

[ PREPROCESSOR INSTRUCTION TABLE ]

UNIT INSTRUCTION	Q24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CALL	0	0	0	1	0	0	0	0	RA5	RA4	RA3	(X)	(X)	(X)	(X)	RA2	RA1	RA0	B/A	NA5	NA4	NA3	NA2	NA1	NA0
RET	0	0	0	0	1	0	0	0	0	0	0	(X)	(X)	(X)	(X)	0	0	0	B/A	1	0	0	0	0	0
JUDGE IF RUN#, COM_AD←NA (J2)	0	0	0	0	0	1	0	0	0	0	0	(X)	(X)	(X)	(X)	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
JMP ON r(SA) (J2,J1)	0	0	0	0	0	0	1	0	0	0	0	(X)	(X)	(X)	(X)	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
WAD3 II RS (WAD3(MSB)←(MSB))	0	0	0	0	0	0	0	1	0	0	0	(X)	(X)	(X)	(X)	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
MULTIPLY' (J1)	0	0	0	0	0	0	0	1	0	1	0	(X)	(X)	(X)	(X)	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
MULTIPLY (J1)	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
[RAM]←[RAM#]	0	0	1	RR5	RR4	RR3	RR2	RR1	RR0	0	0	0	R#2	R#1	R#0	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
* [RAM#]←WAD1, WAD1←[RAM]	0	0	1	RR5	RR4	RR3	RR2	RR1	RR0	R#5	R#4	R#3	R#2	R#1	R#0	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
[RAM]←[GR1]+[GR2] (J2)	0	1	0	RR5	RR4	RR3	RR2	RR1	RR0	N10	N20	G11	G10	G21	G20	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
[RAM]←[GR1]-[GR2] (J2,J1)	0	1	1	RR5	RR4	RR3	RR2	RR1	RR0	N10	N20	G11	G10	G21	G20	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
[GR3]←[GR1]+[GR2] (J2)	1	0	0	N10	N20	G11	G10	G21	G20	G31	G30	(X)	(X)	(X)	(X)	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
[GR3]←[GR1]-[GR2] (J2,J1)	1	0	1	N10	N20	G11	G10	G21	G20	G31	G30	(X)	(X)	(X)	(X)	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
[GR4]←[RAM]	1	1	0	RR5	RR4	RR3	RR2	RR1	RR0	G41	G40	(X)	(X)	(X)	(X)	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
[RAM]←[GR4]	1	1	1	RR5	RR4	RR3	RR2	RR1	RR0	G41	G40	(X)	(X)	(X)	(X)	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
r(SA)←r(SA)+N3	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	0	0	N31	N30	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)
r(SA)←N3	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	0	1	N31	N30	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)
SA←N4	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	1	N42	N41	N40	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)
RUN←0	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	0	0	0	0	0	0	0	0	0	0

G1	A
00	N1
01	WAD1
10	WAD2
11	WAD3

G2	B
00	N2
01	WAD1
10	WAD2
11	WAD3

G3/G4	S
00	NONE
01	WAD1
10	WAD2
11	WAD3

N1/N2
0
1

N3	
00	0
01	1
10	2
11	3

N4		N4	
000	0	100	4
001	1	101	5
010	2	110	6
011	3	111	7

	B/A
0	A
1	B

MULTIPLY :

WAD3 II RS (MSB←0), WAD2 II LS (LSB←0)  
WAD3(LSB)=1; WAD1←WAD1+WAD2  
WAD3(LSB)=0; WAD1←WAD1+0  
WAD3IN=1 (J1)

MULTIPLY' :

WAD3 II RS (MSB←0), WAD2 II LS (LSB←0)  
WAD2(MSB)=1 (J1)

WA(MSB); (J2), WAEQ; (J1)。

# 《プリプロセッサ・レジスタ構成》

## 内部 24 ビット・バス

↑↑ IDB8 ~ ↓↓ IDB11 CPU AGDC		↑↑ IDB16 ~ ↓↓ IDB23		↑↑ IDB8 ~ ↓↓ IDB15		↑↑ IDB0 ~ ↓↓ IDB7 CPU AGDC			
dADORG(4)	03 01	EADORG				00	00		
dAD1(4)	07 03	EAD1				04	02		
dAD2(4)	0B 05(W)	EAD2				08	04		
↑レジスタ ↑ 4ビット		PDISPS				0C	06		
↑レジスタ ↑ 5ビット		PDISPD				10	08		
MOPST(5)		23	11(W)	MOD1(4), MOD1(4), PMAX(16)				14	0A
OFST(5)		27	13(W)	PTN P				18	0C
↑レジスタ ↑ 5ビット		STACK				1C	0E		
↑ 16 ビット・レジスタ ↑		D				20	10		
		D1				24	12		
		D2				28	14		
		DD				2C	16		
		DD1				30	18		
		DD2				34	1A		
		EAD3				38	1C		
						--	1E		
		X				40	20		
		Y				42	21		
		DX				44	22		
		DY				46	23		
		XS				48	24		
		YS				4A	25		
		XE				4C	26		
		YE				4E	27		
		XC				50	28		
		YC				52	29		
		a/DH				54	2A		
		b/DV				56	2B		
		PITCHS				58	2C		
		PITCHD				5A	2D		
		STMAX				5C	2E		
		PLANES				5E	2F		
		PTN CNT				60	30		
		XCLMIN				62	31		
		YCLMIN				64	32		
		XCLMAX				66	33		
		YCLMAX				68	34		
		DC				6A	35		
		CLIP(2), MAGH(4), MAGV(4)				6C	36		
						--	--		

↑ RAM構成レジスタ↑

【内蔵レジスタのアドレス対応図】

RR		RR	
000000	EADORG	100000	X
000001	dADORG	100001	Y
000010	EAD1	100010	DX
000011	dAD1	100011	DY
000100	EAD2	100100	XS
000101	dAD2	100101	YS
000110	PDISPS	100110	XE
000111		100111	YE
001000	PDISPD	101000	XC
001001		101001	YC
001010	MOD ETC.	101010	a/DH
001011		101011	b/DV
001100	PTN_P	101100	PITCHS
001101		101101	PITCHD
001110	STACK	101110	STMAX
001111		101111	PLANES
010000	D	110000	PTN_CNT
010001	MOFST	110001	XCLMIN
010010	D1	110010	YCLMIN
010011	OFST	110011	XCLMAX
010100	D2	110100	YCLMAX
010101		110101	DC
010110	DD	110110	CLIPMAGH
010111		110111	FLAGS
011000	DD1	111000	
011001		111001	
011010	DD2	111010	
011011		111011	
011100	EAD3	111100	
011101		111101	
011110		111110	
011111		111111	

プリプロセッサ・レジスタ

R#		R#	
000000	X#	100000	X#
000001	Y#	100001	Y#
000010		100010	XCLMAX
000011		100011	YCLMAX
000100	FLAGS#	100100	FLAGS#
000101		100101	
000110	PTN	100110	PTN
000111		100111	
001000	D1#	101000	
001001	D2#	101001	WR1
001010	D#	101010	WR2
001011	D#	101011	WR3
001100	DD1#	101100	XCLMIN
001101	DD2#	101101	YCLMIN
001110	DD#	101110	XCLMIN
001111	DD#	101111	YCLMIN
010000	EADS	110000	MOD ETC.
010001	EADS1	110001	
010010	EADS2	110010	,,dADD,S
010011	EADSP	110011	
010100	EADD	110100	PLANES
010101	EADD1	110101	
010110	EADD2	110110	PLANE B
010111	EADDP	110111	
011000	PITCHS	111000	<del>PTN</del>
011001		111001	
011010	PDISPS	111010	DUMMY
011011		111011	
011100	PITCHD	111100	CLIPMAGH
011101		111101	
011110	PDISPD	111110	
011111		111111	

描画プロセッサ・レジスタ

【内蔵レジスタの CPUマップ】

00	EADORG	-L	20	D	-L	40	X	-L	60	PTN_CNT	-L
01		-M	21		-M	41		-H	61		-H
02		-H	22		-H	42	Y	-L	62	XCLMINL	-L
03	dADORG		23	MOFST		43		-H	63		-H
04	EAD1	-L	24	D1	-L	44	DX	-L	64	YCLMIN	-L
05		-M	25		-M	45		-H	65		-H
06		-H	26		-H	46	DY	-L	66	XCLMAX	-L
07	dAD1		27	OFST		47		-H	67		-H
08	EAD2	-L	28	D2	-L	48	XS	-L	68	YCLMAX	-L
09		-M	29		-M	49		-H	69		-H
0A		-H	2A		-H	4A	YS	-L	6A	DC	-L
0B	dAD2		2B			4B		-H	6B		-H
0C	PDISPS	-L	2C	DD	-L	4C	XE	-L	6C	MAGH,V	
0D		-H	2D		-M	4D		-H	6D	CLIP	
0E			2E		-H	4E	YE	-L	6E	FLAGS	-L
0F			2F			4F		-H	6F	COMMAND	
10	PDISPD	-L	30	DD1	-L	50	XC	-L	70	FLAGSY	-L
11		-M	31		-M	51		-H	71		-H
12		-H	32		-H	52	YC	-L	72	PITCHY	-L
13			33			53		-H	73		-H
14	PMAX	-L	34	DD2	-L	54	a/DH	-L	74	DAD	-L
15		-H	35		-M	55		-H	75		-M
16	MOD1,MOD0		36		-H	56	b/DV	-L	76		-H
17			37			57		-H	77	WC	
18	PTN_P	-L	38	EAD3	-L	58	PITCHS	-L	78	GCSRX	-L
19		-M	39		-M	59		-H	79		-H
1A		-H	3A		-H	5A	PITCHD	-L	7A	GCSRYS	-L
1B			3B			5B		-H	7B		-H
1C	STACK	-L	3C	STATUS		5C	STMAX	-L	7C	GCSRYE	-L
1D		-M	3D	BANK		5D		-H	7D		-H
1E		-H	3E	IRR	-L	5E	PLANES	-L	7E	SYNC	-L
1F			3F		-H	5F		-H	7F		-H

【描画プロセッサ判定命令種類】

J2	J1
EQSP; SP#-SPE#	D#(MSB);
CadA; dADD←dADD±1, dADS-dADD etc	EQdA; dADS-dADD
PAR(MSB); D1#-D2# etc.	IRRAJ; IRRAH=0FH, IRRAV=0FH
WR(MSB); WR1←WR1-1 etc.	
M1=0FH; MOFST1=0FH	M1(MSB); MOFST1(MSB)
OCA; OFST←OFST+MAGH+1	K5=1; MOFST1←MOFST1+K4,K5=1
CO(MSB); X#←X#-XS# etc.	EQCO; X#-XE# etc.
	EQPM; PMAXC-PMAX
	PS(MSB); PS∞, PS(MSB)
EQPTN; PTNOUT← PTNIN § PTNOUT(CS)	JMP ON r(SA)
JMP ON r(SA)	OFST(MSB); OFST(MSB)
	CLIP;

【プリプロセッサ ROMアドレス指定】

CALL_A ;	1	0	1	0	NA5	NA4	NA3	NA2	NA1	NA0
CALL_B ;	1	0	1	1	NA5	NA4	NA3	NA2	NA1	NA0
SRE_A ;	1	0	0	0	RA5	RA4	RA3	RA2	RA1	RA0
SRE_B ;	1	0	0	1	RA5	RA4	RA3	RA2	RA1	RA0
COMMAND ;	0	0	IDB15	IDB14	IDB13	IDB12	IDB11	IDB10	IDB09	IDB08
ORDINARY ;	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

【描画プロセッサ ROMアドレス指定】

CALL_A ;	1	1	1	0	NA5	NA4	NA3	NA2	NA1	0
CALL_B ;	1	1	1	1	NA5	NA4	NA3	NA2	NA1	0
SRE_A ;	1	0	0	RA6	RA5	RA4	RA3	RA2	RA1	RA0
SRE_B ;	1	0	1	RA6	RA5	RA4	RA3	RA2	RA1	RA0
FLAGS#←WAD1 ;	0	0	DB#23	DB#22	DB#21	DB#20	DB#19	DB#18	DB#17	DB#16
ORDINARY ;	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

[ CPU-プリプロセッサ-描画プロセッサ間のコマンド/レジスタ転送]

コマンド名	プリプロセッサ 開始番地	コマンド・コード下位バイト								描画プロセッサ 開始番地	フラグレジスタ																	
		07	06	05	04	03	02	01	00		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00		
READ_DP READ_COL	004 09C	0	0	0	0	0	0	0	0	NONE 099	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	[描画ポインタ読み出し] [表示メモリ内容読み出し]
DOT_D A_DOT_M R_DOT_M	008 00C 010	ES	0	0	PXEN	BPPX	0	0	0	0F8 (DOT) 095 (ES_DOT)	0	0	ES	X	BPPX	1ST	0	X	X	X	X	X	X	X	X	X	X	[ドット描画]
A_LINE_M0 A_LINE_M1 A_LINE_M2 A_LINE_D0 A_LINE_D1 A_LINE_D2 A_LINE_D3 R_LINE_M0 R_LINE_M1 R_LINE_M2 R_LINE_D0 R_LINE_D1 R_LINE_D2 A_REC R_REC	014 018 01C 020 024 028 02C 030 034 038 03C 040 044 048 04C	ES	IP	0	PXEN	BPPX	ESH	1	0	020 (LINE) 021 (CON_LINE) 0E9 (ES_LINE) 0E5 (CON_ES_LINE)	0	0	ES	RL	BPPX	ESH	1	X	X	X	X	X	XF	SP			[直線描画]  [四辺形描画]	
CRL ARC CSEC ELPS EARC ESEC	050 054 058 05C 060 064	CF	IP	0	PXEN	BPPX	0	0	028 (CRL) 029 (CON_CRL)	0	0	X	RL	BPPX	X	X	CF	X	SPE	MF	XF	SP			[円描画] [円弧描画] [円弧扇形描画] [楕円描画] [楕円弧描画] [楕円弧扇形描画]			
PAINT	068	TL	0	1	SS	<del>PRD</del>	0	0	034 (PAINT)	0	1	SRCHSEL	1ST	PPUSH	1ST	SS	TL	BL	UD	CA	B2	B1	N2	N1	[任意閉領域内塗りつぶし] 二段目表記は PUSH,POP 時			
A_REC_FILL R_REC_FILL A_REC_FILL	08C 090 09E	TL	0	1	SS	WL	WR	0	014 (REC_FIL)	0	1	POL	X	WL	WR	1ST	SS	TL	X	X	X	X	X	X	X	[四辺形内塗りつぶし]		
CRL_FILL ELPS_FILL	050 05C	TL	0	1	SS	WL	WR	0	02C (CRL_FIL)	0	1	POL	RL	WL	WR	1ST	SS	TL	X	X	X	X	XF	SP		[円内塗りつぶし] [楕円内塗りつぶし]		
A_TRI_FILL A_TRA_FILL R_TRA_FILL	06C 070 074	TL	0	1	SS	WL	WR	0	0BC (POL_FIL)	0	1	POL	RL	WL	WR	1ST	SS	TL	X	FL	SPL	X	X	FR	SPR	[三角形内塗りつぶし] [台形内塗りつぶし]		

