding of an SOH or STX during a transmit or a receive operation. It is turned off at the end of the transmit or receive operation.

**Control Mode Light**

This indicator is used only on systems that have the station select feature installed. The light is turned on by an EOT sequence during a transmit, receive, or receive initial monitor operation, when the station select feature is installed. It is turned off by the decoding of an SOH or STX.

**Digit Present Light**

This light indicates that a digit has been obtained from storage for the auto call unit when the auto call feature has been installed.

**Auto Call Unit Power Off Light**

The ACU PWR OFF light indicates that the auto call unit (special feature) power is off.

**Call Request Light**

On systems with the auto call feature installed, this light indicates that the communication adapter has received an SIO auto call instruction and is performing an auto call operation.

**Data Line in Use Light**

On systems with the auto call unit installed, the DT LINE IN USE light indicates that the data line occupied line from the auto call unit is on.

**Test Mode Light**

This light indicates that the program has placed the adapter in a test mode of operation.

**External Test Switch Light**

The EXT TEST SW light indicates that the switch at the data set end of the medium speed data set cable is in the test position. For high speed data sets, this light is on when the local test switch on the CE panel is in the on position.

**Rate Select Switch**

This switch, which is present only on systems installed outside the U.S.A. that have the rate selection feature as well, controls the rate of transmission and reception of data.



## CE CONTROLS

CE control switches should be changed only when the CPU is stopped.

### ADDRESS/DATA Switches (Below Console Display) (ALD PA111)

These four switches set up addresses or data. An address (16 bits) is loaded into the instruction address register (IAR). Data can be entered into main storage — 8 bits when operating in test mode or 16 bits via a program.

## SYSTEM RESET Key (ALD PC101)

When SYSTEM RESET is pressed, the system enters an immediate idle state. CPU registers, controls, and status indicators are reset and the processor clock is allowed to 'idle'.

The program instruction address register (P-IAR) and program status register (P-PSR) are both reset to zero by a system reset. The active PMR is set to ATT off, SP off, MI off, I/O 17th and 18th bits off, and privileged mode on.

The system must be in the PROCESS mode of operation for the pushbutton to be operative. After power on, the system reset key should be pressed prior to any CE operation. Refer to page 6-2 for system reset cycle description.

## CE Key Switch (ALD PC111)

This switch, when turned to the CE position, prevents the customer use meter from running.

*Note:* A processor check may occur if the switch position is changed while the clock is running.

### CHECK RESET Key (ALD PC101)

This key is pressed to cause a reset of the Processor and/or Input/Output check conditions.

A check reset removes the current error conditions, thus allowing the processor to resume its operation after the Start key is pressed.

## BSCA STEP Key (ALD HE160)

This key, which is effective only when the BSCA is in step mode, causes the BSCA to advance one bit time each time the key is pressed.

### ADDRESS COMPARE Light (ALD PB121)

This light is on whenever the register display is positioned to SAR and the ADDRESS/DATA switches match the SAR/MSAR (depends on setting of the SAR display switch). The system will not stop unless one of the ADDRESS COMPARE switches is set to stop (stop on I cycle or stop on E cycle).

### I/O CHECK Light (ALD PB121)

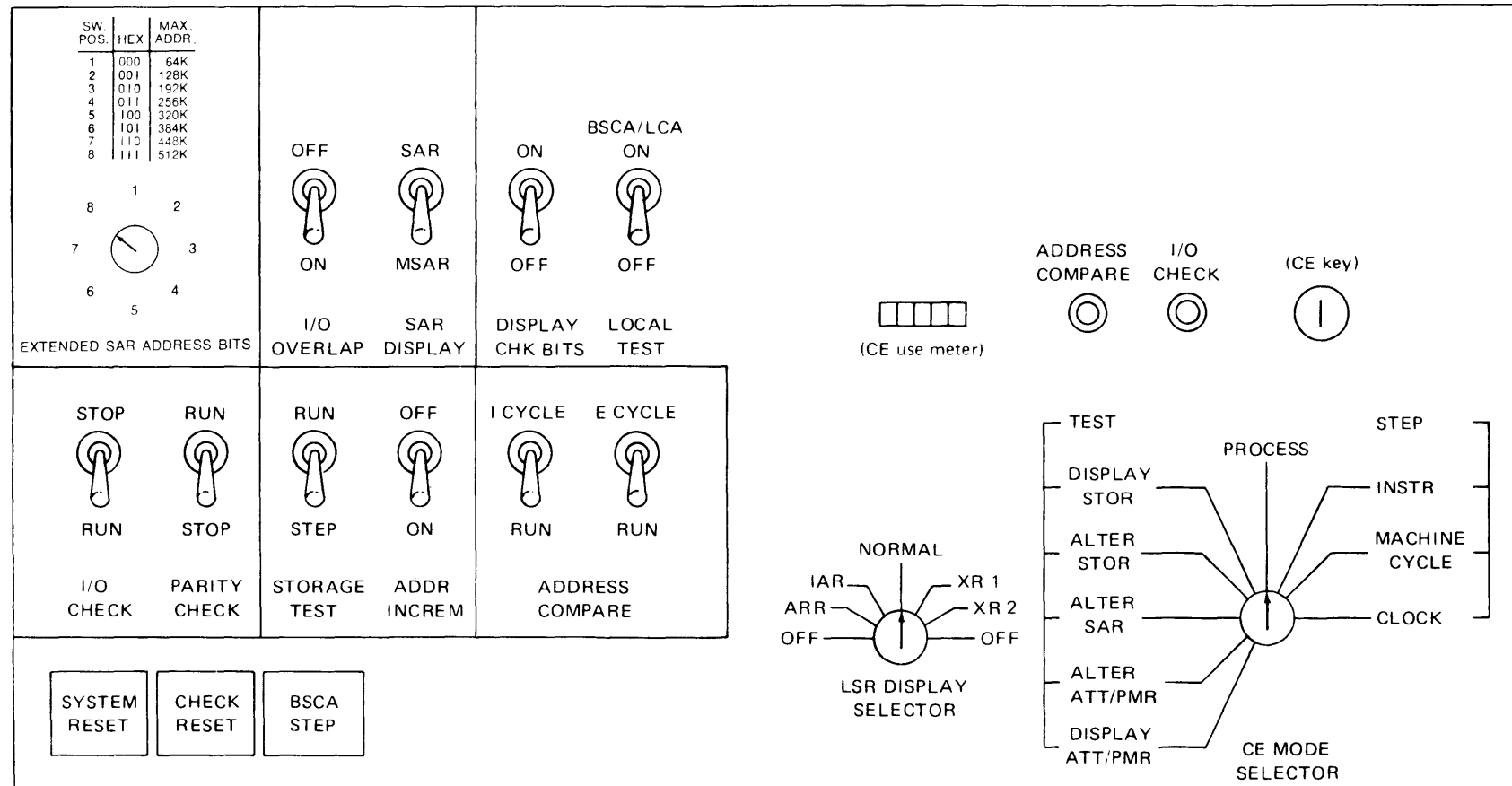
This light is turned on when certain I/O errors (read check, punch check, etc) are detected by an addressed I/O device. It is turned off when a system reset occurs, CHECK RESET is pressed, or at the next SIO.

### LSR DISPLAY SELECTOR (ALD PC111)

This rotary switch selects the local storage register (LSR) to be displayed in position 2 of the roller display.

The LSRs that can be displayed with this switch are: IAR, ARR, XR 1, and XR 2. The selected LSR is displayed whenever the CPU is in a wait state.

When the switch is set to NORMAL, the LSR displayed is the one being used by the program. Keep the selector switch set at NORMAL while running. The LSR parity check display is for the selected LSR. If the switch is not set to NORMAL, parity checks may not be displayed or the LSR display may show wrong information.



### CE MODE SELECTOR (ALD PA101)

This rotary switch selects one of three processor operating modes: (1) TEST; (2) PROCESS; (3) STEP. PROCESS is the mode for normal system operation.

**Notes:**

- To prevent a processor check, the CPU should be in a halt state before changing the position of the CE MODE SELECTOR switch.
- After power on, the CE MODE SELECTOR switch should be placed in the PROCESS mode position and the SYSTEM RESET key pressed. This restores main storage to correct parity.

#### TEST **A**

The switch settings under test mode permit the following:

- DISPLAY STOR** (display storage). The contents of the storage location specified by SAR are transferred into the B register when START is pressed, and into the Q register when the key is released.

The ATT is not active and the address used is the 16 bits from the IAR plus the >64K, 128K and >256K bit values (set with the >64K, >128K and >256K PH CE). If the value in the IAR is not the desired address, it can be changed using an alter SAR operation.

- ALTER STOR** (alter storage). Pressing START allows transfer of the data set up in the right-most two address/data switches, into the A register. Releasing START causes this data to be placed in storage and the Q register. The storage address is specified by the IAR and the >64K, >128K, and >256K PHs. Refer to Alter Storage Cycle description.

- ALTER SAR.** The address set up in the address/data switches and the EXTENDED SAR ADDRESS BITS switch is transferred into the IAR and the >64K, >128K and >256K PHs.

When alter SAR is used to enter an address in preparation for an alter/display storage operation the address entered must be the real address. The ATT is disabled in test mode. When alter SAR is used to enter an address for manual branching to a routine, the address entered must be a logical address. The EXTENDED SAR ADDRESS BITS switch is ignored when the mode selector is switched to PROCESS or STEP.

Refer to page 6-14 for Alter SAR cycle description.

- Alter ATT/PMR.** Pressing START transfers data from the two right-most console data switches into the A register. Releasing START transfers this same data into the ATT register or PMR specified by the two left-most switches.\*

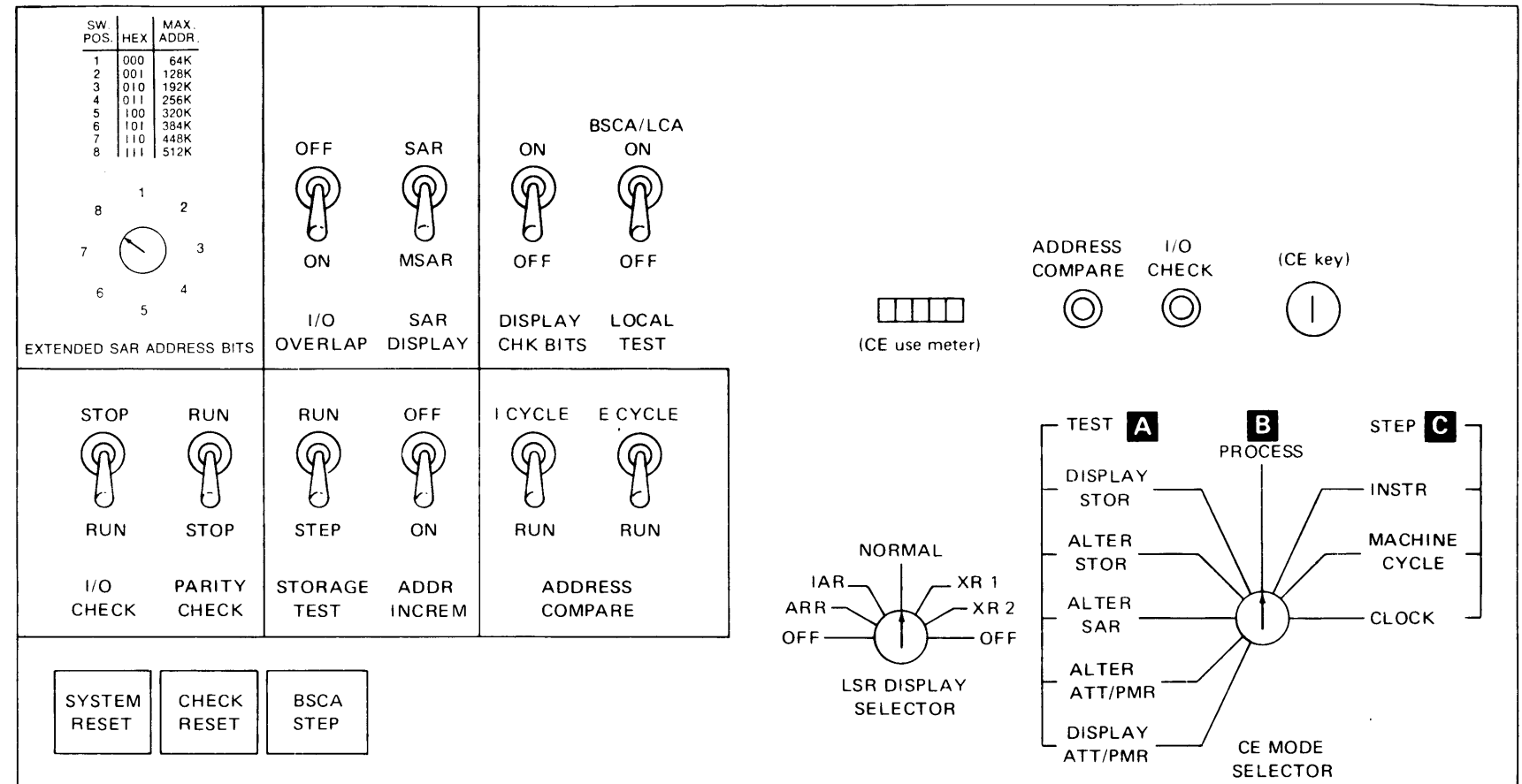
*Note:* Invalid addresses are not detected in test mode.

- DISPLAY ATT/PMR.** If an ATT register address is set into the two left-most address switches,\* pressing START transfers the address to the ATT where one of the 32 registers is selected for display. The content of the selected register is displayed in roller position 6.

If the two left-most address switches select a PMR register,\* pressing START transfers the address into the Q register. This, in turn, selects the PMR and displays the contents of that register in roller position 7.

Refer to page 2-54 for a complete description of the ATT registers. Refer to page 2-28 for a complete description of the PMR.

\*Hex 00 through 1F are used to address the 32 ATT registers. Hex 20 and 28 through 2F (30 and 38 through 3F for the I/O > 256K PMR bit) are used to address the nine program mode registers (PMR). (Hex 20 = program level PMR, 28 = interrupt 0 PMR, 29 = interrupt 1 PMR.)



CE Panel

#### PROCESS **B**

The CE mode selector switch is set to this position for normal system operation.

#### STEP **C**

In the step mode, the rotary switch setting controls the manner in which the processor performs the stored program.

- INSTR** (instruction). Each time START is pressed and released, one complete instruction is performed. The I phase is performed while the key is pressed, and the E phase, if any, when it is released.

*Note:* Any SIO instruction that causes the clock to run, as described under CLOCK in the following paragraph, also causes the next sequential instruction to be executed without pressing START.

- MACHINE CYCLE.** Each time START is pressed and released, the instruction is advanced through one machine cycle. Pressing START causes data in storage to be accessed and modified as required, and causes the result to be displayed in the arithmetic and logical unit (ALU) indicators of the roller display. Release of START, depending upon the operation being performed, writes either the old data or the new result back into storage.

- CLOCK.** Each time START is pressed, the clock advances through an odd-numbered clock; each time START is released, the clock advances through an even-numbered clock. The integrity of I/O data transfers is preserved by allowing the clock to run from I phase end of every executable start I/O instruction to the time the device is finished transferring data. START is not functional while this I/O transfer is taking place.

*Note:* The halt identifier lights do not turn on in any step mode switch setting.

## CE Switches

### EXTENDED SAR ADDRESS BITS Switch

This switch is used to enter SAR bits E13, E14, or E15 for an alter/display storage or an address compare operation above 256K, 128K, or 64K of storage respectively.

The switch bit is set in a PH; then, on an alter/display storage or address compare stop, the bit is gated into SAR E13, E14, or E15.

The switch is inactive in process mode (except for address compare stop).

### >64K ADDR BIT Switch (ALD PA101)

This switch is used to enter SAR bit E15 for an alter/display storage or an address compare operation when storage above 64K must be addressed.

The switch bit is set into a PH, then, on an alter/display storage or address compare stop, is gated into SAR E15.

The switch is inactive in process mode (except for address compare stop).

### SAR/MSAR Display Switch (ALD 'sel MSAR sw' PA101)

This switch controls the display in roller position one.

If the switch is in the SAR position, the untranslated or logical address is displayed.

If the switch is in the MSAR position, the actual address sent to main storage is displayed. If translation is active, the ATT contents and SAR positions 5-15 are displayed. If translation is inactive all SAR positions including E13, E14, and E15 are displayed.

SAR E13, E14, and E15 may be set by an I/O LSR or the > 256K, > 128K, and > 64K PHs.

### I/O OVERLAP Switch (ALD PC121)

This switch controls the system so that I/O operations may be executed in either an overlap or a nonoverlap mode. With the switch in the normal ON position, I/O operations are executed in an overlap mode. When the switch is in the off position, I/O operation is completed prior to execution of the next sequential instruction (nonoverlap). To avoid processor checks, be sure the CPU is stopped before changing this switch.

### DISPLAY CHK (Check) BITS Switch (ALD PC121)

This switch is used only in display storage mode; when set ON, it displays the six main storage check bits for the current half-word. These check bits are displayed in positions 0-5 of the Q register display lights. Q register bit positions 6 and 7 are always on.

These lights are accurate only when SAR is odd.

### BSCA/LCA LOCAL TEST (ALD HE160)

This switch sets the high speed data set into local test mode and causes data to be wrapped around through the data set with a SIO loop test instruction.

### I/O CHECK Switch (ALD PA101)

This switch forces the processor to come to an immediate stop on an I/O error. The switch is normally set to RUN. With the switch set to STOP, the processor stops on an I/O error and the console display indicates the processor status at the time the error stop occurred.

A check reset followed by pressing START allows the program to continue.

### PARITY CHECK Switch (ALD PA101)

This switch allows processor parity errors to be ignored.

The switch is normally set to STOP. This causes the processor to come to an immediate stop whenever a parity error is detected. Pressing CHECK RESET followed by START allows the program to continue. With the PARITY CHECK switch set to RUN, parity errors are detected and displayed, but the processor is not stopped.

### STORAGE TEST Switch (ALD PA101)

This switch enables the altering or displaying of storage as follows:

1. When the switch is set to STEP, a storage location is accessed each time START is pressed.
2. When the switch is set to RUN, pressing START causes storage to loop on either the same location repetitively or all addresses sequentially within one BSM. (See *Addr Increm [Address Increment] Switch.*)

*Note 1:* SAR bits E13, E14, and E15 are not changed during console operations. Looping occurs only in the BSM addressed at the beginning of the operation.

*Note 2:* The STORAGE TEST switch must be in the STEP position to avoid a processor check while switching the CE mode selector between ALTER STOR and DISPLAY STOR.

### ADDR INCREM (Address Increment) Switch (ALD PA101)

This switch enables address incrementing in CE test mode switch settings of alter storage or display storage. With the switch set ON, the contents of SAR are incremented by 1 after each storage access. When the switch is set OFF, SAR is not incremented.

*Note:* SAR bits E13, E14, and E15 are not changed during console operations. Looping occurs only in the BSM addressed at the beginning of the operation.

**ADDRESS COMPARE Switches (ALD PA101)**

The two ADDRESS COMPARE switches **A** are used with SAR DISPLAY to stop on I cycles, and/or stop on E cycles, and/or I/O cycles of either a real or logical address.

When the SAR DISPLAY switch is set to SAR, the logical address is compared to the address set in the four console address switches. The >64K, >128K, and >256K PHs are ignored.

When the SAR DISPLAY switch is set to MSAR, the real address is compared to the address set in the four console address switches plus the EXTENDED SAR ADDRESS BITS switch.

When both compare switches are set to RUN, no processor stop will occur. When the ADDRESS COMPARE switch is set to I CYCLE, an address compare stop will occur in I cycles.

When the ADDRESS COMPARE switch is set to E CYCLE, an address compare stop will occur on E cycles and I/O cycles.

If both switches are set to stop, an address compare stop will occur whenever a compare is detected.

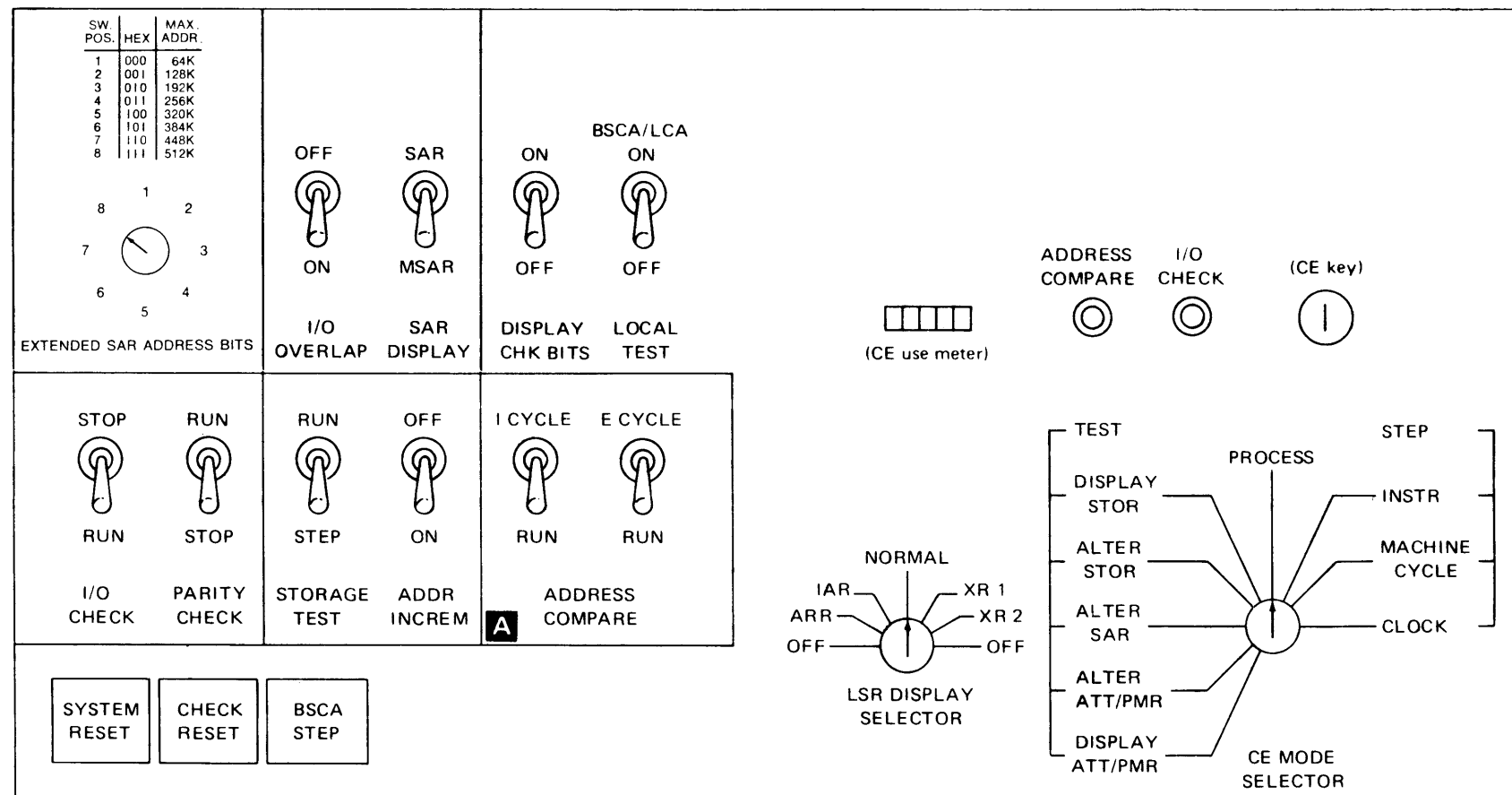
*Note:* When an address compare occurs, processing stops and all cycle lights are off. Since the translate function is gated with either ADDRESS COMPARE switch being set to RUN, the SAR display will not necessarily match the switches.

If either switch is set to stop and a match occurs, the processor will stop at the completion of this cycle. The clock continues to run. If an I/O operation is not complete, that operation continues until completed. Therefore, after stopping the processor with an address match, the I/O device may change SAR to some address different from the one set into the address switches.

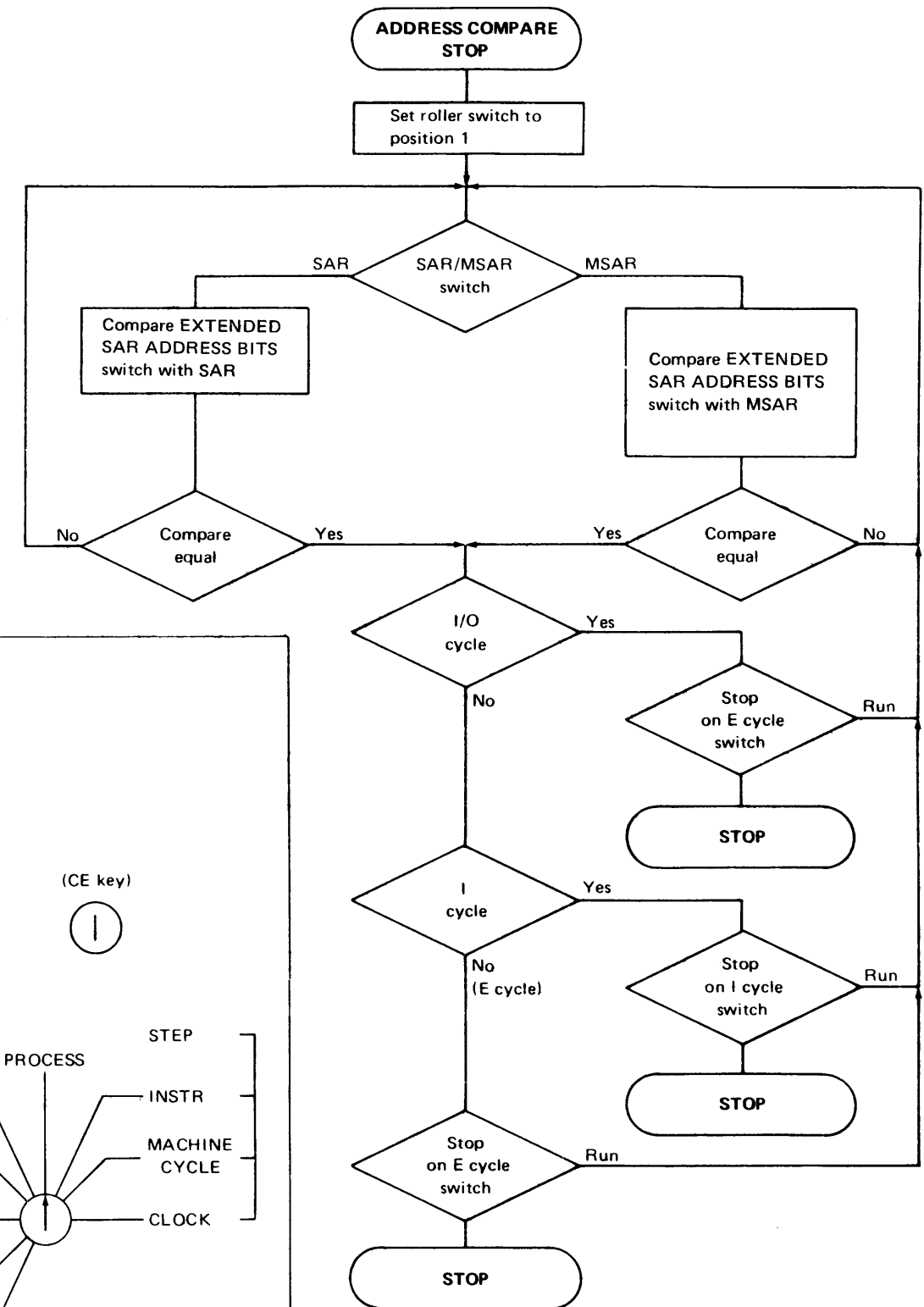
The processor is restarted by pressing START.

Address compare circuits are shown on page 2-5.

*Note:* To prevent stopping on an address match (used as a sync point), tie up 01A-B3R2M02 to 01A-B3U5B02. Sync point is 01A-B3R2U12 (ALD KA222).



CE Panel



This page intentionally left blank

# POWER SUPPLY

## INTRODUCTION

### AC/DC Voltage

#### AC POWER

**A** The 5415 CPU supplies primary ac input power to the following devices:

- 1442 Card Read/Punch
- 2501 Card Reader
- 2560 MFCM (50 Hz – 220 Vac and above)
- 3277 Display
- 3284 Printer
- 5415 Power Supplies
- 5424 MFCU

**B** The 5421 supplies primary ac power to the 1403. It also supplies primary ac power for the 2560 when the input power is 60 hertz or 200 Vac 50 hertz.

