
str_lib

vhdl_lib

2021-08-17

目次

1	str_lib	3
1.1	Supported type	3
1.2	STR_LIB_CONFIG	3
1.2.1	STR_LIB_CONFIG.set_append_preamble()	4
1.2.2	STR_LIB_CONFIG.get_append_preamble	4
1.3	print()	4
1.3.1	Example	4
1.4	replace	5
1.4.1	Example	5
1.5	to_str()	5
1.5.1	PRINT_TYPE	6
1.6	“+”(plus)	6
1.6.1	Example	7
1.7	“/”(slash)	8
1.7.1	Example	8

1 str_lib

デバッグ出力やログ出力用の、VHDL 文字列ライブラリです。ログ出力用の `print()` や、各タイプから文字列への変換の `to_str()` があります。(textio.writeline(), integer'image() 等, to_string() のラッパーライブラリ)

function/procedure

`+(plus)`

`/(slash)`

`print()`

`replace()`

`to_str()`

1.1 Supported type

`print()`, `to_str()`, `+`の右辺, `/`の右辺で、使用できるタイプを以下に示します。

[*] `print()` は、`vectortype` は非対応 (Vivado 制限)。

some type	vector type
bit	bit_vector
boolean	boolean_vector
integer	integer_vector
real	real_vector
std_logic	std_logic_vector
	signed
	unsigned

1.2 STR_LIB_CONFIG

str_lib 共通の設定を変更します。

1.2.1 STR_LIB_CONFIG.set_append_preamble()

```
procedure set_append_preamble(b: boolean);
```

append_preamble を設定します。

一部ベクター出力時に “()” を付与し出力するかの制御を行います。

to_str() のオプション引数、append_preamble がついている関数に対し影響します。

STR_LIB_CONFIG.set_append_preamble(True); の場合、以降 procedure 呼び出し時の append_preamble を無視し、“()” を付与しません。

STR_LIB_CONFIG.set_append_preamble(False); の場合、以降 procedure 呼び出し時の append_preamble で “()” の付与します。

1.2.2 STR_LIB_CONFIG.get_append_preamble

```
impure function get_append_preamble return boolean;
```

現在の append_preamble を読み出します。

1.3 print()

```
procedure print(v: character; end_line: boolean:=true);
```

```
procedure print(v: string; end_line: boolean:=true);
```

```
procedure print(v: bit; end_line: boolean:=true);
```

```
procedure print(v: boolean; end_line: boolean:=true);
```

```
procedure print(v: integer; end_line: boolean:=true);
```

```
procedure print(v: real; end_line: boolean:=true);
```

```
procedure print(v: time; end_line: boolean:=true);
```

```
procedure print(v: std_logic; end_line: boolean:=true);
```

textio ライブラリの write(), writeline() のラッパー関数です。

サポートするタイプの v を文字列に変換し、標準出力に出力します。

end_line=true 時にフラッシュします。end_line=false 時はバッファリングし、次の end_line=true でフラッシュを行います。

vector タイプは対象外です。

vector タイプを出力したい場合は、print(to_str(vec_type)); のように to_str() を通す必要があります。

1.3.1 Example

```
1 process
```

```

2     variable v_int: integer:=123;
3     variable v_slv: std_logic_vector(8 downto 0):='1' & X"AB";
4 begin
5     print("Hello world! " + v_int); -- String + Integer
6     print("StdLogiVector=" + v_slv & ", ", false); -- false: not flush
7     print("Integer=" & to_str(v_int, HEX)); -- flush
8
9     print("Finish @" + now); -- show Simulation time
10    finish(0);
11 end process;

```

```

1 Hello world! 123
2 StdLogiVector=0x1AB, Integer=0x0000007B
3 Finish @0fs

```

1.4 replace

`impure function replace(str, search, rep:string) return string;`

文字列 strの中から、文字列 searchを検索し、文字列 repに置換する。

1.4.1 Example

```
1 print(replace("ABcdefg", "AB", "012"));
```

```
1 012cdefg
```

1.5 to_str()

`impure function to_str(v: character) return string;`

`impure function to_str(v: bit) return string;`

`impure function to_str(v: boolean) return string;`

`impure function to_str(v: integer; ptype: PRINT_TYPE:=SIGNED_DEFAULT_TYPE;`

`prefix: string:="0x") return string;`

`impure function to_str(v: real) return string;`

`impure function to_str(v: time) return string;`

`impure function to_str(v: std_logic) return string;`

`impure function to_str(btv: bit_vector; ptype: PRINT_TYPE:=LOGIC_DEFAULT_TYPE
; prefix: string:="0x") return string;`

