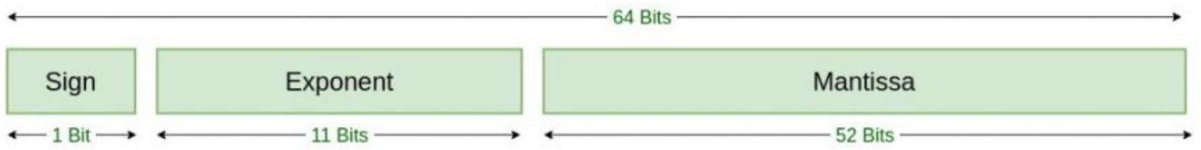
Data Type: Real

Real Data Type

Real data types are utilized in computer programs to represent an estimated value of a real number, as the actual real numbers are not countable and cannot be exactly represented with a finite amount of information in computers.

How Does Computer Store Real Number

- **The most significant bit (MSB)** is used to store the **sign** of the number.
- The next **11 bits** are used to store the **exponent.**
- The remaining **52 bits** are used to store the **mantissa.**



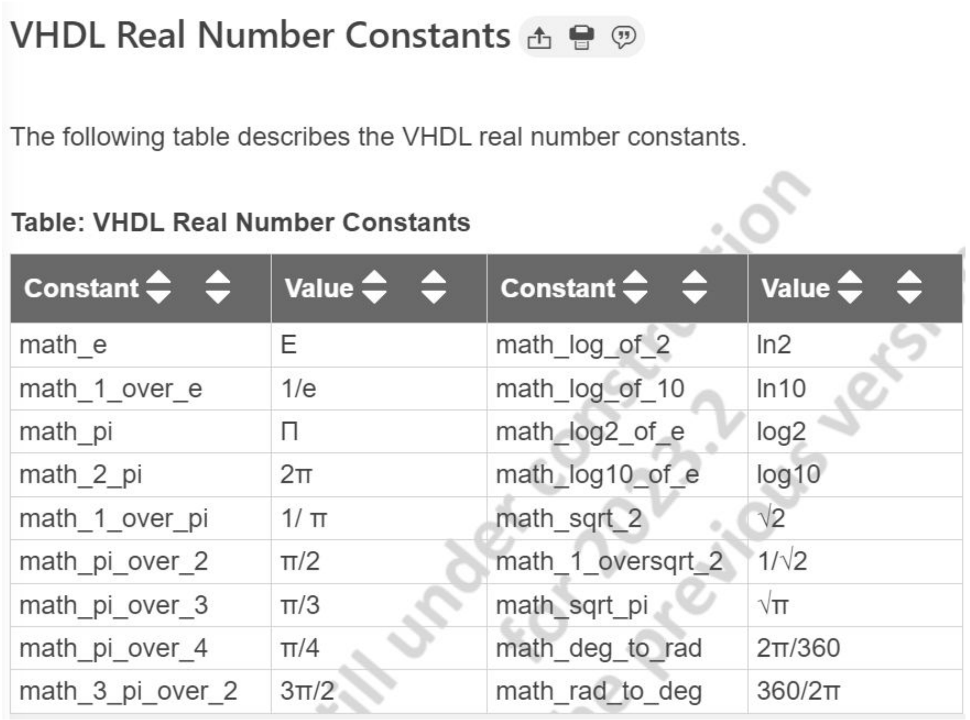
Double Precision
IEEE 754 Floating-Point Standard

https://www.geeksforgeeks.org/ieee-standard-754-floating-point-numbers/

- EX: $5.5_{10} = 101.1_2 = 1.011 \times 2^2$

  => 0 10000000001 011000 …. 00

VHDL Constants



VHDL Real Number Constants

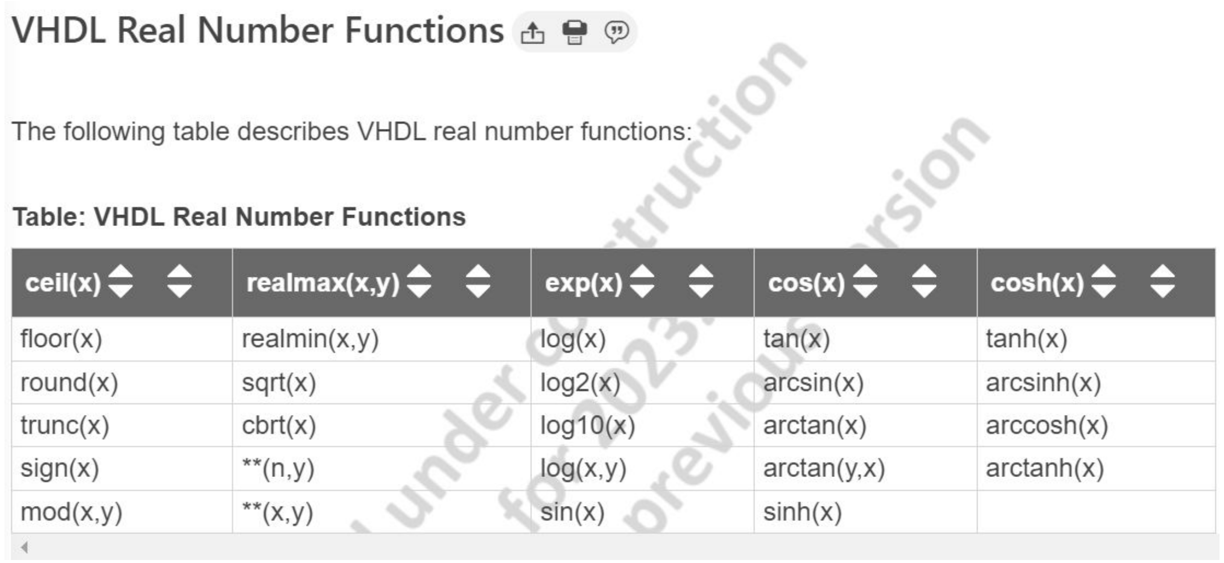
The following table describes the VHDL real number constants.

Table: VHDL Real Number Constants

Constant	Value	Constant	Value
math_e	E	math_log_of_2	ln2
math_1_over_e	1/e	math_log_of_10	ln10
math_pi	Π	math_log2_of_e	log2
math_2_pi	2π	math_log10_of_e	log10
math_1_over_pi	1/ π	math_sqrt_2	√2
math_pi_over_2	π/2	math_1_oversqrt_2	1/√2
math_pi_over_3	π/3	math_sqrt_pi	√π
math_pi_over_4	π/4	math_deg_to_rad	2π/360
math_3_pi_over_2	3π/2	math_rad_to_deg	360/2π

https://docs.xilinx.com/r/en-US/ug901-vivado-synthesis/VHDL-Real-Number-Constants