#### DC POWER

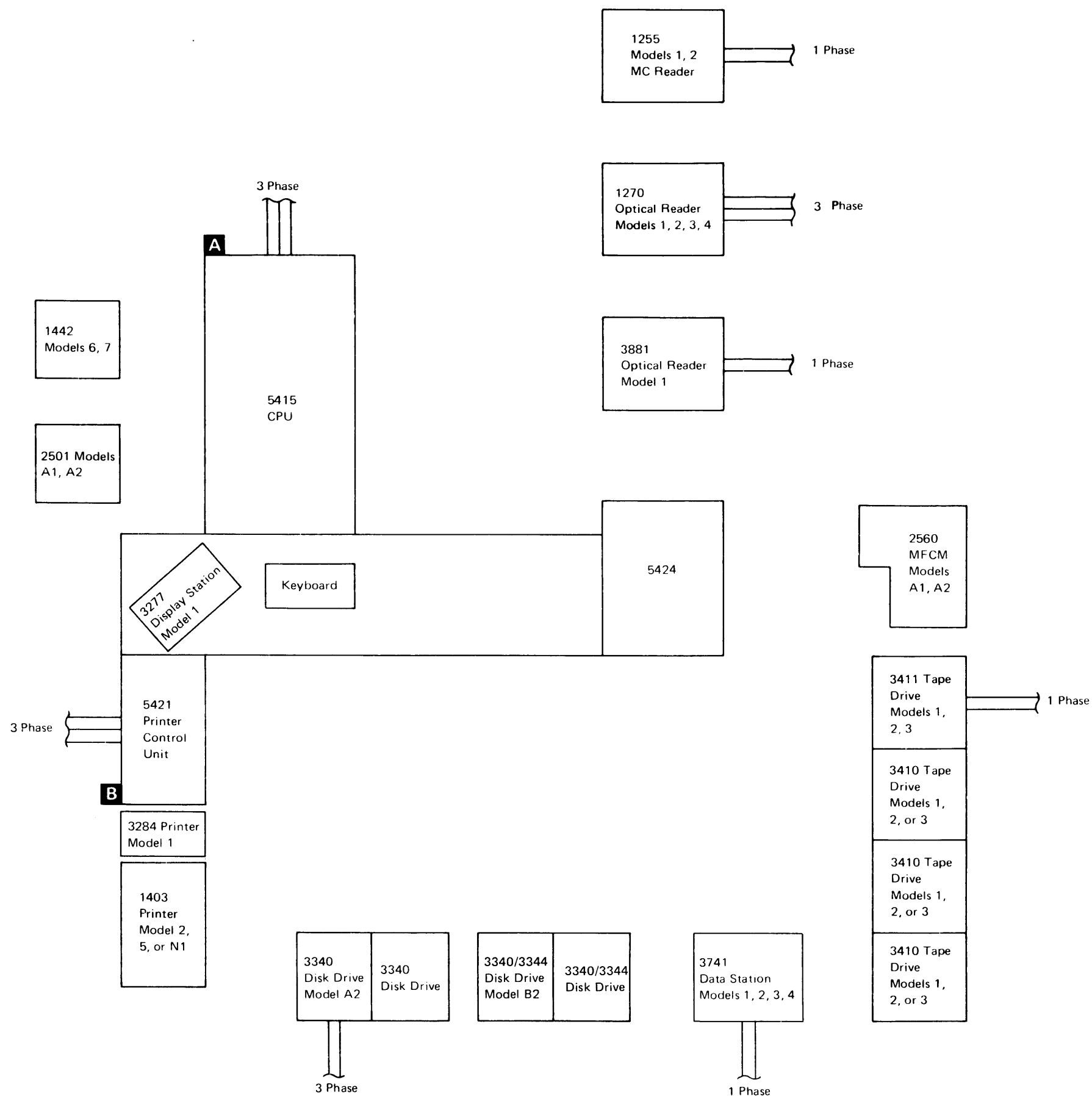
**A** The following dc voltages are used by the 5415 CPU:

- Voltages (dc) developed within the CPU:

- 4 Vdc (A gate basic)
- 4 Vdc (B gate basic)
- 4 Vdc (B gate feature)
- +6 Vdc (basic)
- +8.5 Vdc and +3.4 Vdc (FET storage)
- +24 Vdc (EPO and sequencing)
- +24 Vdc (3340 without 5424)
- 12 Vdc (BSCA or BSCC feature)
- ±12 Vdc (MLTA and BSCA features)
- ±3 Vdc (1442, 2501, 2560 features)
- +5 Vdc (3277 Attachment)
- +5 Vdc (BSCC feature)
- 5 Vdc (BSCC feature with DDSA or 38LS option)

- Voltages (dc) supplied by I/O devices for use by the CPU:

*Note:* See manual of individual I/O device for dc voltage requirements.



## INTRODUCTION

### Basic Unit

The primary power input (ac) is distributed to bulk supplies located in the CPU. The bulk supplies supply unregulated filtered dc to the regulator assemblies. The regulators provide the voltage regulation required to operate the system logic. The regulated dc output is distributed to gates A and B and to the appropriate I/O devices.

### Input Power Requirements

The input power requirements for System/3 are three-phase power at 30A. Domestic and World Trade input voltage requirements are:

- 60 Hertz: 200 Vac, 208 Vac, and 230 Vac ( $\pm 10\%$ )
- 50 Hertz: 200 Vac, 220 Vac, 235 Vac, 380 Vac, and 408 Vac ( $\pm 10\%$ )

### Parts Replacement

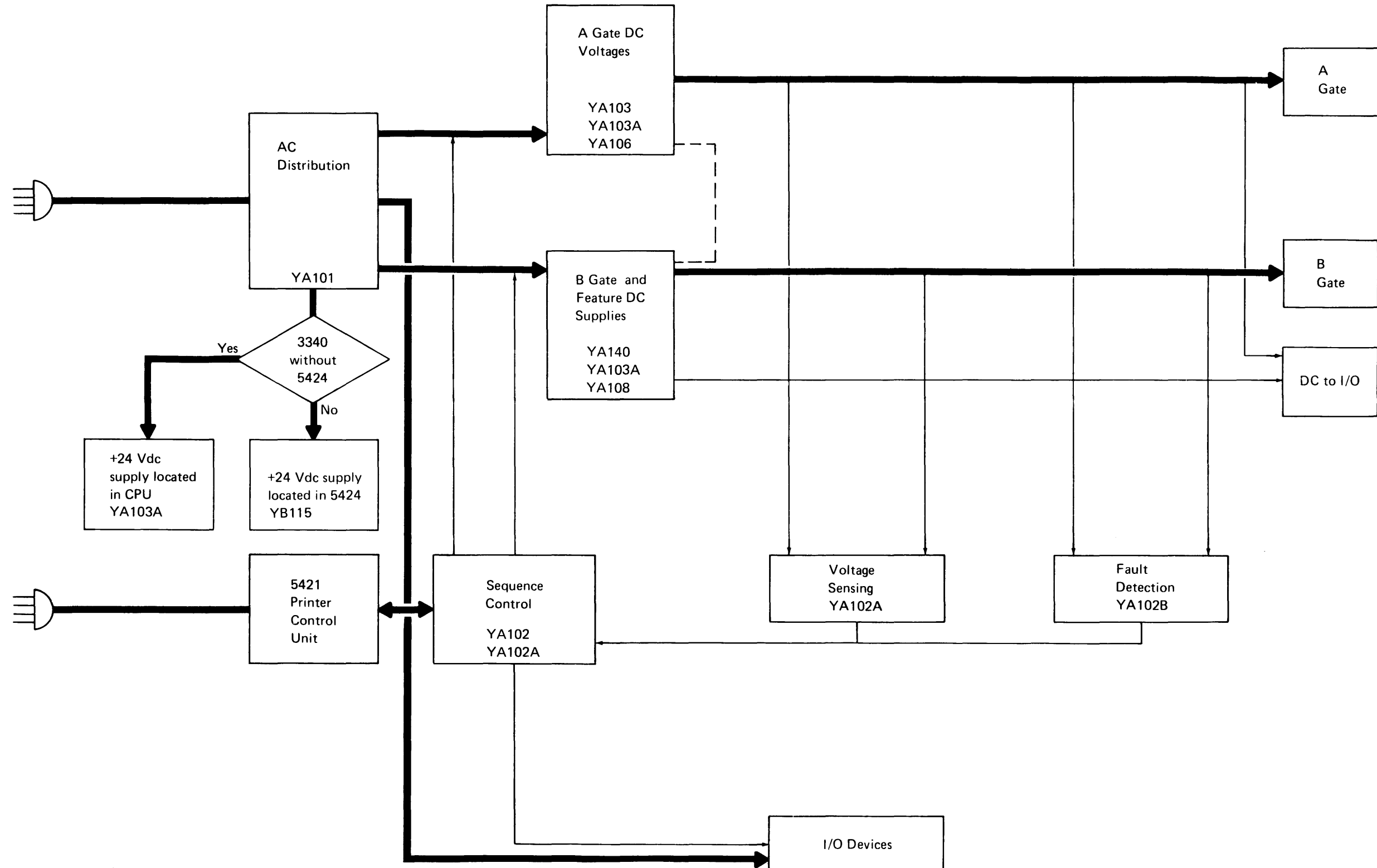
The power system is designed for replacement of power supply subassemblies rather than discrete components. The exceptions include fuses, voltage regulator cards, and relays. However, in large assemblies like the primary control box or bulk supply No. 1, it will be necessary to replace components (filter capacitors, etc.).

### Checks and Adjustments

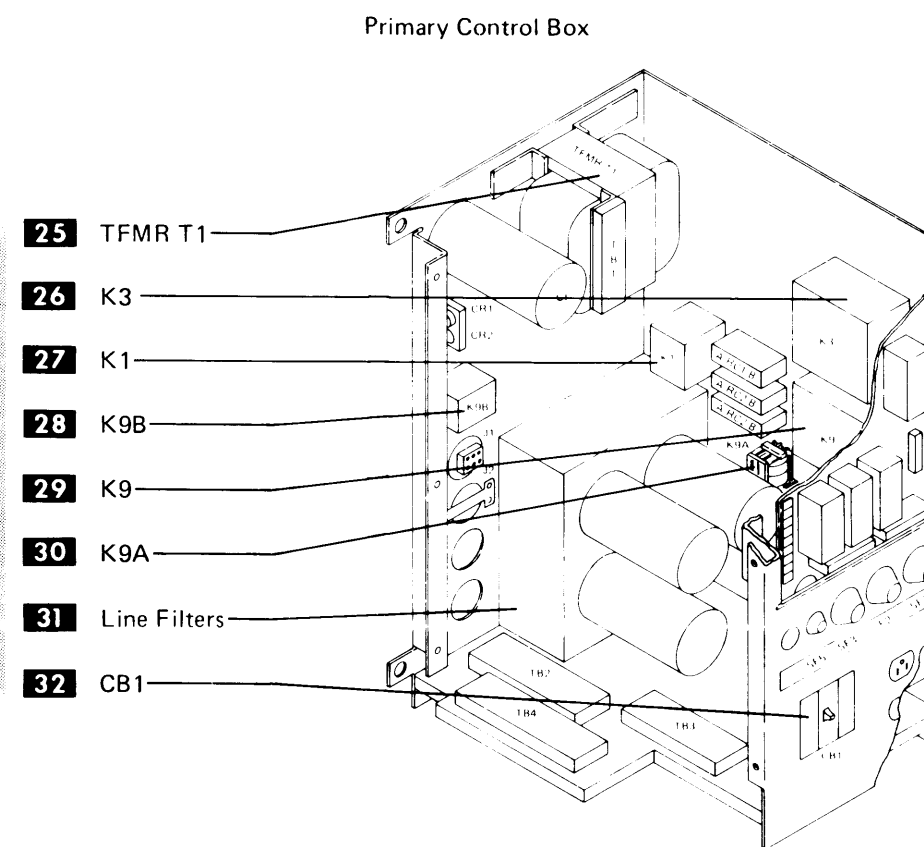
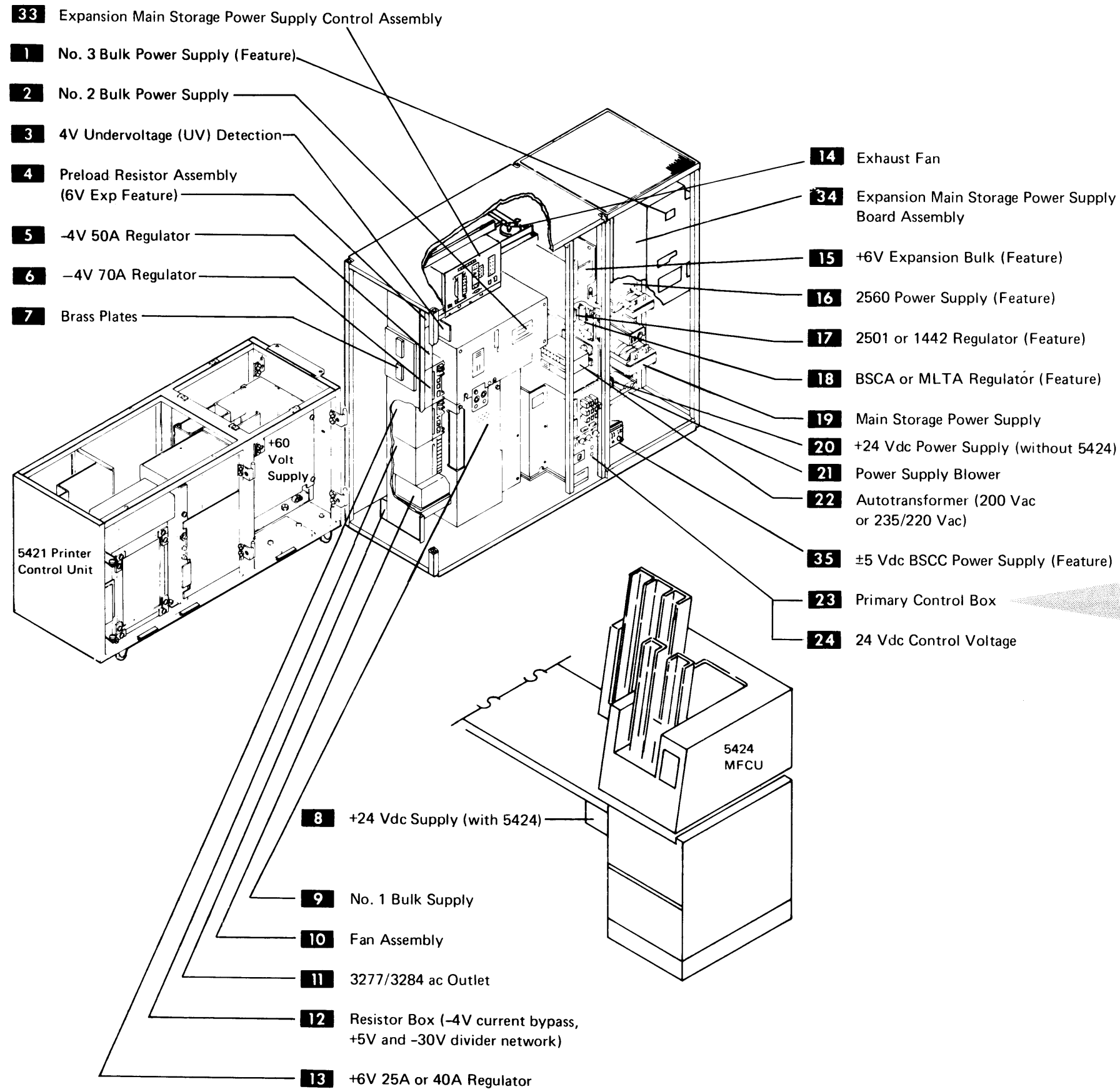
**DANGER:** After the emergency power switch is opened, power is available at K1, K3, and K9B input terminals and at transformer (T1) terminals.

All voltage measurements should be made in a normal environment (temperature between 68 degrees and 86 degrees F) with a *recently calibrated* Weston\* 901 meter or its equivalent.

\*Trademark of Weston, Inc.



**POWER SUPPLIES AND COOLING**

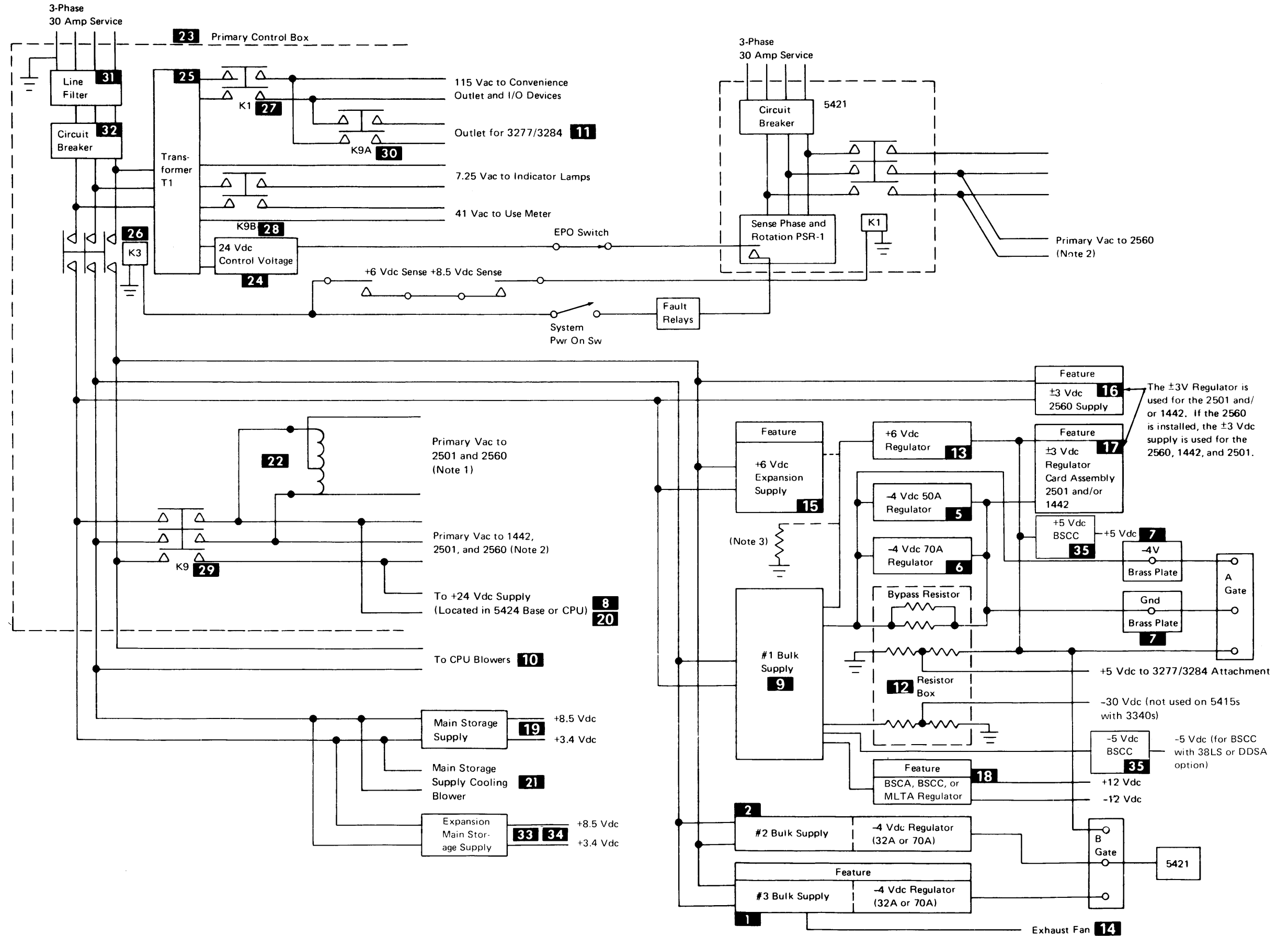




# POWER SUPPLIES AND COOLING

## Notes:

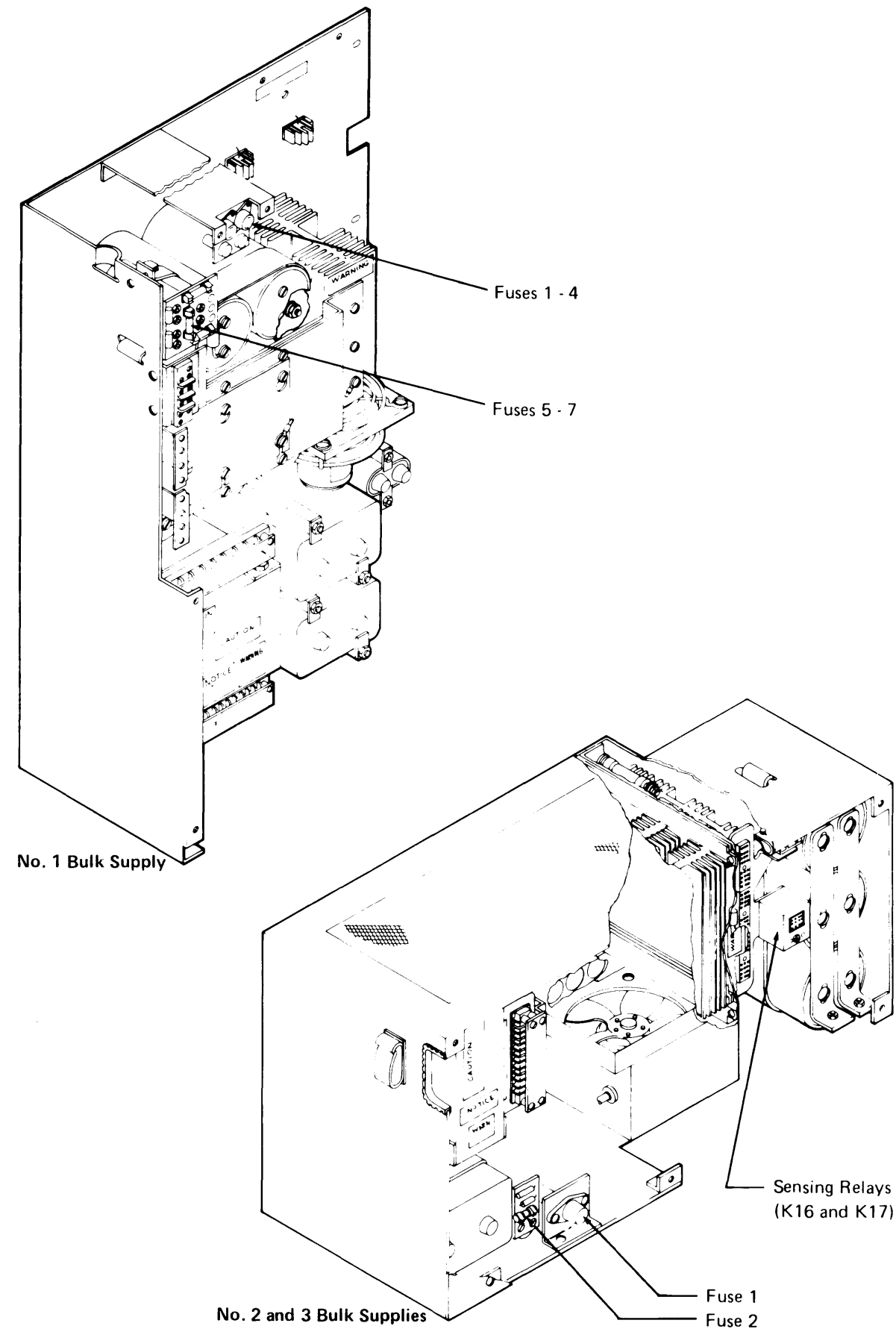
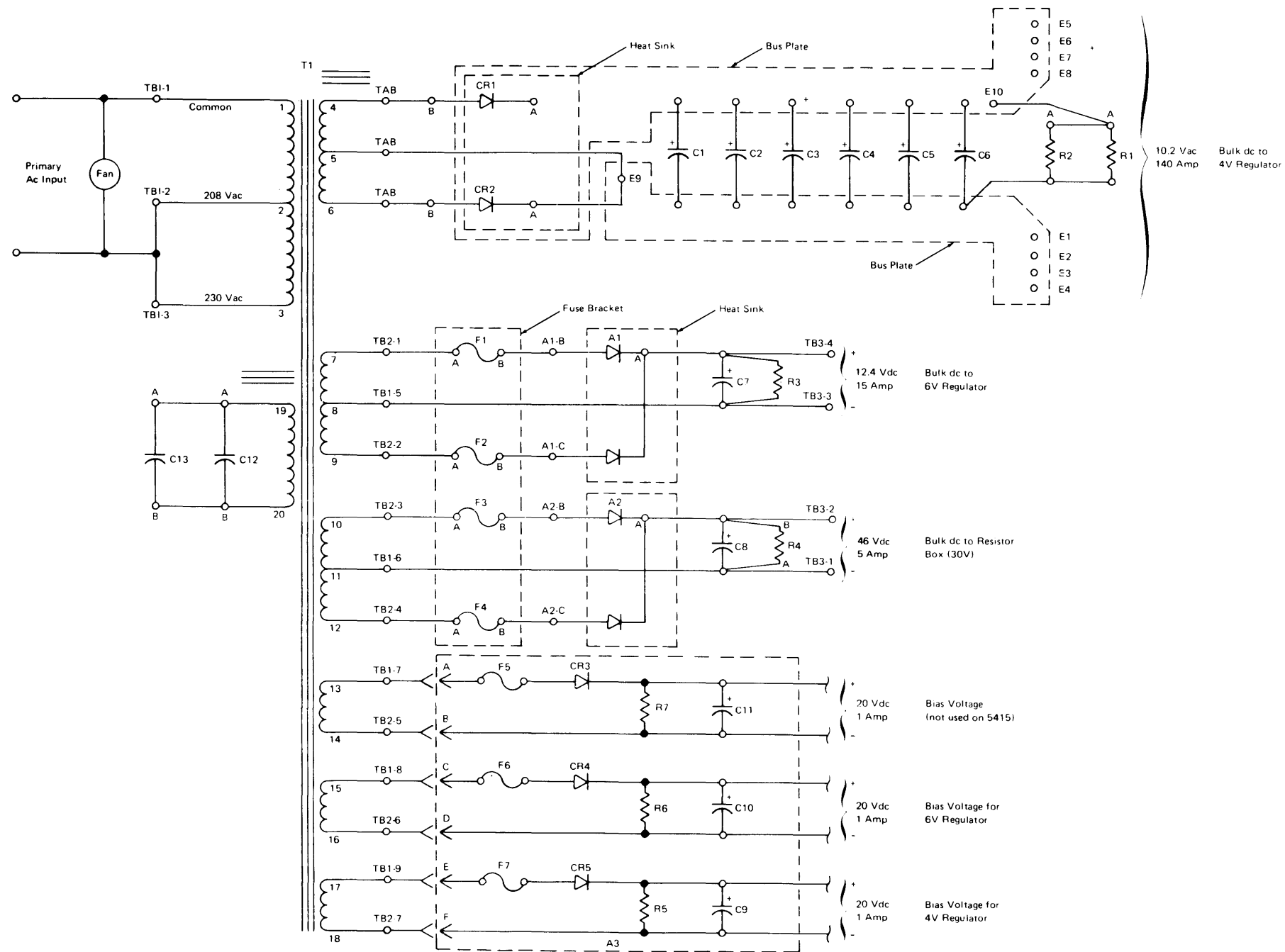
1. If the primary power input is 235 Vac or 408 Vac, an autotransformer is required to supply power to the 2560 or 2501 (if these features are installed).
2. Primary power for the 2560 is supplied by the 5421 on all 60 Hz and 200 Vac 50 Hz systems.
3. Resistor assembly is used to load the No. 1 bulk supply when the 6 Vdc expansion feature is installed. Power to the 6V regulator is supplied by the 6 Vdc expansion supply.



### DC Bulk Supplies

Each bulk power supply contains a ferro-resonant transformer with multiple secondary windings. The transformer outputs are rectified, filtered, and made available to dc regulators for additional regulation. The No. 1 bulk supply is shown in the example. The No. 2 and No. 3 bulk supplies are similar but contain dc bulk and bias voltages for the 4V regulators only.

**Note:** All outputs (dc bulk and bias voltages) are floating. The outputs are referenced to ground via external wiring.



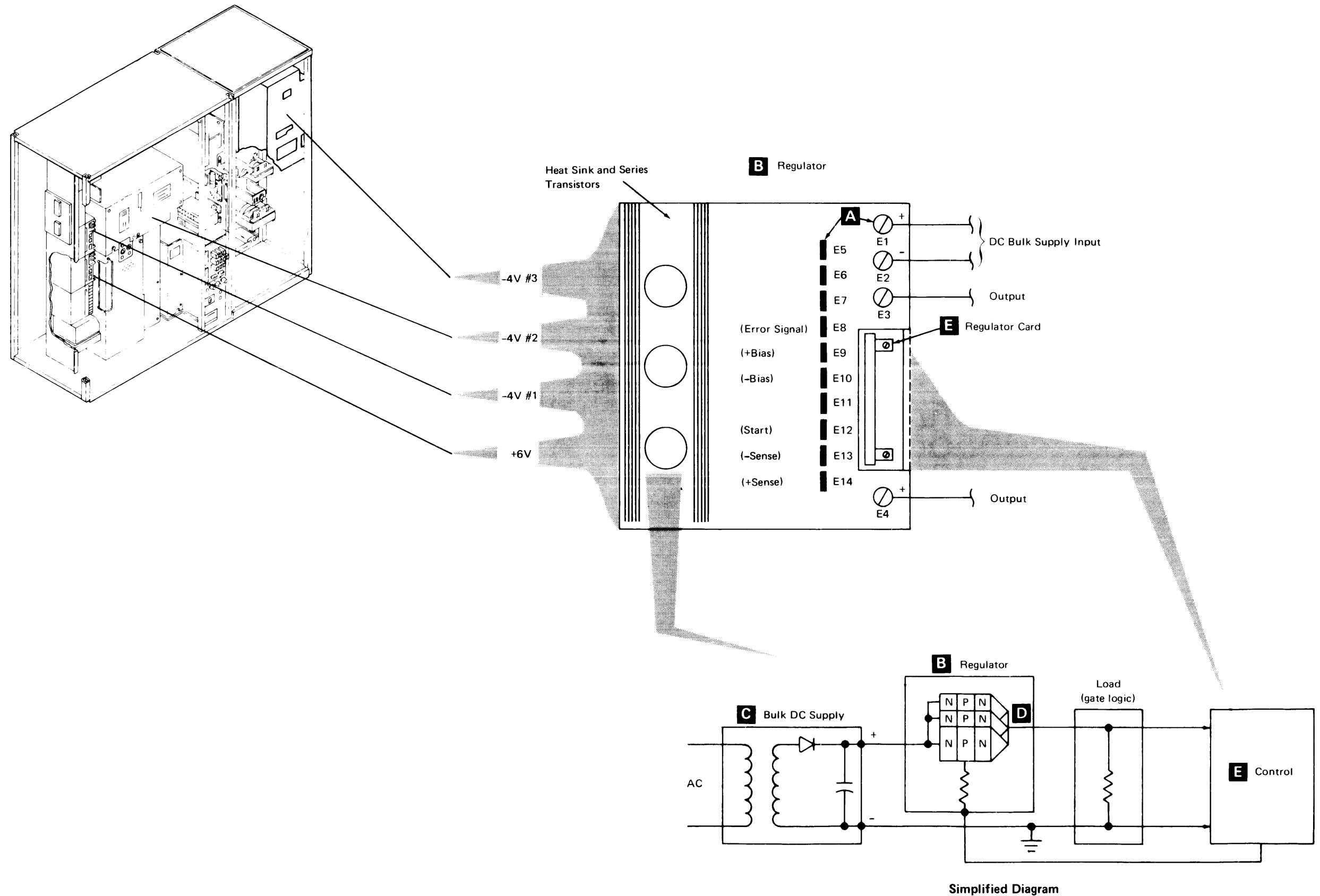
## POWER SUPPLY REGULATORS

The -4V and +6V regulators have identical terminals (E1 - E14) **A** that perform identical functions,

The individual regulators **B** receive the unregulated, filtered dc voltage from the bulk power supplies **C** and provide the voltage regulation required to operate the system logic.

The dc regulators function as a variable resistance **D** in series with the load.

