

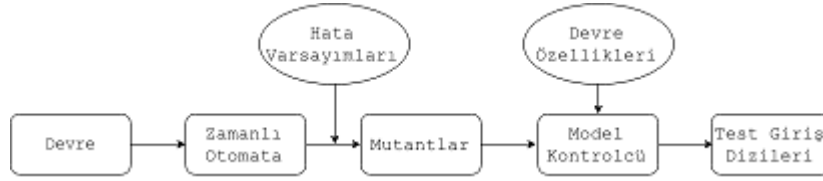
Sayısal Devrelerin Model Kontrol Tabanlı Testi

Savaş Takan

savastakan@iyte.edu.tr

İzmir Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği Bölümü 35430 Urla/İzmir

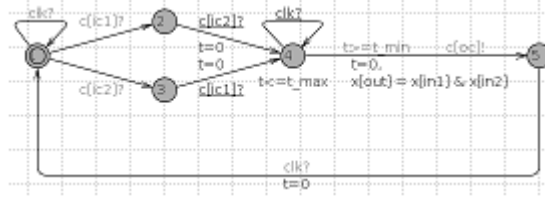
Özet Bu tezin amacı yazılım mühendisliği alanında yaygınca kullanılan bir test yöntemi olan model doğrulama tabanlı test yönteminin, devrelerin gecikme hata testinin denetlenmesi işlemine uygulamaktır. Devre gecikme hatası devrelerin istenilen zamanda istenilen işlevin yerine getirilmemesinden kaynaklanmaktadır. Bu metodolojide, öncelikle devre bir zamanlı otomata olarak modellenir ve böylece devrenin zamansal karakteristikleri ortaya çıkarılır. Ardından model sürekli olarak belirli varsayımlar çerçevesinde gecikme hataları oluşması için mutasyona uğratılır ve tüm mutant modeller verilen niteliklere göre model doğrulayıcı tarafından kontrol edilir. Test giriş dizisi, model doğrulayıcı tarafından geri döndürülen karşıt örneklerden oluşur. Aşağıda metodolojinin kısa özeti verilmiştir.



Şekil 1. Proje Metodolojisi

Günümüzde, VLSI devrelerdeki gecikme kusurlarının ifade edilebilmesi amacıyla kullanılan çeşitli modeller vardır. Örneğin, geçiş gecikme hatası, kapı gecikme hatası, yol gecikme hatası, segment gecikme hatası, çizgi gecikme hatası vb.[1]. Kapıların gecikmeleri zaman aralıkları ile ifade edilir ve her kapı üretim çeşitliliği sebebiyle farklı gecikme sürelerine sahip olabilir. Örneğin, 2-girişli bir AND kapısı şu formül ile ifade edilebilir $o(t) := i_1(t)_{\text{AND}} i_2(t)$. Giriş sinyalleri olan $i_1(k)$ ve $i_2(k)$ 'nin t_1 ve t_2 zamanlarında yenilediğini ve sırasıyla $i_1(k+1)$ ve $i_2(k+1)$ olduğunu varsayalım. Kapının yayılım gecikmesi $\delta = [t_{\min}, t_{\max}]$ aralığı ile verilmiştir. Bu yüzden, yeni çıkışın $t = \max\{t_1, t_2\} + \delta$ zamanında stabil bir duruma geçtiği varsayılır. Sıralı devreler için kural, sonraki-durum mantıksal çıkışlarının bir sonraki clock geçişinden önce stabil olması gerekliliğidir. Eğer birincil giriş ve sonraki-durum çıkışını oluşturan durum girişleri boyunca uzanan bir yolda art arda sıralı olan kapıların gecikmelerinin toplamı bir clock devrini aşmaktaysa, bu hatalı durum geçişine sebep olabilir ve sonuç olarak devrenin birincil çıkışlarının hatalı olmasına neden olabilir.

Zamanlı otomata, sistemlerin zamansal özelliklerini tanımlamak için çok değerli bir araçtır. Her kapının zamana bağlı davranışını ifade edebilmek için onu bir zamanlı otomata olarak temsil edebiliriz. Bellek elemanları, flip-floplar, clocklar ve devrenin diğer elemanları da aynı şekilde birer zamanlı otomata olarak ifade edilebilirler. Böylelikle, tüm devre zamanlı otomatalardan oluşan bir ağ yapısına çevrilmiş olur. Aşağıda 2-girişli bir AND kapısının zamanlı otomata olarak ifade edilmiş hali görülmektedir.



Şekil 2. AND Kapısı için bir Zamanlı Otomata

UPPAAL, zamanlı otomata teorisine dayanan ve iyi bilinen bir model kontrolcüsüdür. Ayrıca klasik zamanlı otomataları, birden fazla otomatanın birbiri ile iletişim içinde olmasını sağlayan haberleşme kanalları ile geliştirmektedir.[3][4] Biz devredeki her bir kablo için bir haberleşme kanalı kullanmaktayız. 2-girişli bir kapı üç kanala sahiptir, bir adet çıkış ve iki adet giriş için. Kapının çıkışı stabil hale geldiğinde kapının haberleşme kanalı işaretlenmektedir. Otomata 1-2-4 ya da 1-3-4 durumlarını hangi giriş sinyalinin önce geldiğine bakarak izlemektedir. 4. duruma ulaştığında, en az t_{min} kadar bir süre beklemektedir fakat t_{max} olmadan da o durumu terk etmektedir. Bunu yaparken mantıksal işlemi olan $x[out] = x[in1] \& x[in2]$ işlemini yapar ve $c[oc]$ haberleşme kanalını işaretleyerek çıkışı stabil olduğunu haber verir. Model kontrolcüsü sistemi bu zamanlama aralığıyla genişletir ve verilen özellikleri kontrol eder. Metodoloji model doğrulama tabanlı test yöntemini, devrelerin gecikme hata testinin denetlenmesi işlemine uygular. Bu metodolojinin avantajı model doğrulama tabanlı testler optimum test takımı bulmak için tam kapsamlı arama yapabilmesidir. Öte yandan bu metodolojin dezavantajı ise bu işlemi yaparken yüksek miktarda işlem gücü ve hafıza harcamasıdır. Bu dezavantaj bazı sadeleştirme yöntemleri ile aşılabacaktır.

Anahtar Kelimeler: Devre Tasarımı, Devre Testi, Model Tabanlı Test, Zamanlı Otomata

Kaynaklar

1. Jha, Niraj K. and Gupta, Sandeep, "Testing of digital systems,", 2003.
2. Majhi, Ananta K. and Agrawal, Vishwani D., "Tutorial: Delay Fault Models and Coverage,", 1997.

3. Gerd Behrmann and Re David and Kim G. Larsen, "A tutorial on uppaal," Entertainment Computing, 2004.
4. S. C. J. Bakkes, P. H. M. Spronck, and H. Jaap van den Herik, "Uppaal 4.0," 2006.