GEBZE TEKNİK ÜNİVERSİTESİ

ELEKTRONİK MÜHENDİSLİĞİ

ELM235

LOJİK DEVRE TASARIM LABORATUVARI

LAB 5 Deney Raporu

Ardışık Lojik Devre ve Sonlu Otomat Tasarımı

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

İçindekiler

1. Giriş

Bu deneyde aşağıda verilen maddeler amaçlanmaktadır:

1. Donanım tanıma dillerini (DTD) kullanarak devre tasarımı yapmak.

2. Ardışık lojik devreleri tasarlamak

3. Sentezleyici araçları kullanarak, DTD ile tanımlanan aritmetik devreleri FGPA için sentezlemek.

4. Simülasyon araçları kullanarak, otomatik simülasyon yaptırmak.

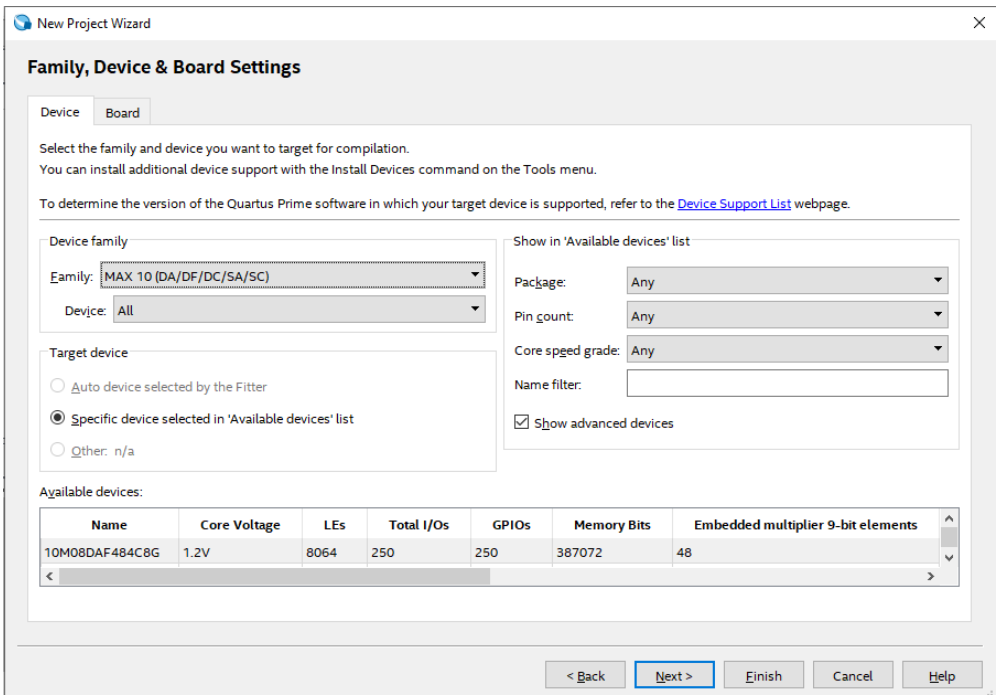
5. Devrelerin çalışma prensiplerine ve isterlere göre çalıştığını doğrulamak için senaryo oluşturup onları test edebilmek.

2. Problemler

2.1. Problem I – Yukarı Serbest Sayıcı Devresi

2.1.1. 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.



Şekil 1.Yeni Proje Oluştururken Board Seçimi

a) Yukarı Serbest Sayıcı Devresini Donanım Tanımlama Dili Kullanarak Gerçeklenmesi

