



GEBZE TEKNİK ÜNİVERSİTESİ
ELEKTRONİK MÜHENDİSLİĞİ

ELM235

LOJİK DEVRE TASARIM LABORATUVARI

LAB 1 Deney Raporu

Lojik Devreler ve Tasarım Laboratuvarı Dersine Hazırlık

Hazırlayanlar
1) 210102002053 – Şule Nur Demirdağ
2) 210102002132 – Aysu Aktürk

1. Giriş

Bu rapor donanım tanımlama dillerini (DTD) kullanarak devre tasarımı yapmak, sentezleyici araçlar kullanarak DTD ile tanımlanan devreleri FBGA için sentezlemek, devrelerde oluşan gecikmeleri, kusurları gözlemek ve K-map gibi konular içermektedir. Ayrıca bu kusurların düzeltilme yolları hakkında laboratuvar çalışması ve sonuçlarını içermektedir.

2. Problemler

2.1. Problem 1 – İstenmeyen Darbe Sinyalleri: Glitchler

2.1.1. Teorik Araştırma

2.1.1.1. Glitch Nedir?

Logic devrelerinde "glitch", istenmeyen ve geçici olarak oluşan sinyal dalgalanmalarıdır. Glitchler, bir mantıksal devre veya diğer elektronik sistemlerin işlevselliğini etkileyebilir veya bozabilir.

Glitchler, genellikle devrelerdeki gecikmeler veya zamanlama sorunları nedeniyle oluşur. Örneğin, bir devredeki bir geçişte bir giriş sinyali değiştiğinde, gecikme nedeniyle diğer sinyallerin değişimi biraz gecikir ve bu, kısa süreli bir dalgalanma veya "glitch" olarak ortaya çıkabilir. Bazı durumlarda, glitchler ciddi sorunlara neden olabilir. Örneğin, bir mikroişlemcinin kontrol sinyallerindeki bir glitch, yanlış komutların çalıştırılmasına veya hatta sistem çökmesine neden olabilir. Bu nedenle, elektronik tasarımcılar, devrelerindeki glitchleri minimize etmek için önlemler alırlar. Glitchleri önlemek için kullanılan bazı teknikler şunlardır:

- Glitch giderme devreleri veya filtreler kullanmak
- Gecikme yaratan elemanlar arasında eşit uzunlukta yol oluşturmak
- Zamanlama sorunlarına neden olan giriş sinyallerini azaltmak veya kontrol etmek
- Tasarımı doğru bir şekilde yerleştirmek ve rota etmek

Bu teknikler, bir devrenin daha istikrarlı ve güvenilir bir şekilde çalışmasını sağlayarak, glitchlerin etkisini azaltabilir veya tamamen ortadan kaldırabilir.

2.1.2. Deneyin Yapılışı

$$\text{Denklem 1: } Y = AB'C + C'D$$

a) Doğruluk Tablosu

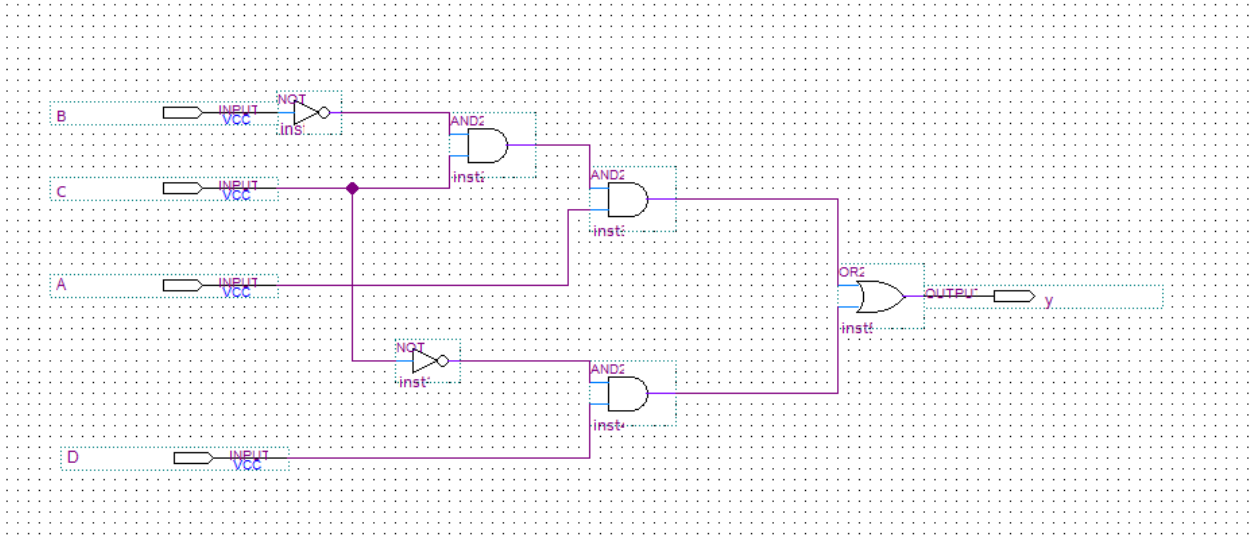
Tablo 1: Problem 1 Denklem 1 için doğruluk tablosu

