CEB7E

Elektronik Mühendisliği

Laboratuvar Çalışması 3

Donanım Tanımlama Dilleri Uygulamaları: Birleşik Devreler

A. Amaçlar

- 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,

B. Bilgilendirme

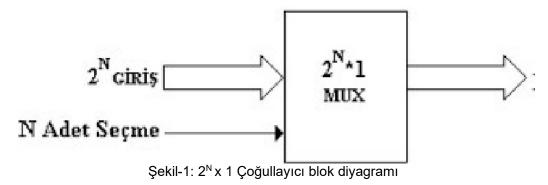
- 1. Lab'a başlamadan önce ders kadrosu tarafından verilen **ModelSim ve Quartus Prime kullanımı** adlı eğitim dökümanını takip edin, ve orada yapılan aşamaları uygulayabildiğinizden emin olun.
- 2. Dosya adlarini her zaman küçük harflerle labX_gY_pZ.uzantı olarak kaydedin. X yerine lab numarasını, Y yerine grup numaranızı, Z yerine problem numarasını ekleyin. Örneğin lab3, SystemVerilog, Problem 1 ve grup numaranız 18 ise, lab3_g18_p1.sv olarak kaydedin. Devre adını da lab3_g18_p1 olarak belirleyin. Test tezgah (Test tezgahı) isimlerinin başına tb_ önekini koyun, ve aksi belirtilmedikçe girişler arası bekleme süreleri için 10ns kullanın.
- 4. Rapor hazırlarken bu dokümanın sonundaki EK bölümünde olduğu gibi bir metin kutusu içerisine mono fontlarla (Courier New, Roboto Mono, Consolas gibi) kodlarınızı ekleyin. Kodlarınızı ayrı bir dosya olarak göndermeyin. Resim olarak eklemeyin, ve formatlamanıza özen gösterin.
- 5. Ek bölümünde verildiği gibi bir şablonu her kodunuzun başına ekleyin.

C. Teslim Edilecekler

1. Laboratuvar raporu tek bir PDF dosyası halinde olacak ve farklı bir formatta teslim ettiğiniz raporlar geçersiz sayılacaktır. İçerik olarak Lab0 daki örnek rapora bağlı kalınız. Tüm devre şemalarını ve simülasyon sonuçlarını gösteren dalga şekillerini ve zamanlama diyagramlarını eklemeyi unutmayınız.

D. Problemler

Problem 1 − 2^N x 1 Çoğullayıcı (MUX) Tasarımı



A. 2x1 MUX devresini DTD kullanarak kapı düzeyinde (structural) sadece NAND kapıları kullanarak tasarlayınız..

B. Test tezgahı oluşturarak, devrenin bütün girişlere karşı nasıl davrandığını gözlemleyin. Tasarladığınız DTD modelini ve ilgili fonksiyonel test tezgahı kodunu girişlere bütün olası kombinasyonları her 10ns lik adımlarda uygulayacak ve çıkış sinyalini gözlemleyecek şekilde DTD ile yazarak simüle ediniz ve ölçüm sonuçlarını raporlayınız. ModelSim dalga formu şemasını File→Export→Image.. ile PNG olarak çıkarabilirsiniz ya da tüm ModelSim projesinin ekran görüntüsü alabilirsiniz.

Problem $2-2^N \times 1$ Çoğullayıcı (MUX) ile Temel Lojik Kapıların ve Büyük Çoğullayıcıların Tasarımı

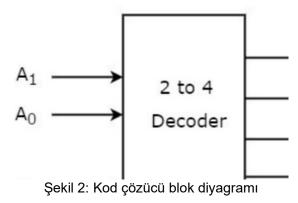
A. Problem 1 deki 2^Nx1 MUX'ı kullanarak AND, OR, NAND, NOR kapılarını tasarlayınız.

B. Tüm kapıları birlikte test edebileceğiniz bir test tezgahı oluşturarak devrenin bütün girişlere karşı nasıl davrandığını gözlemleyin. Tasarladığınız DTD modelini ve ilgili fonksiyonel test tezgahı kodunu girişlere bütün olası kombinasyonları her 10ns lik adımlarda uygulayacak ve çıkış sinyalini gözlemleyecek şekilde DTD ile yazarak simüle ediniz ve ölçüm sonuçlarını raporlayınız. ModelSim dalga formu şemasını File—Export—Image.. ile PNG olarak çıkarabilirsiniz ya da tüm ModelSim projesinin ekran görüntüsü alabilirsiniz.

C. Problem 1 deki 2^Nx1 MUX'ı kullanarak hiyerşik düzende sırasıyla 4x1, 8x1 ve 16x1 MUX devrelerini tasarlayınız. 4x1 MUX'u 8x1 MUX'da ve 8x1 MUX'u 16x1 MUX'da kullanabilirsiniz.

D. 16x1 MUX için Test tezgahı oluşturarak, devrenin rastgele seçilmiş 4 giriş için farklı seçim girişine karşı nasıl davrandığını gözlemleyin. Her 10ns lik adımlarda seçim girişine veri uygulayacak ve çıkış sinyalini gözlemleyecek şekilde devreyi DTD ile yazarak simüle ediniz ve ölçüm sonuçlarını raporlayınız. ModelSim dalga formu şemasını File—Export—Image.. ile PNG olarak çıkarabilirsiniz ya da tüm ModelSim projesinin ekran görüntüsü alabilirsiniz.