VHDL Functions



VHDL Real Number Functions

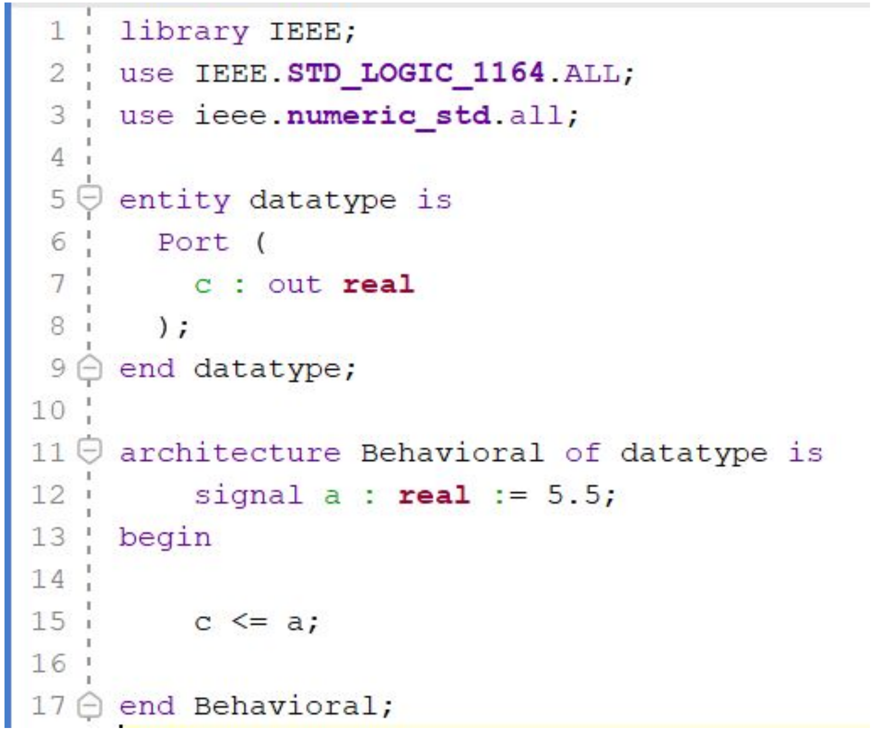
The following table describes VHDL real number functions:

Table: VHDL Real Number Functions

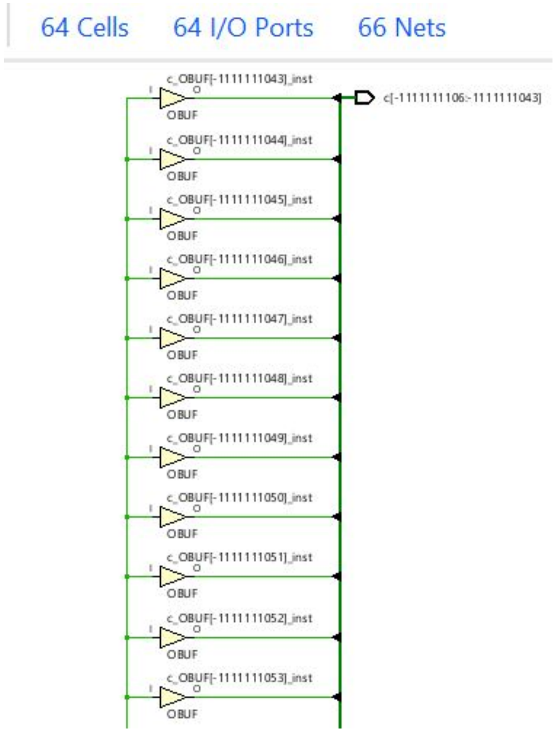
ceil(x)	realmax(x,y)	exp(x)	cos(x)	cosh(x)
floor(x)	realmin(x,y)	log(x)	tan(x)	tanh(x)
round(x)	sqrt(x)	log2(x)	arcsin(x)	arcsinh(x)
trunc(x)	cbrt(x)	log10(x)	arctan(x)	arccosh(x)
sign(x)	**(n,y)	log(x,y)	arctan(y,x)	arctanh(x)
mod(x,y)	**(x,y)	sin(x)	sinh(x)	

https://docs.xilinx.com/r/en-US/ug901-vivado-synthesis/VHDL-Real-Number-Constants

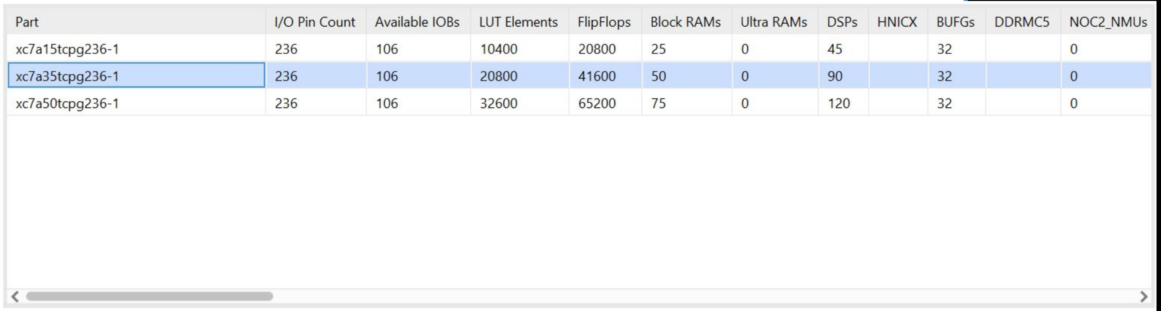
VHDL Example



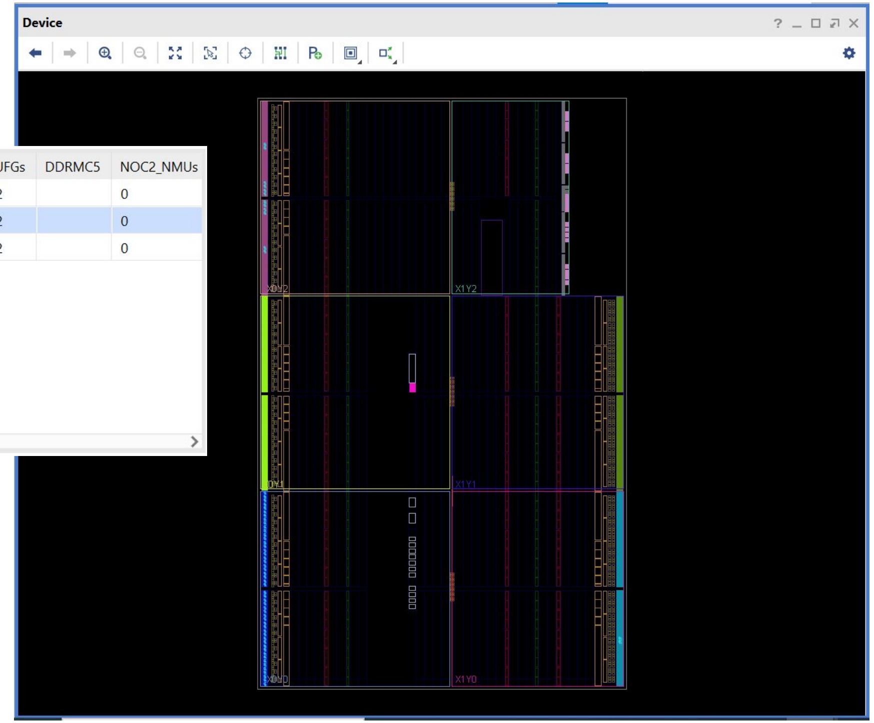
```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity datatype is
  Port (
    c : out real
  );
end datatype;

architecture Behavioral of datatype is
    signal a : real := 5.5;
begin

    c <= a;

end Behavioral;
```