The control circuit regulator card **E** continually monitors the voltage across the load and adjusts the current flow through the series transistors in the regulator to maintain a constant voltage output.



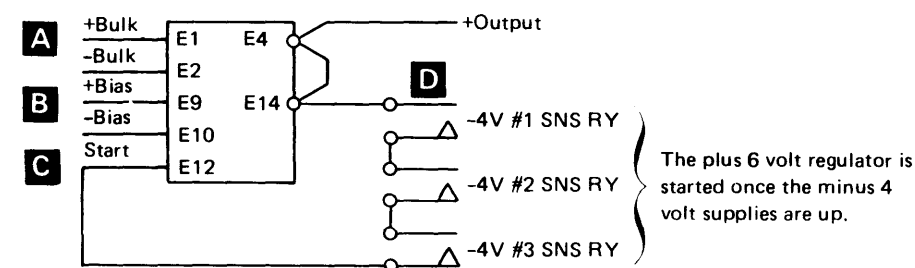
Simplified Diagram

## POWER SUPPLY REGULATORS

### Start-Up Control

A regulator requires three inputs to enable a regulated output voltage:

1. Bulk supply voltage **A** (E1 and E2) provides the desired regulator input voltage. This voltage is reduced to the desired level by the series transistors in the regulator.
2. Bias voltage **B** (E9 and E10) provides operating and reference voltages for the regulator.
3. Start signal **C** (E12) enables the regulator to turn on. This input is connected through relay sense logic **D** to the most positive output of the supply.

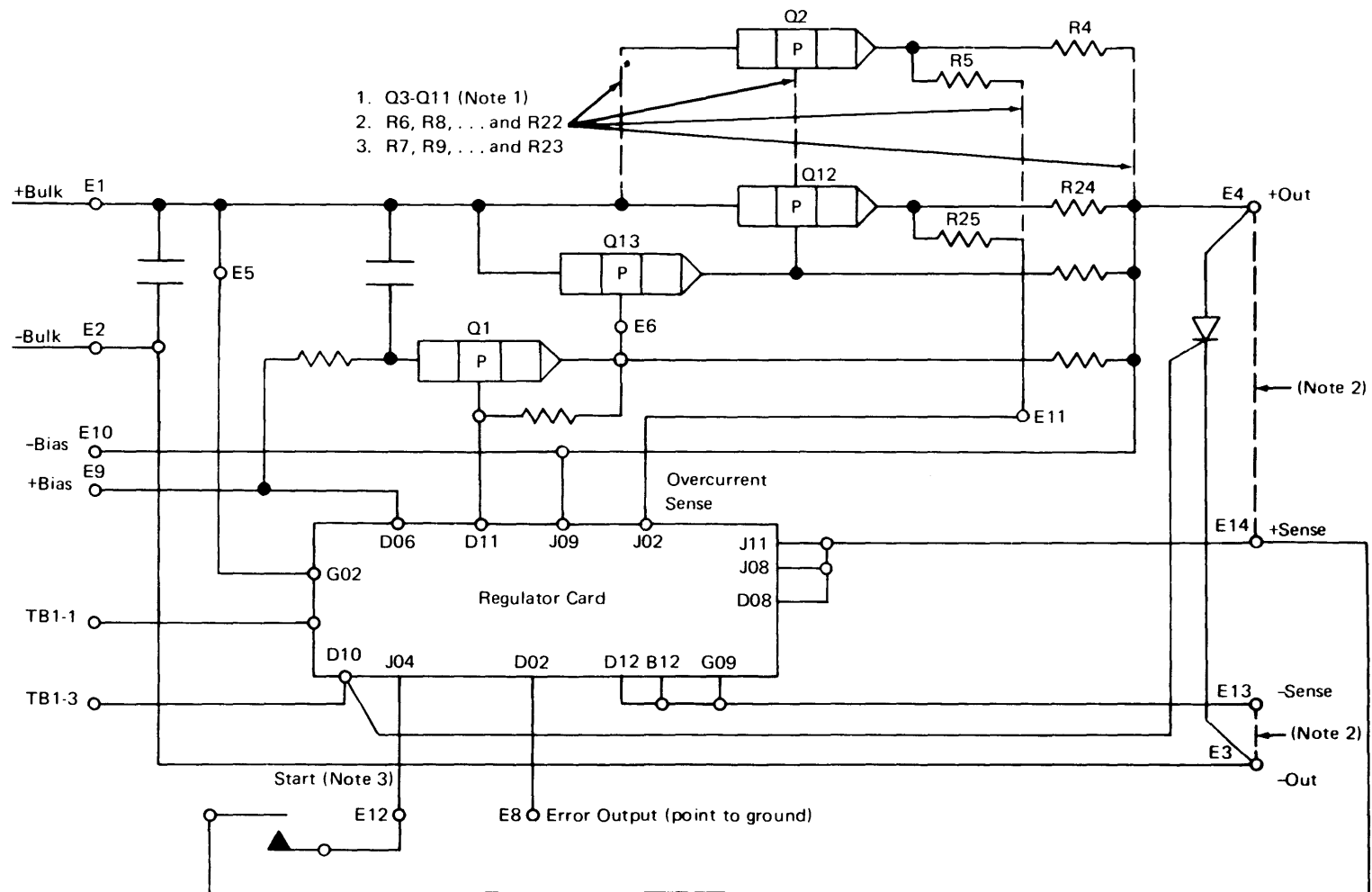


Simplified Diagram (+6V regulator is shown)

### Voltage Regulation

Regulator output terminals E3 and E4 are connected to terminals E13 and E14 (sense - and +) of the regulator card. A differential amplifier on the regulator card compares the output voltage at terminal E13 with an internal reference voltage. The output of the differential amplifier is applied to the base of transistors Q1 and Q13 from terminal D11.

If an increase in output voltage is sensed, a negative voltage is applied to the base of transistors Q1 and Q13 causing the current to decrease through these transistors. A decrease in current results in a decrease in current through transistors Q2 through Q12, thereby causing a decrease in output voltage. For a decrease in output voltage, the current through transistors Q1 and Q13 increases causing an increase in current through transistors Q2 through Q12. Therefore, output voltage increases to the normal level.



1. Q3-Q11 (Note 1)
2. R6, R8, ... and R22
3. R7, R9, ... and R23

#### Notes:

1. To increase current output, add additional transistors in parallel.
2. Connected externally to provide feedback reference to regulator card.
3. To start +6 Vdc regulator, pick K5-2, K17-1, and K19-1. To start -4 Vdc regulators (number 1 through number 3) E12 must be at ground.

## POWER SUPPLY REGULATORS

### Overvoltage Protection

An overvoltage protection circuit in the voltage regulator card monitors terminal E13 and E14. If the output voltage rises beyond the maximum normal limits, the regulator card turns off Q1 and Q13. Transistors Q1 and Q13 then turn off transistors Q2 through Q12 reducing the output voltage to zero. This action protects the logic circuits from an overvoltage condition. If an overvoltage condition exists, the overvoltage circuit grounds terminal E8 (point to ground) via terminal D02 to indicate a fault condition. Grounding terminal E8 causes the OV/OC relay to energize. Energizing the OV/OC relay de-energizes contactor K3 which removes power from the primary of the bulk supply.

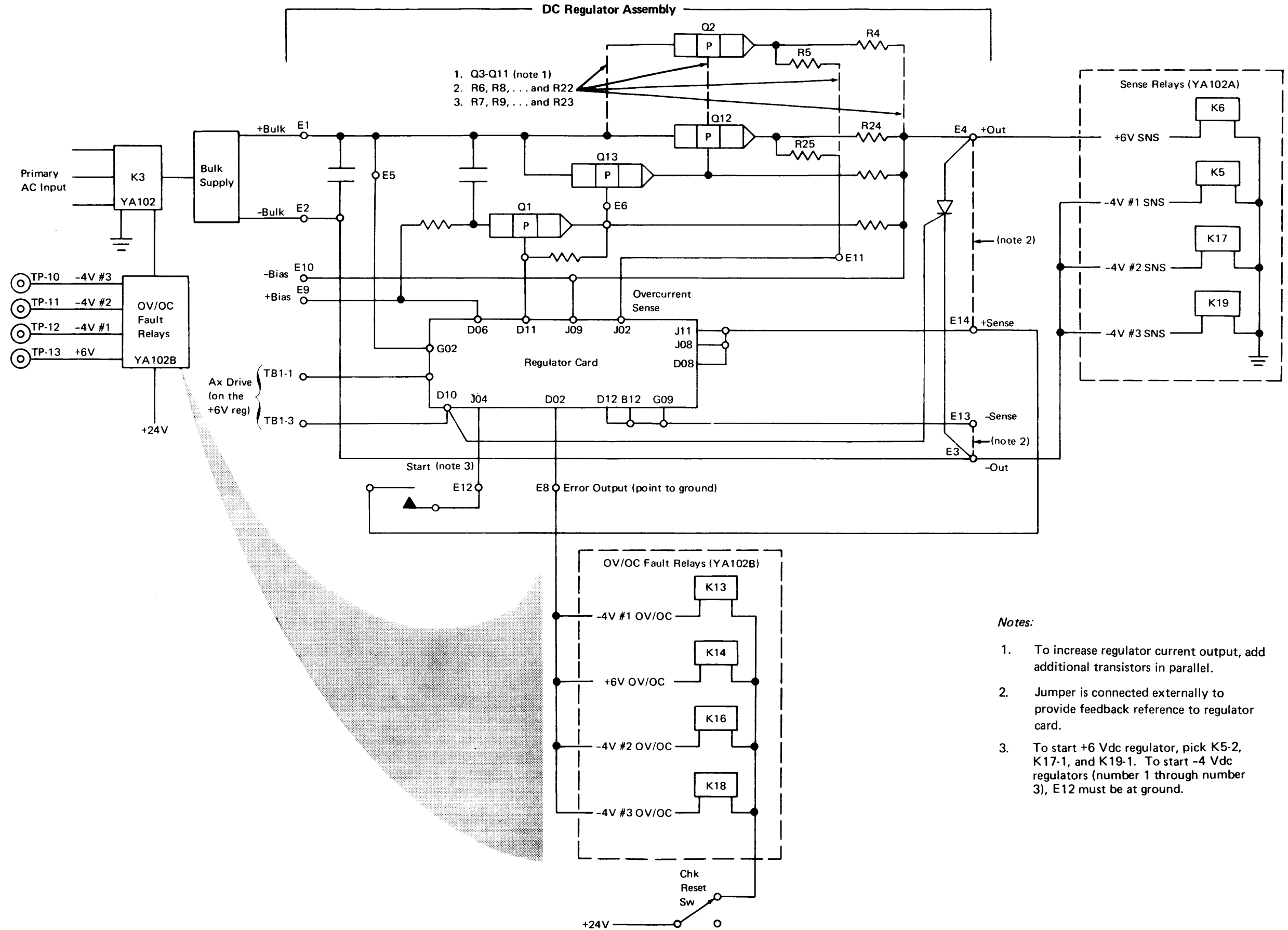
The fault that caused the power down is determined by probing the test points on the power control box (see page 8-28). Fault relays are reset by pressing CHECK RESET if the power switch is off.

### Overcurrent Protection

The overcurrent protection circuit protects the regulator if load current exceeds the limits of the regulator. An overcurrent condition is sensed at terminal J02 of the regulator card.

The voltage drop across R4 through R24 (even-numbered resistors) is proportional to the load current. R5 through R25 (odd-numbered resistors) average these individual voltages. The average voltage feeds a differential amplifier in the regulator card via terminal J02. If the predetermined current limit for the average voltage is exceeded, a fault indication energizes the OV/OC relay by grounding terminal E8. If the OV/OC relays K13, K14, K16, or K18 are energized contactor K3 is de-energized which removes power from the primary of the bulk supply.

The fault that caused the power down is determined by probing the test points on the power control box (see page 8-28). Fault relays are reset by pressing CHECK RESET if the power switch is off.



#### Notes:

1. To increase regulator current output, add additional transistors in parallel.
2. Jumper is connected externally to provide feedback reference to regulator card.
3. To start +6 Vdc regulator, pick K5-2, K17-1, and K19-1. To start -4 Vdc regulators (number 1 through number 3), E12 must be at ground.

## POWER SUPPLY REGULATORS

### Undervoltage Protection

An undervoltage protection circuit for the -4V supply and the +6V supply is necessary because damage to the print magnets can occur if +6V is applied to the 5424 MFCU or the 1403 printer when the -4V regulator output is low or decreases to 0. In order to protect the 5424 MFCU and the 1403 printer, a silicon-controlled rectifier (SCR) is wired across the output of the +6V regulator. The SCR starts to conduct if:

1. The +6V regulator senses an overvoltage or an overcurrent condition.
2. The -4V undervoltage control assembly senses an undervoltage condition.

When the SCR conducts, the +6V output is effectively shorted. This results in an overcurrent condition which, in turn, causes an immediate system power down.

### -4 Volt Undervoltage Circuit (Reg No. 1)

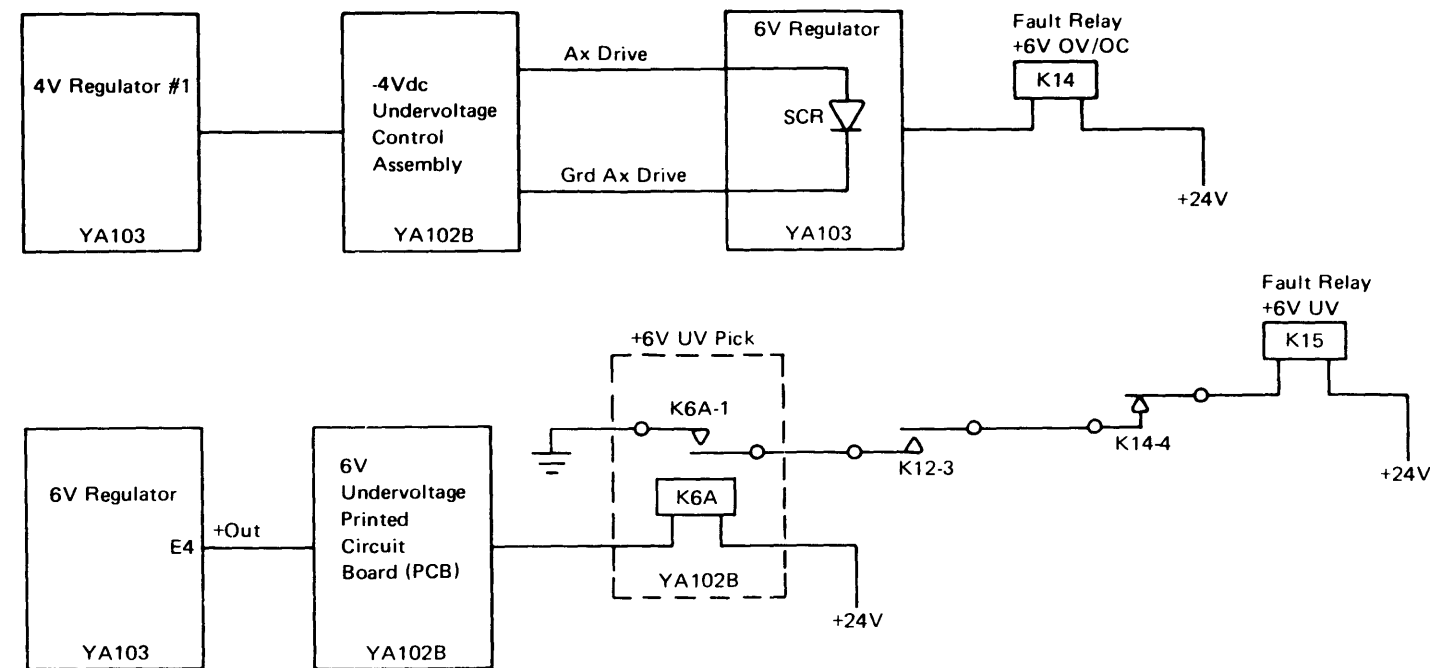
The -4V undervoltage control assembly (Ax Drive) monitors the -4V brass plate. If the -4V output decreases to -3.2 to -3.8 volts an error signal is generated. The error signal fires an SCR in the +6V regulator effectively shorting its output. This shorts out the +6V regulator output causing an overcurrent condition. The +6V regulator senses this simulated overcurrent condition and, as a result, picks K14. This, in turn, de-energizes K3, causing an immediate system power off. The action of shorting (axing) the +6V regulator when a -4V No. 1 undervoltage condition is sensed protects the 5424 and the logic circuits.

### +6 Volt Undervoltage Circuit

The +6V undervoltage circuit monitors the +6V regulator E4 (+ out). If a +6 Vdc undervoltage is sensed, K6A is picked, which picks K15 through the K6A-1 point. This causes the system to power down by dropping relay K9 (ac to I/O).

#### Notes:

1. The -4 Vdc undervoltage control assembly shorts the output of the +6V regulator with an SCR when a -4V undervoltage (-3.2V to -3.8V) condition is detected.
2. Relay K3 (ac to bulk PS) is dropped when the K14-2 n/c points open.
3. Relay K6A is picked if the +6V regulator output drops to +5.7 Vdc or less.



**BASIC -4 Vdc NO. 1 LOGIC SUPPLY  
(Includes Add-On and Bypass Resistor)**

**Basic -4 Vdc No. 1 Supply **A****

The -4 Vdc No. 1 supply provides regulated -4 Vdc for the system logic (A gate, etc.). The -4 Vdc assembly also provides start control, regulation, and ov/oc detection (see page 8-8 for description).

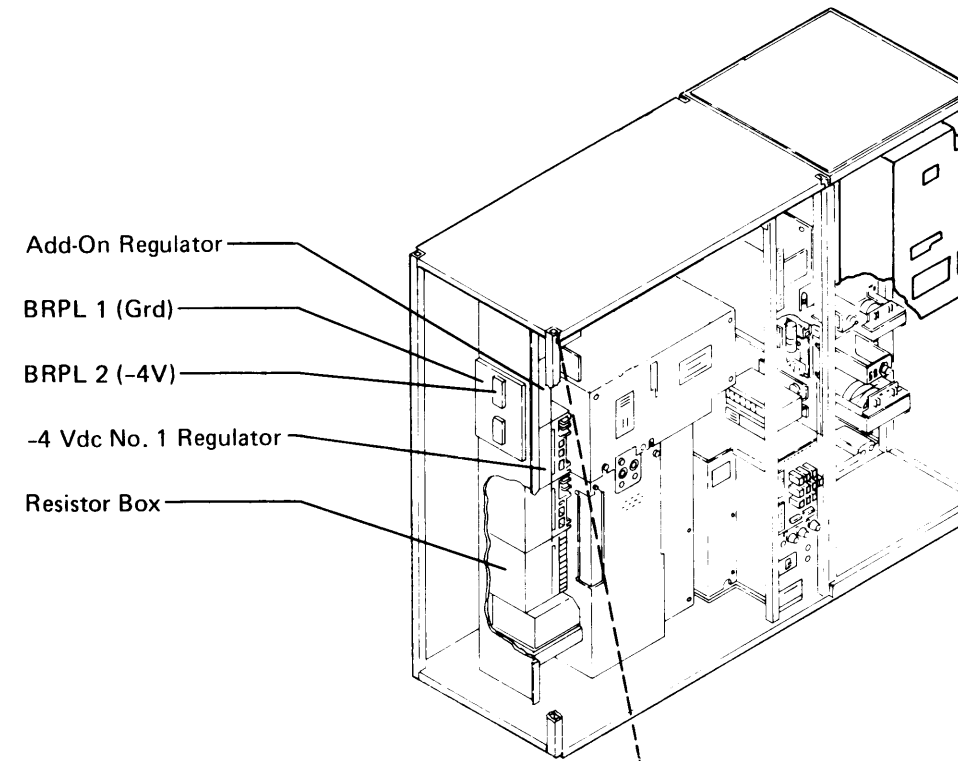
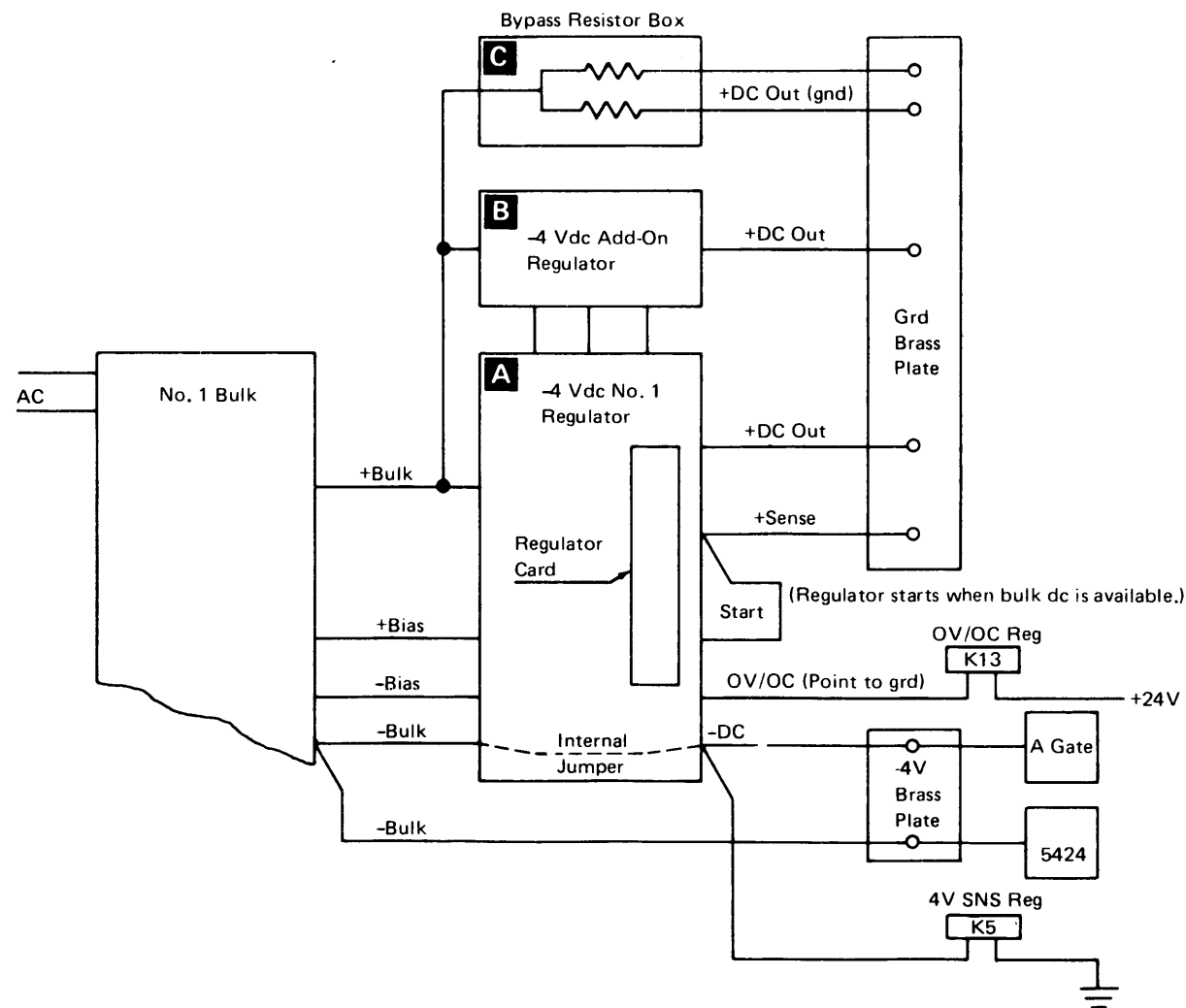
The -4 Vdc add-on and resistor box assembly expand the capacity of the -4 Vdc No. 1 supply.

**Add-On Regulator **B****

The add-on regulator contains series transistors wired in parallel with the basic -4 Vdc No. 1 regulator. These transistors are controlled by the regulator card in the -4V 70 amp regulator for the No. 1 power supply.

**Bypass Resistors **C****

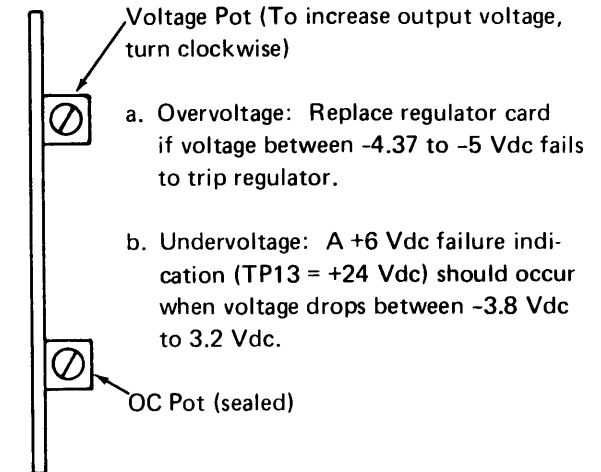
The resistor box assembly contains two 0.32 ohm resistors in parallel that shunt current around the -4V No. 1 and add-on regulators. The maximum current through these resistors is less than minimum required by the CPU, thereby maintaining voltage regulation.



Gate A

**Voltage Adjustment**

1. Connect the meter between brass plate #2 (-4V) and brass plate #1 (ground) behind the CPU console.
2. Set the voltage for -4.20V.



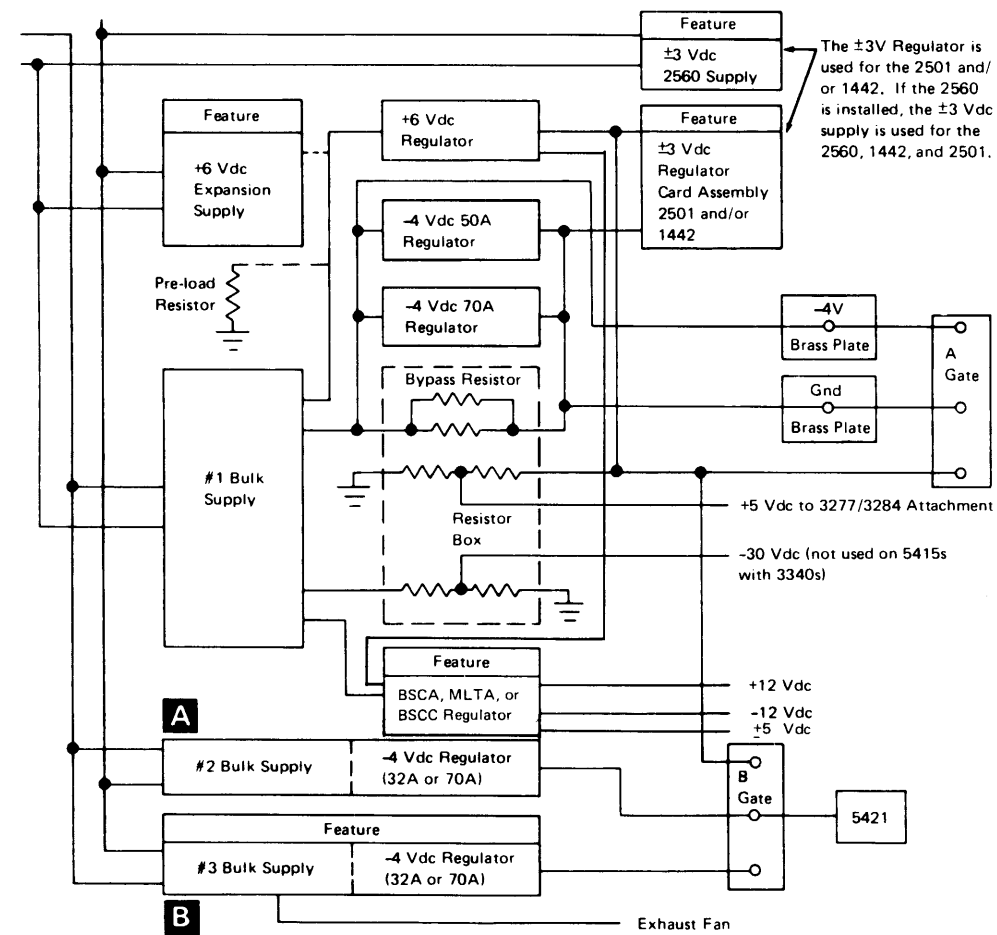
End View of -4V Regulator Card

3. Connect the meter across A-A1C2B06 (-4V) and A-A1C2D08 (ground). This voltage should measure between -3.85V and -4.2V.
4. If the voltage measured in step 3 is out of tolerance, readjust the -4V supply or check distribution cables for high resistance connections.

**B GATE -4V LOGIC SUPPLY NO. 2 (Upper Boards) and B GATE -4V FEATURE LOGIC SUPPLY NO. 3 (Lower Boards)**

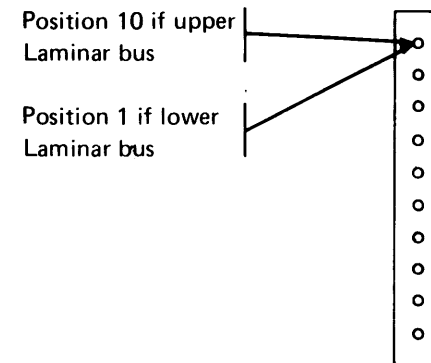
The -4V logic supply No. 2 **A** provides -4 volts for the B gate. As features are installed on the B gate, the -4V feature logic supply No. 3 **B** is added to meet the increased power requirements (lower B gate boards).

Each logic supply (No. 2 and No. 3) has its own bulk transformer, regulator, and cooling system.