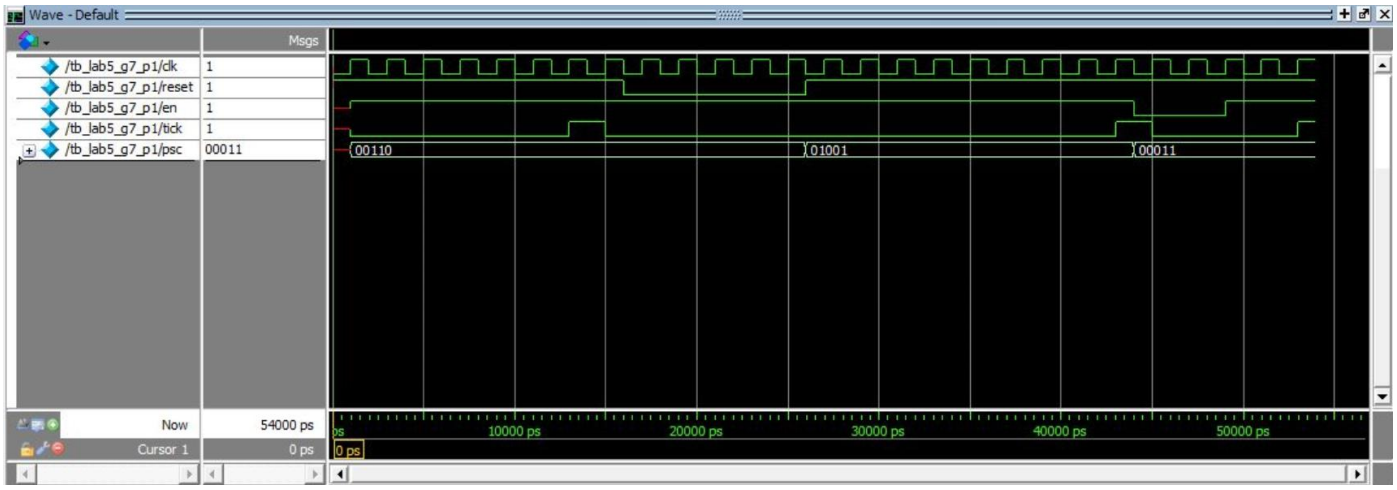
Problem 1'de istenilen devre DTD kullanarak tasarlandı. Tasarlanan DTD modelini çıkış sinyali gözlemlenecek şekilde simüle edilmiştir. Yazılan kodlar ve zamanlama diyagramı aşağıdaki gibidir:

```
/* lab5_g7_p1.sv
*  Hazırlayanlar: Umut Mehmet ERDEM – Emre TANER
*  Notlar: ELM235 2023 Bahar Lab5 – Problem 1
*  Yukarı Serbest Sayıcı Devresi
*/
`timescale 1ns/1ps
module lab5_g7_p1(
      input logic clk, reset, en,
      input logic [4:0] psc,
      output logic tick
      );
logic [4:0] psc_prv;
logic [4:0] counter = 5'b0;
always_ff @(posedge clk)
begin
      if(!reset) counter = 5'b0;
      if(psc_prv != psc && psc_prv !== 5'bXXXXX) counter = 5'b00001;
      if(en && counter!=psc) counter <= counter + 1;
      if(counter == psc && en) begin
            tick <= 1'b1;
            counter <= 5'b0;
      end
      else tick <= 1'b0;
      psc_prv = psc;
      $display("cntr:%b  psc_prv:%b   psc:%b", counter, psc_prv, psc);
end
endmodule
```

Problemde istenilen artışı sağlamak için öncelikle bir sayıcı(counter) atanmıştır. Testbench dosyasında ikinci bir psc değeri girilmesi durumunda counter değerinin 1'e düşürülmesi için her always_ff döngüsünde psc değerinin önceki değerini tutan bir psc_prv değişkeni oluşturulmuştur. enable, 1 olduğu zaman counter değeri arttırılmıştır. counter, psc' ye eşit olduğu durumda ve enable 1 olduğu durumda tick 1 olmuş ve counter 0' a düşürülmüştür. enable, 0 olduğu zaman tick 0' a düşürülmüştür.

```
/* tb_lab5_g7_p1.sv
*  Hazırlayanlar: Umut Mehmet ERDEM – Emre TANER
*  Notlar: ELM235 2023 Bahar Lab5 – Problem 1 Testbench
*  Yukarı Serbest Sayıcı Devresi Testbench
*/
// Zaman birimi ve simülasyon çozunurlugu
`timescale 1ns/1ps
module tb_lab5_g7_p1();

// Test tezgahlarinda port bulunmaz
       logic clk, reset, en, tick;
       logic [4:0] psc;

// Test edilecek modulun yaratimi ve port baglantilarinin yapilmasi
// dut = device under test
       lab5_g7_p1 dut(clk, reset, en, psc, tick);

// always blogu surekli calistirilacak
// clock oluşturmak için clk sinyalini bir lojik
// seviyeye çekip belli bir süre bekleyip, evirerek geri çekiyoruz

always begin
       clk = 0; #1; clk = 1; #1;
end

initial begin
       reset=1; #1
       en = 1; psc = 5'b00110; #15
       reset=0; #10
       reset=1; psc = 5'b01001; #18
       en = 0; psc= 5'b00011; #5
       en = 1; psc= 5'b00011; #5
       $stop; // simülasyonu durdur ve beklet. Bitirmek için
end
endmodule
```

Testbench dosyasında enable, reset, psc değerlerinin farklı kombinasyonları verilmiş ve Şekil 2'de zamanlama diyagramı elde edilmiştir. Grafik incelendiğinde doğru olduğu görülmüştür.