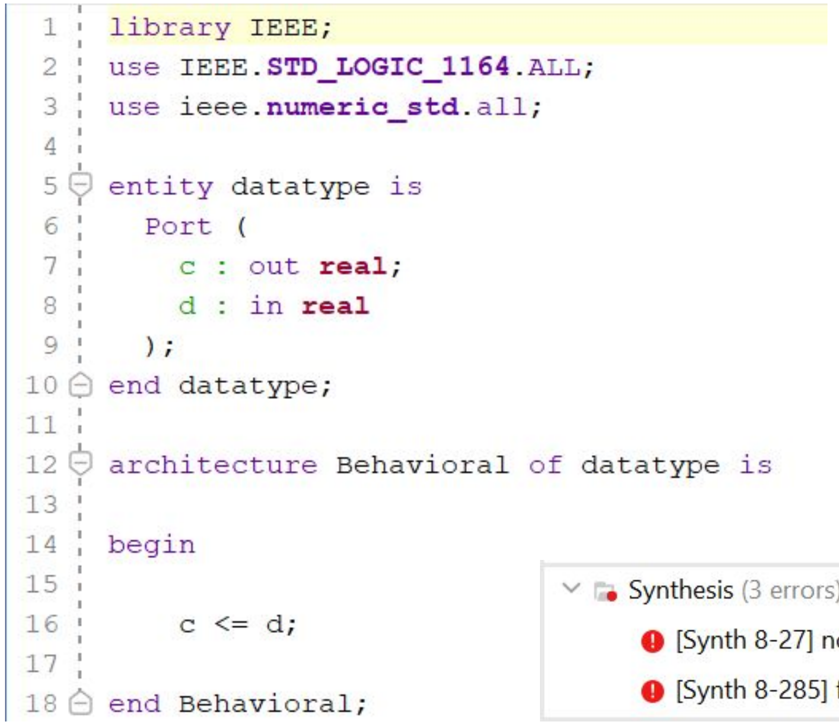
VHDL Example cont.



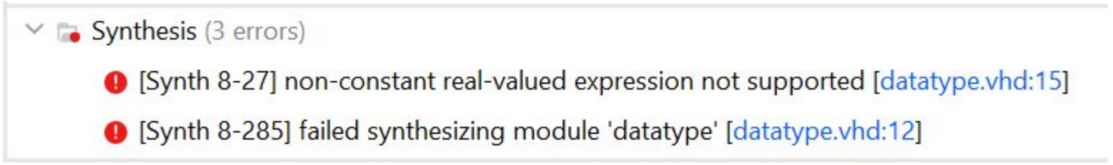
Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs	HNICX	BUFGs	DDRMC5	NOC2_NMUs
xc7a15tcpg236-1	236	106	10400	20800	25	0	45		32		0
xc7a35tcpg236-1	236	106	20800	41600	50	0	90		32		0
xc7a50tcpg236-1	236	106	32600	65200	75	0	120		32		0



VHDL Example cont.



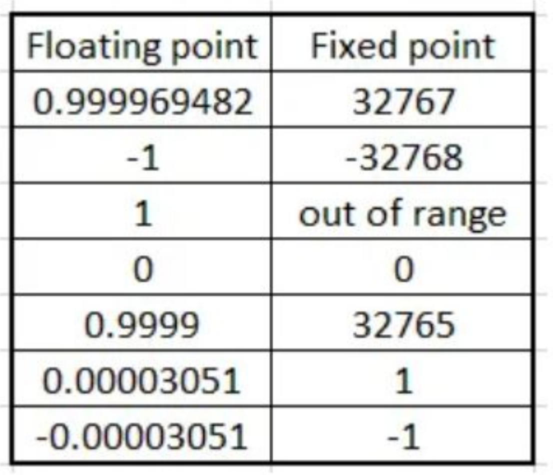
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity datatype is
  Port (
    c : out real;
    d : in real
  );
end datatype;

architecture Behavioral of datatype is

begin

    c <= d;