コマンド名	描画プロセッサ 開始番地	コマンド・コード下位バイト								描画プロセッサ 開始番地	フラグレジスタ																
		07	06	05	04	03	02	01	00		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
A_COPY_EE A_COPY_CE A_COPY_EC A_COPY_CC R_COPY_CC	078	ROT(321)	0	SD_SEL	FAST	0			030 (COPY)	(1) (0) (0) RL	SD_SEL	1ST SL	ROT(321)	(0)	X	X	X	X	[コピー]								
	07C									[1 0 (0) (0) SD_SEL 1 0 ROT(321) 0 0 0 0 0]																	
	080																										
	084	ROT(321)	0	SD_SEL		0	1			008 (90_COPY)	1 0 0 RL	SD_SEL	1ST SL	ROT(321)	0	X	XF	SP	[傾斜コピー]								
	088									[1 0 (0) (0) SD_SEL 1 1 ROT(321) 0 0 XF SP]																	
		ROT(321)	1	SD_SEL		0	0		010 (ES_COPY)	1 0 ES RL	SD_SEL	1ST	1STM	ROT(21)	0	1STV	0	ESHESV	[任意倍率拡大縮小コピー]								
		ESH ROT(21)ESV		SD_SEL		1	1			[1 0 1 0 SD_SEL 1 1 ROT(21) 0 0 0 ESHESV]																	
		ESH ROT(21)ESV		SD_SEL		1	0		06C (FR_COPY)	0 0 ES RL	SD_SEL	ESHESV	X	XFS	SPS	BUG	XF	SP	[任意角回転/任意倍率 拡大縮小コピー]								
		ESH ROT(21)ESV		SD_SEL		1	0			[0 0 1 0 SD_SEL ESH ESV 0 XFS SPS BUG XF SP]																	
PUT/GET_E PUT/GET_C	094 098	0	ROT(21)	0	SD_SEL	1	PUT		01C (PUT)	1 1 0 RL	SD_SEL	1STPUT	0	ROT(21)	0	X	FULL	X	EMPT	[ブット/ゲット]							
										[1 1 0 0 SD_SEL 1 PUT 0 ROT(21) 0 1 0 0 1]																	
PUT-E GET-E PUT-C GET-C	094 096 098 09A	0	ROT(21)	1	SD_SEL	1	0		018 (GET)	1 0 0 RL	SD_SEL	1STPUT	0	ROT(21)	1	0	FULL	1ST#EMP	[90度回転ゲット]								
										[1 0 0 0 SD_SEL 1 0 0 ROT(21) 1 0 0 1 1]																	
									004	(POL_FIL_DUMMY);	XE#←X#,	PTN_CNT←WR3															
									024	(CRL_DUMMY);	XE#←X#,	YE#←Y#															
									088	(REC_FIL_DUMMY);	PTN_CNT←WR3																
									0FC	(READ1_DUMMY);	X#←→XS#,	Y#←→YS#															
									???	(PAINT_READ_COL);																	

SD\_SEL = 00 = MS MOD 0  
01 = MS MOD 1

(0),(1) はプリプロセッサによる設定無し。描画プロセッサによる設定を意味する。

[描画プロセッサ・フラグの意味合い]

- (1) プロセッサ・エラー . . . . . FLAG15・14・13=[111]
- (2) 一般ハードウェア制御 . . . . . FLAG15・14 = [00] → DOT, LINE, REC, CRL, ELPS, FR\_COPY  
 FLAG15・14 = [01] → PAINT, REC\_FILL, CRL\_FILL, POL\_FILL  
 FLAG15・14 = [10] → COPY, 90\_COPY, ES\_COPY  
 FLAG15・14 = [11] → PUT, GET, 90\_GET
- (3) (±) 命令制御 . . . . . FLAG12 = [0] → FLAG7・6・5・4 を参照し、±制御を実行  
 FLAG12 = [1] → FLAG2・1・0 を参照し、±制御を実行

099  
075  
079  
(LIFE?? DUMMY)  
DUMMY

[描画プロセッサ命令表]

Q42 ----- Q32	Q31 ---- Q28	Q27 ----- Q22	Q21 ----- Q17	Q16 ----- Q10	Q09 ----- Q00
[FIELD A] EAD, MASK ETC.	[FIELD B] PMA, MASK	[FIELD C] PTN, WR, CORD	[FIELD D] r(SA)	[FIELD E] FLEXIBLE OPERATION	[FIELD F] NEXT ADDRESS

\* FIELD A \*

UNIT INSTRUCTION	Q42	41	40	39	38	37	36	35	34	33	32
\$ [EA1]← [R#1]	0	0	0	R12	R11	R10	D/S	E11	E10	0	1
\$ [R#1]← [EA1]	0	0	0	R12	R11	R10	D/S	E11	E10	1	0
MASK← [N1]	0	0	1	0	0	1	X	N12	N11	N10	*PT
[EA5]← [EA1] + [EA2]	0	0	1	RW1	RW0	0	D/S	E11	E10	E21	E20
[EA5]← [EA1] - [EA2]	0	0	1	RW1	RW0	1	D/S	E11	E10	E21	E20
[EA3]← [EA1]	RW1	RW0	E33	E32	E31	E30	D/S	E11	E10	X	1
[EA3]← [EA1]±[EA4]	RW1	RW0	E33	E32	E31	E30	D/S	E11	E10	E40	0

E3, D/S	EA3	D/S, E1	EA1	D/S, E2	EA2	D/S, E4	EA4	D/S, E5	EA5	R1	N1	RW	
1XX 0	EADOUT, EADS	0 00	EADS	X 00	"5"	X 0	"1"	0 00	EADOUT, EADS	000	D1#	000	0
X1X 0	EADOUT, EADS1	0 01	EADS1	X 01	"1"	0 1	PITCHS	0 01	EADOUT, EADS1	001	D2#	001	1
XX1X 0	EADOUT, EADS2	0 10	EADS2	0 10	PITCHS	1 1	PITCHD	0 10	EADOUT, EADS2	010	D#	010	2
XXX1 0	EADOUT, EADSP	0 11	EADSP	0 11	PDISPS			0 11	EADOUT, EADSP	011	D#	011	3
1XX 1	EADOUT, EADD	1 00	EADD	1 10	PITCHD			1 00	EADOUT, EADD	100	DD1#	100	4
X1X 1	EADOUT, EADD1	1 01	EADD1	1 11	PDISPD			1 01	EADOUT, EADD1	101	DD2#	101	5
XX1X 1	EADOUT, EADD2	1 10	EADD2					1 10	EADOUT, EADD2	110	DD#	110	6
XXX1 1	EADOUT, EADDP	1 11	EADDP					1 11	EADOUT, EADDP	111	DD#	111	7



\* FIELD B \*

UNIT INSTRUCTION	Q31	30	29	28	MR2	N1
PMAXC-PMAX EQPM (J1)	0	0	0	1	00	*dADD* (*MCFST*)
A10←[N1]	0	0	1	N10	01	dADD
JMP ON r(SA) (J2, J1)	0	1	0	0	10	"0"
'WR1←WR1-MAGH-1 (-16)' (J2)	0	1	0	1	11	"OFH"
D#(MSB) (J1)	0	1	1	0		
/D#(MSB) (J1)	0	1	1	1		
MASKA←[MR2]	1	0	M21	M20		
MASKB←[MR2]	1	1	M21	M20		

ES\_COPY; \*MORST\*



\* FIELD C \*

UNIT INSTRUCTION	Q27	26	25	24	23	22	(21)	(20)
XCLMIN-X#,X#-XCLMAX	0	0	0	0	0	1		
YCLMIN-Y#,Y#-YCLMAX, CLIP (J1)	0	0	0	0	1	0		
dADD←dADD+1	0	0	0	1	0	0		
dADS←dADD (J2,J1)	0	0	0	1	0	1		
dADD←dADD±1 CadA (J2)	0	0	0	1	1	0		
dADS←dADS±1 CadA (J2)	0	0	0	1	1	1		
PS∞	0	0	1	0	0	0		
PS∞ PS(MSB) (J1)	0	0	1	0	0	1		
PS←0	0	0	1	0	1	X		
WR1←DH	0	0	1	1	0	0		
WR1←WR2	0	0	1	1	0	1		
WR2←DV	0	0	1	1	1	0		
WR3←PTN_CNT	0	0	1	1	1	1		
*2 WR1←WR1-MAGH-1 (-16) (J2)	0	1	0	0	0	0		
WR1←WR1-1 WR(MSB) (J2)	0	1	0	0	0	1		
WR2←WR2-1 WR(MSB) (J2)	0	1	0	0	1	0		
WR3←WR3-1 WR(MSB) (J2)	0	1	0	0	1	1		
*1 WR1←WR1+*dADD* (dADD)	0	1	0	1	0	0		
WR1←WR1+1	0	1	0	1	0	1		
WR2←WR2+1	0	1	0	1	1	0		
WR3←WR3+1	0	1	0	1	1	1		
\$V WR1←X#	0	1	1	0	0	0		
WR1←0	0	1	1	0	0	1		
WR2←0	0	1	1	0	1	0		
WR3←0	0	1	1	0	1	1		
DH←WR1	0	1	1	1	0	0		
DV←WR2	0	1	1	1	0	1		
PTN_CNT←WR3	0	1	1	1	1	0		
WR3←WR3+K4	0	1	1	1	1	1		
\$V MPTN←IRSR	1	0	0	0	0	0		
\$V PTN←IRSR	1	0	0	0	0	1		
PTN←PTNIN	1	0	0	0	1	0		
PTN←PTNOUT	1	0	0	0	1	1		
\$V IRSR(C)←PTNOUT	1	0	0	1	0	0		
\$V IRSR(O)←PTN	1	0	0	1	0	1		
PTNOUT←PTNIN	1	0	0	1	1	0		
PTN←MPTN	1	0	0	1	1	1		
PTNOUT← PTNIN \$ [DR2]	1	0	1	D22	D21	D20		
X#←[X1]	1	1	0	0	0	X10		
X#←→[X1]	1	1	0	0	1	X10		
X#-[X1] (J2) EQCO (J1)	1	1	0	1	0	X10		
[X1]←X#	1	1	0	1	1	X10		
Y#←[Y1]	1	1	1	0	0	Y10		
Y#←→[Y1]	1	1	1	0	1	Y10		
Y#-[Y1] (J2) EQCO (J1)	1	1	1	1	0	Y10		
[Y1]←Y#	1	1	1	1	1	Y10		

X1		Y1	
0	XS#	0	YS#
1	XE#	1	YE#

D2	ORDINARY COND.	MS_COPY	FIG	GET/PUT & FL1
000	PTNOUT	PTNOUT (M1) *		
001	PTNOUT(BC)	PTNOUT (INV) *		
010	PTNOUT(CS) EQPTN (J2) *	<del>PTNOUT (M1) *</del>		
011	( PTNOUT←1 ) *	PTNOUT (M1) *		
100	PTN	PTN (M1)	PTN (FIG)	PTNOUT←PTN * %
101	IRSR	IRSR (M1)		PTNOUT←IRSR * %
110	IRRR	IRRR (M1)		¥ PTNOUT←IRRR * %
111	MPTN	MPTN (M1)		① PTNOUT←MPTN * %

\*; MASK←1, ¥; SHIFT COUNT = 0, ①: 三外高起動, %; ZCHG 両起動

MS?

#1 COPY & PUT/GET=1; \*dADD\*  
#2 (ES\_COPY & ESH=1)=1; MAGH

\* FIELD D \*

UNIT INSTRUCTION	Q21	20	19	18	17 (16)
$r(SA) \leftarrow r(SA) + N3$	0	0	0	N31	N30
$r(SA) \leftarrow N3$	0	0	1	N31	N30
$SA \leftarrow N4$	0	1	N42	N41	N40
SP# = SPE# EQSP (J2) (MOD/COL ← PLANES/B)	1	0	0	0	0
'JMP ON r(SA)'	1	0	0	0	1
'XCLMIN-X#,X#-XCLMAX'	1	0	0	1	0
'YCLMIN-Y#,Y#-YCLMAX', CLIP (J1)	1	0	0	1	1
$X# \leftarrow X# - [X1]$ (J2, J1)	1	0	1	0	X10
$Y# \leftarrow Y# - [Y1]$ (J2, J1)	1	0	1	1	Y10
$X# \leftarrow X# \pm 1$	1	1	0	0	0
$X# \leftarrow X# \pm 16$	1	1	0	0	1
$Y# \leftarrow Y# \pm 1$	1	1	0	1	0
$Y# \leftarrow Y# \pm 16$	1	1	0	1	1
$X# \leftarrow X# + 16$	1	1	1	0	0
$X# \leftarrow X# - 16$	1	1	1	0	1
$Y# \leftarrow Y# + 1$	1	1	1	1	0
$Y# \leftarrow Y# - 1$	1	1	1	1	1

N3	
00	"0"
01	"1"
10	"2"
11	"3"

N4	
000	"0"
001	"1"
010	"2"
011	"3"
100	"4"
101	"5"
110	"6"
111	"7"

X1	
0	XS#
1	XE#

Y1	
0	YS#
1	YE#

$\{ r(SA) \leftarrow r(SA) + 0 \}$  27118.  
 00H

\* FIELD E (NO.1) \*

UNIT INSTRUCTION	Q16	15	14	13	12	11	10 (09)	(08)
MOD/COL←PLANES/B	0	0	0	0	0	1	X	
dADD,B1,N1←B_ROM	0	0	0	0	1	0	X	
dADS,B2,N2←B_ROM	0	0	0	0	1	1	X	
SA←N4	0	0	0	1	X	X	X	
\$v [PAINT]←PTNIN (POP)	0	0	1	0	PA2	PA1	PA0	
\$v PTNOUT←[PAINT] (PUSH)	0	0	1	1	PA2	PA1	PA0	
OFST←OFST'	0	1	0	0	0	0	0	
OFST'←OFST	0	1	0	0	0	0	1	
*1 OFST←OFST+MAGH+1 (OFST+16) OCA (J2)	0	1	0	0	0	1	0	
OFST←OFST±1	0	1	0	0	0	1	1	
OFST(MSB) (J1)	0	1	0	0	1	0	0	
MOFST←MOFST'	0	1	0	1	0	0	0	
MOFST←MOFST1+1	0	1	0	1	0	0	1	
MOFST1←MOFST+K4, K5=1 (J1)	0	1	0	1	0	1	0	
MOFST1=OFH (J1),MOFST1(MSB)=1 (J2)	0	1	0	1	1	0	0	
IRRAH=OFH IRRAJ (J1)	0	1	1	0	0	0	0	
IRRAH←#dADD#	0	1	1	0	0	0	1	
IRRAV=0 IRRAJ (J1)	0	1	1	0	0	1	0	
IRRAV←#dADD#	0	1	1	0	0	1	1	
IRRR←IRR(H),IRRAV←IRRAV+1	0	1	1	0	1	0	0	
\$v IRR(H)←PTNOUT,IRRAH←IRRAH+1	0	1	1	0	1	0	1	
IRRR←IRR(V),IRRAV←IRRAV+1	0	1	1	0	1	1	0	

