

Bilgisayar Mimarisi ve Sayısal Sistemler Eğitiminde İnternet ve e-Öğrenme

Elektronik Y. Müh. M. Niyazi Saral
Tübider Okul Bilişim - <http://www.okulbilisim.com>
nsaral@cizgi.com.tr

Dr. Yıldırım Topcu
Çizgi Söğüt Gölgesi - <http://csg.cizgi.com.tr/>
ytopcu@gmail.com

Özet: *Teknolojik donanımların giderek karmaşık hale geldiği ve soyutluk derecesinin sürekli yükseldiği bilgisayar mimarisi ve sayısal sistemler gibi mühendislik alanlarında eğitim amaçlı kullanılacak simülasyonlar ve benzeri eğitim araçları hakkında bilgiler verilmiş, bu alanda tasarlanabilecek eğitim modelleri ve platformlar için önerilerde bulunulmuştur. Çalışma kapsamında incelenen simülasyonlar teknik özellikleri, yazılım tasarımı kriterleri ve pedagojik ihtiyaçlar açısından değerlendirilmiştir. Dünyada ve Türkiye’de takip edilen eğitim stratejileri konusunda değerlendirmeler yapılarak öneriler sunulmuştur.*

Anahtar Kelimeler: Sayısal devreler, bilgisayar mimarisi, simülasyon, HDL (Hardware Description Language), PBL (Project Based Learning)

Giriş

Bilgi çağı olarak adlandırılan son 30 yılda, geçmişteki 5000 yıldan daha fazla bilgi üretildiği ifade edilmektedir. Bilgi toplumunu oluşturan bireylerin en temel özellikleri bilişim yeteneklerinin ön planda olmasıdır. Bilgi toplumunun eğitiminde en son teknolojileri kullanabilen, zaman ve mekân sınırı olmayan, düşük maliyetli ve öğrenci odaklı yaklaşımlar giderek önem kazanmakta ve daha çok rağbet görmektedir [1]. Bilgi toplumunda bilgisayar ve bağlı bilim alanlarındaki temel eğitiminin de özellikle kendi süreçleri içinde etkin öğrenme teknikleri çerçevesinde e-öğrenme ve diğer internet araçları ile ele alınması en doğru yoldur.

Bu çalışmanın amacı Türkiye’de ve dünyada bilgisayar bilimi temel konularından sayısal devreler, bilgisayar organizasyonu ve mimarisi, mikro denetleyiciler ve mikro işlemciler eğitiminde izlenen stratejiler konusunda değerlendirmeler yapmak, bu alanda kullanılacak animasyon, simülasyonlar, ders videoları ve benzeri ücretsiz erişim sağlanabilen kaynaklar hakkında bilgiler vermek, kısıtlı imkânlar ile verilen eğitim için kaynak ve bilgi oluşturmaktır. Ayrıca önerilen uzaktan eğitim stratejileri sosyal oluşumculuk (Social Constructivism), karma öğrenme (Blended Learning) gibi eğitim metodolojileri açısından değerlendirilmiştir.

Bilgisayar mimarisi eğitimi ve internet tabanlı deneysel araçlar

Özellikle sayısal devreler, bilgisayar organizasyonu ve mimarisi, mikro denetleyiciler ve mikro işlemciler eğitiminde çok kısa süreler içinde sürekli yenilenen, giderek büyüyen ve daha bütünlük, daha karmaşık hale gelen bilgi işlem donanımları nedeniyle mevcut eğitsel araçların geçerliliği çabucak ortadan kalkmakta ve deneysel araç olarak animasyonların ötesinde simülasyonlara duyulan ihtiyaç giderek artmaktadır [2].

Simülasyonlar sayesinde gerçek hayatta ve fiziki eğitim ortamında oluşturulması çok zor, zaman alıcı veya pahalı deneysel ortamlar oldukça kolay, hızlı ve pratik biçimde oluşturulabilmektedir. Öğrencilere sağlanacak bu laboratuvar çalışmalarında, ev ödevlerinde ve projelerde, simülasyonlar sayesinde öğrencilerin gerçek hayatta gözlemleri oldukça zor olan çalışma ilkeleri ve tekniklerini adım adım ve tekrar tekrar izleyebilmelerine imkân tanınır ve dolayısıyla eğitim kalitesi yükselir. Simülasyonlar ile çalışma internet ortamlarında ve/veya e-öğrenim sitelerinde yapılabilir. İnceleyeceğimiz simülasyonlardan çoğu İnternet’ten ücretsiz olarak temin edilebilmektedir, bu şekilde diğer eğitimler için yararlı ve zaman tasarrufu sağlayan bir kaynak oluşturmaktadırlar. Bununla beraber, belirli bir ders veya konu için doğru simülasyonu aramanın kendisi zaman tüketici bir süreç olabilir. Bu sebeple, çalışmamızda müfredatın değişik konularına uygun farklı simülasyonlar önerilmiş ve bunlar üzerinde yoğunlaşmıştır.

