



GEBZE TEKNİK ÜNİVERSİTESİ
ELEKTRONİK MÜHENDİSLİĞİ

ELM235

LOJİK DEVRE TASARIM LABORATUVARI

LAB 3 Deney Raporu

Lojik Devreler ve Tasarım Laboratuvarı Dersine Hazırlık

Hazırlayanlar
1) 210102002053 – Şule Nur Demirdağ

1. GİRİŞ

Bu rapor donanım tanımlama dillerini (DTD) kullanarak devre tasarımı yapmak, sentezleyici araçlar kullanarak DTD ile tanımlanan devreleri FBGA için sentezlemek, yalnızca NAND kapıları kullanarak decoder ve multiplexer tasarımı gerçekleştirmek gibi amaçlar içermektedir.

2. PROBLEMLER

2.1. Problem 1 - $2^N \times 1$ Çoğullayıcı (MUX) Tasarımı

2.1.1. Teorik Araştırma

2.1.1.1. Multiplexer (Çoğullayıcı) Nedir?

Multiplexer (MUX), birden fazla giriş sinyalinden yalnızca bir tanesini seçerek çıkışa ileten bir elektronik devredir. Bu işlev nedeniyle MUX, "veri seçici" veya "veri yönlendirici" olarak da adlandırılır.

MUX, özellikle veri iletimi ve işlemede yaygın olarak kullanılır. Örneğin, bir MUX, bir bilgisayarın CPU'sundan veri yolu aracılığıyla gelen verileri RAM veya diğer donanım birimlerine yönlendirmek için kullanılabilir.

Bir MUX, tipik olarak bir kontrol sinyali ile birlikte birden fazla veri girişi ve tek bir çıkışa sahiptir. Kontrol sinyali, hangi girişin çıkışa yönlendirileceğini belirler. Örneğin, iki girişli bir MUX'ta, kontrol sinyali "0" olduğunda ilk giriş çıkışa yönlendirilirken, kontrol sinyali "1" olduğunda ikinci giriş çıkışa yönlendirilir.

MUX, diğer mantık kapıları ile birleştirilerek daha karmaşık devrelerin tasarlanmasında da kullanılabilir.

2.1.2. Deneyin Yapılışı

a) Doğruluk Tablosu

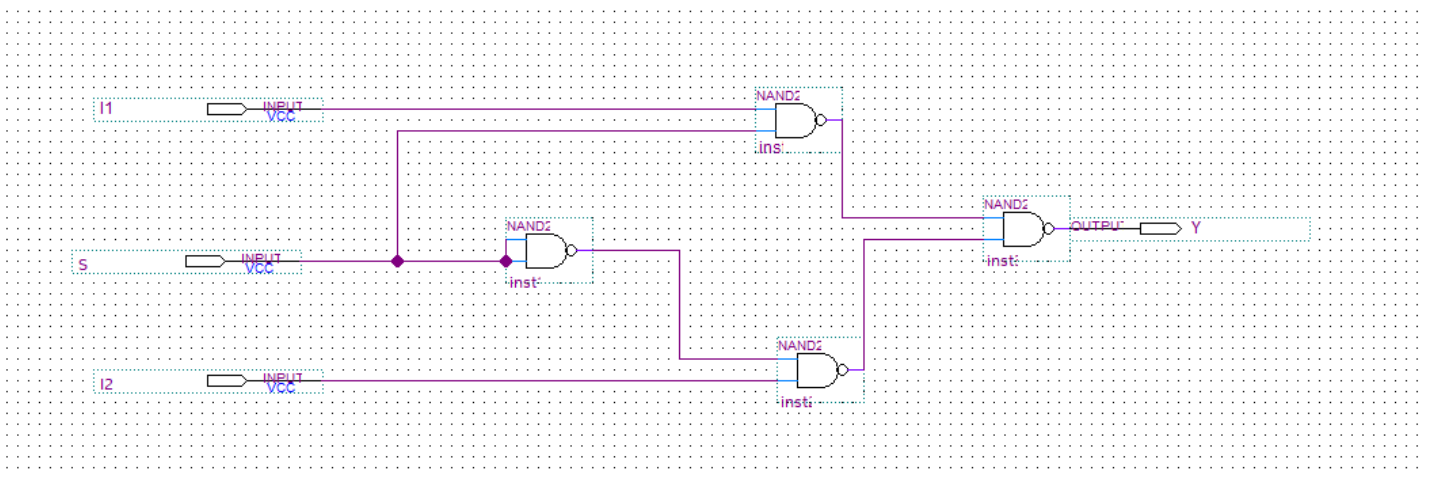
Tablo 1: 2x1 MUX Doğruluk Tablosu

I ₂	I ₁	S	Y
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

Doğruluk tablosundan yola çıkarak mintermlerini bulup tablonun denklemi çıkarıldı.

