

GEBZE

TEKNİK ÜNİVERSİTESİ



GEBZE TEKNİK ÜNİVERSİTESİ
ELEKTRONİK MÜHENDİSLİĞİ

ELM 235

LOJİK DEVRE TASARIM LABORATUVARI

LAB 0x1 Deney Raporu

Birleşimli Mantık Devreleri

Hazırlayanlar
1801022024 Muhammed Cemal Eryiğit
1801022077 Burak Kamil Çiftci

Labın Amacı

- Donanım tanımlama dilleri ile çözücü (decoder) tasarlanması,
- Sentezleyici araçları kullanarak, tanımlanan devreleri FPGA için sentezlemek,
- Structural tasarım prensiplerini kavramak ve uygulamak,
- Simülasyon araçları kullanarak devre çalışmasını doğrulamak.

Giriş

Bu deney kapsamında ELM235 laboratuvar dersinde MODELSİM ve QUARTUS PRIME Programları kullanıldı. Programı kullanmayı öğretmeyi amaçlayan giriş niteliğinde kodlar yazıldı. Basit devre kombinasyonları incelendi. Devreyi Quartus programıyla sentezleyerek ne kadar yer kapladığını, sentezlenen devrenin RTL ve eşleştirme ardı devre şemaları çıkarıldı.

Problem - 1 Basit bir devre tasarımı ve simülasyonu

$$Y = AB + A\bar{C} + \bar{B}C$$

1.1 Teorik Araştırma

Bu problem için teorik bir araştırma bulunmamaktadır.

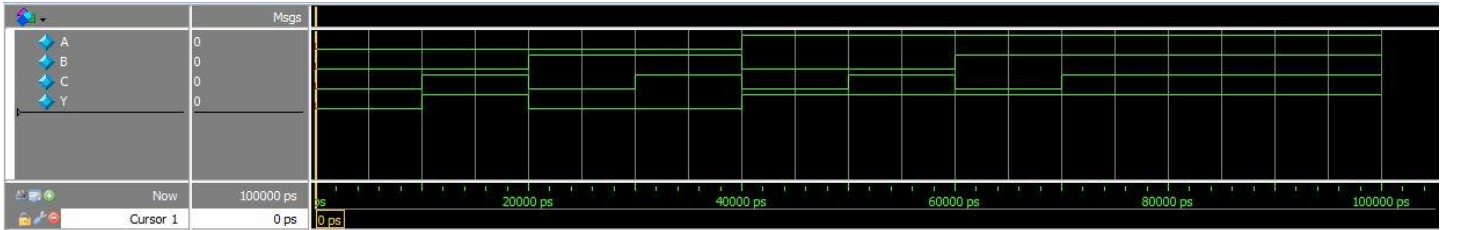
1.2 Deneyin Yapılışı

A) Denklem 1 deki Boolean denklemini HDL ile tasarlayın.

```
/* Hazırlayanlar:
* M.Cemal Eryigit
* Burak Kamil Ciftci
* ELM235 2021 Bahar Lab1 - Problem 1
*/
module lab1_g24_p1 (
input logic A, B, C,
output logic Y
);
assign Y = A & B | A & ~C | ~B & C;
endmodule
```

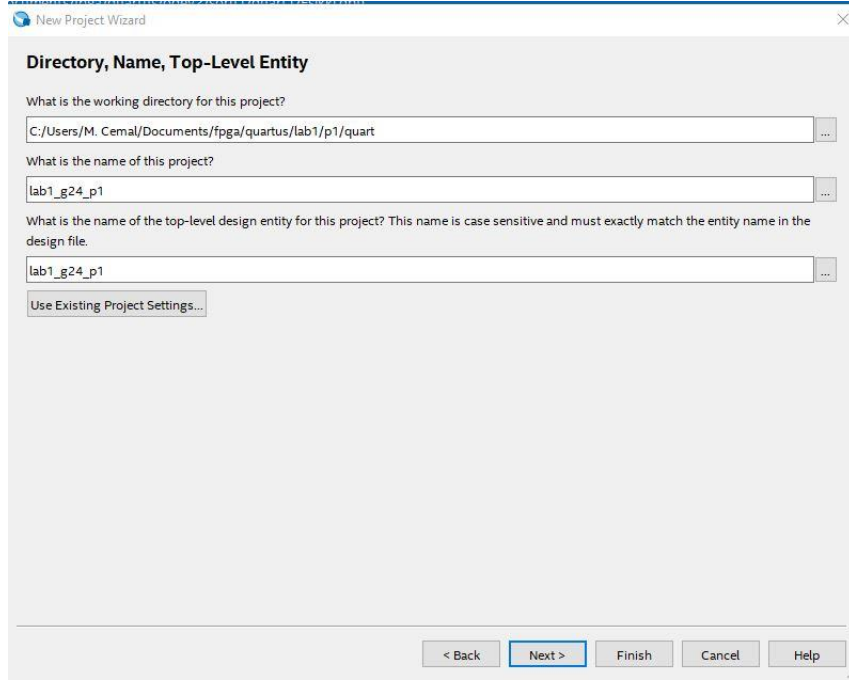
B) Testbench oluşturarak, devrenin bütün girişlere karşı nasıl davrandığını gözlemleyin

```
/* Hazirlayanlar:
* M.Cemal Eryigit
* Burak Kamil Ciftci
* ELM235 2021 Bahar Lab1 - Problem 1 Testbench
*/
`timescale 1ns/1ps
module tb_lab1_g24_p1 ();
logic A, B, C;
logic Y;
lab1_g24_p1 dut0(.A(A), .B(B), .C(C), .Y(Y));
initial begin
A=0; B=0; C=0; #10; // wait 10 ns
A=0; B=0; C=1; #10; // wait 10 ns
A=0; B=1; C=0; #10; // wait 10 ns
A=0; B=1; C=1; #10; // wait 10 ns
A=1; B=0; C=0; #10; // wait 10 ns
A=1; B=0; C=1; #10; // wait 10 ns
A=1; B=1; C=0; #10; // wait 10 ns
A=1; B=1; C=1; #10; // wait 10 ns
#20; // wait 20 ns after completion
$stop; // stop the simulation
end
endmodule
```



Şekil 1 Dalga görüntüsü

C) Devreyi Quartus'da sentezleyerek devrenin ne kadar yer kapladığını (resource utilization report), sentezlenen RTL ve eşleştirme ardı devre şemalarını ekleyerek yorumlayınız.