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

$$Y = \sum_m (1,5,9,10,11,13)$$

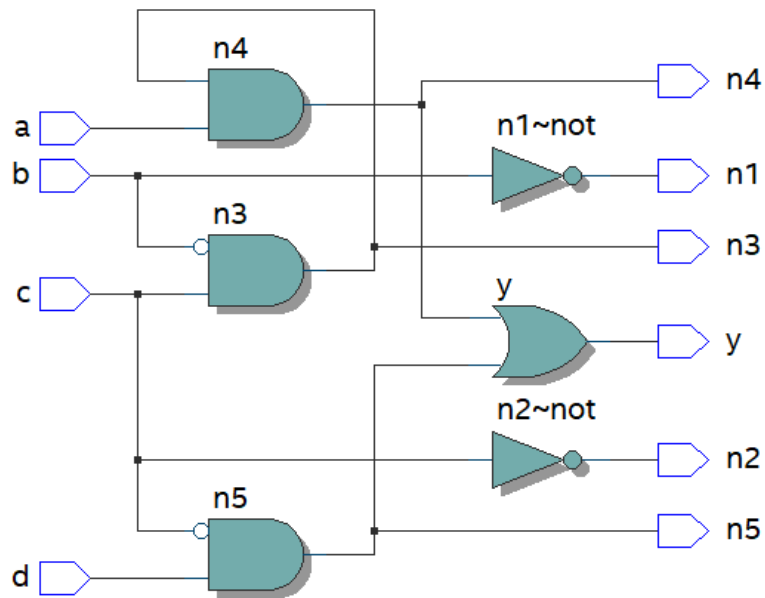
Tablo 1 'de A, B, C, D sütunları A, B, C, D değişkenlerinin alabileceği tüm olasılıkları içermektedir. Y sütunu ise bu olasılıkların Y denklemi için karşılık geldiği sonuçları ifade etmektedir.