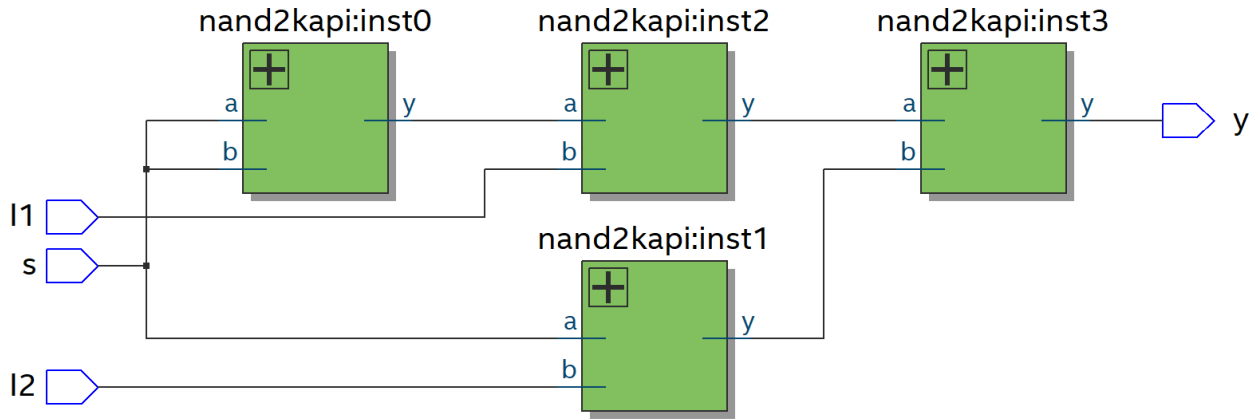
$$\text{Denklem 1: } Y = S'I_1 + SI_2$$

b) Devre Şeması



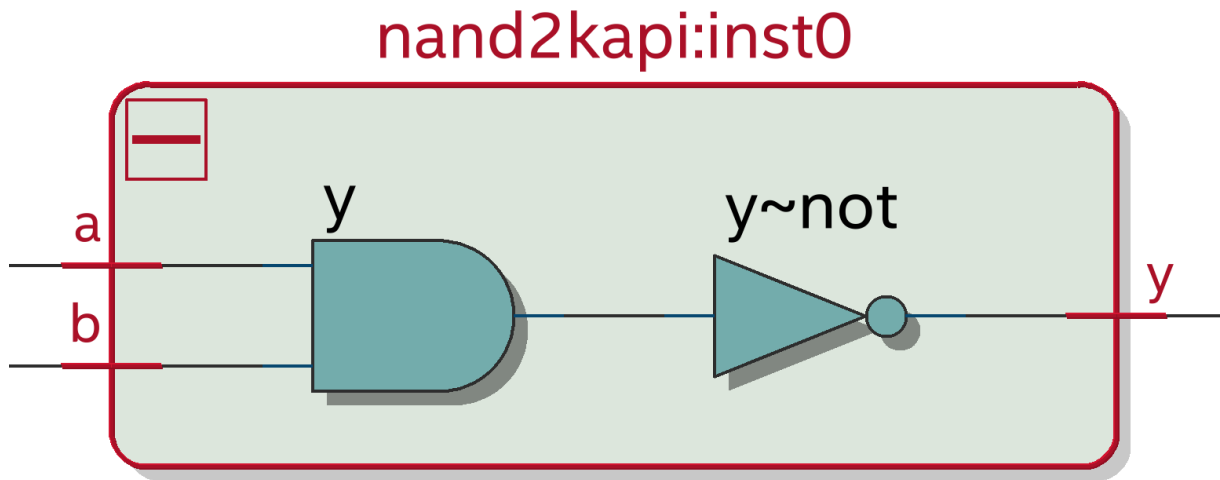
Şema 1: 2x1 MUX Devre Şeması

c) Devrenin RTL Şeması



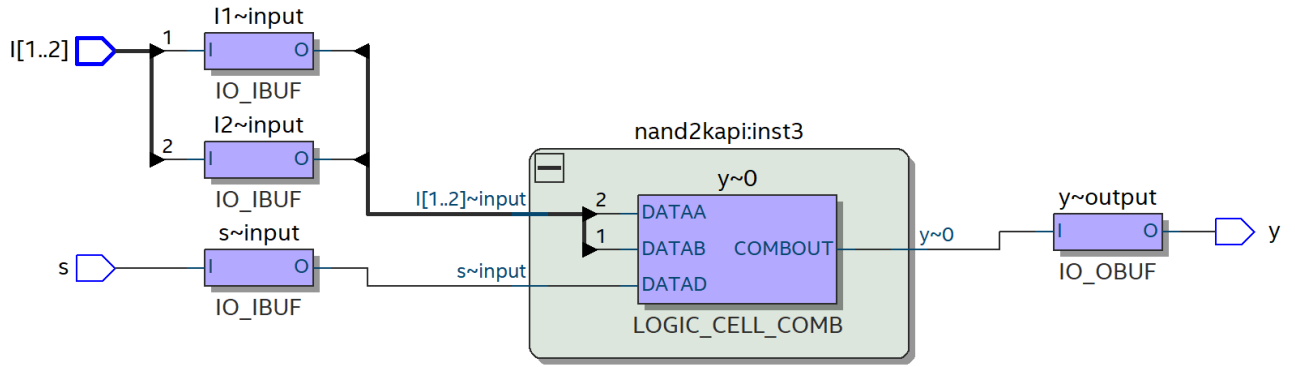
Şema 2: 2x1 MUX RTL Şeması

Yukarıda şemada gösterilen örneğin nand2kapi:inst0 bir NAND kapısını ifade etmektedir. Structural tarzda modüller kullanılarak oluşturulduğu için RTL şemasında bu şekilde gözlemlendi. Aşağıda Şema 3'te nand2kapi:inst0 NAND kapısı gösterilmiştir.