\*1 (ES\_COPY & ESH=1); MAGH, ES\_COPY; OCA

PA	(PUSH)
000	EADS2(L)
001	UD,CA,BL,WR1
010	UD,CA,BL,WR2
011	(UD,CA,BL,WR3)
100	X#
101	Y#
110	EADDP(L),dADD,dADS
111	EADDP(H)

PA	(POP)
000	EADS2(L)
001	UD,CA,BL,WR1
010	(UD,CA,BL,WR2)
011	(UD,CA,BL,WR3)
100	X#
101	Y#
110	EADDP(L),dADD,dADS
111	EADDP(H)

N4	
000	"0"
001	"1"
010	"2"
011	"3"
100	"4"
101	"5"
110	"6"
111	"7"

PUSH/POP SEQUENCE
EADDP(L),dADD,dADS
EADDP(H)
UD,CA,BL,WR1
UD,CA,BL,WR2
X#
Y#
EADS2(L)

\* FIELD E (NO.2) \*

UNIT INSTRUCTION	Q16	15	14	13	12	11	10	(09)	(08)	(07)	(06)	(05)	(04)
D1# ← D1# + 2*DD#	1	0	0	0	0	1	0						0
D1# ← D1# + 2*DD1#	1	0	0	0	0	1	1						0
D# ← D# + D1#	1	0	0	1	0	0	0						0
D# ← D# + D2#	1	0	0	1	0	0	1						0
D# ← D# + DD#	1	0	0	1	0	1	0						0
D# ← D# + DD1#	1	0	0	1	0	1	1						0
D# ← D# + D2#/2	1	0	0	1	1	0	1						0
D# ← D# + DD1#/2	1	0	0	1	1	1	1						0
DD# ← DD# + DD2#	1	0	1	1	0	0	1						0
DD# ← DD# + DD1#	1	0	1	1	0	1	1						0
D1# - D2#	1	0	0	0	0	0	1						1
D2# - 1	1	0	0	0	1	0	0						1
D2# ← D2# - 2*DD#	1	0	0	0	1	1	0						1
D2# ← D2# - 2*DD1#	1	0	0	0	1	1	1						1
D# ← D# - D1#	1	0	0	1	0	0	0						1
D# ← D# - D2#	1	0	0	1	0	0	1						1
D# ← D# - D1#/2	1	0	0	1	1	0	0						1
D# ← D# - D2#/2	1	0	0	1	1	0	1						1
D# ← D# - DD#/2	1	0	0	1	1	1	0						1
DD1# - DD2#	1	0	1	0	0	0	1						1
DD# ← DD# - 2*#	1	0	1	1	1	0	0						1
													(NA6)
[dAD] ← [dAD]	1	1	0	dA3	dA2	dA1	dA0						

UNIT INSTRUCTION	Q16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CALL	1	1	1	0	RA6	RA5	RA4	RA3	RA2	RA1	B/A	NA5	NA4	NA3	NA2	NA1	RA0
SRE	1	1	1	1	0	0	0	0	0	0	B/A	0	0	0	0	0	0

all 0's

B/A		dAD		dAD	
0	A	0000	dADD ← dADS	1000	dADS ← dADD
1	B	0001	dADD ← dADDP	1001	dADS ← dADS1
		0010	dADD ← dADD1	1010	dADS ← dADS2
		0011	dADD ← dADD2	1011	dADS ← dADS2/2
		0100	dADDP ← dADS	1100	dADS1 ← dADD
		0101	dADDP ← dADD	1101	dADS1 ← dADS
		0110	dADD1 ← dADD	1110	dADS2 ← dADS
		0111	dADD2 ← dADD	1111	dADS ← dADS + dADS2 + 1 CadA (J2)

\* FIELD F \*

	09	08	07	06	05	04	03	02	01	00
NEXT ADDRESS ASSIGNMENT	NA9	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0

【 D# 系演算器命令デコード法】

16	15	14	13	12	11	10	9		
1	0	DD/D, AREG			BREG			S/A	
X	X	0	0	0	X	X	X	D1#_SEL	
X	X	0	0	1	X	X	X	D2#_SEL	
X	X	0	1	0	X	X	X	D#_SEL	
X	X	0	1	1	X	X	X	D#_SEL, HALF	
X	X	1	0	0	X	X	X	(DD1#)	
X	X	1	0	1	X	X	X	(DD2#)	
X	X	1	1	0	X	X	X	(DD#)	
X	X	1	1	1	X	X	X	(DD#)	
X	X	X	1	X	0	0	X	DB←D1#	
X	X	0	X	X	0	1	X	DB←D2#	
X	X	X	X	X	1	0	X	DB←DD#	
X	X	X	X	X	1	1	X	DB←DD1#	
X	X	1	X	X	0	1	X	DB←DD2#	
X	X	X	0	X	0	0	X	DB←1	
X	X	X	0	X	0	X	X	TRIAL OPE.	
X	X	0	0	X	1	X	X	DOUBLE	

【 X# 系命令デコード法】

27	26	25	24	23	22	
1	1	Y/X	0	X	X	X, Y ← B
1	1	Y/X	X	X	X	A ← X, Y
1	1	Y/X	X	1	0	XS, YS ← A
1	1	Y/X	X	X	0	B ← XS, YS
1	1	Y/X	X	1	1	XE, YE ← A
1	1	Y/X	X	X	1	B ← XE, YE
1	1	X	1	0	X	S/A, EQCO JUDGE (J1)
1	0	0	0	1	Y/X	B ← XCM, YCM: S/A (BOTH WR & X# OP.)
1	0	0	0	1	1	CLIP (J1)

21	20	19	18	17	
1	0	1	X	X	±
1	1	X	Y/X	X	┌───┐
1	X	1	Y/X	X	
1	1	X	X	X	CaI ← 1
1	0	1	Y/X	0	B ← XS, YS
1	0	1	Y/X	1	B ← XE, YE
1	1	0	X	1	┌───┐
1	1	1	0	X	
1	0	0	1	Y/X	B ← XCM, YCM: S/A (BOTH WR & X#)
1	0	0	1	1	CLIP (J1)
1	0	1	X	X	S/A CaCO (J2), EQCO (J1)
1	1	1	X	1	S/A

【論理演算部論理】

<p>REPLACE; MOD=0000 <math>D \leftarrow \underline{S}</math>            0001 <math>D \leftarrow S</math>            0010 <math>D \leftarrow 0</math>            0011 <math>D \leftarrow 1</math></p>	<p>MASK = 1; <math>D \leftarrow S</math>            MASK ≠ 1; <math>D \leftarrow D</math></p> <p><math>S \leftarrow 0</math></p>
<p>EXOR; MOD=0100 <math>D \leftarrow D (+) \underline{S}</math>            0101 <math>D \leftarrow \underline{D} (+) \underline{S}</math>            0110 <math>D \leftarrow D</math>            0111 <math>D \leftarrow D</math></p>	<p>MASK · S = 1; <math>D \leftarrow \bar{D}</math>            MASK · S ≠ 1; <math>D \leftarrow D</math></p> <p><math>S \leftarrow 0</math></p>
<p>AND; MOD=1000 <math>D \leftarrow D \cdot \bar{S}</math>            1001 <math>D \leftarrow \underline{D} \cdot \underline{S}</math>            1010 <math>D \leftarrow \underline{D} \cdot S</math>            1011 <math>D \leftarrow D \cdot S</math></p>	<p>MASK · S = 1; <math>D \leftarrow 0</math>            MASK · S ≠ 1; <math>D \leftarrow D</math></p>
<p>OR; MOD=1101 <math>D \leftarrow D + \underline{S}</math>            1101 <math>D \leftarrow \underline{D} + S</math>            1110 <math>D \leftarrow \underline{D} + \underline{S}</math>            1111 <math>D \leftarrow D + S</math></p>	<p>MASK · S = 1; <math>D \leftarrow 1</math>            MASK · S ≠ 1; <math>D \leftarrow D</math></p>

$$D \leftarrow D \ \& \ S$$


---

$PTNOUT \leftarrow PTNIN \ \& \ PTNOUT$   
 $PTNOUT \leftarrow PTNIN \ \& \ PTN$   
 $PTNOUT \leftarrow PTNIN \ \& \ IRSR$   
 $PTNOUT \leftarrow PTNIN \ \& \ IRRR$   
 $PTNOUT \leftarrow PTNIN \ \& \ MPTN$



【フラグレジスタ内容により実行形態が変化する描画プロセッサ命令】

MASKA←*dADD* (*MOFST*) MASKB←*dADD* (*MOFST*)	FL15=0; ASSUME ROT=0000 FL15=1; ASSUME ROT (?) FL15=1 & FL14=0 & FL13=1; (*MOFST*) OTHERS ; (*dADD*)
WR1←WR1+(*dADD*) (dADD)	FL15=1; (*dADD*) FL15=0; (dADD)
WR1←WR1-(MAGH+1) (16)	FL15=1 & FL14=0 & FL13=1 & FL01=1; (MAGH+1) OTHERS ; (16)
PTNOUT← PTNIN § PTNOUT	FL15=1 & FL14=0 & FL11=0 & FL10=1; PTNOUT← PTNIN § PTNOUT (MO) OTHERS; PTNOUT← PTNIN § PTNOUT
PTNOUT← PTNIN § PTN	FL15=1 & FL14=0 & FL11=0 & FL10=1; PTNOUT← PTNIN § PTN (M1) FL15=0 & FL14=0; PTNOUT← PTNIN § PTN (FIG) OTHERS; PTNOUT← PTNIN § PTN
PTNOUT← PTNIN § IRSR	FL15=1 & FL14=0 & FL11=0 & FL10=1; PTNOUT← PTNIN § IRSR (M1) OTHERS; PTNOUT← PTNIN § IRSR
PTNOUT← PTNIN § IRRR	FL15=1 & FL14=0 & FL11=0 & FL10=1; PTNOUT← PTNIN § IRRR (M1) OTHERS; PTNOUT← PTNIN § IRRR
PTNOUT← PTNIN § MPTN	FL15=1 & FL14=0 & FL11=0 & FL10=1; PTNOUT← PTNIN § MPTN (M1) OTHERS; PTNOUT← PTNIN § MPTN
OFST←OFST+(MAGH+1) (16) , OCA	FL15=1 & FL14=0 & FL13=1 & FL01=1; (MAGH+1) OTHERS ; (16) FL15=1 & FL14=0 & FL13=1 ; (OCA JUDGE) OTHERS ; (NONE)
IRRAH←*dADD* IRRAV←*dADD*	

FL15, 14	
0 0	FIG (DOT,LINE,REC,ELPS,FR_COPY)
0 1	FILL (PAINT,REC_FILL,CRL_FILL,POL_FILL)
1 0	COPY (COPY,90_COPY,ES_COPY)
1 1	PUT/GET (PUT,GET,90_GET)

FL13; ES  
FL01; ESH

【 KROM の使用法】

K1(4); 水平拡大時の OFST 量を表す。MAGH に等しい (演算時には MAGH+1 )。

その他の場合には、0FHを与える。

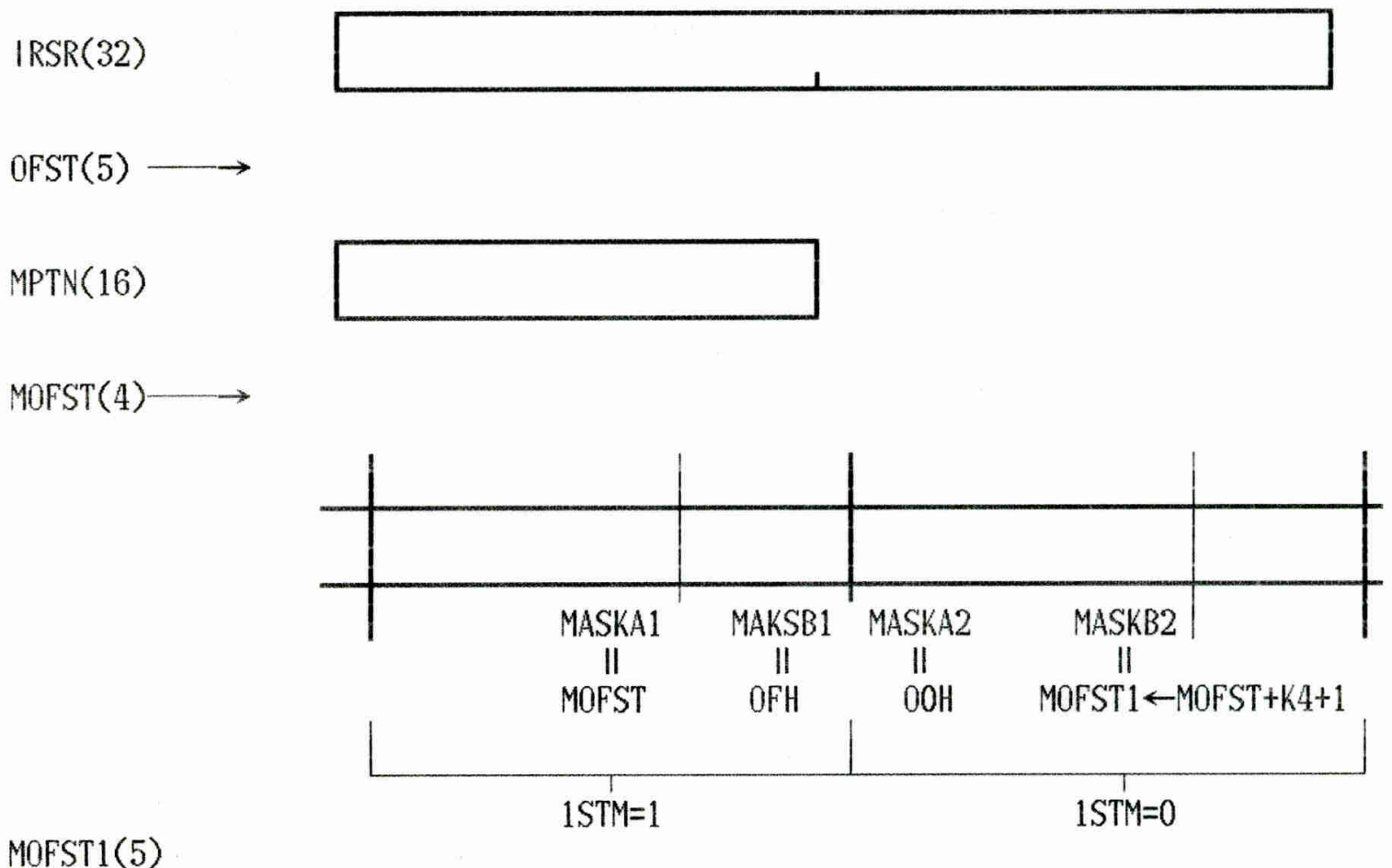
K4(4); 1ライン上での最終ワード描画において、あと何ドット描画可能であることを表す。

