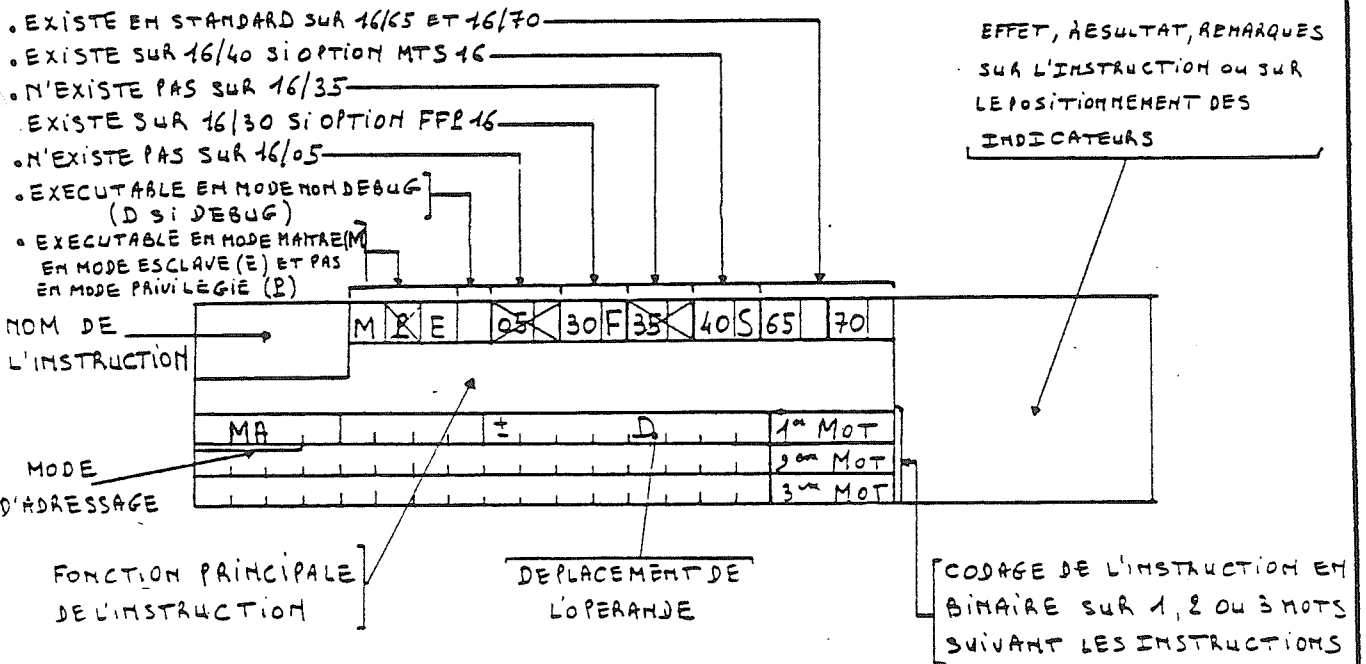


LES INSTRUCTIONS DE BASE, LES INSTRUCTIONS EN STANDARD OU EN OPTION SUIVANT LES UNITES CENTRALES (MTS 16- FFP 16- CDA) SONT LISTEES PAR ORDRE ALPHABETIQUE.

LES NOUVELLES INSTRUCTIONS, EXECUTABLES SUR 16/35 ET 16/70, QUI CONSTITUENT L'OPTION IS 2 16 SONT A LA FIN DE CETTE LISTE, EGALEMENT PAR ORDRE ALPHABETIQUE A PARTIR DU FOLIO 21

REMARQUE : LES INSTRUCTIONS FFL 16 (F) SONT TOUTES OPTIONNELLES SAUF POUR LE 16/35 QUI DE SERIE POSSEDE LE PLOTTANT MICROPROGRAMME.

## SIGNIFICATION DES SIGLES ET SYMBOLES



A REGISTRE A  
 (A) CONTENU DU REGISTRE A  
 ((A)) CONTENU DU CONTENU DE A  
 (A)1 BIT 1 DU REGISTRE A  
 (A)8-15 CONTENU DE A DE 8 A 15  
 AE ADRESSE EFFECTIVE  
 AEB OCTET DE L'ADRESSE EFFECTIVE  
 Rd REGISTRE DESTINATAIRE  
 RS REGISTRE SOURCE  
 MA MODE D'ADRESSAGE  
 ± D. DEPLACEMENT SIGNE  
 V VALEUR (OPERANDE IMMEDIAT)

Q ADRESSE  
 NS NO DETACHE SOFT EN COURS : MEMOIRE '0000  
 NL NO DETACHE SOFT NON PRETE : MEMOIRE '0001  
 ASTF FILE DES TACHES SOFT A REESES  
 ESTF FILE DES TACHES SOFT ELIGIBLES  
 RSTF FILE DES TACHES SOFT PRETES  
 P COMPTeur D'UN SEMAPHORE  
 U UNION  
 ∩ INTERSECTION  
 :: COMPARAISON  
 ⊕ DISTINCTION  
 := CONTIENT

**Bull**



**SPS 5**

CODES D'ORDRE

N° Document

71 F7 31MS

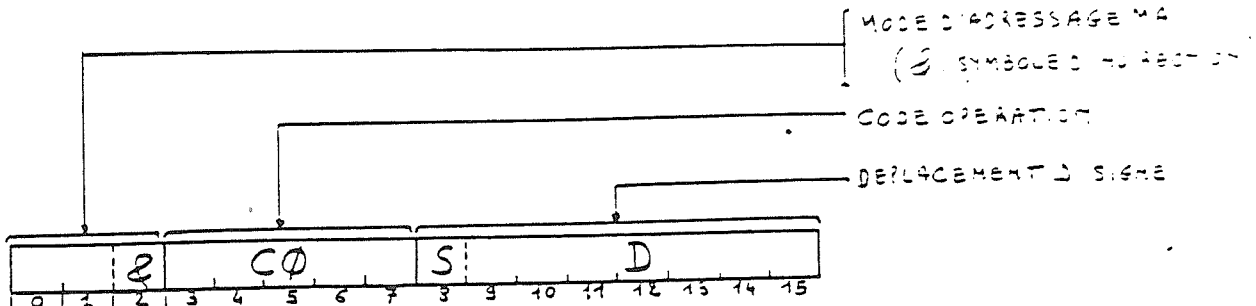
Date

547

Page

P. 7.1

# MODES D'ADRESSAGE



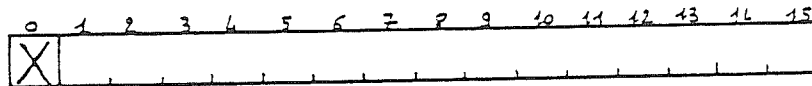
- si BIT 2 = 0 : ADRESSAGE DIRECT DE MOTS

- 0 1 0 →  $AE := (C) + DS$  ADRESSAGE PAR RAPPORT A C
- 1 0 0 →  $AE := (L) + DS$  ADRESSAGE PAR RAPPORT A L
- 1 1 0 →  $AE := (W) + DS$  ADRESSAGE PAR RAPPORT A W

- si BIT 2 = 1 : ADRESSAGE INDIRECT DE MOTS

- 0 1 1 → ADRESSAGE DU RELAIS D'INDIRECTION  $AI := (C) + DS$
- 1 0 1 → ADRESSAGE DU RELAIS D'INDIRECTION  $AI := (L) + DS$
- 1 1 1 → ADRESSAGE DU RELAIS D'INDIRECTION  $AI := (W) + DS$

LE RELAIS D'INDIRECTION A LA CONFIGURATION SUIVANTE:



- 0  $AE := (AI)_{1-15}$  → ADRESSAGE INDIRECT SIMPLE
- 1  $AE := (AI)_{1-15} + (X)$  → ADRESSAGE INDIRECT POST-INDEXE

## CODAGE DES REGISTRES

$R_s - R_d - R_L - R_e$

A	B	X	Y	C	L	W	K
000	001	010	011	100	101	110	111

### CODES D'ORDRE



N° Document

Date

Page

71 F7 31MS

547

P. 7.2

## REGISTRE ST

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MS	SVCS	PR	LCM	IPM	IΦM	V	C	DEBUG	/	MAINT	SCHED	STΦP	/	POP	/

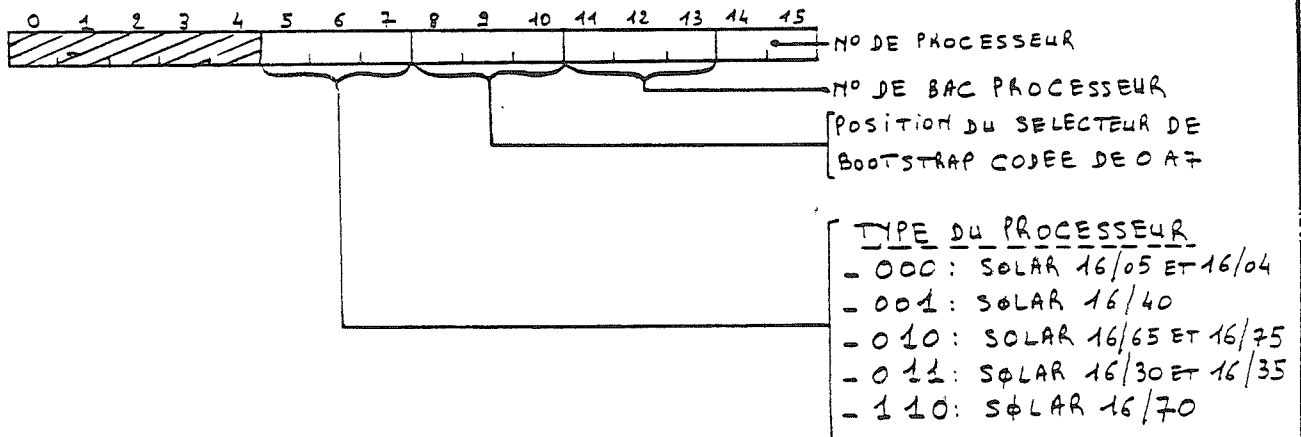
0	0	0	MODE ESCLAVE		
0	0	1	MODE PRIVILEGIE		
1	X	X	MODE MAITRE		

DANS LE CAS OÙ LE MODE PRIVILEGIE EST INEXISTANT (SOLAR AUTRE QUE L'UC. 16/70), LA POSITION DU BIT PR EST NON SIGNIFICATIVE.

- LCM: MASQUE DES INTERRUPTIONS DU CANAL LDC (A"1", INTERRUPTIONS MASQUEES)
- IPM: MASQUE DES INTERRUPTIONS INTERPROCESSEURS (A"1", INTERRUPTIONS MASQUEES)
- IΦM: MASQUE GLOBAL DES INTERRUPTIONS DE PROGRAMME Y COMPRENANT CERTAINS DEFECTS SECTEURS (A"1", INTERRUPTIONS MASQUEES)
- V ET C: INDICATEURS POSITIONNES PAR CERTAINES INSTRUCTIONS
- DEBUG: A"1", LE PROCESSEUR EST EN MODE MISE AU POINT. LES ERREURS DE PARITE MEMOIRE NE SONT PLUS INTERPRETEES COMME TELLES MAIS COMME DES POINTS D'ARRET.
- MAINT: BIT NORMALEMENT A"0"; A"1" LE PROCESSEUR PASSE DANS UN MODE RESERVE A LA MAINTENANCE OÙ LES ERREURS DE PARITE NE PROVOQUENT PLUS D'ALARMS OU DE POINTS D'ARRET.
- SCHED: A"1" SI LE PROCESSEUR A ETE INTERROMPU AU COURS DE SON PASSAGE DANS LE SCHEDULER.
- STΦP: A"0" LE PROCESSEUR EST EN MARCHE PROGRAMME ET EXECUTE DONC DES INSTRUCTIONS; A"1" IL EST A L'ARRET PROGRAMME ET NE TRAITE QUE LES APPELS CANAUX EVENTUELS
- POP: SIGNIFICATIF SI DEBUG=1. DANS CE CAS, A"1" MODE MISE AU POINT PURITRE, A"0" MODE MISE AU POINT PROGRAMME.

N.B: ST = '9C00 A L'INIT CALCULATEUR EN CAS DE RESTART.

## STATUS B LU PAR RDSI



**Bull**



**SPS 5**

CODES D'ORDRE

N° Document

71 F7 31MS

Date

547

Page

P. 7.3

# INSTRUCTIONS DE SECTION ISP 46

## INSTRUCTIONS BASEES

BLA	based load A
BSTA	based store A
BXM	based exchange memory with A
BOLD	based double load
BOST	based double store
BLBY	based load byte
BSTBY	based store byte
BMOVE	based move
BRBTM	based reset bit in memory
BSBTM	based set bit in memory
BDBTM	based discover bit in memory

## INSTRUCTIONS SUR CDA

CLA	CDA load A
CSTA	CDA store A
CXM	CDA exchange memory with A
COLD	CDA double load
COST	CDA double store
CLBY	CDA load byte
CSTBY	CDA store byte

## INSTRUCTIONS DIVERSES

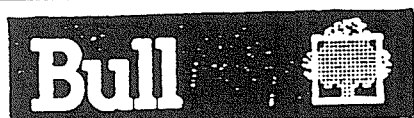
XCTX	commutation de contexte appelant
XENT	entrée de procédure langage
XSOR	sortie de procédure langage

## INSTRUCTIONS PRIVILEGIEES (NON EXECUTABLES EN MODE ESCLAVE)

ACQ	acquit	MVTM	move to master locations
ACT	activate	MVTS	move to slave locations
ARM	arm	RBP	reset breakpoint
BLA	based load A	ROHV	read HV
BSTA	based store A	RDSI	read system identification
BXM	based exchange memory with A	RLSE	release
BLBY	based load byte	RDOE	read SLO SLE
BSTBY	based store byte	ROMB	read-only memory branch
BOLD	based double load	RQST	request
BOST	based double store	RST	read ST
BMOVE	based move	SBP	set breakpoint
BDBTM	based discover bit in memory	SIO	start IO
BRBTM	based reset bit in memory	SST	set ST
BSBTM	based set bit in memory	STAR	store A relative
DBP	discover breakpoint	STEP	step
DIT	disable interrupt	WAIT	wait
EIT	enable interrupt	WOE	write SLO SLE
IPI	inter-processor interrupt	XCTX (1)	commutation de contexte appelant
LAR	load A relative	XIMR	exchange IM with register