Simülasyonların animasyonlu eğitim araçlarından temel farkı, özellikle e-öğrenme sürecinde asenkron eğitim alan bir öğrencinin bir anlamda interaktif kılınabilmesi, yapma/pratik etme ve kullanma yeteneklerinin daha etkin olabilmesidir. Simülasyonlar yaygın öğrenimde e-öğrenme süreçlerinde; örgün öğrenimde ise laboratuvar, ev ödevleri ve proje çalışmalarında

idealdir. Simulatörler çoğu zaman diğer seçenek olan gerçek elemanlar ile deneyerek yapmaktan daha da etkin süreçlerdir. Öğrencilerin zaten bildikleri ve çok basit kurulum (örneğin teller ile bağlamak, deneysel kitleri hazırlamak ve diğer çalışma şartlarını oluşturmak, vs.) ön hazırlıkları için vakit kaybedilmesinin önlenmesi gibi gerçek hayattan farklı olarak öğrencinin hata yapması da özellikle istenir. Örneğin bir uçak simulatöründe uçağın çok dik hale getirilmesi gibi gerçek hayatta telafisi pek mümkün olmayan pilot hataları ancak bir simulatörde tecrübe edilebilir. Aynı şekilde bir bilgisayar tasarımında ortak veri yoluna çıkış yapan birden fazla bileşenin zamanlama sorunu ancak simulatörde denenebilir.

Simulatörler konusunda yapılan çalışmalar, eğitimcilerin mevcut simulatörlerin kapasite ve yetenekleri konusunda yeterince bilgi sahibi olmadıklarını veya amaca yönelik özel simulatörlerin varlığı konusunda yeterince fikir sahibi olmadıklarını göstermektedir [2]. Bu makalede dünyadaki üniversitelerde, e-öğrenim sitelerinde ve diğer internet ortamlarında bilgisayar mimarisi ve sayısal sistemler eğitiminde kullanılan simulatörlerin hemen hemen tümü incelenmiş, avantaj ve dezavantajları değerlendirilerek her konu için ücretsiz ve en uygun olanları belirlenmiştir.

Bilgisayar bilimi temel konularının (sayısal devreler, bilgisayar organizasyonu ve mimarisi, mikro denetleyiciler ve mikroişlemciler) tümü için ayrı ayrı simulatörler seçilebileceği gibi lise düzeyi, üniversite düzeyi ve/veya daha yüksek seviyeler için farklı ve ileriye yönelik simulatörler de seçilebilir. Örneğin, üniversite düzeyi için seçilmesi gereken simulatör basit mantıksal geçitler için de kullanılabilir gibi son aşamada mutlaka HDL (Hardware Description Language) gibi bir yüksek seviye tasarım diline de uyumlu olmalıdır [3-4]. Dolayısıyla, simulatörlerin eğitim süreçlerinde nasıl ve nerelerde kullanılacağı, eğitimin hangi hedefleri için kullanılacağı gibi yüksek öneme sahip soruların öncelikli olarak cevaplandırılması zorunludur. Çalışmamızda pedagojik yaklaşımlardan simulatör ve benzeri yardımcı eğitim araçlarının eğitsel faydalarına, teknik ve tasarım özelliklerine, müfredat ile ilişkisine kadar eğitim süreçlerini ilgilendiren tüm detaylara değinilmeye çalışılmıştır.

Eğitim Yaklaşımları ve Sayısal Sistemler Eğitimi

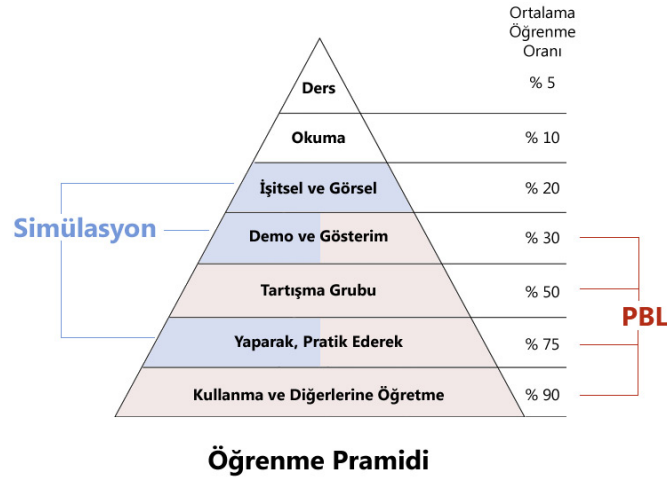
Sosyal Oluşumculara (*Social Constructivism*) göre öğrenme sosyal bir süreçtir. Davranış değişimi (yani öğrenme) pasif bir süreçle dış güçler tarafından gerçekleştirilemez. Anlamli öğrenmeler, bireyler sosyal etkinliklere katıldığında gerçekleşir [5].

