HI4ハードウェア設計実験  
第2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：2024/07/10

# 目的

第1回実験で学んだことを活かし、7セグメントの使い方や、case文の使い方などを学ぶ。

# 実験内容

## 課題1

* + 1. 課題内容

2ビットの入力どうしの加算結果を出力するプログラムを作成しなさい。入力はトグルスイッチ、出力は任意の単体LEDを用いること。

* + 1. プログラム

作成したプログラムを以下に示す。

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity add is

    port (

        A, B: in std\_logic\_vector(1 downto 0);

        Z : out std\_logic\_vector(2 downto 0)

    );

end add;

architecture rtl of add is

begin

    Z <= ('0' & A) + ('0' & B);

end architecture rtl;

* + 1. ピン設定情報

ピン設定情報を図1に示す。

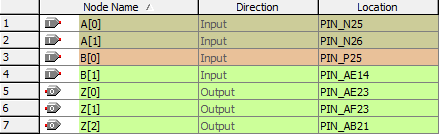


図1 課題1ピン設定情報

* + 1. 結果

結果を真理値表にしたものを表1に示す。

表1 課題1真理値表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 入力 | | | | 出力 | | |
| A[1] | A[0] | B[1] | B[0] | Z[2] | Z[1] | Z[0] |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |

* + 1. 考察

この回路は2ビットどうしの加算結果を出力するものである。

2ビットどうしの加算なので出力は3ビットにするとよいことがわかる。

* 1. 課題2
     1. 課題内容

トグルスイッチから入力される4ビットの値の16進表現を1つの7セグメントに出力するプログラムを作成しなさい。

* + 1. プログラム

作成したプログラムを以下に示す。

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity segment is

    port(

        A: in std\_logic\_vector(3 downto 0);

        Z: out std\_logic\_vector(6 downto 0)

    );

end segment;

architecture rtl of segment is

begin

    process(A)

    begin

        case A is

            when "0000" => Z <= "1000000";

            when "0001" => Z <= "1111001";

            when "0010" => Z <= "0100100";

            when "0011" => Z <= "0110000";

            when "0100" => Z <= "0011001";

            when "0101" => Z <= "0010010";

            when "0110" => Z <= "0000010";

            when "0111" => Z <= "1011000";

            when "1000" => Z <= "0000000";

            when "1001" => Z <= "0010000";

            when "1010" => Z <= "0001000";

            when "1011" => Z <= "0000011";

            when "1100" => Z <= "1000110";

            when "1101" => Z <= "0100001";

            when "1110" => Z <= "0000110";

            when "1111" => Z <= "0001110";

        end case;

    end process;

end rtl;

* + 1. ピン設定情報

ピン設定情報を図2に示す。

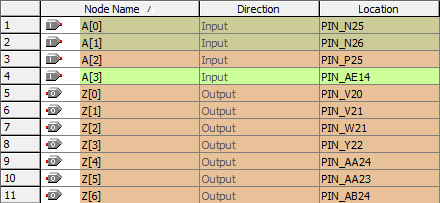


図2 課題2ピン設定情報

* + 1. 結果

結果を真理値表にしたものを表2に示す。

また、Z[0]～Z[6]までをまとめてZとし、Zは7セグメントで表示された数字やアルファベットとする。

表2 課題2真理値表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入力 | | | | 出力 |
| A[3] | A[2] | A[1] | A[0] | Z |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 3 |
| 0 | 1 | 0 | 0 | 4 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 0 | 6 |
| 0 | 1 | 1 | 1 | 7 |
| 1 | 0 | 0 | 0 | 8 |
| 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 1 | 0 | A |
| 1 | 0 | 1 | 1 | b |
| 1 | 1 | 0 | 0 | C |
| 1 | 1 | 0 | 1 | d |
| 1 | 1 | 1 | 0 | E |
| 1 | 1 | 1 | 1 | F |