The screenshot shows the 'New Project Wizard' window, specifically the 'Directory, Name, Top-Level Entity' step. It contains three text input fields for project configuration. The first field is for the working directory, the second for the project name, and the third for the top-level design entity name. All three fields are filled with the same text: 'C:/Users/M. Cemal/Documents/fpga/quartus/lab1/p1/quart', 'lab1_g24_p1', and 'lab1_g24_p1' respectively. There is a 'Use Existing Project Settings...' button below the third field. At the bottom, there are navigation buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Directory, Name, Top-Level Entity

What is the working directory for this project?

C:/Users/M. Cemal/Documents/fpga/quartus/lab1/p1/quart

What is the name of this project?

lab1_g24_p1

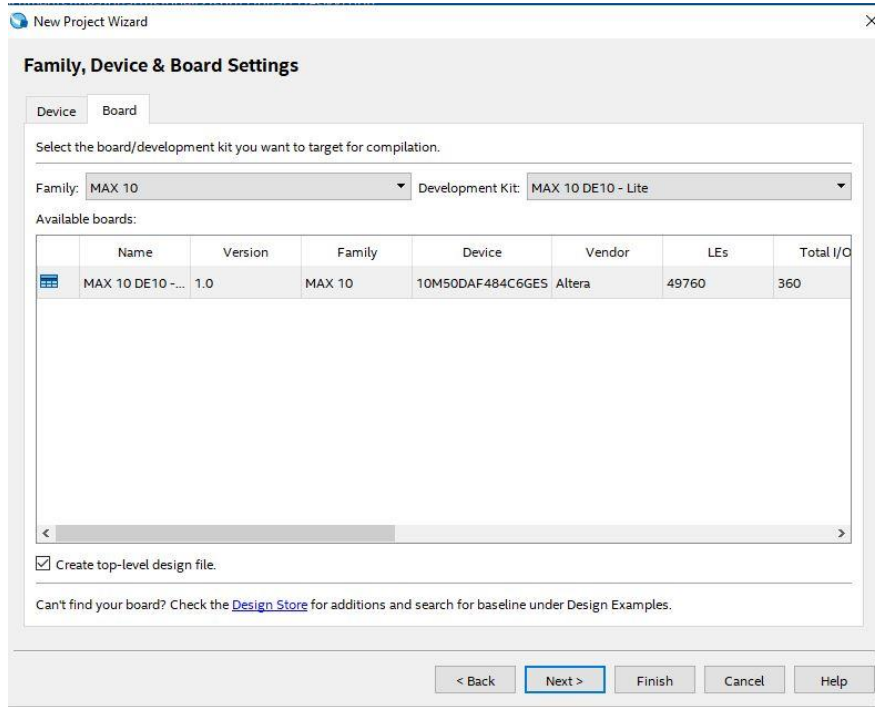
What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

lab1_g24_p1

Use Existing Project Settings...

< Back Next > Finish Cancel Help

Şekil 2 Proje için dosya konumu seçimi



The screenshot shows the 'New Project Wizard' window, specifically the 'Family, Device & Board Settings' step. It has two tabs: 'Device' and 'Board'. The 'Board' tab is selected. The screen prompts the user to 'Select the board/development kit you want to target for compilation.' Below this, there are two dropdown menus: 'Family' set to 'MAX 10' and 'Development Kit' set to 'MAX 10 DE10 - Lite'. A table titled 'Available boards:' lists the available boards. The table has columns: Name, Version, Family, Device, Vendor, LEs, and Total I/O. The first row shows 'MAX 10 DE10 - Lite' with version 1.0, family MAX 10, device 10M50DAF484C6GES, vendor Altera, 49760 LEs, and 360 Total I/O. Below the table, there is a checkbox 'Create top-level design file.' which is checked. At the bottom, there is a link 'Design Store' and a note 'Can't find your board? Check the Design Store for additions and search for baseline under Design Examples.' At the bottom of the window, there are navigation buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Family, Device & Board Settings

Device Board

Select the board/development kit you want to target for compilation.

Family: MAX 10 Development Kit: MAX 10 DE10 - Lite

Available boards:

Name	Version	Family	Device	Vendor	LEs	Total I/O
MAX 10 DE10 - Lite	1.0	MAX 10	10M50DAF484C6GES	Altera	49760	360

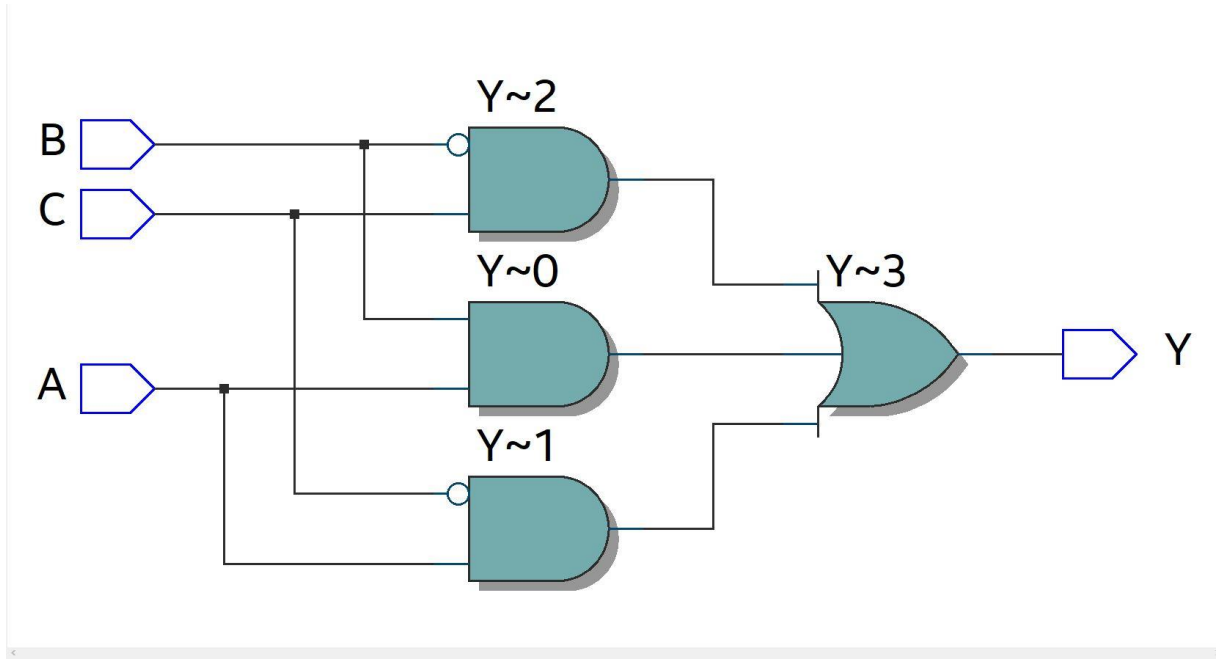
☒ Create top-level design file.