縮小時、K5(1)=1の場合には描画実行を停止する。

K5(1); 描画プロセッサの条件判定入力に接続される。

「1」のとき、描画実行を停止する。

【OFST、MOFST、MOFST1の関係】





(3) K4; 拡大時

θ  
WR1(15-4)=~~1~~ のときの WR1(3-0)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
1	7	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
2	4	A	F	F	F	F	F	F	F	F	F	F	F	F	F	F
3	3	7	B	F	F	F	F	F	F	F	F	F	F	F	F	F
4	2	5	9	C	F	F	F	F	F	F	F	F	F	F	F	F
5	2	4	7	A	C	F	F	F	F	F	F	F	F	F	F	F
6	1	4	6	8	A	D	F	F	F	F	F	F	F	F	F	F
7	1	3	5	7	9	B	D	F	F	F	F	F	F	F	F	F
8	1	3	4	6	8	A	B	D	F	F	F	F	F	F	F	F
9	1	2	4	5	7	9	A	B	D	F	F	F	F	F	F	F
A	0	2	3	5	6	8	9	B	C	E	F	F	F	F	F	F
B	0	2	3	4	6	7	8	A	B	D	E	F	F	F	F	F
C	0	1	3	4	5	6	8	9	A	B	D	E	F	F	F	F
D	0	1	2	4	5	6	7	8	9	A	C	D	E	F	F	F
E	0	1	2	3	4	5	6	8	9	A	B	C	D	E	F	F
F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

MAGH

DH(15-4)≠0

のとき

K4=0FH

(4) K4; 縮小時

θ  
WR1(15-4)=~~1~~ のときの WR1(3-0)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	K	K	K	K	K	K	K	0	0	0	0	0	0	0	0	0
1	K	K	K	0	0	0	0	0	0	0	0	1	1	1	1	1
2	K	K	0	0	0	0	0	1	1	1	1	1	1	2	2	2
3	K	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3
4	K	0	0	0	1	1	1	2	2	2	2	3	3	3	4	4
5	K	0	0	1	1	1	2	2	2	3	3	4	4	4	5	5
6	K	0	0	1	1	2	2	3	3	3	4	4	5	5	6	6
7	0	0	0	1	2	2	3	3	4	4	5	5	6	6	7	7
8	0	0	1	1	2	2	3	4	4	5	5	6	6	7	7	8
9	0	0	1	2	2	3	3	4	5	5	6	7	7	8	8	9
A	0	0	1	2	2	3	4	5	5	6	7	7	8	9	9	A
B	0	1	1	2	3	4	4	5	6	7	7	8	9	A	A	B
C	0	1	1	2	3	4	5	6	6	7	8	9	A	A	B	C
D	0	1	2	3	3	4	5	6	7	8	9	A	A	B	C	D
E	0	1	2	3	4	5	6	7	7	8	9	A	B	C	D	E
F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

MAGH

DH(15-4)≠0

のとき

K4=MAGH

【 N/16 倍縮小 / 16/N 倍拡大 入出力対応表】

縮小時の算式 ;  $DOT \times (16/N)$  の四捨五入

ESH	MAGH	OUTPUT DATA BUS INTERCONNECTION																MAG
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	0;0000	0	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	1/16	
0	1;0001	0	8	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	(8)	2/16	
0	2;0010	0	5	B	(C)	(C)	(C)	(C)	(C)	(C)	(C)	(C)	(C)	(C)	(C)	(C)	3/16	
0	3;0011	0	4	8	C	(C)	(C)	(C)	(C)	(C)	(C)	(C)	(C)	(C)	(C)	(C)	4/16	
0	4;0100	0	3	6	A	D	(D)	(D)	(D)	(D)	(D)	(D)	(D)	(D)	(D)	(D)	5/16	
0	5;0101	0	3	5	8	B	D	(D)	(D)	(D)	(D)	(D)	(D)	(D)	(D)	(D)	6/16	
0	6;0110	0	2	5	7	9	B	E	(E)	(E)	(E)	(E)	(E)	(E)	(E)	(E)	7/16	
0	7;0111	0	2	4	6	8	A	C	E	(E)	(E)	(E)	(E)	(E)	(E)	(E)	8/16	
0	8;1000	0	2	4	5	7	9	B	C	E	(E)	(E)	(E)	(E)	(E)	(E)	9/16	
0	9;1001	0	2	3	5	6	8	A	B	D	E	(E)	(E)	(E)	(E)	(E)	10/16	
0	A;1010	0	1	3	4	6	7	9	A	C	D	F	(F)	(F)	(F)	(F)	11/16	
0	B;1011	0	1	3	4	5	7	8	9	B	C	D	F	(F)	(F)	(F)	12/16	
0	C;1100	0	1	2	4	5	6	7	9	A	B	C	E	F	(F)	(F)	13/16	
0	D;1101	0	1	2	3	5	6	7	8	9	A	B	D	E	F	(F)	14/16	
0	E;1110	0	1	2	3	4	5	6	7	9	A	B	C	D	E	F	(F)	15/16
0	F;1111	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	16/16
ESH	MAGH	OUTPUT DATA BUS INTERCONNECTION																MAG
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
1	F;1111	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	16/16
1	E;1110	0	1	2	3	4	5	6	7	7	8	9	A	B	C	D	E	16/15
1	D;1101	0	1	2	3	3	4	5	6	7	8	9	A	A	B	C	D	16/14
1	C;1100	0	1	2	2	3	4	5	6	6	7	8	9	A	A	B	C	16/13
1	B;1011	0	1	1	2	3	4	4	5	6	7	7	8	9	A	A	B	16/12
1	A;1010	0	1	1	2	3	3	4	5	5	6	7	7	8	9	9	A	16/11
1	9;1001	0	0	1	2	2	3	4	4	5	5	6	7	7	8	9	9	16/10
1	8;1000	0	0	1	1	2	3	3	4	4	5	5	6	7	7	8	8	16/9
1	7;0111	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	16/8
1	6;0110	0	0	1	1	1	2	2	3	3	4	4	5	5	5	6	6	16/7
1	5;0101	0	0	0	1	1	2	2	2	3	3	3	4	4	5	5	5	16/6
1	4;0100	0	0	0	1	1	1	2	2	2	2	3	3	3	4	4	4	16/5
1	3;0011	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	16/4
1	2;0010	0	0	0	0	0	1	1	1	1	1	1	2	2	2	2	2	16/3
1	1;0001	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	16/2
1	0;0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16/1

拡大時の算式 ;  $(16/N) \times N'$  の四捨五入 (N'; 整数)

【繰り返し／間引き回数制御】

MAGV	↓	→																														↓		
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1
1	0	2	4	6	8	A	C	E	0	2	4	6	8	A	C	E	0	2	4	6	8	A	C	E	0	2	4	6	8	A	C	E	0	2
2	1	4	7	A	D	0	3	6	9	C	F	2	5	8	B	E	1	4	7	A	D	0	3	6	9	C	F	2	5	8	B	E	1	7
3	1	5	9	D	1	5	9	D	1	5	9	D	1	5	9	D	1	5	9	D	1	5	9	D	1	5	9	D	1	5	9	D	1	5
4	2	7	C	1	6	B	0	5	A	F	4	9	E	3	8	D	2	7	C	1	6	B	0	5	A	F	4	9	E	3	8	D	2	7
5	2	8	E	4	A	0	6	C	2	8	E	4	A	0	6	C	2	8	E	4	A	0	6	C	2	8	E	4	A	0	6	C	2	8
6	3	A	1	8	F	6	D	4	B	2	9	0	7	E	5	C	3	A	1	8	F	6	D	4	B	2	9	0	7	E	5	C	3	A
7	3	B	3	B	3	B	3	B	3	B	3	B	3	B	3	B	3	B	3	B	3	B	3	B	3	B	3	B	3	B	3	B	3	B
8	4	D	6	F	8	1	A	3	C	5	E	7	0	9	2	B	4	D	6	F	8	1	A	3	C	5	E	7	0	9	2	B	4	D
9	4	E	8	2	C	6	0	A	4	E	8	2	C	6	0	A	4	E	8	2	C	6	0	A	4	E	8	2	C	6	0	A	4	E
A	5	0	B	6	1	C	7	2	D	8	3	E	9	4	F	A	5	0	B	6	1	C	7	2	D	8	3	E	9	4	F	A	5	0
B	5	1	D	9	5	1	D	9	5	1	D	9	5	1	D	9	5	1	D	9	5	1	D	9	5	1	D	9	5	1	D	9	5	1
C	6	3	0	D	A	7	4	1	E	B	8	5	2	F	C	9	6	3	0	D	A	7	4	1	E	B	8	5	2	F	C	9	6	3
D	6	4	2	0	E	C	A	8	6	4	2	0	E	C	A	8	6	4	2	0	E	C	A	8	6	4	2	0	E	C	A	8	6	4
E	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8	7	6
F	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

初期値作成;  $dADS2=MAGV$  とし、  $dADS \leftarrow dADS2/2$  を実行。

演算;  $dADS, dADS1=$  初期値、  $dADS2=MAGV$  とし

$dADS \leftarrow dADS+dADS2+1$  を実行。

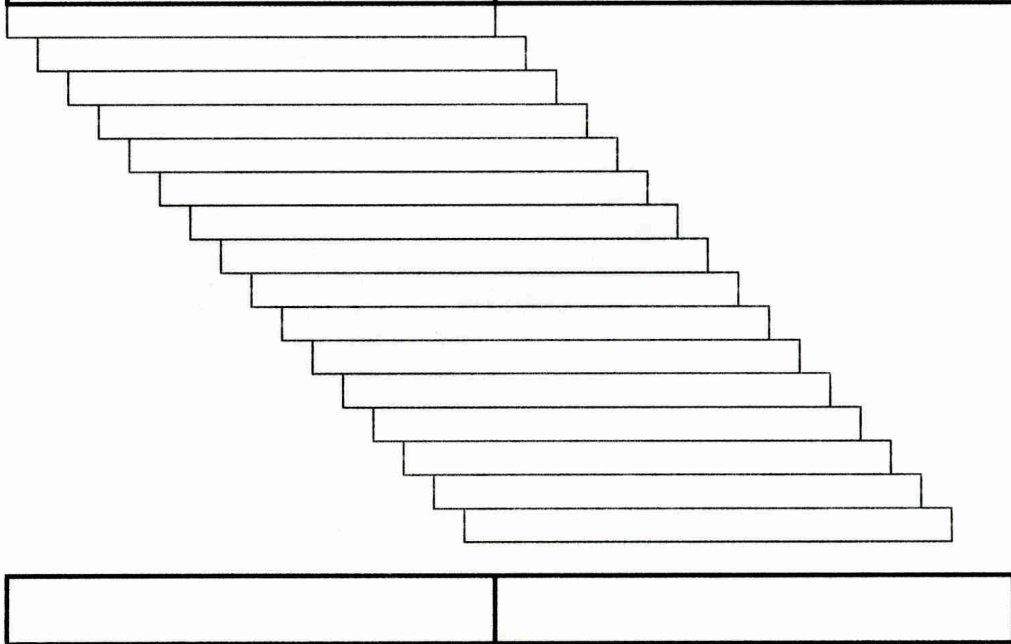
下線部で示す桁上げ信号発生までが「繰り返し／間引き回数」となる。

【 32 ビット・シフト】

OFST	BITS CONNECTED WITH DB#0 ~ DB#15							
0	1111	1111	1111	1111	0000	0000	0000	0000
1	0111	1111	1111	1111	1000	0000	0000	0000
2	0011	1111	1111	1111	1100	0000	0000	0000
3	0001	1111	1111	1111	1110	0000	0000	0000
4	0000	1111	1111	1111	1111	0000	0000	0000
5	0000	0111	1111	1111	1111	1000	0000	0000
6	0000	0011	1111	1111	1111	1100	0000	0000
7	0000	0001	1111	1111	1111	1110	0000	0000
8	0000	0000	1111	1111	1111	1111	0000	0000
9	0000	0000	0111	1111	1111	1111	1000	0000
10	0000	0000	0011	1111	1111	1111	1100	0000
11	0000	0000	0001	1111	1111	1111	1110	0000
12	0000	0000	0000	1111	1111	1111	1111	0000
13	0000	0000	0000	0111	1111	1111	1111	1000
14	0000	0000	0000	0011	1111	1111	1111	1100
15	0000	0000	0000	0001	1111	1111	1111	1110
16	0000	0000	0000	0000	1111	1111	1111	1111
17	1000	0000	0000	0000	0111	1111	1111	1111
18	1100	0000	0000	0000	0011	1111	1111	1111
19	1110	0000	0000	0000	0001	1111	1111	1111
20	1111	0000	0000	0000	0000	1111	1111	1111
21	1111	1000	0000	0000	0000	0111	1111	1111
22	1111	1100	0000	0000	0000	0011	1111	1111
23	1111	1110	0000	0000	0000	0001	1111	1111
24	1111	1111	0000	0000	0000	0000	1111	1111
25	1111	1111	1000	0000	0000	0000	0111	1111
26	1111	1111	1100	0000	0000	0000	0011	1111
27	1111	1111	1110	0000	0000	0000	0001	1111
28	1111	1111	1111	0000	0000	0000	0000	1111
29	1111	1111	1111	1000	0000	0000	0000	0111
30	1111	1111	1111	1100	0000	0000	0000	0011
31	1111	1111	1111	1110	0000	0000	0000	0001

OFST < 16; IRSR00 ~ IRSR15	OFST < 16; IRSR16 ~ IRSR31
OFST ≥ 16; IRSR16 ~ IRSR31	OFST ≥ 16; IRSR00 ~ IRSR15

- OFST=00,16
- OFST=01,17
- OFST=02,18
- OFST=03,19
- OFST=04,20
- OFST=05,21
- OFST=06,22
- OFST=07,23
- OFST=08,24
- OFST=09,25
- OFST=10,26
- OFST=11,27
- OFST=12,28
- OFST=13,29
- OFST=14,30
- OFST=15,31



IRSR(32)

## 【表示メモリ内容論理演算機能】

関与するレジスタ・・・ MOD0, MOD1, PLANES, PLANE<sub>B</sub>

### (1) 表示メモリ内容読み出し時の論理演算

(a) MULTIPLE SOURCE COPY (PTNOUT← PTNIN § PTNOUT(M<sub>0</sub>))

PLANES内容が「1」のプレーンに対してのみ MOD<sub>0</sub> 内容による論理演算  
「0」のプレーンに対しては NO OPERATION。

( MOD1 内容は表示メモリ書き込み時に使用)

(b) 任意閉領域内塗りつぶし時の境界点検索 (PTNOUT← PTNIN § PTNOUT(BC))

