



IBM

Field Engineering

Maintenance Diagrams

2020 Processing Unit

System/360 Model 20

(Machines with serial no. 50,000 and above)

Volume 2

Preface

This publication (Volume 2) and its companion publication (Volume 1, Form Y33-1024) constitute the Field Engineering Maintenance Diagrams manual for the IBM 2020 Processing Unit (machines with serial number 50,000 and above) in the IBM System/360 Model 20. Volume 2 contains operations information (Section 5) and Volume 1 contains information on the following:

- Diagnostic techniques (Section 1)
- Error conditions (Section 2)
- Data flow (Section 3)
- Functional units (Section 4)
- Power (Section 6)
- Microprograms (Appendix B)

Both volumes are used for maintenance, instruction, and recall.

The material in these volumes supplements the information contained in the following manuals:

1. Field Engineering Theory of Operation, *2020 Processing Unit, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1021.
2. Field Engineering Maintenance Manual, *2020 Processing Unit, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1035.

Associated Publications

The following Field Engineering Maintenance Diagrams manuals contain information on the features which may be installed on the 2020 Processing Unit:

1. *1403 Printer Models 2, 7, N1 Attachment Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1018.
2. *2152 Printer-Keyboard Attachment Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1026.
3. *2203 Printer Attachment Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1022.
4. *2520 Card Read Punch Attachment Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1028.

5. *2560 Multi-Function Card Machine Attachment Feature, 2501 Card Reader Attachment Feature, 1442 Card Punch Model 5 Attachment Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1033.
6. *Binary Synchronous Communications Adapter, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1039.
7. *Input/Output Channel Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1017.
8. *Storage Control Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1037.

Information on the serial I/O channel feature is contained in Field Engineering Theory of Operation, Maintenance Diagrams, *Serial I/O Channel Attachment Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1040.

The associated Field Engineering Theory of Operations manual for the features are:

1. *1403 Printer Models 2, 7, N1 Attachment Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1020.
2. *2152 Printer-Keyboard Attachment Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1025.
3. *2203 Printer Attachment Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1027.
4. *2520 Card Read Punch Attachment Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1029.
5. *2560 Multi-Function Card Machine Attachment Feature, 2501 Card Reader Attachment Feature, 1442 Card Punch Model 5 Attachment Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1034.
6. *Binary Synchronous Communications Adapter, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1038.
7. *Input/Output Channel Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1019.
8. *Storage Control Feature, System/360 Model 20 (Machines with serial no. 50,000 and above)*, Form Y33-1036.

Second Edition (November 1969)

This volume is a major revision of, and obsoletes, Y33-1042-0.

Changed diagrams are denoted by the symbol ● to the left of the caption, small changes being also indicated by vertical lines to the left of the changes.

Changes are continually made to the specifications herein; any such changes will be reported in subsequent revisions or FE Supplements.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Laboratories, Product Publications, Dept 784, 703 Boeblingen/Wuertt, P.O. Box 210, Germany.

© Copyright International Business Machines Corporation 1969

Contents

Operations

CPU Operations (Part 1)	5- 1
CPU Operations (Part 2) Microinstructions	5- 1
Microprogram List Explanation	5- 2

Microinstruction Charts

(Flowchart = Part 1; Timing Chart = Part 2)

RI Formats

Load Byte Intermediate (2 parts)	5- 3
Insert Byte Left (2 parts)	5- 4
AND-OR-Exclusive OR Immediate (2 parts)	5- 5
Add Immediate (2 parts)	5- 6
Test under Mask and Skip if Zero and Not Zero (2 parts)	5- 7
Translate and Branch Short (Direct Addressing) (2 parts)	5- 8
Translate and Branch Short (Indirect Addressing) (2 parts)	5- 9
Translate and Branch Long (Direct Addressing) (2 parts)	5-10
Translate and Branch Long (Indirect Addressing) (2 parts)	5-11

RD Formats

Load Halfword (Direct Addressing) (2 parts)	5-12
Load Halfword (Indirect Addressing) (2 parts)	5-13
Store Halfword (Direct Addressing) (2 parts)	5-14
Store Halfword (Indirect Addressing) (2 parts)	5-15
Branch and Store (Direct Addressing) (2 parts)	5-16
Branch and Store (Indirect Addressing) (2 parts)	5-17
Branch on Binary Zero-Minus-Plus, Address Check (Direct Addressing) (2 parts)	5-18
Branch on Binary Zero-Minus-Plus, Address Check (Indirect Addressing) (2 parts)	5-19
Branch Unconditional (Direct Addressing) (2 parts)	5-20
Branch Unconditional (Indirect Addressing) (2 parts)	5-21

FF Formats

Store Zone Register (2 parts)	5-22
Load Zone Register (2 parts)	5-23
Move Halfword and Split (DD) (2 parts)	5-24
Move Halfword and Split (DX) (2 parts)	5-25
Shift Left/Right and Move (DD) (2 parts)	5-26
Shift Left/Right and Move (DX) (2 parts)	5-27
Shift Left/Right and Move (XD) (2 parts)	5-28
Shift Left/Right and Move (XX)	5-29
Move Halfword (DD) (2 parts)	5-30
Move Halfword (DX) (2 parts)	5-31
Move Halfword (XD) (2 parts)	5-32
Move Halfword (XX, ALC) (2 parts)	5-33
Move Byte (DD) (2 parts)	5-34
Move Byte (DX) (2 parts)	5-35
Move Byte (XD) (2 parts)	5-36
Move Byte (XX, ALC) (2 parts)	5-37
Move Numeric/Zone (DD) (2 parts)	5-38
Move Numeric/Zone (DX) (2 parts)	5-39
Move Numeric/Zone (XD) (2 parts)	5-40

Move Numeric/Zone (XX, ALC) (2 parts)	5-41
Add/Subtract Halfword, Add/Subtract Halfword and Set Carry (DD) (2 parts)	5-42
Add/Subtract Halfword, Add/Subtract Halfword and Set Carry (DX) (2 parts)	5-43
Add/Subtract Halfword, Add/Subtract Halfword and Set Carry (XD) (2 parts)	5-44
Add/Subtract Halfword, Add/Subtract Halfword and Set Carry (XX, ALC) (2 parts)	5-45
AND-OR-Exclusive OR Byte or Halfword (DD) (2 parts)	5-46
AND-OR-Exclusive OR Byte or Halfword (DX) (2 parts)	5-47
AND-OR-Exclusive OR Byte or Halfword (XD) (2 parts)	5-48
AND-OR-Exclusive OR Byte or Halfword (XX, ALC) (2 parts)	5-49
Compare Logical Byte or Halfword (DD) (2 parts)	5-50
Compare Logical Byte or Halfword (DX) (2 parts)	5-51
Compare Logical Byte or Halfword (XD) (2 parts)	5-52
Compare Logical Byte or Halfword (XX, ALC) (2 parts)	5-53
Add/Zero and Add Packed Byte (DD) (2 parts)	5-54
Add/Zero and Add Packed Byte (DX) (2 parts)	5-55
Add/Zero and Add Packed Byte (XD) (2 parts)	5-56
Add/Zero and Add Packed Byte (XX, ALC) (2 parts)	5-57
Subtract Packed Byte/Perform Packed Complement (DD) (2 parts)	5-58
Subtract Packed Byte/Perform Packed Complement (DX) (2 parts)	5-59
Subtract Packed Byte/Perform Packed Complement (XD) (2 parts)	5-60
Subtract Packed Byte/Perform Packed Complement (XX, ALC) (2 parts)	5-61
Set Decimal Sign (DD) (2 parts)	5-62
Set Decimal Sign (DX) (2 parts)	5-63
Set Decimal Sign (XD) (2 parts)	5-64
Set Decimal Sign (XX) (2 parts)	5-65
Halt and Display Halfword (2 parts)	5-66

I/O Instructions

SENS CPU I/O (Direct Addressing) (2 parts)	5-67
SENS CPU I/O (Indirect Addressing) (2 parts)	5-68
SENS I/O (ALC) (2 parts)	5-69
Control I/O (Direct Addressing) (2 parts)	5-70
Control I/O (Indirect Addressing) (2 parts)	5-71
Control I/O (ALC) (2 parts)	5-72

MANOP Charts

(Flowchart = Part 1; Timing Chart = Part 2)

Storage Display/Scan (2 parts)	5-73
Storage Alter/Fill (2 parts)	5-74
Local Store Display (2 parts)	5-75
Local Store Alter (2 parts)	5-76
Initial Control Program Load	5-77
Storage Test Run 1 and 3 (Load Runs) (2 parts)	5-78
Storage Test Run 2 and 4 (Compare Runs) (2 parts)	5-79
CPU LOG IN (2 parts)	5-80

Cycle-Stealing Charts

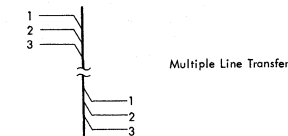
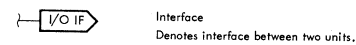
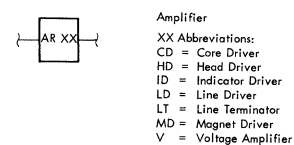
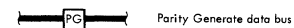
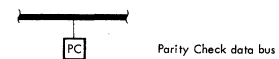
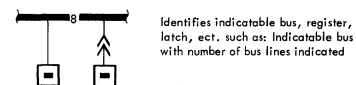
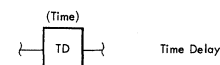
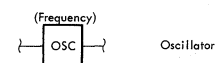
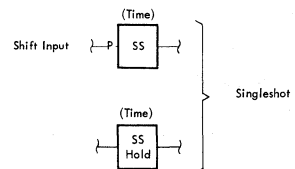
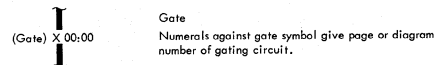
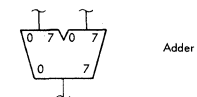
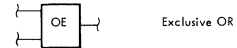
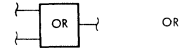
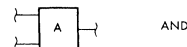
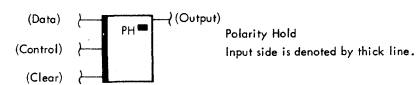
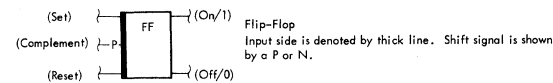
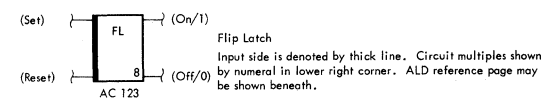
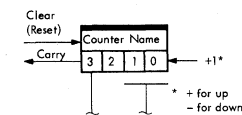
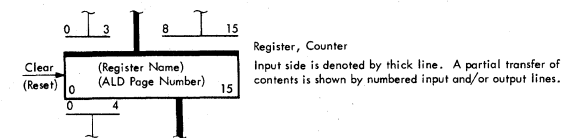
(Flowchart = Part 1; Timing Chart = Part 2)

CPU Cycle Steal Operation (2 parts)	5-81
-------------------------------------	------

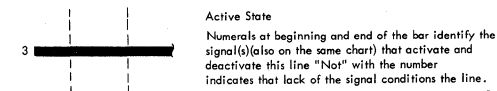
Note: The diagrams in this manual have a code number to the right of the caption. This is a publishing control number and is unrelated to the subject matter.

Legend

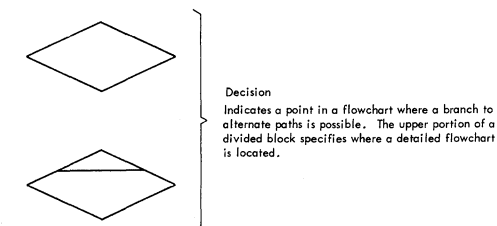
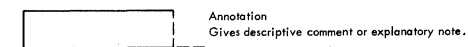
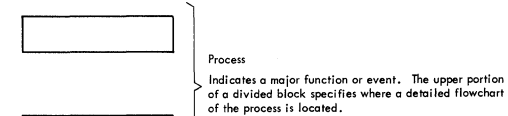
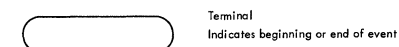
1. Logic Diagrams



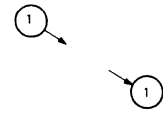
2. Timing Charts



3. Flowcharts



4. General



On-Page Connector
Indicates connection between two parts of the same diagram. Arrow leaving symbol points (line-of-sight) to correspondingly-numbered symbol.



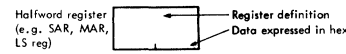
On-Page Connector
Indicates connection between two parts of the same diagram. Alphanumeric grid coordinate of complementary connector shown beneath.



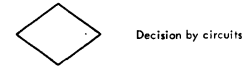
Off-Page Connector
Indicates connection between diagrams located on separate pages. Location of correspondingly-lettered symbol shown adjacent.

Diag 1-2

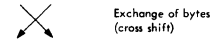
5. Special Symbols



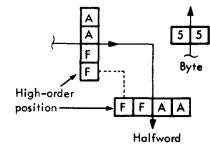
Halfword register (e.g. SAR, MAR, LS reg)
Register definition
Data expressed in hex



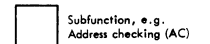
Decision by circuits



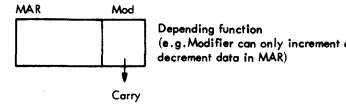
Exchange of bytes (cross shift)



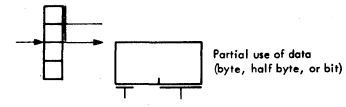
Information blocks showing which data (hex) is on a line, and an input or output of a functional unit



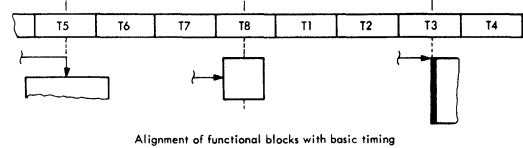
Subfunction, e.g. Address checking (AC)



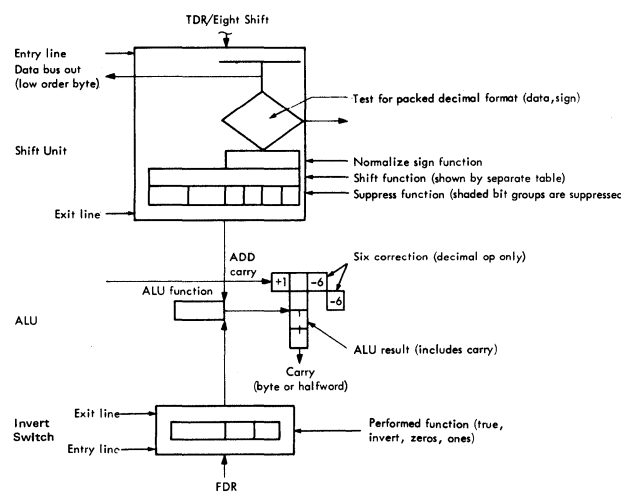
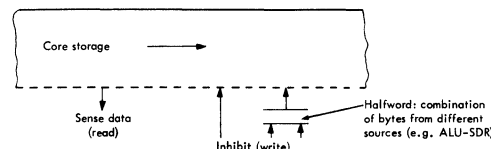
Depending function (e.g. Modifier can only increment or decrement data in MAR)



Partial use of data (byte, half byte, or bit)



Alignment of functional blocks with basic timing



Logical unit comprising shift unit, ALU, and invert switch. Functions not used within a specified operation are deleted. The whole logical unit is controlled by signals timed by cycles (eg. FDR true 0-15 during cycle 1). For charts representing more than one different operation, the different functions are repeated and identified by mnemonics

6. Standard Signal Arrangement

Signal grouping according to functional units	No	Name	ALD
LS zone selection	1	Sense Trap Request Lines	LA103
	2	LS New PL Zone Gate	KA511
	3	LS Current PL Zone Gate	
	4	New/Current PL	
	5	CE LS Select	CC222
Spare	6		
LS register selection, LS input and output control	7	Fixed X-Address	LA412
	8	'To Reg' Select	LA402
	9	'From Reg' Select	LA411
	10	LS to SAR	KB411
	11	LS to MAR	
	12	LS to FDR	KB401
	13	LS to TDR	
	14	MAR to LS	LA702-712
	15	Set ALU (I/O Bus) to LS	
	16	LS Write	LA302/313
Spare	17		
MAR/modifier control	18	Set Address Check	RA501
	19	Branch Go	RA502
	20	Increment by 1	RA402
	21	Increment by 2	RA403
	22	Decrement by 1	RA401
	23	Decrement by 2	RA402
	24	Prevent Mod-SAR-Inh Check	KA511
Spare	25		
SDR control	26	Prevent Storage Use	MA402
	27	SDR to Inh	MA401
	28	SDR to Op Reg	KB102
Spare	29		
TDR/shift unit control	30	SDR to TDR	KB402
	31	Eight Shift Control	RB162
	32	Shift by 2 or 4	RB161
	33	No Shift	RB162
	34	Test Packed Byte or Sign	RA502
	35	Normalize Sign Active	
36	Suppress	RB171	
Spare	37		
FDR/invert switch control	38	ALU to FDR	AA303
	39	Reset FDR/Retain FDR 0-7	AA303/KB411
40	Invert Switch Control	RB301	
Spare	41		
ALU control	42	ALU Control Gate	AA301
	43	Additional Carry	AA302
	44	Six Correction 8-11/12-15	
	45	Set Carry Latch	AA402
	46	Set Condition Code Latches	
	47	ALU to Inh	MA401
	48	ALU to SAR	KB411
Spare	49		
I/O bus control	50	Data Switch to Op Reg	KB402
	51	Op Reg to Address Bus	KA541
	52	I/O Display Address Out	
	53	Allow Strobe	
	54	SENS Strobe/Control Strobe	BA102
	55	Sense Reset/CTRL Strobe	
	56	Prevent ALU and SU Check	BA103
	57	I/O Bus to FDR	
Spare	58		
Spare	59		
Spare	60		
Spare	61		
Spare	62		
Spare	63		
Spare	64		
Spare	65		
Spare	66		
Set CPU checks	67	Set ALU Test Latch	CC121
	68	Set Process Check	CC122
	69	Set LSA Check	CC221
	70	Set Mod Check	CC101
	71	Set SU Check	CC102
	72	Set ALU Check	
	73	Set Bus Check	
	74	Set SAR Check	CC101
	75	Set Inh Check	

Combined signal which shows when new and current PL may differ (higher priority trap request). Depending upon the zone gates, the last instruction is finished in the currently selected LS zone while the next instruction is already initiated using the new LS zone.

Combined signal. The number above the signal shows which LS register is addressed.

Combined signal comprising: TDR 0-7 to SU 0-7 } No shift 8
TDR 8-15 to SU 8-15 } Shift left by 4
TDR 0-7 to SU 8-15 } Shift right by 2
TDR 8-15 to SU 0-7 } Cross shift

Combined signal comprising: Shift left by 2
Shift left by 4
Shift right by 2
Shift right by 4

Combined signal No shift 0-7 comprising: No shift bit 8-15 } No shift 0-15

Combined signal comprising: Suppr SU bit 0-7
Suppr SU bit 0-3
Suppr SU bit 4-7
Suppr SU bit 8-11
Suppr SU bit 12-15
Suppr SU bit 8-9
Suppr SU bit 10-11
Suppr SU bit 12-13
Suppr SU bit 14-15

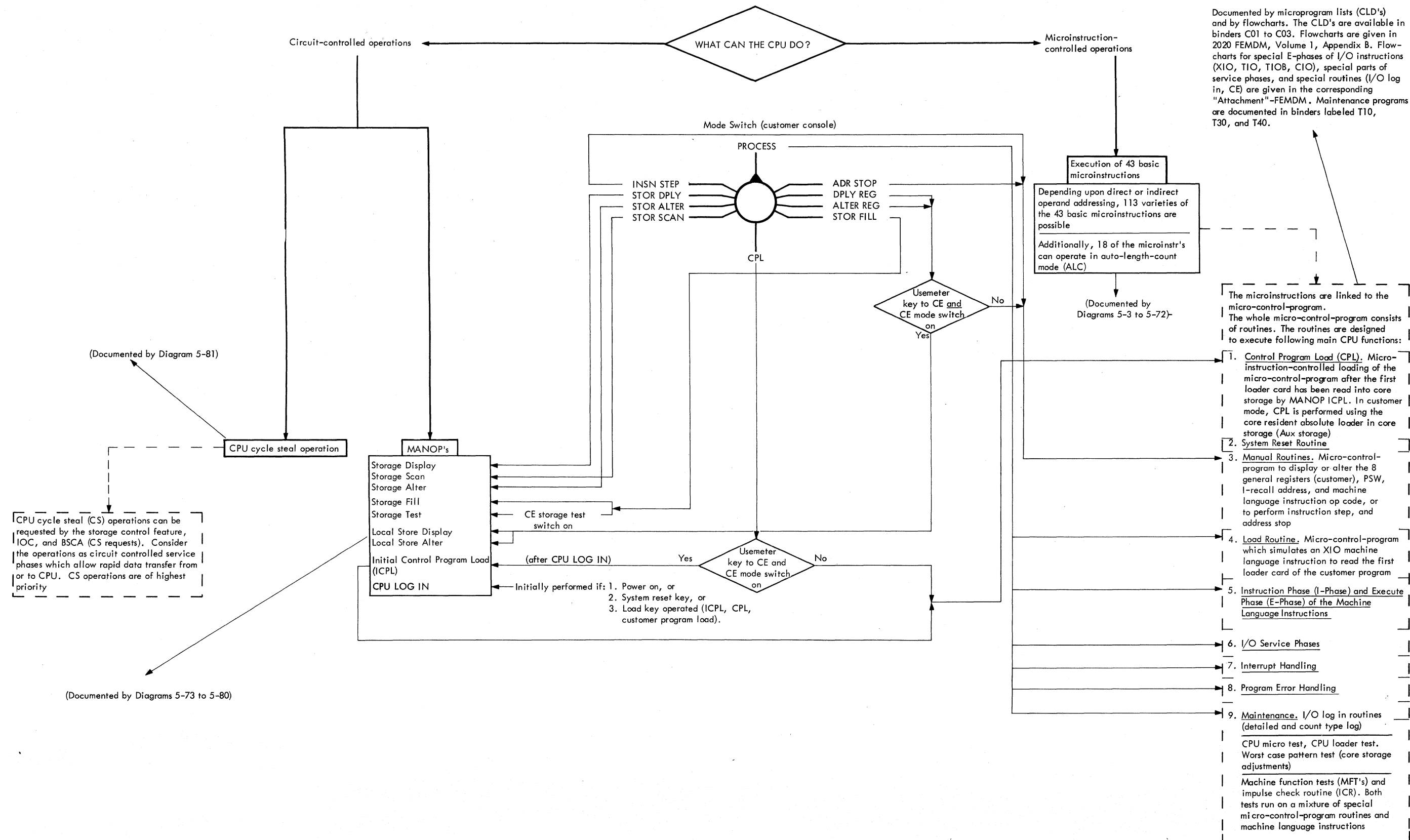
Combined signal comprising: FDR true 0-15
FDR invert 8-15 } True = true
FDR invert 0-7 } Invert = invert
FDR invert 0-7 is forced by FDR invert 8-15. Invert and true = force zeros
If FDR invert 0-7 is not required, it is } Not invert not true = force ones suppressed.

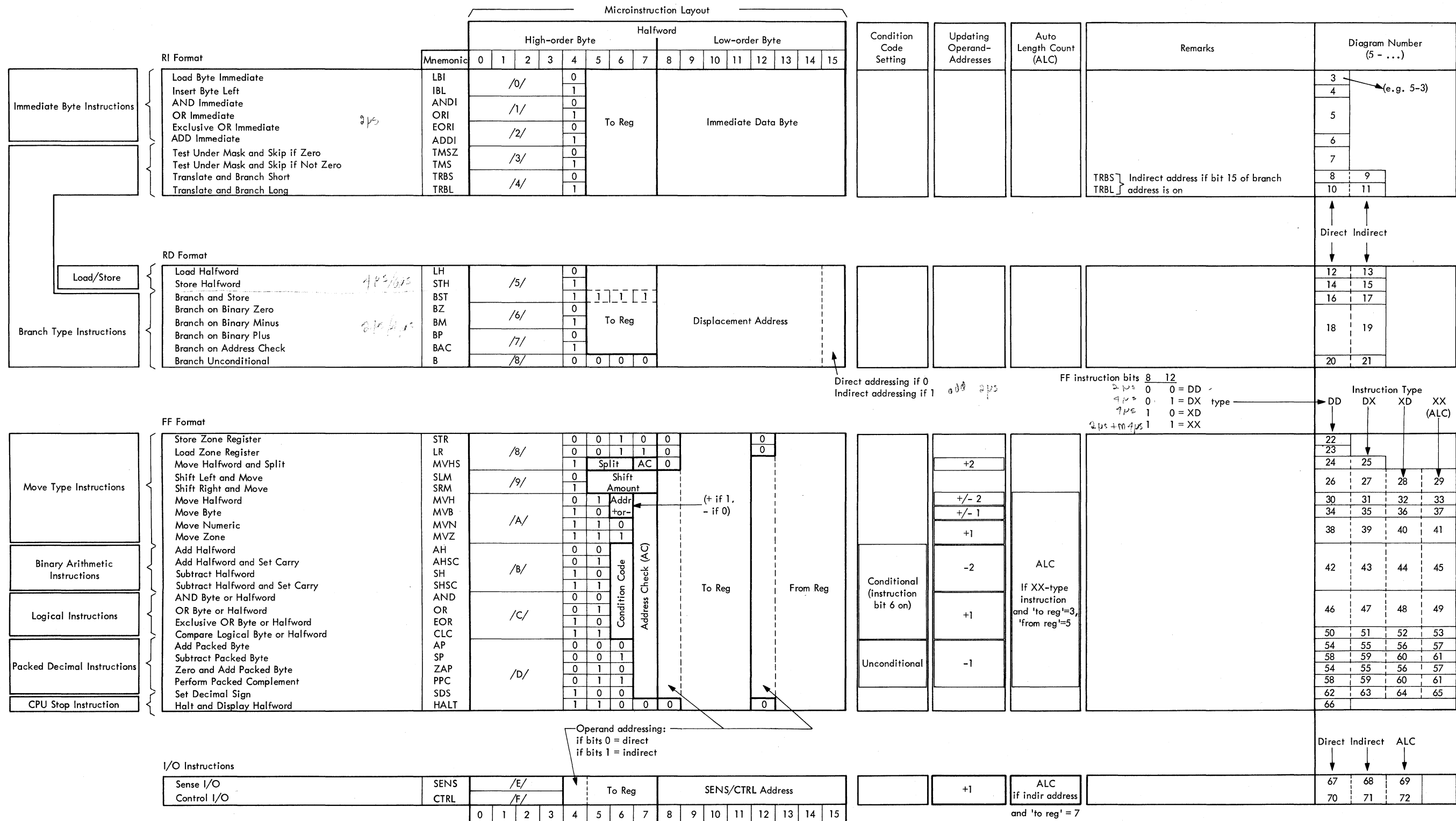
Combined signal comprising: Adder gate
AND gate
OR gate
OE = no gate active

Signals prefixed by "Set" are internal card signals which cannot be measured. They indicate when the specified function is executed. Reference to a single ALD page or to the first of a group of pages.

Abbreviations

AC	Address Check	MANOP	(Circuit-Controlled) Manual Operation
ADDR, Addr, ADR	Address	MAR	Modify Address-Register
ALC	Auto Length Count	MFT	Machine Function Test
ALU	Arithmetic and Logic Unit	Mnem	Mnemonic
ASCII	American Standard Code for Information Interchange		
Aux	Auxiliary		
		NSI	Next Sequential Instruction
BSCA	Binary Synchronous Communications Adapter		
		OE	Exclusive OR
CC	Condition Code	Op	Operation
CE	Customer Engineer	Op Reg	Operation Register
Chk, Ck	Check	Oprnd	Operand
CLD	Control Logic Diagram (microprogram list)		
Col Bin	Column Binary	PL	Program Level
CPL	Control Program Load	Pri	Primary
CPU	Central Processing Unit	PSW	Program Status Word
CS	Cycle Steal		
Cust	Customer		
CY	Cycle	Rd	Read
		Reg	Register
Diag	Diagnostic, Diagram	Req	Request
Displ, Dply	Display	Rst	Reset
DR	Data Register (display customer console)		
		SAR	Storage Address Register
EBCDIC	Extended Binary-Coded-Decimal Interchange Code	SC	Set Carry
Ex	Execution	SDR	Storage Data Register
		Stor	Storage
FDR	From-Data Register	SU	Shift Unit
FL	Flip Latch (Latch)	Sw	Switch
		Syst	System
IAR	(Micro) Instruction Address Register	T	Time (pulse)
ICPL	Initial Control Program Load	TDR	To-Data Register
ICR	Impulse Check Routine		
Indir	Indirect	u-instr	Microinstruction
Inh	Inhibit (switch)		
Insn, Instr	Instruction	Wkg	Working
IOC	Input/Output Channel		
I/O	Input/Output (Device)	Δ CY	Delta Cycle
I-Recall Addr	Instruction-Recall Address		
LC	Length Count	/.../	Representation of Hexadecimal Numbers
LS	Local Store		





ADDRESS CHECK (AC, bit 7 of FF format instructions).
When the bit is on, the operand addresses (in 'to' or 'from reg' if indirect address) are checked that they are not outside customer storage area. For halfword instructions (e.g. MVHS, MVH, AH), the addresses are also checked for halfword boundary (address must be even). An address check stops the CPU by program check (trap request 2).

SPLIT (bits 5 and 6 of MVHS).
During MVHS, a halfword is split. The result of the split is set into two adjacent registers. The binary value of bits 5 and 6 (0-3) defines the split mode. Details of split are given in the MVHS flowchart.

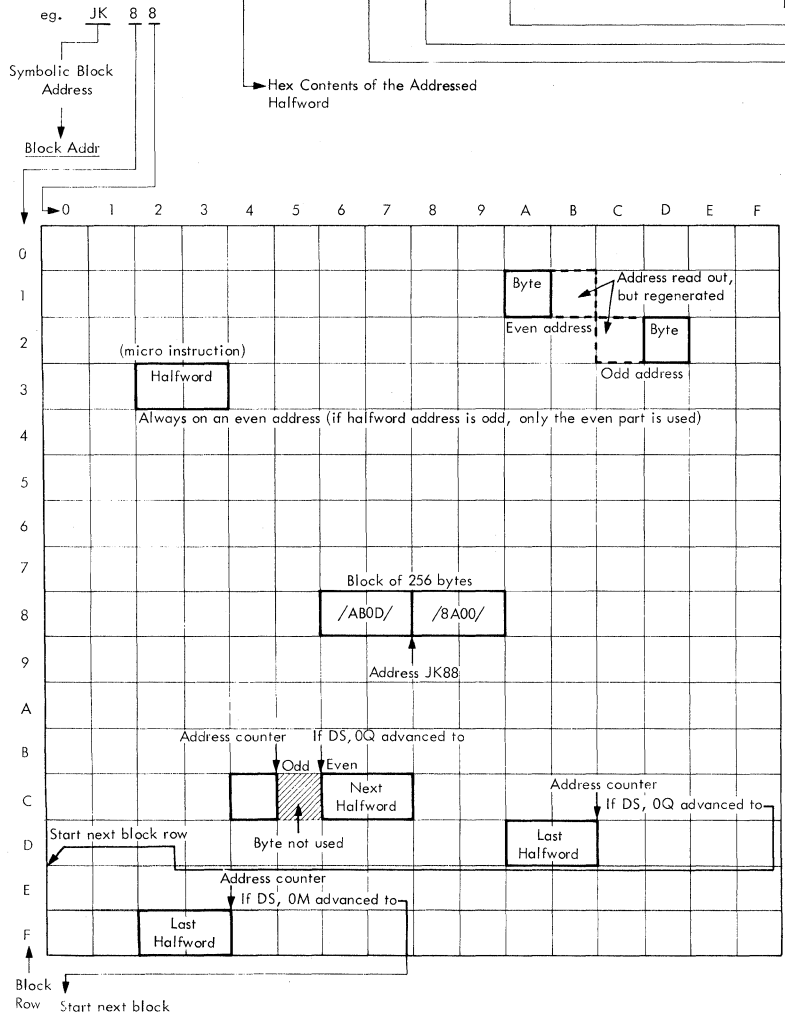
SHIFT AMOUNT (bits 5, 6, and 7 of SLM, SRM).
The pattern in these bits specifies the number of bits by which the operand halfword has to be shifted.

Bits	5	6	7	Shift
0	0	0	0	Shift by 0 (No shift, only move)
0	0	0	1	Shift by 2
0	1	0	0	Shift by 4
1	0	0	0	No shift by 6 (bits 6 and 7 on simultaneously)
1	0	1	0	Shift by 8
1	0	1	1	Shift by 10
1	1	0	0	Shift by 12

'TO REG', 'FROM REG'.
These fields may contain any binary value from 0 to 7. This binary value is used to select one of eight local storage registers in a local store zone. For FF and I/O instructions, the selected register is used as data register if the high-order bit in the 'to or from reg' field is off (direct addressing). The selected register is used as address register for a core storage operand if the high-order bit is on (indirect addressing).
When the 'to reg' specifies local store register 7 (IAR, instruction address register), instructions which set data into a local store register as a result of their operation (load type instruction) are treated as no operation.

Example of Microprogram List (CLD)

ADPM	INST	TO	LABEL	MNEM	OPFRANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX	COMMENTS
JK74	6184	JK84		RZ	1, DIGSFL	RR TO IAR, 0-7/INST, 0-14 IF R1 = 0	
JK76	29FF			ADDI	1, -2	R1 ** R1 + 'FFFF'	
JK78	7164	JK64		BP	1, SIGNST	RR TO IAR, 0-7/INST, 8-14 IF R1 ,GT, 0	
JK7A	6980	JK90		BM	1, SIGNST	RR TO IAR, 0-7/INST, 8-14 IF R1 ,LT, 0	
JK7C	4A66			*	FIELD SEPARATION CHARACTER	CC, R6 ** R6 - R6	
JK7E	806F	JK66		*	SIGNI TCANCE START CHARACTER	RR TO IAR, 0-7/INST, 8-14	
JK80	7684	JK84		SIGNST	RP 6, DIGSFL	RR TO IAR, 0-7/INST, 8-14 IF R6 ,GT, 0	
JK82	0461			OK	6, 1	R6 ** R6 ,OR, R1	
JK84	68AA	JKAA		DIGSFL	RM 0, SCNDGT	RR TO IAR, 0-7/INST, 8-14 IF R0 ,LT, 0	
JK86	AB0D			MVR	0, 51, INC, AC	R0 ** R0, 0-7/BY(R5, AC, +1)	
JK88	8A00			MVHS	0, 0, 1	R0 ** 'A000'/R0, 12-15, R1 ** '00'/R0, 0-11	



The real values of the symbolic block addresses depend upon system configuration and main-storage size. The references of symbolic and real block addresses are given in the LINK LIST which can only be printed immediately following the loading of the microprogram.

Common Symbols

- 'To reg' = 0-7.
- 'From reg' = 0-7.
- l the corresponding reg or label is used for indirect addressing.
- AC = address check performed.
- INC = the indirect address (in reg) is incremented.
- DEC = the indirect address (in reg) is decremented.
- CC = condition code set.
- NC = no condition code.
- X'a...' = any hex digit 0 to F.
- [] = optional function.

The statement is represented by a formula which defines the detailed CPU functions required to perform a microinstruction. The detailed CPU functions are also outlined in the operation flowcharts.

EXAMPLE: AP 4I, 2I, AC

Result = CC, C/BY(R4, AC) * BY(R4, -1), D+, BY(R2, AC, -1), D+, C

Consider as an arrow

Direction of Reading

The result of the operation (see "Summary") is set into the byte (BY) addressed by LS reg 4 (R4, AC). The address is checked. A carry, when decimal adding the operands, is saved by setting the carry latch (C).

- Define operands: BY, D+, BY, D+, C (Two bytes and a previous carry are decimal added.)
- Define addressing: BY (R4), BY (R2) (One byte is addressed by LS reg 4, the other by LS reg 2.)
- What happens with the addresses? (R4, -1), (R2, AC, -1) (Both addresses are decremented by 1. The address in LS reg 2 is checked.)

Summary: The byte addressed by LS reg 2 and a present previous carry are decimal added to the byte addressed by LS reg 4. The addresses are decremented by 1 and checked. NOTE: AC specification is valid for all indirect addresses in an instruction.

Statement Symbols

A	= Logical and	PIUL	= Customer Console Display
AC	= Address Check	PL	= Program Level
BR	= Branch	R	= Local Storage Register
BY	= Byte	SKIP	= Skip Over the Next Instruction
C	= Carry	SL	= Shift Left
CC	= Condition Code	SR	= Shift Right
COMP	= Compare	TDR	= Dataflow Register
D+	= Decimal Plus	UNTIL	= Length's-count Reduction Until.....
D-	= Decimal Minus	VALID SIGN	= Decimal Sign
ESTR	= Customer Console Display		Hex A to F
FDR	= Dataflow Register		= Binary Add
GT	= Greater Than		= Binary Complement Add
HW	= Halfword		= Precedes a Bit Notation
IAR	= Microprogram IAR		= Separator
INDIR	= Indirect		= Connection of Values
INST	= Instruction		= Contains an Address
INVALID	= Invalid Address		= Bits 8-15 (Example)
LT	= Less Than		= Secondary Defined Value
NE	= Not Equal		= Direction of Data Transfer (←)
NORMALIZED	= Decimal Sign For EBCDIC or USASCII		= Hexadecimal Representation
NOT C	= Logical "!" if No Carry		
OE	= Exclusive OR		
OR	= Logical OR		

Alphabetic Listing of Micro-instructions

Mnemonic	Diagram
ADDI	5-6
AH	5-42, 43, 44, 45
AHSC	5-42, 43, 44, 45
AND	5-46, 47, 48, 49
ANDI	5-5
AP	5-54, 55, 56, 57
B	5-20, 21
BAC	5-18, 19
BM	5-18, 19
BP	5-18, 19
BST	5-16, 17
BZ	5-18, 19
CLC	5-50, 51, 52, 53
CTRL	5-70, 71, 72
EOR	5-46, 47, 48, 49
EORI	5-5
HALT	5-66
IBL	5-4
LBI	5-3
LH	5-12, 13
LR	5-23
MVB	5-34, 35, 36, 37
MVH	5-30, 31, 32, 33
MVHS	5-24, 25
MVN	5-38, 39, 40, 41
MVZ	5-38, 39, 40, 41
OR	5-46, 47, 48, 49
ORI	5-5
PPC	5-58, 59, 60, 61
SDS	5-62, 63, 64, 65
SENS	5-67, 68, 69
SH	5-42, 43, 44, 45
SHSC	5-42, 43, 44, 45
SLM	5-26, 27, 28, 29
SP	5-58, 59, 60, 61
SRM	5-26, 27, 28, 29
STH	5-14, 15
STR	5-22
TMS	5-7
TMSZ	5-7
TRBL	5-10, 11
TRBS	5-8, 9
ZAP	5-54, 55, 56, 57

DEFINE CONSTANT. The constants are hexadecimal digits (up to 16), characters (up to 32) or addresses. The addresses are of four different types (A, E, B, D).
 Type A = halfword address defined by label or by label ± n (n = 0 to 255).
 Type E = an internal label is similar to an external label. The constant is the address defined by the external label.
 Type B = the constant is the block address (high-order byte) of the address defined by the label.
 Type D = the constant is the low-order byte of the address defined by the label or by the label ± n (n = 0 to 255).
 ORIGIN. Command for the address counter of the assembler to go either to the ORG address ± n (if n = 0 to 255) or to the address specified by the label.
 START. The start address of a section is defined. The required label is used as section name.
 ENTRY/EXTERN. The label defines an address at which the section is entered from another section (ENTRY) or to which the program goes on leaving the section.
 DEFINE STORAGE. Command for the assembler to advance the address counter.
 The counter is advanced either according to n (N = 0 to 255) or to a storage boundary defined by OH, OQ, or OM:
 OH = continue at next halfword address.
 OQ = continue at start address of next block row.
 OM = continue at start address of next block. (see examples in the block on the left)
 EQUATE. The required label is equated either with the direct address (X'aaaa'), with the address defined by the label, or with the address defined by the label ± n (n = 0 to 255).
 For EQU* the required label is equated with the address presently in the address counter.

Microinstructions

X	LBI	'To reg', X 'aa'
X	IBL	or
X	ANDI	'To reg', n
X	EORI	or
X	ADDI	'To reg', -n
X	TMSZ	
X	TMS	
X	TRBS	'To reg', label [X 'a']
X	TRBL	'To reg', B (label)
X	LH	
X	STH	To reg, label [±n][, l]
X	BZ	
X	BM	
X	BP	
X	BAC	
X	BST	Label [±n][, l]
X	B	
X	LR	'To reg', 'from reg'
X	STR	
X	HALT	
X	SLM	'To reg' [l], 'from reg' [l], n
X	MVHS	'To reg', 'from reg' [l], n, [AC]
X	MVH	'To reg' [l], 'from reg' [l], INC [AC]
X	MVB	'To reg' [l], 'from reg' [l], DEC [AC]
X	MVN	
X	MVZ	
X	AP	
X	ZAP	'To reg' [l], 'from reg' [l], [AC]
X	SP	
X	PPC	
X	SDS	
X	AH	
X	AHSC	'To reg' [l], 'from reg' [l], CC [AC]
X	SH	or
X	SHSC	'To reg' [l], 'from reg' [l], NC [AC]
X	AND	
X	OR	
X	EOR	
X	CLC	
X	SENS	'To reg' [l], X'aa' or
X	CTRL	'To reg' [l], label [±n]
X	DC	A (label [±n])
X	DC	E (label, label)
X	DC	B (label)
X	DC	D (label [±n])
X	DC	C 'ABCDEFGH.....'
X	DC	X 'aaaa.....'
X	ORG	* ±n
X	ORG	Label
X	START	X 'aaaa'
X	ENTRY	Label
X	EXTRN	Label
X	DS	n
X	DS	OH
X	DS	OQ
X	DS	OM
X	EQU	X 'aaaa' or*
X	EQU	Label [±n]
X	END	Label

Assembler Instructions

n = 0 through 127
-n = -1 through -128

n = 0 through 255, however, the result of label ± n must be an address inside block boundary

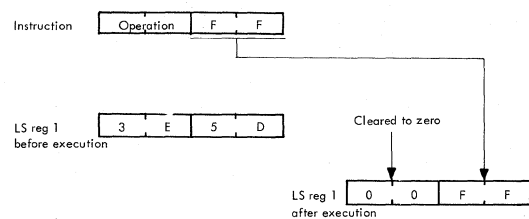
n = 0, 2, 4, 8, 10, 12 (Shift amount)

n = 0 to 3 (Split mode)

n = 0 through 255, however, the result of label ± n must be an address inside block boundary

END. The assembler branches to the address defined by the label (start microprogram) and executes microinstructions.

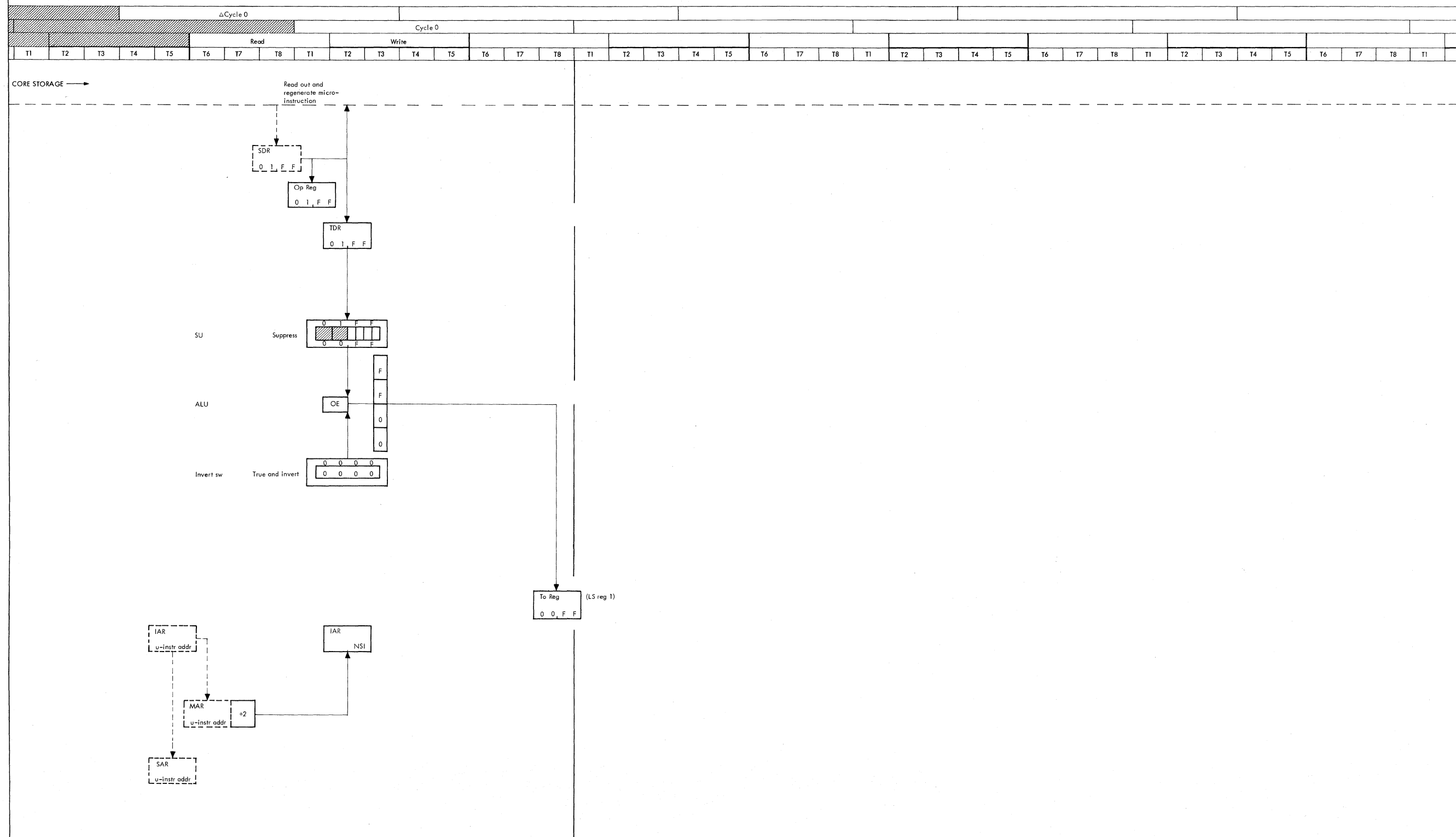
The immediate data byte is loaded into the low-order byte of the LS register specified in the 'to reg' field of the instruction. The high-order byte is set to zero.



Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code					To Reg		Immediate Data								
0	0														

Mnemonic
LBI
Format
RI
Type

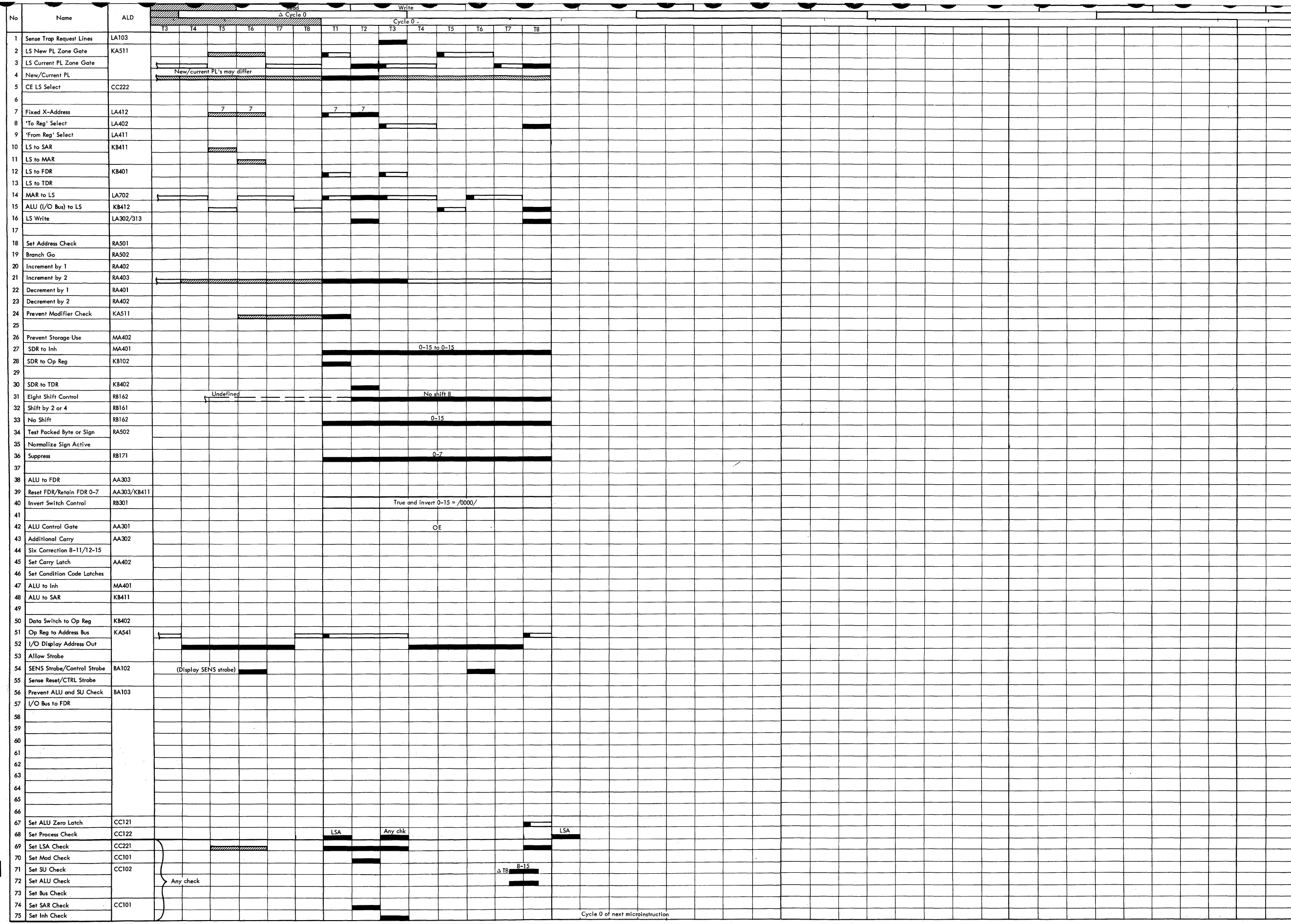
INST MNEM OPERANDS STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
 01FF LBI 1,X'FF' R1+X'00FF'

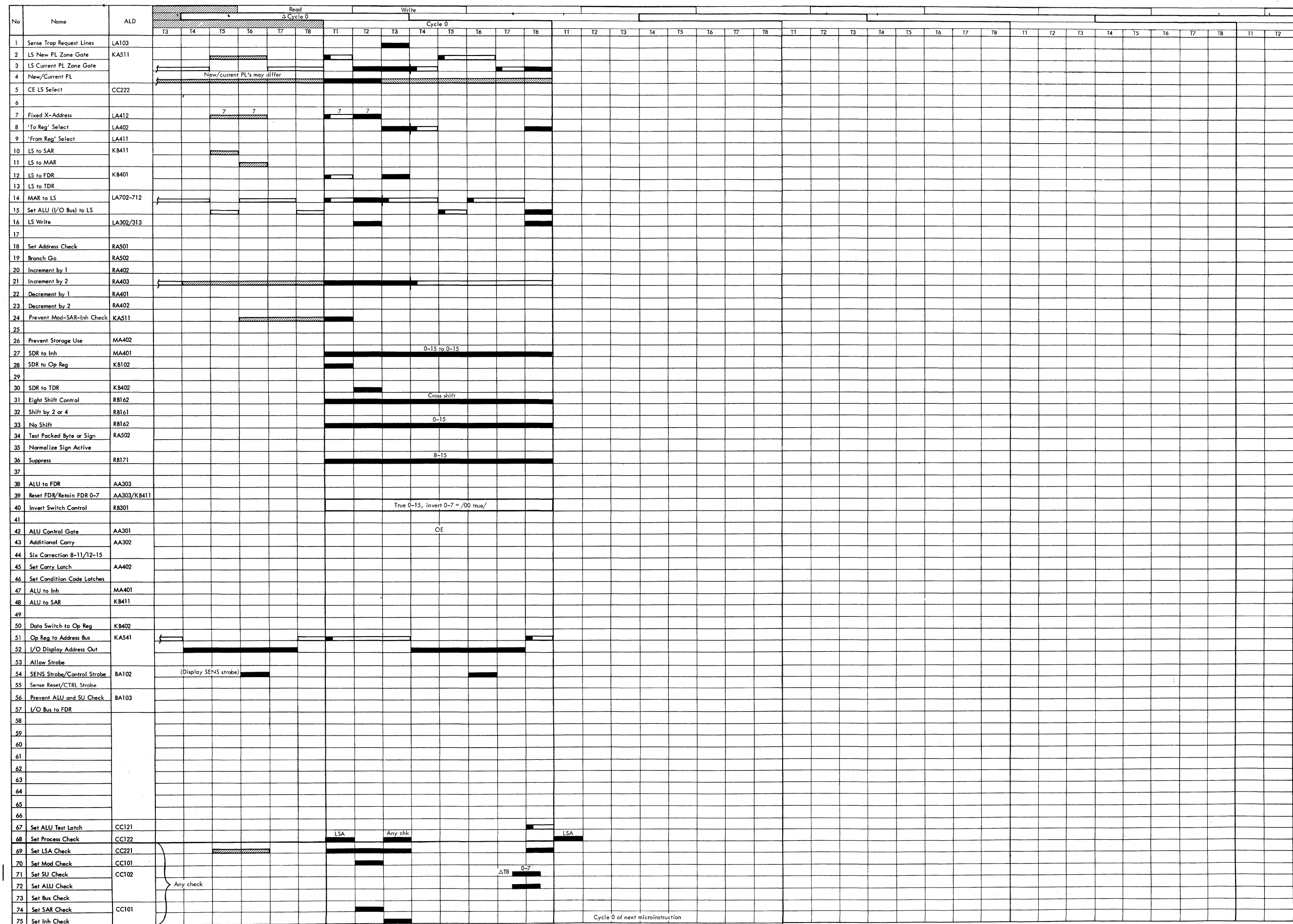


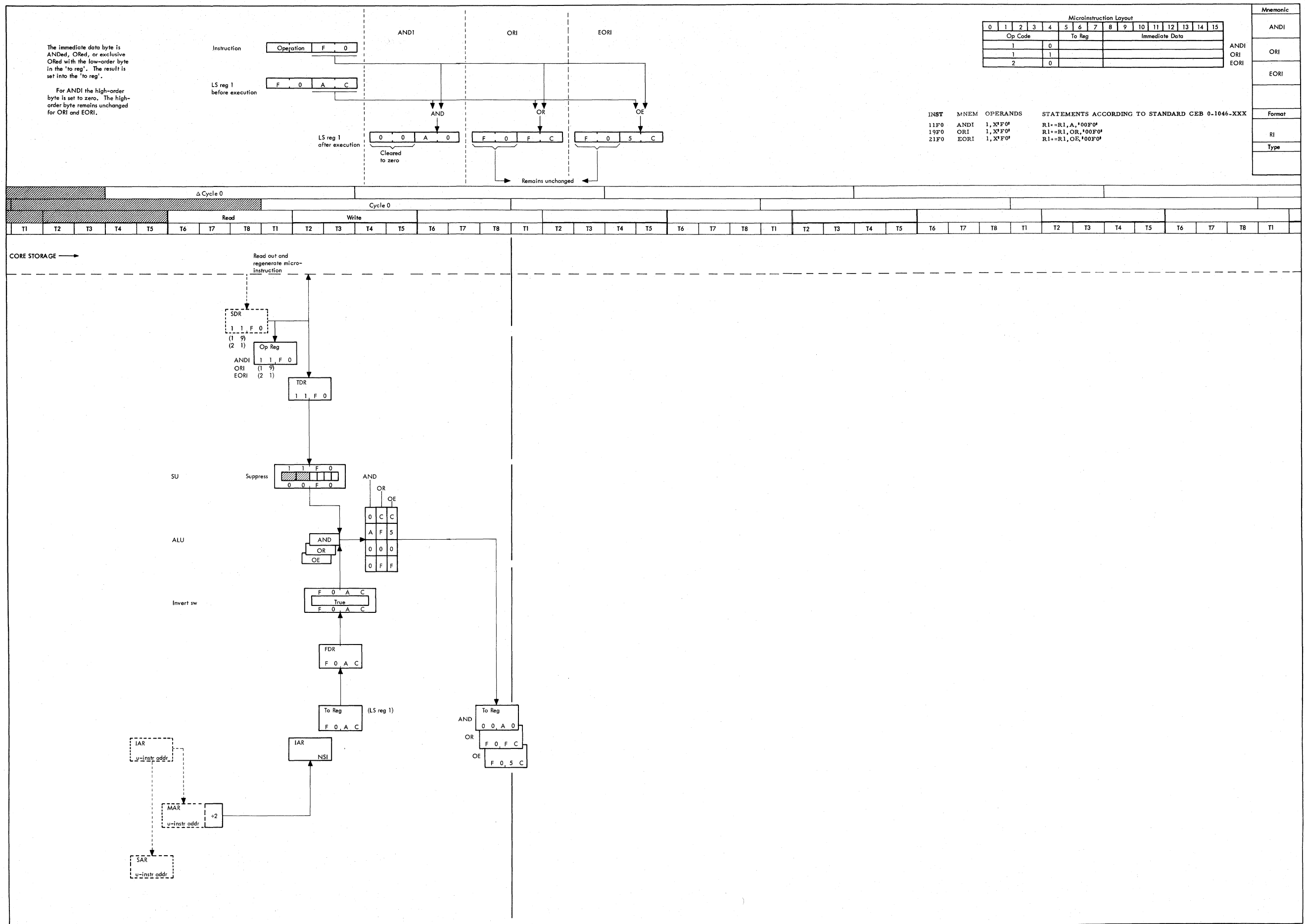
Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



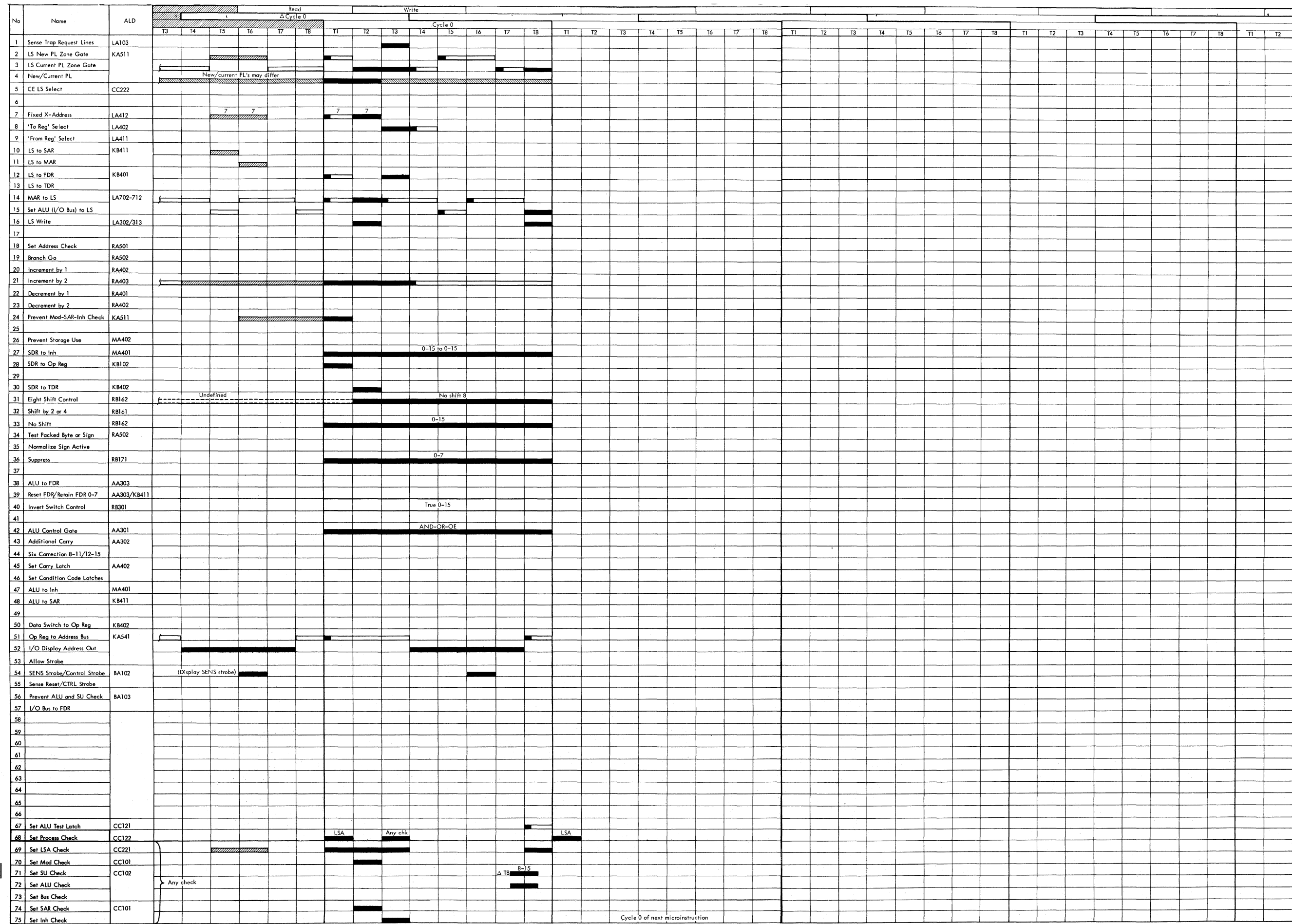


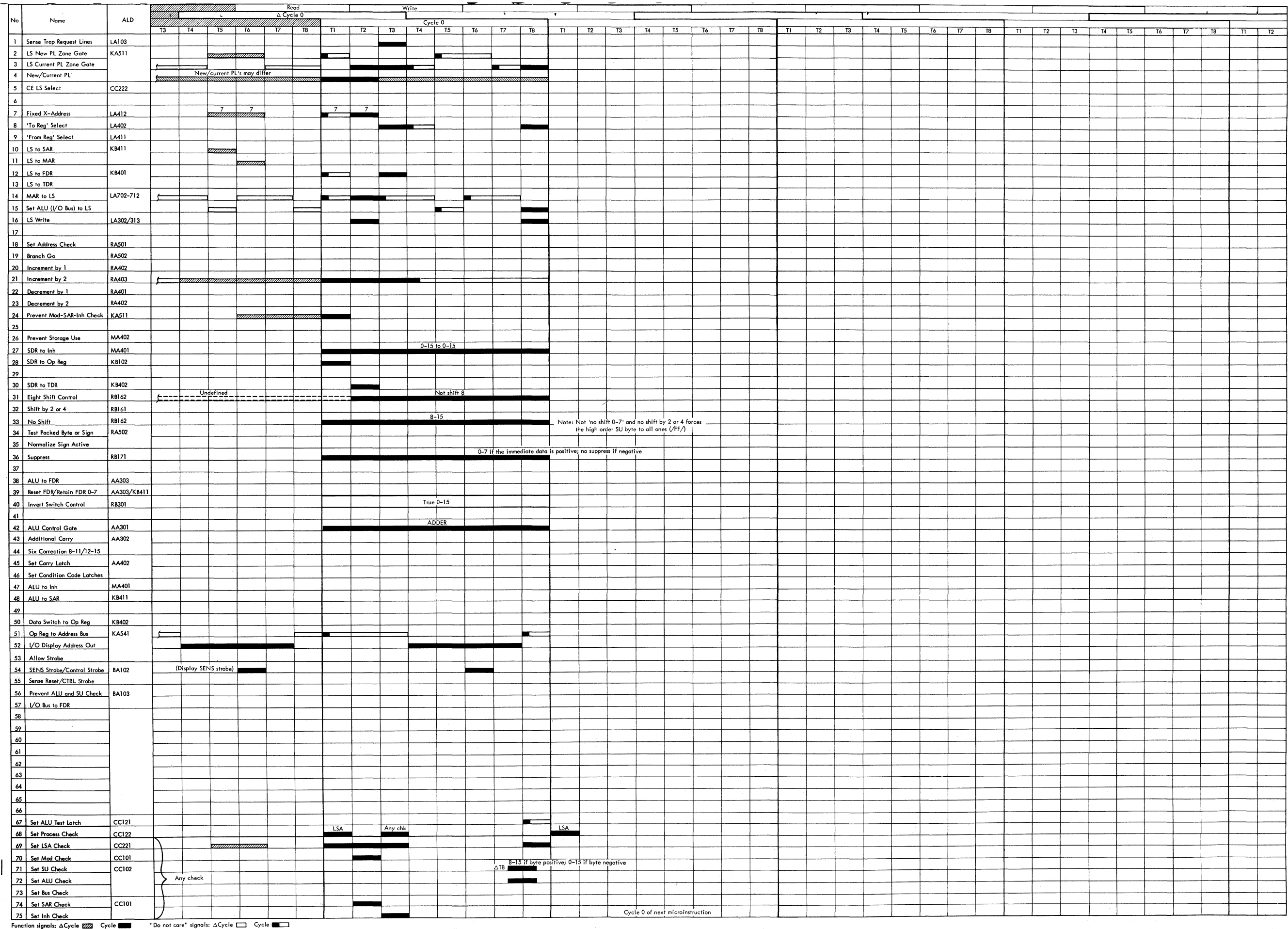


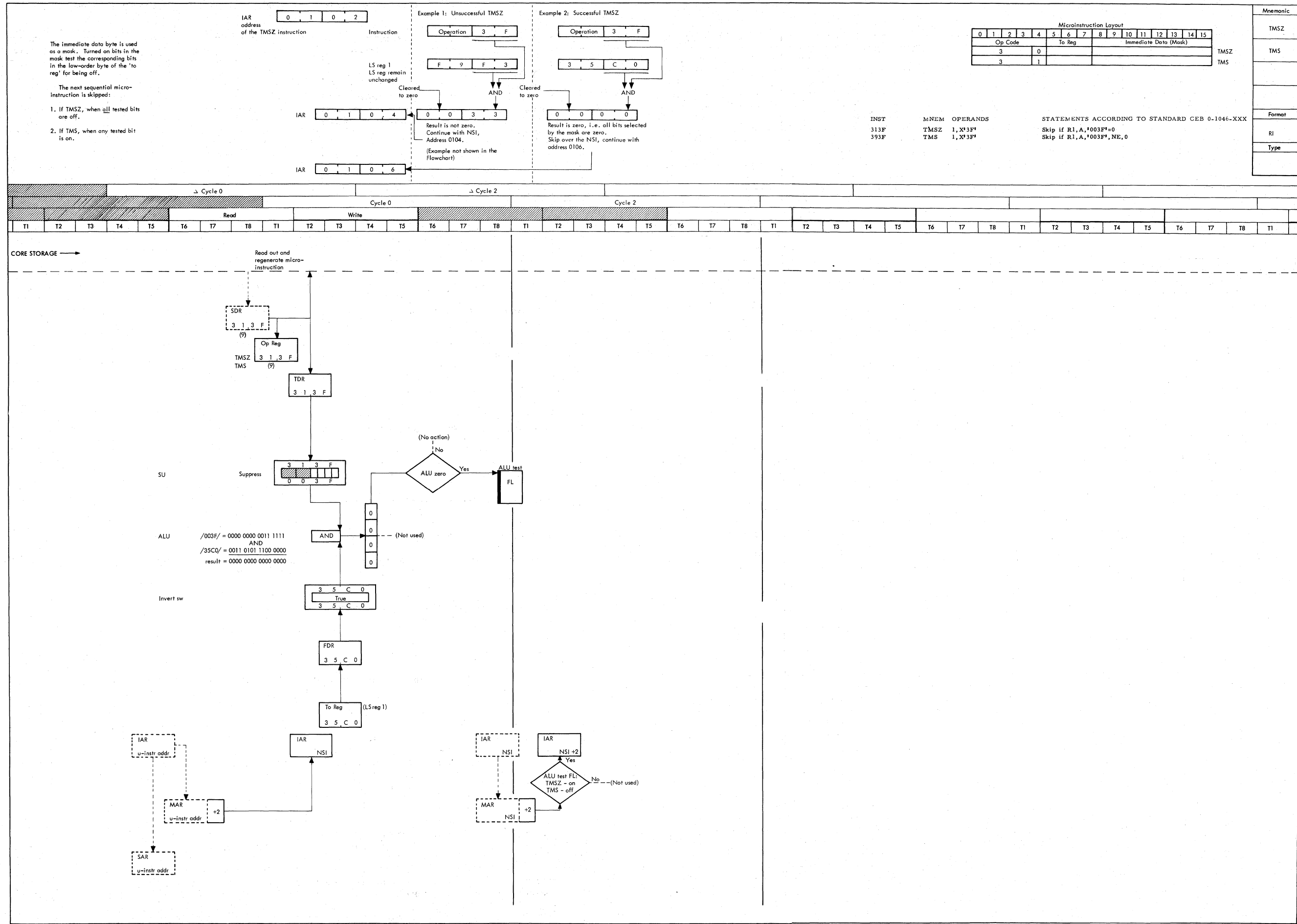
Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

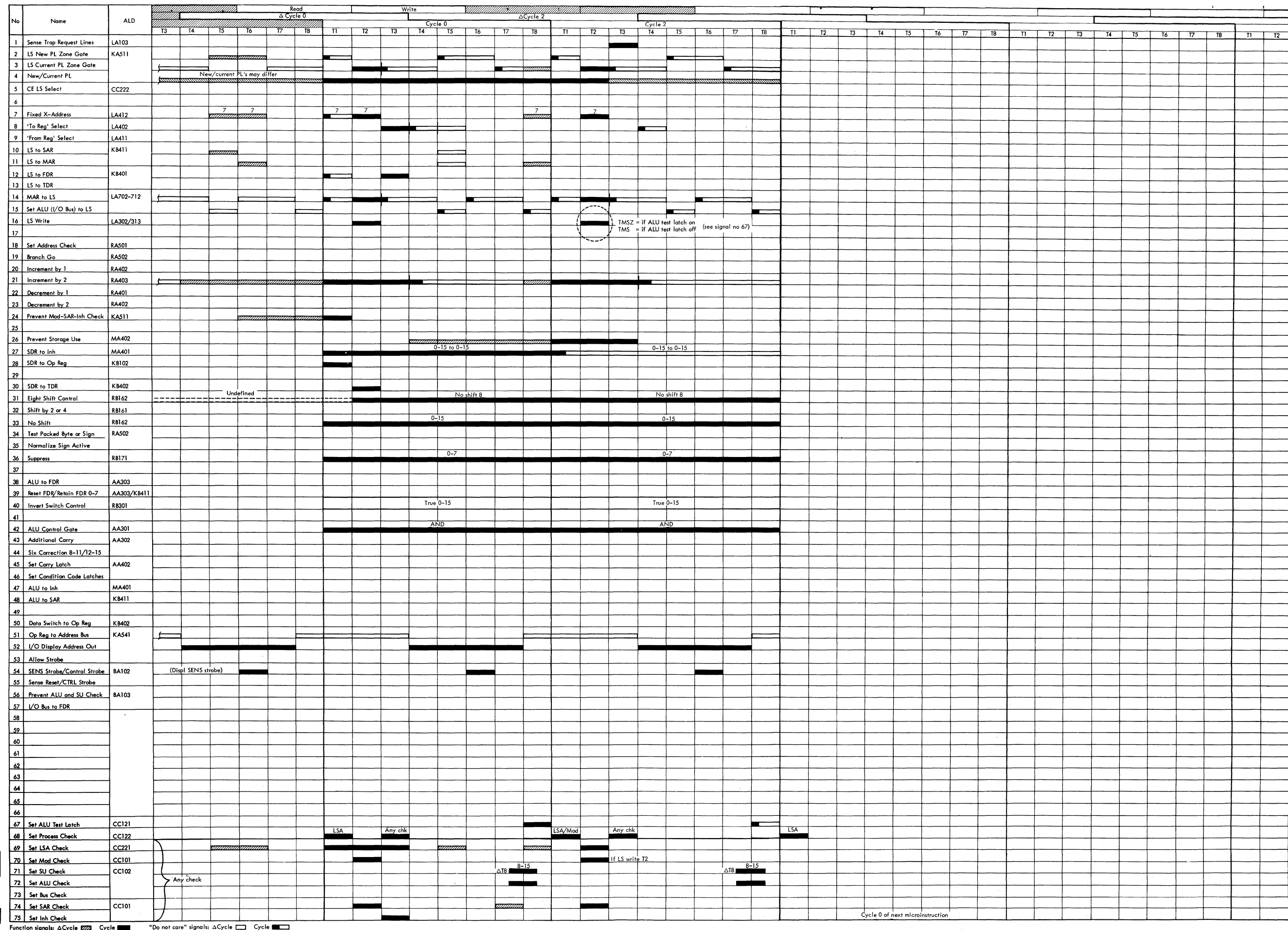




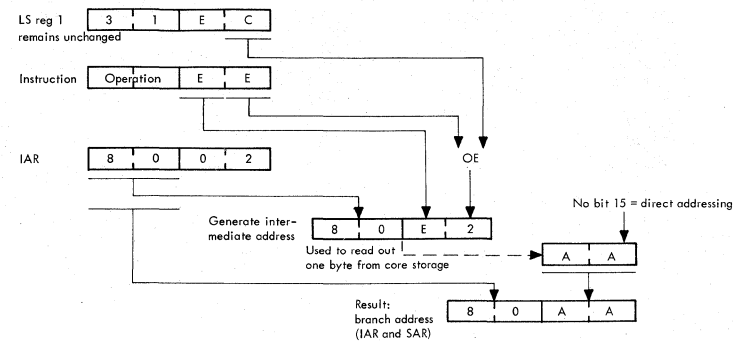


Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle



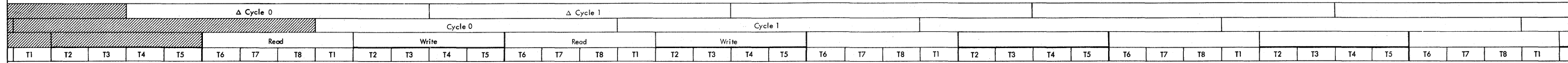
A branch address is combined with the current block address (high-order byte of IAR) and an even byte is read out by an intermediate address (even byte = direct addressing). The intermediate address consists of the current block address as the high-order byte while the low-order byte is obtained by exclusive OR'ing the immediate data byte with the low-order four-bit digit (hexadecimal) in the 'to reg'. The 'to reg' remains unchanged.



Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code				To Reg				Immediate Data (Addr)							
4				0											

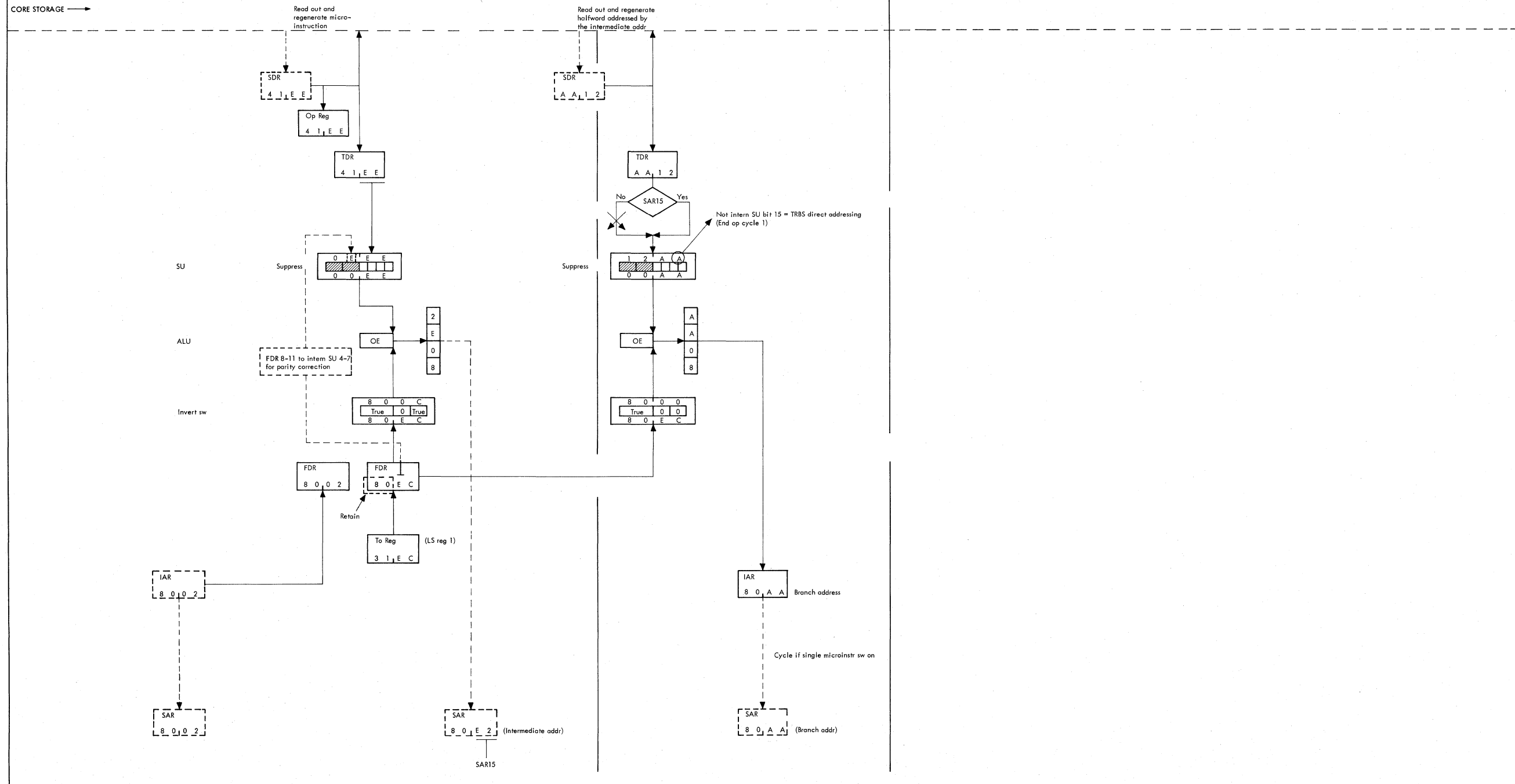
INST 41EE MNEM TRBS OPERANDS 1, LABEL STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
 BR TO IAR.0-7/BY(IAR.0-7/INST.8-11/X), X=INST.12-15, OE, R1.12-15, IF BY.15=1 BR INDIR.

Mnemonic	TRBS
Format	RI
Type	Direct Addressing

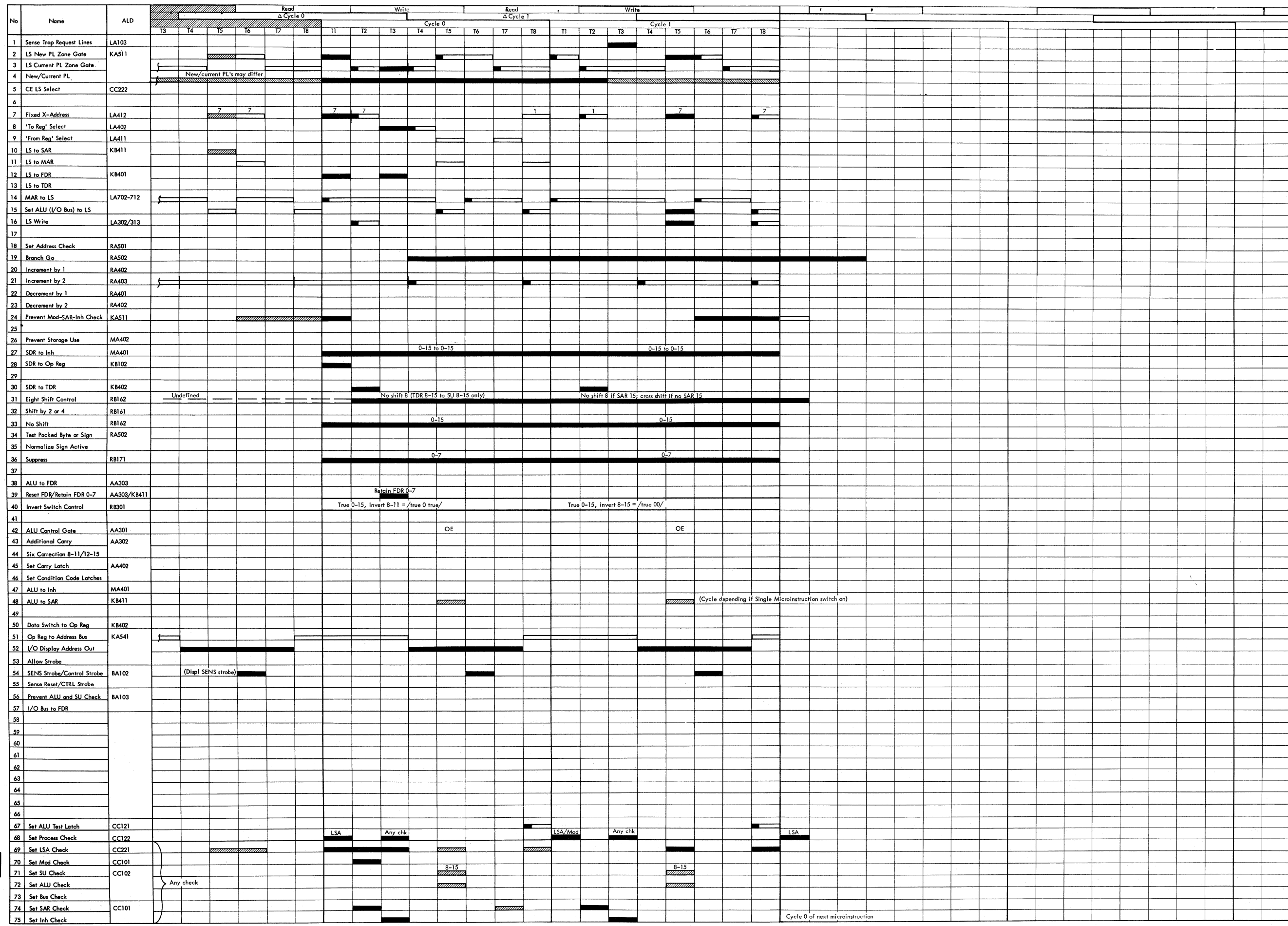


Note: For "Do not care" functions refer to timing chart below.

