

MODUL 10

IMPLEMENTASI RANGKAIAN DIGITAL PENJUMLAH, PENGURANG DAN PENGALI DI FPGA MENGGUNAKAN VERILOG-HDL

1. Tujuan Praktikum

Setelah mempraktekan topik ini, praktikan diharapkan dapat :

- Praktikan dapat memahami konsep dasar rangkaian aritmatika, termasuk penjumlahan, pengurangan, dan perkalian pada bilangan biner.
- Praktikan dapat memahami dan menggunakan bahasa Verilog dengan menjelaskan setiap sirkuit dalam Verilog, meningkatkan kemampuan merancang rangkaian digital.
- Praktikan dapat memahami dan mengimplementasikan code Verilog pada platform FPGA DE10-Lite

2. Dasar Teori

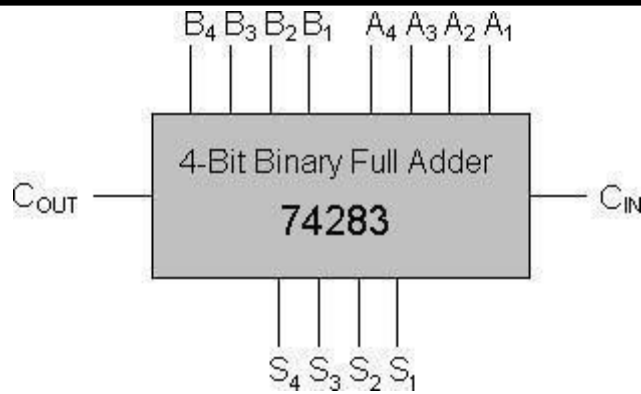
2.1 Sistem Bilangan Biner

Sistem Bilangan Biner adalah representasi angka menggunakan basis, 0 dan 1. Pentingnya representasi biner dalam rangkaian digital terletak pada penyimpanan dan pemrosesan informasi dalam bentuk biner. Setiap digit biner menunjukkan keadaan logika dalam rangkaian, 0 untuk logika rendah (OFF) dan 1 untuk logika tinggi (ON). Operasi matematika seperti penjumlahan, pengurangan, dan pengalian dalam rangkaian digital dilakukan dengan menggunakan bilangan biner.

Logika Kombinasional merupakan jenis logika di mana output hanya bergantung pada kombinasi saat ini dari input. Ini melibatkan penggunaan dasar-dasar logika seperti gerbang logika AND, OR, NOT, NAND, NOR, XOR, dan XNOR, serta persamaan logika Boolean untuk menggambarkan hubungan antara input dan output dalam fungsi logika. Dalam merancang fungsi-fungsi dasar seperti penjumlahan, pengurangan, dan pengalian dalam rangkaian digital, logika kombinasional menjadi kunci dalam memastikan operasi yang tepat dan efisien.

2.2 Adder

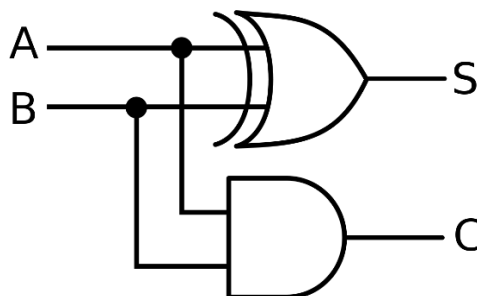
Penjumlah atau Adder adalah komponen elektronika digital yang dipakai untuk menjumlahkan dua buah angka dalam sistem bilangan biner. Dalam komputer dan mikroprosesor, Adder biasanya berada di bagian ALU (Arithmetic Logic Unit). Sistem bilangan yang dipakai dalam proses penjumlahan, selain bilangan biner, juga 2's complement untuk bilangan negatif, bilangan BCD (binary-coded decimal), dan excess- jika sistem bilangan yang dipakai adalah 2's complement, maka proses operasi penjumlahan dan operasi pengurangan akan sangat mudah dilakukan.



Gambar 1 IC 74283 (4-bit Adder)

2.2 Half Adder

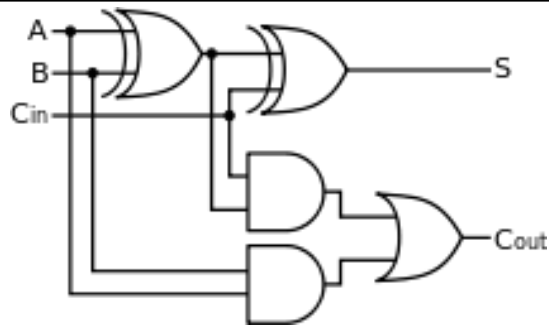
Sebuah Half Adder merupakan rangkaian elektronik yang bertugas melakukan penjumlahan dari dua bilangan biner. Dengan dua input dan dua output, ini memungkinkan untuk menghasilkan hasil penjumlahan serta carry-out. Namun, kekurangannya terletak pada validitasnya yang terbatas hanya untuk penjumlahan pertama. Penggunaan berulang dari Half Adder tidak dapat menjamin kebenarannya karena carry-in yang diperlukan untuk penjumlahan berikutnya tidak dipertimbangkan. Meskipun demikian, Half Adder merupakan fondasi penting untuk membangun rangkaian lebih kompleks seperti Full Adder yang digunakan dalam operasi penjumlahan biner yang lebih luas. Rangkaian ini memiliki dua input dan dua buah output, salah satu outputnya dipakai sebagai tempat nilai pindahan (carry) dan yang lain sebagai hasil dari penjumlahan (sum).



Gambar 2 Rangkaian Half Adder

2.3 Full Adder

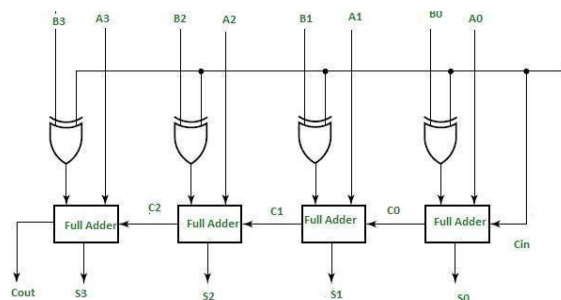
Full Adder adalah rangkaian elektronik yang bekerja melakukan perhitungan penjumlahan sepenuhnya dari dua buah bilangan biner, yang masing-masing terdiri dari satu bit. Rangkaian ini memiliki tiga input dan dua buah output, salah satu input merupakan nilai dari pindahan penjumlahan (carry in). Kemudian sama seperti pada half adder salah satu outputnya dipakai sebagai tempat nilai pindahan (carry out) dan yang lain sebagai hasil dari penjumlahan (sum).



Gambar 3 Rangkaian Full Adder

2.4 Binary Adder-Subtractor

Binary Adder-Subtractor adalah salah satu yang mampu menambah dan mengurangi bilangan biner dalam satu sirkuit itu sendiri. Operasi yang dilakukan tergantung pada nilai biner yang dimiliki oleh sinyal kontrol. Ini adalah salah satu komponen dari ALU (Unit Logika Aritmatika). Sirkuit ini Membutuhkan pengetahuan prasyarat Gerbang Exor, Penambahan dan Pengurangan Biner, Penambah Lengkap. Mari kita pertimbangkan dua angka biner 4-bit A dan B sebagai input ke Sirkuit Digital untuk operasi dengan digit.



Gambar 4 6 Adder-Subtractor

Jika nilai K (garis Kontrol) adalah 1, keluaran th dari B0 (exor) $K = B0'$ (Komplemen B0). Dengan demikian operasi akan menjadi $A + (B0')$. Sekarang pengurangan komplemen 2 untuk dua angka A dan B diberikan oleh $A + B'$. Ini menunjukkan bahwa ketika $K = 1$, operasi yang dilakukan pada empat angka bit adalah pengurangan.

Demikian pula jika Nilai $K = 0$, $B0$ (exor) $K = B0$. Operasi adalah $A + B$ yang merupakan penambahan biner sederhana. Ini menunjukkan bahwa Ketika $K = 0$, operasi yang dilakukan pada empat angka bit adalah tambahan.

3. Lembar Kegiatan Praktikum

3.1 Alat dan Bahan

- Software Quartus II
- ModelSim
- Mouse
- Laptop

3.2 Langkah Praktikum Modul 10

PART 1

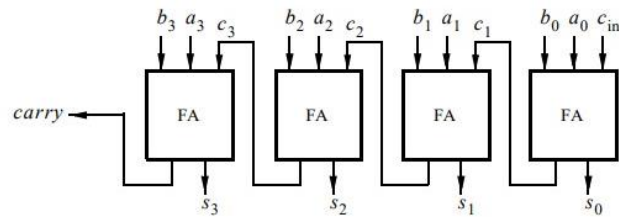


Figure 1: A four-bit ripple carry adder.

