

# 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

## 1. 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。

## 2. 実験内容

### 2.1. 課題1

#### 2.1.1. 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。

#### 2.1.2. 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;
```

#### 2.1.3. ピン設定情報

ピン設定情報を図1に示す。








		Node Name #	Direction	Location
1		A[0]	Input	PIN_N25
2		A[1]	Input	PIN_N26
3		B[0]	Input	PIN_P25
4		B[1]	Input	PIN_AE14
5		Z[0]	Output	PIN_AE23
6		Z[1]	Output	PIN_AF23
7		Z[2]	Output	PIN_AB21

図1 課題1 ピン設定情報

### 2.1.4. 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

### 2.1.5. 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。

## 2.2. 課題 2

### 2.2.1. 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。

### 2.2.2. 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;

```

### 2.2.3. ピン設定情報

ピン設定情報を図 2 に示す。












	Node Name /	Direction	Location
1	 A[0]	Input	PIN_N25
2	 A[1]	Input	PIN_N26
3	 A[2]	Input	PIN_P25
4	 A[3]	Input	PIN_AE14
5	 Z[0]	Output	PIN_V20
6	 Z[1]	Output	PIN_V21
7	 Z[2]	Output	PIN_W21
8	 Z[3]	Output	PIN_Y22
9	 Z[4]	Output	PIN_AA24
10	 Z[5]	Output	PIN_AA23
11	 Z[6]	Output	PIN_AB24

図 2 課題 2 ピン設定情報

#### 2.2.4. 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

#### 2.2.5. 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。

### 2.3. 課題 3

#### 2.3.1. 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。

#### 2.3.2. 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;

```

### 2.3.3. ピン設定情報

ピン設定情報を図 3 に示す。























		Node Name /	Direction	Location
1		A1[0]	Input	PIN_N25
2		A1[1]	Input	PIN_N26
3		A1[2]	Input	PIN_P25
4		A1[3]	Input	PIN_AE14
5		A2[0]	Input	PIN_AF14
6		A2[1]	Input	PIN_AD13
7		A2[2]	Input	PIN_AC13
8		A2[3]	Input	PIN_C13
9		Z1[0]	Output	PIN_V20
10		Z1[1]	Output	PIN_V21
11		Z1[2]	Output	PIN_W21
12		Z1[3]	Output	PIN_Y22
13		Z1[4]	Output	PIN_AA24
14		Z1[5]	Output	PIN_AA23
15		Z1[6]	Output	PIN_AB24
16		Z2[0]	Output	PIN_AB23
17		Z2[1]	Output	PIN_V22
18		Z2[2]	Output	PIN_AC25
19		Z2[3]	Output	PIN_AC26
20		Z2[4]	Output	PIN_AB26
21		Z2[5]	Output	PIN_AB25
22		Z2[6]	Output	PIN_Y24

図 3 課題 3 ピン設定情報

### 2.3.4. 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

### 2.3.5. 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。

## 2.4. 課題 4

### 2.4.1. 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

### 2.4.2. 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';
            Y <= '0';
        elsif(A = B) then
            X <= '1';
            W <= '0';
            Y <= '0';
        else
            Y <= '1';
            X <= '0';
            W <= '0';
        end if;
    end process;
end rtl;

```

### 2.4.3. ピン設定情報

ピン設定情報を図 4 に示す。








		Node Name	Direction	Location
1		A[1]	Input	PIN_N26
2		A[0]	Input	PIN_N25
3		B[1]	Input	PIN_AE14
4		B[0]	Input	PIN_P25
5		W	Output	PIN_AB21
6		X	Output	PIN_AF23
7		Y	Output	PIN_AE23

図 4 課題 4 ピン設定情報



#### 2.4.4. 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

#### 2.4.5. 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。

### 2.5. 課題 5

#### 2.5.1. 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。

#### 2.5.2. 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;

```

### 2.5.3. ピン設定情報

ピン設定情報を図 5 に示す。









	Node Name /	Direction	Location
1	 led_out[0]	Output	PIN_AE23
2	 led_out[1]	Output	PIN_AF23
3	 led_out[2]	Output	PIN_AB21
4	 led_out[3]	Output	PIN_AC22
5	 sw_in[0]	Input	PIN_N25
6	 sw_in[1]	Input	PIN_N26
7	 sw_in[2]	Input	PIN_P25
8	 sw_in[3]	Input	PIN_AE14

図 5 課題 5 ピン設定情報

### 2.5.4. 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

### 2.5.5. 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。

## 2.6. 応用課題

### 2.6.1. 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。

### 2.6.2. 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;

```

### 2.6.3. ピン設定情報

全加算器のピン設定情報を図 6 に示す。






		Node Name	Direction	Location
1		A	Input	PIN_N25
2		B	Input	PIN_N26
3		Cin	Input	PIN_P25
4		Cout	Output	PIN_AF23
5		S	Output	PIN_AE23

図 6 課題 6 全加算器ピン設定情報

### 2.6.4. 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

### 2.6.5. 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。

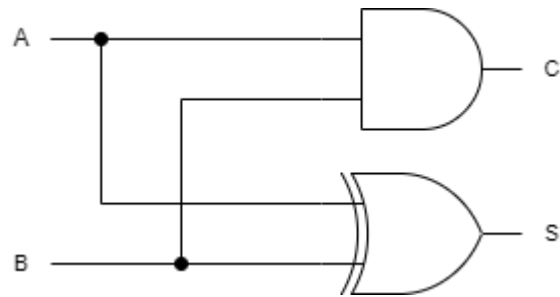


図 7 半加算器



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りを考慮する。

$C_{in}$  が 0 の時は下位からの桁上りがない状態、 $C_{in}$  が 1 の時は桁上りがある状態である。

したがって、出力を  $C_{out}S$  の順に 2 進数で見ると、 $C_{out}S = A + B + C_{in}$  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。

## 3. 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。

## 4. 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>