PLANE<sub>B</sub>内容が「1」のプレーンに対して 「AND」による論理演算

PLANE<sub>B</sub>内容が「0」のプレーンに対しては「ソース反転 AND」による論理演算

( MOD0、MOD1 内容は表示メモリ書き込み時に使用)

(c) 任意閉領域内塗りつぶし時のスタック内容検索 (PTNOUT← PTNIN § PTNOUT(CS))

無条件に、「ソース反転 AND」による論理演算を実行し、結果が零であることを判定する。

### (2) 表示メモリ書き込み時の論理演算

(a) MULTIPLE SOURCE COPY (PTNOUT← PTNIN § PTNOUT(M<sub>1</sub>))

MOD<sub>1</sub> 内容による論理演算 (PLANESや P<sub>MAX</sub> 内容には依存しない)

(b) 任意閉領域内塗りつぶし時を含む、その他の描画

PLANES内容が「1」のプレーンに対しては MOD1 内容による論理演算

PLANES内容が「0」のプレーンに対しては MOD0 内容による論理演算

MULTIPLE SOURCE によるコピー描画を設定した場合、PLANES の最下位ビットは必ず  
”1” とすること。(これを怠ると、ソース側論理演算を正常に実行しない)

(例) PLANES=0000000000001100, EADS=0, PITCHS=10000H

↓

PLANES=0000000000000011, EADS=20000H, PITCHS=10000H



【任意倍率拡大縮小コピー】

転送元領域に対して水平方向および垂直方向について独立に拡大／縮小、および、その倍率を任意に設定できる。

(1)水平方向の拡大/ 縮小選択 . . . ESH

ESH	
0	水平方向縮小
1	水平方向拡大

(2)垂直方向の拡大/ 縮小選択 . . . ESV

ESV	
0	垂直方向縮小
1	垂直方向拡大

(3)水平方向の拡大／縮小倍率選択 . . . MAGH

(4)垂直方向の拡大／縮小倍率選択 . . . MAGV

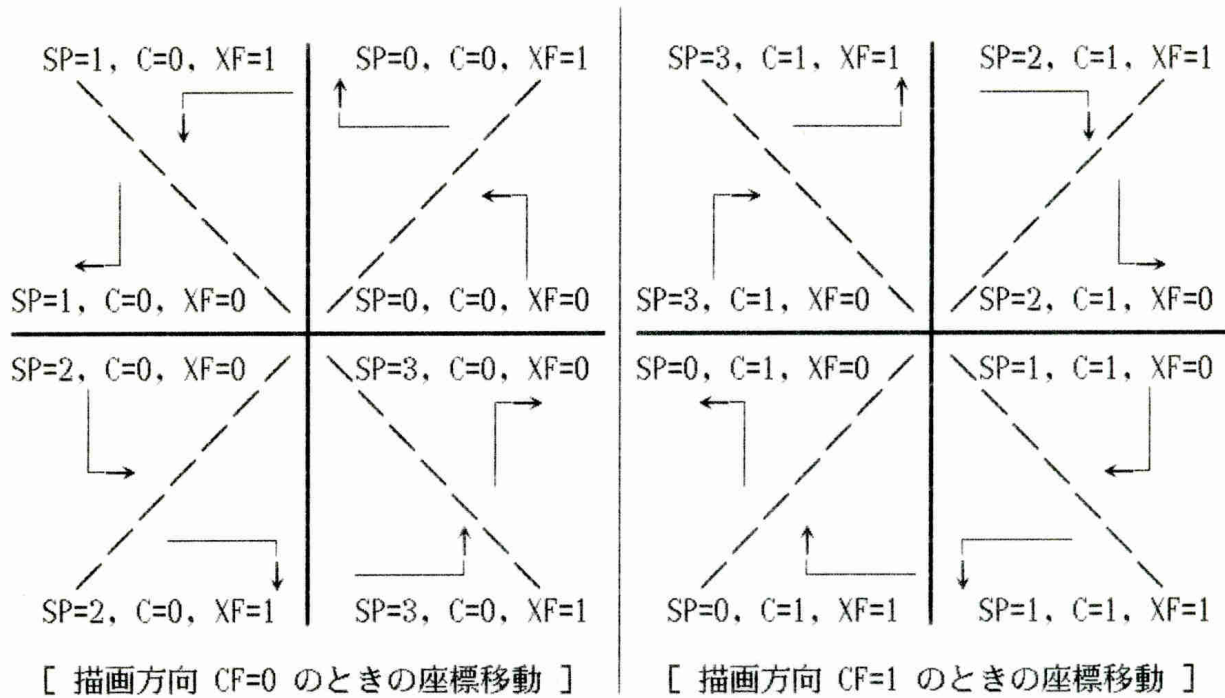
◎相互の関連性

	ESH=0															ESH=1																	
MAGH	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	MAGH
倍率	$\frac{1}{16}$	$\frac{2}{16}$	$\frac{3}{16}$	$\frac{4}{16}$	$\frac{5}{16}$	$\frac{6}{16}$	$\frac{7}{16}$	$\frac{8}{16}$	$\frac{9}{16}$	$\frac{10}{16}$	$\frac{11}{16}$	$\frac{12}{16}$	$\frac{13}{16}$	$\frac{14}{16}$	$\frac{15}{16}$	$\frac{16}{16}$	$\frac{16}{16}$	$\frac{16}{15}$	$\frac{16}{14}$	$\frac{16}{13}$	$\frac{16}{12}$	$\frac{16}{11}$	$\frac{16}{10}$	$\frac{16}{9}$	$\frac{16}{8}$	$\frac{16}{7}$	$\frac{16}{6}$	$\frac{16}{5}$	$\frac{16}{4}$	$\frac{16}{3}$	$\frac{16}{2}$	$\frac{16}{1}$	倍率

	ESV=0															ESV=1																	
MAGV	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	MAGV
倍率	$\frac{1}{16}$	$\frac{2}{16}$	$\frac{3}{16}$	$\frac{4}{16}$	$\frac{5}{16}$	$\frac{6}{16}$	$\frac{7}{16}$	$\frac{8}{16}$	$\frac{9}{16}$	$\frac{10}{16}$	$\frac{11}{16}$	$\frac{12}{16}$	$\frac{13}{16}$	$\frac{14}{16}$	$\frac{15}{16}$	$\frac{16}{16}$	$\frac{16}{16}$	$\frac{16}{15}$	$\frac{16}{14}$	$\frac{16}{13}$	$\frac{16}{12}$	$\frac{16}{11}$	$\frac{16}{10}$	$\frac{16}{9}$	$\frac{16}{8}$	$\frac{16}{7}$	$\frac{16}{6}$	$\frac{16}{5}$	$\frac{16}{4}$	$\frac{16}{3}$	$\frac{16}{2}$	$\frac{16}{1}$	倍率

【インクリメンタル・ポイントによる座標移動】

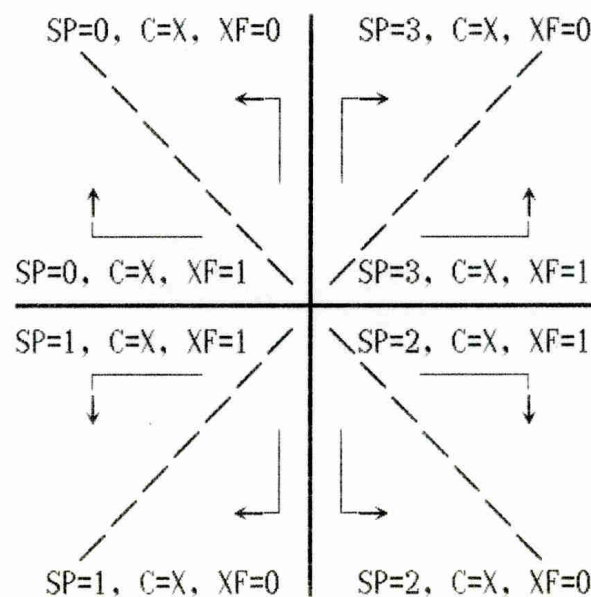
円／圆弧／楕円／楕圆弧



描画開始点は X, Y 軸上の点。開始点における SP, XF の関係は以下の通り。

	SP=0	SP=1	SP=2	SP=3
CF=0	XF=0	XF=1	XF=0	XF=1
CF=1	XF=1	XF=0	XF=1	XF=0

直線

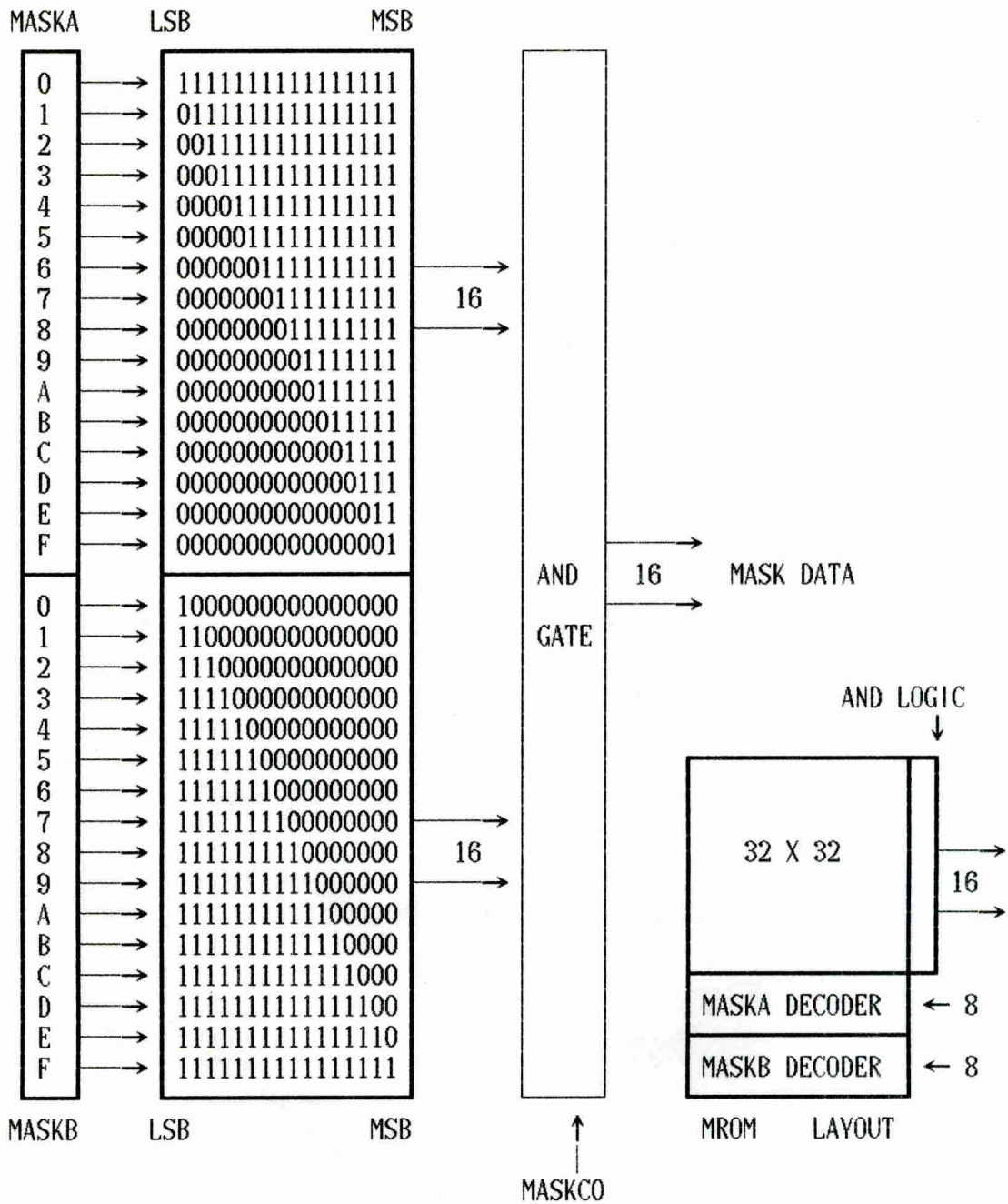


【±制御】

D<0	SP	C	XF	X#←X#+1 dADD←dADD+1 EADD←EADD+1	X#←X#-1 dADD←dADD-1 EADD←EADD-1	Y#←Y#+1 EADD←EADD-P	Y#←Y#-1 EADD←EADD+P
0	0	-	-		⊙	⊙	
0	1	-	-		⊙		⊙
0	2	-	-	⊙			⊙
0	3	-	-	⊙		⊙	
1	0	-	0			⊙	
1	1	-	0				⊙
1	2	-	0				⊙
1	3	-	0			⊙	
1	0	-	1		⊙		
1	1	-	1		⊙		
1	2	-	1	⊙			
1	3	-	1	⊙			

$X\# \leftarrow X\# \pm [N]$ ,  $Y\# \leftarrow Y\# \pm [N]$ ,  $EADD \leftarrow EADD \pm PITCHS$ ,  $dADD \leftarrow dADD \pm [N]$  など直線／楕円弧描画における演算命令の ADD/SUBの切り換え、演算の禁止を制御する。

【描画マスク発生用 ROM MROM】



ワード内の描画領域を決定する 16 ビットの信号を発生する ROM。

各々 4ビットのレジスタ MASKA、MASKB によってアドレスされる。

MASKA内容から MASKB内容までのドットが、描画領域として選択される。

具体例： MASKA=4, MASKB=0BH、MASKCO=0のとき、

MASK 出力 = 0000111111110000 となる。

MASKC は、描画時のマスク制御や閉領域内塗りつぶし時の境界点検出の際に用いられる 4ビットのレジスタ。その最下位ビット ( MASKCO ) によって、MROM 出力を反転するかないかを制御できる。 MASKCO=1のとき反転。レジスタ MASKC の上位ビットの機能については別紙参照。

【境界点検出用 ROM (BROM)】

OUT(4) : 境界点のドット・アドレス。

NEXT : 描画プロセッサの判定入力となる。ワード内に境界点が存在していない場合に「1」となる。

BACK : 境界点検出の結果、1 アドレス前のアドレスに境界点が存在していたことが判明した場合に「1」となる。

1←0 境界点検出 (MASK\_C=X00)

0→1 境界点検出 (MASK\_C=X01)