* + 1. 考察

このプログラムは4ビットの入力の16進表現を7セグメントに出力するものである。

0～Fまでを7セグメントで出力するのに規則性はないため、case文を使用して入力で場合分けしている。

bとdは7セグメントで表示するときは大文字で表現できないため、小文字で表示している。

* 1. 課題3
     1. 課題内容

課題2で作成した回路をコンポーネントとして利用し、8ビットの入力を2桁の16進(7セグメントを使用)で出力するプログラムを作成しなさい。

* + 1. プログラム

作成したプログラムを以下に示す。

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity segment2 is

    port(

        A1, A2 : in std\_logic\_vector(3 downto 0);

        Z1, Z2 : out std\_logic\_vector(6 downto 0)

    );

end segment2;

architecture rtl of segment2 is

    component segment

        port(

            A: in std\_logic\_vector(3 downto 0);

            Z: out std\_logic\_vector(6 downto 0)

        );

    end component;

    begin

        seg1: segment port map(A => A1, Z => Z1);

        seg2: segment port map(A => A2, Z => Z2);

end rtl;

* + 1. ピン設定情報

ピン設定情報を図3に示す。

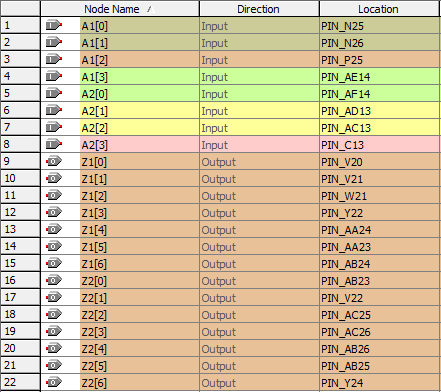


図3 課題3ピン設定情報

* + 1. 結果

課題2と同様に結果を真理値表にしたものを表3に示す。

表3 課題3真理値表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 入力 | | | | | | | | 出力 | |
| A2[3] | A2[2] | A2[1] | A2[0] | A1[3] | A1[2] | A1[1] | A1[0] | Z2 | Z1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 2 | 2 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 3 | 3 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 4 | 4 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 5 | 5 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 6 | 6 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 7 | 7 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 8 | 8 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 9 | 9 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | A | A |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | b | b |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | C | C |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | d | d |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | E | E |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | F | F |

* + 1. 考察

この回路は課題2をコンポーネントとして利用して、8ビットの入力を2桁の16進で出力するものである。

出力結果は課題2とほとんど変わらず、A1の入力がZ1にA2の入力がZ2に対応している。

回路をコンポーネントとして利用することで、プログラムを簡潔に書くことができる。

* 1. 課題4
     1. 課題内容

以下に示すような2ビットの比較回路を作成しなさい。

入力：A, B (各2ビット)

出力： A > Bの時：W = 1, X = 0, Y = 0

A = Bの時：W = 0, X = 1, Y = 0

A < Bの時：W = 0, X = 0, Y = 1

* + 1. プログラム

作成したプログラムを以下に示す。

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity compare is

    port(

        A, B: in std\_logic\_vector(1 downto 0);

        W, X, Y: out std\_logic);

end compare;

architecture rtl of compare is

begin

    process(A, B)

    begin

        if(A > B) then

            W <= '1';

            X <= '0';

            X <= '0';

        elsif(A = B) then

            X <= '1';

            W <= '0';

            Y <= '0';

        else

            Y <= '1';

            X <= '0';

            W <= '0';

        end if;

    end process;

end rtl;

* + 1. ピン設定情報

ピン設定情報を図4に示す。

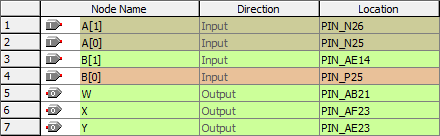


図4 課題4ピン設定情報

* + 1. 結果

結果を真理値表にしたものを表4に示す。

表4 課題4真理値表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 入力 | | | | 出力 | | |
| B[1] | B[0] | A[1] | A[0] | W | X | Y |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

* + 1. 考察

この回路は2ビットどうしの入力を比較し、比較結果を出力するものである。

if文を使用することで、場合分けをしている。

* 1. 課題5
     1. 課題内容

4ビット入力に対する2の補数回路を作成しなさい。

* + 1. プログラム

作成したプログラムを以下に示す。

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity work35 is

    port(

        sw\_in: in std\_logic\_vector (3 downto 0);

        led\_out: out std\_logic\_vector(3 downto 0)