NTL (National Training Laboratories) tarafından yapılan çalışmalar sonucu geliştirilen “öğrenme piramidi” (Şekil 1) incelendiğinde öğrencilerin pasif bir süreçte bilgi aldıkları ders, okuma, işitsel-görsel ve demo-gösterim tekniklerinin öğrenmeye katkısı %5 ile %30 arasında değişirken aktif bir süreçte yer aldıkları tartışma grupları, yapma/pratik etme, kullanma ve diğerlerine öğretme tekniklerinin öğrenmeye katkısı sırasıyla %50, %75 ve %90 değerlerine ulaşmaktadır [6-7].

Uzaktan öğrenme yöntem ve uygulamaları söz konusu olduğunda bunlardan ayrı düşünülmemeyecek bir kavram olan karma öğrenme (*Blended Learning*) konusuna da atıfta bulunmak gereklidir. Karma öğrenme, uzaktan eğitim yöntemlerinin bireylerde iç motivasyon, zamanı etkin kullanma ve sorumluluk sahibi olma gibi gereksinimleri yanında asosyalleştirme (sürekli bilgisayar karşısında ve tek başına kalma durumu) tehlikesine karşın modern yöntemlerin geleneksel yöntemlerle birleştirildiği bir öğrenme biçimidir [8]. Bu nedenle işbirliğine dayalı (*Cooperative/Collaborative Learning*) çalışmaların yapılabildiği yazılımlar, zaman zaman kişileri belirli projeler için bir araya getiren proje tabanlı etkinlikler (*Project Based Learning, PBL*) öğrenmenin verdiği hazzı ve dolayısıyla kalitesini arttıracaktır.

Özellikle sayısal devreler, bilgisayar organizasyonu ve mimarisi, mikro denetleyiciler ve mikroişlemciler eğitimi gibi soyutluk derecesinin giderek arttığı ve dolayısıyla anlaşılabilirliğinin giderek güçleştiği bir alanda tasarlanacak eğitim modelinde NTL tarafından ortaya konan bu öğrenme piramidinin ve sayılan öğrenim metodolojilerinin dikkate alınması gerektiği açıktır.

Öğretmen-öğrenci etkileşimine ve ders vermeye dayalı klasik eğitim yöntemleri incelendiğinde, müfredatın birbirinden bağımsız küçük bilgi ve beceri parçacıkları şeklinde verilmesi, teknik yetersizlik veya yürütme güçlüğü nedeniyle beceri eğitiminin arzu edilen seviyede olamaması, iletişim becerilerine gereken önemin verilememesi ve disiplinler arası ilişkilerin sağlanamaması gibi önemli yetersizlikler görülmektedir [9]. Bu yetersizlikler tek bir kişinin çözüme ulaşmasının çok güç olduğu, sürekli olarak daha kapsamlı ve karmaşık problemlerin ortaya çıktığı mühendislik dünyasında öğrencilerin gerçek hayata yeterince hazır olamaması ile sonuçlanmaktadır.



Şekil 1. Bilgisayar mimarisi ve sayısal sistemler eğitiminde Öğrenme Piramidi

Klasik eğitim yöntemlerinin bu olumsuzluklarına karşı geliştirilen proje tabanlı eğitim (PBL), mühendislik eğitiminin vazgeçilmez bir unsuru haline gelmiştir. Teknolojik yenilikler ve internet kullanımının yaygınlaşmasına paralel olarak gelişen uzaktan öğrenim araçları, yeni kaynaklar ve yeni servisler sunarak proje tabanlı eğitimin de gelişimine katkıda bulunmuştur. Çalışmamız içinde incelenen simülasyonlar ve diğer internet tabanlı araçlar bu yeniliklere birer örnek teşkil etmektedir.