(1) Exécutable uniquement en mode maître

### CODES D'ORDRE



SPS5

N° Document

71 F7 31MS

Date

547

Page

P. 7.4

ACK	M	P	E	05	30	35	40	65	70	(X) := N° DE S/MIV 0 A15 SI NORMAL 0 A47 SI EXCEPTION (X) INCHANGE SI PAS DE S/MIV SOUS TACHE SOFT, RAZ VETC INTERDITE DANS TACHE HARD	V	C	0 0 PAS DE S/MIV 1 0 S/M NORMAL 0 1 S/M EXCEPTION
	RECONNAISSANCE S/MIV. HARD										0	0	
0,0,0,1 1,1,1,0 0,0,0,0 1,1,0,1													

ACQ	M	P	<input checked="" type="checkbox"/>	05	30	35	40	65	70	RAZ DU BIT DE MIV CORRESPONDANT A LA TACHE QUI EXECUTE ACQ RAZ IPM ET IPM DE ST LANCEMENT DE LA TACHE HARD OU SOFT LA PLUS PRIORITAIRE
	ACQUITTEMENT D'UNE INTERRUPTION									
0,0,0,1 1,1,1,0 0,0,0,0 0,0,0,0										

ACT	M	P	<input checked="" type="checkbox"/>	05	30	35	40	65	70	- Si (AE) = 1, MISE A 1 DU BIT DONT LE N° EST DANS Y. - $\beta = \beta + 1$ - Si $\beta > 0$ POURSUITE DE LA TACHE - Si $\beta \leq 0$ EST F $\alpha = 1$ { N° > 2 POURSUITE TACHE { N° < 2 LANCEMENT SCHEDULE - RAZ IPM ET IPM DE ST - (AE) PAIS = $\beta$ = COMPTEUR ET (AE) 1A7 = $\alpha$ = N° TACHE
	ACTIVATION D'UNE TACHE SOFT									
MA 1 1,1,0,0 ± D										

ACTD	M	P	E	05	30	35	40	65	70	CHANGEMENT DE CONTEXTE LANCEMENT TACHE HARD LE N° DE S/MIV EST DANS LA MEMOIRE DU CONTEXTE DE LA TACHE HARD (OCTET DROIT)
	APPEL TACHE HARD O S/MIV = 9									
0,0,0,1 1,1,1,0 0,0,0,0 0,1,0,1										

AD	M	P	E	05	30	35	40	65	70	(A) := (A) + (AE)	V	C	1 1 DEBORDEMENT REPORT
	ADDITION										1	1	
MA 0 1,0,0,1 ± D													

ADCR	M	P	E	05	30	35	40	65	70	(Rd) := (Rd) + C C = bit 7 DE ST	V	C	1 1 DEBORDEMENT REPORT
	ADDITION DE L'INDICATEUR C AU REGISTRE DESTINATION (Rd)										1	1	
0,0,1,0 1,1,0,1 1,0,0,0 0, Rd													

ADR	M	P	E	05	30	35	40	65	70	(Rd) := (Rs) + (Rd)	V	C	1 1 DEBORDEMENT REPORT
	ADDITION DE REGISTRES										1	1	
0,0,1,0 1,1,0,0 0,0 Rs Rd													

ADRI	M	P	E	05	30	35	40	65	70	(Rd) := (Rd) ± V LE BIT 2 (SIGNE) EST ETENDU AUX BITS 0 A 7.	V	C	1 1 DEBORDEMENT REPORT
	ADDITION IMMEDIATE A UN REGISTRE										1	1	
0,0,0,0 1 Rd ± V													

ADR2	M	P	E	05	30	35	40	65	70	$(P) := (Rs) - (R)$ - LA VALEUR DE R EST CELLE DE L'OPERATION ELLE MEME
	ADDITION D'UN REGISTRE A R									
0 0 1 0 1 1 1 1 1 0 Rs 0 0 0										

AND	M	P	E	05	30	35	40	65	70	$(A) := (A) \cap (AE)$ - INTERSECTION DE 1 ET 1 VAUT 1, LES INTERSECTIONS DE 1 ET 0, 0 ET 1, 0 ET 0 VALENT 0
	INTERSECTION									
MA 1 0 0 1 1 1 1 D										

ANDI	M	P	E	05	30	35	40	65	70	$(A) := (A) \cap V$ - LE BIT 2 DE V EST ETENDU DE 0 A 7 - INTERSECTION DE 1 ET 1 VAUT 1, LES INTERSECTIONS DE 1 ET 0, 0 ET 1, 0 ET 0 VALENT 0
	INTERSECTION IMMEDIATE									
0 0 1 V 1 0 0 1 1 V										

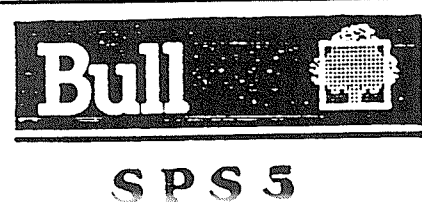
ANDR	M	P	E	05	30	35	40	65	70	$(Rd) := (Rd) \cap (Rs)$ - INTERSECTION DE 1 ET 1 VAUT 1, LES INTERSECTIONS DE 1 ET 0, 0 ET 1, 0 ET 0 VALENT 0
	INTERSECTION ENTRE REGISTRES									
0 0 1 0 1 1 0 1 0 0 Rs Rd										

ARM	M	P	<input checked="" type="checkbox"/>	05	30	35	40	65	70	- n : NUMERO DE TACHE - $ASTF(n) = 1$ si BIT 2 = 1 $n = n + (X)$ - RAZ = 0n ET 1n DE ST - LA TACHE EN COURS EST POURSUIVIE SI ELLE EST PLUS PRIORITAIRE QUE n OU SI $ESTF(n) = 0$ S'IL Y A LANCHEMENT SCHEDULER
	ARRIVEMENT D'UNE TACHE SOFT									
0 0 0 1 1 1 1 1 X n										

BR	M	P	E	05	30	35	40	65	70	$(P) := (AE)$
	BRANCHEMENT									
MA 0 0 1 0 1 1 D										

BSR	M	P	E	05	30	35	40	65	70	SI MS = 1 ET SVCS = 0 $(K) := (K) + 1$ $(P) := (AE)$ $(K) := (P) + 1$ SI MS = 0 ET SVCS = 1 $(K) := (K) + 1$ $(P) := (AE)$ $(K) := (K) - 1$ LES BITS MS ET SVCS SONT ECHANGEES LE RETOUR JEVA SE FAIRE PAR RSR
	BRANCHEMENT A UN SOUS PROGRAMME									
MA 0 0 1 1 0 1 D										

CLSR	M	P	E	05	30	35	40	65	70	$(Rd) := (Rd) \cap (\overline{Rs})$ LES BITS A 1 DU Rs METTENT A 0 LES BITS CORRESPONDANTS DU Rd
	MISE A ZERO SELECTIVE DES BITS D'UN REGISTRE									
0 0 1 0 1 1 0 1 0 1 Rs Rd										



CODES D'ORDRE		
N° Document	Date	Page
71 F7 31MS	547	P. 7.6

CMR	M	P	E	05	30	35	40	65	70	$(Rd) := (\overline{Rs})$
	COMPLEMENTATION DE REGISTRES									
0 0 1 0 1 1 1 0 1 0 Rs Rd										

CP	M	P	E	05	30	35	40	65	70	$(A) :: (AE)$	V C	0 1	$(A) < (AE)$
	COMPARAISON ALGEBRAIQUE DE A											1 0	$(A) = (AE)$
MA 1 0 1 0 1 ± D										0 0	$(A) > (AE)$		

CPBY	M	P	E	05	30	35	40	65	70	$(AE) :: (A) \& 15$ AVEC BITS 0-7 A 0 ADRES. DIRECT: OCTET GAUCHE $AE = (BASE) \& D$ - ADRES. INDIRECT: OCTET GAUCHE $AE = (BASE) \& D$ - ADRES. INDIRECT INDEXE $AE = ((BASE) \& D) + (X)$ X PAIR: OCTET GAUCHE X IMPAIR: OCTET DROIT	V C	0 1	$(A) < OCTET$
	COMPARAISON ALGEBRAIQUE DE A AVEC LE CONTENU DE L'OCTET MEMOIRE ADRESSE											1 0	$(A) = OCTET$
MA 0 0 0 1 0 ± D										0 0	$(A) > OCTET$		

CPI	M	P	E	05	30	35	40	65	70	$(A) :: V$ LE BIT 2 EST ETENDU AUX BITS 0 A 7 COMPARAISON ALGEBRAIQUE	V C	0 1	$(A) < V$
	COMPARAISON IMMEDIATE DE A											1 0	$(A) = V$
0 0 V 1 0 1 0 1 V										0 0	$(A) > V$		

CPR	M	P	E	05	30	35	40	65	70	$(Rd) : (Rs)$ COMPARAISON ALGEBRAIQUE	V C	0 1	$(Rd) < (Rs)$
	COMPARAISON DE REGISTRES											1 0	$(Rd) = (Rs)$
0 0 1 0 1 1 1 0 1 1 Rs Rd										0 0	$(Rd) > (Rs)$		

CPZ	M	P	E	05	30	35	40	65	70	$(AE) :: 0$ COMPARAISON ALGEBRAIQUE	V C	0 1	$(AE) < 0$
	COMPARAISON D'UNE MEMOIRE A ZERO											1 0	$(AE) = 0$
MA 0 0 1 0 0 ± D										0 0	$(AE) > 0$		

CPZR	M	P	E	05	30	35	40	65	70	$(Rd) :: 0$ COMPARAISON ALGEBRAIQUE	V C	0 1	$(Rd) < 0$
	COMPARAISON D'UN REGISTRE A ZERO											1 0	$(Rd) = 0$
0 0 1 0 1 1 1 0 0 1 0 0 0 Rd										0 0	$(Rd) > 0$		

DPB	M	P	<del>D</del>	05	30	35	40	65	70	- SANS DAPS: (Y) = 1 <sup>er</sup> TESTEE - AVEC DAPS (Y) + (SLφ) = 1 <sup>er</sup> TESTEE (SLφ) = DERNIERE TESTEE - L'INSTRUCTION S'ETERMINE SI EPREVA PAIRTE ou NEPAIRTE ou si (Y) = 0	V C	0 1	ARRÊT SUR ARRÊT
	RECHERCHE DES POINTS D'ARRÊT HORS MODE DEBUG, AISQUE D'ALANES.											1 0	ARRÊT SUR MEMOIRE
0 0 0 1 1 1 1 0 0 0 0 1 0 0 0 0										0 0	ARRÊT SUR ARRÊT (Y) = 0		

CODES D'ORDRE



N° Document	Date	Page
71 F7 31MS	547	P. 7.7

DBT	M P E	05	30	35	40	65	70	$(X) = \text{N}^\circ \text{ DU } 1^{\text{er}} \text{ BIT}$ (EN BINAIRE)	V C
	RECHERCHE DU 1 <sup>er</sup> BIT A 1 DE A ET B								
0,0,0,1,1,1,1,0,0,0,0,0,0,0,1,1								X INCHANGE SI AU MOINS UN BIT A 1	0 1 UN BIT A 1
									0 1 AUCUN BIT A 1

DC	M P E	05	30	35	40	65	70	$(AE) := (AE) - 1$ BIT 15 A 1	V C
	DECREMENTATION MEMOIRE								
MA 1,1,0,0,1,1,1,1,1,1,1,1,1,1,1,1									0 1 $(AE) < 0$
									1 0 $(AE) = 0$
								0 0 $(AE) > 0$	

DIT	M P E	05	30	35	40	65	70	- MISE A 1 DU BIT IPM DE ST - INTERRUPTIONS + DEF AUT SECTEUR MASQUE - SI CPU 65 (ROM VU 32) OU CPU 70 ON MASQUE L'INTERRUPTION INTERPROCESSEUR - IM NON MODIFIE; CE MASQUE N'EST PAS MODIFIE PAR LES CHANGES
	INHIBITION DES INTERRUPTIONS							
0,0,0,1,1,1,1,0,0,0,0,1,0,0,1,1								

DRBM	M P E	05	30	35	40	65	70	- $(X) := \text{N}^\circ \text{ DU } 1^{\text{er}} \text{ BIT A 1}$ AAZ DU BIT DE V ETC SI RECHERCHE POSITIVE - $(A) = 0$ 1 <sup>er</sup> MOT DE LA CHAINE $(X) = \text{N}^\circ \text{ 1}^{\text{er}} \text{ BIT A CHERCHER}$ $(Y) = \text{N}^\circ \text{ DERNIER BIT DE LA CHAINE}$ - LA RECHERCHE SE FAIT PAR JOUETS.	V C
	RECHERCHE 1 <sup>er</sup> BIT A 1 D'UNE CHAINE DE BITS								
0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,1								0 1 PAS DE BIT A 1 $(X) := (Y)$	
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0									0 0 BIT TROUVE

DV	M P E	05	30	35	40	65	70	- QUOTIENT DANS A $(A) := (A, B) / (AE)$ - RESTE DANS B	V
	DIVISION								
MA 0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1								0 DIVISION CORRECTE	
									1 DIVISION IMPOSSIBLE

EIT	M P E	05	30	35	40	65	70	- MISE A 0 DU BIT IPM DE ST - DENASQUAGE DES INTERRUPTIONS ET DU DEF AUT SECTEUR - SI CPU 65 (ROM VU 32) OU CPU 70, DENASQUAGE INTERRUPTION INTERPROCESSEUR (IPM=0) - LE MASQUE IM N'EST PAS MODIFIE.
	REACTIVATION DES INTERRUPTIONS							
0,0,0,1,1,1,1,0,0,0,0,1,0,1,0,0								

EOR	M P E	05	30	35	40	65	70	- $(A) := (A) \oplus (AE)$ - LA DISJONCTION DE 1 ET 0, 0 ET 1 VAUT 1 - LA DISJONCTION DE 0 ET 0, 1 ET 1 VAUT 0
	DISJONCTION							
MA 1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0								

EORI	M P E	05	30	35	40	65	70	- $(A) := (A) \oplus V$ - LE BIT 2 DE V EST ETENDU AUX BITS 0 A 7 - LA DISJONCTION DE 1 ET 0, 0 ET 1 VAUT 1 - LA DISJONCTION DE 0 ET 0, 1 ET 1 VAUT 0
	DISJONCTION IMMEDIATE DE A							
0,0,V,1,0,1,0,0,0,0,0,0,0,0,0,0,0								