Pada part1 ini, diminta untuk mengimplementasikan rangkaian yang ditunjukkan dalam gambar dibawah ini , yang umumnya dikenal sebagai akumulator. Tujuannya adalah untuk secara berulang menambahkan nilai masukan A ke dirinya sendiri. Rangkaian ini memiliki dua keluaran penting: carry out dari penjumlahan dan sinyal overflow. Jika kita memperlakukan input A sebagai angka 2's-complement, itu berarti input tersebut bisa positif atau negatif. Jadi, sinyal overflow harus diaktifkan jika hasil penjumlahan melebihi kapasitas yang dapat direpresentasikan oleh angka 2's-complement. Hal ini penting untuk mengetahui kapan hasilnya tidak benar, terutama saat berurusan dengan angka negatif. Oleh karena itu, jika hasil penjumlahannya melebihi rentang yang dapat direpresentasikan oleh angka 2's-complement, sinyal overflow harus diatur menjadi 1 untuk menandakan hasil yang tidak benar.

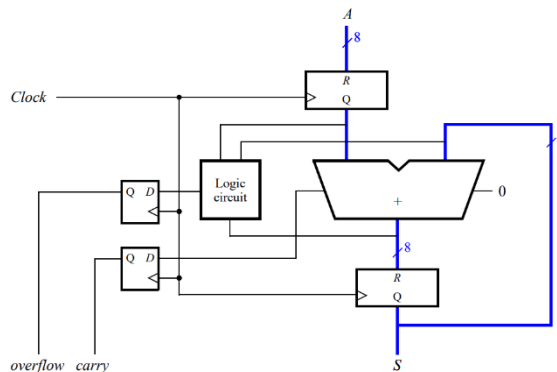


Figure 2: An eight-bit accumulator circuit.

1. Tuliskan code Verilog dibawah ini

```
1  module part1 (KEY, SW, LEDR, HEX3, HEX2, HEX1, HEX0);
2      input [1:0] KEY;
3      input [7:0] SW;
4      output [9:0] LEDR;
5      output [0:6] HEX3, HEX2, HEX1, HEX0;
6
7      wire [7:0] A, A_reg, S, S_reg;
8      wire carry, carry_reg;
9      wire Clock, Resetn, overflow, overflow_reg;
10
11     assign Resetn = KEY[0];
12     assign Clock = KEY[1];
13     assign A = SW;
14     // instantiate module regn (R, Clock, Resetn, Q);
15     regn U_A (A, Clock, Resetn, A_reg);
16
17     assign {carry, S} = A_reg + S_reg;
18     // Display the adder outputs
19     assign LEDR[7:0] = S;
20
21     // instantiate module regn (R, Clock, Resetn, Q);
22     regn U_S (S, Clock, Resetn, S_reg);
23     regn #(n(1)) U_carry (carry, Clock, Resetn, carry_reg);
24
25     // check for overflow
26     assign overflow = (A_reg[7] ^ S_reg[7]) & (A_reg[7] ^ S[7]);
27     regn #(n(1)) U_overflow (overflow, Clock, Resetn, overflow_reg);
28     assign LEDR[9] = overflow_reg;
29     assign LEDR[8] = carry_reg;
30
31     // drive the displays through 7-seg decoders
32     hex7seg digit_3 (A_reg[7:4], HEX3);
33
34     hex7seg digit_2 (A_reg[3:0], HEX2);
35     hex7seg digit_1 (S_reg[7:4], HEX1);
36     hex7seg digit_0 (S_reg[3:0], HEX0);
37 endmodule
38
39 module regn (R, Clock, Resetn, Q);
40     parameter n = 8;
41     input [n-1:0] R;
42     input Clock, Resetn;
43     output [n-1:0] Q;
44     reg [n-1:0] Q;
45
46     always @(posedge Clock or negedge Resetn)
47         if (Resetn == 0)
48             Q <= {n{1'b0}};
49         else
50             Q <= R;
51 endmodule
52
53 module hex7seg (hex, display);
54     input [3:0] hex;
55     output [0:6] display;
56
57     reg [0:6] display;
58
59     /*
60      *      0
61      *      ---
62      *      |      |
63      *      5|  6 |1
64      *      |      |
65      *      ---
66      *      |      |
67      */
```

Modul Praktikum Sisdig

```

65      *
66      *      4 | | 2
67      *
68      *      | |
69      *      | |
70      */
71      always @ (hex)
72      case (hex)
73          4'h0: display = 7'b0000001;
74          4'h1: display = 7'b1001111;
75          4'h2: display = 7'b0010010;
76          4'h3: display = 7'b0000110;
77          4'h4: display = 7'b1001100;
78          4'h5: display = 7'b0100100;
79          4'h6: display = 7'b0100000;
80          4'h7: display = 7'b0001111;
81          4'h8: display = 7'b0000000;
82          4'h9: display = 7'b0000100;
83          4'ha: display = 7'b0001000;
84          4'hb: display = 7'b1100000;
85          4'hc: display = 7'b0110001;
86          4'hd: display = 7'b1000010;
87          4'he: display = 7'b0110000;
88          4'hf: display = 7'b0111000;
89      endcase
90  endmodule
91

```

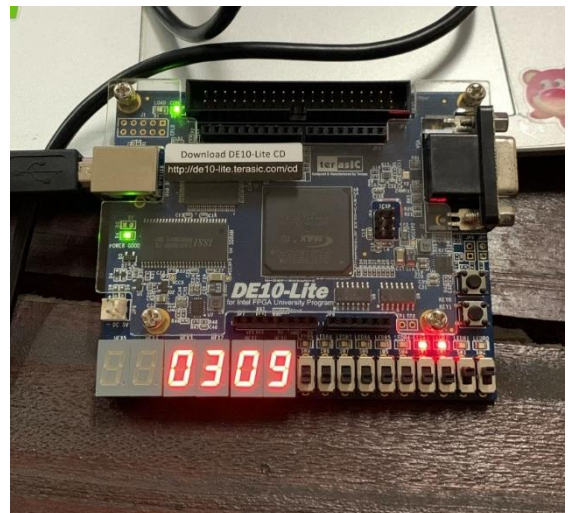
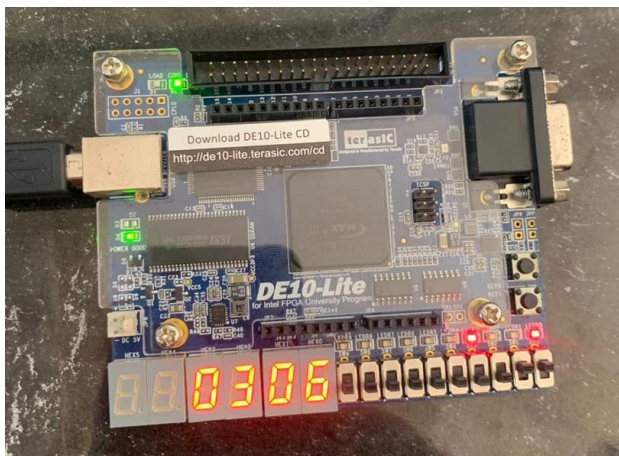
2. Lalu compile dan lanjutkan masukan Pin Planner

Node Name	Direction	Location	I/O Bank	VREF Group	Pin Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	IOCT Preservati
OUT HEX0[0]	Output	PIN_C14	7	B7_NO	PIN_C14	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX0[1]	Output	PIN_E15	7	B7_NO	PIN_E15	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX0[2]	Output	PIN_C15	7	B7_NO	PIN_C15	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX0[3]	Output	PIN_C16	7	B7_NO	PIN_C16	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX0[4]	Output	PIN_E16	7	B7_NO	PIN_E16	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX0[5]	Output	PIN_D17	7	B7_NO	PIN_D17	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX0[6]	Output	PIN_C17	7	B7_NO	PIN_C17	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX1[0]	Output	PIN_C18	7	B7_NO	PIN_C18	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX1[1]	Output	PIN_D18	6	B6_NO	PIN_D18	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX1[2]	Output	PIN_E18	6	B6_NO	PIN_E18	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX1[3]	Output	PIN_B16	7	B7_NO	PIN_B16	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX1[4]	Output	PIN_A17	7	B7_NO	PIN_A17	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX1[5]	Output	PIN_A18	7	B7_NO	PIN_A18	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX1[6]	Output	PIN_B17	7	B7_NO	PIN_B17	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX2[0]	Output	PIN_B20	6	B6_NO	PIN_B20	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX2[1]	Output	PIN_A20	7	B7_NO	PIN_A20	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX2[2]	Output	PIN_B19	7	B7_NO	PIN_B19	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX2[3]	Output	PIN_A21	6	B6_NO	PIN_A21	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX2[4]	Output	PIN_B21	6	B6_NO	PIN_B21	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX2[5]	Output	PIN_C22	6	B6_NO	PIN_C22	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX2[6]	Output	PIN_B22	6	B6_NO	PIN_B22	3.3-V LVTTTL		8mA (default)	2 (default)		