Şekil 2. Problem 1'in Zamanlama Diyagramı

2.1.2. Sonuçların Yorumu

Şekil 2'de clk değeri 0 olduğundan en, tick, psc sinyalleri kırmızı olarak gözükmekte ve X değerini almaktadır. Enable ve reset portunun 1 olduğu durumda counter değeri 0'dan başlayarak her yükselen kenarda artmış ve psc portuna eşit olduğunda tick 1' e yükselmiştir. Reset portu, 0 olduğu zaman counter değeri 0 olmuştur.

2.1. Problem II- Yavaşlatılabilir Aşağı Sayıcı Tasarımı

2.2.1. Deneyin Yapılışı

a) Yavaşlatılabilir Aşağı Sayıcı Tasarımını Donanım Tanımlama Dili Kullanarak Gerçeklenmesi

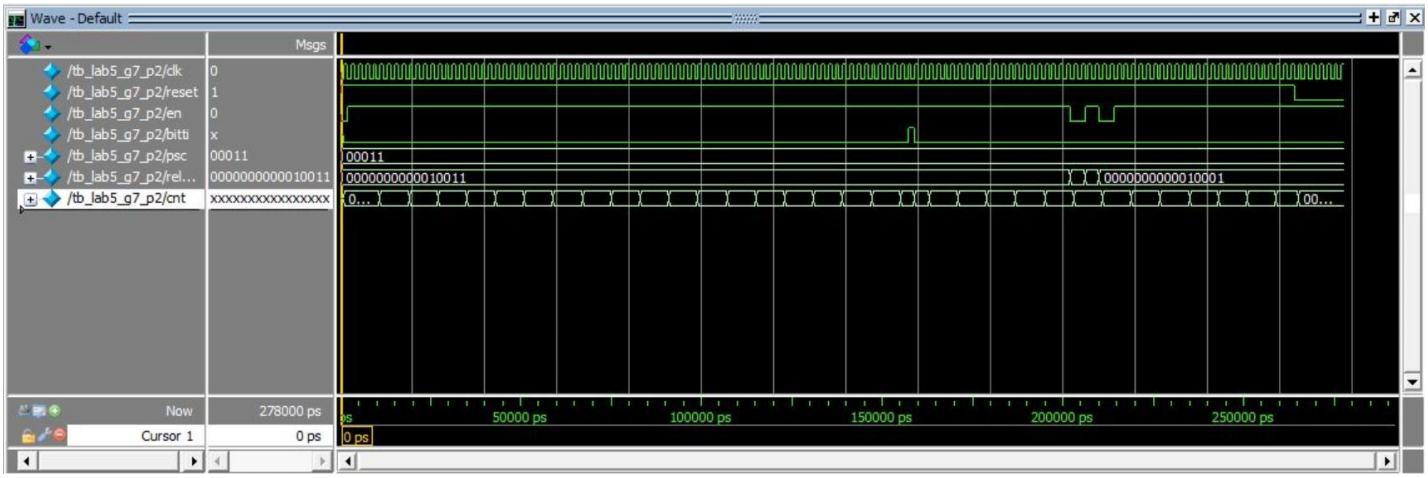
Problem 2'de istenilen tasarımı DTD kullanarak tasarlandı. Tasarlanan DTD modelini çıkış sinyali gözlemlenecek şekilde simüle edilmiştir. Yazılan kodlar ve zamanlama diyagramı aşağıdaki gibidir:

```
/* lab5_g7_p2.sv
*  Hazırlayanlar: Umut Mehmet ERDEM – Emre TANER
*  Notlar: ELM235 2023 Bahar Lab5 – Problem 2
*  Yavaşlatılabilir Aşağı Sayıcı Tasarımı
*/
`timescale 1ns/1ps
module lab5_g7_p2(
	input logic clk, reset, en,
	input logic [4:0] psc,
	input logic [15:0] reload,
	output logic [15:0] cnt,
	output logic bitti
);
logic tick;
lab5_g7_p1 p1(clk, reset, en, psc, tick);
always_ff @(posedge clk) begin
	if(!reset) cnt = 16'b0;
	if(bitti == 1'b1) begin cnt = reload; bitti = 1'b0; end
	if(!en && reload!=16'b0) begin cnt <= reload; bitti = 1'b0; end
	if(en) begin
		if(cnt == 16'b0 && reset) bitti = 1'b1;
		else begin
			if(p1.tick == 1'b1) cnt <= cnt - 1;
		end
	end
	$display("cntr:%b  tick:%b", cnt, p1.tick);
end
endmodule

module lab5_g7_p1(
	input logic clk, reset, en,
	input logic [4:0] psc,
	output logic tick
	);
logic [4:0] psc_prv;
logic [4:0] counter = 5'b0;
always_ff @(posedge clk)
begin
	if(!reset) counter = 5'b0;
	if(psc_prv != psc && psc_prv !== 5'bXXXXX) counter = 5'b00001;
	if(en && counter!=psc) counter <= counter + 1;
	if(counter == psc && en) begin
		tick <= 1'b1;
		counter <= 5'b0;
	end
	else tick <= 1'b0;
	psc_prv = psc;
end
endmodule
```