Can't find your board? Check the [Design Store](#) for additions and search for baseline under Design Examples.

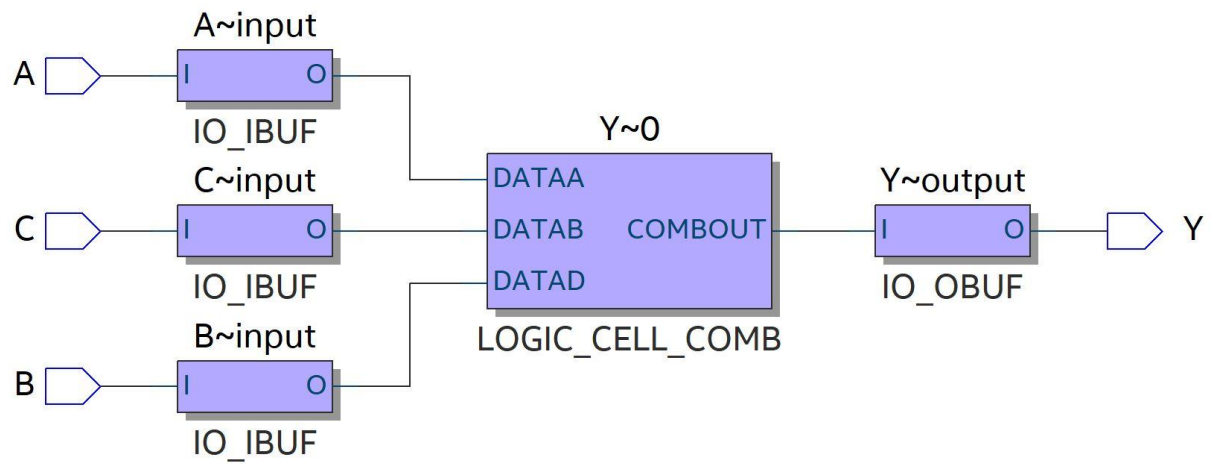
< Back Next > Finish Cancel Help

Şekil 3 Proje için board seçimi

Proje oluşturma aşamasında Family Device&Board Settings kısmında Board sekmesinde Development Kit 'MAX10 DE10 Lite' seçildi.



Şekil 4 Lojik devrenin RTL devre şeması



Şekil 5 Lojik devrenin eşleştirme ardı devre şeması

Şekil 4 ve Şekil 5 de Netlist Viewer sentezi sonrası RTL Viewer ve Technology Map Viewer (Eşleştirme ardı şeması) gösterildi.

Flow Summary

<<Filter>>

Flow Status	Successful - Mon Jun 28 11:31:17 2021
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	lab1_g24_p1
Top-level Entity Name	lab1_g24_p1
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	2 / 49,760 (< 1 %)
Total registers	0
Total pins	4 / 360 (1 %)
Total virtual pins	0
Total memory bits	0 / 1,677,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 288 (0 %)
Total PLLs	0 / 4 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 2 (0 %)

Compilation Report - lab1_g24_p1

Analysis & Synthesis Resource Usage Summary

<<Filter>>

	Resource	Usage
1	Estimated Total logic elements	1
2		
3	Total combinational functions	1
4	Logic element usage by number of LUT inputs	
1	-- 4 input functions	0
2	-- 3 input functions	1
3	-- <=2 input functions	0
5		
6	Logic elements by mode	
1	-- normal mode	1
2	-- arithmetic mode	0
7		
8	Total registers	0
1	-- Dedicated logic registers	0
2	-- I/O registers	0
9		
10	I/O pins	4
11		
12	Embedded Multiplier 9-bit elements	0
13		
14	Maximum fan-out node	Y~0
15	Maximum fan-out	1
16	Total fan-out	8
17	Average fan-out	0.89

Analysis & Synthesis Resource Utilization by Entity

<<Filter>>

Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers	Memory Bits	UFM Blocks	DSP Elements	DSP 9x9	DSP 18x18	Pins	Virtual Pins	ADC blocks	Full Hierarchy Name	Entity Name	Library Name
1 lab1_g24_p1	1 (1)	0 (0)	0	0	0	0	0	4	0	0	lab1_g24_p1	lab1_g24_p1	work

Şekil 6 Analiz ve Kaynak Sentez Kullanım Özeti

1.3 Deneyin Yorumu

Oluşturulmak istenilen devre için MODELSİM HDL dilinde kodlandı. Daha sonra bu devre test edilmek için testbench kodu yazıldı. Girişlerin alabileceği bütün değerler için devre teste sokuldu. Analiz ve sentez özetinde devrenin gerçekleşebilmesi için 2 lojik eleman ve 4 pin gerekmektedir. FPGA üzerinde bulunan toplam pinlerin %1 inden daha az miktarda pin kullanıldı.

Problem 2 – Adder Tasarımı

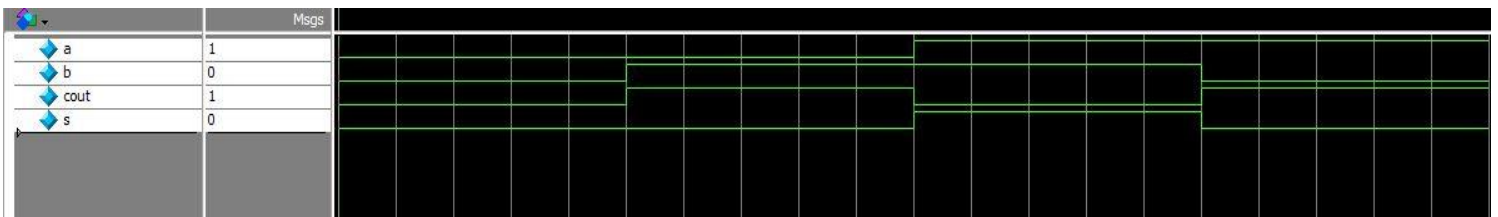
1.1 Teorik Araştırma

Bu problem için teorik bir araştırma bulunmamaktadır.