Node Name	Direction	Location	I/O Bank	VREF Group	Pin Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	IOCT Preservati
OUT HEX3[0]	Output	PIN_F21	6	B6_NO	PIN_F21	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX3[1]	Output	PIN_E22	6	B6_NO	PIN_E22	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX3[2]	Output	PIN_E21	6	B6_NO	PIN_E21	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX3[3]	Output	PIN_C19	7	B7_NO	PIN_C19	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX3[4]	Output	PIN_C20	6	B6_NO	PIN_C20	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX3[5]	Output	PIN_D19	6	B6_NO	PIN_D19	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT HEX3[6]	Output	PIN_E17	6	B6_NO	PIN_E17	3.3-V LVTTTL		8mA (default)	2 (default)		
IN KEY[1]	Input	PIN_A7	7	B7_NO	PIN_A7	3.3-V ...rigger		8mA (default)			
IN KEY[0]	Input	PIN_B8	7	B7_NO	PIN_B8	3.3 V ...rigger		8mA (default)			
OUT LEDR[9]	Output	PIN_B11	7	B7_NO	PIN_B11	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT LEDR[8]	Output	PIN_A11	7	B7_NO	PIN_A11	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT LEDR[7]	Output	PIN_D14	7	B7_NO	PIN_D14	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT LEDR[6]	Output	PIN_E14	7	B7_NO	PIN_E14	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT LEDR[5]	Output	PIN_C13	7	B7_NO	PIN_C13	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT LEDR[4]	Output	PIN_D13	7	B7_NO	PIN_D13	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT LEDR[3]	Output	PIN_B10	7	B7_NO	PIN_B10	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT LEDR[2]	Output	PIN_A10	7	B7_NO	PIN_A10	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT LEDR[1]	Output	PIN_A9	7	B7_NO	PIN_A9	3.3-V LVTTTL		8mA (default)	2 (default)		
OUT LEDR[0]	Output	PIN_A8	7	B7_NO	PIN_A8	3.3-V LVTTTL		8mA (default)	2 (default)		
IN SW[7]	Input	PIN_A14	7	B7_NO	PIN_A14	3.3-V LVTTTL		8mA (default)			
IN SW[6]	Input	PIN_A13	7	B7_NO	PIN_A13	3.3-V LVTTTL		8mA (default)			
IN SW[5]	Input	PIN_B12	7	B7_NO	PIN_B12	3.3-V LVTTTL		8mA (default)			
IN SW[4]	Input	PIN_A12	7	B7_NO	PIN_A12	3.3-V LVTTTL		8mA (default)			
IN SW[3]	Input	PIN_C12	7	B7_NO	PIN_C12	3.3-V LVTTTL		8mA (default)			
IN SW[2]	Input	PIN_D12	7	B7_NO	PIN_D12	3.3-V LVTTTL		8mA (default)			
IN SW[1]	Input	PIN_C11	7	B7_NO	PIN_C11	3.3-V LVTTTL		8mA (default)			
IN SW[0]	Input	PIN_C10	7	B7_NO	PIN_C10	3.3-V LVTTTL		8mA (default)			

Modul Praktikum Sisdig

3. Buka programmer dan sambungkan di FPGA DE10-Lite
 - Hasil di FPGA



PART2

Untuk memperluas rangkaian dari part1 agar rangkaian bisa melakukan penambahan dan pengurangan angka. Caranya adalah dengan menambahkan satu input baru yang namanya add_sub ke dalam rangkaian. Ketika add_sub diatur menjadi 1, rangkaian Anda akan melakukan operasi pengurangan dengan mengurangkan nilai A dari S. Sebaliknya, ketika add_sub diatur menjadi 0, rangkaian Anda akan melakukan operasi penambahan dengan menambahkan nilai A ke S, seperti yang dilakukan pada Bagian I. Dengan adanya perubahan ini, rangkaian akan mampu melakukan operasi penambahan dan pengurangan berdasarkan nilai dari input add_sub.

1. Tuliskan codingan dibawah ini

```
1 module part2 (KEY, SW, LEDR, HEX3, HEX2, HEX1, HEX0);
2   input [1:0] KEY;
3   input [9:0] SW;
4   output [9:0] LEDR;
5   output [0:6] HEX3, HEX2, HEX1, HEX0;
6
7   wire [7:0] A, A_reg, S, S_reg;
8   wire carry, carry_reg, Add_Sub;
9   wire Clock, Resetn, overflow, overflow_reg;
10
11   assign Resetn = KEY[0];
12   assign Clock = KEY[1];
13   assign A = SW[7:0];
14   // instantiate module regn (R, Clock, Resetn, Q);
15   regn U_A (A, Clock, Resetn, A_reg);
16
17   assign Add_Sub = SW[9];
18   assign {carry, S} = S_reg + ((A_reg ^ {8{Add_Sub}}) + Add_Sub);
19   // Display the adder outputs
20   assign LEDR[7:0] = S;
21
22   // instantiate module regn (R, Clock, Resetn, Q);
23   regn U_S (S, Clock, Resetn, S_reg);
24   regn #(n(1)) U_carry (carry, Clock, Resetn, carry_reg);
25
26   // check for overflow
27   assign overflow = ((Add_Sub ^ A_reg[7]) ^ S_reg[7]) & ((A_reg[7] ^ Add_Sub) ^ S[7]);
28   regn #(n(1)) U_overflow (overflow, Clock, Resetn, overflow_reg);
29   assign LEDR[9] = overflow_reg;
30   assign LEDR[8] = carry_reg;
31
32   // drive the displays through 7-seg decoders
33   hex7seg digit_3 (A_reg[7:4], HEX3);
34   hex7seg digit_2 (A_reg[3:0], HEX2);
35   hex7seg digit_1 (S_reg[7:4], HEX1);
36   hex7seg digit_0 (S_reg[3:0], HEX0);
37 endmodule
38
39 module regn (R, Clock, Resetn, Q);
40   parameter n = 8;
41   input [n-1:0] R;
42   input Clock, Resetn;
43   output [n-1:0] Q;
44   reg [n-1:0] Q;
45
46   always @(posedge Clock or negedge Resetn)
47     if (Resetn == 0)
48       Q <= {n{1'b0}};
49     else
50       Q <= R;
51 endmodule
52
53 module hex7seg (hex, display);
54   input [3:0] hex;
55   output [0:6] display;
56
57   reg [0:6] display;
58
59   /*
60   *      0
61   *      ---
62   *      _!_ !
63   */
```

Modul Praktikum Sisdig

```

63      *      5 | 1
64      *      | 6 |
65      *      | 2 |
66      *      | 4 |
67      *      | 3 |
68      *      | 2 |
69      *      | 1 |
70      */
71
72      always @ (hex)
73      case (hex)
74      4'h0: display = 7'b0000001;
75      4'h1: display = 7'b1001111;
76      4'h2: display = 7'b0010010;
77      4'h3: display = 7'b0000110;
78      4'h4: display = 7'b1001100;
79      4'h5: display = 7'b0100100;
80      4'h6: display = 7'b0100000;
81      4'h7: display = 7'b0001111;
82      4'h8: display = 7'b0000000;
83      4'h9: display = 7'b0000100;
84      4'hA: display = 7'b0001000;
85      4'hB: display = 7'b1100000;
86      4'hC: display = 7'b0110001;
87      4'hD: display = 7'b1000010;
88      4'hE: display = 7'b0110000;
89      4'hF: display = 7'b0111000;
90      endcase
91 endmodule
92

