

HI4 ハードウェア設計実験

第 1 回実験

実験者：HI4 45 号 山口惺司

共同実験者：HI4 46 号 吉田旺雅

提出日：2024/06/26

1. 目的

Quartus の基本操作を確認しながら,単純な組合せ回路を作成し,作成した回路をコンポーネントとして別の回路で階層的に利用する方法を習得する.

2. 実験内容

2.1. 課題 1

2.1.1. 課題内容

「1つのスイッチの入力をそのまま1つのLEDに出力する」という回路を作成する.

2.1.2. プログラム

作成した test プログラムを図 1 に示す.

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity test is
5  port(
6      sw_in : in std_logic;
7      led_out : out std_logic);
8  end test;
9
10 architecture rtl of test is
11 begin
12     led_out <= sw_in;
13 end rtl;
```

図 1 test.vhd プログラム

2.1.3. ピン設定情報

test.vhd のピン設定情報を表 1 に示す.

表 1 test.vhd ピン設定情報

Node Name	Direction	Location
sw_in	Input	PIN_W26
led_out	Output	PIN_Y18

2.1.4. 結果

PIN_W26 を入力用スイッチとして,PIN_Y18 を出力用 LED として,利用した.
スイッチを押すと,LED が光った.

2.1.5. 考察

エンティティ宣言部分では入力モード型の変数 sw_in, 出力モード型の変数 led_out がそれぞれ 1bit で宣言されているため,二つの変数には 0 か 1 が入る.

スイッチを押すと sw_in に 1 が入り,アーキテクチャ宣言にて led_out に sw_in の値が代入されることで led_out に 1 が入り,LED が光ったと考えられる.

2.2. 課題 2

2.2.1. 課題内容

2 入力の論理積の結果を出力する回路 andgate.vhd を作成し実行せよ.

また,andgate.vhd で定義した andgate 回路をコンポーネントとして宣言し,実行する andtest.vhd を作成せよ.

2.2.2. プログラム

作成した andgate.vhd プログラムを図 2, andtest.vhd プログラムを図 3 に示す.

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity andgate is
5      port(
6          sw_in: in std_logic_vector (1 downto 0);
7          led_out : out std_logic);
8  end andgate;
9
10 architecture rtl of andgate is
11 begin
12     led_out <= (not sw_in(0) and not sw_in(1));
13 end rtl;
```

