



GEBZE TEKNİK ÜNİVERSİTESİ
ELEKTRONİK MÜHENDİSLİĞİ

ELM235

LOJİK DEVRE TASARIM LABORATUVARI

LAB 2 Deney Raporu

Donanım Tanımlama Dilleri Uygulamaları: Birleşik Devreler

Hazırlayanlar
1) 200102002025 – Umut Mehmet ERDEM
2) 200102002066 – Emre TANER

İçindekiler Tablosu

1. Giriş	2
2. Problemler.....	2
3. Sonuçlar ve Genel Yorumlar	23
4. Referanslar	23

1. Giriş

Bu deneyde aşağıda verilen maddeler amaçlanmaktadır:

1. Donanım tanıma dillerini (DTD) kullanarak devre tasarımı yapmak.
2. Sentezleyici araçları kullanarak, DTD ile tanımlanan devreleri FPGA için sentezlemek.
3. Simülasyon araçları kullanarak, otomatik simülasyon yaptırmak.
4. Devrede oluşan gecikmeleri gözlemlemek.

2. Problemler

2.1. Problem I – İstenmeyen Darbe Sinyalleri: Glitchler

2.1.1. Teorik Araştırma

Glitch'in tanımı şu şekildedir:

Glitch, sistemde kısa ömürlü bir arızadır. Genellikle kendini düzelten geçici bir hatayı tanımlamak için kullanılır ve bu nedenle sorunu gidermek zordur. Bu terim, bilgisayar ve elektronik endüstrilerinde, devre bükülmelerinde ve ayrıca video oyunu oynayanlar tarafından kullanılmaktadır.

2.1.2 Deneyin Yapılışı

Quartus Prime üzerinden yeni bir proje açılmış açılan projenin üst düzey tasarım varlığının adı (name of top-level design entity) Modelsim programı üzerinden oluşturulan System Verilog (.sv) dosyası ile aynı şekilde isimlendirilmiştir. İleriki adımlarda “.sv” dosyası ve kullanılacak board Şekil 1’ deki gibi seçilmiştir.

New Project Wizard

Family, Device & Board Settings

Device Board

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family

Family: MAX 10 (DA/DF/DC/SA/SC)

Device: All

Target device

☐ Auto device selected by the Filter

☒ Specific device selected in 'Available devices' list

☐ Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: Any

Core speed grade: Any

Name filter:

☒ Show advanced devices

Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit elements
10M08DAF484C8G	1.2V	8064	250	250	387072	48

< >

< Back Next > Finish Cancel Help

Şekil 1.Yeni Proje Oluştururken Board Seçimi

a) Verilen Boole Denklemi Donanım Tanımlama Dili Kullanarak Gerçeklenmesi

Denklem 1: $Y = AB'C + C'D$

Denklem 1'de verilen boole denklemi DTD kullanarak bir modül olarak gerçekleştirildi ve her kapağa 2ns gecikme vererek atama yapıldı. Yazılan kod aşağıdaki gibidir:

```
/* lab2_g7_p1.sv
* Hazırlayanlar: Umut Mehmet ERDEM - Emre TANER
* Notlar: ELM235 2023 Bahar Lab2 - Problem 1
* Denklem 1'de verilen boole denklemi donanım tanımlama dili kullanarak gerçekleştirilmesi
*/
`timescale 1ns/1ps
module lab2_g7_p1(
    input logic A, B, C, D,
    output logic Y
);

wire n1,n2,n3,n4,n5;

assign #2 n1 = ~B;
assign #2 n2 = A & n1;
assign #2 n3 = C & n2;
assign #2 n4 = ~C;
assign #2 n5 = D & n4;
assign #2 Y = n3 | n5;
endmodule
```

b) Tasarlanılan DTD Modelinin Simüle Edilmesi

Tasarlanılan DTD modelini ve ilgili fonksiyonel test tezgâhı kodunu girişlere bütün olası kombinasyonları her 10ns'lik adımlarda uygulanıldı ve çıkış sinyali gözlemlenecek şekilde simüle edildi. Yazılan kod aşağıdaki gibidir:

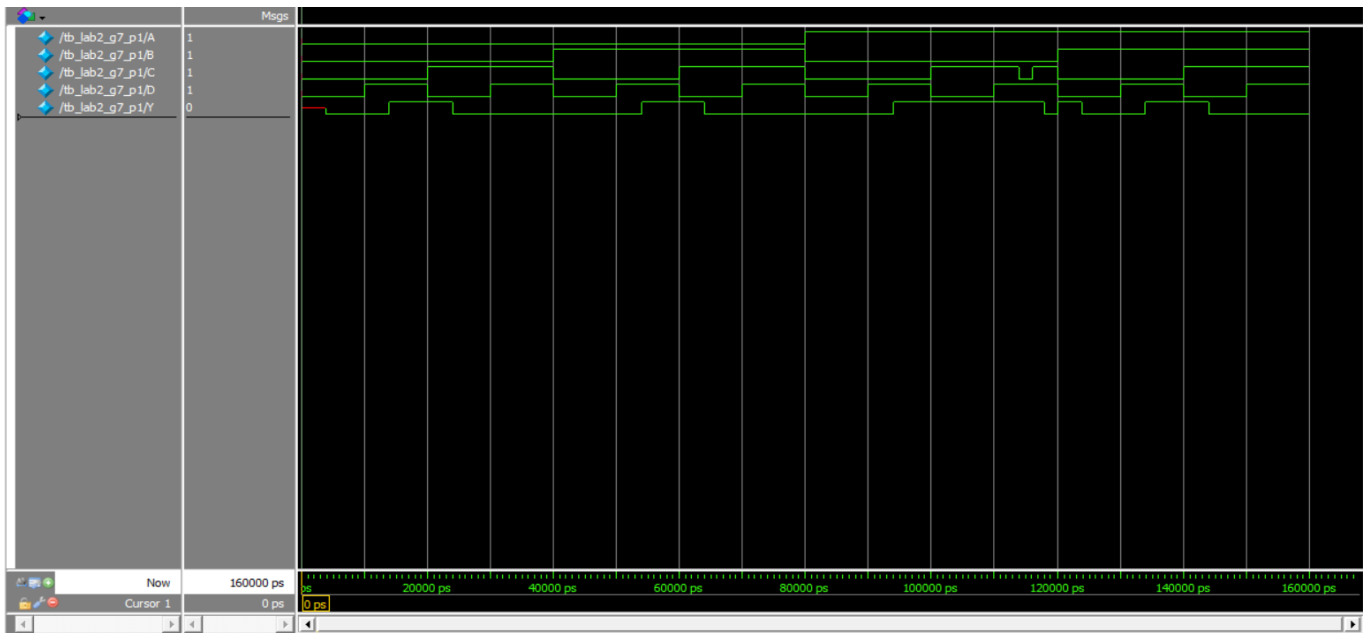
```
/* tb_lab2_g7_p1.sv
* Hazırlayanlar: Umut Mehmet ERDEM - Emre TANER
* Notlar: ELM235 2023 Bahar Lab2 - Problem 1 Testbench
* Denklem 1'de verilen boole denklemi donanım tanımlama dili kullanarak gerçekleştirilmesi
* Bütün olası girişlere göre çıkış gözlemlenir.
*/
// Zaman birimi ve simülasyon çözünürlüğü
`timescale 1ns/1ps
```

```

module tb_lab2_g7_p1();
// Test tezgahlarında port bulunmaz
    logic A, B, C, D; // test tezgahi giriş sinyal tanımları
    logic Y; // test tezgahi çıkış sinyal tanımları
// Test edilecek modulün yaratımı ve port bağlantılarının yapılması
// dut = device under test
    Lab2_g7_p1 dut0(A, B, C, D, Y);
// Bu kısımda sinyaller test edilen devreye sıralı olarak uygulanır.
// Sonuçlar test edilen devre çıkışlarında gözlenebilir.
initial begin
    A = 0; B = 0; C = 0; D = 0;      #10 // 0000 -10 ns bekle
    D = 1;                          #10 //0001
    C = 1; D = 0;                    #10 //0010
    D = 1;                          #10 //0011
    B = 1; C = 0; D = 0;             #10 //0100
    D = 1;                          #10 //0101
    C = 1; D = 0;                    #10 //0110
    D = 1;                          #10 //0111
    A = 1; B = 0; C = 0; D = 0;      #10 //1000
    D = 1;                          #10 //1001
    C = 1; D = 0;                    #10 //1010
    C = 1; D = 1;                    #4 //1011
    C = 0; D = 1;                    #2
    C = 1; D = 1;                    #4
    B = 1; C = 0; D = 0;             #10 //1100
    D = 1;                          #10 //1101
    C = 1; D = 0;                    #10 //1110
    D = 1;                          #10 //1111
    $stop; // simülasyonu durdur
end
endmodule

