

Type	Symbol	Description	Example (a = 4 bit, b = 5 bit)
Arithmetic	+	integer, natural, positive, signed and unsigned	c (exactly 5 bit) = a + b (if a and b are signed/unsigned)
	-		c (exactly 5 bit) = a - b (if a and b are signed/unsigned)
	*		c (exactly 9 bit i.e. 4+5) = a * b (if a and b are signed/unsigned)
	/		c (exactly 5 bit)=a / b (if a and b are signed/unsigned & ignores the decimal values e.g. 5/-2 = -2)
	abs	integer, natural, positive	abs(a) (absolute value of a)
	rem		a rem b (returns remainder with sign of a)
	mod		a mod b (returns remainder with sign of b)
	**	natural, positive, constant positive integer	a ** b (e.g. 2**3 = 8)
Boolean	not	not	not(a)
	or	or	a or b
	nor	nor	a nor b
	and	and	a and b
	nand	nand	a nand b
	xor	xor	a xor b
	xnor	xnor	a xnor b
Relational	>	greater than	a > b
	<	less than	a < b
	>=	greater than or equal	a >= b
	<=	less than or equal	a <= b
	==	equal	a == b
	/=	not equal	a /= b
Concatenation	&	can be used for shift operations	'0' & "101" = "0101"
			"101" & '0' & "11"= "101011"
Assignment	<=	assign value to singal	a <= "1001"
	:=	assign value to variable	b := "1001"
		assign initial value of signals and variables	signal a : std_logic := '1'
			variable a : std_logic := '1'
	=>	assign value using 'others'	a <= (others => '0')