

#### 4. SYNOPSIS' STD LOGIC ARITH

#### 4.1. PREDEFINED TYPES

<b>UNSIGNED</b> (na to   downto na)	Arrays of STD_LOGIC
<b>SIGNED</b> (na to   downto na)	Integer, 0 or 1
<b>SMALL INT</b>	

## 4.2. OVERLOADED OPERATORS

Left	Op	Right	Return
<b>abs</b>		sg	sg,lv
<b>+</b>		un	un,lv
<b>+</b> , <sup>-</sup>		sg	sg,lv
<b>+</b> , <sup>*</sup> , <sup>/</sup>		un	un,lv
<b>sg</b>		sg	sg,lv
<b>sg</b>		un	sg,lv
<b>+</b> , <sup>*</sup> , <sup>/</sup> , <sub>c</sub>		in	un,lv
<b>+</b> , <sub>c</sub>		in	sg,lv
<b>+</b> , <sub>c</sub>		u/l	un,lv
<b>+</b> , <sub>c</sub>		u/l	sg,lv
<b>&lt;</b> , <b>&lt;</b> , <b>&lt;</b> , <b>&lt;</b> , <b>&lt;</b> , <b>=</b> , <b>=</b> , <b>=</b> , <b>=</b>		un	bool
<b>&lt;</b> , <b>&lt;</b> , <b>&lt;</b> , <b>&lt;</b> , <b>&lt;</b> , <b>=</b> , <b>=</b> , <b>=</b> , <b>=</b>		sg	bool
<b>&lt;</b> , <b>&lt;</b> , <b>&lt;</b> , <b>&lt;</b> , <b>&lt;</b> , <b>=</b> , <b>=</b> , <b>=</b> , <b>=</b> , <sub>c</sub>		in	bool
<b>&lt;</b> , <b>&lt;</b> , <b>&lt;</b> , <b>&lt;</b> , <b>&lt;</b> , <b>=</b> , <b>=</b> , <b>=</b> , <b>=</b> , <sub>c</sub>		in	bool
<b>4.3. PREDEFINED FUNCTIONS</b>			
<b>SHL</b> (un, un)		un	
<b>SHR</b> (un, un)		un	
<b>SHL</b> (sg, un)		sg	
<b>SHR</b> (sg, un)		sg	
<b>EXT</b> (lv, in)		lv	zero-extn
<b>SEXT</b> (lv, in)		lv	sign-extn

### 4.3. PREDEFINED FUNCTIONS

SHL(un, un)	un	zero-extend
SHR(un, un)	un	sign-extend
SHL(sg, un)	sg	
SHR(sg, un)	sg	
EXT(lv, in)	lv	
SEXT(lv, in)	lv	

## 4.4. CONVERSION FUNCTIONS

From	To	Function
un,lv	sg	<b>SIGNED</b> (from)
sg,lv	un	<b>UNSIGNED</b> (to)
sg,un	lv	<b>STD_LOGIC_VECTOR</b> (from)
un,sg	in	<b>CONV_INTEGER</b> (from)
in,un,sg,u	un	<b>CONV_UNSIGNED</b> (from, size)
in,un,sg,u	sg	<b>CONV_SIGNED</b> (from, size)
in,un,sg,u	lv	<b>CONV STD LOGIC VECTOR</b> (from, size)

## 5. SYNOPSIS' STD LOGIC MISC

## 5.1. PREDEFINED FUNCTIONS

AND\_REDUCE( $v \mid uv$ )       $u/v$   
OR\_REDUCE( $v \mid uv$ )       $u/v$   
XOR\_REDUCE( $v \mid uv$ )       $u/v$

## 6. SYNOPSIS' STD LOGIC UNSIGNED

## 6.1. OVERLOADED OPERATORS

Left	Op	Right Return
$\vee$	$+$	$\vee$
$\vee$	$+^*$	$\vee$
$\vee$	$+^C$	$\vee$
$\vee$	$+^{\neg C}$	$\vee$
$\vee$	$<^{\wedge}$	$\vee$
$\vee$	$<^{\wedge} \vee$	$\text{bool}$
$\vee$	$<^{\wedge} \vee =$	$\text{bool}$
$\vee$	$<^{\wedge} \vee = /$	$\text{bool}$
$\vee$	$<^{\wedge} \vee = /_c$	$\text{bool}$

## 6.2. CONVERSION FUNCTIONS

From	To	Function
lv	in	<b>CONV INTEGER</b> (from)