CODES D'ORDRE		
N <sup>o</sup> Document	Date	Page
71 F7 31MS	547	P. 7.8



EΦRR	M	P	E	05	30	35	40	65	70	- (Rd) := (Rd) ⊕ (As) - LA DISJONCTION DE 1 ET 0, 0 ET 1 VAUT 1 - LA DISJONCTION DE 0 ET 0, 1 ET 1 VAUT 0
	DISTINCTION DE REGISTRES									
0, 0, 1, 0, 1, 1, 0, 0, 1, 1									Rs, Rd	

EXTENSIONS	M	P	E	05	30	35	40	65	70	→ SI EXTENSION EXISTANTE - MS = 1 ET SVCS = 0 (K) := (K)+1, (X) := (X), (P) := (P)+2 - MS = 0 OU MS = 1 ET SVCS = 1 (K) := (K)+1, ((K)+S64) := (X), (P) := (P)+2. PUIS EXECUTION D'UNE SVC "n".
	HUIT COSES RESERVES AUX EXTENSIONS									
0, 0, 1, 1, 1									n, 1 <sup>er</sup> MOT	
									2 <sup>er</sup> MOT	

FABS	M	P	E	05	30	35	40	65	70	- (A, B) :=  (A, B)  - RAZ DE VET C
	VALEUR ABSOLUE DE A ET B									
0, 0, 1, 1, 1, 0, 0, 0									1 <sup>er</sup> MOT	
0, 0, 0, 0, 1, 0, 0, 0									2 <sup>er</sup> MOT	

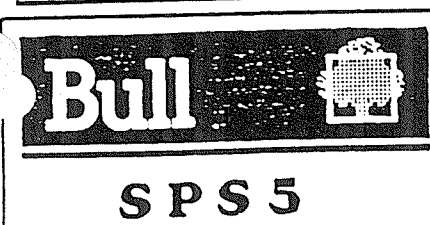
FAD	M	P	E	05	30	35	40	65	70	(A, B) := (A, B) + (AE, AE+1)	V C 1 0 OVERFLOW + BIT 1 0 1 UNDERFLOW + BIT 0 DU 1 <sup>er</sup> MOT CORRECT 0 0 CORRECT
	ADDITION FLOTTANTE										
0, 0, 1, 1, 1, 0, 0, 0									1 <sup>er</sup> MOT		
MA 0 0, 0, 1, 0									2 <sup>er</sup> MOT		

FCAM	M	P	E	05	30	35	40	65	70	(A, B) := (AE, AE+1)	V C 0 1 A, B < nb memoire 1 0 A, B = nb memoire 0 0 A, B > nb memoire
	COMPARAISON DE A, B A UN NOMBRE EN MEMOIRE										
0, 0, 1, 1, 1, 0, 0, 0									1 <sup>er</sup> MOT		
MA 0 1, 1, 1, 0									2 <sup>er</sup> MOT		

FCAZ	M	P	E	05	30	35	40	65	70	(A, B) := 0	V C 0 1 A, B < 0 1 0 A, B = 0 0 0 A, B > 0
	COMPARAISON DE A, B A ZERO										
0, 0, 1, 1, 1, 0, 0, 0									1 <sup>er</sup> MOT		
C, 0, 1, 0, 1, 0, 0, 0									2 <sup>er</sup> MOT		

FCMZ	M	P	E	05	30	35	40	65	70	(AE, AE+1) := 0	V C 0 1 nb < 0 1 0 nb = 0 0 0 nb > 0
	COMPARAISON A ZERO D'UN NOMBRE EN MEMOIRE										
0, 0, 1, 1, 1, 0, 0, 0									1 <sup>er</sup> MOT		
MA 0 1, 1, 0, 0									2 <sup>er</sup> MOT		

FDV	M	P	E	05	30	35	40	65	70	(A, B) := (A, B) / (AE, AE+1)	V C 1 0 OVERFLOW + BIT 2 0 1 UNDERFLOW + BIT 0 DU 1 <sup>er</sup> MOT CORRECT 0 0 CORRECT
	DIVISION FLOTTANTE										
0, 0, 1, 1, 1, 0, 0, 0									1 <sup>er</sup> MOT		
MA 0 0, 1, 1, 0									2 <sup>er</sup> MOT		



CODES D'ORDRE		
N° Document	Date	Page
71 F7 31MS	547	P. 7.9

FIX	M	P	E	<input checked="" type="checkbox"/>	30	F	35	40	F	65	F	70	F	(A) := FIX(A, B) (B) := 0	V	C	0 0 CORRECT 1 0 NOT CORRECT - BIT 3 (1 <sup>er</sup> MOT DU CORRECT)
	CONVERSION FLOTTANT → ENTIER																
	0,0,1,1	1,0,0,0												1 <sup>er</sup> MOT			
0,0,0,1	0,0,0,0												2 <sup>em</sup> MOT				

FLD	M	P	E	<input checked="" type="checkbox"/>	30	F	35	40	F	65	F	70	F	(A) := (AE) (B) := (AE+1) RAZ DE V ETC			
	CHARGEMENT DE A ET B																
	0,0,1,1	1,0,0,0												1 <sup>er</sup> MOT			
MA	0	0,0,0,0	±		D								2 <sup>em</sup> MOT				

FLT	M	P	E	<input checked="" type="checkbox"/>	30		35	40		65		70		(A, B) := FLOTTANT(A) RAZ DE V ETC			
	CONVERSION ENTIER → FLOTTANT																
	0,0,1,1	1,0,0,0												1 <sup>er</sup> MOT			
0,0,0,1	1,0,0,0												2 <sup>em</sup> MOT				

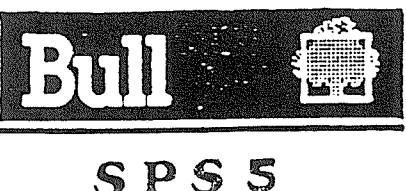
FMP	M	P	E	<input checked="" type="checkbox"/>	30	F	35	40	F	65	F	70	F	(A, B) := (A, B) x (AE, AE+1)	V	C	0 0 CORRECT 1 0 OVERFLOW - BIT 2 0 1 UNDERFLOW - BIT 2 DU 1 <sup>er</sup> MOT DU CORRECT
	MULTIPLICATION FLOTTANTE																
	0,0,1,1	1,0,0,0												1 <sup>er</sup> MOT			
MA	0	0,1,0,0	±		D								2 <sup>em</sup> MOT				

FNEG	M	P	E	<input checked="" type="checkbox"/>	30	F	35	40	F	65	F	70	F	(A, B) := -(A, B) RAZ DE V ETC			
	INVERSION DE SIGNE																
	0,0,1,1	1,0,0,0												1 <sup>er</sup> MOT			
0,0,0,0	0,0,0,0												2 <sup>em</sup> MOT				

FSB	M	P	E	<input checked="" type="checkbox"/>	30	F	35	40	F	65	F	70	F	(A, B) := (A, B) - (AE, AE+1)	V	C	0 0 CORRECT 1 0 OVERFLOW - BIT 2 0 1 UNDERFLOW - BIT 2 DU 1 <sup>er</sup> MOT DU CORRECT
	SOUSTRACTION FLOTTANTE																
	0,0,1,1	1,0,0,0												1 <sup>er</sup> MOT			
MA	0	1,0,1,0	±		D								2 <sup>em</sup> MOT				

FST	M	P	E	<input checked="" type="checkbox"/>	30		35	40		65		70		(AE) := (A) (AE+1) := (B) RAZ DE V ETC			
	RANGEMENT DE A ET B																
	0,0,1,1	1,0,0,0												1 <sup>er</sup> MOT			
MA	0	1,0,0,0	±		D								2 <sup>em</sup> MOT				

HALT	M	P	E		05	30	35	40	65	70	- L'UT N'EXECUTE PLUS D'INSTRUCTION PENDANT UNE TEMPO DEFINIE PAR (X) := n 1 < n < 65536 si n = 1 TEMPO DE 2 MS - FIN D'INSTRUCTION si (X) = 0, si IT LOC OU PROGRAMME, OU DEF AUT SECTEUR.				
	ATTENTE D'INTERUPTION														
	0,0,0,1	1,1,1,0	0,0,0,0	0,0,0,0	0,0,0,1										





JALE	M	P	E	05	30	35	40	65	70
	SAUT SI (A) ≤ 0								
0, 0, 1, 0, 0, 1, 1, 1 ± D									

si (A) ≤ 0 (P) := (P) ± D  
 si (A) > 0 (P) := (P) + 1

JANE	M	P	E	05	30	35	40	65	70
	SAUT SI (A) ≠ 0								
0, 0, 1, 0, 0, 0, 1, 0 ± D									

si (A) ≠ 0 (P) := (P) ± D  
 si (A) = 0 (P) := (P) + 1

JC	M	P	E	05	30	35	40	65	70
	SAUT SI REPORT								
0, 0, 0, 0, 0, 1, 0, 1 ± D									

si C = 1 (P) := (P) ± D  
 si C = 0 (P) := (P) + 1

MEME INSTRUCTION QUE JL

JCV	M	P	E	05	30	35	40	65	70
	SAUT SI REPORT OU DEBOAEMENT								
0, 0, 0, 0, 0, 1, 1, 1 ± D									

MEME INSTRUCTION QUE JE

V	C
0	1
1	0
0	0

(P) := (P) ± D  
 (P) := (P) + 1

JDX	M	P	E	05	30	35	40	65	70
	DECREMENTATION DE X								
0, 0, 0, 1, 1, 0, 0, 1 ± D									

(X) := (X) - 1  
 si (X) > 0 (P) := (P) ± D  
 si (X) ≤ 0 (P) := (P) + 1  
 si X = 0 PAS DE SAUT

JE	M	P	E	05	30	35	40	65	70
	SAUT SI EGAL								
0, 0, 0, 0, 0, 1, 1, 0 ± D									

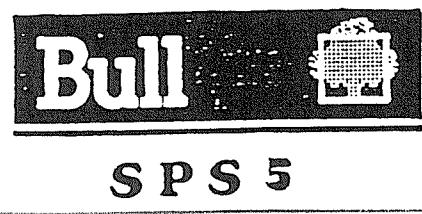
- CETTE INSTRUCTION SUIT UNE COMPARAISON  
 - MEME INSTRUCTION QUE JV

JG	M	P	E	05	30	35	40	65	70
	SAUT SI PLUS GRAND								
0, 0, 0, 0, 0, 0, 1, 1 ± D									

- CETTE INSTRUCTION SUIT UNE COMPARAISON  
 - MEME INSTRUCTION QUE JMCV

JGE	M	P	E	05	30	35	40	65	70
	SAUT SI PLUS GRAND OU EGAL								
0, 0, 0, 0, 0, 0, 0, 1 ± D									

- CETTE INSTRUCTION SUIT UNE COMPARAISON  
 - MEME INSTRUCTION QUE JMC



CODES D'ORDRE

N° Document	Date	Page
71 F7 31MS	547	P. 7.12

JIX	M	P	E	05	30	35	40	65	70	$(X) := (X) + 1$ si $(X) < 0$ $(P) := (P) \pm D$ si $(X) \geq 0$ $(P) := (P) + 1$ si $X = 0$ pas de SAUT
	INCRÉMENTATION DE X									
	0,0,0,1   1,0,0,0   ±   D									

JL	M	P	E	05	30	35	40	65	70	- CETTE INSTRUCTION SUIT UNE COMPARAISON - MÊME INSTRUCTION QUE JC
	SAUT SI PLUS PETIT									
	0,0,0,0   0,1,0,1   ±   D									

JLE	M	P	E	05	30	35	40	65	70	- CETTE INSTRUCTION SUIT UNE COMPARAISON - MÊME INSTRUCTION QUE JCV
	SAUT SI PLUS PETIT OU ÉGAL									
	0,0,0,0   0,1,1,1   ±   D									

JMP	M	P	E	05	30	35	40	65	70	- $(P) := (P) \pm D$ - AUTRE CODE INSTRUCTION POSSIBLE 20..
	SAUT INCONDITIONNEL									
	0,0,0,0   0,0,0,0   ±   D									

JNC	M	P	E	05	30	35	40	65	70	- si $C = 0$ $(P) := (P) \pm D$ - si $C = 1$ $(P) := (P) + 1$ - MÊME INSTRUCTION QUE JGE
	SAUT SI PAS DE REPORT									
	0,0,0,0   0,0,0,1   ±   D									

JNCV	M	P	E	05	30	35	40	65	70	- MÊME INSTRUCTION QUE JG	<table border="1"> <tr><th>V</th><th>C</th></tr> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> </table> $(P) := (P) \pm D$ $(P) := (P) + 1$	V	C	0	0	1	0	0	1
	V	C																	
	0	0																	
1	0																		
0	1																		
SAUT SI PAS DE REPORT ET PAS DE DEBORDEMENT																			
0,0,0,0   0,0,1,1   ±   D																			

JNE	M	P	E	05	30	35	40	65	70	- CETTE INSTRUCTION SUIT UNE COMPARAISON - MÊME INSTRUCTION QUE JNV
	SAUT SI DIFFÉRENT									
	0,0,0,0   0,0,1,0   ±   D									

JNV	M	P	E	05	30	35	40	65	70	- si $V = 0$ $(P) := (P) \pm D$ - si $V = 1$ $(P) := (P) + 1$ - MÊME INSTRUCTION QUE JNE
	SAUT SI PAS DE DEBORDEMENT									
	0,0,0,0   0,0,1,0   ±   D									



CODES D'ORDRE		
N° Document	Date	Page
71 F7 31MS	547	P. 7.13

JV	M	P	E	05	30	35	40	65	70	- SI $V = 1$ $(R) := (R) \pm D$ - SI $V = 0$ $(R) := (R) + 1$ - MÊME INSTRUCTION QUE JE
	SAUT SI DEBOARDÈMENT									
0,0,0,0,0,1,1,0,±, D										

LA	M	P	E	05	30	35	40	65	70	$(A) := (AE)$
	CHARGEMENT DE A									
MA 1,0,0,0,0,±, D										

LAD	M	P	E	05	30	35	40	65	70	$(A) := AE$ C'EST LA VALEUR DE L'W EFFECTIVE QUI EST CHARGÉE ET NON SON CONTENU
	CHARGEMENT DE A PAR L'ADRESSE EFFECTIVE									
MA 1,1,1,0,1,±, D										

