

Syntaxe	Action	nzp	Codage															
			Op-code				Arguments											
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT DR,SR	DR ← not SR	*	1	0	0	1	DR	SR	1	1	1	1	1	1	1	1		
ADD DR,SR1,SR2	DR ← SR1 + SR2	*	0	0	0	1	DR	SR1	0	0	0	SR2						
ADD DR,SR1,Imm5	DR ← SR1 + SEXT(Imm5)	*	0	0	0	1	DR	SR1	1	Imm5								
AND DR,SR1,SR2	DR ← SR1 and SR2	*	0	1	0	1	DR	SR1	0	0	0	SR2						
AND DR,SR1,Imm5	DR ← SR1 and SEXT(Imm5)	*	0	1	0	1	DR	SR1	1	Imm5								
LEA DR,label	DR ← PC + SEXT(PCOffset9)	*	1	1	1	0	DR	PCOffset9										
LD DR,label	DR ← mem[PC + SEXT(PCOffset9)]	*	0	0	1	0	DR	PCOffset9										
ST SR,label	mem[PC + SEXT(PCOffset9)] ← SR		0	0	1	1	SR	PCOffset9										
LDR DR,BaseR,Offset6	DR ← mem[BaseR + SEXT(Offset6)]	*	0	1	1	0	DR	BaseR	Offset6									
STR SR,BaseR,Offset6	mem[BaseR + SEXT(Offset6)] ← SR		0	1	1	1	SR	BaseR	Offset6									
LDI DR,label	DR ← mem[mem[PC + SEXT(PCOffset9)]]	*	1	0	1	0	DR	PCOffset9										
STI SR,label	mem[mem[PC + SEXT(PCOffset9)]] ← SR		1	0	1	1	SR	PCOffset9										
BR[n][z][p] label	Si (cond) PC ← PC + SEXT(PCOffset9)		0	0	0	0	n	z	p	PCOffset9								
NOP	No Operation		0	0	0	0	0	0	0	000000000								
JMP BaseR	PC ← BaseR		1	1	0	0	0	0	0	BaseR	000000							
RET (≡ JMP R7)	PC ← R7		1	1	0	0	0	0	0	1	1	1	000000					
JSR label	R7 ← PC; PC ← PC + SEXT(PCOffset11)		0	1	0	0	1	PCOffset11										
JSRR BaseR	R7 ← PC; PC ← BaseR		0	1	0	0	0	0	0	BaseR	000000							
RTI	cf. interruptions		1	0	0	0	0000000000000											
TRAP Trapvect8	R7 ← PC; PC ← mem[Trapvect8]		1	1	1	1	0	0	0	0	Trapvect8							
Réservé			1	1	0	1												