b) Devre Şeması



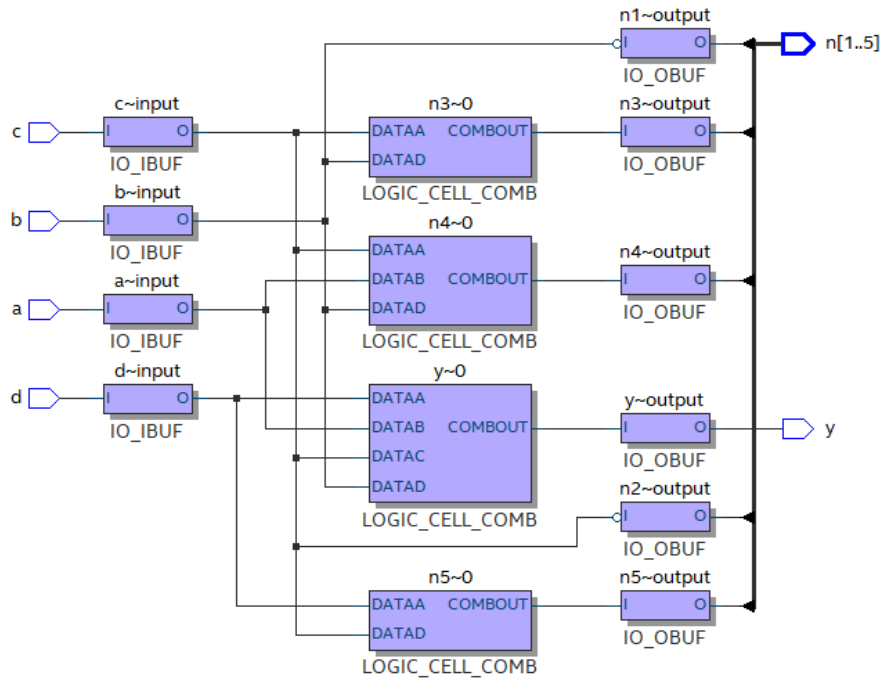
Şema 1: Denklem 1 Devre Şeması

c) Devrenin RTL Şeması



Şema 2: Denklem 1 Devresinin RTL Şeması

d) Eşleştirme Sonrası Devre Şeması



Şema 3: Denklem 2 Devresinin Eşleştirme Sonrası Devre Şeması

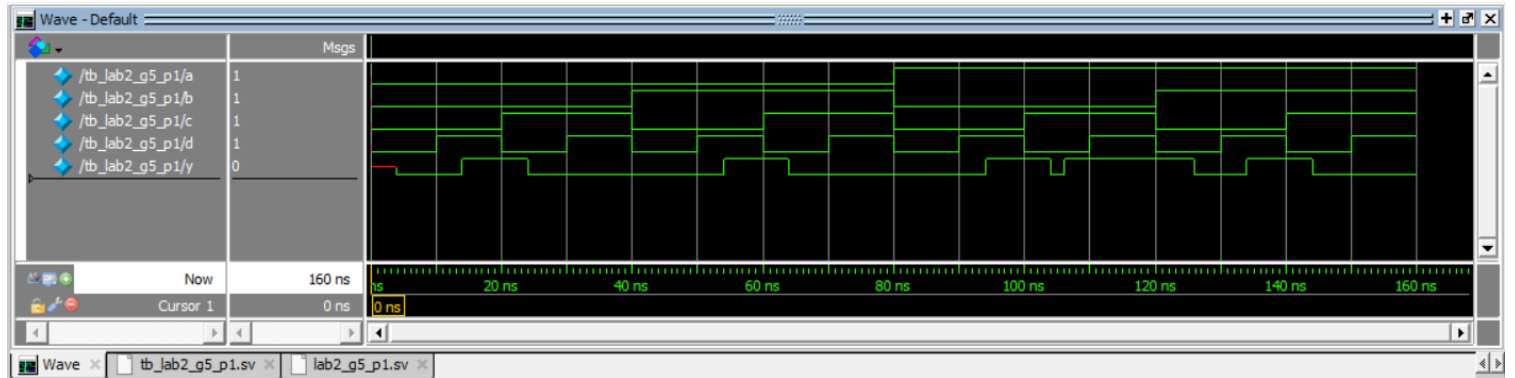
e) Analiz ve Sentez Özeti

Flow Summary	
<<Filter>>	
Flow Status	Successful - Sun May 07 13:30:27 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	lab2_g5_p1
Top-level Entity Name	lab2_g5_p1
Family	MAX 10
Device	10M08DAF484C8G
Timing Models	Final
Total logic elements	5 / 8,064 (< 1 %)
Total registers	0
Total pins	10 / 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 %)

f) Analiz ve Sentez Kaynak Kullanım Özeti

Analysis & Synthesis Resource Usage Summary		
<<Filter>>		
	Resource	Usage
1	Estimated Total logic elements	4
2		
3	Total combinational functions	4
4	▼ Logic element usage by number of LUT inputs	
1	-- 4 input functions	1
2	-- 3 input functions	1
3	-- <=2 input functions	2
5		
6	▼ Logic elements by mode	
1	-- normal mode	4
2	-- arithmetic mode	0
7		
8	▼ Total registers	0
1	-- Dedicated logic registers	0
2	-- I/O registers	0
9		
10	I/O pins	10
11		
12	Embedded Multiplier 9-bit elements	0
13		
14	Maximum fan-out node	c~input
15	Maximum fan-out	5
16	Total fan-out	27
17	Average fan-out	1.13

g) Simülasyon



Grafikte gösterildiği gibi 100ns ve 110ns aralığında glitch gözlenmiştir. Glitch'in gözlenme nedeni tasarlanan devrenin o noktada farklı uzunlukta yolları sahip olmasıdır.

h) Glitch'in Giderilmesi

Denklemler 1 devresinin (2.1.2.a.) doğruluk tablosundan yola çıkılarak oluşturulan Karnaugh haritası aşağıda gösterilmiştir.

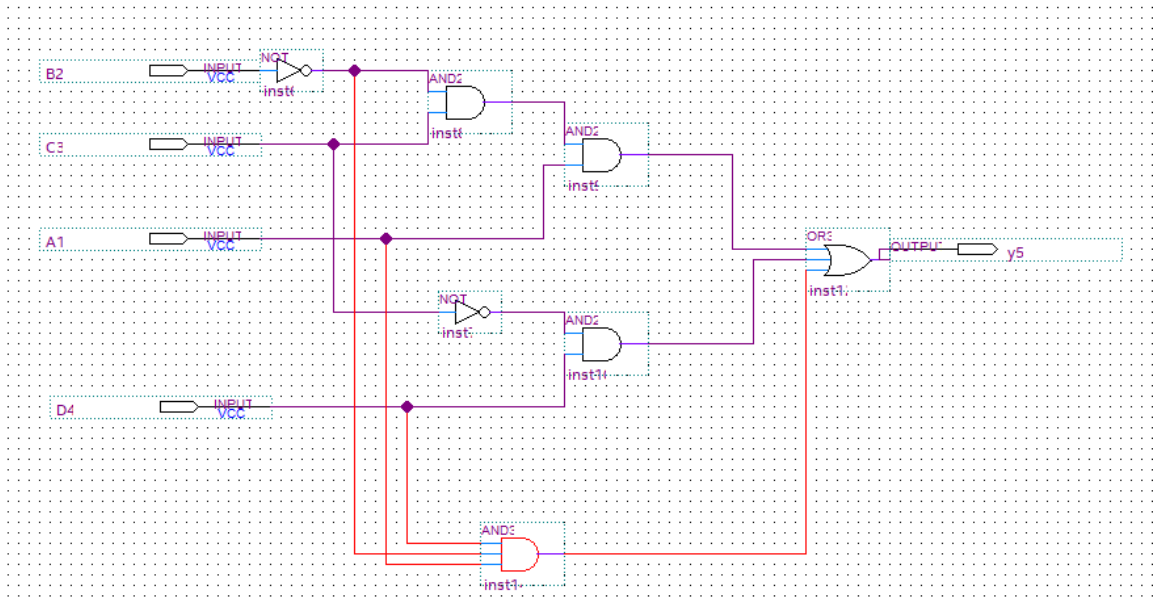
a,b \ c,d	00	01	11	10
00	0	1	0	0
01	0	1	0	0
11	0	1	0	0
10	0	1	1	1

$$AB'D$$

Yukarıdaki Karnaugh haritasından yola çıkarak glitch oluşabilecek durum belirlendi ve bu glitch'in oluşmasını engellemek için denkleme eklenebilecek ifade belirlendi. Denklem 2 glitch giderildikten sonra oluşturulan denklemi ifade etmektedir.