Şema 3: nand2kapi:inst0 adlı NAND kapısının ayrıntılı incelenmesi

d) Eşleştirme Sonrası Devre Şeması



Şema 4: 2x1 MUX Eşleştirme Sonrası Devre Şeması

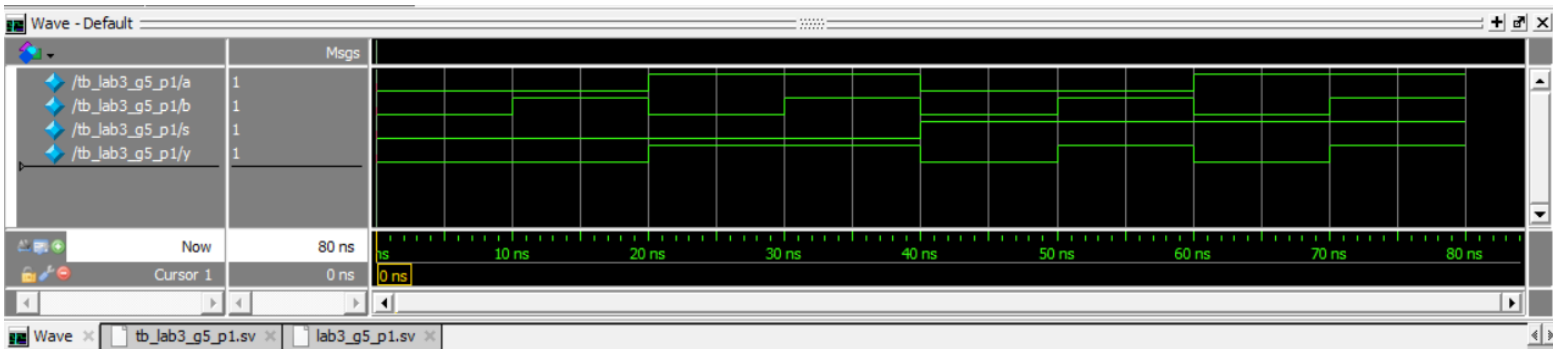
e) Analiz ve Sentez Özeti

Flow Summary	
<<Filter>>	
Flow Status	Successful - Sun May 14 20:58:37 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	lab3_g5_p1
Top-level Entity Name	lab3_g5_p1
Family	MAX 10
Device	10M08DAF484C8G
Timing Models	Final
Total logic elements	2 / 8,064 (< 1 %)
Total registers	0
Total pins	4 / 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 %)

f) Analiz ve Sentez Kaynak Kullanım 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	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	na...~0
15	Maximum fan-out	1
16	Total fan-out	8
17	Average fan-out	0.89

g) Simülasyon



A, I_1 'yi B I_2 'yi temsil etmektedir.

2.2. Problem 2 - 2N x 1 Çoğullayıcı (MUX) ile Temel Lojik Kapıların ve Büyük Çoğullayıcıların Tasarımı

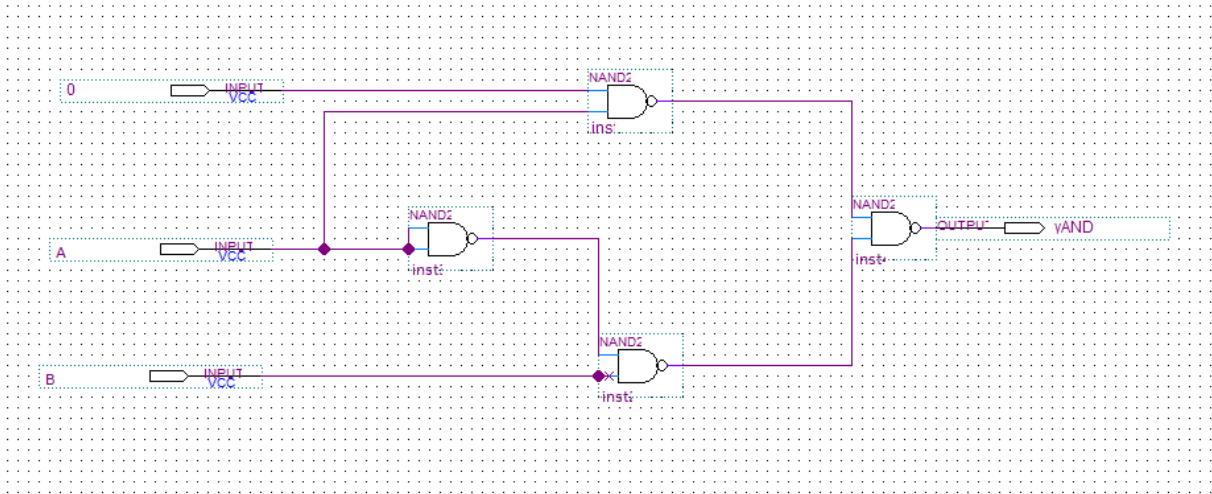
2.2.1. Teorik İşlemler ve Devre Şemaları

a) AND Kapısı

Denklem 1' deki ifadeyi kullanarak kolayca 2x1 MUX kullanarak AND kapısını oluşturabiliriz.