Aktif öğrenmenin sağlanması amacıyla geliştirilen bu yeni yaklaşımda öğretmen ve öğrencilerin aktiviteleri klasik pedagojilerde olduğundan daha farklıdır. Öğretmenler daha çok yönlendirici rol üstlenerek proje olarak tasarlanan çalışma için amaç, içerik, aktiviteler, çalışma metotları, grup üyelerinin görevleri, sunumlar ve projenin değerlendirme stratejilerini belirler. Bir proje için zaman planlaması, kullanılacak teknoloji, maliyet, kontrol ve değerlendirme sistemi gibi tüm detaylar baştan belirlenir. Proje görevlere bölünür, her bir görev belirli aktiviteler içerir ve ürün sunumuyla sonuçlanır. Öğrenciler projenin başlangıç aşamasından sunum aşamasına kadar etkin bir şekilde rol alır. Akran denetimi ve liderliği, zamanlanmış görevler, tartışma grupları ve değerlendirme süreçlerinde yer alma gibi aktif roller üstlenerek bir anlamda öğretmenin yönetim ve denetim görevlerini paylaşır. Farklı disiplinlerin uygulamalarına yer verilen, grup içinde çalışma, iletişim, karar alma ve teorik bilgilerden yararlanarak çözümler üretme gibi becerileri geliştiren bu kapsamlı aktivite sonunda gerçek hayata yakın şekilde ve arzu edilen en üst seviyede bilgi ve beceriler edinilebilmektedir [10].

Gerekli altyapı, araç ve servisleri sağlayarak böylesi kompleks aktiviteleri destekleyen araçlar en etkin öğrenme sistemleridir. Özellikle sayısal devreler, bilgisayar organizasyonu ve mimarisi gibi mühendislik eğitimleri söz konusu olduğunda bu destekleyici sistemlere en güzel örnekler çalışmamızda her yönüyle incelenen DigSim, Logisim, Deeds ve Hades gibi simülasyon araçlarıdır. Bu araçların ne kadar etkili olduğu hiç şüphesiz tartıştığımız pedagojik kriterler doğrultusunda ne kadar etkili kullanılabildiğine bağlıdır.

Simülasyonun Eğitimsel Yararları

Wolffe ve arkadaşları bilgisayar mimarisi ve sayısal sistemler eğitiminde yardımcı araç olarak kullanılan simülasyonların başlıca yararlarını şöyle sıralamıştır: (1) öğrenciler bilgisayar mimarisi ve organizasyonunun altında yatan detayları çoklu soyutlama seviyelerinde öğrenebilirler; (2) öğrenciler istedikleri zaman ve yerde içeriğe kolay bir şekilde erişebilirler böylelikle e-öğrenme hedefleri içinde asenkron eğitim için kullanılabilirliğine imkan sağlar; (3) hepsi için geçerli olmasa dahi çoğu konu başlığı için en son içerik kullanıma sunulur; (4) birçok simülasyon ders kitapları ile uyumludur; ve (5) düşük veya sıfır altyapı maliyeti mevcuttur [2].

İnternet üzerinden erişilen simülasyonlar öğrencilerin sayısal devreleri programlamasından kendi yeni mimarilerini oluşturmalarına kadar geniş bir alanda uygulama yapmalarına imkân sağlamaktadır. Yazılımda bir bilgisayar sisteminin

simülasyonu yapmak donanımda gerçek bir bilgisayar yapmaya benzer. Öğrencilerin hata yapıp düzeltmelerine olanak sağlamak konusunda daha esnek ve üstelik daha ucuzdur.

Simülatörlerde, öğrenciler oldukça basit bir şekilde çalışma ekranlarında tasarladıkları devrelere yeni bileşenleri dâhil ettiğinde ve fare kullanarak bu bileşenler arasında bağlantılar oluşturduğunda sayısal devreler kolayca yapılandırılmış olur. Öğrencilerin bu şekilde tasarladığı ve çalışma prensiplerini irdelediği devreler daha sonra yeniden yüklenmek ve simüle edilmek üzere dosyalar şeklinde kaydedilebilir.

Simülatör kullanan öğrencinin ilk ihtiyacı bir grafik ekranında giriş değerlerini kontrol edebildiği ve bunların devre üzerinde etkilerini görebildiği canlandırma yapabilmesidir. Bu, sayısal bir devrenin incelenebileceği en basit yöntemdir ve bu öğrenim şekli yeni başlayanlar için oldukça yararlı ve kullanışlıdır. Öğrenci grafik ekranında seçtiği ve/veya oluşturduğu sayısal devrelerin giriş uçlarına deneysel sayısal değerler ("0" veya "1") vererek çıkış uçlarındaki sonuçlarını irdeler. Bir sonraki adım, devrenin zamana karşı davranışlarını incelemektir.