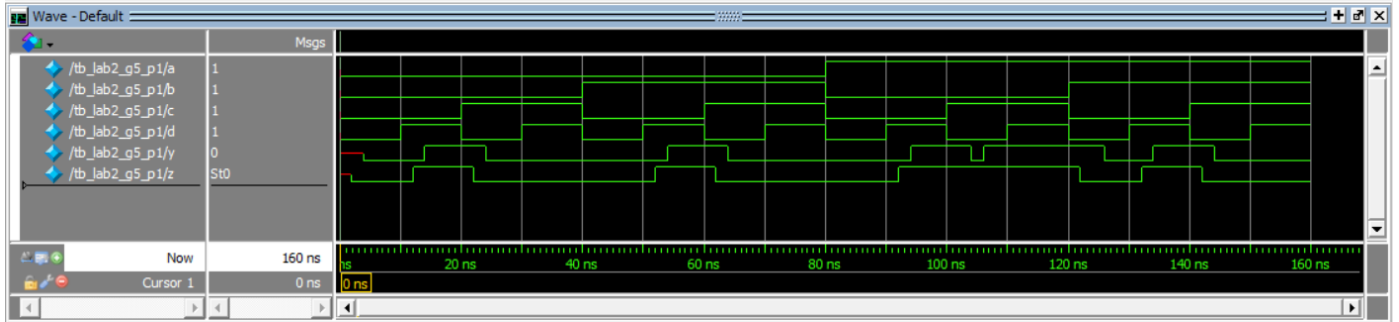
$$\text{Denklem 2: } Z = AB'D + AB'C + C'D$$

Glitch Giderildikten Sonra Oluşan Devre Şeması



Şema 4: Glitch Giderildikten Sonraki Devre Şeması

Glitch Giderildikten Sonra Yapılan Simülasyon



Grafik 2: Glitch Giderildikten Sonra Yapılan Simülasyon

Yukarıda verilen grafik denklem 1 (Y) ve denklem 2 (Z)' ye ait simülasyon sonuçlarını içermektedir. Sonuçlar denklem 1' de Karnough haritası yardımıyla tespit edilen glitch'e sebep olan durum düzeltildikten sonra denklem 2 (Z)'de tekrar glitch oluşmadığı gözlemlenmektedir.

2.2. Problem 2 – Çözücüler (Decoder) ve Uygulamaları

2.2.1. Teorik Araştırma

2.2.1.1. 7BLE (Yedi Segment LED Display Modülü) Nedir?

Yedi Segment LED Display modülü ondalık sayıların gösteriminde kullanılan elektronik bir cihazdır ve içerisinde bulunan yedi adet LED ile aydınlatılan yedi ayrı bölmeden oluşmaktadır. Elemanın yapısında bulunan yedi LED'in her biri bir segment olarak adlandırılır. Bunun sebebi sayısal rakamların kesim formları görüntülenmek üzere parçalar halinde aydınlatılmış olmasıdır. Ayrıca Yedi Segment Display'in yapısında ek olarak bir LED daha bulunmaktadır. Bu LED sayısal göstergelerde küsuratlı sayıların ondalık noktalarının gösterilmesi için kullanılır. Bu ilave LED simülasyonlarda veya elemanın veri sayfasında ondalık nokta anlamında gelen DP (decimal point) ile adlandırılır. Her bir parçanın uçları modülün alt kısmında bulunan ayrı bir bacaklara bağlıdır. Görüntü, bacaklara bir devre bağlanıp LED ya da LCD parçalar üzerinden akım geçmesi sağlanarak üretilir. Yedi segment gösterciler dijital saatlerde, elektronik sayaçlar gibi elektronik cihazlarda kullanılmaktadır.

2.2.1.2. Karnaugh Haritası (KM ya da K-Map) Nedir?

Dijital elektronikte en önemli kurallardan biri az maliyetli devreler oluşturmaktır. Bunu yapabilmek için sadeleştirme yöntemleri kullanılır. Bu yöntemlerinden biri de Karnaugh Haritasıdır. İnsanların örüntü tanıyabilme kabiliyetini kullanarak karışık hesaplamaları sadeleştirirken potansiyel hata durumlarının hızlıca fark edilmesini ve ortadan kaldırılmasını kolaylaştırır. Gerekli Boolean sonuçları iki boyutlu doğruluk tablosundan Karnaugh haritasına aktarılır. Karnaugh haritası, dışındaki hücrelerin Gray kodu ile sıralandığı ve bu hücrelerin yatay-düşey birleşimlerinin temsil ettiği her hücrenin bir giriş durumunun temsil edildiği bir haritadır. Haritadan 1 ve 0'ların standart formlardan

birini oluşturan ideal gruplar belirlenir. Bu gruplar ihtiyaç duyulan Boolean ifadesinin asgari terimle yazılmasında kullanılabilir. Çarpımların toplamı ifadesi her zaman VE kapılarının bir VEYA kapısını beslemesiyle gerçekleştirilir. Toplamların çarpımı ifadesi ise VEYA kapılarının bir VE kapısını beslemesiyle gerçekleştirilir.