```

c) Zamanlama Diyagramı



Şekil 2. Problem 1'in Zamanlama Diyagramı

Oluşan zamanlama diyagramı incelendiğinde girişe A=1, B=0, C=1, D=1 uygulandığında Y çıktısında geçici bir hata olduğu gözlemlenmektedir. Buna Glitch denmektedir. Hatanın sebebi C'deki anlık değişimdir.

d) Boole Denklemin Doğruluk Tablosu ve Olası Glitch

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

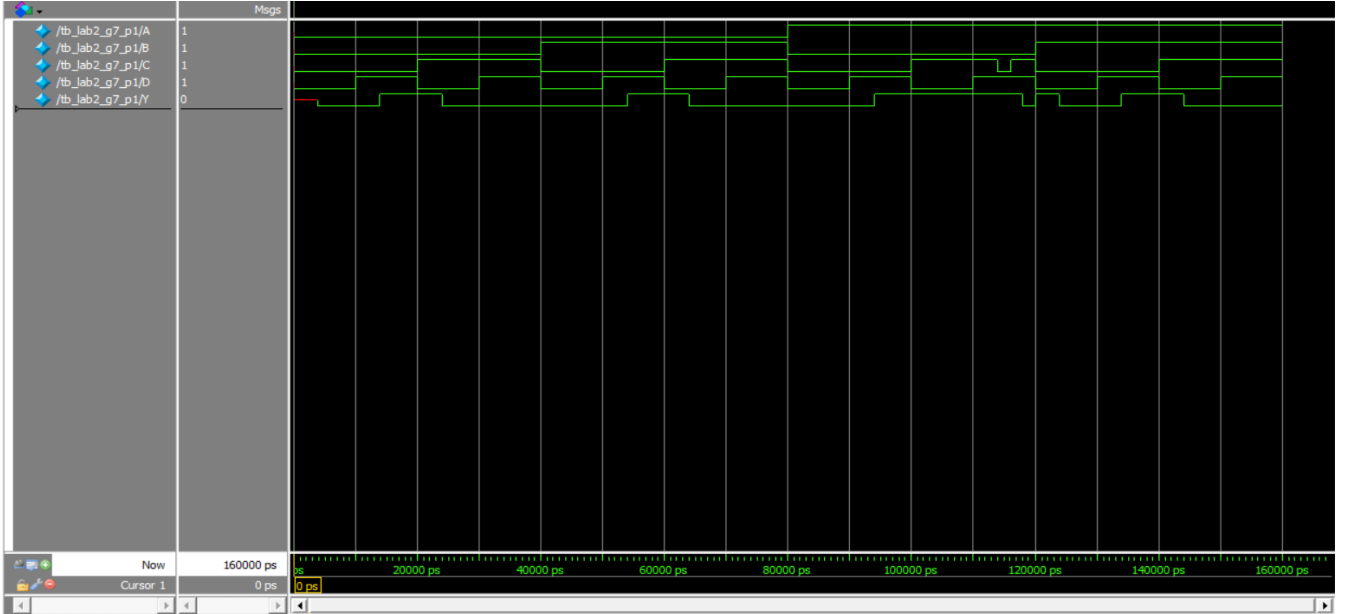
Tablo 1. Denklem 1'in Doğruluk Tablosu

AB\CD	C'D'	C'D	CD	CD'
A'B'	0	1	0	0
A'B	0	1	0	0
AB	0	1	0	0
AB'	0	1	1	1

Tablo 2. Denklem 1'in Karnaugh Map'i

Denklem 1'in Karnaugh Map'i incelendiğinde AB'C'D ve AB'CD girişlerinde sistemin 1 çıkışı verildiği görülmektedir. K-Map'te sadeleştirme yapılırken 2 ve 2'nin katları olacak şekilde ve hiçbir 1 dışarıda kalmayacak şekilde daire içine alınıp sadeleştirme yapılır. Daire içine almadaki bir diğer kural ise daire sayısının en az sayıda olmasıdır. Bundan dolayı AB'C'D ve AB'CD bloğundaki 1'ler daire içine alınmıyor. Fakat daire içine alınmadığından ve C'nin anlık değişmesi koşuluyla bir Glitch oluşmaktadır. Daire içine alınmayan bu blokların doğruluk tablosunda 1011 ve 1001 girişlerine denk geldiği görülmektedir. 1011 → 1001 (C girişi anlık olarak değişiyor)

1011 girişinde C'yi anlık olarak değiştirilmesinin test tezgâhı koduna konulduğunda, oluşan zaman diyagramı aşağıdaki gibidir:



Şekil 3. Bulunan Koşulun Girişe Uygulanması Sonucu Oluşan Zamanlama Diyagramı

e) Glitch'in Giderilmesi

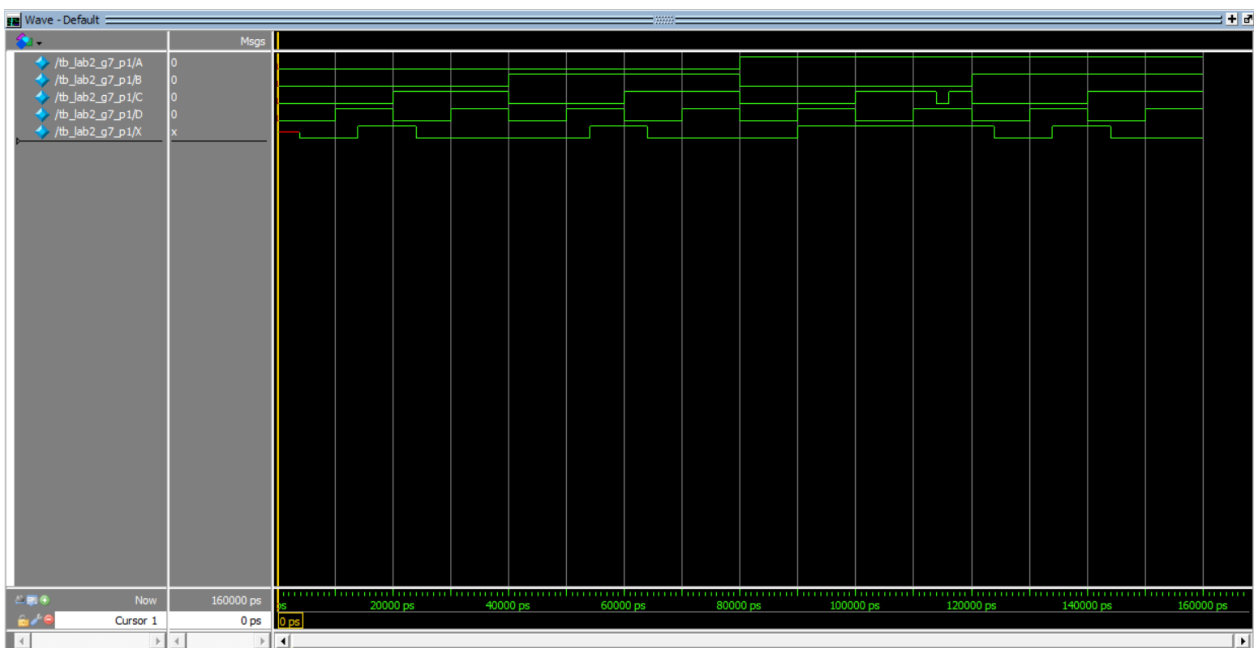
Glitch'in düzeltilmesi için K-Map'te birlikte daire içine alınmayan 1011 ve 1001 bloklarını daire içine alınıp sadeleştirme yapıldıktan sonra denklem 1'e eklenmelidir. Bu sayede C'nin anlık değişiminden kaynaklı Glitch giderilmiş olunacaktır.

Güncellenmiş DTD modülünün kodu ve güncelleme sonucu oluşan zamanlama diyagramı aşağıdaki gibidir:

```
/* lab2_g7_p1.sv
* Hazırlayanlar: Umut Mehmet ERDEM - Emre TANER
* Notlar: ELM235 2023 Bahar Lab2 - Problem 1
* Denklem 1'de verilen boole denkleminin glitch'inin giderilmesi
*/
`timescale 1ns/1ps
module lab2_g7_p1(
    input logic A, B, C, D,
    output logic X
);


wire n1,n2,n3,n4,n5;

assign #2 n1 = ~B;
assign #2 n2 = A & n1;
assign #2 n3 = C & n2;
assign #2 n4 = ~C;
assign #2 n5 = D & n4;
assign #2 X = n3 | n5 | A & ~B & D;
endmodule
```




Şekil 4. Giderilmiş Glitch Denkleminin Zamanlama Diyagramı

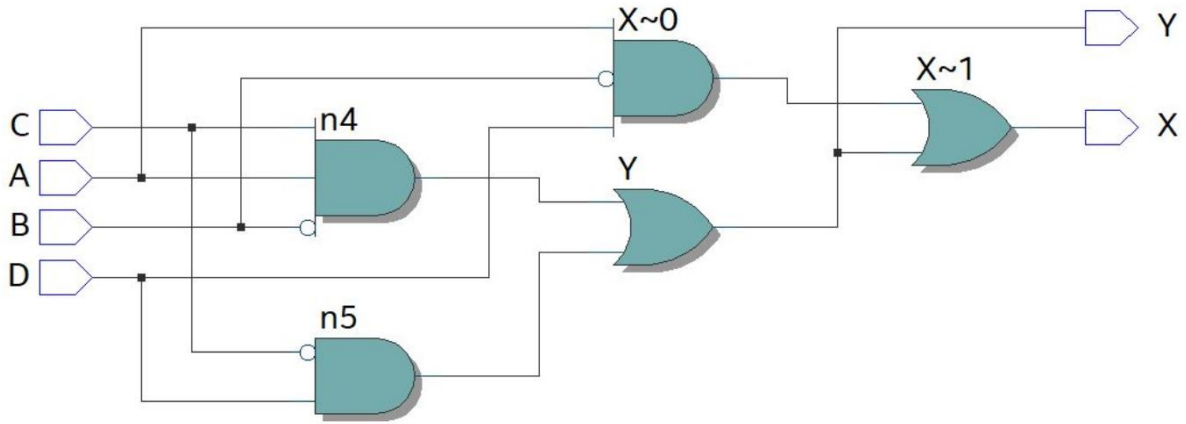
f) Analiz, Sentez, Kaynak Kullanım Özeti ve RTL, Eşleştirme Sonrası Teknoloji Şeması

Flow Summary	
 <<Filter>>	
Flow Status	Successful - Thu May 04 21:10:32 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	lab2_g7_p1
Top-level Entity Name	lab2_g7_p1
Family	MAX 10
Device	10M08DAF484C8G
Timing Models	Final
Total logic elements	2 / 8,064 (< 1 %)
Total registers	0
Total pins	6 / 250 (2 %)
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 %)

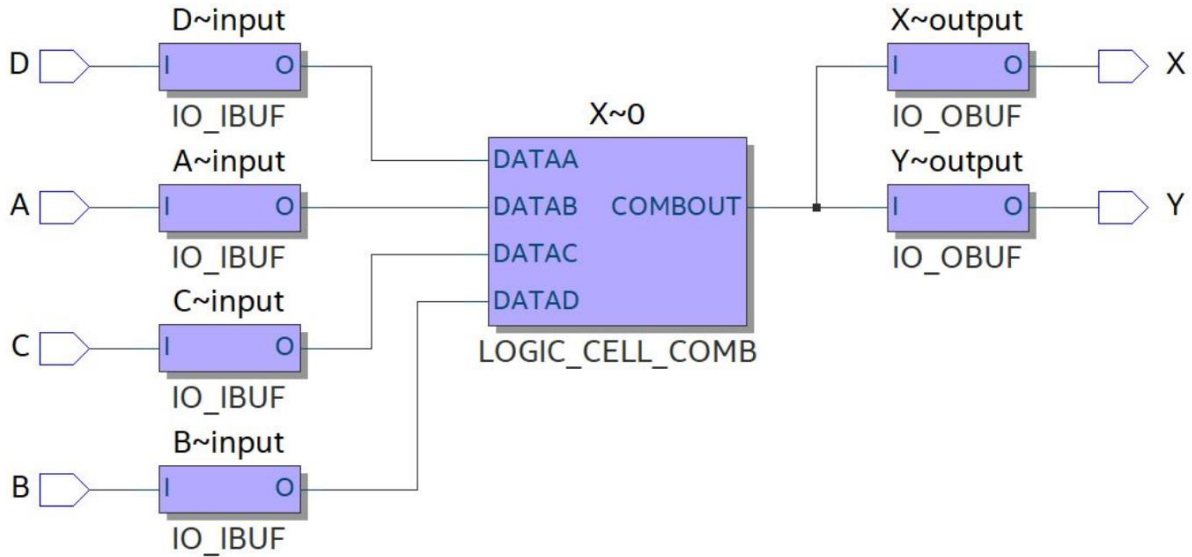
Şekil 5. Problem 1'in Analiz ve Sentez Özeti