end Behavioral;
```



```
Synthesis (3 errors)
    [Synth 8-27] non-constant real-valued expression not supported [datatype.vhd:15]
    [Synth 8-285] failed synthesizing module 'datatype' [datatype.vhd:12]
```

Fixed Point Number

In computing, fixed-point is a technique used to represent non-integer numbers by storing a fixed number of digits in their fractional part.



Floating point	Fixed point
0.999969482	32767
-1	-32768
1	out of range
0	0
0.9999	32765
0.00003051	1
-0.00003051	-1

Fixed-point to the floating-point conversion table.

Fixed Point Number

Let F be the floating-point number to convert it to the fixed point number

1. Multiply it by 2^n
   a. where n is the number of bits to the right of the binary point.
2. Round the value to the nearest integer
3. If F is negative take two’s complement of the value arrived at step 2.

Fixed Point Number Example

$0.9999_{10} = 0.11111111111110010111_2$

$\Rightarrow$ fixed point number when n = 15

$\Rightarrow 111111111111100_2 = 32765_{10} = 0.9999 \times 2^{15}$

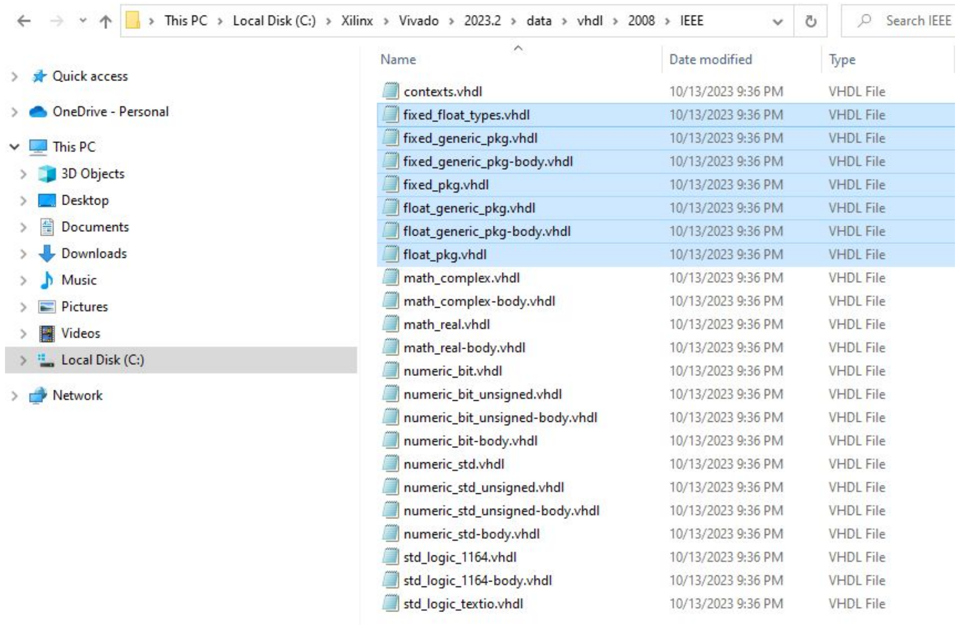
To convert back to decimal

$\Rightarrow 32765_{10} / 2^{15} = 0.99990844726_{10}$

Fixed-Point Package

Has different built in functions / data types

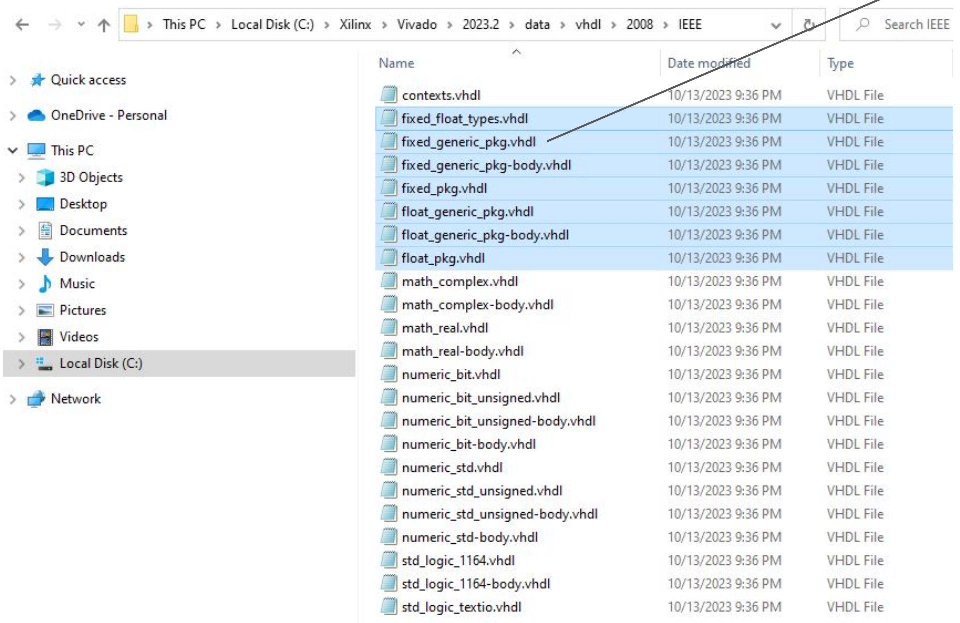
- to_sfixed
- to_real
- sfixed
- real_vector



This PC › Local Disk (C:) › Xilinx › Vivado › 2023.2 › data › vhdl › 2008 › IEEE

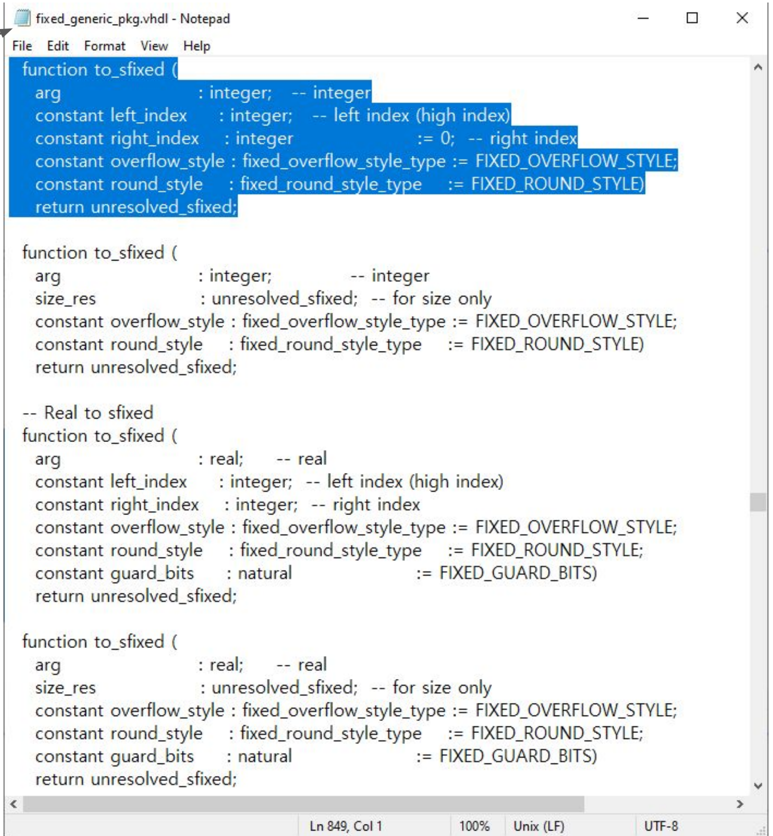
Name	Date modified	Type
contexts.vhdl	10/13/2023 9:36 PM	VHDL File
fixed_float_types.vhdl	10/13/2023 9:36 PM	VHDL File
fixed_generic_pkg.vhdl	10/13/2023 9:36 PM	VHDL File
fixed_generic_pkg-body.vhdl	10/13/2023 9:36 PM	VHDL File
fixed_pkg.vhdl	10/13/2023 9:36 PM	VHDL File
float_generic_pkg.vhdl	10/13/2023 9:36 PM	VHDL File
float_generic_pkg-body.vhdl	10/13/2023 9:36 PM	VHDL File
float_pkg.vhdl	10/13/2023 9:36 PM	VHDL File
math_complex.vhdl	10/13/2023 9:36 PM	VHDL File
math_complex-body.vhdl	10/13/2023 9:36 PM	VHDL File
math_real.vhdl	10/13/2023 9:36 PM	VHDL File
math_real-body.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_bit.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_bit_unsigned.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_bit_unsigned-body.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_bit-body.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_std.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_std_unsigned.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_std_unsigned-body.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_std-body.vhdl	10/13/2023 9:36 PM	VHDL File
std_logic_1164.vhdl	10/13/2023 9:36 PM	VHDL File
std_logic_1164-body.vhdl	10/13/2023 9:36 PM	VHDL File
std_logic_textio.vhdl	10/13/2023 9:36 PM	VHDL File

Fixed-Point Package - to_sfixed



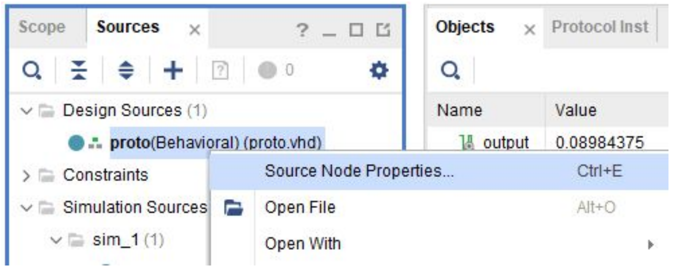
This PC › Local Disk (C:) › Xilinx › Vivado › 2023.2 › data › vhdl › 2008 › IEEE

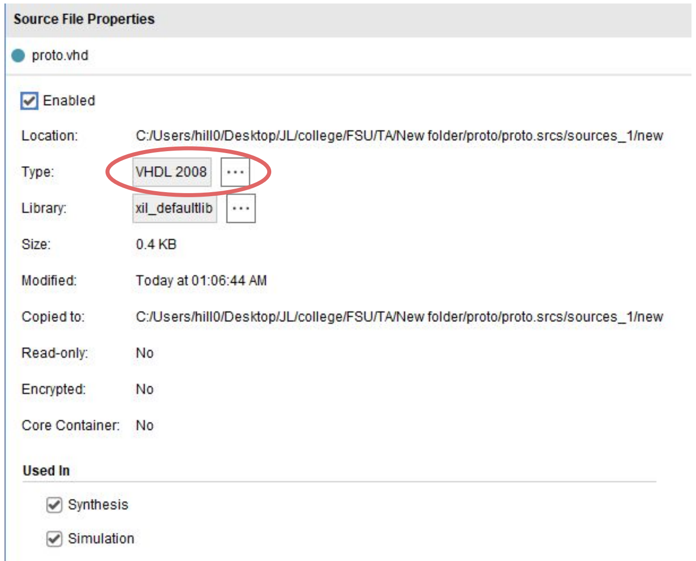
Name	Date modified	Type
contexts.vhdl	10/13/2023 9:36 PM	VHDL File
fixed_float_types.vhdl	10/13/2023 9:36 PM	VHDL File
fixed_generic_pkg.vhdl	10/13/2023 9:36 PM	VHDL File
fixed_generic_pkg-body.vhdl	10/13/2023 9:36 PM	VHDL File
fixed_pkg.vhdl	10/13/2023 9:36 PM	VHDL File
float_generic_pkg.vhdl	10/13/2023 9:36 PM	VHDL File
float_generic_pkg-body.vhdl	10/13/2023 9:36 PM	VHDL File
float_pkg.vhdl	10/13/2023 9:36 PM	VHDL File
math_complex.vhdl	10/13/2023 9:36 PM	VHDL File
math_complex-body.vhdl	10/13/2023 9:36 PM	VHDL File
math_real.vhdl	10/13/2023 9:36 PM	VHDL File
math_real-body.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_bit.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_bit_unsigned.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_bit_unsigned-body.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_bit-body.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_std.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_std_unsigned.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_std_unsigned-body.vhdl	10/13/2023 9:36 PM	VHDL File
numeric_std-body.vhdl	10/13/2023 9:36 PM	VHDL File
std_logic_1164.vhdl	10/13/2023 9:36 PM	VHDL File
std_logic_1164-body.vhdl	10/13/2023 9:36 PM	VHDL File
std_logic_textio.vhdl	10/13/2023 9:36 PM	VHDL File