```

2. Lalu Compile dan lanjutkan masukan Pin Planner

- Pin Planner

Node Name	Direction	Location	I/O Bank	VREF Group	Pin Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	IOCT Preservati
out HEX0[0]	Output	PIN_C14	7	B7_NO	PIN_C14	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX0[1]	Output	PIN_E15	7	B7_NO	PIN_E15	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX0[2]	Output	PIN_C15	7	B7_NO	PIN_C15	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX0[3]	Output	PIN_C16	7	B7_NO	PIN_C16	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX0[4]	Output	PIN_E16	7	B7_NO	PIN_E16	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX0[5]	Output	PIN_D17	7	B7_NO	PIN_D17	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX0[6]	Output	PIN_C17	7	B7_NO	PIN_C17	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX1[0]	Output	PIN_C18	7	B7_NO	PIN_C18	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX1[1]	Output	PIN_D18	6	B6_NO	PIN_D18	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX1[2]	Output	PIN_E18	6	B6_NO	PIN_E18	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX1[3]	Output	PIN_B16	7	B7_NO	PIN_B16	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX1[4]	Output	PIN_A17	7	B7_NO	PIN_A17	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX1[5]	Output	PIN_A18	7	B7_NO	PIN_A18	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX1[6]	Output	PIN_B17	7	B7_NO	PIN_B17	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX2[0]	Output	PIN_B20	6	B6_NO	PIN_B20	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX2[1]	Output	PIN_A20	7	B7_NO	PIN_A20	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX2[2]	Output	PIN_B19	7	B7_NO	PIN_B19	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX2[3]	Output	PIN_A21	6	B6_NO	PIN_A21	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX2[4]	Output	PIN_B21	6	B6_NO	PIN_B21	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX2[5]	Output	PIN_C22	6	B6_NO	PIN_C22	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX2[6]	Output	PIN_B22	6	B6_NO	PIN_B22	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX3[0]	Output	PIN_F21	6	B6_NO	PIN_F21	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX3[1]	Output	PIN_E22	6	B6_NO	PIN_E22	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX3[2]	Output	PIN_E21	6	B6_NO	PIN_E21	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX3[3]	Output	PIN_C19	7	B7_NO	PIN_C19	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX3[4]	Output	PIN_C20	6	B6_NO	PIN_C20	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX3[5]	Output	PIN_D19	6	B6_NO	PIN_D19	3.3-V LVTTTL		8mA (default)	2 (default)		
out HEX3[6]	Output	PIN_E17	6	B6_NO	PIN_E17	3.3-V LVTTTL		8mA (default)	2 (default)		
in KEY[1]	Input	PIN_A7	7	B7_NO	PIN_A7	3.3 V ...rigger		8mA (default)			
in KEY[0]	Input	PIN_B8	7	B7_NO	PIN_B8	3.3 V ...rigger		8mA (default)			
out LEDR[9]	Output	PIN_B11	7	B7_NO	PIN_B11	3.3-V LVTTTL		8mA (default)	2 (default)		
out LEDR[8]	Output	PIN_A11	7	B7_NO	PIN_A11	3.3-V LVTTTL		8mA (default)	2 (default)		
out LEDR[7]	Output	PIN_D14	7	B7_NO	PIN_D14	3.3-V LVTTTL		8mA (default)	2 (default)		
out LEDR[6]	Output	PIN_E14	7	B7_NO	PIN_E14	3.3-V LVTTTL		8mA (default)	2 (default)		
out LEDR[5]	Output	PIN_C13	7	B7_NO	PIN_C13	3.3-V LVTTTL		8mA (default)	2 (default)		
out LEDR[4]	Output	PIN_D13	7	B7_NO	PIN_D13	3.3-V LVTTTL		8mA (default)	2 (default)		
out LEDR[3]	Output	PIN_B10	7	B7_NO	PIN_B10	3.3-V LVTTTL		8mA (default)	2 (default)		
out LEDR[2]	Output	PIN_A10	7	B7_NO	PIN_A10	3.3-V LVTTTL		8mA (default)	2 (default)		
out LEDR[1]	Output	PIN_A9	7	B7_NO	PIN_A9	3.3-V LVTTTL		8mA (default)	2 (default)		
out LEDR[0]	Output	PIN_A8	7	B7_NO	PIN_A8	3.3-V LVTTTL		8mA (default)	2 (default)		
in SW[9]	Input	PIN_F15	7	B7_NO	PIN_F15	3.3-V LVTTTL		8mA (default)			
in SW[8]	Input	PIN_B14	7	B7_NO	PIN_B14	3.3-V LVTTTL		8mA (default)			
in SW[7]	Input	PIN_A14	7	B7_NO	PIN_A14	3.3-V LVTTTL		8mA (default)			
in SW[6]	Input	PIN_A13	7	B7_NO	PIN_A13	3.3-V LVTTTL		8mA (default)			
in SW[5]	Input	PIN_B12	7	B7_NO	PIN_B12	3.3-V LVTTTL		8mA (default)			
in SW[4]	Input	PIN_A12	7	B7_NO	PIN_A12	3.3-V LVTTTL		8mA (default)			
in SW[3]	Input	PIN_C12	7	B7_NO	PIN_C12	3.3-V LVTTTL		8mA (default)			
in SW[2]	Input	PIN_D12	7	B7_NO	PIN_D12	3.3-V LVTTTL		8mA (default)			
in SW[1]	Input	PIN_C11	7	B7_NO	PIN_C11	3.3-V LVTTTL		8mA (default)			
in SW[0]	Input	PIN_C10	7	B7_NO	PIN_C10	3.3-V LVTTTL		8mA (default)			

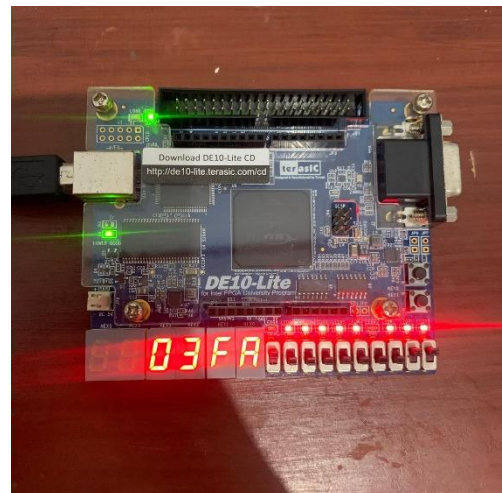
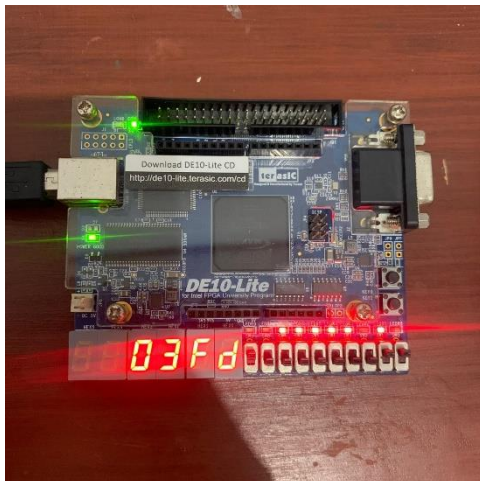
Modul Praktikum Sisdig

3. Buka programmer dan sambungkan di FPGA DE10-Lite
 - Hasil di FPGA

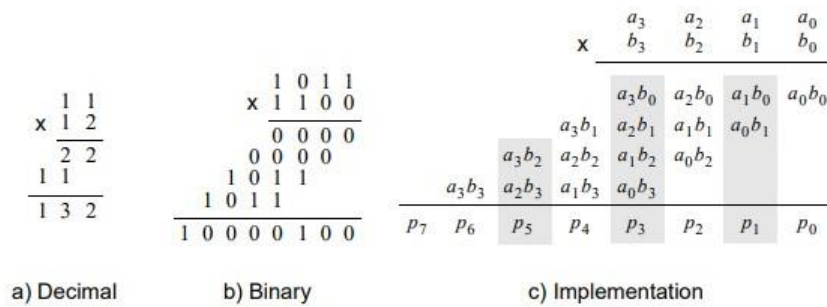
Penjumlahan



Pengurangan (switch 9 harus bernilai 1)



PART 3



Gambar perkalian bilangan biner

Pada Bagian III, sebuah pengganda array dibuat menggunakan modul full adder. Pada level yang lebih tinggi, satu baris full adder berfungsi sebagai pengganda n-bit, dan rangkaian pengganda array dapat direpresentasikan. Setiap pengganda n-bit menambahkan versi yang digeser dari A untuk setiap baris yang diberikan, dan produk parsial dari baris di atasnya. Mengabstraksi rangkaian pengganda sebagai urutan penambahan memungkinkan kita membangun pengganda yang lebih besar. Gunakan pendekatan ini untuk membuat rangkaian pengganda 8 x 8 dengan masukan dan keluaran terdaftar

1. Tuliskan codingan dibawah ini

```

1 // Implements an unsigned multiplier P = A x B.
2 // inputs: SW11-8 = A, SW3-0 = B
3 // outputs: HEX5-4 shows the product P, HEX2 shows A, HEX0 shows B
4 module part3 (SW, HEX5, HEX4, HEX2, HEX0);
5     input [7:0] SW;
6     output [0:6] HEX5, HEX4, HEX2, HEX0;
7
8     wire [3:0] A, B;
9     wire [7:0] P;
10    wire [3:1] C_b1; // carries for row that ANDs with B1
11    wire [5:2] PP1; // partial products from row that ANDs with B1
12    wire [3:1] C_b2; // carries for row that ANDs with B2
13    wire [6:3] PP2; // partial products from row that ANDs with B2
14    wire [3:1] C_b3; // carries for row that ANDs with B3
15
16    assign A = SW[7:4];
17    assign B = SW[3:0];
18    assign P[0] = A[0] & B[0];
19
20    // module fa (a, b, ci, s, co);
21    fa b1_a0 (A[1] & B[0], A[0] & B[1], 1'b0, P[1], C_b1[1]);
22    fa b1_a1 (A[2] & B[0], A[1] & B[1], C_b1[1], PP1[2], C_b1[2]);
23    fa b1_a2 (A[3] & B[0], A[2] & B[1], C_b1[2], PP1[3], C_b1[3]);
24    fa b1_a3 (1'b0, A[3] & B[1], C_b1[3], PP1[4], PP1[5]);
25
26    // module fa (a, b, ci, s, co);
27    fa b2_a0 (PP1[2], A[0] & B[2], 1'b0, P[2], C_b2[1]);
28    fa b2_a1 (PP1[3], A[1] & B[2], C_b2[1], PP2[3], C_b2[2]);
29    fa b2_a2 (PP1[4], A[2] & B[2], C_b2[2], PP2[4], C_b2[3]);
30    fa b2_a3 (PP1[5], A[3] & B[2], C_b2[3], PP2[5], PP2[6]);
31