$$\text{Denklem 1: } Y = S'A + SB$$

AND kapısı sonucu elde edilmesi için $Y = AB$ olmalıdır. Bunun için yukarıdaki denklemde S yerine A, A yerine 0, B yerine ise yine B konulması gerekir.



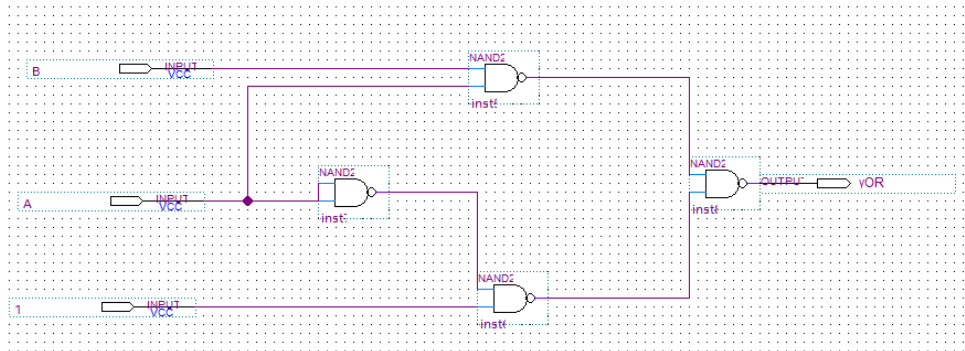
Şema 5: AND kapısı devre şeması

b) OR Kapısı

Denklem 1' deki ifadeyi kullanarak kolayca 2x1 MUX kullanarak OR kapısını oluşturabiliriz.

$$\text{Denklem 1: } Y = S'A + SB$$

OR kapısı sonucu elde edilmesi için $Y = A + B$ olmalıdır. Bunun için yukarıdaki denklemde S yerine A, A yerine B, B yerine ise yine 1 konulması gerekir.



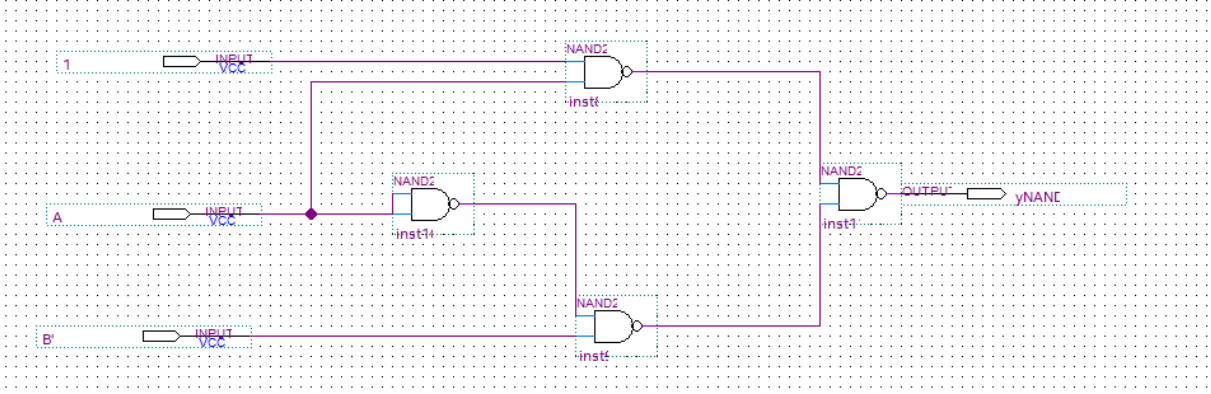
Şema 6: OR Kapısı Devre Şeması

c) NAND Kapısı

Denklem 1' deki ifadeyi kullanarak kolayca 2x1 MUX kullanarak NAND kapısını oluşturabiliriz.

$$\text{Denklem 1: } Y = S'A + SB$$

NAND kapısı sonucu elde edilmesi için $Y = A' + B'$ olmalıdır. Bunun için yukarıdaki denklemde S yerine A, A yerine 1, B yerine ise yine B' konulması gerekir.