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	1
2	-- 3 input functions	0
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	6
11		
12	Embedded Multiplier 9-bit elements	0
13		
14	Maximum fan-out node	X~0
15	Maximum fan-out	2
16	Total fan-out	12
17	Average fan-out	0.92

Şekil 6. Problem 1'in Analiz ve Sentez Kaynak Kullanımı Özeti



Şekil 7. Problem 1'in RTL Şeması



Şekil 8. Problem 1'in Eşleştirme Sonrası Teknoloji Şeması

2.1.3. Sonuçların Yorumu

Problem 1'de verilen denklemde hangi koşulda Glitch oluşabileceği ve oluşan bu Glitch'in çıkış sinyalinde ne gibi bir etkisi olduğu Şekil 3'te görülmektedir. K-Map yardımıyla oluşan bu Glitch'in nasıl düzeltilebileceği Tablo 2'de ve düzeltildikten sonra çıkış sinyalinin düzeldiği Şekil 4'te görülmektedir. Buna göre Glitch'in düzeltilmesinde K-Map'in önemli bir yol olduğu söylenebilir.

2.2. Problem II- Boole Cebri Teoremlerini Kullanarak Lojik Devre Sadeleştirme

2.2.1. Teorik Araştırma

Yedi segment Led görüntü modülü ondalık sayıların gösteriminde kullanılan elektronik bir görüntülme cihazdır. Yedi segment göstericiler dijital saatlerde, elektronik sayaçlarda ve diğer birçok elektronik cihazda sayısal bilgi görüntülemek için yaygın olarak kullanılmaktadır. Çizgi şeklindeki LED ya da LCD parçaların sekiz rakamının görünüşüne benzer şekilde bir modül üzerine dizilmesi ile oluşturulur. Her bir parçanın uçları modülün alt kısmında bulunan ayrı bir bacaklara bağlıdır. Görüntü belirli bacaklar bir devreye bağlanıp LED ya da LCD parçalar üzerinden akım geçmesi sağlanarak üretilir.

2.2.2. Deneyin Yapılışı

Quartus Prime üzerinden yeni bir proje açılmış açılan projenin üst düzey tasarım varlığının adı (name of top-level design entity) Modelsim programı üzerinden oluşturulan System Verilog (.sv) dosyası ile aynı şekilde isimlendirilmiştir. İleriki adımlarda “.sv” dosyası ve kullanılacak board Şekil 2’ deki gibi seçilmiştir.

New Project Wizard

Family, Device & Board Settings

Device Board

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family

Family: MAX 10 (DA/DF/DC/SA/SC)

Device: All

Target device

☐ Auto device selected by the Fitter

☒ Specific device selected in 'Available devices' list

☐ Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: Any

Core speed grade: Any

Name filter:

☒ Show advanced devices

Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit elements
10M08DAF484C8G	1.2V	8064	250	250	387072	48

< Back Next > Finish Cancel Help

Şekil 9. Yeni Proje Oluştururken Board Seçimi

a)

X3	X2	X1	X0	A	B	C	D	E	F	G
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	x	x	x	x	x	x	x
0	0	1	0	1	0	0	1	1	1	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	x	x	x	x	x	x	x
0	1	0	1	0	0	0	1	1	1	0
0	1	1	0	x	x	x	x	x	x	x
0	1	1	1	x	x	x	x	x	x	x
1	0	0	0	x	x	x	x	x	x	x
1	0	0	1	x	x	x	x	x	x	x
1	0	1	0	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x
1	1	0	0	1	0	0	1	1	1	0
1	1	0	1	1	1	0	1	1	0	1
1	1	1	0	1	0	1	1	0	1	1
1	1	1	1	x	x	x	x	x	x	x

Tablo 3. 7BLE Çalışma Tablosuna Göre Oluşan Doğruluk Tablosu

$$A = F(x_3, x_2, x_1, x_0) = \Sigma(2, 3, 12, 13, 14)$$

$$B = F(x_3, x_2, x_1, x_0) = \Sigma(3, 13)$$

$$C = F(x_3, x_2, x_1, x_0) = \Sigma(3, 14)$$

$$D = F(x_3, x_2, x_1, x_0) = \Sigma(2, 3, 5, 12, 13, 14)$$

$$E = F(x_3, x_2, x_1, x_0) = \Sigma(2, 5, 12, 13)$$

$$F = F(x_3, x_2, x_1, x_0) = \Sigma(2, 5, 12, 14)$$

$$G = F(x_3, x_2, x_1, x_0) = \Sigma(0, 2, 3, 13, 14)$$

b)

$X_3X_2 \backslash X_1X_0$	$X_1'X_0'$	$X_1'X_0$	X_1X_0	X_1X_0'
$X_3'X_2'$	0	x	1	1
$X_3'X_2$	x	0	x	x
X_3X_2	1	1	x	1
X_3X_2'	x	x	x	x

Tablo 4. A Çıkışının Karnaugh Map'i

$$A = x_1 + x_3$$

$X_3X_2 \backslash X_1X_0$	$X_1'X_0'$	$X_1'X_0$	X_1X_0	X_1X_0'
$X_3'X_2'$	0	x	1	0
$X_3'X_2$	x	0	x	x
X_3X_2	0	1	x	0
X_3X_2'	x	x	x	x

Tablo 5. B Çıkışının Karnaugh Map'i

$$B = x_1x_0 + x_3x_0$$

$X_3X_2 \backslash X_1X_0$	$X_1'X_0'$	$X_1'X_0$	X_1X_0	X_1X_0'
$X_3'X_2'$	0	x	1	0
$X_3'X_2$	x	0	x	x
X_3X_2	0	0	x	1
X_3X_2'	x	x	x	x

Tablo 6. C Çıkışının Karnaugh Map'i

$$C = x_2'x_0 + x_2x_1$$

X3X2\X1X0	X1'X0'	X1'X0	X1X0	X1X0'
X3'X2'	0	x	1	1
X3'X2	x	1	x	x
X3X2	1	1	x	1
X3X2'	x	x	x	x

Tablo 7. D Çıkışının Karnaugh Map'i

$$D = x_1 + x_2$$

X3X2\X1X0	X1'X0'	X1'X0	X1X0	X1X0'
X3'X2'	0	x	0	1
X3'X2	x	1	x	x
X3X2	1	1	x	0
X3X2'	x	x	x	x

Tablo 8. E Çıkışının Karnaugh Map'i

$$E = x_2x'_1 + x'_3x_1x'_0$$

X3X2\X1X0	X1'X0'	X1'X0	X1X0	X1X0'
X3'X2'	0	x	0	1
X3'X2	x	1	x	x
X3X2	1	0	x	1
X3X2'	x	x	x	x

Tablo 9. F Çıkışının Karnaugh Map'i

$$F = x_1x'_0 + x'_3x_2 + x_2x'_0$$

$X_3X_2 \backslash X_1X_0$	$X_1'X_0'$	$X_1'X_0$	X_1X_0	X_1X_0'
$X_3'X_2'$	1	x	1	1
$X_3'X_2$	x	0	x	x
X_3X_2	0	1	x	1
X_3X_2'	x	x	x	x