Bir öğrenim aracı olarak günümüz simülatörleri, basit bir kullanıcı arabirimi ve herhangi bir zamanda erişilebilen kapsamlı bir çevrimiçi yardım sistemiyle birlikte kullanım kolaylığı sağlamaktadır. Kurulumları çok kısa bir süre içerisinde yapılır ve yardım sisteminde yer alan kısa yönlendirme talimatlarıyla sayısal devreler birkaç dakika içerisinde oluşturulabilir.

Geçmiş dönemlerde laboratuvar uygulamalarında devre prototiplerinin yapımı ve test edilmesi ön planda tutulmaktaydı. Bugün başlı başına bir laboratuvar gibi kullanılan simülatörler, sayısal devrelerin işletim prensibi gösteriminden ölçümleme uygulamalarına, tasarım ve prototip oluşturmada test etme ve arıza bulmaya kadar birçok amaca hizmet etmektedir [2].

Simülasyon Araçlarının Sınıflandırılması

Bilgisayar mimarisi ve sayısal devreler eğitiminde kullanılacak düzeyde çok sayıda simülatör mevcuttur. Bu simülatörlerin kategorizasyonu konusunda yapılan çalışmalar simülatörlerin genel olarak aşağıdaki şekilde sınıflandırılabilirliği üzerinde görüş birliğine varmıştır [2-4];

- Sayısal Mantık Devresi Simülatörleri (Digital Logic Simulators)
- Basit Kuramsal Makine Simülatörleri (Simple Hypothetical Machine Simulators)
- Orta Seviye Komut Kümesi Simülatörleri (Intermediate Instruction Set Simulators)
- Gelişmiş Mikro mimari Simülatörleri (Advanced Microarchitecture Simulators)
- Çoklu İşlemci Simülatörleri (Multi-Processor Simulators)
- Öğretici Sistem Simülatörleri (Educational System Simulators)

Sırasıyla bu simülatörlerin yapı ve işlevleri hakkında önemli hususlara değineceğiz:

Modern ve gelişmiş bilgisayarlar esasında çok sayıda ve oldukça basit yapıların bir araya gelmesiyle oluşmaktadır. Bir eğitimci ister MOS transistörleri ve Boole mantık geçitlerinden, isterse bir bütün olarak mantık devrelerinden başlamayı seçsin, amaç daima aynıdır; basitten karmaşığa doğru hiyerarşik yapıya uygun bir eğitim modeli... Sayısal sistemlerin bu en temel seviyesine hitap eden **sayısal mantık devresi simülatörleri** şu donanım özelliklerinin işleyişini tanımlayabilmektedir; anahtarlama teorisi ile temel sayısal elemanlar, devre analizi (uygulama, minimum hale getirme), zamanlama sistemleri (yayıma gecikmeleri, riskler), flip-floplar/kayıt devreleri/yazıcılar, mantık yapıları (çoklayıcılar, bilgi dağıtıcılar, karşılaştırıcılar, toplama devreleri) ve depolama elemanları (ROM, PROM, RAM).

Sayısal Mantık Devresi Simülatörleri (Digital Logic Simulators)	<ul style="list-style-type: none">• Digital Simulator (http://www.mit.edu/people/eichin/thesis/usrdoc.html)• MIT Digital Logic Simulator (http://web.mit.edu/ara/www/ds.html)• Iowa Logic Simulator (http://www.cs.uiowa.edu/~jones/logicsim/)• Simcir – The circuit Simulator (http://www.d-project.com/simcir/)• EasySim - Digital Logic Simulator (http://www.research-systems.com/easysim/easysim.htm)• DigSim – Digital Workshop (http://akademi.okulbilisim.com/elearning/simulator.aspx)• Java Digital Breadboard Simulator (http://www.cs.york.ac.uk/netpro/bboard/)
---	---

Simulatörler kullanılarak öğrenmenin gerçek bir makinede elde edilen deneyimle aynı olmadığı şeklindeki iddialara hak veriyoruz; evet, simulatörler kesinlikle daha iyi olabilir! Gerçek makineler giderek daha karmaşık hale geldikçe, bilgisayar mimarisi ve organizasyonuna ilişkin başlangıç derslerinde zorunlu olarak bulunması gereken kavramları öğretmek için maalesef kullanışlı olamamaktadır.

Basit kuramsal makine simulatörleri öğrenciler için bir sistemin iç işleyişine erişim sağlayarak (gerçek işlemciler bunu sağlayamaz) önemli bir rol üstlenmektedir. Uygun bir şekilde tasarlanmaları halinde, basit kuramsal makine simulatörleri ile aşağıda belirtilen temel kavramları canlandırmada üstündürler; Von Neumann mimarisi, mikro program konsepti, ardışık zamanlama devreleri, karmaşık sayı gösterimleri ve yapıları, temel komut kümeleri, komutların çözümlenmesi, komut alma ve yürütme döngüleri, yazıcıların kullanılması.