LSB	MSB	OUT	NEXT	BACK	LSB	MSB	OUT	NEXT	BACK
XXXXXXXX	XXXXXXXX1	0	0	* 1	00000000	00000001	14	0	0
XXXXXXXX	XXXXXXXX10	15	0	0	00000000	0000001X	13	0	0
XXXXXXXX	XXXXX100	14	0	0	00000000	000001XX	12	0	0
XXXXXXXX	XXXX1000	13	0	0	00000000	00001XXX	11	0	0
XXXXXXXX	XXX10000	12	0	0	00000000	0001XXXX	10	0	0
XXXXXXXX	XX100000	11	0	0	00000000	001XXXXX	9	0	0
XXXXXXXX	X1000000	10	0	0	00000000	01XXXXXX	8	0	0
XXXXXXXX	10000000	9	0	0	00000000	1XXXXXXX	7	0	0
XXXXXXX1	00000000	8	0	0	00000001	XXXXXXXX	6	0	0
XXXXXX10	00000000	7	0	0	0000001X	XXXXXXXX	5	0	0
XXXXX100	00000000	6	0	0	000001XX	XXXXXXXX	4	0	0
XXX10000	00000000	5	0	0	00001XXX	XXXXXXXX	3	0	0
XX100000	00000000	4	0	0	0001XXXX	XXXXXXXX	2	0	0
X1000000	00000000	3	0	0	001XXXXX	XXXXXXXX	1	0	0
10000000	00000000	2	0	0	01XXXXXX	XXXXXXXX	0	0	0
00000000	00000000	1	0	0	1XXXXXXX	XXXXXXXX	15	0	* 1
		0	1	0	00000000	00000000	0	* 1	0

0←1 境界点検出 (MASK\_C=X10)

1→0 境界点検出 (MASK\_C=X11)

LSB	MSB	OUT	NEXT	BACK	LSB	MSB	OUT	NEXT	BACK
XXXXXXXX	XXXXXXXX0	15	0	0	11111111	11111110	15	0	0
XXXXXXXX	XXXXXXXX01	14	0	0	11111111	1111110X	14	0	0
XXXXXXXX	XXXXX011	13	0	0	11111111	111110XX	13	0	0
XXXXXXXX	XXXX0111	12	0	0	11111111	11110XXX	12	0	0
XXXXXXXX	XXX01111	11	0	0	11111111	1110XXXX	11	0	0
XXXXXXXX	XX011111	10	0	0	11111111	110XXXXX	10	0	0
XXXXXXXX	X0111111	9	0	0	11111111	10XXXXXX	9	0	0
XXXXXXXX	01111111	8	0	0	11111111	0XXXXXXX	8	0	0
XXXXXX0	11111111	7	0	0	11111110	XXXXXXXX	7	0	0
XXXXXX01	11111111	6	0	0	1111110X	XXXXXXXX	6	0	0
XXXXX011	11111111	5	0	0	111110XX	XXXXXXXX	5	0	0
XXX01111	11111111	4	0	0	11110XXX	XXXXXXXX	4	0	0
XX011111	11111111	3	0	0	1110XXXX	XXXXXXXX	3	0	0
X0111111	11111111	2	0	0	110XXXXX	XXXXXXXX	2	0	0
X0111111	11111111	1	0	0	10XXXXXX	XXXXXXXX	1	0	0
01111111	11111111	0	0	0	0XXXXXXX	XXXXXXXX	0	0	0
11111111	11111111	0	1	0	11111111	11111111	0	* 1	0

dADD,B1,N1←B\_ROM またはdADS,B2,N2←B\_ROM 命令時、\* で示したビットが dADD または dADSの 5ビット目に OR される。

【境界点検出制御用レジスタ MASKC】

ドット・マスク用レジスタ MASKA, MASKB が生成するマスク出力を基にして境界点検索を行う B\_ROMのアドレス制御を行う 3ビットのレジスタ。以下に示す機能を持つ。

MASKC	機 能	
X00	1←0 境界点検出	BROM に供給されるデータの うち MROM 出力が「1」である ビットを強制的に「0」とする
X01	0→1 境界点検出	
X10	0←1 境界点検出	BROM に供給されるデータの うち MROM 出力が「1」である ビットを強制的に「1」とする
X11	1→0 境界点検出	

MASKC	機 能
0XX	MROM 出力反転せず
1XX	MROM 出力反転

【境界点検索 MASKA, MASKB, MASKC設定早見表】

	MASKA	MASKB	MASKC
1/0	dADD	dADD	46
1←0 ワード内	dADD	OFH	0
1←0 ワード間	0	OFH	4
0←1 ワード内	dADD	OFH	2
0←1 ワード間	0	OFH	6
1→0 ワード内	0	dADD	3
1→0 ワード間	0	OFH	7
0→1 ワード内	0	dADD	1
0→1 ワード間	0	OFH	5

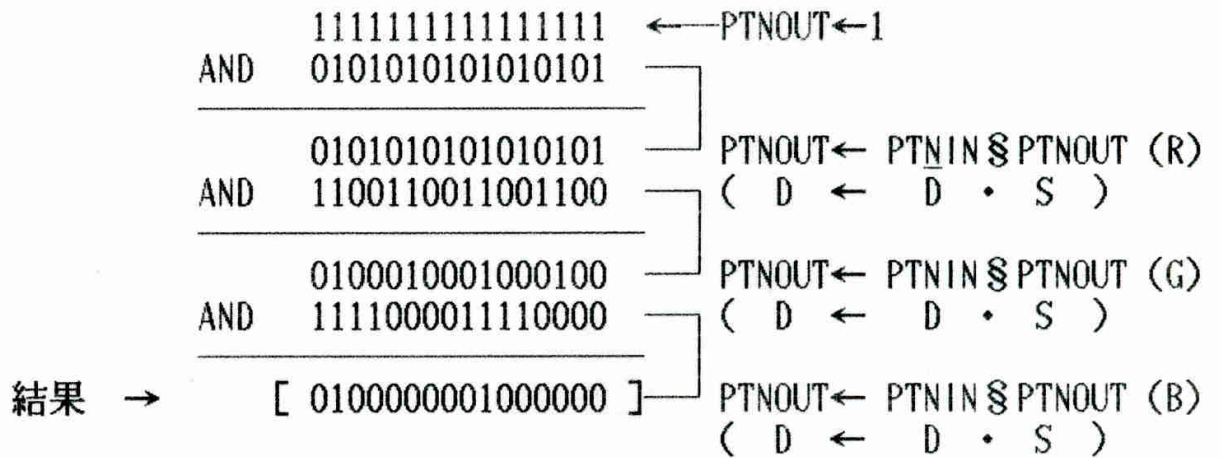
【境界点検索時に使用する論理演算】

【表示メモリ各プレーンの内容】

LSB	MSB	
1010101010101010		RED
1100110011001100		GREEN
1111000011110000		BLUE

【境界色指定による境界点検索】

境界色： CYAN (PLANE<sub>B</sub>=0000000000000110)

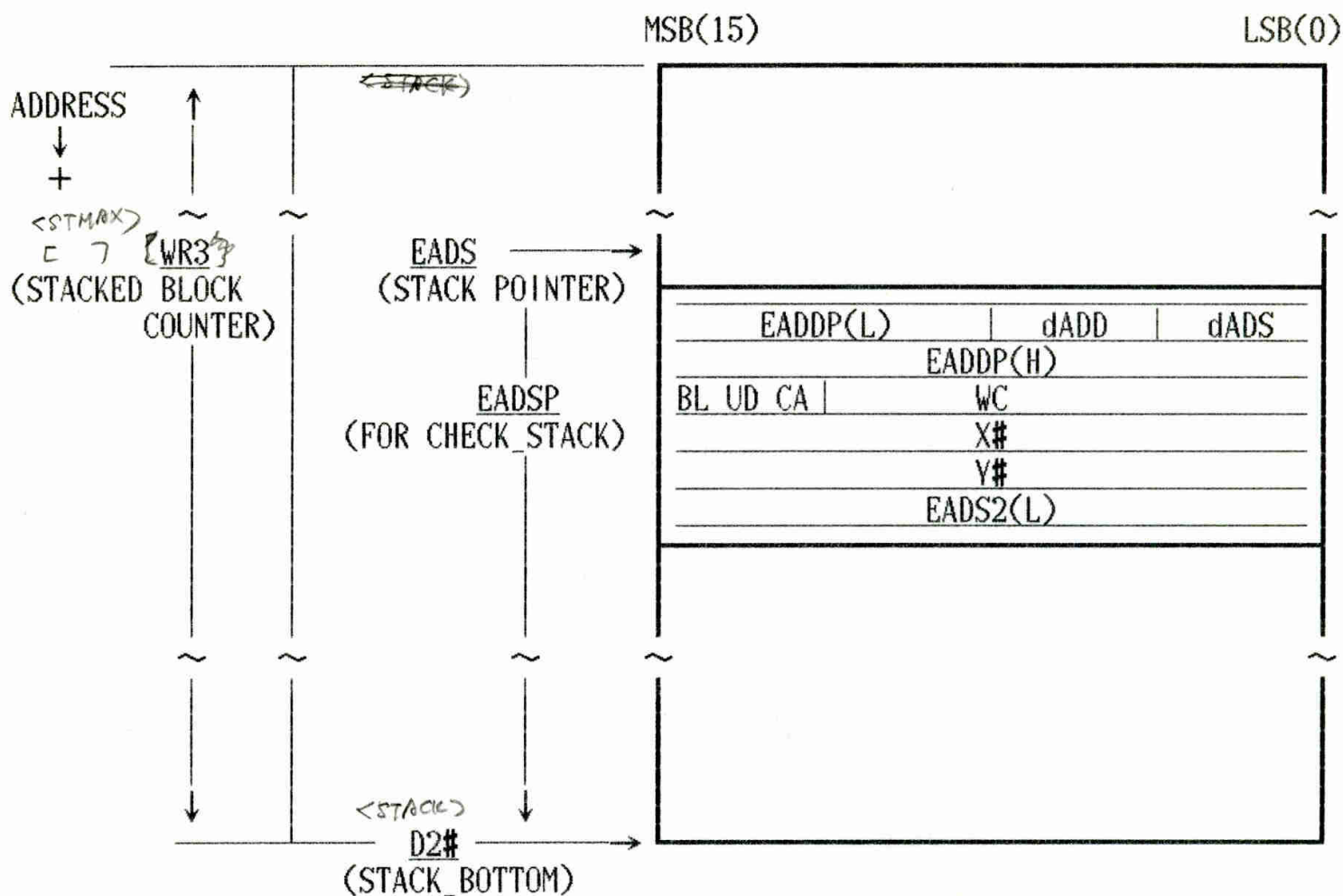


【境界色としない色の指定による境界点検索】

境界色としない色： CYAN (PLANE<sub>B</sub>=0000000000000110)

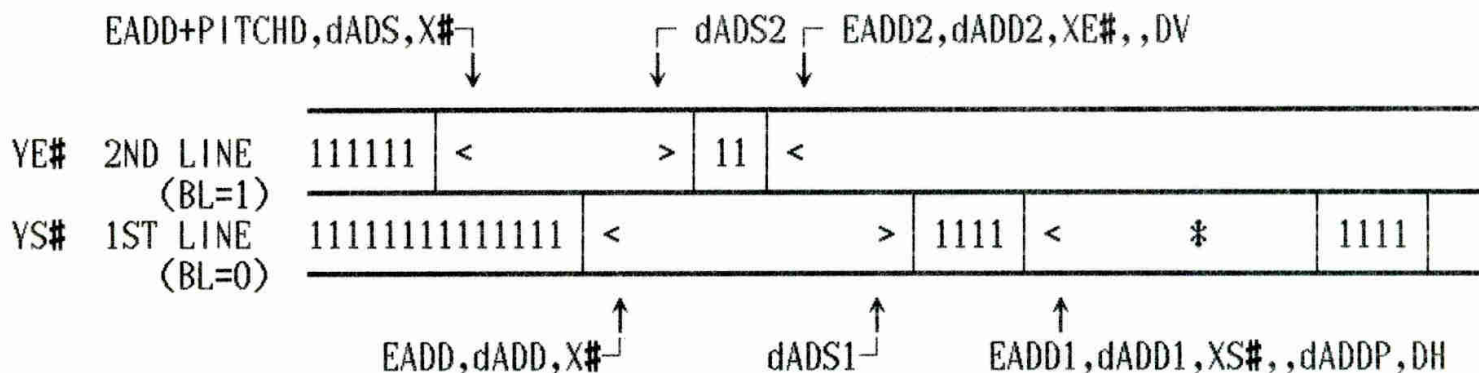
上記結果の反転 → [ 1011111110111111 ]

# 【スタック制御】



< > ; コーサ処理  
 [ ] ; 20720577k

# 【アドレス・レジスタ使用法】



2ND LINE SEARCH;  
 1ST LINE SEARCH;

dADS, B2, N2 ← B\_ROM  
 dADD, B1, N1 ← B\_ROM

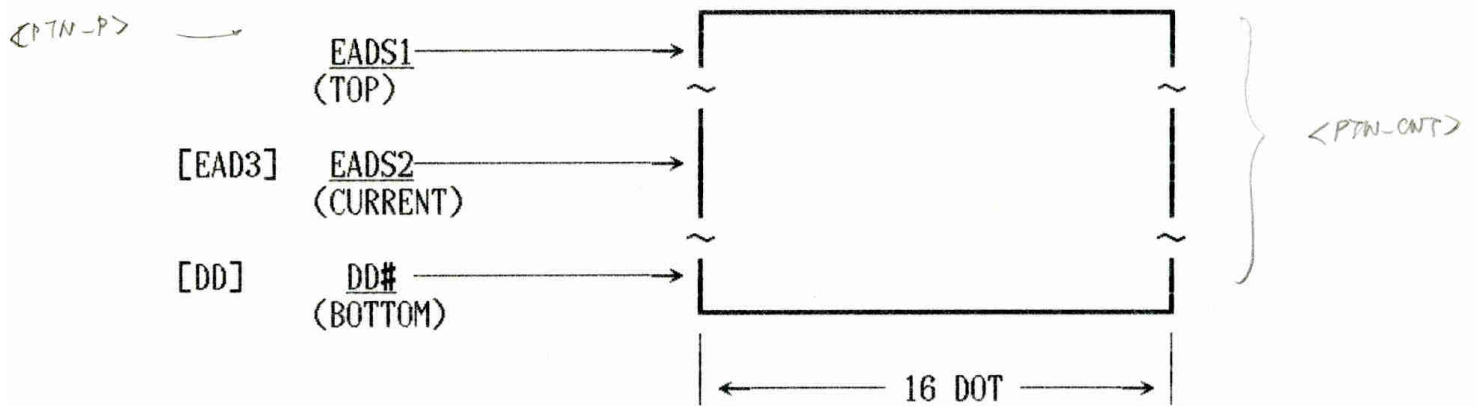
PUSH, POP;

dADD, dADS, EADDP(L): EADDP(H)