Tablo 10. G Çıkışının Karnaugh Map'i

$$G = x_2' + x_1 + x_3x_0$$

A çıkışında 1 adet, B çıkışında 3 adet, C çıkışında 4 adet, D çıkışında 1 adet, E çıkışında 7 adet, F çıkışında 8 adet, G çıkışında 4 adet olmak üzere toplamda 28 adet devre elamanı kullanılmıştır.

c)

```

/* lab2_g7_p2.sv
* Hazırlayanlar: Umut Mehmet ERDEM – Emre TANER
* Notlar: ELM235 2023 Bahar Lab2 – Problem 2
* Çözücü devresinin DTD modelini davranışsal tarzda yazılması
*/
// 1ns birim zaman ve 1ps çözünürlükte çalışır.
`timescale 1ns/1ps
module lab2_g7_p2(
    input logic [3:0] x,
    output logic [6:0] sev_segments
);

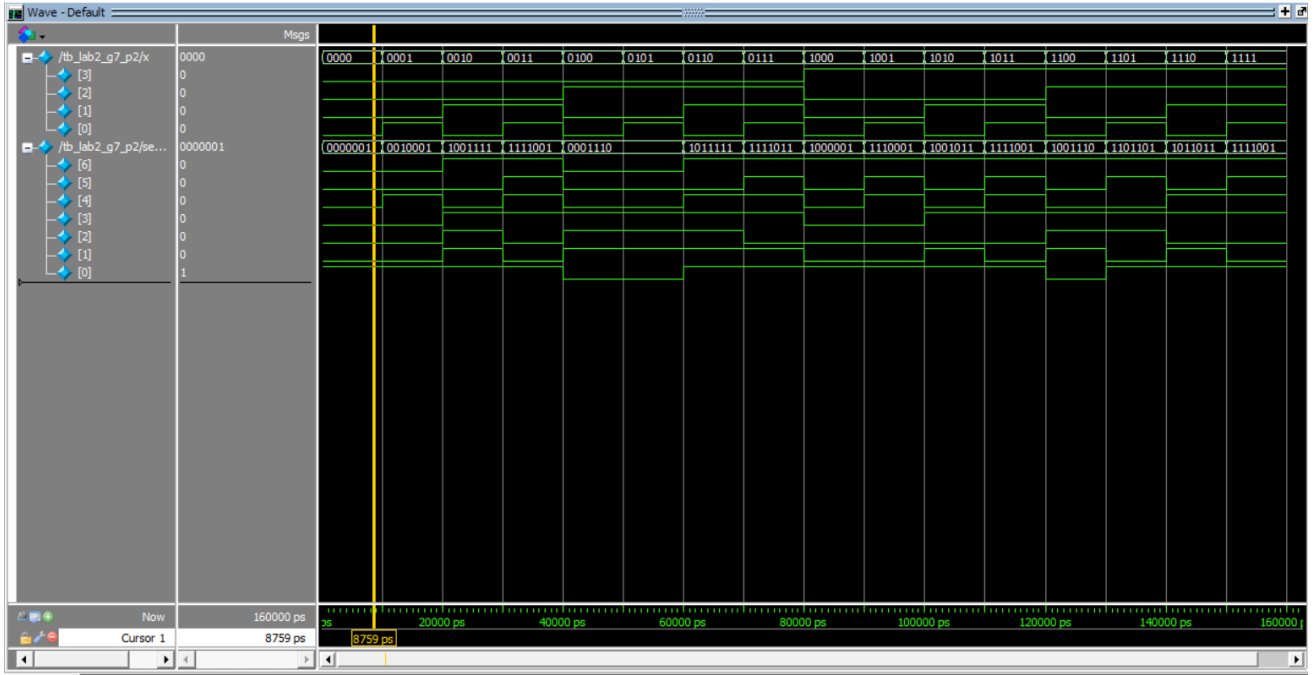
assign sev_segments[6] = x[1] | x[3];
assign sev_segments[5] = x[1]&x[0] | x[3]&x[0];
assign sev_segments[4] = ~x[2]&x[0] | x[2]&x[1];
assign sev_segments[3] = x[1] | x[2];
assign sev_segments[2] = x[2]&~x[1] | ~x[3]&x[1]&~x[0];
assign sev_segments[1] = x[1]&~x[0] | ~x[3]&x[2] | x[2]&~x[0];
assign sev_segments[0] = ~x[2] | x[1] | x[3]&x[0];
endmodule

```


d)

```
/* tb_lab2_g7_p2.sv
* Hazırlayanlar: Umut Mehmet ERDEM - Emre TANER
* Notlar: ELM235 2023 Bahar Lab2 - Problem 2 Testbench
* Çözücü devresinin DTD modelini davranışsal tarzda yazılması
* Bütün olası girişlere göre çıkış gözlemlenir.
*/
// Zaman birimi ve simülasyon çözünürlüğü
`timescale 1ns/1ps
module tb_lab2_g7_p2();
// Test tezgahlarında port bulunmaz
    logic [3:0] x; // test tezgahi giriş sinyal tanımları
    logic [6:0] sev_segments; // test tezgahi çıkış sinyal tanımları
// Test edilecek modülün yaratımı ve port bağlantılarının yapılması
// dut = device under test
    Lab2_g7_p2 dut0(x, sev_segments);
// Bu kısımda sinyaller test edilen devreye sıralı olarak uygulanır.
// Sonuçlar test edilen devre çıkışlarında gözlemlenebilir.
initial begin
    x[3] = 0; x[2] = 0; x[1] = 0; x[0]=0; #10 // 0000 -10 ns bekle
    x[0] = 1; #10 // 0001
    x[1] = 1; x[0] = 0; #10 // 0010
    x[0] = 1; #10 // 0011
    x[2] = 1; x[1] = 0; x[0] = 0; #10 // 0100
    x[0] = 1; #10 // 0101
    x[1] = 1; x[0] = 0; #10 // 0110
    x[0] = 1; #10 // 0111
    x[3] = 1; x[2] = 0; x[1] = 0; x[0]=0; #10 // 1000
    x[0] = 1; #10 // 1001
    x[1] = 1; x[0] = 0; #10 // 1010
    x[0] = 1; #10 // 1011
    x[2] = 1; x[1] = 0; x[0] = 0; #10 // 1100
    x[0] = 1; #10 // 1101
    x[1] = 1; x[0] = 0; #10 // 1110
    x[0] = 1; #10 // 1111
    $stop; // simülasyonu durdur
end
endmodule
```

