

		Mnemonička naredba	Operacija	Zastavice nakon naredbe			
				C	V	N	Z
Vrsta naredaba	Aritmetičko-logičke	ADD <i>src1, src2, dest</i>	src1 + src2 → dest	c	ov	s	z
		ADC <i>src1, src2, dest</i>	src1 + src2 + C → dest	c	ov	s	z
		SUB <i>src1, src2, dest</i>	src1 - src2 → dest	b	ov	s	z
		SBC <i>src1, src2, dest</i>	src1 - src2 - C → dest	b	ov	s	z
		CMP <i>src1, src2</i>	src1 - src2	b	ov	s	z
		AND <i>src1, src2, dest</i>	src1 AND src2 → dest	0	0	s	z
		OR <i>src1, src2, dest</i>	src1 OR src2 → dest	0	0	s	z
		XOR <i>src1, src2, dest</i>	src1 XOR src2 → dest	0	0	s	z
		SHL <i>src1, src2, dest</i>	src1 pomakni logički ulijevo za src2 bitova → dest	izlazni bit	0	s	z
		SHR <i>src1, src2, dest</i>	src1 pomakni logički udesno za src2 bitova → dest	izlazni bit	0	s	z
		ASHR <i>src1, src2, dest</i>	src1 pomakni aritmetički udesno za src2 bitova → dest	izlazni bit	0	s	z
		ROTL <i>src1, src2, dest</i>	src1 rotiraj ulijevo za src2 bitova → dest	izlazni bit	0	s	z
		ROTR <i>src1, src2, dest</i>	src1 rotiraj udesno za src2 bitova → dest	izlazni bit	0	s	z
	Registarske	MOVE <i>src2, dest</i>	src2 → dest				
		MOVE SR, <i>dest</i>	SR → dest (najviši bitovi od dest pune se ničicama)				
		MOVE <i>src2</i> , SR	src2 → SR (najniži bajt od src2 se puni u SR)	ovisi o najnižem bajtu od src2			
	Memorijske	LOAD{B,H} <i>dest, (adr20)</i>	MEM(ext adr20) → dest				
		LOAD{B,H} <i>dest, (adreg+offset20)</i>	MEM(adreg + ext offset20) → dest				
		STORE{B,H} <i>src1, (adr20)</i>	src1 → MEM(ext adr20)				
		STORE{B,H} <i>src1, (adreg+offset20)</i>	src1 → MEM(adreg + ext offset20)				
		PUSH <i>src1</i>	R7 - 4 → R7; src1 → MEM(R7)				
		POP <i>dest</i>	MEM(R7) → dest; R7 + 4 → R7				
	Upravljačke	JP{uvjet} <i>adr20</i>	ext adr20 → PC				
		JP{uvjet} ( <i>adreg</i> )	adreg → PC				
		JR{uvjet} <i>adr</i>	PC + ext offset20 → PC (prevoditelj pretvara adr u offset20)				
		CALL{uvjet} <i>adr20</i>	R7 - 4 → R7; PC → MEM(R7); ext adr20 → PC				
		CALL{uvjet} ( <i>adreg</i> )	R7 - 4 → R7; PC → MEM(R7); adreg → PC				
		RET{uvjet}	MEM(R7) → PC; R7 + 4 → R7				
		RETI{uvjet}	MEM(R7) → PC; R7 + 4 → R7; 1 → GIE				
		RETN{uvjet}	MEM(R7) → PC; R7 + 4 → R7; 1 → IIF				
		HALT{uvjet}	zaustavi procesor				

- src1, src2, dest* i *adreg* označuju opće registre R0 do R7. *Src2* dodatno može biti i broj koji se mora moći dobiti predznačnim proširenjem 20-bitnog broja.
- adr20* označuje adresu koja se mora moći dobiti predznačnim proširenjem 20-bitnog broja.
- adr* označuje adresu čija se udaljenost od naredbe JR mora moći prikazati sa 20 bita.
- offset20* označuje 20-bitni odmak koji mora biti predznačeni broj.
- Oznaka {B,H} označuje da na tom mjestu može pisati slovo B ili H ili ništa. Npr. može se pisati LOADB, LOADH ili LOAD. Ove naredbe učitavaju podatke širine 8, 16 i 32 bita iz memorije u registar. Preostali viši bitovi registra se popunjavaju ničicama.
- Naredbe STOREB, STOREH i STORE spremaju podatke širine 8, 16 i 32 bita iz registra u memoriju. Iz registra se čita samo potreban broj najnižih bitova.
- oznaka *MEM(A)* označuje memorijsku lokaciju čija adresa je A.
- {uvjet} označuje jedan od uvjeta zadanih (prvim stupcem) tablice uvjeta. Uvjet se može ispustiti i tada je uvijek istinit.
- ext* označuje operaciju predznačnog proširivanja do širine 32 bita.
- u stanjima zastavica koriste se sljedeće oznake: c=prijenos, b=posudba, ov=preljev, s=predznak, z=ništica, a prazno mjesto znači da se zastavica ne mijenja.

