BLACKVOXEL Virtual CPU 1.0 Opcodes

Op	Time	CVNZ	Instruction	Description
01	2		nop	No operation. Do Nothing.
41	2		brk	Stop CPU, switch to step mode
81	2		sleep	Break time quantum
02	8	XX	move.b #_8BitsImmediateValue, rx	Move 8 bit immediate value to register
42	10	XX	move.w #16BitsImmediateValue, rx	Move 16 bit immediate value to register
82	14	XX	move.l #32BitsImmediateValue, rx	Move 32 bits immediate value to register
03	8	XX	move.b (ry),rx	Move 8 bits memory data pointed by source register to destination register
43	10	XX	move.w (ry),rx	Move 16 bits memory data pointed by source register to destination register
83	14	XX	move.l (ry),rx	Move 32 bits memory data pointed by source register to destination register
04	8	XX	move.b ry,(rx)	Move 8 bits from source register to memory pointed by destination register
44	10	XX	move.w ry,(rx)	Move 16 bits from source register to memory pointed by destination register
84	14	XX	move.l ry,(rx)	Move 32 bits from source register to memory pointed by destination register
05	6	XX	move.b ry,rx	Move 8 bits from source register to destination register
45	6	XX	move.w ry,rx	Move 16 bits from source register to destination register
85	6	XX	move.l ry,rx	Move 32 bits from source register to destination register
06	10	XX	move.b ry,disp(rx)	Move 8 bits from source register to memory pointed by destination register adding displacement
46	12	XX	move.w ry,disp(rx)	Move 16 bits from source register to memory pointed by destination register adding displacement
86	16	XX	move.l ry,disp(rx)	Move 32 bits from source register to memory pointed by destination register adding displacement
07	10	XX	move.b disp(ry),rx	Move 8 bits from memory data pointed by source register adding displacement to destination register
47	12	XX	move.w disp(ry),rx	Move 16 bits from memory data pointed by source register adding displacement to destination register
87	16	XX	move.l disp(ry),rx	Move 32 bits from memory data pointed by source register adding displacement to destination register.
08	12	XX	move.b ry,disp(rx+rz*m)	Move 8 bits from source register to memory location pointed by the sum of source register + dispmacement + offset register * m
48	14	XX	move.w ry,disp(rx+rz*m)	Move 16 bits from source register to memory location pointed by the sum of source register + dispmacement + offset register * m
88	18	XX	move.l ry,disp(rx+rz*m)	Move 32 bits from source register to memory location pointed by the sum of source register + dispmacement + offset register * m
09	12	XX	move.b disp(ry+rz*m),rx	Move 8 bits from memory location pointed by the sum of source register + dispmacement + offset register * m to destination register
49	14	XX	move.w disp(ry+rz*m),rx	Move 16 bits from memory location pointed by the sum of source register + dispmacement + offset register * m to destination register
89	18	XX	move.l disp(ry+rz*m),rx	Move 32 bits from memory location pointed by the sum of source register + dispmacement + offset register * m to destination register
8B	4	XX	movex.un #_4BitsSignedValue,rx	Move 4 bits immediate unsigned value expanded to 32 bits destination register
8A	4	XX	movex.sn #_4BitsSignedValue,rx	Move 4 bits immediate signed value expanded to 32 bits destination register
0B	6	XX	movex.b #8BitsUnsignedValue,rx	Move 8 bits immediate unsigned value expanded to 32 bits destination register
0A	6	XX	movex.sb #_8BitsSignedValue,rx	Move 8 bits immediate signed value expanded to 32 bits destination register
4B	8	XX	movex.w #16BitsUnsignedValue,rx	Move 16 bits immediate unsigned value expanded to 32 bits destination register
4A	8	XX	movex.sw #16BitsSignedValue,rx	Move 16 bits immediate signed value expanded to 32 bits destination register
0C	8+4r		pushregs rx-rx/rx...	Push selected registers (by bitmask) to stack
4C	8+4r		popregs rx-rx/rx...	Pop selected registers (by bitmask) to stack
0D	6	XXX	inc.l #4BitSignedValue,rx	Increment selected register with the immediate 4 bit value
8D	2	XXX	inclr.l	Increment last used indirect register or offset
4D	6	XXX	dec.l #4BitSignedValue,rx	Decrement selected register with the immediate 4 bit value
CD	2	XXX	declr.l	Decrement last used indirect register or offset
0E	6	XXXX	add.b ry,rx	8 bit Addition from the source register to the destination register
4E	6	XXXX	add.w ry,rx	16 bit Addition from the source register to the destination register
8E	6	XXXX	add.l ry,rx	32 bit Addition from the source register to the destination register
0F	6	XXXX	sub.b ry,rx	8 bit substraction from the source register to the destination register
4F	6	XXXX	sub.w ry,rx	16 bit substraction from the source register to the destination register
8F	6	XXXX	sub.l ry,rx	32 bit substraction from the source register to the destination register
10	6	XX	and.b ry,rx	8 bit boolean AND operation from the source register to the destination register
50	6	XX	and.w ry,rx	16 bit boolean AND operation from the source register to the destination register
90	6	XX	and.l ry,rx	32 bit boolean AND operation from the source register to the destination register