2.2.2. Deneyin Yapılışı

a) Doğruluk Tablosu

X ₃	X ₂	X ₁	X ₀	A	B	C	D	E	F	G	ÇIKTI
0	0	0	0	0	0	0	0	0	0	1	-
0	0	0	1	x	x	x	x	x	x	x	x
0	0	1	0	1	0	0	1	1	1	1	E
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	x	x	x	x	x	x	x	x
0	1	0	1	0	0	0	1	1	1	0	L
0	1	1	0	x	x	x	x	x	x	x	x
0	1	1	1	x	x	x	x	x	x	x	x
1	0	0	0	x	x	x	x	x	x	x	x
1	0	0	1	x	x	x	x	x	x	x	x
1	0	1	0	x	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x	x
1	1	0	0	1	0	0	1	1	1	0	C
1	1	0	1	1	1	0	1	1	0	1	2
1	1	1	0	1	0	1	1	0	1	1	5
1	1	1	1	x	x	x	x	x	x	x	x

Tablo 2: 7BLE için Doğruluk Tablosu

Kullanılmayacak gerekli olmayan değerler için don't care (X) ifadeleri kullanıldı.

b) Doğruluk Tablosundaki A B C D E F G Denklemlerinin Sadeleştirilmesi

i) A Sütunu için K-Map Yardımı ile En Sade Boole Cebri İfadesinin Oluşturulması ve

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	0 0	- 1	1 3	1 2
01	- 4	0 5	- 7	- 6
11	1 12	1 13	- 15	1 14
10	- 8	- 9	- 11	- 10

X3

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	0 0	- 1	1 3	1 2
01	- 4	0 5	- 7	- 6
11	1 12	1 13	- 15	1 14
10	- 8	- 9	- 11	- 10

X1

$$A = \sum_m (2, 3, 12, 13, 14) = X_3 + X_1$$

- A denklemini gerçeklemek için toplamda 1 adet OR kapısı kullanıldı.

j) B Sütunu için K-Map Yardımı ile En Sade Boole Cebri İfadesinin Oluşturulması

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	0 0	- 1	1 3	0 2
01	- 4	0 5	- 7	- 6
11	0 12	1 13	- 15	0 14
10	- 8	- 9	- 11	- 10

X3X0

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	0 0	- 1	1 3	0 2
01	- 4	0 5	- 7	- 6
11	0 12	1 13	- 15	0 14
10	- 8	- 9	- 11	- 10

X3X0

$$B = \sum_m (3, 13) = X_3 X_0 + X_1 X_0$$

- B denklemini gerçeklemek için toplamda 1 adet OR, 2 adet AND kapısı kullanıldı.

k) C Sütunu için K-Map Yardımı ile En Sade Boole Cebri İfadesinin Oluşturulması

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	0	-	1	0
01	-	0	-	-
11	0	0	-	1
10	-	-	-	-

$X_2 X_1$

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	0	-	1	0
01	-	0	-	-
11	0	0	-	1
10	-	-	-	-

$X_1 X_0$

$$C = \sum_m (3, 14) = X_2 X_1 + X_1 X_0$$

- C denklemini gerçeklemek için toplamda 1 adet OR kapısı, 2 adet AND kapısı kullanıldı.

l) D Sütunu için K-Map Yardımı ile En Sade Boole Cebri İfadesinin Oluşturulması

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	0	-	1	1
01	-	1	-	-
11	1	1	-	1
10	-	-	-	-

X_2

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	0	-	1	1
01	-	1	-	-
11	1	1	-	1
10	-	-	-	-

X_1

$$D = \sum_m (2, 3, 5, 12, 13, 14) = X_2 + X_1$$

- D denklemini gerçeklemek için toplamda 1 adet OR kapısı kullanıldı.

m) E Sütunu için K-Map Yardımı ile En Sade Boole Cebri İfadesinin Oluşturulması

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	0 0	- 1	0 3	1 2
01	- 4	1 5	- 7	- 6
11	1 12	1 13	- 15	0 14
10	- 8	- 9	- 11	- 10

$$X_2 X_1'$$

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	0 0	- 1	0 3	1 2
01	- 4	1 5	- 7	- 6
11	1 12	1 13	- 15	0 14
10	- 8	- 9	- 11	- 10

$$X_3' X_1 X_0'$$

$$E = \sum_m (2, 5, 12, 13) = X_2 X_1' + X_3' X_1 X_0'$$

- E denklemini gerçeklemek için toplamda 1 adet OR kapısı, 3 adet AND kapısı ve 3 adet NOT kapısı kullanıldı.