```

Modul Praktikum Sisdig

```

31
32 // module fa (a, b, ci, s, co);
33 fa b3_a0 (PP2[3], A[0] & B[3], 1'b0, P[3], C_b3[1]);
34 fa b3_a1 (PP2[4], A[1] & B[3], C_b3[1], P[4], C_b3[2]);
35 fa b3_a2 (PP2[5], A[2] & B[3], C_b3[2], P[5], C_b3[3]);
36 fa b3_a3 (PP2[6], A[3] & B[3], C_b3[3], P[6], P[7]);
37
38 // drive the displays through a 7-seg decoders
39 hex7seg digit_5 (P[7:4], HEX5);
40 hex7seg digit_4 (P[3:0], HEX4);
41 hex7seg digit_3 (A, HEX2);
42 hex7seg digit_2 (B, HEX0);
43
44 endmodule
45
46 module fa (a, b, ci, s, co);
47 input a, b, ci;
48 output s, co;
49
50 wire a_xor_b;
51
52 assign a_xor_b = a ^ b;
53 assign s = a_xor_b ^ ci;
54 assign co = (~a_xor_b & b) | (a_xor_b & ci);
55 endmodule
56
57 module hex7seg (hex, display);
58 input [3:0] hex;
59 output [0:6] display;
60
61 reg [0:6] display;
62
63
64 *
65 * 0
66 * ---
67 * 5 | 1
68 * 6 |
69 * ---
70 * 4 | 2
71 * 3 |
72 * ---
73 * 3
74 */
75
76 always @ (hex)
77 case (hex)
78 4'h0: display = 7'b0000001;
79 4'h1: display = 7'b1001111;
80 4'h2: display = 7'b0010010;
81 4'h3: display = 7'b0000110;
82 4'h4: display = 7'b1001100;
83 4'h5: display = 7'b0100100;
84 4'h6: display = 7'b0100000;
85 4'h7: display = 7'b0001111;
86 4'h8: display = 7'b0000000;
87 4'h9: display = 7'b0000100;
88 4'ha: display = 7'b0001000;
89 4'hb: display = 7'b1100000;
90 4'hc: display = 7'b0110001;
91 4'hd: display = 7'b1000010;
92 4'he: display = 7'b0110000;
93 4'hf: display = 7'b0111000;
94 endcase
95 endmodule
96

```

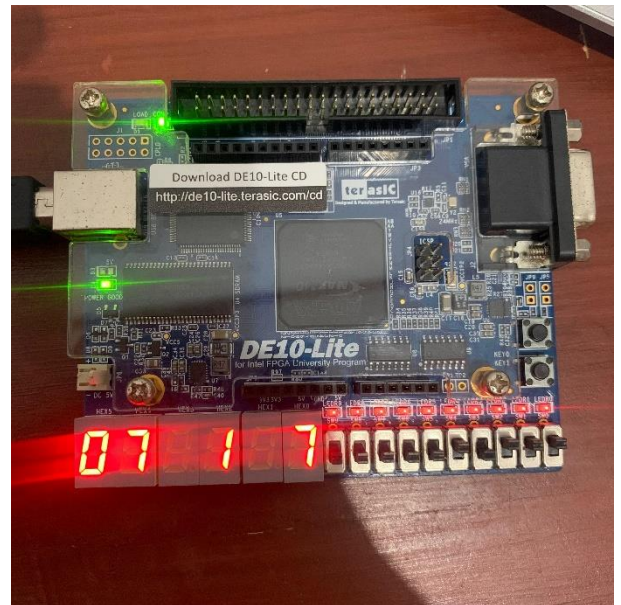
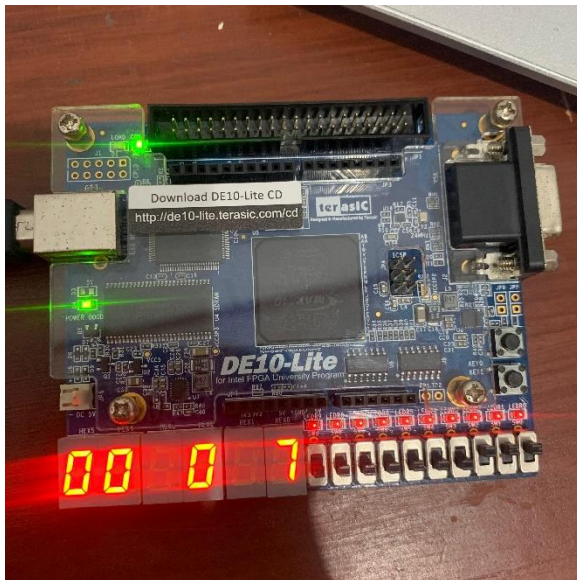
1. Lalu Compile dan lanjutkan masukan Pin Planner
- Pin Planner

Node Name	Direction	Location	I/O Bank	VREF Group	itter Location	I/O Standard	Reserved	urrent Streng	Slew Rate	ifferential Pai	ict Preservati
HEX0[0]	Output	PIN_C14	7	B7_NO	PIN_C14	2.5 V		12mA ..ault	2 (default)		
HEX0[1]	Output	PIN_E15	7	B7_NO	PIN_E15	2.5 V		12mA ..ault	2 (default)		
HEX0[2]	Output	PIN_C15	7	B7_NO	PIN_C15	2.5 V		12mA ..ault	2 (default)		
HEX0[3]	Output	PIN_C16	7	B7_NO	PIN_C16	2.5 V		12mA ..ault	2 (default)		
HEX0[4]	Output	PIN_E16	7	B7_NO	PIN_E16	2.5 V		12mA ..ault	2 (default)		
HEX0[5]	Output	PIN_D17	7	B7_NO	PIN_D17	2.5 V		12mA ..ault	2 (default)		
HEX0[6]	Output	PIN_C17	7	B7_NO	PIN_C17	2.5 V		12mA ..ault	2 (default)		
HEX2[0]	Output	PIN_B20	6	B6_NO	PIN_B20	2.5 V		12mA ..ault	2 (default)		
HEX2[1]	Output	PIN_A20	7	B7_NO	PIN_A20	2.5 V		12mA ..ault	2 (default)		
HEX2[2]	Output	PIN_B19	7	B7_NO	PIN_B19	2.5 V		12mA ..ault	2 (default)		
HEX2[3]	Output	PIN_A21	6	B6_NO	PIN_A21	2.5 V		12mA ..ault	2 (default)		
HEX2[4]	Output	PIN_B21	6	B6_NO	PIN_B21	2.5 V		12mA ..ault	2 (default)		
HEX2[5]	Output	PIN_C22	6	B6_NO	PIN_C22	2.5 V		12mA ..ault	2 (default)		
HEX2[6]	Output	PIN_B22	6	B6_NO	PIN_B22	2.5 V		12mA ..ault	2 (default)		
HEX4[0]	Output	PIN_F18	6	B6_NO	PIN_F18	2.5 V		12mA ..ault	2 (default)		
HEX4[1]	Output	PIN_E20	6	B6_NO	PIN_E20	2.5 V		12mA ..ault	2 (default)		
HEX4[2]	Output	PIN_E19	6	B6_NO	PIN_E19	2.5 V		12mA ..ault	2 (default)		
HEX4[3]	Output	PIN_J18	6	B6_NO	PIN_J18	2.5 V		12mA ..ault	2 (default)		
HEX4[4]	Output	PIN_H19	6	B6_NO	PIN_H19	2.5 V		12mA ..ault	2 (default)		
HEX4[5]	Output	PIN_F19	6	B6_NO	PIN_F19	2.5 V		12mA ..ault	2 (default)		
HEX4[6]	Output	PIN_F20	6	B6_NO	PIN_F20	2.5 V		12mA ..ault	2 (default)		
HEX5[0]	Output	PIN_J20	6	B6_NO	PIN_J20	2.5 V		12mA ..ault	2 (default)		
HEX5[1]	Output	PIN_K20	6	B6_NO	PIN_K20	2.5 V		12mA ..ault	2 (default)		
HEX5[2]	Output	PIN_L18	6	B6_NO	PIN_L18	2.5 V		12mA ..ault	2 (default)		
HEX5[3]	Output	PIN_N18	6	B6_NO	PIN_N18	2.5 V		12mA ..ault	2 (default)		
HEX5[4]	Output	PIN_M20	6	B6_NO	PIN_M20	2.5 V		12mA ..ault	2 (default)		
HEX5[5]	Output	PIN_N19	6	B6_NO	PIN_N19	2.5 V		12mA ..ault	2 (default)		
HEX5[6]	Output	PIN_N20	6	B6_NO	PIN_N20	2.5 V		12mA ..ault	2 (default)		
SW[7]	Input	PIN_A14	7	B7_NO	PIN_A14	2.5 V		12mA ..ault			