1.2 Deneyin Yapılışı

A) Half-adder tasarlayın.

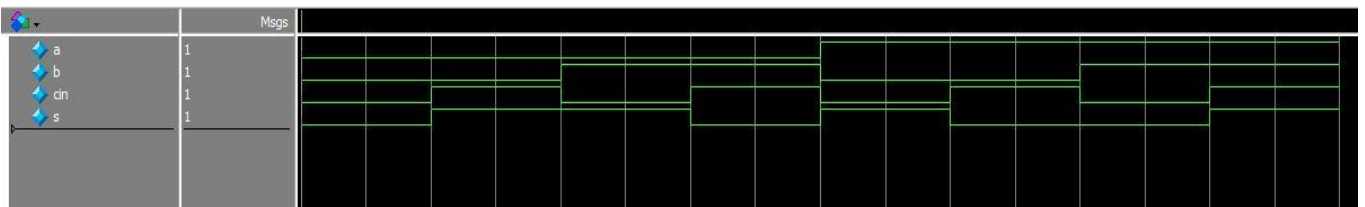
```
/* Hazirlayanlar:  
* M.Cemal Eryigit  
* Burak Kamil Ciftci  
* ELM235 2021 Bahar Lab1 - Problem 2  
*/  
module lab1_g24_p2a (  
input logic a , b  
output logic sum,carry  
);  
assign sum=a^b;  
assign carry=a&b;  
endmodule
```



Şekil 7 dalga görüntüsü

B) A da tasarladığınız half-adderları modül olarak kullanarak full-adder tasarlayın

```
module halfadder (  
  input logic a , b  
  output logic sum,carry  
);  
  assign sum=a^b;  
  assign carry=a&b;  
endmodule  
  
module lab1_g24_p2a(  
  input logic a,b  
  input logic cin,sum,cout  
);  
  logic s1,c1,c2;  
  halfadder h1 (.a(a), .b(b), .sum(s1), .carry(c1));  
  halfadder h2(.a(s1), .b(cin), .sum(sum), .carry(c2));  
  assign cout = c1 | c2;  
endmodule
```



Şekil 8 dalga görüntüsü

C) B de tasarladığınız full-adderları modül olarak kullanarak 4-bitlik ripple carry adder tasarlayın.


```

/* Hazirlayanlar:
* M.Cemal Eryigit
* Burak Kamil Ciftci
* ELM235 2021 Bahar Lab1 - Problem 2
*/

module halfadder (
input logic a ,b,
output logic sum, carry
);
assign sum=a^b;
assign carry=a&b;
endmodule

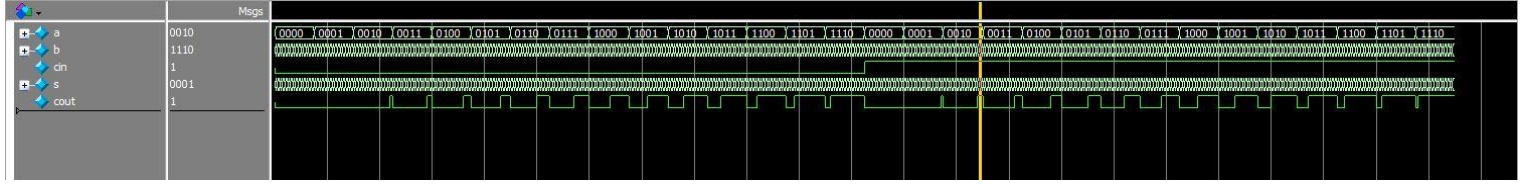
module fulladder(
input logic a,b,cin,
output logic sum,cout
);
logic s1,c1,c2;
halfadder h1 (.a(a), .b(b), .sum(s1), .carry(c1));
halfadder h2(.a(s1), .b(cin), .sum(sum), .carry(c2));
assign cout = c1 | c2;
endmodule

module lab1_g24_p2(
input logic [3:0] a,b,cin,
output logic [3:0]sum2 ,cout
);
logic v1,v2,v3,v4;
fulladder du1(.a(a[0]),.b(b[0]),.cin(cin),.sum(sum2[0]),.cout(v1));
fulladder du2(.a(a[1]),.b(b[1]),.cin(v1),.sum(sum2[1]),.cout(v2));
fulladder du3(.a(a[2]),.b(b[2]),.cin(v2),.sum(sum2[2]),.cout(v3));
fulladder du4(.a(a[3]),.b(b[3]),.cin(v3),.sum(sum2[3]),.cout(cout));
endmodule

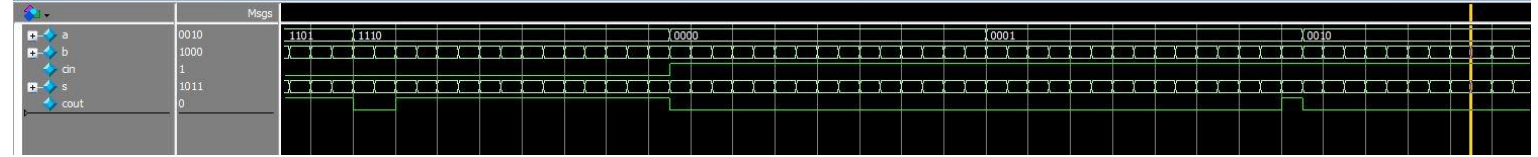
```

D) Testbench oluşturarak, devrenin bütün girişlere karşı nasıl davrandığını ve sonuçların doğruluğunu gözlemleyin.

```
`timescale 1ns/1ps
module tb_lab1_g24_p2 ();
    logic [3:0] a, b;
    logic cin;
    logic [3:0] s;
    logic cout;
    lab1_g24_p2 dut0(.a(a), .b(b), .cin(cin), .cout(cout),
.sum2(s));
    initial
    begin
        for (int c=0; c<2; c++) begin
            cin = c;
            for (int i=0; i<2**4-1; i++) begin
                for (int j=0; j<2**4-1; j++) begin
                    a = i; b = j;
                    #10;
                end
            end
        end
        end
        $stop;
    end
endmodule
```