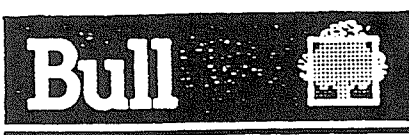
LAI	M	P	E	05	30	35	40	65	70	$(A) := V$ - LE BIT 2 (SIGNE) EST ÉTENDU AUX BITS 0 À 7
	CHARGEMENT IMMÉDIAT DE A									
0,0,V,1,0,0,0,0, V										

LAR	M	P	E	05	30	35	40	65	70	$(A) := (Y) \pm (SL\phi)$ $(Y) = W$ RELATIVE À $SL\phi$ SI DRPS $(Y) = W$ ABSOLUE SI PAS DRPS - ALARME DRPS SI DEBOARDÈMENT
	CHARGEMENT DE A PAR LE CONTENU D'UNE ADRESSE RELATIVE									
0,0,0,1,1,1,1,0,0,0,0,1,1,0,1,0										

LB	M	P	E	05	30	35	40	65	70	$(B) := (AE)$
	CHARGEMENT DE B									
MA 1,0,1,1,1,±, D										

LBI	M	P	E	05	30	35	40	65	70	$(B) := V$ LE BIT 2 (SIGNE) EST ÉTENDU AUX BITS 0 À 7.
	CHARGEMENT IMMÉDIAT DE B									
0,0,V,1,0,1,1,1, V										

LBY	M	P	E	05	30	35	40	65	70	$(A)_{8-15} := (AE)_b$ $(A)_{0-7} = 0$ $- AE_b = (BASE \pm D) \pm (X/2)$ NON INDEXÉ : OCTET GAUCHE INDEXÉ : X PAIR OCTET GAUCHE X IMPAIR OCTET DROIT
	CHARGEMENT D'OCTETS DANS A									
MA 0,0,0,0,0,±, D										



CODES D'ORDRE

N° Document	Date	Page
71 F7 31MS	547	P. 7.14

LR	M	P	E	05	30	35	40	65	70	(Rd) := (Rs)
	CHARGEMENT DE REGISTRE									
	0,0,1,0,1,0,1,1,1,1, Rs, Rd									

LRM	M	P	E	05	30	35	40	65	70	- LES REGISTRES SPECIFIES PAR UN 1 SONT CHARGES DE AVERS K PAR LES MOTS MEMOIRE QUI SUIVENT L'INSTRUCTION LRM - A LA FIN (R) := (R) + m + 1 (m) = nb de REGISTRES - INSTRUCTION INTERRUPTIBLE ENTRE CHAQUE REGISTRE
	CHARGEMENT MULTIPLE DE REGISTRES									
	0,0,0,1,1,1,0,1, A, B, X, Y, C, L, W, K									

LRP	M	P	E	05	30	35	40	65	70	- (Rd) := (R) - LA VALEUR DE R EST CELLE DE L'INSTRUCTION ELLE MEME.
	CHARGEMENT DE R DANS UN REGISTRE									
	0,0,1,0,1,1,1,1,1,1,0,0,0, Rd									

LX	M	P	E	05	30	35	40	65	70	(X) := (AE)
	CHARGEMENT DE X									
	MA 1,0,0,0,1, D									

LXI	M	P	E	05	30	35	40	65	70	(X) := V LE BIT 2 (SIGNE) EST ETENDU AUX BITS 0 A 7
	CHARGEMENT IMMEDIAT DE X									
	0,0,V,1,0,0,0,1, V									

LY	M	P	E	05	30	35	40	65	70	(Y) := (AE)
	CHARGEMENT DE Y									
	MA 1,0,1,1,0, D									

LYI	M	P	E	05	30	35	40	65	70	(Y) := V LE BIT 2 (SIGNE) EST ETENDU AUX BITS 0 A 7
	CHARGEMENT IMMEDIAT DE Y									
	0,0,V,1,0,1,1,0, V									

MOVE	M	P	E	05	30	35	40	65	70	- LE TRANSFERT SE FAIT PAR LES W DECREISSANTES - (A) = w DEBUT ZONE D'ORIGINE, (X) = nb de MOTS - (B) = w DEBUT ZONE DESTINATION - INSTRUCTION INTERRUPTIBLE ENTRE CHAQUE MOT. - INEFFECTIVE SI (X) <= 0 APRES DECREMENTATION.
	DEPLACEMENT D'UNE ZONE MEMOIRE									
	0,0,0,1,1,1,1,0,0,0,0,0,1,0,0,1									



CODES D'ORDRE		
N° Document	Date	Page
71 F7 31MS	547	P. 7.15

MP	M	P	E	05	30	35	40	65	70
	MULTIPLICATION								
M,A 0 1 1 1 0 = D									

$(A, B) := (A) \times (AE)$   
 $- V = C = 0$

MVTM	M	P	<del>E</del>	05	30	35	40	65	70
	TRANSLATION D'UNE ZONE ESCLAVE VERS UNE ZONE MAITRE								
0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0									

$(A) = \omega$  RELATIVE DEBUT ZONE D'ORIGINE  
 $\omega$  ABSOLUE =  $(A) + (SLD)$   
 $(B) = \omega$  ABSOLUE DEBUT ZONE DESTINATION  
 $(X) = nb$  DE MOTS A TRANSLATER  
 - INEFFECTIVE SI  $(X) \leq 0$  APRES DECREMENTATION  
 - ALARME DRPS SI DEBOURDEMENT

MVTS	M	P	<del>E</del>	05	30	35	40	65	70
	TRANSLATION D'UNE ZONE MAITRE VERS UNE ZONE ESCLAVE								
0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 1									

$(A) = \omega$  ABSOLUE DEBUT ZONE D'ORIGINE  
 $(B) = \omega$  RELATIVE DEBUT ZONE DESTINATION  
 $\omega$  ABSOLUE =  $(B) + (SLD)$   
 $(X) = nb$  DE MOTS A TRANSLATER  
 - INEFFECTIVE SI  $(X) \leq 0$  APRES DECREMENTATION  
 - ALARME DRPS SI DEBOURDEMENT

NGR	M	P	E	05	30	35	40	65	70
	OPPOSITION DE REGISTRES								
0 0 1 0 1 1 1 0 0 0 R <sub>s</sub> R <sub>d</sub>									

$(R_d) := -(R_s)$

VIC	1 1	$(R_s) = -32768$
	0 0	$(R_s) = 0$
	0 1	AUTRES CAS

NORM	M	P	E	<del>05</del>	30F	35	40F	65F	70F
	NORMALISATION								
0 0 1 1 1 0 0 0 0 1 <sup>er</sup> Mot									
0 0 1 0 0 0 0 0 0 2 <sup>er</sup> Mot									

$(A, B) := (A, B)$  NORMALISE

VIC	0 0	CORRECT
	1 0	OVERFLOW + BIT 1
	0 1	UNDERFLOW + BIT 0
		DU 1 <sup>er</sup> MOT DU CONTENU

NOP	M	P	E	05	30	35	40	65	70
	PASSAGE EN SEQUENCE								
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0									

$(P) := (P) + 1$

- AUTRE CODE INSTRUCTION POSSIBLE 2400

OR	M	P	E	05	30	35	40	65	70
	UNION								
M,A 1 0 0 1 0 = D									

$(A) := (A) \cup (AE)$

- UNION DE 1 ET 0, 0 ET 1, 1 ET 1 VAUT 1  
 - UNION DE 0 ET 0 VAUT 0

ORI	M	P	E	05	30	35	40	65	70
	UNION IMMEDIATE								
0 0 V 1 0 0 1 0 V									

$(A) := (A) \cup V$

- LE BIT 2 DE V EST ETENDU AUX BITS 0 A 7  
 - UNION DE 1 ET 0, 0 ET 1, 1 ET 1 VAUT 1  
 - UNION DE 0 ET 0 VAUT 0

CODES D'ORDRE

**Bull**



SPS 5

N° Document

71 F7 31MS

Date

547

Page

P. 7.16



ORR	M	P	E	05	30	35	40	65	70	- (Rd) := (Rd) U (Rs) - UNION DE 1 ETC, CET 1, 1 ET 1 VAUT 1 - UNION DE 0 ETC VAUT 0
	UNION DE REGISTRES									
0, 0, 1, 0 1, 1, 0, 0 1, 0 Rs, Rd										

PLR	M	P	E	05	30	35	40	65	70	- POUR CHAQUE REGISTRE SPECIFIE PAR UN 1 • SI MS = 1 ET SVCS = 0 $(REG) := ((K))$ $(K) := (K) - 1$ • SI MS = 0 OU MS = 1 ET SVCS = 1 $(REG) := ((K) + (SLD))$ $(K) := (K) - 1$ - LES REGISTRES SONT DEPILEES DANS L'ORDRE K A A.
	DEPILEMENT DE REGISTRES									
0, 0, 0, 1 1, 0, 1, 1 K, W, L, C, Y, X, B, A										

PSR	M	P	E	05	30	35	40	65	70	- POUR CHAQUE REGISTRE SPECIFIE PAR UN 1 • SI MS = 1 ET SVCS = 0 $(K) := (K) + 1$ $((K)) := (REG)$ • SI MS = 0 OU MS = 1 ET SVCS = 1 $(K) := (K) + 1$ $((K) + (SLD)) := (REG)$ - LES REGISTRES SONT EMPILES DANS L'ORDRE A A K
	EMPILEMENT DE REGISTRES									
0, 0, 0, 1 1, 0, 1, 0 A, B, X, Y, C, L, W, K										

PTY	M	P	E	05	30	35	40	65	70	- LE CALCUL SE FAIT SUR LES 8 BITS DE L'OCTET - V EST MIS A 0. - SI C = 0 OCTET PAIR SI C = 1 OCTET IMPAIR
	CALCUL DE LA PARITE DE L'OCTET DROIT DE A									
0, 0, 0, 1 1, 1, 1, 0 0, 0, 0, 0 1, 1, 1										

PULL	M	P	E	05	30	35	40	65	70	- (A) := (AE) puis n = n - 1 AVEC $(Y) = W$ DE DEBUT DE PILE $n = ((Y))_{8-15}$ : POINTEUR DE PILE $N = ((Y))_{0-7}$ : LONGUEUR DE PILE $AE = (Y) + n$	<table border="1"> <tr><td>V</td><td>C</td></tr> <tr><td>0</td><td>0</td><td>PILE NON VIDE</td></tr> <tr><td>0</td><td>1</td><td>PILE VIDE</td></tr> <tr><td>1</td><td>0</td><td>PILE DEJA VIDE</td></tr> </table>	V	C	0	0	PILE NON VIDE	0	1	PILE VIDE	1	0	PILE DEJA VIDE
	V	C																				
0	0	PILE NON VIDE																				
0	1	PILE VIDE																				
1	0	PILE DEJA VIDE																				
DEPILEMENT DANS A D'UNE PILE POINTEE PAR Y																						
0, 0, 0, 1 1, 1, 1, 0 0, 0, 0, 0 1, 1, 0, 0																						

PUSH	M	P	E	05	30	35	40	65	70	- n = n + 1 puis (AE) := (A) AVEC $(Y) = W$ DE DEBUT DE PILE $n = ((Y))_{8-15}$ : POINTEUR DE PILE $N = ((Y))_{0-7}$ : LONGUEUR DE PILE $AE = (Y) + n$	<table border="1"> <tr><td>V</td><td>C</td></tr> <tr><td>0</td><td>0</td><td>PILE NON PLEINE</td></tr> <tr><td>0</td><td>1</td><td>PILE PLEINE</td></tr> <tr><td>1</td><td>0</td><td>PILE DEJA PLEINE</td></tr> </table>	V	C	0	0	PILE NON PLEINE	0	1	PILE PLEINE	1	0	PILE DEJA PLEINE
	V	C																				
0	0	PILE NON PLEINE																				
0	1	PILE PLEINE																				
1	0	PILE DEJA PLEINE																				
EMPILEMENT DE A DANS UNE PILE POINTEE PAR Y																						
0, 0, 0, 1 1, 1, 1, 0 0, 0, 0, 0 1, 0, 1, 1																						

QUIT	M	P	E	05	30	35	40	65	70	- (ASTF) <sub>RS</sub> = 0 - SAUVEGARDE ET LANCEMENT DU SCHEDULEUR - MISE A 0 DES BITS IQM ET IPM DEST - INTERDIT SOUS TACHE HAAD
	ACQUITTEMENT D'UNE TACHE SOFT									
0, 0, 0, 1 1, 1, 1, 0 0, 0, 0, 0 0, 1, 1, 0										

RBP	M	P	<del>D</del>	05	30	35	40	65	70	- SANS DRPS $\curvearrowright$ POINT D'ARRET = (Y) - AVEC DRPS $\curvearrowright$ POINT D'ARRET = (Y) + (SLD) - HORS MODE DEBUG : ALARME PARITE - ALARME DRPS SI DEBORDEMENT
	EFFACEMENT D'UN POINT D'ARRET									
0, 0, 0, 1 1, 1, 1, 0 0, 0, 0, 1 0, 0, 1, 0										



CODES D'ORDRE		
N° Document	Date	Page
71 F7 31MS	547	P. 7.17

RBT	M	P	E	05	30	35	40	65	70
	MISE A 0 D'UN BIT DE A ET B								
0, 0, 1, 0, 1, 0, 0, 0, 0, 0, X, n									

$(A, B)_n = 0 \quad 0 \leq n \leq 31$   
 si bit 10 = 1  $n = n + (X)$

RBTM	M	P	E	05	30	35	40	65	70
	RAZ D'UN BIT D'UNE CHAÎNE DE BITS								
0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1 1 <sup>er</sup> Mot									
X, n 2 <sup>er</sup> Mot									

$(A) = 0$  du 1<sup>er</sup> MOT DE LA CHAÎNE  
 $n = n^o$  DU BIT DE LA CHAÎNE  
 si bit 0 = 1  $n = n + (X)$

RCDA	M	P	E	05	30	35	40	65	70
	TRANSFERT D'UN SEGMENT DE ZONE CPA VERS UNE ZONE MEMOIRE RELATIVE A LA TACHE								
0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0 1 <sup>er</sup> Mot									
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 2 <sup>er</sup> Mot									

$(A) = 0$  DEBUT ZONE D'ORIGINE  $\rightarrow$   $\omega$  RELATIVE AU DEBUT DE ZONE CPA ('0012' ('0012'): MEME FORMAT QUE SL $\phi$ )  
 $(B) = \omega$  DEBUT ZONE DESTINATION  $\rightarrow$   $\omega$  ABSOLUE ou RELATIVE SELON L'ETAT DE MS  
 $(X) = nb$  DE MOTS A TRANSFERER  
 - TRANSFERT PAR LES  $\omega$  DECAISSANTS, FIN DE TRANSFERT SUR  $(X) = 0$

