Bird Computer "Sparrow" BC24

24 Bit RISC Microprocessor Core

Programming Reference Page (Preliminary)

Conditions

Encoding cccc	Mnemonic cc	Condition	Flag Test
0000	eq	equal	z
0001	ne	not equal	~z
0010	mi	minus	n
0011	pl	plus	~n
0100	lt	less than	n ^ v
0101	ge	greater than or equ	~ (n ^ v)
0110	le	less than or equa	(n ^ v) | z
0111	gt	greater than	~((n ^ v) | z
1000	ltu	unsigned less tha	c
1001	geu	unsigned greater or equ	~c
1010	leu	unsigned less or equa	c | z
1011	gtu	unsigned greater tha	~(c | z)
1100	vs	overflow	v
1101	vc	no overflow	~v

Special Reg

TTTTT	Reg	Description
11011	SR	Status

Misc.

ooo	Mne	Description
100	ld	load special registe
101	st	store special registe

Legend

Bit Pattern Code

AAAAA = source register 'Ra'

BBBBB = source register 'Rb'

TTTTT = target register 'Rt

iiiiiiii = immediate consta

d = displacement bi

cccc = condition

x = unused, reserved =

For branch test purposes the C and V flags are set by ALU operations, while the Z flag is based on a direct comparison of the register to zero, and the N flag is the sign bit of the register.

While the syntax of the assembler requires specifying a register for all conditional branches, the register is ignored for simple overflow and carry conditions and any register may be specified.

Jump / Call / Return Grou

Mnemonic	Bit Pattern
call label	0001 aaaa aaaaaaaa aaaaaaaa
call d8[Ra]	0000 1100 xxx AAAAA dddddddd
jmp d8[Ra]	0000 1101 xxx AAAAA dddddddd
ret	0000 1101 xxx 11110 dddddddd
iret	0000 1101 xxx 11111 dddddddd
bcc Ra,label	0000 100 cccc AAAAA dddddddd
icall d8[Ra]	0000 1110 xxx AAAAA dddddddd

ALU Operations

Description	Assembler Syntax	Bit Pattern
Reg - Reg	<op> Rt, Ra, Rb	ooooo 0 TTTTT AAAAA xxx BBBBB
Reg - Imm	<op> Rt, Ra, #im	ooooo 1 TTTTT AAAAA iiiiiiii

Shift

Description	Assembler Syntax	Bit Pattern
Reg - Reg	<op> Rt, Ra, Rb	00101 0 TTTTT AAAAA x oo BBBBB
Reg - Imm	<op> Rt, Ra, #im	00101 1 TTTTT AAAAA x oo iiiii

Miscellaneous Instruction

Mne	Bit Pattern	Description
nop	00000000 00000000 xxxxxxxx	No OPeratio
stop	00000000 00000001 xxxxxxxx	Stop processor

Shift / Rotate

oo	Mne
01	shl / rol
10	ashr
11	shr / ror

Logic

ooo oo	Mne
010 00	and
010 01	or
010 10	xor

Add / Sub

ooo oo	Mne
011 00	add
011 01	addc
011 10	sub
011 11	subc

Load / Store

ooo	Mne
001	lb
010	ldb
011	lw
101	sb
110	sdb
111	sw

Load / Store (ld / st)

Assembler Syntax	Effective Address Generation	Bit Pattern	Description
<op> Rt,d8[Ra]	EA = d8 + [Rn]	11 ooo 1 TTTTT AAAAA dddddddd	Register Indirect with Displaceme

Extended Mnemonics

neg Rt,Rb	sub Rt,R0,Rb	
cmp Ra,Rb	sub R1,Ra,Rb	
cmpz Ra	sub R1,Ra,#0	Compare to zer
not Rt,Ra	xor Rt,Ra,#-1	
test Ra,Rb	and R1,Ra,Rb	
clr Rt	and Rt,R0	
clr <ea>	st R0,<ea>	
clr <spr>	ld <spr>,R0	

Notes:

Bit 7 of the shift count for single bit shifts SHL / SHR converts the shift into a rotate operation for extended shifts.

Const Prefix (CON16 #)

101111 xx iiiiiiii iiiiiiii

Constant prefix can be used to extend: immediate values or load / store / branch

Miscellaneous Instruction

Description	Assembler Syntax	Bit Pattern
Reg - Reg	<op> <spr>, Rb	00100 0 TTTTT xx ooo xxx BBBBB
Reg - Imm	<op> <spr>, #im	00100 1 TTTTT xx ooo iiiiiiii