n) F Sütunu için K-Map Yardımı ile En Sade Boole Cebri İfadesinin Oluşturulması

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	0 0	- 1	0 3	1 2
01	- 4	1 5	- 7	- 6
11	1 12	0 13	- 15	1 14
10	- 8	- 9	- 11	- 10

$$X_2 X_0'$$

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	0 0	- 1	0 3	1 2
01	- 4	1 5	- 7	- 6
11	1 12	0 13	- 15	1 14
10	- 8	- 9	- 11	- 10

$$X_1 X_0'$$

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	0 0	- 1	0 3	1 2
01	- 4	1 5	- 7	- 6
11	1 12	0 13	- 15	1 14
10	- 8	- 9	- 11	- 10

$$X_3' X_1' X_0$$

$$F = \sum_m (2, 5, 12, 14) = X_2 X_0' + X_1 X_0' + X_3' X_1' X_0$$

- F denklemini gerçeklemek için toplamda 2 adet OR kapısı, 4 adet AND kapısı ve 4 adet NOT kapısı kullanıldı.

o) G Sütunu için K-Map Yardımı ile En Sade Boole Cebri İfadesinin Oluşturulması

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	1 0	- 1	1 3	1 2
01	- 4	0 5	- 7	- 6
11	0 12	1 13	- 15	1 14
10	- 8	- 9	- 11	- 10

X_2'

$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	1 0	- 1	1 3	1 2
01	- 4	0 5	- 7	- 6
11	0 12	1 13	- 15	1 14
10	- 8	- 9	- 11	- 10

X_1

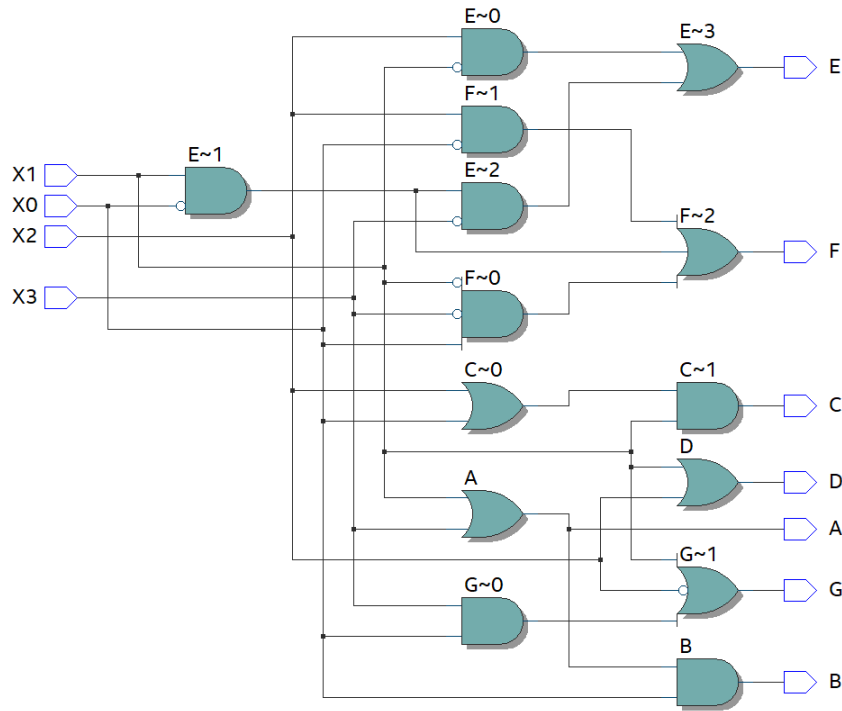
$x_3, x_2 \backslash x_1, x_0$	00	01	11	10
00	1 0	- 1	1 3	1 2
01	- 4	0 5	- 7	- 6
11	0 12	1 13	- 15	1 14
10	- 8	- 9	- 11	- 10

$$G = \sum_m (0,2,3,13,14) = X_2' + X_1 + X_3X_0$$

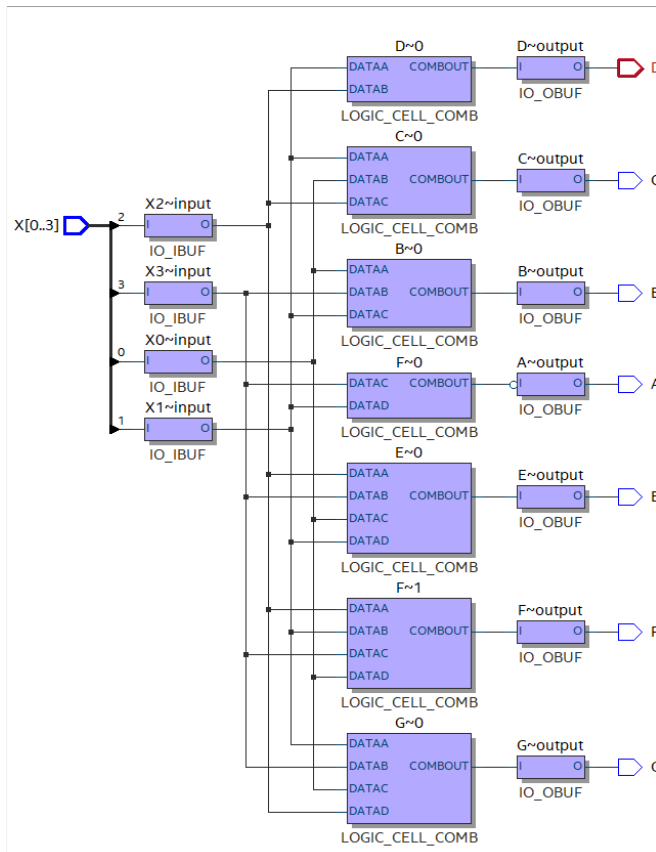
X_3X_0

- G denklemini gerçeklemek için toplamda 2 adet OR kapısı, 1 adet NOT kapısı, 1 adet AND kapısı kullanıldı.