RDHV	M	P	E	05	30	35	40	65	70
	CHARGEMENT DE A PAR HV								
0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1									

$(A) = (HV)$   
 $HV_i = 1 \rightarrow$  TACHE HARDWARE EN COURS DE TRAITEMENT  
 - BIT LE PLUS A GAUCHE  $\rightarrow$  TACHE HARD EN COURS  
 - LES AUTRES  $\rightarrow$  TACHES HARD INTERROMPUES

RDOE	M	P	E	05	30	35	40	65	70
	LECTURE DE SL $\phi$ ET DE SLE								
0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1									

$(A) := (SL\phi) \quad (B) := (SLE)$   
 - SANS DAPS : RAZ DE A ET B

A = SL $\phi$	512K	64K	32K	4K	2K	256	128	16
B = SLE	01010101	32K	4K	2K	256	128	16	

RDSI	M	P	E	05	30	35	40	65	70
	IDENTIFICATION PROCESSEUR								
0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1									

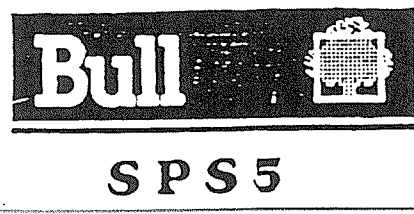
A: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  $\rightarrow$  MODU PROCESSEUR  
 TYPE DU PROCESSEUR  
 SOLAR 16105 // 16104 : 000  
 SOLAR 16140 : 001  
 SOLAR 16165 // 16175 : 010  
 SOLAR 16130 // 16135 : 011  
 SOLAR 16170 : 110  
 MODU BAC PROCESSEUR  
 POSITION DU SELECTEUR DE BOOTSTRAP CODEE DE 0 A 7

RLSE	M	P	E	05	30	35	40	65	70
	LIBERATION D'UN ACCES A UNE RESSOURCE								
MA 1 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1									

$P = P + 1$   
 - si  $P > 0$  POURSUITE DE LA TACHE  
 si  $P \leq 0$  RAZ BIT "n" PLUS PRIORITAIRE DANS FILE SEMAPHORE (ESTF) $_{h=1}$   
 - RAZ BITS P0M ET IPM DANS ST  
 - si MS > n POURSUITE TACHE, SINON LANCEMENTS SCHEDULE  
 $P = (AE)_{8-15} \quad (AE) :=$  SEMAPHORE D'EXCLUSION

ROMB	M	P	E	05	30	35	40	65	70
	BRANCHEMENT A UN MICRO PROGRAMME								
0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1									

$(Y)_{4-15} = \omega$  MICRO PROGRAMME A LANCER  
 - NE DOIT ETRE UTILISEE QUE POUR SE BRANCHER DANS DES MICRO PROGRAMMES SPECIFIQUES ET CONNUS



CODES D'ORDRE		
N <sup>o</sup> Document	Date	Page
71 F7 31MS	547	P. 7.18



SBP	M	P	E	05	30	35	40	65	70	- SANS JAPS $\omega$ POINT D'ARRÊT = 1 - AVEC JAPS $\omega$ POINT D'ARRÊT = $\gamma_i + S.C.$ - MODS MOJE DEBUE, ALAAME PARITE - ALAAME JAPS SI DEBUEMENT
	POSITIONNEMENT D'UN POINT D'ARRÊT									
0,0,0,1 1,1,1,0 0,0,0,1 0,0,0,1										

SBR	M	P	E	05	30	35	40	65	70	$(R_d) := (R_d) - (R_s)$ V/C 1: REPEAT 1: DEBOAEMENT
	SOUSTRATION DE REGISTRES									
0,0,1,0 1,1,0,0 0,1 R <sub>s</sub> R <sub>d</sub>										

SBS	M	P	E	05	30	35	40	65	70	(A) <sub>0-7</sub> = 0 OBLIGATOIEMENT (A) <sub>8-15</sub> = OCTET RECHACHE (B) = $\omega$ DEBUT DE ZONE (X) = NO <sup>14</sup> OCTET ACHACHEA (Y) = NO <sup>TOTAL</sup> D'OCTETS - si C = 1 L'OCTET N'A PAS ETE TROUVE OU X > Y - si C = 0 L'OCTET A ETE TROUVE (X) := AARE OCTET
	RECHACHE D'UN OCTET DANS UNE ZONE									
0,0,0,1 1,1,1,0 0,0,0,0 1,0,1,0										

SBT	M	P	E	05	30	35	40	65	70	$(A, B)_n := 1$ si BIT 10 = 1 $n = n + (X)$ MODULO 32
	MISE A 1 D'UN BIT DE A ET B									
0,0,1,0 1,0,0,0 0,1 X n										

SBTM	M	P	E	05	30	35	40	65	70	- (A) = $\omega$ DU 1 <sup>er</sup> MOT DE LA CHAINE - n = NO DU BIT DE LA CHAINE - si BIT 10 = 1 (2 <sup>em</sup> MOT) $n = n + (X)$
	MISE A 1 D'UN BIT D'UNE CHAINE DE BITS									
0,0,1,1 1,1,1,1 0,0,0,0 0,0,1,0 1 <sup>er</sup> Mot										
X n 2 <sup>em</sup> Mot										

SCLD	M	P	E	05	30	35	40	65	70	- n = nb DE PAS DE DECALAGE si BIT 10 = 1 $n = n + (X)$ MODULO 32 - LE DERNIER BIT SORTI EN A0 VAUT 0 si C=0, VAUT 1 si C=1 - si n=0 C=0
	DECALAGE FERME A GAUCHE DE A ET B									
0,0,1,0 1,0,1,0 1,1 X n										

SCLS	M	P	E	05	30	35	40	65	70	- n = nb DE PAS DE DECALAGE si BIT 10 = 1 $n = n + (X)$ MODULO 32 - LE DERNIER BIT SORTI EN A0 VAUT 0 si C=0, VAUT 1 si C=1 - si n=0 C=0
	DECALAGE FERME A GAUCHE DE A									
0,0,1,0 1,0,1,0 1,0 X n										

SCRD	M	P	E	05	30	35	40	65	70	- n = nb DE PAS DE DECALAGE si BIT 10 = 1 $n = n + (X)$ MODULO 32 - LE DERNIER BIT SORTI EN B15 VAUT 0 si C=0, VAUT 1 si C=1 - si n=0 C=0
	DECALAGE FERME A DROITE DE A ET B									
0,0,1,0 1,0,1,0 0,1 X n										



CODES D'ORDRE		
N° Document	Date	Page
71 F7 31MS	547	P. 7.20

SCRS	M	P	E	05	30	35	40	65	70
	DECALAGE FERME A DROITE DE A								
0, 0, 1, 0, 1, 0, 1, 0, 0, 0, X, n									

- n = nb DE PAS DE DECALAGE  
 si BIT 10 = 1 n = n + (X) MODULO 32  
 - LE DERNIER BIT SORTI EN A15 VAUT 0  
 si c = 0, VAUT 1 si c = 1  
 - si n = 0 c = 0

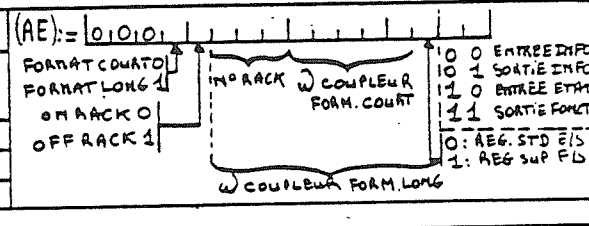
SCY	M	P	E	05	30	35	40	65	70
	MISE A 1 DE L'INDICATEUR C								
0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0									

- c = 1 = BIT 7 DE ST

SFQ	M	P	E	<del>05</del>	<del>30</del>	<del>35</del>	<del>40</del>	65	70
	SUPPRIMER DANS UNE LISTE L'ELEMENT PRECEDANT CELUI SPECIFIE								
0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1 <sup>14</sup> MOT									
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <sup>2</sup> MOT									

- (A) = 0 DE L'ELEMENT SPECIFIE  
 - (A) = 0 DE L'ELEMENT SUPPRAINE  
 INEFFECTIVE SI LA CHAINE A UN SEUL ELEMENT.

SIO	M	P	<del>X</del>	05	30	35	40	65	70
	LANCEMENT D'ENTREE SORTIE								
M, A 0 0, 1, 1, 1, 1, D									



SLLD	M	P	E	05	30	35	40	65	70
	DECALAGE LOGIQUE OUVERT A GAUCHE DE A ET B								
0, 0, 1, 0, 1, 0, 0, 1, 1, 1, X, n									

- n = nb DE PAS DE DECALAGE  
 si BIT 10 = 1 n = n + (X) MODULO 32  
 - LE DERNIER BIT SORTI EN A0 VAUT 0 si c = 0, VAUT 1 si c = 1  
 - si n = 0 c = 0

SLLS	M	P	E	05	30	35	40	65	70
	DECALAGE LOGIQUE OUVERT A GAUCHE DE A								
0, 0, 1, 0, 1, 0, 0, 1, 1, 0, X, n									

- n = nb DE PAS DE DECALAGE  
 si BIT 10 = 1 n = n + (X) MODULO 32  
 - LE DERNIER BIT SORTI EN A0 VAUT 0 si c = 0, VAUT 1 si c = 1  
 - si n = 0 c = 0

SLQ	M	P	E	<del>05</del>	<del>30</del>	<del>35</del>	<del>40</del>	65	70
	SUPPRIMER DANS UNE LISTE L'ELEMENT SUIVANT CELUI SPECIFIE								
0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1 <sup>14</sup> MOT									
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <sup>2</sup> MOT									

- (A) = 0 DE L'ELEMENT SPECIFIE  
 - (A) = 0 DE L'ELEMENT SUPRAINE  
 INEFFECTIVE SI LA CHAINE A UN SEUL ELEMENT

SLRD	M	P	E	05	30	35	40	65	70
	DECALAGE LOGIQUE OUVERT A DROITE DE A ET B								
0, 0, 1, 0, 1, 0, 0, 1, 0, 1, X, n									

- n = nb DE PAS DE DECALAGE  
 si BIT 10 = 1 n = n + (X) MODULO 32  
 - LE DERNIER BIT SORTI EN B15 VAUT 0 si c = 0, VAUT 1 si c = 1  
 - si n = 0 c = 0

SLRS	M	P	E	05	30	35	40	65	70
	DECALAGE LOGIQUE OUVERT A DROITE DE A								
0 0 1 0 1 0 0 1 0 0 X									

- n = nb DE PAS DE DECALAGE
- si bit 40 = 1 n = n + X MODULO 32
- LE DERNIER BIT SORTI EN PAS VAUT 0 SI C=0, VAUT 1 SI C=1
- si n=0 C=0

SST	M	P	E	05	30	35	40	65	70
	MISE A 1 SELECTIVE DE ST								
0 0 0 1 1 1 1 0 0 0 0 1 0 1 0 1									

- LES BITS A 1 DE A RETIENNENT A 1 LES BITS CORRESPONDANTS DE ST, PUIS CHARGEMENT DE LA NOUVELLE VALEUR DE ST DANS A
- LE FORCAGE DE CERTAINS BITS EST INEFFECTIF

STA	M	P	E	05	30	35	40	65	70
	RANGEMENT DE A								
M A 0 1 1 0 1 1									

$$(AE) := (A)$$

STAR	M	P	X	05	30	35	40	65	70
	RANGEMENT DE A DANS UNE ADRESSE RELATIVE								
0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 1									

- $((Y) + (SLP)) := (A)$
- AVEC DAPS PRESENT  $(Y) = \omega$  RELATIVE A SLP
- SANS DAPS  $(Y) = \omega$  ABSOLUE (SLP=0)
- ALARME DAPS SI DECALAGEMENT

STB	M	P	E	05	30	35	40	65	70
	RANGEMENT DE B								
M A 0 1 1 0 0 1									

$$(AE) := (B)$$

STBY	M	P	E	05	30	35	40	65	70
	RANGEMENT D'UN OCTET								
M A 0 0 0 0 1 1									

- $(AE) := (A)_{8-15}$
- $AE = (BASE = D) + \frac{(X)}{2}$
- NON INDEXE : OCTET GAUCHE
- INDEXE : X PAIR → OCTET GAUCHE, X IMPAIR → OCTET DROIT

STEP	M	P	X	05	30	35	40	65	70
	PAS A PAS INSTRUCTION								
0 0 0 1 1 1 1 0 0 0 0 0 0 1 0 0									

- RAZ DU BIT DE LA TACHE DANS #V
- LANCEMENT SCHEDULER
- EXECUTION D'UNE INSTRUCTION DE LA TACHE LA PLUS PAUCRITAIRES
- LANCEMENT THO S/M Z
- RAZ DES BITS IOM ET IPM DE ST
- INSTRUCTION INTERDITE DANSTACHE SOFT

STX	M	P	E	05	30	35	40	65	70
	RANGEMENT DE X								
M A 0 1 0 1 1 1									

$$(AE) := (X)$$



CODES D'ORDRE		
N° Document	Date	Page
71 F7 31MS	547	P. 7.22

STY	M	P	E	05	30	35	40	65	70	(AE) := (Y)
	RANGEMENT DE Y									
M A 0 1 0 1 0 1 0 1 0 1 D										

STZ	M	P	E	05	30	35	40	65	70	(AE) := 0
	RAZ D'UN MOT MEMOIRE									
M A 1 1 1 1 0 1 0 1 0 1 D										

SUPQ	M	P	E	<del>05</del>	<del>30</del>	<del>35</del>	<del>40</del>	<del>65</del>	<del>70</del>	- (A) = W DE L'ELEMENT A SUPPRIMER - INEFFECTIVE SI LA CHAINE A 1 SEUL ELEMENT
	SUPPRIMER DANS UNE LISTE L'ELEMENT SPECIE									
0,0,1,1,1,1,1,0,0,0,0,0,1,0,1 1 <sup>er</sup> MOT										
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 2 <sup>er</sup> MOT										