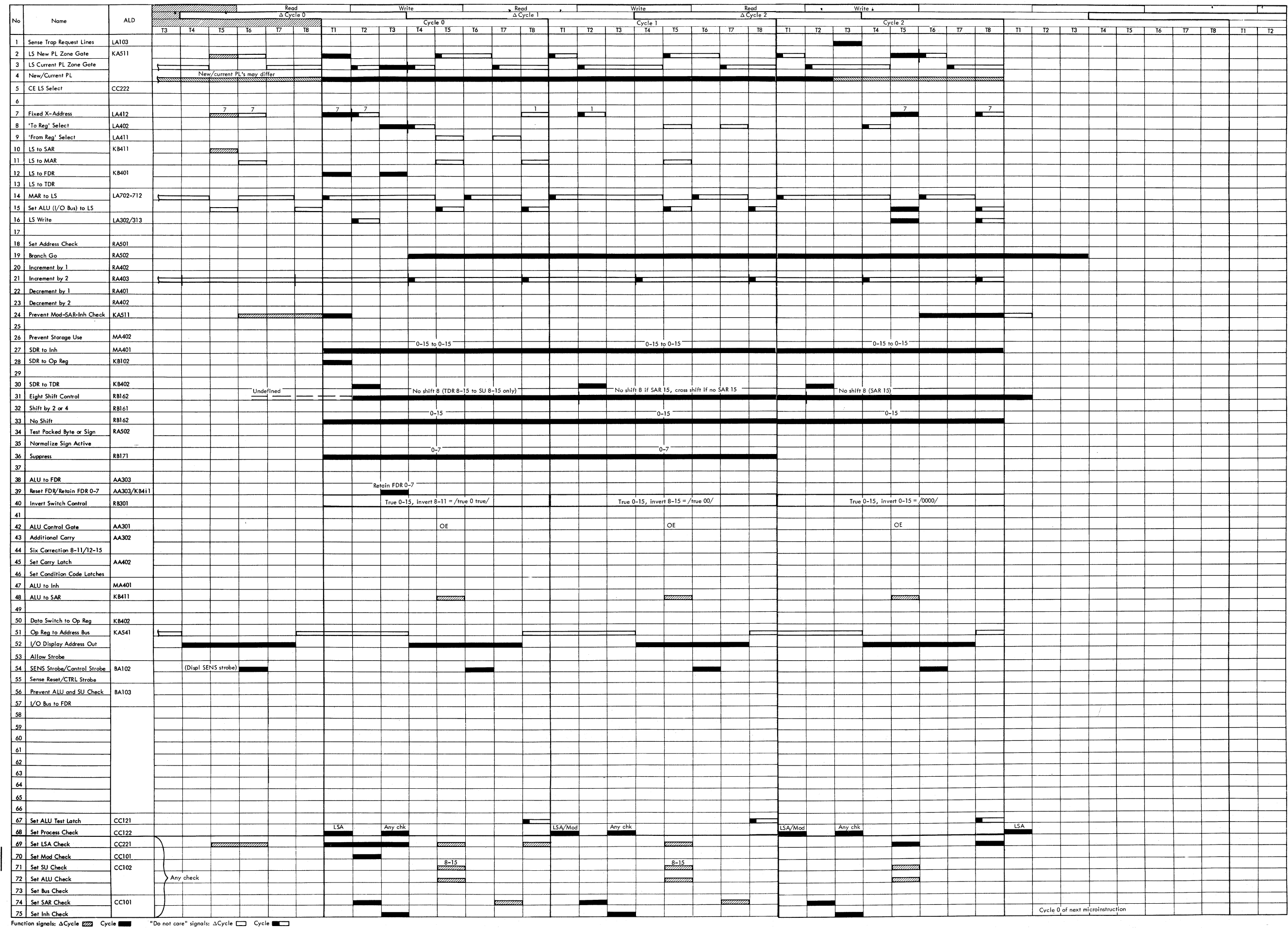
"Do not care" signals:
 ΔCycle
 Cycle

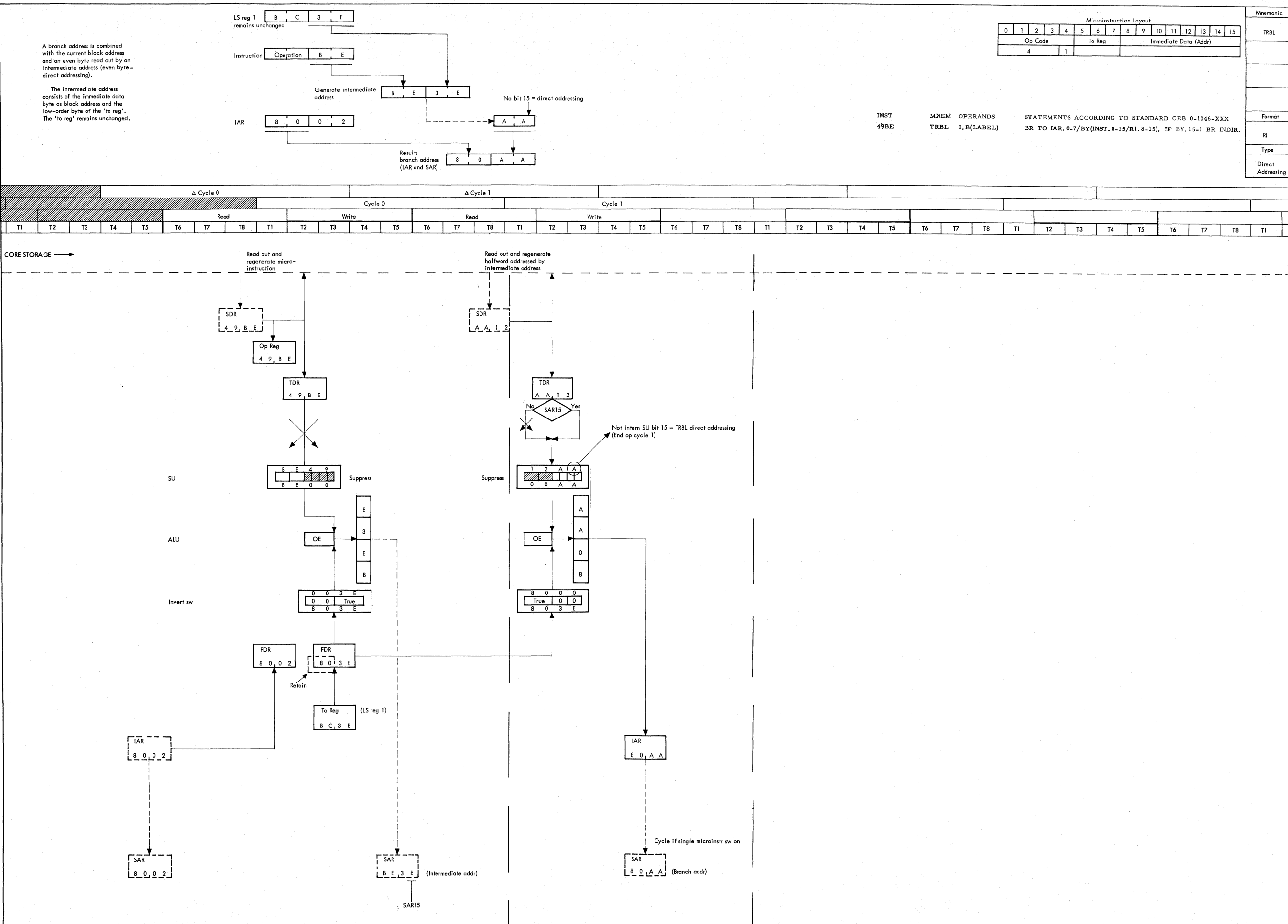


Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



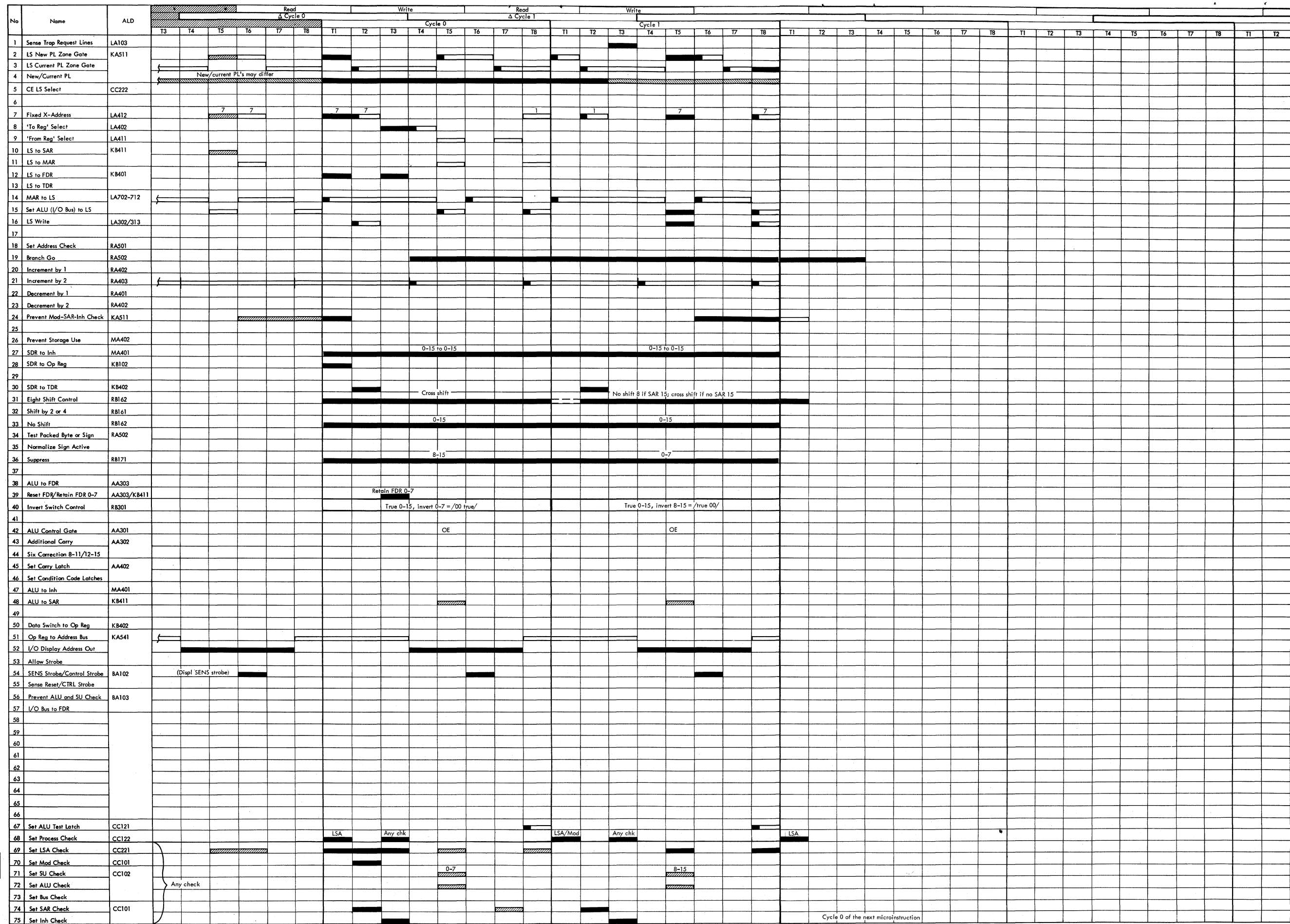
Function signals: Δ Cycle Cycle "Do not care" signals: Δ Cycle Cycle

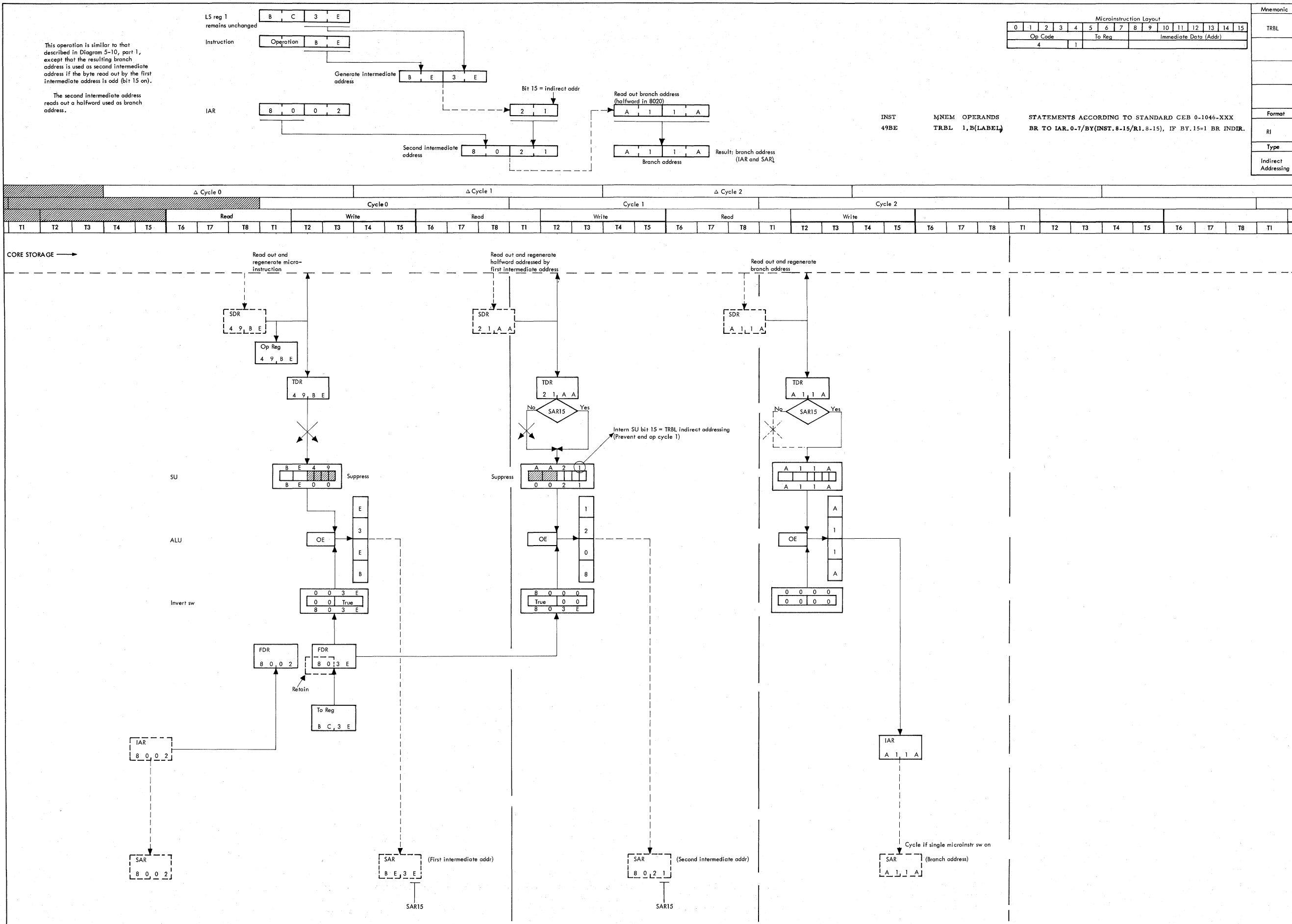




Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

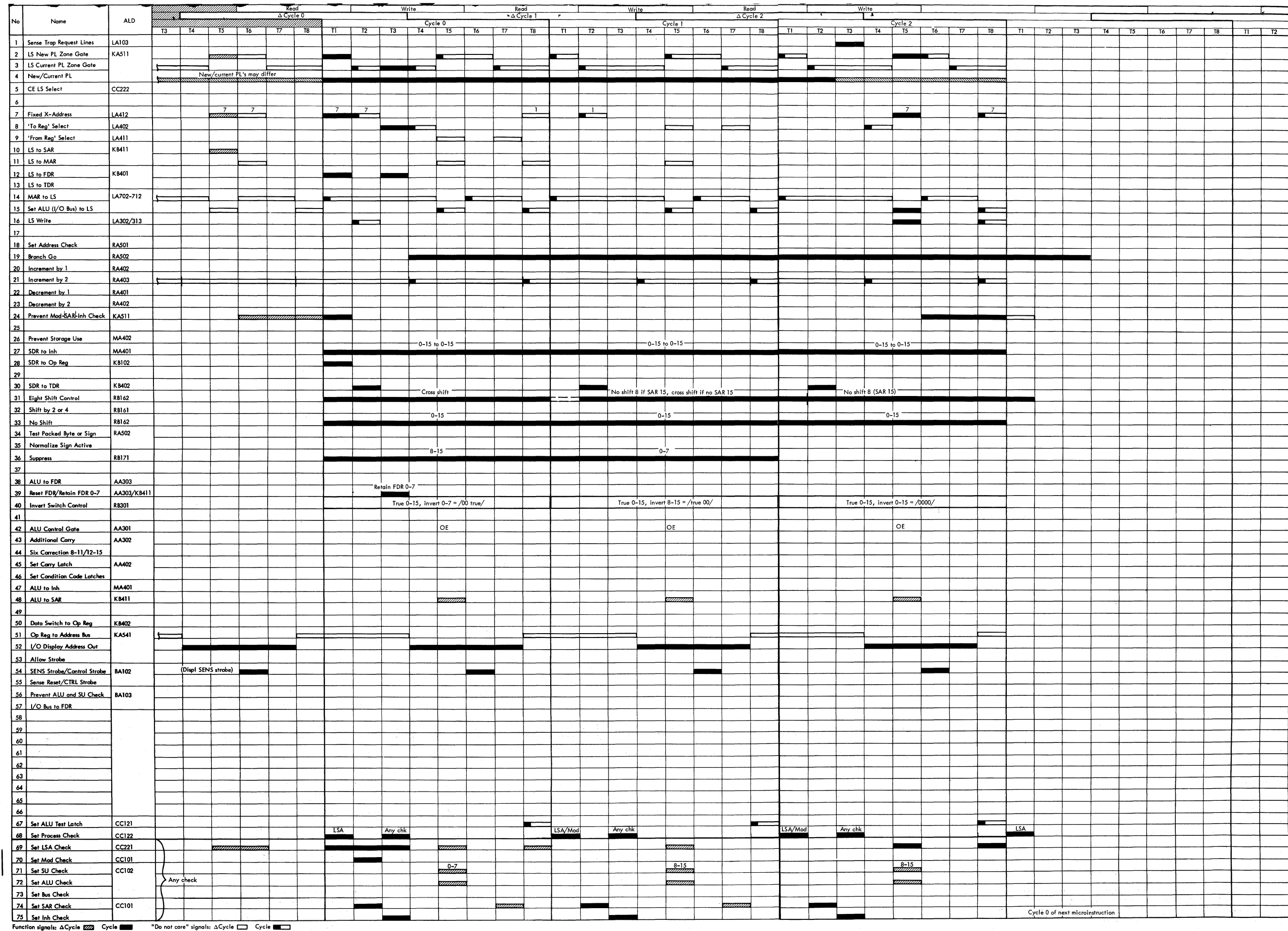


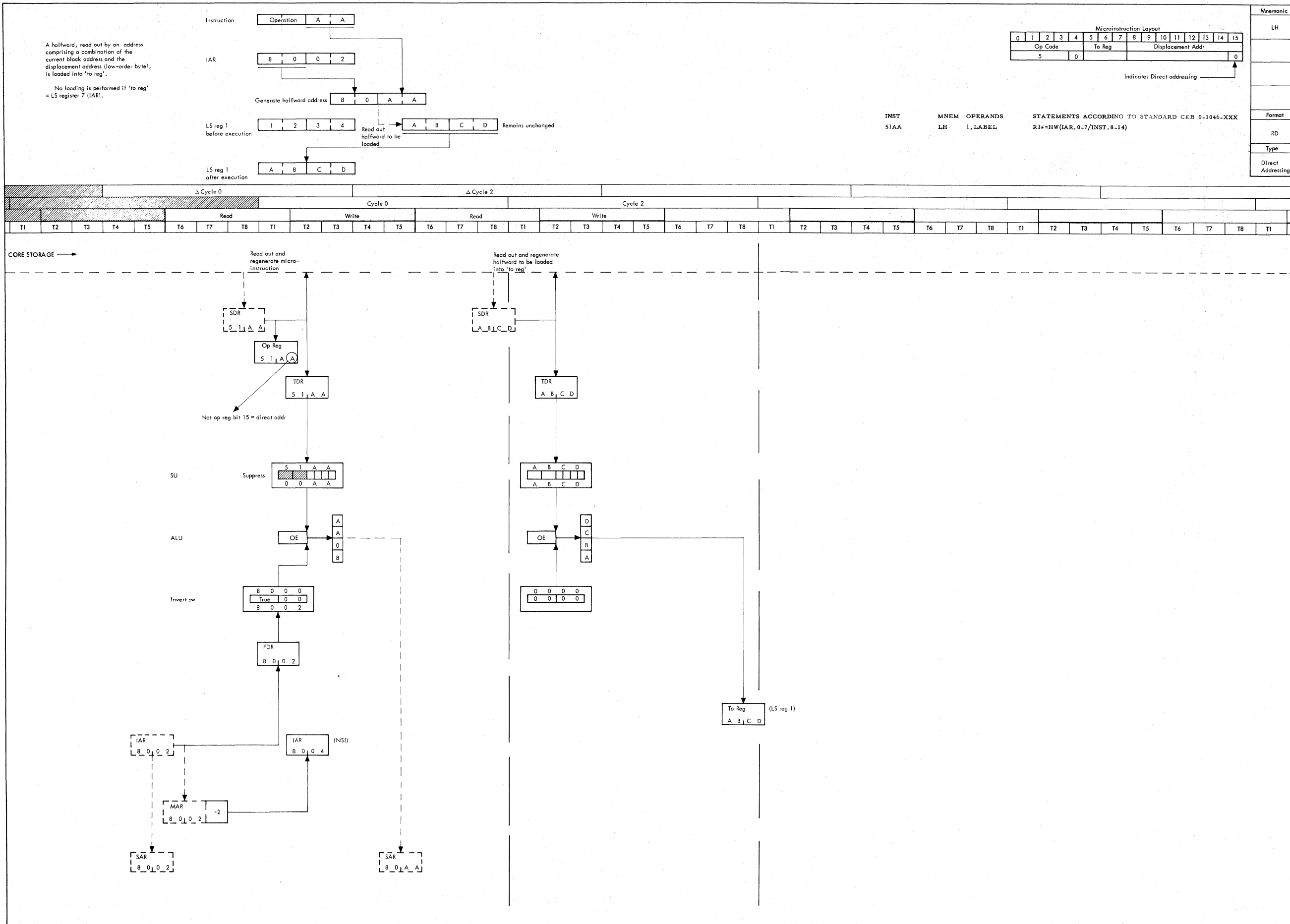


Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

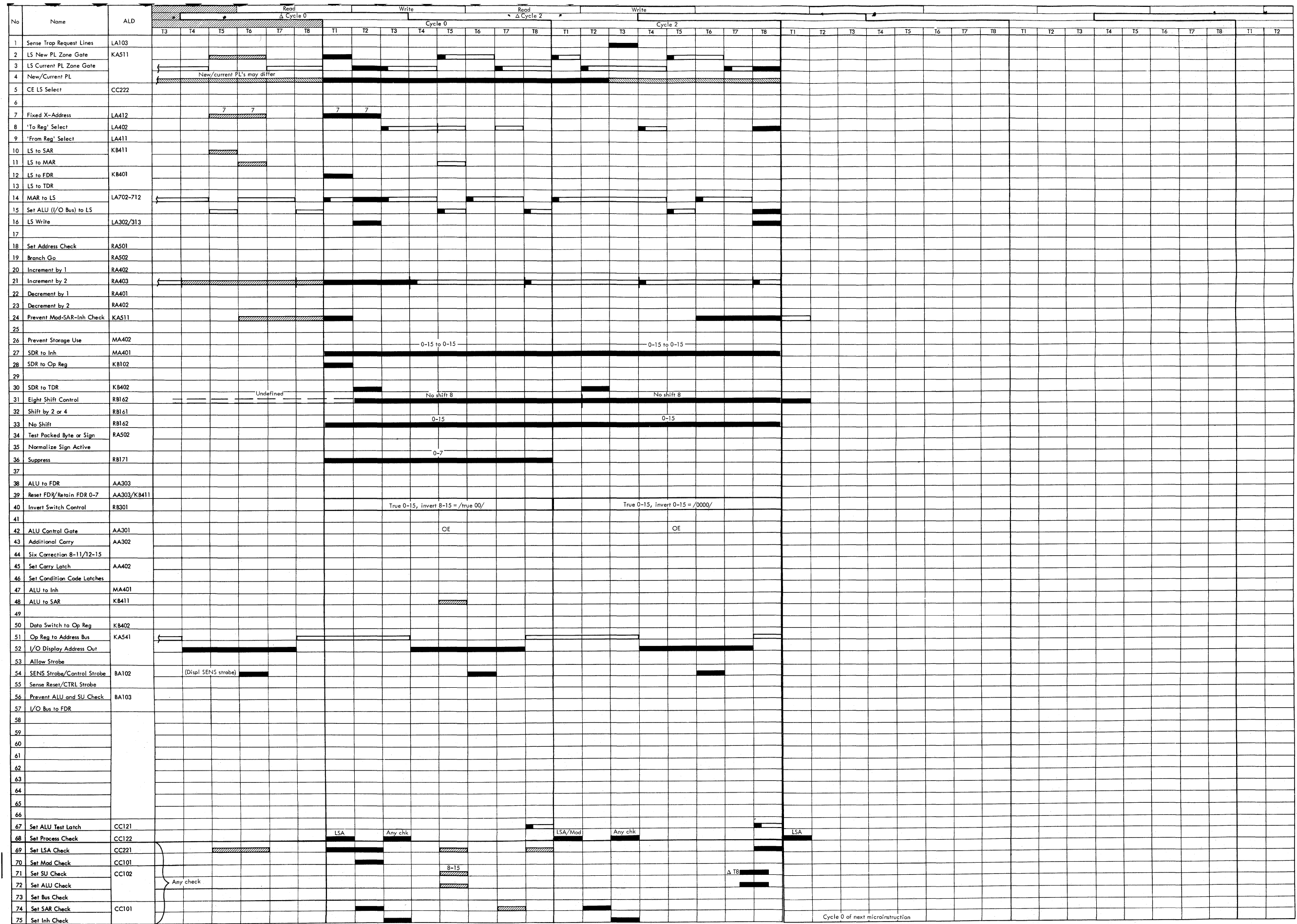




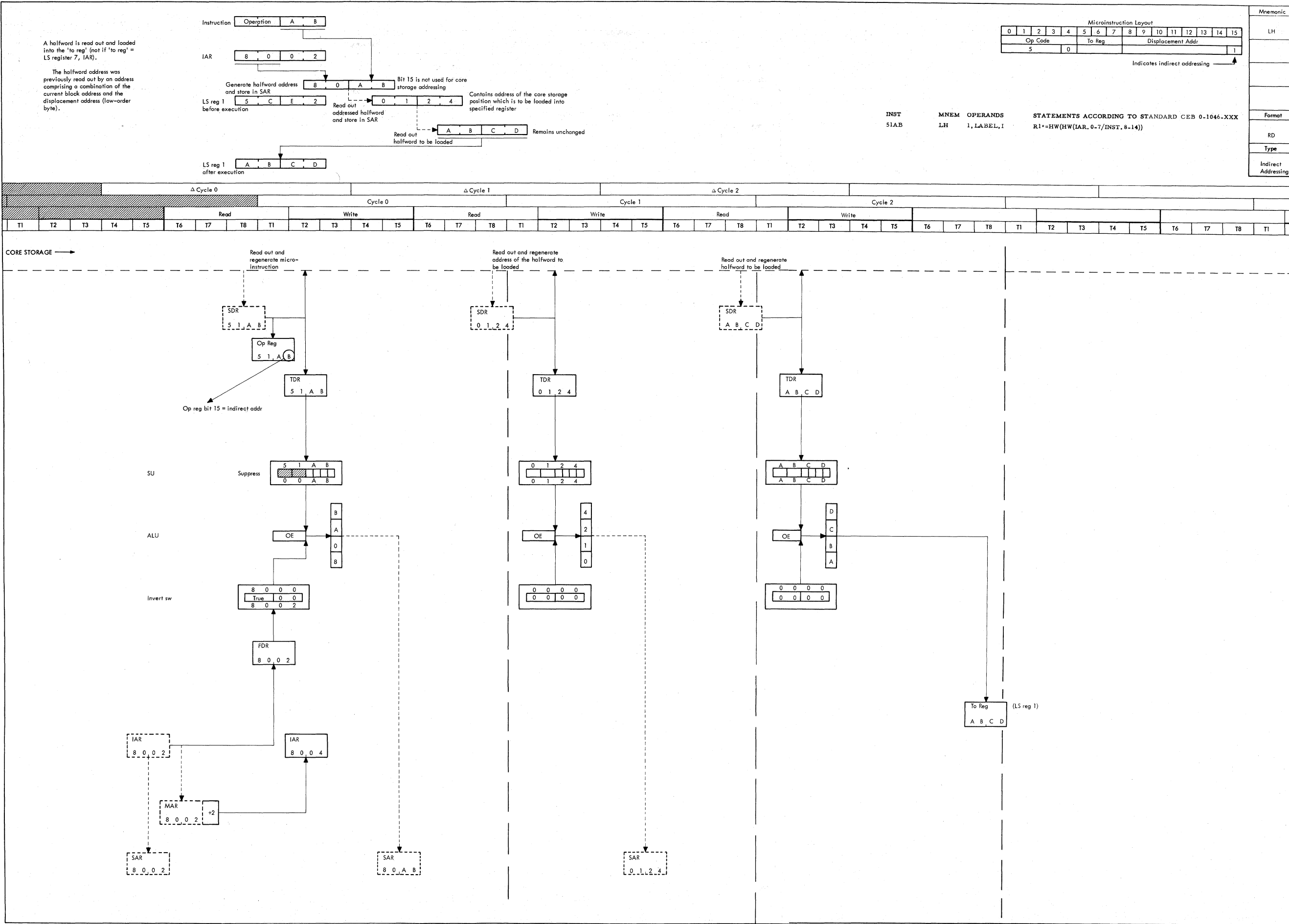
Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 Δ Cycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



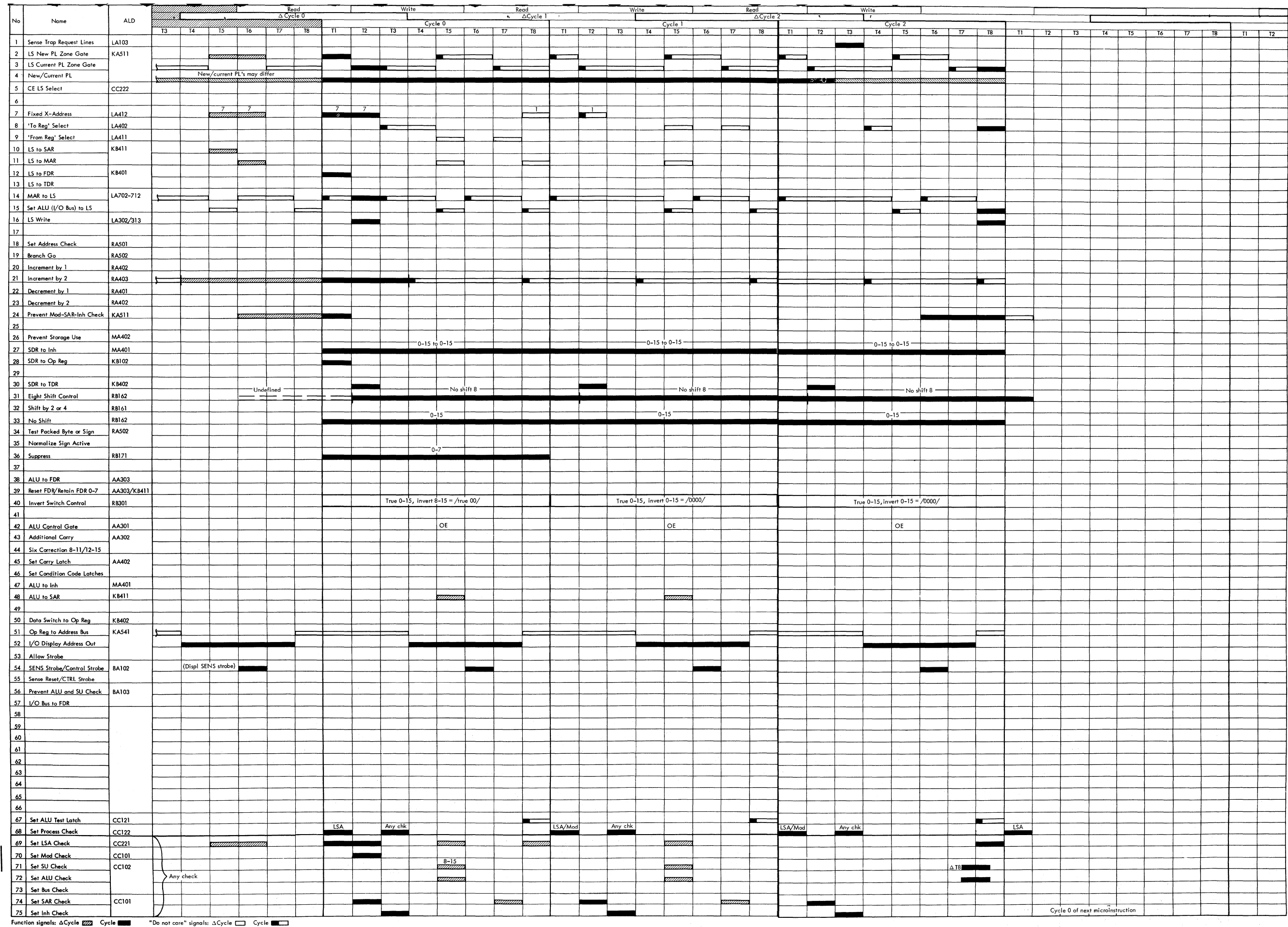
● Diagram 5-12. Load Halfword (Direct Addressing) (Part 2 of 2) (03717A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)



Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



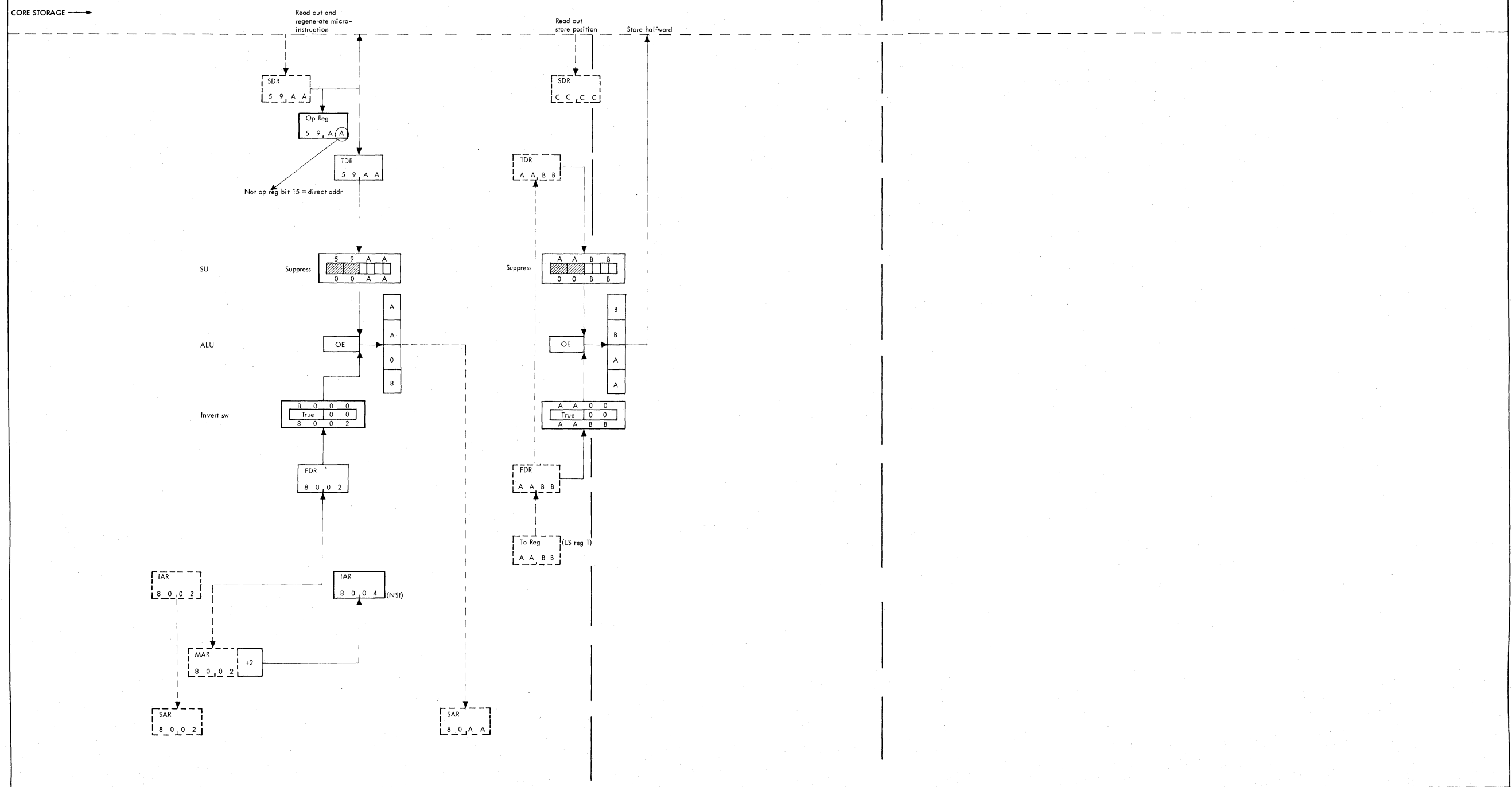
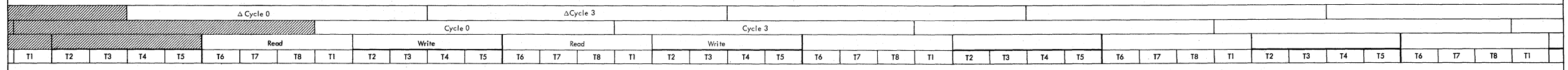
The "to reg" halfword is stored in core storage. The address comprises a combination of the current block address and the displacement address (low-order byte).

Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code					To Reg					Displacement Addr					
5					1					0					

Indicates direct addressing

INST 59AA MNEM STH OPERANDS 1, LABEL STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX HW(IAR, 0-7/INST, 8-14)--R1

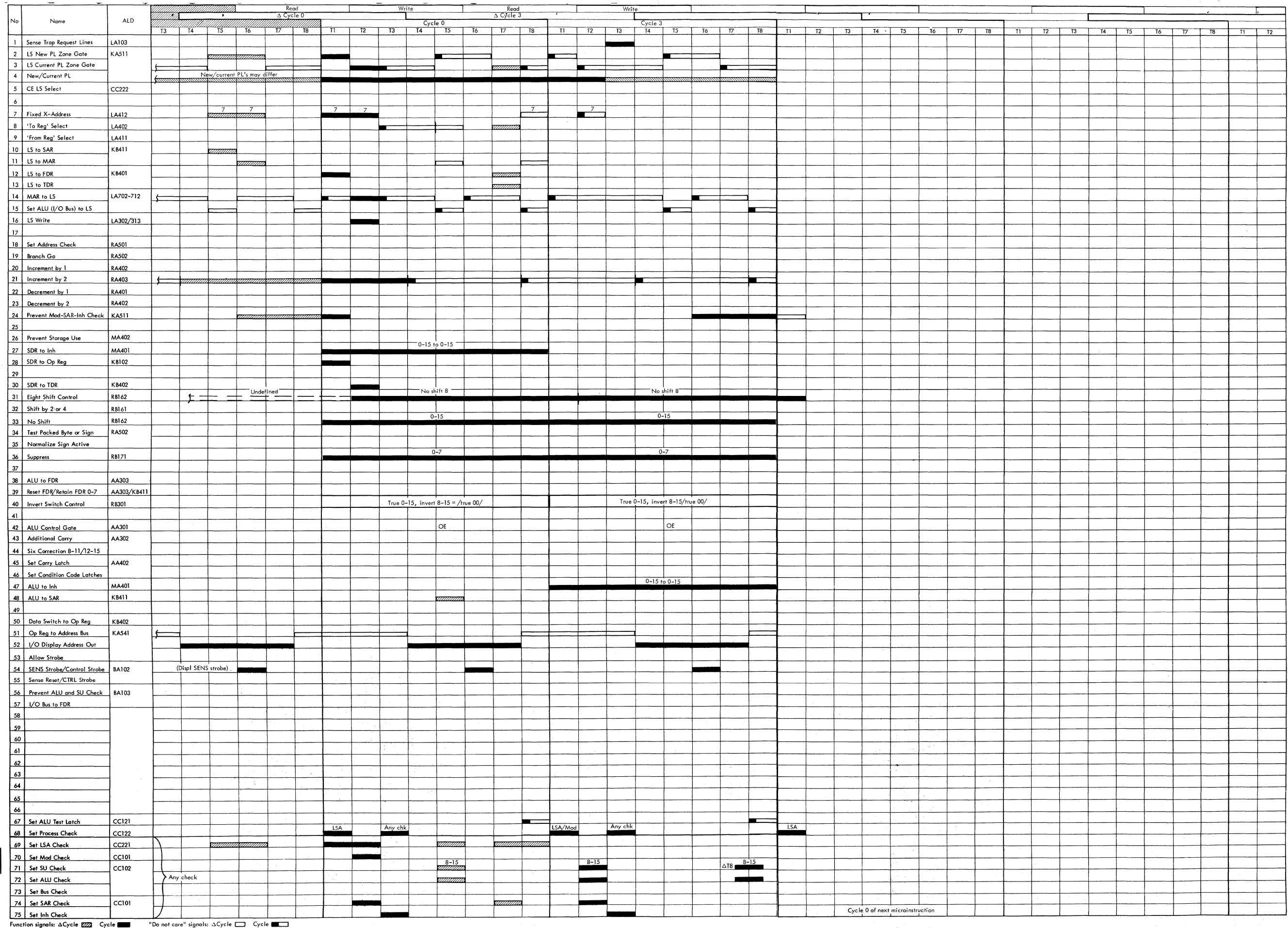
Mnemonic	STH
Format	RD
Type	Direct Addressing



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

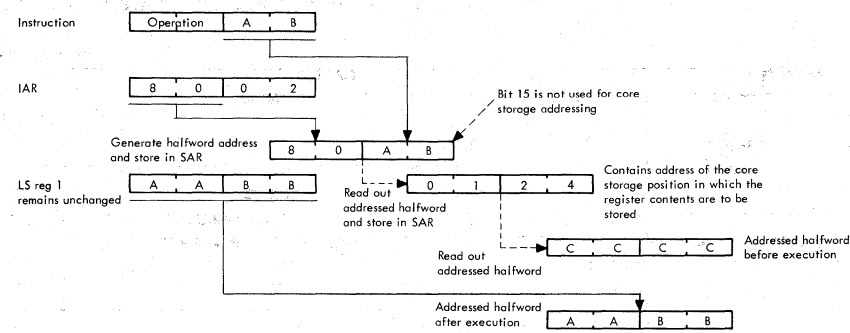
Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle



Function signals: Δ Cycle Cycle "Do not care" signals: Δ Cycle Cycle

The 'to reg' halfword is stored into core storage. The store address was previously read out by an address comprising a combination of the current block address and the displacement address (low-order byte).

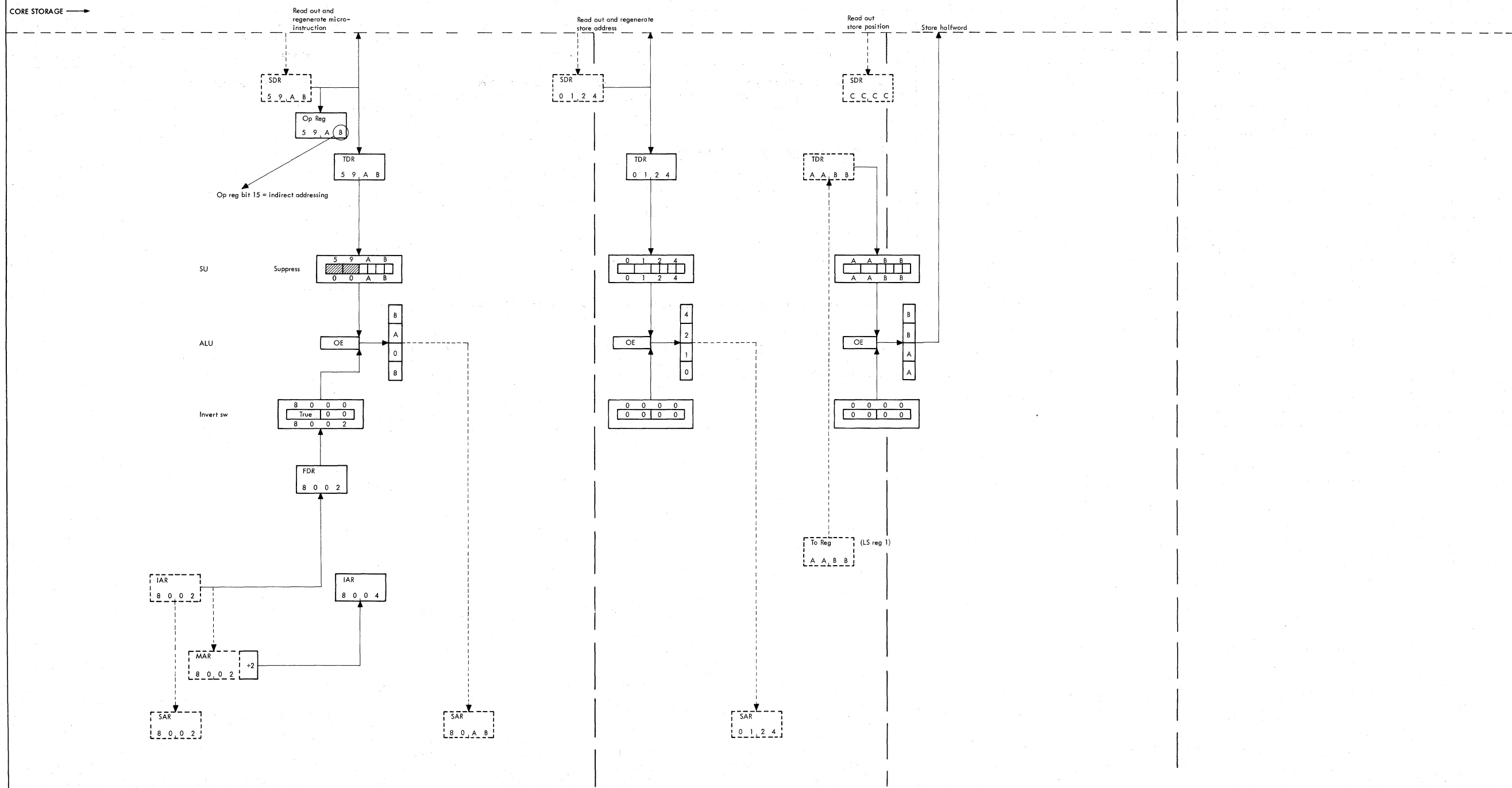
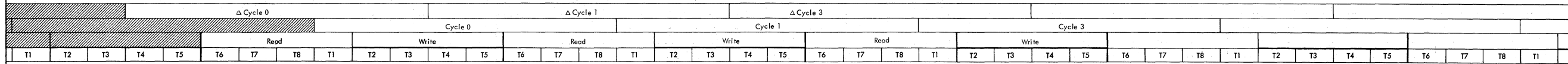


Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code					To Reg					Displacement Addr					
5					1					1					

Indicates indirect addressing

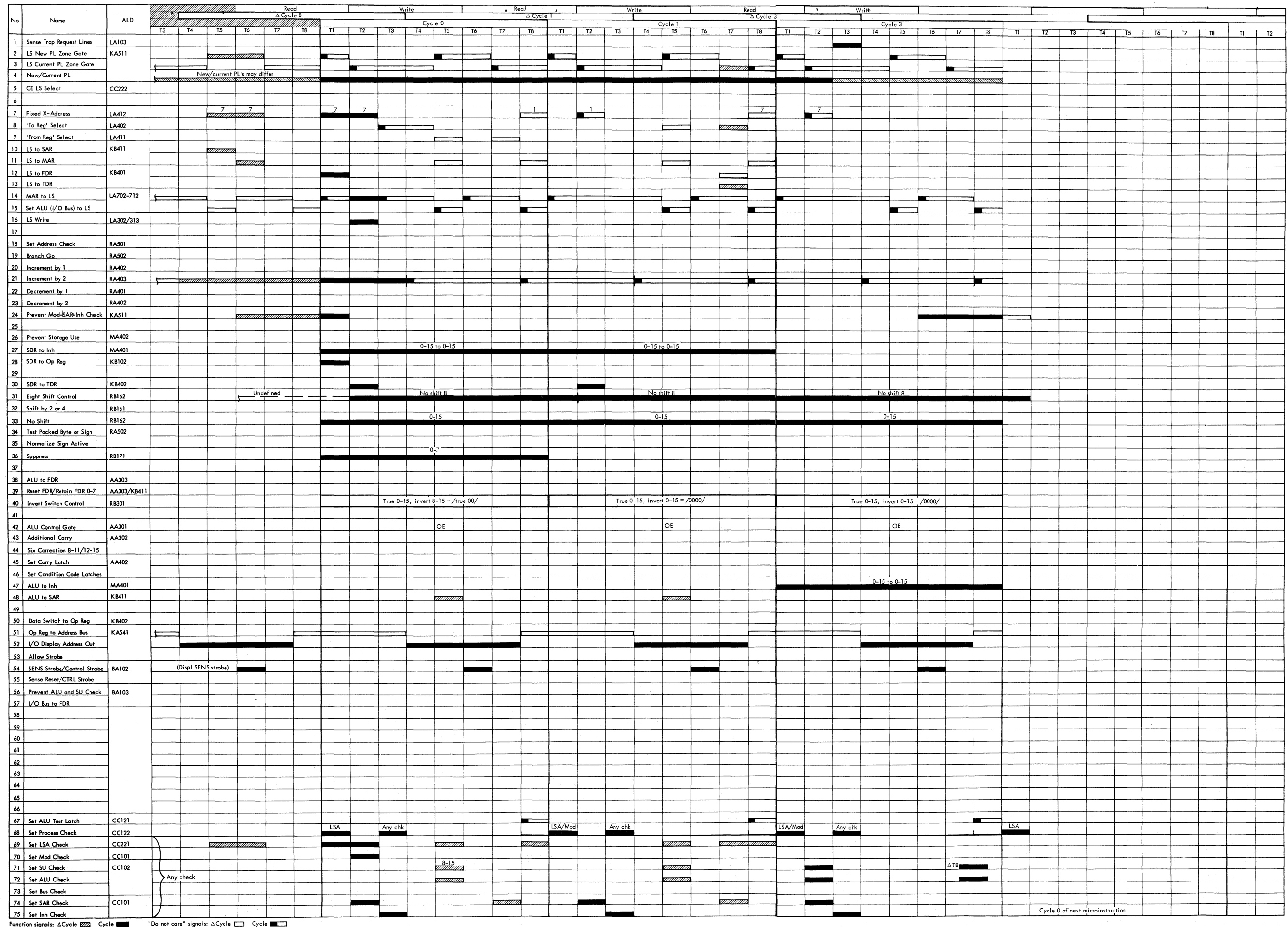
INST 59AB MNEM OPERANDS STH 1, LABEL, I STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX HW(HW(IAR, 0-7/INST, 8-14))=R1

Mnemonic	STH
Format	
RD	
Type	
Indirect Addressing	



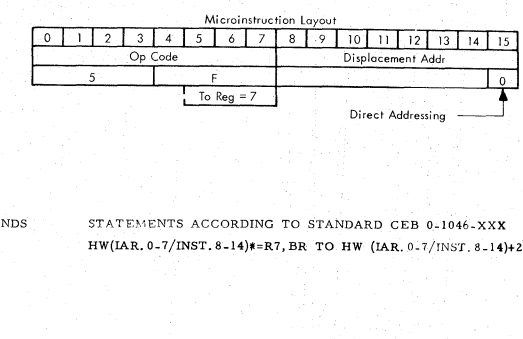
Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle



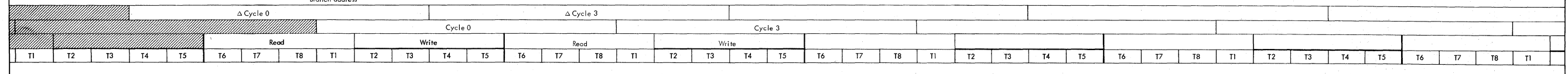
● Diagram 5-15. Store Halfword (Indirect Addressing) (Part 2 of 2) (03720A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

The updated microinstruction address (IAR + 2) is stored in core storage. The store address comprises a combination of the current block address and the displacement address (low-order byte). The store address is incremented by 2 and used to read out the next microinstruction (branch to store address + 2). The contents of LS register 4 are destroyed.



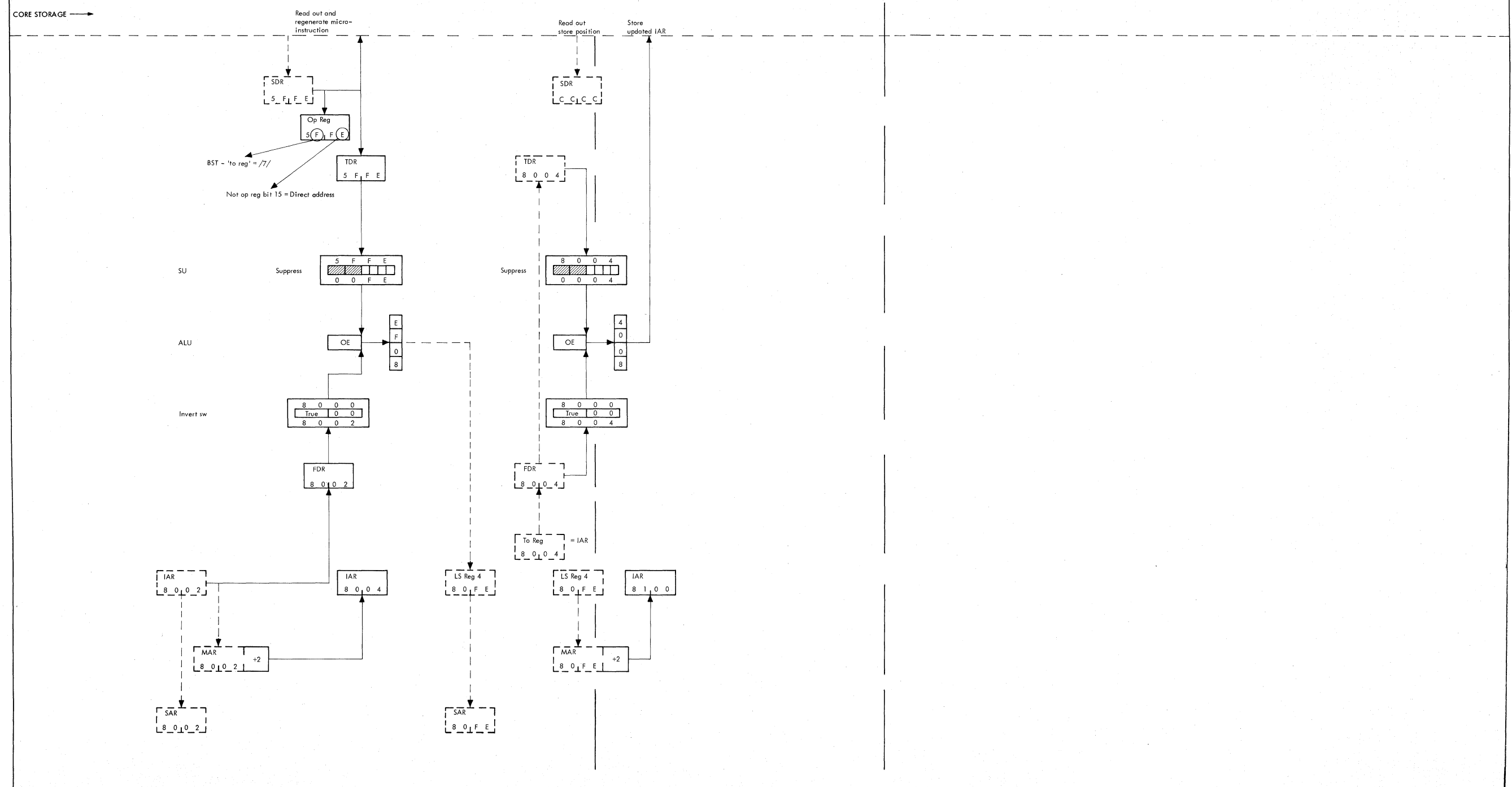
INST 5FFE MNEM BST OPERANDS LABEL STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX HW(IAR.0-7/INST.8-14)*R7, BR TO HW (IAR.0-7/INST.8-14)+2

Mnemonic	BST
Format	RD
Type	Direct Addressing

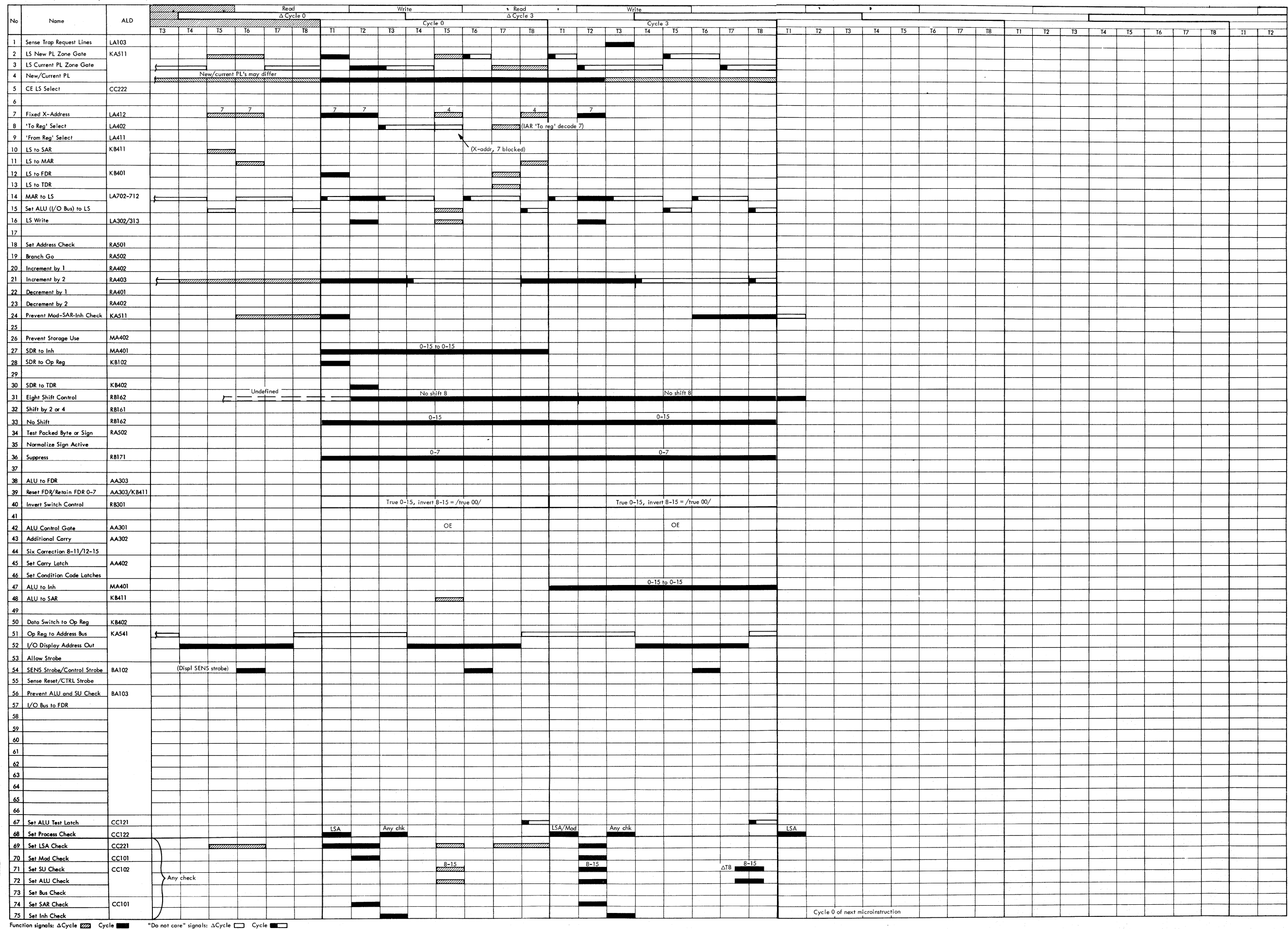


Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

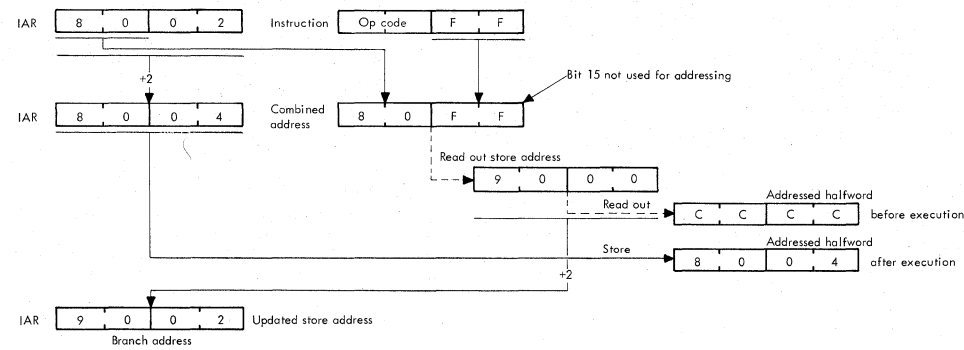


Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



Function signals: Δ Cycle Cycle "Do not care" signals: Δ Cycle Cycle

The updated microinstruction address (IAR + 2) is stored in core storage.
 The store address is read out by an address comprising a combination of the current block address and the displacement address (low-order byte).
 The store address is incremented by 2 and used to read out the next microinstruction (branch to store address + 2). The contents of LS register 4 are destroyed.



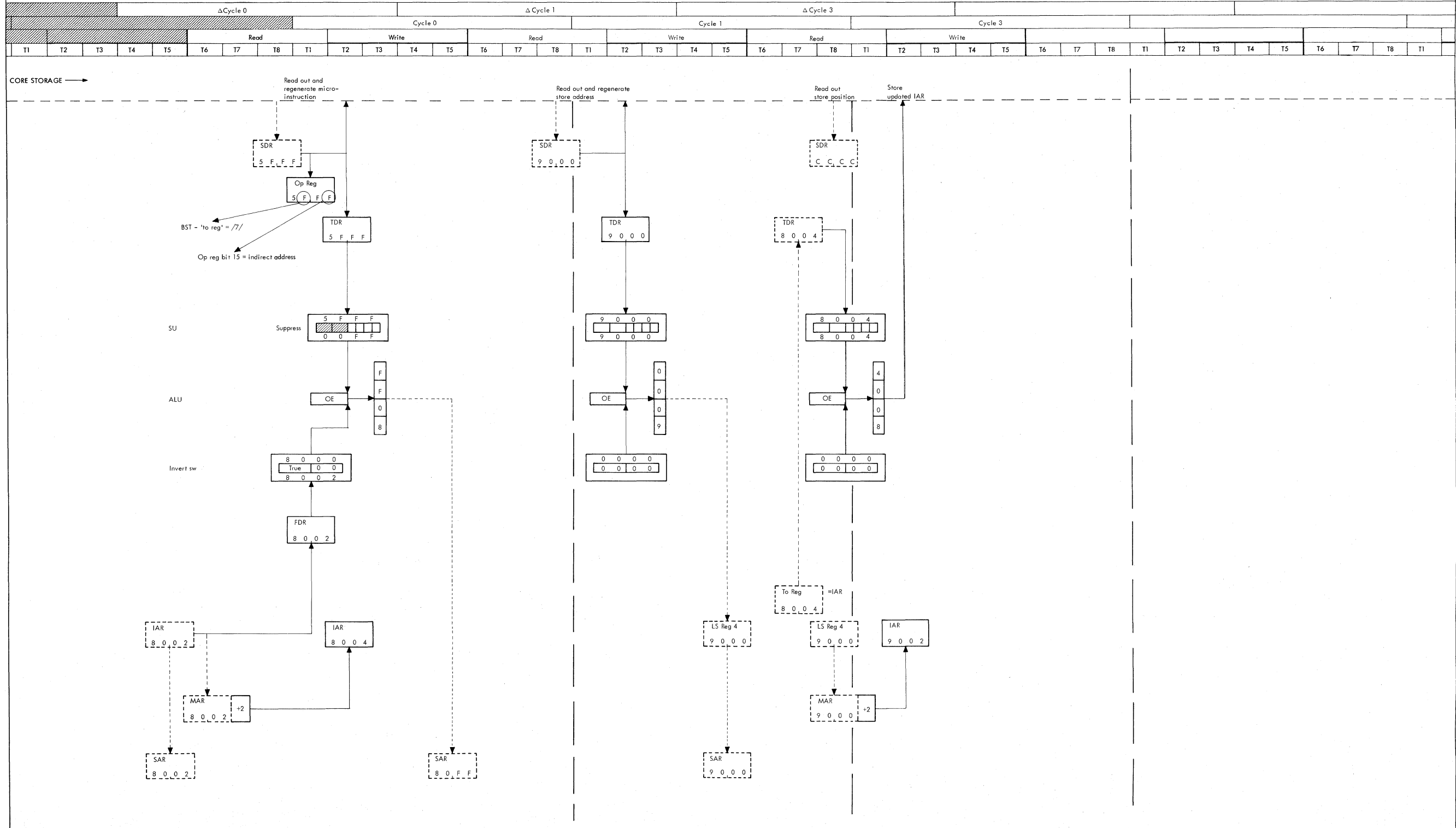
Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code								Displacement Addr							
5								F							
								To Reg = 7							

INST 5FFF MNEM BST OPERANDS LABEL, 1 STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
 HW (HW(IAR, 0-7/INST, 8-14)) += R7, BR TO HW(IAR, 0-7/INST, 8-14)+2

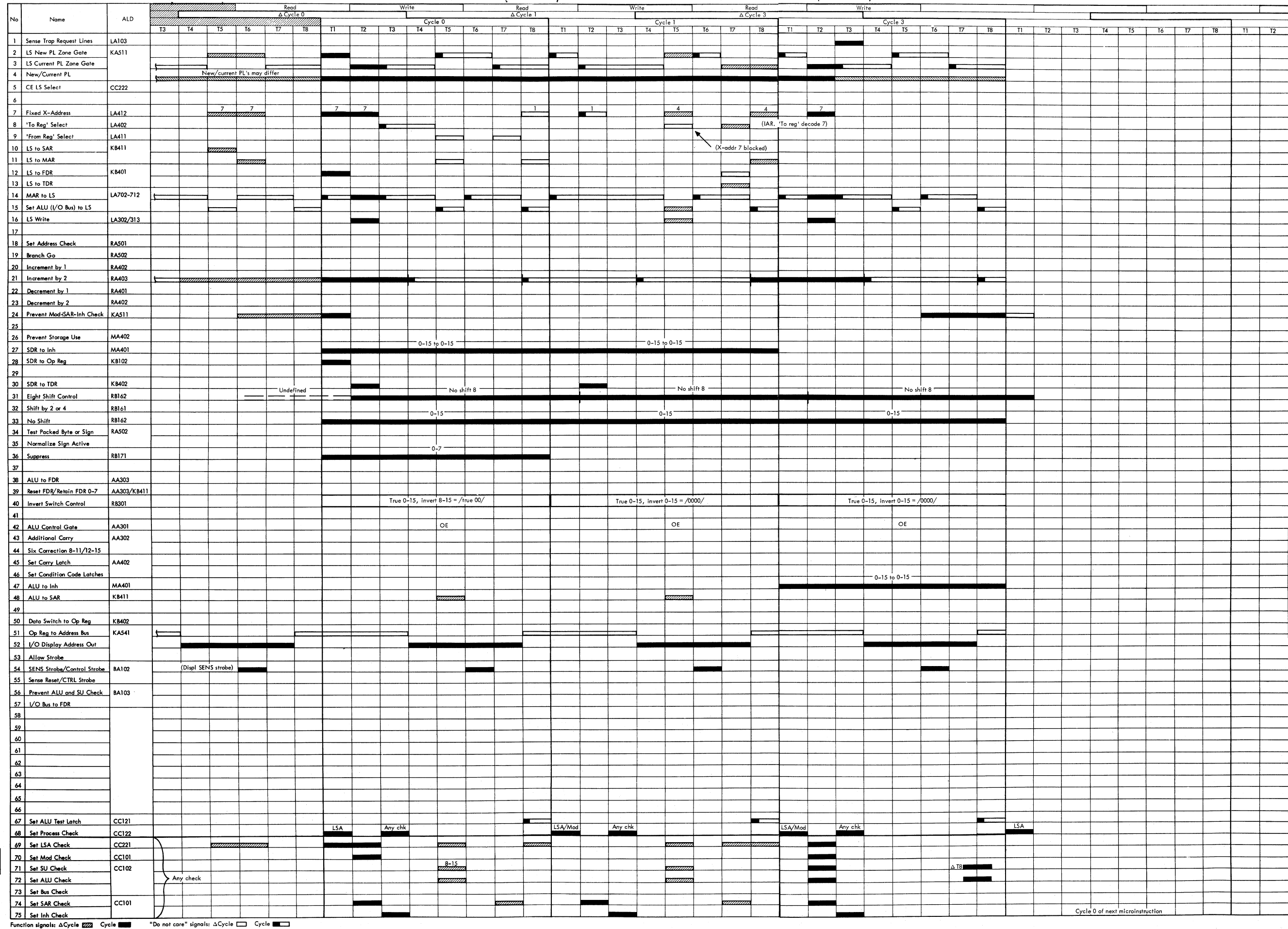
Mnemonic	BST
Format	RD
Type	Indirect Addressing

Note: For "Do not care" functions refer to timing chart below.

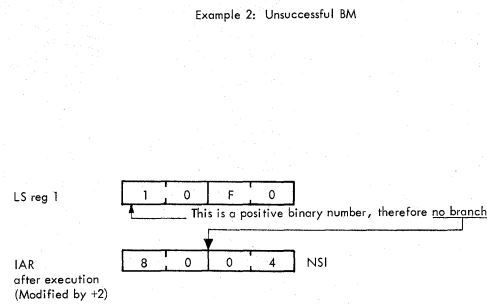
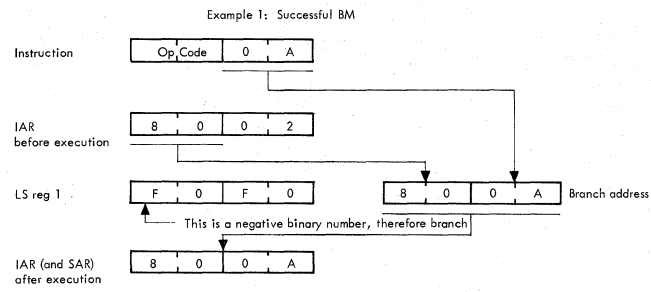
"Do not care" signals:
 ΔCycle
 Cycle



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



The 'to reg' is tested for being zero, minus, plus (not zero), or on invalid address (outside customer area). Successful testing causes a branch. The branch address is a combination of the current block address and the displacement address (low-order byte). If no branch, the updated microinstruction address is used.



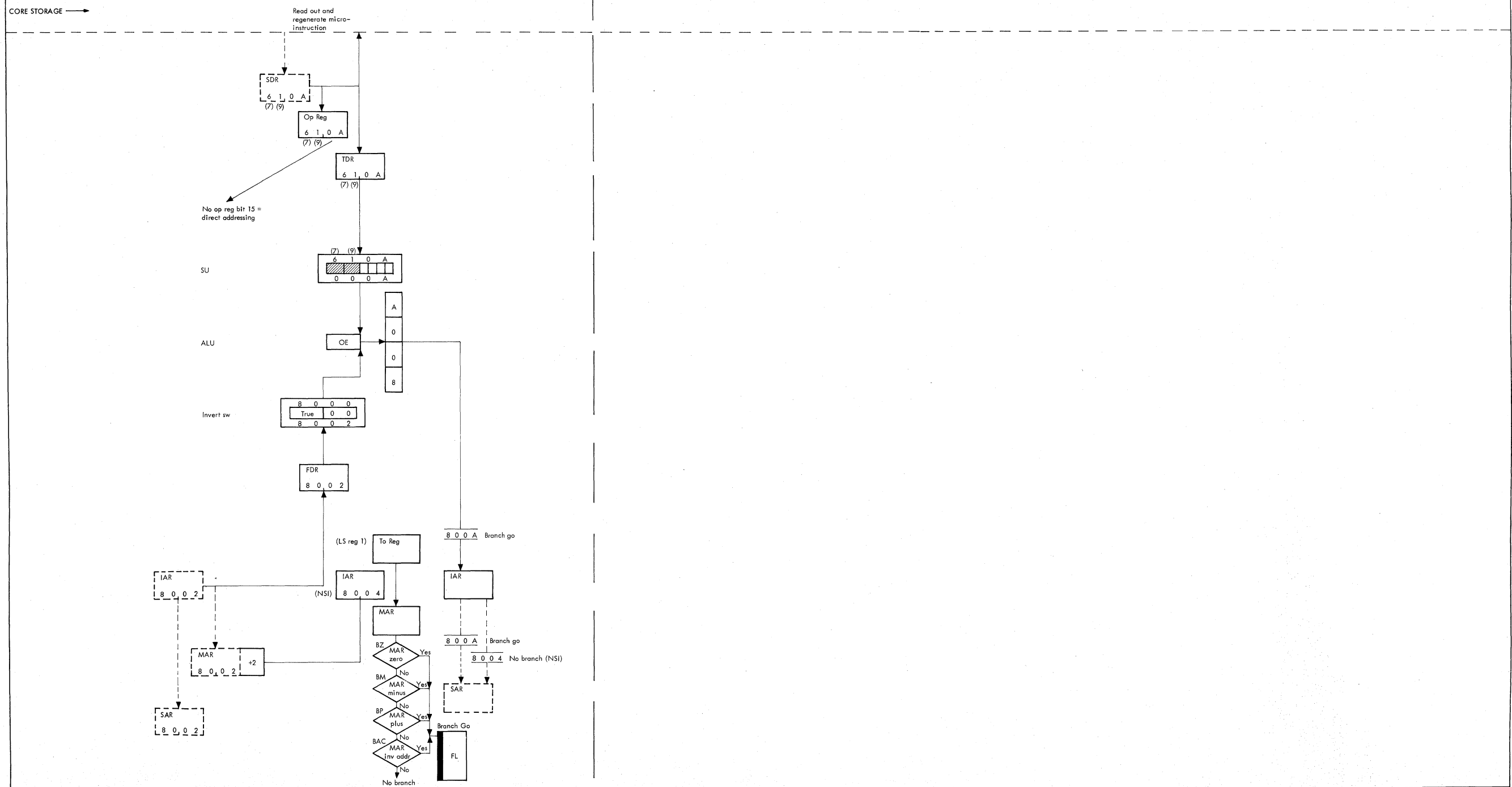
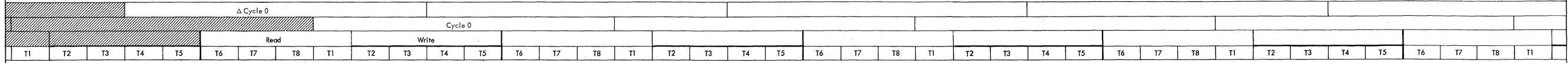
Microinstruction Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code					Displacement Addr.										
6	0														0
6	1														0
7	0														0
7	1														0

Direct addressing

INST	MNEM	OPERANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
610A	BZ	1, LABEL	BR TO IAR, 0-7/INST, 8-14 IF R1=0
690A	BM	1, LABEL	BR TO IAR, 0-7/INST, 8-14 IF R1, LT, 0
710A	BP	1, LABEL	BR TO IAR, 0-7/INST, 8-14 IF R1, GT, 0
790A	BAC	1, LABEL	BR TO IAR, 0-7/INST, 8-14 IF R1 INVALID

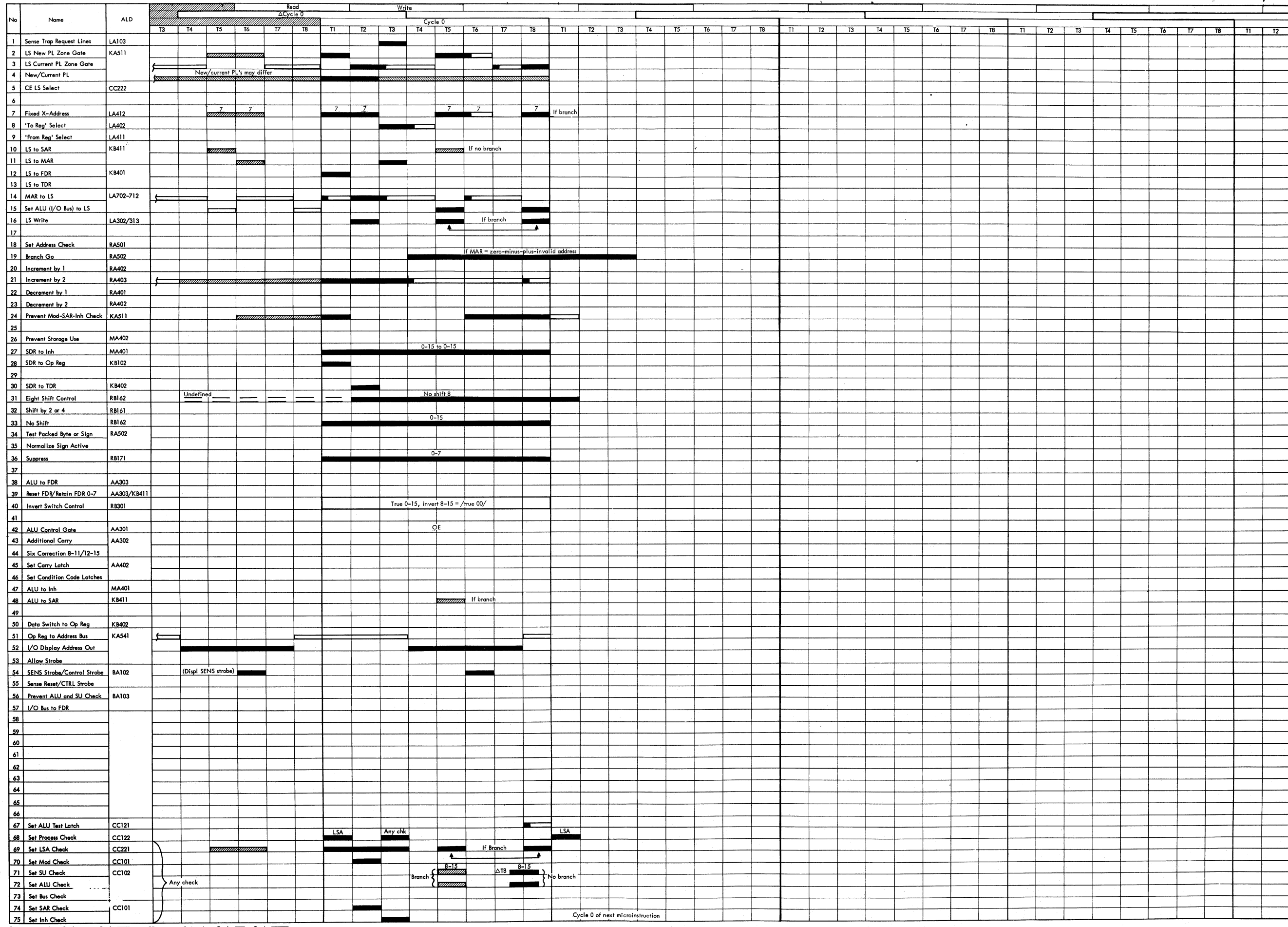
BZ
BM
BP
BAC
Format
RD
Type
Direct Addressing



Note: For "Do not care" functions refer to timing chart below.

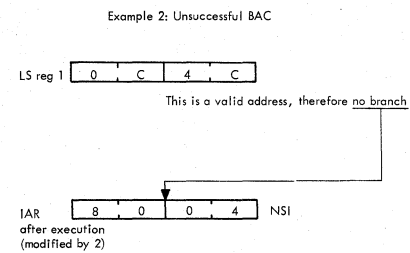
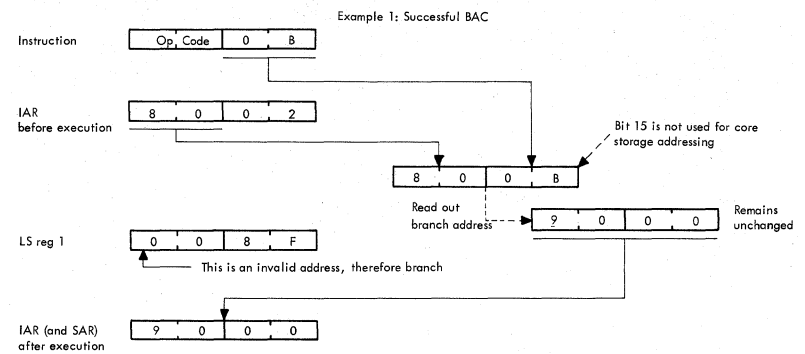
"Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



Function signals: Δ Cycle Cycle "Do not care" signals: Δ Cycle Cycle

The 'to reg' is tested for being zero, minus, plus (not zero), or an invalid address (outside customer area).
 Successful testing causes a branch. The branch address is read out by an address comprising a combination of the current block address and the displacement address (low-order byte). If no branch the update microinstruction address is used.