c) Devrelerin RTL Şeması



d) Eşleştirme Sonrası Devre Şeması



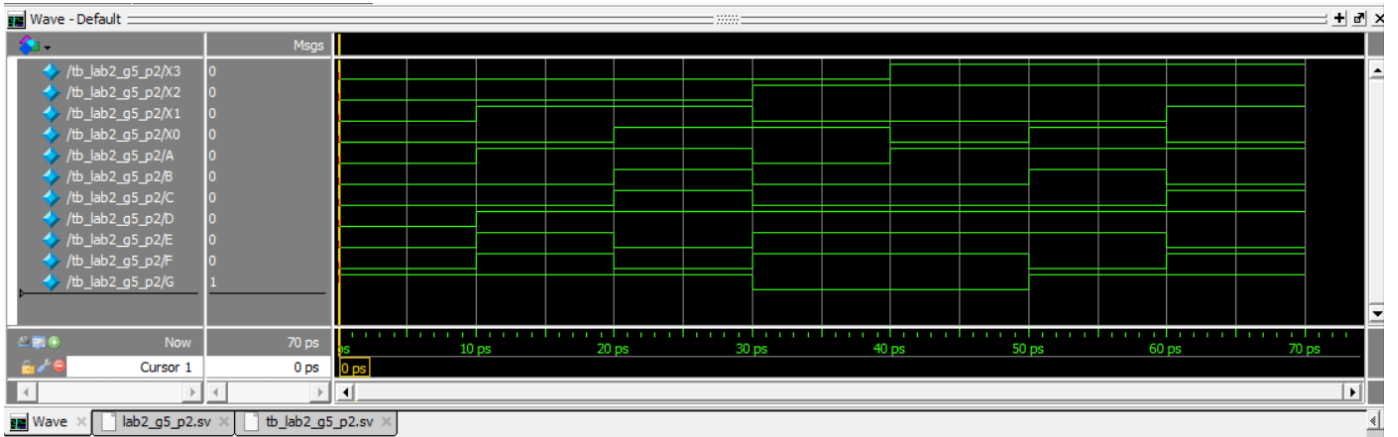
e) Analiz ve Sentez Özeti

Flow Summary	
<<Filter>>	
Flow Status	Successful - Sun May 07 17:22:25 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	lab2_g5_p2
Top-level Entity Name	lab2_g5_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 %)

f) Analiz ve Sentez Kaynak Kullanım 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	X1~...put
15	Maximum fan-out	7
16	Total fan-out	40
17	Average fan-out	1.38

g) Simülasyon



Grafik 3: A B C D E F G Denklemlerinin gerçekleşmesi

Grafik 3. te X_3, X_2, X_1, X_0 tüm olasılıkları için A, B, C, D, E, F, G denklemleri gerçekleştirildi. Her 10ps'lik aralık 7BLE seven segment display de görüntülenmek istenen değer için gerekli bool ifadeleri içermektedir. Örneğin 10ps ve 20ps arası E harfi seven segment display de görüntülenir. Çünkü A, B, C, D, E, F, G sırasıyla 1001111 değerini alır ve 1 değerini alan bölmelerin ledleri yanar sonuç olarak E harfi görüntülenir.

Referanslar:

<https://www.elektrikport.com/makale-detay/7-segment-display-nedir/18475#ad-image-0>

https://tr.wikipedia.org/wiki/Yedi_segment_LED_g%C3%B6r%C3%BCnt%C3%BC_mod%C3%BCI%C3%BC#:~:text=Yedi%20segment%20LED%20g%C3%B6r%C3%BCnt%C3%BC%20mod%C3%BCI%C3%BC%20ondal%C4%B1k%20say%C4%B1lar%C4%B1n%20g%C3%B6steriminde%20kullan%C4%B1lan%20elektronik,g%C3%B6r%C3%BCnt%C3%BClemek%20i%C3%A7in%20yayg%C4%B1n%20olarak%20kullan%C4%B1lmaktadır