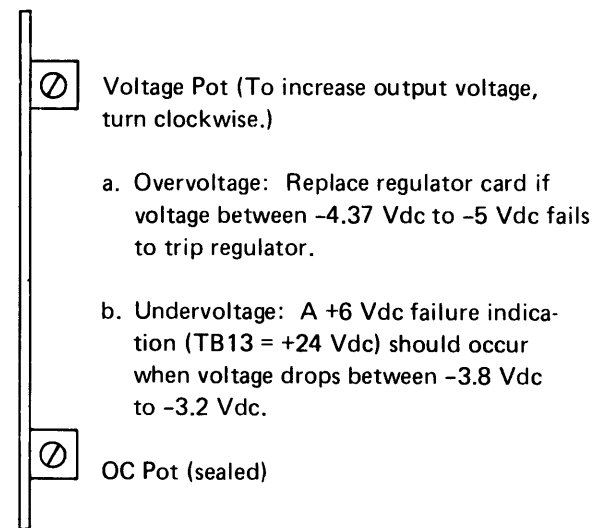


**Voltage Adjustment**

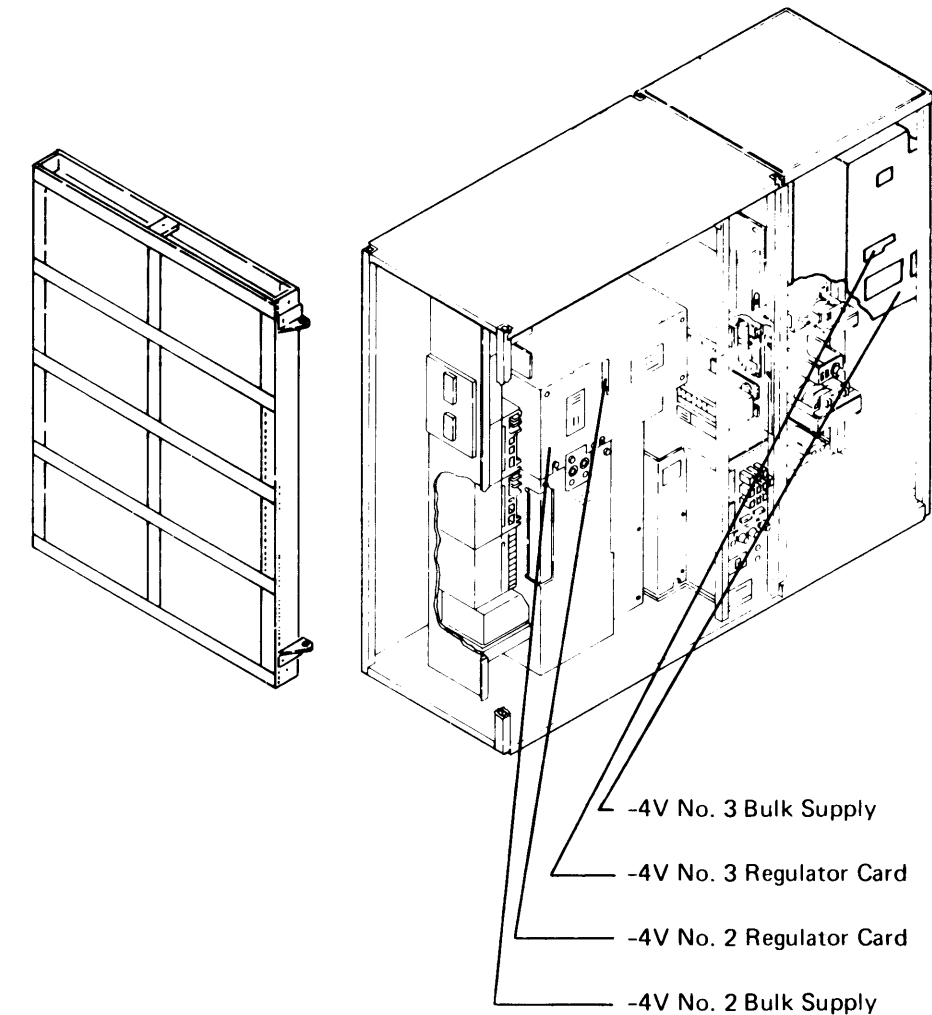
1. Connect the meter across the -4V load between position 6 and ground (position 7) on B gate upper laminar bus if adjusting -4V supply No. 2 or B gate; lower laminar bus if adjusting -4V supply No. 3.



2. Set the voltage adjustment potentiometer to -4.05 Vdc.

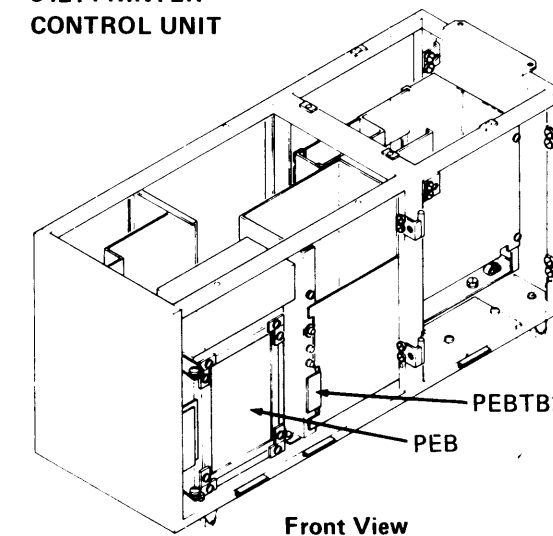


**End View of -4V Regulator Card**



3. Connect the meter across PEBTB 2-7 (ground) and PEBTB 2-8 (-4V) on the printer electronics gate. This voltage should measure between -4.20V and -3.85V.
4. If voltage measured in step 3 is out of tolerance readjust the -4V No. 2 supply or check distribution cables for high resistance connections.

**5421 PRINTER CONTROL UNIT**



**Front View**

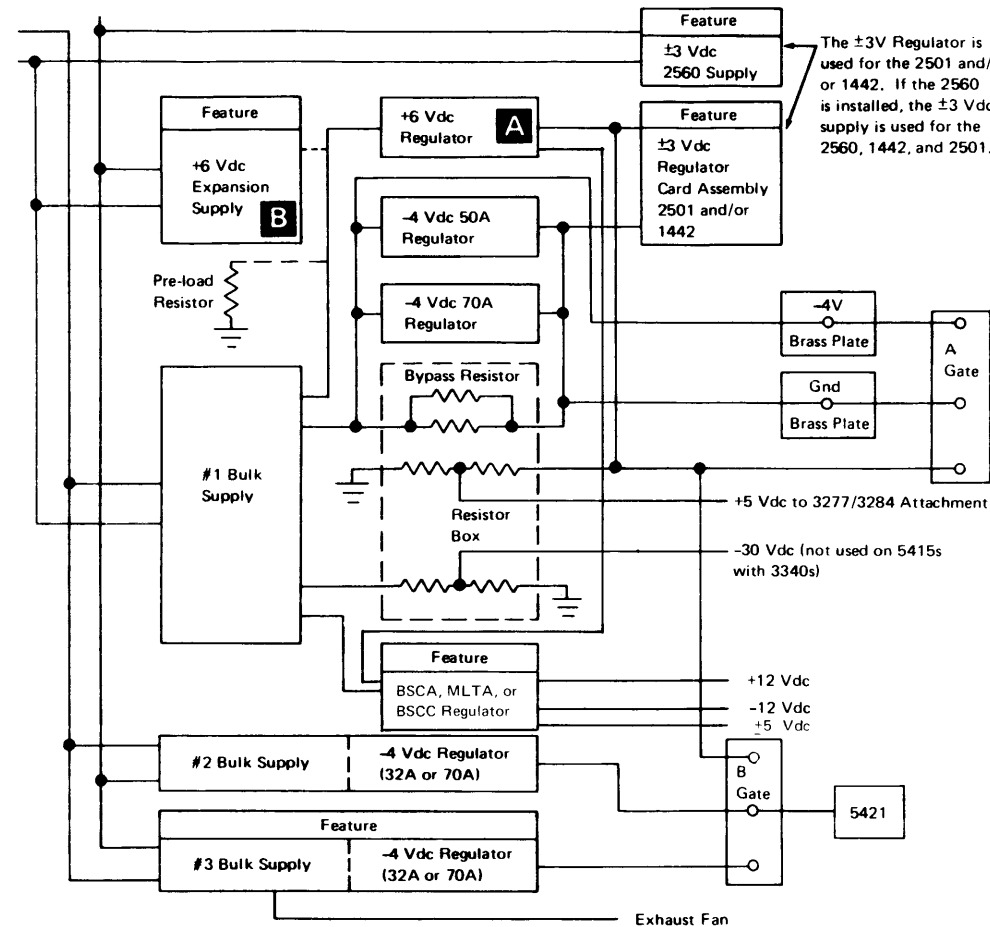


## +6 VOLT LOGIC SUPPLY

### +6 Vdc Regulator **A**

The +6 Vdc regulator provides a regulated 6 Vdc output to operate the system logic (A and B gates). A regulator card in the regulator assembly provides the regulated output and circuit protection.

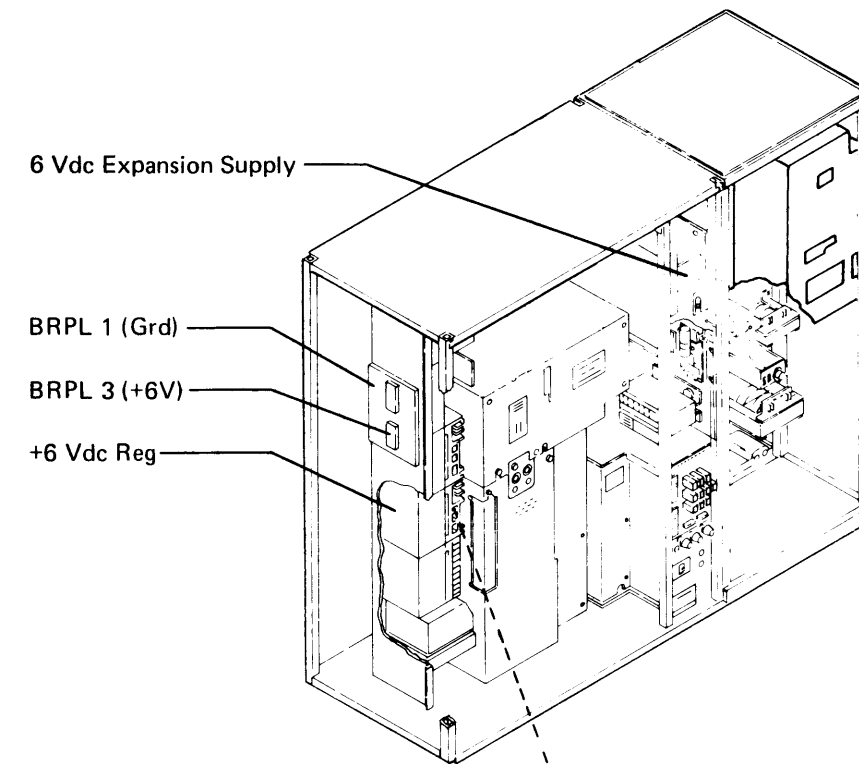
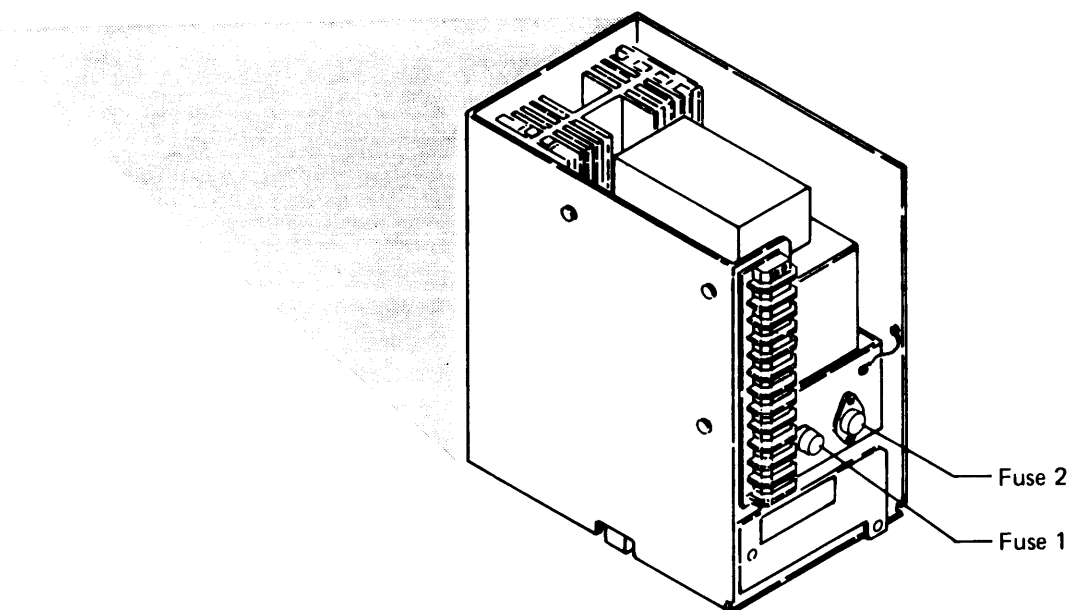
The 6V output is also used by the resistor box to supply (via voltage divider network) +5 Vdc to the 3277/3284 attachment.



### +6V Expansion Bulk Supply (Feature) **B**

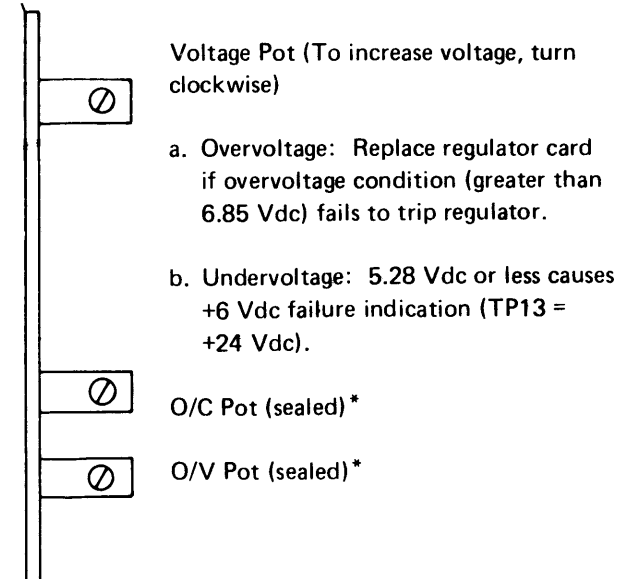
The +6V expansion bulk supply is required when the feature configuration requires more current at +6 Vdc than the No. 1 bulk is capable of supplying.

The installation of the +6V expansion bulk is accomplished by removing the present 6V regulator from the No. 1 bulk output, adding a resistor load (preload) in its place, and installing the 6V expansion bulk (see diagram).



### Voltage Adjustment

1. Connect meter between brass plate #3 (+6 Vdc terminal) and brass plate #1 (ground terminal) behind the CPU console.
2. Set voltage adjustment potentiometer for +6 Vdc.

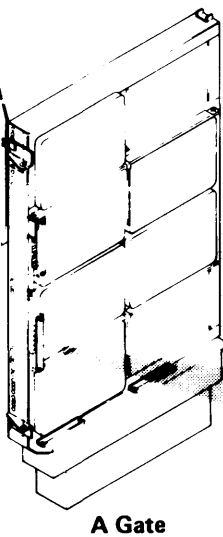
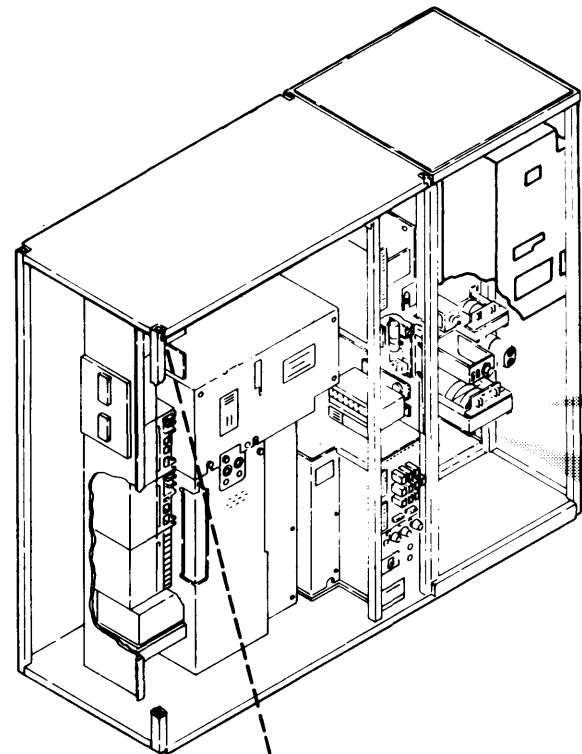


End View of +6V Regulator Card

\*These potentiometers may or may not be on the card.

**MAIN STORAGE POWER SUPPLY**

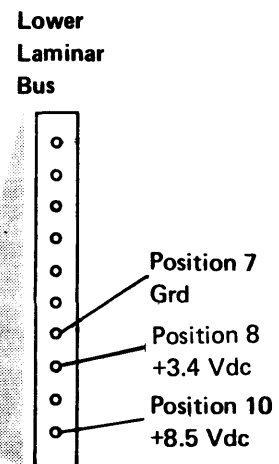
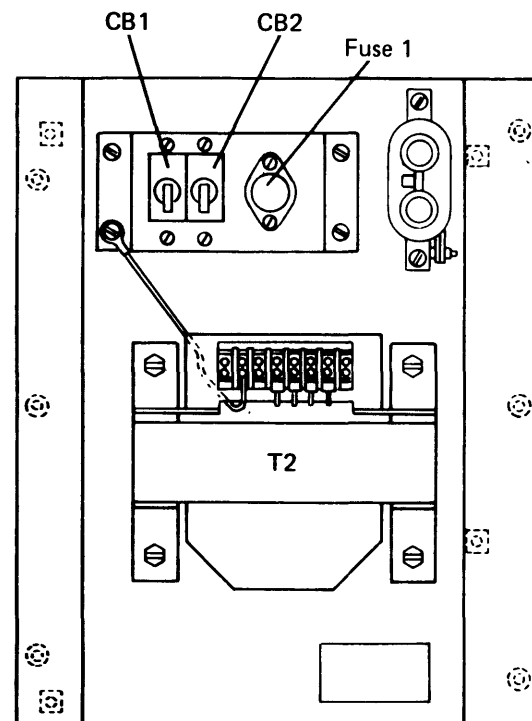
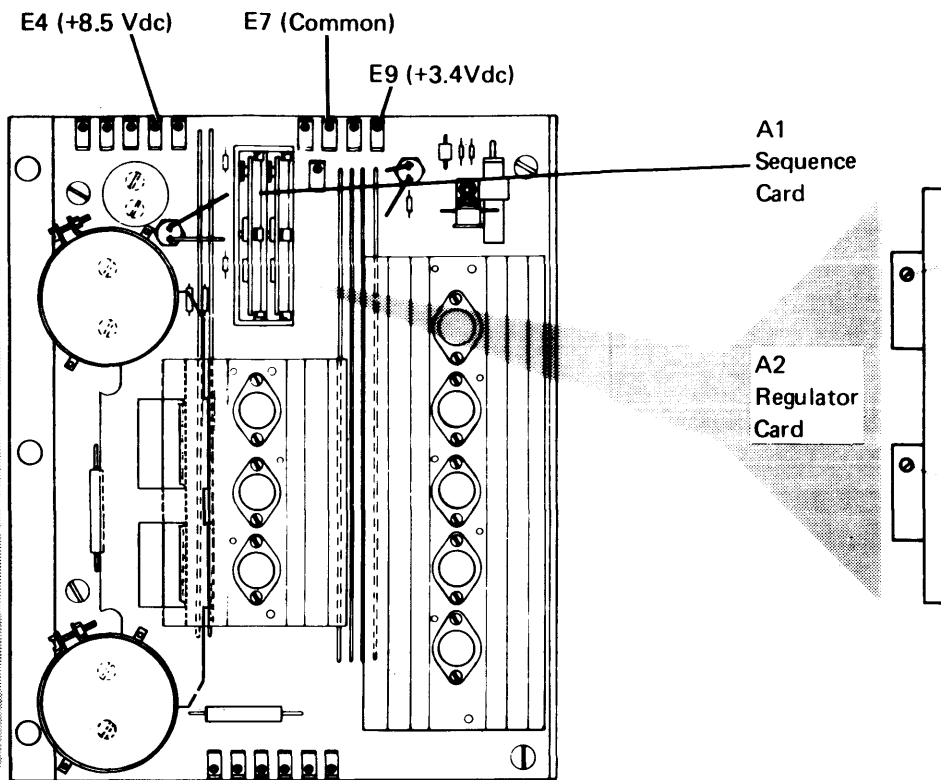
The main storage power supply receives primary ac input power via relay K3 and generates two bulk dc voltages. The bulk voltages are regulated within the supply to +3.4 Vdc and +8.5 Vdc. The +3.4 Vdc and +8.5 Vdc supply provides power for CPU main storage and to B gate for the BSCC control store (FET) cards and 3340/3344 attachment (FET) B4 card.



**A Gate**

**Left Side View**

**Right Side View**



**Voltage Adjustments**

**+3.4 Vdc**

1. Connect meter across E7 (common) and E9 (+3.4 Vdc).
2. Set voltage for 3.45 Vdc.
3. Connect meter between A gate lower laminar bus position 8 (+3.4 Vdc) and position 7 (grd). This voltage should measure +3.4 Vdc. If not, readjust the +3.4 Vdc supply.

**+8.5 Vdc**

1. Connect meter across E7 (common) and E4 (+8.5 Vdc).
2. Set voltage for +8.55 Vdc.
3. Connect meter between A gate lower laminar bus position 10 (+8.5 Vdc) and position 7 (grd). This voltage should measure +8.5 Vdc. If not, readjust +8.5 Vdc supply.

**MAIN STORAGE POWER SUPPLY (Part 2 of 2)** The following inputs are required to sequence up the main storage power supply:

**Sequencing**

1. AC Power – AC to bulks via K3.
2. +24V Control Voltage (E2) – Powers the sequence card.
3. +24V Start/Stop (E3) – Signals the sequence card to begin monitoring the -4V sense line.
4. -4V Sense (E1) – The supply is turned on when this line reaches approximately -3 Vdc (monitored at the brass plate). The No. 2 -4 Vdc power supply, the No. 3 -4 Vdc

power supply (if channel bank No. 3 FET protect is installed), and expansion main storage power supply (if expansion main storage is installed) are all powered up.

**Overvoltage (OV)/Undervoltage (UV) Protection**

**Sequence Card (A1)**

The sequence card provides sequence control for the +3.4 Vdc and +8.5 Vdc regulators. The +3.4 Vdc supply is sequenced up first on power-up and down last on power-down.

**Regulator Card (A2)**

The regulator card provides voltage regulation for the two series regulators of which the outputs are +3.4 Vdc and +8.5 Vdc. The regulator card contains two potentiometers, one for adjusting the +3.4 Vdc level and one for adjusting the +8.5 Vdc level.

**+3.4 Vdc**

A +3.4 Vdc OV (+4.2 Vdc) or UV (+2.5 Vdc) is sensed by the sequence card, which provides gate signals to bias on both Q1 and Q2 (SCRs). The SCRs cause short circuits across the +3.4 Vdc and 8.5 Vdc supplies, tripping CB1 and CB2. At the same time, the sequence card turns off both regulators.

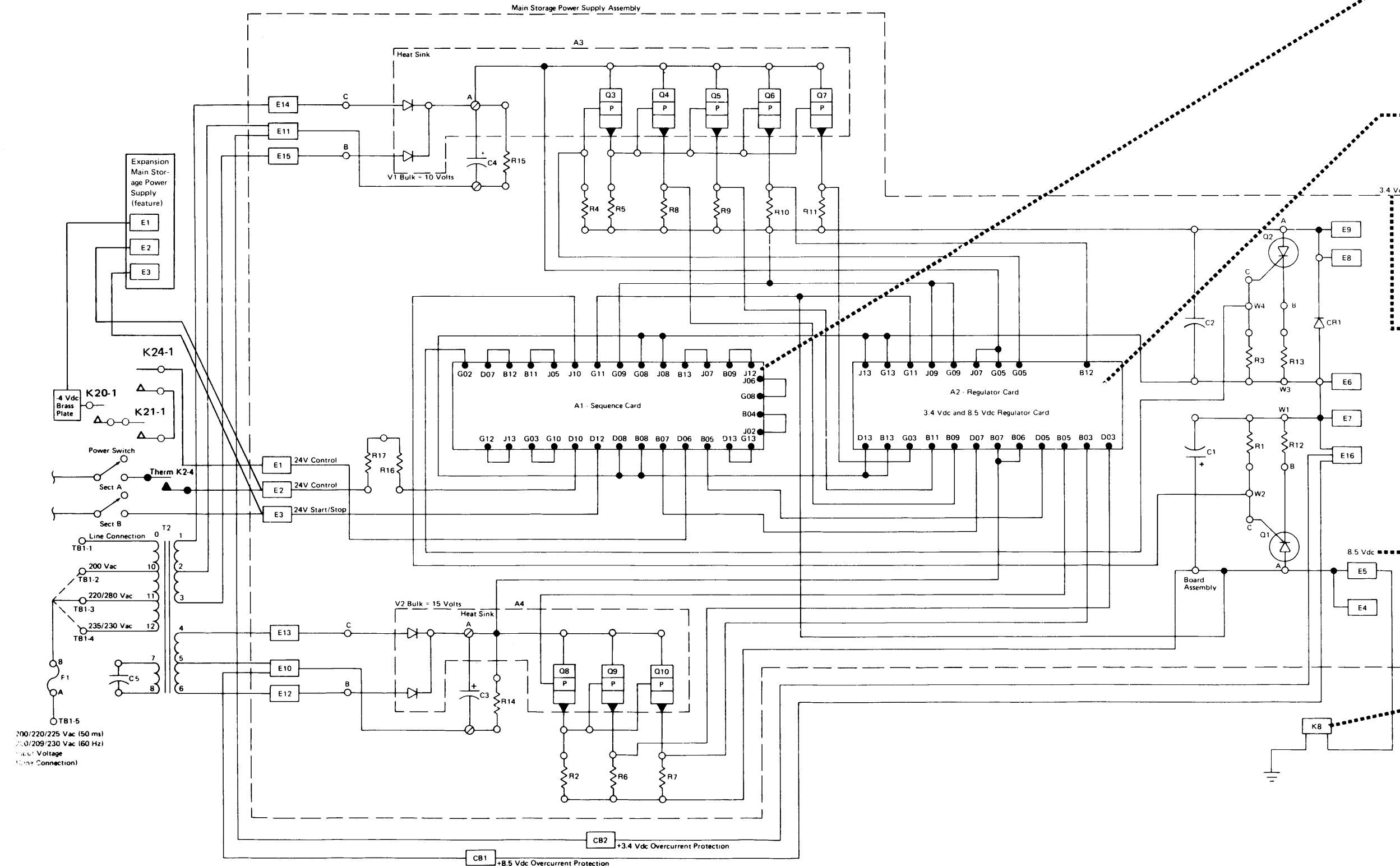
*Note:* CB1 and CB2 may or may not trip depending on the cause of the overvoltage, but both regulators stay off until system power is recycled on.

**+8.5 Vdc**

A +8.5 Vdc OV (+9.5 Vdc) or UV (+2.5 Vdc) is sensed by the sequence card, which provides a gate signal to bias on Q1 (SCR). Q1 causes a short circuit across the +8.5 Vdc. This trips CB1 which disconnects the bulk voltage to the +8.5 Vdc regulator.

**Sense Relay**

The sense relay (K8) is energized when all the +3.4 Vdc and +8.5 Vdc supplies, fed by the main storage power supply, are up.

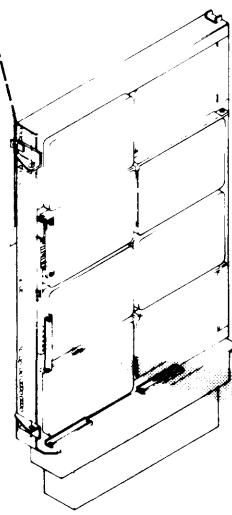
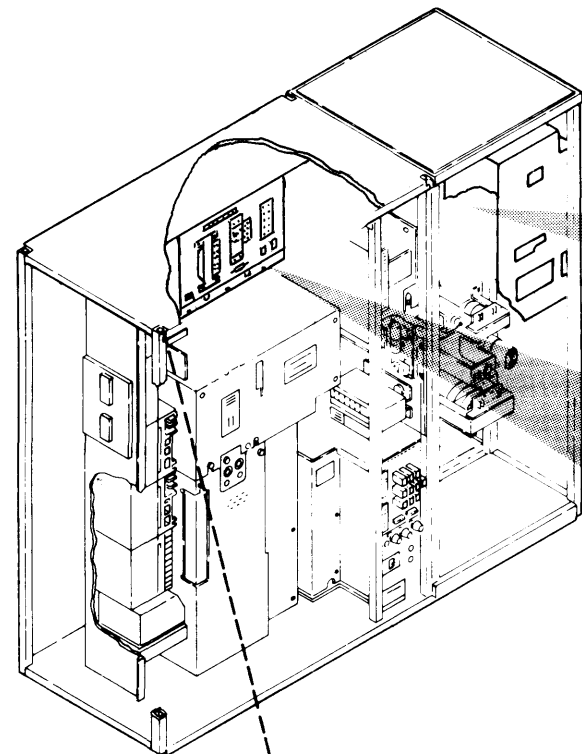


200/220/225 Vac (50 ms)  
200/209/230 Vac (60 Hz)  
Input Voltage  
(Line Connection)

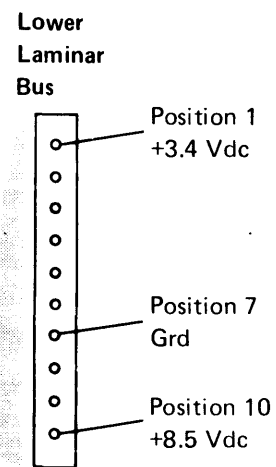
### EXPANSION MAIN STORAGE POWER SUPPLY (Part 1 of 2)

The expansion main storage power supply is electrically identical to the main storage power supply but is physically different due to packaging. It is installed when greater than 256K memory is being installed. When installed, this power supply supplies the +8.5 Vdc and the +3.4 Vdc power to the 01A-A4 board position for expansion main storage.

The expansion main storage power supply receives primary ac input power via relay K3 and generates two bulk dc voltages. The bulk voltages are regulated within the supply to +3.4 Vdc and +8.5 Vdc.