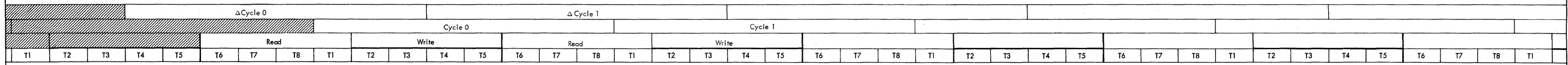


Microinstruction Layout

Op Code	To Reg	Displacement Addr
6	0	1
6	1	1
7	0	1
7	1	1

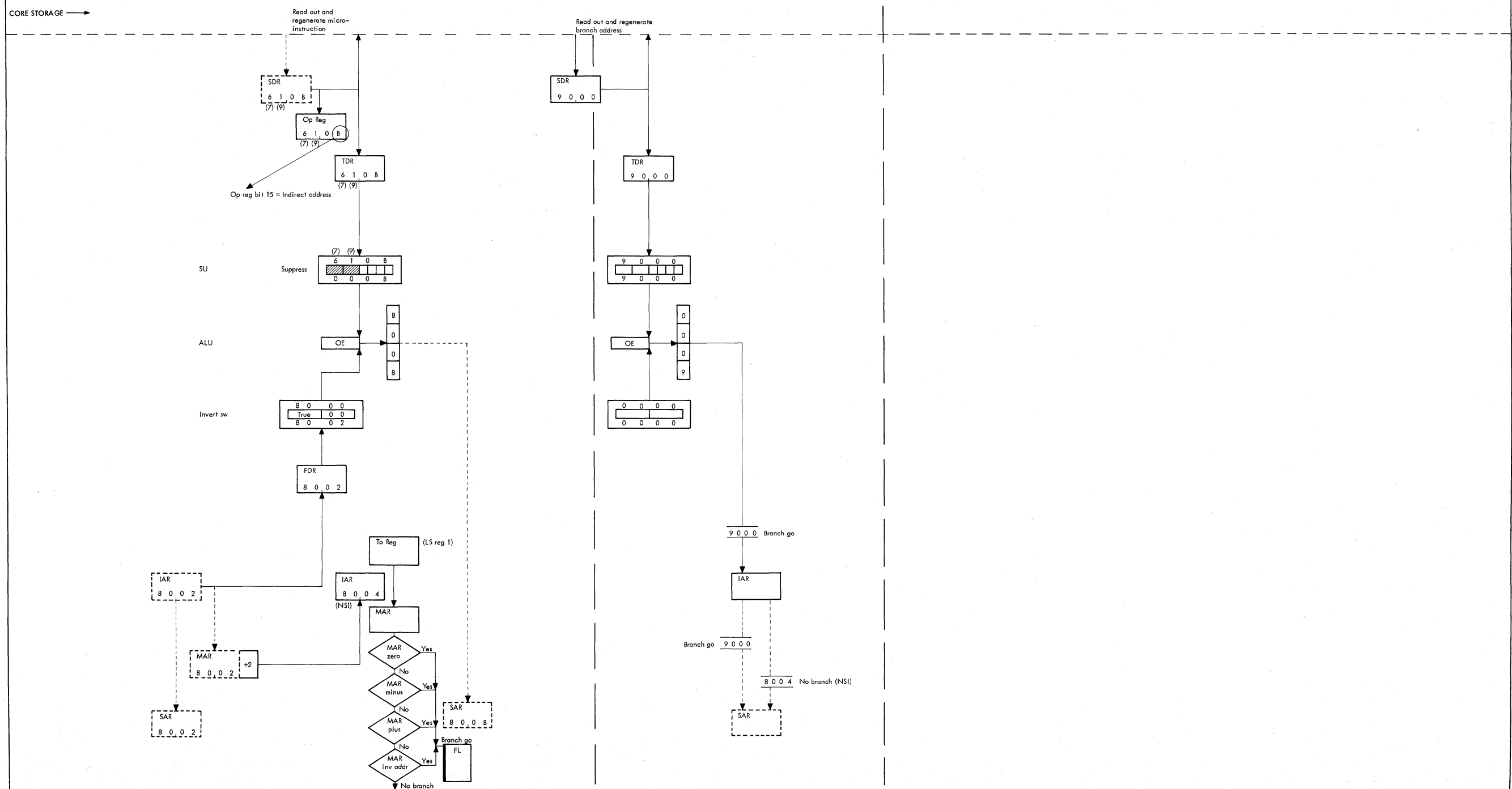
INST	MNEM	OPERANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
610B	BZ	1, LABEL, I	BR TO HW (IAR. 0-7/INST. 8-14) IF R1=0
690B	BM	1, LABEL, I	BR TO HW (IAR. 0-7/INST. 8-14) IF R1, LT, 0
710B	BP	1, LABEL, I	BR TO HW (IAR. 0-7/INST. 8-14) IF R1, GT, 0
790B	BAC	1, LABEL, I	BR TO HW (IAR. 0-7/INST. 8-14) IF R1 INVALID

Mnemonic
BZ
BM
BP
BAC
Format
RD
Type
Indirect Addressing

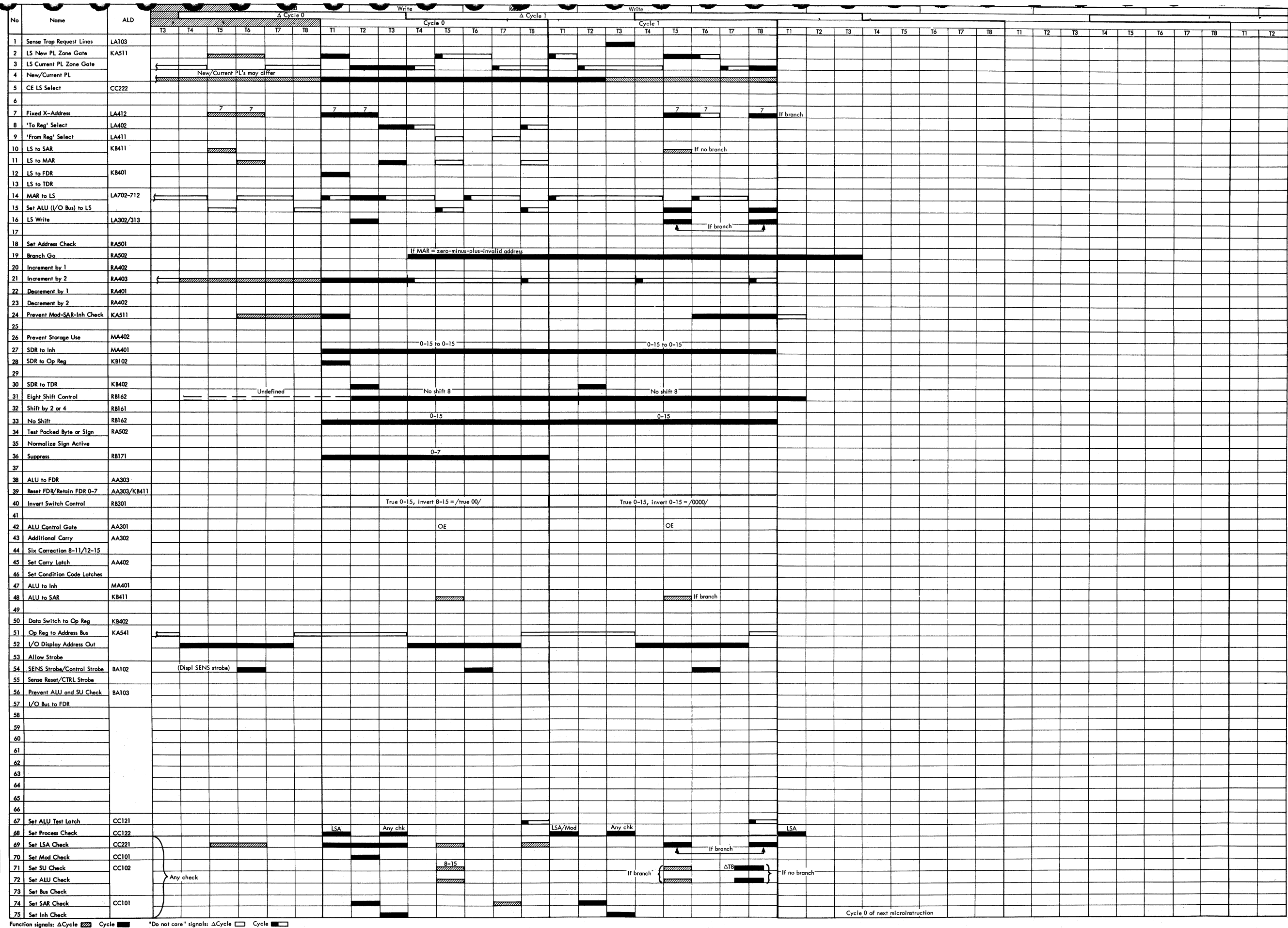


Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

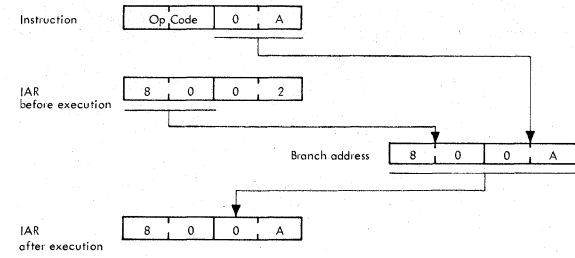


Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



Function signals: ΔCycle Cycle "Do not care" signals: ΔCycle Cycle

An unconditional branch is performed. The branch address is a combination of the current block address and the displacement address (low-order byte).

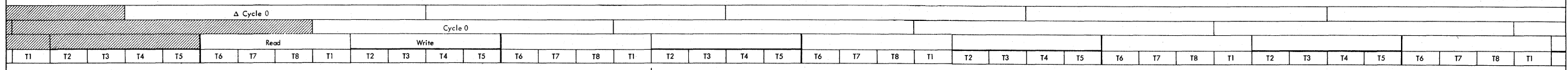


Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code								Displacement Addr							
8 0 0 0 0								0							
To Reg															

Direct Addressing

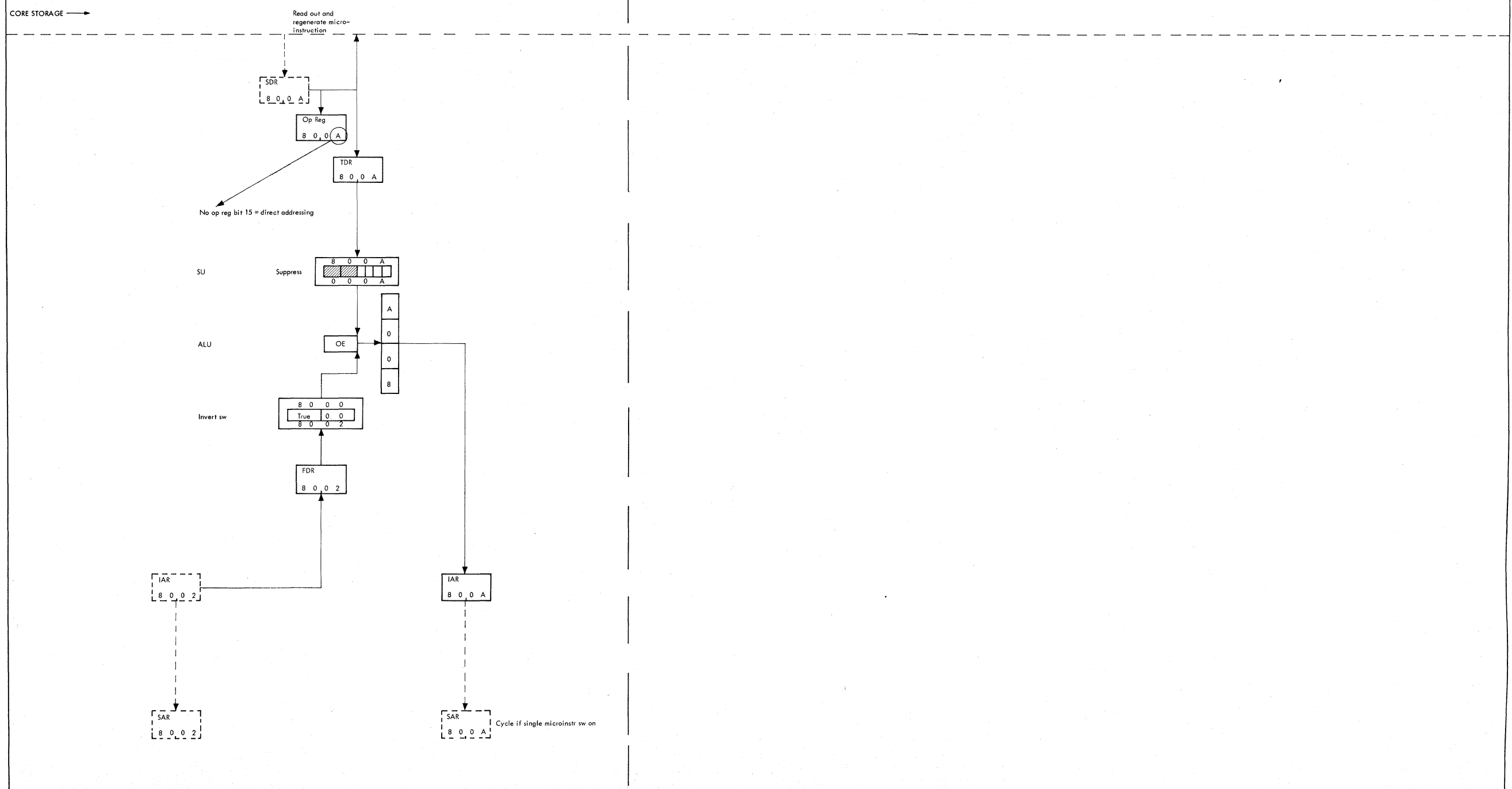
INST 800A MNEM B OPERANDS LABEL STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX BR TO IAR.0-7/INST.8-14

Mnemonic	B
Format	
RD	
Type	
Direct Addressing	

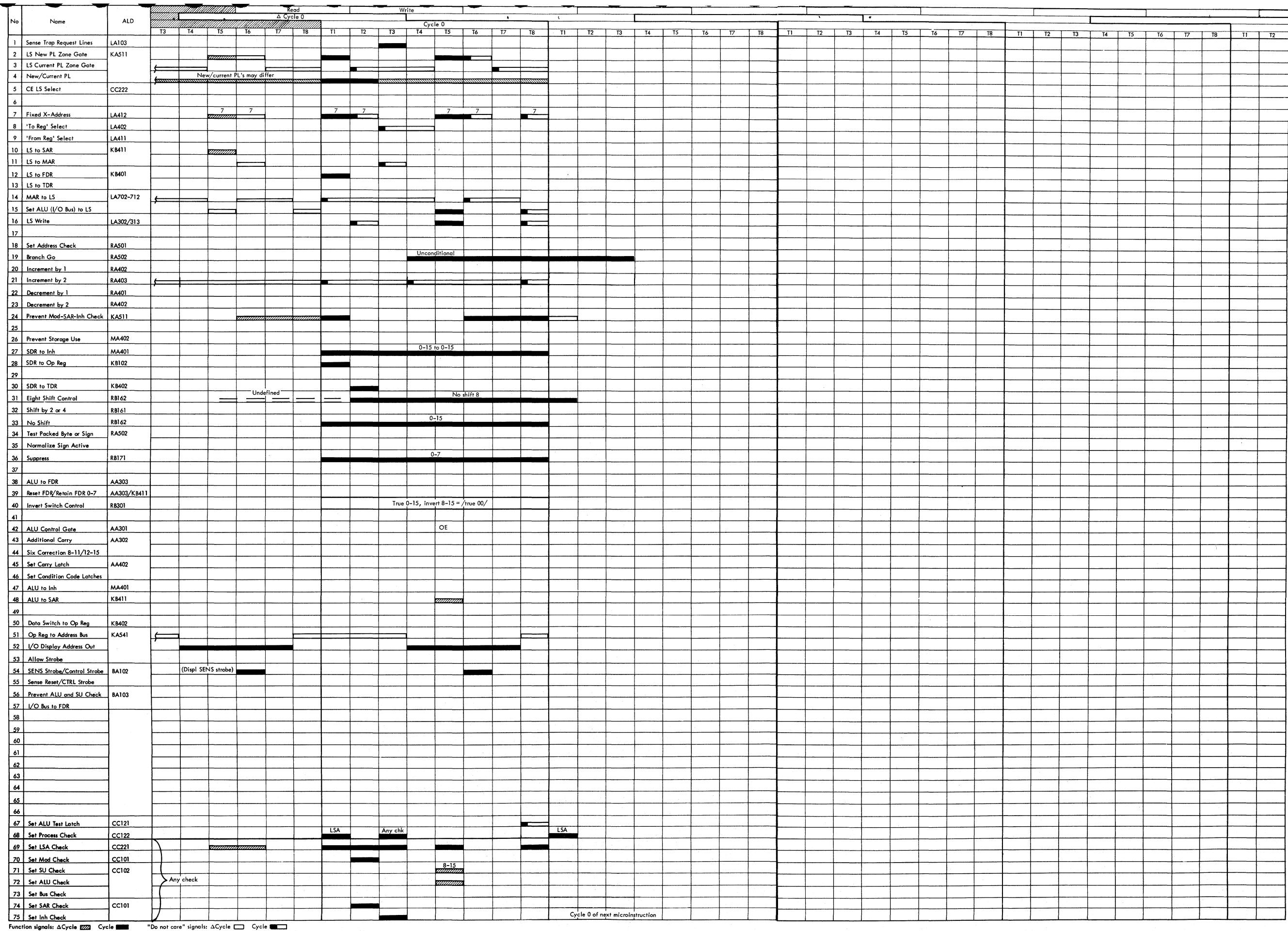


Note: For "Do not care" functions refer to timing chart below.

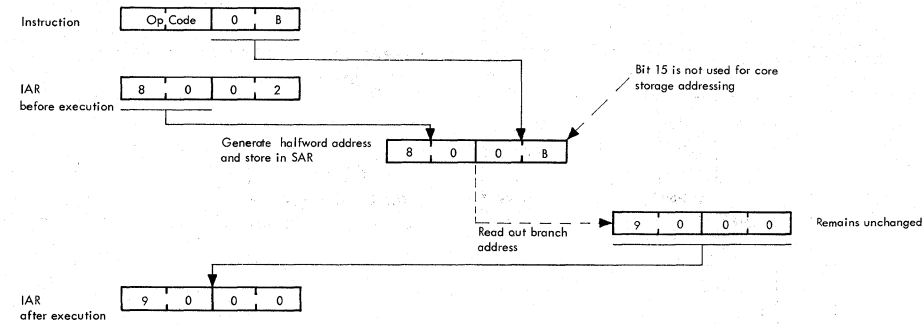
"Do not care" signals:
 ΔCycle
 Cycle



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



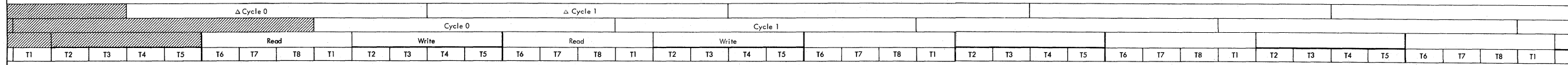
An unconditional branch is performed. The branch address is read out by an address comprising a combination of the current block address and the displacement address (low-order byte)



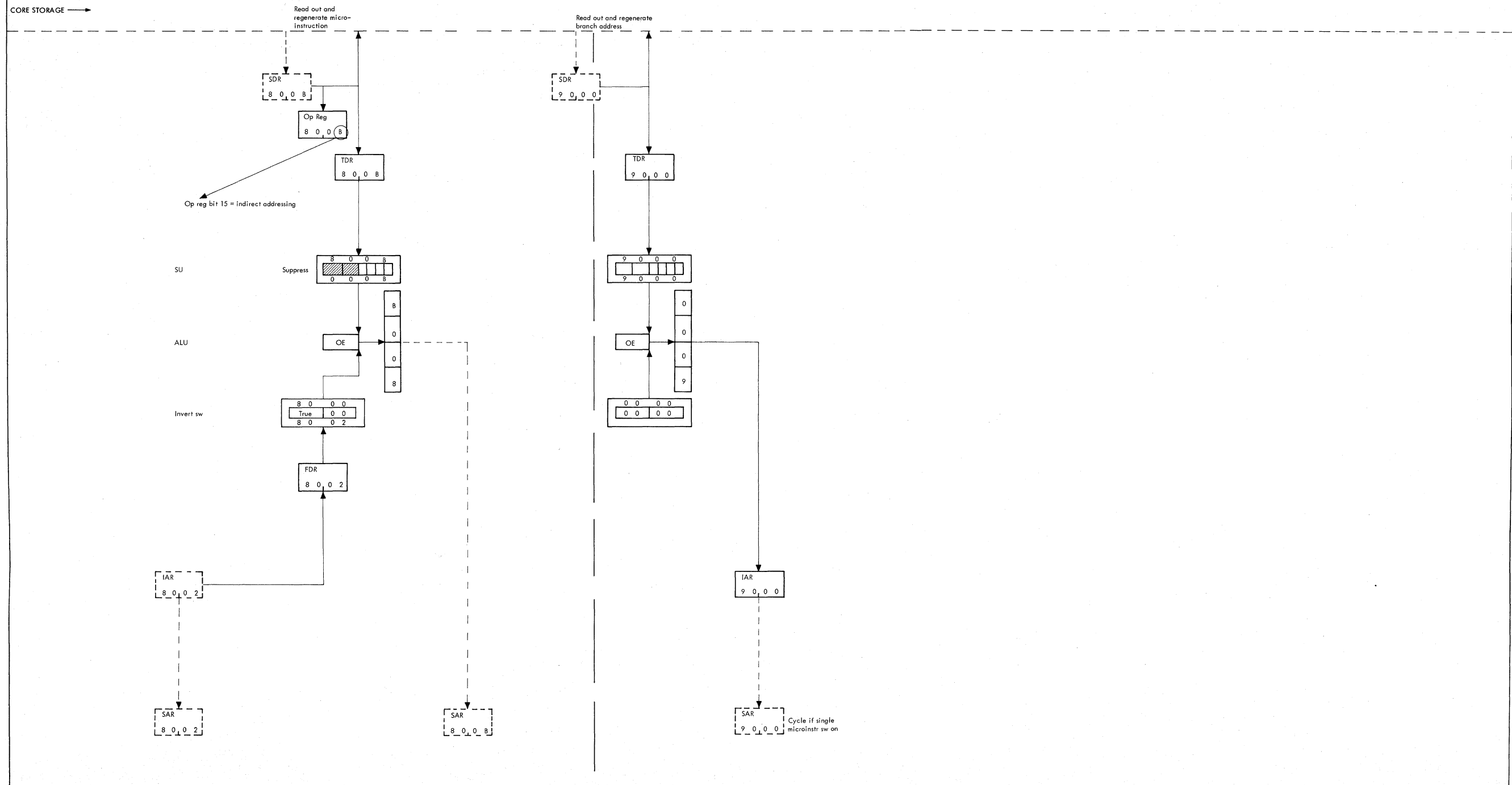
Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code								Displacement Add							
8	0	0	0	0	0	0	0								1

Mnemonic	B
Format	
RD	
Type	
Indirect Addressing	

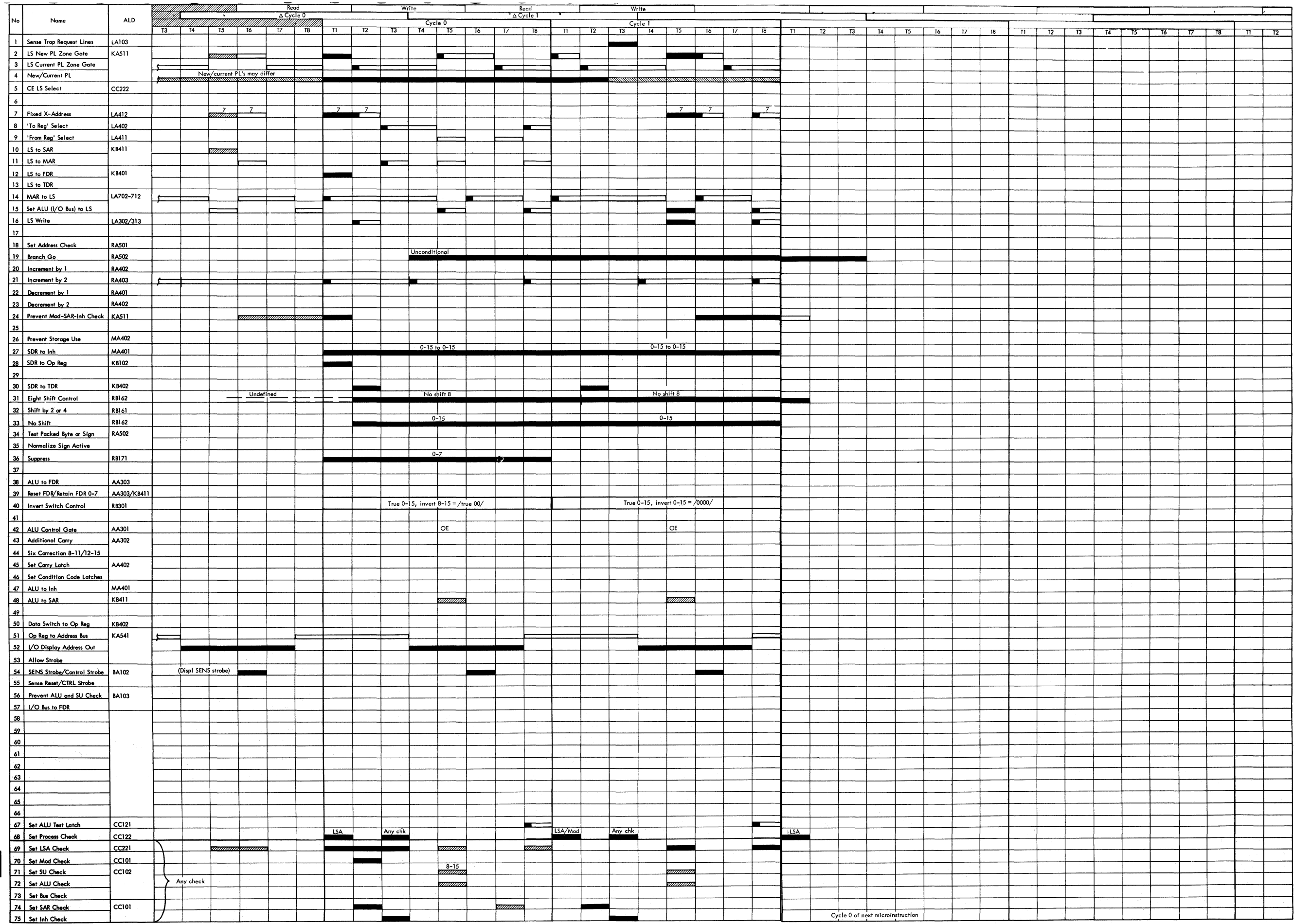
INST 800B MNEM B OPERANDS LABEL,I STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX BR TO HW (IAR,0-7/INST,8-14)



Note: For "Do not care" functions refer to timing chart below.
 "Do not care" signals:
 ΔCycle
 Cycle



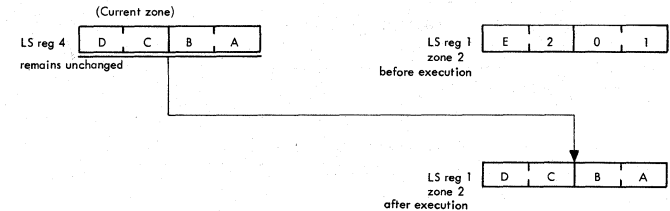
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



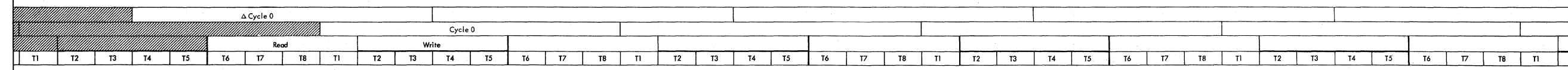
Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code								To Reg				From Reg			
8								0 0 1 0 0				0			
Zone															

Mnemonic	STR
Format	FF
Type	

LS register 4 of the current LS zone (defined by the current program level) is stored into the 'from reg' of the LS zone specified in the 'to reg' field.

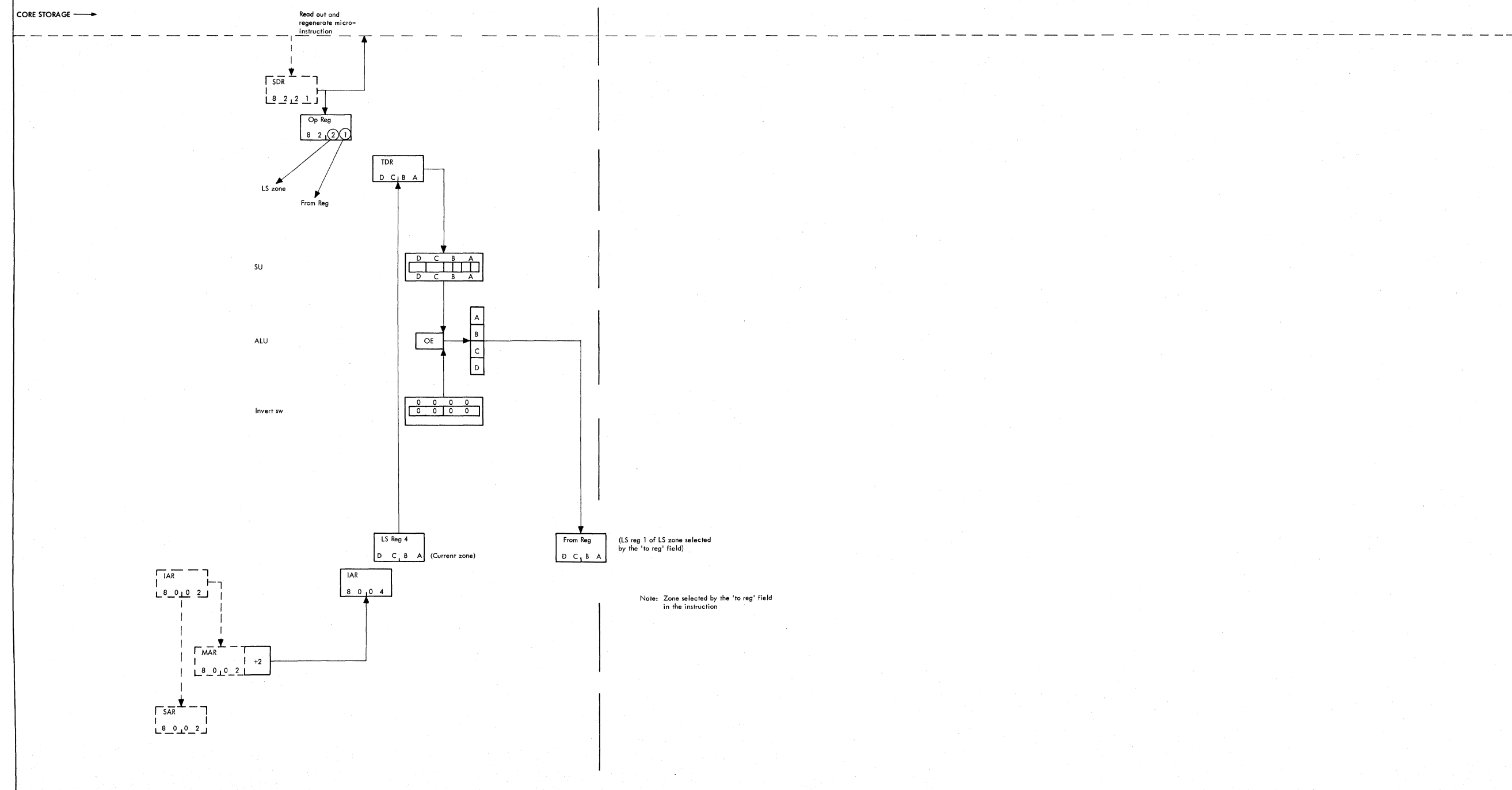


INST 8221 MNEM STR OPERANDS 2,1 STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX R1 PL2=R4

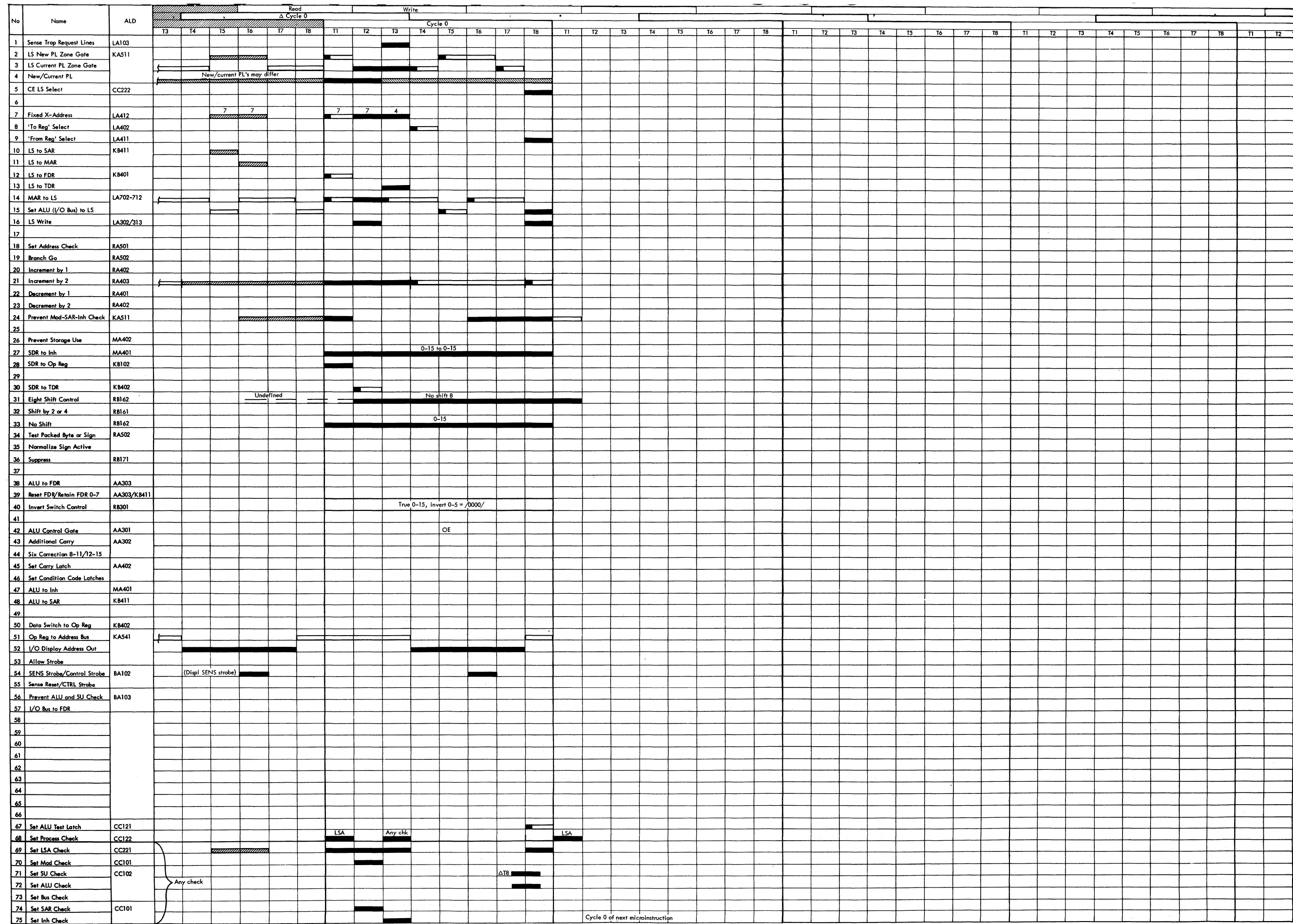


Note: For "Do not care" functions refer to timing chart below.

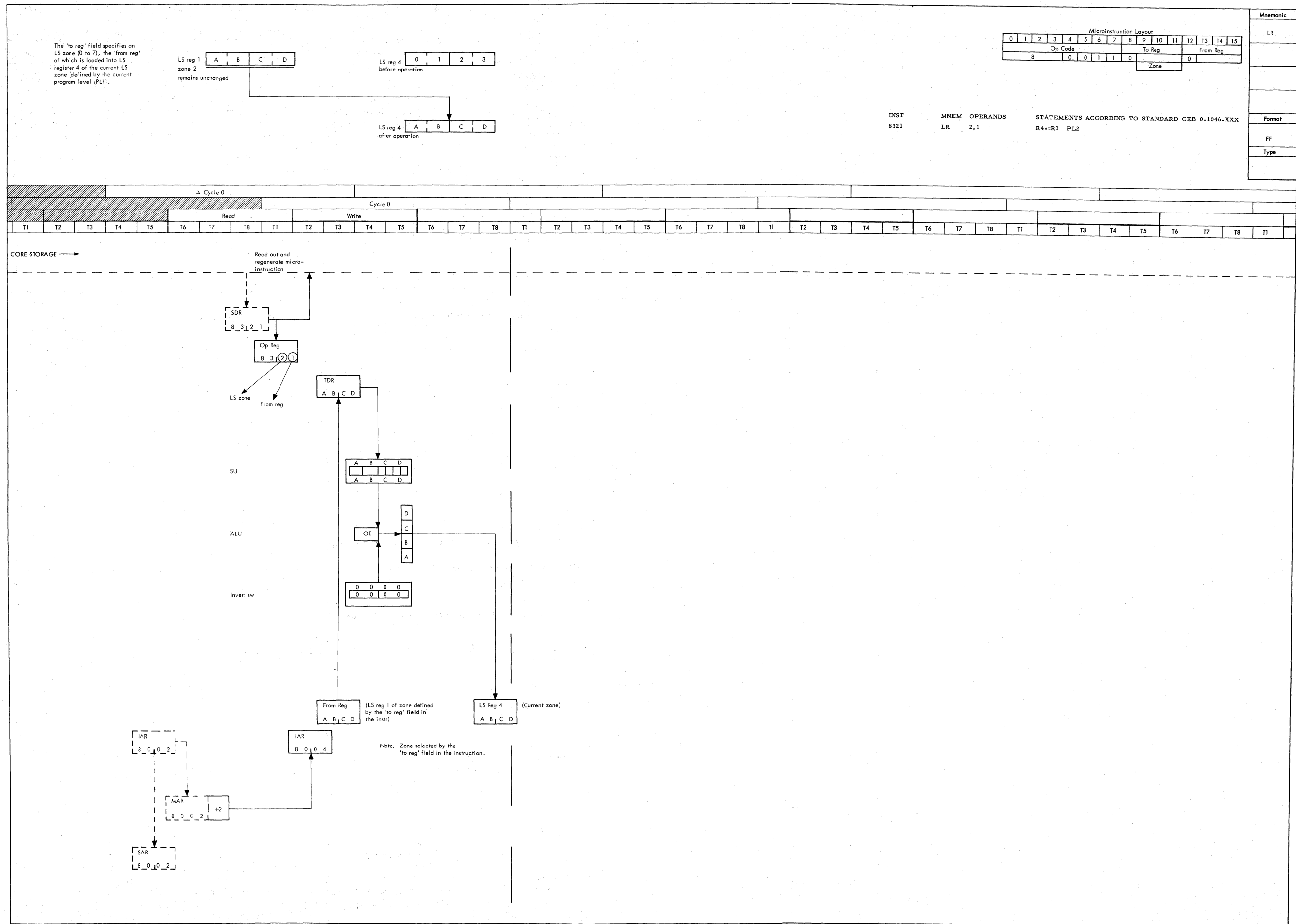
"Do not care" signals:
 ΔCycle
 Cycle



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



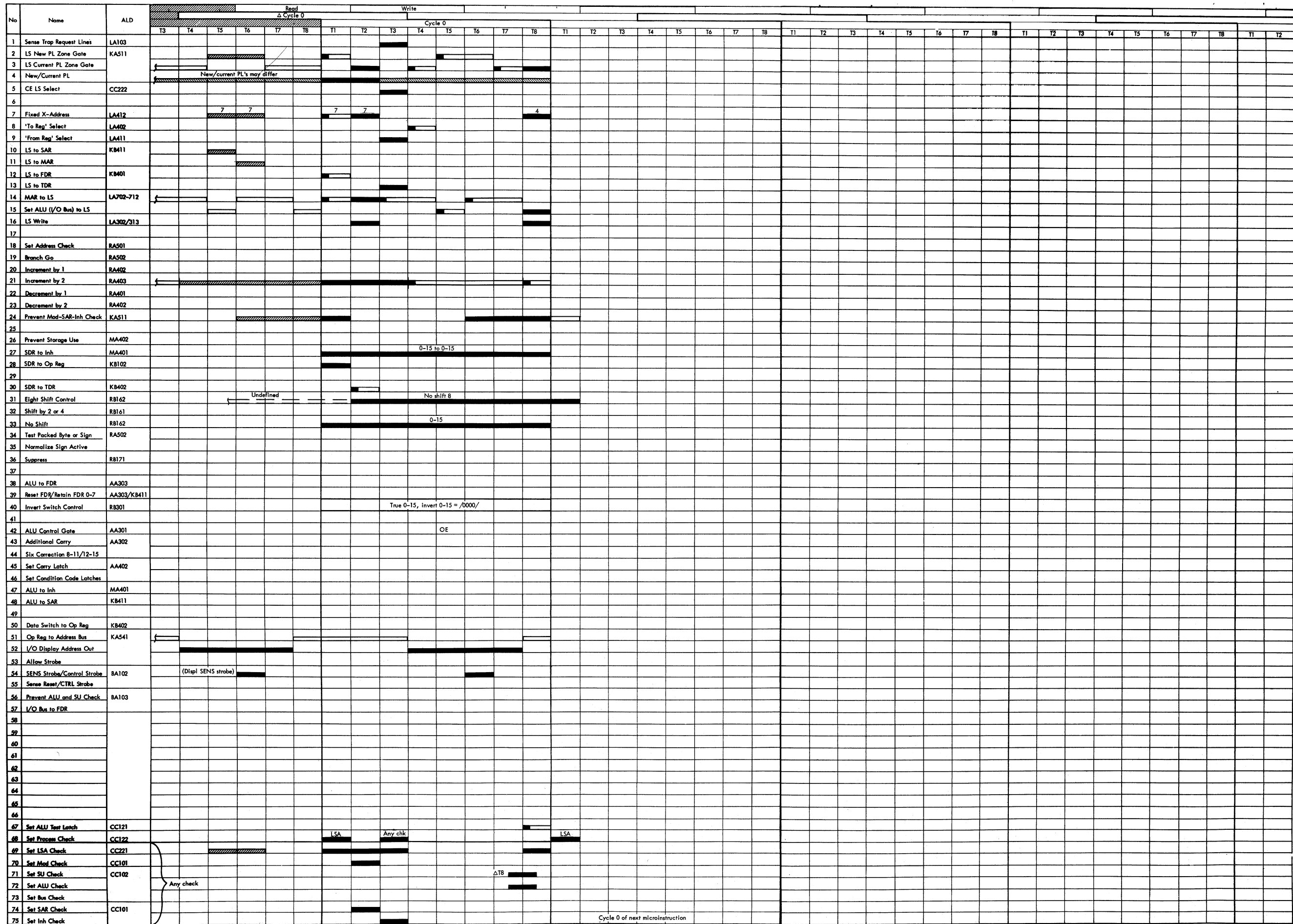
Function signals: ΔCycle Cycle "Do not care" signals: ΔCycle Cycle



Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

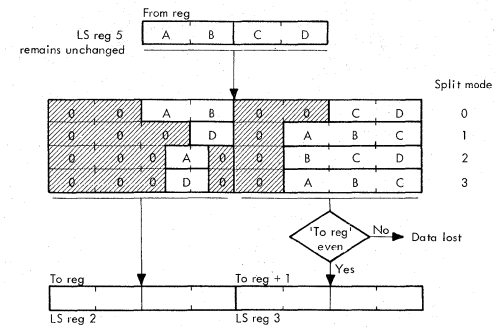
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



Function signals: ΔCycle Cycle "Do not care" signals: ΔCycle Cycle

The 'from reg' is split according to the selected split mode (0 to 3). The result of the split is set into the 'to reg' and the next highest LS register ('to reg' + 1). The 'to reg' must be an even numbered register (but not 0) otherwise the data to be placed into the next highest LS register will be lost.

Address check is ignored. The 'from reg' remains unchanged. 'From reg' and 'to reg' may be the same LS register.



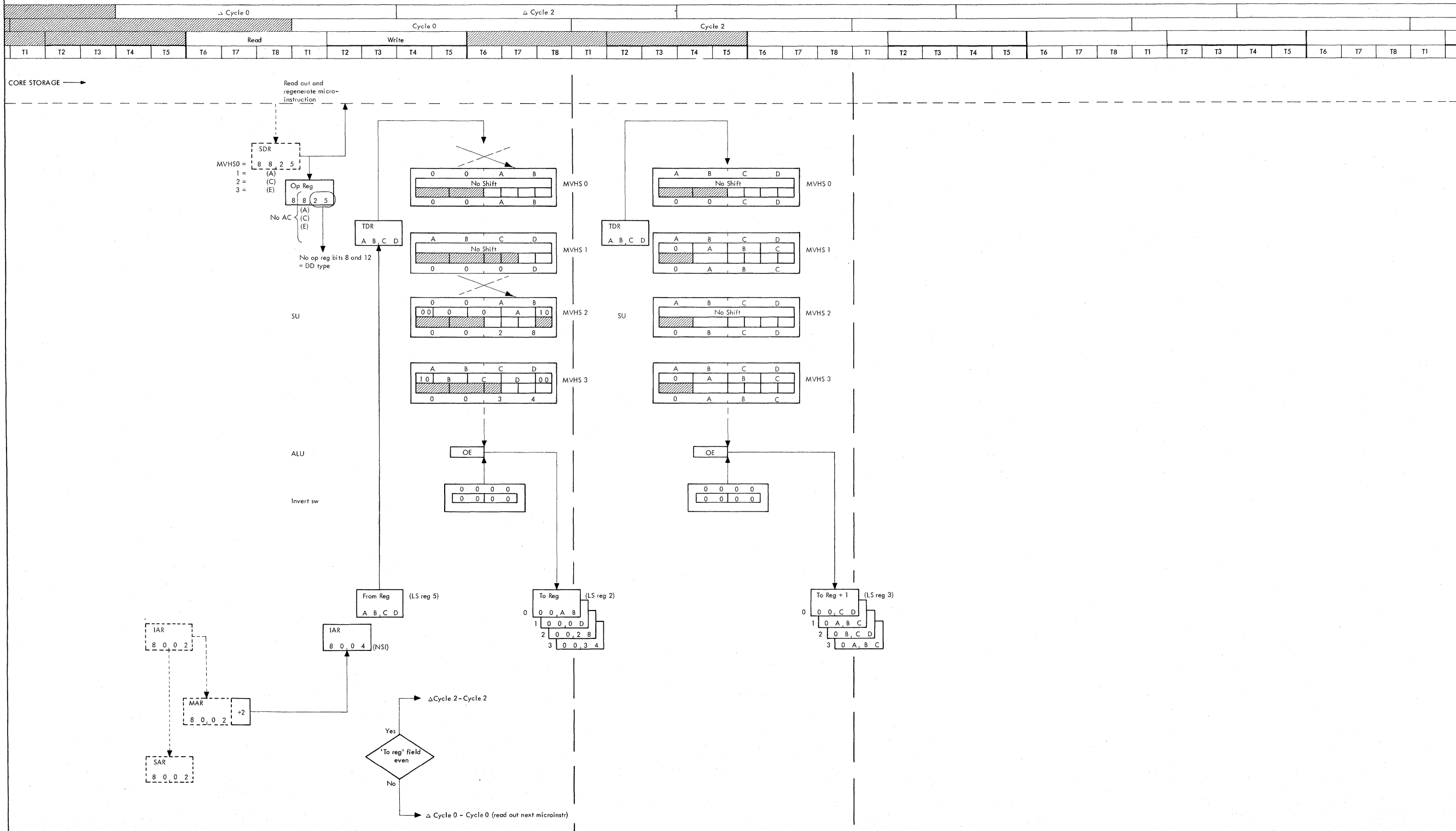
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code				Split	AC	To Reg				From Reg					
8				1	0	0	0				0				
				0	1	DD				Without effect					
				1	0										
				1	1										

INST	MNEM	OPERANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
8A25	MVHS	2, 5, 0	R2 = 1001/R5, 0-7, R3 = 1001/R5, 8-15
8A25	MVHS	2, 5, 1	R2 = 10001/R5, 12-15, R3 = 101/R5, 0-11
8C25	MVHS	2, 5, 2	R2 = SL 2 10001/R5, 0-3, R3 = 101/R5, 4-15
8E25	MVHS	2, 5, 3	R2 = SL 2 10001/R5, 12-15, R3 = 101/R5, 0-11

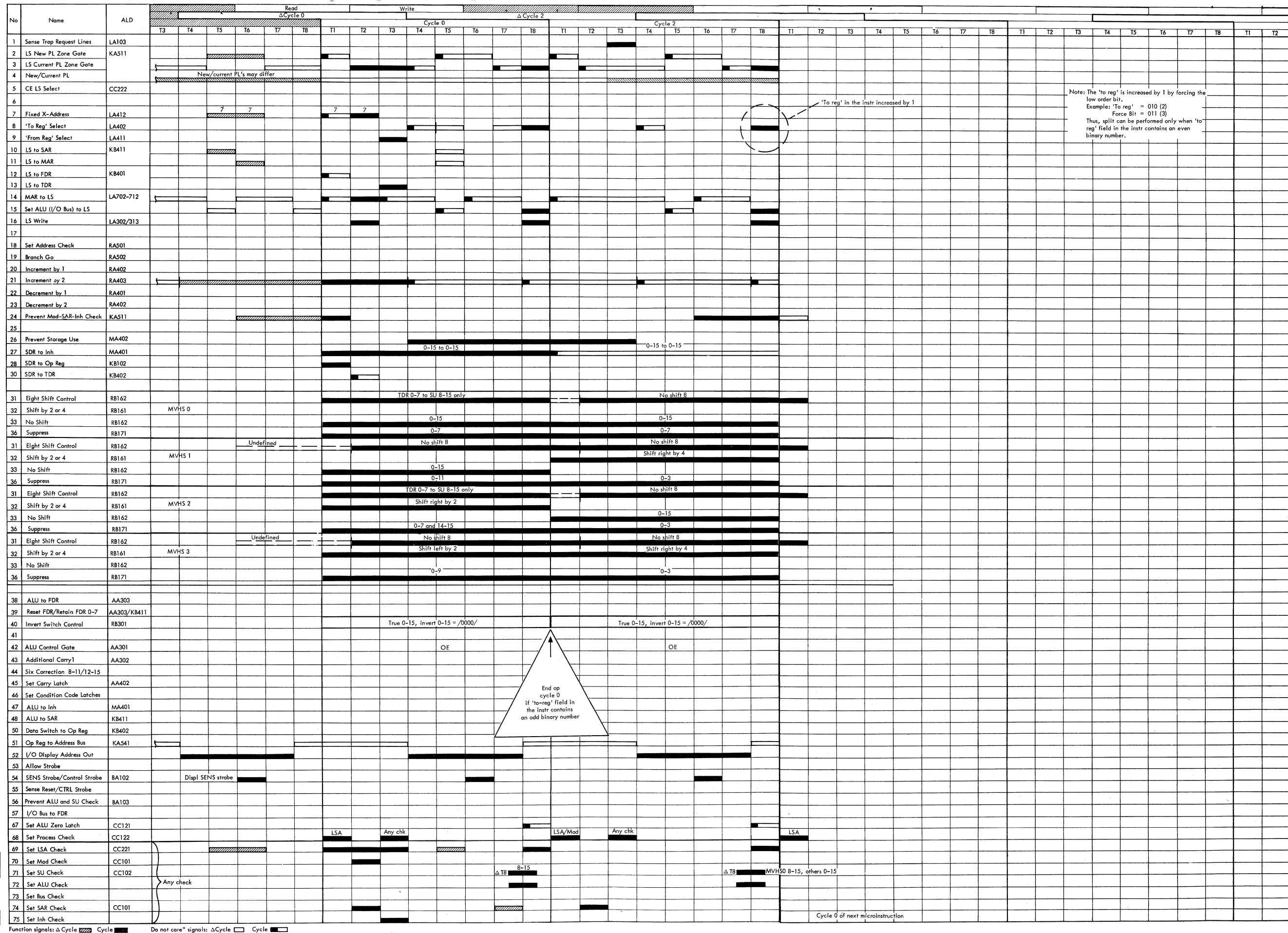
Mnemonic	Format	Type
MVHS		
MVHS 0		
MVHS 1		
MVHS 2		
MVHS 3		
FF		
DD		

Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle



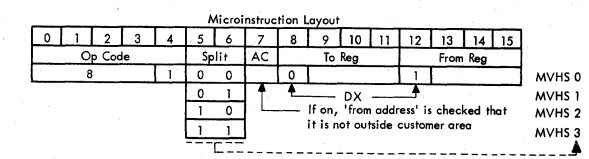
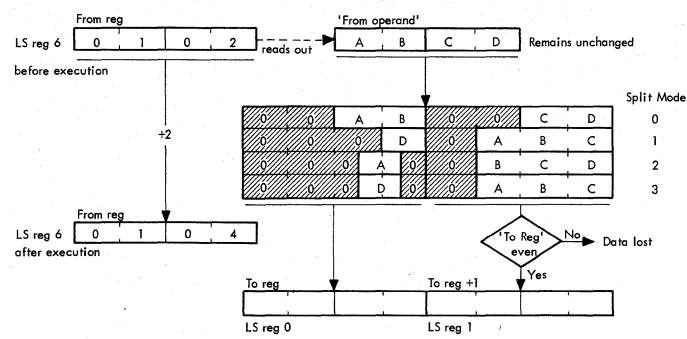
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



● Diagram 5-24. Move Halfword and Split (DD) (Part 2 of 2) (03729A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

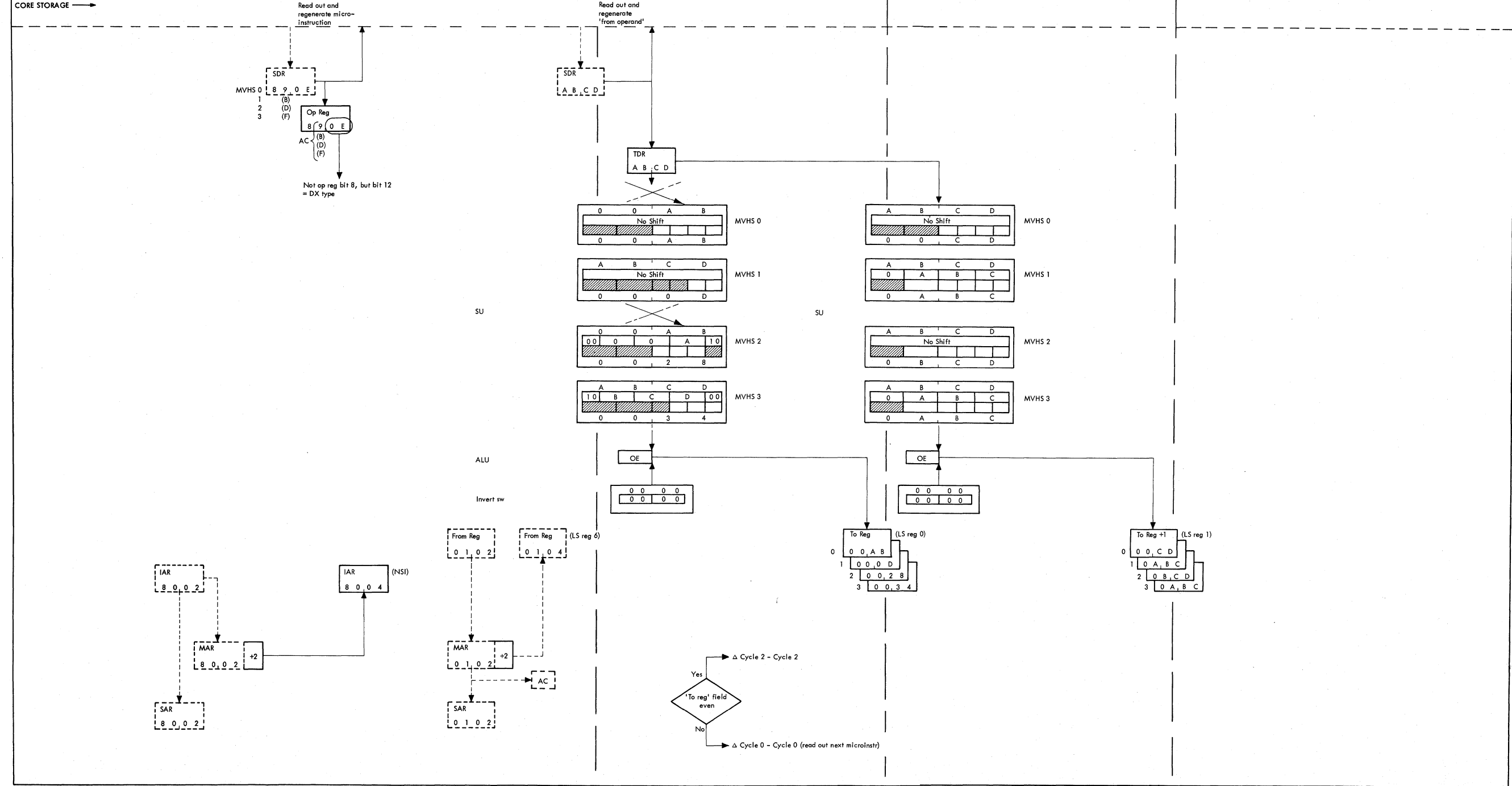
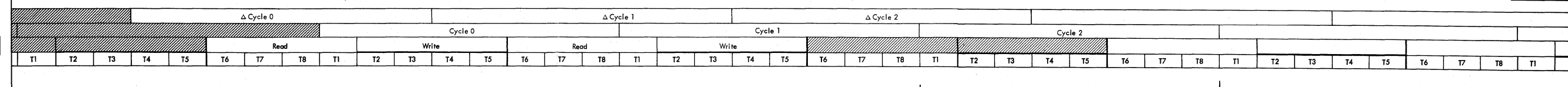
The halfword read out by the 'from reg' is split according to the selected split mode (0-3). The result of the split is set into 'to reg' and the next higher LS register ('to reg' + 1). The 'to reg' must be an even numbered register (but not 6) otherwise the data to be placed into the next higher LS register is lost.

If the AC bit (instruction bit 7) is on, an address check occurs when the 'from reg' address is outside customer area. The 'from reg' address is incremented by 2. The core storage halfword read out by the 'from reg' remains unchanged.



INST	MNEM	OPERANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
890E	MVHS	0,6I,0,AC	R0 \rightarrow 00 ⁰ /HW(R6,AC), 0-7, R1 \rightarrow 00 ⁰ /HW(R6,+2), 8-15
8B0E	MVHS	0,6I,1,AC	R0 \rightarrow 00 ⁰ /HW(R6,AC), 12-15, R1 \rightarrow 0 ⁰ /HW(R6,+2), 0-11
8D0E	MVHS	0,6I,2,AC	R0 \rightarrow 8L 2 ⁰⁰⁰ /HW(R6,AC), 0-3, R1 \rightarrow 0 ⁰ /HW(R6,+2), 4-15
8F0E	MVHS	0,6I,3,AC	R0 \rightarrow 5L 2 ⁰⁰⁰ /HW(R6,AC), 12-15, R1 \rightarrow 0 ⁰ /HW(R6,+2), 0-11

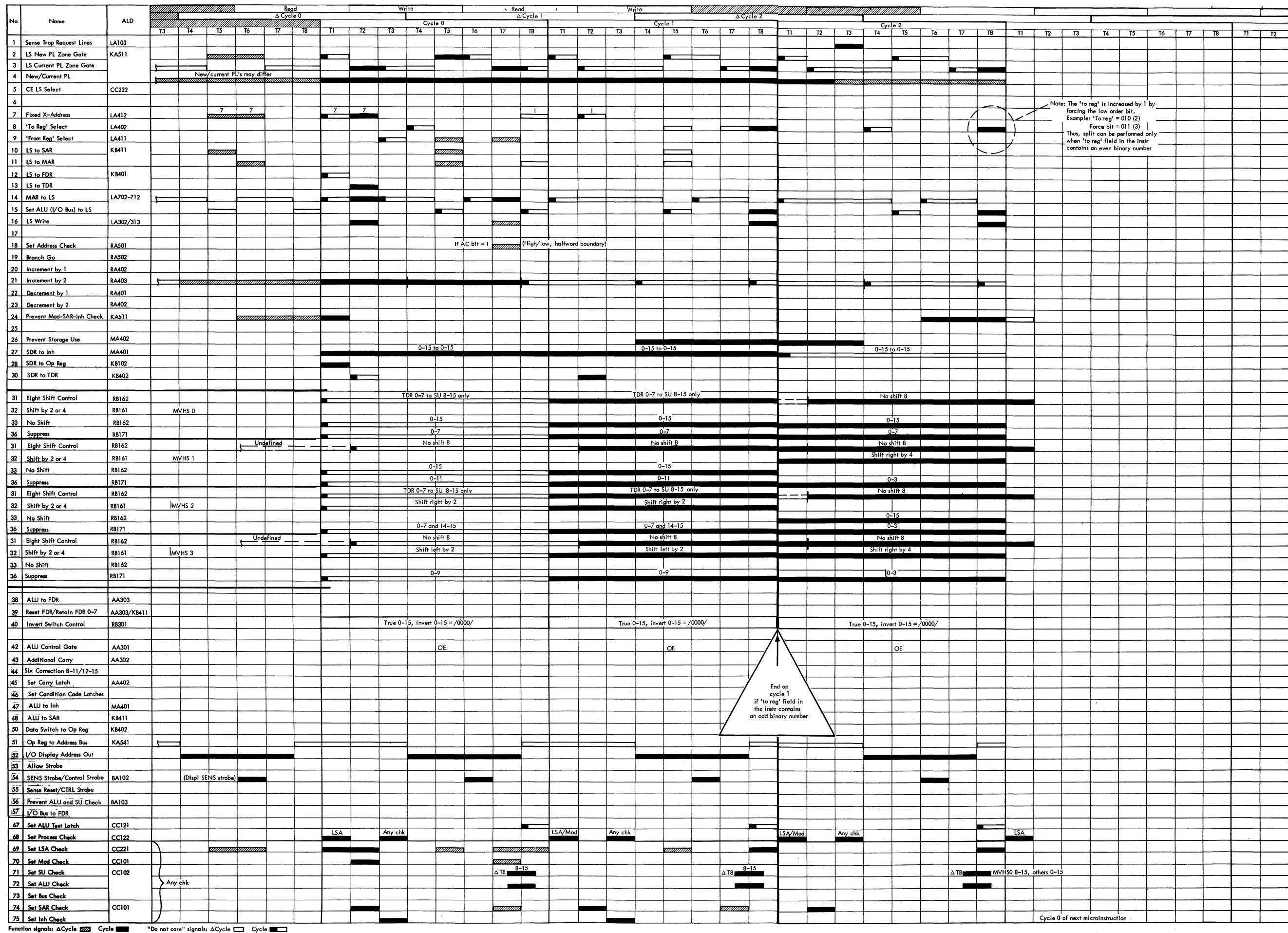
Mnemonic
MVHS
FF
DX



Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 Δ Cycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

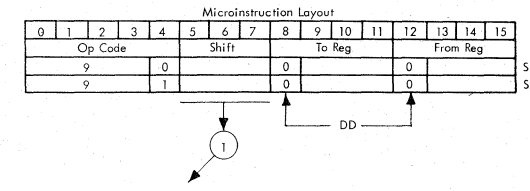
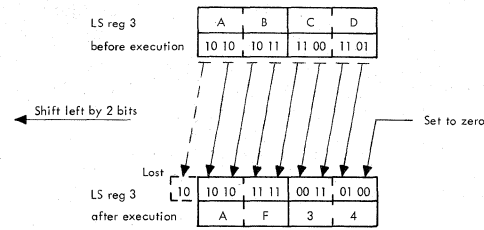


● Diagram 5-25. Move Halfword and Split (DX) (Part 2 of 2) (03730A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

The 'from reg' is shifted left or right as specified in the shift-amount field (bits 5, 6, 7 of the microinstruction). The shift result is set into 'to reg'. The 'from reg' remains unchanged.

'From reg' and 'to reg' may be the same LS register.

Example: Shift left by 2

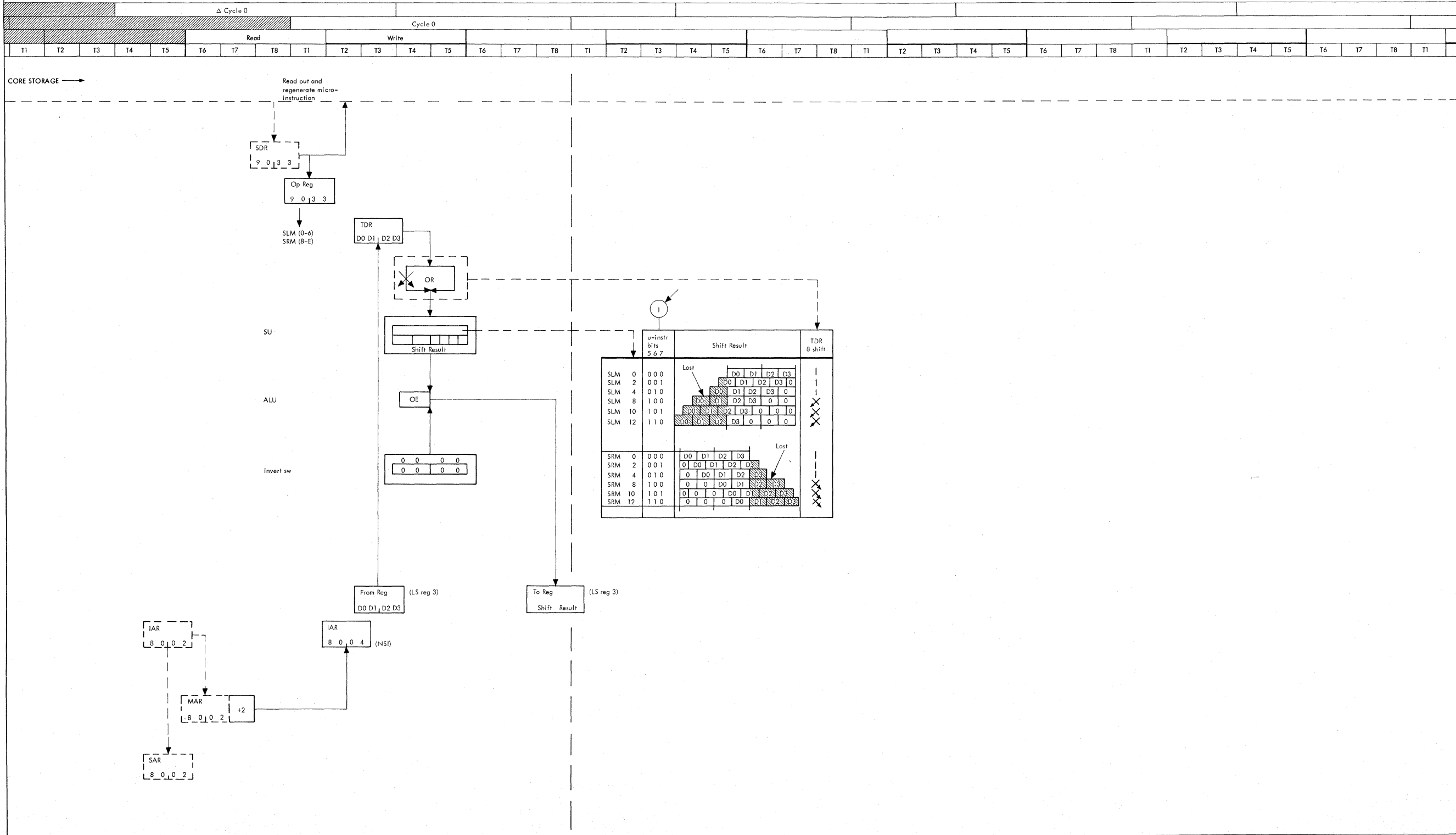


INST 9033 MNEM SLM OPERANDS 3,3,0 STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX R3=SL0 R3

Mnemonic
SLM
SRM
Format
FF
Type
DD

Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

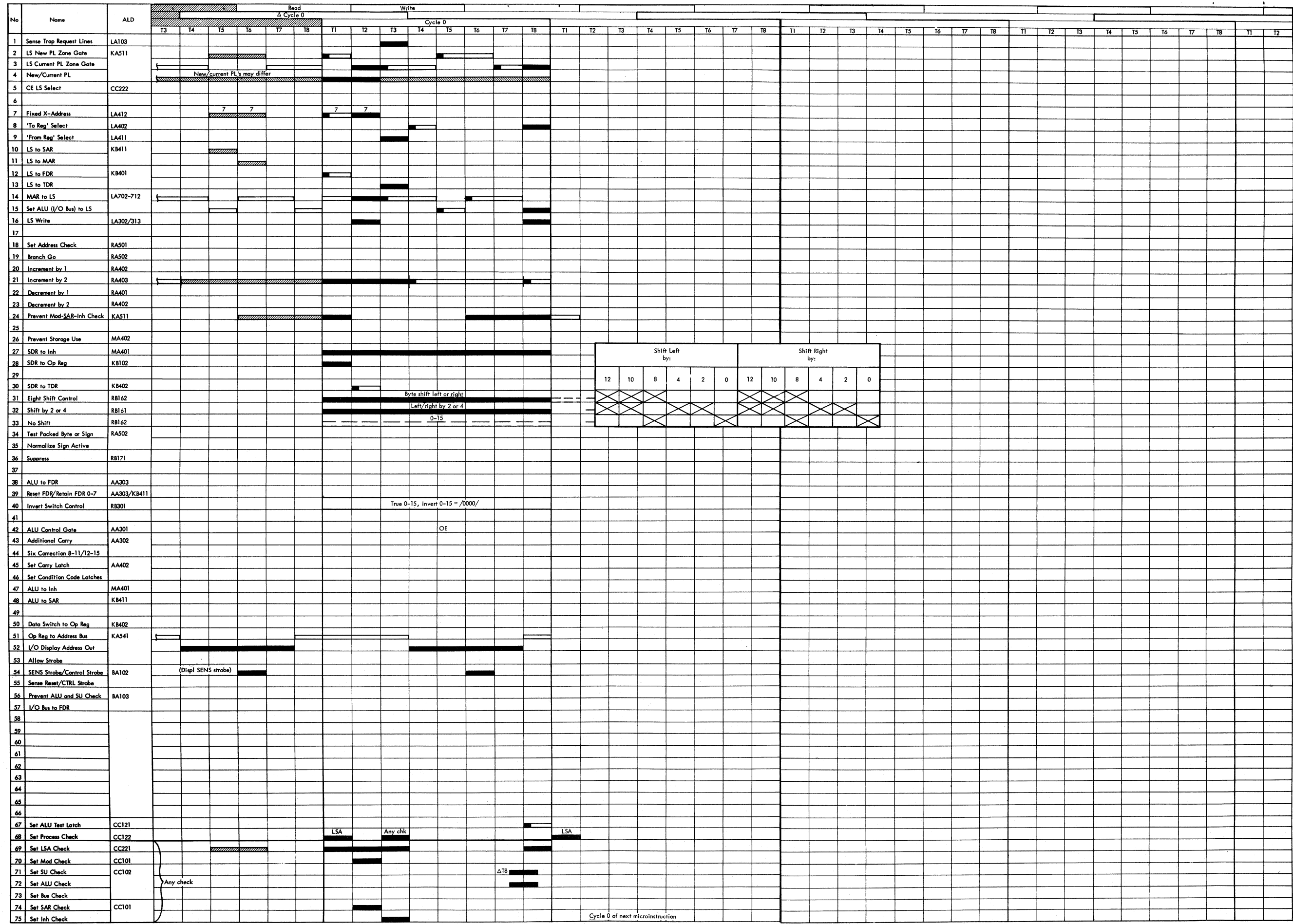
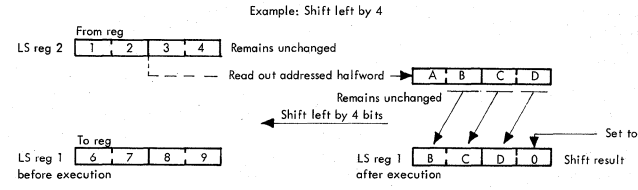


Diagram 5-26. Shift Left/Right and Move (DD) (Part 2 of 2) (03731A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

The halfword read out by the 'from reg' is shifted left or right as specified in the shift-amount field (bits 5, 6, 7 of the microinstruction).

The shift result is set into the 'to reg'.

The core storage halfword read out by the 'from reg' remains unchanged.



Microinstruction Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code				Shift				To Reg				From Reg			
9	0							0					1		
9	1							0					1		

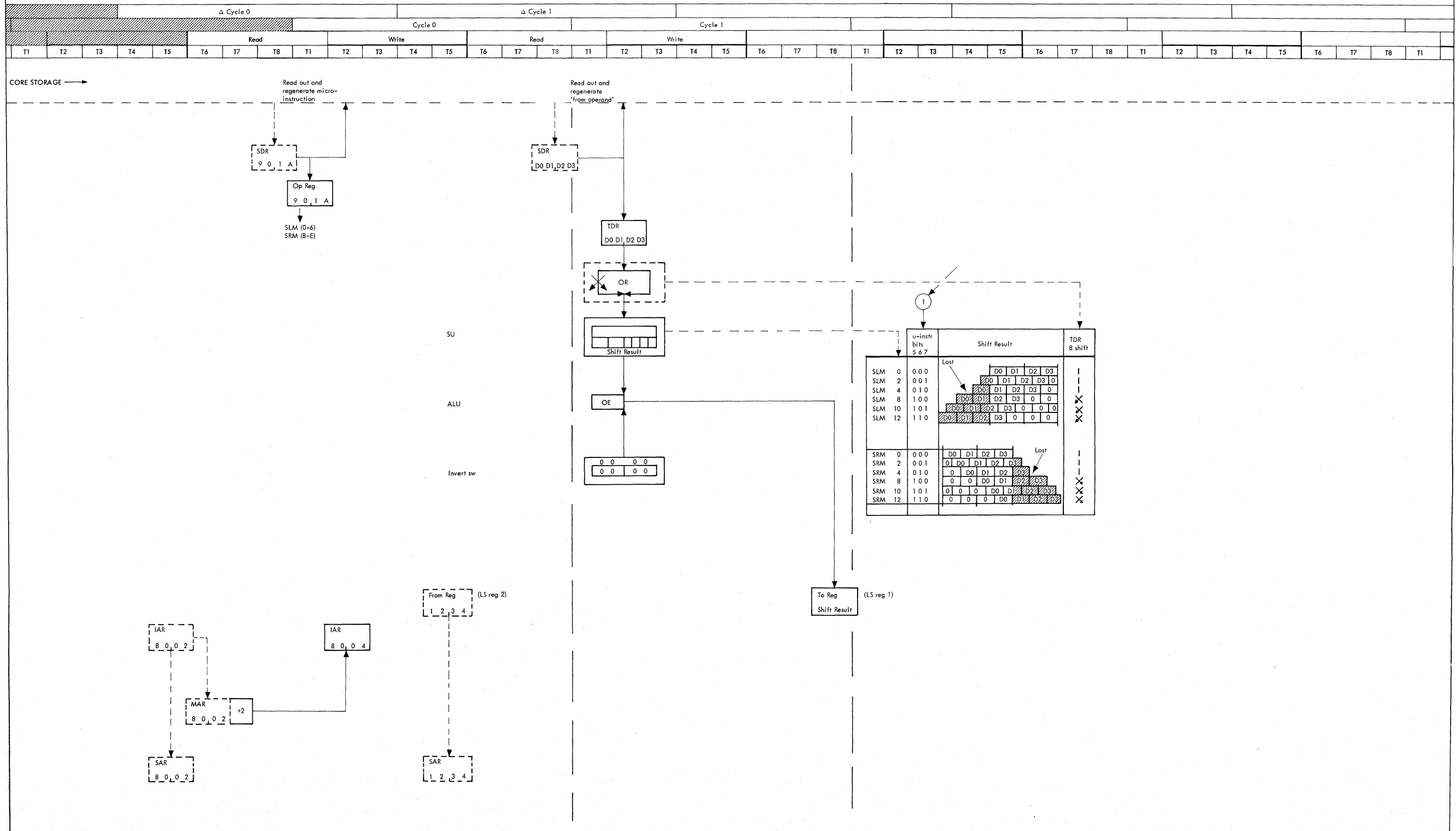
SLM SRM

INST 901A MNEM SLM OPERANDS 1,2I,0 STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX R1+=SL0 HW(R2)

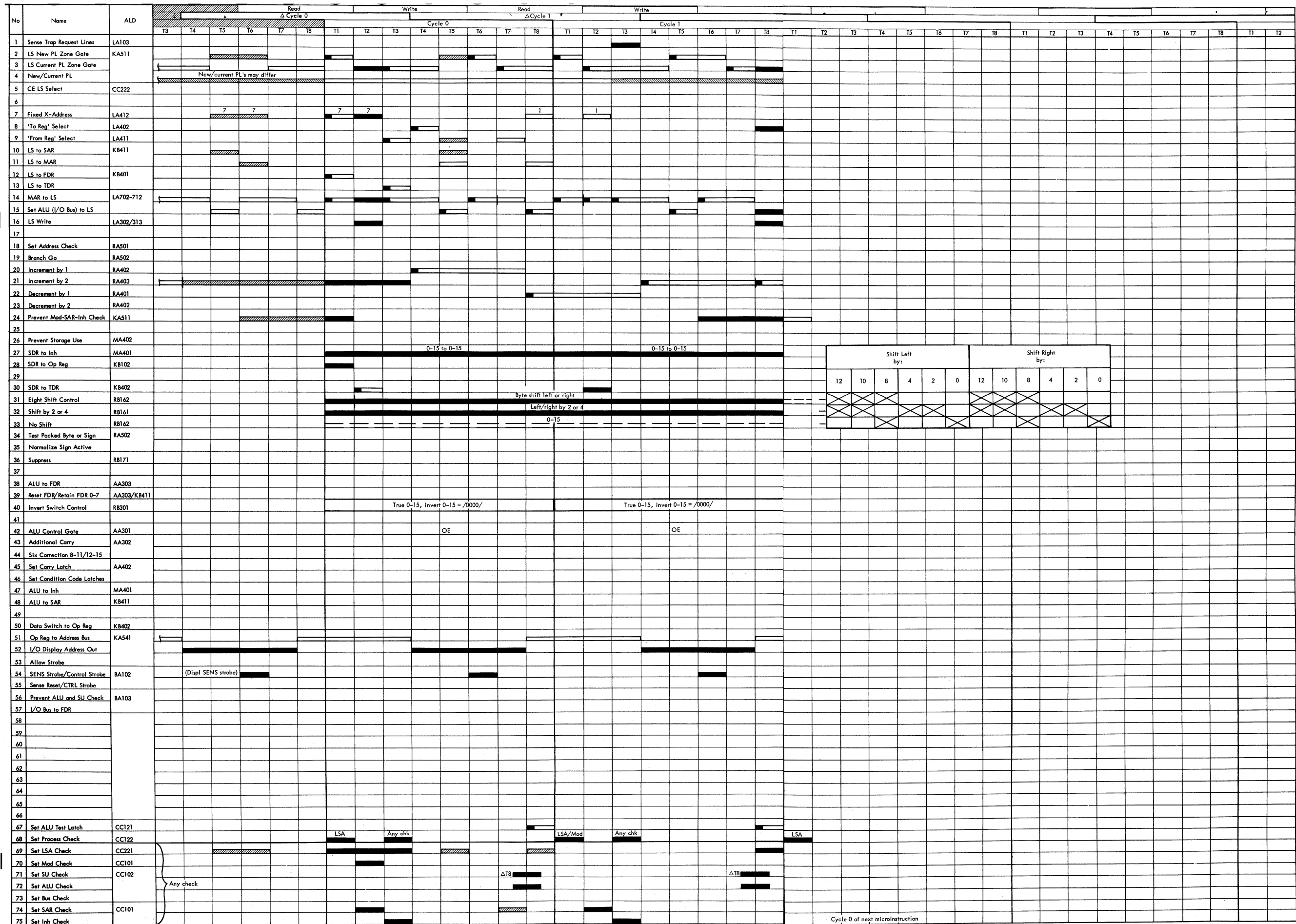
Mnemonic
SIM
SPM
Format
FF
Type
DX

Note: For "Do not care" functions refer to timing chart below.

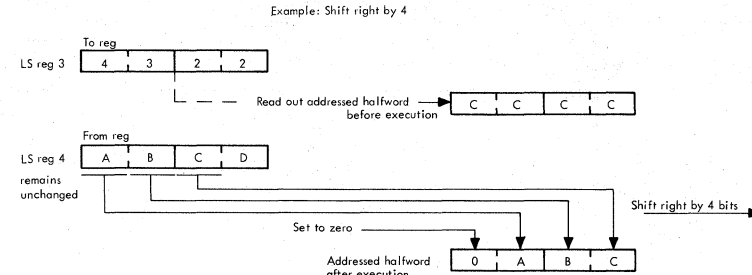
"Do not care" signals:
 ΔCycle
 Cycle



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



The 'from reg' is shifted left or right as specified in the shift amount field (bits 5, 6, 7 of the microinstruction). The shift result is stored into the core storage halfword addressed by the 'to reg'. The 'from reg' remains unchanged.