`impure function to_str(blv: boolean_vector; append_parenthesis: boolean:=
True) return string;`

```

impure function to_str(intv: integer_vector; ptype: PRINT_TYPE:=SIGNED_DEFAULT_TYPE
; prefix: string:="0x"; append_parenthesis: boolean:=True) return string;
impure function to_str(rlv: real_vector; append_parenthesis: boolean:=True)
return string;
impure function to_str(tmv: time_vector; append_parenthesis: boolean:=True)
return string;
impure function to_str(slv: std_logic_vector; ptype: PRINT_TYPE:=LOGIC_DEFAULT_TYPE
; prefix: string:="0x") return string;
impure function to_str(s: signed; ptype: PRINT_TYPE:=SIGNED_DEFAULT_TYPE;
prefix: string:="0x") return string;
impure function to_str(u: unsigned; ptype: PRINT_TYPE:=UNSIGNED_DEFAULT_TYPE
; prefix: string:="0x") return string;

```

サポートするタイプを文字列 `string` に変換します。

`PRINT_TYPE` を付けた場合、指定フォーマットの文字列へ変換します。

`PRINT_TYPE` 指定なしの場合 (Default) は、上記の各関数宣言を参照。

また `prefix`(Default:“0x”) を付けた場合、`PRINT_TYPE=HEX` の場合に `prefix+HEX` へと変換します。

いくつかの Vector タイプでは、`append_parenthesis=True` の場合、“()” をつけてベクターをまとめます。

(例: True 時:(1,2,3), False 時:1,2,3)

1.5.1 PRINT_TYPE

数値タイプ等で、2 進数 (BIN)、10 進数 (符号有:DEC_S, 無:DEC_U)、16 進数 (HEX) での出力を指定します。

```

1 type PRINT_TYPE is (
2   HEX,
3   DEC_S, -- signed
4   DEC_U, -- unsigned
5   BIN);
6 constant LOGIC_DEFAULT_TYPE: PRINT_TYPE := HEX; -- for bit_vector,
   std_logic_vector
7 constant SIGNED_DEFAULT_TYPE: PRINT_TYPE := DEC_S; -- for intger, real
8 constant UNSIGNED_DEFAULT_TYPE: PRINT_TYPE := DEC_U; -- for intger,
   real

```

1.6 “+”(plus)

```

impure function "+"(l: string; r: character) return string;
impure function "+"(l: string; r: bit) return string;

```

```

impure function "+"(l: string; r: boolean) return string;
impure function "+"(l: string; r: integer) return string;
impure function "+"(l: string; r: real) return string;
impure function "+"(l: string; r: time) return string;
impure function "+"(l: string; r: std_logic) return string;
impure function "+"(l: string; r: bit_vector) return string;
impure function "+"(l: string; r: boolean_vector) return string;
impure function "+"(l: string; r: integer_vector) return string;
impure function "+"(l: string; r: real_vector) return string;
impure function "+"(l: string; r: time_vector) return string;
impure function "+"(l: string; r: std_logic_vector) return string;
impure function "+"(l: string; r: signed) return string;
impure function "+"(l: string; r: unsigned) return string;

```

Operator function.

r を文字列へ変換 (to_str()) し、l と結合した l+r を返します。

string + string はできませんので、標準の "&" を使用してください。

1.6.1 Example

```

1 variable v_bl: boolean:= true;
2 variable v_slv: std_logic_vector(7 down to 0):= x"12";

```

code	return string	description
"hello" & "world"	"helloworld"	string & string, defined on standard lib
"value=" + 1234	"value=1234"	string + integer
"value=" + 3.14	"value=3.140000e+00"	string + real
"values=" + integer_vector'(1,2,3)	"value=(1,2,3)"	string + integervector
"value=" + v_bl	"value=true"	string + boolean
"value=" + v_slv	"value=0x12"	string + std_logic_vector
"a=" + 12 & ", b=" + 34	"a=12, b=34"	complex

1.7 “/”(slash)

```

impure function "/"(l: string; r: character) return string;
impure function "/"(l: string; r: string) return string;
impure function "/"(l: string; r: bit) return string;
impure function "/"(l: string; r: boolean) return string;
impure function "/"(l: string; r: integer) return string;
impure function "/"(l: string; r: real) return string;
impure function "/"(l: string; r: time) return string;
impure function "/"(l: string; r: std_logic) return string;
impure function "/"(l: string; r: bit_vector) return string;
impure function "/"(l: string; r: boolean_vector) return string;
impure function "/"(l: string; r: integer_vector) return string;
impure function "/"(l: string; r: real_vector) return string;
impure function "/"(l: string; r: time_vector) return string;
impure function "/"(l: string; r: std_logic_vector) return string;
impure function "/"(l: string; r: signed) return string;
impure function "/"(l: string; r: unsigned) return string;

```

Operator function.

rを文字列へ変換 (to_str()) し、lとカンマ区切りで結合した l,rを返します。

csv 形式。

1.7.1 Example

code	return string	description
to_str(X"123") / 123 / 1.23	"0x123,123,1.230000e+00"	use "/" for csv