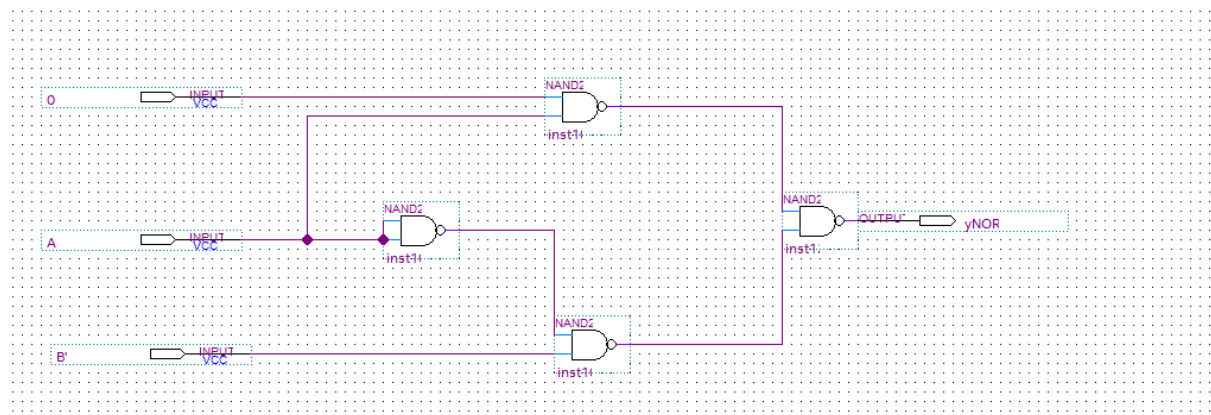
Şema 7: NAND Kapısı devre şeması

d) NOR Kapısı

Denklem 1' deki ifadeyi kullanarak kolayca 2x1 MUX kullanarak NAND kapısını oluşturabiliriz.

$$\text{Denklem 1: } Y = S'A + SB$$

NOR kapısı sonucu elde edilmesi için $Y = A'B'$ olmalıdır. Bunun için yukarıdaki denklemde S yerine A, A yerine B', B yerine ise yine 0 konulması gerekir.