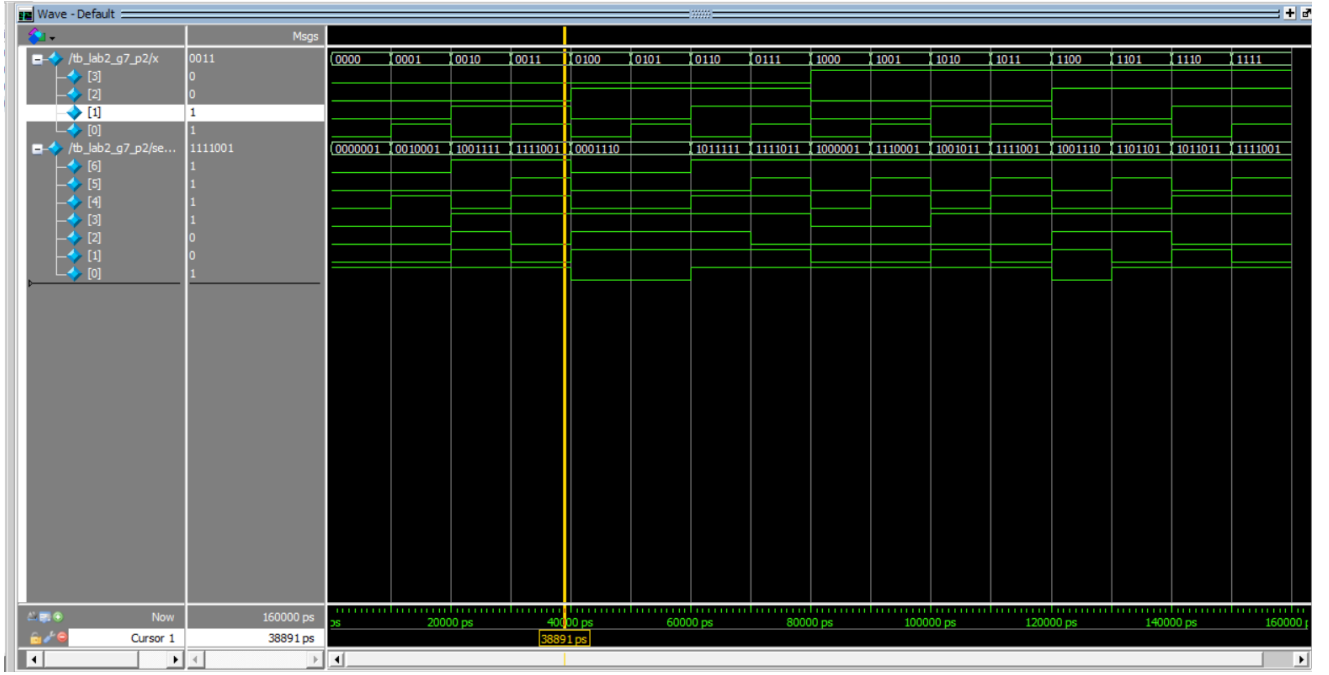
Aşağıdaki zamanlama diyagramlarındaki [6] A LED'ini, [5] B LED'ini, [4] C LED'ini, [3] D LED'ini, [2] E LED'ini, [1] F LED'ini, [0] G LED'ini ifade etmektedir.



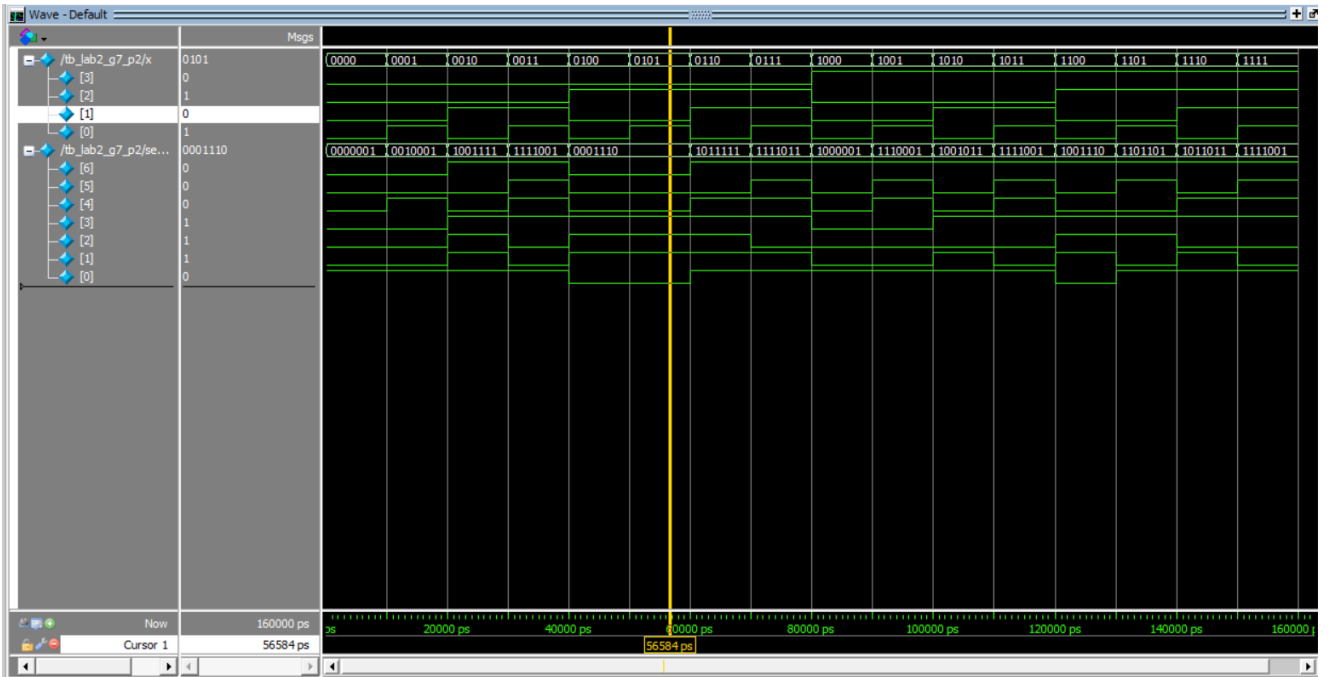
Şekil 10. Girişe '0000' Uygulandığında '1' Çıktısı Veren Zamanlama Diyagramı