```
function to_sfixed (
  arg                     : integer;   -- integer
  constant left_index     : integer;   -- left index (high index)
  constant right_index    : integer                    := 0;  -- right index
  constant overflow_style : fixed_overflow_style_type := FIXED_OVERFLOW_STYLE;
  constant round_style    : fixed_round_style_type    := FIXED_ROUND_STYLE)
  return unresolved_sfixed;

function to_sfixed (
  arg                     : integer;              -- integer
  size_res                : unresolved_sfixed;  -- for size only
  constant overflow_style : fixed_overflow_style_type := FIXED_OVERFLOW_STYLE;
  constant round_style    : fixed_round_style_type    := FIXED_ROUND_STYLE)
  return unresolved_sfixed;

-- Real to sfixed
function to_sfixed (
  arg                     : real;     -- real
  constant left_index     : integer;  -- left index (high index)
  constant right_index    : integer;  -- right index
  constant overflow_style : fixed_overflow_style_type := FIXED_OVERFLOW_STYLE;
  constant round_style    : fixed_round_style_type    := FIXED_ROUND_STYLE;
  constant guard_bits     : natural                   := FIXED_GUARD_BITS)
  return unresolved_sfixed;

function to_sfixed (
  arg                     : real;     -- real
  size_res                : unresolved_sfixed;  -- for size only
  constant overflow_style : fixed_overflow_style_type := FIXED_OVERFLOW_STYLE;
  constant round_style    : fixed_round_style_type    := FIXED_ROUND_STYLE;
  constant guard_bits     : natural                   := FIXED_GUARD_BITS)
  return unresolved_sfixed;
```

Fixed-Point Package Usage




Source File Properties

proto.vhd

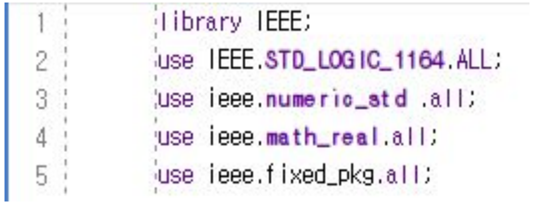
☑ Enabled

Location:	C:/Users/hill0/Desktop/JL/college/FSU/TA/New folder/proto/proto.srcs/sources_1/new
Type:	VHDL 2008
Library:	xil_defaultlib
Size:	0.4 KB
Modified:	Today at 01:06:44 AM
Copied to:	C:/Users/hill0/Desktop/JL/college/FSU/TA/New folder/proto/proto.srcs/sources_1/new
Read-only:	No
Encrypted:	No
Core Container:	No

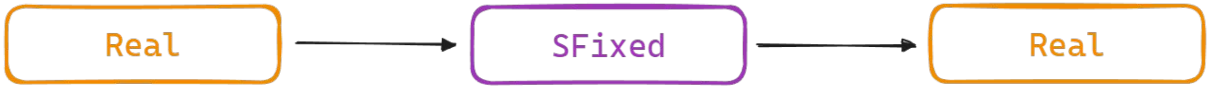
Used In

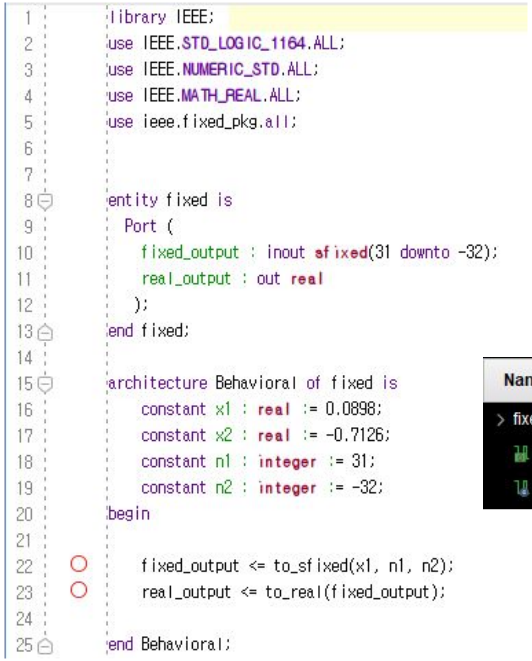
☑ Synthesis

☑ Simulation

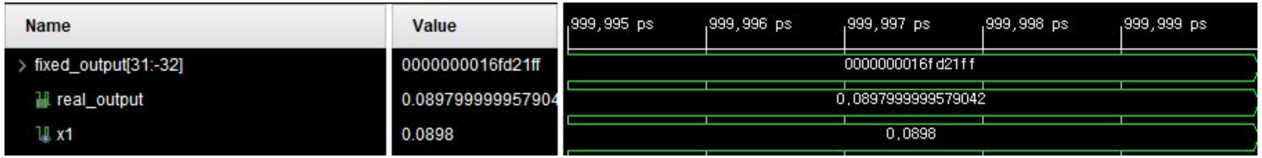

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;
use ieee.math_real.all;
use ieee.fixed_pkg.all;
```

Fixed-Point Package Code Example



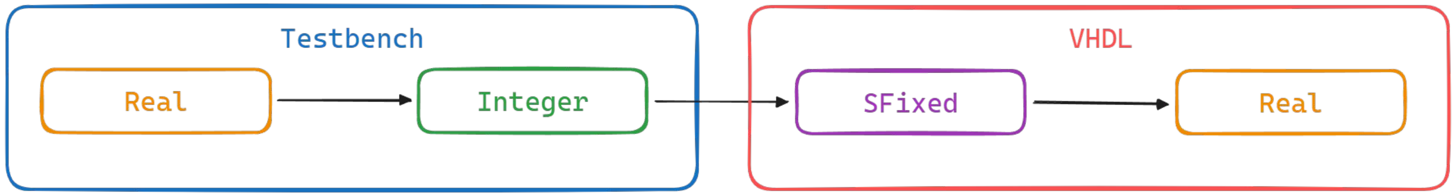


```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
use IEEE.MATH_REAL.ALL;
use ieee.fixed_pkg.all;


entity fixed is
  Port (
    fixed_output : inout sfixed(31 downto -32);
    real_output : out real
   );
end fixed;

architecture Behavioral of fixed is
    constant x1 : real := 0.0898;
    constant x2 : real := -0.7126;
    constant n1 : integer := 31;
    constant n2 : integer := -32;
begin

    fixed_output <= to_sfixed(x1, n1, n2);
    real_output <= to_real(fixed_output);

end Behavioral;
```