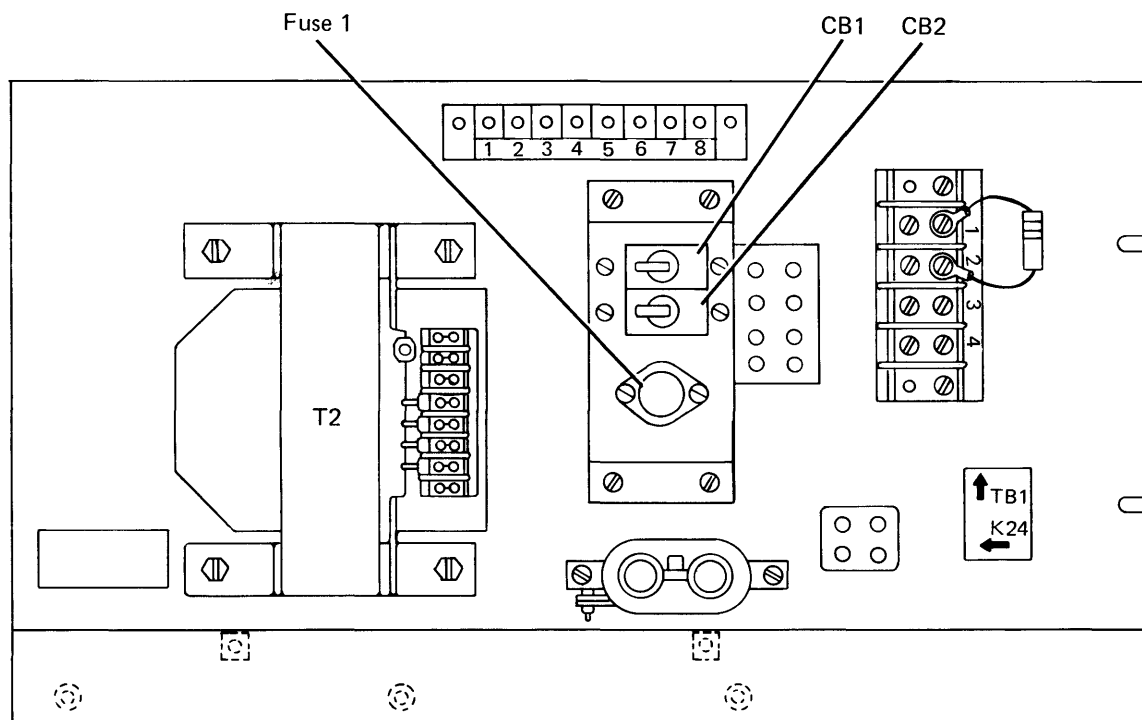
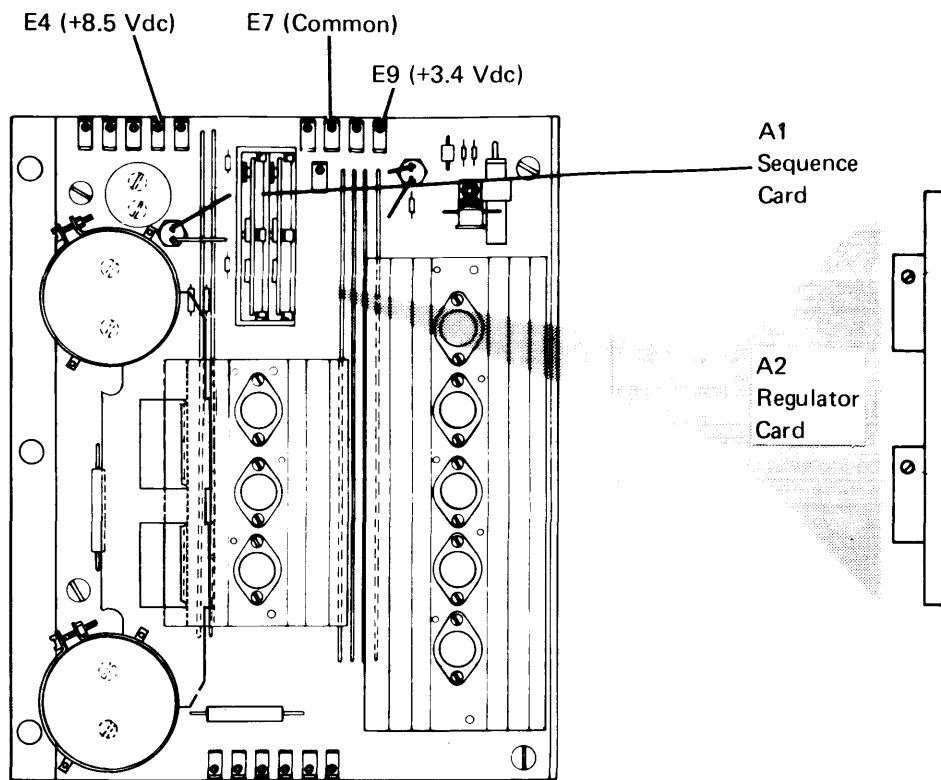


A Gate



Left Side View

Right Side View



### Voltage Adjustments

#### +3.4 Vdc

1. Connect meter across E7 (common) and E9 (+3.4 Vdc).
2. Set voltage for 3.45 Vdc.
3. Connect meter between A gate lower laminar bus position 1 (+3.4 Vdc) and position 7 (grd). This voltage should measure +3.4 Vdc. If not, readjust the +3.4 Vdc supply.

#### +8.5 Vdc

1. Connect meter across E7 (common) and E4 (+8.5 Vdc).
2. Set voltage for +8.55 Vdc.
3. Connect meter between 01A-A4L2D07 (+8.5 Vdc) and position 7 (grd).

# EXPANSION MAIN STORAGE POWER SUPPLY (Part 2 of 2)

## Sequencing

The following inputs are required to sequence up the main storage power supply:

1. AC Power – AC to bulks via K3.
2. +24V Control Voltage (E2) – Powers the sequence card.
3. +24V Start/Stop (E3) – Signals the sequence card to begin monitoring the -4V sense line.
4. -4V Sense (E1) – The supply is turned on when this line reaches approximately -3 Vdc (monitored at the O2 brass plate).

## Overvoltage (OV)/Undervoltage (UV) Protection

### Sequence Card (A1)

The sequence card provides sequence control for the +3.4 Vdc and +8.5 Vdc regulators. The +3.4 Vdc supply is sequenced up first on power-up and down last on power-down.

### Regulator Card (A2)

The regulator card provides voltage regulation for the two series regulators of which the outputs are +3.4 Vdc and +8.5 Vdc. The regulator card contains two potentiometers, one for adjusting the +3.4 Vdc level and one for adjusting the +8.5 Vdc level.

### +3.4 Vdc

A +3.4 Vdc OV (+4.2 Vdc) or UV (+2.5 Vdc) is sensed by the sequence card, which provides gate signals to bias on both Q1 and Q2 (SCRs). The SCRs cause short circuits across the +3.4 Vdc and 8.5 Vdc supplies, tripping CB1 and CB2. At the same time, the sequence card turns off both regulators.

*Note:* CB1 and CB2 may or may not trip depending on the cause of the overvoltage, but both regulators stay off until system power is recycled on.

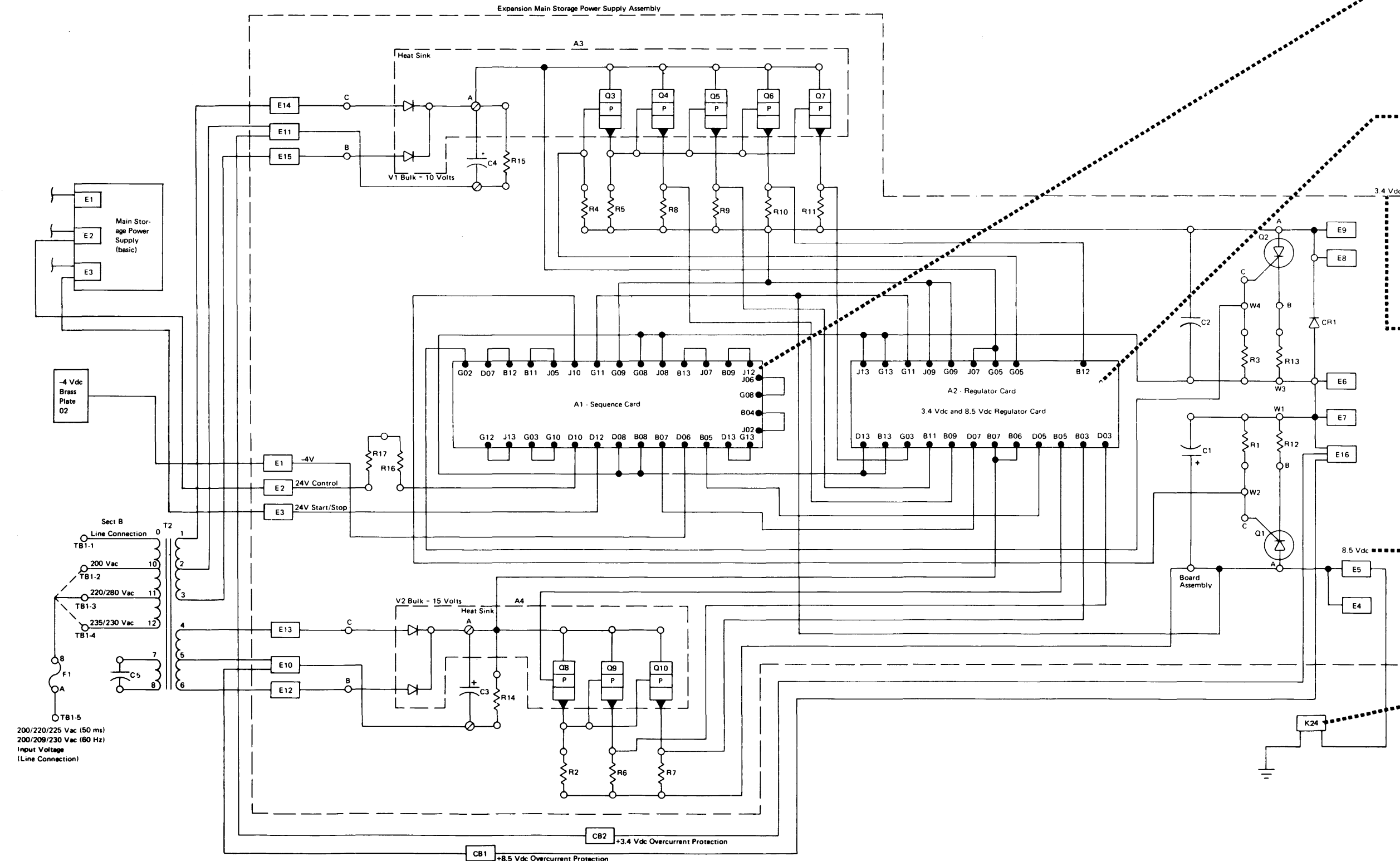
### +8.5 Vdc

A +8.5 Vdc OV (+9.5 Vdc) or UV (+2.5 Vdc) is sensed by the sequence card, which provides a gate signal to bias on Q1 (SCR). Q1 causes a short circuit across the +8.5 Vdc. This trips CB1 which disconnects the bulk voltage to the +8.5 Vdc regulator.

### Sense Relay

The sequence relay (K24) is energized when both the +3.4 Vdc and +8.5 Vdc portions of the expansion main storage power supply are up.

*Note:* K24 points are shown on the main storage power supply page.

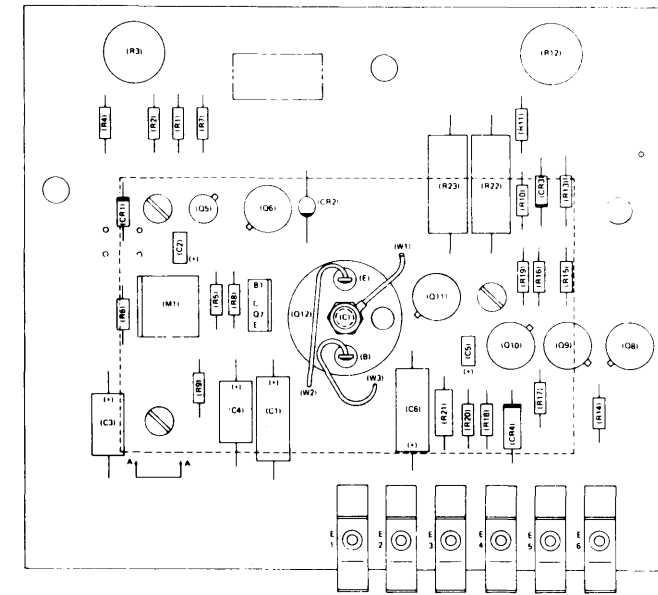


**±3 VOLT LOGIC SUPPLIES (Part 1 of 2)**

The system uses two different ±3 Vdc supplies depending on the card reader attached. If the 1442 and/or 2501 are attached, a ±3 Vdc regulator card **A** supplies the required voltage. If the system uses a 2560 MFCM, a ±3 Vdc power supply **B** is installed. This power supply provides sufficient power for the 1442 and/or 2501 (if installed).

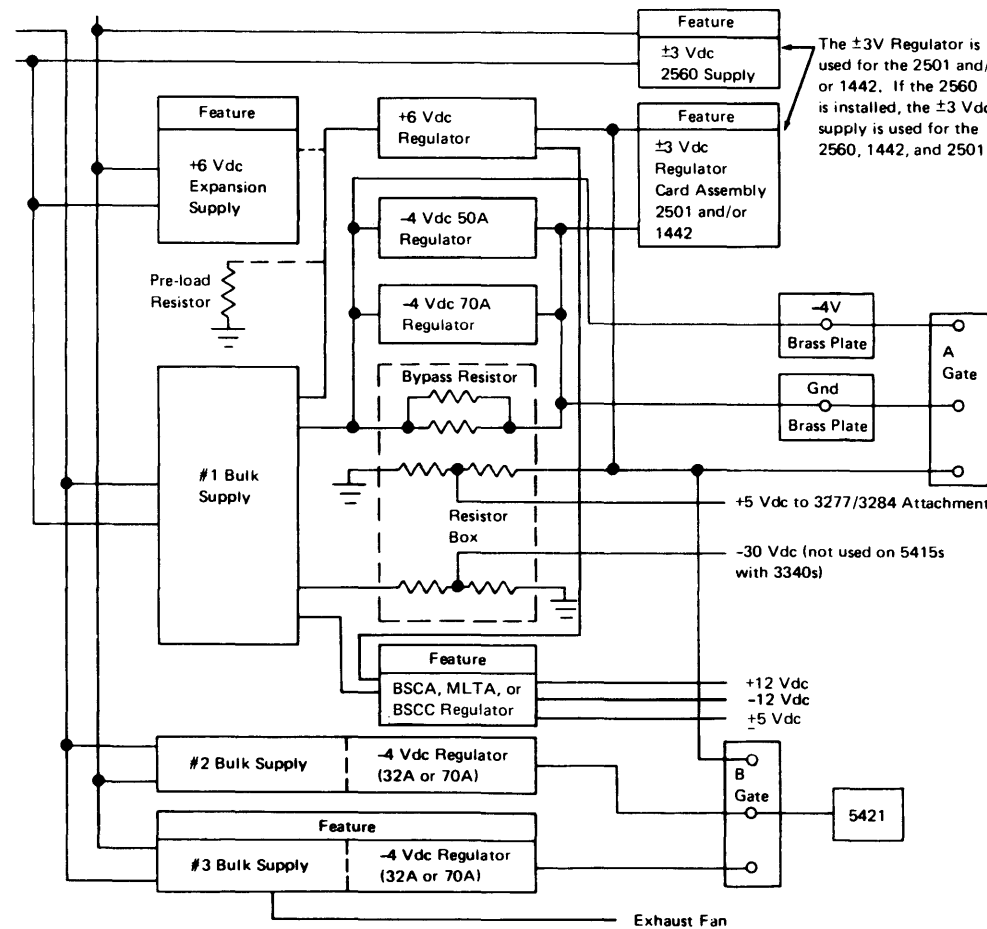
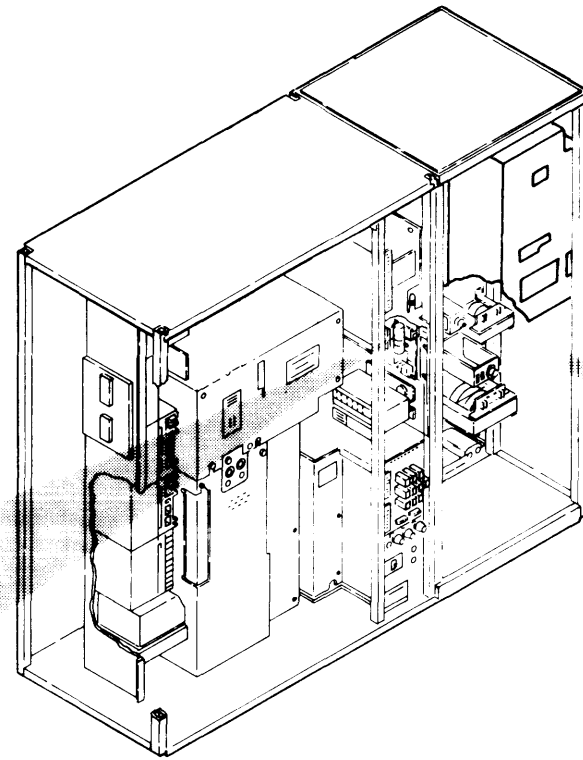
**±3 Volt Regulator Card**

The output of this regulator is +3 Vdc and -3 Vdc. The inputs are -4 Vdc (from the -4V No. 1 supply) for the -3V circuit and +6V (regulated output of the +6 Vdc regulator) for the +3V circuit.



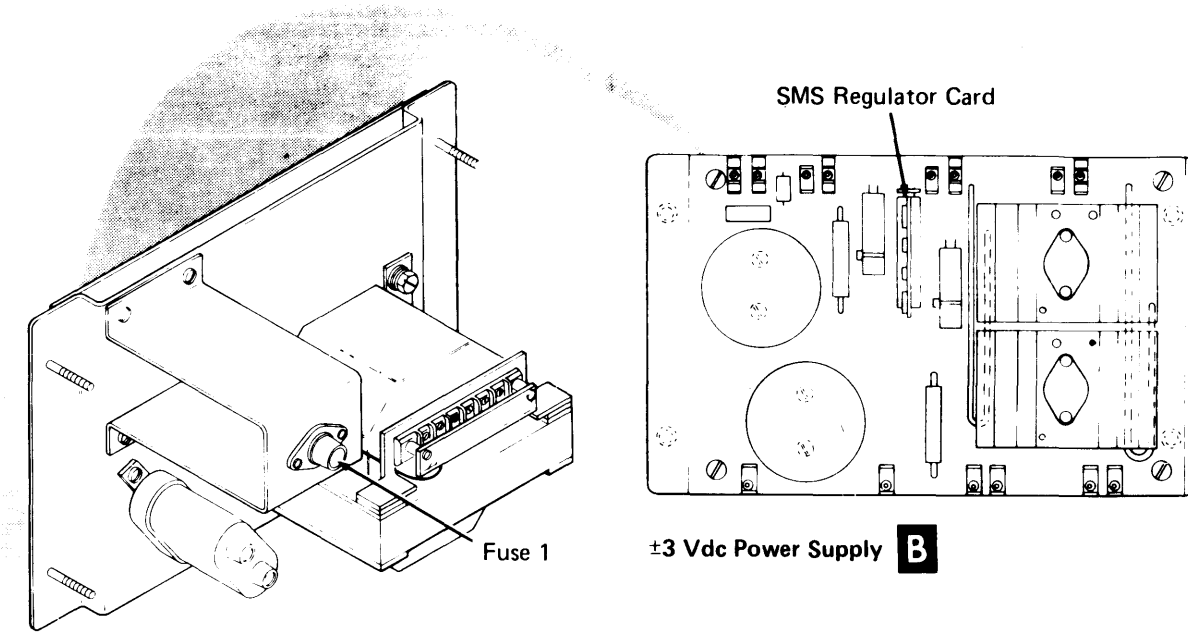
**±3 Vdc Regulator Card A**

If an OC condition occurs in either the +3 Vdc or -3 Vdc circuits, the OC condition is sensed by the regulator that supplies input to the SMS circuit in error. For example, the -4V No. 1 regulator senses an OC in the -3V SMS circuit and TP12 identifies the failure. Likewise, if an OC condition occurs in the +3V SMS circuit, the +6V regulator senses the OC and TP13 identifies the error.

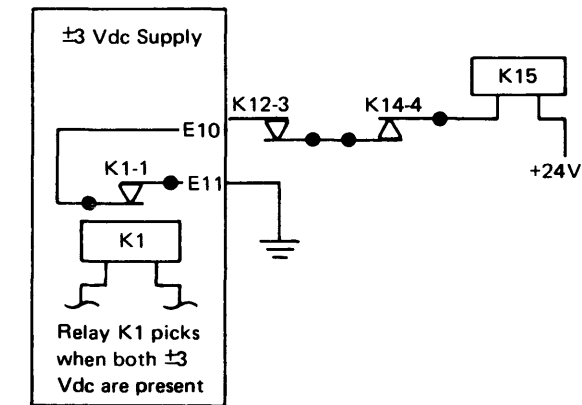


**±3 Volt Power Supply**

The ±3V power supply receives primary ac power through relay K3 and provides a regulated ±3 Vdc output at 6.5 amps. An SMS card provides the regulating circuits.



Both the +3V and -3V outputs must be active to pick relay K1 (located in the ±3V supply). If one or both of the voltages are not up when K12 (power sequence complete) picks, fault relay K15 is energized through the K14-4 points (see diagram). Fuse one provides overcurrent protection.



## ±3 VOLT LOGIC SUPPLIES (Part 2 of 2)

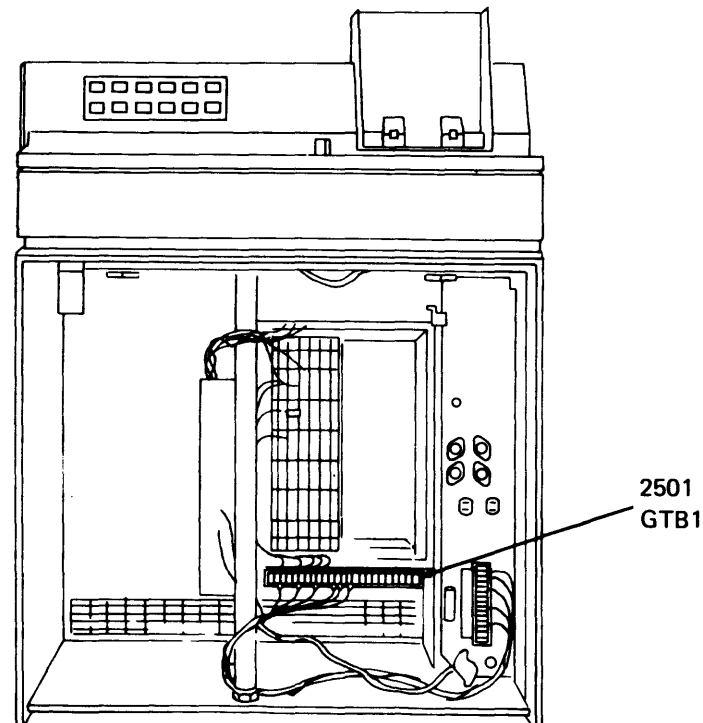
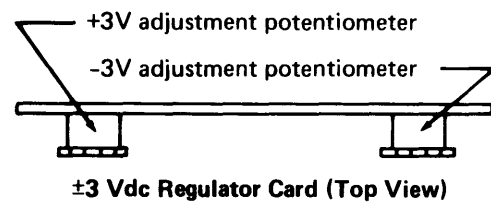
### Voltage Adjustments

#### ±3 Vdc Regulator Card

The ±3 Vdc regulator card output is adjusted while monitoring the voltage at the attached device. If both the 1442 and 2501 are installed the +3V and -3V levels are checked at both devices and the average taken.

#### +3 Vdc Level Adjustment

1. Connect meter between either the 1442 TB8-5 (+3 Vdc) and TB8-8 (grd) or 2501 GTB1-5 (+3 Vdc) and GTB1-8 (grd).
2. Set voltage adjustment potentiometer for +3 Vdc.

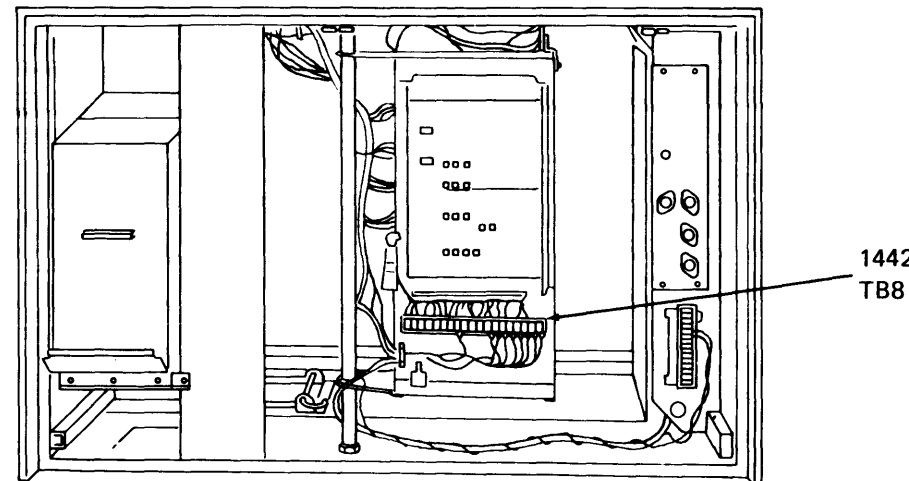


2501 Card Reader

3. If both the 1442 and 2501 are installed, connect the meter between the TB points (1442/2501) not used in step 1. Record the voltage level.
4. If the voltage level recorded differs from that set in step 2, readjust the +3 Vdc potentiometer to the mean.

For example, if the voltage recorded in step 3 is 3.04 Vdc, readjust the +3 Vdc potentiometer until the voltage level at the device used in step 3 is 3.02 Vdc.

*Note:* The voltage at either device is not to be less than ±2.88 Vdc following the mean adjustment.



1442 Card Reader

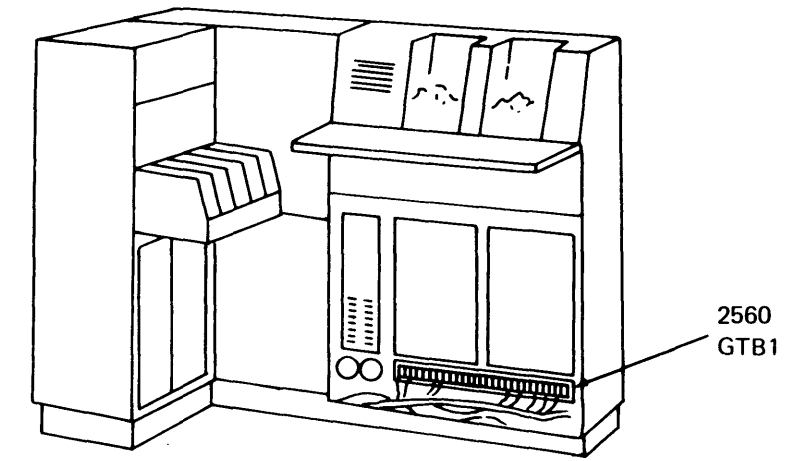
#### -3 Vdc Level Adjustment

5. Connect meter between either the 1442 TB8-11 (-3Vdc) and TB8-8 (grd) or 2501 GTB1-11 (-3Vdc) and GTB1-8 (grd).
6. Set voltage adjustment potentiometer for -3 Vdc.
7. If both the 1442 and 2501 are installed connect the meter between the TB points (1442/2501) not used in step 5. Record the voltage level.
8. Adjust the -3 Vdc potentiometer to voltage mean if necessary.

*Note:* If only the 1442 or 2501 is installed, adjust voltage at that device to the appropriate +3V and -3V levels (use only those steps required).

#### ±3 Vdc Power Supply

The ±3 Vdc power supply is adjusted while monitoring the output voltage at the 2560.



2560 Card Reader

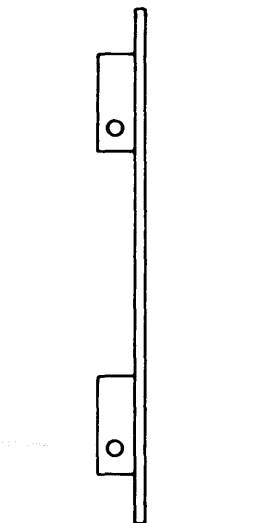
#### -3 Vdc Level Adjustment

1. Connect meter between 2560 GTB1-10 (-3 Vdc) and GTB1-9 (grd).
2. Set voltage adjustment potentiometer for -3 Vdc.

#### +3 Vdc Level Adjustment

3. Connect meter between 2560 GTB1-4 (+3 Vdc) and GTB1-9 (grd).
4. Set voltage adjustment potentiometer for +3 Vdc.

*Note:* If the 1442 and/or 2501 are also installed, adjusting the ±3 Vdc power supply at the 2560 only is sufficient.



SMS Regulator Card

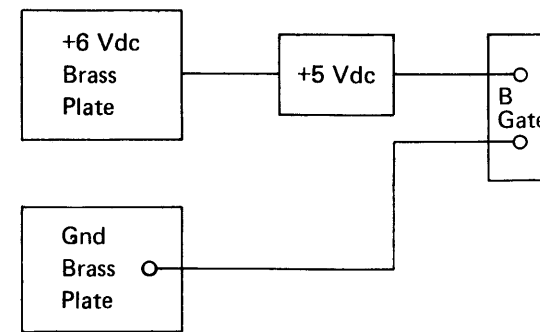
## ±5 VOLT FEATURE POWER SUPPLY

### ±5 Vdc Feature Regulator (BSCC)

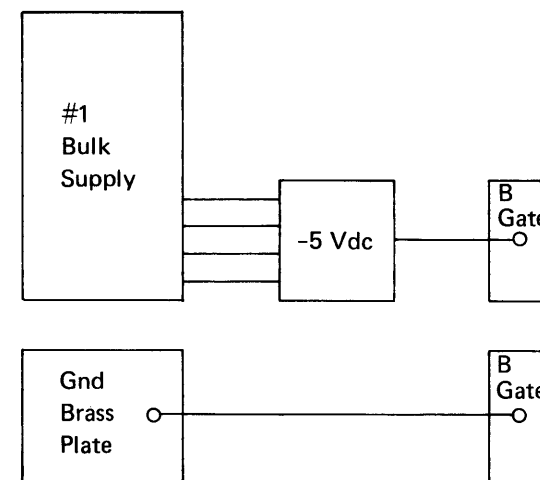
The +5 Vdc power supply is required when BSCC is installed. The +5 Vdc is regulated to +10% and to -9.5% and is derived from the +6 Vdc supply at the brass plate. It uses the ±6 Vdc regulator for overcurrent and overvoltage protection.