Şekil 11. Girişe '0010' Uygulandığında 'E' Çıktısı Veren Zamanlama Diyagramı



Şekil 12. Girişe '0011' Uygulandığında '3' Çıktısı Veren Zamanlama Diyagramı



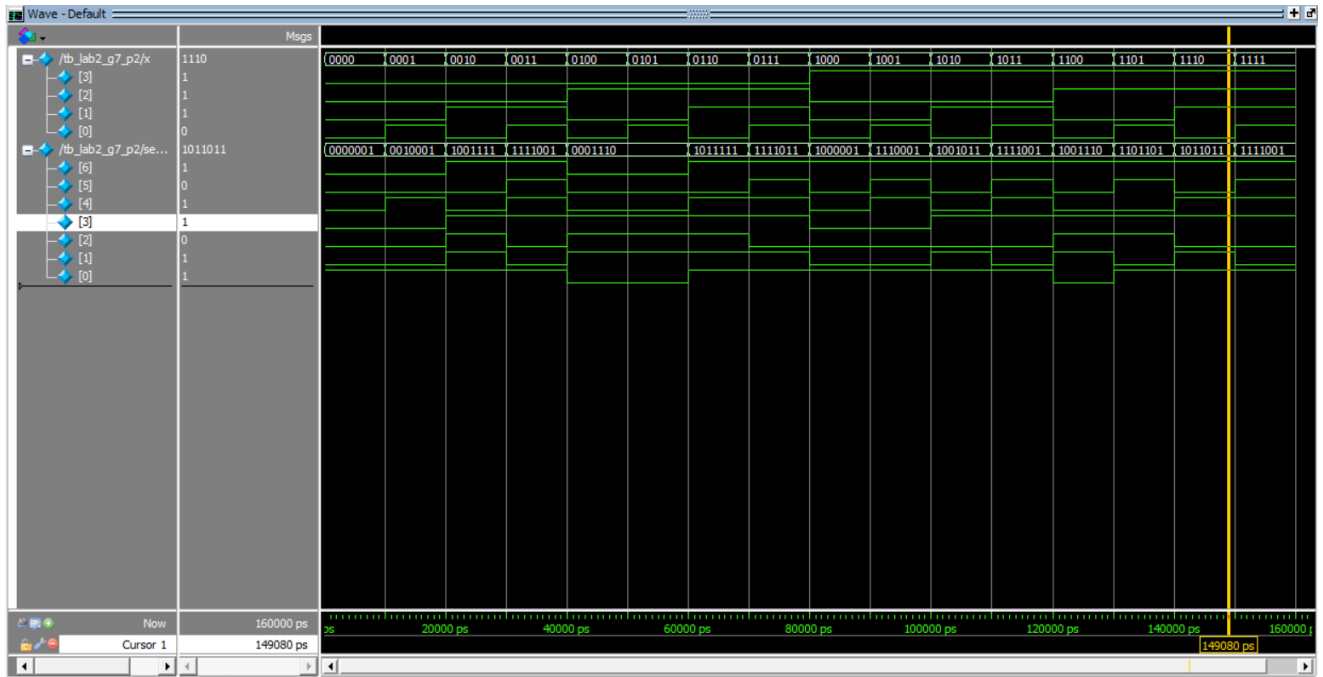
Şekil 13. Girişe '0101' Uygulandığında 'L' Çıktısı Veren Zamanlama Diyagramı