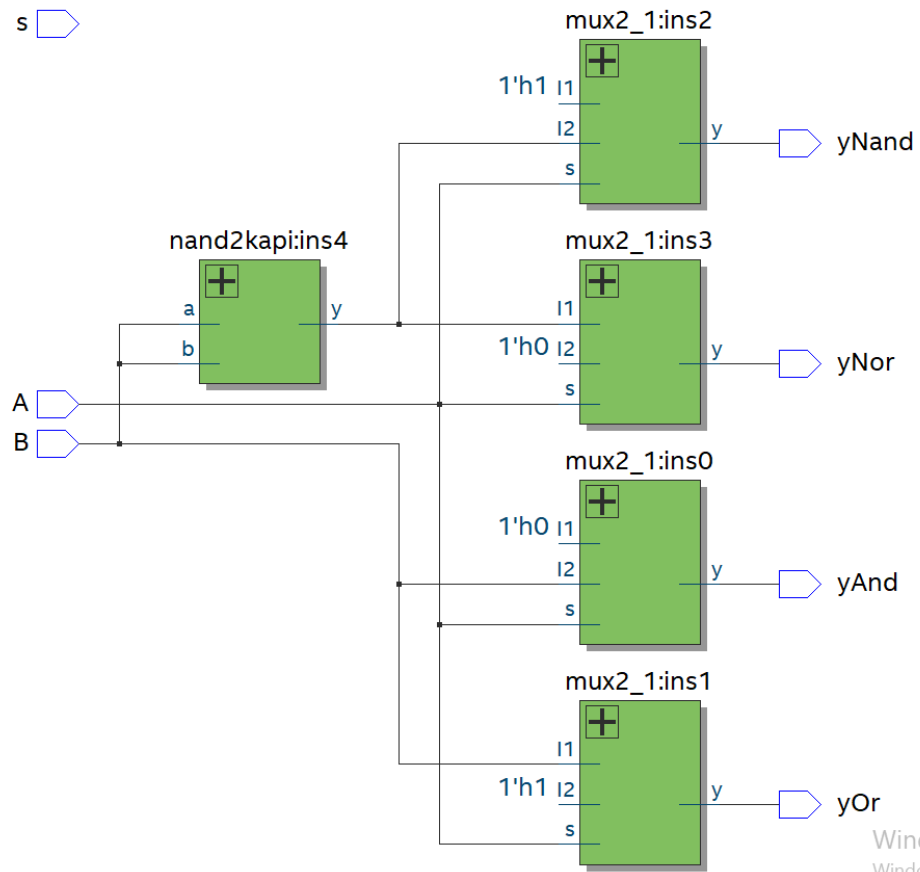
Şema 8: NOR Kapısı devre şeması

e) Simülasyon



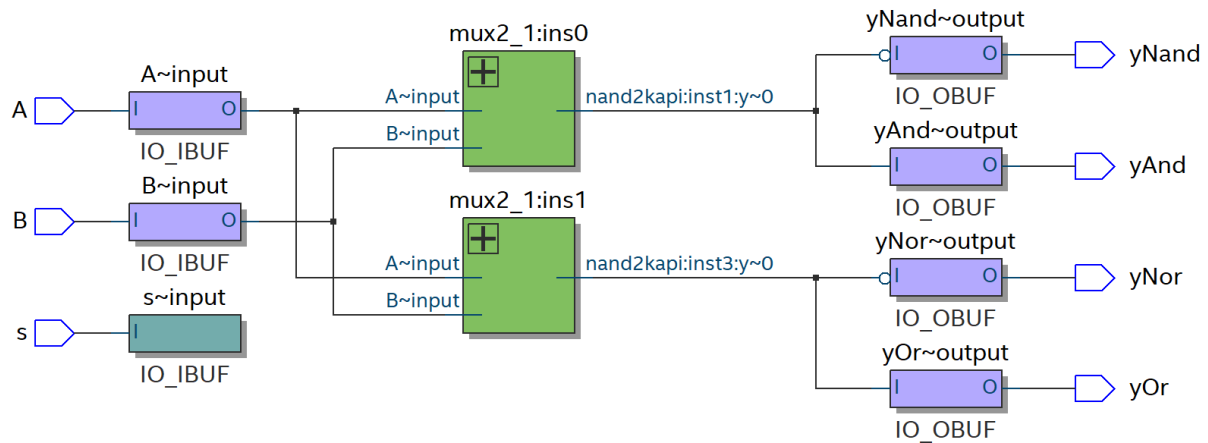
Grafik 1: Simülasyon sonuçları elde edilen kare dalga grafiği

f) Devrenin RTL Şeması




Windows'u
Windows'u etk

g) Eşleştirme Sonrası Devre Şeması



h) Analiz ve Sentez Özeti

Flow Summary	
 <<Filter>>	
Flow Status	Successful - Sun May 14 21:56:29 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	lab3_g5_p2
Top-level Entity Name	lab3_g5_p2
Family	MAX 10
Device	10M08DAF484C8G
Timing Models	Final
Total logic elements	3 / 8,064 (< 1 %)
Total registers	0
Total pins	7 / 250 (3 %)
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 %)

i) Analiz ve Sentez Kaynak Kullanım Özeti

Analysis & Synthesis Resource Usage Summary		
<<Filter>>		
	Resource	Usage
1	Estimated Total logic elements	2
2		
3	Total combinational functions	2
4	▼ Logic element usage by number of LUT inputs	
1	-- 4 input functions	0
2	-- 3 input functions	0
3	-- <=2 input functions	2
5		
6	▼ Logic elements by mode	
1	-- normal mode	2
2	-- arithmetic mode	0
7		
8	▼ Total registers	0
1	-- Dedicated logic registers	0
2	-- I/O registers	0
9		
10	I/O pins	7
11		
12	Embedded Multiplier 9-bit elements	0
13		
14	Maximum fan-out node	mu...~0
15	Maximum fan-out	2
16	Total fan-out	15
17	Average fan-out	0.94

2.3. Problem 3 - 2 x 4 Kod Çözücü (Decoder) Tasarımı

2.3.1. Teorik Araştırma

2.3.1.1. Decoder Nedir?

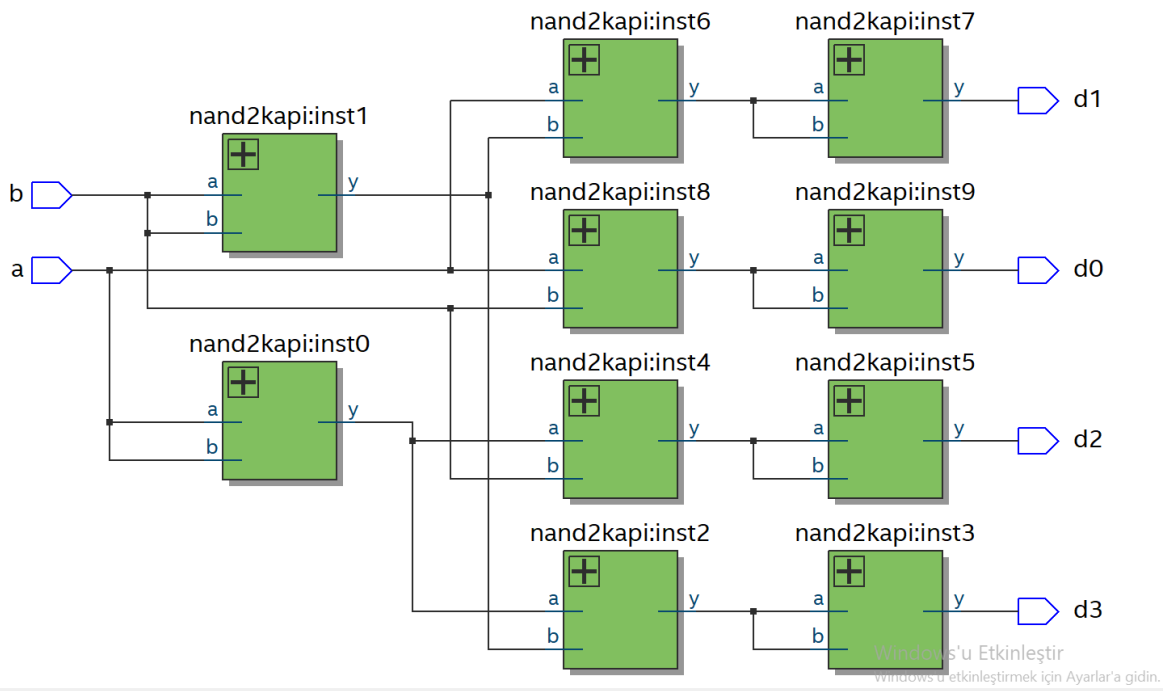
Decoder, birden çok giriş sinyalini alarak, her bir kombinasyon için bir çıkış sinyali üreten bir elektronik devredir. Decoder, "çözücü" veya "çözücü/dönüştürücü" olarak da adlandırılır.