Özetle, kuramsal makine simulatörleri, öğretmen ve öğrencilerin karmaşık makineye bağımlı detaylar içinde kaybolmadan, önemli temel kavramlar üzerine odaklanabilmelerine imkân vermektedir. Simülasyonunu yaptıkları makinelerin basit yapıları sayesinde bu kategorideki araçlar internet tabanlı uygulamalar için de oldukça uygundur.

Basit Kuramsal Makine Simulatörleri (Simple Hypothetical Machine Simulators)	<ul style="list-style-type: none">• CASLE (http://shay.ecn.purdue.edu/~casle/)• CPU Sim (http://web.cs.wpi.edu/~cew/sigcse2002posters/skrien.html)• Simple Computer Emulator (http://beachstudios.com/sc/)• PIPIN (http://akademi.okulbilisim.com/elearning/cpusim.aspx)• Very Simple CPU (http://akademi.okulbilisim.com/elearning/vssimulator.aspx)
---	--

Yukarıda sıraladığımız makine simulatörleri sadece basit adresleme, sınırlı bir komut seti ve çok basit bir hafıza modeli kullanacak şekilde tasarlanmıştır. **Orta seviye komut kümelerine sahip simulatörler** ise aksine daha gerçekçi bir adresleme ve hafıza hiyerarşisi, daha kapsamlı komut seti ve bazen de bir kesme mekanizması içerebilmektedir. Mümkün olduğunca eksiksiz bir tasarım hedefi, bu simulatörlerin daha gerçekçi programlama teknikleriyle donatılması ile sonuçlanmıştır. Oldukça eksiksiz ve gerçekçi olan böylesi komut setine sahip bir simulatör kullanmanın diğer bir yararı, daha geniş ve ileri seviye müfredat içinde önemli olan mühendislik kavramlarını keşfetme olanağı sağlamasıdır. Öğrencilerin büyük bir kısmı, yüksek düzeydeki kavramları, dayandıkları düşük düzeydeki mekanizmaları inceledikten sonra daha iyi anlamaktadır.

Orta Seviye Komut Kümesi Simulatörleri (Intermediate Instruction Set Simulators)	<ul style="list-style-type: none">• LC2 (http://www.mhhe.com/patt/)• SIMHC12 (http://hcs12text.com/index.html)• AMD SimNow! (http://www.x86-64.org/downloads)• SPIM (http://www.cs.wisc.edu/~larus/spim.html)• EduMips64 (http://www.edumips.org/)• MIPS R2000 CPU Simulatörü (http://akademi.okulbilisim.com/elearning/psimulator.aspx)
---	--

Gelişmiş mikro mimari simulatörleri, mikro kod seviyesinde makine dili uygulamalarının (örneğin veri yolları, kontrol birimleri) gözlenmesine olanak sağlayacak şekilde tasarlanmıştır. Bu tür ileri düzey simulatörler, pipeline teknikleri, dallanma tahmini ve komut düzeyi paralelliği gibi performans arttıran tekniklerin avantaj ve dezavantajlarını araştırmak için kullanılabilir. Bu simulatörlerden bazıları mikro programlanabilir özelliktedir; bu sayede öğrenciler eğitim setlerini tasarlayarak deneyler yapabilir. Bu kategoride yer alan simulatörlerin çoğu müfredata paralel olacak şekilde bir ders kitabı ile birleştirilmiştir. Bu nedenlerle bilgisayar mimarisiyle ilgili ileri düzey dersler için uygundur.

Gelişmiş Mikro mimari Simulatörleri (Advanced Microarchitecture Simulators)	<ul style="list-style-type: none">• DLX (ftp://max.stanford.edu/pub/max/pub/hennessy-patterson.software)• Mic-1 Simulator (http://www.ontko.com/mic1/)• SimpleScalar (http://www.simplescalar.com/)• WinDLX (http://cs.uns.edu.ar/~jechaiz/arquitectura/windlx/windlx.html)• TkGate (http://www.tkgate.org/)• Deeds (http://esng.dibe.unige.it/netpro/Deeds/Index.htm)• Logisim (http://akademi.okulbilisim.com/elearning/logisim.aspx)• Hades (http://akademi.okulbilisim.com/elearning/hades.aspx)
--	--

Çoklu işlemci simülatörleri tek işlemci simülatörlerinden önemli ölçüde farklılıklar gösterir. Bu farklılıkların ilki, çoklu işlemci simülasyonunun tek işlemcilerde olmayan özellikleri de (örneğin paylaşılan bağlantı ağları ve paylaşılan hafıza) gerektirmesidir. Diğer önemli fark, eş zamanlı uygulama ihtiyacıdır; doğru bir simülasyon farklı işlemciler üzerinde komutların aynı zamanda gerçekleştiği olgusunu yansıtmak zorundadır. Üçüncü bir fark ise bir simülasyonu tamamlamak için gereken süredir. Simülasyon süresi benzetimi yapılan işlemcilerin sayısı ile doğru orantılı olarak artma eğiliminde olduğundan bu konu teknik açıdan sıkıntılıdır.

