CARTE DE PROGRAMMATION

MODÈLE DE PROGRAMMATION

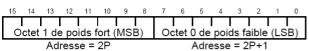
R0	
R1	
R2	
R3	
R4	
R5	
R6	
R7	Registres généraux à N bits (N=16 ou 32)
R8	contiennent donnée ou adresse
R9	
R10	
R11	
R12	
R13	
R14	

SP R15	Stack Pointer (en fait R15)
	pointe sur le dernier mot empilé
PC	Programming Counter
	pointe sur le mot suivant l'instruction
SR	Status Register

indique l'état de la machine

Mot mémoire pour N=16 bits d'adresse 2P

(le CPU est "big endian")



Registre d'état SR :

ı	15	14	13	12	11	10	9	8	7	6	In 5	dica 4	teui 3	rs (F	Flag:	s) 0
	0	0	0	0	0	0	0	0	0	0	In 5 WF	IF	ZF	VF	CF	NF

FLAGS	SIGNIFICATION	USAGE
ZF	Zero Flag	=1⇔ Le résultat de la dernière instruction est nul
CF	Carry Flag	Retenue de la dernière instruction (0 si pas retenue)
NF	Negative Flag	Bit de signe (de gauche, MSB, n°N-1) du résultat de la dernière instruction.
VF	oVerflow Flag	=1⇔ Le résultat de la dernière instruction a débordé de l'intervalle autorisé pour un entier signé en code complément à 2 : [-2 ^{N-1} 2 ^{N-1} -1]
IF	Interrupt Flag	=1⇒ Les interruptions sont validées; 0 ⇒ inhibées
WF	Wait Flag	=1⇔ Le CPU est arrêté et attend une interruption

GROUPE I : OPÉRATIONS À TROIS REGISTRES OPÉRANDES

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1		I OP3			CR	sa			CR	sb			CI	l ₹d	I

Снамр	SIGNIFICATION
OP3	Code d'opération à trois opérandes
CRsa	Code de registre source A (e.g. 1111 pour R15)
CRsb	Code de registre source B (e.g. 1010 pour R10)
CRd	Code de registre destination (e.g. 0000 pour R0)

Actions générales : Rsa $\ensuremath{\mathsf{op3}}$ Rsb \rightarrow Rd , nouveaux indicateurs \rightarrow SR

OP3	Mnémo-	SIGNIFICATION	ACTION PRINCIPALE	I۱	IDI	СА	TE	UR	S
	NIQUE3			W	-	Z	٧	U	Ν
000	ADC	ADd Carry	$Rsa + Rsb + CF \to Rd$			*	*	*	*
001	XOR	EXclusive OR	Rsa			*	0	0	*
010	DIV	DIVision signée & reste	$Rsa \div Rsb o Rd$ $Rsa \% \; Rsb \; o Rsa$			*	*	*	*
011	MUL	MULtiplication signée	Rsa imes Rsb o Rd			*	*	*	*
100	AND	AND	$Rsa \wedge Rsb \to Rd$			*	0	0	*
101	OR	OR	$Rsa \vee Rsb \to Rd$			*	0	0	*
110	ADD	ADD	$Rsa + Rsb \to Rd$			*	*	*	*
111	SUB	SUBstract	$Rsa + \neg Rsb + 1 \to Rd$			*	*	*	*

GROUPE II: OPÉRATIONS À DEUX REGISTRES OPÉRANDES

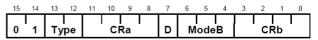
0	1	0	0		0	P2	l		CI	l Rs	l		СІ	₹d	
Сн	АМР		Sic	NIFI	CAT	ON									
OF	2		cod	code d'opération à eux opérandes											
CF	≷s		Со	Code registre source											
CF	₹d		Со	de r	eais	tre o	desti	natio	on						

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Actions générales: $\mathsf{Op2}(\mathsf{Rs}) \to \mathsf{Rd}$, nouveaux indicateurs $\to \mathsf{SR}$