Decoder, dijital sistemlerde kullanılan anahtarlama devrelerinde yaygın olarak kullanılır. Bir Decoder, genellikle bir adres çizgisi üzerindeki bilgiyi çözümlmek için kullanılır. Örneğin, bellek adresi için bir Decoder, bellek sisteminin gerekli alanına doğru verilerin yönlendirilmesini sağlar.

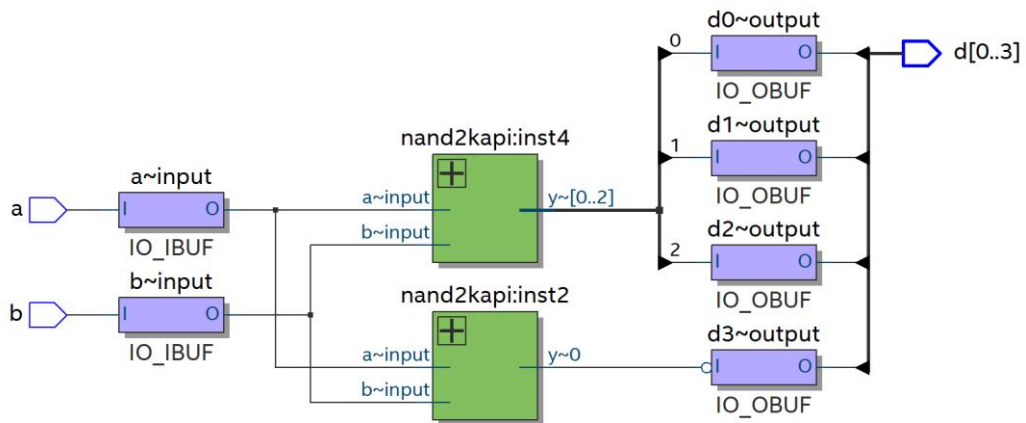
Decoder, doğrudan doğruya binary bir veri girişini kabul ederek, örneğin 2^N adet çıkışa sahip olacak şekilde tasarlanabilir. Bu durumda, her bir çıkışa yalnızca bir tek giriş kombinasyonu denk gelecektir. Decoder, genellikle AND mantık kapıları ve invertörlerden oluşur.

Decoder, diğer mantık kapıları ile birleştirilerek daha karmaşık devrelerin tasarlanmasında da kullanılabilir. Örneğin, bir Decoder ve bir MUX'un birleştirilmesiyle, önceden belirlenmiş koşulların karşılandığı durumlarda farklı veri kaynaklarından veri seçimi yapmak için bir seçim devresi oluşturulabilir.

a) RTL Devre Şeması



b) Eşleştirme Sonrası Devre Şeması









c) Analiz ve Sentez Özeti

Flow Summary	
<<Filter>>	
Flow Status	Successful - Sun May 14 22:26:04 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	lab3_g5_p3
Top-level Entity Name	lab3_g5_p3
Family	MAX 10
Device	10M08DAF484C8G
Timing Models	Final
Total logic elements	4
Total registers	0
Total pins	6
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0

d) 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	0
2	-- 3 input functions	0
3	-- <=2 input functions	4
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	6
11		
12	Embedded Multiplier 9-bit elements	0
13		
14	Maximum fan-out node	a~input
15	Maximum fan-out	4
16	Total fan-out	18
17	Average fan-out	1.13

e) Simülasyon

 /tb_lab3_g5_p3/a	0				
 /tb_lab3_g5_p3/b	0				
 /tb_lab3_g5_p3/d0	0				
 /tb_lab3_g5_p3/d1	0				
 /tb_lab3_g5_p3/d2	0				
 /tb_lab3_g5_p3/d3	1				

E. EKLER: Problem 1

```
`timescale 1ns/1ns

module nand2kapi(
    input logic a,b,
    output logic y
);

    assign y = ~( a & b );

endmodule

/* lab2_g5_p1.sv
*
* Hazirlayanlar:
* Sule Nur Demirdas
*
* Notlar:
* ELM235 2023 Bahar Lab2 - Problem 1
* 2x1 MUX denkleminin gerçeklemesi
*
*/

`timescale 1ns/1ns

module lab3_g5_p1(
```

```

        input logic I1, I2, s,
        output logic y

    );

/*
    assign nand1 = ~(s & s);
    assign nand2 = ~(s & I2);
    assign nand3 = ~(nand1 & I1);
    assign y = ~(nand3 & nand2);
*/

logic n2,n3,n4;

nand2kapi inst0(s,s,n2);
nand2kapi inst1(s,I2,n3);
nand2kapi inst2(n2,I1,n4);
nand2kapi inst3(n4,n3,y);

endmodule

/* tb_lab2_g5_p1.sv
*
* Hazirlayanlar:
* Sule Nur Demirdas
*
* Notlar:
* ELM235 2023 Bahar Lab3 - Problem 1 Testbench
* 2x1 MUX gerçeklemesi
*
*/