Modul Praktikum Sisdig

SW[6]	Input	PIN_A13	7	B7_N0	PIN_A13	2.5 V		12mA ...ault)				
SW[5]	Input	PIN_B12	7	B7_N0	PIN_B12	2.5 V		12mA ...ault)				
SW[4]	Input	PIN_A12	7	B7_N0	PIN_A12	2.5 V		12mA ...ault)				
SW[3]	Input	PIN_C12	7	B7_N0	PIN_C12	2.5 V		12mA ...ault)				
SW[2]	Input	PIN_D12	7	B7_N0	PIN_D12	2.5 V		12mA ...ault)				
SW[1]	Input	PIN_C11	7	B7_N0	PIN_C11	2.5 V		12mA ...ault)				
SW[0]	Input	PIN_C10	7	B7_N0	PIN_C10	2.5 V		12mA ...ault)				

- Hasil FPGA



PART 4

Setiap penjumlah n-bit menambahkan versi bergeser dari A untuk baris tertentu dan hasil perkalian parsial dari baris di atasnya. Mengabstraksi rangkaian perkalian sebagai urutan penjumlahan memungkinkan kita untuk membangun perkalian yang lebih besar. Perkalian seharusnya terdiri dari penjumlah n-bit yang disusun dalam struktur seperti yang ditunjukkan dalam Gambar 5. Gunakan pendekatan ini untuk menerapkan rangkaian perkalian 8 x 8 dengan masukan dan keluaran yang terdaftar.

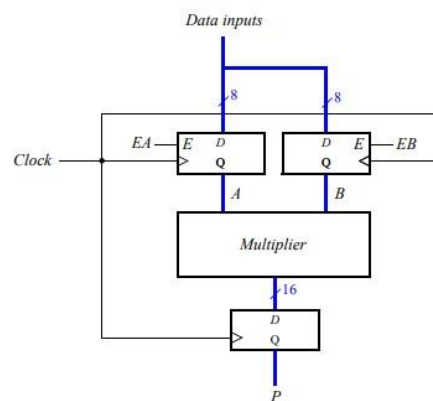


Figure 6: A registered multiplier circuit.

1. Tuliskan codingan dibawah ini

```

1 // Implements an unsigned multiplier P = A x B.
2 // inputs: SW7-0, SW9 select A, SW8 selects B, KEY[0] is reset, KEY[1] is clock
3 // outputs: HEX3-0 shows the product P
4 module part4 (SW, KEY, LEDR, HEX3, HEX2, HEX1, HEX0);
5     input [9:0] SW;
6     input [1:0] KEY;
7     output [9:0] LEDR;
8     output [0:6] HEX3, HEX2, HEX1, HEX0;
9
10    wire Resetn, Clock;
11    reg [7:0] A, B;
12    reg [15:0] P_reg;
13    wire [15:0] PP0, PP1, PP2, PP3, PP4, PP5, PP6;
14    wire [15:0] P;
15
16    assign Resetn = KEY[0];
17    assign Clock = KEY[1];
18    always @(posedge Clock) | part4
19
20    begin
21        if (~Resetn)
22            begin
23                A <= 8'b0; B <= 8'b0; P_reg <= 16'b0;
24            end
25        else
26            begin
27                if (SW[9]) A <= SW[7:0];
28                if (SW[8]) B <= SW[7:0];
29                P_reg <= P;
30            end
31    end
32

```

```

33 assign LEDR[7:0] = SW[9] ? A : (SW[8] ? B : 8'b0);
34 assign LEDR[9:8] = SW[9:8];
35
36 assign PP0 = { 8'd0, A & {8{B[0]}} };
37 assign PP1 = PP0 + { 7'd0, A & {8{B[1]}} }, 1'd0 };
38 assign PP2 = PP1 + { 6'd0, A & {8{B[2]}} }, 2'd0 };
39 assign PP3 = PP2 + { 5'd0, A & {8{B[3]}} }, 3'd0 };
40 assign PP4 = PP3 + { 4'd0, A & {8{B[4]}} }, 4'd0 };
41 assign PP5 = PP4 + { 3'd0, A & {8{B[5]}} }, 5'd0 };
42 assign PP6 = PP5 + { 2'd0, A & {8{B[6]}} }, 6'd0 };
43 assign P = PP6 + { 1'd0, A & {8{B[7]}} }, 7'd0 };
44
45 // drive the display through a 7-seg decoder
46 hex7seg digit_3 (P_reg[15:12], HEX3);
47 hex7seg digit_2 (P_reg[11:8], HEX2);
48 hex7seg digit_1 (P_reg[7:4], HEX1);
49 hex7seg digit_0 (P_reg[3:0], HEX0);
50 endmodule
51
52 module hex7seg (hex, display);
53 input [3:0] hex;
54 output [0:6] display;
55
56 reg [0:6] display;
57
58 /*
59 *      0
60 *      ---
61 *      |   |
62 *      5   1
63 *      |   |
64 *      6   |
65 *      |   |
66 *      |   |
67 *      4   2
68 *      |   |
69 *      ---
70 *      3
71 */
72 always @ (hex)
73   case (hex)
74     4'h0: display = 7'b0000001;
75     4'h1: display = 7'b1001111;
76     4'h2: display = 7'b0010010;
77     4'h3: display = 7'b0000110;
78     4'h4: display = 7'b1001100;
79     4'h5: display = 7'b0100100;
80     4'h6: display = 7'b0100000;
81     4'h7: display = 7'b0001111;
82     4'h8: display = 7'b0000000;
83     4'h9: display = 7'b0001100;
84     4'hA: display = 7'b0001000;
85     4'hb: display = 7'b1100000;
86     4'hc: display = 7'b0110001;
87     4'hd: display = 7'b1000010;
88     4'he: display = 7'b0110000;
89     4'hf: display = 7'b0111000;
90   endcase
91 endmodule

```

1. Lalu Compile dan lanjutkan masukan Pin Planner

- Pin Planner

Node Name	Direction	Location	I/O Bank	VREF Group	Pin Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	I/O Preservation
HEX0[0]	Output	PIN_C14	7	B7_NO	PIN_C14	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX0[1]	Output	PIN_E15	7	B7_NO	PIN_E15	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX0[2]	Output	PIN_C15	7	B7_NO	PIN_C15	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX0[3]	Output	PIN_C16	7	B7_NO	PIN_C16	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX0[4]	Output	PIN_E16	7	B7_NO	PIN_E16	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX0[5]	Output	PIN_D17	7	B7_NO	PIN_D17	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX0[6]	Output	PIN_C17	7	B7_NO	PIN_C17	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX1[0]	Output	PIN_C18	7	B7_NO	PIN_C18	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX1[1]	Output	PIN_D18	6	B6_NO	PIN_D18	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX1[2]	Output	PIN_E18	6	B6_NO	PIN_E18	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX1[3]	Output	PIN_B16	7	B7_NO	PIN_B16	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX1[4]	Output	PIN_A17	7	B7_NO	PIN_A17	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX1[5]	Output	PIN_A18	7	B7_NO	PIN_A18	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX1[6]	Output	PIN_B17	7	B7_NO	PIN_B17	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX2[0]	Output	PIN_B20	6	B6_NO	PIN_B20	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX2[1]	Output	PIN_A20	7	B7_NO	PIN_A20	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX2[2]	Output	PIN_B19	7	B7_NO	PIN_B19	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX2[3]	Output	PIN_A21	6	B6_NO	PIN_A21	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX2[4]	Output	PIN_B21	6	B6_NO	PIN_B21	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX2[5]	Output	PIN_C22	6	B6_NO	PIN_C22	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX2[6]	Output	PIN_B22	6	B6_NO	PIN_B22	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX3[0]	Output	PIN_F21	6	B6_NO	PIN_F21	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX3[1]	Output	PIN_E22	6	B6_NO	PIN_E22	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX3[2]	Output	PIN_E21	6	B6_NO	PIN_E21	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX3[3]	Output	PIN_C19	7	B7_NO	PIN_C19	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX3[4]	Output	PIN_C20	6	B6_NO	PIN_C20	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX3[5]	Output	PIN_D19	6	B6_NO	PIN_D19	3.3-V LVTTTL		8mA (default)	2 (default)		
HEX3[6]	Output	PIN_E17	6	B6_NO	PIN_E17	3.3-V LVTTTL		8mA (default)	2 (default)		
KEY[1]	Input	PIN_A7	7	B7_NO	PIN_A7	3.3 V ...rigger		8mA (default)			