Active low olduğu durumda reset 0' da cnt 0' a atanmıştır. enable' ın 0 olduğu ve reload'ın 0 olmadığı durumda cnt reload a eşit ve bitti de 0 a eşit olacaktır. enable' ın 1 olduğu durumda cnt' un 0 olduğu ve resetin 1 olduğu noktada bitti 1 olmuştur. Diğer durumda p1 ile tanımladığımız modülde tick 1 olarak çıkarsa cnt 1'er olarak düşmüştür.

```
/* tb_lab5_g7_p2.sv
*  Hazırlayanlar: Umut Mehmet ERDEM – Emre TANER
*  Notlar: ELM235 2023 Bahar Lab5 – Problem 2 Testbench
*  Yavaşlatılabilir Aşağı Sayıcı Tasarımı Testbench
*/
// Zaman birimi ve simülasyon çozunurlugu
`timescale 1ns/1ps
module tb_lab5_g7_p2();
// Test tezgahlarinda port bulunmaz
       logic clk, reset, en, bitti;
       logic [4:0] psc;
       logic [15:0] reload, cnt;

// Test edilecek modulun yaratimi ve port baglantilarinin yapilmasi
// dut = device under test
       lab5_g7_p2 dut(clk, reset, en, psc, reload, cnt, bitti);
// Bu kisimda sinyaller test edilen devreye sirali olarak uygulanir.
// Sonuclar test edilen devre cikislarinda gozlenebilir.

always begin
       clk = 0; #1; clk = 1; #1;
end

initial begin
       reset=1; en=0; psc= 5'b00011; reload=16'h0013; #2
       en=1; #200
       en=0; reload=16'hC01F; #4
       en=1; reload=16'h0; #4
       en=0; reload=16'h0011; #4
       en=1; #50
       reset=0; #14
       $stop;
end
endmodule
```

Testbench dosyasında başta girdi olarak reset portu 1 olarak verilmiştir. Enable portu 0 olarak verilmiş aynı süre zarfında psc ve reload portları tanımlanmıştır. Enable portunun başta 0 olarak girilmediği durumda reload cnt portuna atanmayacağından psc portu çalışsa dahi cnt de düşme gözlenmeyecektir. Daha sonra, enable portu 1 yapılarak reload portunun girdisi cnt ye atanmış ve psc portunda verilen değere göre tick değeri gelerek cnt değerinin düşmesi sağlanmıştır. Enable' ın 1 olduğu durum için 200ps verildiğinde psc değeri düşük olduğundan p1 döngüsü hızlanmış ve cnt değeri hızlı düşmüştür bu nedenle bitti portu 1 olmuş devamında tekrar başa dönüp aynı şekilde 200ps dolana kadar çalışmıştır. Reset portu 0 olduğu durumda cnt değeri 0 olmuştur. enable portunun 1 olduğu durumda reload portunun değeri atanmış, atanan değer enable 1 olduğunda cnt portuna atanamamıştır.