OP2	Mnémo-	SIGNIFICATION	ACTION PRINCIPALE	I	ND	ICA	TE	UR	s
UPZ	NIQUE2		ACTION PRINCIPALE	W	—	Ζ	٧	O	Z
0000	RLC	Rotate Left through Carry	$(Rs \mathrel{<<} 1) + CF \rightarrow Rd$			*	0	*	*
0001	RRC	Rotate Right through Carry	(Rs >> 1) + (CF << (N-1) → Rd			*	0	*	*
0010	SRL	Shift Right Logical	$Rs >> 1 \to Rd$			*	0	*	*
0011	SRA	Shift Right Arithmetic	$Rs / 2 \rightarrow Rd$			*	0	*	*
0100	NOT	Not	$\negRs o Rd$			*	0	0	*
0101	SBB	SuBstract Borrow	Rs + ¬CF + 1111 → Rd			*	*	*	*
0110	SHL	SHift Left	$Rs \ll 1 \rightarrow Rd$			*	0	*	*
0111	NEG	NEGate	$\neg Rs + 1 \to Rd$			*	*	*	*
1000	INP	INPut data	IO[Rs] o Rd			*	0	0	*
1001	OUT	OUTput data	$Rs \rightarrow IO[Rd]$			*	0	0	*
1010	SWB	SWap Bytes	$\begin{array}{l} \text{Rs.LSByte} \\ \rightarrow \text{Rd.MSByte}, \\ \text{Rs.MSByte} \\ \rightarrow \text{Rd.LSByte} \end{array}$			*	0	0	*
1011	RLB*	Rotate Left Barrel	rotation de 4 bits à gauche			*	0	0	*
1100	ANI	ANd Immediate	$Rs \wedge IE \to Rd$			*	0	0	*
1101	EXT*	EXTend sign	$\begin{array}{l} extend(Rs.LSByte) \to \\ Rd \end{array}$			*	0	0	*
1110	ADI	ADd Immediate	$Rs + IE \to Rd$			*	*	*	*
1111	CMP	CoMPare	Rs +			*	*	*	*

Groupe III: Transferts



Снамр	SIGNIFICATION
Туре	Code de taille d'opérande
D	Direction du transfert
ModeB	Code du Mode d'adressage de l'opérande B
CRa	Code du registre opérande A
CRb	Code du registre B; 0000 si direct ou immédiat

Note : L'opérande B est indiqué par le registre Rb et le mode d'adressage ModeB. L'opérande A est le contenu du registre Ra.

	Түре	Mnémo Nique	Nом	TAILLE EN OCTETS	Notes
١	01	В	Byte	1	
١	10	W	Word	2	
١	11	L	Long word	4	Pour N=32 bits

D	Mnémo	SIGNIFICA-	ACTION		ND	ICA	TE	JRS	;
	NIQUE	TION	PRINCIPALE	W	1	Ζ	٧	С	Ν
0	ST	STore	$Ra o Op\'erande \; B$			*	0	0	*
1	LD	LoaD	Ra ← Opérande B			*	0	0	*

ModeB	Nом	EA	OPÉRANDE B
000	Immédiat	PC	E
001	Registre		Opérande B = Rb
010	Indirect	Rb	Opérande B = M[Rb]
011	Indirect- Post-incrémenté	Rb	Opérande B = M[Rb]; Rb ← Rb + taille(type)
100	Indirect- Pré-décrémenté	Rb - taille(type)	Rb ← Rb - taille(type); Opérande B = M[Rb]
101	Direct	ΙE	M[IE]
110	Indexé	IE+Rb	M[Rb+IE]
111	Indirect- pré-indexé	M[IE+Rb]	M[M[Rb+IE]]

GROUPE IV: SAUT RELATIF LONG JCC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0		С	C		1	N	l /lod			C	R	

Снамр	SIGNIFICATION
CC	Code de la Condition de saut
Mode	Code du Mode d'adressage de l'opérande déplacement
CR	Code du Registre de l'opérande déplacement

Jump if Condition : Saute à une instruction si la condition est vérifiée.

Action principale: Condition vérifiée ⇒ PC ← PC + déplacement

CODE CONDITION	Mnémo Condition	SIGNIFICATION	CONDITION SUR LES INDICATEURS
0001	MP	no condition (always JuMP)	1
0010	EQ	EQual	ZF
0011	NE	Not Equal	¬ZF
0100	GE	Greater or Equal	¬(NF ∀ VF)
0101	LE	Lower or Equal	(NF ∀ VF) ∨ ZF
0110	GT	GreaTer	¬(NF ∀ VF) ∧ ¬ZF
0111	LW	LoWer	NF ∀ VF
1000	AE CC	Above or Equal, Carry Cleared	¬CF
1001	BE	Below or Equal	CF ∨ ZF
1010	AB	ABove	¬CF∧ ¬ZF
1011	BL CS	BeLow, Carry Set	CF
1100	vs	oVerflow Set	VF
1101	VC	oVerflow Cleared	¬VF

GROUPE V: INSTRUCTIONS À UN REGISTRE OPÉRANDE

-						 	 7				 		
	0	0	0	0	1	OP1	0	N	/lod	e	С	R	

Снамр	SIGNIFICATION
OP1	Code d'opération à 1 opérande
Mode	Code du Mode d'adressage de l'opérande A
CR	Code du registre R déterminant A selon le Mode