11	6	XX	or.b ry,rx	8 bit boolean OR operation from the source register to the destination register
51	6	XX	or.w ry,rx	16 bit boolean OR operation from the source register to the destination register
91	6	XX	or.l ry,rx	32 bit boolean OR operation from the source register to the destination register
12	6	XX	xor.b ry,rx	8 bit boolean XOR operation from the source register to the destination register
52	6	XX	xor.w ry,rx	16 bit boolean XOR operation from the source register to the destination register
92	6	XX	xor.l ry,rx	32 bit boolean XOR operation from the source register to the destination register
13	6	X XX	asr.b ry,rx	8 bit arithmetic shift right of the destination register with shift count from the source register
53	6	X XX	asr.w ry,rx	16 bit arithmetic shift right of the destination register with shift count from the source register
93	6	X XX	asr.l ry,rx	32 bit arithmetic shift right of the destination register with shift count from the source register
14	6	X XX	lsl.b ry,rx	8 bit logical shift left of the destination register with shift count from the source register
54	6	X XX	lsl.w ry,rx	16 bit logical shift left of the destination register with shift count from the source register
94	6	X XX	lsl.l ry,rx	32 bit logical shift left of the destination register with shift count from the source register
15	6	X XX	lsr.b ry,rx	8 bit logical shift right of the destination register with shift count from the source register
55	6	X XX	lsr.w ry,rx	16 bit logical shift right of the destination register with shift count from the source register
95	6	X XX	lsr.l ry,rx	32 bit logical shift right of the destination register with shift count from the source register
16	6	X XX	rol.b ry,rx	8 bit left rotation with carry of the destination register with shift count from the source register
56	6	X XX	rol.w ry,rx	16 bit left rotation with carry of the destination register with shift count from the source register
96	6	X XX	rol.l ry,rx	32 bit left rotation with carry of the destination register with shift count from the source register
17	6	X XX	ror.b ry,rx	8 bit right rotation with carry of the destination register with shift count from the source register
57	6	X XX	ror.w ry,rx	16 bit right rotation with carry of the destination register with shift count from the source register
97	6	X XX	ror.l ry,rx	32 bit right rotation with carry of the destination register with shift count from the source register
18	150	XX	umul.b ry,rx	8 bit unsigned multiplication of the destination register with the source register
58	150	XX	umul.w ry,rx	16 bit unsigned multiplication of the destination register with the source register
98	150	XX	umul.l ry,rx	32 bit unsigned multiplication of the destination register with the source register
19	150	XX	smul.b ry,rx	8 bit signed multiplication of the destination register with the source register
59	150	XX	smul.w ry,rx	16 bit signed multiplication of the destination register with the source register
99	150	XX	smul.l ry,rx	32 bit signed multiplication of the destination register with the source register
1A	6	XXXX	cmp.b ry,rx	8 bit compare of source and destination registers
5A	6	XXXX	cmp.w ry,rx	16 bit compare of source and destination registers
9A	6	XXXX	cmp.l ry,rx	32 bit compare of source and destination registers
1B	6		jmp (rx)	Jump to the location designed by the destination register
5B	14		jsr (rx)	Jump to the subroutine location designed by the destination register. Return address is on the stack.
60	8		bra #16BitsSignedValue	Jump to a location designed by relative immediate signed value
1C	8		beq #16BitsSignedValue	Conditionnal jump if values are equal
1D	8		bne #16BitsSignedValue	Conditionnal jump if values are not equal
5E	8		bcc #16BitsSignedValue	Conditionnal jump if carry is clear
5F	8		bcs #16BitsSignedValue	Conditionnal jump if carry is set
5C	8		bvc #16BitsSignedValue	Conditionnal jump if no overflow
5D	8		bvs #16BitsSignedValue	Conditionnal jump if overflow
1F	8		bmi #16BitsSignedValue	Conditionnal jump if negative
1E	8		bpl #16BitsSignedValue	Conditionnal jump if positive
9C	8		bhi #16BitsSignedValue	Conditionnal jump if higher
9D	8		bhs #16BitsSignedValue	Conditionnal jump if higher or same
9E	8		bls #16BitsSignedValue	Conditionnal jump if lower or same
9F	8		blo #16BitsSignedValue	Conditionnal jump if lower
DC	8		bhi.s #16BitsSignedValue	Conditionnal jump if higher(Signed)
DD	8		bhs.s #16BitsSignedValue	Conditionnal jump if higher or same (Signed)
DE	8		bls.s #16BitsSignedValue	Conditionnal jump if lower or same (Signed)
DF	8		blo.s #16BitsSignedValue	Conditionnal jump if lower (Signed)
20	8		bso #16BitsSignedValue	Conditionnal jump on shift overflow
9B	12		rts	Return from subroutine
DB	16	XXXX	rti	Return from interrupt
21	6	XX	ext.bw rx	Expand first 8 bits (signed) of the destination register to 16 bits
61	6	XX	ext.bl rx	Expand first 8 bits (signed) of the destination register to 32 bits
A1	6	XX	ext.wl rx	Expand first 16 bits (signed) of the destination register to 32 bits
40	6		rsr.l rx	Read from special register
80	6	XXXX	wsr.l rx	Write to special register