Şekil 3. Problem 2'nin Zamanlama Diyagramı

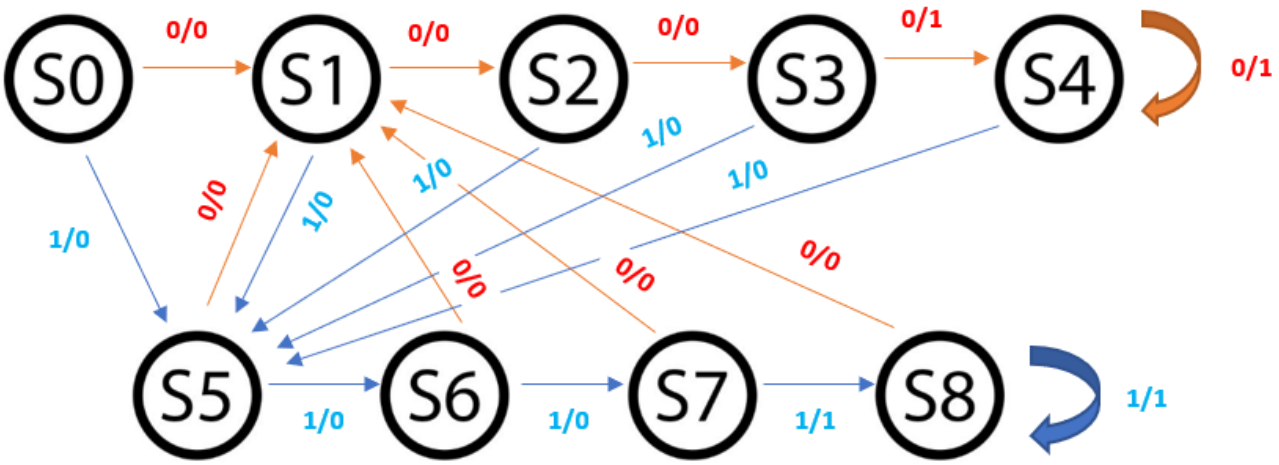
2.2.2. Sonuçların Yorumu

Şekil 3'te verilen zamanlama diyagramında psc değerine göre cnt portunun 0 olduğu durumda bitti portu 1 olacaktır reload değerinin enable portunun 0 olduğu durumda atanıp 1 olduğu durumda uzun bir süre çalıştırılmasıyla belli bir noktada bitti portu 1 olup daha sonra tekrar 0' a düşürülmüş ve cnt portu reload' a eşitlenip tekrar aynı işlem gerçekleşmiştir. Reset portu, 0 olduğu durumda cnt portunun değeri 0'a eşitlenmiştir çünkü active low reset 0 da aktif hale gelmektedir.

2.3. Problem III- Sonlu Durum Makinesi (FSM) Tasarımı: Bit Deseni Dedektörü

2.3.1. Deneyin Yapılışı

a) Sonlu Durum Makinesi (FSM) Tasarımının Durum Geçiş Diyagramının Çizilmesi



Şekil 4. Problem 3'ün Durum Geçiş Diyagramı

b) Sonlu Durum Makinesi (FSM) Tasarımının Donanım Tanımlama Dili Kullanarak Gerçeklenmesi

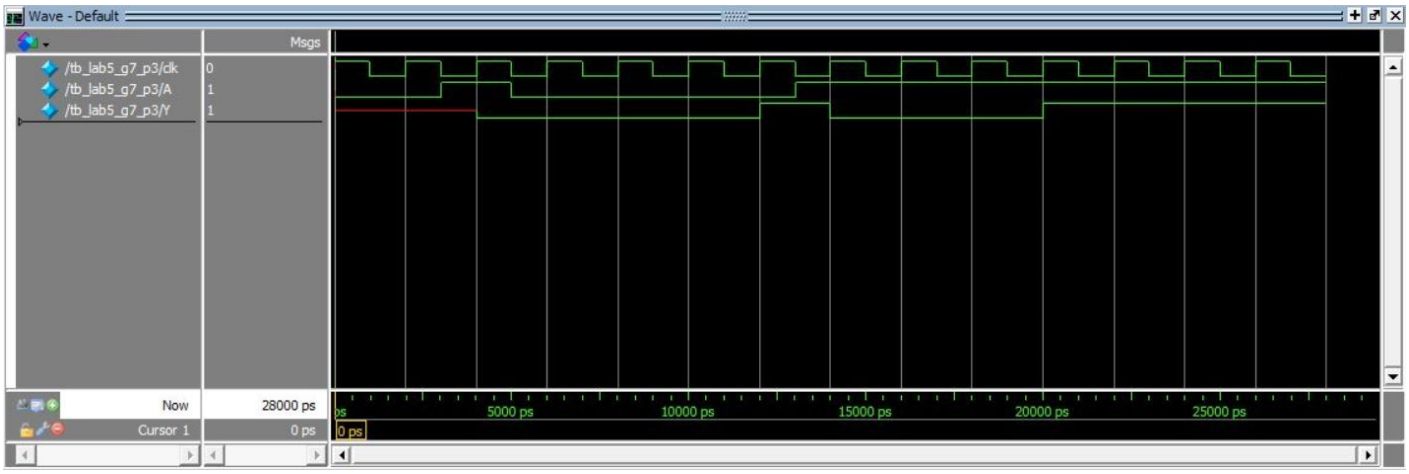
Problem 3'te istenilen tasarımı DTD kullanarak tasarlandı. Tasarlanan DTD modelini çıkış sinyali gözlemlenecek şekilde simüle edilmiştir. Yazılan kodlar ve zamanlama diyagramı aşağıdaki gibidir:

```
/* lab5_g7_p3.sv
*  Hazırlayanlar: Umut Mehmet ERDEM – Emre TANER
*  Notlar: ELM235 2023 Bahar Lab4 – Problem 3
*  Sonlu Durum Makinesi (FSM) Tasarımı: Bit Deseni Dedektörü
*/
`timescale 1ns/1ps
module lab5_g7_p3(
	input logic clk, A,
	output logic Y
);
	logic A_prv;
	logic [2:0] counter = 3'b0;
	always_ff @(posedge clk) begin
		if (A_prv == A || A_prv === 1'bX) begin
			counter = counter + 1;
			if (counter >= 4)
				Y <= 1'b1;
		end
		else begin
			counter = 3'b001;
			Y <= 1'b0;
		end
		A_prv <= A;
		$display("Aprev:%b  A:%b   cntr:%b", A_prv, A, counter);
	end
endmodule
```