The -5 Vdc power supply is required when BSCC and either the Data-Phone Digital Service Adapter (DDSA) or the 38LS modem is installed. The -5 Vdc is regulated to +10% and -9.5%, and is derived from the No. 1 bulk power supply transformer (ac) at the 12 Vac bulk level (6 Vdc regulated bulk). The 12 Vac is rectified and regulated with an ICVR and a pass transistor. The regulator is current protected by a feedback circuit within itself. The 12 Vac input is fused ahead of the rectifier and regulator. The -5 Vdc power supply is power sequenced up with the No. 2 bulk power supply. If -5 Vdc is not present, relay K17 cannot pick, the system will not power up, and the power check indicator will turn on.

### +5 Vdc



### -5 Vdc



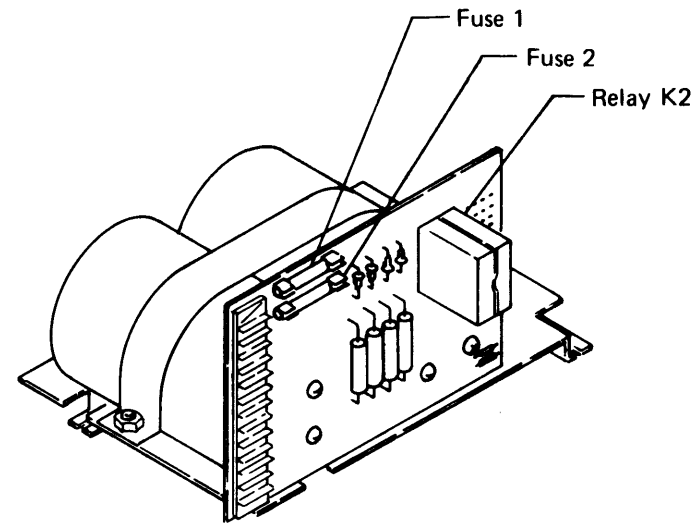


**±12/-12 VOLT FEATURE POWER SUPPLIES**

**±12 Vdc Feature Regulator (MLTA and BSCA) **A****

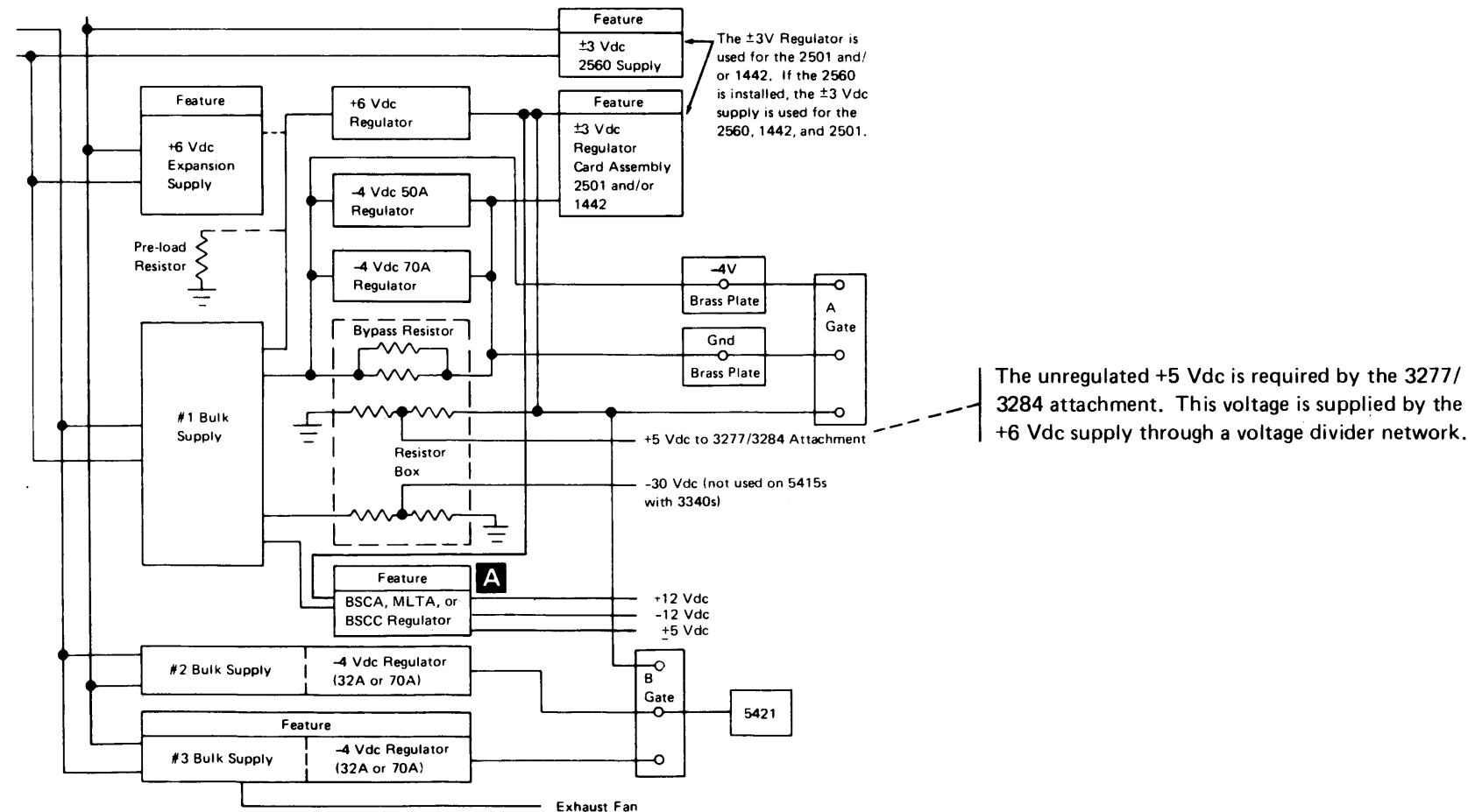
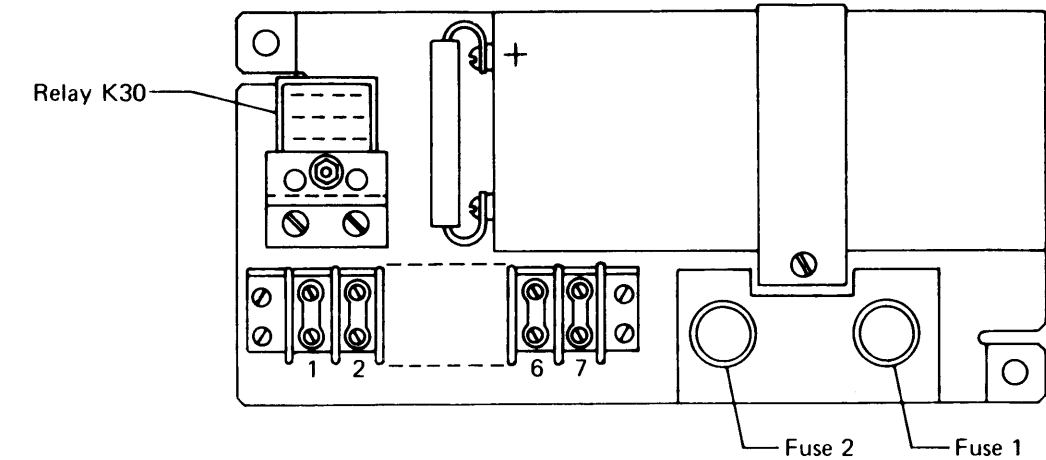
The ±12 Vdc supply is required when the MLTA feature is installed. This supply is also required with BSCA and not MLTA if the BSCA feature contains the 1200 bps integrated modem. The ±12 Vdc supply is not required if BSCA without 1200 bps integrated modem is installed (BSCA without this modem uses -12 Vdc only).

A secondary winding of the bulk supply (also used by +6V basic regulator) is used to supply unregulated ac voltage to the ±12V supply where a full-wave rectifier provides ±12 Vdc to the MLTA feature. Both inputs to the ±12V supply are fused.

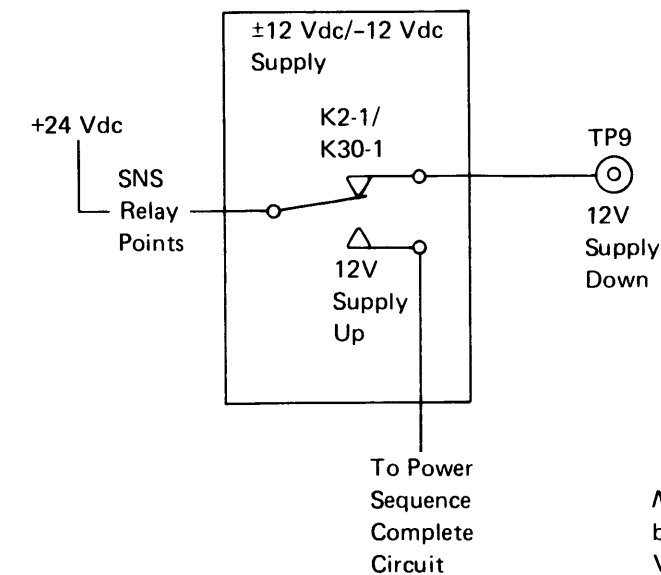


**-12 Vdc Feature Supply (BSCA or BSCC, Without 1200 bps Modem and not MLTA) **A****

The -12 Vdc feature power supply is required only when the BSCA or BSCC features without 1200 bps integrated modem is installed. The description given for ±12 Vdc regulator is valid for this supply except for the fact that +12 Vdc is not provided.



Test point 9 monitors the ±12 Vdc (MLTA) or -12 Vdc (BSCA or BSCC) supplies. If the supplies fail to sequence up, test point 9 indicates the error.



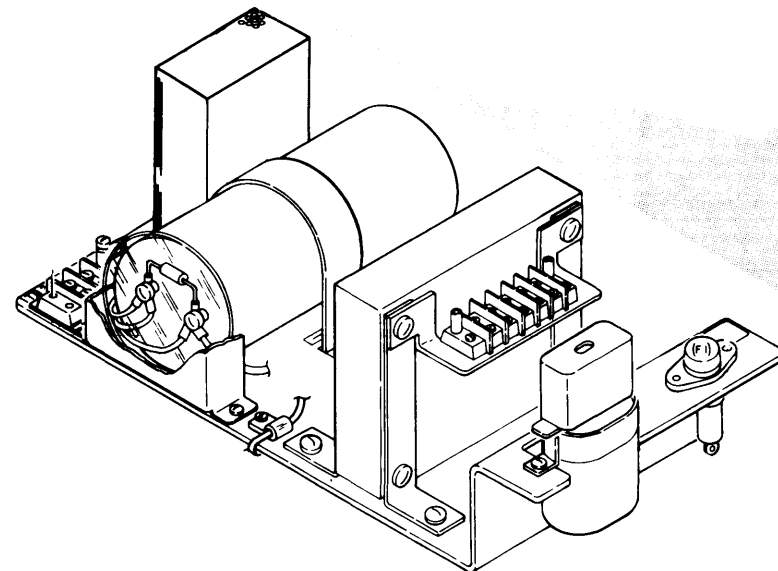
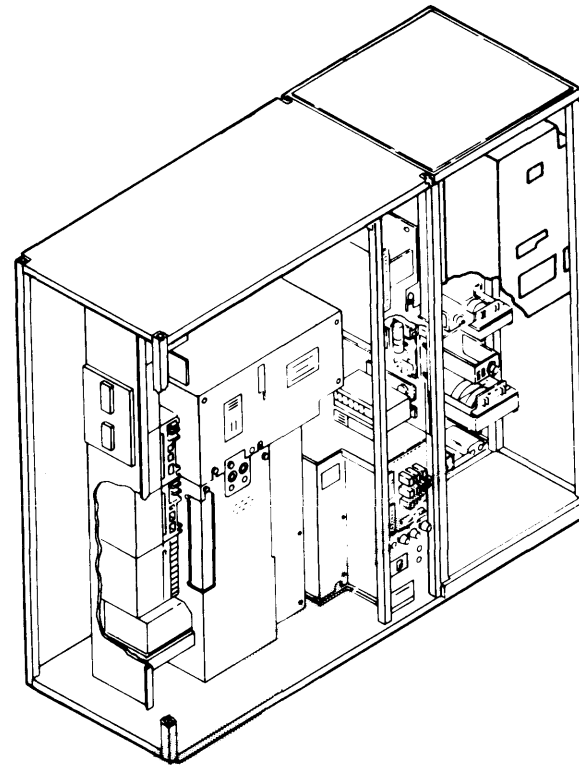
*Note:* Relay K2 (±12 Vdc supply) picks when both ±12 Vdc levels are present. Relay K30 (-12 Vdc supply) picks when -12 Vdc is present.

This page intentionally left blank.

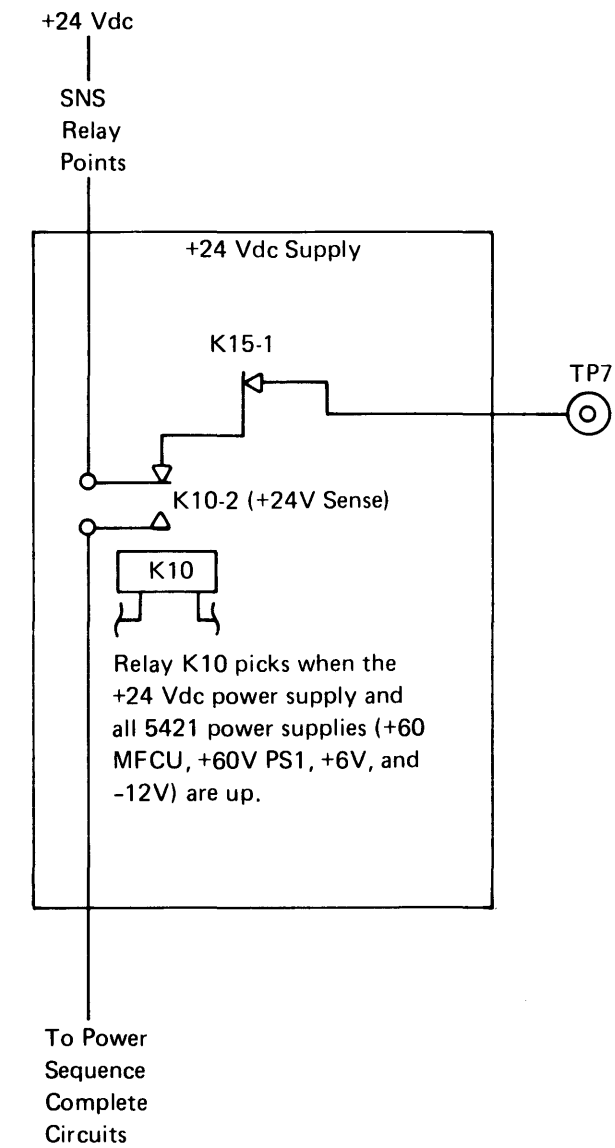
## +24 VOLT LOGIC SUPPLY

The +24 Vdc (25 amp) power supply is normally contained in the 5424. Those system configurations without a 5424 installed have a +24 Vdc 5 amp supply mounted in the 5415D. This supply is shown in the diagram.

The +24 Vdc power supply receives primary ac power via relay K9 and provides an unregulated +24 Vdc output at 4.5 amps. The +24 Vdc output is used by the 5421 and 2560.



Test point 7 monitors the +24 Vdc supply. If the supply fails to sequence up, test point 7 indicates the error.



## POWER SEQUENCE (Part 1 of 3)

These pages describe the sequential action of the system power supplies and their functional units. The purpose of the relays and coils is described on page 8-28. Reference logic pages YA102 through YA103A.

### Power Up Sequence

Normal conditions with system POWER switch set OFF, main CB on, line source on, and no power faults:

1. K13 and K14 are not energized (EPO must not be pulled and no point to ground fault).
    - a. K13-1 is *not* picked (allows +24V to be passed to K14-2).
    - b. K14-2 is *not* picked (allows +24V to be passed through power switch section A in the off position to the thermal switches to energize K2).
  2. +24 Vdc control voltage is available (TP2 = 24 Vdc).
  3. K1 is energized (convenience outlet on).
  4. K2 is energized if:
    - a. No thermal switch is open.
    - b. +24 Vdc control is available (main CB on).
- Note:* Abnormal power down conditions are described on page 8-27.
5. Lamp test switch is active (only TH CHK and PWR CHK lights will light with lamp test).

1

1

Move the system POWER switch to ON.

Sections A, B, and C of the power switch transfer.

1. Section C turns on PWR CHK light (turns off when K12-2 picks). Section C is also used to inhibit a point-to-ground fault reset while the POWER switch is set ON.
2. Section B provides +24V control to the 3340/3344, +24V control to the main storage supply, and +24V to the n/o K6-2 contacts.
3. Section A energizes K3 coil through K2-4 n/o points. (The thermal relay [K2] is now held through K2-3 n/o points.)

K3 relay picks:

1. Power (ac) is applied to all fans.
2. Power (ac) is applied to No. 1, 2, and 3 bulk power supplies, to the +6V expansion power supply, and to the -5V power supply (when installed).
3. Power (ac) is applied to the main storage supply basic and expansion main storage power supply.
4. Power (ac) is applied to ±3V supply for 2560 (when installed).

2

2

Bulk logic supply voltage energizes regulators:

1. Bias voltage (20 Vdc ± 10%) is applied to -4V (No. 1, 2, and 3) and +6V regulators (terminals E9 and E10 of each).
2. Bulk voltages are applied to each regulator (terminals E1 and E2 of each).
  - a. 9.37V to 11.3 Vdc to -4V regulator
  - b. 10.77V to 13.5 Vdc to +6 Vdc regulator
3. 46 Vdc (±10%) is applied to the resistor box assembly (30V section).
4. Approximately 25 Vac is applied to the 12V supply (BSCA/MLTA feature).
5. K30 is energized (senses BSCA -12V supply) or K2 is energized (senses ±12V supply).
6. K30-1 or K2-1 picks (TP9 = 0 Vdc).

*Note:* The ±12V power supply relay K2 should not be confused with K2 in the system (mounted on relay panel).

3

(To next page)

**POWER SEQUENCE (Part 2 of 3)**

3

**-4V regulators power up:**

1. Expansion memory power supply powers up (when -4V No. 1 output reaches -3V).
2. K5 is energized (-4V No. 1 regulator output sense).
3. K17 is energized (-4V No. 2 regulator output sense).
4. K19 is energized (-4V No. 3 feature regulator output sense).
5. K5-1 picks (TP2 = 0 Vdc).
6. K17-2 picks (TP3 = 0 Vdc).
7. K19-2 picks (TP4 = 0 Vdc) — -4V No. 3 feature must be installed.
8. K5-2, K17-1, and K19-1 pick (all three picked apply the start-up voltage to terminal E12 of the +6V regulator).
9. K20, K21 (feature), and K24 (feature) pick (passes -4V to the E1 terminal of the main storage power supply).

Main storage power supply (basic) powers up when K20, K21, and K24 have picked:

1. K8 is energized (+8.5V sense).
2. K8-1 picks plus K6-2 picked (energizes K9, K9A, and K9B; and passes +24V to the 5421).
3. K8-2 picks (TP6 = 0 Vdc).

4

4

+6V regulator and +5V supply (if installed) powers up:

1. K6 is energized (+6V regulator output sense).
2. K6-2 picks (along with K8-1 picked, energizes K9, K9A, and K9B and also passes +24V to the 5421).
3. K6-3 picks (activates -4V UV detection circuit).
4. K6-4 picks (TP5 = 0 Vdc).

K9, K9A, and K9B relays pick:

1. K9B-1 picks (passes 7.25 Vac to lamp distribution TB). Clock 9 is forced on.
2. K9B-2 picks (passes 41 Vac to usage meter control card).
3. K9A-1 and 3 pick (passes ac voltage to 3277 and 3284).
4. K9-1, 2 and 3 pick (passes ac voltage to the +24 Vdc power supply and I/O devices).

5

5

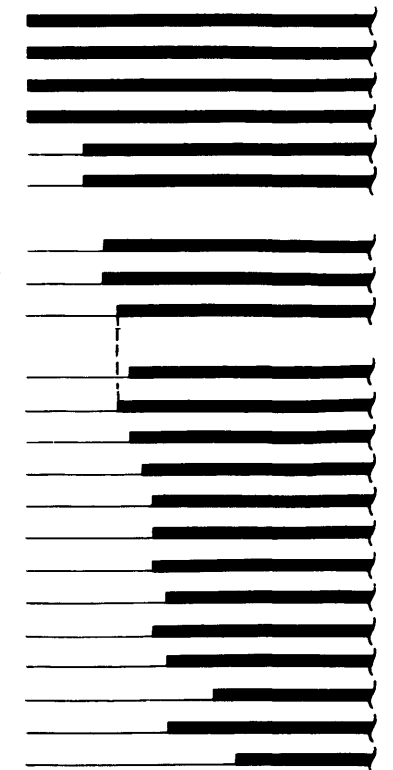
+24V supply in 5415/5424 and +60V P/S in 5421 power up:

1. K10 and K11 are energized (+24V and +60V sense).  
*Note:* For K10 to be picked, all 5421 voltages must be up and sensed and K7 in the 5421 must be picked. This allows the +24V in the 5415/5424 to be passed through the n/o points of K7 to pick K10.
2. K10-2 picks (TP7 = 0 Vdc).
3. K11-2 picks (TP8 = 0 Vdc). K12 is energized.
4. K12-1 picks (passes power sequence complete signal to the 5421 and passes +24V to 2560 feature).
5. K12-2 picks (PWR CHK light goes off).
6. K12-3 picks (inhibits K15 from being energized during a normal power on and off sequence).
7. K12-4 picks ('power on reset' is deactivated and clock 9 goes off).

System is ready for processing.

**Power On Sequence**

1. Main CB On (power switch OFF)
2. +24 Vdc control voltage
3. K1 (convenience outlets)
4. K2 (TH CHK light turns off)
5. Turn Power switch ON
6. K3 (ac voltage to logic supplies and fans) and -5 Vdc power on (BSCC only)
7. K5 (-4V power on)
8. K24 Expansion main storage power supply
9. K5-2, K17-1 (+6V power on) and +5 Vdc power on (BSCC only)
10. K6 (6V sensed)
11. (+3.4V power on)
12. (8.5V power on)
13. K8 (8.5V sensed)
14. K9-B (lamp and meter voltage)
15. K9-A (ac voltage to 3277 and 3284)
16. K9 (ac voltage to I/O devices)
17. (+24V power on in 5415/5424)
18. K1 in 5421 (5421 start up)
19. K7 in 5421 (+6V, -12V, +60V in 5421)
20. K10 (+24V sensed)
21. K11 (+60V sensed)
22. K12 (remove POR)



**POWER SEQUENCE (Part 3 of 3)**

**Normal Power Down Sequence**

Move the POWER switch to OFF.

Sections A, B, and C of the power switch transfer.

1. Section C deactivates the -4V UV detection circuit. (Applies 'Ax inhibit' to -4V UV detection circuit.)
2. Section A opens. K10-1 and K11-1 hold K3 energized until both the +24V supply and +60V supply are powered down on a normal power off sequence.
3. Section B de-energizes:  
K9, K9A, and K9B (I/O supplies).  
+3.4/8.5V main storage supply.

Main storage supply and the expansion main storage power supply (feature) powers down.

1. K8 is de-energized.
2. K8-1 drops (used in power on sequence).
3. K8-2 drops (TP6 = 24 Vdc)
4. K12 is de-energized.
5. K12-2 drops (PWR CHK light comes on).
6. K12-3 drops (inhibits K15 from being energized).
7. K12-4 drops (activates 'power on reset' to the CPU).

+60V and +24V supplies power down. (I/O supplies de-energize more slowly than the main storage supply.)

1. K10 and K11 are de-energized.
2. K10-2 and K11-2 drop (TPs).
3. K10-1 and K11-1 (de-energize K3).



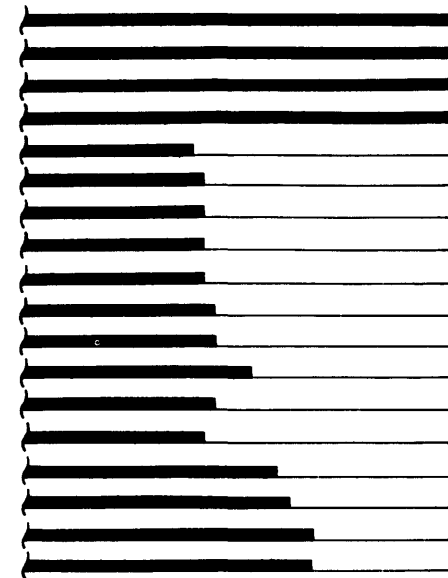
K3 drops (removes bulk and bias voltages to -4V, +6V regulators and voltage divider).

1. -4V regulators (1, 2, and 3) and +6V regulators power down.
2. K5 is de-energized.
3. K5-1, K17-2, and K19-2 drop (TPs).
4. K5-1 drops (TP2 = 24 Vdc and PWR CHK light goes off).
5. K5-2, K17-1, and K19-1 drop (used in power on sequence).
6. K6 is de-energized.
7. K6-4 drops (TP5).

System is in normal power off state.

**Power Off Sequence**

1. Main CB
2. +24 Vdc control voltage
3. K1 (convenience outlet)
4. K2 (thermal interlock)
5. Turn POWER switch OFF
6. K9 (ac voltage to I/O devices)
7. K9A (ac voltage to 3277 and 3284)
8. K9B (lamp and meter voltage)
9. K1 in 5421 (+60, +6, and -12 Vdc)
10. K11 (+60 Vdc sensed)
11. K10 (+24 Vdc sensed)
12. K12 (turn on POR)
13. K7 in 5421
14. K8 and K24 (+8.5 Vdc  
+3.4 Vdc)
15. K3 (ac voltage to logic supplies and fans)
16. K6 (+6 Vdc sensed)
17. K5 (-4 Vdc sensed)



## THERMAL AND POWER CHECKS

### Abnormal Power Indications

An abnormal power down can be caused by a power check or a thermal check.

A PWR CHK light indicates that one of the following has occurred:

- An abnormal power down sequence **A**
- An abnormal power up sequence **B**
- An overvoltage, overcurrent, or an undervoltage power failure **C**
- A thermal condition exists (TH CHK light will be on also) **D**

A thermal check indicates that one of the following areas has overheated:

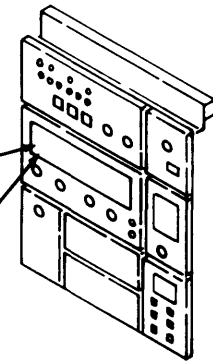
- A or B gate logic
- Bulk supply
- Regulator stack
- Main storage supply
- Expansion main storage supply (feature)
- 5421 and 2560 supplies
- 5421 logic gate

**A** An *abnormal power down sequence* occurs only when the POWER switch is moved to OFF (the operator intends to power down the system). If all system power supplies power down as expected, no power check occurs. If they do not, the PWR CHK light stays on (comes on during normal power down sequence) and +24 Vdc may or may not be available on one of the test points.

Any one of the following faults could cause an abnormal power down sequence.

1. Section A, B, or C of the power switch failed to open.
2. K3 relay did not drop.
3. K9 relay did not drop.

4. K12 did not drop.



One failure is examined to show how the power supply elements react: With the POWER switch OFF, and if relay K3 fails to drop ac power continues to be supplied to the -4V and +6V regulators. Since K5, K17, and K6 are never de-energized, K5-1, K17-2, and K6-4 never drop. +24 Vdc is measured on TP6. The PWR CHK light is on because +24V is available through K5-1, CR13 and K12-2.

An *abnormal power down sequence* should not be confused with an overvoltage (OV) or overcurrent (OC) fault occurring during a power down. If an OV or OC occurs during a power down sequence, the system powers down as expected and the PWR CHK light does *not* stay on but goes off as soon as all supplies are down. However, a fault relay (K13, K14, K16, or K18) latches so when the POWER switch is moved to ON, an immediate power check occurs with no power supply sequencing. (Normally an OV/OC causes the system to immediately power down at some point in the sequence up state or after all supplies are up.)

**B** An *abnormal power up sequence* can occur after the POWER switch is moved to ON. This power failure can occur if any one of the system power supplies fail to power up or if any one of the sense relays fail to sense an associated power supply output. The system does *not* abruptly power down with this fault but inhibits power complete or stops sequencing up at the point of failure for some supplies. +24 Vdc is measured on the TP of the supply that failed to power up. For Example: If either the main storage power supply or the expansion main storage power supply (feature) has an internal failure and its output never reaches +8.5V, K8 is not energized and relays K8-1 and K8-2 are not transferred. With K8-1 not picked, K9, K9A, and K9B are not energized (I/O supplies). With K8-2 not picked, +24 Vdc appears at TP6. K12 is not energized and the PWR CHK light stays on.

**C** A -4V or +6V overvoltage, overcurrent, or a -4V undervoltage power failure can be detected during normal power on. With this type of failure, the system powers down *abruptly* as soon as the fault is detected. +24V is measured on the TP of the supply that detected an OV, OC, or UV power failure. The respective regulators sense voltage and current load drain and if preset limits are exceeded, a point-to-ground signal is presented on pin E8 of the respective regulator. This point-to-ground signal (ground potential) is used to energize the appropriate OV/OC sense relay which, in turn, immediately drops all system power. Undervoltage power failures also cause abrupt system power down. Two special UV detection circuits are used to sense the output of the -4V and +6V regulators. If either regulator's output drops below criteria, K6/K15 (+6V UV) or K14 (-4V UV) is energized and powers down the system. However, a +6V UV does *not* power down the system abruptly but causes a power check with all CPU logic supplies up.

Overcurrent (OC) power failure example: system is in a normal power up state.

1. A -4V short-to-ground fault occurs (can be anywhere on the system where -4V from the No. 1 bulk is used).
2. The fault causes excessive current drain on the -4V regulator.
3. A -4V OC fault is sensed by the -4V regulator card.
4. A point-to-ground signal is presented to K13. (The ground side of K13 is normally open to ground.)
5. K13 is energized.
6. K13-2 picks (latches K13 on).
7. K13-1 picks (TP12 = +24 Vdc). When this relay transfers, +24 Vdc control voltage is removed from all relay coils (K2, K3, and K9 in particular).
8. System powers down abruptly.
9. PWR CHK comes on as soon as K12 drops and the POWER switch stays ON.

If the POWER switch is moved to OFF, the PWR CHK light goes off, but comes on again each time the switch is moved to on. The system stays latched in the fault state until the fault relay is reset and the point-to-ground fault is removed.