Şekil 9 Dalga görüntüsü (örnek bir nokta gösterilmiştir)



Şekil 10 Dalga görüntüsü (örnek bir alan gösterilmiştir)

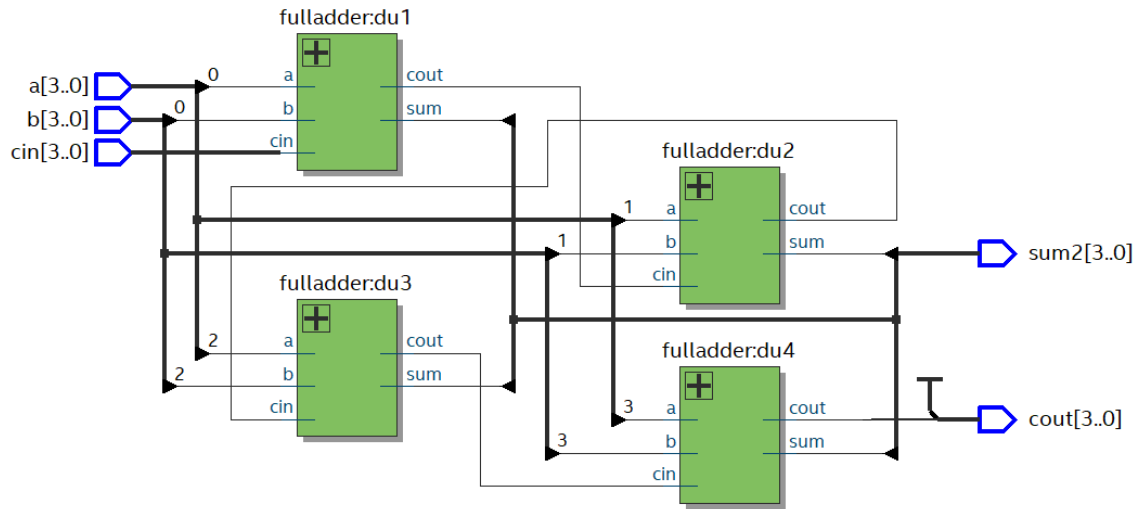
E) Devreyi Quartus'da sentezleyerek devrenin ne kadar yer kapladığını, sentezlenen RTL ve eşleştirme ardı devre şemalarını ekleyerek yorumlayınız.

Flow Summary					
 <<Filter>>					
Flow Status		Successful - Tue Jun 29 23:37:25 2021			
Quartus Prime Version		20.1.1 Build 720 11/11/2020 SJ Lite Edition			
Revision Name		lab1_g24_p2			
Top-level Entity Name		lab1_g24_p2			
Family		MAX 10			
Device		10M08DAF484C8G			
Timing Models		Final			
Total logic elements		9 / 8,064 (< 1 %)			
Total registers		0			
Total pins		20 / 250 (8 %)			
Total virtual pins		0			
Total memory bits		0 / 387,072 (0 %)			
Embedded Multiplier 9-bit elements		0 / 48 (0 %)			
Total PLLs		0 / 2 (0 %)			
UFM blocks		0 / 1 (0 %)			
ADC blocks		0 / 1 (0 %)			

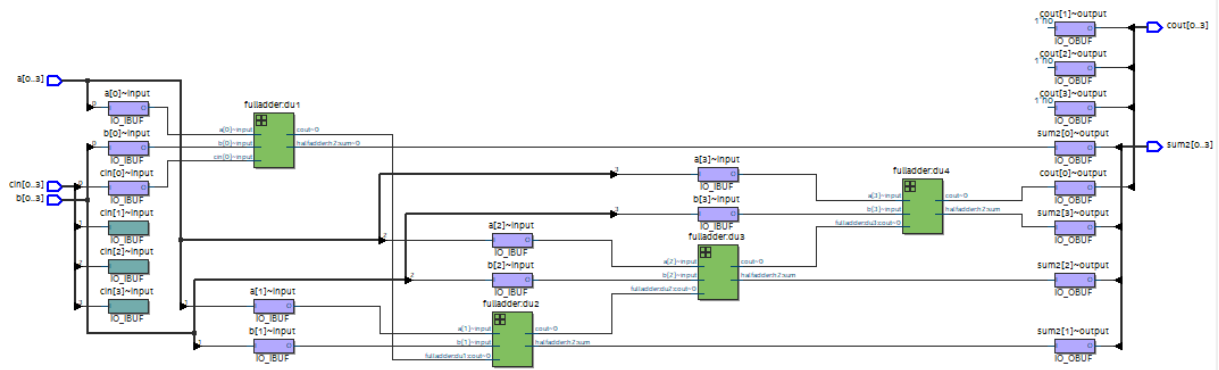
Resource		Usage
1	Estimated Total logic elements	8
2		
3	Total combinational functions	8
4	Logic element usage by number of LUT inputs	
1	-- 4 input functions	0
2	-- 3 input functions	8
3	-- <=2 input functions	0
5		
6	Logic elements by mode	
1	-- normal mode	8
2	-- arithmetic mode	0
7		
8	Total registers	0
1	-- Dedicated logic registers	0
2	-- I/O registers	0
9		
10	I/O pins	20
11		
12	Embedded Multiplier 9-bit elements	0
13		
14	Maximum fan-out node	fulladder:0
15	Maximum fan-out	2
16	Total fan-out	49
17	Average fan-out	1.02

	Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers	Memory Bits	UFM Blocks	DSP Elements	DSP 9x9	DSP 18x18	Pins	Virtual Pins	ADC blocks	Full Hierarchy Name	Entity Name	Library Name
1	lab1_g24_p2	8 (0)	0 (0)	0	0	0	0	0	20	0	0	lab1_g24_p2	lab1_g24_p2	work
1	fulladder:du1	2 (1)	0 (0)	0	0	0	0	0	0	0	0	lab1_g24_p2_fulladder:du1	fulladder	work
1	halfadder:h2	1 (1)	0 (0)	0	0	0	0	0	0	0	0	lab1_g24_p2_halfadder:h2	halfadder	work
2	fulladder:du2	2 (1)	0 (0)	0	0	0	0	0	0	0	0	lab1_g24_p2_fulladder:du2	fulladder	work
1	halfadder:h2	1 (1)	0 (0)	0	0	0	0	0	0	0	0	lab1_g24_p2_halfadder:h2	halfadder	work
3	fulladder:du3	2 (1)	0 (0)	0	0	0	0	0	0	0	0	lab1_g24_p2_fulladder:du3	fulladder	work
1	halfadder:h2	1 (1)	0 (0)	0	0	0	0	0	0	0	0	lab1_g24_p2_halfadder:h2	halfadder	work
4	fulladder:du4	2 (1)	0 (0)	0	0	0	0	0	0	0	0	lab1_g24_p2_fulladder:du4	fulladder	work
1	halfadder:h2	1 (1)	0 (0)	0	0	0	0	0	0	0	0	lab1_g24_p2_halfadder:h2	halfadder	work

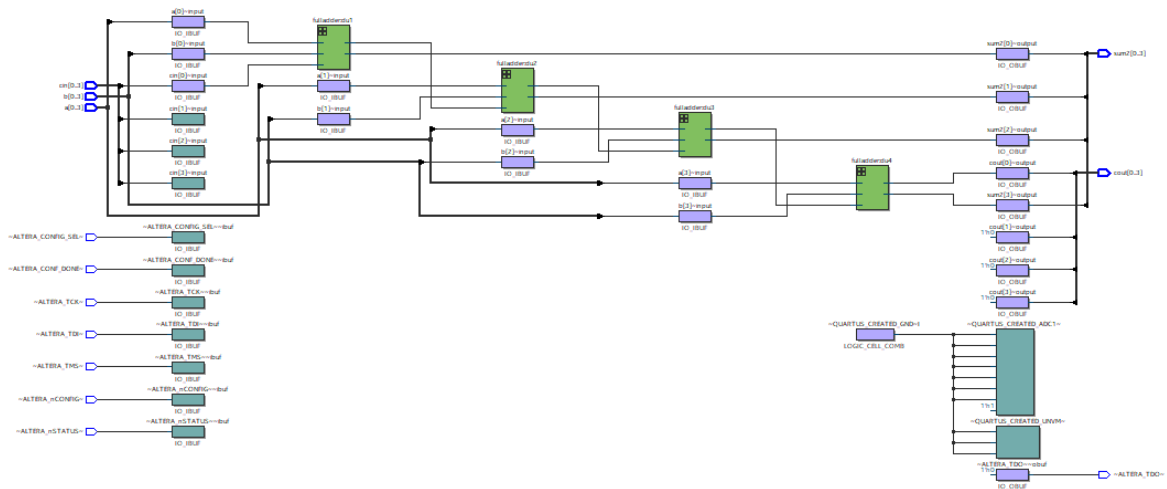
Şekil 11 Analiz ve Sentez Özetleri



Şekil 12 RTL Devresi



Şekil 13.a Techmap Şeması



Şekil 13.b Techmap Şeması

Problem - 3 Çözücü tasarımı

1.1 Teorik Araştırma

Bu problem için teorik bir araştırma bulunmamaktadır.

1.2 Deneyin Yapılışı

A) Şekil 1'deki 7BE elemanının bölmeleri Tablo 1 de verilen gösterimlerde yakmak için 4 girişli 7 çıkışlı bir *çözücü* (decoder) devresinin doğruluk tablosunu oluşturunuz ve boolean expressionlarını çıkarınız. İsterseniz K-Map ile sadeleştirip öyle devam edebilirsiniz.

Hexadecimal encodings for displaying the digits 0 to F ^{[12][13]}										
Digit	Display	gfedcba	abcdefg	a	b	c	d	e	f	g
0	0	0x3F	0x7E	on	on	on	on	on	on	
1	1	0x06	0x30		on	on				
2	2	0x5B	0x6D	on	on		on	on		on
3	3	0x4F	0x79	on	on	on	on			on
4	4	0x66	0x33		on	on			on	on
5	5	0x6D	0x5B	on		on	on		on	on
6	6	0x7D	0x5F	on		on	on	on	on	on
7	7	0x07	0x70	on	on	on				
8	8	0x7F	0x7F	on	on	on	on	on	on	on
9	9	0x6F	0x7B	on	on	on	on		on	on
A	A	0x77	0x77	on	on	on		on	on	on
b	b	0x7C	0x1F			on	on	on	on	on
C	C	0x39	0x4E	on			on	on	on	
d	d	0x5E	0x3D		on	on	on	on		on
E	E	0x79	0x4F	on			on	on	on	on
F	F	0x71	0x47	on				on	on	on

Şekil 14 Örnek Truth Table

X3	X2	X1	X0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	1	0	0	1	1	0	0	0	0
0	0	1	1	1	1	1	0	0	0	1
0	1	1	1	0	1	1	0	0	0	1
1	0	0	1	0	0	1	0	0	1	0
1	0	1	1	0	0	0	0	1	1	0
1	1	1	1	0	1	0	0	1	0	0

$$\begin{aligned}a &\rightarrow \overline{X_3}X_2 \\b &\rightarrow \overline{X_3}\overline{X_2}\overline{X_0} + X_2X_1X_0 \\c &\rightarrow \overline{X_3}\overline{X_2}\overline{X_0} + \overline{X_3}X_2X_1X_0 + X_3\overline{X_2}\overline{X_1}X_0 \\d &\rightarrow \overline{X_3}\overline{X_2}\overline{X_1}\overline{X_0} \\e &\rightarrow \overline{X_3}\overline{X_2}\overline{X_1}\overline{X_0} + X_3X_1X_0 \\f &\rightarrow \overline{X_3}\overline{X_2}\overline{X_1}\overline{X_0} + X_3\overline{X_2}X_0 \\g &\rightarrow \overline{X_3}X_1X_0\end{aligned}$$