https://tr.wikipedia.org/wiki/Karnaugh_haritas%C4%B1

<https://teknolojiprojeleri.com/elektronik/karno-karnaugh-haritasi-nedir>

EKLER

1. Problem 1 SystemVerilog DTD tasarımı ve test tezgahı

```
/* lab2_g5_p1.sv
*
* Hazirlayanlar:
* Sule Nur Demirdas
* Aysu Akturk
*
* Notlar:
* ELM235 2023 Bahar Lab2 - Problem 1
*  $Y = A \text{ and NOT } B \text{ and } C \text{ or NOT } C \text{ and } D$  denkleminin gereklemesi
* Ama: Devrede oluřan gecikmeleri ve glicthleri gozlemlemek
*
*/

`timescale 1ns/1ns

module lab2_g5_p1(
    input logic a, b, c,d,
    output logic n1, n2 , n3, n4, n5 , y, z
);

assign #2 n1 = ~b;
assign #2 n2 = ~c;
assign #2 n3 = n1 & c;
assign #2 n4 = n3 & a;
assign #2 n5 = n2 & d;
assign #2 y = n4 | n5;

assign #2 z = n5 | a & n1 & d | n4; // Glicth giderilmesi iin
oluřturulan denklem

endmodule
```

```

/* tb_lab2_g5_p1.sv
*
* Hazirlayanlar:
* Sule Nur Demirdas
* Aysu Akturk
*
* Notlar:
* ELM235 2023 Bahar Lab2 - Problem 1 Testbench
*  $Y = A \text{ and NOT } B \text{ and } C \text{ or NOT } C \text{ and } D$  denkleminin simulasyonu
* Bütün olası girişlere göre çıkış gözlemlenir.
*
*/

`timescale 1ns/1ns

module tb_lab2_g5_p1();

logic a, b, c, d; // Test tezgahi giri? de?i?kenleri tanimlari
logic n1, n2, n3, n4, n5, y; // Test tezgahi cikis de?iskenleri
tanimlari

lab2_g5_p1 dut0(a, b, c, d, n1, n2, n3, n4, n5, y, z );

initial begin

    // Truth table daki tum durumlar tanimlanir

    a = 0; b = 0; c = 0; d = 0; #10 // Degiskenlere degerler
    atandiktan sonra 10 us bekle

    a = 0; b = 0; c = 0; d = 1; #10

    a = 0; b = 0; c = 1; d = 0; #10

    a = 0; b = 0; c = 1; d = 1; #10

```

```
a = 0; b = 1; c = 0; d = 0; #10
a = 0; b = 1; c = 0; d = 1; #10
a = 0; b = 1; c = 1; d = 0; #10
a = 0; b = 1; c = 1; d = 1; #10
a = 1; b = 0; c = 0; d = 0; #10
a = 1; b = 0; c = 0; d = 1; #10
a = 1; b = 0; c = 1; d = 0; #10
a = 1; b = 0; c = 1; d = 1; #10
a = 1; b = 1; c = 0; d = 0; #10
a = 1; b = 1; c = 0; d = 1; #10
a = 1; b = 1; c = 1; d = 0; #10
a = 1; b = 1; c = 1; d = 1; #10
```

```
$stop;
```

```
end
```

```
endmodule
```

2. Problem 2 SystemVerilog DTD tasarımı ve test tezgahı

```
/* lab2_g5_p2.sv
*
* Hazirlayanlar:
* Sule Nur Demirdas
* Aysu Akturk
*
* Notlar:
* ELM235 2023 Bahar Lab2 - Problem 2
* 7BLE Seven segment display de istenen deęerlerin gozlenmesi icin
uygulanan
*denklemlerin geręeklenmesi
*/
module lab2_g5_p2(
    input logic X3, X2, X1, X0,
    output logic A, B,C,D,E,F, G
);

assign A = X3 | X1;
assign B = X0 & (X3 | X1);
assign C = X1 & (X2 | X0);
assign D = X1 | X2 ;
assign E = X2 & ~X1 | ~X0 & X1 & ~X3;
assign F = ~X0 & X1 | ~X3 & ~X1 & X0 | ~X0&X2;
assign G = ~X2 | X1 | X0 & X3;

endmodule
```

```

/* tb_lab2_g5_p2.sv
*
* Hazirlayanlar:
* Sule Nur Demirdas
* Aysu Akturk
*
* Notlar:
* ELM235 2023 Bahar Lab2 - Problem 2 Testbench
* Bütün olası giri?lere göre ç?k?? gözlemlenir.
*  $A = X3 \mid X1,$ 
*  $B = X0 \ \& \ (X3 \mid X1),$ 
*  $C = X1 \ \& \ (X2 \mid X0),$ 
*  $D = X1 \mid X2,$ 
*  $E = X2 \ \& \ \sim X1 \mid \sim X0 \ \& \ X1 \ \& \ \sim X3,$ 
*  $F = \sim X0 \ \& \ X1 \mid \sim X3 \ \& \ \sim X1 \ \& \ X0 \mid \sim X0 \ \& \ X2,$ 
*  $G = \sim X2 \mid X1 \mid X0 \ \& \ X3$ 
* denklemlerinin gerçekleşmesi
*/
module tb_lab2_g5_p2();

logic X3, X2, X1, X0; // Test tezgahi giri? de?i?kenleri tanımları
logic A, B, C, D, E, F, G; // Test tezgahi çıkış de?iskenleri tanımları

lab2_g5_p2 dut0(X3, X2, X1, X0, A, B, C, D, E, F, G);

initial begin

    X3 = 0; X2 = 0; X1 = 0; X0 = 0; #10

```

```
x3 = 0; x2 = 0; x1 = 1; x0 = 0; #10  
x3 = 0; x2 = 0; x1 = 1; x0 = 1; #10  
x3 = 0; x2 = 1; x1 = 0; x0 = 1; #10  
x3 = 1; x2 = 1; x1 = 0; x0 = 0; #10  
x3 = 1; x2 = 1; x1 = 0; x0 = 1; #10  
x3 = 1; x2 = 1; x1 = 1; x0 = 0; #10
```

```
$stop;
```

```
end
```

```
endmodule
```