The fault relay can be reset by setting the POWER switch to OFF and pressing CHECK RESET. With the POWER switch set OFF, +24V energizes K13 coil through the closed check reset switch. With check reset open, K13 de-energizes and K13-2 drops (latch) and K13-1 drops. K2 is now energized by the +24V control voltage and the system is ready for a normal power up sequence.

The preceding description can be used similarly for -4V OV/OC, -4V UV, and +6V OV/OC power failures. Only different supplies and relays are involved.

A +6V UV power failure does *not* cause abrupt system power down. Instead, special solid state circuits sense the +6V regulator output and, if +5.7 Vdc or lower is sensed, coil K6A is energized. Relay K6A-1 picks and causes K15 to energize. Then K15-4 picks and de-energizes K9, K9A, and K9B. This relay drops ac power to the I/O devices, which causes K10 and K11 to drop. The PWR CHK light comes on, but the system does *not* power down since +24V control is not interrupted to K3 or K2.

Undervoltage control circuits are used to ensure that:

- a) +6V is removed when the +6V output falls below +5.3 Volts.
- b) -4V is always present when +6V is present in the 5424, or CPU logic gate.

If the above is not controlled, damage results in I/O device control circuits and electro-magnetic components.

Normally a power supply itself *cannot* cause an OC failure. If an OC condition prevails, an I/O device, logic circuits, or cables have caused the failure. If the supply is abnormally overloaded, an OC condition prevails over an UV condition. Even though the regulated supply voltage may drop, normally the OC sensing by the regulator has powered the system down *before* UV can be detected.

**D** A *thermal check*, caused by one of the following overheated areas, causes the system to power down in a sequential manner rather than abruptly.

Thermal check example: System is in a normal power on state.

1. A thermal switch opens from an overheating condition.
2. +24V is interrupted to K2 coil, K2 de-energizes.
3. K2-3 and K2-4 transfer.
4. K2-3 interrupts +24V to K9, K9A, and K9B (I/O supplies power down) and passes +24V to the thermal light.
5. K2-4 transfers, and K3 is held energized through K10-1 and K11-1 until both the +24V power supply and the +60V power supply are down.
6. System powers down as a normal power down sequence, except that the TH CHK light is on.

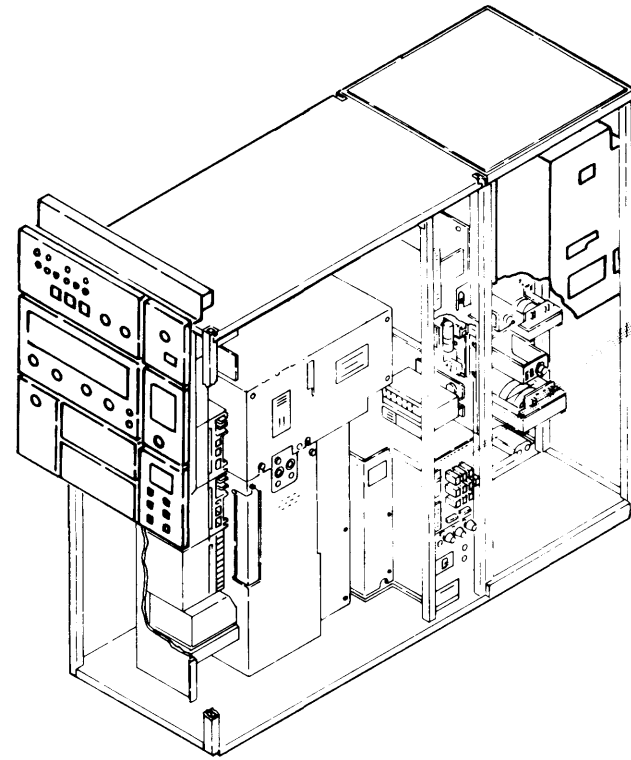
If the POWER switch is set OFF, the TH CHK light stays on if the thermal fault still exists. If the fault is corrected, moving the switch to OFF allows K2 to be energized again and the TH CHK light goes off.

**THERMAL AND POWER CHECKS**

**Power Check and Thermal Check Indicators**

The PWR CHK light comes on during the power on sequence and goes off when the power on sequence is completed. It also lights with the TH CHK light (see chart below) when an overtemperature condition occurs or whenever any power trouble is present.

A 'power on reset' occurs every time the PWR CHK lights. PWR CHK stays off if the 24 Vdc output of the control transformer/rectifier pack (T/R Pac) is missing.



**Test Points (TPs)**

Test points (TPs) are on the power control box. When a voltage failure occurs, check these test points in numerical sequence to determine the voltage that failed.

If the power on sequence is not completed, the PWR CHK light remains on and the TPs from TP2 to TP9 indicate where the sequence stopped.

For example, a +6V regulator sequencing failure is indicated if TPs 2-4 were zero volts and +24V appeared at TP5.

The machine powers down in any of the conditions detected in TP10-14. Twenty-four volts is readable in TP10-14 until CHECK RESET is pressed. Loss of either the -4V or +6V while the machine is running powers down the system and +24V is present at TP10-14. Loss of +24V while the machine is running does not cause power down but activates 'power on reset' stopping operation of the machine.

For example, an overvoltage/overcurrent failure in the -4V No. 1 regulator occurred if +24V appeared at TP12.

Fault	POWER ON/OFF Switch	Power Check Ind	Thermal Ind	Action
Internal Power Supply Malfunction	On	On	Off	1. Turn power off. 2. Correct problem. 3. Press check reset. 4. Turn power on.
Thermal Condition	On	On	On	1. Turn power off. 2. Power check indicator goes off. 3. Thermal light stays on until condition is removed.
Customer Power Source Loss	On	On	On	1. Turn power off. 2. All indicators turn off. 3. Turn power on and continue operation.
Emergency Power Off (EPO) Activated	On	Off	Off	1. Turn power off. 2. Correct problem. 3. Restore EPO interlock. 4. Turn power on.

Test Points	Type of Failure If TP = 24 Vdc (Nominal)	Power Supply Checked
TP1	Ground	
TP2	Sequence	-4V #1 logic supply (A gate)
TP3	Sequence	-4V #2 logic supply (B gate) -5V logic supply (BSCC only)
TP4	Sequence	-4V #3 logic supply (Feat) B gate
TP5	Sequence	+6V Supply
TP6	Sequence	+3.4V/+8.5V main storage power supply
TP7	Sequence	5415/5424 (+24V basic) 5421 (+60V PS1, -12V, +6V, +60V multilevel)
TP8	Sequence	5421 (+60V supply SNS circuit)
TP9	Sequence	Feature supply (-12V supply or ±12V supply)
TP10	OV/OC	-4V #3 logic supply (Feature) B gate
TP11	OV/OC	-4V #2 logic supply (B gate) -5V logic supply (feature)
TP12	OV/OC	-4V #1 logic supply (A gate)
TP13	OV/OC UV	+6V supply -4V #1 logic supply (A gate)
TP14	Sequence	±3V feature supply (2560)
	UV	+6V supply



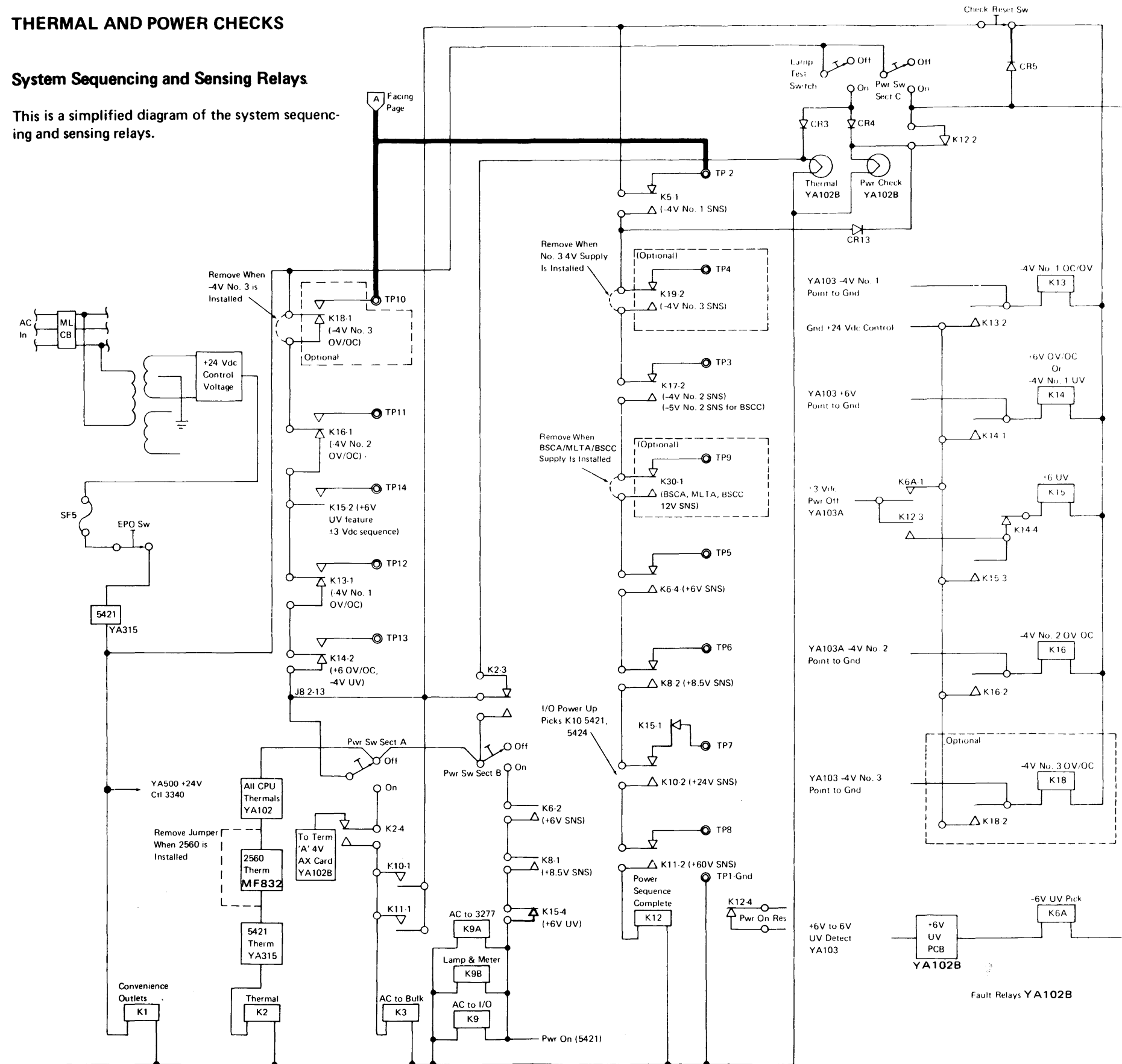
Facing page



## THERMAL AND POWER CHECKS

### System Sequencing and Sensing Relays

This is a simplified diagram of the system sequencing and sensing relays.



### Power On Sequence

K1 picks if the interlock between the CPU and 5421 is complete and the 24V control voltage is present through the EPO switch. A quick service check for this 24V control voltage power supply can be made by pressing LAMP TEST while power is off and observing the TH CHK and PWR CHK lights. If they light, the 24V supply is functioning.

K2 picks if the CPU, 2560 (if installed), and 5421 thermals are closed, and all fault relays are de-energized.

**Note:** K1 and K2 pick with the POWER switch on or off.

K3 picks when the POWER switch is turned ON.

K12 picks when the power on sequence is complete.

### Abnormal Power Off

The five causes for an abnormal power off sequence are:

1. Overvoltage (OV)
2. Overcurrent (OC)
3. Undervoltage (UV)
4. Thermal (overheating – normal power off sequence)
5. Emergency power off (EPO) switch opened

### Overvoltage and Overcurrent Power Off Sequence

Whenever an overvoltage or an overcurrent condition is sensed, one of the OV/OC relays, K13, K14, K16, or K18 is picked. Energizing an OV/OC relay results in de-energizing contactor K3. De-energizing contactor K3 removes power from the logic and main storage supply.

On an abnormal power off, the power check indicator turns on to indicate a failure. Test points indicate the power supply that failed. The energized OV/OC relay contacts hold the relay energized until CHECK RESET is pressed with the POWER switch OFF.

After an overvoltage, overcurrent, or an undervoltage failure, CHECK RESET must be pressed with the POWER switch set OFF to de-energize the OV/OC/UV relay and to allow a power on sequence.

### Undervoltage Power Off Sequence

Only the -4V and the +6V outputs sense for undervoltage conditions. If the -4V No. 1 regulator UV circuit senses an undervoltage condition, the -4V UV circuit (a separate card) immediately signals the +6V regulator to short via the SCR across the +6V regulator output. This is a +6V simulated overcurrent condition and the OV/OC/UV relay K14 energizes. The K14-2 contacts removed +24V from contactor K3. Contactor K3, in turn, removes power to the logic and main storage bulk supply. This results in an immediate system power off.

Because K14 OV/OC/UV relay energizes, +24V is present at TP13 to indicate a +6V power failure. However, a +6V overvoltage, a +6V overcurrent, or a -4V undervoltage could cause the failure condition (see MAPs, Maintenance Analysis Procedures).

### Thermal Power Off Sequence

A thermal condition causes relay K2 to be de-energized. The K2-3 contacts turn on the TH CHK light to indicate overheating. Power then sequences off the same as a normal power off sequence by opening the power switch circuit.

The TH CHK light and the PWR CHK light are on when the system power off sequence ends. Turning the POWER switch OFF turns off the PWR CHK light. The TH CHK light remains on until the over-temperature condition has been corrected and the POWER switch has been turned OFF. Power can then be restored to the system by turning the POWER switch ON.

### Emergency Power Off

Pulling the emergency power off switch removes +24V to K1, K2, K3, etc causing system power to drop immediately.

**Note:** In a normal system power off state, TP2 will read +24 Vdc. Because of a system power failure (power check), +24 Vdc measured on TP2 indicates the -4 Vdc failed to sequence on.

## COIL/RELAY FUNCTIONAL DESCRIPTIONS

The coils/relays described below are classified into two groups: sequence relays or fault relays. Sequence relays allow the system to sequentially power up or power down. Fault relays identify a failing power supply and, in some cases, cause the system to abruptly power down to avoid circuit/component damage.

### Sequence Relays

K1\*\*\*  
 K1 (I/O P/S)  
 \*K2\*\* ( $\pm 12V$  sense)  
 MLTA/BSCA  
 K3 (bulk power)  
 \*K5 (-4V No. 1 sense)  
 \*K6 (+6V sense)  
 \*K8 (main storage supply sense)  
 K9 (I/O supply and +24 Vdc power supply input)  
 K9A (115 Vac distribution)  
 K9B (41 Vac and 7.25 Vac distribution)  
 \*K10 (+24V sense – 5415/5424)  
 \*K11 (+60V sense – 5421)  
 K12 (power supply up sense)  
 \*K17 (-4V No. 2 sense and -5V regulator)  
 \*K19 (-4V No. 3 sense)  
 \*K30 (-12V sense) with medium speed BSCA/BSCC and not MLTA

### Fault Relays

K2 (thermal)  
 K6A (+6V UV)  
 K12 (power check)  
 $\Delta$ K13 (-4V OV/OC)  
 $\Delta$ K14 (-4V UV or +6V OV/OC)  
 K15 (+6V UV or +3V sense)  
 $\Delta$ K16 (-4V No. 2 OV/OC)  
 $\Delta$ K18 (-4V No. 3 OV/OC)

\*These relays are also fault relays because they identify the failing power supply.

\*\*This K2 is located in  $\pm 12V$  supply, and not in the sequence control box.

\*\*\*This K1 is located on the -5 Vdc power supply and is a sense relay.

$\Delta$ These relays cause an abrupt system power off; all others do not.

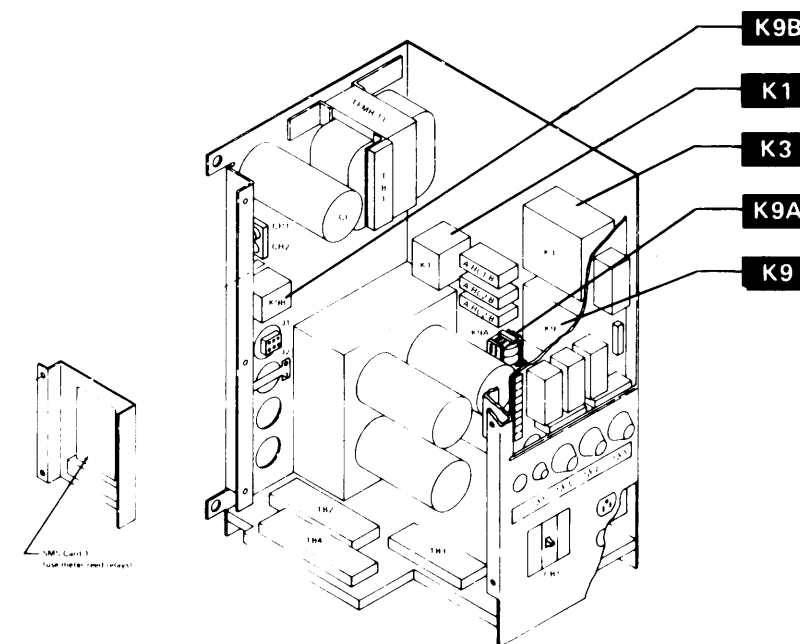
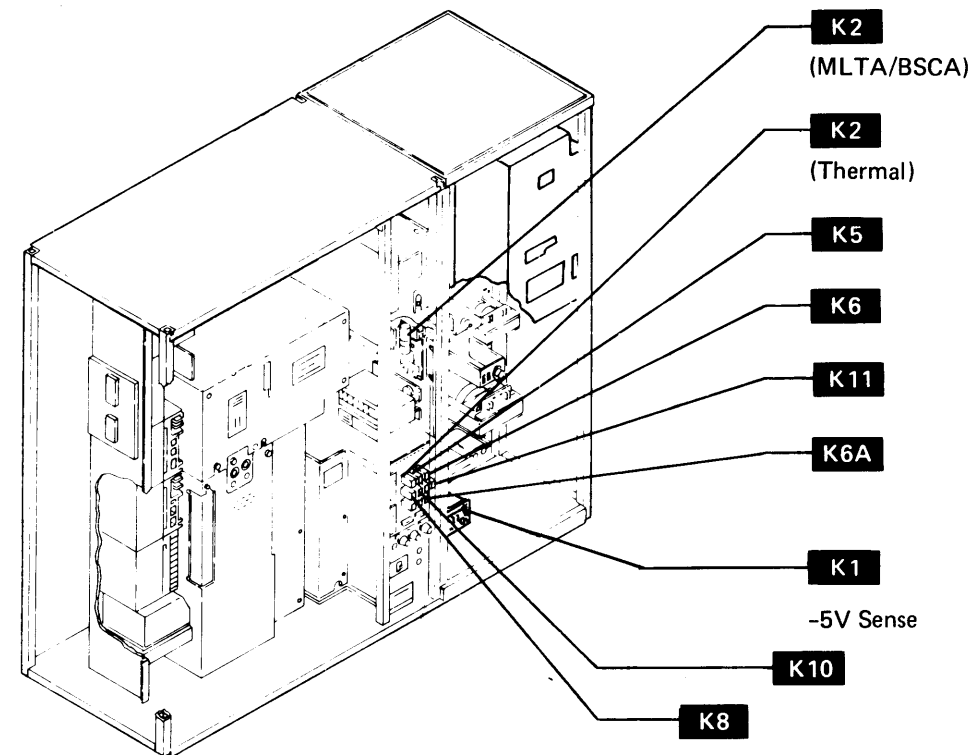
Note: K4 and K7 are not used.

### K1 Convenience Outlet (YA102)

- Pick
  - 24V control voltage power up
  - EPO switch closed (pushed in)
- Drop
  - EPO switch opened (pulled out)
  - Loss of 24V control voltage
- Function
  - K1-1, 2 provides control of 115 Vac to system convenience outlets.

### K1 -5V Sense

- Pick
  - At about 5V when -5V regulator is powered up
- Drop
  - Loss of -5V
  - K17 dropping
- Function
  - Protects the FET substrate
  - Gives a TP3 indication when failure occurs



Primary Control Box

### K2 Thermal Relay, Relay Panel (YA102)

- Pick
  - System in normal power off state
  - All thermal switches closed
- Drop
  - Any system thermal open
- Function
  - K2-1 not used
  - K2-2 not used
  - K2-3
    - Provides hold voltage for K2 coil when POWER is ON.
    - Interrupts hold current to K9, K9A, K9B coils and drops power-on signals to 5421 and 3340/3344 devices on a thermal fault.
    - Provides power to the TH CHK indicator light when a thermal fault is detected.
    - Provides input voltage to main storage supply sequence card.
  - K2-4
    - On normal sequence on, provides a path to allow K3 to pick if no thermal fault is present.
    - Provides for sequential shutdown of K3 on a power fault condition.
    - Inhibits -4V UV sense after a thermal fault.
    - Drops power-up signal to main storage supply.

### K2 $\pm 12V$ Supply Sense, Located on Power Supply (MLTA/BSCA-YA140)

- Pick
  - When the MLTA  $\pm 12V$  supply output is approximately  $\pm 12V$ dc.
- Drop
  - Loss of either +12V or -12V output.
- Function
  - K2-2
    - Provides a path for +24V to TP9 when  $\pm 12V$  supply is not up.
    - Provides the  $\pm 12V$  link in the power complete sequence chain.

Note: The  $\pm 12V$  Vdc supply is required when the MLTA feature is installed. This supply is also used with BSCA and not MLTA if the BSCA feature contains the 1200 bps integrated modem. This supply is not required if BSCA without 1200 bps modem is installed (BSCA and/or BSCC without this modem uses -12 Vdc only).

## COIL/RELAY FUNCTIONAL DESCRIPTIONS

### **K3** AC Voltage to Bulk Power Supply (YA102)

- Pick
  - Transferring the power switch (section A) after being in a normal power off state.
- Drop
  - A -4V OV/OC power fault
  - A -4V UV or +6V OV/OC power fault
  - Loss of +24V. +60V levels after a thermal fault.
  - Loss of +24V. +60V levels after transferring the power switch to the "OFF" position.
- Function
  - K3-1, 2, 3 controls ac distribution to all CPU bulk supplies:
    - a. No. 1 logic -4V/-5V/+6V/-30V and 25 Vac to  $\pm 12V$  supply
    - b. No. 2 logic -4V
    - c. No. 3 logic -4V
    - d. Main storage power supply +8.5V/+3.4V
    - e. -6V expansion +6V
    - f.  $\pm 3V$  2560  $\pm 3V$
    - g. +8.5V +3.4V expansion main storage power supply

*Note:* The outputs of the +6V regulator, +8.5V regulator, and +3.4V regulator have additional controlling functions.

- Provides ac voltage to all CPU cooling fans.

### **K5** -4V No. 1 (A gate) Sense (YA102A)

- Pick
  - -4V No. 1 (A gate) output at approximately 4 Vdc
- Drop
  - Loss of -4V No. 1 (A gate) output
- Function
  - K5-1
    - a. Provides path for +24V to TP2 when -4V No. 1 (A gate) level is not up.
    - b. Provides control of the PWR CHK light during power down sequence.
    - c. Provides the -4V No. 1 (A gate) link in the power complete (K12) sequence chain.
  - K5-2
    - a. Provides the -4V No. 1 (A gate) link in the start up control for the +6V regulator.

### **K6** +6V (+5V for BSCC only) Sense (YA102A)

- Pick
  - +6V/+5V output at approximately +6 Vdc
- Drop
  - Loss of +6V output
- Function
  - K6-2 provides the +6V up-link of the chain required to energize K9, K9A, and K9B relays that provide ac voltage to I/O devices attached to the system.
  - K6-3 provides voltage to -4V UV sense circuit. This line ensures that a -4V UV fault will not be sensed before the +6V supply is up.
  - K6-4
    - a. Provides a path for +24V to TP5 when +6V level is not up. For +24V to appear on TP5, all -4V supplies, -12V or  $\pm 12V$  supply, and  $\pm 3V$  2560 supply (if installed) must be up.
    - b. Provides the +6V link in the power complete sequence chain.

### **K6A** +6V UV Detect (YA102B)

- Pick
  - Loss of +6V output while system is in a normal power on state.
- Drop
  - +6V power restored
- Function
  - K6A-1 provides path for picking the +6 UV fault relay (K15) when system is in a normal power on state, and a +6V undervoltage is detected.

*Note:* K6A is energized with +24V only when the ground side of this coil is connected to ground. Transistors Q1 and Q2 sample the +6V regulator output and if +6V is available, Q1 and Q2 do *not* conduct and Q2 presents an open circuit to the ground side of K6A and inhibits this coil from being energized. If +6V is not available, transistors Q1 and Q2 conduct and the ground side of K6A is at ground potential and allows K6A to be energized. This causes K15 fault relay to pick which, in turn, causes a power check (K9 drops, K10 drops, K12 drops, PWR CHK light on).

### **K8** Main Storage Supply Sense (YA102A)

- Pick
  - +8.5V output at approximately 8.5 Vdc
- Drop
  - Loss of +8.5V output
- Function
  - K8-1 provides the +8.5V required to energize K9, K9A, and K9B relays that provide ac voltage to the I/O devices attached to the system.
  - K8-2
    - a. Provides a path for +24V to TP6 when +8.5V level is not up. For +24V to appear on TP6, all -4V supplies, -12V or  $\pm 12V$  supply,  $\pm 3V$  2560 supply (if installed), and +6V supply must be up.
    - b. Provides the +8.5V link in the power complete sequence chain.

*Note:* The +8.5V supply depends on the +3.4V supply being up. The +3.4V supply is *not* associated with any relay coil. However, the availability of this voltage provides a start-up to the +8.5V supply. Hence, both must be up before K8 can pick.

### **K9** I/O AC Power (YA101 and YA102)

- Pick
  - +6V power up and +8.5V power up
- Drop
  - Any OV/UV/OC fault on -4V or +6V
  - Thermal fault
  - POWER switch being set OFF
  - Loss of the +8.5V level when in a normal power up state
- Function
  - K9-1, 2, 3 controls ac voltage distribution to the +24 Vdc power supply and the following I/O devices:
    - a. 2560
    - b. 5424
    - c. 2501
    - d. 1442

*Note:* Primary power for the 2560 is supplied by the 5421 on all 60 Hz and 200 Vac 50 Hz systems.

### **K9A** I/O AC Power (YA101 and YA102)

- Pick
  - +6V power up and +8.5V power up
- Drop
  - Any OV/UV/OC fault on -4V or +6V
  - Thermal fault
  - POWER switch being set OFF
  - Loss of the +8.5V level when in a normal power up state
- Function
  - K9A-1, 3 controls ac power distribution to the 3277 and 3284.

### **K9B** 7.25 Vac and 41 Vac Distribution (Lamp and Meter – YA102)

- Pick
  - +6V power up and +8.5V power up
- Drop
  - Any OV/UV/OC fault on -4V or +6V
  - Thermal fault
  - POWER switch being set OFF
  - Loss of the +8.5V level when in a normal power up state
- Function
  - K9B-1 controls distribution of the 7.25 Vac power to the indicator lamp bus.
  - K9B-2 controls distribution of the 41 Vac power to the meter control card.

### **K10** I/O Power Sense (YA102A)

- Pick
  - All 5421 power supplies (+60V MFCU, +60V PS1, +6V, and -12V) and 5415/5424 +24V power supplies are up.
- Drop
  - Loss of any 5421 power supply (+60V MFCU, +60V PS1, +6V, or -12V)
  - Loss of 5415/5424 +24V power supply.
- Function
  - K10-1
    - a. Provides path to hold K3 energized after a thermal fault or setting the POWER switch OFF.
    - b. Provides path to supply +24V to regulator for main storage supply sequence/sense circuits.

- K10-2
  - a. Provides a path for +24V to TP7 when +24V in 5415/5424 or one of the 5421 5421 supply levels is not up. For +24V to appear on TP7, all -4V supplies, -12V or  $\pm 12V$  supply,  $\pm 3V$  2560 supply (if installed), +6V supply, and +8.5V supply must be up.
  - b. Provides the +24V and all 5421 supplies link in the power complete sequence chain.

### **K11** +60V Sense (YA102A)

- Pick
  - +60V supply in 5421 up to approximately +60V
- Drop
  - Loss of +60V output
- Function
  - K11-1
    - a. Provides a path to hold K3 energized after a thermal fault or setting the POWER switch OFF.
    - b. Provides a path to supply +24V to regulator for main storage supply sequence/sense circuits.
  - K11-2
    - a. Provides a path for +24V to TP8 when +60V in 5421 is not up. For +24V to appear on TP8, all -4V supplies, -12V/  $\pm 12V$  supply,  $\pm 3V$  2560 supply (if installed), +6V supply, +8.5V supply, +24V in 5415/5424, and all 5421 supplies must be up.

*Note:* Because loss on any 5421 supply that includes +60V also controls TP7, TP8 should never indicate +24V unless the sense circuit itself is failing.

- K11-2
  - b. Provides the +60V link in the power complete sequence chain.

*Note:* While it appears that K10 and K11 have duplicate functions, the holding on of K3 until +60V to the MFCU is down (K11-1 n/o) is sufficient justification for K11 being present.

## COIL/RELAY FUNCTIONAL DESCRIPTIONS

### K12 Power Sequence Complete

- Pick
  - All  $-4\text{V}$  power supplies sensed up (K5, K17, K19),  $-12\text{V}$  or  $\pm 12\text{V}$  power supply sensed up (K30, K2 in MLTA supply),  $\pm 3\text{V}$  power supply sensed up (K12),  $+6\text{V}$  power supply sensed up (K6),  $+8.5\text{V}$  power supply sensed up (K8),  $+24\text{V}$  in 5415/5424 and all 5421 supplies sensed up (K10), and  $+60\text{V}$  power supply in 5421 sensed up (K11).
- Drop
  - Loss of any of the above supply outputs
  - OV/OC fault on  $-4\text{V}$  or  $+6\text{V}$  levels
  - UV fault on  $-4\text{V}$  level
- Function
  - K12-1
    - a. Provides power sequence complete to 5421 and 5424.
    - b. Provides  $+24\text{V}$  to 2560 switch lines (2).
    - c. Provides control line to 3411.
  - K12-2
    - a. Provides  $+24\text{V}$  to PWR CHK light with POWER switch ON and K12 not picked, or with POWER switch OFF though K12 not picked and  $-4\text{V}$  No. 1 supply (K5) still up.
  - K12-3
    - a. Enables picking of  $+6\text{V}$  UV fault relay (K15) but only after power sequence is complete (inhibits a  $+6\text{V}$  UV fault during a normal power up and power down sequence).
  - K12-4
    - a. Disables 'power on reset' signal to CPU (clock 9 no longer on).