Şekil 15 Truth Table for ELC235

B) Bu devreyi HDL ile tasarlayın. (Her bir 7BE çıkışı için bir assign olacaktır.)

```
/* Hazırlayanlar:
 * M.Cemal Eryigit
 * Burak Kamil Ciftci
 * ELM235 2021 Bahar Lab2 - Problem 1
 */
module lab1_g24_p3 (
input logic x3, x2, x1, x0,
output logic a, b, c, d, e, f, g
);
assign a = ~x3 & x2;
assign b = ~x3 & ~x2 & ~x0 | x2 & x1 & x0;
assign c = ~x3 & ~x2 & ~x0 | ~x3 & x2 & x1 & x0;
assign d = ~x3 & ~x2 & ~x1 & x0;
assign e = ~x3 & ~x2 & ~x1 & ~x0 | x3 & x1 & x0;
assign f = ~x3 & ~x2 & ~x1 & ~x0 | x3 & ~x2 & x0;
assign g = ~x3 & x1 & x0;

endmodule
```

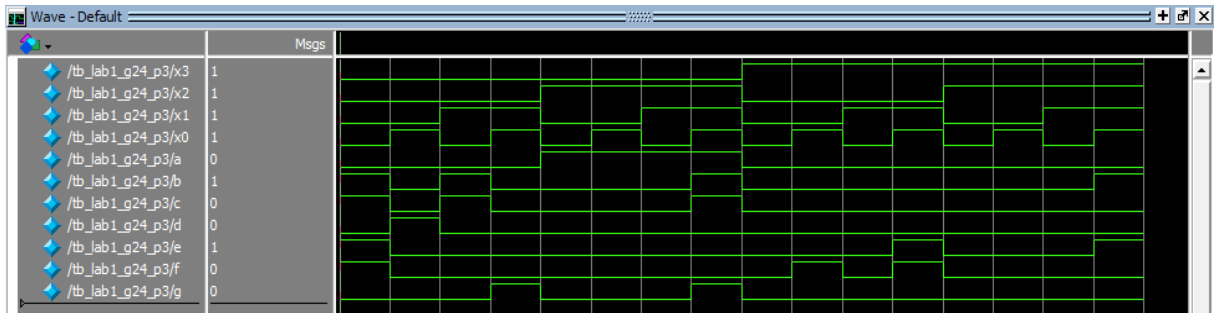
```
/* Hazırlayanlar:
 * M.Cemal Eryigit
 * Burak Kamil Ciftci
 * ELM235 2021 Bahar Lab2 - Problem 1
 */
`timescale 1ns/1ps
module tb_lab1_g24_p3 ();
logic x3, x2, x1, x0;
logic a, b, c, d, e, f, g;
lab1_g24_p3 dut0(x3, x2, x1, x0, a, b, c, d, e, f, g);
initial begin
x3 = 0; x2= 0; x1= 0; x0= 0; #10;
x0= 1; #10;
x1= 1; x0= 0; #10;
x0= 1; #10;
x2= 1; x1= 0; x0= 0; #10;
x0= 1; #10;
x1= 1; x0= 0; #10;
x0= 1; #10;
x3 = 1; x2= 0; x1= 0; x0= 0; #10;
x0= 1; #10;
x1= 1; x0= 0; #10;
x0= 1; #10;
x2= 1; x1= 0; x0= 0; #10;
```

```

x0= 1; #10;
x1= 1; x0= 0; #10;
x0= 1; #10;
$stop;
end
endmodule

```

C) Testbench oluşturarak, 16 farklı giriş kombinasyonuna göre çıkış dalga şeklini gözlemleyiniz. Tabloda verilen girişlerde doğru çıkışların olduğunu doğrulayınız.



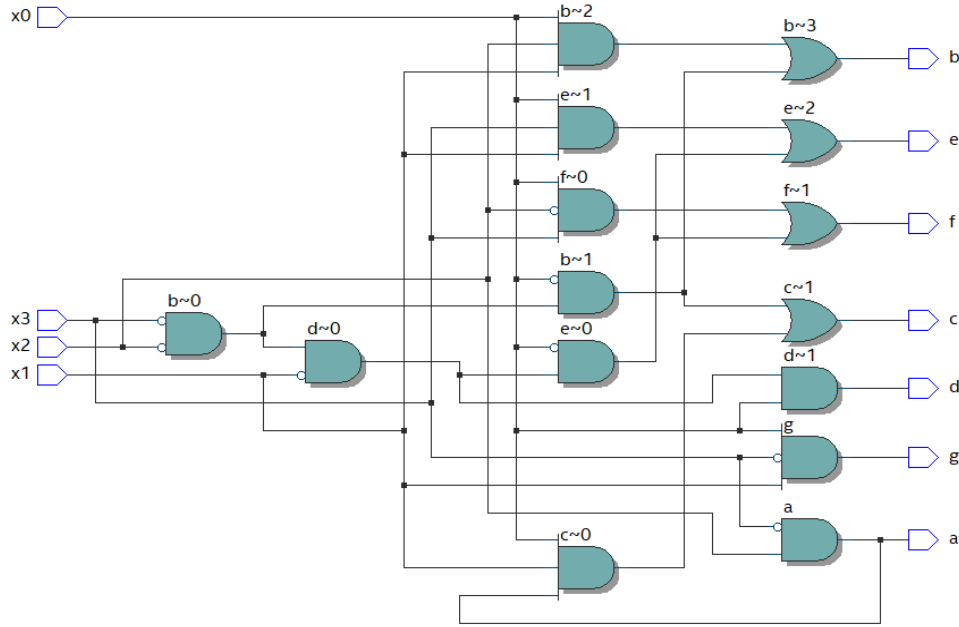
Şekil 16 Dalga görüntüsü

D) Devreyi Quartus'da sentezleyerek devrenin ne kadar yer kapladığını, sentezlenen RTL ve eşleştirme ardı devre şemalarını ekleyerek yorumlayınız.

