UoS educational processor - instruction set summary

Registers and flags

RA: 00 RB: 01 RC: 10 RD: 11

Flags are set after "cmp dst, src" and used for conditional jumps:

ZF: Zero flag
SF: Sign flag
CF: Carry flag

OF: Overflow flag (not

implemented)

Legend

rn or rm indicate a register (one of RA, RB, RC, RD) as source or destination. It is encoded by two bits in the instruction encoding.

i indicates an immediate,i.e. a constant specified in the instruction encoding.

General instruction format

instr(1513)	<pre>instruction(128)</pre>	instruction(70)
Opcode	depends on the	src
	instruction	

Move instruction (opcode 000)

Format:

mov dst, src

Instructions			i	nstruct	Instruction(7											
Move	Or	CO	de	\overline{R} / I	dd#m	sd#m	dr	eg		Sl	CC	(i	or	rn	n)	
mov rn, rm	0	0	0	0	0	0	r	r	_	_	-	-	_	_	r	r
mov rn, i	0	0	0	1	0	0	r	r	i	i	i	i	i	i	i	i
mov rn, [rm]	0	0	0	0	0	1	r	r	_	_	ı	-	_	_	r	r
mov rn, [i]	0	0	0	1	0	1	r	r	i	i	i	i	i	i	i	i
mov [rn], rm	0	0	0	0	1	0	r	r	-	-	-	-	-	-	r	r
mov [rn], i	0	0	0	1	1	0	r	r	i	i	i	i	i	i	i	i

ALU operations (opcode 001, 010, 011)

Format:

Two operands logic/arithmetic

Instructions			i	nstruct	ion(1	58)			I	nst	cru	cti	Lon	.(7	0)
ALU 2 op	or	CO	de	\overline{R}/I	ALU	op	dr	eg		SI	CC	(i	or	rr	n)	
add rn, rm	0	0	1	0	0	0	r	r	_	_	_	_	_	_	r	r
add rn, i	0	0	1	1	0	0	r	r	i	i	i	i	i	i	i	i
sub rn, rm	0	0	1	0	0	1	r	r	-	-	-	-	-	_	٢	r
sub rn, i	0	0	1	1	0	1	r	r	i	i	i	i	i	i	i	i
and rn, rm	0	0	1	0	1	0	r	r	-	-	-	-	-	_	٢	r
and rn , i	0	0	1	1	1	0	r	r	i	i	i	i	i	i	i	i
or rn, rm	0	0	1	0	1	1	r	r	-	-	-	-	-	_	٢	r
or rn, i	0	0	1	1	1	1	r	r	i	i	i	i	i	i	i	i
xor rn, rm	0	1	0	0	0	0	r	r	_	-	-	-	_	-	r	r
xor rn, i	0	1	0	1	0	0	r	r	i	i	i	i	i	i	i	i

Two operands comparison

	Test	opcode	\overline{R}/I	ALU c	p	dr	eg	im	me	dit	e o	or	rm		
ĺ	cmp rn, rm	0 1 0	0	0	1	r	r	-	-	_	-	-	_	r	r

UoS educational processor - instruction set summary

cmp rn	i	Λ	1	Λ	1	0	1	r	r	i	i	i	i	i	i	i	i
CHIP III,	, т	0		U	l -	U	l -										

The comparison is realized by performing a subtraction of dst-src and the result is stored in flags; dst is unchanged:

dst=src: Zero flag set, Carry flag clear
dst>src: Zero flag clear, Carry flag clear
dst<src: Zero flag clear, Carry flag set</pre>

One operand logic/arithmetic

Instructions			i	nstruct	ion(1			Instruction(7									
ALU 1 op	op	COC	de	ALU op	ALU op												
not rn	0	1	1	0	0	0	r	r	_	_	_	-	_	-	-	_	
shr rn	0	1	1	0	0	1	r	r	-	-	-	-	-	-	-	_	
ror rn	0	1	1	0	1	0	r	r	_	_	_	-	_	-	-	_	
asr rn	0	1	1	0	1	1	r	r	-	-	-	-	-	-	-	_	
rol rn	0	1	1	1	0	0	r	r	_	_	_	_	_	_	_	_	

Jumps (opcode 101)

Format:

jxxx src

Unconditional jump: jmp

Conditional jump: je/jz: jump zero/equal, ZF=1

jne/jnz: jump not zero/not equal, ZF=0

ja: jump above (ZF=0, CF=0)
jb: jump below (ZF=0, CF=1)

Instructions	in	st:	ruc	tion(15	58)			Ir	ıst:	ruc	ti	on(7.	.0)		
Jump Unsigned: JA, JAE, JB, JBE	ob	CO	de	\overline{R}/I	•	Jump	ė		im	med	dit	e d	or	rn		
jmp rn	1	0	1	0	0	0	0	0	-	_	_	_	_	_	r	r
jmp i	1	0	1	1	0	0	0	0	i	i	i	i	i	i	i	i
je/jz rn	1	0	1	0	0	0	0	1	-	-	1	-	ı	1	٢	r
je/jz i	1	0	1	1	0	0	0	1	i	i	i	i	i	i	i	i
jne/jnz r <i>n</i>	1	0	1	0	1	0	0	1	_	_	-	_	ı	ı	r	r
jne/jnz i	1	0	1	1	1	0	0	1	i	i	i	i	i	i	i	i
ja r <i>n</i>	1	0	1	0	0	0	1	0	-	_	ı	-	1	ı	۲	r
ja i	1	0	1	1	0	0	1	0	i	i	i	i	i	i	i	i
jb rn	1	0	1	0	1	0	1	1	_	_	_	_	_	_	r	r
jb i	1	0	1	1	1	1	1	i	i	i	i	i	i	i	i	

External interface (opcode 110)

Format:

out src in dst

Instructions			i	nstruct	ion(1	Instruction(70												
IO	or	CO	de	\overline{R} / I	IO type dreg				reg src (i or rn)									
out rn	1	1	0	0	0	0	-	_	_	_	_	_	_	_	r	r		
out i	1	1	0	1	0	0	-	_	i	i	i	i	i	i	i	i		
in rn	1	1	0	_	0	1	r	r	-	-	-	_	_	_	-	1		