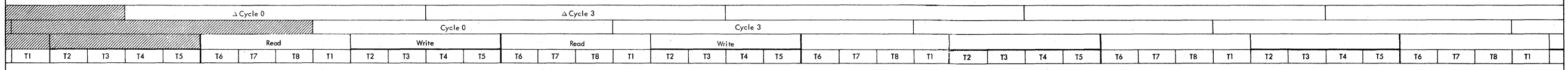


Microinstruction Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code					Shift			To Reg		From Reg					
9	0							1						0	
9	1							1						0	

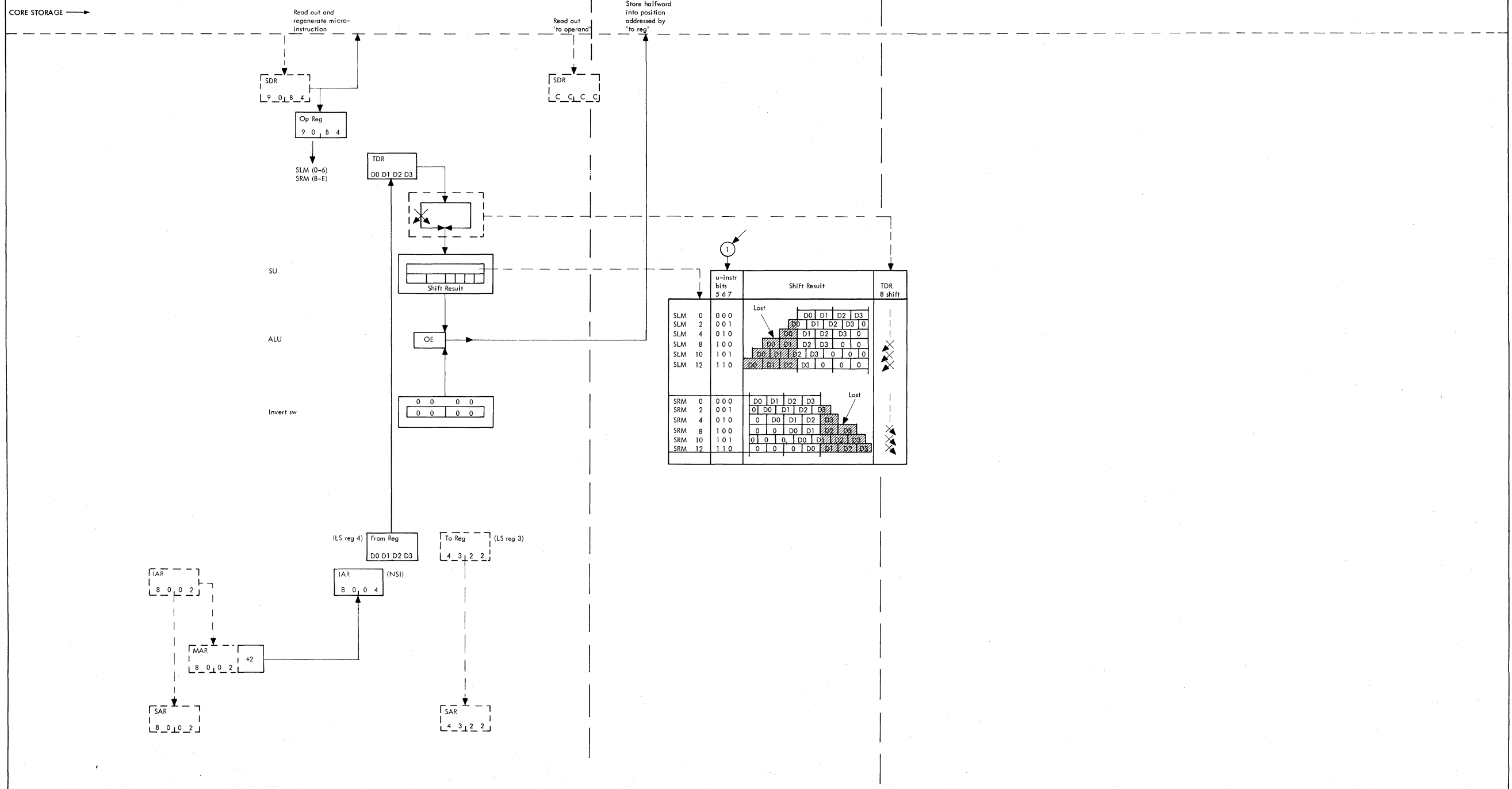
INST 90B4 MNE M OPERANDS STATEMENTS ACCORDING TO STANDARD CIB 0-1046-XXX
 SRM SLM 31,4,0 HW(R3)=SL0 R4

Mnemonic
SLM
SRM
Format
FF
Type
XD

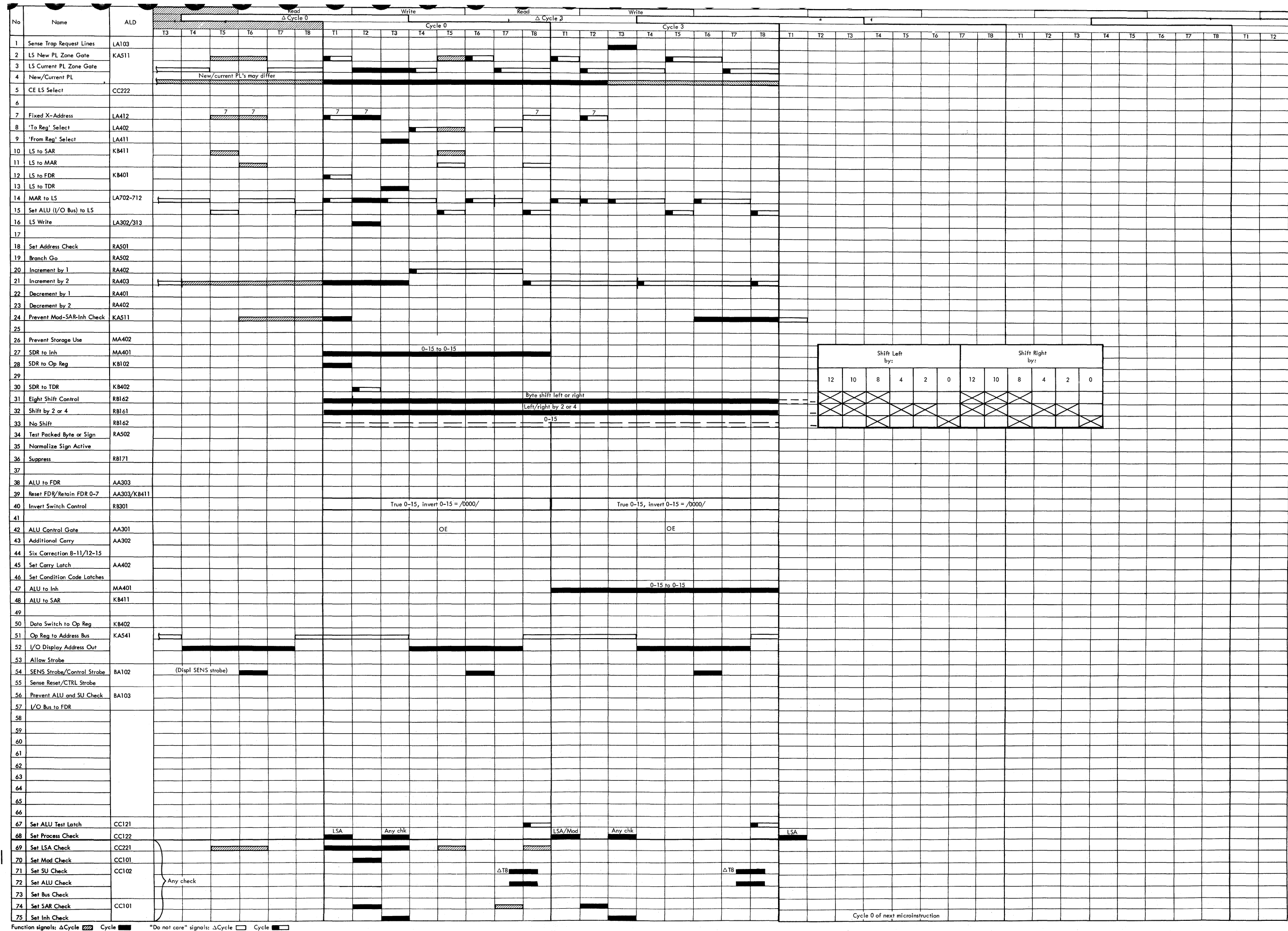


Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle



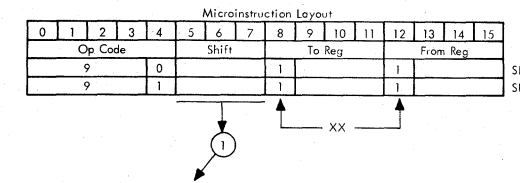
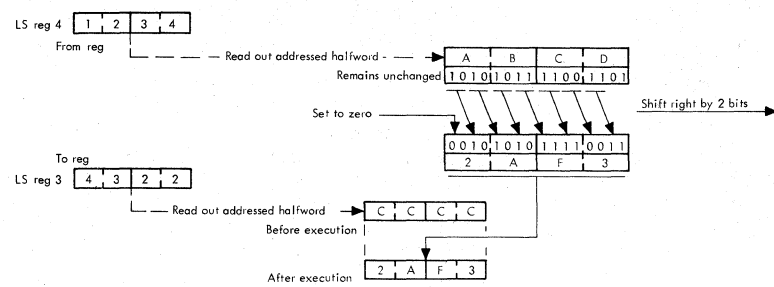
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



● Diagram 5-28. Shift Left/Right and Move (XD) (Part 2 of 2) (03733A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

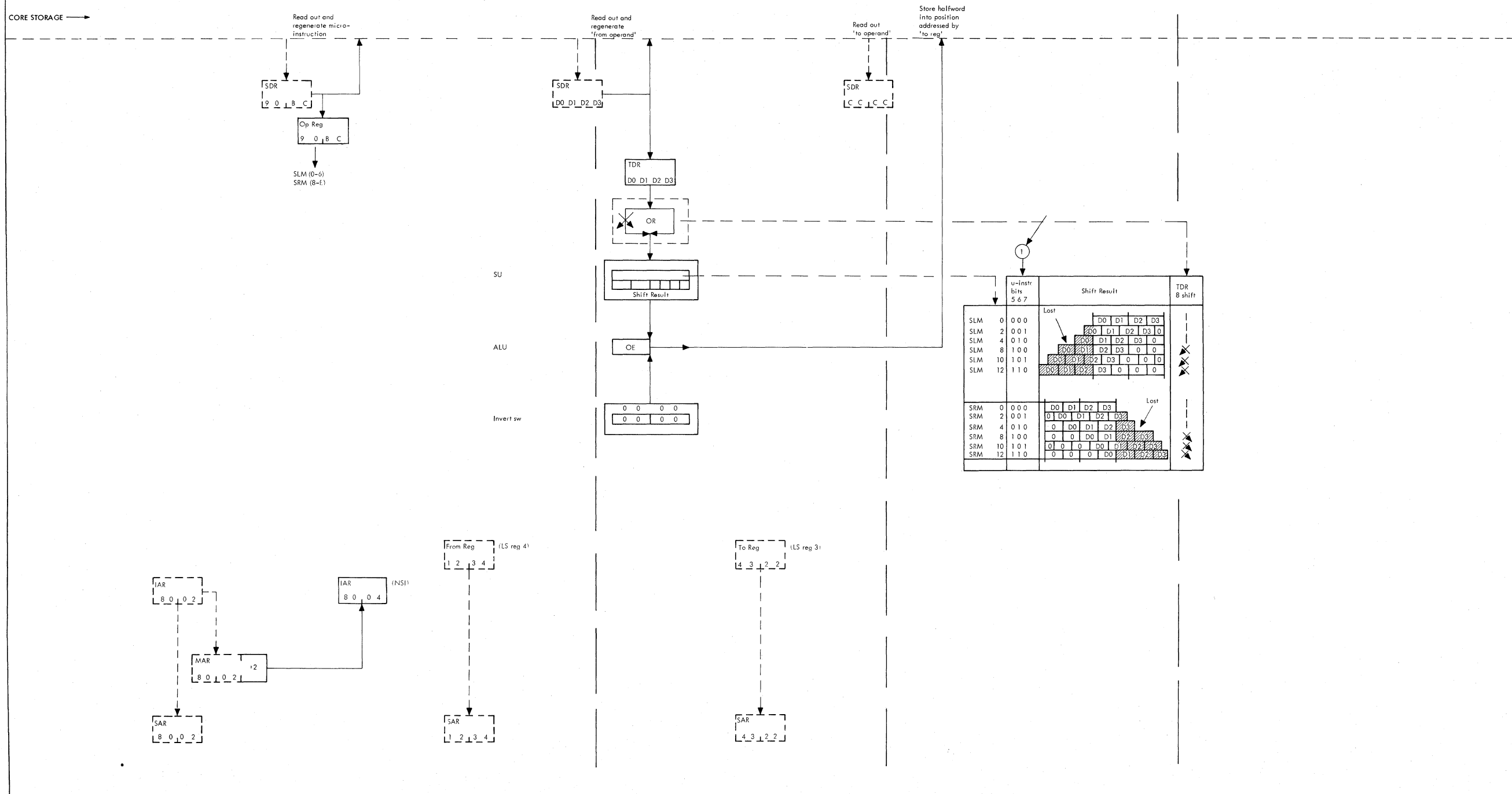
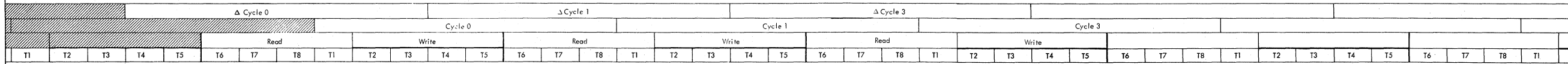
The halfword read out by the 'from reg' is shifted left or right as specified in the shift-amount fields (bits 5, 6, 7 of the microinstruction). The shift result is stored into the halfword addressed by 'to reg'.

The halfword addressed by 'from reg' remains unchanged. 'From reg' and 'to reg' may be the same LS register (same address) or both registers may contain the same address.



INST 90BC MNE: SLM OPERANDS 3I, 4I, 0 STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
HW(R3)--SL0 HW(R4)

Mnemonic	SLM
Format	FF
Type	XX

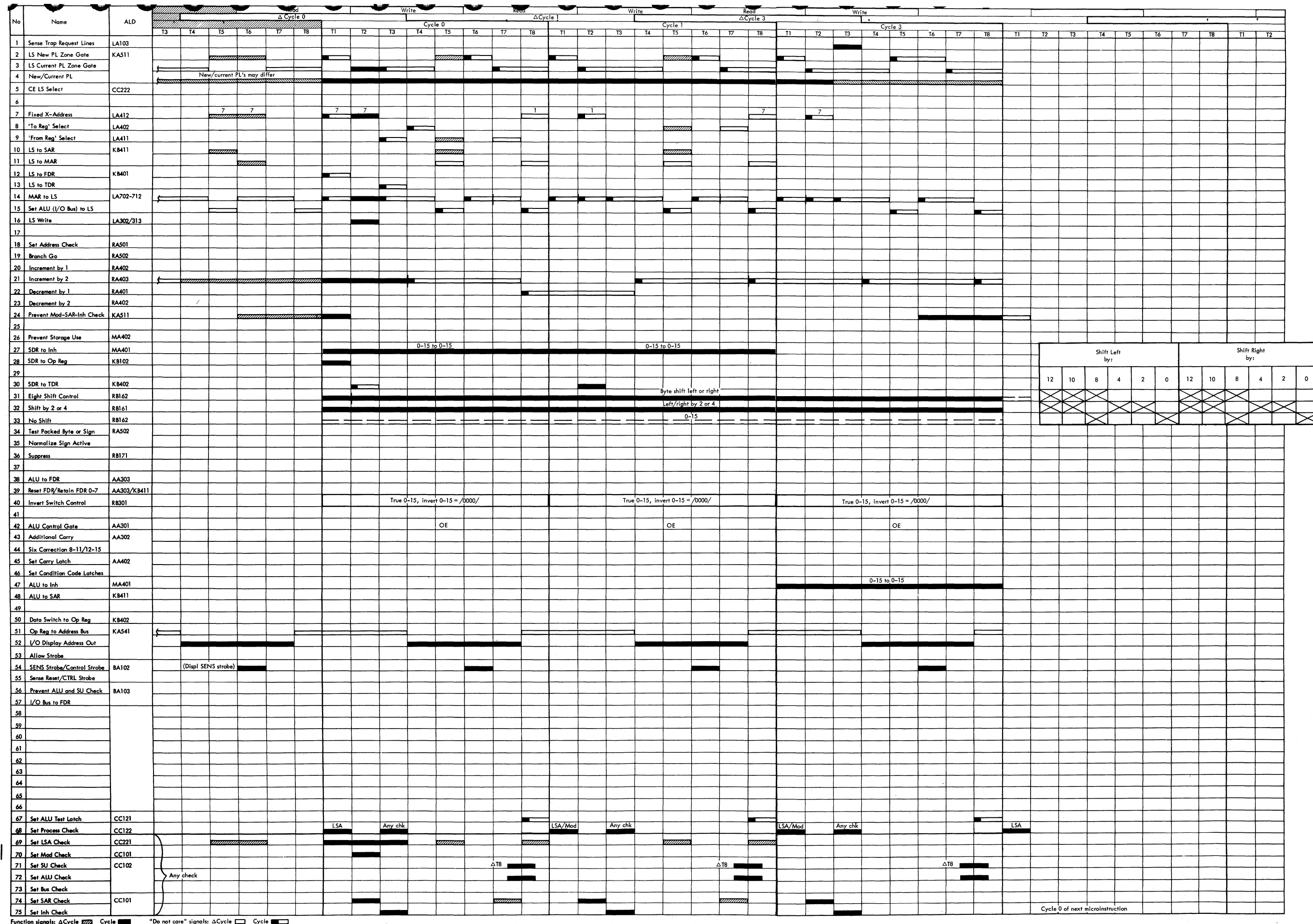


	u-instr bits 5 6 7	Shift Result	TDR 8 shift
SLM 0	0 0 0	Lost	---
SLM 2	0 0 1	D0 D1 D2 D3 0	---
SLM 4	0 1 0	D1 D2 D3 0 0	---
SLM 8	1 0 0	D2 D3 0 0 0	---
SLM 10	1 0 1	D3 0 0 0 0	---
SLM 12	1 1 0	D3 0 0 0 0	---
SRM 0	0 0 0	D0 D1 D2 D3	---
SRM 2	0 0 1	0 D0 D1 D2 D3	---
SRM 4	0 1 0	0 0 D0 D1 D2	---
SRM 8	1 0 0	0 0 0 D0 D1	---
SRM 10	1 0 1	0 0 0 0 D0	---
SRM 12	1 1 0	0 0 0 0 D0	---

Note: For "Do not care" functions refer to timing chart below.

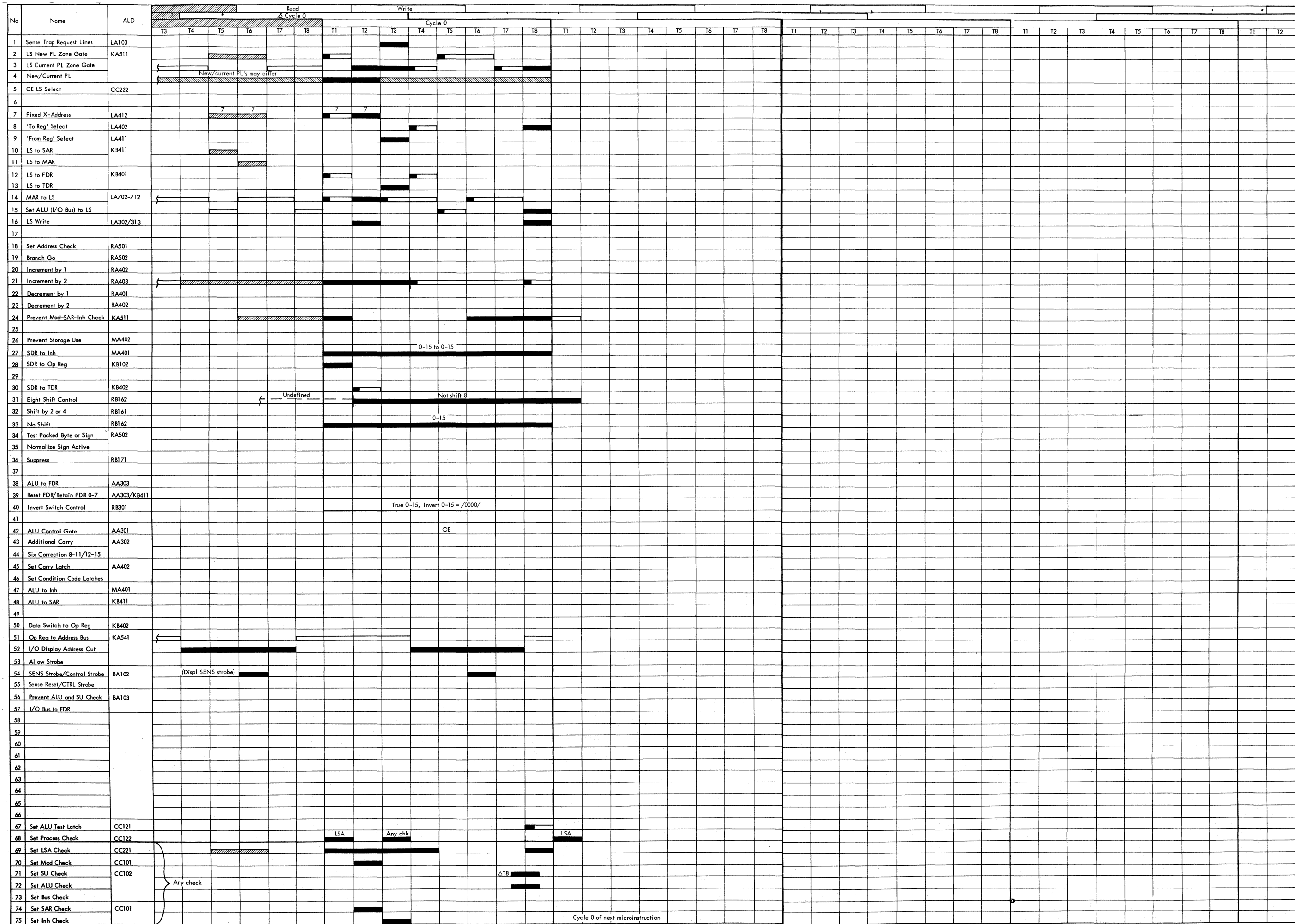
"Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



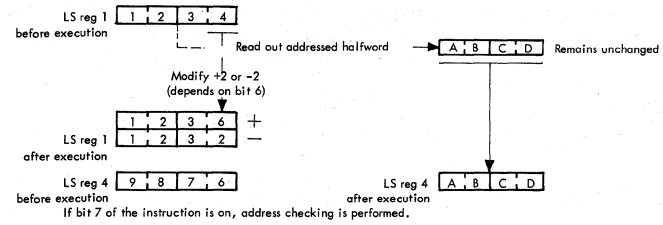
Shift Left by:						Shift Right by:					
12	10	8	4	2	0	12	10	8	4	2	0
X	X	X	X	X	X	X	X	X	X	X	X

Function signals: ΔCycle Cycle "Do not care" signals: ΔCycle Cycle



The halfword read out by the 'from reg' is moved into 'to reg'.

If the AC bit (instruction bit 7) is on, an address check occurs when the 'from reg' address is outside customer area or not on halfword boundary (not even). The 'from reg' address is incremented by 2 (instruction bit 6 on) or decremented by 2 (instruction bit 6 off).



Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code						±	AC	To Reg				From Reg			
A						0	1	0				1			

The 'from addr' is: incr by 2 if on
decr by 2 if off

If on, the 'from addr' is checked that it is not outside customer area

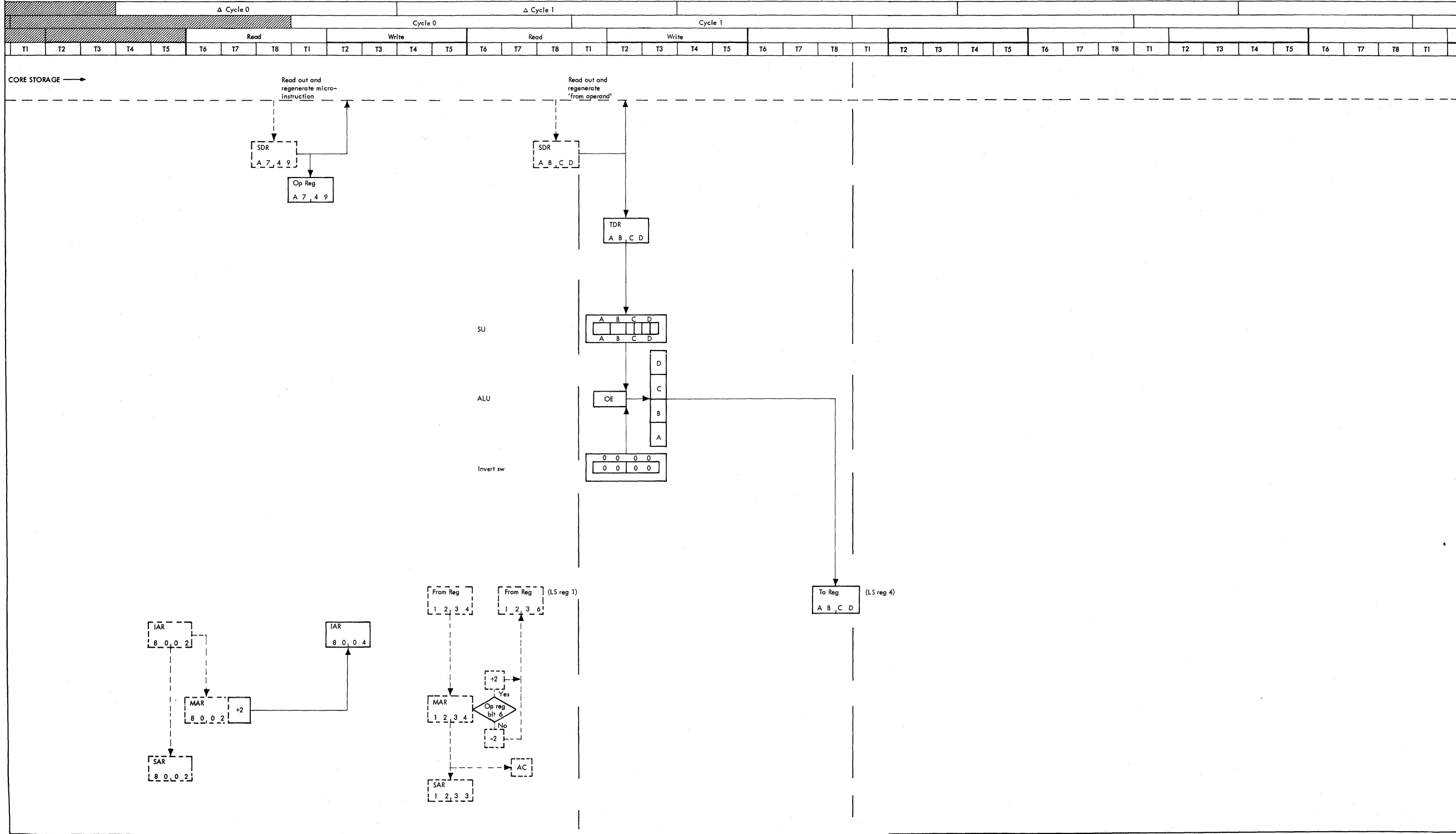
DX

INST A749 MNEM MVH OPERANDS 4, 11, INC, AC STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX R4+= HW(R1, AC, +2)

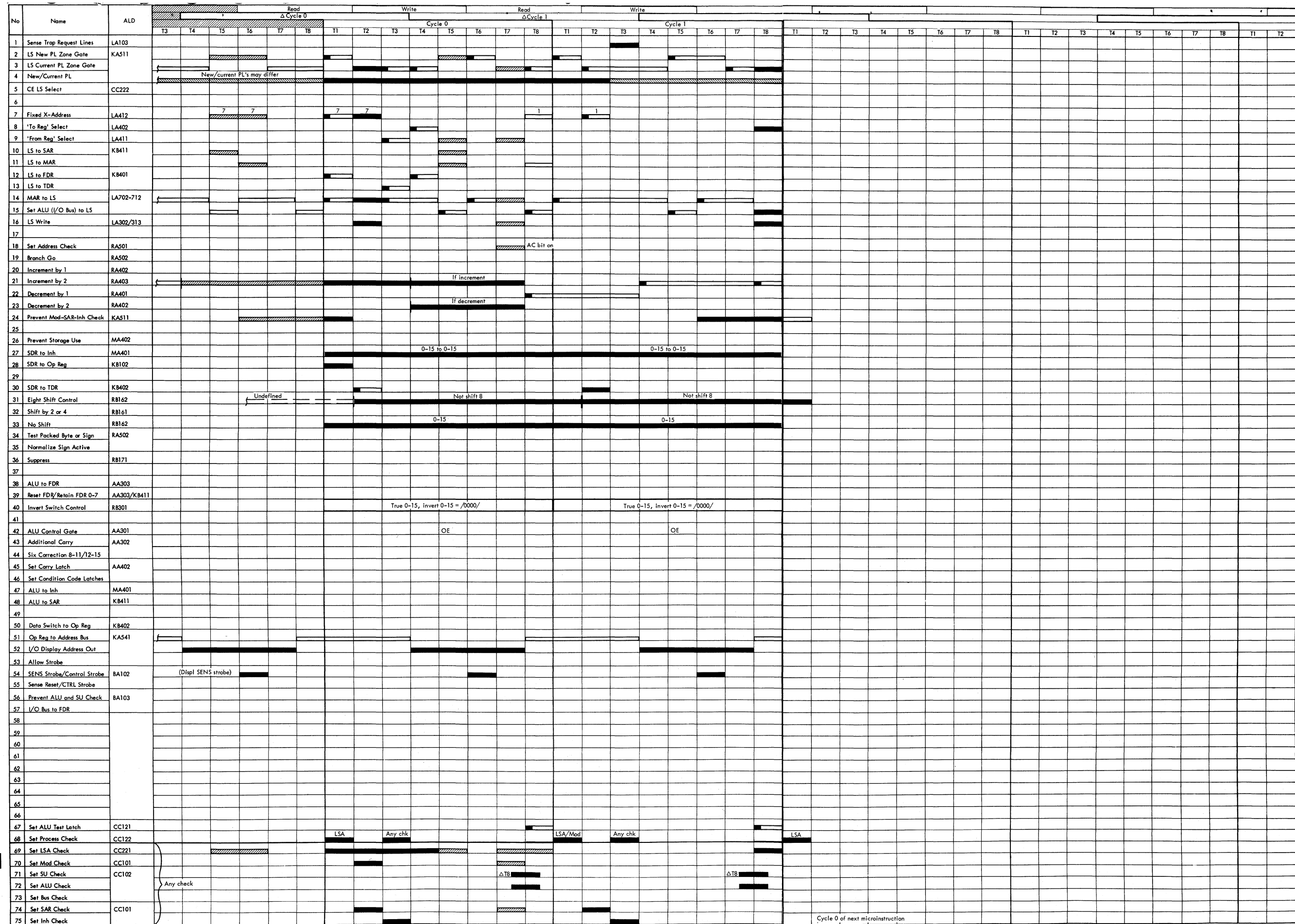
Mnemonic
MVH
Format
FF
Type
DX

Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle



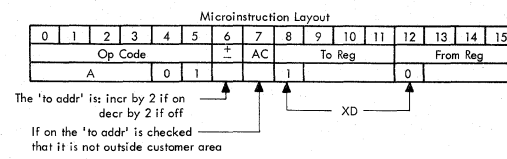
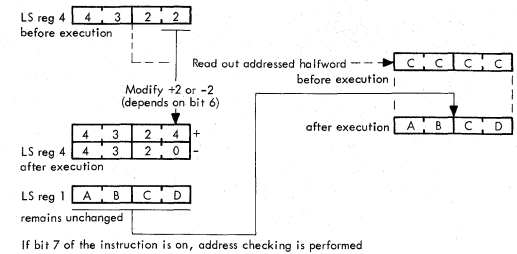
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



Function signals: ΔCycle Cycle "Do not care" signals: ΔCycle Cycle

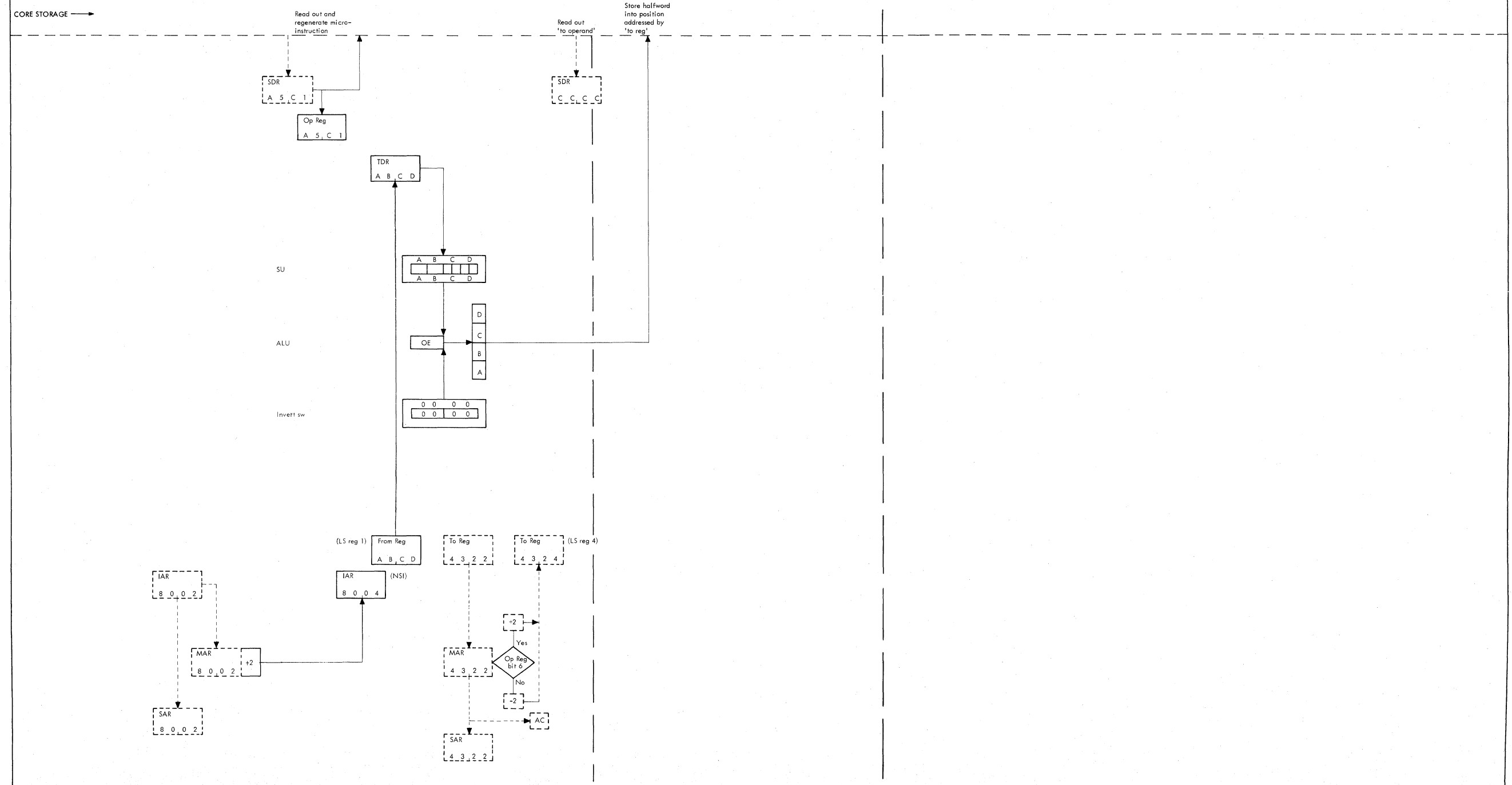
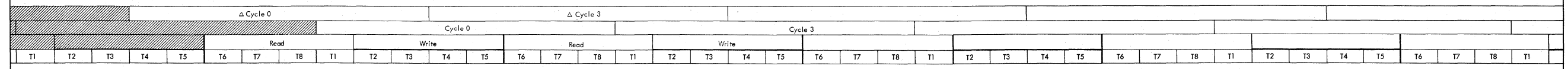
The 'from reg' is moved into the core storage halfword addressed by the 'to reg'. The 'from reg' remains unchanged. If the AC bit (instruction bit 7) is on, an address check occurs when the 'to reg' address is outside customer area or not on halfword boundary (not even).

The 'to reg' address is incremented by 2 (instruction bit 6 on) or decremented by 2 (instruction bit 6 off).



INST A5C1 MNE M V H OPERANDS 4I, 1, DEC, AC STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX HW(R4, AC, -2)=R1

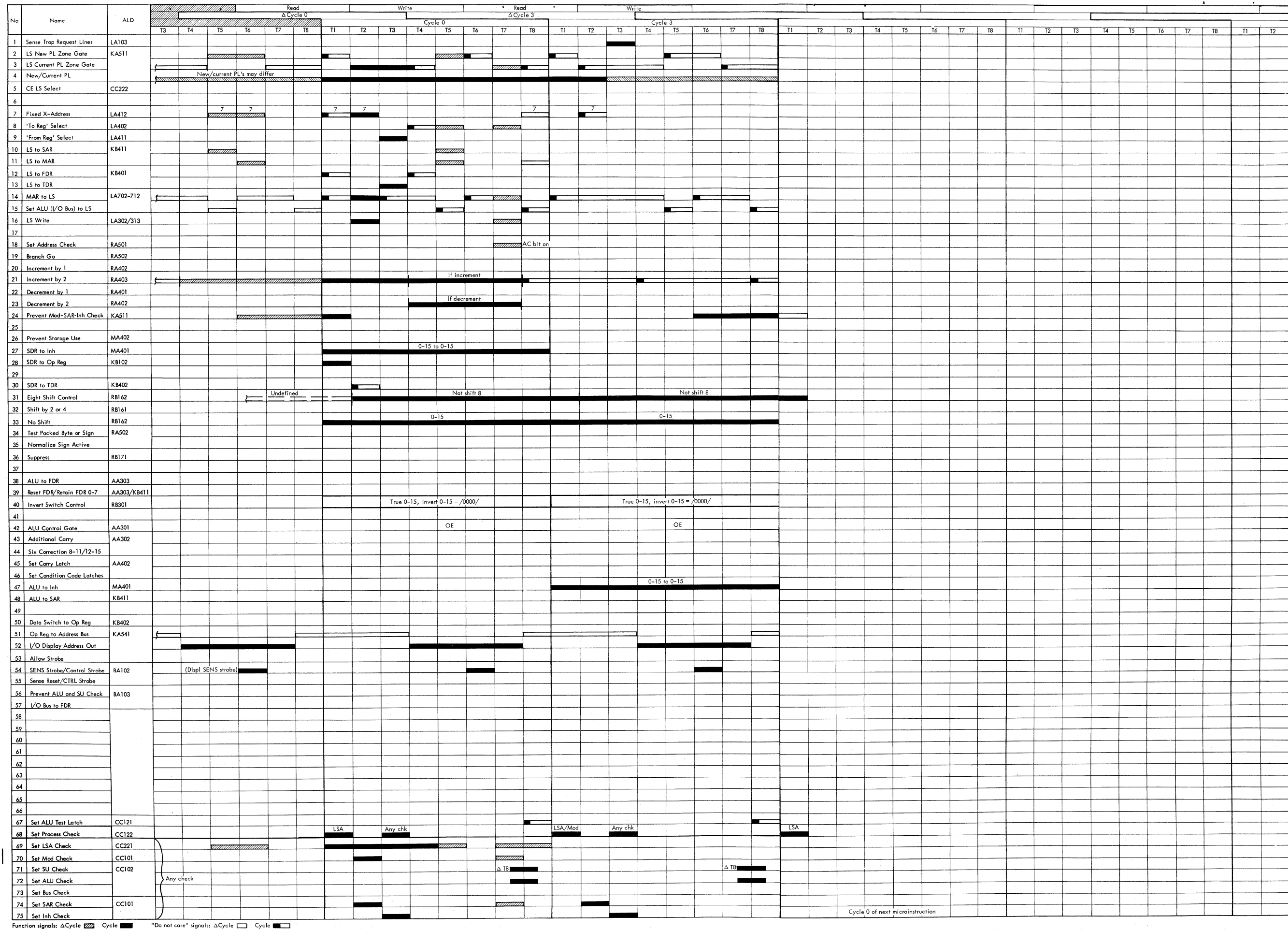
Mnemonic	MVH
Format	FF
Type	XD



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

Note: For "Do not care" functions refer to timing chart below.

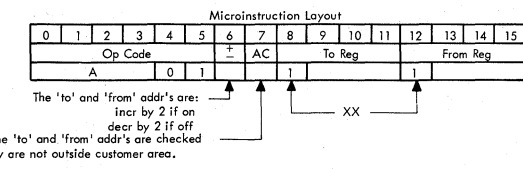
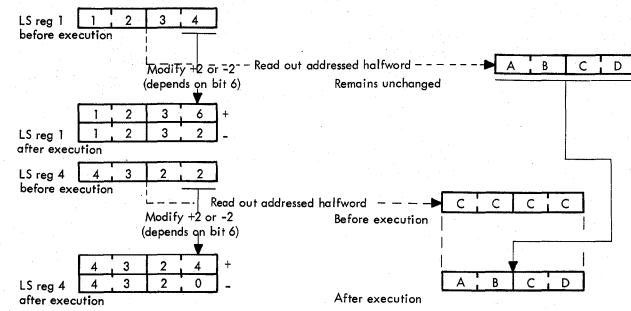
"Do not care" signals:
 ΔCycle
 Cycle



The halfword addressed by the 'from reg' is moved into the halfword addressed by 'to reg'. The 'from' halfword remains unchanged.

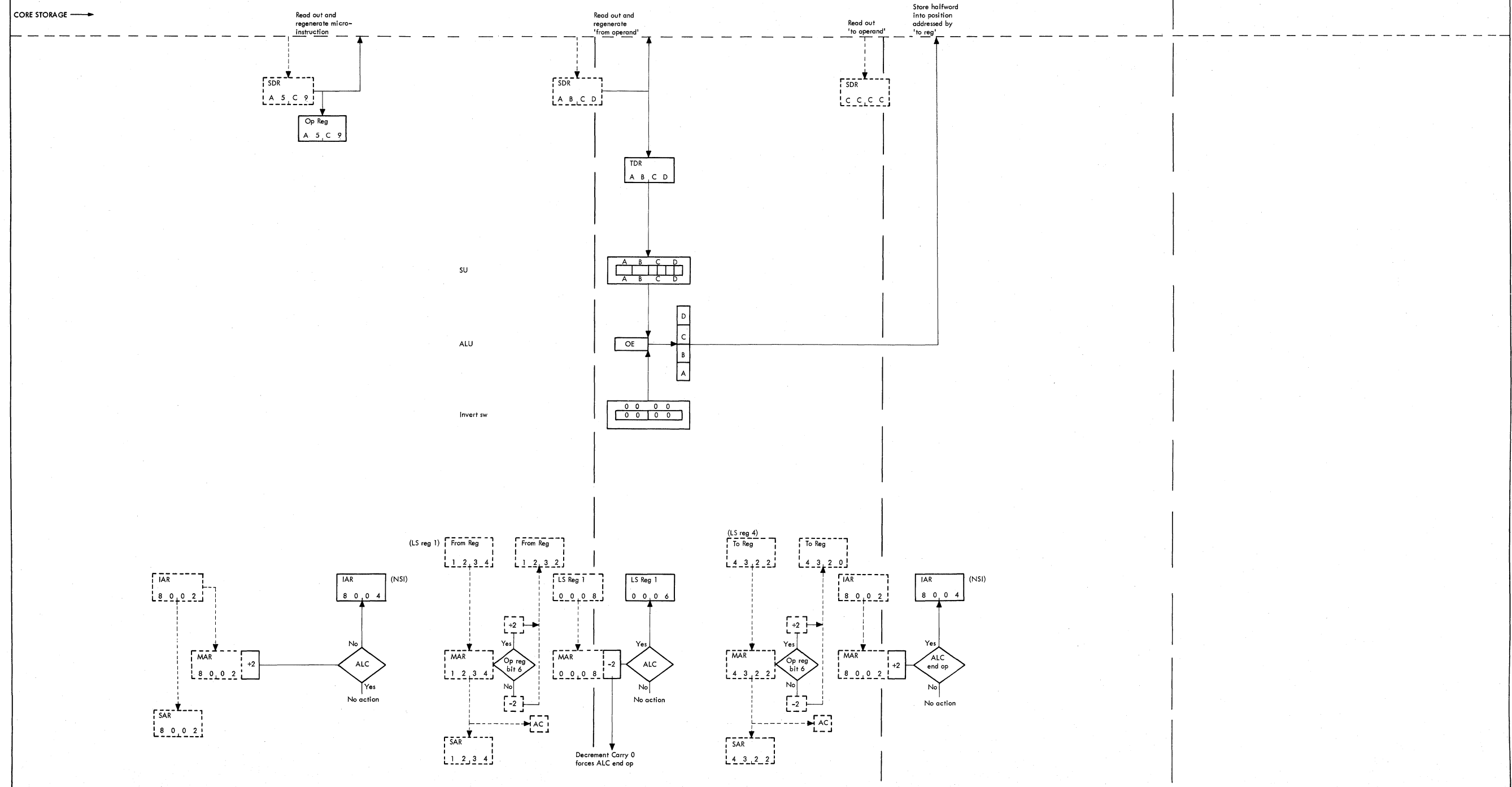
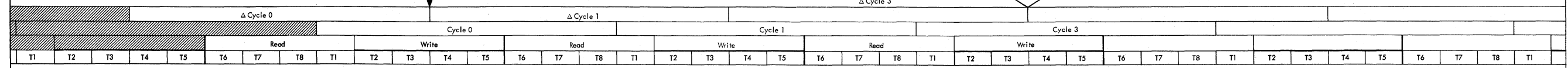
Both addresses are either incremented by 2 (instruction bit 6 on) or decremented by 2 (instruction bit 6 off). If the AC bit (instruction bit 7) is on, an address check occurs when the 'from reg' or 'to reg' address is outside customer area or not on halfword boundary.

The operation is performed in ALC mode when 'to reg' = 3 and 'from reg' = 5. LS register 1 contains the field length. The field length must reflect the number of bytes to be moved (always an even number) reduced by 2. For ALC, the operand addresses are updated according to instruction bit 6 (+2 or -2) every time a halfword is moved.



Mnemonic	MVH
Format	FF
Type	XX (ALC)

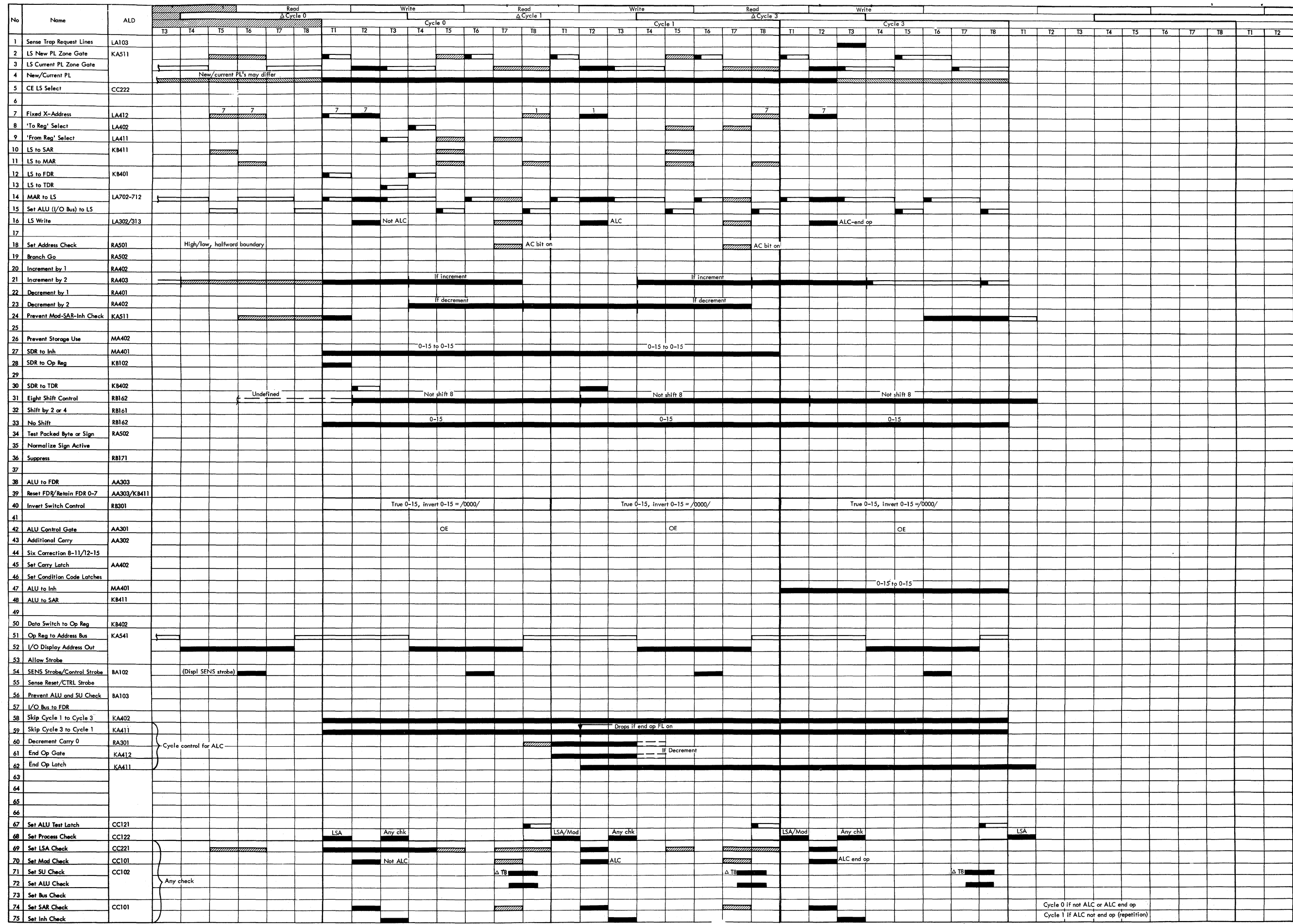
INST: A5C9, A7BD
 MNEM: MVH
 OPERANDS: 4I, 1I, DEC, AC; 3I, 5I, INC, AC
 STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX: HW(R4, AC, -2) += HW(R1, AC, -2); HW(R3, AC, +2, UNTIL R1, LT, 0) += HW(R5, AC, +2)



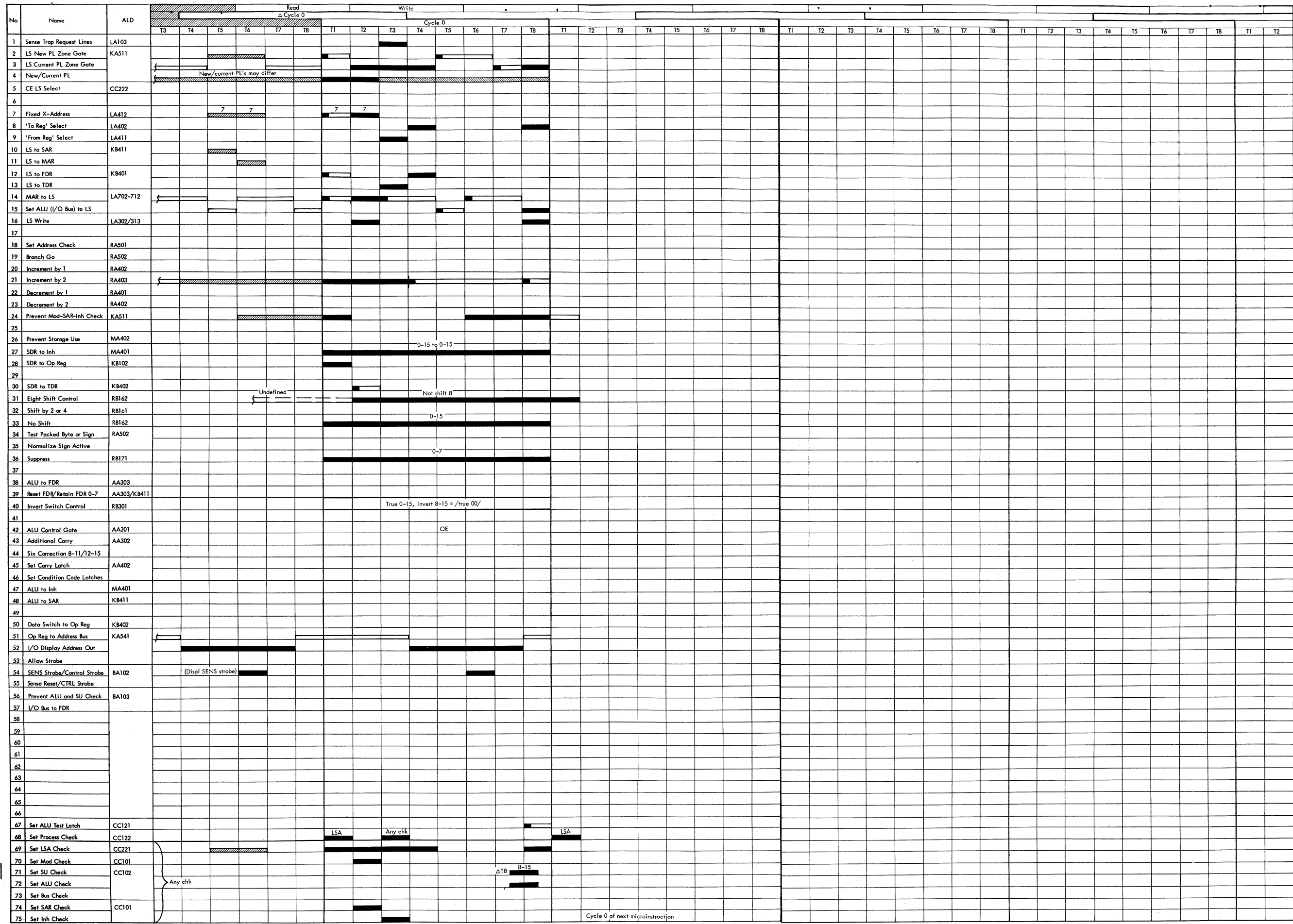
Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 Δ Cycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



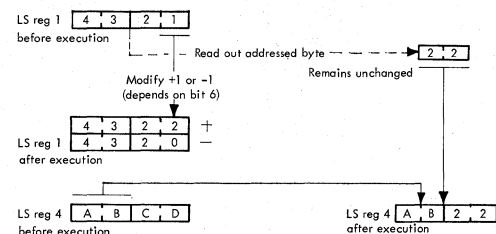
● Diagram 5-33. Move Halfword (XX, ALC) (Part 2 of 2) (03738A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)



Function signals: ΔCycle Cycle "Do not care" signals: ΔCycle Cycle

The byte addressed by the 'from reg' is moved into the low-order byte of the 'to reg'. The 'from' byte and the high-order byte of 'to reg' remain unchanged. If the AC bit (instruction bit 7) is on, an address check occurs when the 'from reg' address is outside customer area.

The 'from reg' address is incremented by 1 (instruction bit 6 on) or decremented by 1 (instruction bit 6 off)

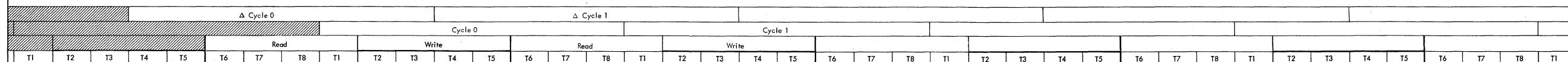


Microinstruction Layout																
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Op Code							±	AC	To Reg				From Reg			
A							1	0	0				1			

The 'from addr' is: incr by 1 if on decr by 1 if off
If on, the 'from'addr is checked that it is not outside customer area

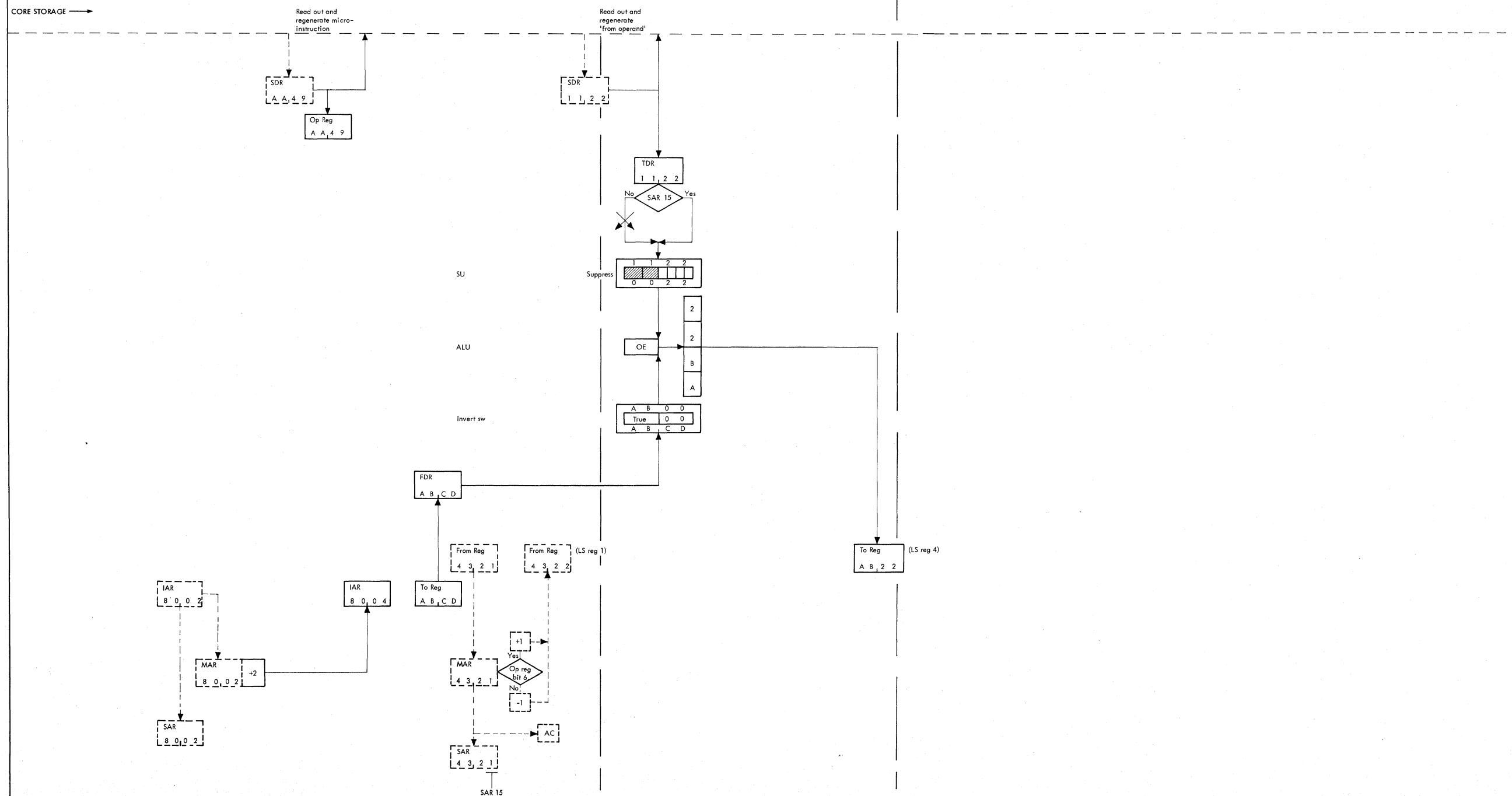
INST: AA49, MNEM: MVB, OPERANDS: 4, 11, INC, STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX: R4←R4.0-7/BY(R1,+1)

Mnemonic	MVB
Format	
FF	
Type	
DX	

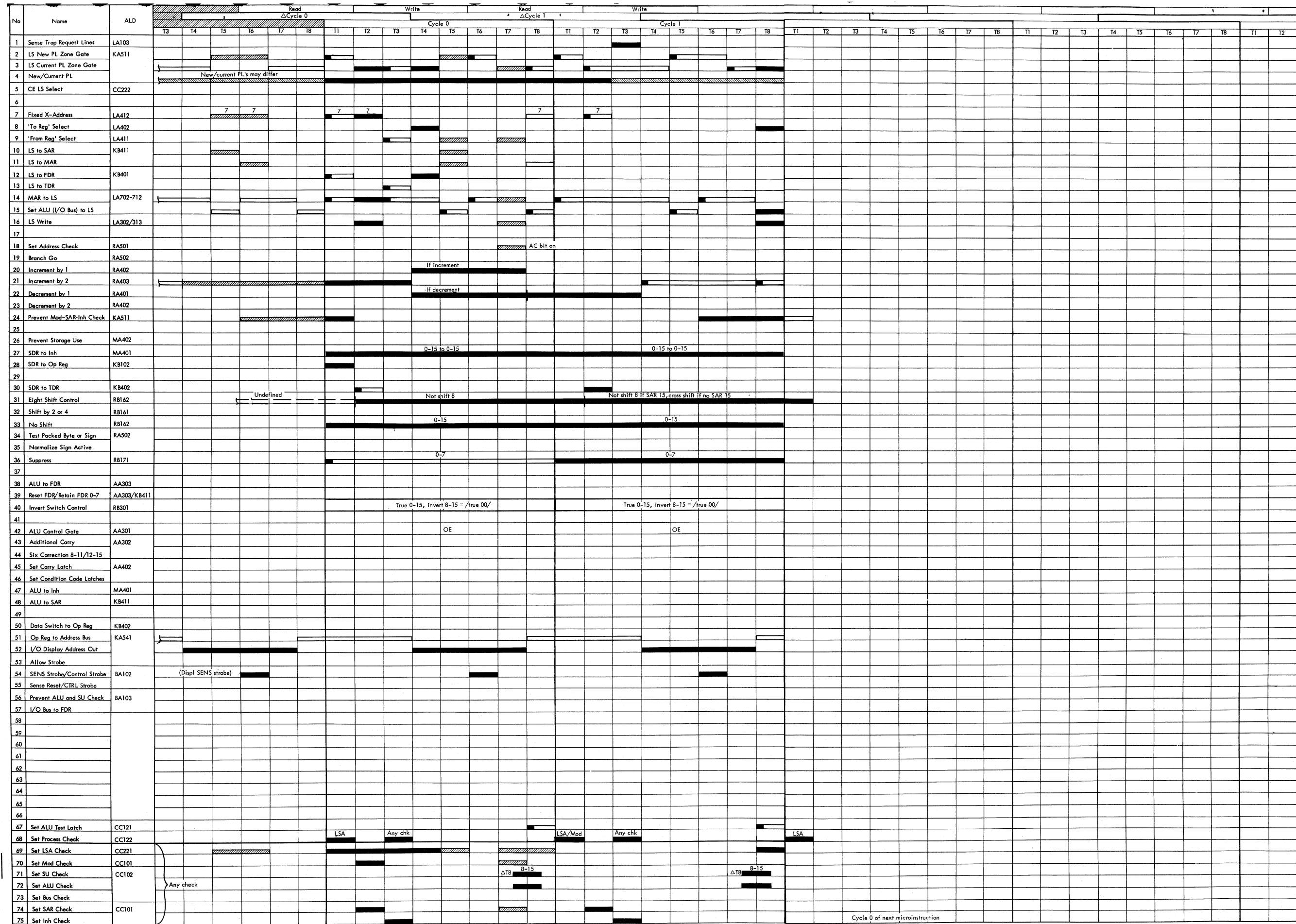


Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
ΔCycle
Cycle

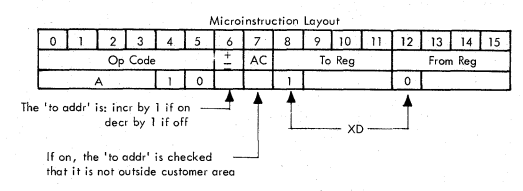
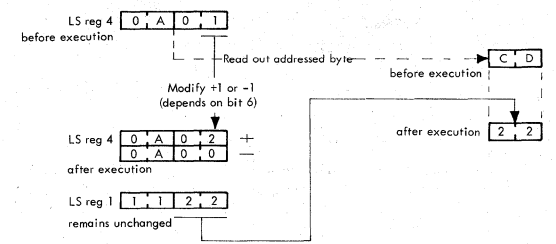


Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



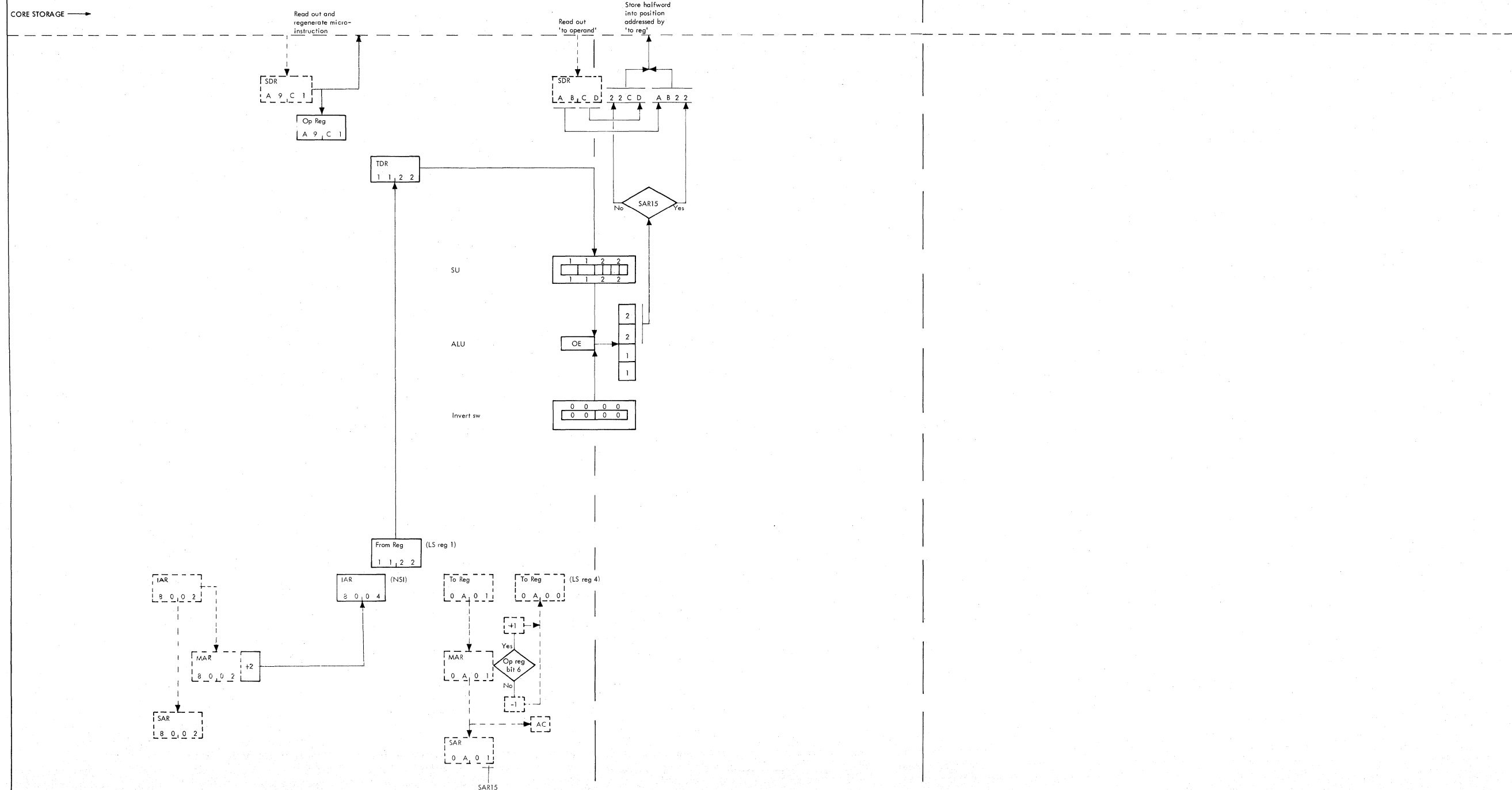
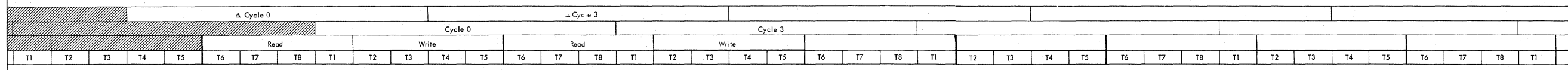
The low-order byte of the 'from reg' is moved into the byte addressed by the 'to reg'. The 'from reg' remains unchanged. If the AC bit (instruction bit 7) is on, an address check occurs when the 'to reg' address is outside customer area.

The 'to reg' address is incremented by 1 (instruction bit 6 on) or decremented by 1 (instruction bit 6 off).



INST: A9C1
 NAME: MVB
 OPERANDS: 4I, 1, D, -, AC
 STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX: BY(R4, AC, -1)+=R1, 8-15

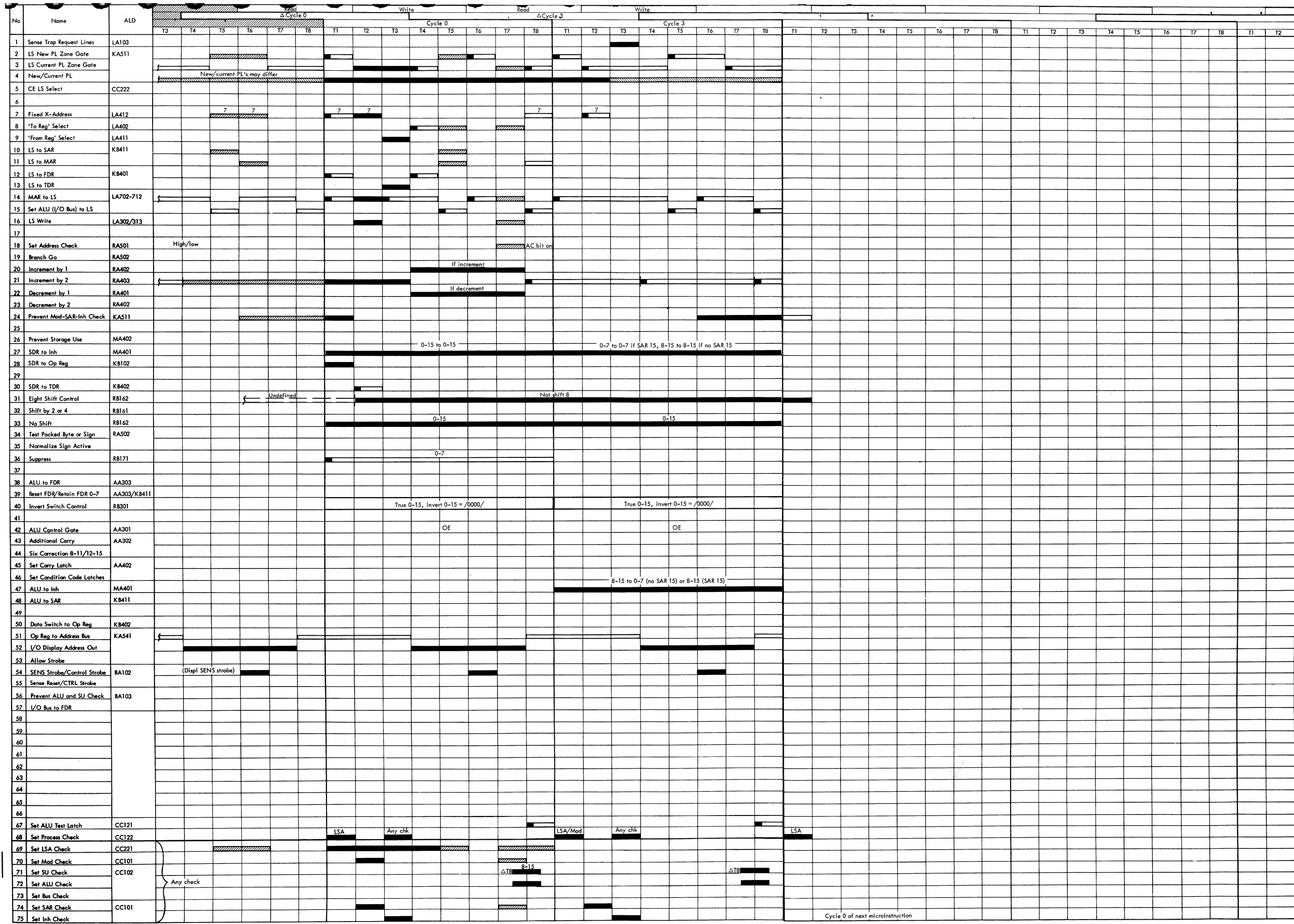
Mnemonic	MVB
Format	FF
Type	XD



Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

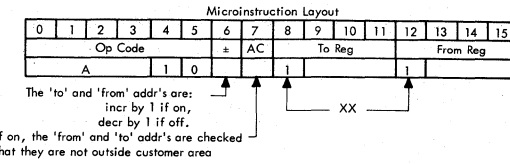
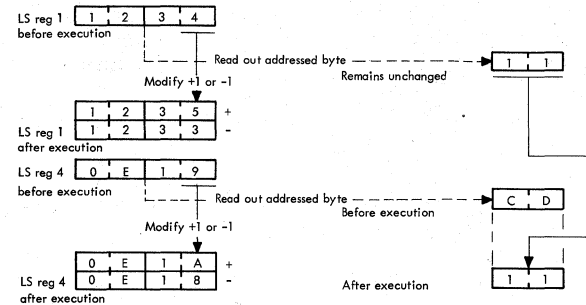
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



The byte addressed by the 'from reg' is moved into the byte addressed by the 'to reg'.

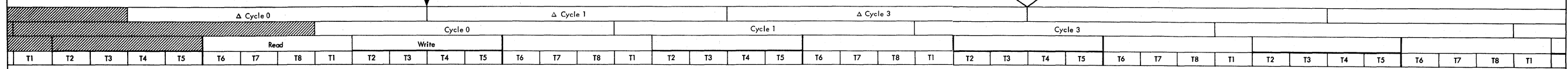
The 'from' byte remains unchanged. Both addresses are either incremented by 1 (instruction bit 6 on) or decremented by 1 (instruction bit 6 off). If the AC bit (instruction bit 7) is on, an address check occurs when the 'from reg' or 'to reg' address is outside customer area.

The operation is performed in ALC mode when 'to reg' = 3 and 'from reg' = 5. LS register 1 contains the field length. The field length is the real number of bytes to be moved, reduced by 1. For ALC, the operand addresses are updated according to instruction bit 6 (+1 or -1) every time a byte is moved.



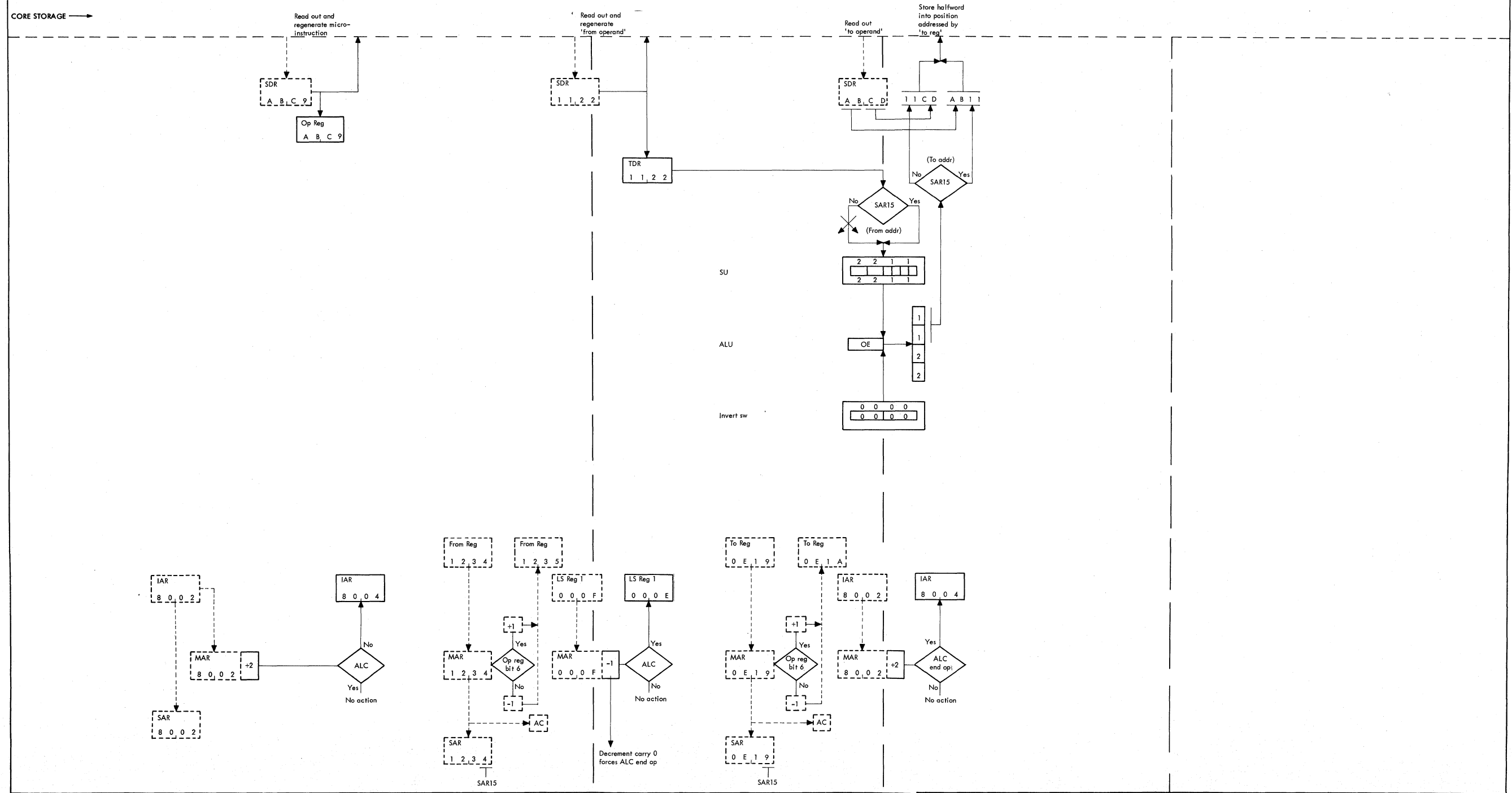
INST: ABC9, ABBD
 MNEM: MYB, MYB
 OPERANDS: 4I, 1I, INC, AC; 3I, 5I, INC, AC
 STATEMENTS ACCORDING TO STANDARD GEB 0-1046-XXX: BY(R4, AC, +1) = BY(R1, AC, +1); BY(R3, AC, +1, UNTIL R1, LT, 0) = BY(R5, AC, +1)

Mnemonic	MVB
Format	FF
Type	XX (ALC)

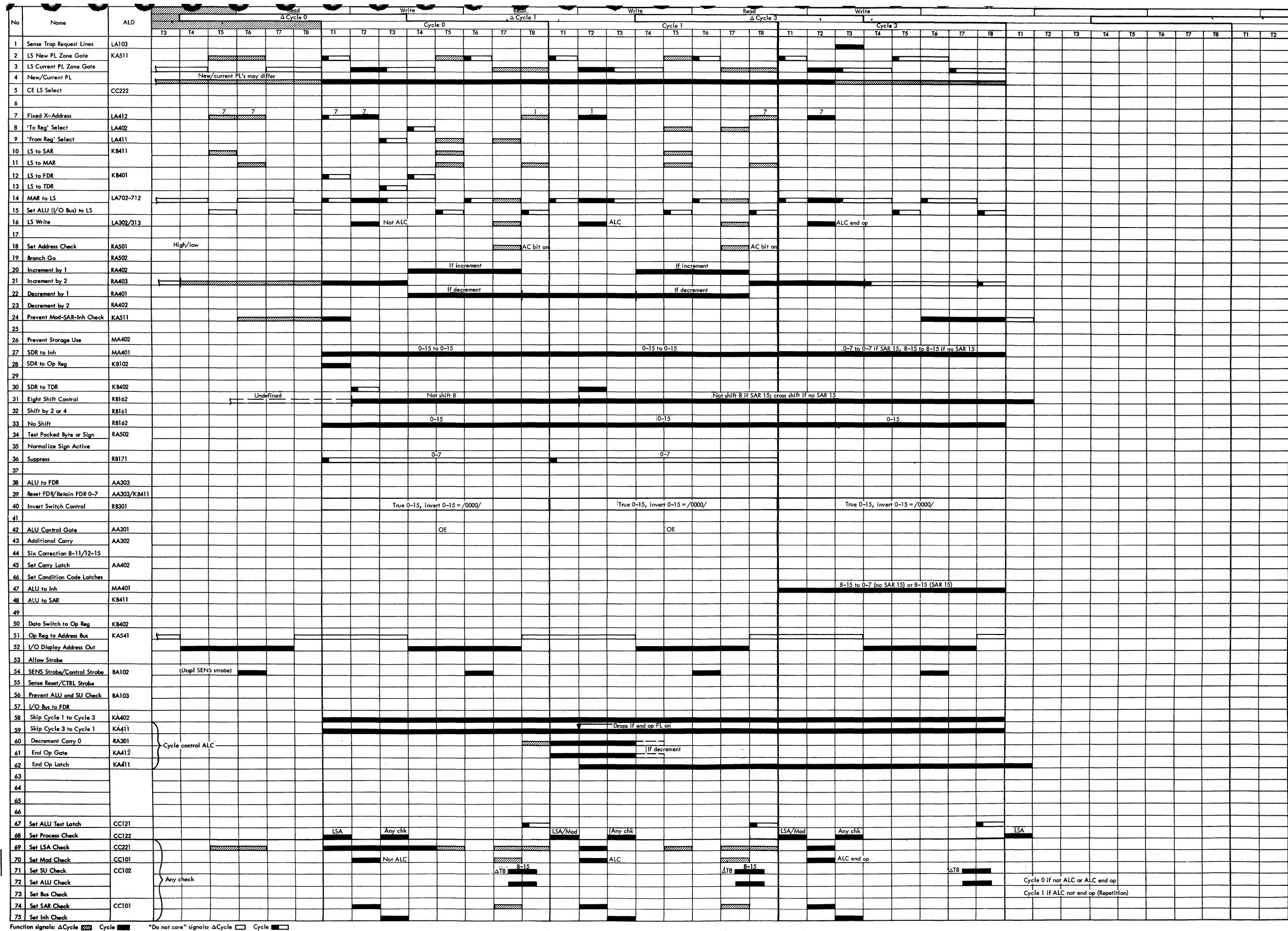


Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle



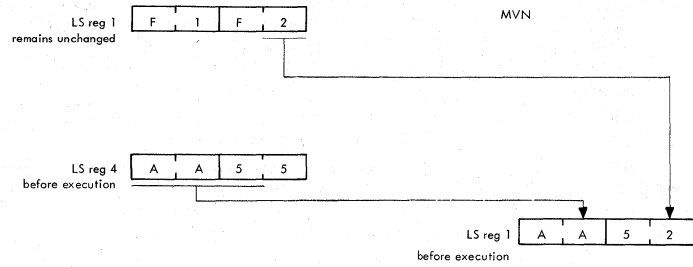
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



Function signals: ΔCycle Cycle "Do not care" signals: ΔCycle Cycle

The numeric (zone) of the low-order byte of the 'from reg' is moved into the numeric (zone) of the low-order byte of the 'to reg'.

