1164 PACKAGES QUICK REFERENCE CARD

Revision 2.2

()	Grouping	[ ]	Optional
{}	Repeated	|	Alternative
bold	As is	CAPS	User Identifier
italic	VHDL-93	c	commutative

b	::= BIT
bv	::= BIT_VECTOR
u/l	::= STD_ULOGIC/STD_LOGIC
uv	::= STD_ULOGIC_VECTOR
lv	::= STD_LOGIC_VECTOR
un	::= UNSIGNED
sg	::= SIGNED
in	::= INTEGER
na	::= NATURAL
sm	::= SMALL_INT (subtype INTEGER range 0 to 1)

1. IEEE's STD_LOGIC_1164

1.1 Logic Values

'U'	Uninitialized
'X'/'W'	Strong/Weak unknown
'0'/'L'	Strong/Weak 0
'1'/'H'	Strong/Weak 1
'Z'	High Impedance
'-'	Don't care

1.2 Predefined Types

STD_ULOGIC	Base type
Subtypes:	
STD_LOGIC	Resolved STD_ULOGIC
X01	Resolved X, 0 & 1
X01Z	Resolved X, 0, 1 & Z
UX01	Resolved U, X, 0 & 1
UX01Z	Resolved U, X, 0, 1 & Z
STD_ULOGIC_VECTOR(na **to | downto** na**)**	Array of STD_ULOGIC
STD_LOGIC_VECTOR(na **to | downto** na**)**	Array of STD_LOGIC

1.3 Overloaded Operators

Description	Left	Operator	Right
bitwise-and	u/l,uv,lv	**and**, **nand**	u/l,uv,lv
bitwise-or	u/l,uv,lv	**or**, **nor**	u/l,uv,lv
bitwise-xor	u/l,uv,lv	**xor**, **xnor**	u/l,uv,lv
bitwise-not		**not**	u/l,uv,lv

1.4 Conversion Functions

From	To	Function
u/l	b	**TO_BIT(**from[**,** xmap]**)**
uv,lv	bv	**TO_BITVECTOR(**from[**,** xmap]**)**
b	u/l	**TO_STDULOGIC(**from**)**
bv,uv	lv	**TO_STDLOGICVECTOR(**from**)**
bv,lv	uv	**TO_STDULOGICVECTOR(**from**)**

2. IEEE's NUMERIC_STD

2.1 Predefined Types

UNSIGNED(na **to | downto** na**)**	Array of STD_LOGIC
SIGNED(na **to | downto** na**)**	Array of STD_LOGIC

2.2 Overloaded Operators

Left	Op	Right	Return
	abs	sg	sg
	-	sg	sg
un	**+,-,*,/,rem,mod**	un	un
sg	**+,-,*,/,rem,mod**	sg	sg
un	**+,-,*,/,rem,mod** $_c$	na	un
sg	**+,-,*,/,rem,mod** $_c$	in	sg
un	**<,>,<=,>=,=,/=**	un	bool
sg	**<,>,<=,>=,=,/=**	sg	bool
un	**<,>,<=,>=,=,/=** $_c$	na	bool
sg	**<,>,<=,>=,=,/=** $_c$	in	bool

2.3 Predefined Functions

SHIFT_LEFT(un, na**)**	un
SHIFT_RIGHT(un, na**)**	un
SHIFT_LEFT(sg, na**)**	sg
SHIFT_RIGHT(sg, na**)**	sg
ROTATE_LEFT(un, na**)**	un
ROTATE_RIGHT(un, na**)**	un
ROTATE_LEFT(sg, na**)**	sg
ROTATE_RIGHT(sg, na**)**	sg
RESIZE(sg, na**)**	sg
RESIZE(un, na**)**	un
STD_MATCH(u/l, u/l**)**	bool
STD_MATCH(uv, uv**)**	bool
STD_MATCH(lv, lv**)**	bool
STD_MATCH(un, un**)**	bool
STD_MATCH(sg, sg**)**	bool

2.4 Conversion Functions

From	To	Function
un,lv	sg	**SIGNED(**from**)**
sg,lv	un	**UNSIGNED(**from**)**
un,sg	lv	**STD_LOGIC_VECTOR(**from**)**
un,sg	in	**TO_INTEGER(**from**)**
na	un	**TO_UNSIGNED(**from, size**)**
in	sg	**TO_SIGNED(**from, size**)**

3. IEEE's NUMERIC_BIT

3.1 Predefined Types

UNSIGNED(na **to | downto** na**)**	Array of BIT
SIGNED(na **to | downto** na**)**	Array of BIT

3.2 Overloaded Operators

Left	Op	Right	Return
	abs	sg	sg
	-	sg	sg
un	**+,-,*,/,rem,mod**	un	un
sg	**+,-,*,/,rem,mod**	sg	sg
un	**+,-,*,/,rem,mod** $_c$	na	un
sg	**+,-,*,/,rem,mod** $_c$	in	sg
un	**<,>,<=,>=,=,/=**	un	bool
sg	**<,>,<=,>=,=,/=**	sg	bool
un	**<,>,<=,>=,=,/=** $_c$	na	bool
sg	**<,>,<=,>=,=,/=** $_c$	in	bool

3.3 Predefined Functions

SHIFT_LEFT(un, na**)**	un
SHIFT_RIGHT(un, na**)**	un
SHIFT_LEFT(sg, na**)**	sg
SHIFT_RIGHT(sg, na**)**	sg
ROTATE_LEFT(un, na**)**	un
ROTATE_RIGHT(un, na**)**	un
ROTATE_LEFT(sg, na**)**	sg
ROTATE_RIGHT(sg, na**)**	sg
RESIZE(sg, na**)**	sg
RESIZE(un, na**)**	un

3.4 Conversion Functions

From	To	Function
un,bv	sg	**SIGNED(**from**)**
sg,bv	un	**UNSIGNED(**from**)**
un,sg	bv	**BIT_VECTOR(**from**)**
un,sg	in	**TO_INTEGER(**from**)**
na	un	**TO_UNSIGNED(**from**)**
in	sg	**TO_SIGNED(**from**)**