    );

end work35;

architecture rtl of work35 is

begin

    led\_out <= not sw\_in + 1;

end rtl;

* + 1. ピン設定情報

ピン設定情報を図5に示す。

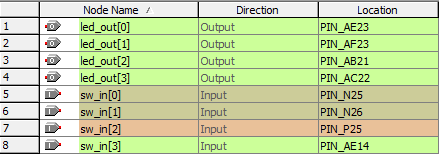


図5 課題5ピン設定情報

* + 1. 結果

結果を真理値表にしたものを表5に示す。

表5 課題5真理値表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 入力 | | | | 出力 | | | |
| sw\_in[3] | sw\_in [2] | sw\_in [1] | sw\_in [0] | led\_out[3] | led\_out [2] | led\_out [1] | led\_out [0] |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |

* + 1. 考察

この回路は4ビットの入力に対する2の補数を出力する回路である。

2の補数は、与えられた2進数の数値の0と1を入れ替え,1を足すことで表現できる。

表5を見ると2の補数が正しく出力されているため、この回路は正しいと言える。

* 1. 応用課題
     1. 課題内容

ウェブのプログラムなどを参考に、各自オリジナルの回路を設計し実行しなさい。オリジナルの回路は組み合わせ回路とする。

今回、作成した回路は半加算器と全加算器である。

半加算器の回路をコンポーネントとして全加算器を作成した。

* + 1. プログラム

半加算器のプログラムを以下に示す.

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity halfadder is

    port(

        A, B: in std\_logic;

        C, S: out std\_logic

    );

end halfadder;

architecture rtl of halfadder is

begin

    C <= A and B;

    S <= A xor B;

end rtl;

全加算器のプログラムを以下に示す.

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity fulladder is

    port(

        A, B, Cin : in std\_logic;

        Cout, S : out std\_logic

    );

end fulladder;

architecture rtl of fulladder is

    signal C1, C2, S1 : std\_logic;

component halfadder

    port(

        A, B: in std\_logic;

        C, S: out std\_logic

    );

end component;

begin

    half1: halfadder port map(A => A, B => B, C => C1, S => S1);

    half2: halfadder port map(A => S1, B => Cin, C => C2, S => S);

    Cout <= C1 or C2;

end rtl;

* + 1. ピン設定情報

全加算器のピン設定情報を図6に示す。

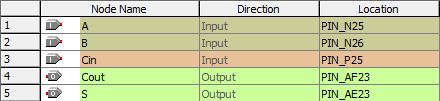


図6 課題6全加算器ピン設定情報

* + 1. 結果

全加算器の出力を真理値表にしたものを表6に示す。

表6 応用課題 全加算器の真理値表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入力 | | | 出力 | |
| Cin | A | B | Cout | S |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

* + 1. 考察

半加算器の回路図を図7, 全加算器の回路図を図8に示す。

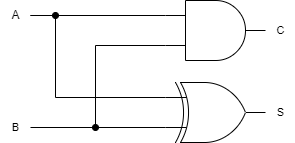


図7 半加算器



図8 全加算器

図7,8より全加算器は半加算器を2つ使用していることが分かるため、半加算器の回路を作成し、コンポーネントとして宣言することで全加算器の回路を作成した。

全加算器は下位からの桁上がりを考慮する。

Cinが0の時は下位からの桁上がりがない状態、Cinが1の時は桁上がりがある状態である。

したがって、出力をCoutSの順に2進数で見る時、CoutS=A+B+Cinであり、これは表6の真理値表と一致するため、結果は正しいと言える。

# 感想

7セグメントのピン設定が、設定するピンの数が多く、大変だった。

応用課題の全加算器を作成するにあたって、今まで学んだことを活用できたのでよかった。

# 参考文献

工業大学生ももやまのうさぎ塾

https://www.momoyama-usagi.com/entry/info-calc-sys09