Şekil 14. Girişe '1100' Uygulandığında 'C' Çıktısı Veren Zamanlama Diyagramı

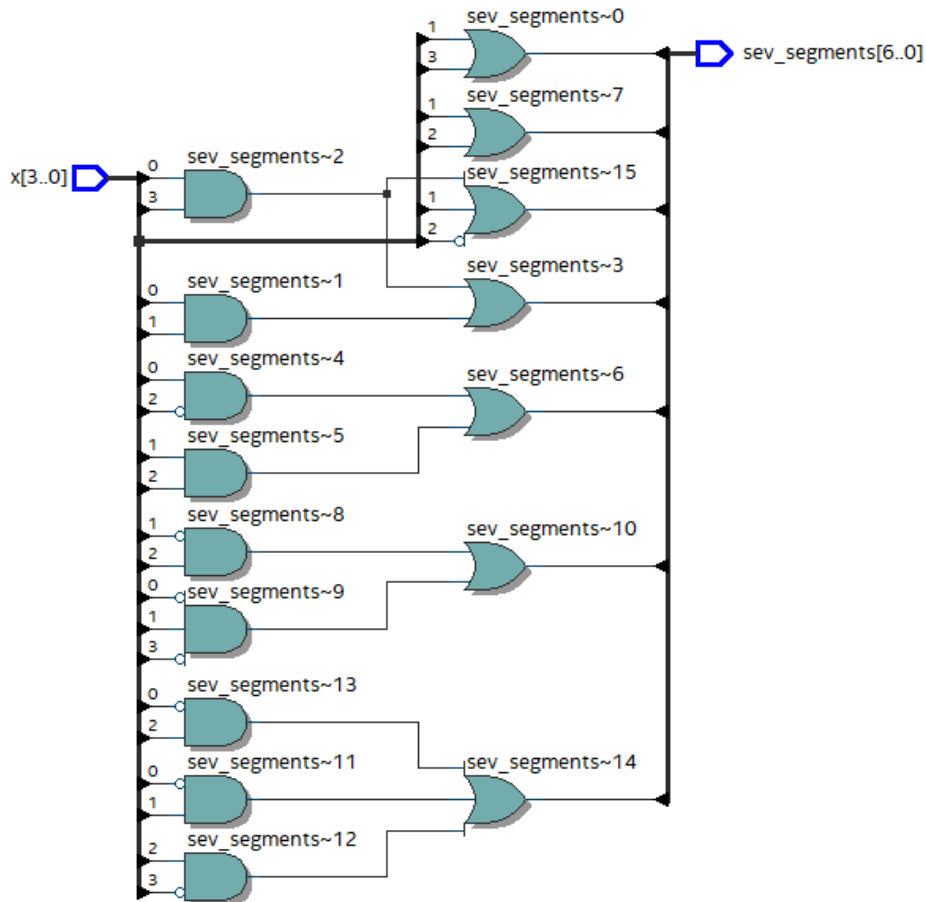


Şekil 15. Girişe '1101' Uygulandığında '2' Çıktısı Veren Zamanlama Diyagramı

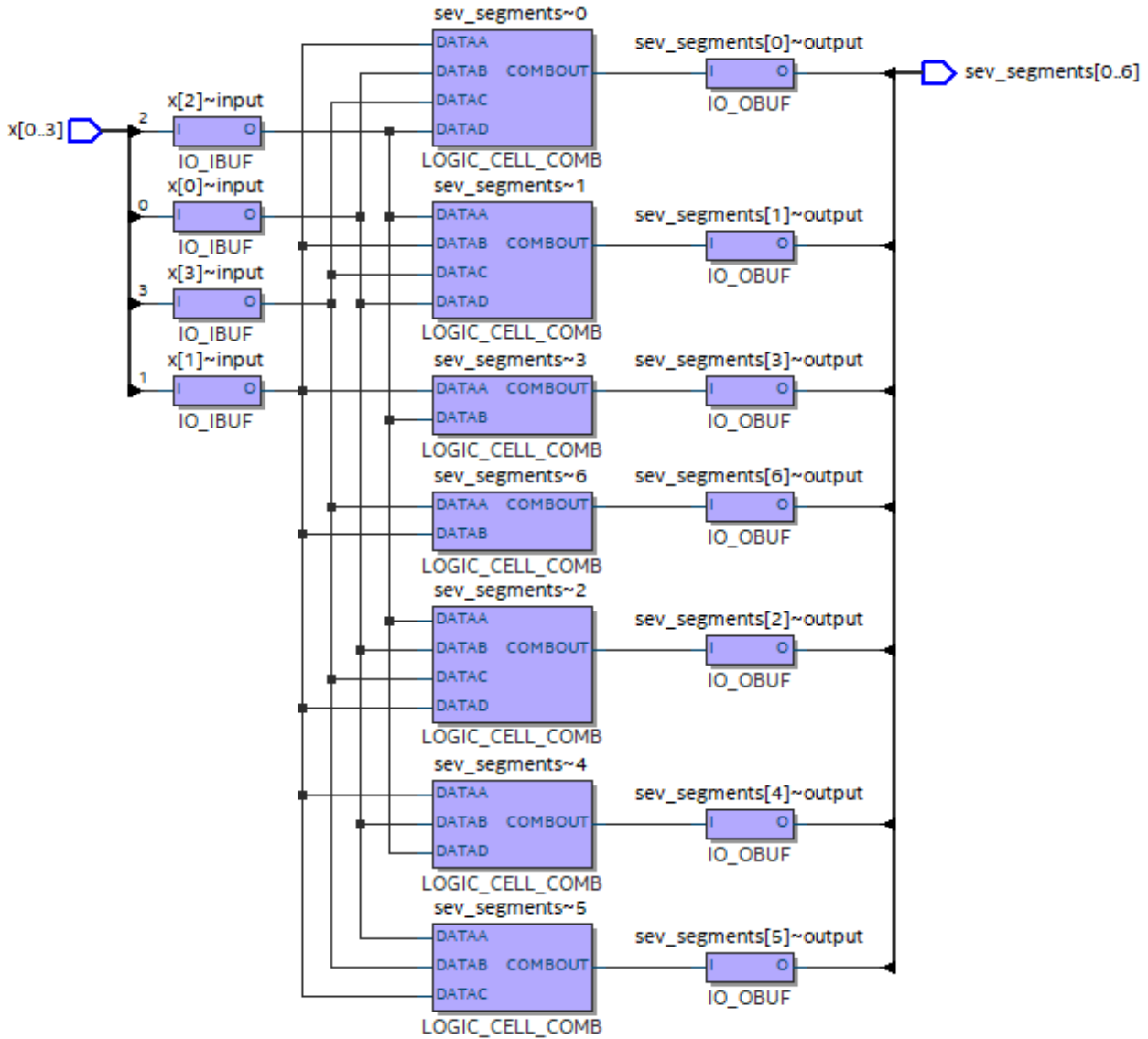


Şekil 16. Girişe '1110' Uygulandığında '5' Çıktısı Veren Zamanlama Diyagramı

e)



Şekil 17. Problem 2'nin RTL şeması



Şekil 18. Problem 1'in Eşleştirme Sonrası Teknoloji Şeması

Şekil 17' de gösterilen x 4 bitlik değişkeninde K-maplerde bulduğumuz 7 bitlik A, B, C, D, E, F, G çıkış fonksiyonlarına göre benzer devre şeması çıktığı gözlenmektedir. RTL şemasında gösterilen devrelerin her biri Şekil 18' de post-mapping şemasında logic-cell-comb'larda gösterilmektedir. Bu nedenle Şekil 18' de x değişkeninin 4 girişi' nin fonksiyonlara göre mantık hücre kombinasyonlarına girmesi ile 7 çıkış bulunmuştur ve sev_segments değişkeninde bit olarak gösterilmektedir.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Thu May 04 21:34:23 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	lab2_g7_p2
Top-level Entity Name	lab2_g7_p2
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 %)

Şekil 19. Problem 2'nin Analiz ve Sentez Özeti

Analysis & Synthesis Resource Usage Summary		
<<Filter>>		
	Resource	Usage
1	Estimated Total logic elements	7
2		
3	Total combinational functions	7
4	▼ Logic element usage by number of LUT inputs	
1	-- 4 input functions	3
2	-- 3 input functions	2
3	-- <=2 input functions	2
5		
6	▼ Logic elements by mode	
1	-- normal mode	7
2	-- arithmetic mode	0
7		
8	▼ Total registers	0
1	-- Dedicated logic registers	0
2	-- I/O registers	0
9		
10	I/O pins	11
11		
12	Embedded Multiplier 9-bit elements	0
13		
14	Maximum fan-out node	x[1...put
15	Maximum fan-out	7
16	Total fan-out	40
17	Average fan-out	1.38

Şekil 20. Problem 2'nin Analiz ve Sentez Kaynak Kullanımı Özeti

Şekil 19’ da Total pins olarak 4 bit giriş, 7 bit çıkış olmak üzere 11 pin gözlenmektedir. Total logic elements, 8 tanedir. Şekil 20’ de Şekil 19’a göre farklı olan veriler; combinational functions, Şekil 18’ de gösterilen logic_cell_comb’ lara eşittir. Estimated Total logic elements değeri, combinational functions değerlerine eşittir. LUT inputs (Look up table input) değerleri, combinational functions’ ta gösterilen logic_cell’lerin giriş miktarına göre sayıları verilmiştir.

f)

Şekil 17, 18, 19 ve 20’ deki verilere bakılarak kullanılan pinlerin toplamının Problem 1’de Şekil 5, 6, 7 ve 8’ deki görsellere bakılarak daha fazla olduğu anlaşılabılır. Aynı şekilde kullanılan logic_cell_comb miktarı da Problem 2’de daha fazla kullanılmıştır.

2.2.3. Sonuçların Yorumu

Verilen laboratuvar çalışmasında 2. Problem de istenen tablo değerlerine göre, BCD Display üzerinde desen çıkarma işleminde ortak katota göre pinlere güç verileceğinden ve ortak katot olmasının BCD Display üzerindeki pinlerin ortak toprak noktalarına bağlı olmasını ifade ettiğinden buna bağlı olarak VCC pinlerinin ayrı olmasına bakılarak 1 değerinin pinlerin yanması, 0 değerinin pinlerin sönmesini ifade ettiği anlaşılmaktadır. Buna göre, giriş ve çıkış pinlerinin truth table’ ı oluşturulmuş ve K-map tablosu yapılmıştır. Buna göre her bir pinin fonksiyonları çıkarılmış ve kodu yazılmıştır. Gelen değerlerin zamanlama diyagramında incelenmiş ve doğruluğu teyit edilmiştir.

3. Sonuçlar ve Genel Yorumlar

Yapılan laboratuvar çalışmasında, yaptığımız adımların belli bir hiyerarşi içerisinde teorik bilginin simüle edilip daha sonra sentezlenerek ürünün ortaya çıktığı görülmüştür. Buna göre; simüle edilecek denklemin öncelikle doğruluk tablosu çıkarılmış, daha sonra çıkarılan doğruluk tablosuna ve Boole denklemine göre ModelSim üzerinden kodu yazılmıştır. Glitch’lerin nasıl oluştuğu ve nasıl giderilmesi gerektiği araştırılmış ve bunun analizi zamanlama diyagramı üzerinde incelenmiştir. BCD display’lerin nasıl kullanıldığı, ortak katot veya ortak anot olması durumunda pinlerin nasıl değiştiği anlaşılmıştır. Değişkenlerin birden fazla bit olarak nasıl DTD dilinde tanımlanacağı anlaşılmıştır ve buna göre kodlar yazılıp çıktı sonuçları incelenmiş ve doğru olduğu görülmüştür.

4. Referanslar

[1]https://tr.wikipedia.org/wiki/Yedi_segment_LED_g%C3%B6r%C3%BCnt%C3%BCmod%C3%BCl%C3%BC

[2] <https://tr.wikipedia.org/wiki/Glitch>

[3] Intel® Quartus® Prime Software Suite. URL: <https://www.intel.com.tr/content/www/tr/tr/software/programmable/quartus-prime/overview.html> Accessed: 21.02.2020

[4] Quartus Prime Introduction Using Schematic Designs. URL: ftp://ftp.intel.com/Pub/fpgaup/pub/Intel_Material/16.0/Tutorials/Schematic/Quartus_II_Introduction.pdf Accessed: 21.02.2020