Tablica uvjeta			
uvjet	značenje	odnos operanada	način ispitivanja zastavica
_C	carry		C = 1
_NC	not carry		C = 0
_V	overflow		V = 1
_NV	not overflow		V = 0
_N	negative		N = 1
_NN	not negative		N = 0
_M	minus		N = 1
_P	plus (positive)		N = 0
_Z	zero		Z = 1
_NZ	not zero		Z = 0
_EQ	equal	X = Y	Z = 1
_NE	not equal	X ≠ Y	Z = 0
_ULE	unsigned less or equal	X ≤ Y	C = 1 ili Z = 1
_UGT	unsigned greater than	X > Y	C = 0 i Z = 0
_ULT	unsigned less than	X < Y	C = 1
_UGE	unsigned greater or equal	X ≥ Y	C = 0
_SLE	signed less or equal	X ≤ Y	(N ⊕ V) = 1 ili Z = 1
_SGT	signed greater than	X > Y	(N ⊕ V) = 0 i Z = 0
_SLT	signed less than	X < Y	(N ⊕ V) = 1
_SGE	signed greater or equal	X ≥ Y	(N ⊕ V) = 0

10	9	8	7	6	5	4	3	2	1	0
INT2	INT1	INT0	GIE	EINT2	EINT1	EINT0	Z	V	C	N

GIE, EINTi: 0 zabranjuje, a 1 dozvoljava maskirajuće prekide  
IIF: 0 zabranjuje, a 1 dozvoljava nemaskirajući prekid

## Sklop FRISC-CT

adresa	pisanje	čitanje
PA	upis u LR (i DC)	trenutačno stanje DC-a
PA + 4	upis u CR	čitanje stanja spremnosti
PA + 8	potvrda prihvata prekida (tj. brisanje status-bistabila)	-
PA + 12	obavijest o kraju posluživanja	-

### Upravljačka riječ (CR)

31-2	1	0
-	STOP/START	INT
	0 - brojilo je zaustavljeno 1 - brojilo broji	0 - ne postavlja prekid 1 - postavlja prekid

## Sklop FRISC-PIO

adresa	pisanje	čitanje
PA	- izlazna upravljačka riječ - ulazna upravljačka riječ - maska (8-bitna, 1 označava bit koji se ispituje)	stanje PIO-sklopa
PA + 4	upis podatka u DR/PIOD	čitanje DR/PIOD
PA + 8	potvrda prihvata prekida (tj. brisanje status-bistabila)	-
PA + 12	obavijest o kraju posluživanja	-

### Izlazni upravljački registar (OCR)

31-3	2	1	0
-	MODE	INT	0
	0 - izlazni način 1 - postavljanje bitova	0 - ne postavlja prekid 1 - postavlja prekid	mora biti 0

### Ulazni upravljački registar (ICR)

31-6	5	4	3	2	1	0
-	AND/OR	ACTIVE	MASK FOLLOWS	MODE	INT	1
	0 - OR 1 - AND	0 - aktivna je 0 1 - aktivna je 1	0 - ne slijedi maska 1 - slijedi maska	0 - ulazni način 1 - ispitivanje bitova	0 - ne postavlja prekid 1 - postavlja prekid	mora biti 1

## Sklop FRISC-DMA

adresa	pisanje	čitanje
PA	upis adrese izvora	čitanje adrese izvora
PA + 4	upis adrese odredišta	čitanje adrese odredišta
PA + 8	upis u brojač podataka	čitanje brojača podataka
PA + 12	upis upravljačke riječi	čitanje bistabila stanja
PA + 16	pokretanje prijenosa	-
PA + 20	potvrda prihvata prekida (tj. brisanje status-bistabila)	-

### Upravljačka riječ

31-4	3	2	1	0
-	DESTINATION	SOURCE	MODE	INT
	0 - memorija 1 - vanjska jedinica	0 - memorija 1 - vanjska jedinica	0 - zaustavljanje procesora 1 - krađa ciklusa	0 - ne postavlja prekid 1 - postavlja prekid