
sim_lib

vhdl_lib

2021-08-17

目次

1	sim_lib	3
1.1	check()	3
1.2	make_clock()	3
1.2.1	Example	3
1.3	make_reset()	4
1.3.1	Example	4
1.4	rand()	4
1.4.1	Example	4
1.5	wait_clock()	4
1.5.1	Example	4

1 sim_lib

シミュレーション向けの VHDL ライブラリです。

Function/Procedure

`check()`

`make_clock()`

`make_reset()`

`rand_slv()`

`wait_clock()`

1.1 check()

```
procedure check(data, exp: std_logic_vector; msg:string:="" ; show_result:
boolean:=false);
```

```
procedure check(data, exp: unsigned; msg:string:="" ; show_result:boolean:=
false);
```

```
procedure check(data, exp: signed; msg:string:="" ; show_result:boolean:=
false);
```

データ (data) と期待値 (exp) の比較を行い、不一致であれば、Error を report(アサート) します。show_result が True 時は、一致の場合でもログを出力します。また、msg に任意の文字列が入っている場合は、同時にログに出力します。

1.2 make_clock()

```
procedure make_clock(signal clk: out std_logic; constant half_period: in
time);
```

クロックを生成します (clk)。half_period には、半周期の時間を入力します。

1.2.1 Example

```
1 process begin
2   make_clock(clk, 5 ns); -- 100MHz=10ns clock
3 end process;
```

1.3 make_reset()

```
procedure make_reset(signal rstn: out std_logic; signal clk: in std_logic;  
constant cyc: in natural:=1);
```

負論理のリセットを生成します (rstn)。clk には、クロック信号、cyc に clk 何周期分遅延させるかを指定します。

1.3.1 Example

```
1 process begin  
2     make_reset(rstn, clk, 5); -- reset  
3 end process;
```

1.4 rand()

```
impure function rand_slv(constant size: positive ) return std_logic_vector;
```

呼び出し毎にランダムな、size で指定したビット幅の std_logic_vector を返します。Seed は固定されているため、Sim 毎では同値となります。

1.4.1 Example

```
1 variable slv: std_logic_vector(7 downto 0);  
2 ~~~  
3 slv := rand_slv(8);
```

1.5 wait_clock()

```
procedure wait_clock(signal clk: std_logic; constant num: natural:=1);
```

クロック信号の立ち上がりを num 回待ちます。

1.5.1 Example

```
1 wait_clock(clk, 5); -- wait clock rising, 5times
```