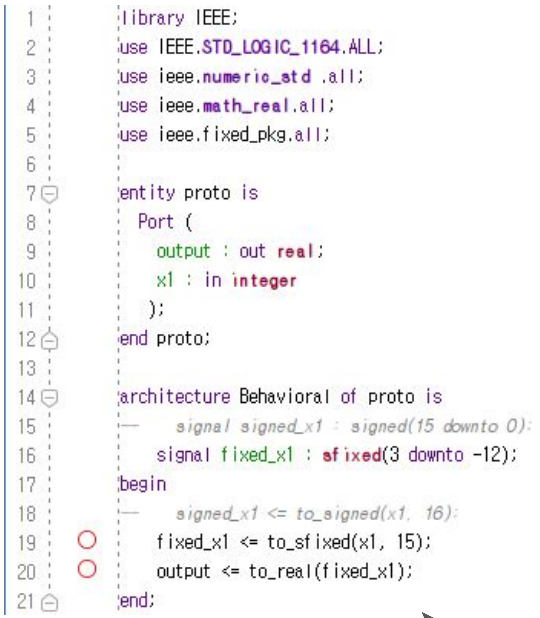


```
Name                  Value               999,995 ps   999,996 ps   999,997 ps   999,998 ps   999,999 ps
> fixed_output[31:-32] 0000000016fd21ff                         0000000016fd21ff
  real_output          0.089799999957904                        0.0897999999579042
  x1                   0.0898                                   0.0898
```

Fixed-Point Package Code Example



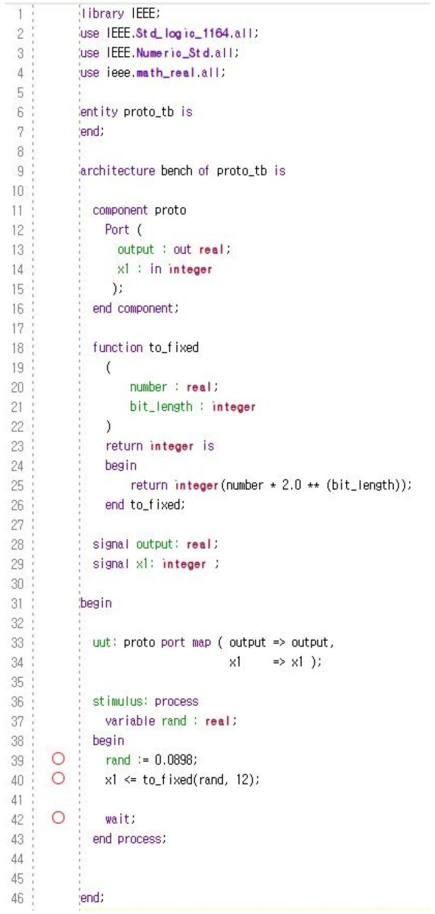
Fixed-Point Package Code Example



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std .all;
use ieee.math_real.all;
use ieee.fixed_pkg.all;

entity proto is
  Port (
    output : out real;
    x1 : in integer
   );
end proto;

architecture Behavioral of proto is
--    signal signed_x1 : signed(15 downto 0);
    signal fixed_x1 : sfixed(3 downto -12);
begin
--    signed_x1 <= to_signed(x1, 16);
    fixed_x1 <= to_sfixed(x1, 15);
    output <= to_real(fixed_x1);
end;
```

VHDL Code

Testbench Code



```
library IEEE;
use IEEE.Std_logic_1164.all;
use IEEE.Numeric_Std.all;
use ieee.math_real.all;

entity proto_tb is
end;

architecture bench of proto_tb is

  component proto
    Port (
      output : out real;
      x1 : in integer
     );
  end component;

  function to_fixed
    (
        number : real;
        bit_length : integer
    )
    return integer is
    begin
        return integer(number * 2.0 ** (bit_length));
    end to_fixed;

  signal output: real;
  signal x1: integer ;

begin

  uut: proto port map ( output => output,
                        x1     => x1 );

  stimulus: process
    variable rand : real;
  begin
    rand := 0.0898;
    x1 <= to_fixed(rand, 12);

    wait;
  end process;


end;
```

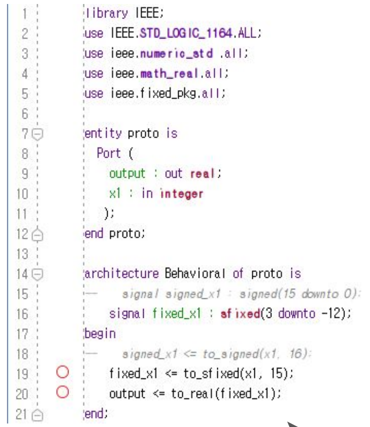
Fixed-Point Package Code Limitation

The package cannot be used within the testbench file

- It will not compile

The output cannot take any non-constant real value

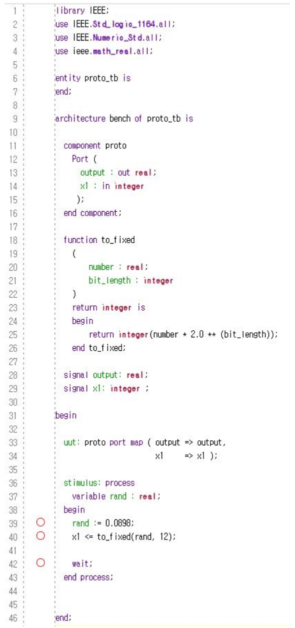
- Have to output in different data type



```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std .all;
use ieee.math_real.all;
use ieee.fixed_pkg.all;

entity proto is
  Port (
    output : out real;
    x1 : in integer
   );
end proto;

architecture Behavioral of proto is
--    signal signed_x1 : signed(15 downto 0);
    signal fixed_x1 : sfixed(3 downto -12);
begin
--    signed_x1 <= to_signed(x1, 16);
    fixed_x1 <= to_sfixed(x1, 15);
    output <= to_real(fixed_x1);
end;
```

VHDL Code

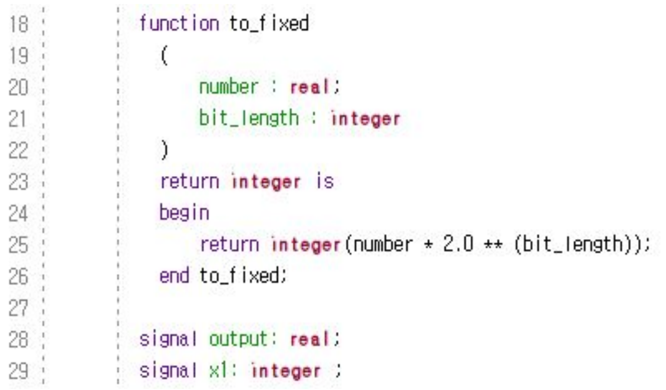
Testbench Code



```vhdl
library IEEE;
use IEEE.Std_logic_1164.all;
use IEEE.Numeric_Std.all;
use ieee.math_real.all;

entity proto_tb is
end;

architecture bench of proto_tb is

  component proto
    Port (
      output : out real;
      x1 : in integer
     );
  end component;

  function to_fixed
    (
        number : real;
        bit_length : integer
    )
    return integer is
    begin
        return integer(number * 2.0 ** (bit_length));
    end to_fixed;

  signal output: real;
  signal x1: integer ;

begin

  uut: proto port map ( output => output,
                        x1     => x1 );

  stimulus: process
    variable rand : real;
  begin
    rand := 0.0898;
    x1 <= to_fixed(rand, 12);

    wait;
  end process;