SVC	M	P	E	05	30	35	40	65	70	(X) := n - si MS=0 → MS:=1 SVCS:=1 (K):=(K)+1 ((K)+(SLA)) = (P)+1 • BR=0 → (P) = ('05) • BR=1 → (P) = ('1F) - si MS=1 ET SVCS=0 → (K)=(K)+1 ET ((K))=(P)+1 (P)=(P) - si MS=1 ET SVCS=1 → (K)=(K)+1 ET ((K)+(SLA))=(P)+1 (P) = ('04)
	APPEL DU SUPERVISEUR									
0,0,0,1,1,1,0,0,0,0,0,0,0,0,0 n										

SWBR	M	P	E	05	30	35	40	65	70	- (Rd) <sub>0-7</sub> := (Rs) <sub>8-15</sub> (Rd) <sub>8-15</sub> := (Rs) <sub>0-7</sub>
	ECHANGE D'OCTETS									
0,0,1,0,1,1,1,1,0,0,0,0,0,0,0 Rs Rd										

TBT	M	P	E	05	30	35	40	65	70	- n = n° DE BIT TESTE - si BIT 10 = 1 n = n + (X) MODULO 32 - si C=0 LE BIT VAUT 0 - si C=1 LE BIT VAUT 1
	TEST D'UN BIT DE A ET B									
0,0,1,0,1,0,0,0,1,1,1,1,0,0,0 n										

WAIT	M	P	<del>E</del>	05	30	35	40	65	70	- (d) := (MS) NO TACHE DANS SEMAPHORE - β = P-1 - si β > 0 POURSUITE DE LA TACHE - si β < 0 → (ESTF)MS = 0 ET LANCEMENT SCHEDULER - RAZ BITS SPM ET EPM DANS ST - si SEMAPHORE PARAMETRE GESTION FILE PAR PROGRAMMEUR - (AE) = SEMAPHORE β = (AE) <sub>8-15</sub> α = (AE) <sub>1-7</sub>
	MISE EN ATTENTE D'UNE TACHE SOFT									
M A 1 1 1 1 1 1 1 1 1 1 D										

WCDA	M	P	E	<del>05</del>	<del>30</del>	<del>35</del>	<del>40</del>	<del>65</del>	<del>70</del>	- (B) = W DEBUT ZONE ORIGINE → W ABSOLUE OU RELATIVE SELON L'ETAT DE MS - (A) = W DEBUT ZONE DESTINATION → W RELATIVE AU DEBUT DE ZONE CDA ('0018) - (X) = nb DE MOTS A TRANSFERER - TRANSFERT PAR LES W DECREOISSANTES - Fm DE TRANSFERT SUR (X) = 0 - ('0018) : MERE FORMAT QUE SLA
	TRANSFERT D'UN SEGMENT DE LA ZONE TACHE VERS LA ZONE CDA									
0,0,1,1,1,1,1,0,0,0,0,0,0,0,0 1 <sup>er</sup> MOT										
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 2 <sup>er</sup> MOT										

CODES D'ORDRE



N° Document	Date	Page
71 F7 31MS	547	P. 7.23

WOE	M	<del>P</del>	<del>E</del>	05	30	35	40	65	70	- (SLD) := (A) (SLE) := (B) - INEFFECTIVE si PAS DE CARTE A = SLD <table border="1"><tr><td>52K</td><td>64K</td><td>75K</td><td>86K</td><td>97K</td></tr></table> B = SLE <table border="1"><tr><td>72K</td><td>84K</td><td>95K</td><td>106K</td><td>117K</td></tr></table>	52K	64K	75K	86K	97K	72K	84K	95K	106K	117K
	52K	64K	75K	86K	97K															
72K	84K	95K	106K	117K																
CHARGEMENT DE SLD ET SLE																				
0,0,0,1,1,1,0,0,0,1,1,1,0																				

XIMR	M	<del>P</del>	<del>E</del>	05	30	35	40	65	70	(IM) := (Rd) (Rd) := (IM)
	ECHANGE DU REGISTRE IM AVEC UN REGISTRE									
0,0,1,0,1,1,1,1,0,1,0,0,0, Rd										

XM	M	P	E	05	30	35	40	65	70	(AE) := (A) (A) := (AE)
	ECHANGE DE A AVEC UNE MEMOIRE									
,MA 0,0,0,1,1,1, D										

XMR	M	P	E	<del>05</del>	<del>30</del>	<del>35</del>	<del>40</del>	65	70	((Y) + (SLD)) := (A) (A) := ((Y) + (SLD)) - DISPONIBLE SUR CPU 65 QUE SI LA CARTE CPU ROM EST AU V432
	ECHANGE DE A AVEC LE MOT MEMOIRE ADRESSE PAR Y + SLD									
0,0,1,1,1,1,1,0,0,1,0,1,1,0,0,0										

XR	M	P	E	05	30	35	40	65	70	(Rd) := (Rs) (Rs) := (Rd)
	ECHANGE DE REGISTRES									
0,0,1,0,1,0,1,1,1,0, Rs, Rd										

	M	P	E	05	30	35	40	65	70	

	M	P	E	05	30	35	40	65	70	

	M	P	E	05	30	35	40	65	70	



SDS 5

CODES D'ORDRE

N° Document	Date	Page
71 E7 31MS	547	P. 7.24



BDBTM	M P E	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES MOTS MEMOIRE POINTES PAR LES MOTS 2 ET 3	VC	0 0 BIT TROUVE EST AU RANG (X) 0 1 BIT A 1 NON RENCONTRE (X) = (Y) si (X) > Y RECHERCHE NEGATIVE
	RECHERCHE A PARTIR DU RANG (X) JUSQU'AU RANG (Y)-1 DU 7 <sup>e</sup> BIT A 1 DANS UNE FILE DE BITS EN MEMOIRE							AE20 = PARAM1 * 16 + PARAM2 PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS	0 0	
	0, 0, 1, 1	1, 1, 0, 0	0, 1, 0, 0	0, 1, 1, 1	1 <sup>er</sup> MOT			1 <sup>er</sup> MOT		
MA	0, 0, 0, 0	±	D	2 <sup>em</sup> MOT			2 <sup>em</sup> MOT			
MA	0, 0, 0, 0	±	D	3 <sup>em</sup> MOT			3 <sup>em</sup> MOT			

BDBTM	M P E	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES REGISTRES R1 ET R2 DESIGNES PAR LE MOT 2	VC	0 0 BIT TROUVE EST AU RANG (X) 0 1 BIT A 1 NON RENCONTRE (X) = (Y) si (X) > Y RECHERCHE NEGATIVE
	RECHERCHE A PARTIR DU RANG (X) JUSQU'AU RANG (Y)-1 DU 7 <sup>e</sup> BIT A 1 DANS UNE FILE DE BITS EN MEMOIRE							AE20 = PARAM1 * 16 + PARAM2 PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS	0 0	
	0, 0, 1, 1	1, 1, 0, 0	0, 0, 0, 0	0, 0, 0, 0	1, 1, 1, 1	1 <sup>er</sup> MOT		1 <sup>er</sup> MOT		
0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0	R1	R2	2 <sup>em</sup> MOT				

BDLD	M P E	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES MOTS MEMOIRE POINTES PAR LES MOTS 2 ET 3	- si BIT 11 = 0 AE20 = PARAM1 * 16 + PARAM2 - si BIT 11 = 1 AE20 = PARAM1 * 16 + (PARAM2 + X) PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS X = REGISTRE D'INDEX; (PARAM2 + X) EFFECTIVE SUR 16 BITS	
	LE CONTENU DU MOT ADRESSE EST CHARGE DANS A LE CONTENU DU MOT SUIVANT DANS B							AE20 = PARAM1 * 16 + PARAM2 PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS		
	0, 0, 1, 1	1, 1, 0, 0	0, 1, 0, X	0, 1, 0, 0	1 <sup>er</sup> MOT			1 <sup>er</sup> MOT		
MA	0, 0, 0, 0	±	D	2 <sup>em</sup> MOT			2 <sup>em</sup> MOT			
MA	0, 0, 0, 0	±	D	3 <sup>em</sup> MOT			3 <sup>em</sup> MOT			

BDLD	M P E	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES REGISTRES R1 ET R2 DESIGNES PAR LE MOT 2	- si BIT 11 = 0 AE20 = PARAM1 * 16 + PARAM2 - si BIT 11 = 1 AE20 = PARAM1 * 16 + (PARAM2 + X) PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS X = REGISTRE D'INDEX; (PARAM2 + X) EFFECTIVE SUR 16 BITS	
	LE CONTENU DU MOT ADRESSE EST CHARGE DANS A LE CONTENU DU MOT SUIVANT DANS B							AE20 = PARAM1 * 16 + PARAM2 PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS		
	0, 0, 1, 1	1, 1, 0, 0	0, 0, 0, X	0, 1, 0, 0	1 <sup>er</sup> MOT			1 <sup>er</sup> MOT		
0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0	R1	R2	2 <sup>em</sup> MOT				

BDST	M P E	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES MOTS MEMOIRE POINTES PAR LES MOTS 2 ET 3	- si BIT 11 = 0 AE20 = PARAM1 * 16 + PARAM2 - si BIT 11 = 1 AE20 = PARAM1 * 16 + (PARAM2 + X) PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS X = REGISTRE D'INDEX; (PARAM2 + X) EFFECTIVE SUR 16 BITS	
	LE CONTENU DE A EST RANGE DANS LE MOT ADRESSE, LE CONTENU DE B DANS LE MOT SUIVANT.							AE20 = PARAM1 * 16 + PARAM2 PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS		
	0, 0, 1, 1	1, 1, 0, 0	0, 1, 0, X	0, 1, 0, 1	1 <sup>er</sup> MOT			1 <sup>er</sup> MOT		
MA	0, 0, 0, 0	±	D	2 <sup>em</sup> MOT			2 <sup>em</sup> MOT			
MA	0, 0, 0, 0	±	D	3 <sup>em</sup> MOT			3 <sup>em</sup> MOT			

BDST	M P E	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES REGISTRES R1 ET R2 DESIGNES PAR LE MOT 2	- si BIT 11 = 0 AE20 = PARAM1 * 16 + PARAM2 - si BIT 11 = 1 AE20 = PARAM1 * 16 + (PARAM2 + X) PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS X = REGISTRE D'INDEX; (PARAM2 + X) EFFECTIVE SUR 16 BITS	
	LE CONTENU DE A EST RANGE DANS LE MOT ADRESSE, LE CONTENU DE B DANS LE MOT SUIVANT							AE20 = PARAM1 * 16 + PARAM2 PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS		
	0, 0, 1, 1	1, 1, 0, 0	0, 0, 0, X	0, 1, 0, 1	1 <sup>er</sup> MOT			1 <sup>er</sup> MOT		
0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0	R1	R2	2 <sup>em</sup> MOT				

BLA	M P E	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES MOTS MEMOIRE POINTES PAR LES MOTS 2 ET 3	- si BIT 11 = 0 AE20 = PARAM1 * 16 + PARAM2 - si BIT 11 = 1 AE20 = PARAM1 * 16 + (PARAM2 + X) PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS X = REGISTRE D'INDEX; (PARAM2 + X) EFFECTIVE SUR 16 BITS	
	LE CONTENU DU MOT MEMOIRE ADRESSE EST CHARGE DANS A							AE20 = PARAM1 * 16 + PARAM2 PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS		
	0, 0, 1, 1	1, 1, 0, 0	0, 1, 0, X	0, 0, 0, 0	1 <sup>er</sup> MOT			1 <sup>er</sup> MOT		
MA	0, 0, 0, 0	±	D	2 <sup>em</sup> MOT			2 <sup>em</sup> MOT			
MA	0, 0, 0, 0	±	D	3 <sup>em</sup> MOT			3 <sup>em</sup> MOT			

BLA	M P E	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES REGISTRES R1 ET R2 DESIGNES PAR LE MOT 2	- si BIT 11 = 0 AE20 = PARAM1 * 16 + PARAM2 - si BIT 11 = 1 AE20 = PARAM1 * 16 + (PARAM2 + X) PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS X = REGISTRE D'INDEX; (PARAM2 + X) EFFECTIVE SUR 16 BITS	
	LE CONTENU DU MOT MEMOIRE ADRESSE EST CHARGE DANS A							AE20 = PARAM1 * 16 + PARAM2 PARAM1 = BASE FORMAT SLO PARAM2 = DEPLACEMENT SUR 16 BITS		
	0, 0, 1, 1	1, 1, 0, 0	0, 0, 0, X	0, 0, 0, 0	1 <sup>er</sup> MOT			1 <sup>er</sup> MOT		
0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0	0, 0, 0, 0	R1	R2	2 <sup>em</sup> MOT				

CODES D'ORDRE



N° Document	Date	Page
71 F7 31MS	547	P. 7.25

<b>BLBY</b>	M	P	X	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES MOTS MEMOIRE POINTES PAR LES MOTS 2 ET 3 $AE20 = PARAM1 * 16 + PARAM2$ $PARAM1 = BASE FORMAT SLQ$ $PARAM2 = DEPLACEMENT SUR 16 BITS$ $X = REGISTRE D'INDEX; (PARAM2 * 16)$ EFFECTUEES SUR 16 BITS $X PAIR \rightarrow$ OCTET GAUCHE ; $X IMPAIR \rightarrow$ OCTET DROIT
	BITS 7 A 15 DE A SONT CHARGES PAR L'OCTET									
	MEMOIRE ADRESSE, BITS 0 A 7 DE A SONT MIS A ZERO									
	0,0,1,1	1,1,0,0	0,1,0,0	0,0,1,1	1 <sup>er</sup> Mot					
MA	0,0,0,0	±	D	2 <sup>em</sup> Mot						
MA	0,0,0,0	±	D	3 <sup>em</sup> Mot						

<b>BLBY</b>	M	P	X	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES REGISTRES R1 ET R2 DESIGNES PAR LE MOT 2 $AE20 = PARAM1 * 16 + (PARAM2 * (X/2))$ $PARAM1 = BASE FORMAT SLQ$ $PARAM2 = DEPLACEMENT SUR 16 BITS$ $X = REGISTRE D'INDEX; (PARAM2 * (X/2))$ EFFECTUEES SUR 16 BITS $X PAIR \rightarrow$ OCTET GAUCHE ; $X IMPAIR \rightarrow$ OCTET DROIT
	BITS 8 A 15 DE A SONT CHARGES PAR L'OCTET									
	MEMOIRE ADRESSE, BITS 0 A 7 DE A SONT MIS A ZERO									
	0,0,1,1	1,1,0,0	0,0,0,0	0,0,1,1	1 <sup>er</sup> Mot					
0,0,0,0	0,0,0,0	0,0	R2	R2	2 <sup>em</sup> Mot					