## 7. SYNOPSIS' STD LOGIC SIGNED

## 7.1. OVERLOADED OPERATORS

Left	Op	Right	Return
<b>abs</b>		lv	lv
<b>+</b>		lv	lv
<b>*</b>		lv	lv
<b>-</b>		lv	lv
<b>^C</b>		u/l	lv
<b>^C</b>		lv	lv
<b>&lt;</b>		lv	bool
<b>&lt;=</b>		lv	bool
<b>&lt;=</b>		in	bool
<b>&lt;=</b>		in	bool

## 7.2. CONVERSION FUNCTIONS

From lv	To in	Function CONV INTEGER(from)
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
21	21	21
22	22	22
23	23	23
24	24	24
25	25	25
26	26	26
27	27	27
28	28	28
29	29	29
30	30	30
31	31	31
32	32	32
33	33	33
34	34	34
35	35	35
36	36	36
37	37	37
38	38	38
39	39	39
40	40	40
41	41	41
42	42	42
43	43	43
44	44	44
45	45	45
46	46	46
47	47	47
48	48	48
49	49	49
50	50	50
51	51	51
52	52	52
53	53	53
54	54	54
55	55	55
56	56	56
57	57	57
58	58	58
59	59	59
60	60	60
61	61	61
62	62	62
63	63	63
64	64	64
65	65	65
66	66	66
67	67	67
68	68	68
69	69	69
70	70	70
71	71	71
72	72	72
73	73	73
74	74	74
75	75	75
76	76	76
77	77	77
78	78	78
79	79	79
80	80	80
81	81	81
82	82	82
83	83	83
84	84	84
85	85	85
86	86	86
87	87	87
88	88	88
89	89	89
90	90	90
91	91	91
92	92	92
93	93	93
94	94	94
95	95	95
96	96	96
97	97	97
98	98	98
99	99	99
100	100	100

## 8. SYNOPSIS' STD LOGIC TEXTIO

Read/write binary values

```

READ(line, u/l, [good]);
READ(line, uv, [good]);
READ(line, lv, [good]);
WRITE(line, u/l, [justify], [width]);
WRITE(line, uv, [justify], [width]);
WRITE(line, lv, [justify], [width]);

```

Read/write octal values

```

OREAD(line, uv, [good]);
OREAD(line, lv, [good]);
OWRITE(line, uv, [justify], [width]);
OWRITE(line, lv, [justify], [width]);

```

Read/write hexadecimal values

```

HREAD(line, uv, [good]);
HREAD(line, lv, [good]);
HWRITE(line, uv, [justify], [width]);
HWRITE(line, lv, [justify], [width]);

```

## 9. CADENCE'S STD LOGIC ARITH

## 9.1. OVERLOADED OPERATORS

Left	Op	Right	Return
uv	+	uv	uv
uv	+	lv	lv
u/l	$+_{*3}^*$	u/l	u/l
lv	$+_{*3}^*$	lv	lv
lv	$+_{*3}^*/_C$	u/l	lv
lv	$+_{*3}^*_C$	in	lv
uv	$+_{*3}^*$	uv	uv
uv	$+_{*3}^*$	u/l	uv
uv	$+_{*3}^*_C$	in	uv
lv	$\wedge$	in	bool
uv	$\wedge$	in	bool

## 9.2. PREDEFINED FUNCTIONS

C-like ?: replacements:

<b>COND_OP</b> (bool, lv, lv)	lv
<b>COND_OP</b> (bool, uv, uv)	uv
<b>COND</b> (bool, u/l, u/l)	u/l

Shift operations:

SH\_LEFT(lv, na)  
SH\_LEFT(luv, na)  
SH\_RIGHT(lv, na)  
SH\_RIGHT(luv, na)

Resize functions:

<b>ALIGN_SIZE</b> (lv, na)	lv
<b>ALIGN_SIZE</b> (uv, na)	uv
<b>ALIGN_SIZE</b> (w/l, na)	lv
<b>ALIGN_SIZE</b> (w/l, na)	uv

### 9.3. CONVERSION FUNCTIONS

From lv,uv,u/	To in lv	Function <b>TO_INTEGER</b> (from)
in		<b>TO_STDLOGICVECTOR</b> (from, size)
in	uv	<b>TO_STDLOGICVECTOR</b> (from, size)

© 1995 Qualis Design Corporation. Permission to reproduce and distribute strictly verbatim copies of this document in whole is hereby granted.

**Qualis Design Corporation**  
Beaverton, OR USA  
+1-503-531-0377 FAX: +1-503-531-0378  
E-mail: [info@qualis.com](mailto:info@qualis.com)

**Also available:** VHDL Quick Reference Card  
Verilog HDL Quick Reference Card