end;
```

Fixed-Point Alternative

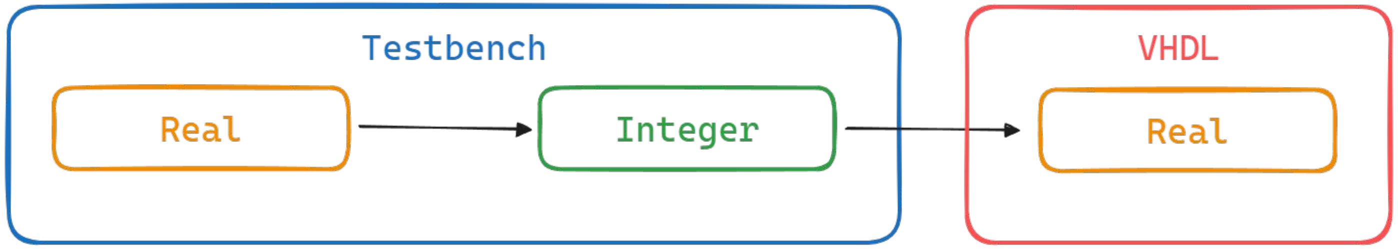
Creating a function that does similar action as the to_sfixed

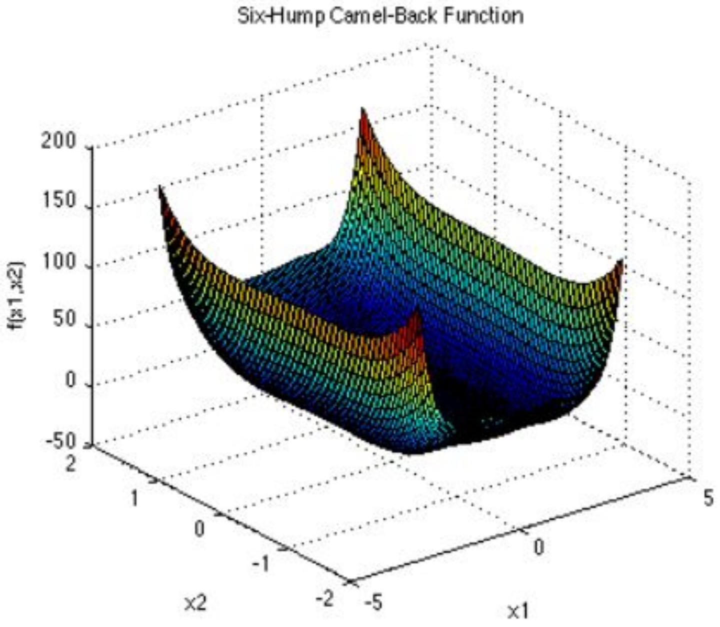


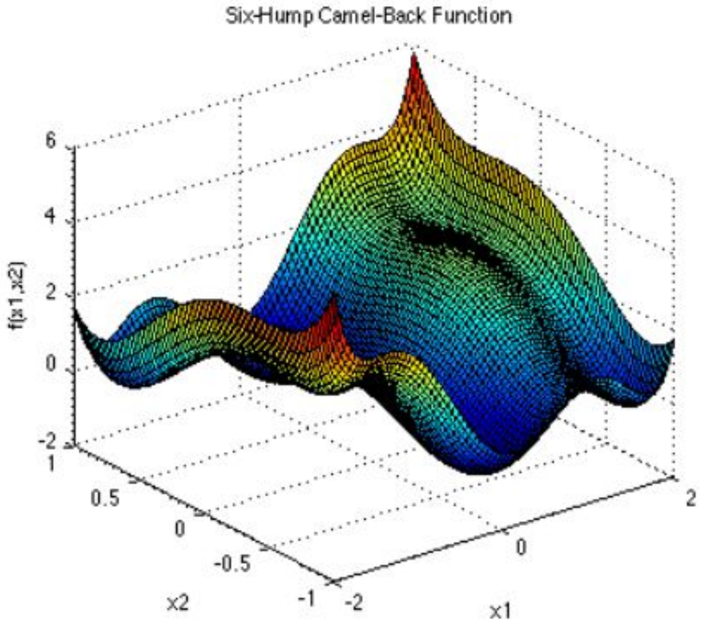
```
function to_fixed
  (
      number : real;
      bit_length : integer
  )
  return integer is
  begin
      return integer(number * 2.0 ** (bit_length));
  end to_fixed;

signal output: real;
signal x1: integer ;
```

Keep the data type as "integer"

Fixed-Point Alternative







$$f(\mathbf{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

Global Minimum:

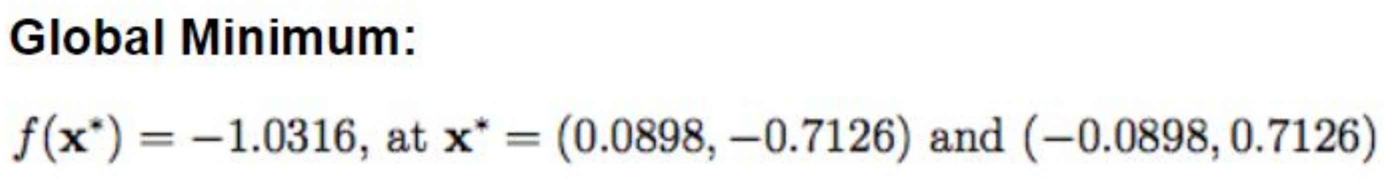
https://www.sfu.ca/~ssurjano/camel6.html

$f(\mathbf{x}^*) = -1.0316$, at $\mathbf{x}^* = (0.0898, -0.7126)$ and $(-0.0898, 0.7126)$

Fixed-point Example

Convert back to decimal

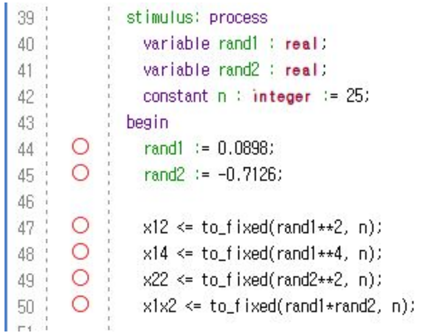
=> $-34610194_{10} / 2^{25} = -1.03146415949_{10}$



Global Minimum:

$f(\mathbf{x}^*) = -1.0316$, at $\mathbf{x}^* = (0.0898, -0.7126)$ and $(-0.0898, 0.7126)$

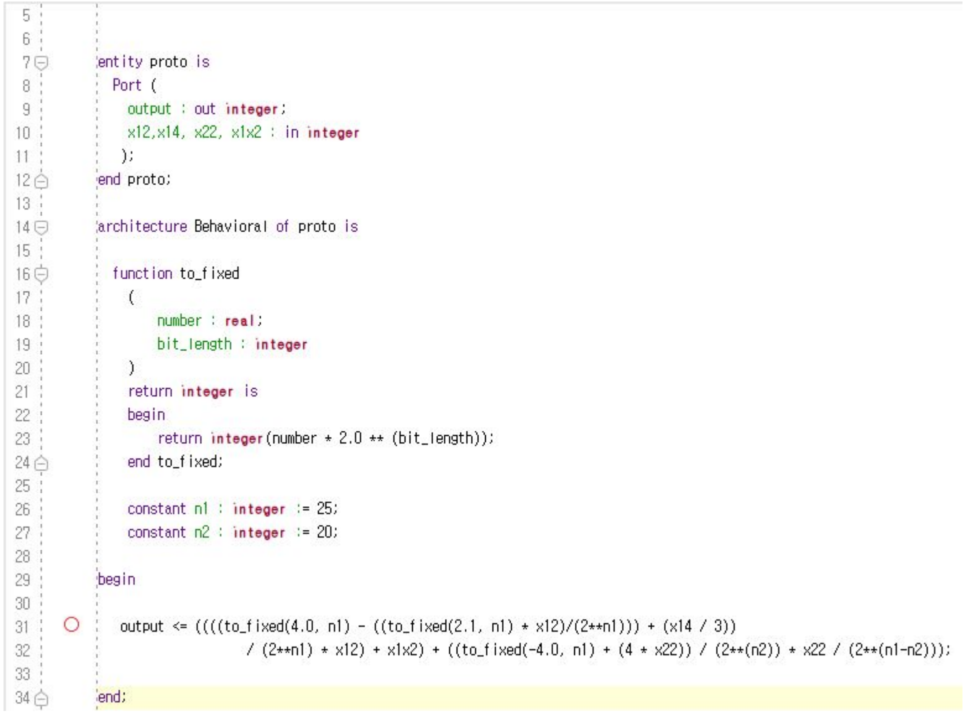
Fixed-point Example