The 'from reg', the highorder byte of the 'to reg', and the zone (numeric) of low-order byte of the 'to reg' remain unchanged. Address check is ignored.



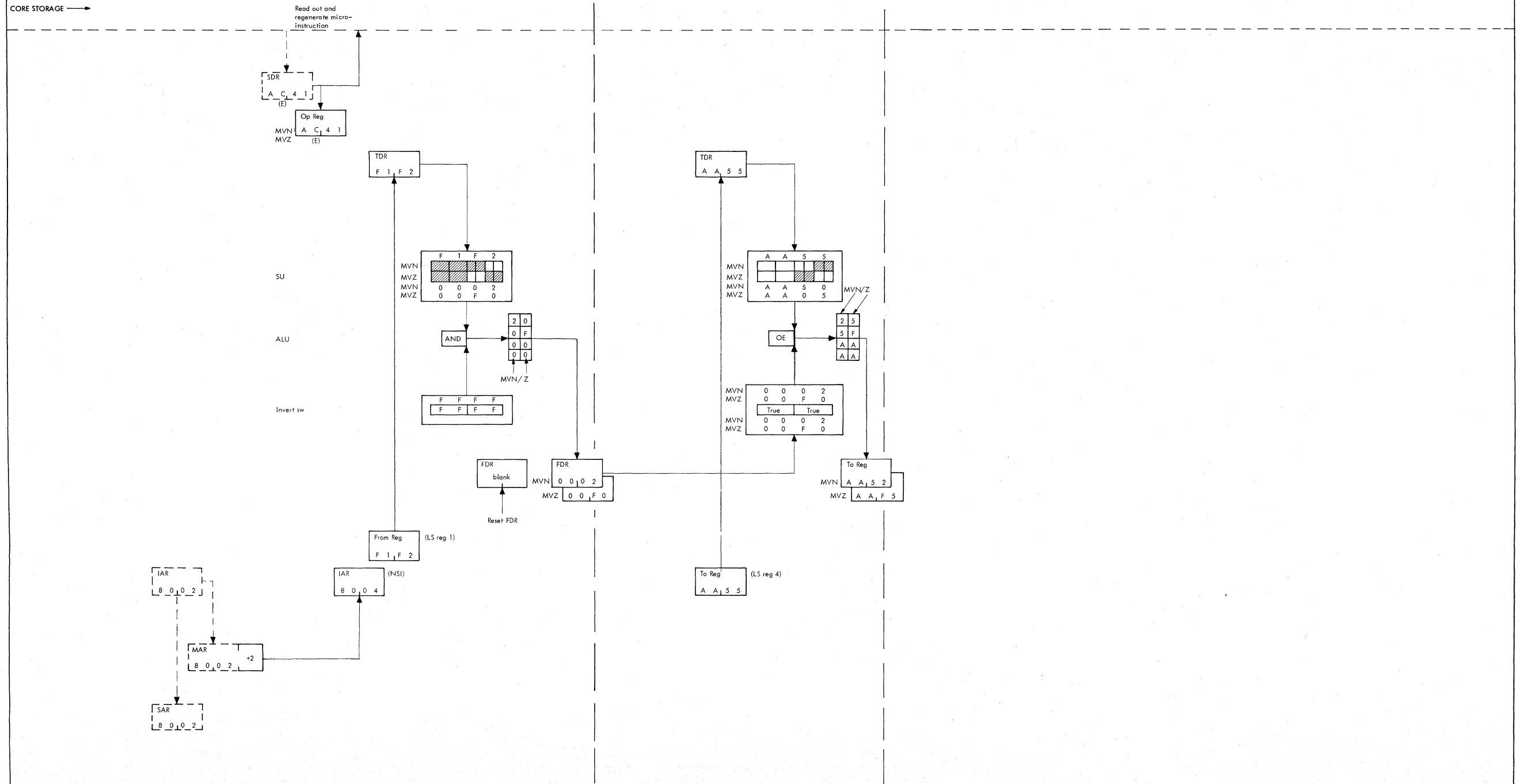
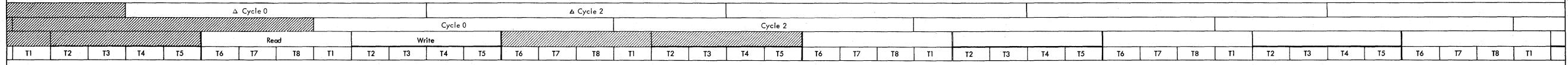
Microinstruction Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Op Code								AC	To Reg				From Reg			
A	1	1	0					0				0				
A	1	1	1					0				0				

Ignored → DD →

INST AC41 AE41 MNEM MVN MVZ OPERANDS 4,1 4,1 STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX R4=R4,0-11/R1,12-15 R4=R4,0-7/R1,8-11/R4,12-15

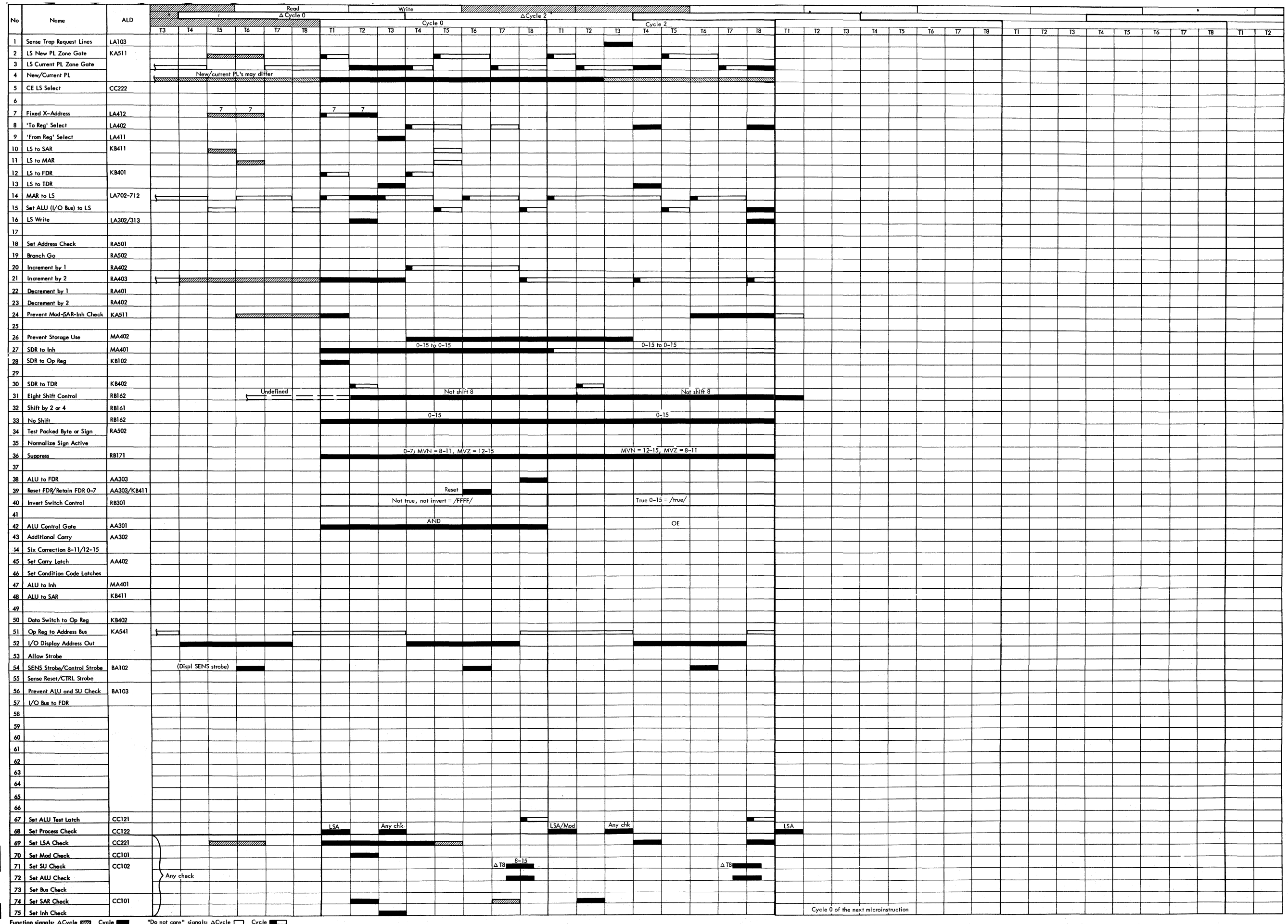
Mnemonic
MVN
MVZ
Format
FF
Type
DD



Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

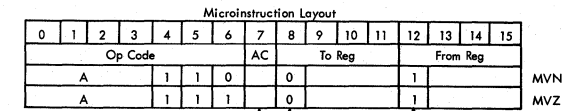
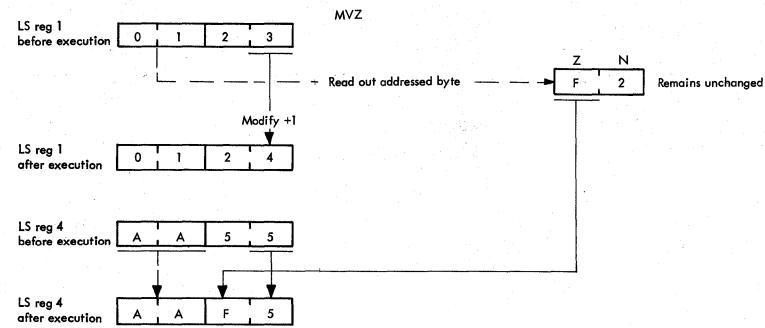


● Diagram 5-38. Move Numeric/Zone (DD) (Part 2 of 2) (03743A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

The numeric (zone) of the byte addressed by the 'from reg' is moved into the numeric (zone) of the low-order byte of the 'to reg'.

The 'from reg' address is incremented by 1. If the AC bit (instruction bit 7) is on, an address check occurs when the 'from reg' address is outside customer area.

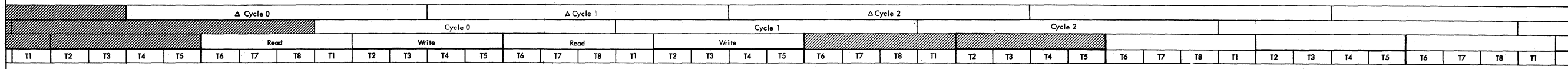
The 'from' byte, high-order byte of the 'to reg', and the zone (numeric) of the low-order byte of the 'to reg' remain unchanged.



If on, the 'from addr' is checked that it is not outside customer area

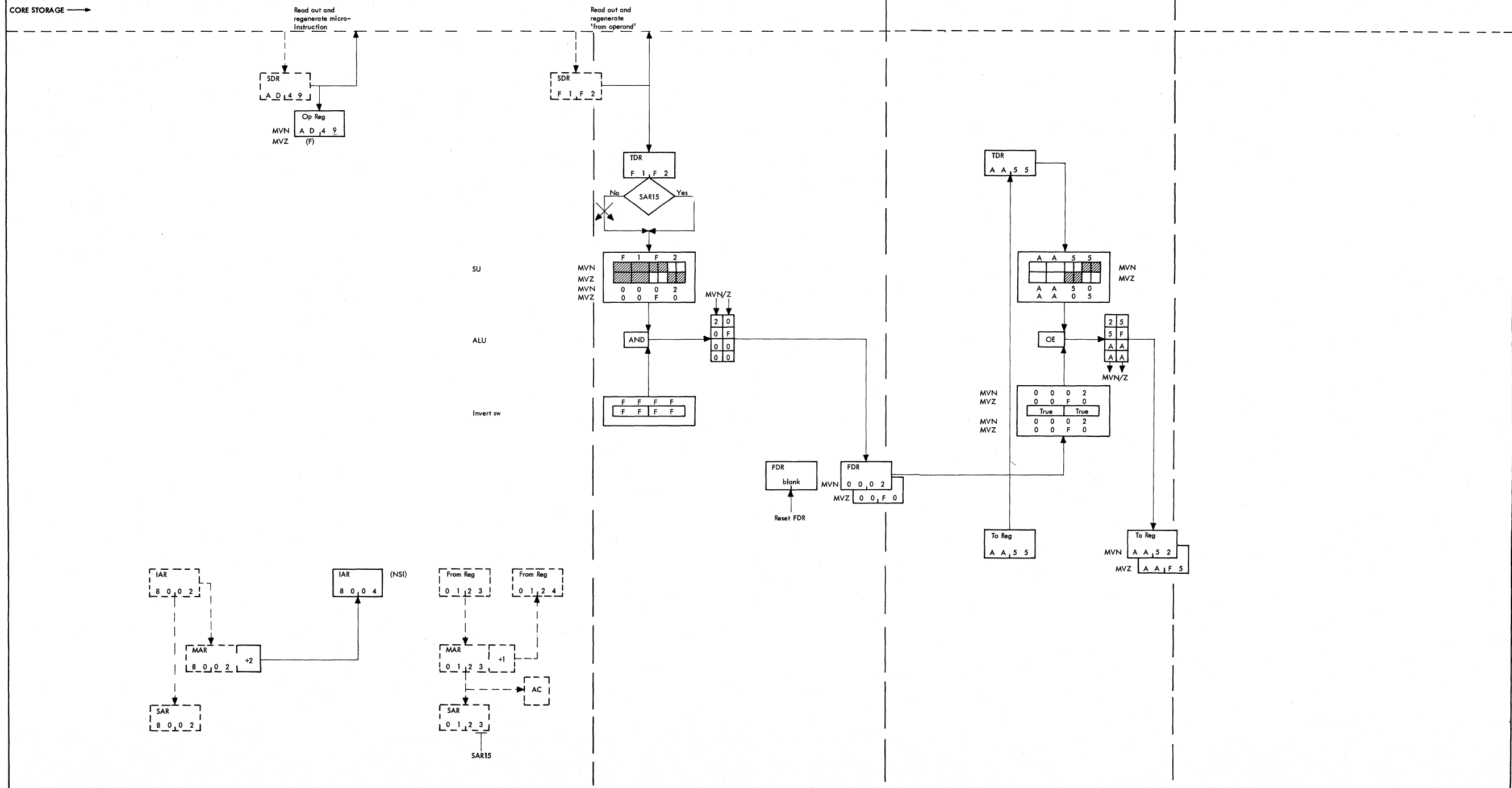
INST: AD49, AF49
 MNEM: MVN, MVZ
 OPERANDS: 4, 11, AC
 STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX: R4=R4.0-11/BY(R1, AC, +1), 4-7; R4=R4.0-7/BY(R1, AC, +1), 0-3/R4. 12-15

Mnemonic
MVN
MVZ
Format
FF
Type
DX

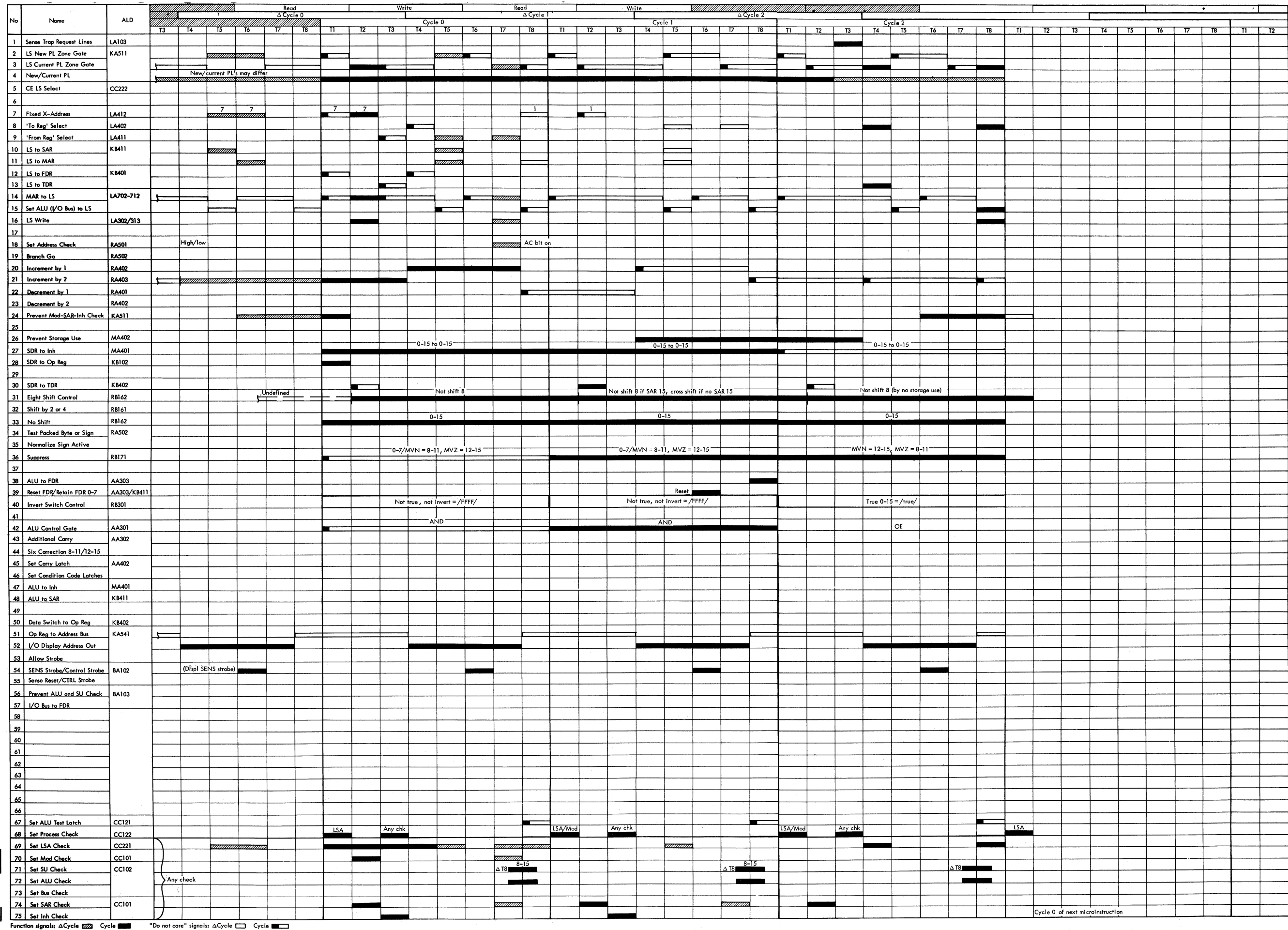


Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle



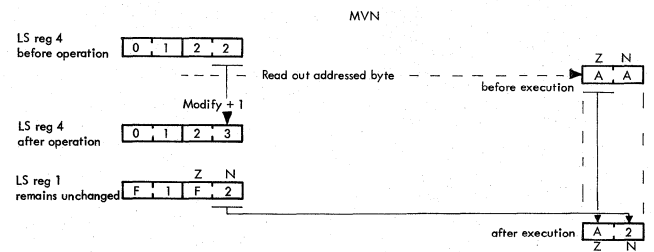
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



The numeric (zone) of the low-order byte of the 'from reg' is moved into the numeric (zone) of the byte addressed by the 'to reg'.

The 'from reg' and the zone (numeric) of the 'to' byte remain unchanged.

The 'to reg' address is incremented by 1. If the AC bit (instruction bit 7) is on, an address check occurs when the 'to reg' address is outside customer area.



Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code							AC	To Reg				From Reg			
A	1	1	0				1					0			
A	1	1	1				1					0			

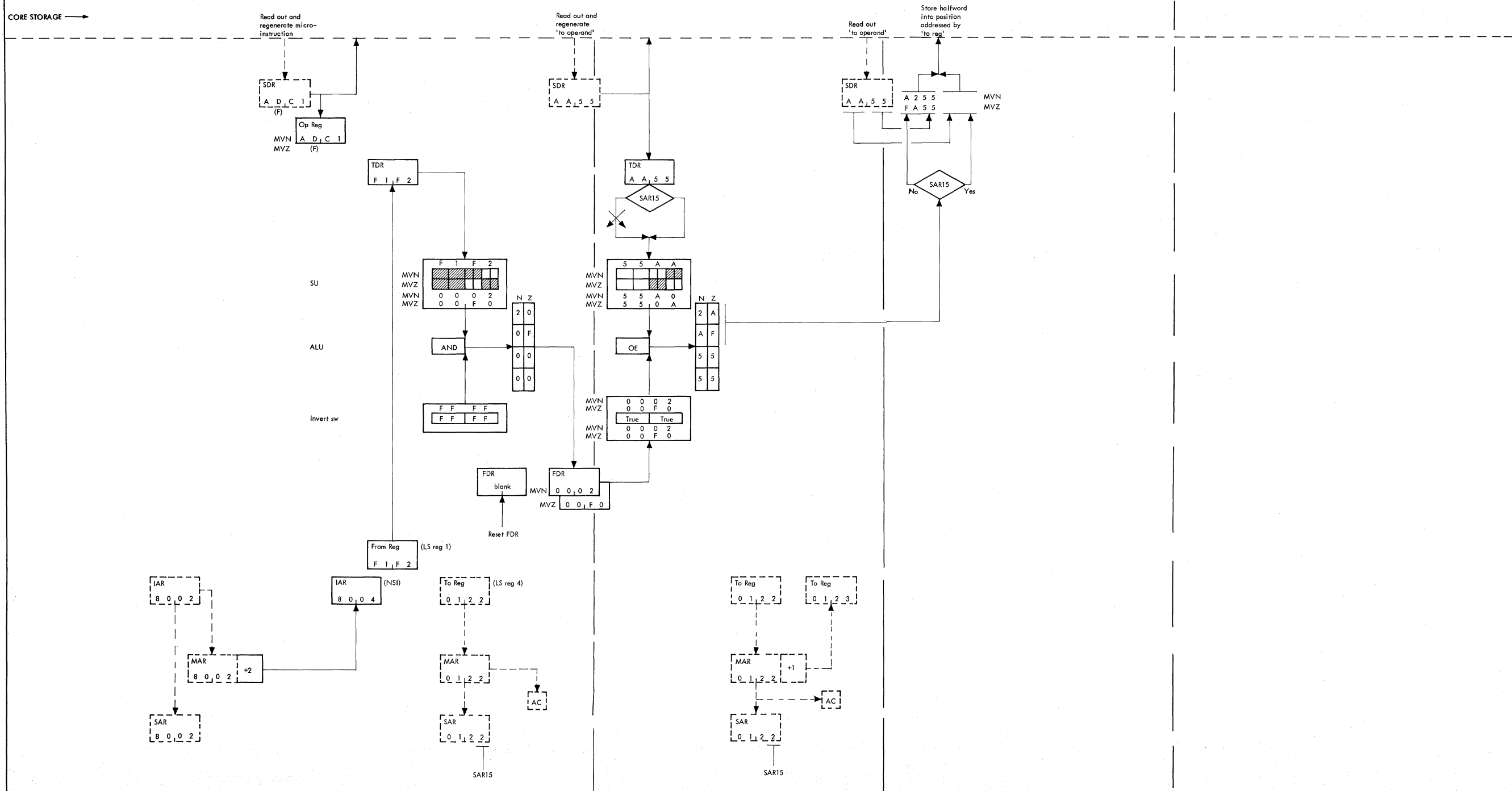
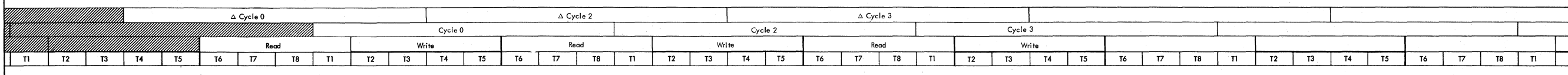
If on, the 'to addr' is checked that it is not outside customer area

INST MNEM OPERANDS STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX

ADCI MVN 41,1,AC BY(R4,AC,+1),4-7=R1,12-15

AFCI MVZ 41,1,AC BY(R4,AC,+1),0-3=R1,8-11

Mnemonic
MVN
MVZ
Format
FF
Type
XD



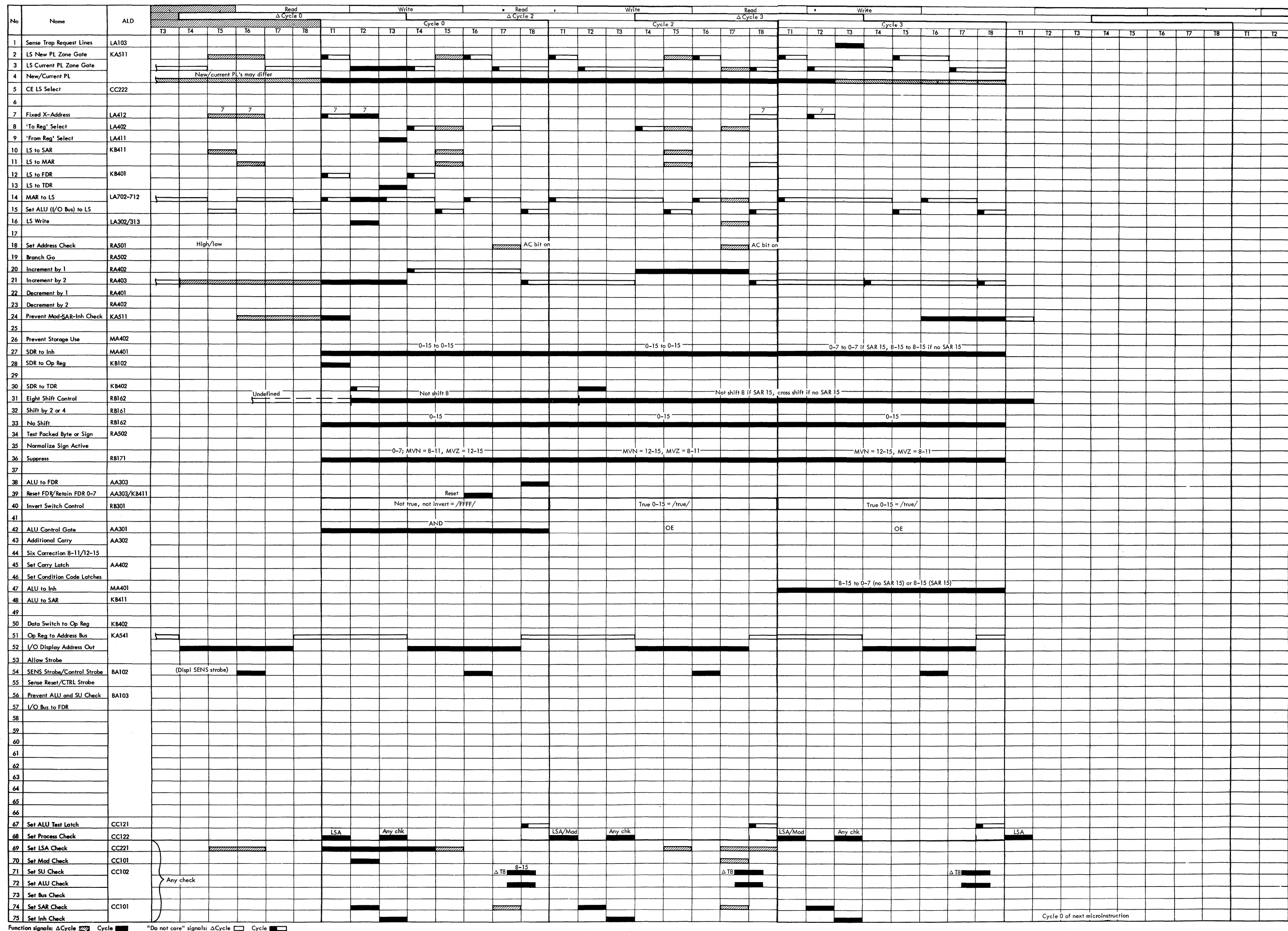
Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:

ΔCycle

Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



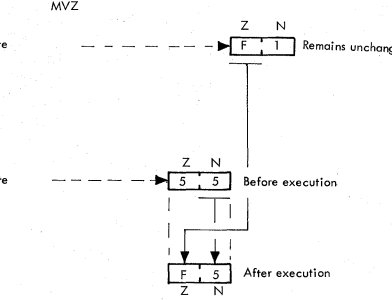
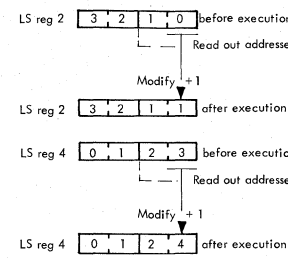
The numeric (zone) of the byte addressed by the 'from reg' is moved into the numeric (zone) of the byte addressed by the 'to reg'.

The 'from' byte and the zone (numeric) of the 'to' byte remain unchanged.

Both operand addresses are incremented by 1. If the AC bit (instruction bit 7) is on, an address check occurs when the 'from reg' or 'to reg' address is outside customer area.

The operation is performed in ALC mode when 'to reg' = 3 and 'from reg' = 5. LS register 1 contains the field length. The field length is the real number of numerics (zones) to be moved reduced by 1.

For ALC, the operand addresses are incremented by 1 every time one numeric (zone) has been moved.



Microinstruction Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code							AC	To Reg				From Reg			
A							1	1				1			
A							1	1				1			

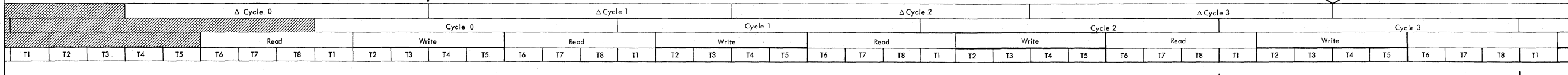
If on, the 'from' and 'to' add's are checked that they are not outside customer area

INST MNEM OPERANDS STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
 ADCA MVN 4I,2I,AC BY(R4,AC,+1).4-7*=BY(R2,AC,+1).4-7
 AFCA MVZ 4I,2I,AC BY(R4,AC,+1).0-3*=BY(R2,AC,+1).0-3
 AFBD MVZ 3I,5I,AC BY(R3,AC,+1,UNTIL R1,LT,0).0-3*=BY(R5,AC,+1).0-3

Mnemonic

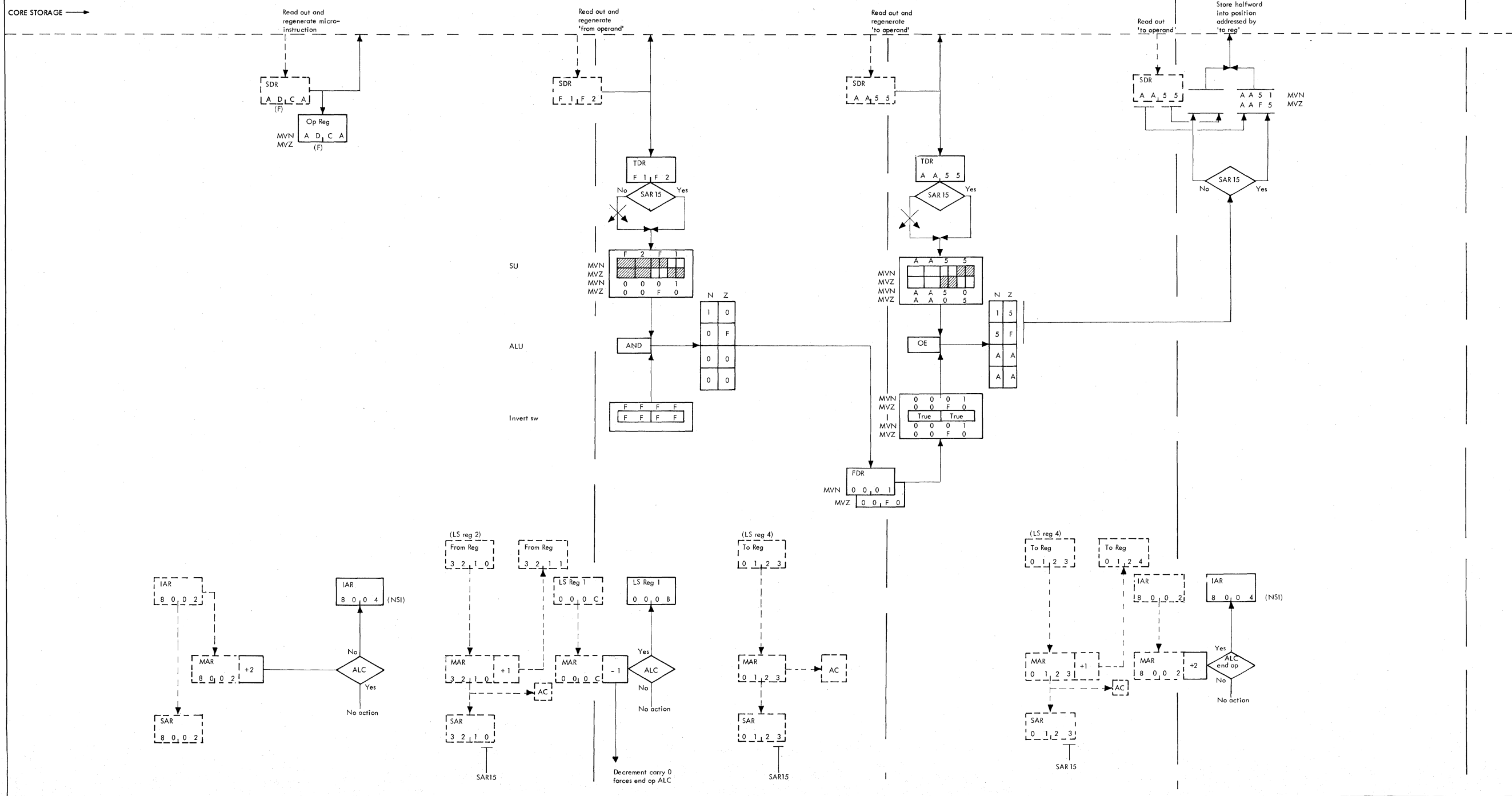
MVN
MVZ
FF
XX (ALC)

Repeat Δ Cycle 1 - Δ Cycle 2 - Δ Cycle 3

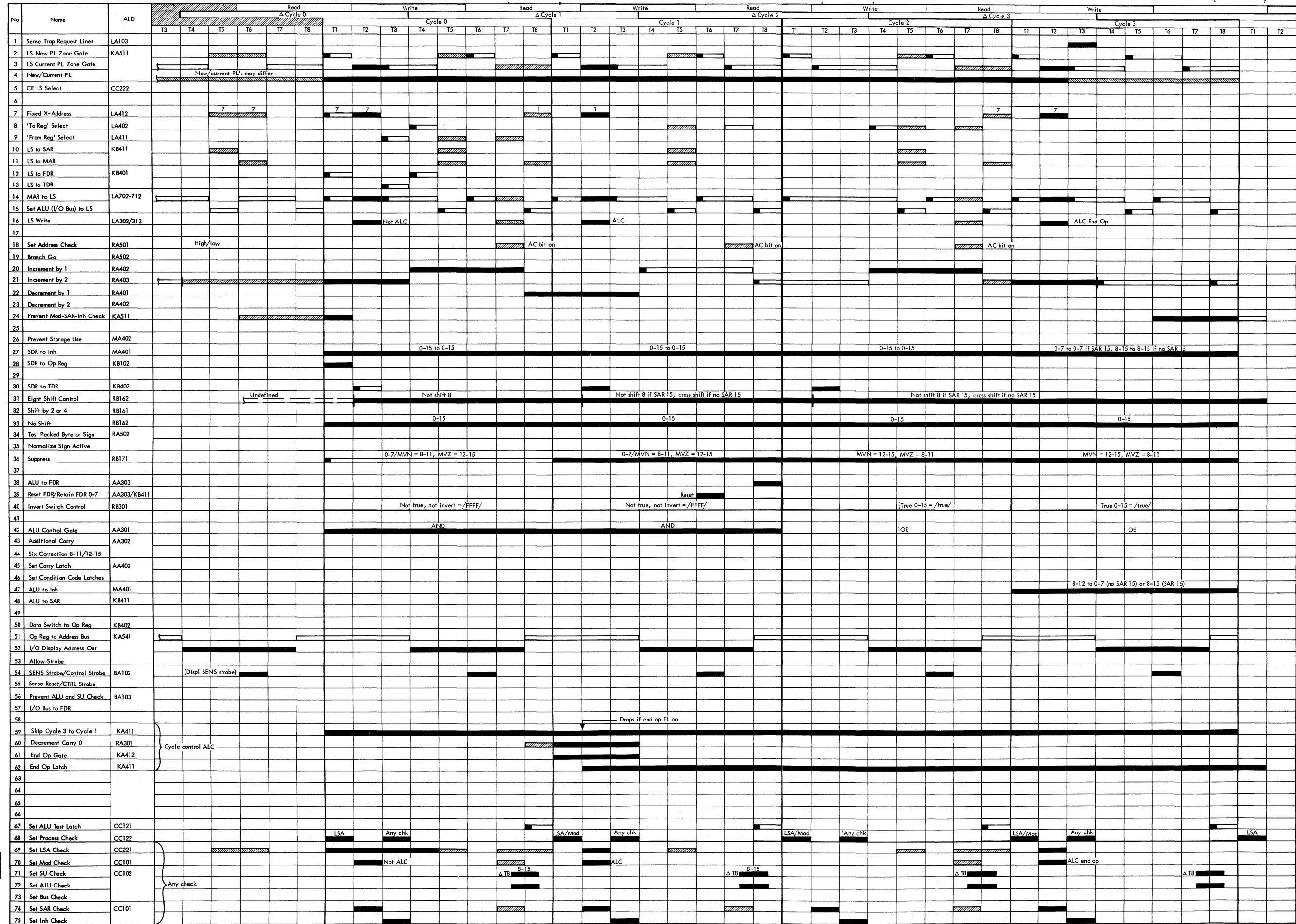


Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 Δ Cycle
 Cycle



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



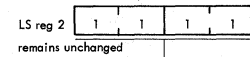
Function signals: Δ Cycle Cycle "Do not care" signals: Δ Cycle Cycle

The 'from reg' is added to (or subtracted from) the 'to reg'. The result is set into 'to reg'. The 'from reg' remains unchanged. The condition code latches are set if the CC bit (instruction bit 6) is on.

For set carry (SC) instructions a carry out of the halfword turns on the carry latch, and a previous carry is implemented in the addition (or subtraction).

For SHSC a previous carry must be simulated by turning on the carry latch (aux carry latch). The carry latch can be turned on by a CTRL microinstruction.

AH/AHSC



LS reg 4 (Negative binary number) before execution

Add to

No previous carry

LS reg 4 after execution

For SC No carry (carry latch off)

Condition Code			
0	1	2	3
0	1	0	0

Result less than zero

Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code						CC	AC	To Reg				From Reg			
B	0	0				0						0			
B	0	1				0						0			
B	1	0				0						0			
B	1	1				0						0			

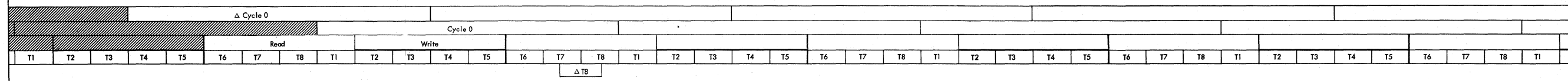
Set condition code if on

Ignored

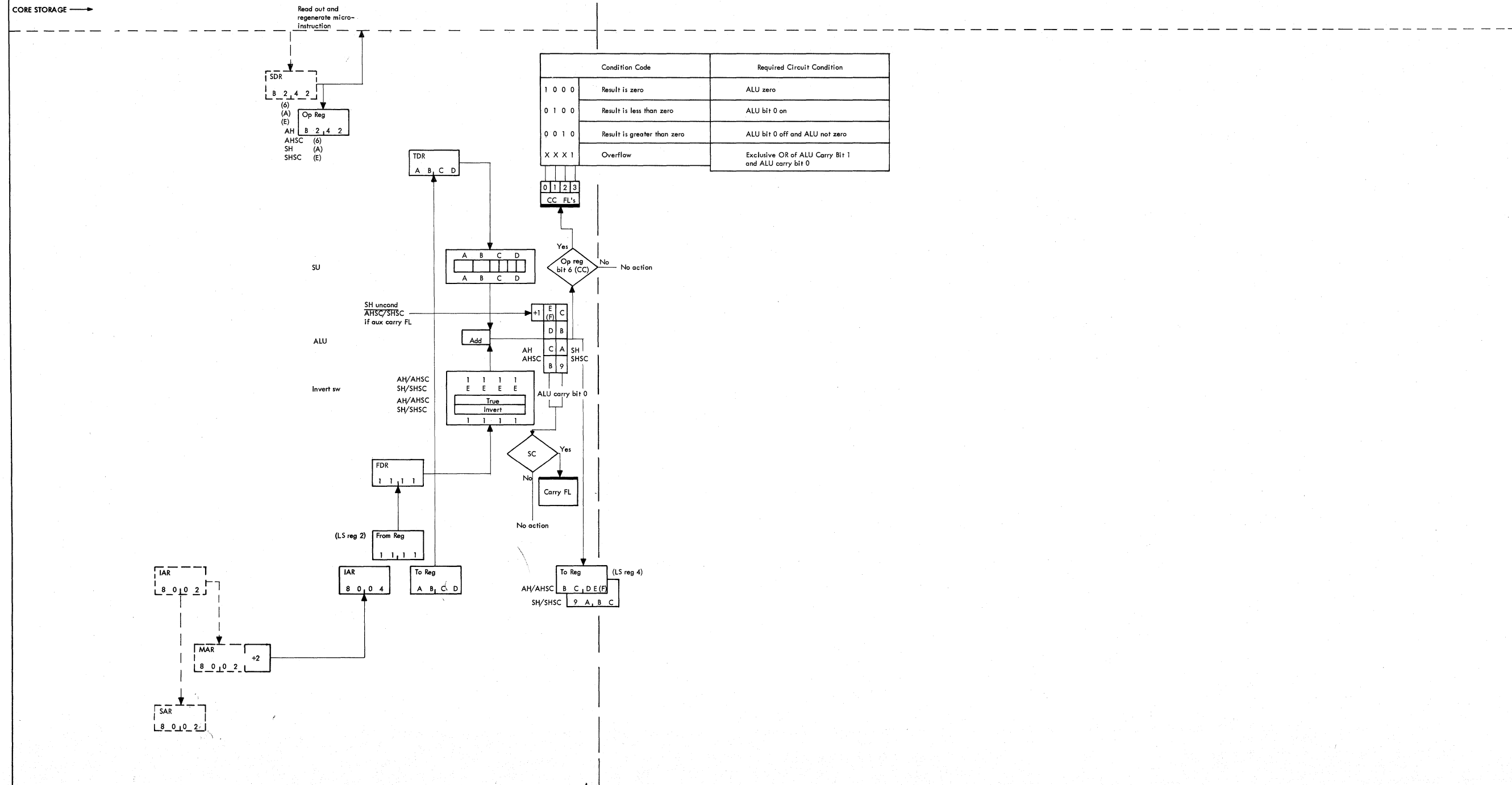
DD

INST	MNEM	OPERANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
B242	AH	4, 2, CC	CC, R4=R4+R2
B642	AHSC	4, 2, CC	CC, C/R4=R4+R2+C
BA42	SH	4, 2, CC	CC, R4=R4-R2
BE42	SHSC	4, 2, CC	CC, C/R4=R4-R2-NOT C

Mnemonic
AH
AHSC
SH
SHSC
Format
FF
Type
DD



Note: For "Do not care" functions refer to timing chart below.
 "Do not care" signals:
 ΔCycle
 Cycle



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

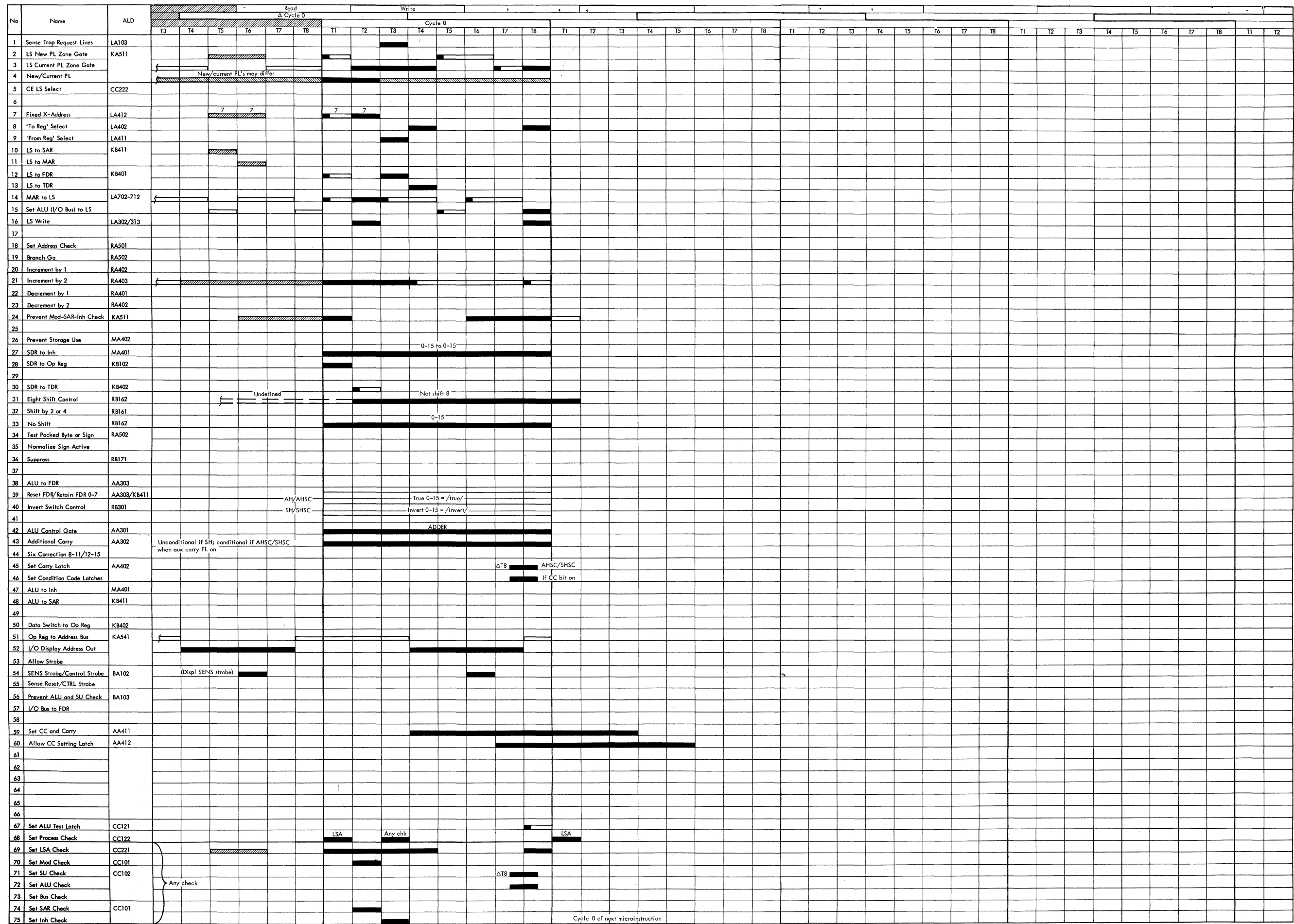
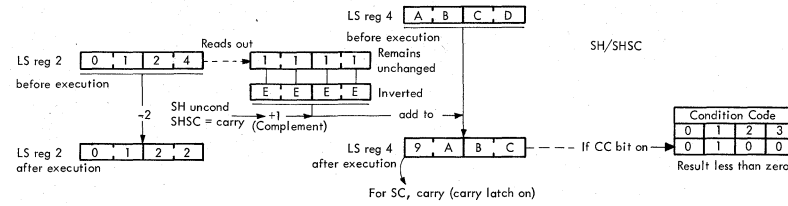


Diagram 5-42. Add/Subtract Halfword, Add/Subtract Halfword and Set Carry (DD) (Part 2 of 2) (03747A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

The halfword addressed by the 'from reg' is added to (or subtracted from) the 'to reg'. The result is set into 'to reg'. The 'from' halfword remains unchanged. The condition code latches are set if the CC bit (instruction bit 6) is on.

For SHSC a previous carry must be simulated, if not already present, to obtain a valid two's complement of the 'from operand'. The carry latch can be turned on by a ctrl/10, bit 11, as well as during arithmetical ALU operations. For SC instructions a carry during addition (subtraction) turns on the carry latch.

The 'from reg' address is decremented by 2. If the AC bit (instruction bit 7) is on, an address check occurs when the 'from reg' address is outside customer area or not on halfword boundary.



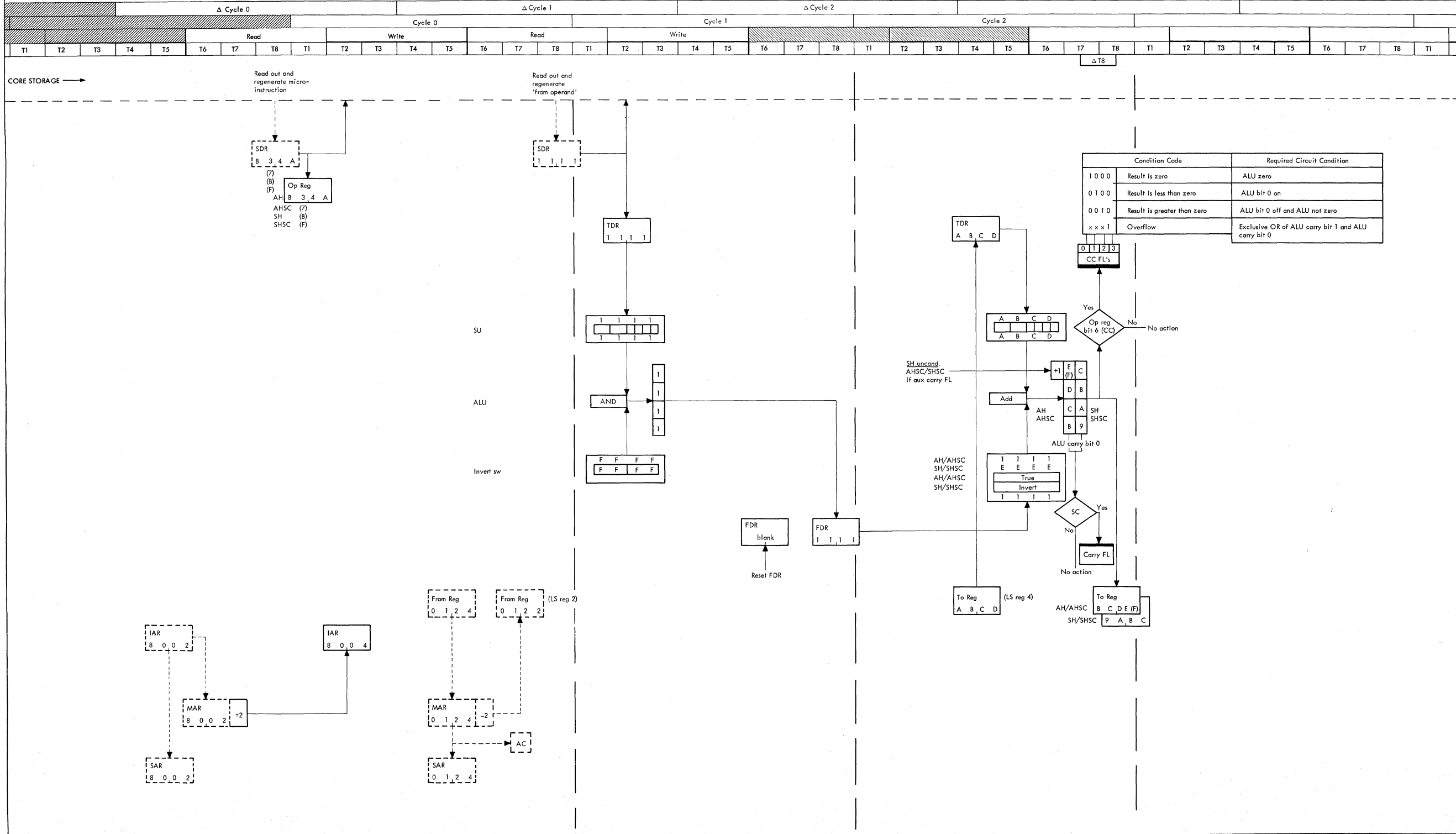
Microinstruction Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code						CC	AC	To Reg			From Reg				
B	0	0				0								1	
B	0	1				0								1	
B	1	0				0								1	
B	1	1				0								1	

Set condition code if on. If on, the 'from addr' is checked that it is not outside customer area. DX

INST	MNEM	OPERANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
B34A	AH	4, 2I, CC, AC	CC, R4 += R4 + HW(R2, AC, -2)
B74A	AHSC	4, 2I, CC, AC	CC, C/R4 += R4 + HW(R2, AC, -2) + C
BB4A	SH	4, 2I, CC, AC	CC, R4 += R4 - HW(R2, AC, -2)
BF4A	SHSC	4, 2I, CC, AC	CC, C/R4 += R4 - HW(R2, AC, -2) - NOT C

Mnemonic	AH
	AHSC
	SH
	SHSC
Format	
	FF
Type	
	DX



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

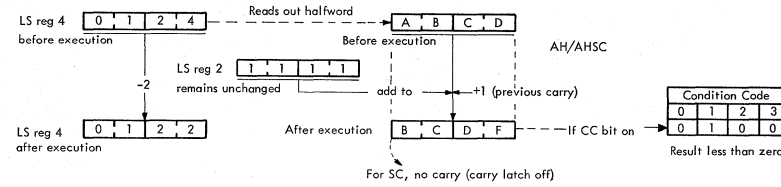
Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 Δ Cycle
 Cycle

The 'from reg' is added to (subtracted from) the halfword addressed by the 'to reg'. The result is set into the halfword addressed by the 'to reg'. The 'from reg' remains unchanged. The condition code latches are set if the CC bit (instruction bit 6) is on. A previous carry (aux carry latch on) is implemented in the addition (subtraction).

For SHSC a previous carry must be simulated, if not already present, to obtain a valid two's complement of the 'from operand'. The carry latch can be turned on by a ctrl /10/, bit 11, as well as during arithmetical ALU operations. For SC instructions, a carry during addition (subtraction) of the two halfwords turns on the carry latch.

The 'to reg' address is decremented by 2. If the AC bit (instruction bit 7) is on, an address check occurs when the 'to reg' address is outside customer area or not on halfword boundary.



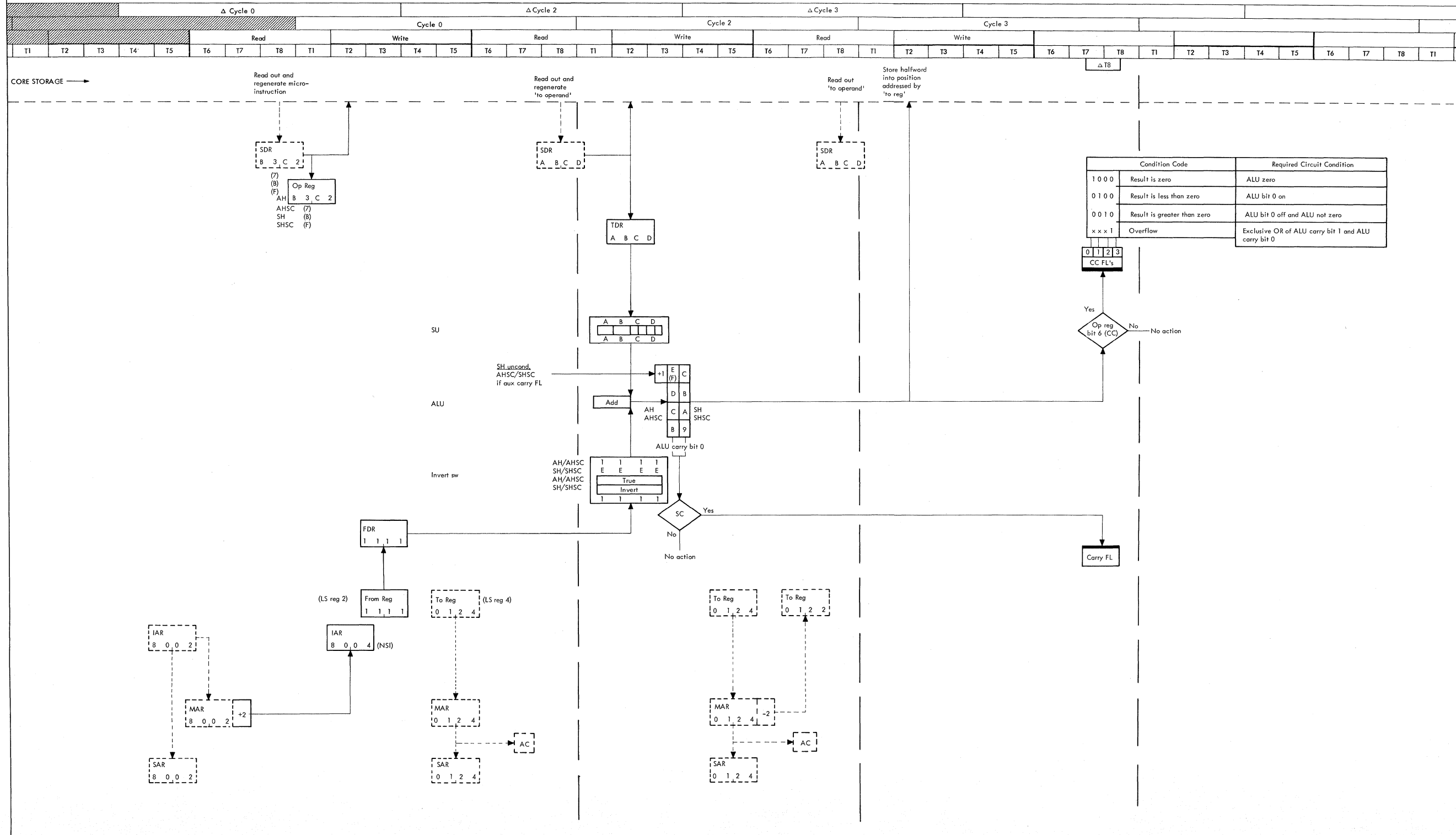
Microinstruction Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code						CC	AC	To Reg				From Reg			
B	0	0					1					0			
B	0	1					1					0			
B	1	0					1					0			
B	1	1					1					0			

Set condition code if on
If on, the 'to addr' is checked that it is not outside customer area

INST	MNEM	OPERANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
B3C2	AH	4I, 2, CC, AC	CC, HW(R4, AC) += HW(R4, -2) + R2
B7C2	AHSC	4I, 2, CC, AC	CC, C/HW(R4, AC) += HW(R4, -2) + R2 + C
BBC2	SH	4I, 2, CC, AC	CC, HW(R4, AC) += HW(R4, -2) - R2
BFC2	SHSC	4I, 2, CC, AC	CC, C/HW(R4, AC) += HW(R4, -2) - R2 - NOT C

Mnemonic
AH
AHSC
SH
SHSC
Format
FF
Type
XD



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
Δ Cycle
Cycle

The halfword addressed by the 'from reg' is added to (subtracted from) the halfword addressed by the 'to reg'. The result is set into the halfword addressed by the 'to reg'. The 'from' halfword remains unchanged.

The condition code latches are set if the CC bit (instruction bit 6) is on.

A previous carry (aux carry latch on) is implemented in the addition (subtraction).

by a ctrl/10, bit 11, as well as during arithmetical ALU operations.

For SC instructions a carry during addition (subtraction) of the two halfwords turns on the carry latch.

Both operand addresses are decremented by 2. If the AC bit (instruction bit 7) is on, an address check occurs when the 'from reg' or 'to reg' address is outside customer area or not on halfword boundary.

The operation is performed in ALC mode when 'to reg' = 3 and 'from reg' = 5; LS registers 1 contains the field length. The field length must reflect the

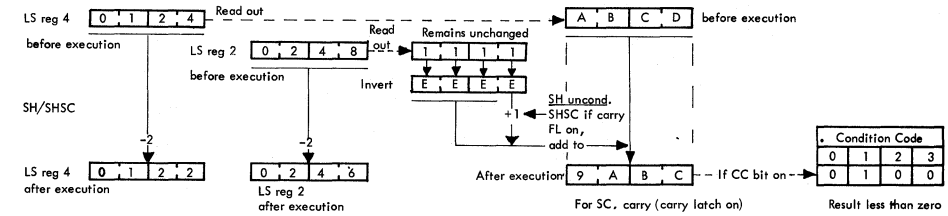
number of bytes to be operated (always an even number) reduced by 2.

For ALC, the operand addresses are updated by -2 every time a halfword is added (subtracted).

INST	MNEM	OPERANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
B3CA	AH	4I, 2I, CC, AC	CC, HW(R4, AC) += HW(R4, -2) + HW(R2, AC, -2)
B7CA	AHSC	4I, 2I, CC, AC	CC, C/HW(R4, AC) += HW(R4, -2) + HW(R2, AC, -2) + C
BBCA	SH	4I, 2I, CC, AC	CC, HW(R4, AC) += HW(R4, -2) - HW(R2, AC, -2)
BFC A	SHSC	4I, 2I, CC, AC	CC, C/HW(R4, AC) += HW(R4, -2) - HW(R2, AC, -2) - NOT C
B7BD	AHSC	3I, 5I, CC, AC	CC, C/HW(R3, AC) += HW(R3, -2, UNTIL R1, LT, 0) + HW(R2, AC, -2) + C

Op Code	CC	AC	To Reg	From Reg
B	0	0		
B	0	1		
B	1	0		
B	1	1		

Mnemonic
AH
AHSC
SH
SHSC
Format
FF
Type
XX (ALC)



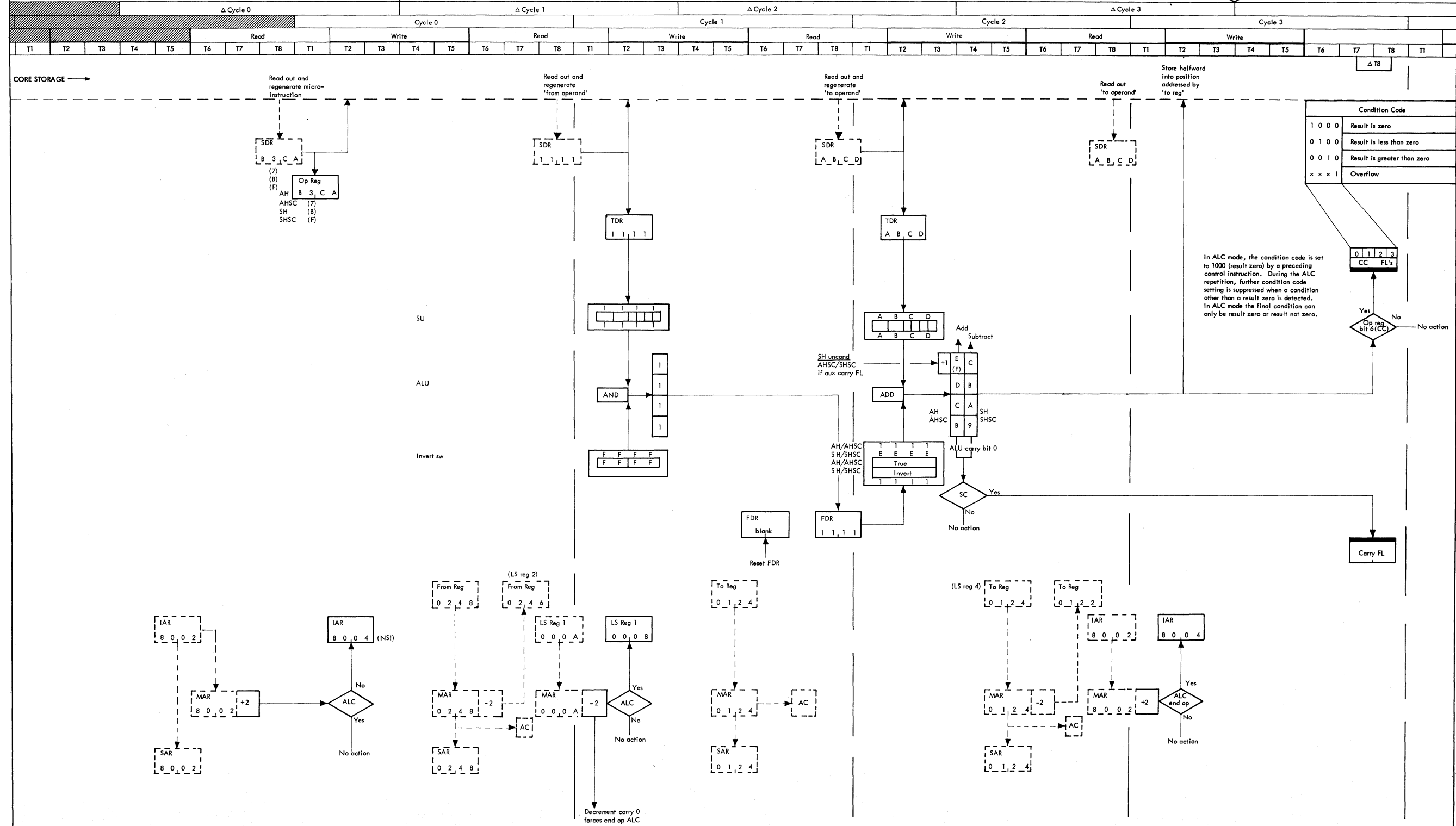
Set condition code if on. If on, the 'from' and 'to' addresses are checked that they are not outside customer area.

0	1	2	3
0	1	2	3
0	1	0	0

Repeat Δ Cycle 1 - Δ Cycle 2 - Δ Cycle 3. ALC end op? Yes/No.

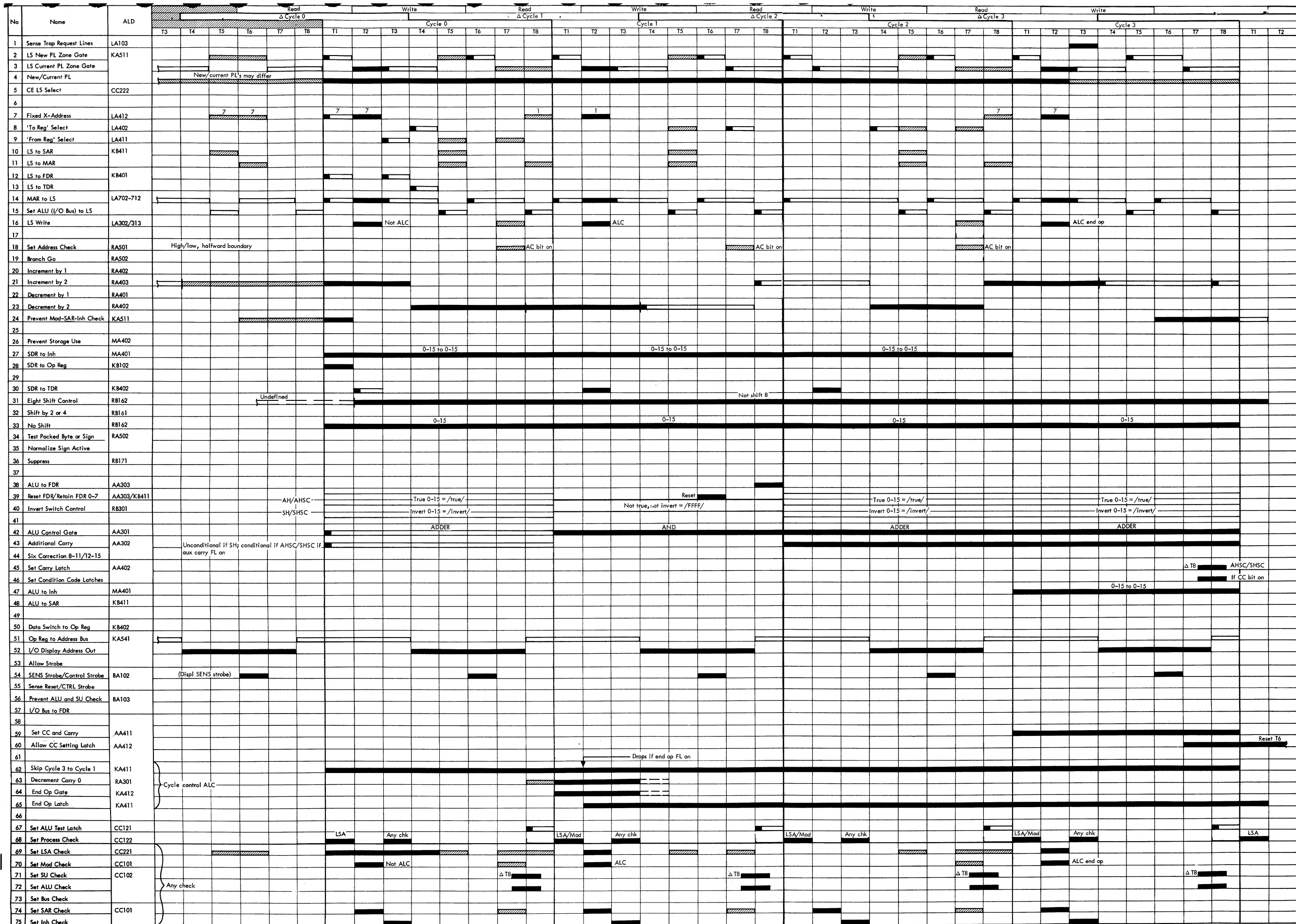
Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 Δ Cycle
 Cycle



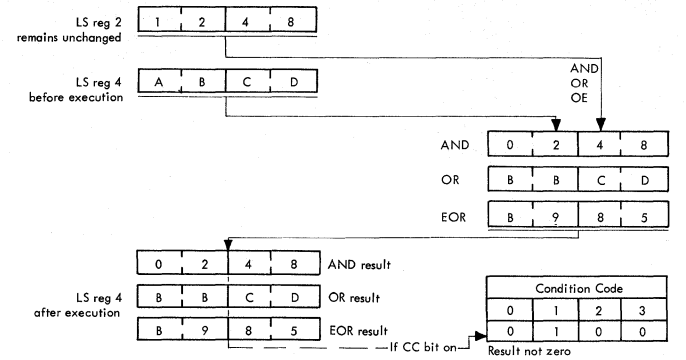
In ALC mode, the condition code is set to 1000 (result zero) by a preceding control instruction. During the ALC repetition, further condition code setting is suppressed when a condition other than a result zero is detected. In ALC mode the final condition can only be result zero or result not zero.

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



● Diagram 5-45. Add/Subtract Halfword, Add/Subtract Halfword and Set Carry (XX, ALC) (Part 2 of 2) (03750A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

The 'from reg' is ANDed, ORed, or exclusive ORed with the 'to reg'. The result is set into 'to reg'. The 'from reg' remains unchanged. The condition code latches are set if the CC bit (instruction bit 6) is on.
The address check is ignored.



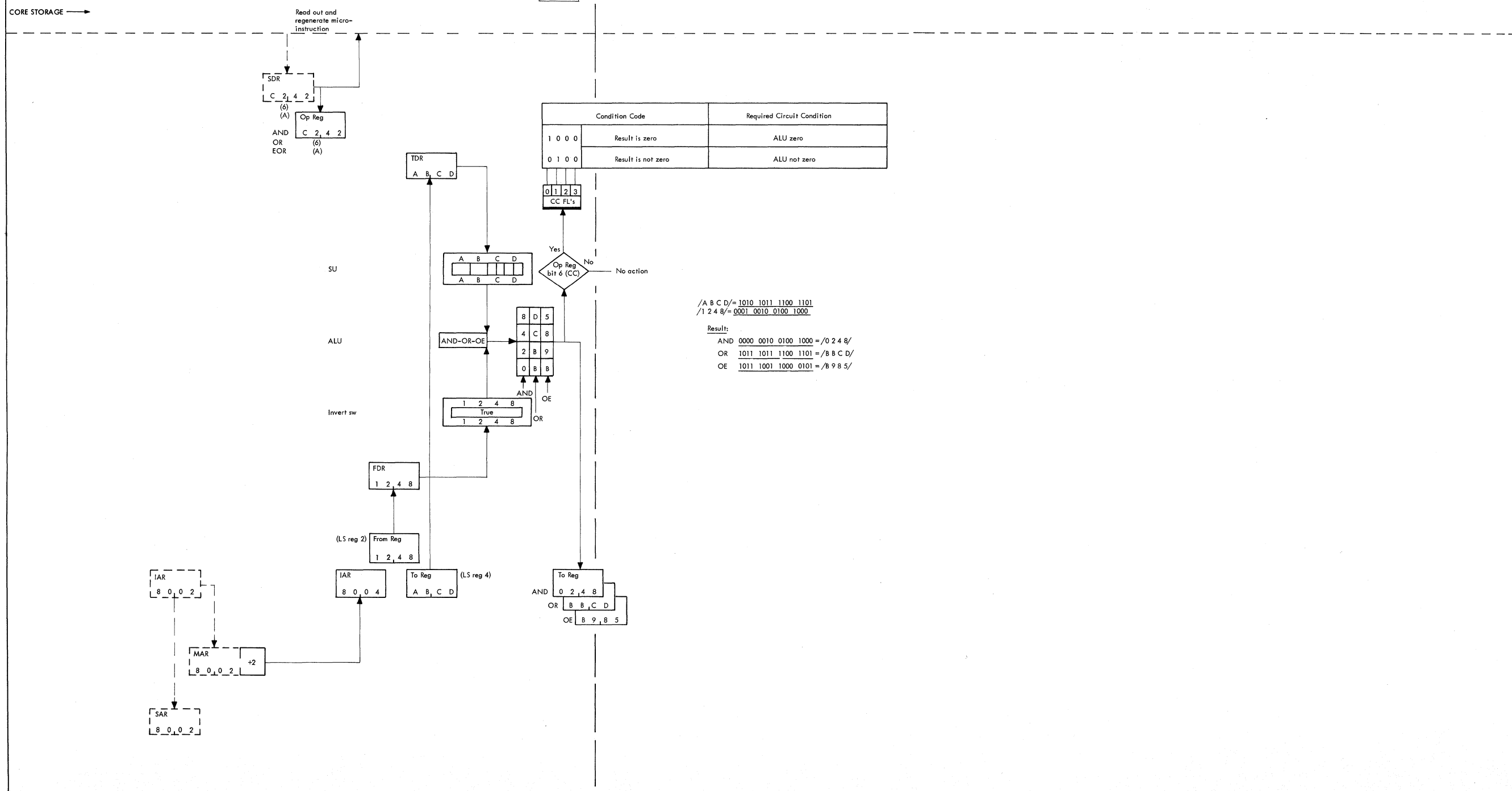
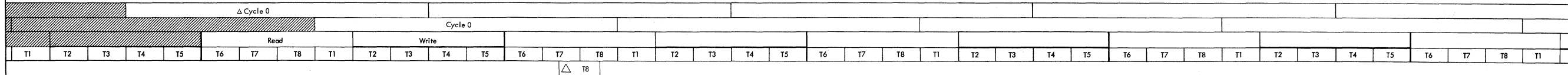
Microinstruction Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code						CC	AC	To Reg				From Reg			
C	0	0				0						0			
C	0	1				0						0			
C	1	0				0						0			

Set condition code if on → CC
Ignored → AC
DD → To Reg

Mnemonic
AND
OR
EOR
Format
FF
Type
DD

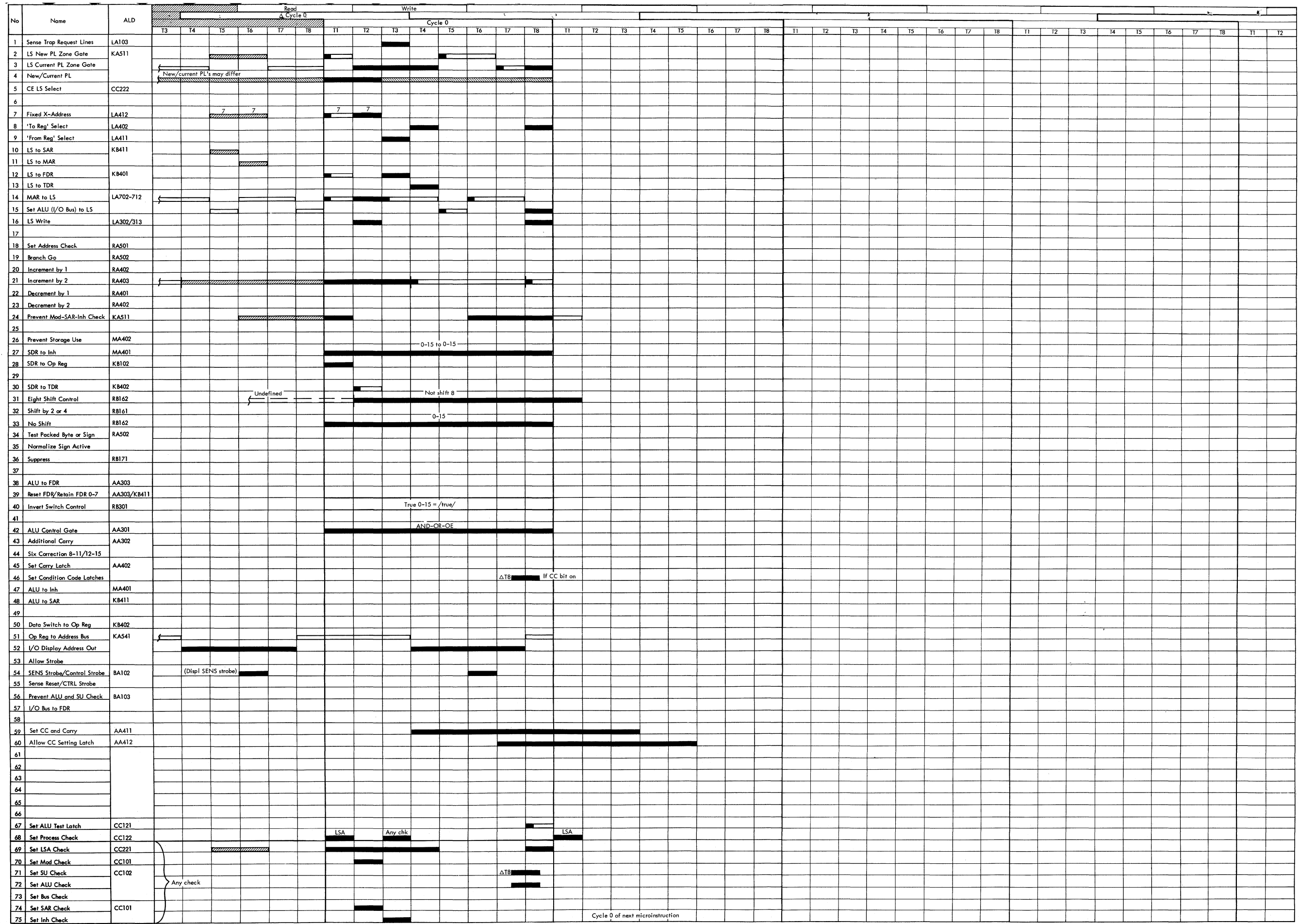
INST	MNEM	OPERANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
C242	AND	4, 2, CC	CC, R4 += R4, A, R2
C642	OR	4, 2, CC	CC, R4 += R4, OR, R2
CA42	EOR	4, 2, CC	CC, R4 += R4, OE, R2



Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 Δ Cycle
 Cycle

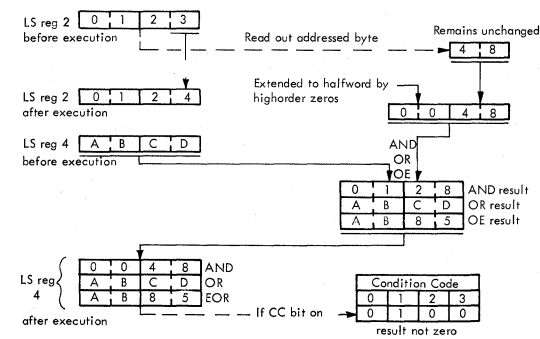
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



Function signals: ΔCycle Cycle "Do not care" signals: ΔCycle Cycle

The byte addressed by the 'from reg' is ANDed, ORed, or exclusive ORed with the halfword in 'to reg'. The 'from' byte is extended to a halfword by high-order zeros. The result is set into 'to reg'. The 'from' byte remains unchanged. The condition code latches are set if the CC bit (instruction bit 6) is on.