Problem 3 – 2 x 4 Kod Çözücü (Decoder) Tasarımı



A. 2x4 DECODER devresini DTD kullanarak kapı düzeyinde (structural) sadece NAND kapıları kullanarak tasarlayınız..

B. Test tezgahı oluşturarak, devrenin bütün girişlere karşı nasıl davrandığını gözlemleyin. Tasarladığınız DTD modelini ve ilgili fonksiyonel test tezgahı kodunu girişlere bütün olası kombinasyonları her 10ns lik adımlarda uygulayacak ve çıkış sinyalini gözlemleyecek şekilde DTD ile yazarak simüle ediniz ve ölçüm sonuçlarını raporlayınız. ModelSim dalga formu şemasını File—Export—Image.. ile PNG olarak çıkarabilirsiniz ya da tüm ModelSim projesinin ekran görüntüsü alabilirsiniz.

Problem 4 – 4 x 16 Kod Çözücü (Decoder) Hiyerarşik Tasarımı

C. Problem 3'teki 2x4 DECODER ve NAND kapılarını kullanarak 3x8 ve 4x16 DECODER devrelerini DTD ile yapısal (structural) tarzda tasarlayınız.

D. 4x16 DECODERiçin Test tezgahı oluşturarak, devrenin rastgele seçilmiş 4 giriş için farklı seçim girişine karşı nasıl davrandığını gözlemleyin. Her 10ns lik adımlarda seçim girişine veri uygulayacak ve çıkış sinyalini gözlemleyecek şekilde devreyi DTD ile yazarak simüle ediniz ve ölçüm sonuçlarını raporlayınız. ModelSim dalga formu şemasını File—Export—Image.. ile PNG olarak çıkarabilirsiniz ya da tüm ModelSim projesinin ekran görüntüsü alabilirsiniz.

Problem 5 – Çoğullayıcı (MUX) ve Çözücüler (DECODER) ile Birleşik Devre Tasarımı

A. $F_1(A,B,C,D) = \sum \{1,3,7,11,12,13,15\}$ ve $F_2(A,B,C,D) = \prod \{2,3,4,5,7\}$ olarak verilen fonksiyonları daha önce tasarladığınız MUX ve Decoder devrelerini kullanarak DTD ile yapısal şekilde tasarlayınız.

- B. Test tezgahı oluşturarak, devrenin bütün girişlere karşı nasıl davrandığını gözlemleyin. Tasarladığınız DTD modelini ve ilgili fonksiyonel test tezgahı kodunu girişlere bütün olası kombinasyonları her 10ns lik adımlarda uygulayacak ve çıkış sinyalini gözlemleyecek şekilde DTD ile yazarak simüle ediniz ve ölçüm sonuçlarını raporlayınız. ModelSim dalga formu şemasını File→Export→Image.. ile PNG olarak çıkarabilirsiniz ya da tüm ModelSim projesinin ekran görüntüsü alabilirsiniz.
- C. Devrenin ne kadar yer kapladığını (resource utilization report) sentezlenen RTL ve eşleştirme ardı devre şemalarını da ekleyerek karşılaştırmalı şekilde yorumlayınız.
- D. Her bir fonksiyon için K-Map kullanarak fonksiyonların en sade Boole cebri ifadesini bulunuz. Toplamda kaç devre elemanı kullandığınızı (NOT kapıları dahil) belirtiniz.
- E. Elde ettiğiniz sonuçları C. deki sentez sonuçlarıyla karşılaşırın ve hangi tasarımın daha ekonomik olacağını yorumlayınız.

Referanslar

- [1] https://www.electronics-tutorials.ws/combination/comb 5.html
- [2] https://www.sathyabama.ac.in/sites/default/files/course-material/2020-10/Unit2 0.pdf

E. Ekler

1. Örnek bir SystemVerilog DTD tasarımı ve test tezgahı

```
/* lab1 g0 p10.sv
* Hazırlayanlar:
* Ihsan Cicek
* Notlar:
* ELM235 2023 Bahar Lab1 - Problem 10
* Y = NOT A and B denkleminin gerçeklemesi
*/
module lab1_g0_p10 (
input logic a, b,
output logic y
      assign y = -a \& b;
endmodule
/* tb_lab1_g0_p10.sv
* Hazırlayanlar:
* Ihsan Cicek
* Notlar:
* ELM235 2023 Bahar Lab1 - Problem 10 Testbench
* Y = NOT A and B denkleminin simulasyonu
* Bütün olası girişlere göre çıkış gözlemlenir.
*/
// Zaman birimi ve simulasyon cozunurlugu
`timescale 1ns/1ps
module tb lab1 g0 p10 ();
// Test tezgahlarinda port bulunmaz
logic a tb, b tb; // test tezgahi giris sinyal tanimlari
logic y tb; // test tezgahi cikis sinyal tanimlari
// Test edilecek modulun yaratimi ve port baqlantilarinin yapilmasi
// mut = module under test,
lab1_g0_p10 mut0(.a(a_tb), .b(b_tb), .y(y_tb));
// Bu kisimda sinyaller test edilen devreye sirali olarak uygulanir.
// Sonuclar test edilen devre cikislarinda gozlenebilir.
initial begin
      a tb = 0; b tb = 0;
      \#10; // a = 0, b = 0 yap, 10 ns bekle
      b tb = 1;
      #10; // a = 0, b = 1 yap, 10 ns bekle
      a tb = 1;
      \#10; // a = 1 yap, b = 1, 10 ns bekle
      b tb = 0;
      #10; // a = 1, b = 0 yap, 10 ns bekle
      #20; // simulasyondan sonra 20 ns bekle
      $finish; // sımulasyonu sonlandır
end
endmodule
```