```



```

`timescale 1ns/1ns

module tb_lab3_g5_p1();

logic a, b, s; // Test tezgahi giri? de?i?kenleri tanimlari
logic y; // Test tezgahi cikis de?iskenleri tanimlari

lab3_g5_p1 dut0(a, b, s, y);

initial begin

    // Truth table daki tum durumlar tanimlanir

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

    a = 0; b = 1; s = 0; #10
    a = 1; b = 0; s = 0; #10
    a = 1; b = 1; s = 0; #10
    a = 0; b = 0; s = 1; #10
    a = 0; b = 1; s = 1; #10
    a = 1; b = 0; s = 1; #10
    a = 1; b = 1; s = 1; #10

    $stop;

end

endmodule

```

F. EKLER Problem 2

```
`timescale 1ns/1ns
```

```
module nand2kapi(
```

```
    input logic a,b,
```

```
    output logic y
```

```
);
```

```
    assign y = ~( a & b );
```

```
endmodule
```

```
* mux2_1.sv
```

```
*
```

```
* Hazirlayanlar:
```

```
* Sule Nur Demirdas
```

```
*
```

```
* Notlar:
```

```
* ELM235 2023 Bahar Lab3 - Problem 2
```

```
* 2x1 MUX Gerçelemesi
```

```
*
```

```
*/
```

```
`timescale 1ns/1ns
```

```

module mux2_1(
    input logic I1, I2, s,
    output logic y
);

logic n2,n3,n4;

nand2kapi inst0(s,s,n2);
nand2kapi inst1(s,I2,n3);
nand2kapi inst2(n2,I1,n4);
nand2kapi inst3(n4,n3,y);

endmodule

/* tb_lab3_g5_p2.sv
*
* Hazirlayanlar:
* Sule Nur Demirdas
*
* Notlar:
* ELM235 2023 Bahar Lab3 - Problem 2 Testbench
* 2x1 MUX simülasyonu
* Bütün olası giri?lere göre ç?k?? gözlemlenir.
*
*/
`timescale 1ns/1ns

```

```
module tb_lab3_g5_p2();

logic a, b, s; // Test tezgahi giri? de?i?kenleri tanimlari
logic yAnd, yOr, yNand, yNor; // Test tezgahi cikis
de?iskenleri tanimlari

lab3_g5_p2 dut0(a, b, s, yAnd, yOr, yNand, yNor);

initial begin

    // Truth table daki tum durumlar tanimlanir

    a = 0; b = 0; s = 1; #10
    a = 0; b = 1; s = 1; #10
    a = 1; b = 0; s = 1; #10
    a = 1; b = 1; s = 1; #10

    $stop;

end

endmodule
```

G. EKLER – PROBLEM 3

```
/* lab3_g5_p3.sv
*
* Hazirlayanlar:
* Sule Nur Demirdas
*
* Notlar:
* ELM235 2023 Bahar Lab2 - Problem 1
* Decoder gerceklemesi
* Amaç: 2x4 Decoder NAND kapıları kullanarak gerceklemek
*
*/

`timescale 1ns/1ns

module lab3_g5_p3(
    input logic a,b,
    output logic d0,d1,d2,d3
);

    logic n1,n2,n3,n4,n5,n6;

    nand2kapi inst0(a,a,n1);
    nand2kapi inst1(b,b,n2);
    nand2kapi inst2(n1,n2,n3);
    nand2kapi inst3(n3,n3,d3);

    nand2kapi inst4(n1,b,n4);
```

```
nand2kapi inst5(n4,n4,d2);
```

```
nand2kapi inst6(a,n2,n5);
```

```
nand2kapi inst7(n5,n5,d1);
```

```
nand2kapi inst8(a,b,n6);
```

```
nand2kapi inst9(n6,n6,d0);
```

```
endmodule
```

```
/* tb_lab3_g5_p3.sv
```

```
*
```

```
* Hazirlayanlar:
```

```
* Sule Nur Demirdas
```

```
*
```

```
* Notlar:
```

```
* ELM235 2023 Bahar Lab3 - Problem 3 Testbench
```

```
* 2x4 Decoder simulasyonu
```

```
* Bütün olası giri?lere göre ç?k?? gözlemlenir.
```

```
*
```

```
*/
```

```
`timescale 1ns/1ns
```

```
module tb_lab3_g5_p3();
```

```
logic a, b; // Test tezgahi giri? de?i?kenleri tanimlari
```

```
logic d0,d1,d2,d3; // Test tezgahi cikis de?iskenleri tanimlari
```

```
lab3_g5_p3 dut0(a,b,d0,d1,d2,d3);
```

```
initial begin
```

```
    // Truth table daki tum durumlar tanimlanir
```

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

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

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

```
        a = 1; b = 1; #10
```

```
    $stop;
```

```
end
```

```
endmodule
```