The 'from reg' address is incremented by 1. If the AC bit (instruction bit 7) is on, an address check occurs when the from reg address is outside customer area.



Microinstruction Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code						CC	AC	To Reg				From Reg			
C	0	0				0		0				1			
C	0	1						0						1	
C	1	0						0						1	

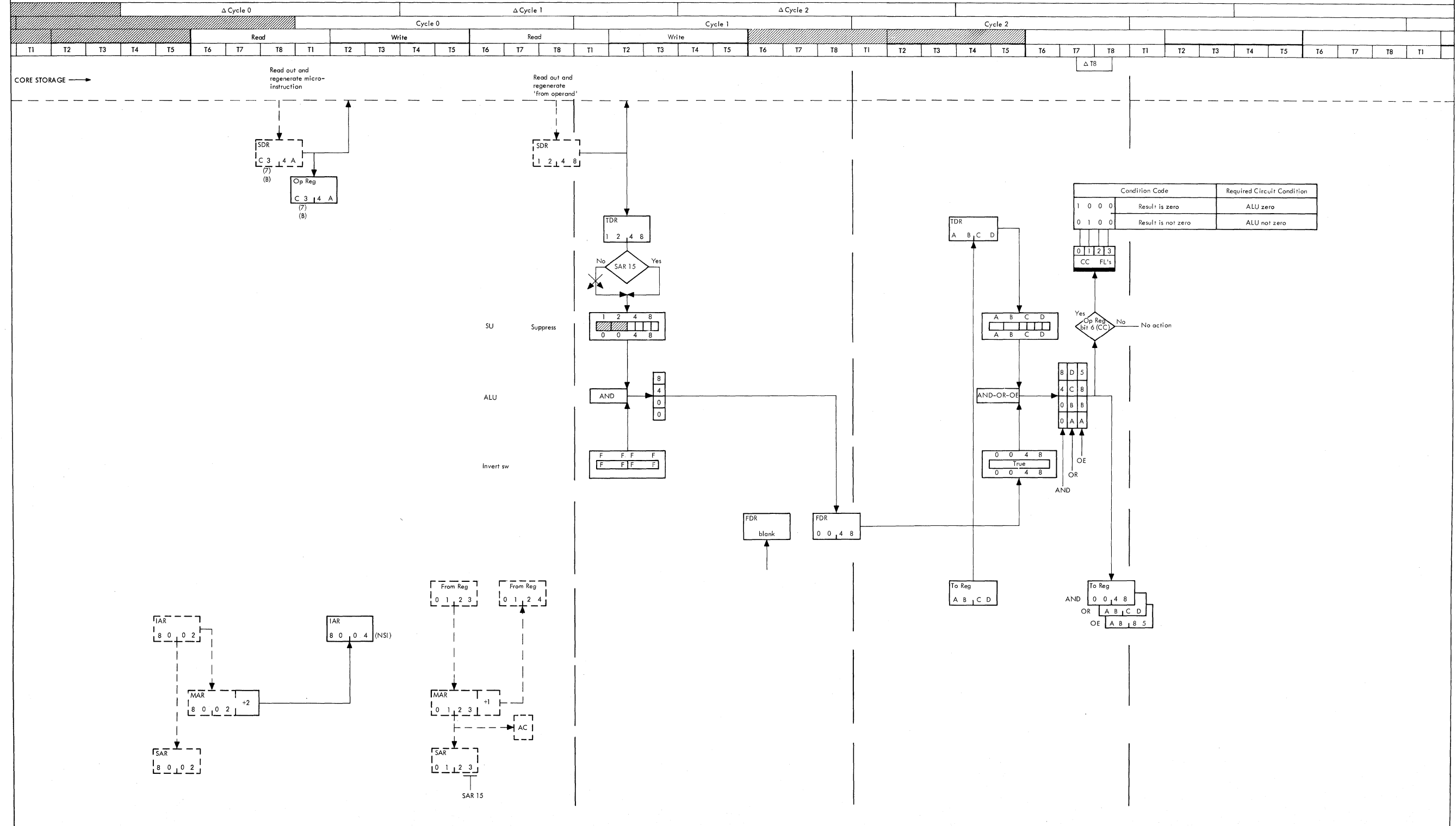
Set condition code if on → CC
If on, the 'from addr' is checked that it is not outside customer area → AC
DX → To Reg

INST	MNEM	OPERANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
C34A	AND	4,2I,CC,AC	CC,R4==R4,A,'00'/BY(R2,AC,+1)
C74A	OR	4,2I,CC,AC	CC,R4==R4,OR,'00'/BY(R2,AC,+1)
CB4A	EOR	4,2I,CC,AC	CC,R4==R4,OE,'00'/BY(R2,AC,+1)

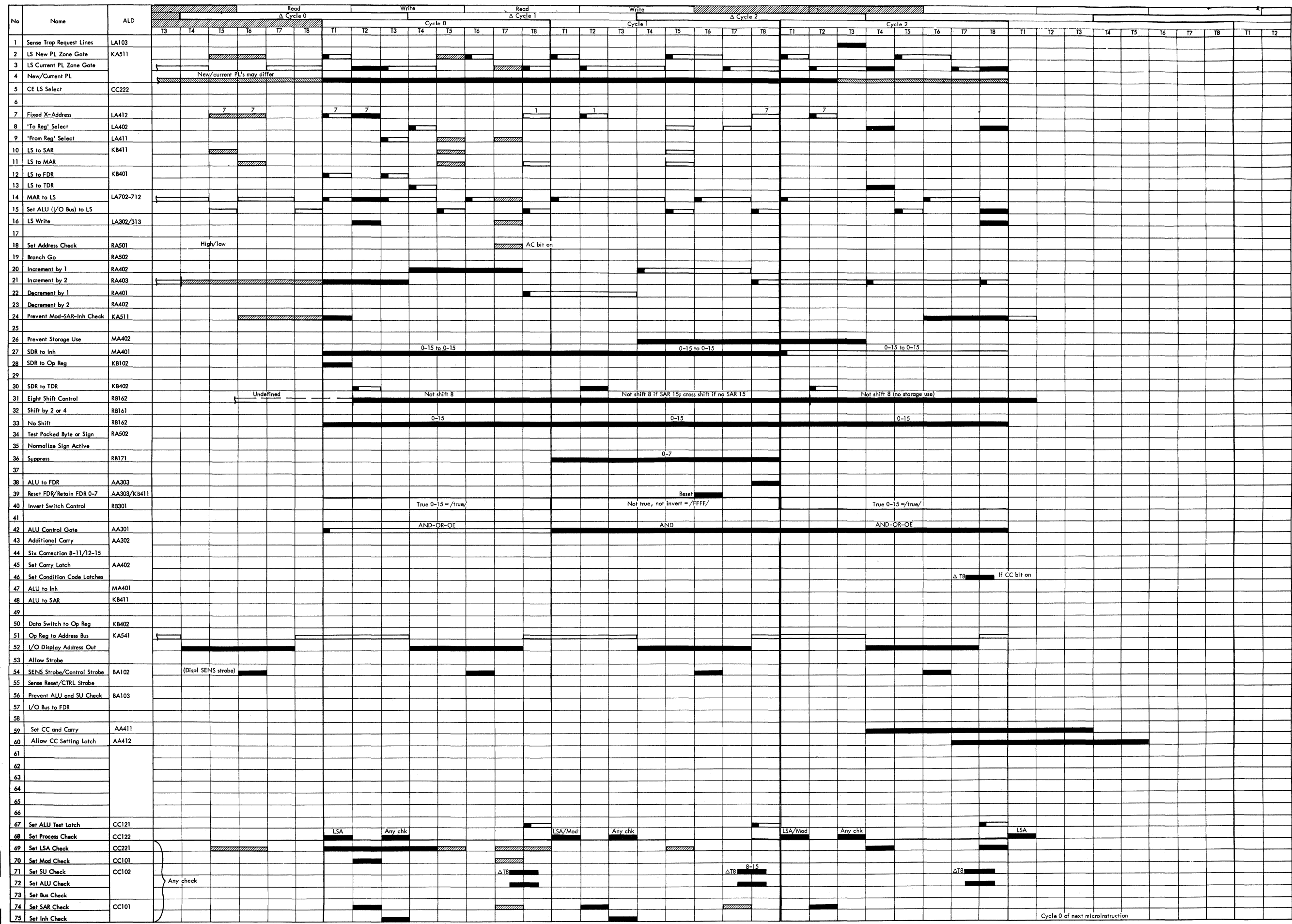
Mnemonic
AND
OR
EOR
Format
FF
Type
DX

Note: For "Do not care" functions refer to timing chart below.

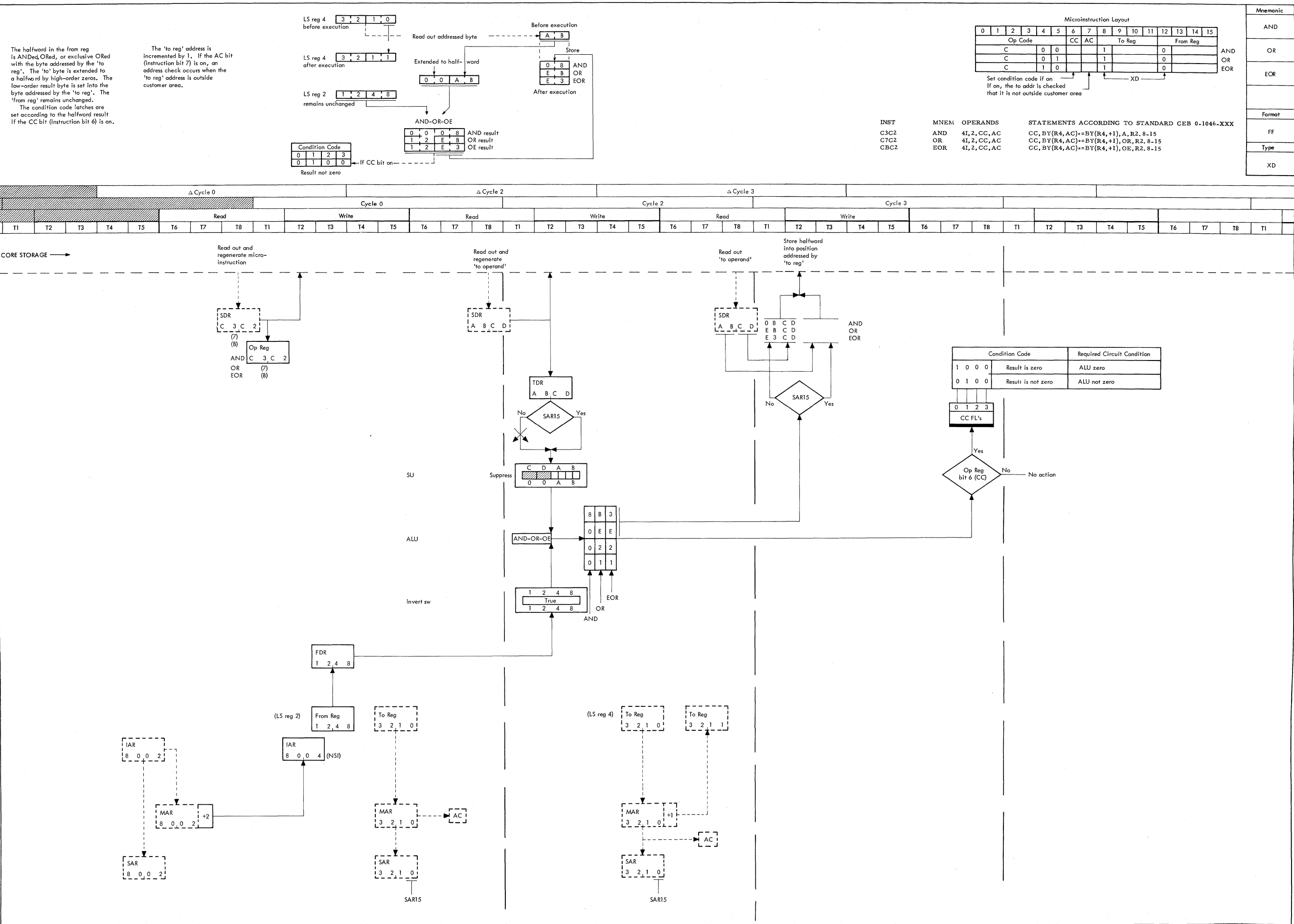
"Do not care" signals:
 ΔCycle
 Cycle



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

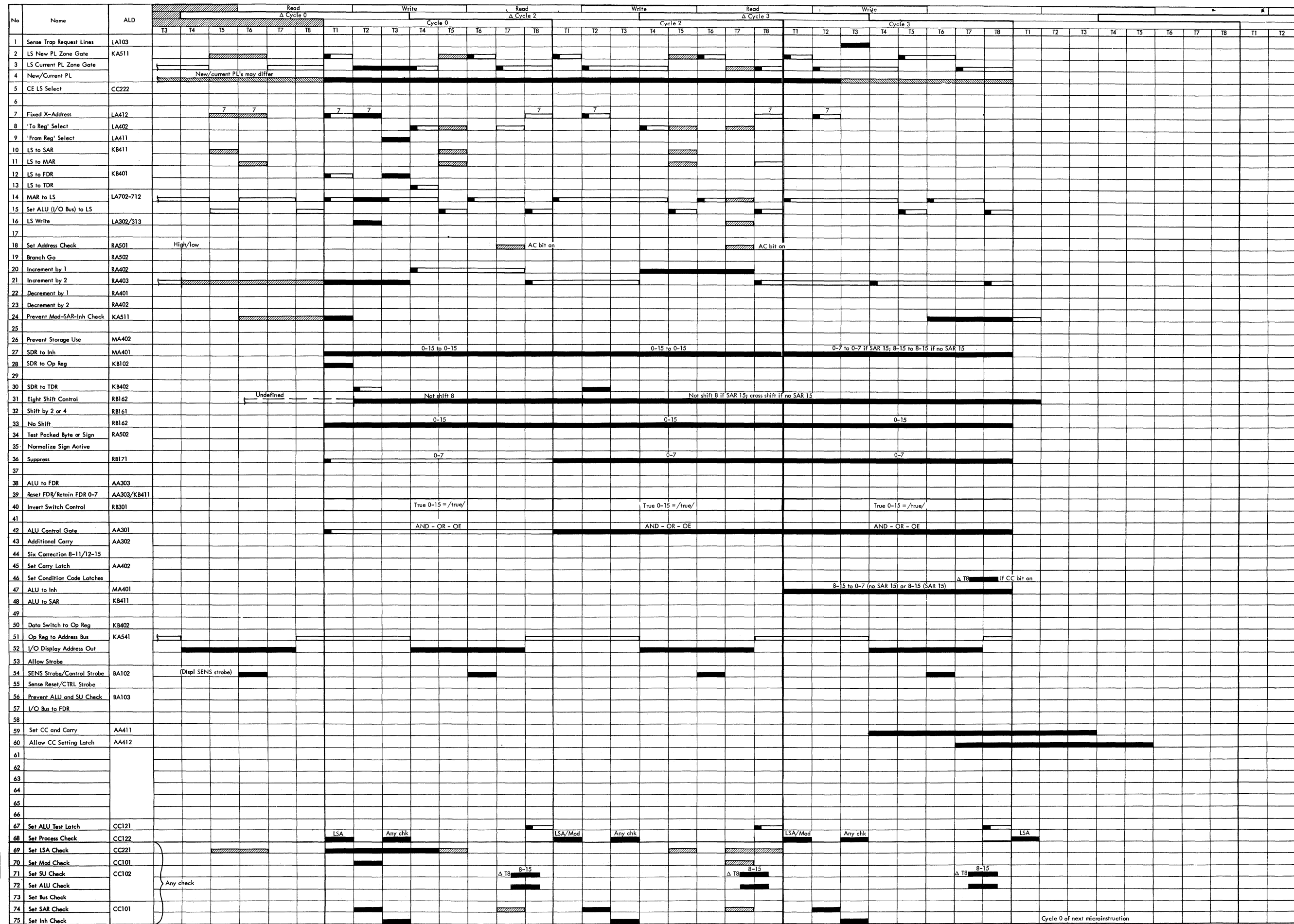


● Diagram 5-47. AND-OR-Exclusive OR Byte or Halfword (DX) (Part 2 of 2) (03752A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)



Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 Δ Cycle
 Cycle



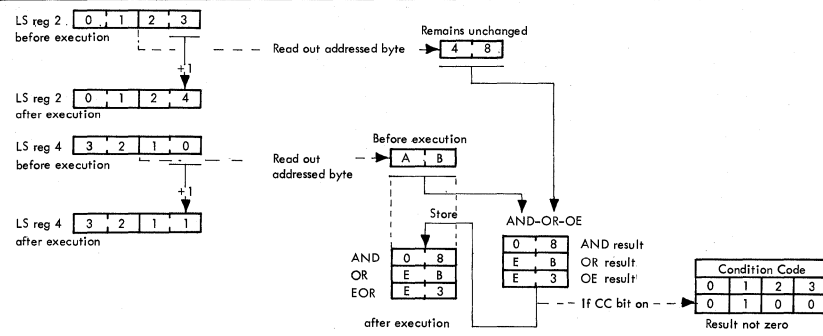
● Diagram 5-48. AND-OR-Exclusive OR Byte or Halfword (XD) (Part 2 of 2) (03753A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

The byte addressed by the 'from reg' is ANDed, ORed, or exclusive ORed with the byte addressed by the 'to reg'. The result is set into the byte addressed by the 'to reg'. The 'from' byte remains unchanged.

The condition code latches are set, according to the result, if the CC bit (instruction bit 6) is on. Both operand addresses are incremented by 1.

If the AC bit (instruction bit 7) is on, an address check occurs when the 'to reg' or 'from reg' address is outside customer area.

The operation is performed in ALC mode when 'to reg' = 3 and 'from reg' = 5; LS register 1 contains the field length. The field length is the real number of bytes to be ANDed, ORed, or exclusive ORed reduced by 1. The operand addresses are incremented by 1 every time a byte has been operated.



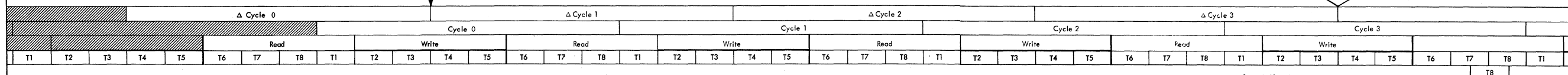
Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code		CC		AC		To Reg		From Reg							
C	0	0													
C	0	0													
C	1	0													

Set condition code if on: CC, AC, To Reg, From Reg, XX

If on, the 'from' and 'to' addrs are checked that they are not outside customer area

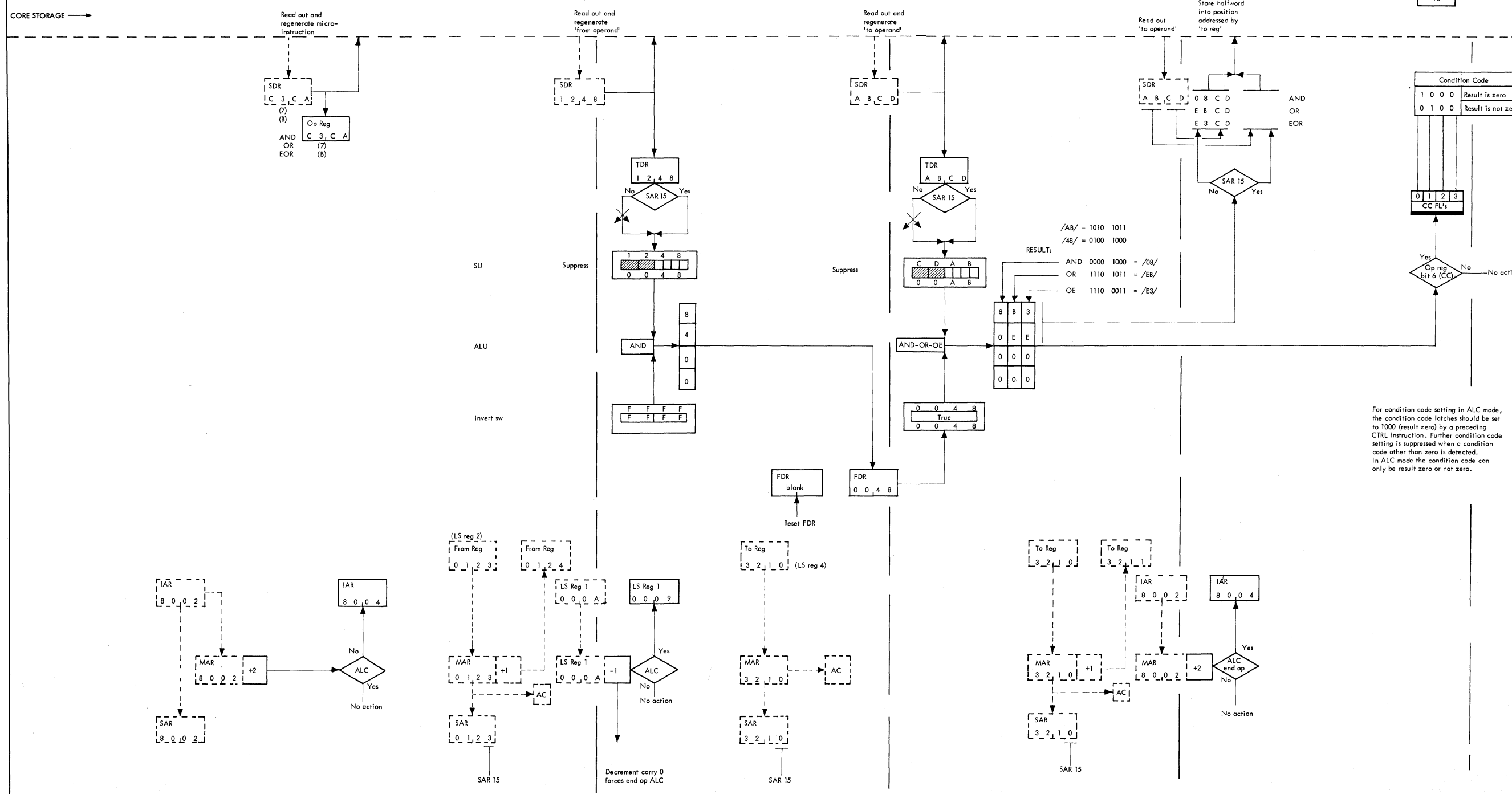
INST	MNEM	OPERANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
C3CA	AND	41, 21, CC, AC	CC, BY(R4, AC) += BY(R4, +1), A, BY(R2, AC, +1)
C7CA	OR	41, 21, CC, AC	CC, BY(R4, AC) += BY(R4, +1), OR, BY(R2, AC, +1)
CBGA	EOR	41, 21, CC, AC	CC, BY(R4, AC) += BY(R4, +1), OE, BY(R2, AC, +1)
C3BD	AND	31, 51, CC, AC	CC, BY(R3, AC, UNTIL R1, LT, 0) += BY(R3, +1), A, BY(R5, AC, +1)

Mnemonic	Format	Type
AND	FF	XX (ALC)
OR	FF	XX (ALC)
EOR	FF	XX (ALC)



Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

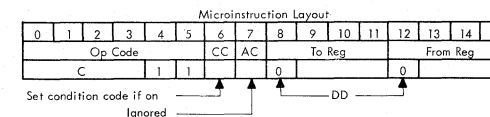
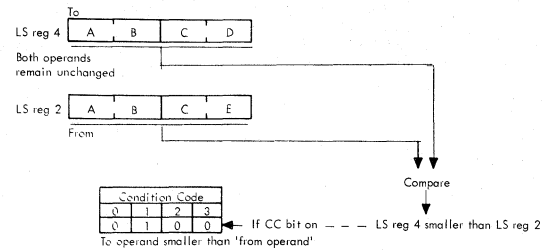


For condition code setting in ALC mode, the condition code latches should be set to 1000 (result zero) by a preceding CTRL instruction. Further condition code setting is suppressed when a condition code other than zero is detected. In ALC mode the condition code can only be result zero or not zero.

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

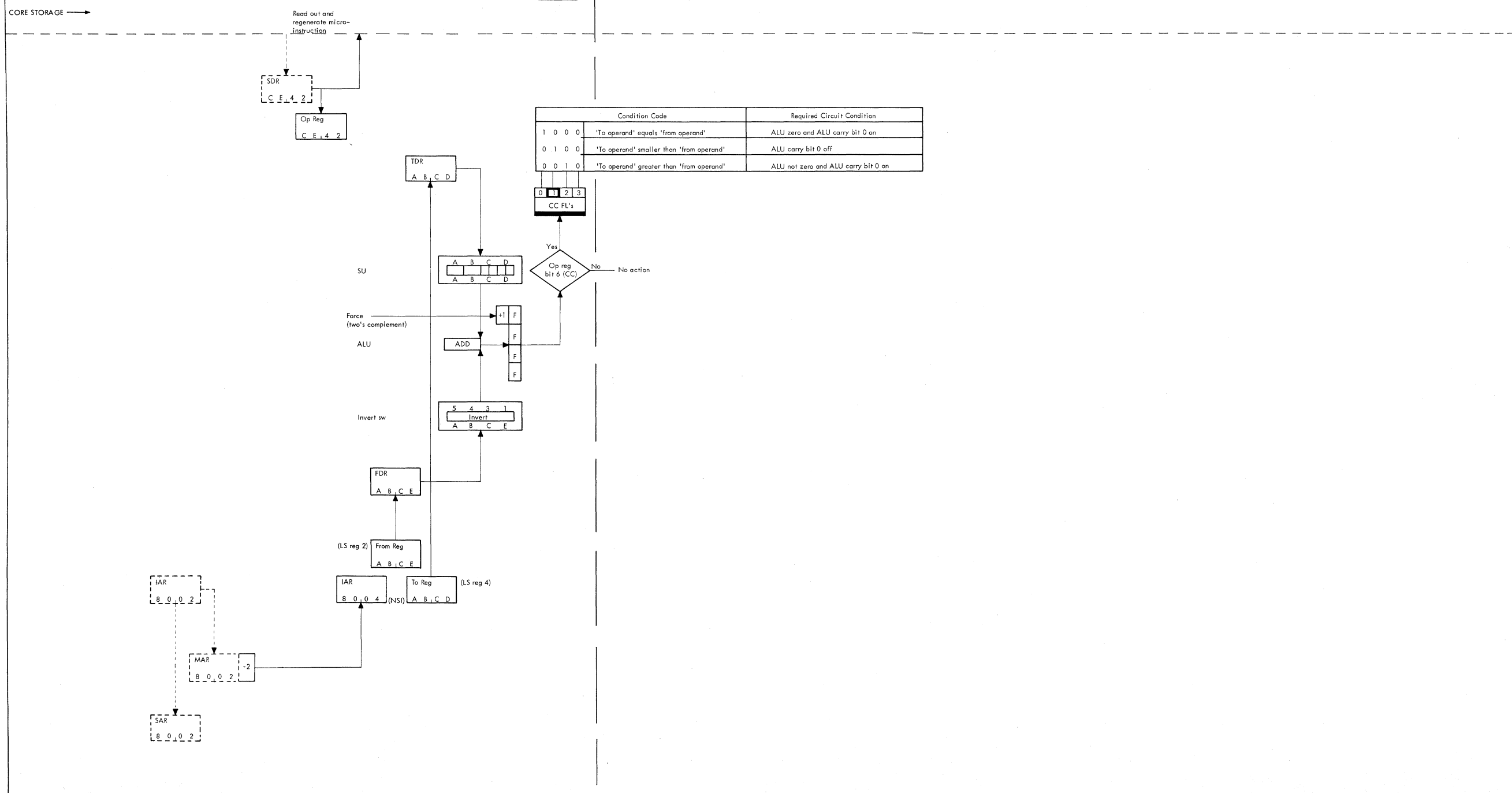
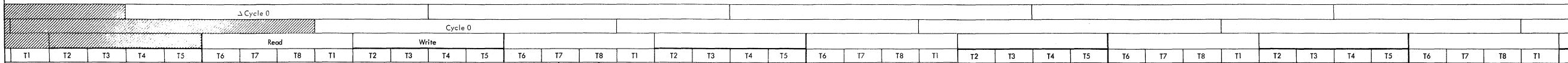
The 'from reg' is compared with the 'to reg'. 'To reg' and 'from reg' remain unchanged. The result of the comparison sets the condition code latches if the CC bit (instruction bit 6) is on. The comparison is done by subtracting the 'from operand' from the 'to operand'.

The address check is ignored.



INST M.NEM OPERANDS STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
GE42 CLC 4,2,CC CC,R4,COMP,R2

Mnemonic	CLC
Format	FF
Type	DD



Note: For "Do not care" functions refer to timing chart below.
"Do not care" signals:
ΔCycle
Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

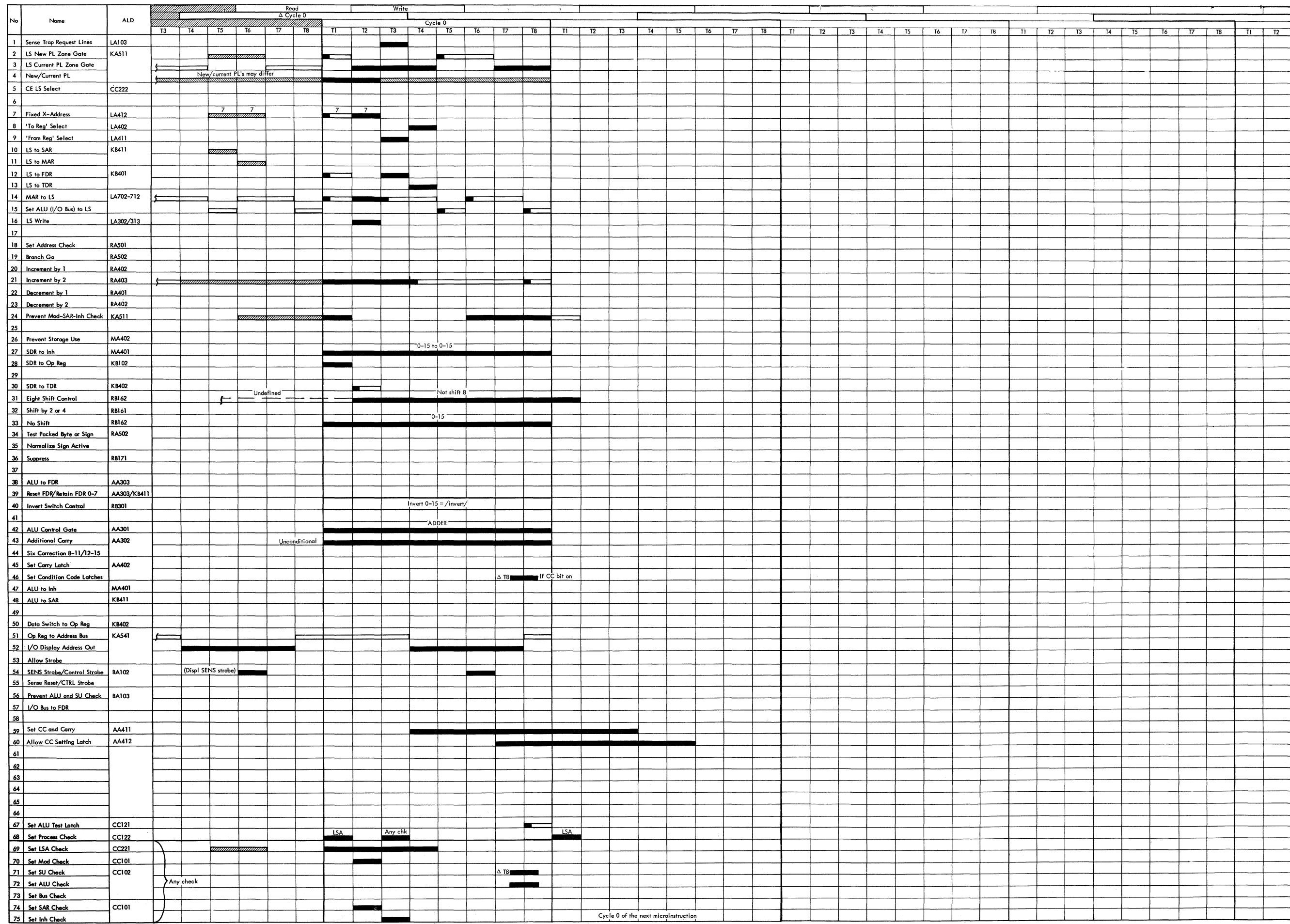


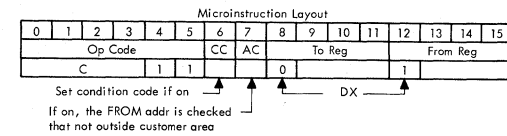
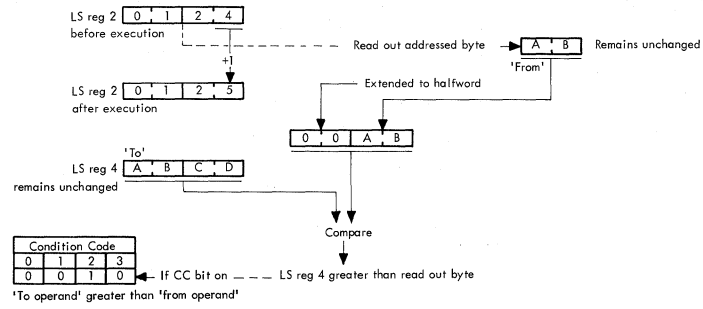
Diagram 5-50. Compare Logical Byte or Halfword (DD) (Part 2 of 2) (03755A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

The byte addressed by the 'from reg' is compared with the halfword in the 'to reg'. The 'from' byte and the 'to reg' remain unchanged. The result of the comparison sets the condition code latches if the CC bit (instruction bit 6) is on.

The comparison is done by subtracting the 'from operand' from the 'to operand'. The 'from' byte is extended to a halfword by high-order zeros.

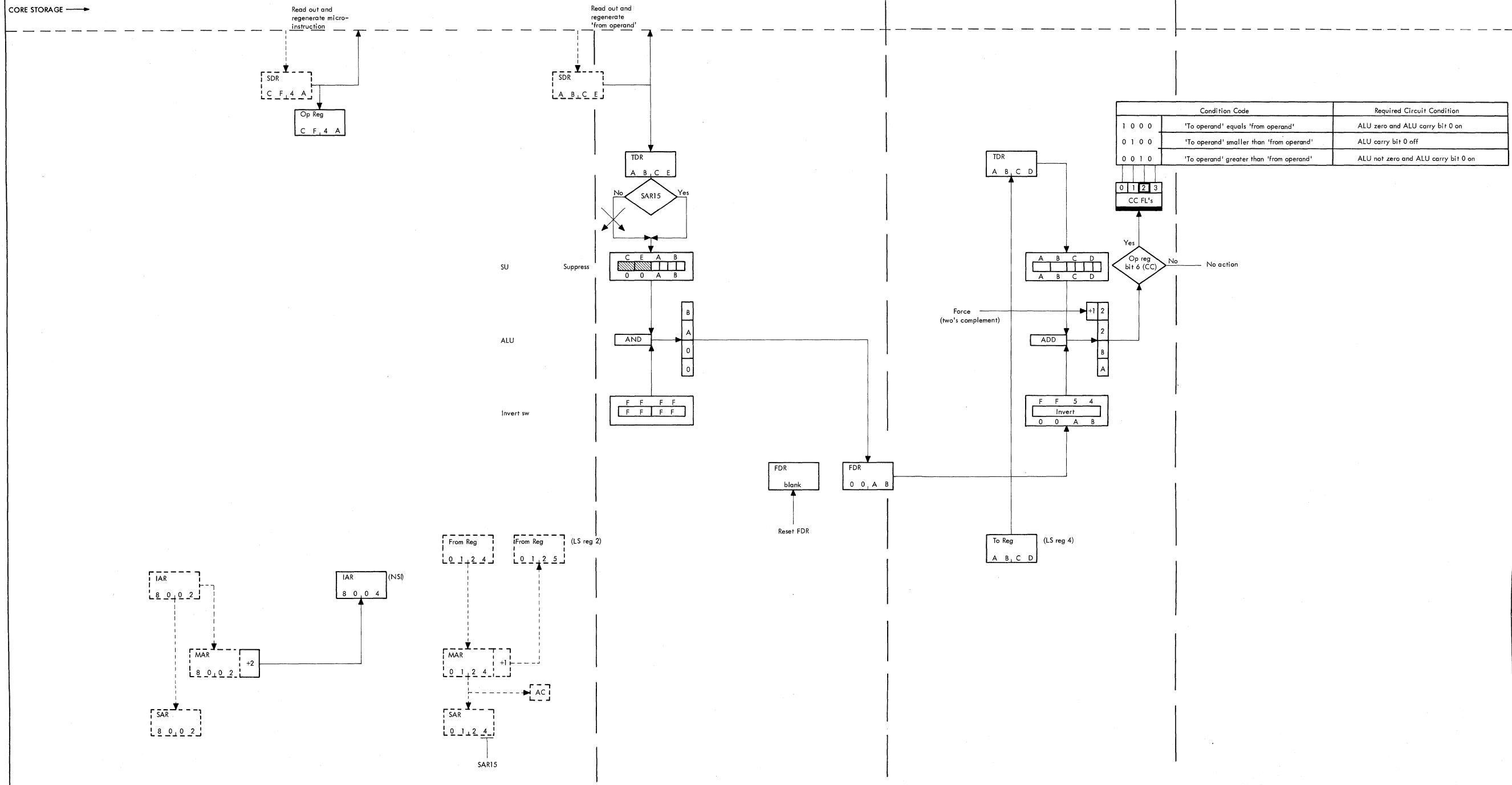
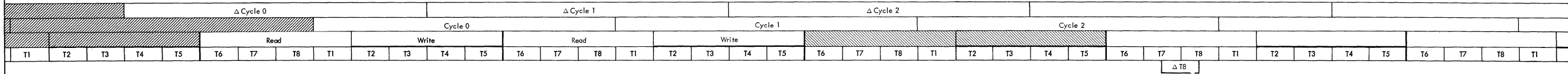
The 'from reg' address is incremented by 1.

If the AC bit (instruction bit 7) is on, an address check occurs when the 'from reg' address is outside customer area.



INST MNEM OPERANDS STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
 CF4A CLC 4,2I,CC,AC CC,R4,COMP,'00'/BY(R2,AC,+1)

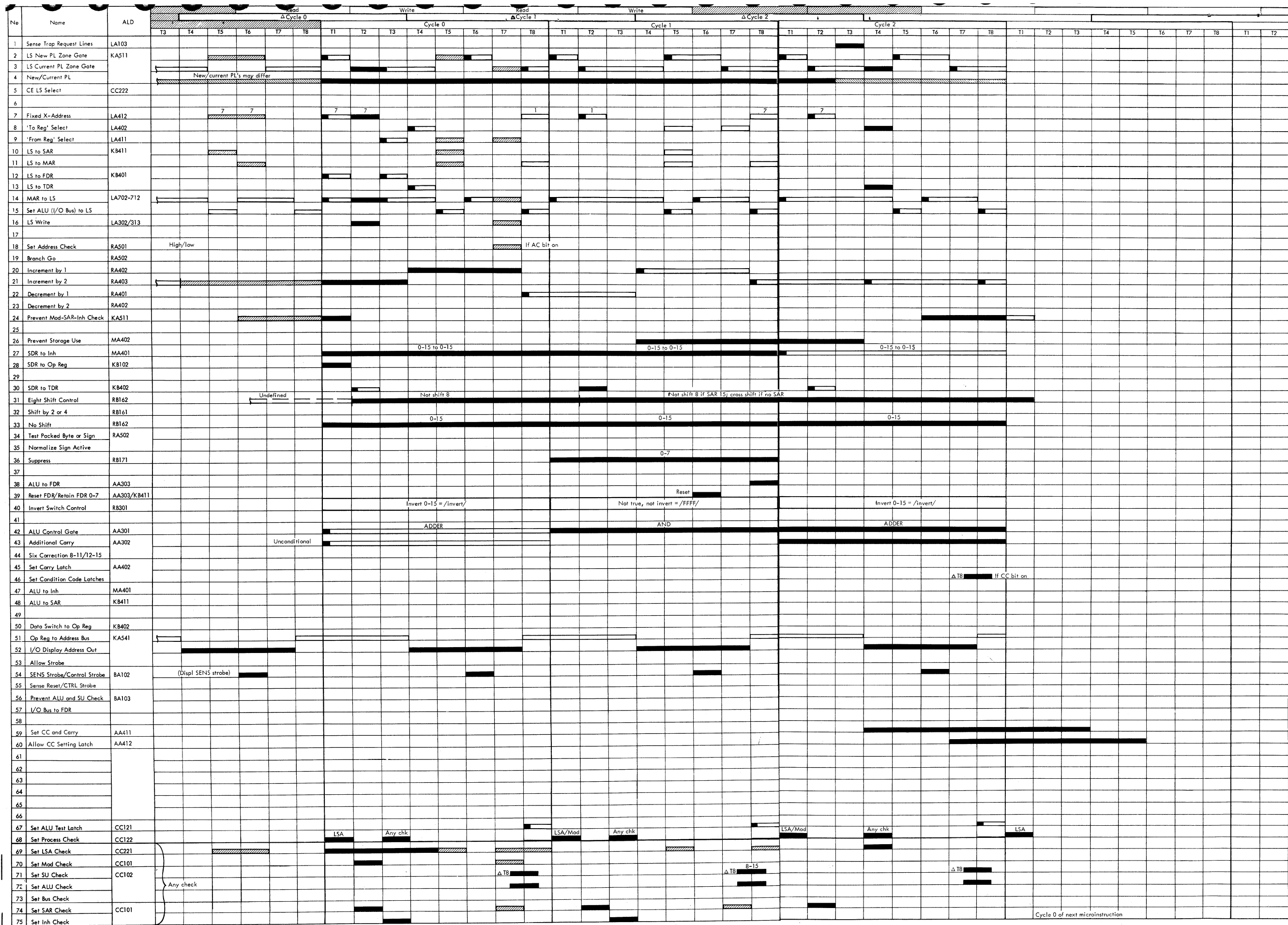
Mnemonic	CLC
Format	FF
Type	DX



Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

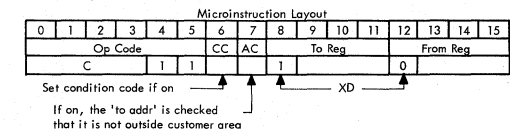
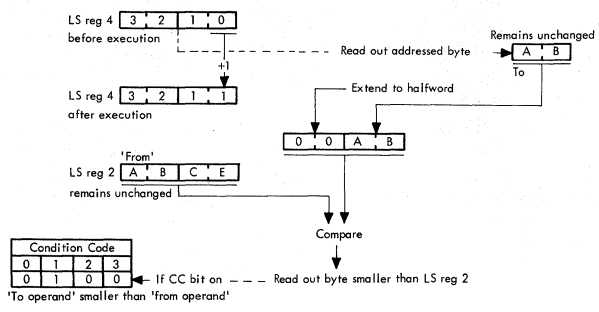
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



The halfword in the 'from reg' is compared with the byte addressed by the 'to reg'. The 'from reg' and the 'to' byte remain unchanged.

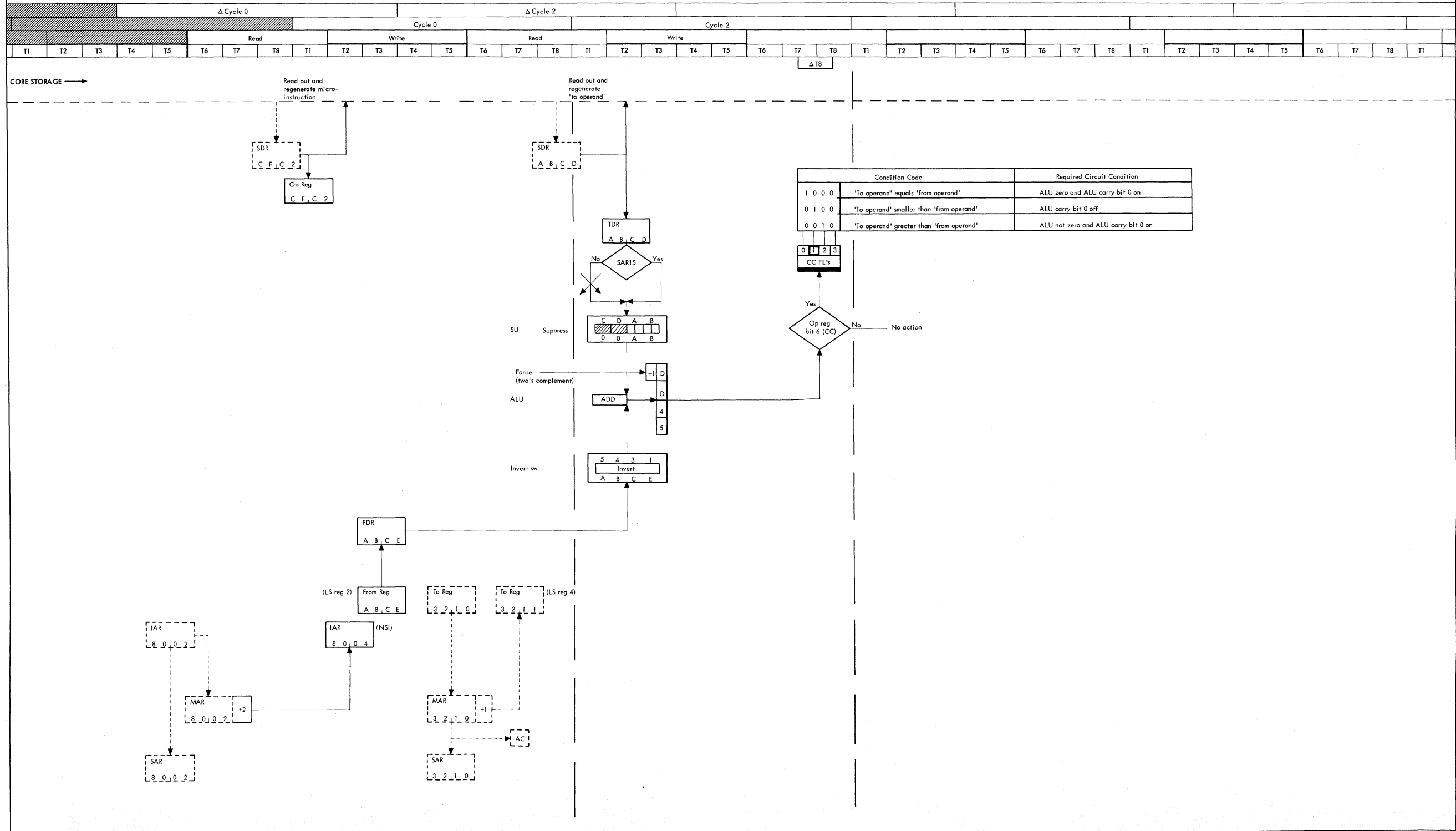
The 'to reg' address is incremented by 1. If the AC bit (instruction bit 7) is on, an address check occurs when the 'to reg' address is outside customer area.

The result of the comparison sets the condition code latches if the CC bit (instruction bit 6) is on. The comparison is done by subtracting the 'from operand' from the 'to operand'. The 'to' byte is extended to a halfword by high-order zeros.



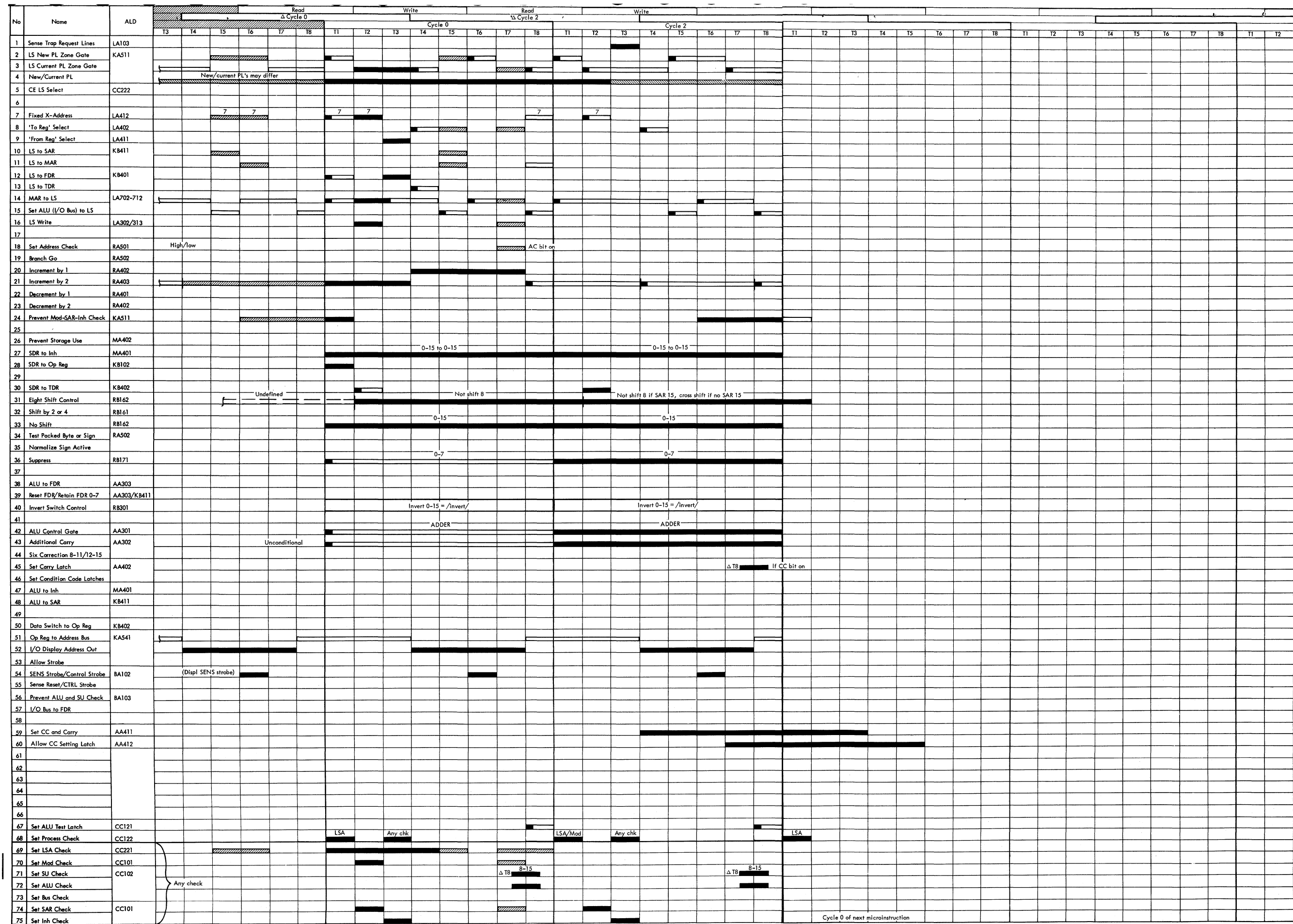
INST MLEM OPERANDS STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
 CFC2 CLC 4I,2,CC,AC CC,'00'/BY(R4,AC,+1),COMP,R2

Mnemonic	CLC
Format	
Type	FF
	XD

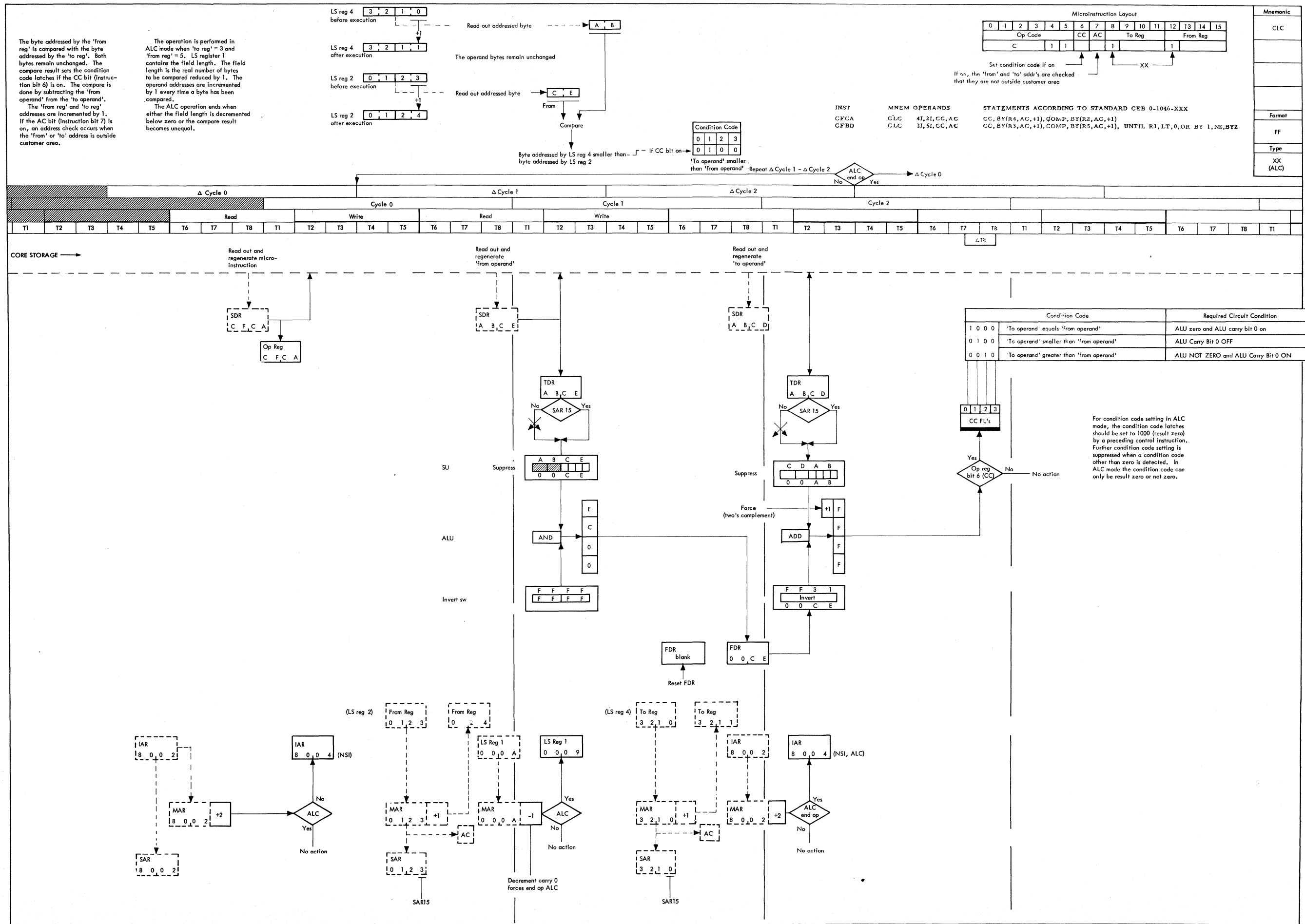


Note: For "Do not care" functions refer to timing chart below.
 "Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

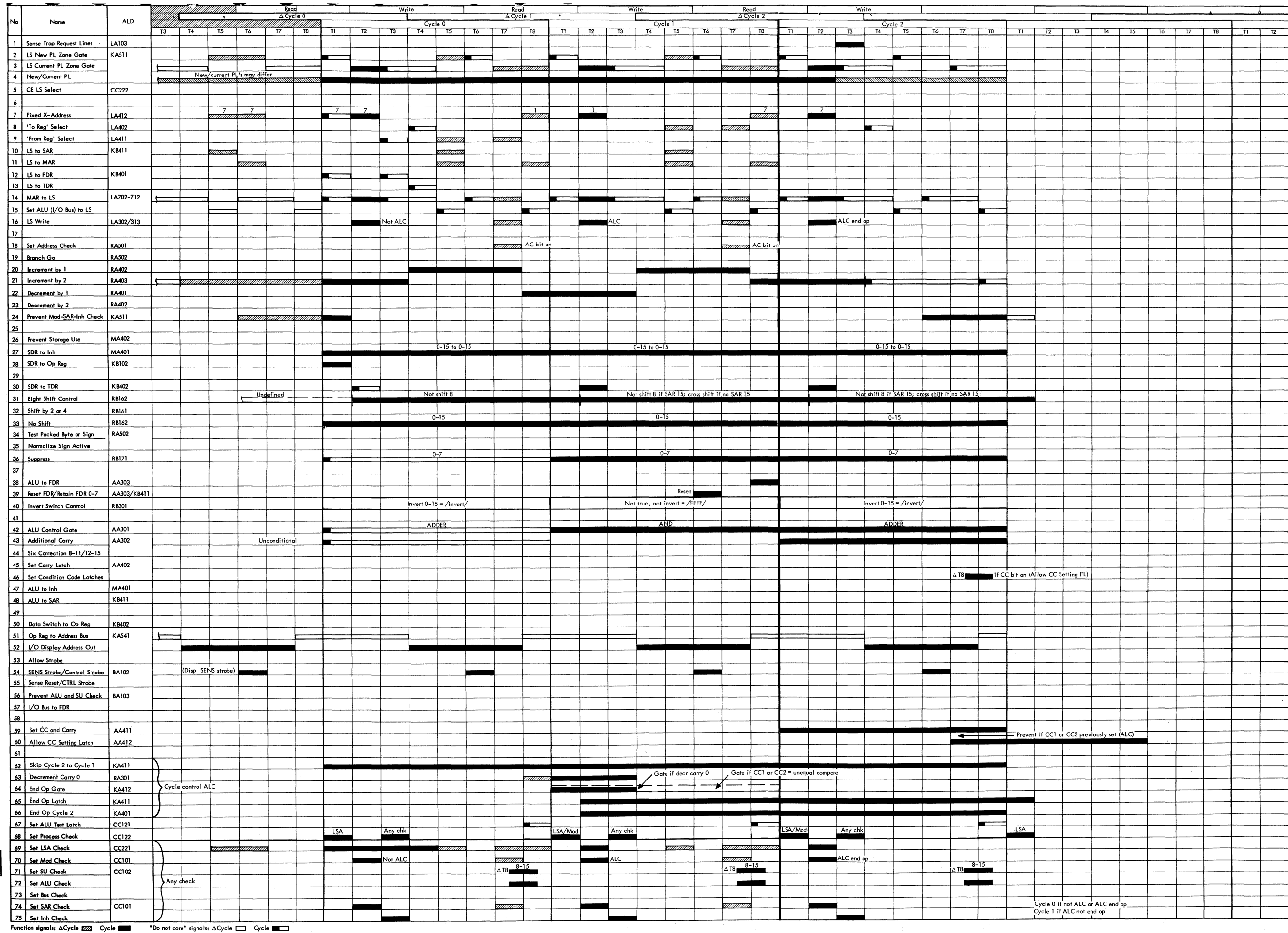


● Diagram 5-52. Compare Logical Byte or Halfword (XD) (Part 2 of 2) (03757A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)



Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle



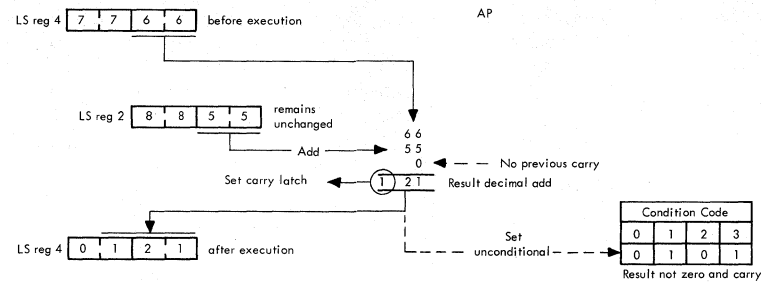
● Diagram 5-53. Compare Logical Byte or Halfword (XX, ALC) (Part 2 of 2) (03758A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

The two decimal digits in the low-order byte of the 'from reg' are added to the two decimal digits in the low-order byte of the 'to reg' (AP) or to zero (ZAP).

The result, which may consist of three decimal digits, is set into 'to reg'. The 'from reg' remains unchanged.

A previous carry (aux carry latch on) is implemented in the addition. A carry, out of the low-order byte, latches on the carry latch. The condition code latches are set.

A data check occurs when the low-order byte of the 'to reg' is not in packed decimal format. Address check is ignored.

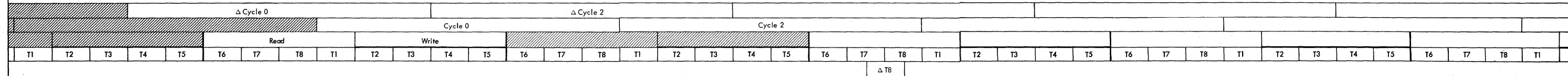


Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code							AC	To Reg							From Reg
D	0	0	0				0						0		
D	0	1	0												0

Ignored → DD →

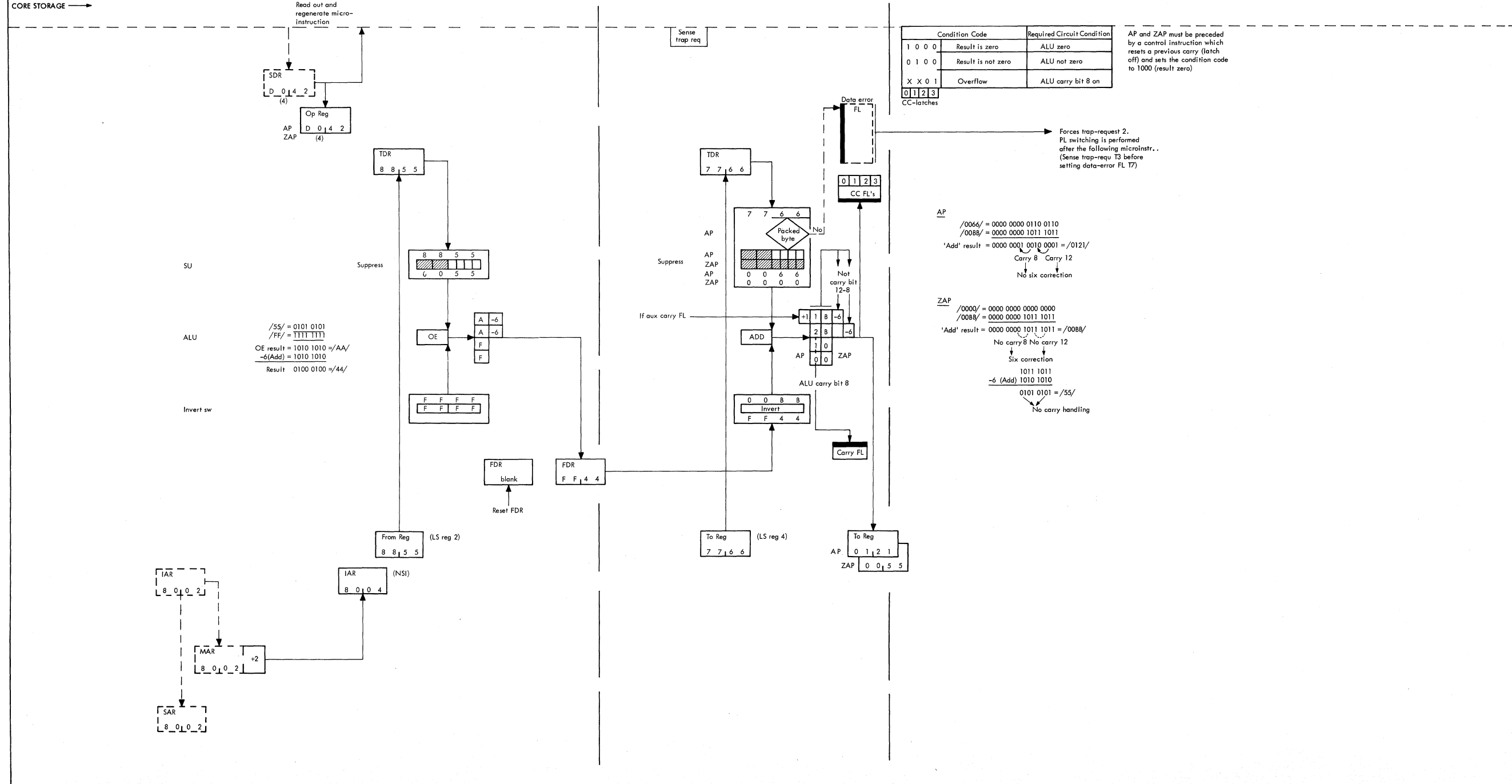
Mnemonic
AP
ZAP
DD

INST: D042, D442
 M'NEM: AP, ZAP
 OPERANDS: 4, 2
 STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
 CC, C/R4=100/R4, 8-15, D8, *00/R2, 8-15, D+, C
 CC, C/R4=100/R2, 8-15, D+, C

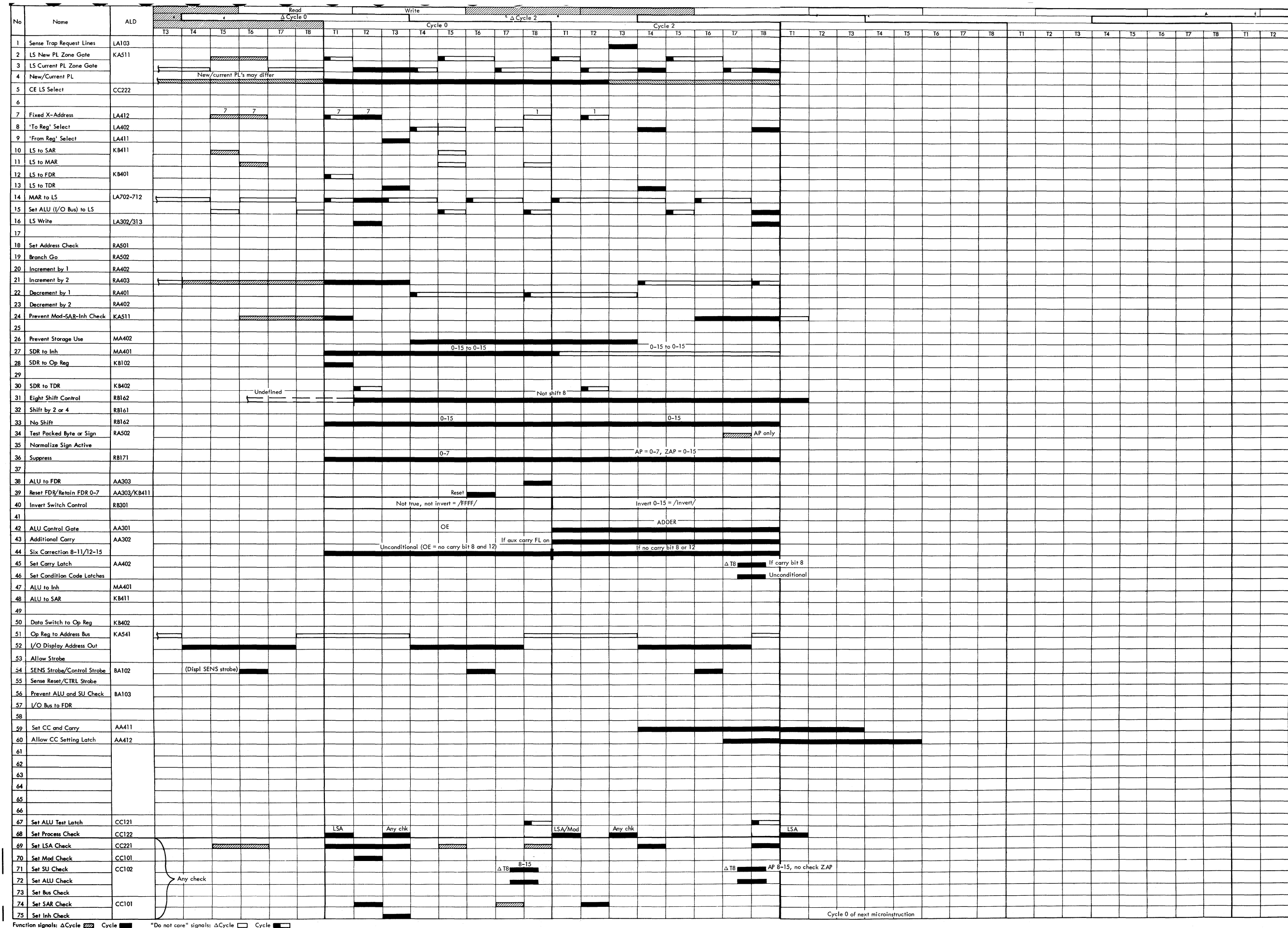


Note: For "Do not care" functions refer to timing chart below.

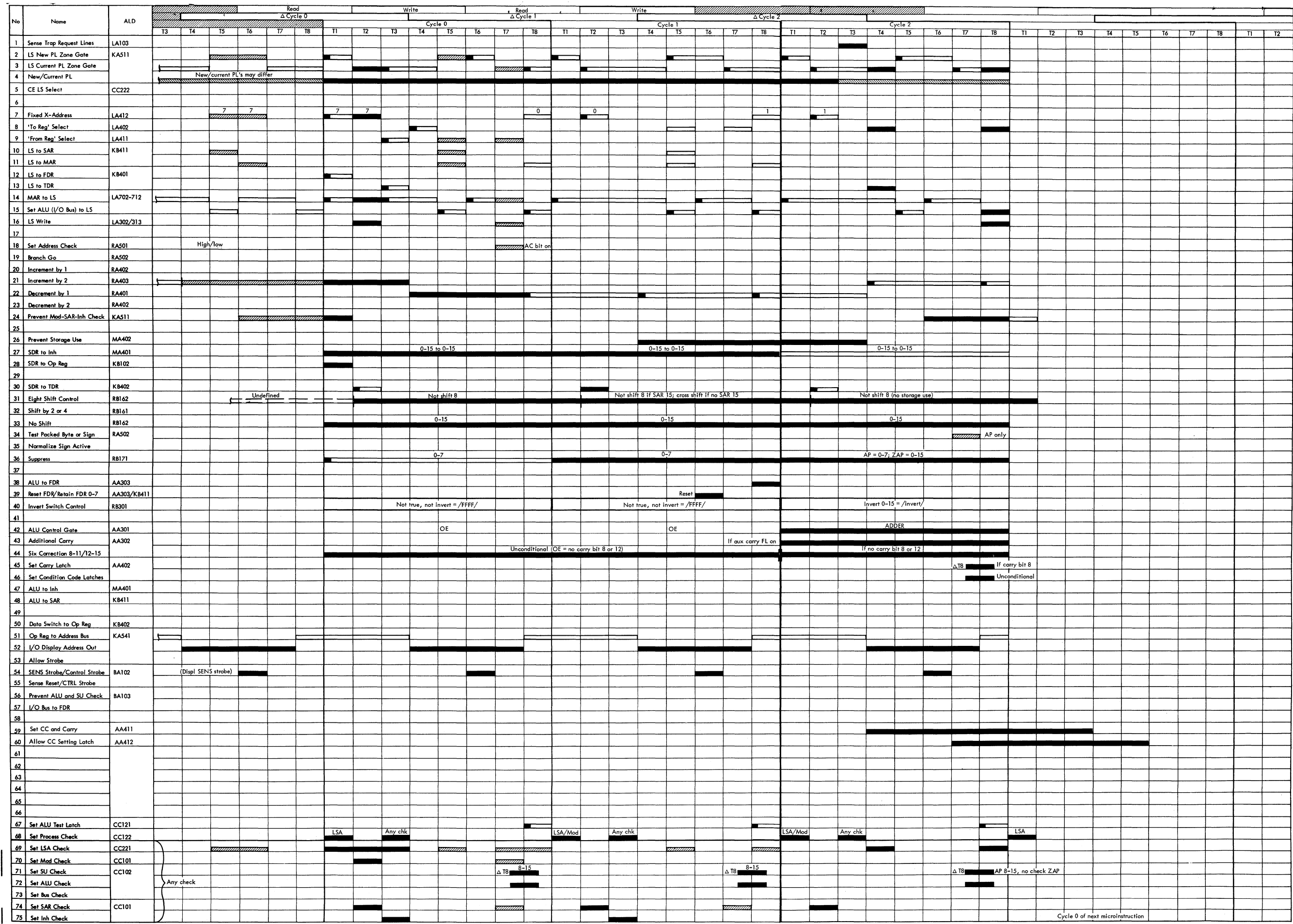
"Do not care" signals:
 ΔCycle (white box)
 Cycle (black box)



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



● Diagram 5-54. Add/Zero and Add Packed Byte (DD) (Part 2 of 2) (03759A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)



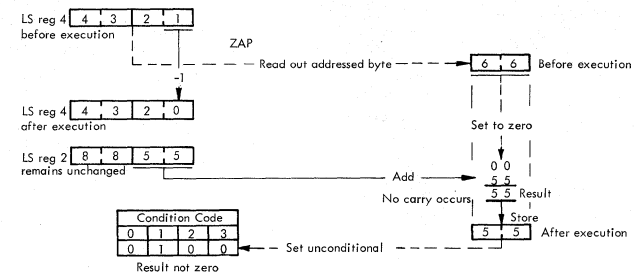
● Diagram 5-55. Add/Zero and Add Packed Byte (DX) (Part 2 of 2) (03760A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

The two decimal digits in the low-order byte of the 'from reg' are added to the two decimal digits in the byte addressed by the 'to reg' (AP) or to zero (ZAP). The result (low-order two decimal digits) is set into the byte addressed by the 'to reg'. The 'from reg' remains unchanged.

A previous carry (aux carry latch) is implemented in the addition. A carry out of the low-order byte turns on the carry latch. The condition code latches are set.

A data check occurs when the low-order byte of the 'from reg' or the byte addressed by the 'to reg' is not in packed decimal format.

The 'to reg' address is decremented by 1. If the AC bit (instruction bit 7) is on, an address check occurs when the 'to reg' address is outside customer area.



Microinstruction Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code							AC	To Reg				From Reg			
D	0	0	0	0	0	0	1					0			
D	0	1	0	0	0	0	1					0			

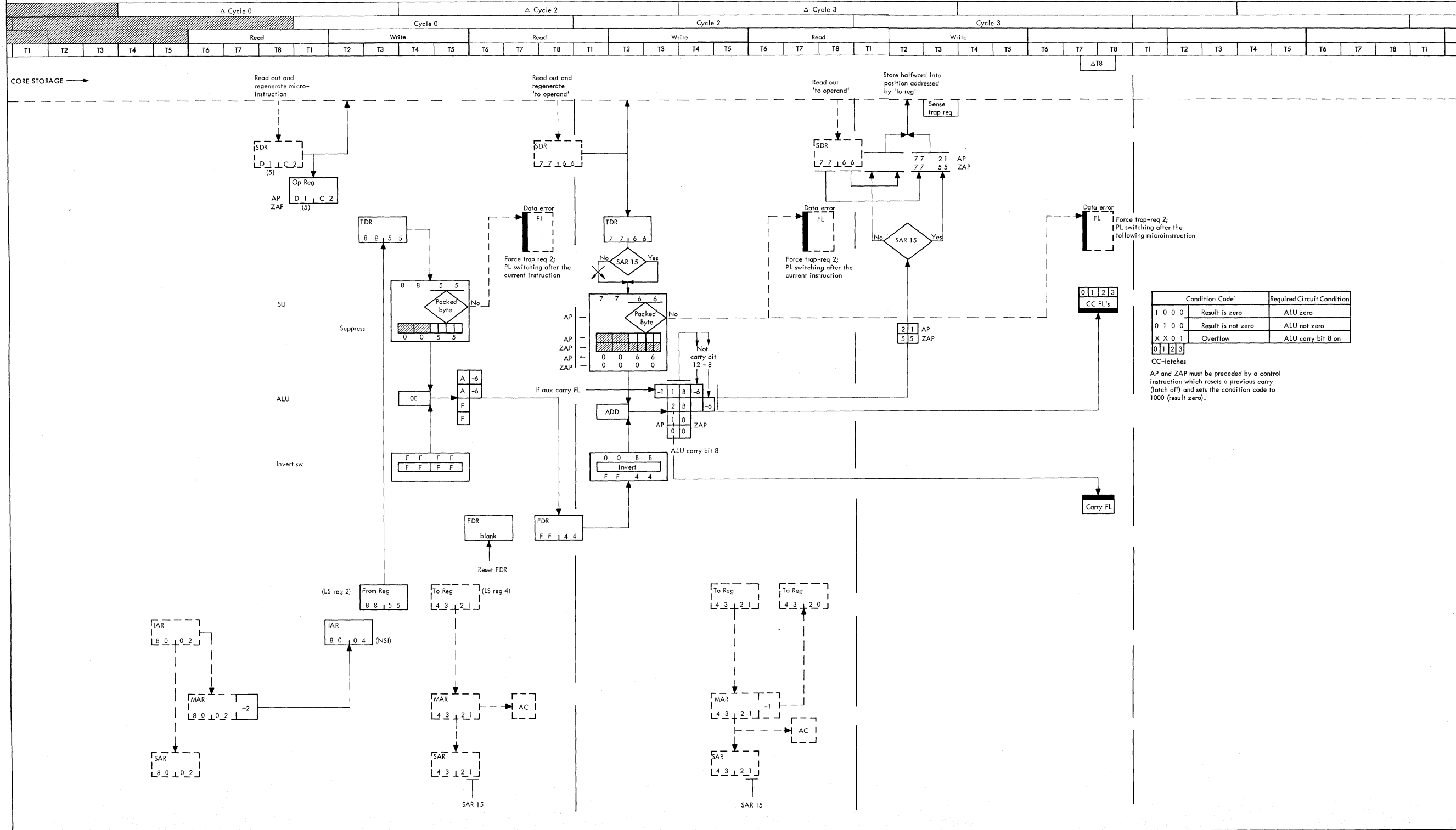
If on, the 'to addr' is checked that it is not outside customer area

INST	MNEM	OPERANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
D1C2	AP	41, 2, AC	CC, C/BY(R4, AC) = BY(R4, -1), D+, R2, 8-15, D+, C
D5C2	ZAP	41, 2, AC	CC, C/BY(R4, AC, -1) = R2, 8-15, D+, C

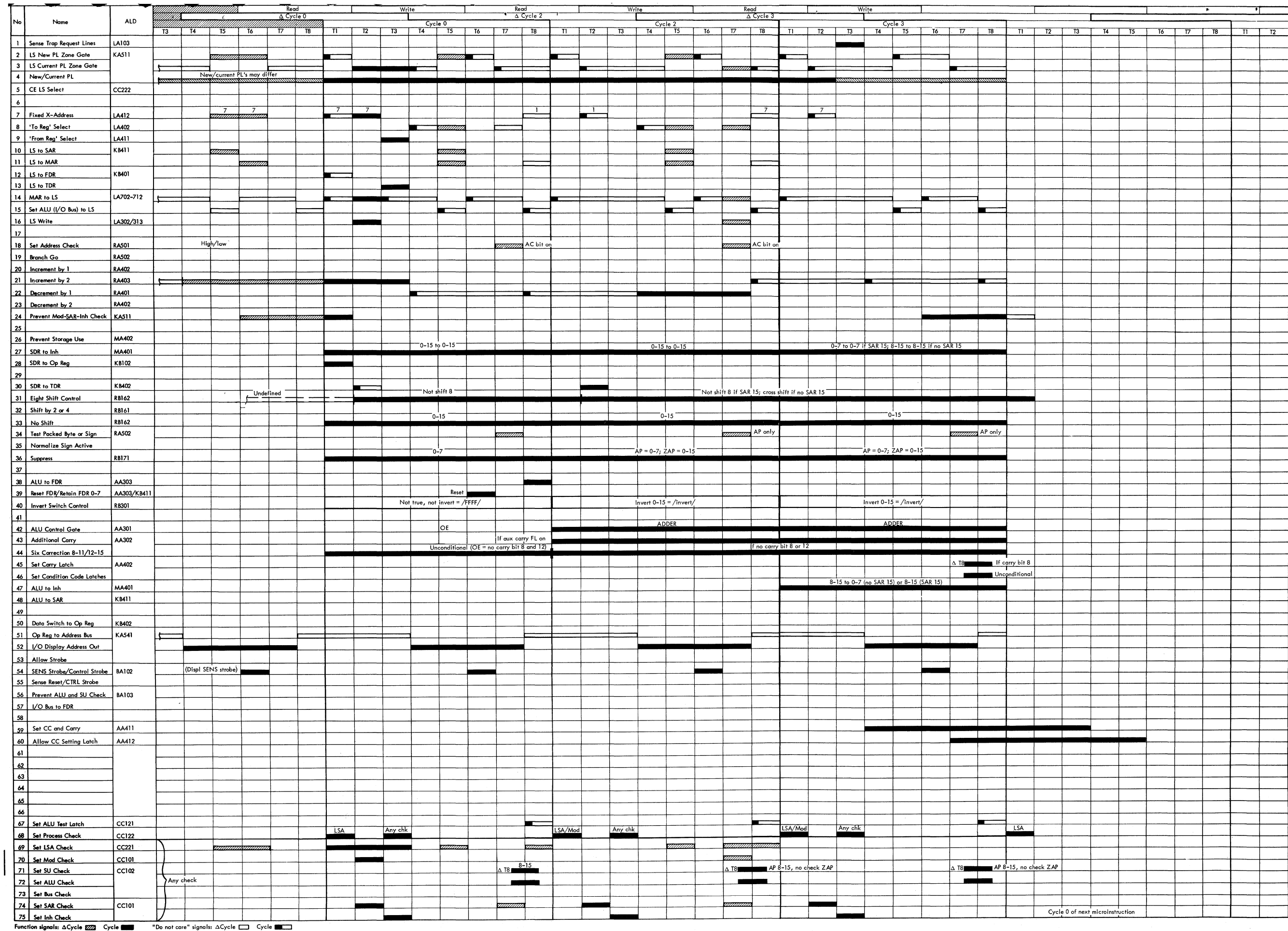
Mnemonic	AP
Format	ZAP
Type	FF
	XD

Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 Δ Cycle
 Cycle



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



● Diagram 5-56. Add/Zero and Add Packed Byte (XD) (Part 2 of 2) (03761A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

The two decimal digits in the byte addressed by the 'from reg' are added to the two decimal digits in the byte addressed by the 'to reg' (AP) or to zero (ZAP). The result (two decimal digits) is set into the byte addressed by the 'to reg'. The 'from' byte remains unchanged.