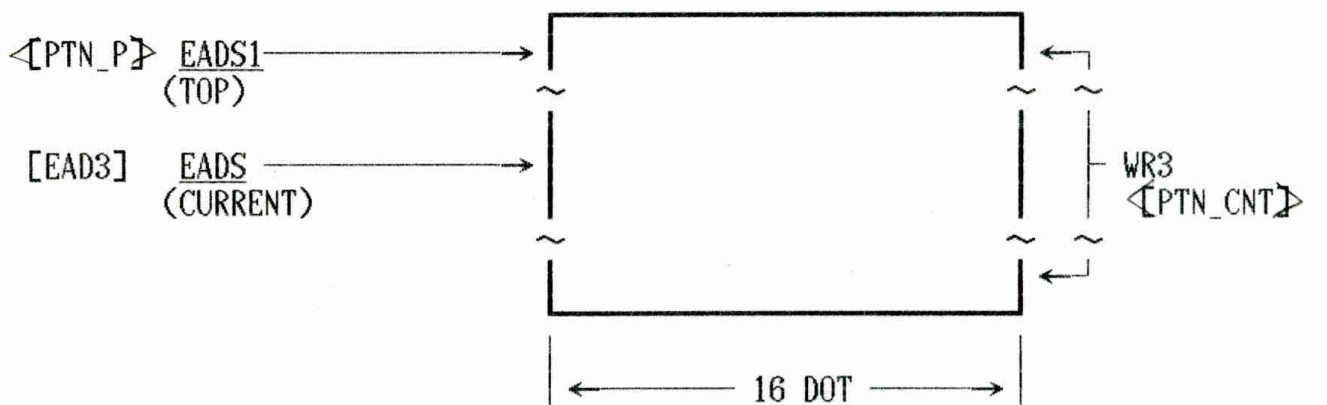
## 【塗りつぶしパターン制御】

### (1) PAINT



INTER-PLANE READ; EADSP, PDISPS

### (2) FILL



INTER-PLANE READ; EADSP, PDISPS

< > 2-# 13 位  
[ ] 717D 5 8 6

## 【各種フラグによるフロー制御】

### (1) SRCH\_SEL (SA=6)

左方向検索時 (LEFT\_SRCH);

[ (\*1←0\*) ] 検索の実行 : SRCH\_SEL = "00"

[ (\*1←0\*) ] 検索の実行 : SRCH\_SEL = "01"

[ (\*0←1\*) ] 検索の実行 : SRCH\_SEL = "10"

右方向検索時 (RIGHT\_SRCH);

[ (\*1←0\*) ] 検索の実行 : SRCH\_SEL = "00"

[ (\*0→1\*) ] 検索の実行 : SRCH\_SEL = "01"

[ (\*1→0\*) ] 検索の実行 : SRCH\_SEL = "10"

### (2) 1STP (SA=5)

## 【塗りつぶしスタック制御】

### ①探索後のプッシュ

すべての探索が未だ完了はしていないが、左右の境界点が検出された時。

- (1) STACK\_POINTER (EADS)+1 のアドレスから STACK\_BOTTOM (D2#) のアドレスまでdADD,dADS,EADDP(L)および EADDP(H) の 2ワードについてのアドレス比較を実行する。まず初めにdADD,dADS,EADDP(L)について比較を行い一致が取れた場合には引き続き EADDP(H) についての比較を行う。一致が取れなかった場合にはスタック領域のチェックを行った後、次のスタックデータ・ブロックのアドレス比較を行う。
- (2) 一致検出がなされた場合、キャンセルフラグ (CA) をセットするだけで、プッシュは実行しない。
- (3) すべてのデータブロックについてアドレス一致が取れなかった場合には、プッシュを実行する。

### ②探索後の描画実行

すべての探索（上下 2ラインについて）が完了した時のプッシュまたは描画可能な部分についての描画。必ず右方向探索を行った後終了する。第一ラインは第二ラインに対して優先して描画実行される（ポップ順序が早い）。

- (1) ①-(1)と同様にアドレス比較を行う。
- (2) 一致検出がなされた場合、キャンセルフラグ (CA) をセットするだけで、描画は実行せず、次にポップ動作を実行する。
- (3) すべてのデータブロックについてアドレス一致が取れなかった場合には、描画を実行し、描画開始点を基に探索を開始する。

### ③ポップ

プッシュデータを検出できなかった時の探索終了後、または、探索後またはポップ後の描画実行後、探索に入れられない場合。

- (1) 最終データブロックを読み出す。
- (2) CANCEL FLAGが立っている場合には、描画を実行した後、再びポップ動作に入

る。

- (3) CANCEL FLAGが立っていない場合には、描画を実行した後、描画開始点を基にした探索を開始する。

#### 【塗りつぶし実行の条件】

- (1)  $\begin{bmatrix} 0 \rightarrow 1 \\ 0 \rightarrow 1 \end{bmatrix}$  探索において、基線/第二線共に同一 X座標の境界点を検出したとき、  
第二線データをプッシュ。  
基線データを描画 (PAINT\_WRITE\_1)。
- (2)  $\begin{bmatrix} 1 \rightarrow 0 \\ 0 \rightarrow 1 \end{bmatrix}$  探索において、0→1 境界点を先に検出したとき、  
基線データを描画 (PAINT\_WRITE\_1)。
- (3)  $\begin{bmatrix} 0 \rightarrow 1 \\ 1 \rightarrow 0 \end{bmatrix}$  探索において、0→1 境界点を先に検出したとき、  
第二線データを描画 (PAINT\_WRITE\_2)。
- (4)  $\begin{bmatrix} 0 \rightarrow 1 \\ \text{IGNORE} \end{bmatrix}$  探索において、基線の描画済み境界と同一な X座標に境界点を検出したとき、  
第二線データを描画 (PAINT\_WRITE\_2)。
- (5) POP時、POP データが基線データであるとき、  
基線データを描画 (PAINT\_WRITE\_1)。
- (6) POP時、POP データが第二線データであるとき、  
第二線データを描画 (PAINT\_WRITE\_2)。

【境界点判定法】

① ユーザ指定探索開始点 (dADD) の 1/0判定

設定: MASKA←dADD, MASKB←dADD, MASKC←4, dADD,B1,N1←B\_ROM

原型データ (dADD=0BH)	XXXXXXXXXXOXXXX	原型データ (dADD=0BH)	XXXXXXXXXX1XXXX
マスクデータ	111111111101111	マスクデータ	111111111101111
結果	0000000000000000 NEXT=1 "0"	結果	0000000000010000 NEXT=0 "1"

② 1←0 境界点判定

(A) ワード内境界点判定 (1回目のみ)

設定: MASKA←dADD, MASKB←0FH, MASKC←0, dADD,B1,N1←B\_ROM

原型データ (dADD=0BH)	0000000000XXXXX	原型データ (dADD=0BH)	XXXXX10000XXXXX
マスクデータ	000000000011111	マスクデータ	000000000011111
結果	0000000000000000 NEXT=1 [境界点無し]	結果	XXXXX10000000000 NEXT=0, dADD=7 [境界点有り]

(B) ワード間境界点判定 (2回目以降)

設定: MASKA←0, MASKB←0FH, MASKC←4, dADD,B1,N1←B\_ROM

原型データ	0000000000000000	原型データ	XXXXX10000000000
マスクデータ	0000000000000000	マスクデータ	0000000000000000
結果	0000000000000000 NEXT=1 [境界点無し]	結果	XXXXX10000000000 NEXT=0, dADD=7 [境界点有り]

### ③ 0←1 境界点判定

#### (A) ワード内境界点判定 (1回目のみ)

設定: MASKA←dADD, MASKB←0FH, MASKC←2, dADD,B1,N1←B\_ROM

原型データ 1111111111XXXX  
(dADD=0BH)

原型データ XXXXX011111XXXX  
(dADD=0BH)

マスクデータ 000000000011111

マスクデータ 000000000011111

---

結果 111111111111111

---

結果 XXXXX01111111111

NEXT=1

NEXT=0, dADD=7

[境界点無し]

[境界点有り]

#### (B) ワード間境界点判定 (2回目以降)

設定: MASKA←0, MASKB←0FH, MASKC←6, dADD,B1,N1←B\_ROM

原型データ 1111111111111111

原型データ XXXXX01111111111

マスクデータ 0000000000000000

マスクデータ 0000000000000000

---

結果 1111111111111111

---

結果 XXXXX01111111111

NEXT=1

NEXT=0, dADD=7

[境界点無し]

[境界点有り]

#### ④ 1→0 境界点判定

##### (A) ワード内境界点判定 (1回目のみ)

設定: MASKA←dADD, MASKB←0FH, MASKC←3, dADD,B1,N1←B\_ROM

原型データ XXXXX1111111111  
(dADD=05H)

マスクデータ 1111100000000000

結果 1111111111111111

NEXT=1

[境界点無し]

原型データ XXXXX111110XXXXX  
(dADD=05H)

マスクデータ 1111100000000000

結果 1111111110XXXXX

NEXT=0, dADD=A

[境界点有り]

##### (B) ワード間境界点判定 (2回目以降)

設定: MASKA←0, MASKB←0FH, MASKC←7, dADD,B1,N1←B\_ROM

原型データ 1111111111111111

マスクデータ 0000000000000000

結果 1111111111111111

NEXT=1

[境界点無し]

原型データ 1111111110XXXXX

マスクデータ 0000000000000000

結果 1111111110XXXXX

NEXT=0, dADD=A

[境界点有り]

⑤ 0→1 境界点判定

(A) ワード内境界点判定 ( 1回目のみ)

設定: MASKA←dADD, MASKB←0FH, MASKC←1, dADD,B1,N1←B\_ROM

原型データ XXXXX000000000000  
(dADD=05H)

原型データ XXXXX000001XXXXX  
(dADD=05H)

マスクデータ 1111100000000000

マスクデータ 1111100000000000

結果 0000000000000000

結果 0000000001XXXXX

NEXT=1

NEXT=0, dADD=A

[境界点無し]

[境界点有り]

(B) ワード間境界点判定 ( 2回目以降)

設定: MASKA←0, MASKB←0FH, MASKC←5, dADD,B1,N1←B\_ROM

原型データ 0000000000000000

原型データ 0000000001XXXXX

マスクデータ 0000000000000000

マスクデータ 0000000000000000

結果 0000000000000000

結果 0000000001XXXXX

NEXT=1

NEXT=0, dADD=A

[境界点無し]

[境界点有り]

《左方向検索判定条件》

(1) (\*1←0\*)  
 (\*0←1\*) 検索時 ;

XXXX	XX1 <u>0</u> 0000	XXXX	1 <u>0</u> 000000	XXX1	<u>0</u> 0000000
XXXX	X0111111	XXXX	01111111	XXX0	11111111
dADS > dADD (3) (1)		dADS > dADD (1) (0)		dADS > dADD (16) (15)	

XXXX	XXX10000	XXXX	XXXX1000	XXXX	00000000
XXXX	XXXX <u>0</u> 111	XXXX	XXXXXX <u>0</u> 1	XXXX	<u>0</u> 1111111
dADS = dADD (4) (4)		dADS < dADD (5) (6)		dADS = dADD (0) (0)	

(2) (\*1←0\*)  
 (\*1←0\*) 検索時 ;

XXXX	XX1 <u>0</u> 0000	XXXX	1 <u>0</u> 000000	XXXX	XXX1 <u>0</u> 000	XXX1	<u>0</u> 0000000
XXXX	10000000	XXXX	00000000	XXXX	XXX1 <u>0</u> 000	XXX1	<u>0</u> 0000000
dADS > dADD (3) (1)		dADS > dADD (1) (0)		dADS = dADD (4) (4)		dADS = dADD (16) (16)	

XXXX	00000000	XXXX	XXXX1000
XXXX	1 <u>0</u> 000000	XXXX	XXXXX1 <u>0</u>
dADS < dADD (0) (1)		dADS < dADD (5) (6)	



(3) (\*0←1\*)  
 (\*1←0\*) 検索時 ;

XXXX	X0111111
XXXX	XX1 <u>0</u> 0000

dADD > dADS  
 (3) (1)

XXXX	01111111
XXXX	1 <u>0</u> 000000

dADD > dADS  
 (1) (0)

XXX0	11111111
XXX1	<u>0</u> 0000000

dADD > dADS  
 (16) (15)

XXXX	XXXX <u>0</u> 111
XXXX	XXX10000

dADD = dADS  
 (4) (4)

XXXX	XXXXXX <u>0</u> 1
XXXX	XXXX1000

dADD < dADS  
 (5) (6)

XXXX	<u>0</u> 1111111
XXXX	00000000

dADS = dADD  
 (0) (0)

《右方向検索判定条件》

- (1) (\*0→1\*)  
(\*1→0\*) 検索時 ;

00000001XX   XX	0000000001   XX	0000000000   1X	*; ボロ-反転
111111110X   XX	1111111110   XX	1111111111   0X	
dADS < dADD (12) (14)	dADS < dADD (14) (15)	dADS < dADD *(31) (0)	
0000001XXX   XX	000001XXXX   XX	0000000000   XX	
111110XXXX   XX	1110XXXXXX   XX	1111111110   XX	
dADS = dADD (11) (11)	dADS > dADD (10) (9)	dADS > dADD (16) (15)	

- (2) (\*0→1\*)  
(\*0→1\*) 検索時 ;

00000001XX   XX	0000001XXX   XX	0000000000   XX	
0000000001   XX	0000001XXX   XX	0000000001   XX	
dADS < dADD (12) (14)	dADS = dADD (11) (11)	dADS > dADD (16) (14)	
0000000001   XX	0000000000   1X	0000000000   XX	*; ボロ-反転
0000000000   XX	0000000000   1X	1111111110   XX	
dADS < dADD (14) (16)	dADS = dADD *(31) *(31)	dADS < dADD (16) (15)	

(2) (\*1→0\*)  
 (\*0→1\*) 検索時 ;

111111110X	XX
00000001XX	XX

dADS > dADD  
 (14) (12)

1111111110	XX
0000000001	XX

dADS > dADD  
 (15) (14)

1111111111	0X
0000000000	1X

dADS > dADD  
 (0) \*(31)

\*; ボロー反転

111110XXXX	XX
0000001XXX	XX

dADS = dADD  
 (11) (11)

1110XXXXXX	XX
000001XXXX	XX

dADS < dADD  
 (19) (10)

1111111110	XX
0000000000	XX

dADS < dADD  
 (15) (16)

【コピー [COPY]】

	ROT=0000 (0)	ROT=0010 (2)	ROT=0100 (4)	ROT=0110 (6)	ROT=1000 (8)	ROT=1010 (A)	ROT=1100 (C)	ROT=1110 (E)
EADS	EADS+1, EADS+P				EADS-1, EADS-P			
ICHG(RD)	0				1			
ICHG(WR)	0	1	0		1		0	
MASKA	1ST=1; dADD	WR1-15~ 4≠0; 0		1ST=1; dADD		DH15~ 4≠0; 0		1ST=1; dADD
	1ST=0; 0	WR1-15~ 4=0; /WR1-3~0/		1ST=0; 0		DH15~ 4=0; /WR1-3~0/		1ST=0; 0
MASKB	WR1-15~ 4≠0; OFH	1ST=1; dADD		WR1-15~ 4≠0; OFH		1ST=1; dADD		WR1-15~ 4≠0; OFH
	WR1-15~ 4=0; WR1-3~0	1ST=0; OFH		WR1-15~ 4=0; WR1-3~0		1ST=0; OFH		WR1-15~ 4=0; WR1-3~0
EADD	EADD+1, EADD+P	EADD-1, EADD-P	EADD-1, EADD+P	EADD+1, EADD-P	EADD+1, EADD+P	EADD-1, EADD-P	EADD-1, EADD+P	EADD+1, EADD-P