Modul Praktikum Sisdig

in	KEY[0]	Input	PIN_B8	7	B7_NO	PIN_B8	3.3 V ...rigger	8mA (default)			
out	LEDR[9]	Output	PIN_B11	7	B7_NO	PIN_B11	3.3-V LVTTTL	8mA (default)	2 (default)		
out	LEDR[8]	Output	PIN_A11	7	B7_NO	PIN_A11	3.3-V LVTTTL	8mA (default)	2 (default)		
out	LEDR[7]	Output	PIN_D14	7	B7_NO	PIN_D14	3.3-V LVTTTL	8mA (default)	2 (default)		
out	LEDR[6]	Output	PIN_E14	7	B7_NO	PIN_E14	3.3-V LVTTTL	8mA (default)	2 (default)		
out	LEDR[5]	Output	PIN_C13	7	B7_NO	PIN_C13	3.3-V LVTTTL	8mA (default)	2 (default)		
out	LEDR[4]	Output	PIN_D13	7	B7_NO	PIN_D13	3.3-V LVTTTL	8mA (default)	2 (default)		
out	LEDR[3]	Output	PIN_B10	7	B7_NO	PIN_B10	3.3-V LVTTTL	8mA (default)	2 (default)		
out	LEDR[2]	Output	PIN_A10	7	B7_NO	PIN_A10	3.3-V LVTTTL	8mA (default)	2 (default)		
out	LEDR[1]	Output	PIN_A9	7	B7_NO	PIN_A9	3.3-V LVTTTL	8mA (default)	2 (default)		
out	LEDR[0]	Output	PIN_A8	7	B7_NO	PIN_A8	3.3-V LVTTTL	8mA (default)	2 (default)		
in	SW[9]	Input	PIN_F15	7	B7_NO	PIN_F15	3.3-V LVTTTL	8mA (default)			
in	SW[8]	Input	PIN_B14	7	B7_NO	PIN_B14	3.3-V LVTTTL	8mA (default)			
in	SW[7]	Input	PIN_A14	7	B7_NO	PIN_A14	3.3-V LVTTTL	8mA (default)			
in	SW[6]	Input	PIN_A13	7	B7_NO	PIN_A13	3.3-V LVTTTL	8mA (default)			
in	SW[5]	Input	PIN_B12	7	B7_NO	PIN_B12	3.3-V LVTTTL	8mA (default)			
in	SW[4]	Input	PIN_A12	7	B7_NO	PIN_A12	3.3-V LVTTTL	8mA (default)			
in	SW[3]	Input	PIN_C12	7	B7_NO	PIN_C12	3.3-V LVTTTL	8mA (default)			
in	SW[2]	Input	PIN_D12	7	B7_NO	PIN_D12	3.3-V LVTTTL	8mA (default)			
in	SW[1]	Input	PIN_C11	7	B7_NO	PIN_C11	3.3-V LVTTTL	8mA (default)			
in	SW[0]	Input	PIN_C10	7	B7_NO	PIN_C10	3.3-V LVTTTL	8mA (default)			
<<new node>>											

Modul Praktikum Sisdig

- Hasil di FPGA

Pada hasil di FPGA ini merupakan perkalian bilangan biner, Switch 3-0 Bagian B Dan Switch 4-7 Bagian A. (Syarat perkalian bisa digunakan adalah swich 8 & 9 bernilai 1)



Hasil perkalian :

$$21 \times 21 = 441$$



Part 5

Pada bagian IV, ditunjukkan bagaimana mengimplementasikan perkalian $A \times B$ sebagai rangkaian penjumlahan berurutan, dengan mengakumulasikan versi bergeser dari A satu baris pada satu waktu. Cara lain untuk mengimplementasikan rangkaian ini adalah dengan melakukan penambahan menggunakan pohon penambah. Pohon penambah adalah metode untuk menambahkan beberapa angka bersama-sama secara paralel. Ide ini diilustrasikan dalam Gambar 7. Dalam gambar tersebut, angka-angka A, B, C, D, E, F, G, dan H ditambahkan bersama-sama secara paralel. Penjumlahan $A + B$ terjadi secara simultan dengan $C + D$, $E + F$, dan $G + H$. Hasil dari operasi-operasi ini kemudian ditambahkan secara paralel lagi, sampai jumlah akhir P dihitung.

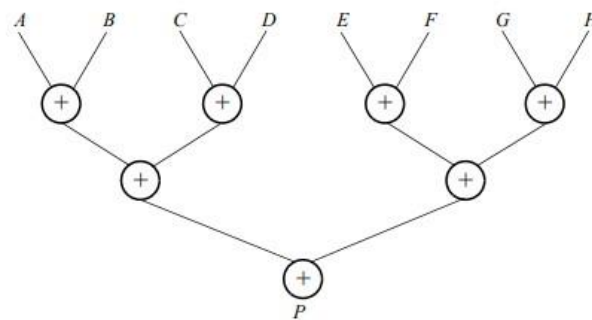


Figure 7: An example of adding 8 numbers using an adder tree.

1. Tuliskan codingan dibawah ini

```
1 // Implements an unsigned multiplier P = A x B.
2 // inputs: SW7-0, SW9 select A, SW8 selects B, KEY[0] is reset, KEY[1] is clock
3 // outputs: HEX3-0 shows the product P
4 module part5 (SW, KEY, LEDR, HEX3, HEX2, HEX1, HEX0);
5     input [9:0] SW;
6     input [1:0] KEY;
7     output [9:0] LEDR;
8     output [0:6] HEX3, HEX2, HEX1, HEX0;
9
10    wire Resetn, Clock;
11    reg [7:0] A, B;
12    reg [15:0] P_reg;
13    wire [15:0] PP0, PP1, PP2, PP3, PP4, PP5, PP6;
14    wire [15:0] P;
15
16    assign Resetn = KEY[0];
17    assign Clock = KEY[1];
18    always @(posedge Clock)
19    begin
20        if (~Resetn)
21        begin
22            A <= 8'b0; B <= 8'b0; P_reg <= 16'b0;
23        end
24        else
25        begin
26            if (SW[9]) A <= SW[7:0];
27            if (SW[8]) B <= SW[7:0];
28            P_reg <= P;
29        end
30    end
31
32    assign LEDR[7:0] = SW[9] ? A : (SW[8] ? B : 8'b0);
```

```

33 assign LEDR[9:8] = SW[9:8];
34
35 assign PP0 = { 8'd0, A & {8{B[0]}} } + { 7'd0, A & {8{B[1]}} , 1'd0 };
36 assign PP1 = { 6'd0, A & {8{B[2]}} , 2'd0 } + { 5'd0, A & {8{B[3]}} , 3'd0 };
37 assign PP2 = { 4'd0, A & {8{B[4]}} , 4'd0 } + { 3'd0, A & {8{B[5]}} , 5'd0 };
38 assign PP3 = { 2'd0, A & {8{B[6]}} , 6'd0 } + { 1'd0, A & {8{B[7]}} , 7'd0 };
39 assign PP4 = PP0 + PP1;
40 assign PP5 = PP2 + PP3;
41 assign P = PP4 + PP5;
42
43 // drive the displays through 7-seg decoders
44 hex7seg digit_3 (P_reg[15:12], HEX3);
45 hex7seg digit_2 (P_reg[11:8], HEX2);
46 hex7seg digit_1 (P_reg[7:4], HEX1);
47 hex7seg digit_0 (P_reg[3:0], HEX0);
48 endmodule
49
50 module hex7seg (hex, display);
51 input [3:0] hex;
52 output [0:6] display;
53
54 reg [0:6] display;
55
56 /*
57 *      0
58 *      ---
59 *      |   |
60 *      5|   |1
61 *      | 6 |
62 *      ---
63 *      |   |
64 *      4|   |2
65
66 *      |   |
67 *      ---
68 *      3
69 */
69 always @ (hex)
70     case (hex)
71         4'h0: display = 7'b0000001;
72         4'h1: display = 7'b1001111;
73         4'h2: display = 7'b0010010;
74         4'h3: display = 7'b0000110;
75         4'h4: display = 7'b1001100;
76         4'h5: display = 7'b0100100;
77         4'h6: display = 7'b0100000;
78         4'h7: display = 7'b0001111;
79         4'h8: display = 7'b0000000;
80         4'h9: display = 7'b0001100;
81         4'hA: display = 7'b0001000;
82         4'hb: display = 7'b1100000;
83         4'hc: display = 7'b0110001;
84         4'hd: display = 7'b1000010;
85         4'he: display = 7'b0110000;
86         4'hF: display = 7'b0111000;
87     endcase
88 endmodule
89