A previous carry (aux carry latch on) is implemented in the addition. A carry, out of the low-order byte, turns on the carry latch. The condition code latches are set. A data check occurs when the 'from' byte or the 'to' byte is not in packed decimal format.

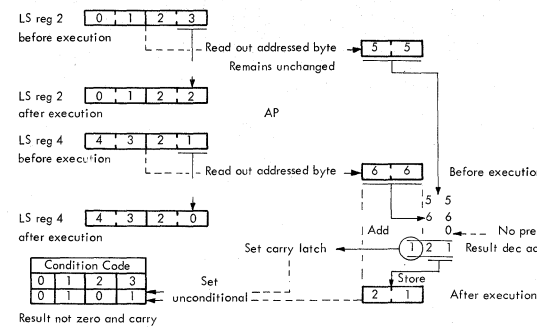
Both operand addresses are decremented by 1. If the AC bit (instruction bit 7) is on, an address check occurs when the operand addresses are outside customer area.

The operation is performed in ALC mode when 'to reg' = 3 and 'from reg' = 5. Each operand has its own field length and the field lengths may be different. LS register 0 contains the field length of the 'from operand'. LS register 1 contains the field length of the 'to operand'.

The field length is the real number of bytes in an operand field reduced by 1.

The operand addresses are incremented by 1 every time a byte is operated.

If the 'to operand' field is too small to accept all digits of the 'from operand' field the condition code is set to overflow (CC latch 3 on).



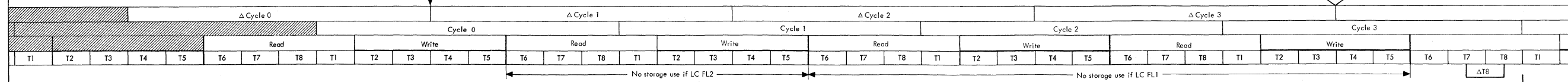
Microinstruction Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Op Code							AC	To Reg				From Reg				
D	0	0	0	0	0	0	1					1				
D	0	1	0													

If on, the 'from' and 'to' addr's are checked that they are not outside customer area

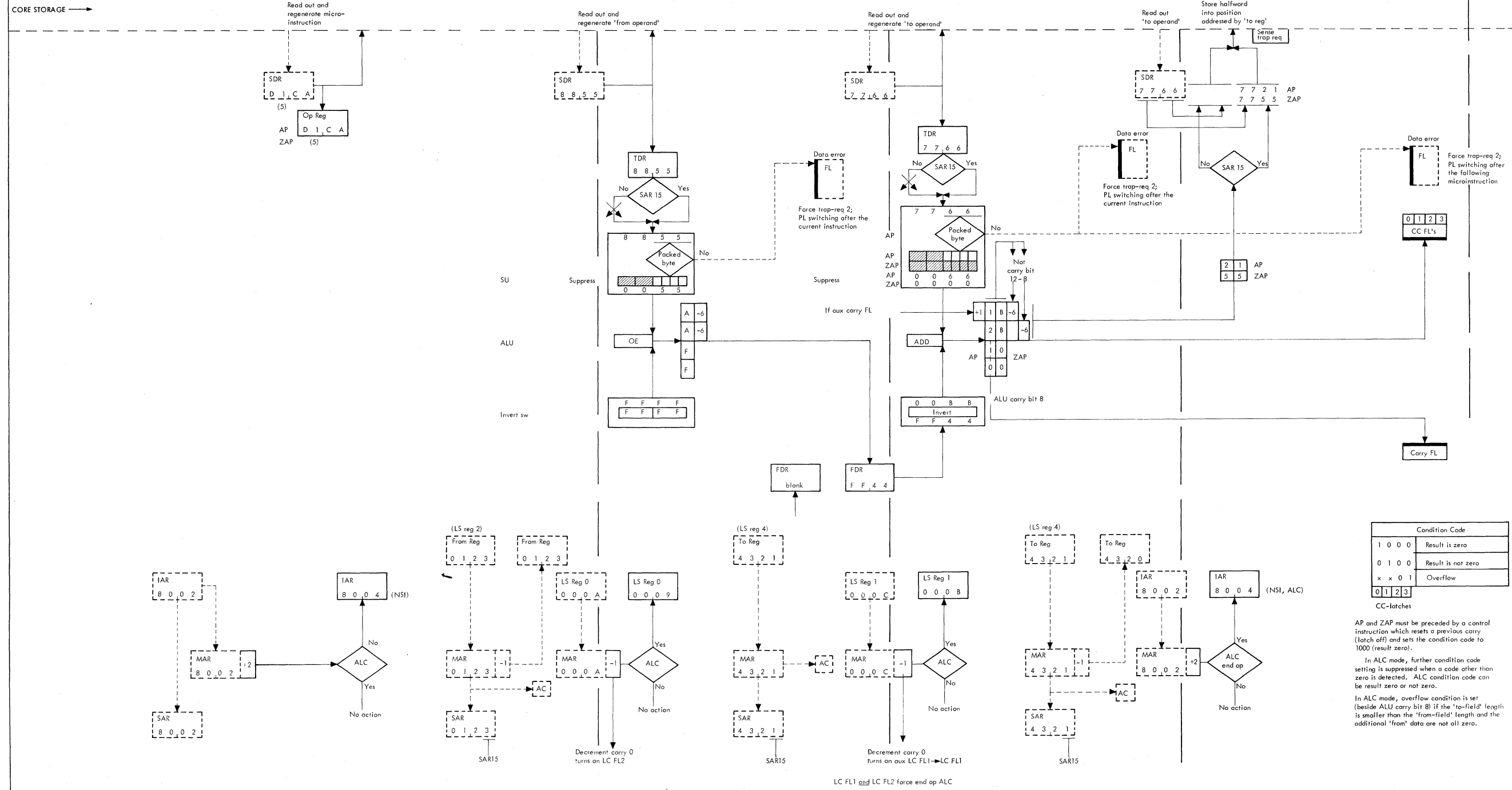
INST	MNEM	OPERANDS	STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
D1CA	AP	4I, 2I, AC	CC, C/BY(R4, AC) += BY(R4, -1), D+, BY(R2, AC, -1), D+, C
D5CA	ZAP	4I, 2I, AC	CC, C/BY(R4, AC, -1) += BY(R2, AC, -1), D+, C
D1BD	AP	3I, 5I, AC	CC, C/BY(R3, AC, UNTIL R1, LT, 0) += BY(R3, -1), D+, BY(R5, AC, -1), UNTIL R0, LT, 0), D+, C

AP	
ZAP	
Format	
FF	
Type	
XX	(ALC)

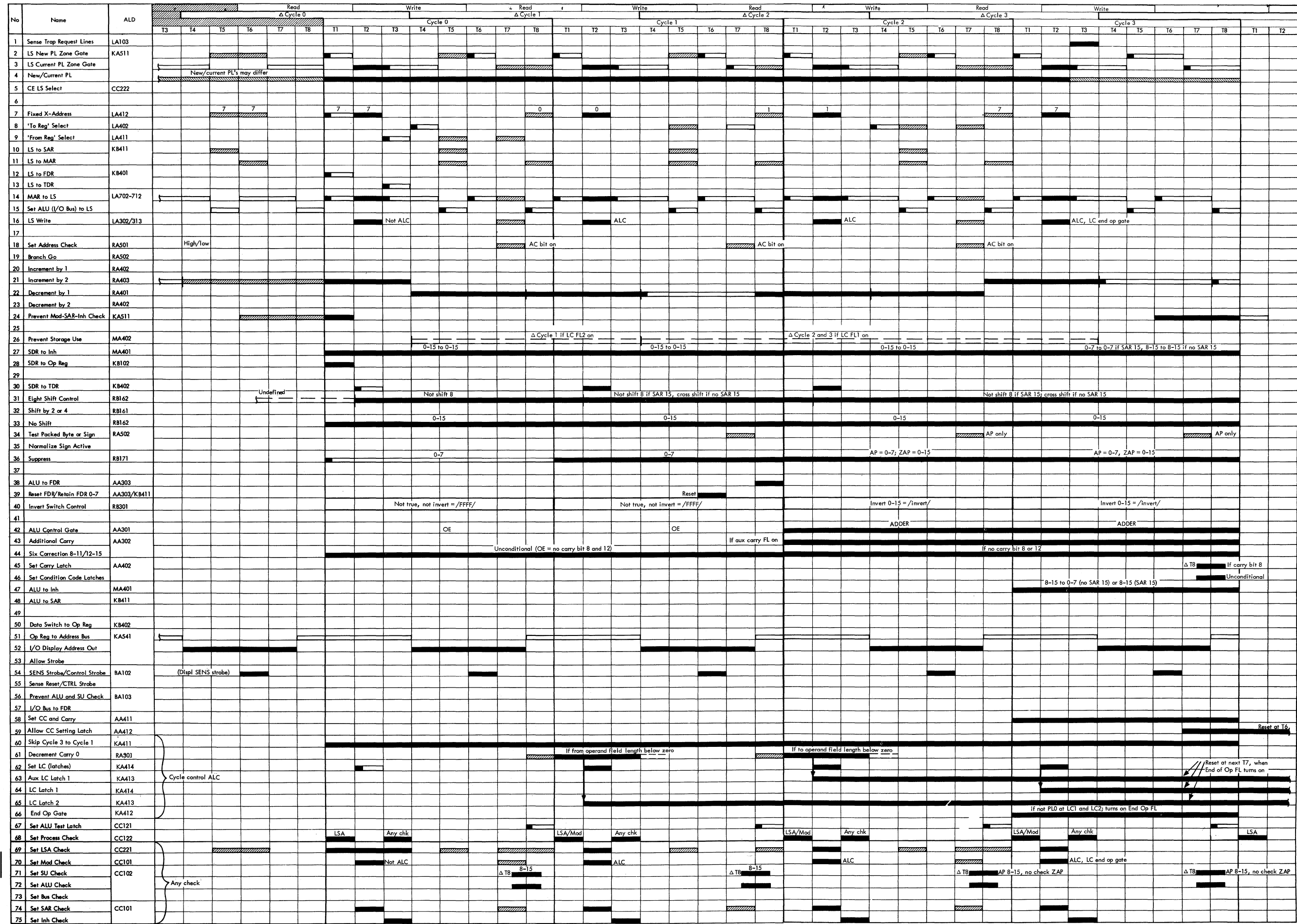


Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle



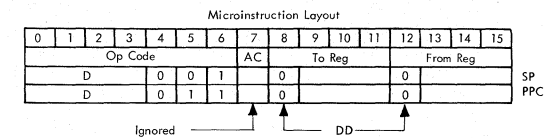
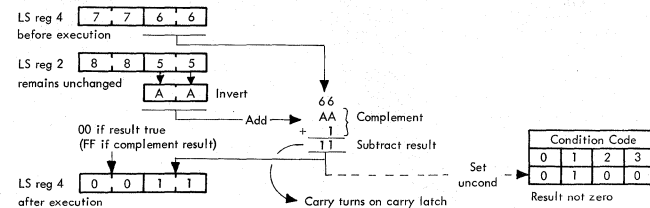
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



Function signals: Δ Cycle Cycle "Do not care" signals: Δ Cycle Cycle

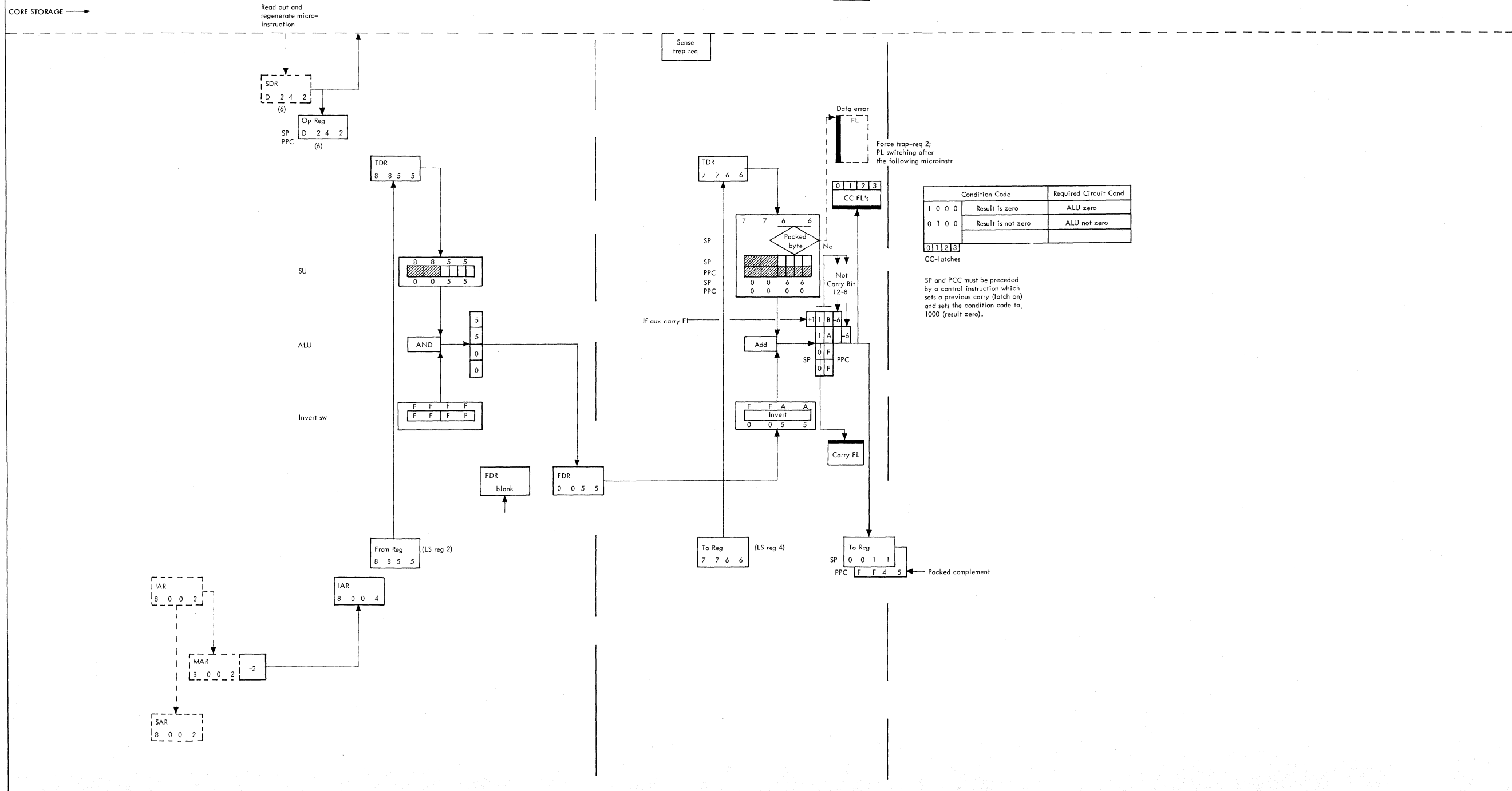
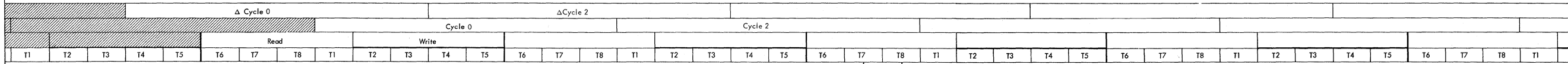
For SP, the two decimal digits in the low-order byte of the 'from reg' are subtracted from the two decimal digits in the low-order byte of the 'to reg'. The result is set into 'to reg'.
 For PPC, the tens complement of the two decimal digits in the low-order byte of the 'from reg' is set into 'to reg'. The 'from reg' remains unchanged.
 A previous carry (aux carry latch on) is implemented in the operation.

Before initiating SP or PPC, the previous carry must be simulated by turning on the carry latch by a control /10, bit 11 to obtain the correct complement.
 A carry, out of the low-order byte, turns on the carry latch. The condition code latches are set. A data check occurs when the low-order byte of the 'to reg' is not in packed decimal format. Address check is ignored.



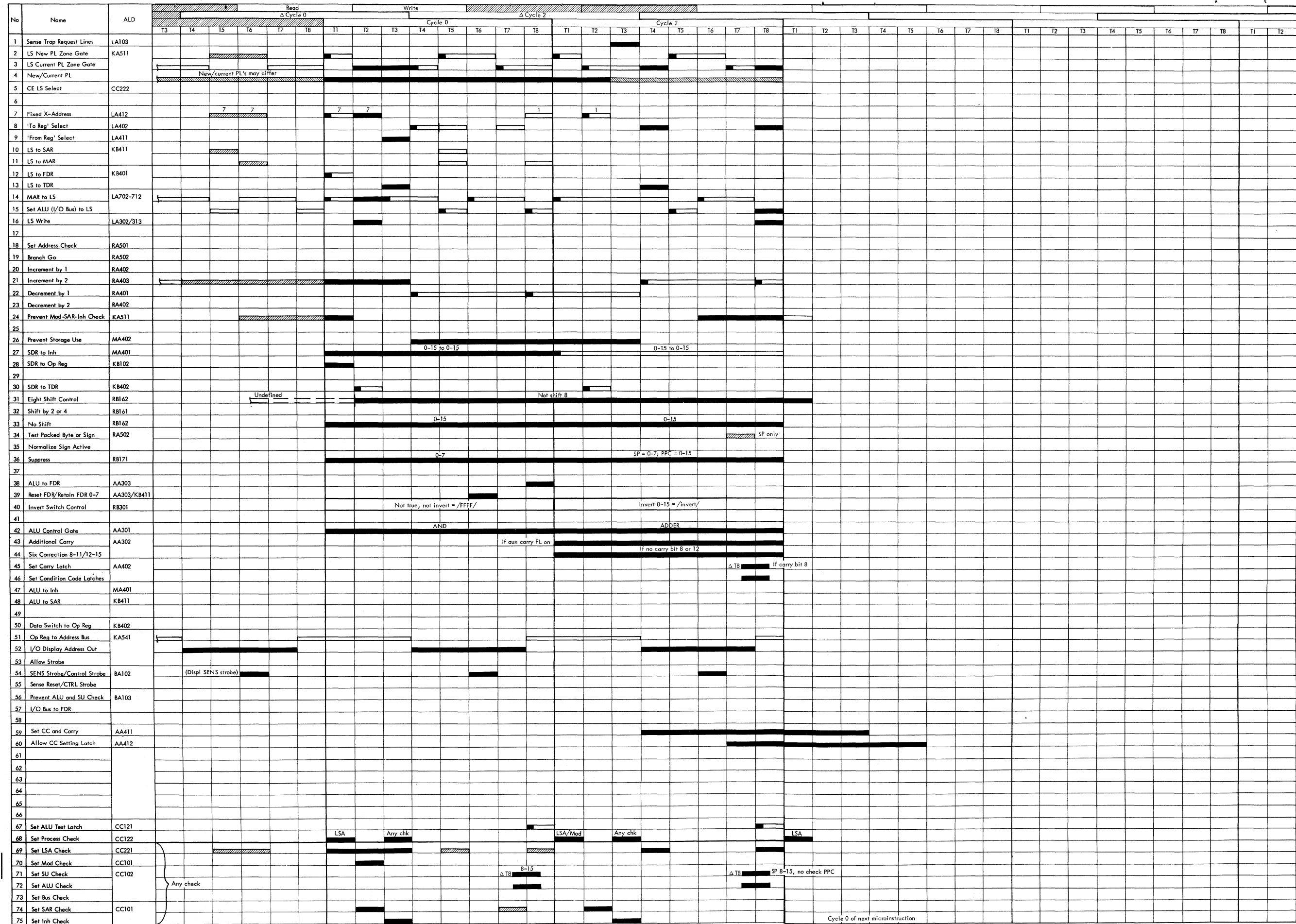
Mnemonic
SP
PPC
Format
FF
Type
DD

INST D242 D642
 MNEM SP PPC
 OPERANDS 4,2 4,2
 STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
 CC, C/R4+= '00'/R4, 8-15, D-, '00'/R2, 8-15, D-, C
 CC, C/R4+= 'FF'/X, X = '99', D-, R2, 8-15, D+, C



Note: For "Do not care" functions refer to timing chart below.
 "Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



● Diagram 5-58. Subtract Packed Byte/Perform Packed Complement (DD) (Part 2 of 2) (03763A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

For SP, the two decimal digits in the byte addressed by the 'from reg' are subtracted from the two decimal digits in the low-order byte of the 'to reg'. The result is set into 'to reg'. For PPC, the tens complement of the two decimal digits in the byte addressed by the 'from reg' is set into 'to reg'. The 'from' byte remains unchanged.

A previous carry (aux carry latch on) is implemented in the operation.

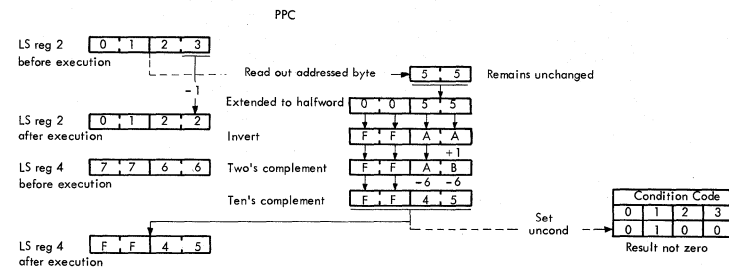
Before initiating SP or PPC, the previous carry must be simulated by turning on the carry latch by a control/10, bit 11, to obtain the correct complement.

A carry, out of the low-order byte, turns on the carry latch. The condition code latches are set.

A data check occurs when the low-order byte of the 'to reg' is not in packed decimal format.

The 'from reg' address is decremented by 1.

If the AC bit (instruction bit 7) is on, an address check occurs when the 'from address' is outside customer area.



Microinstruction Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Op Code							AC	To Reg							From Reg		
D	0	0	1	0							1						
D	0	1	1	0							1						

If on, the 'from addr' is checked that it is not outside customer area

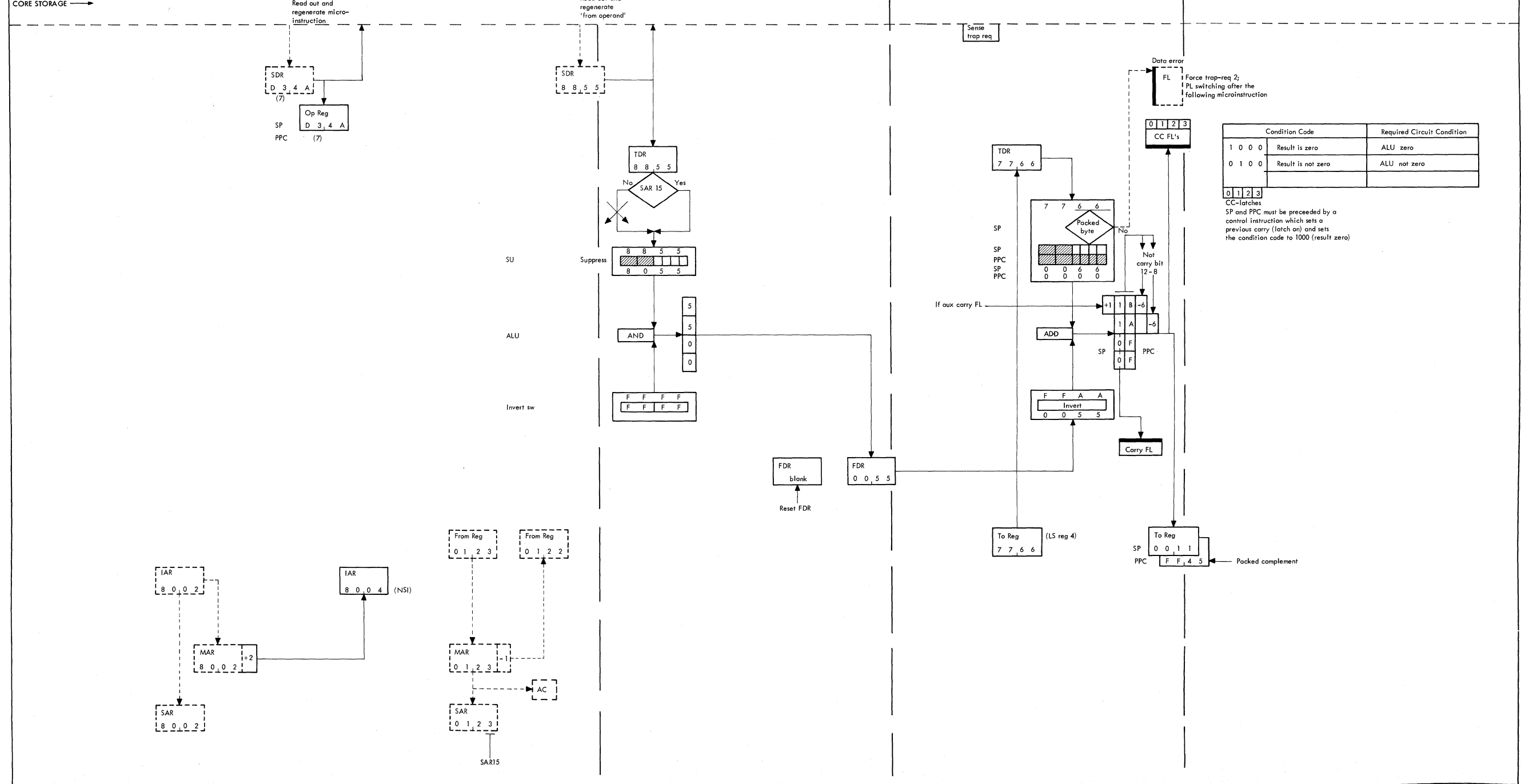
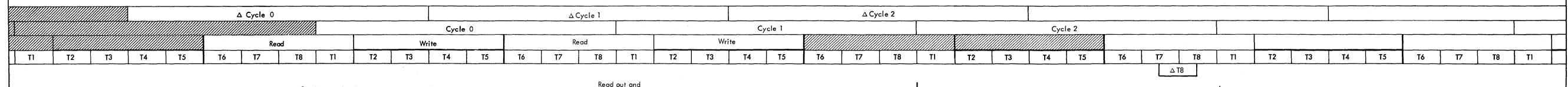
INST: D34A, D74A

MNEM: SP, PPC

OPERANDS: 4, 2I, AC; 4, 2I, AC

STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX: CC, C/R4= '00'/R4. 8-15, D-, '00'/BY(R2, AC, -1), D-, C; CC, C/R4= 'FF'/X, X = '99', D-, BY(R2, AC, -1), D+, C

Mnemonic	SP
Format	PPC
Type	FF
DX	DX



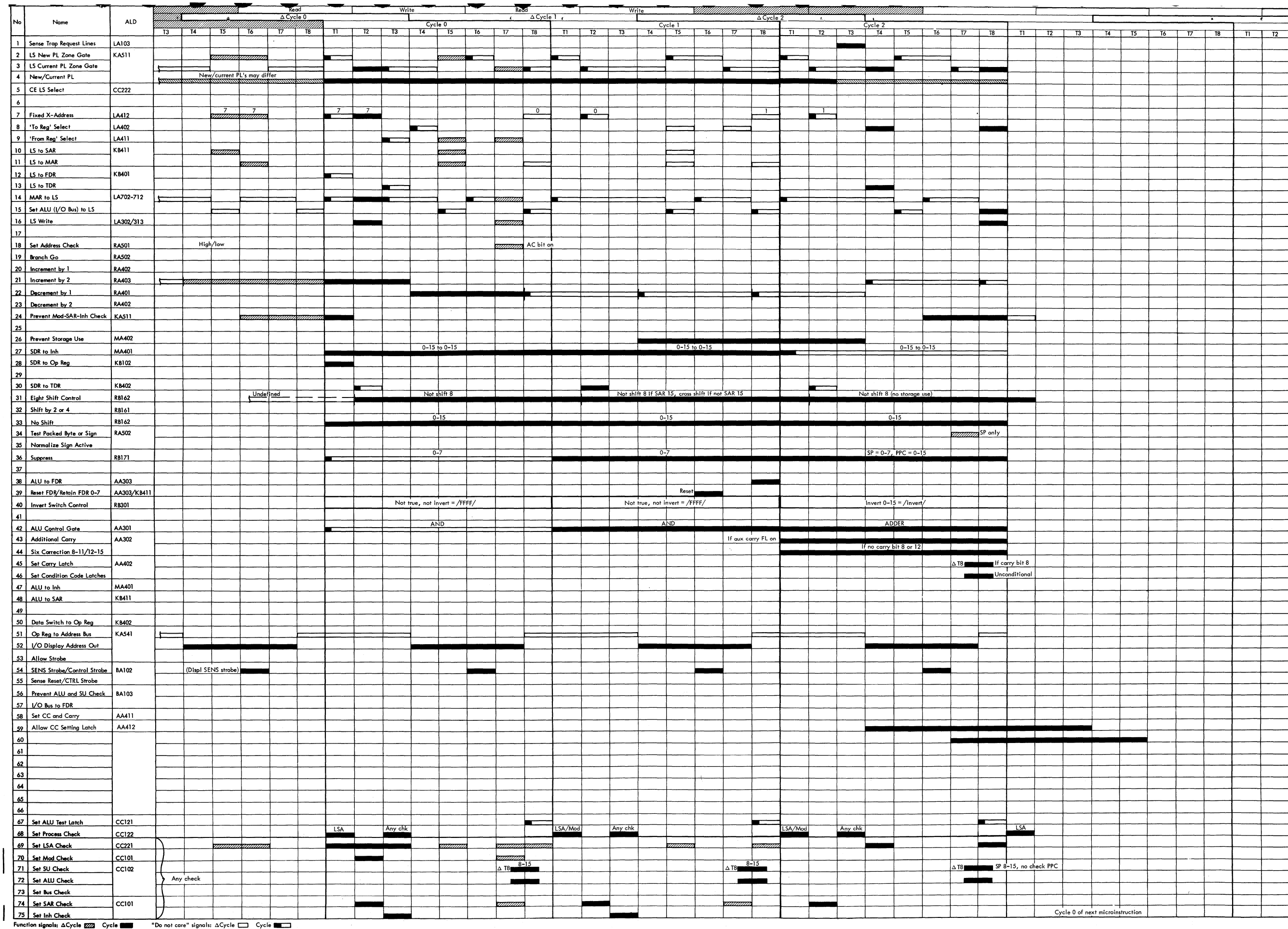
Condition Code	Required Circuit Condition
1 0 0 0	Result is zero
0 1 0 0	Result is not zero

CC-latches
SP and PPC must be preceded by a control instruction which sets a previous carry (latch on) and sets the condition code to 1000 (result zero)

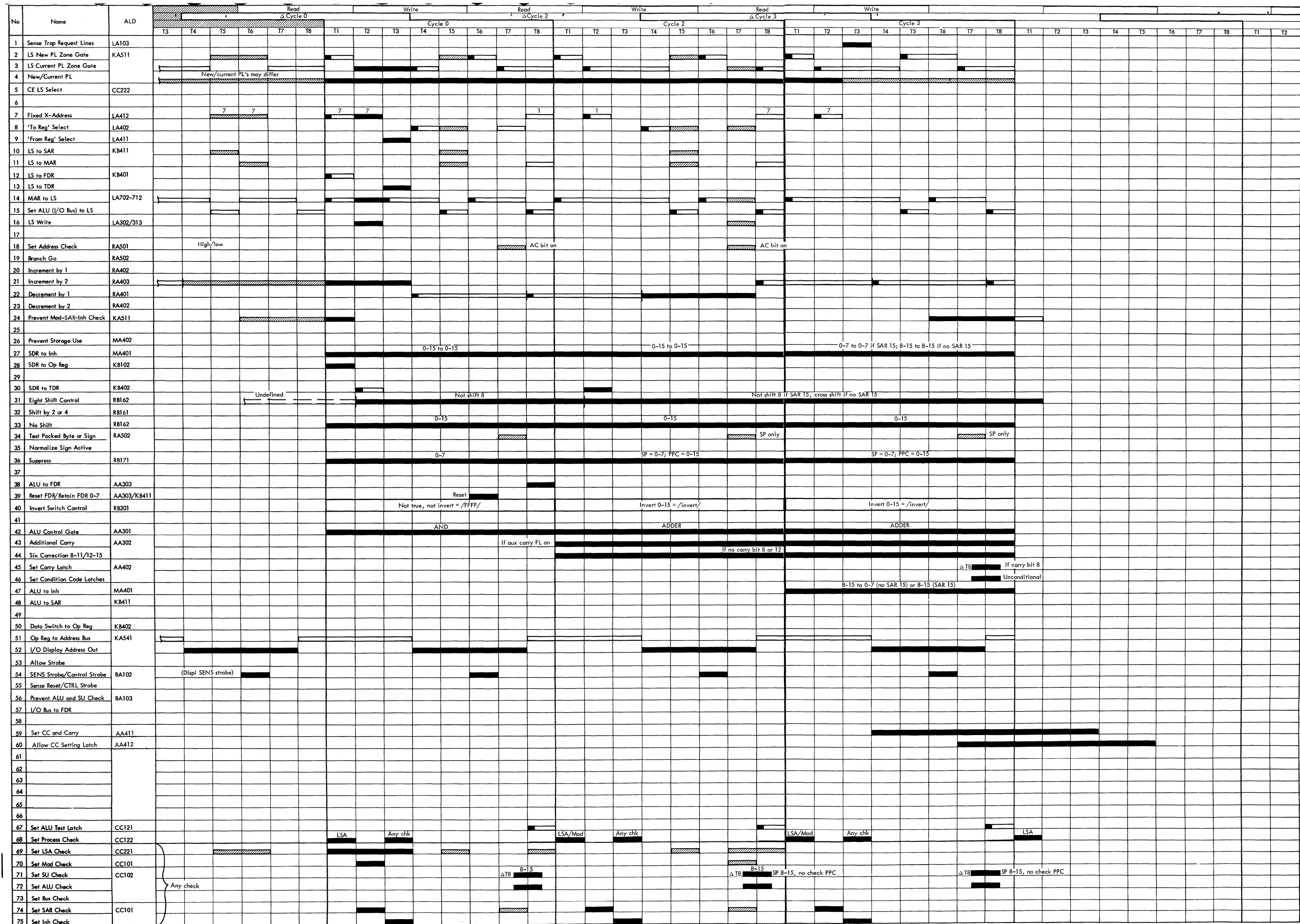
Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



● Diagram 5-59. Subtract Packed Byte/Perform Packed Complement (DX) (Part 2 of 2) (03764A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)



Function signals: Δ Cycle Cycle "Do not care" signals: Δ Cycle Cycle

For SP, the two decimal digits in the byte address by the 'from reg' are subtracted from the two decimal digits in the byte addressed by the 'to reg'. The result is set into the byte addressed by the 'to reg'.

For PPC, the complement of the two decimal digits in the byte addressed by the 'from reg' is set into the byte addressed by the 'to reg'.

The 'from' byte remains unchanged.

A previous carry (aux carry latch on) is implemented in the operation.

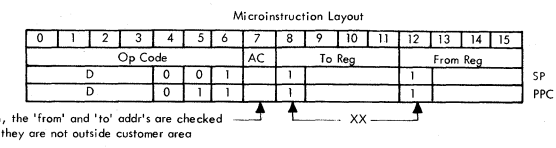
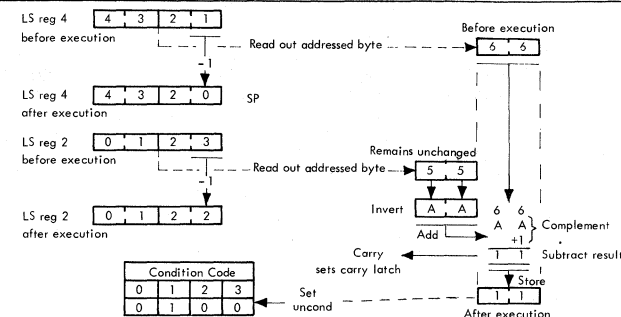
A carry out of the low-order byte turns on the carry latch. A data check occurs when one of the operand bytes is not in packed decimal format.

The 'from' and 'to' addresses are decremented by 1. If the AC bit (instruction bit 7) is on, an address check occurs when the 'from' or 'to' address is outside customer area.

The operation is performed in ALC mode when 'to reg' = 3 and 'from reg' = 5. Each operand has its own field length. The field length may be different. LS register 0 contains the field length of the 'from operand'. LS register 1 contains the field length of the 'to operand'.

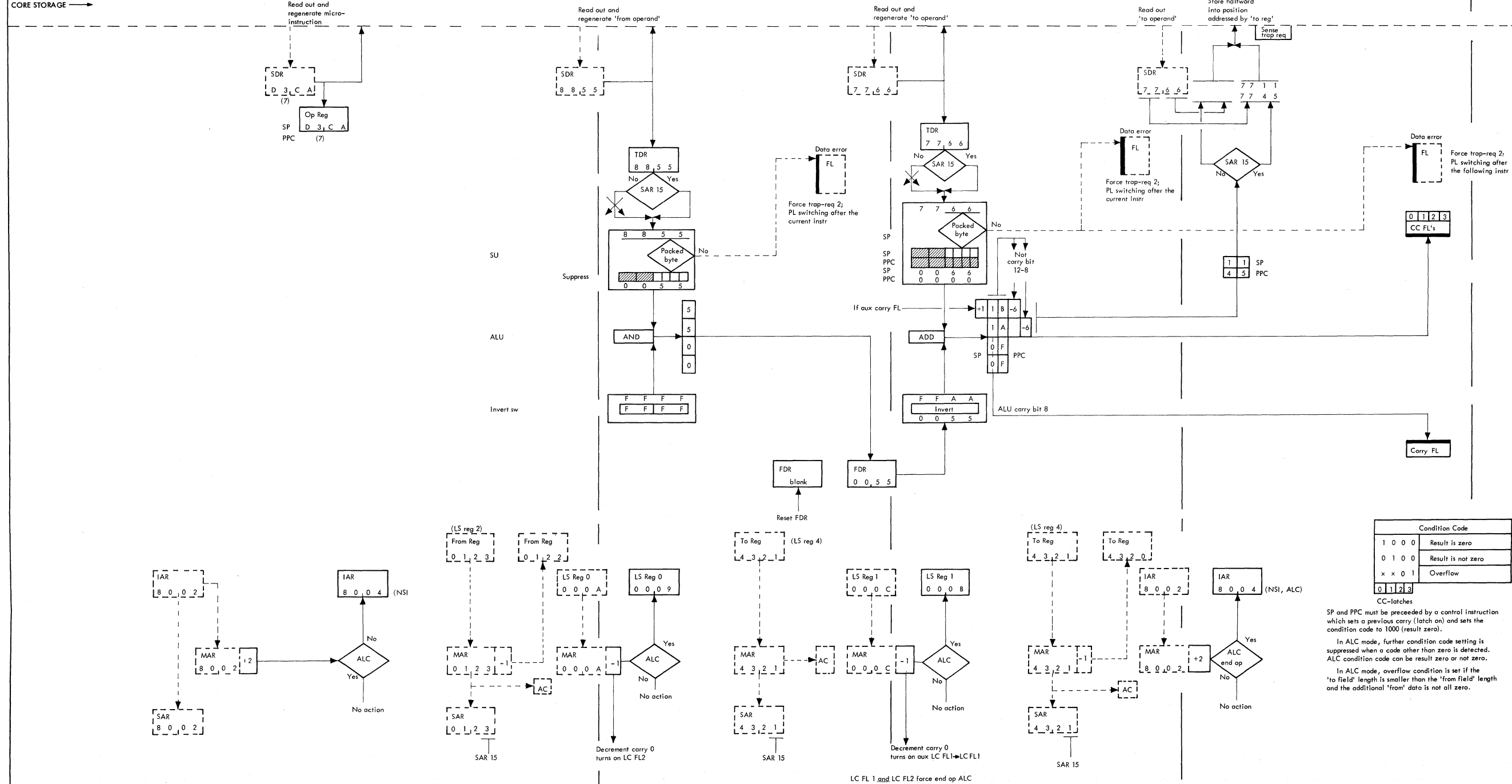
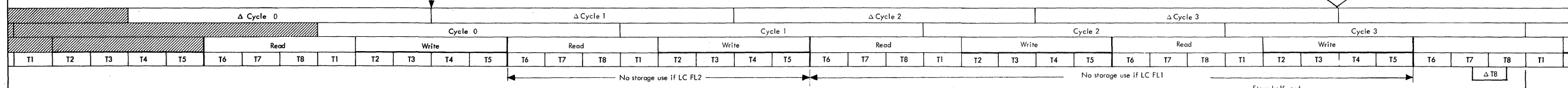
The field length is the real number of bytes in an operand field reduced by 1.

The operand addresses are decremented by 1 every time one byte has been operated.



INST MNEM OPERANDS STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
 D3CA SP 41,21,AC CG,C/BY(R4,AC)+=RY(R4,-1),D-,RY(R2,AC,-1),D-,C
 D7CA PPC 41,21,AC CG,C/BY(R4,AC,-1)+=RY(R4,AC,-1),D+,C
 D3BD SP 31,51,AC CC,C/BY(R3,AC,UNTIL R1,LT,0)+=BY(R3,-1),D-,BY(R5,AC,-1),UNTIL R0,LT,0),D-,C

Mnemonic	SP
Format	PPC
Type	XX (ALC)



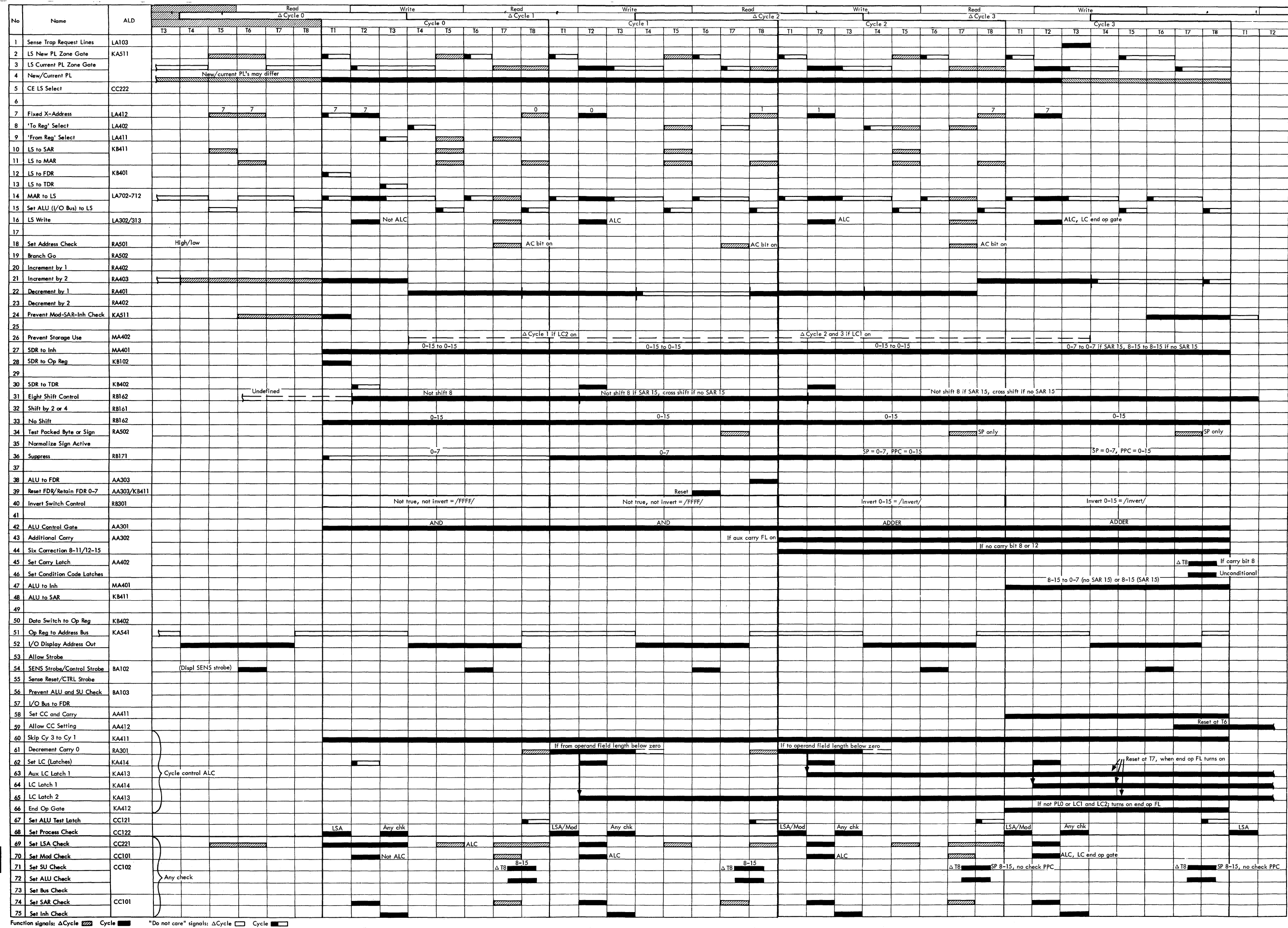
Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

Condition Code			
1 0 0 0	Result is zero		
0 1 0 0	Result is not zero		
x x 0 1	Overflow		
0 1 2 3	CC-latches		

SP and PPC must be preceded by a control instruction which sets a previous carry (latch on) and sets the condition code to 1000 (result zero).
 In ALC mode, further condition code setting is suppressed when a code other than zero is detected. ALC condition code can be result zero or not zero.
 In ALC mode, overflow condition is set if the 'to field' length is smaller than the 'from field' length and the additional 'from' data is not all zero.

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



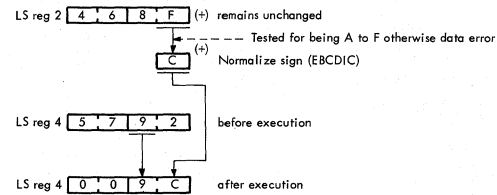
● Diagram 5-61. Subtract Packed Byte/Perform Packed Complement (XX, ALC) (Part 2 of 2) (03766A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

The packed decimal sign (low-order four bits) in the 'from reg' is translated into the standard sign according to the selected code. The code is selected by the USASCII latch (on = USASCII, off = EBCDIC). The USASCII latch can be turned on by a control/11, bit 14, not bit 12; and it can be turned off by a control/11, not bit 14, not bit 12.

The standardized sign is set into the low-order four bit position of the 'to reg'. The

high-order four bit position of the low-order byte in the 'to reg' and the 'from reg' remain unchanged. The high-order byte of the 'to reg' is set to zero. 'From reg' and 'to reg' may be the same LS register. Address check is ignored.

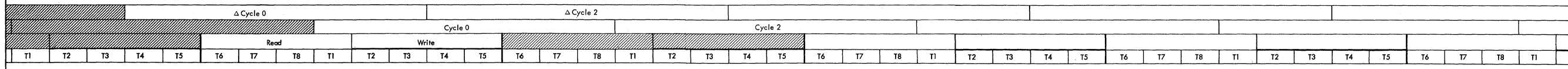
The sign is not checked for validity. If the sign pattern is invalid, the normalize function is inhibited. The invalid sign is set into the 'to reg'.



Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code							AC	To Reg			From Reg				
D							1	0	0	0	DD				

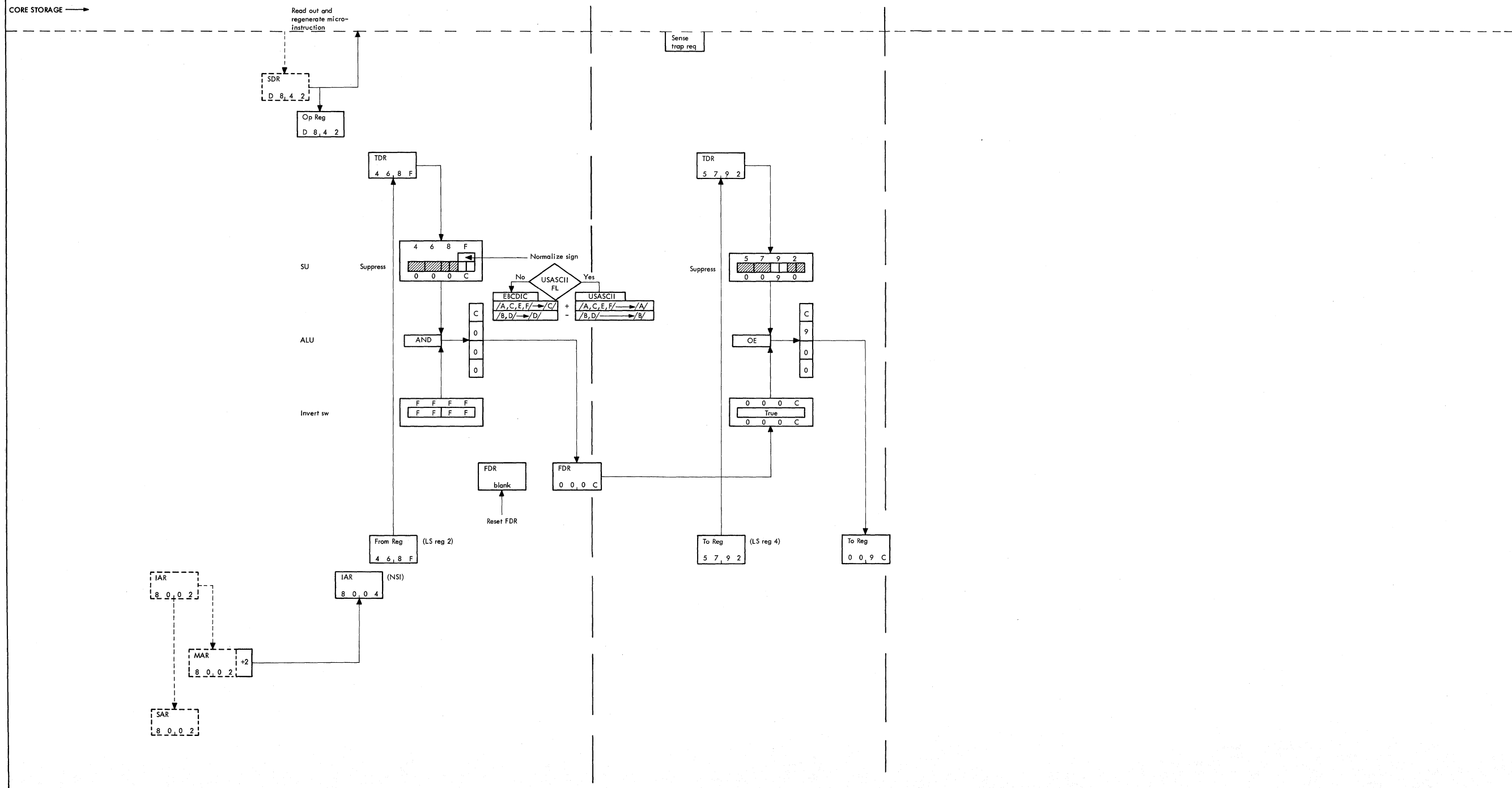
INST M542 MNEM OPERANDS R4#R4.0-11/R2.12-15 TEST FOR VALID SIGN AND NORMALIZE

Mnemonic
SDS
Format
FF
Type
DD

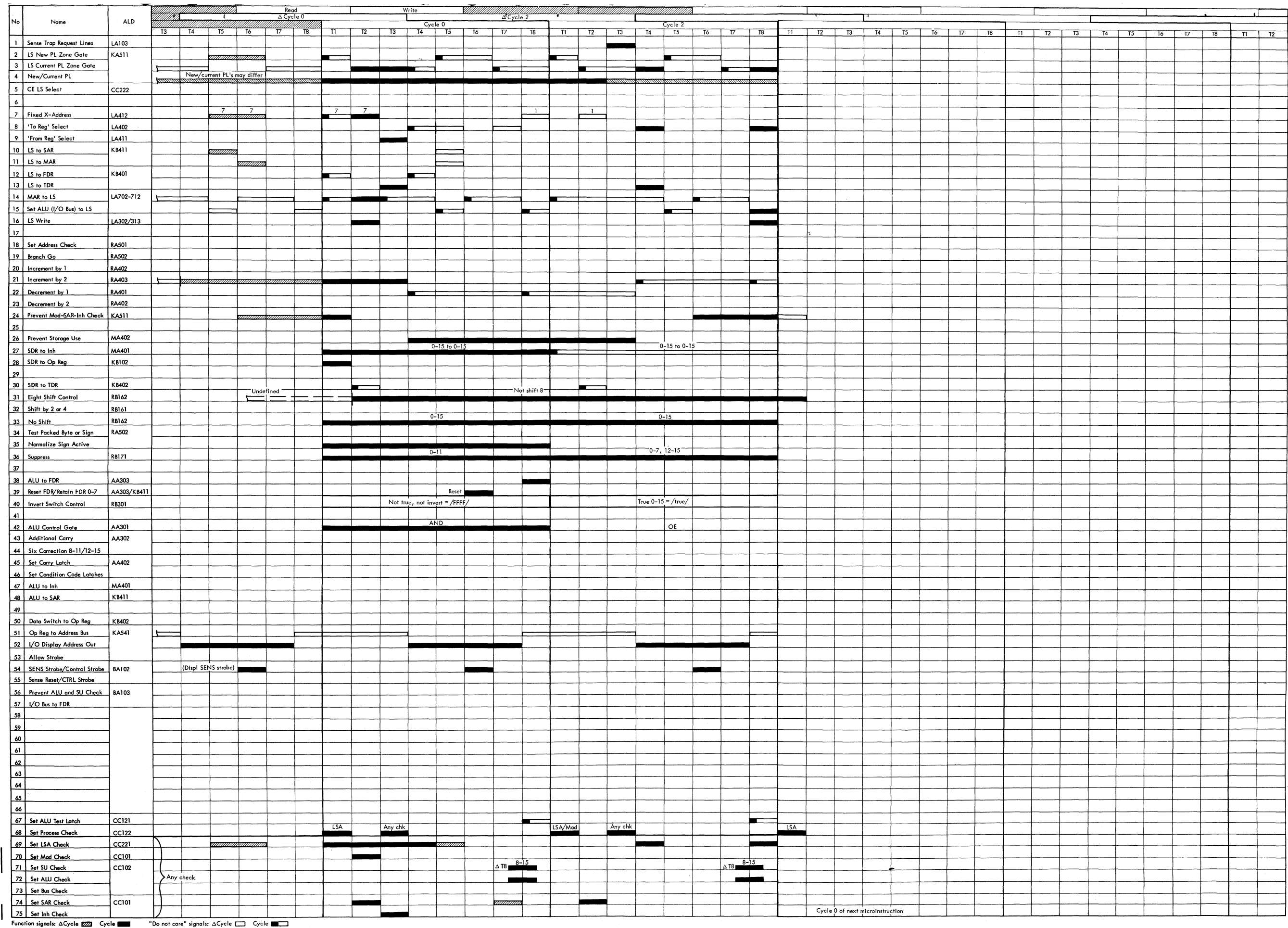


Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 Δ Cycle
 Cycle



Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



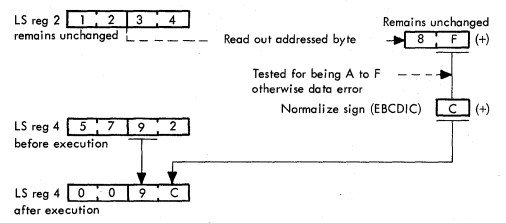
● Diagram 5-62. Set Decimal Sign (DD) (Part 2 of 2) (03767A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

The packed decimal sign in the low-order four bit position of the byte addressed by the 'from reg' is standardized, according to the selected code, and set into the low-order four bits of the 'to reg'. The high-order four bits of the low-order byte in 'to reg' as well as the 'from' byte remain unchanged. The high-order byte in 'to reg' is set to zero.

If the AC bit (instruction bit 7) is on, an address check occurs when the 'from address' is outside customer area. The 'from address' is not updated.

The code is specified by the USASCII latch (on = USASCII, off = EBCDIC). The USASCII latch can be turned on by a control/11, bit 14, not bit 12, and can be turned off by control/11, not bit 14 and not bit 12.

The sign is not checked for validity. If the sign pattern is invalid, the normalize function is inhibited. The invalid sign is set into the 'to reg'.



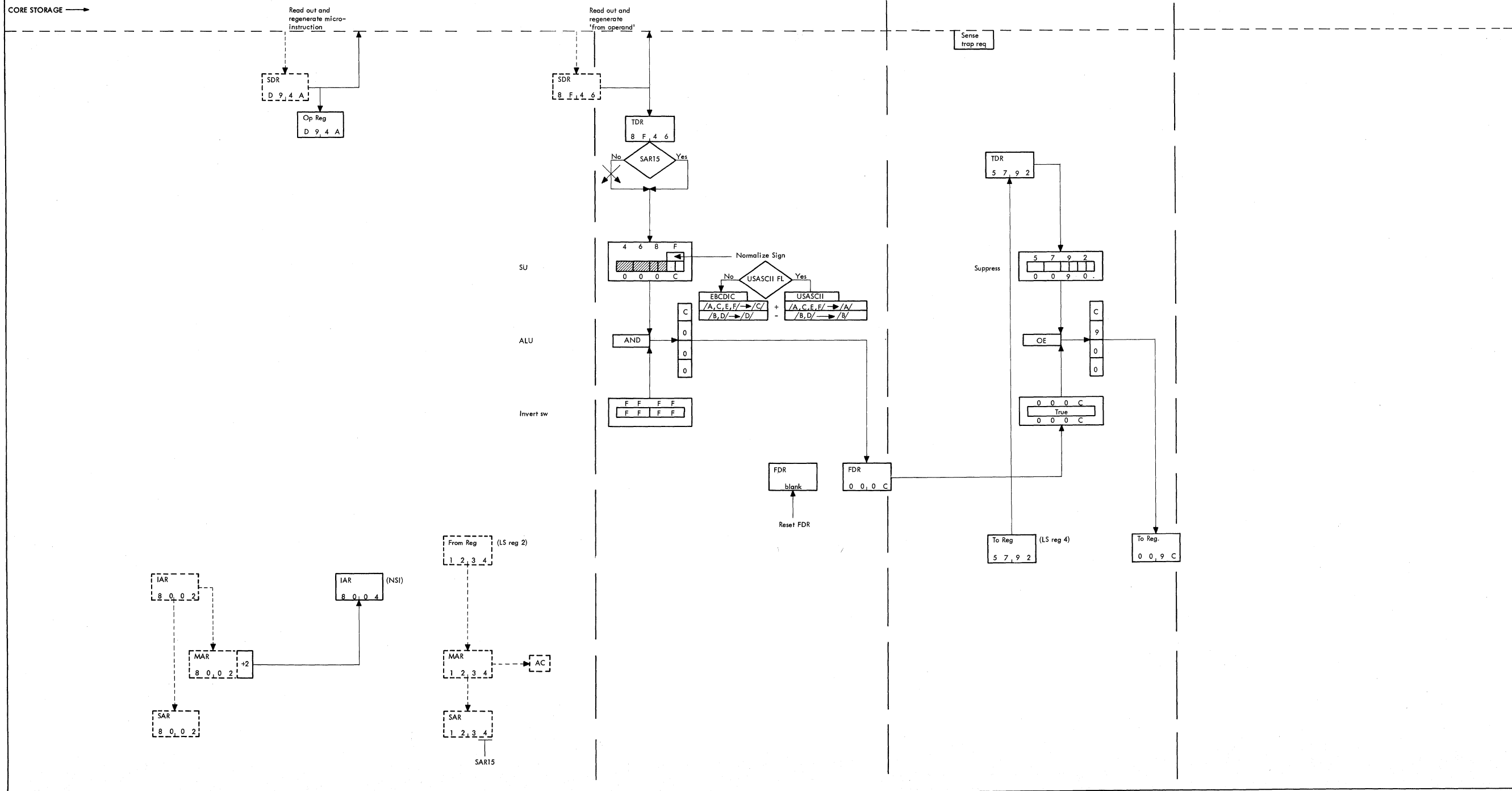
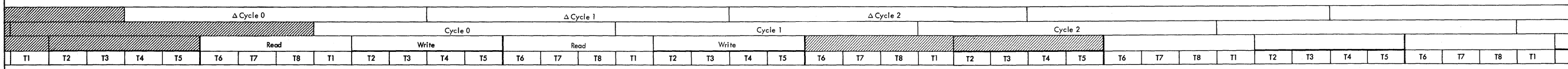
Microinstruction Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code							AC	To Reg				From Reg			
D							1	0	0	0	1				

If on, the 'from addr' is checked that it is not outside customer area

INST D94A MNEM SDS OPERANDS 4,2I,AC STATEMENTS ACCORDING TO STANDARD GEB 0-1046-XXX R4=R4.0-11/BY(R2,AC).4-7 TEST FOR VALID SIGN AND NORMALIZE

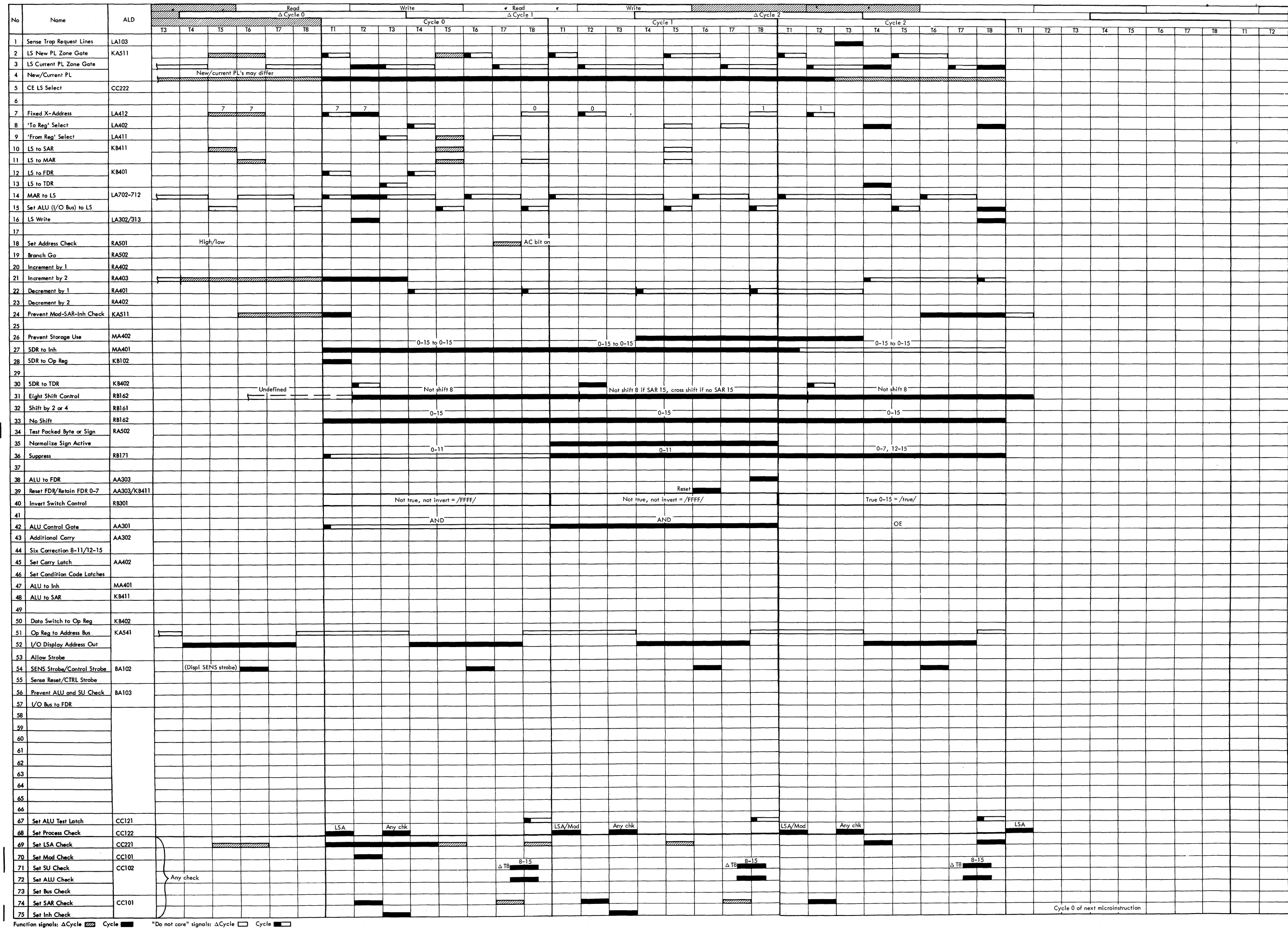
Mnemonic	SDS
Format	FF
Type	DX



Note: For "Do not care" functions refer to timing chart below.

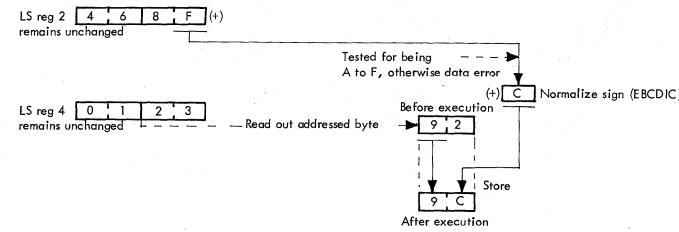
"Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



The packed decimal sign in the low-order four bits of the 'from reg' is standardized according to the selected code and set into the low-order four bit position of the byte addressed by the 'to reg'. The high-order four bits of the 'to' byte as well as the 'from' bits remain unchanged. If the AC bit (instruction bit 7) is on, an address check occurs when the 'to address' is outside customer area. The 'to address' is not updated.

The code is specified by the USASCII (on = USASCII, off = EBCDIC). The USASCII latch can be turned on by a control 11/, bit 14, not bit 12, and can be turned off by a control 11/, not bit 14 and not bit 12. A data check occurs when the low-order four bits in the from reg do not contain any hexadecimal value /A/ to /F/.



Microinstruction Layout																
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Op Code							AC	To Reg					From Reg			
D	1	0	0				1						0			

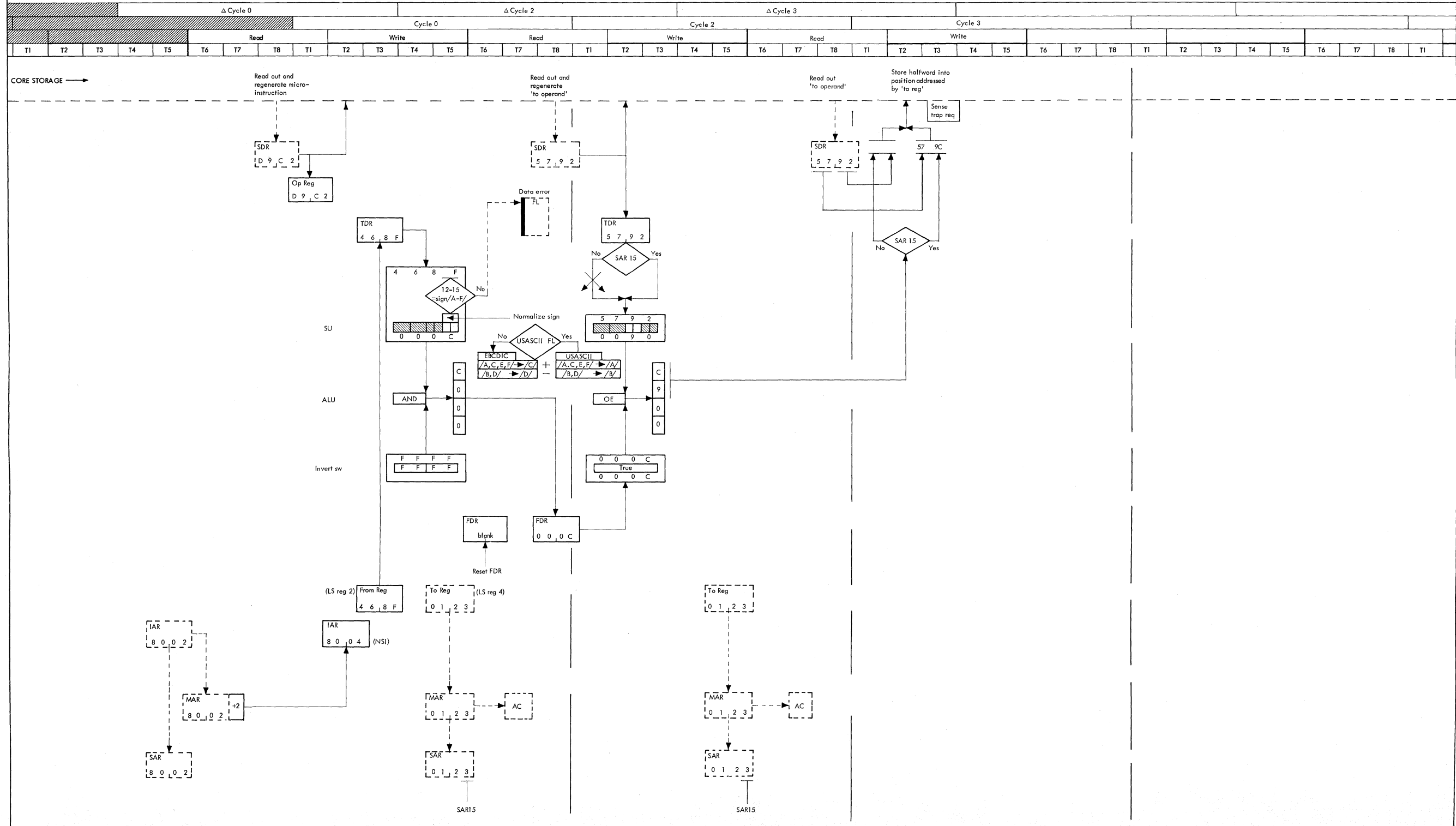
If on, the 'to addr' is checked that it is not outside customer area

INST MNEM OPERANDS STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
 D9C2 SDS 4I,2,AC BY(R4,AC).4-7=R2.12-15 TEST FOR VALID SIGN AND NORMALIZE

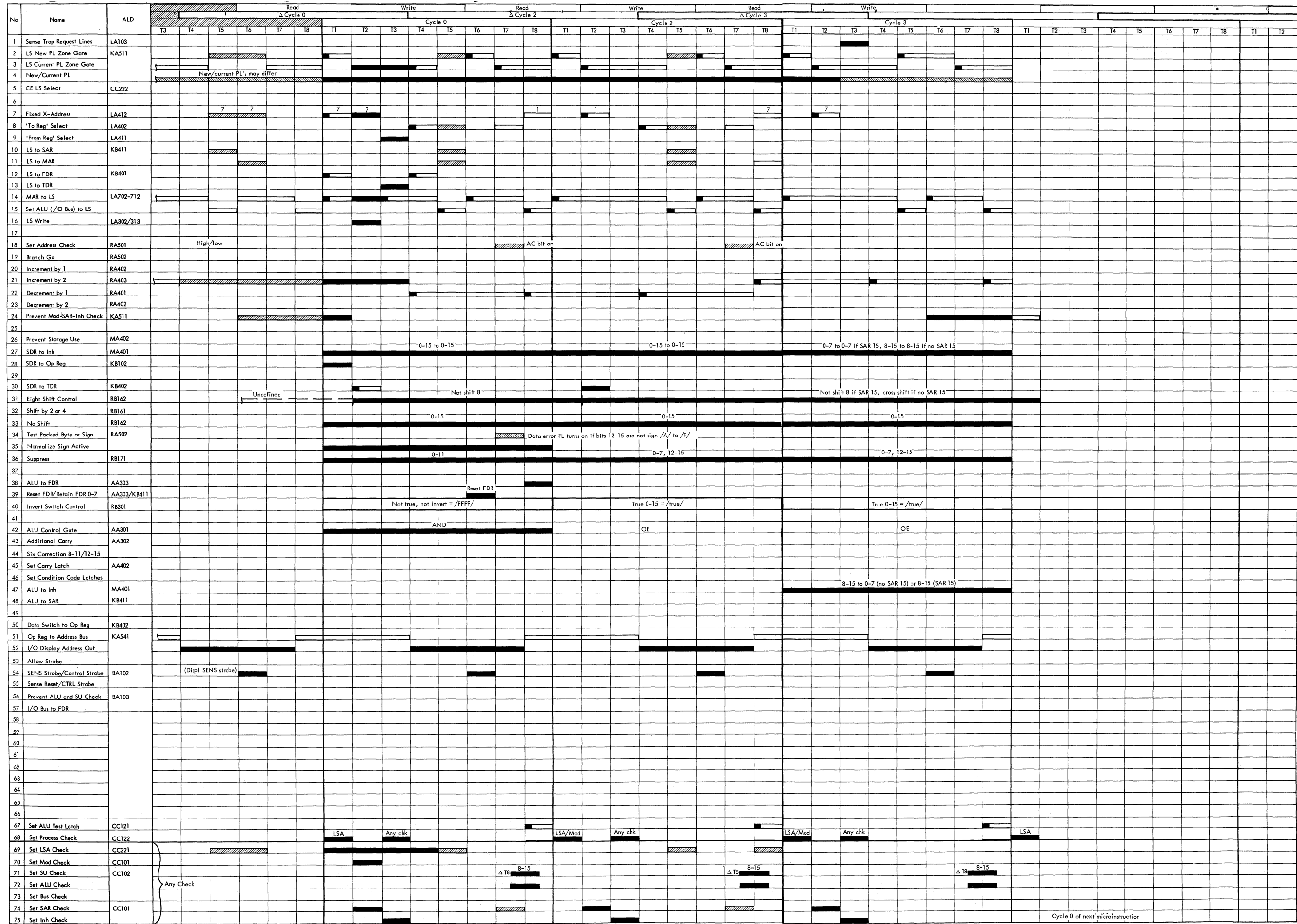
Mnemonic	SDS
Format	FF
Type	XD

Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle



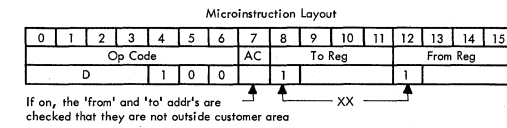
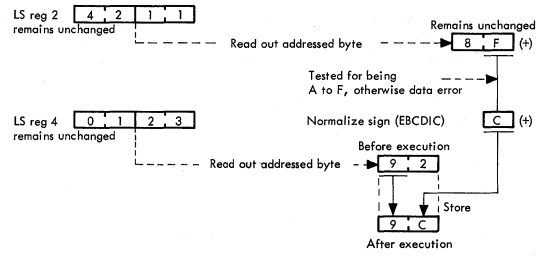
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



● Diagram 5-64. Set Decimal Sign (XD) (Part 2 of 2) (03769A)

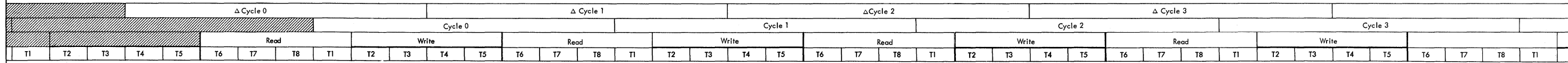
The packed decimal sign in the four low-order bits of the byte addressed by the 'from reg' is standardized according to the selected code and set into the four low-order bits of the byte addressed by the 'to reg'. The four high-order bits of the 'to' byte as well as the 'from' byte remain unchanged. 'From reg' and 'to reg' may contain the same address or may be the same LS register. If the AC bit (instruction bit 7) is on, an address check occurs when one of the operand addresses is outside customer area. The operand addresses are not updated.

The code is specified by the USASCII latch (on = USASCII, off = EBCDIC). The USASCII latch can be turned on by a ctrl/11/, bit 14, not bit 12, and can be turned off by ctrl/11/, not bit 14 and not bit 12. A data check occurs when the four low-order bits of the 'from' byte do not contain any hexadecimal value /A/ to /F/.



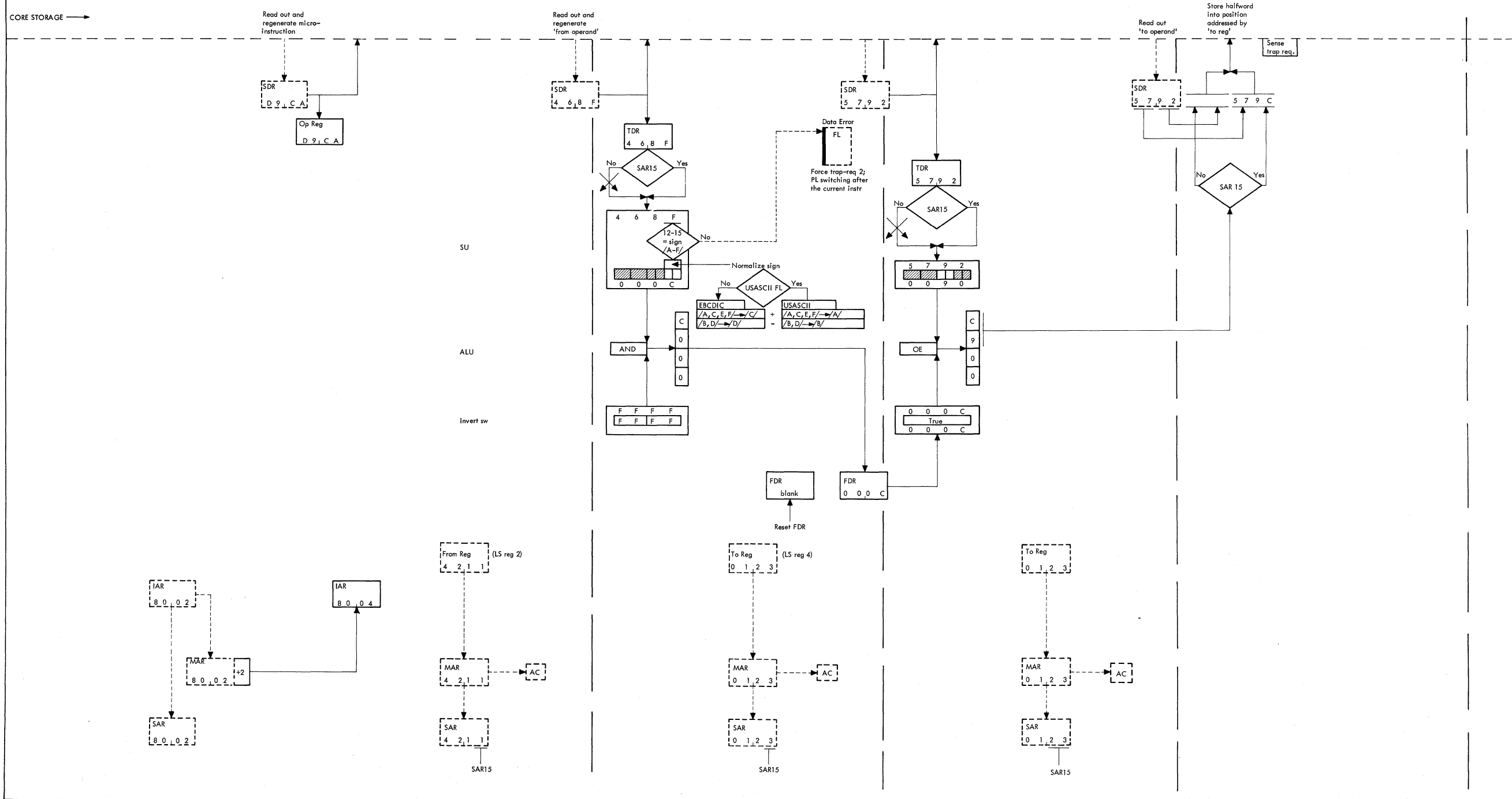
If on, the 'from' and 'to' addr's are checked that they are not outside customer area

Mnemonic	SDS
Format	FF
Type	XX

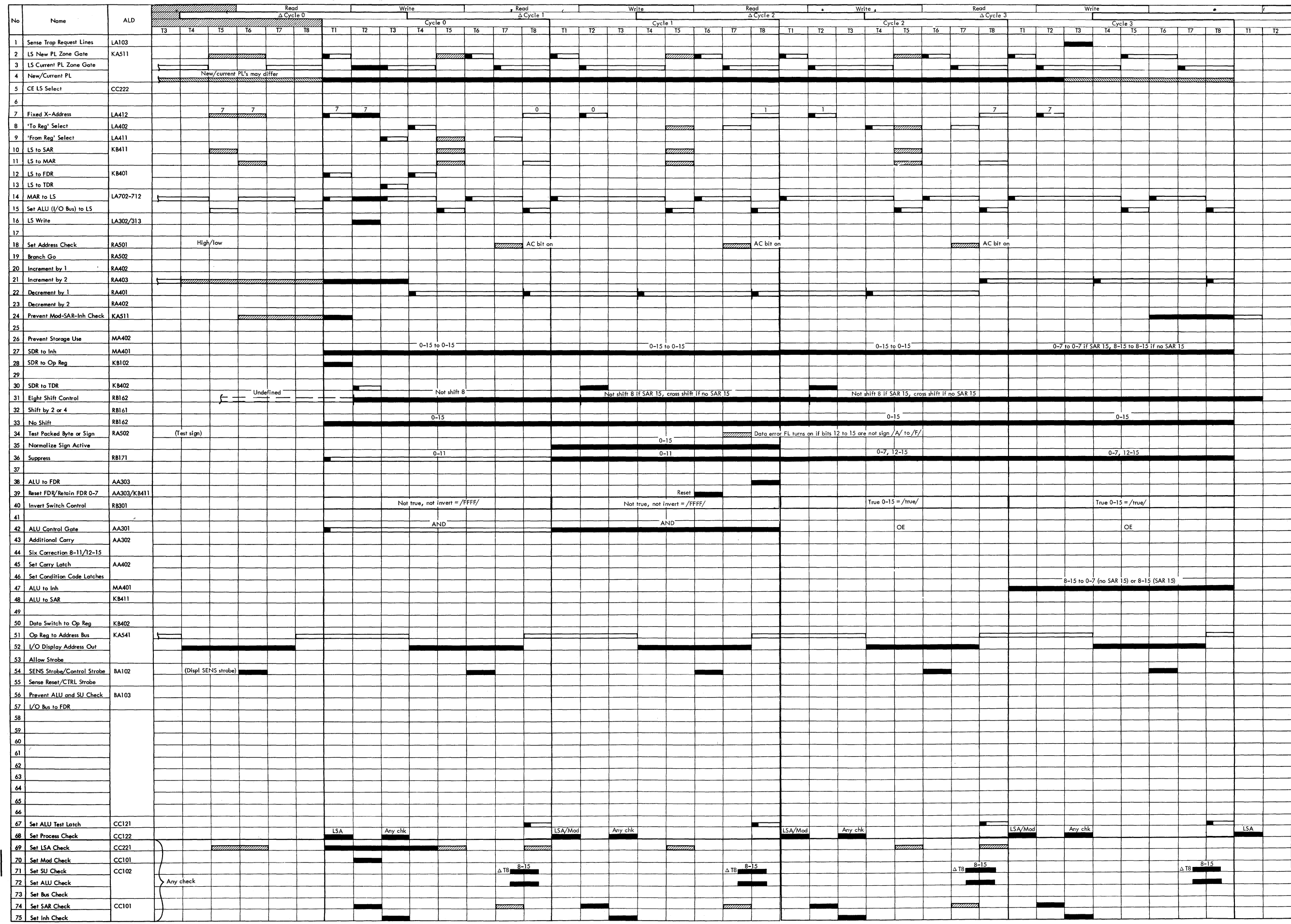


Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

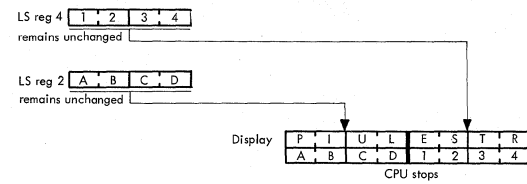


Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



Function signals: ΔCycle Cycle "Do not care" signals: ΔCycle Cycle

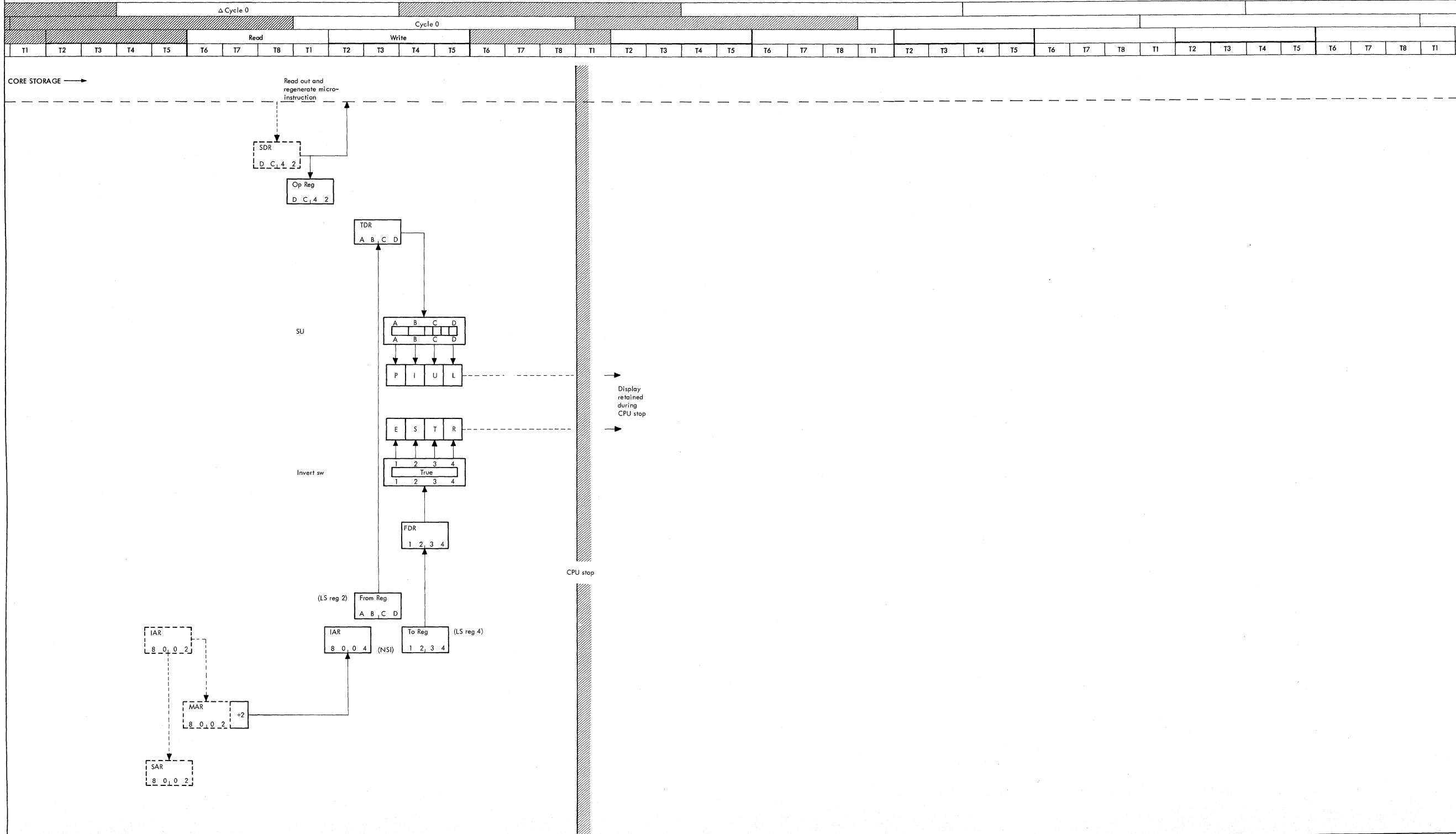
The CPU stops after executing the instruction. The 'from reg' is displayed in data register P, I, U, L. The 'to reg' is displayed in data register E, S, T, R. The CPU can be started again by operating the start key. The next microinstruction to be executed will be that addressed by the updated IAR (HALT instruction address + 2).



Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code								To Reg				From Reg			
D	1	1	0	0	0								0		

INST DC42 MNEM HALT OPERANDS 4, 2 STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
FDR=E, S, T, R←R4, TDR=P, I, U, L←R2

Mnemonic	HALT
Format	FF
Type	



Note: For "Do not care" functions refer to timing chart below.
"Do not care" signals:
ΔCycle
Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

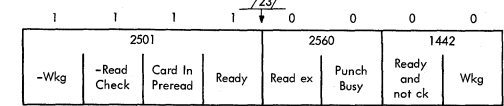


A byte or halfword provided by a sense source (CPU or attachment) is set into 'to reg'. The sense source is selected by the SENS/CTRL address in the low-order byte of the instruction (256 different sources are possible). The SENS/CTRL address is set on the address bus. The sense byte is set on the data bus in the attachment to which the current SENS/CTRL address is co-ordinated. Except for the halfword SENS's (SENS/CTRL addresses /14/ and /15/), the

SENS/CTRL address on the returning address bus is exclusive ORed with the original SENS/CTRL address in the Instruction. If both addresses are identical the result is zero. The result is set into the high-order byte of the 'to reg'.

Instruction Op Code 2 3

SENS/CTRL address to attachment ---> Activates sense source



LS reg 4 3 A B 4 before execution

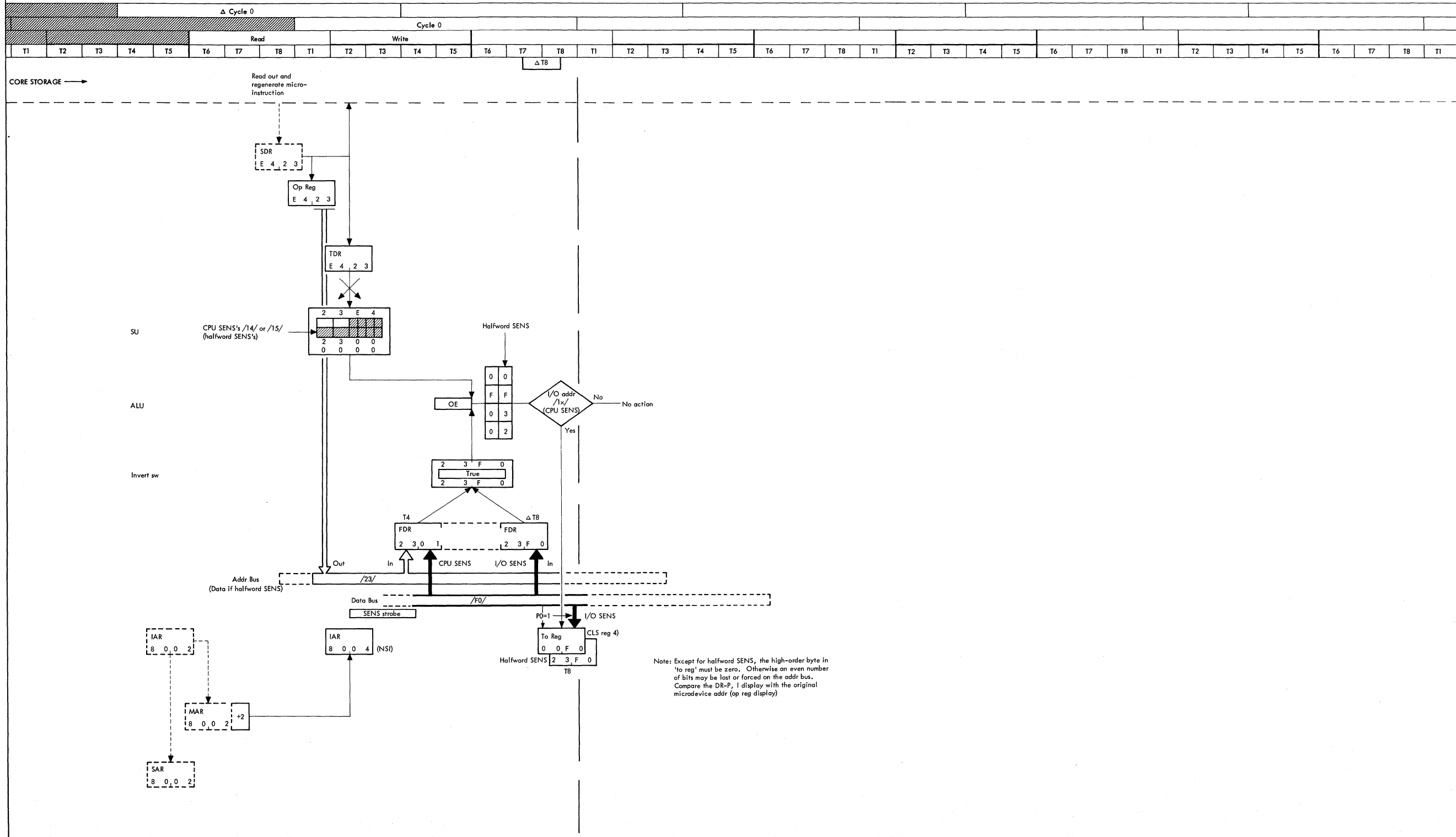
LS reg 4 0 0 F 0 after execution

Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code				To Reg				SENS/Ctrl Addr							
E				0											

direct addressing

INST E423 MNEM SENS OPERANDS 4, X'23' STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX R4 += *001/SENSE 23

Mnemonic	SENS
Format	
I/O	
Type	Direct Addressing

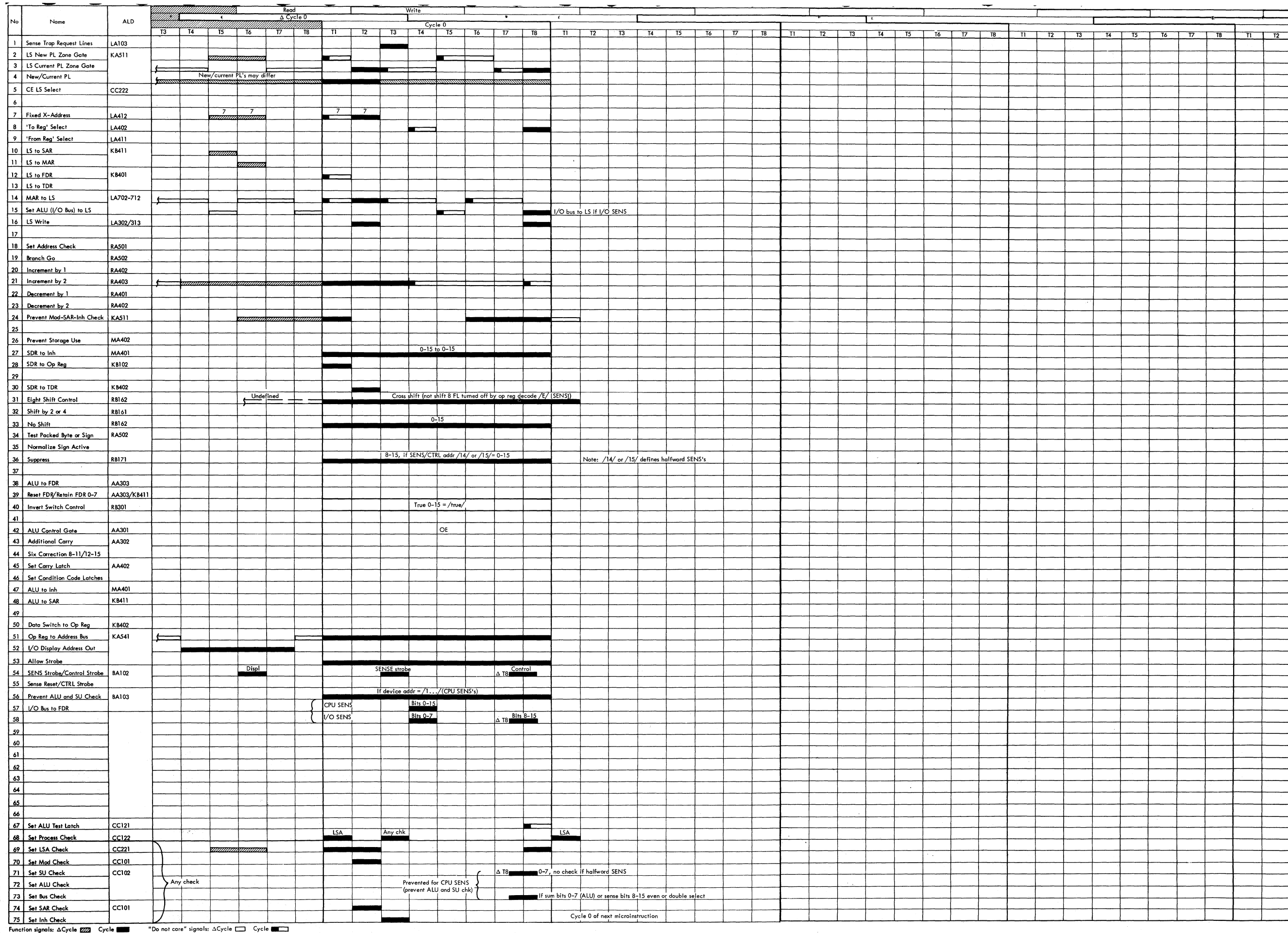


Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

Note: Except for halfword SENS, the high-order byte in 'to reg' must be zero. Otherwise an even number of bits may be lost or forced on the addr bus. Compare the DR-P, I display with the original microdevice addr (op reg display)

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



A byte provided by a sense source (CPU or attachment) is stored into the byte addressed by the 'to reg'. The 'to address' is incremented by 1. The sense source is selected by the SENS/CTRL address in the low-order byte of the instruction (256 different sources are possible). The SENS/CTRL address is set on the address bus. The sense byte is set on the data bus in the attachment to which the current SENS/CTRL address is co-ordinated. The SENS/CTRL address on the returning address bus is exclusive ORed with the original SENS/CTRL address in the instruction. If both addresses are

identical the result is zero. The result can be displayed in the high-order ALU byte.

Instruction Op Code 2 3

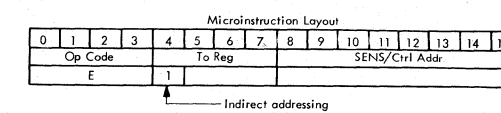
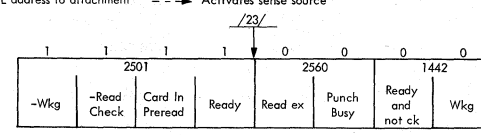
----- Sens/CTRL address to attachment -----> Activates sense source

LS reg 4 2 E E 2

Read out addressed byte

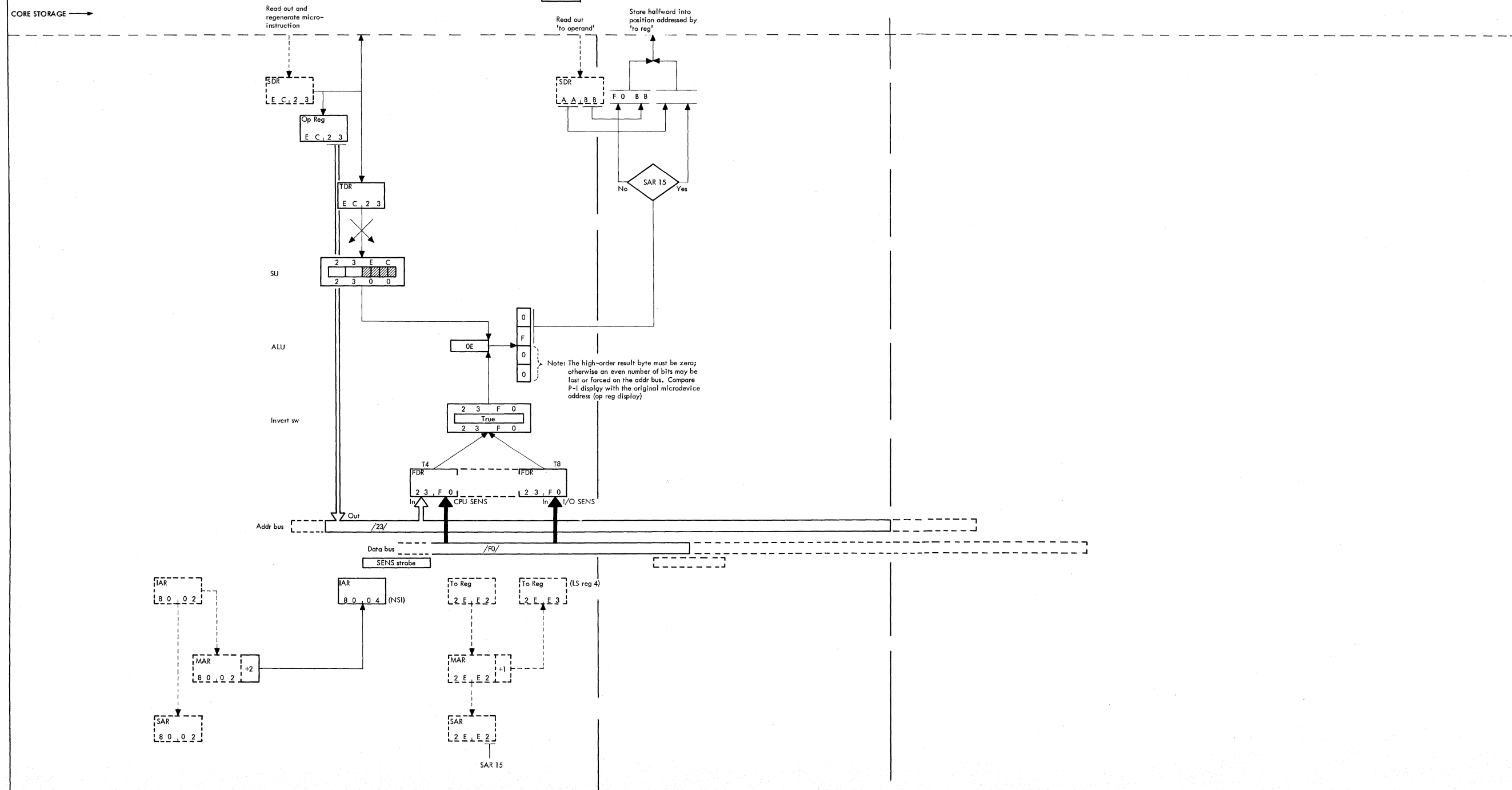
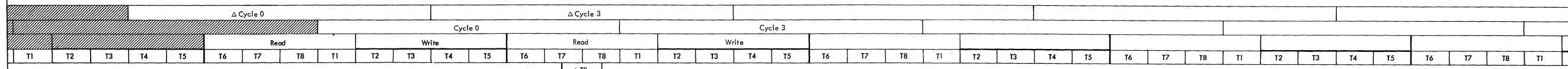
LS reg 4 2 E E 3

after execution



INST EC23 MNEM SENS OPERANDS 41, X'23' STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX BY(R4, +1) += SENSE 23

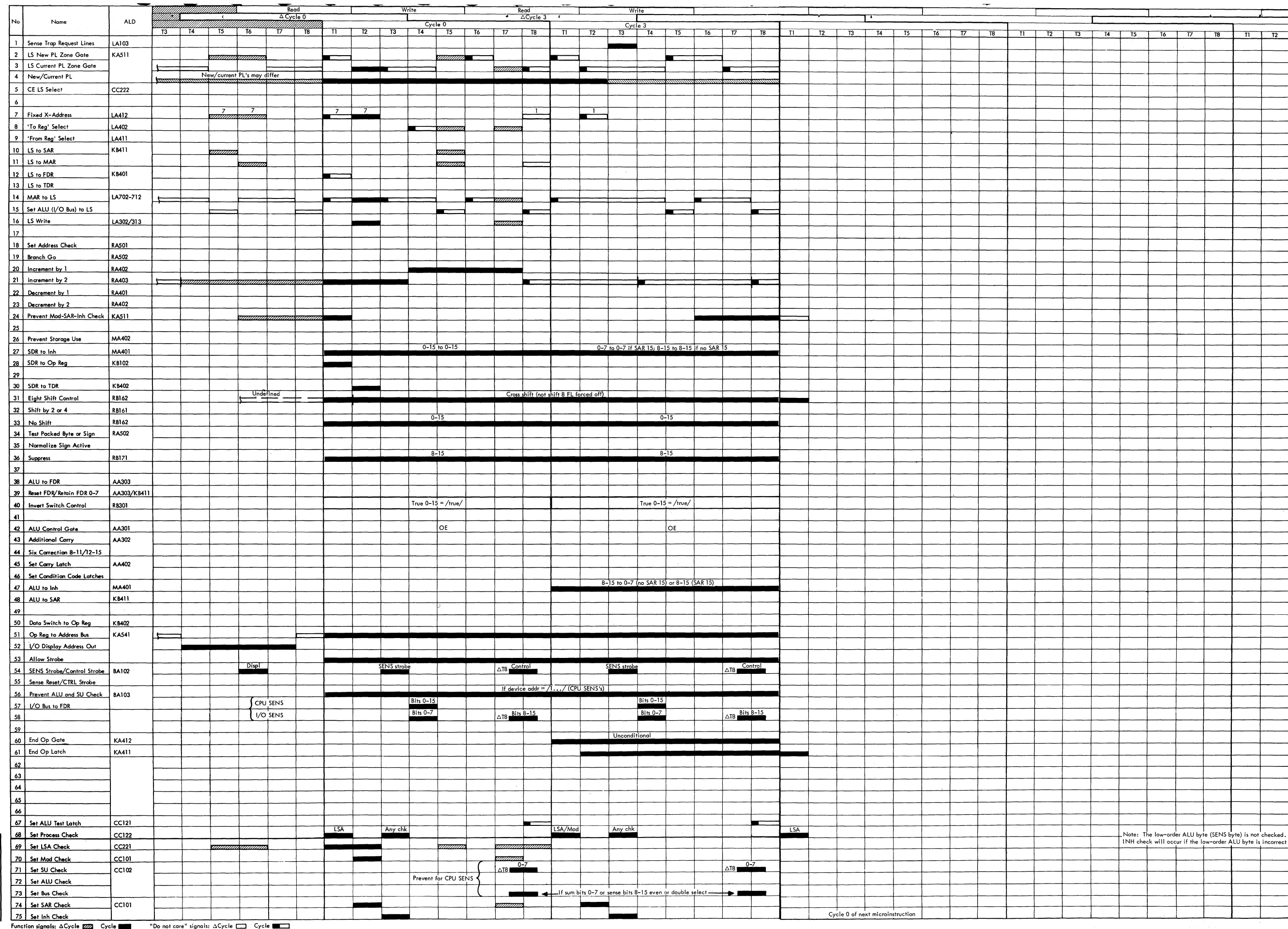
Mnemonic	SENS
Format	
I/O	
Type	
Indirect Addressing	



Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



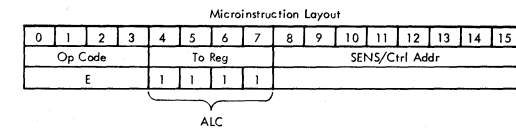
● Diagram 5-68. SENS CPU I/O (Indirect Addressing) (Part 2 of 2)

(03773A)

2020 ≥ 50,000 FEMDM Vol 2 (8/69)

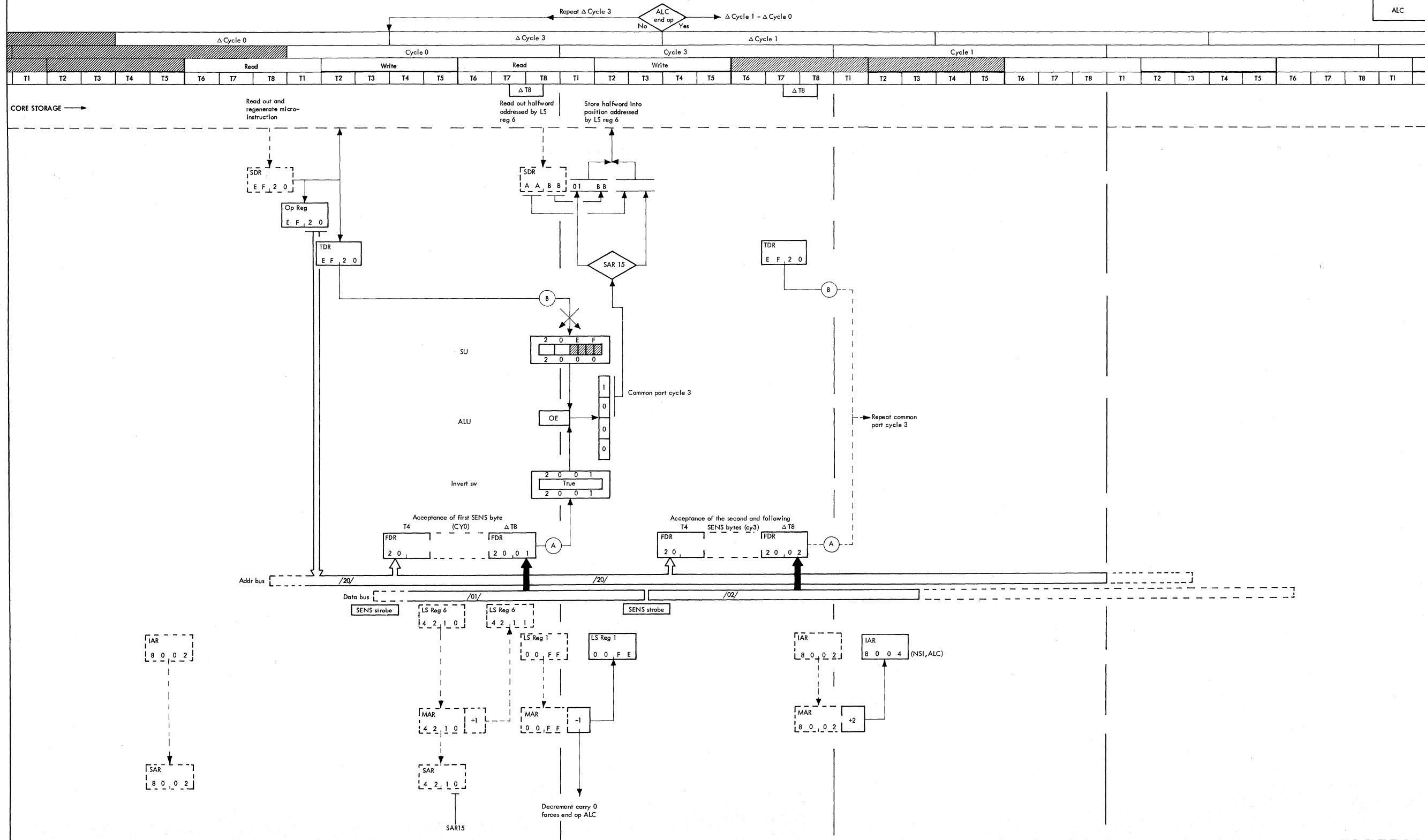
If the 'to reg' field contains // (11), an indirectly addressed SENS (see Diagram 5-69) is executed in ALC mode. LS register 1 contains the field length (real number of bytes to be sensed reduced by 1). LS register 6 contains the address of the byte into which the first sense byte must be stored. The field length is decremented by 1, and the store address is incremented by 1, every time a sense byte is stored.

This SENS ALC can be used to question a sense source with a repetition rate of 2 microseconds (diagnostic applications, ICR).



INST MNEM OPERANDS STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
 EF20 SENS 7I,X*20⁴ BY(R6,+1, UNTIL R1, LT, 0)*= SENSE 20

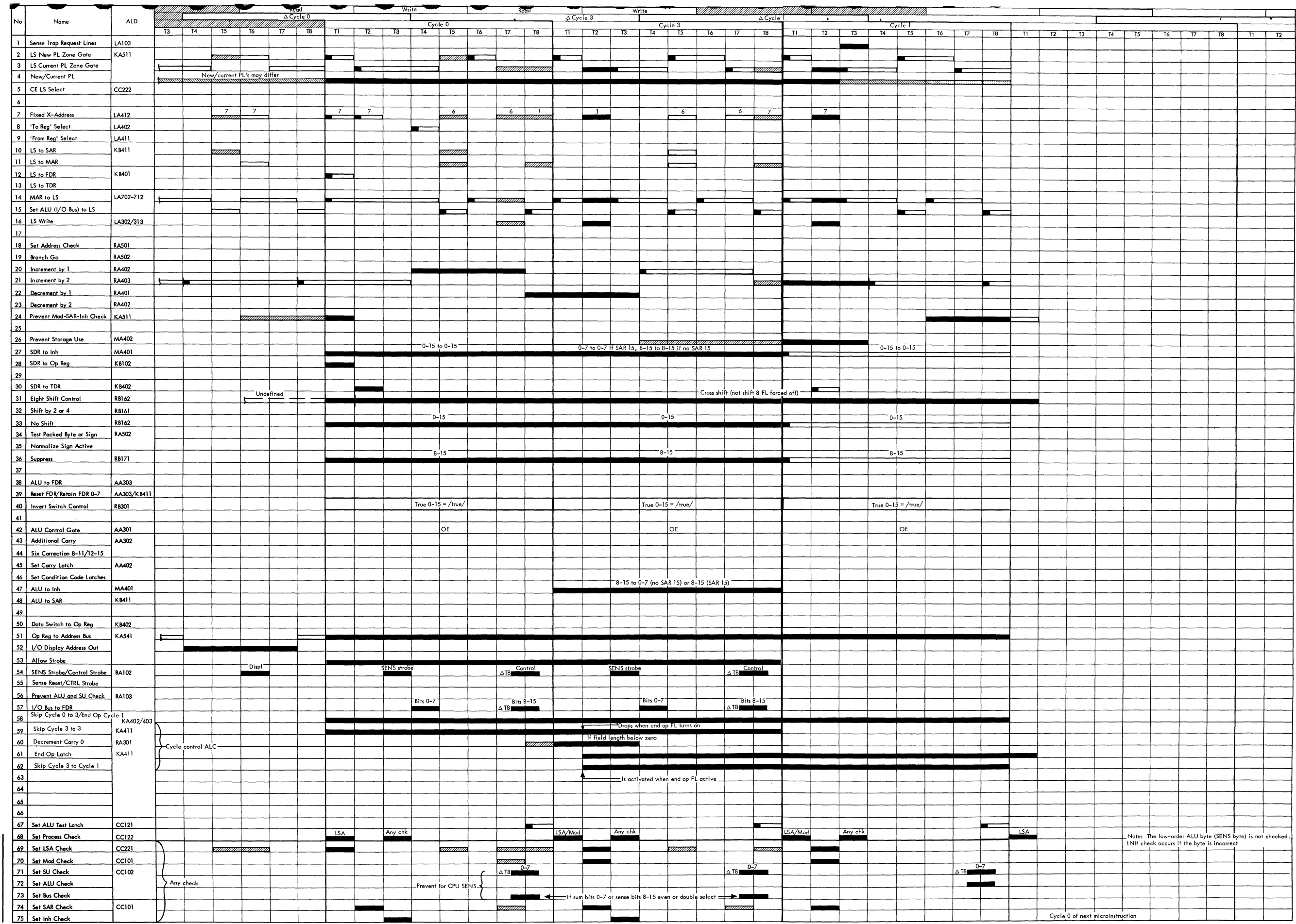
Mnemonic
SENS
Format
I/O
Type
ALC



Note: For "Do not care" functions refer to timing chart below.

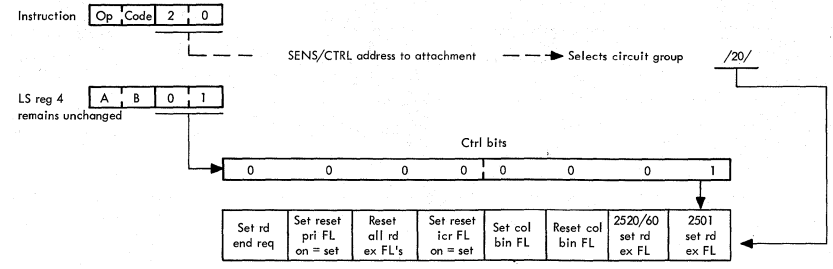
"Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



The low-order byte in 'to reg' is set onto the DATA BUS. The SENS/CTRL address (low-order byte of the instruction) is set on the address bus and selects the circuit group (in CPU or attachment) which shall be controlled according to the CTRL data on the data bus. The 'to reg' remains unchanged.

The returning address and data bus are set into FDR for display (E-S = returning SENS/CTRL addr, T-R returning ctrl data) if the CPU stops after the instruction. The data bus output is displayed in U-L.

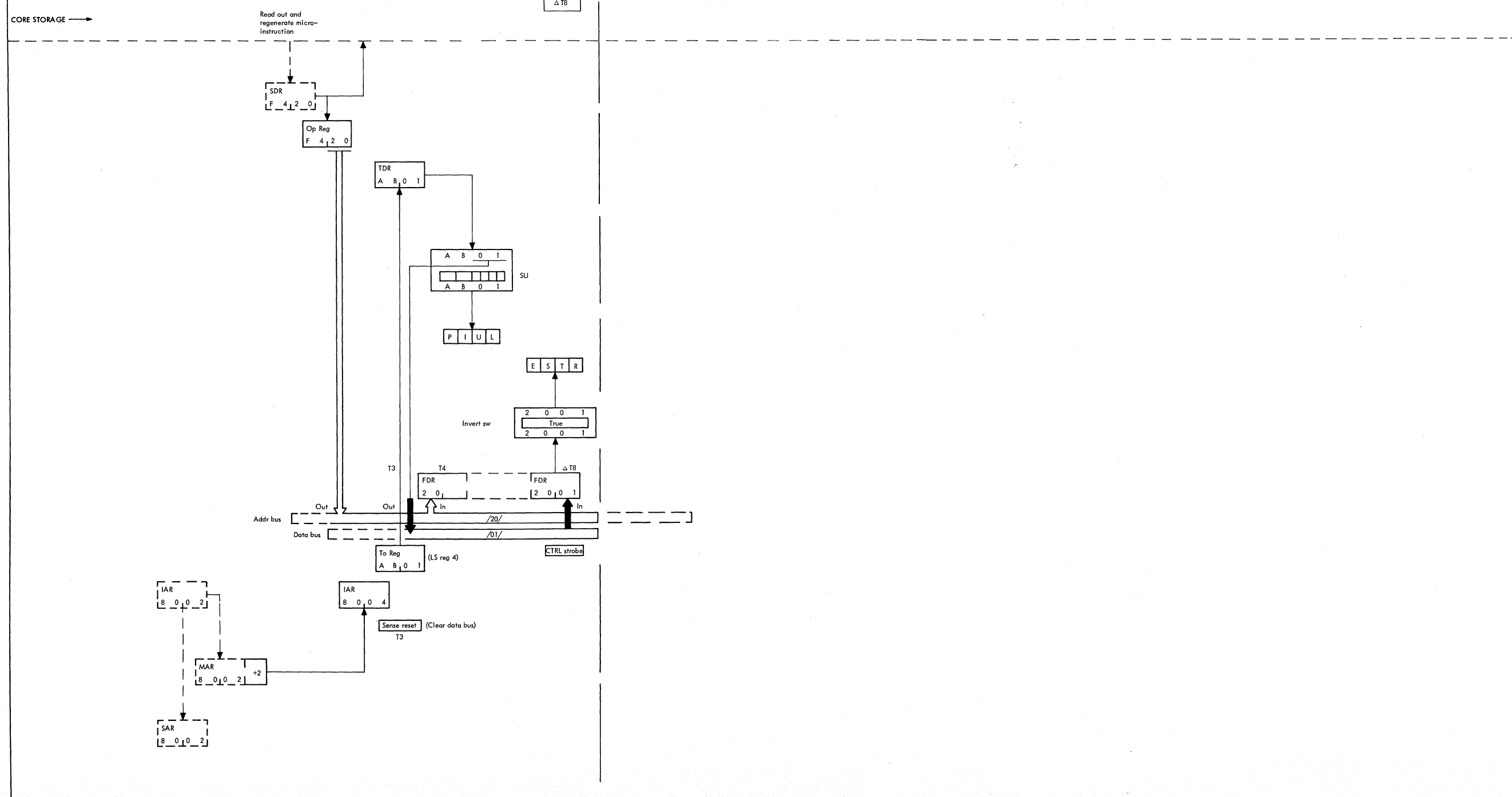
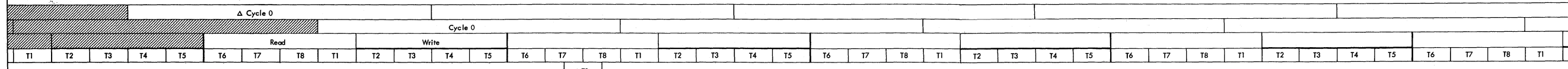


Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code				To Reg				SENS/Ctrl Addr							
F				0											

Direct addressing

INST F420
 MNEM CTRL 4, X'20'
 OPERANDS
 STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
 CONTROL 20/R4.8-15

Mnemonic
CTRL
Format
I/O
Type
Direct Addressing



Note: For "Do not care" functions refer to timing chart below.

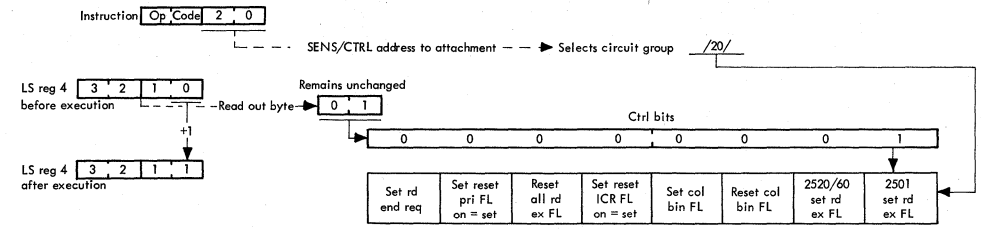
"Do not care" signals:
 ΔCycle
 Cycle

Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

The byte addressed by the 'to reg' is set onto the data bus. The SENS/CTRL address (low-order byte of the instruction) is set on the address bus and selects the circuit group (in CPU or attachment) which shall be controlled according to the CTRL data on the data bus. The 'to' byte remains unchanged. The 'to' address is incremented by 1.

The returning address and data bus are set into FDR for display (E-S = returning SENS/CTRL addr, T-R returning CTRL data) if the CPU stops after the instruction.

The data bus output is displayed in U-L.

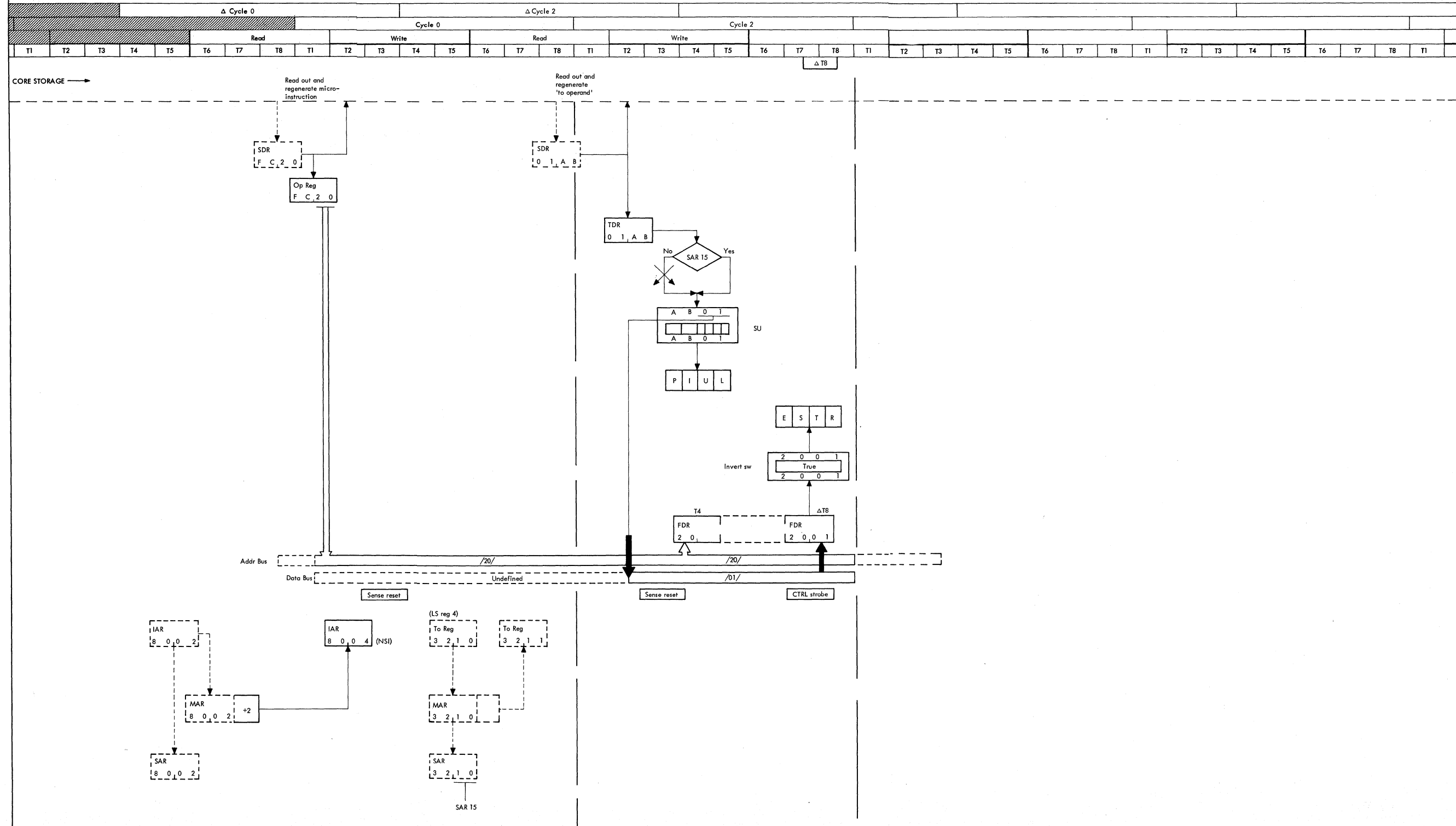


Microinstruction Layout															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Op Code				To Reg		SENS/Ctrl Addr									
F	1														

Indirect addressing

INST FC20 MMEM OPERANDS CTRL 41, X*20* STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX CONTROL 20/BY(R4,+1)

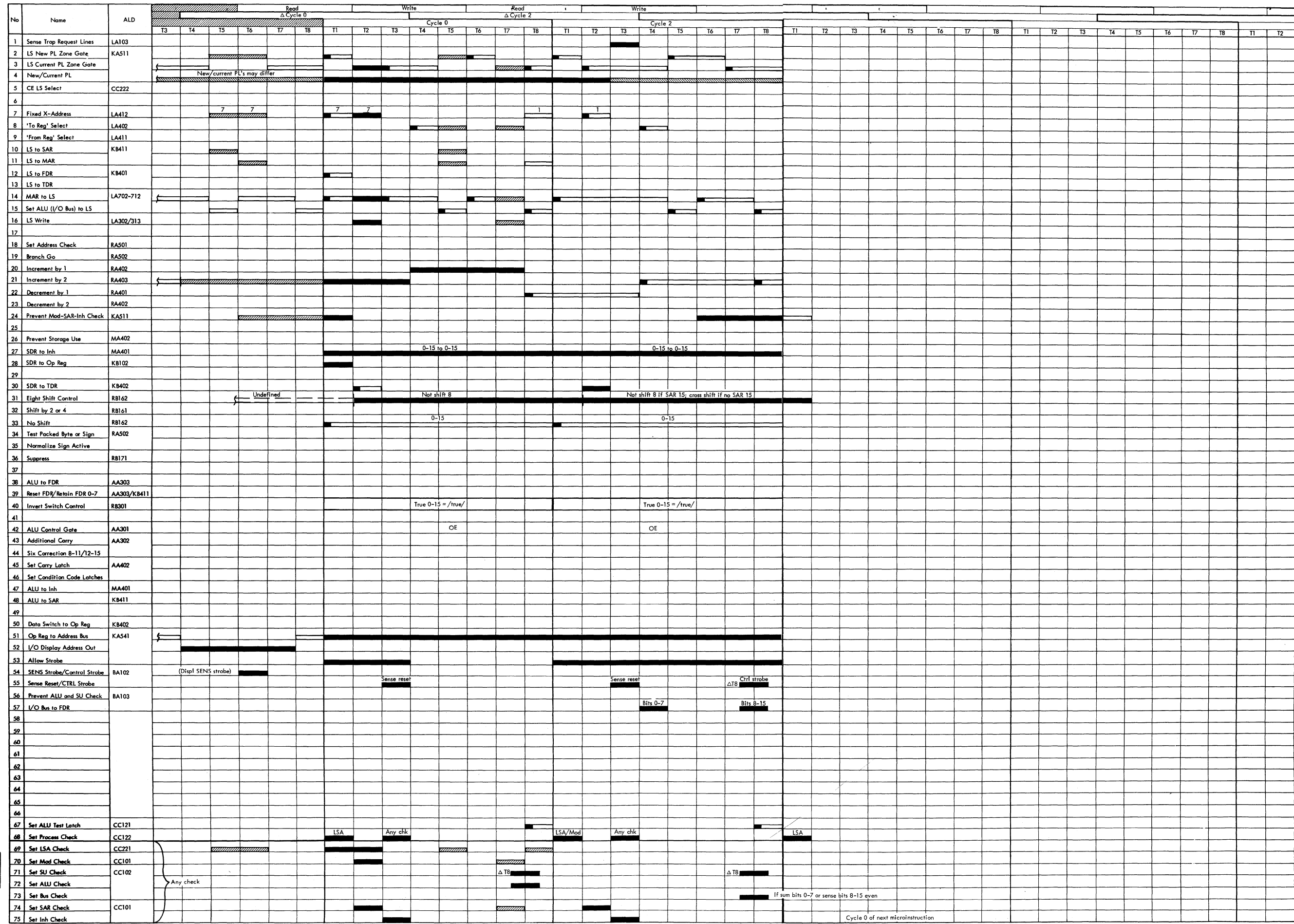
Mnemonic
CTRL
Format
I/O
Type
Indirect Addressing



Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
 ΔCycle
 Cycle

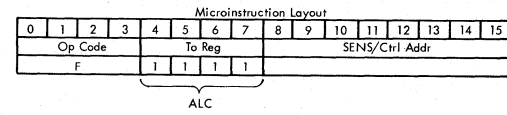
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.



Function signals: Δ Cycle Cycle "Do not care" signals: Δ Cycle Cycle

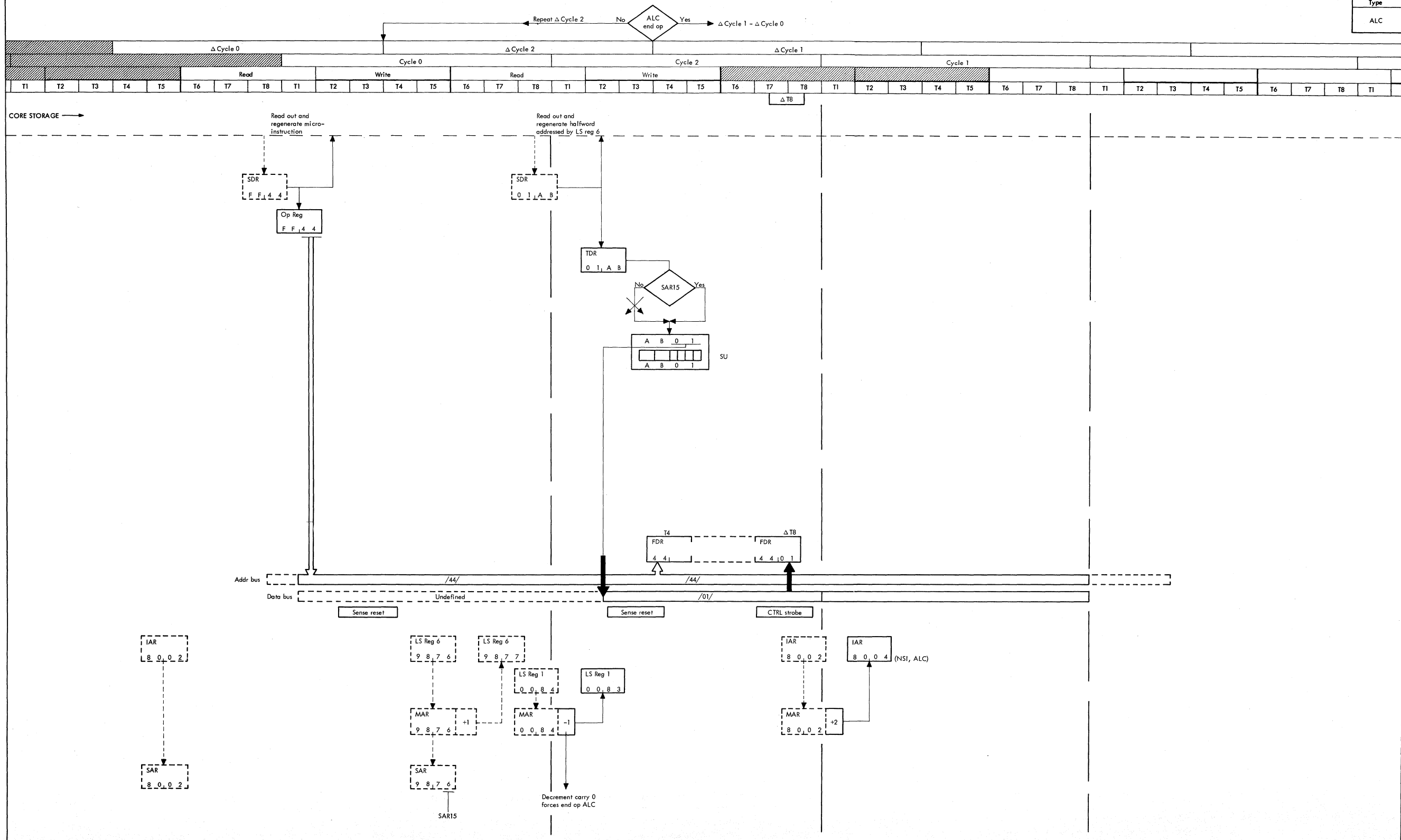
The indirect addressed ctrl (see Diagram 5-71) is executed in ALC mode when the 'to reg' field contains 77(1111). LS register 1 contains the field length (real number of bytes, to be transferred to the addressed circuit group, reduced by 1).
LS register 6 contains the address of the first CTRL data byte. The field length and the CTRL data address is updated

every time a byte is transferred. The field length is decremented by 1, and the CTRL data address is incremented by 1.
The CTRL ALC is used for high speed transfer of a string of bytes (one byte within 2 micro-seconds) to a single circuit group (eg. print buffer).



INST FF44 MNEM OPERANDS STATEMENTS ACCORDING TO STANDARD CEB 0-1046-XXX
CTRL 71, X'44' CONTROL 44/BY(R6,+1, UNTIL R1,LT,0)

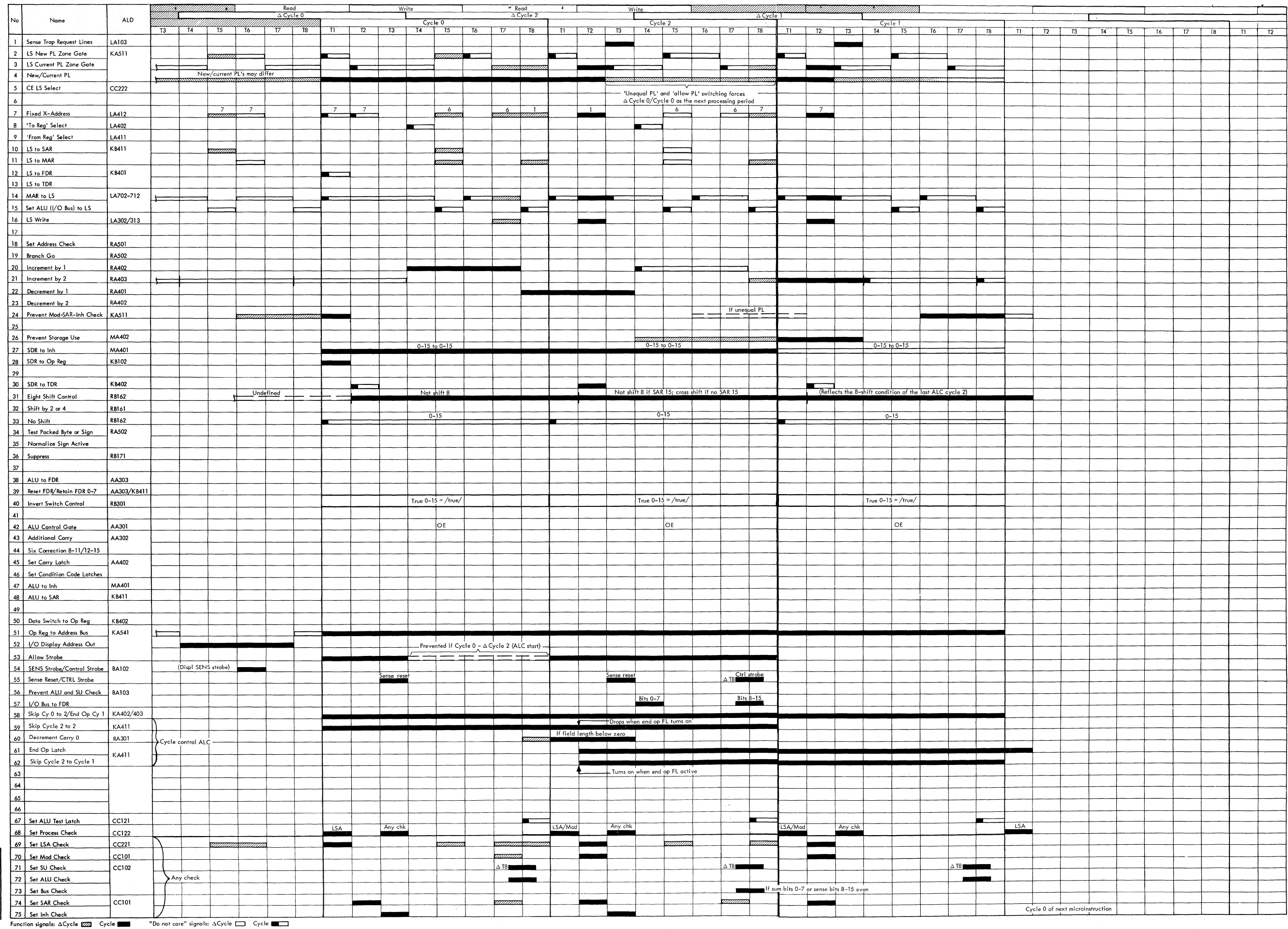
Mnemonic
CTRL
Format
I/O
Type
ALC



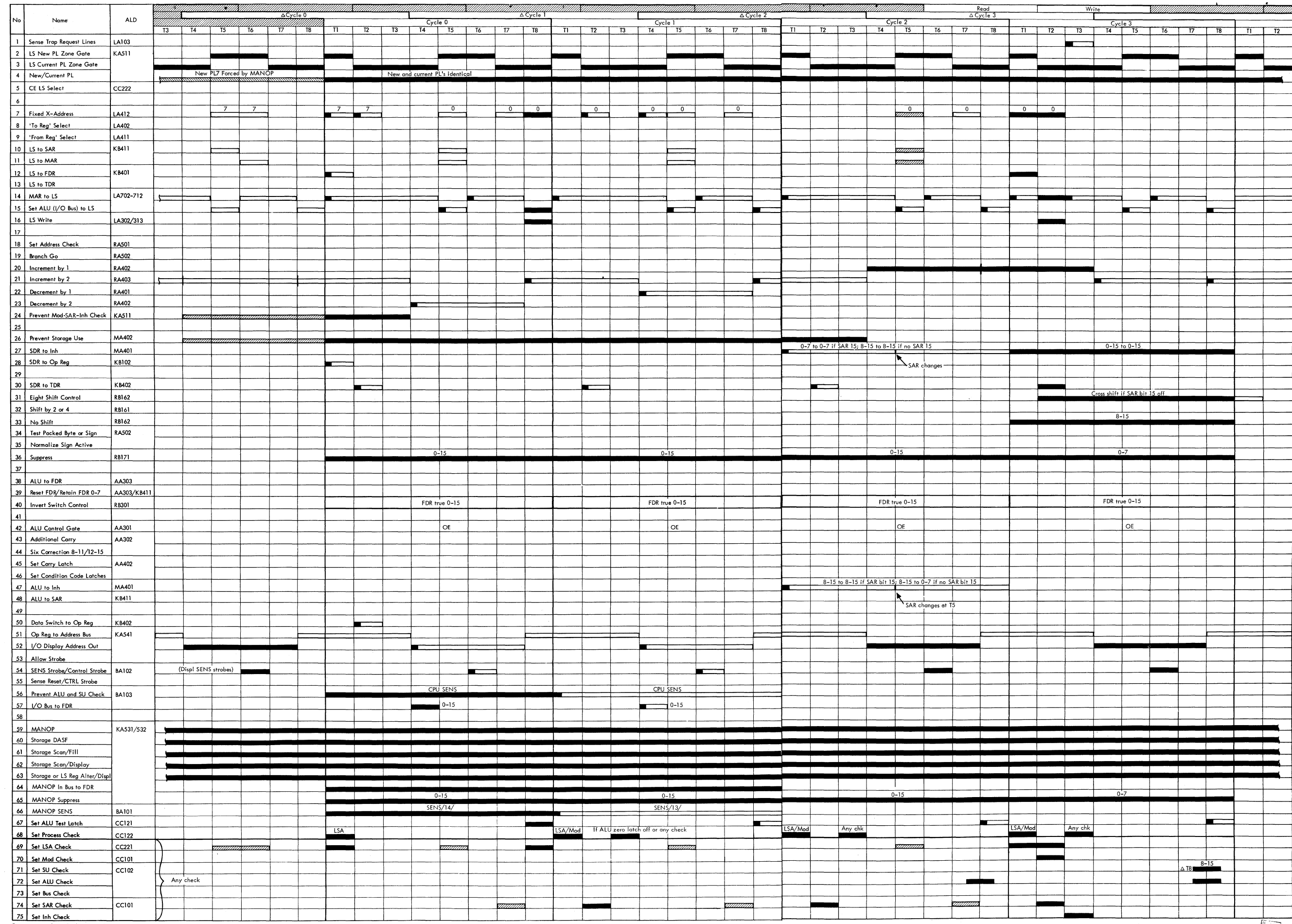
Note: For "Do not care" functions refer to timing chart below.

"Do not care" signals:
ΔCycle
Cycle

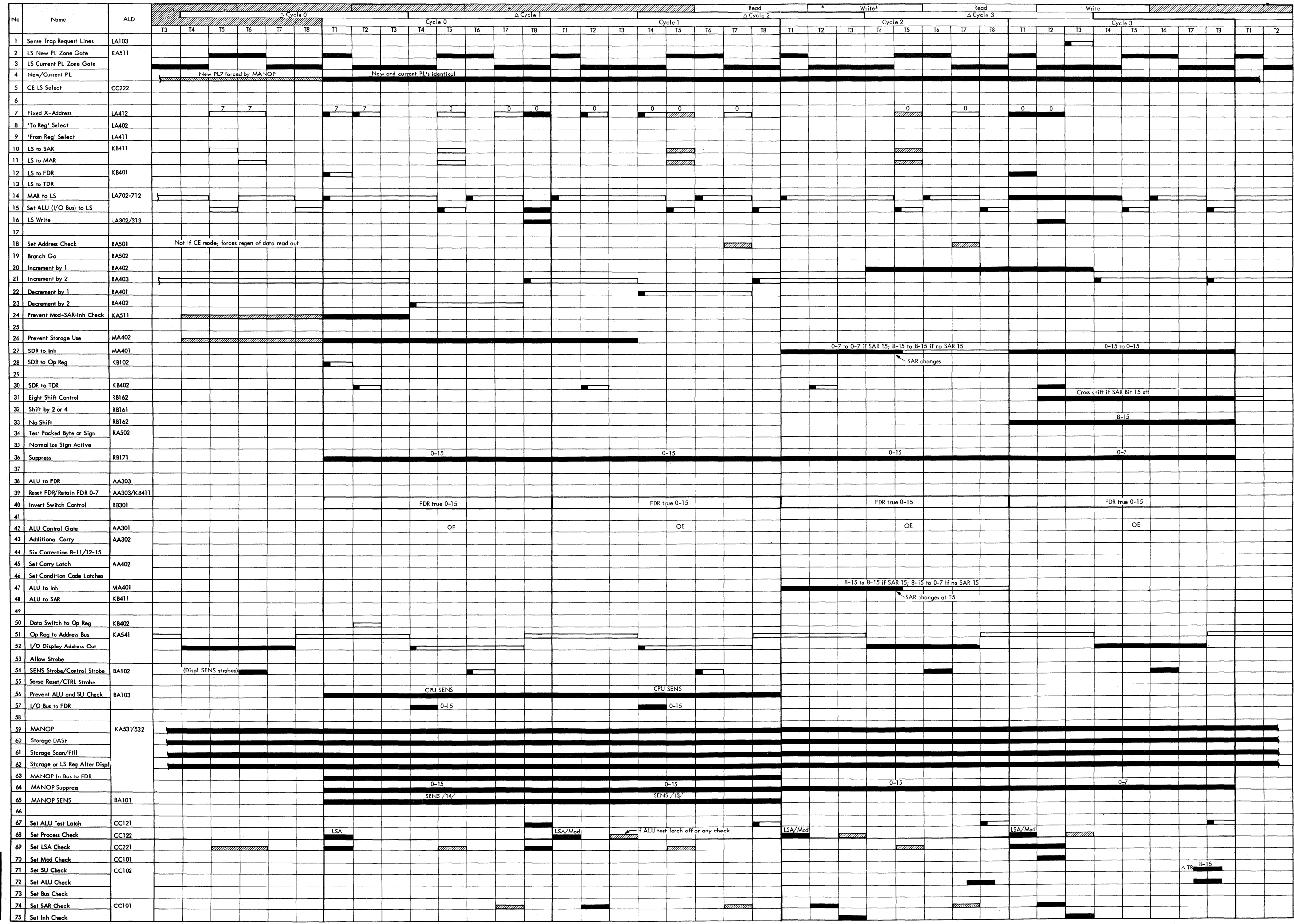
Functions performed during cycle time are shown by full lines and functions performed during Δ cycle time are shown by dotted lines.

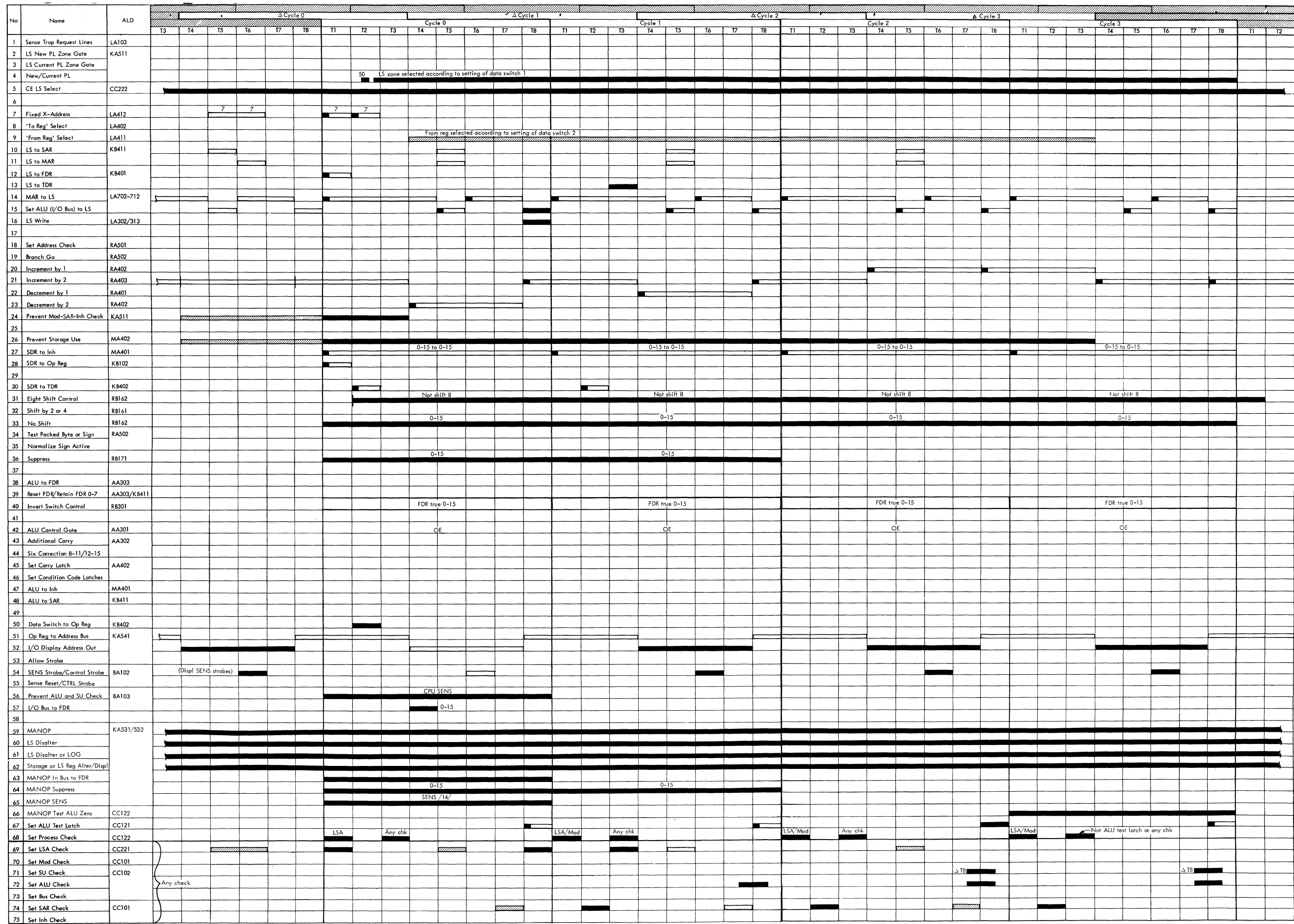


● Diagram 5-72. Control I/O (ALC) (Part 2 of 2) (03777A) 2020 ≥ 50,000 FEMDM Vol 2 (8/69)

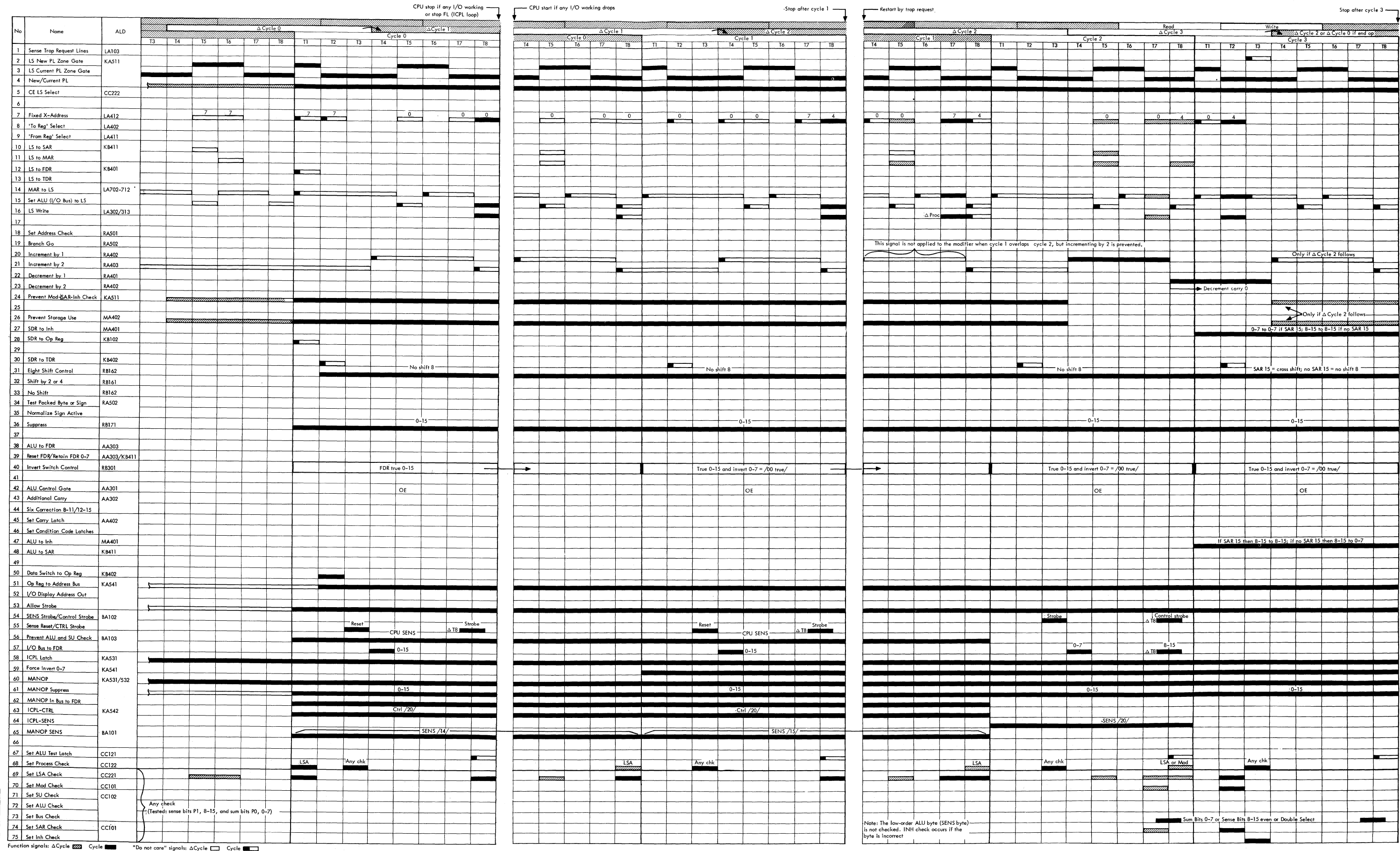


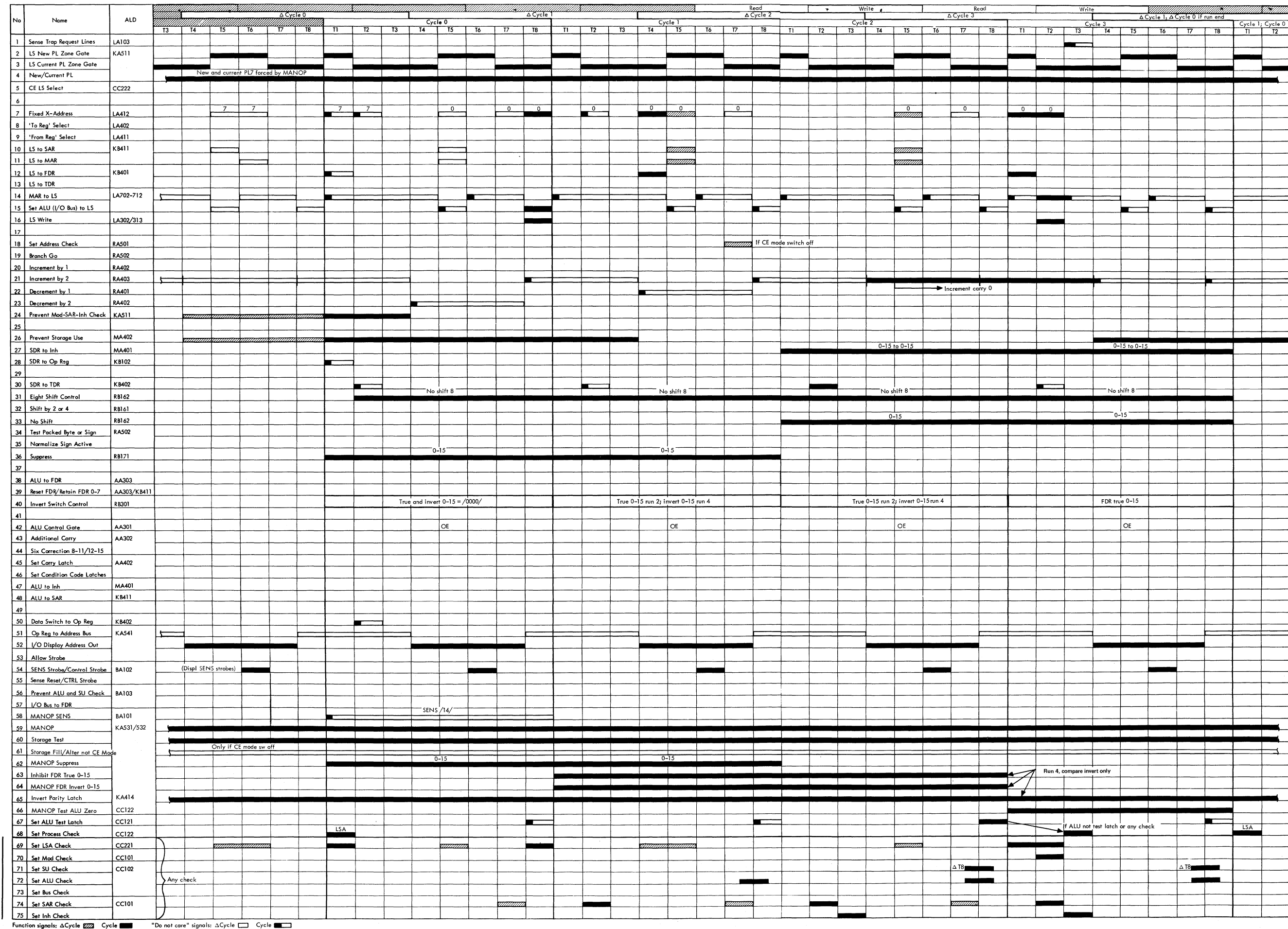
Function signals: ΔCycle Cycle "Do not care" signals: ΔCycle Cycle

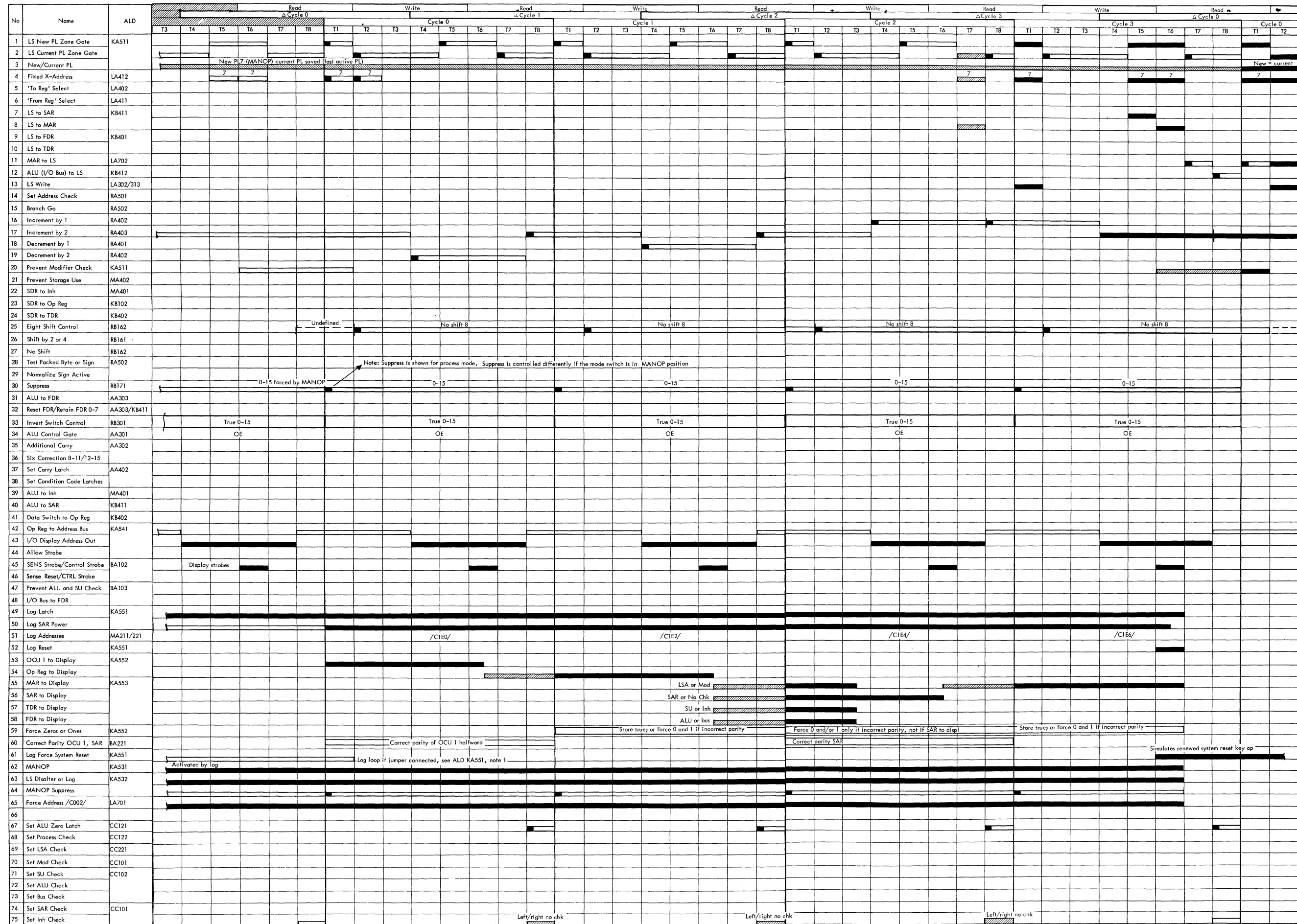




Function signals: ΔCycle Cycle "Do not care" signals: ΔCycle Cycle







READER'S COMMENT FORM

2020 Processing Unit, System/360
Model 20 (Machines with serial no.
50,000 and above), Volume 2 FEMDM

Form SY33-1042-1

How did you use this publication?

- As a reference source
As a class-room text
As a self-study text

Based on your own experience, rate this publication:

- As a reference source—Very Good Good Fair Poor Very Poor
As a text—Very Good Good Fair Poor Very Poor

What is your occupation?

We would appreciate your specific comments; please give page and line references where appropriate. If you wish a reply, be sure to include your name and address.

READER'S COMMENT FORM

2020 Processing Unit, System/360
Model 20 (Machines with serial no.
50,000 and above), Volume 2 FEMDM

Form SY33-1042-1

How did you use this publication?

- As a reference source
As a class-room text
As a self-study text

Based on your own experience, rate this publication:

- As a reference source—Very Good Good Fair Poor Very Poor
As a text—Very Good Good Fair Poor Very Poor

What is your occupation?

We would appreciate your specific comments; please give page and line references where appropriate. If you wish a reply, be sure to include your name and address.

YOUR COMMENTS, PLEASE . . .

This Field Engineering manual is part of a library that serves as a reference source for customer engineers. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

Fold

Fold

FIRST CLASS
PERMIT NO. 1359
WHITE PLAINS, N.Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM CORPORATION
112 EAST POST ROAD,
WHITE PLAINS, N.Y. 10601.



Attention: Department 813 (B)

Fold

Fold



International Business Machines Corporation
Field Engineering Division
112 East Post Road, White Plains, N. Y. 10601

YOUR COMMENTS, PLEASE . . .

This Field Engineering manual is part of a library that serves as a reference source for customer engineers. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

Fold

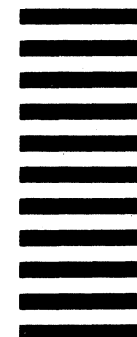
Fold

FIRST CLASS
PERMIT NO. 1359
WHITE PLAINS, N.Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM CORPORATION
112 EAST POST ROAD,
WHITE PLAINS, N.Y. 10601.



Attention: Department 813 (B)

Fold

Fold



International Business Machines Corporation
Field Engineering Division
112 East Post Road, White Plains, N. Y. 10601

Cut along this line

Cut along this line

FE
System
Maintenance
Library

System

CUT HERE

SY33-1042-1

Printed in U.S.A. SY33-1042-1



International Business Machines Corporation
Field Engineering Division
112 East Post Road, White Plains, N.Y. 10601