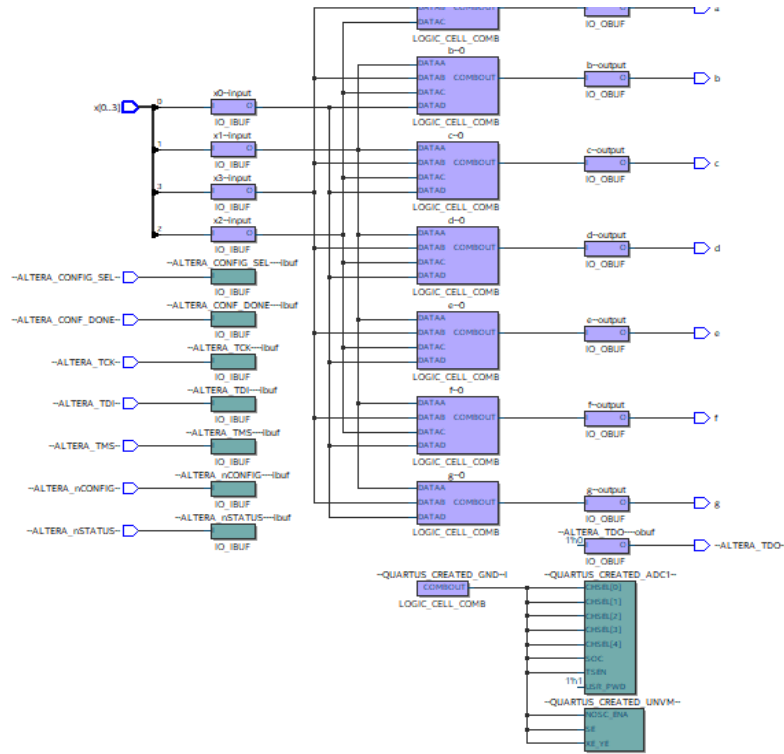
Flow Summary		Analysis & Synthesis Resource Usage Summary							
<<Filter>>									
Flow Status	Successful - Tue Jun 29 15:01:22 2021								
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition								
Revision Name	lab1_g24_p3								
Top-level Entity Name	lab1_g24_p3								
Family	MAX 10								
Device	10M08DAF484C8G								
Timing Models	Final								
Total logic elements	8 / 8,064 (< 1 %)								
Total registers	0								
Total pins	11 / 250 (4 %)								
Total virtual pins	0								
Total memory bits	0 / 387,072 (0 %)								
Embedded Multiplier 9-bit elements	0 / 48 (0 %)								
Total PLLs	0 / 2 (0 %)								
UFM blocks	0 / 1 (0 %)								
ADC blocks	0 / 1 (0 %)								

<<Filter>>													
Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers	Memory Bits	UFM Blocks	DSP Elements	DSP 9x9	DSP 18x18	Pins	Virtual Pins	ADC blocks	Full Hierarchy Name	Entity Name	Library Name
1 lab1_g24_p3	7 (7)	0 (0)	0	0	0	0	0	11	0	0	lab1_g24_p3	lab1_g24_p3	work

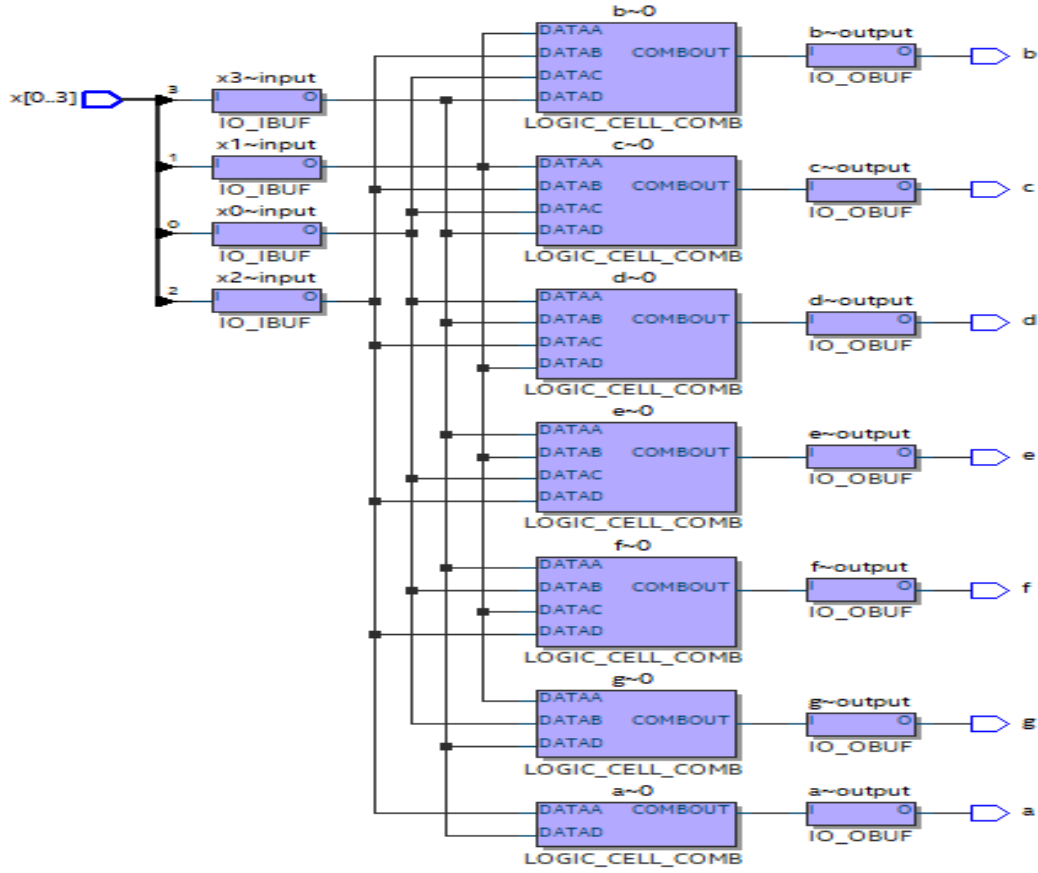
Şekil 17 Analiz ve Sentez Özeti



Şekil 18 RTL Devresi



Şekil 19.a Techmap Şeması



Şekil 19.b Techmap Şeması

1.3 Deneyin Yorumu

7 bölmeli ekran (7BE) (7-segment display) üzerinde tasarım yapılması istenmektedir. Her bir ABCDEFG bölmesi için ortak bir anode, ayrı bir cathode bulunmaktadır. Başka bir deyişle, ortak pine lojik 1, ABCDEFG girişlerine ise lojik 0 gönderildiğinde istenilen bölme yanacak şekilde çözücü tasarımı yapılmıştır.

Referanslar

1. https://en.wikipedia.org/wiki/Seven-segment_display
2. <https://github.com/fcayci/sv-digital-design>
3. <https://www.geeksforgeeks.org>