Sonuç olarak, iyi performansa sahip doğru çoklu işlemci simülatörlerinin geliştirilmesi başlı başına bir araştırma alanıdır. Yararlı eğitim yardımcıları olan çoklu işlemci simülatörler aynı zamanda bilgisayar mimarisine yönelik çalışmalarda aktif olarak kullanılan simülatörlerdir. Bu simülatörlerin kullanılması diğer kategorilerde yer alan tekli işlemci simülatörlerinin kullanılmasından daha karmaşık olabileceği için en uygun kullanma yeri ileri düzey bilgisayar mimarisi dersleridir.

Çoklu İşlemci Simülatörleri (Multi-Processor Simulators)	<ul style="list-style-type: none">• ABSS (http://arithmetic.Stanford.edu/~lemon/abss.html)• MINT (http://www.cs.rochester/u/veenstra/)• Proteus (http://www.ee.lsu.edu/koppel/proteus.html)• RSIM (http://rsim.cs.uiuc.edu/rsim/)• SimOS (http://simos.stanford.edu/introduction.html)• Wisconsin Simulator Page (http://www.cs.wisc.edu/arch/www/tools.html)
---	---

Başlangıç dersleri için en uygun olan simülatörler bilgisayarın işleyişini tanımlarken betimleyici ve canlandırıcı olmalıdır. Ancak ileri düzey öğrencileri anlamlı analiz, değerlendirme ve karşılaştırmaya olanak sağlayan detaylı performans istatistiklerine hazırlıklıdır. Bu öğrencilerin öğrendikleri ilk ders önbellek erişim oranı gibi CPU dışındaki faktörlerin performansı etkilediğidir. Son kategorimiz olan **öğretici sistem simülatörleri** işte bu ihtiyaca cevap verecek hafıza hiyerarşilerini modellemek, örneğin ağ simülatörleri gibi tamamen ayrı bir birim olarak ele alınabilecek modelleri analiz etmek için kullanılan simülatör örneklerini içermektedir [11-13].

Öğretici Sistem Simülatörleri (Didactic System Simulators)	<ul style="list-style-type: none">• Cacheprof (http://www.cacheprof.org)• CACTI (http://www.research.compaq.com/wrl/people/jouppi/cacti.html)• Dinero IV (http://www.cs.wisc.edu/~markhill/DineroIV/)• PRIMA (http://www.dsi.unimo.it/staff/st36/imagelab/prima.html)• Xcache (http://www.prism.uvsq.fr/archi/softs/XCache/)• PCSpim-Cache (http://www.disca.upv.es/spetit/spim.htm)• PSATSIM (http://www.ces.clemson.edu/~tarek/psatsim/)• R2NP (http://www.inf.pucminas.br/professores/cota/r2np/r2np-e.html)• MSCSim (http://www.msccsim.com/)• WebMIPS (http://bellerofonte.dii.unisi.it/WEBMIPS/)
---	---

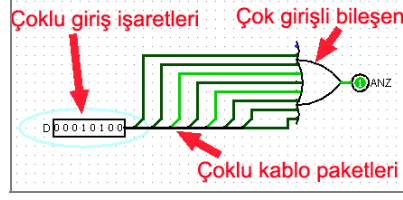
Şu ana kadar geniş bir seviye aralığındaki derslere kolaylıkla entegre edilebilecek öğretim materyalleri olan simülatörler değerlendirilmiştir. Önerilen simülatörlerin yukarıda değinilen eğitim yaklaşımları dikkate alınarak diğer öğrenim araçlarıyla birleştirilmesiyle sayısal sistemler ve bilgisayar mimarisi gibi gün geçtikçe soyutluk derecesi artan mühendislik eğitimlerinde aktif öğrenmeyi mümkün kılacaktır.