図 2 andgate.vhd プログラム

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity andtest is
5      port(
6          sw_in1, sw_in2 : in std_logic_vector (1 downto 0);
7          led_out1, led_out2 : out std_logic);
8  end andtest;
9
10 architecture rtl of andtest is
11     component andgate
12     port(
13         sw_in : in std_logic_vector (1 downto 0);
14         led_out : out std_logic);
15     end component;
16
17 begin
18     and1: andgate port map (sw_in => sw_in1, led_out => led_out1);
19     and2: andgate port map (sw_in => sw_in2, led_out => led_out2);
20 end rtl;
```

図 3 andtest.vhd プログラム

2.2.3. ピン設定情報

andtest.vhd のピン設定情報を表 2 に示す.

表 2 andtest.vhd ピン設定情報

Node Name	Direction	Location
sw_in1[0]	Input	PIN_N23
sw_in1[1]	Input	PIN_G26
sw_in2[0]	Input	PIN_W26
sw_in2[1]	Input	PIN_P23
led_out1	Output	PIN_AA20
led_out2	Output	PIN_Y18

2.2.4. 結果

andgate では 2 つのプッシュボタンを押すと LED が光った

andtest では 4 つのプッシュボタン,2 つの LED があり,任意の 2 つのプッシュボタンを押すと,任意の LED が光った.

表 2 のピン設定の場合だと,PIN_G26 と PIN_N23 のボタンを押すと PIN_Y18 の LED が光り,PIN_P23 と PIN_W26 のボタンを押すと PIN_AA20 の LED が光る.

2.2.5. 考察

andgate では `sw_in: in std_logic_vector(1 downto 0);` という文で `sw_in` は 2bit の変数として宣言され `sw_in` に二つの入力を割り当てることができるようになる.

実行結果を真理値表に表すと表 3 のようになる.

表 3 andgate 実行結果の真理値表

sw_in(0)	sw_in(1)	led_out
0	0	0
0	1	0
1	0	0
1	1	1

`led_out <= (not sw_in(0) and not sw_in(1));` という文において,なぜ not がついているかというとプッシュボタンを押していない状態の時は'1',押しているときは'0' がそれぞれの `sw_in` に入るため,not を入れることで AND ゲートの動作にしている.

andtest では andgate をコンポーネントとして宣言することで簡単に andgate を呼び出すことができる. そのため,同じプログラムを複数回呼び出すときにはコンポーネントとして宣言すると効率が良くなるということがわかる.

andgate によって `sw_in1(0)` と `sw_in1(1)` の論理積が計算され `led_out1` に代入,`sw_in2(0)` と `sw_in2(1)` の論理積が計算され `led_out2` に代入されている.

2.3. 応用課題

2.3.1. 課題内容

課題で作成したプログラムを変更して動作が変わることを確認したり,ピン設定を変えることで入出力のデバイスが変更されることを確認したりして,VHDL の特徴をつかめ.

2.3.2. プログラム

変更した課題 2.1 のプログラム test_o.vhd を図 4, 変更した課題 2.2 のプログラム andtest_o.vhd を図 75 示す.

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity test_o is
5  port(
6      sw_in1, sw_in2 : in std_logic;
7      led_out1, led_out2 : out std_logic);
8  end test_o;
9
10 architecture rtl of test_o is
11 begin
12     led_out1 <= sw_in1;
13     led_out2 <= sw_in2;
14 end rtl;
```

図 6 test_o.vhd プログラム

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity andtest_o is
5  port(
6      sw_in1, sw_in2 : in std_logic_vector (1 downto 0);
7      led_out1, led_out2 : out std_logic);
8  end andtest_o;
9
10 architecture rtl of andtest_o is
11     component andgate
12     port(
13         sw_in : in std_logic_vector (1 downto 0);
14         led_out : out std_logic);
15     end component;
16
17 begin
18     and1: andgate port map (sw_in => sw_in1, led_out => led_out1);
19     and2: andgate port map (sw_in => sw_in2, led_out => led_out2);
20 end rtl;
```

図 7 andtest_o.vhd プログラム

2.3.3. ピン設定情報

test_o.vhd のピン設定情報を表 4, andtest_o.vhd のピン設定情報を表 5 に示す.

表 4 test_o.vhd のピン設定情報

Node Name	Direction	Location
sw_in1	Input	PIN_G26
sw_in2	Input	PIN_N23
led_out1	Output	PIN_AA20
led_out2	Output	PIN_Y18

表 5 andtest_o.vhd のピン設定情報

Node Name	Direction	Location
sw_in1[0]	Input	PIN_N23
sw_in1[1]	Input	PIN_G26
sw_in2[0]	Input	PIN_W26
sw_in2[1]	Input	PIN_P23
led_out1	Output	PIN_AF22
led_out2	Output	PIN_W19

2.3.4. 結果

test_o.vhd では PIN_G26 のスイッチを押すと,PIN_AA20 の LED が光り,PIN_N23 のスイッチを押すと,PIN_Y18 が光る.

andtest_o.vhd では課題 2.2 と同じような動作をする.

2.3.5. 考察

test_o.vhd の test.vhd からの変更点として,入力と出力を 1 つずつ増やし,スイッチを押すと対応する LED が光るようになっている.

andtest_o.vhd の andtest.vhd からの変更点として,ピン設定を変更している.

ピン設定を変更しても正しく動作した.

3. 感想

慣れない言語でエラーが出ることもあったが,正しく動作し,仕組みを理解することができた.

プログラムの書き方やピンの設定の仕方,装置に実装するまでの全体の流れをつかむことができたため,今後の実験に活かすことができるだろう.

4. 参考文献

ALTERA DE2 User Manual