3.problemde 4 tane ard arda 0 veya 1 girdileri girildiğinde Y portunun 1 olması istenildiğinden 3 bitlik counter değişkeni oluşturulmuş ve bu counter değeri her bir A port girişinin aynı olduğu durumlarda 1'er olarak arttırılmıştır. A portunun önceki değerinin tutulması ve şu anki değeri ile karşılaştırılarak counter değerinin 1'er olarak arttırılması için A_prv değişkeni oluşturulmuştur. A_prv ve A değişkeninin farklı geldiği durumda counter 1'e düşmüş ve Y portu 0 olmuştur.

```
/* tb_lab5_g7_p3.sv
*  Hazırlayanlar: Umut Mehmet ERDEM – Emre TANER
*  Notlar: ELM235 2023 Bahar Lab5 – Problem 3 Testbench
*  Sonlu Durum Makinesi (FSM) Tasarımı: Bit Deseni Dedektörü Testbench
*/
// Zaman birimi ve simülasyon çozunurlugu
`timescale 1ns/1ps
module tb_lab5_g7_p3();
// Test tezgahlarinda port bulunmaz
	logic clk, A, Y;
// Test edilecek modulun yaratimi ve port baglantilarinin yapilmasi
// dut = device under test
	lab5_g7_p3 dut(clk, A, Y);
// Sonuclar test edilen devre cikislarinda gozlenebilir.
always begin
	clk=1;#1; clk=0;#1;
end

initial begin
	A = 0;#3
	A = 1;#2
	A = 0;#8
	A = 1;#15
	$stop;
end
endmodule
```

Testbench' te A girişi belli ps sürelerinde 0 ve 1 yapılarak farklı kombinasyonlar test edilmiştir.



Şekil 5. Problem 3'ün Zamanlama Diyagramı

2.3.2. Sonuçların Yorumu

Şekil 4'te A' nın girişinin değişip değişmediği ve bu duruma göre 4 kere aynı sinyalin gelmesi ile Y portu 1 olmuştur. Buna göre zamanlama diyagramının doğru olduğu görülmüştür.

3. Sonuçlar ve Genel Yorumlar

Yapılan laboratuvar çalışmasında belli bir sayıcı portu ile verilen girdi değerlerine enable, clk ve reset portları kullanılarak arttırma ve azaltma işlemi yapılmıştır. Bunun yanında sinyallerin benzer geldiği durumlar belli koşullara göre kontrol edilmiştir. Her bir problem için oluşan koşullara göre çıktılar üretilmiş ve alınan hatalar kod üzerinde düzeltilmiştir. FSM (Finite State Machine) ne olduğu, nasıl kullanıldığı ve kod üzerinde nasıl entegre edildiği öğrenilmiştir.

4. Referanslar

[1] https://www.allaboutcircuits.com/textbook/digital/chpt-11/finite-state-machines/

[2] https://web.mit.edu/6.111/www/f2017/handouts/L06.pdf