```

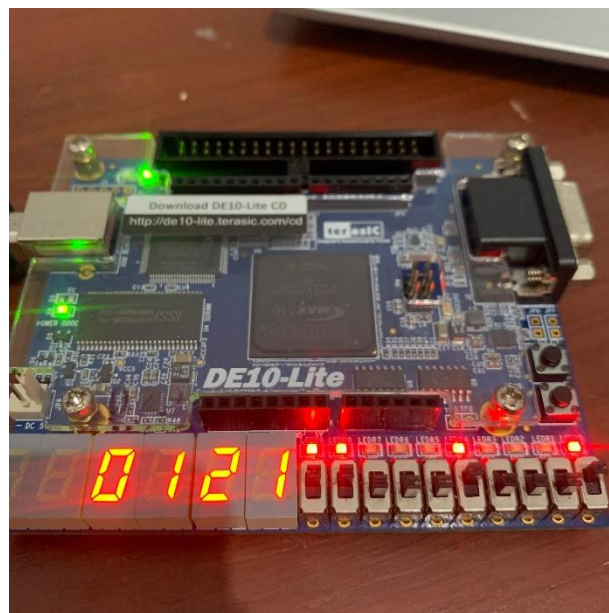
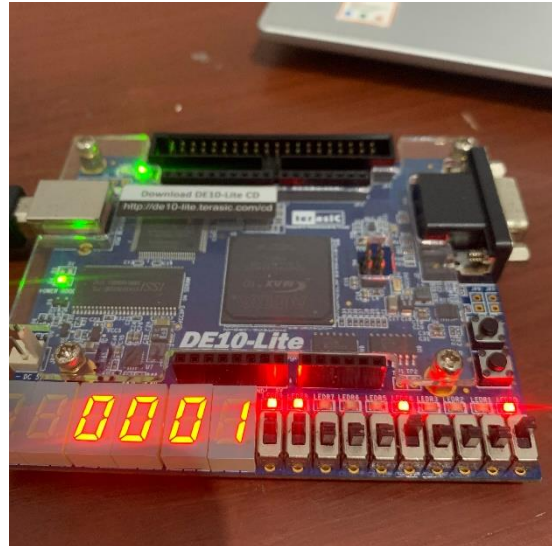
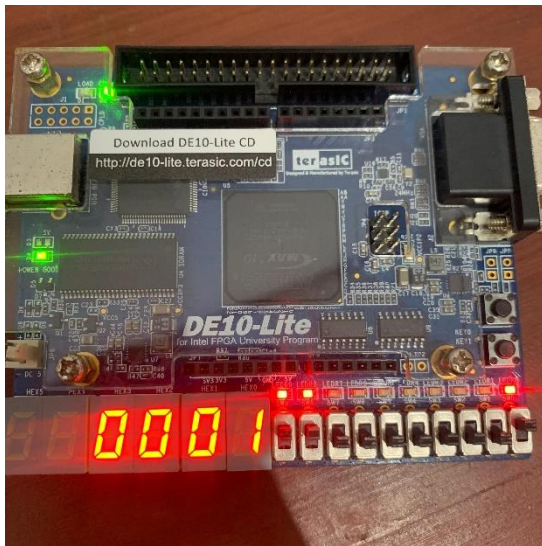
2. Lalu Compile dan lanjutkan masukan Pin Planner

- Pin Planner

Named: * Edit:												
Node Name	Direction	Location	I/O Bank	VREF Group	itter Locatior	I/O Standard	Reserved	rrrent Streng	Slew Rate	ifferential Pai	ict Preservati	
out HEX0[0]	Output	PIN_C14	7	B7_NO	PIN_C14	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX0[1]	Output	PIN_E15	7	B7_NO	PIN_E15	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX0[2]	Output	PIN_C15	7	B7_NO	PIN_C15	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX0[3]	Output	PIN_C16	7	B7_NO	PIN_C16	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX0[4]	Output	PIN_E16	7	B7_NO	PIN_E16	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX0[5]	Output	PIN_D17	7	B7_NO	PIN_D17	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX0[6]	Output	PIN_C17	7	B7_NO	PIN_C17	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX1[0]	Output	PIN_C18	7	B7_NO	PIN_C18	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX1[1]	Output	PIN_D18	6	B6_NO	PIN_D18	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX1[2]	Output	PIN_E18	6	B6_NO	PIN_E18	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX1[3]	Output	PIN_B16	7	B7_NO	PIN_B16	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX1[4]	Output	PIN_A17	7	B7_NO	PIN_A17	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX1[5]	Output	PIN_A18	7	B7_NO	PIN_A18	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX1[6]	Output	PIN_B17	7	B7_NO	PIN_B17	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX2[0]	Output	PIN_B20	6	B6_NO	PIN_B20	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX2[1]	Output	PIN_A20	7	B7_NO	PIN_A20	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX2[2]	Output	PIN_B19	7	B7_NO	PIN_B19	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX2[3]	Output	PIN_A21	6	B6_NO	PIN_A21	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX2[4]	Output	PIN_B21	6	B6_NO	PIN_B21	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX2[5]	Output	PIN_C22	6	B6_NO	PIN_C22	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX2[6]	Output	PIN_B22	6	B6_NO	PIN_B22	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX3[0]	Output	PIN_F21	6	B6_NO	PIN_F21	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX3[1]	Output	PIN_E22	6	B6_NO	PIN_E22	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX3[2]	Output	PIN_E21	6	B6_NO	PIN_E21	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX3[3]	Output	PIN_C19	7	B7_NO	PIN_C19	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX3[4]	Output	PIN_C20	6	B6_NO	PIN_C20	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX3[5]	Output	PIN_D19	6	B6_NO	PIN_D19	3.3-V LVTTTL		8mA (default)	2 (default)			
out HEX3[6]	Output	PIN_E17	6	B6_NO	PIN_E17	3.3-V LVTTTL		8mA (default)	2 (default)			
in KEY[1]	Input	PIN_A7	7	B7_NO	PIN_A7	3.3 V ...rigger		8mA (default)				
in KEY[0]	Input	PIN_B8	7	B7_NO	PIN_B8	3.3 V ...rigger		8mA (default)				
out LEDR[9]	Output	PIN_B11	7	B7_NO	PIN_B11	3.3-V LVTTTL		8mA (default)	2 (default)			
out LEDR[8]	Output	PIN_A11	7	B7_NO	PIN_A11	3.3-V LVTTTL		8mA (default)	2 (default)			
out LEDR[7]	Output	PIN_D14	7	B7_NO	PIN_D14	3.3-V LVTTTL		8mA (default)	2 (default)			
out LEDR[6]	Output	PIN_E14	7	B7_NO	PIN_E14	3.3-V LVTTTL		8mA (default)	2 (default)			
out LEDR[5]	Output	PIN_C13	7	B7_NO	PIN_C13	3.3-V LVTTTL		8mA (default)	2 (default)			
out LEDR[4]	Output	PIN_D13	7	B7_NO	PIN_D13	3.3-V LVTTTL		8mA (default)	2 (default)			
out LEDR[3]	Output	PIN_B10	7	B7_NO	PIN_B10	3.3-V LVTTTL		8mA (default)	2 (default)			
out LEDR[2]	Output	PIN_A10	7	B7_NO	PIN_A10	3.3-V LVTTTL		8mA (default)	2 (default)			
out LEDR[1]	Output	PIN_A9	7	B7_NO	PIN_A9	3.3-V LVTTTL		8mA (default)	2 (default)			
out LEDR[0]	Output	PIN_A8	7	B7_NO	PIN_A8	3.3-V LVTTTL		8mA (default)	2 (default)			
in SW[9]	Input	PIN_F15	7	B7_NO	PIN_F15	3.3-V LVTTTL		8mA (default)				
in SW[8]	Input	PIN_B14	7	B7_NO	PIN_B14	3.3-V LVTTTL		8mA (default)				
in SW[7]	Input	PIN_A14	7	B7_NO	PIN_A14	3.3-V LVTTTL		8mA (default)				
in SW[6]	Input	PIN_A13	7	B7_NO	PIN_A13	3.3-V LVTTTL		8mA (default)				
in SW[5]	Input	PIN_B12	7	B7_NO	PIN_B12	3.3-V LVTTTL		8mA (default)				
in SW[4]	Input	PIN_A12	7	B7_NO	PIN_A12	3.3-V LVTTTL		8mA (default)				
in SW[3]	Input	PIN_C12	7	B7_NO	PIN_C12	3.3-V LVTTTL		8mA (default)				
in SW[2]	Input	PIN_D12	7	B7_NO	PIN_D12	3.3-V LVTTTL		8mA (default)				
in SW[1]	Input	PIN_C11	7	B7_NO	PIN_C11	3.3-V LVTTTL		8mA (default)				
in SW[0]	Input	PIN_C10	7	B7_NO	PIN_C10	3.3-V LVTTTL		8mA (default)				

Modul Praktikum Sisdig

- Hasil di FPGA



~ Selamat Mengerjakan teman teman ~