<b>BMOVE</b>	M	P	X	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES MOTS MEMOIRE POINTES PAR LES MOTS 2 ET 3 - OPERATION INEFFECTIVE SI (X) < 0 APRES DECREMENT - FIN DE TRANSFERT (X) = 0 - ZONE ORIGINE : $PARAM1 * 16 + A$ - ZONE DESTINATION : $PARAM2 * 16 + B$ - $PARAM1$ ET $PARAM2 =$ BASES EN FORMAT SLQ - $(A + (X-1))$ ET $(B - (X-1))$ EFFECTUEES SUR 16 BITS
	TRANSFERT DE X MOTS DANS LE SENS DES									
	ADRESSES DECAISSANTES POUR X DE 1 A 32768									
	0,0,1,1	1,1,0,0	0,1,0,0	0,1,1,0	1 <sup>er</sup> Mot					
MA	0,0,0,0	±	D	2 <sup>em</sup> Mot						
MA	0,0,0,0	±	D	3 <sup>em</sup> Mot						

<b>BMOVE</b>	M	P	X	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES REGISTRES R1 ET R2 DESIGNES PAR LE MOT 2 - OPERATION INEFFECTIVE SI (X) < 0 APRES DECREMENT - FIN DE TRANSFERT (X) = 0 - ZONE ORIGINE : $PARAM1 * 16 + A$ - ZONE DESTINATION : $PARAM2 * 16 + B$ - $PARAM1$ ET $PARAM2 =$ BASES EN FORMAT SLQ - $(A + (X-1))$ ET $(B - (X-1))$ EFFECTUEES SUR 16 BITS
	TRANSFERT DE X MOTS DANS LE SENS DES									
	ADRESSES DECAISSANTES POUR X DE 1 A 32768									
	0,0,1,1	1,1,0,0	0,0,0,0	0,1,1,0	1 <sup>er</sup> Mot					
0,0,0,0	0,0,0,0	0,0	R1	R2	2 <sup>em</sup> Mot					

<b>BRBTM</b>	M	P	X	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES MOTS MEMOIRE POINTES PAR LES MOTS 2 ET 3 $AE20 = PARAM1 * 16 + PARAM2$ $PARAM1 = BASE FORMAT SLQ$ $PARAM2 = DEPLACEMENT SUR 16 BITS$ - LE RANG DU BIT EST RELATIF AU BIT 0 DU 1 <sup>er</sup> MOT DE LA FILE
	MISE A ZERO DU BIT DE RANG (X) DANS									
	UNE FILE DE BITS EN MEMOIRE									
	0,0,1,1	1,1,0,0	0,1,1,0	0,1,1,1	1 <sup>er</sup> Mot					
MA	0,0,0,0	±	D	2 <sup>em</sup> Mot						
MA	0,0,0,0	±	D	3 <sup>em</sup> Mot						

<b>BRBTM</b>	M	P	X	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES REGISTRES R1 ET R2 DESIGNES PAR LE MOT 2 $AE20 = PARAM1 * 16 + PARAM2$ $PARAM1 = BASE FORMAT SLQ$ $PARAM2 = DEPLACEMENT SUR 16 BITS$ - LE RANG DU BIT EST RELATIF AU BIT 0 DU 1 <sup>er</sup> MOT DE LA FILE
	MISE A ZERO DU BIT DE RANG (X) DANS									
	UNE FILE DE BITS EN MEMOIRE									
	0,0,1,1	1,1,0,0	0,0,1,0	0,1,1,1	1 <sup>er</sup> Mot					
0,0,0,0	0,0,0,0	0,0	R1	R2	2 <sup>em</sup> Mot					

<b>BSBTM</b>	M	P	X	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES MOTS MEMOIRE POINTES PAR LES MOTS 2 ET 3 $AE20 = PARAM1 * 16 + PARAM2$ $PARAM1 = BASE FORMAT SLQ$ $PARAM2 = DEPLACEMENT SUR 16 BITS$ - LE RANG DU BIT EST RELATIF AU BIT 0 DU 1 <sup>er</sup> MOT DE LA FILE
	MISE A UN DU BIT DE RANG (X) DANS UNE									
	FILE DE BITS EN MEMOIRE									
	0,0,1,1	1,1,0,0	0,1,1,0	0,1,1,1	1 <sup>er</sup> Mot					
MA	0,0,0,0	±	D	2 <sup>em</sup> Mot						
MA	0,0,0,0	±	D	3 <sup>em</sup> Mot						

<b>BSBTM</b>	M	P	X	05	30	35	40	65	70	PARAMETRES 1 ET 2 SONT DANS DES REGISTRES R1 ET R2 DESIGNES PAR LE MOT 2 $AE20 = PARAM1 * 16 + PARAM2$ $PARAM1 = BASE FORMAT SLQ$ $PARAM2 = DEPLACEMENT SUR 16 BITS$ - LE RANG DU BIT EST RELATIF AU BIT 0 DU 1 <sup>er</sup> MOT DE LA FILE
	MISE A UN DU BIT DE RANG (X) DANS UNE									
	FILE DE BITS EN MEMOIRE									
	0,0,1,1	1,1,0,0	0,0,1,0	0,1,1,1	1 <sup>er</sup> Mot					
0,0,0,0	0,0,0,0	0,0	R1	R2	2 <sup>em</sup> Mot					

CODES D'ORDRE



SPS 5

N° Document

71 F7 31MS

Date

547

Page

P. 7.26

BSTA	M	P	X	05	30	35	40	65	70								
	LE CONTENU DE A EST RANGE DANS LE MOT MEMOIRE ADRESSE.																
	0	0	1	1	1	1	0	0	0	1	0	X	0	0	0	1	1 <sup>er</sup> Mot
	MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3 <sup>em</sup> Mot

PARAMETRES 1 ET 2 SONT DANS DES MOTS MEMOIRE POINTES PAR LES MOTS 2 ET 3

- si BIT 11 = 0  $AE_{20} = PARAM1 + 16 - PARAM2$
- si BIT 11 = 1  $AE_{20} = PARAM1 + 16 + (PARAM2 + X)$

PARAM1 = BASE FORMAT SL6  
PARAM2 = DEPLACEMENT SUR 16 BITS  
(X) = REGISTRE D'INDEX; (PARAM2 + X) EFFECTUEES SUR 16 BITS

BSTA	M	P	X	05	30	35	40	65	70								
	LE CONTENU DE A EST RANGE DANS LE MOT MEMOIRE ADRESSE.																
	0	0	1	1	1	1	0	0	0	0	0	X	0	0	0	1	1 <sup>er</sup> Mot
	MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3 <sup>em</sup> Mot

PARAMETRES 1 ET 2 SONT DANS DES REGISTRES R1 ET R2 DESIGNES PAR LE MOT 2

- si BIT 11 = 0  $AE_{20} = PARAM1 + 16 + PARAM2$
- si BIT 11 = 1  $AE_{20} = PARAM1 + 16 + (PARAM2 + X)$

PARAM1 = BASE FORMAT SL6  
PARAM2 = DEPLACEMENT SUR 16 BITS  
(X) = REGISTRE D'INDEX; (PARAM2 + X) EFFECTUEE SUR 16 BITS

BSTBY	M	P	X	05	30	35	40	65	70								
	LES BITS 8 A 15 DE A SONT RANGES DANS L'OCTET MEMOIRE ADRESSE																
	0	0	1	1	1	1	0	0	0	1	0	1	0	0	1	1	1 <sup>er</sup> Mot
	MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3 <sup>em</sup> Mot

PARAMETRES 1 ET 2 SONT DANS DES MOTS MEMOIRE POINTES PAR LES MOTS 2 ET 3

$AE_{20} = PARAM1 + 16 + (PARAM2 + (X/2))$

PARAM1 = BASE EN FORMAT SL6  
PARAM2 = DEPLACEMENT SUR 16 BITS  
(X) = REGISTRE D'INDEX; (PARAM2 + (X/2)) EFFECTUEES SUR 16 BITS  
X PAIR → OCTET GAUCHE; X IMPAIR → OCTET DROIT

BSTBY	M	P	X	05	30	35	40	65	70								
	LES BITS 8 A 15 DE A SONT RANGES DANS L'OCTET MEMOIRE ADRESSE.																
	0	0	1	1	1	1	0	0	0	1	0	0	1	0	0	1	1 <sup>er</sup> Mot
	MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3 <sup>em</sup> Mot

PARAMETRES 1 ET 2 SONT DANS DES REGISTRES R1 ET R2 DESIGNES PAR LE MOT 2

$AE_{20} = PARAM1 + 16 + (PARAM2 + (X/2))$

PARAM1 = BASE FORMAT SL6  
PARAM2 = DEPLACEMENT SUR 16 BITS  
(X) = REGISTRE D'INDEX; (PARAM2 + (X/2)) EFFECTUEES SUR 16 BITS  
X PAIR → OCTET GAUCHE; X IMPAIR → OCTET DROIT

BXM	M	P	X	05	30	35	40	65	70								
	LES CONTENUS DE A ET DU MOT ADRESSE SONT ECHANGES EN UN CYCLE MEMOIRE ININTERRUPTIBLE.																
	0	0	1	1	1	1	0	0	0	1	0	X	0	0	1	0	1 <sup>er</sup> Mot
	MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3 <sup>em</sup> Mot

PARAMETRES 1 ET 2 SONT DANS DES MOTS MEMOIRE POINTES PAR LES MOTS 2 ET 3

- si BIT 11 = 0  $AE_{20} = PARAM1 + 16 + PARAM2$
- si BIT 11 = 1  $AE_{20} = PARAM1 + 16 + (PARAM2 + X)$

PARAM1 = BASE FORMAT SL6  
PARAM2 = DEPLACEMENT SUR 16 BITS  
(X) = REGISTRE D'INDEX; (PARAM2 + X) EFFECTUEES SUR 16 BITS

BXM	M	P	X	05	30	35	40	65	70								
	LES CONTENUS DE A ET DU MOT ADRESSE SONT ECHANGES EN UN CYCLE MEMOIRE ININTERRUPTIBLE																
	0	0	1	1	1	1	0	0	0	0	0	X	0	0	1	0	1 <sup>er</sup> Mot
	MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3 <sup>em</sup> Mot

PARAMETRES 1 ET 2 SONT DANS DES REGISTRES R1 ET R2 DESIGNES PAR LE MOT 2

- si BIT 11 = 0  $AE_{20} = PARAM1 + 16 + PARAM2$
- si BIT 11 = 1  $AE_{20} = PARAM1 + 16 + (PARAM2 + X)$

PARAM1 = BASE FORMAT SL6  
PARAM2 = DEPLACEMENT SUR 16 BITS  
(X) = REGISTRE D'INDEX; (PARAM2 + X) EFFECTUEE SUR 16 BITS

CDLD	M	P	X	05	30	35	40	65	70								
	LE CONTENU DU MOT ADRESSE EST CHARGE DANS A, LE CONTENU DU MOT SUIVANT DANS B																
	0	0	1	1	1	1	0	0	1	0	0	X	0	1	0	0	1 <sup>er</sup> Mot
	MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- si BIT 11 = 0  $AE_{20} = \phi CDA + D16$
- si BIT 11 = 1  $AE_{20} = \phi CDA + (D16 + (X))$

$\phi CDA$  = ORIGINE CDA FORMAT SL6  
D16 = DEPLACEMENT SUR 16 BITS CONTENU DANS LE MOT MEMOIRE POINTE PAR LE MOT 2  
(X) = REGISTRE D'INDEX; (D16 + X) EFFECTUEES SUR 16 BITS

CDST	M	P	X	05	30	35	40	65	70								
	LE CONTENU DE A EST RANGE DANS LE MOT ADRESSE, LE CONTENU DE B DANS LE MOT SUIVANT																
	0	0	1	1	1	1	0	0	1	0	0	X	0	1	0	1	1 <sup>er</sup> Mot
	MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- si BIT 11 = 0  $AE_{20} = \phi CDA + D16$
- si BIT 11 = 1  $AE_{20} = \phi CDA + (D16 + (X))$

$\phi CDA$  = ORIGINE CDA FORMAT SL6  
D16 = DEPLACEMENT SUR 16 BITS CONTENU DANS LE MOT MEMOIRE POINTE PAR LE MOT 2  
(X) = REGISTRE D'INDEX; (D16 + X) EFFECTUEE SUR 16 BITS

CODES D'ORDRE



N° Document	Date	Page
71 F7 31MS	547	P. 7.27

**CLA** M P ~~E~~ 05 30 35 40 65 70

LE CONTENU DU MOT MEMOIRE ADRESSE EST CHARGE DANS A

0, 0, 1, 1	1, 1, 0, 0	1, 0, 0, X	0, 0, 0, 0	1 <sup>er</sup> Mot
MA	0, 0, 0, 0	±	D	2 <sup>er</sup> Mot

- si bit 11 = 0  $AE20 = \phi CDA - D16$
- si bit 11 = 1  $AE20 = \phi CDA - (D16 + X)$
- $\phi CDA =$  ORIGINE CDA FORMAT SLQ
- D16 = DEPLACEMENT SUR 16 BITS CONTENU DANS LE MOT MEMOIRE POINTE PAR LE MOT 2
- (X) = REGISTRE D'INDEX; (D16 + X) EFFECTUEE SUR 16 BITS

**CLBY** M P ~~E~~ 05 30 35 40 65 70

LES BITS 8 A 15 DE A SONT CHARGES PAR L'OCTET MEMOIRE ADRESSE, LES BITS 0 A 7 DE A SONT MIS A ZERO

0, 0, 1, 1	1, 1, 0, 0	1, 0, 0, 0	0, 0, 1, 1	1 <sup>er</sup> Mot
MA	0, 0, 0, 0	±	D	2 <sup>er</sup> Mot

- $AE20 = \phi CDA + (D16 + (X/2))$
- $\phi CDA =$  ORIGINE CDA FORMAT SLQ
- D16 = DEPLACEMENT SUR 16 BITS CONTENU DANS LE MOT MEMOIRE POINTE PAR LE MOT 2
- (X) = REGISTRE D'INDEX; (D16 + (X/2)) EFFECTUEE SUR 16 BITS
- X PAIR → OCTET GAUCHE; X IMPAIR → OCTET DROIT