### K13 $-4\text{V}$ No. 1 (A gate) OV/OC Fault

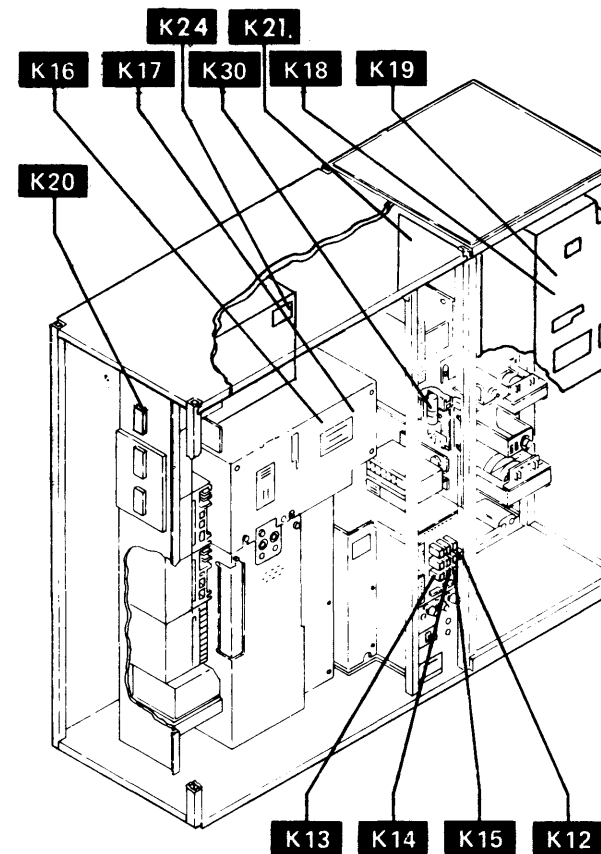
- Pick
  - An overvoltage or overcurrent condition detected by  $-4\text{V}$  No. 1 (A gate) regulator.
- Drop
  - Pressing CHECK RESET with the POWER switch OFF.

- Function
  - K13-1
    - a. Provides a path for  $+24\text{V}$  to TP12.
    - b. Provides control of distribution of  $+24\text{V}$  to all sequence control circuits.
  - K13-2
    - a. Provides a hold path for K13 requiring a manual reset to clear the fault indicator.

### K14 $-4\text{V}$ No. 1 (A gate) UV or $+6\text{V}$ ( $+5\text{V}$ for BSCC only) OV/OC Fault

- Pick
  - An overvoltage or overcurrent condition detected by the  $+6\text{V}$  regulator.
  - An undervoltage condition detected by the  $-4\text{V}$  No. 1 (A-gate) UV sense circuit.
- Drop
  - Pressing CHECK RESET with the POWER switch OFF.
- Drop
  - Pressing CHECK RESET with the POWER switch OFF.
- Function
  - K14-1
    - a. Provides a hold path for K14 requiring a manual reset to clear the fault indicator.
  - K14-2
    - a. Provides a path for  $+24\text{V}$  to TP13.
    - b. Provides control of distribution of  $+24\text{V}$  to all sequence control circuits.
  - K14-3 not used.
  - K14-4
    - a. Inhibits picking of K15 when K14 is picked.

*Note:* K14-4 and K15-1 are used to prevent picking two fault relays when a power fault (OV/OC) occurs. When the system abruptly powers down and K3 is dropped, a race condition (to power down) exists for the  $-4\text{V}$  and  $+6\text{V}$  supplies. K14-4 and K15-1 eliminate erroneous test point indications.



### K15 $+6\text{V}$ ( $+5\text{V}$ for BSCC only) UV Fault

K15 is used as a  $+6\text{V}$  UV detection sense relay and as a  $\pm 3\text{V}$  sequence sense (2560 feature) relay.

- Pick
  - A UV condition causing K6A-1 n/o points to close with power complete (K12-3 n/o points closed) and no  $-4\text{V}$  UV fault or  $+6\text{V}$  OV/OC fault (K14-4 n/c points closed).
- Drop
  - Pressing CHECK RESET with the POWER switch OFF.
- Function
  - K15-1
    - a. Inhibit  $+24\text{V}$  to TP7 on  $+6\text{V}$  UV condition.
  - K15-2
    - a. Provides a path for  $+24\text{V}$  to TP14.
  - K15-3
    - a. Provides a hold path for K15 requiring a manual reset to clear the fault indicator.
  - K15-4
    - a. Provides control of  $+24\text{V}$  to K9, K9A, and K9B coils and power up signal to 5421.

### K16 $-4\text{V}$ No. 2 OV/OC Fault

- Pick
  - An overvoltage or overcurrent condition detected by  $-4\text{V}$  No. 2 regulator.
- Drop
  - Pressing CHECK RESET with POWER switch OFF.
- Function
  - K16-1
    - a. Provides a path for  $+24\text{V}$  to TP11.
    - b. Provides control of distribution of  $+24\text{V}$  to all sequence control circuits.
  - K16-2
    - a. Provides a hold path for K16 requiring a manual reset to clear the fault indicator.

### K17 $-4\text{V}$ No. 2 Sense and $-5\text{V}$ Sense

- Pick
  - $-4\text{V}$  No. 2 output at approximately  $-4\text{Vdc}$  and K1 picked due to  $-5\text{V}$  supply being up
- Drop
  - Loss of  $-4\text{V}$  No. 2 output
  - Loss of  $-5\text{V}$  output
- Function
  - K17-1
    - a. Provides the  $-4\text{V}$  No. 2 link in the start-up control for the  $+6\text{V}$  regulator.
  - K17-2
    - a. Provides a path for  $+24\text{V}$  to TP3 when  $-4\text{V}$  No. 2 level is not up. For  $+24\text{V}$  to appear on TP3,  $-4\text{V}$  No. 1 (A gate) and  $-4\text{V}$  No. 3 supplies must be up.
    - b. Provides the  $-4\text{V}$  No. 2 link and a  $-5\text{V}$  link in the power complete sequence chain

### K18 $-4\text{V}$ No. 3 OV/OC Fault

- Pick
  - An overvoltage or overcurrent condition detected by  $-4\text{V}$  No. 3 regulator.
- Drop
  - Pressing CHECK RESET switch with POWER switch OFF.

- Function
  - K18-1
    - a. Provides a path for  $+24\text{V}$  to TP10.
    - b. Provides control of distribution of  $+24\text{V}$  to all sequence control circuits.
  - K18-2
    - a. Provides a hold path for K18 requiring a manual reset to clear the fault indicator.

### K19 $-4\text{V}$ No. 3 Sense

- Pick
  - $-4\text{V}$  No. 3 output at approximately  $-4\text{Vdc}$
- Drop
  - Loss of  $-4\text{V}$  No. 3 output
- Function
  - K19-1
    - a. Provides the  $-4\text{V}$  No. 3 link in the start-up control for the  $+6\text{V}$  regulator.
  - K19-2
    - a. Provides a path for  $+24\text{V}$  to TP4 when  $-4\text{V}$  No. 3 level is not up. For  $+24\text{V}$  to appear on TP4,  $-4\text{V}$  No. 1 (A gate) supply must be up.
    - b. Provides the  $-4\text{V}$  No. 3 link in the power complete sequence chain.

### K20 B Gate FET Protect

- Pick
  - $-4\text{V}$  No. 2 power supply output at approximately  $-4\text{Vdc}$ .
- Drop
  - Loss of  $-4\text{V}$  No. 2 power supply output.
- Function
  - K20-1
    - a. Prevents basic main storage power supply from operating if the  $-4\text{V}$  No. 2 power supply does not power up or if it powers down abruptly.

## COIL/RELAY FUNCTIONAL DESCRIPTIONS

### **K21** B Gate FET Protect for No. 3 Feature Power Supply

- Pick
  - -4V no. 3 feature power supply output at approximately -4 Vdc
- Drop
  - Loss of -4V no. 3 feature power supply output.
- Function
  - K21-1
    - a. Prevents basic main storage power supply from operating if -4V No. 3 power supply does not power up or if it powers down abruptly.
    - b. Installed only if +8.5V and +3.4V are used on channel bank no. 3.

### **K24** Expansion Main Storage Power Supply (Feature)

- Pick
  - +8.5V expansion main storage power supply powered up.
- Drop
  - Loss of +8.5V expansion main storage power supply output.
- Function
  - Prevents the basic main storage power supply from powering up until after the expansion main storage power supply has powered up

### **K30** -12V Supply Sense (BSCA/BSCC)

- Pick
  - The BSCA/BSCC -12V supply output is at approximately -12 Vdc.
- Drop
  - Loss of -12V supply output
- Function
  - K30-1
    - a. Provides a path for +24V to TP9 when the -12V supply is not up.
    - b. Provides the -12V link in the power complete sequence chain.

*Note:* If MLTA and BSCA features are installed, a  $\pm 12V$  supply is installed instead of the -12V supply. In that case, K2 is used to sense its output.

This page intentionally left blank

# APPENDIX A. CROSS REFERENCE

## FEALD – CARD CROSS-REFERENCE LIST

Function	Board/Card	FEALD	FETMD
A and B registers	B3F2	RA101	2-20, 2-21
ALU	B3H2	AV102	2-8
Aux ALU	B3Q2	AB101	2-23
ALU controls	B3M2	KY101	2-15
CE mode	B3V3	PA101	7-8
Channel bank 2 terminators	B2E4	MD101	4-005
	B2F4	MD111	---
Channel bank 3 terminators	B2E5	MD201	4-005
	B2F5	MD211	---
	B2G5	MD221	---
Channel entry/exit	---	WB101	4-100
Channel in	B2J2	KE101	4-115
Channel out	B2H2	KE201	4-107
Clock controls	B3T2	KC102	2-1
Clock indicators	B3V2	PB121	7-4
CPU to storage	---	WS010	---
Cycle controls	B3S2	KD101	2-2
DBO bank 2	B2F2	MC101	4-005
	B2E2	MC111	---
	B2D2	MC121	---
	B2C2	MC131	---
DBO bank 3	B2F3	MC201	4-005
	B2E3	MC211	---
	B2D3	MC221	---
	B2C3	MC231	---
Display selector drum	B2U5	PC101	7-5
Drum indicators	B2V5	PB101	7-4
External address gate	B3A3	MB101	---
Halt ID Circuits	B2K4	KT201	5-66
Halt ID Indicators	B2V4	PB131	7-3
Initial memory scan	B3U4	KA312	6-4
Instruction and program check control	B3G2	KN101	---
Interrupt card	B3L2	KM101	2-49
Interval timer	B2K3	KT601	2-52
Interval timer oscillator	B2K2	KT701	---
LSR card	B3C2	MA202	2-30
LSR controls	B3P2	KL101	2-34
	B3N2	KG101	2-34
LSR select bank 2	B2D4	KE301	2-36
LSR select bank 3	B2G2	KE401	2-37
LSR select and operator indicators	B3V4	PC111	7-7
Machine cycle indicators	B3U2	PB111	7-4
Main storage	A4/B4	UJ011	3-1
Meter control	B3U3	CS101	---
Miscellaneous console	B3V5	PC121	7-1
Op and Q registers	B3J2	RN101	2-41, 2-44
Oscillator	B2S5	KA101	2-1
PMR	B3E2	KP102	2-28
Processor check latches and display	B3K2	KB101	1-8

Function	Board/Card	FEALD	FETMD
Program check registers	B3D2	KR201	2-46
Rotary bit switches	B2V3	PA111	7-7
Run controls	B3R2	KA202	2-4
SAR card	B3B2	MA102	2-54
Store data LSR to assembler to A register	B3D2	KR242	2-48
Timer	B2M2	KT501	2-52
Use meter	B2S4	CR101	---

LOGIC CARD REFERENCE

Board A-B2 (pin side)

	1	2	3	4	5	6
A	Channel 1 Out	Channel 2 Conn Out	Channel 3 Conn Out	To B3A2	Spare	
B	Channel 1 Out	Channel 2 Conn Out	Channel 3 Conn Out	Channel 2 Conn In	Channel 3 Conn In	
C	Channel 1 Out	Channel 2 Out TLD	Channel 3 Out TLD	Channel 2 Conn In	Channel 3 Conn In	
D	Channel 1 Out	Channel 2 Out TLD	Channel 3 Out TLD	LSR Select Bank 2	Spare	
E	Channel 1 Out	Channel 2 Out TLD	Channel 3 Out TLD	Channel 2 In TLD	Channel 3 In TLD	
F	Channel 1 Out	Channel 2 Out TLD	Channel 3 Out TLD	Channel 2 In TLD	Channel 3 In TLD	
G	Channel 1 In	LSR Select Bank 3		Channel 2 In TLD	Channel 3 In TLD	
H	Channel 1 In	Channel Out				
J	Channel 1 In	Channel In				
K	Channel 1 In	Timer and Unit Record Restart		Halt ID Indicator		
L	Channel 1 In	Timer				
M	Channel 1 In	Timer				
N	Channel 1 In	Timer				
O	Channel 1 In	Timer				
P	Channel 1 In	Timer				
Q	Channel 1 In	Timer				
R	Channel 1 In	Timer				
S	Channel 1 In	Timer Oscillator		Use Meter 1	Oscillator	
T	Y6					
U	Y6				Switch	Z6
V	Y6		Console Switches	P1	Drum (roller) Indicator	Z6
	1	2	3	4	5	6

Board A-B3 (pin side)

	1	2	3	4	5	6
A		To B2A4	Rem. SAR Top Connector Card	Dual Byte Feature Cable		
B		SAR				To A4V2
C		LSR				To A4V2
D		Program Check				
E		PMR				To A4V3
F		A and B Registers				To A4V3
G		Instruction Control				
H		ALU				To A4V4
J		Op and Q Registers, Op Decode				To A4V4
K		Display and Check				
L		Interrupt				
M		ALU Control				To A4V5
N		Register Control				To A4V5
P		LSR Control				
Q		Auxiliary ALU				
R		Run Control				
S		Cycle Control				
T		Clock Control				
U	Y6	Machine Cycle Light	Use Meter 2	Initial Memory Scan		Z6
V	Y6	Clock Light	Toggle Switch	LSR Switch	Misc Switches and Lights	Z6
	1	2	3	4	5	6



# Index

## A

A (add to register) 5-46  
A cycles 5-14  
A-register and controls 2-20  
A/B register parity checking 2-22  
ac power 8-2  
add logical characters (ALC) 5-15  
add to register (A) 5-46  
add zoned decimal (AZ) 5-20  
add-on regulator 8-11  
address bit generation, I/O LSR 2-38  
address bit switch 7-9  
address compare light 7-7  
address compare switches 7-10  
    E cycle stop 7-10  
    I cycle stop 7-10  
address increment switch 7-9  
address parity check 2-57  
address selection, main storage 3-8  
address translate table 2-54  
address violation check 2-56  
addressing 1-12  
addressing main storage checks 3-6, 3-8  
addressing, BSM 3-7  
addressing, main storage 2-54  
advance program level (APL) 5-80  
ALC (add logical characters) 5-15  
alter ATT/PMR 6-12  
alter SAR 6-14  
alter storage 6-5  
ALU  
    auxiliary 2-23  
    card check, auxiliary 2-24  
    controls 2-15  
    description 2-8  
    parity checking 2-18  
    P-bit generation 2-12  
APL (advance program level) 5-80  
auxiliary ALU 2-23  
auxiliary ALU card check 2-24  
auxiliary B register 2-25  
AZ (add zoned decimal) 5-20

## B

B cycles 5-14  
B gate -4V feature logic supply no. 3 8-12  
B gate -4V logic supply no.2 8-12  
B register, auxiliary 2-25  
B-register and controls 2-21

basic -4Vdc no. 1 supply 8-11  
basic unit, power distribution 8-3  
BC (branch on condition) 5-51  
binary  
    addition 2-11  
    subtraction 2-10  
branch on condition (BC) 5-51  
branching 1-13  
BSCA step key 7-7  
BSCA/LCA control panel  
BSCA/LCA local test switch 7-9  
BSM addressing 3-7  
BSM write operation 3-4  
bulk power supplies  
    no. 1, 2, and 3 8-6  
    +6V expansion 8-13  
bypass resistors 8-11  
byte control, main storage 3-6

## C

card check, auxiliary ALU 2-24  
carry check, ALU 2-19  
CCP (command CPU) 5-67  
CE controls 7-7  
CE switches  
    key switch 7-7  
    mode selector switch 7-8  
    channel 4-005, 4-010  
    channel DBO parity 4-120  
    channel interface 4-005, 4-010  
    channel lines 4-100  
    channel terminators 4-005  
    check ALU 2-17  
    extended SAR address bits 2-55, 7-9  
channel terminate feature 1-5  
check bit generation, main storage 3-6, 3-10  
check reset key 7-7  
checks and adjustments 8-3  
CLC (compare logical characters) 5-16  
clock indicators 7-4  
coil/relay functional description 8-26  
command CPU (CCP) 5-67  
command instructions 5-63  
compare logical characters (CLC) 5-16  
condition register 2-23  
console switches 4-110  
conversion  
    decimal to hexadecimal 1-10  
    hexadecimal to decimal 1-10

## CPU

clock 2-1  
data flow 1-8  
    DBO parity 4-125  
    LSRs 2-28  
    timing 1-7  
cycle controls 2-2  
cycle steal priority 5-72

## D

data flow  
    CPU 1-8  
    main storage 3-3  
data format 1-10  
data transfer, I/O cycle 5-71  
DBI and translator 4-115  
DBI parity 4-130  
DBI translator 4-095  
DBO  
    parity 4-120, 4-125  
    translator 4-095  
DBO and translator 4-107  
dc bulk power supplies 8-6  
dc power 8-2  
decimal  
    addition 2-13  
    subtraction 2-11  
display adapter 1-5  
display check bit switch 7-9  
display SAR/MSAR 6-18  
display storage 6-8

## E

ECC (error checking and correction) 3-10  
ED (edit) 5-26  
edit (ED) 5-26  
emergency power off (EPO) switch 7-1  
EPO switch 7-1  
error checking and correction (ECC) 3-10  
error classification logic, main storage 3-11  
errors 1-7  
execute cycles 5-14  
expansion main storage power supply 8-16

## F

fast I cycles 1-7  
fault relays 8-9, 8-30

FEALD/card cross reference list A-1  
fetch data register and control 3-9  
file control panel 7-2

## H

halt program level (HPL) 5-66  
HPL (halt program level) 5-66

## I

I cycles 5-3  
I-op cycle 5-3  
I/O  
    attention light 7-3  
    check light 7-7  
    check switch 7-9  
    cycle 5-70  
    cycle data transfer 5-71  
    cycle steal 1-13  
    instructions 5-68  
    interface 4-005, 4-010  
    interface lines 4-100  
    LSRs 2-32  
    LSR select check 2-40  
    LSR 17th and 18th bit 2-55  
    overlap switch 7-9  
IH cycles 5-7  
IL cycles 5-7  
indexing 5-10  
initial memory scan 6-4  
initial program load (IPL) 6-1  
input power requirements 8-3  
insert and test characters (ITC) 5-30  
instruction and signal reference 5-1  
instruction cycles (fast I cycles) 5-2  
instruction format 1-11  
interrupt  
    priority 2-49  
    not ready to ready 2-59  
    request 2-51  
    routines 1-13  
interrupt level light 7-4  
interrupt mask 2-49  
interrupts 2-49  
interval timer 2-52  
introduction  
    processor 1-6  
    system 1-1  
invalid address check 2-58

invalid device address 4-120  
invalid op code check 2-43  
invalid Q 4-120  
IPL (initial program load) 6-1  
IQ cycle 5-3  
IR cycle 5-63  
ITC (insert and test characters) 5-30  
IX cycles 5-10

## J

JC (jump on condition) 5-65  
jump on condition (JC) 5-65

## K

key switch, CE 7-7

## L

LA (load address) 5-50  
LCA/BSCA control panel 7-6  
LCP (load CPU) 5-53  
LIO (load I/O) 5-73  
Load ATT flowchart 5-56  
load address (LA) 5-50  
load CPU (LCP) 5-53  
load I/O (LIO) 5-73  
load, store CPU flowcharts 5-54  
local storage registers (LSR) 2-30  
locations, power supply 8-4  
logic card reference A-2  
LSR (local storage registers)  
    addressing 2-33  
    controls 2-39  
    description 2-30  
    display selector switch 7-7  
    select  
        check (I/O) 2-40  
        CPU 2-34  
        I/O address bit generation 2-38  
        I/O channel bank 1 2-35  
        I/O channel bank 2 2-36  
        I/O channel bank 3 2-37  
        17th and 18th bit (I/O) 2-55  
LSRs  
    CPU 2-31  
    I/O 2-32

**M**

machine cycle indicators 7-4  
 machine language 1-9  
 main storage  
   addressing 2-54  
   address selection 3-8  
   checks, addressing 3-6, 3-8  
   data flow 3-3  
   introduction 3-1  
   locations 3-2  
   power supply 8-14  
     overvoltage/undervoltage protection 8-15  
     regulator card 8-15  
     sense relay 8-15  
     sequence card 8-15  
     sequencing 8-15  
     voltage adjustment 8-14  
   power supply (expansion) 8-16  
     overvoltage/undervoltage protection 8-17  
     regulator card 8-17  
     sense relay 8-17  
     sequence card 8-17  
     sequencing 8-17  
     voltage adjustment 8-16  
   read operation, BSM 3-5  
 memory  
   diagnostic fetch 5-53, 5-60  
   FDR reset 5-53, 5-60  
 message display unit 7-3  
 Model 10 compatibility 2-42  
 mode selector switch 7-8  
 move characters (MVC) 5-16  
 move hex character (MVX) 5-34  
 move logical immediate (MVI) 5-38  
 MVC (move characters) 5-16  
 MVI (move logical immediate) 5-38  
 MVX (move hex character) 5-34

**N**

not ready to ready interrupt 2-59  
 number  
   conversions 1-9  
   systems 1-9

**O**

one-address instructions description 5-38  
 op register 2-41  
 op- and Q-register parity 2-45  
 operator panel 7-1  
 overcurrent protection 8-9  
 overvoltage protection 8-9

**P**

P-bit generation, ALU 2-12  
 parity check switch 7-9  
 parts replacement (power) 8-3  
 PMR (program mode register) 2-28  
 power check light (lamp) 7-4, 8-28  
 power distribution 8-3, 8-5  
   +5 Vdc to 3277/3284 attachment 8-21  
   ±5 Vdc to BSCC 8-21  
 power off  
   abnormal 8-27, 8-29  
   emergency 8-29  
   OC/OV/UV failure 8-27, 8-29  
   sequence 8-26  
   thermal 8-29  
 power on  
   abnormal 8-27  
   sequence 8-24, 8-25, 8-29  
 power on/off switch 7-3  
 power supply  
   regulators 8-7  
     start up control 8-8  
     ±3 Vdc 8-18, 8-19  
     -4 Vdc (add-on regulator) 8-11  
     -4 Vdc basic no. 1 supply adjustment 8-11  
     -4 Vdc logic supplies no. 2 and no. 3  
       adjustments 8-12  
     -4 Vdc undervoltage circuit (regulator  
       no. 1) 8-10  
     ±5 Vdc feature regulator (BSCC) 8-20  
     +6 Vdc expansion bulk supply (feature) 8-13  
     +6 Vdc regulator adjustment 8-13  
     +6 Vdc undervoltage circuit 8-10  
     ±12 Vdc supply (feature) 8-21  
     -12 Vdc supply (feature) 8-21  
     +24 Vdc supply 8-23  
 privileged mode 2-42  
 privileged op check 2-43  
 processor characteristics 1-6  
 processor check  
   A-B register P check 2-22  
   address violation check (store protect) 2-58  
   ALU P check 2-18  
   carry check 2-19  
   channel DBO P check 4-120  
   channel I/O check 1-7  
   CPU DBO P check 4-125  
   CPU SAR check (SAR/ATT) 2-57  
   DBI P check 4-130  
   DBO P check (CPU DBO) 4-125  
   description 1-7  
   invalid address 2-58  
   invalid device address check (invalid Q) 4-120  
   invalid op check 2-43  
   invalid Q (invalid device address) check 4-120

I/O LSR select check 2-40  
 light 7-3  
 LSR parity check 2-31  
 memory SAR P check 2-57  
 op/Q register P check 2-45  
 privileged op check 2-43  
 SAR/ATT (CPU SAR check) 2-57  
 SDBI check 3-13  
 SDBO check 3-13  
 store protect (address violation) 2-58  
 program check  
   address violation check (store protect) 2-58  
   assembler 2-48  
   description 1-7  
   interrupt 2-49  
   invalid address 2-58  
   invalid device address check (invalid Q) 4-120  
   invalid op check 2-43  
   privileged op check 2-43  
   registers 2-46  
 program load selector switch 7-2  
 program mode register (PMR) 2-28

**Q**

Q register parity 2-45

**R**

read operation, main storage 3-5  
 recomplement 2-14  
 register display unit 7-5  
 roller display 7-5  
 rotary address switch 6-9, 6-15  
 run controls 2-4

**S**

SAR E14 and E15 bit 2-55  
 SAR/MSAR display switch 7-9  
 SBF (set bits off masked) 5-42  
 SBN (set bits on masked) 5-42  
 SCP (store CPU) 5-52  
 SDBI main storage checks 3-13  
 SDBO main storage checks 3-13  
 select check (I/O LSR) 2-40  
 sense I/O (SNS) 5-76  
 sequence relays (sensing) 8-30  
 sequential instruction execution 1-12  
 set bits off masked (SBF) 5-42  
 set bits on masked (SBN) 5-42  
 seventeenth and eighteenth bit (I/O LSR) 2-55

**V**

SLC (subtract logical characters) 5-15  
 SNS (sense I/O) 5-76  
 ST (store register) 5-46  
 start I/O (SIO) 5-68  
 start key 7-3  
 stop key/light 7-3  
 storage  
   protection 2-56  
   test switch 7-9  
 storage protect check 2-58  
 store ATT flowchart 5-55  
 store CPU (SCP) 5-52  
 store data assembler 2-48  
 store register (ST) 5-46  
 store, load CPU flowchart 5-54  
 subtract logical characters (SLC) 5-15  
 subtract zoned decimal (SZ) 5-20  
 supervisor call 5-67  
 supervisor program 2-50  
 syndrome bit  
   decoder 3-10  
   generator 3-10  
 system console 7-1  
 system reset  
   description 6-2  
   key 7-7  
 system sequencing and sensing relays 8-29  
 SZ (subtract zoned decimal) 5-20

**T**

TBF (test bits off masked) 5-42  
 TBN (test bits on masked) 5-42  
 test bits off masked (TBF) 5-42  
 test bits on masked (TBN) 5-42  
 test I/O and branch (TIO) 5-79  
 test points (TPs), power supply 8-28  
 thermal check 8-27, 8-28  
 thermal check light 7-4  
 TIO (test I/O and branch) 5-79  
 two-address instruction 5-3

**U**

uncorrectable error (main storage) 3-11  
 undervoltage protection 8-10  
 unit check 1-7  
 unit record restart 2-59  
 usage meter 7-1

**W**

write operation, main storage BSM 3-4

**Z**

ZAZ (zero and add zoned) 5-21  
 zero and add zoned (ZAZ) 5-21

READER'S COMMENT FORM

Please use this form only to identify publication errors or request changes to publications. Technical questions about IBM systems, changes in IBM programming support, requests for additional publications, etc, should be directed to your IBM representative or to the IBM branch office nearest your location.

Error in publication (typographical, illustration, and so on). No reply.

Inaccurate or misleading information in this publication. Please tell us about it by using this postage-paid form. We will correct or clarify the publication, or tell you why a change is not being made, provided you include your name and address.

Page Number Error

Page Number Comment

Check if reply is requested.

Note: All comments and suggestions become the property of IBM.

Name \_\_\_\_\_

Address \_\_\_\_\_

• No postage necessary if mailed in the U.S.A.

READER'S COMMENT FORM

Please use this form only to identify publication errors or request changes to publications. Technical questions about IBM systems, changes in IBM programming support, requests for additional publications, etc, should be directed to your IBM representative or to the IBM branch office nearest your location.

Error in publication (typographical, illustration, and so on). No reply.

Inaccurate or misleading information in this publication. Please tell us about it by using this postage-paid form. We will correct or clarify the publication, or tell you why a change is not being made, provided you include your name and address.

Page Number Error

Page Number Comment

Check if reply is requested.

Note: All comments and suggestions become the property of IBM.

Name \_\_\_\_\_

Address \_\_\_\_\_

• No postage necessary if mailed in the U.S.A.

Cut Along Line

Fold

Fold

Fold

Fold

FIRST CLASS  
PERMIT NO. 387  
ROCHESTER, MINN.

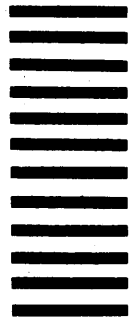
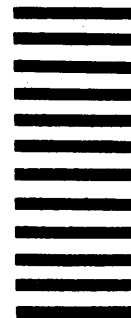
FIRST CLASS  
PERMIT NO. 387  
ROCHESTER, MINN.

**BUSINESS REPLY MAIL**  
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

**BUSINESS REPLY MAIL**  
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

POSTAGE WILL BE PAID BY . . .



IBM Corporation  
General Systems Division  
Development Laboratory  
Publications, Dept. 245  
Rochester, Minnesota 55901

IBM Corporation  
General Systems Division  
Development Laboratory  
Publications, Dept. 245  
Rochester, Minnesota 55901

Fold

Fold

Fold

Fold



International Business Machines Corporation  
General Systems Division  
5775D Glenridge Drive N.E.  
Atlanta, Georgia 30301  
(U.S.A. Only)

General Business Group/International  
44 South Broadway  
White Plains, New York 10601  
U.S.A.  
(International)



International Business Machines Corporation  
General Systems Division  
5775D Glenridge Drive N.E.  
Atlanta, Georgia 30301  
(U.S.A. Only)

General Business Group/International  
44 South Broadway  
White Plains, New York 10601  
U.S.A.  
(International)

System  
Maintenance  
Library



System


— cut here —



International Business Machines Corporation  
General Systems Division  
5775D Glenridge Drive N.E.  
Atlanta, Georgia 30301  
(USA Only)

General Business Group/International  
44 South Broadway  
White Plains, New York 10601  
U.S.A.  
(International)