OP1	Mnémo	SIGNIFICATION	Actions
001	JEA	Jump to Effective Address saut inconditionnel absolu long (saute à l'instruction opérande)	PC←EA
010	JSR	Jump to SubRoutine	SP←SP-T; M[SP] ←PC; PC←EA
011	TRP	TRaP Trappe logicielle: Appelle une fonction système dont le n'n est indiqué par l'opérande A	SP←SP-T; M[SP] ←SR; SP←SP-T; M[SP] ←PC; PC←M[4 x A] ∧ 1110
100	TST	TeST	$CF \leftarrow 0$, $VF \leftarrow 0$, $ZF \leftarrow is_zero(A)$, $NF \leftarrow A_{N-1}$
101	TSR	TeSt and Reset	CF←0, VF←0, ZF←is_zero(A), NF←A _{N-1} ; A←0
110	MSR	Move Status Register	A←SR
111	MPC	Move Program Counter	A←PC

NOTATION	SIGNIFICATION				
IE	"Instruction extension": mot qui suit l'instruction				
EA	"Effective Address": adresse de l'opérande				
T N	Taille du mot CPU en octets ; idem en bits				
M[A]	Case mémoire d'adresse A				
IO[A]	Port d'entrée-sortie numéro A				
R2 _n	Le bit n°n de R2 (bit de droite poids faible = R2 $_0$)				
R1←R2 ou R2→R1	Le registre R1 est chargé avec le contenu de R2				
a1,a2;a3	action a1 <i>simultanée</i> à action a2 <i>puis</i> action a3				
X << 2 X >> 2	X décalé de de 2 bits à gauche; idem à droite				
A/B A%B	quotient de A divisé par B ;reste de A sur B				
+ ×	addition <i>fixée sur N bits</i> ; multiplication				
¬ ∧ ∨ ∀	opérateurs bit à bit: NOT, OR, AND, XOR				

GROUPE VI: INSTRUCTIONS SANS REGISTRE OPÉRANDE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0		OP(0	0	0	0	0	0	0	0

СНАМР	SIGNIFICATION
OP0	Code d'opération sans opérande

OP0	Mnémo0	SIGNIFICATION	Actions
000	NOP	No-OPeration	Aucune action
001	HLT	HaLT	WF←1 Arrête et attend une interruption
010	RTS	ReTurn from Subroutine	PC←M[SP]; SP←SP+T
011	RTI	ReTurn from Interrupt	PC←M[SP]; SP←SP+T; SR←M[SP]; SP←SP+T
100	C C	CLear Carry	CF←0
101	STC	SeT Carry	CF←1
110	DSI	DiSable Interrupts	inhibe les interruptions: IF←0
111	ENI	Enable Interrupts	valide les interruptions: IF←1

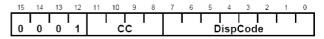
GROUPE VII: OPÉRATIONS RAPIDES



Снамр	SIGNIFICATION
OPQ	Code d'opération rapide
ValueCode	Code complément à 2 de la valeur sur 8 bits (valeur de -128 à 127)
CR	Code du registre destination

OPQ	Mnémo-	SIGNIFICA-	Action	INDICATEURS					
UFQ	NIQUE	TION	ACTION		1	Z	٧	O	Ν
0	LDQ	LoaD Quick	$R \leftarrow ext(ValueCode)$			*	0	0	*
1	ADQ	ADd Quick	$R \leftarrow R + ext(ValueCode)$			*	*	*	*

GROUPE VIII: BRANCHEMENT RELATIF COURT BCC



	SIGNIFICATION			
CC	Code de la Condition de branchement			
DispCode	Code complément à 2 du déplacement sur 8 bits. (déplacement de -128 à 127)			

Branch if Condition: saute à une instruction si la condition est vérifiée

Action principale : Condition vérifiée \Rightarrow PC \leftarrow PC + déplacement

EXCEPTIONS MATÉRIELLES

Nом	SIGNIFICA-	DÉCLEN- CHEMENT	Actions
Reset	initialisation	ligne /RST=0	Efface SR; inhibe les exceptions ; Lance programme adresse démarrage.
Interrupt	interruption sur ligne n1	ligne /IRQi ↓ (périphé- rique)	Termine l'instruction en cours; empile SR; inhibe les exceptions ; appelle le programme de vecteur n°n = i + 32 . Exécutera instruction suivante après
Trap trappe n ^o n		CPU	Empile SR; inhibe les exceptions ; appelle le programme de vecteur n°n <i>Exécutera instruction suivante</i> après
Fault	faute กำ	CPU	Ne finit pas l'instruction en cours; empile SR; inhibe les exceptions; appelle programme de vecteur n ⁿ . Réexécutera instruct. en cours après.
Error	erreur ท _ี ก	CPU	Arrête l'instruction en cours; inhibe les exceptions ; appelle programme de vecteur n ⁿ . Initialisera le CPU après.

- adresse de démarrage = FFFAh pour N=16
- n° vecteur d'exception = INT = n° ligne de requête d'interrup tion + 32
- adresse du vecteur d'exception n°INT = 4 x INT pour N=16 et 32
- adresse du programme d'exception = vecteur \(\Lambda \) 11...110 pour N=16