**CSTA** M P ~~E~~ 05 30 35 40 65 70

LE CONTENU DE A EST RANGE DANS LE MOT MEMOIRE ADRESSE

0, 0, 1, 1	1, 1, 0, 0	1, 0, 0, X	0, 0, 0, 1	1 <sup>er</sup> Mot
MA	0, 0, 0, 0	±	D	2 <sup>er</sup> Mot

- si bit 11 = 0  $AE20 = \phi CDA + D16$
- si bit 11 = 1  $AE20 = \phi CDA + (D16 + (X))$
- $\phi CDA =$  ORIGINE CDA FORMAT SLQ
- D16 = DEPLACEMENT SUR 16 BITS CONTENU DANS LE MOT MEMOIRE POINTE PAR LE MOT 2
- (X) = REGISTRE D'INDEX; (D16 + (X)) EFFECTUEE SUR 16 BITS

**CSTBY** M P ~~E~~ 05 30 35 40 65 70

LES BITS 8 A 15 DE A SONT RANGEES DANS L'OCTET MEMOIRE ADRESSE

0, 0, 1, 1	1, 1, 0, 0	1, 0, 0, 1	0, 0, 1, 1	1 <sup>er</sup> Mot
MA	0, 0, 0, 0	±	D	2 <sup>er</sup> Mot

- $AE20 = \phi CDA + (D16 + (X/2))$
- $\phi CDA =$  ORIGINE CDA FORMAT SLQ
- D16 = DEPLACEMENT SUR 16 BITS CONTENU DANS LE MOT MEMOIRE POINTE PAR LE MOT 2
- (X) = REGISTRE D'INDEX; (D16 + (X/2)) EFFECTUEE SUR 16 BITS
- X PAIR → OCTET GAUCHE; X IMPAIR → OCTET DROIT

**CXM** M P ~~E~~ 05 30 35 40 65 70

LES CONTENUS DE A ET DU MOT ADRESSE SONT ECHANGES EN UN CYCLE MEMOIRE ININTERROMPU

0, 0, 1, 1	1, 1, 0, 0	1, 0, 0, X	0, 0, 1, 0	1 <sup>er</sup> Mot
MA	0, 0, 0, 0	±	D	2 <sup>er</sup> Mot

- si bit 11 = 0  $AE20 = \phi CDA + D16$
- si bit 11 = 1  $AE20 = \phi CDA + (D16 + (X))$
- $\phi CDA =$  ORIGINE CDA FORMAT SLQ
- D16 = DEPLACEMENT SUR 16 BITS CONTENU DANS LE MOT MEMOIRE POINTE PAR LE MOT 2
- (X) = REGISTRE D'INDEX; (D16 + (X)) EFFECTUEE SUR 16 BITS

**XCTX** M ~~E~~ 05 30 35 40 65 70

COMPUTATION DE CONTEXTE

0, 0, 1, 1	1, 1, 0, 0	1, 1, 0, 0	0, 0, 0, 0
------------	------------	------------	------------

1. DEPILEMENT DE 4 MOTS DE LA PILE K COURANTE: C'EST LE NOUVEAU CONTEXTE
2. ECHANGE DE CES 4 MOTS AVEC SLQ, SLE, K ET LES BITS 1 ET 2 DE ST: C'EST LA COMPUTATION
3. ENPILEMENT DE CES 4 MOTS DANS LA NOUVELLE PILE K: C'EST L'ANCIEN CONTEXTE

**XENT** M P ~~E~~ 05 30 35 40 65 70

ENTREE DE PROCEDURE

0, 0, 1, 1	1, 1, 0, 0	1, 1, 0, 0	0, 0, 0, 1	1 <sup>er</sup> Mot
BASE L DE LA PROCEDURE				2 <sup>er</sup> Mot
LONGUEUR DU SEGMENT				3 <sup>er</sup> Mot

- ELLE UTILISE UNE STRUCTURE DE BASE LIEE DE LA ZONE COMMON POINTEE PAR LA BASE C
- C-127 = LIEN STATIQUE
- C-125 = LIEN DYNAMIQUE
- C-118 = SOUS-PROGRAMME D'ERREUR
- SONT MODIFIES: A, B, L, W, LE LIEN STATIQUE LA PILE ET EVENTUELLEMENT K.

**XSOR** M P ~~E~~ 05 30 35 40 65 70

SORTIE DE PROCEDURE

0, 0, 1, 1	1, 1, 0, 0	1, 1, 0, 0	0, 0, 1, 0
------------	------------	------------	------------

- ELLE UTILISE LA MEME ZONE COMMON QUE L'INSTRUCTION XENT
- SONT MODIFIES: L, W, K, LE LIEN STATIQUE



CODES INSTRUCTIONS PAR ORDRE NUMERIQUE (hexadécimal)

0

00 — JMP  
 01 — JNC JGE  
 02 — JNV JNE  
 03 — JNCV JG  
 04 — NOP  
 05 — JC JL  
 06 — JV JE  
 07 — JCV JLE  
 08 — ADRI A  
 09 — ADRI B  
 0A — ADRI X  
 0B — ADRI Y  
 0C — ADRI C  
 0D — ADRI L  
 0E — ADRI W  
 0F — ADRI K

1

10 — LAI + n  
 11 — LXI + n  
 12 — ORI + n  
 13 — ANDI + n  
 14 — EORI + n  
 15 — CPI + n  
 16 — LYI + n  
 17 — LBI + n  
 18 — JIX  
 19 — JDX  
 1A — PSR  
 1B — PLR  
 1C — SVC  
 1D — LRM  
 1E 00 ACQ  
 1E 01 HALT  
 1E 02 RSR  
 1E 03 DBT  
 1E 04 STEP  
 1E 05 ACTD  
 1E 06 QUIT  
 1E 07 PTY  
 1E 08 SCY  
 1E 09 MOVE  
 1E 0A SBS  
 1E 0B PUSH  
 1E 0C PULL  
 1E 0D ACK  
 1E 0E RSV  
 1E 0F ROMB  
 1E 10 DBP  
 1E 11 SBP  
 1E 12 RBP  
 1E 13 DIT  
 1E 14 EIT  
 1E 15 SST

2

1E 16 RST  
 1E 17 ROHV  
 1E 18 IPI  
 1E 19 RDSI  
 1E 1A LAR  
 1E 1B RDOE  
 1E 1C MVTM  
 1E 1D STAR  
 1E 1E WOE  
 1E 1F MVTS  
 1F — ARM  
 20 — JMP  
 21 — JAGE  
 22 — JANE  
 23 — JAG  
 24 — NOP  
 25 — JAL  
 26 — JAE  
 27 — JALE  
 28 0 — RBT n  
 28 1 — RBT 16 + n  
 28 2 — RBT X, n  
 28 3 — RBT X, 16 + n  
 28 4 — SBT n  
 28 5 — SBT 16 + n  
 28 6 — SBT X, n  
 28 7 — SBT X, 16 + n  
 28 8 — IBT n  
 28 9 — IBT 16 + n  
 28 A — IBT X, n  
 28 B — IBT X, 16 + n  
 28 C — TBT n  
 28 D — TBT 16 + n  
 28 E — TBT X, n  
 28 F — TBT X, 16 + n  
 29 0 — SLRS n  
 29 1 — SLRS 16 + n  
 29 2 — SLRS X, n  
 29 3 — SLRS X, 16 + n  
 29 4 — SLRD n  
 29 5 — SLRD 16 + n  
 29 6 — SLRD X, n  
 29 7 — SLRD X, 16 + n  
 29 8 — SLLS n  
 29 9 — SLLS 16 + n  
 29 A — SLLS X, n  
 29 B — SLLS X, 16 + n  
 29 C — SLLD n  
 29 D — SLLD 16 + n  
 29 E — SLLD X, n  
 29 F — SLLD X, 16 + n  
 2A 0 — SCRS n

2A 1 — SCRS 16 + n  
 2A 2 — SCRS X, n  
 2A 3 — SCRS X, 16 + n  
 2A 4 — SCRD n  
 2A 5 — SCRD 16 + n  
 2A 6 — SCRD X, n  
 2A 7 — SCRD X, 16 + n  
 2A 8 — SCLS n  
 2A 9 — SCLS 16 + n  
 2A A — X, n  
 2A B — X, 16 + n  
 2A C — SCLD n  
 2A D — SCLD 16 + n  
 2A E — SCLD X, n  
 2A F — SCLD X, 16 + n  
 2B 0 — SARS n  
 2B 1 — SARS 16 + n  
 2B 2 — SARS X, n  
 2B 3 — SARS X, 16 + n  
 2B 4 — SARD n  
 2B 5 — SARD 16 + n  
 2B 6 — SARD X, n  
 2B 7 — SARD X, 16 + n  
 2B 8 — XR  
 2B 9 — XR  
 2B A — XR  
 2B B — XR  
 2B C — LR  
 2B D — LR  
 2B E — LR  
 2B F — LR  
 2V 0 — ADR  
 2C 1 — ADR  
 2C 2 — ADR  
 2C 3 — ADR  
 2C 4 — SBR  
 2C 5 — SBR  
 2C 6 — SBR  
 2C 7 — SBR  
 2C 8 — ORR  
 2C 9 — ORR  
 2C A — ORR  
 2C B — ORR  
 2C C — EORR  
 2C D — EORR  
 2C E — EORR  
 2C F — EORR  
 2D 0 — ANDR  
 2D 1 — ANDR  
 2D 2 — ANDR  
 2D 3 — ANDR  
 2D 4 — CLSR  
 2D 5 — CLSR  
 2D 6 — CLSR  
 2D 7 — CLSR

2D 8 — ADCR  
 2D 9 — ADCR  
 2D A — ADCR  
 2D B — ADCR  
 2D C — SBCR  
 2D D — SBCR  
 2D E — SBCR  
 2D F — SBCR  
 2E 0 — NGR  
 2E 1 — NGR  
 2E 2 — NGR  
 2E 3 — NGR  
 2E 4 — CPZR  
 2E 5 — CPZR  
 2E 6 — CPZR  
 2E 7 — CPZR  
 2E 8 — CMR  
 2E 9 — CMR  
 2E A — CMR  
 2E B — CMR  
 2E C — CPR  
 2E D — CPR  
 2E E — CPR  
 2E F — CPR  
 2F 0 — SWBR  
 2F 1 — SWBR  
 2F 2 — SWBR  
 2F 3 — SWBR  
 2F 4 — XIMR  
 2F 5 — XIMR  
 2F 6 — XIMR  
 2F 7 — XIMR  
 2F 8 — ADRP  
 2F 9 — ADRP  
 2F A — ADRP  
 2F B — ADRP  
 2F C — LRP  
 2F D — LRP  
 2F E — LRP  
 2F F — LRP

3

30 — LAI - n  
 31 — LXI - n  
 32 — ORI - n  
 33 — ANDI - n  
 34 — EORI - n  
 35 — CPI - n  
 36 — LYI - n  
 37 — LBI - n  
 38 — extension (option FFP16)  
 39 — extension ( " DAP 16)  
 3A — extension ( " VSS 16)  
 3B — extension  
 3C — extension ( " ISP16)  
 3D — extension ( " FFM16)  
 3E — extension  
 3F — extension (option CDA)

CODES D'ORDRE



SPS 5

N° Document

71 F7 31MS

Date

547

Page

P. 7.29

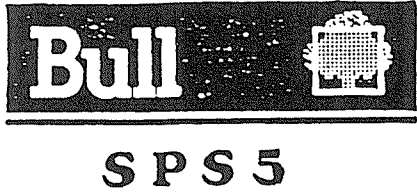
base C		base L		base W		
direct	indirect	direct	indirect	direct	indirect	
<b>4</b>	<b>6</b>	<b>8</b>	<b>A</b>	<b>C</b>	<b>E</b>	
40	60	80	A0	C0	E0	LEBY
41	61	81	A1	C1	E1	STBY
42	62	82	A2	C2	E2	CPBY
43	63	83	A3	C3	E3	XM
44	64	84	A4	C4	E4	CPZ
45	65	85	A5	C5	E5	BR
46	66	86	A6	C6	E6	BSR
47	67	87	A7	C7	E7	SIO
48	68	88	A8	C8	E8	SB
49	69	89	A9	C9	E9	AO
4A	6A	8A	AA	CA	EA	STY
4B	6B	8B	AB	CB	EB	STX
4C	6C	8C	AC	CC	EC	STB
4D	6D	8D	AD	CD	ED	STA
4E	6E	8E	AE	CE	EE	MP
4F	6F	8F	AF	CF	EF	OV
<b>5</b>	<b>7</b>	<b>9</b>	<b>B</b>	<b>D</b>	<b>F</b>	
50	70	90	B0	D0	F0	LA
51	71	91	B1	D1	F1	LX
52	72	92	B2	D2	F2	OR
53	73	93	B3	D3	F3	AND
54	74	94	B4	D4	F4	EOR
55	75	95	B5	D5	F5	CP
56	76	96	B6	D6	F6	LY
57	77	97	B7	D7	F7	LB
58	78	98	B8	D8	F8	IC
59	79	99	B9	D9	F9	OC
5A	7A	9A	BA	DA	FA	RQST
5B	7B	9B	BB	DB	FB	RLSE
5C	7C	9C	BC	DC	FC	ACT
5D	7D	9D	BD	DD	FD	LAD
5E	7E	9E	BE	DE	FE	STZ
5F	7F	9F	BF	DF	FF	WAIT

DECODAGE DE L'OCTET DROIT DES INSTRUCTIONS REGISTRE - REGISTRE

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-4-8-C-	A, A	A, B	A, X	A, Y	A, C	A, L	A, W	A, X	B, A	B, B	B, X	B, Y	B, C	B, L	B, W	B, X
1-5-3-0-	X, A	X, B	X, X	X, Y	X, C	X, L	X, W	X, X	Y, A	Y, B	Y, X	Y, Y	Y, C	Y, L	Y, W	Y, X
2-6-A-E-	C, A	C, B	C, X	C, Y	C, C	C, L	C, W	C, K	L, A	L, B	L, X	L, Y	L, C	L, L	L, W	L, X
3-7-8-F-	W, A	W, B	W, X	W, Y	W, C	W, L	W, W	W, X	X, A	X, B	X, X	X, Y	X, C	X, L	X, W	X, X

ex : 2BC2 se décode LR A, X. On trouve A, X dans la ligne C- et la colonne - 2

CODES D'ORDRE



N° Document	Date	Page
71 F7 31MS	547	P. 7.30