```
stimulus: process
  variable rand1 : real;
  variable rand2 : real;
  constant n : integer := 25;
begin
  rand1 := 0.0898;
  rand2 := -0.7126;

  x12 <= to_fixed(rand1**2, n);
  x14 <= to_fixed(rand1**4, n);
  x22 <= to_fixed(rand2**2, n);
  x1x2 <= to_fixed(rand1*rand2, n);
```

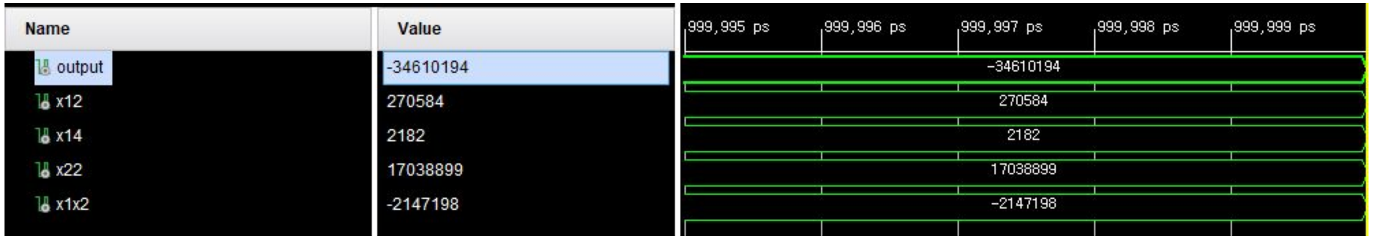
Testbench Code

VHDL Code



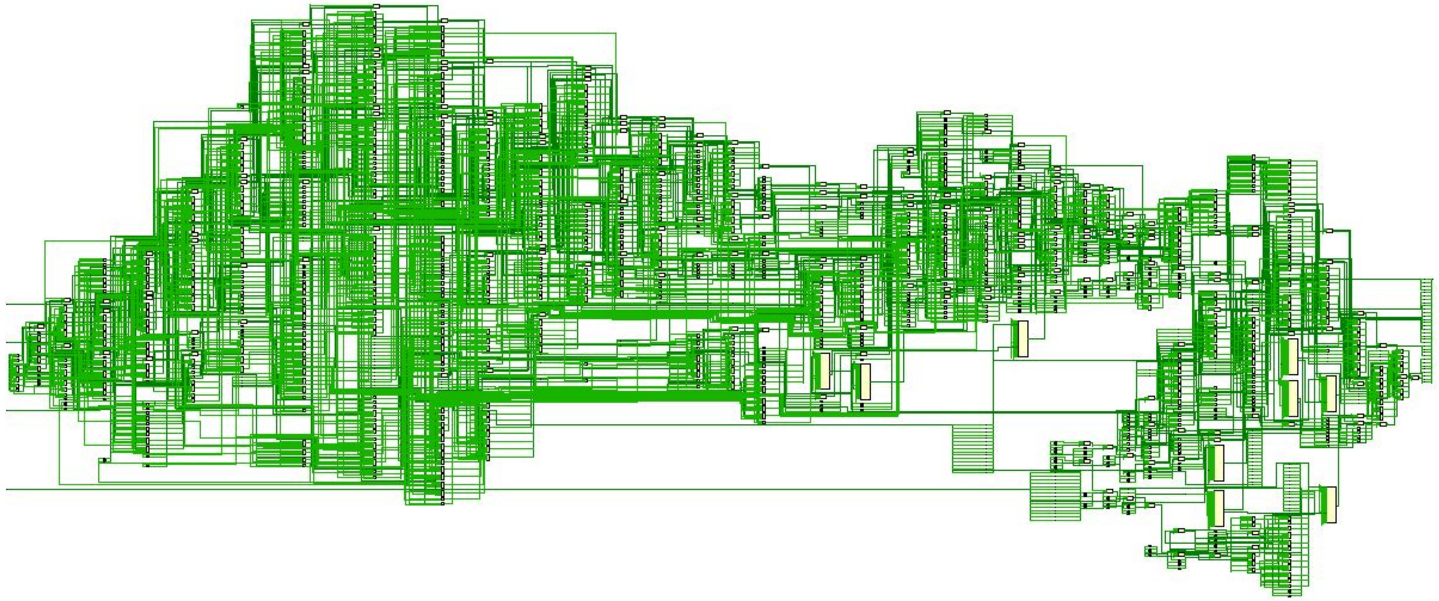
```


entity proto is
  Port (
    output : out integer;
    x12,x14, x22, x1x2 : in integer
    );
end proto;

architecture Behavioral of proto is

  function to_fixed
    (
        number : real;
        bit_length : integer
    )
    return integer is
    begin
        return integer(number * 2.0 ** (bit_length));
    end to_fixed;

    constant n1 : integer := 25;
    constant n2 : integer := 20;

begin

   output <= ((((to_fixed(4.0, n1) - ((to_fixed(2.1, n1) * x12)/(2**n1))) + (x14 / 3))
                        / (2**n1) * x12) + x1x2) + ((to_fixed(-4.0, n1) + (4 * x22)) / (2**(n2)) * x22 / (2**(n1-n2)));

end;
```

Fixed-point Example

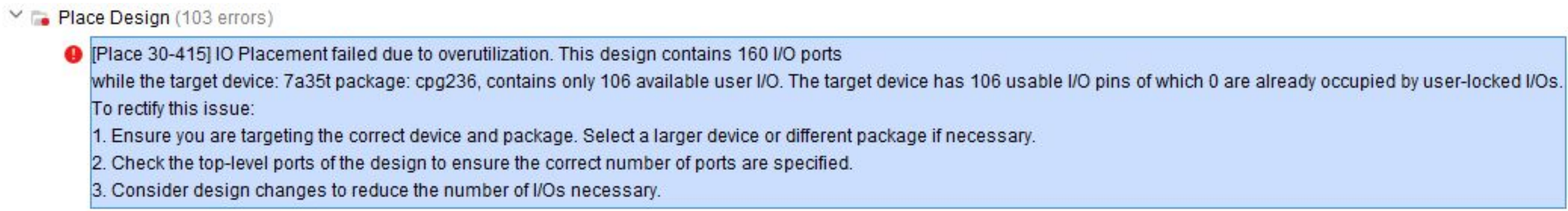


Name	Value
output	-34610194
x12	270584
x14	2182
x22	17038899
x1x2	-2147198

Fixed-point Example



Fixed-point Example



```
Place Design (103 errors)
[Place 30-415] IO Placement failed due to overutilization. This design contains 160 I/O ports
while the target device: 7a35t package: cpg236, contains only 106 available user I/O. The target device has 106 usable I/O pins of which 0 are already occupied by user-locked I/Os.
To rectify this issue:
1. Ensure you are targeting the correct device and package. Select a larger device or different package if necessary.
2. Check the top-level ports of the design to ensure the correct number of ports are specified.
3. Consider design changes to reduce the number of I/Os necessary.
```

Fixed-point Example