	ROT=0000 (0)	ROT=0010 (2)	ROT=0100 (4)	ROT=0110 (6)	ROT=1000 (8)	ROT=1010 (A)	ROT=1100 (C)	ROT=1110 (E)
OFST (COPY)	1ST=1; dADS-dADD	1ST=1; dADS-/dADD/		1ST=1; dADS-dADD	1ST=1; /dADS/-dADD	1ST=1; /dADS/-/dADD/		1ST=1; /dADS/-dADD
	1ST=0; OFST+16							
OFST,OST' (SL_COPY)	1ST=1; dADS-dADD	1ST=1; dADS-/dADD/		1ST=1; dADS-dADD	1ST=1; /dADS/-dADD	1ST=1; /dADS/-/dADD/		1ST=1; /dADS/-dADD
	1ST=0; OFST+16							
dADD± DEPENDENT	NEXT LINE; OFST'±1	NEXT LINE; OFST'±1		NEXT LINE; OFST'±1		NEXT LINE; OFST'±1		NEXT LINE; OFST'±1
DH	DH+dADD	DH+/dADD/		DH+dADD		DH+/dADD/		DH+dADD

【拡大/縮小コピー [ES\_COPY]】

	ROT=0000 (0)	ROT=0010 (2)	ROT=0100 (4)	ROT=0110 (6)	ROT=1000 (8)	ROT=1010 (A)	ROT=1100 (C)	ROT=1110 (E)
EADS	EADS+1, EADS+P				EADS-1, EADS-P			
ICHG(RD)	0				1			
ICHG(WR)	0	1		0		1		0
EADD	EADD+1, EADD+P	EADD-1, EADD-P	EADD-1, EADD+P	EADD+1, EADD-P	EADD+1, EADD+P	EADD-1, EADD-P	EADD-1, EADD+P	EADD+1, EADD-P

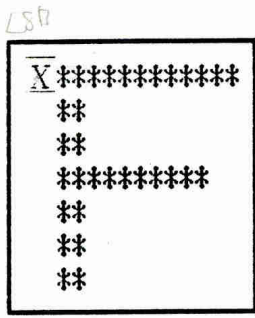
	ROT=0000 (0)	ROT=0010 (2)	ROT=0100 (4)	ROT=0110 (6)
OFST	1ST=1; dADS			
	1ST=0; OFST+K1			
DH	DH			
MASKA	1STM=1; MOFST	(MOFST1(MSB)=1 & 1STM=1)=1; 0		1STM=1; MOFST
	1STM=0; 0	(MOFST1(MSB)=1 & 1STM=1)=0; /MOFST1/		1STM=0; 0
MASKB	(MOFST1(MSB)=1 & 1STM=1)=0; MOFST1	1STM=1; /MOFST/	(MOFST1(MSB)=1 & 1STM=1)=0; MOFST1	
	(MOFST1(MSB)=1 & 1STM=1)=1; OFH	1STM=0; OFH	(MOFST1(MSB)=1 & 1STM=1)=1; OFH	
MOFST	1ST=1; dADD	1ST=1; /dADD/		1ST=1; dADD
	1ST=0; MOFST1			

【90度回転コピー [90\_COPY]】

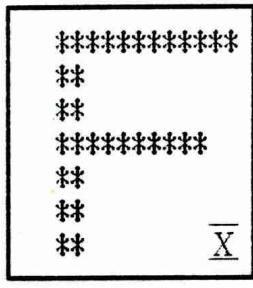
	ROT=0001 (1)	ROT=0011 (3)	ROT=0101 (5)	ROT=0111 (7)	ROT=1001 (9)	ROT=1011 (B)	ROT=1101 (D)	ROT=1111 (F)
EADS	EADS+1, EADS+P				EADS-1, EADS-P			
ICHG(RD)	0				1			
ICHG(WR)	0	1	0		1		0	
MASKA	1ST=1; dADD	WR1-15~ 4≠0; 0	1ST=1; dADD		WR1-15~ 4≠0; 0	1ST=1; dADD		
	1ST=0; 0	WR1-15~ 4=0; /WR1-3~0/	1ST=0; 0		WR1-15~ 4=0; /WR1-3~0/	1ST=0; 0		
MASKB	WR1-15~ 4≠0; OFH	1ST=1; dADD	WR1-15~ 4≠0; OFH		1ST=1; dADD	WR1-15~ 4≠0; OFH		
	WR1-15~ 4=0; WR1-3~0	1ST=0; OFH	WR1-15~ 4=0; WR1-3~0		1ST=0; OFH	WR1-15~ 4=0; WR1-3~0		
EADD	EADD-P, EADD+1	EADD+P, EADD-1	EADD-P, EADD-1	EADD+P, EADD+1	EADD-P, EADD+1	EADD+P, EADD-1	EADD-P, EADD-1	EADD+P, EADD+1

	ROT=0001 (1)	ROT=0011 (3)	ROT=0101 (5)	ROT=0111 (7)	ROT=1001 (9)	ROT=1011 (B)	ROT=1101 (D)	ROT=1111 (F)
IRRAV	1ST#=1; dADS				1ST#=1; /dADS/			
	1ST#=0; 0				1ST#=0; 0			
IRRAH	1ST=1; dADD	1ST=1; /dADD/	1ST=1; dADD	1ST=1; dADD	1ST=1; dADD	1ST=1; /dADD/	1ST=1; dADD	
	1ST=0; 0				1ST=0; 0			

【ROT の定義】

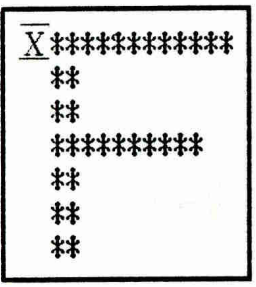


ROT=0XXX

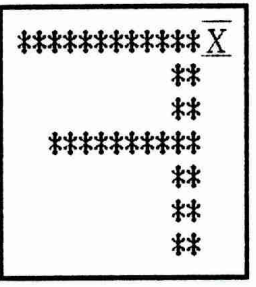


ROT=1XXX

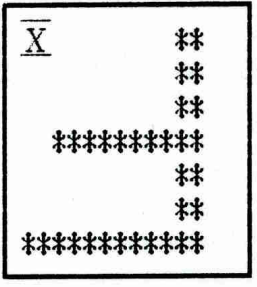
SOURCE



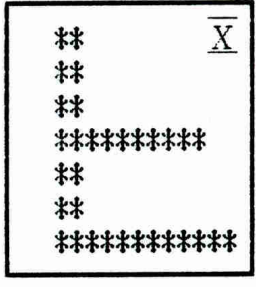
ROT=0000 (0)



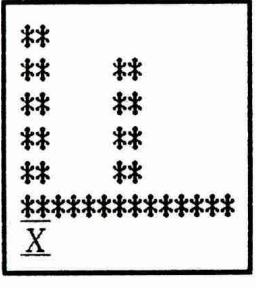
ROT=0100 (4)



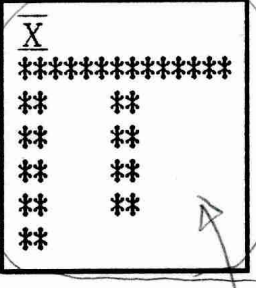
ROT=1000 (8)



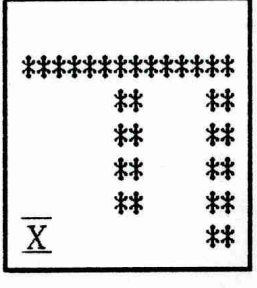
ROT=1100 (C)



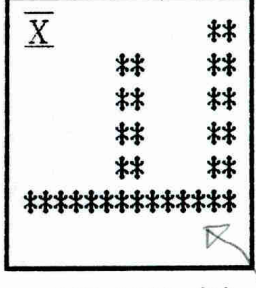
ROT=0001 (1)



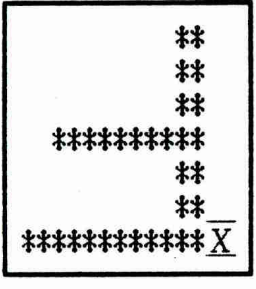
ROT=0101 (5)



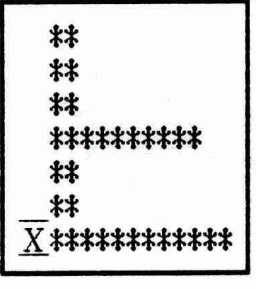
ROT=1001 (9)



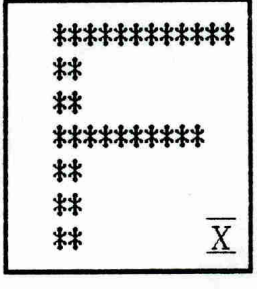
ROT=1101 (D)



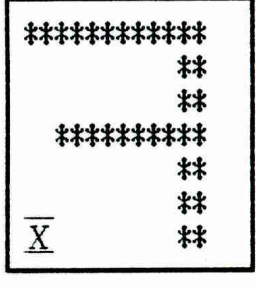
ROT=0010 (2)



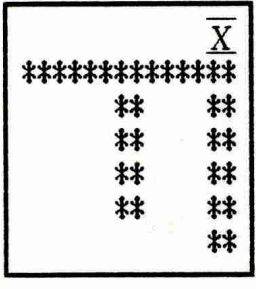
ROT=0110 (6)



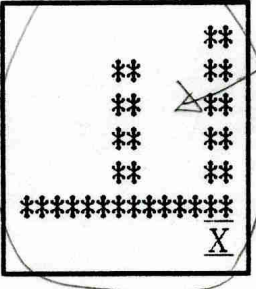
ROT=1010 (A)



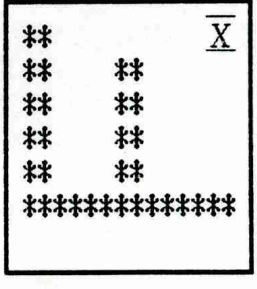
ROT=1110 (E)



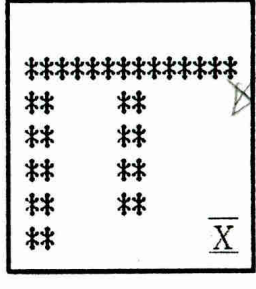
ROT=0011 (3)



ROT=0111 (7)



ROT=1011 (B)



ROT=1111 (F)

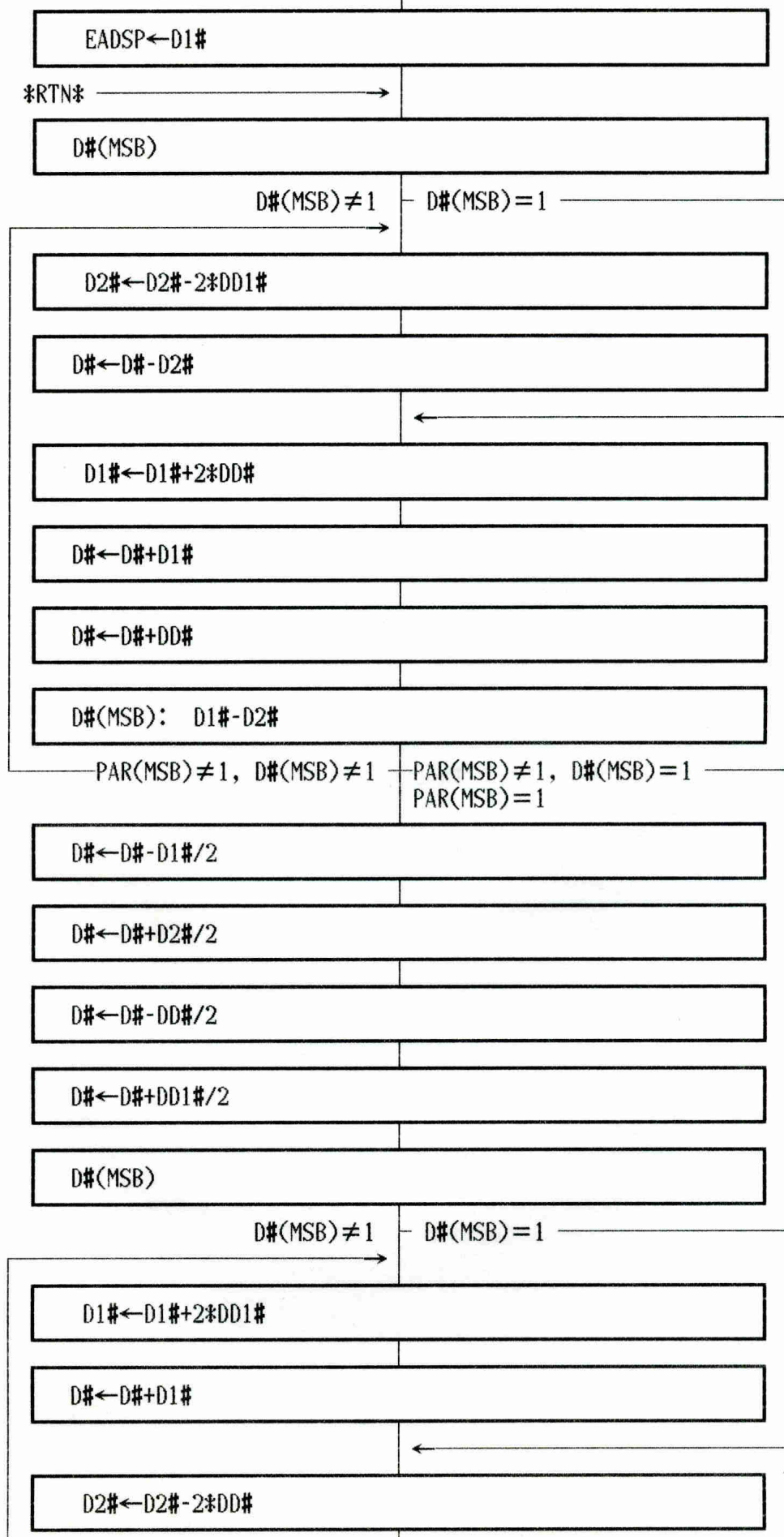
DESTINATION

DESTINATION

21  
16  
17  
25  
19

【円／円弧／楕円／楕円弧描画概略フロー】

[ CRL/CARC/ELPS/EARC ]





$D\# \leftarrow D\# - D2\#$

$D\# \leftarrow D\# + DD1\#$

$D\#(\text{MSB}) : D2\# - 1$

$\text{PAR}(\text{MSB}) \neq 1, D\#(\text{MSB}) \neq 1$

$\text{PAR}(\text{MSB}) \neq 1, D\#(\text{MSB}) = 1$   
 $\text{PAR}(\text{MSB}) = 1$

$DD\# \leftarrow DD1\#$

$DD1\# \leftarrow DD\#$

$D2\# \leftarrow D1\#$

$D1\# \leftarrow \text{EADSP}$

\*RTN\*