Simülatörlerle ilgili teknik kavramlar

Simülatörlerin teknik özellikleri hiç şüphesiz kullanılabilmesi için eğitim alanı ve seviyesi ile yakından ilişkilidir. Simülatörleri bu özelliklerine göre incelediğimizde gerçekten de teknik kapasiteleri bakımından ciddi farklılıkları olduğunu görmekteyiz. Teknik düzeyde yapılacak bu değerlendirmelere geçmeden önce konuyla ilgili kavramların kısaca üzerinden geçilmesinin faydalı olacağını düşünmekteyiz.

Bağlantılar (yollar): Bilgisayar donanım simülatörlerinde genellikle iki nokta arasındaki elektrik bağlantısını simgeleyen yollar, geçitlerin giriş ve çıkışları arasındaki tekil çizgilerdir (Şekil 2). Müfredatın seviyesine paralel olarak seçilebilecek daha gelişmiş simülatörlerde yollar çoklu kablo paketleri olarak gösterilebilmektedir. Bu sayede tasarlanan devreler daha temiz görünmekte ve böylelikle daha karmaşık devrelerin kablo/bağlantı karmaşalarından uzak bir şekilde

tasarlanabilmesine izin verilmektedir. Çoklu bağlantı yolları için kablo sayısının kontrol edilmesi simülâtörün temel görevlerindedir. Bağlantılar sırasında veya simülasyonda mantık geçidi çıkışlarının kontrol edilmesi, örneğin iki standart geçit çıkışlarının doğrudan bağlanması gibi hata durumlarının doğrudan uyarılarla kullanıcıya bildirilmesi gerekir.



Şekil 2. Donanım simülâtörlerinde çoklu bağlantı ve girişlerin gösterimi

Bağlantılardaki sayısal değerler: Tekil bağlantı yollarının simülasyon sırasında grafik arayüzünde farklı renklerle gösterilmesi artık geleneksel hale gelmiştir. Daha gelişmiş simülâtörlerde “0” veya “1” seviyesi yanında başlangıç seviyesinde daha koşullandırılmamış işaretler, simülasyon sırasında tanımlanmamış seviyeler ve hatta mantıksal seviyeleri kuvvetli olmayan (weak logic) işaretler bile farklı renklerle simgelenmektedir. Simülâtörlerdeki kablo paketlerini simgeleyen çoklu işaret göstergelerinin de kafa karıştırıcı nitelikte olmaması gerekmektedir.

Çok girişli geçitler: Başlangıç seviyesindeki simülâtörler 2 veya 3 girişli geçit devrelerini kütüphanelerindeki farklı bileşenler olarak gösterirken, gelişmiş simülâtörler bu tür bileşenlerin giriş ucu sayısı dahi esnek tutulmaktadır. Bunun yanında kontrol edilebilen üç durumlu yol vericileri de ortak veri yolu tasarımları için şarttır.

Saat ve zamanlama diyagramları: Simülâtörlerin zaman içindeki davranışlarının öğrenci tarafından izlenebilmesi için bir saat devresi gerekmektedir. Saat devresi esas alınarak çizilecek zamanlama diyagramlarında özellikle ardışık devrelerin çalışmaları irdelenebilir. Zamanlama diyagramlarının daha gerçekçi olabilmesi için geçitlerin gecikme değerleri simülâtörlerde değiştirilebilir bir parametre olarak algılanabilmeli, yarış şartları (race conditions) ve CPU çevrimi diyagramlarında izlenebilmelidir.

Geleneksek devreler ve Flip-Flop’lar: Simülâtörün kütüphanesi geleneksek basit geçit devreleri, Flip-Flop’ları ve tüm ardışık devreleri içeren kütüphanesi olmalıdır.

Kompozit öğeler: Sayısal devrelerin kolaylıkla kurulabilmesi için kompozit öğeler (aritmetik devreler, çoklayıcılar, bilgi dağıtıcılar, yazıcılar, sayaçlar, bellekler, vs.) bulunmalıdır.

Doğruluk çizelgesi: Giriş seviyesindeki simülâtörler basit mantıksal fonksiyonlar için hantal devre spesifikasyonlarına başvurma gereksinimini ortadan kaldıran doğruluk çizelgelerini içermelidir.

Kara Kutu: Öğrenci simülâtör kütüphanesindeki kompozit öğeler yanında kendi oluşturacağı alt devreleri özetleyebilen “kara kutu”lar tasarlayabilmeli ve devrenin görünümünü sadeleştirmek için isteğe bağlı iç içe alt devreler ekleyebilmelidir. Kara kutu olarak 7400TTL entegreleri gibi gerçek hayata yakın örnekler barındırmalıdır.

Durum Diyagramları: Karmaşık işaretlerin incelenmesi, yollardaki protokollerin irdelenmesi ve örneğin bir işlemcinin kontrol işaretlerinin tasarımı gibi devrelerin değişik konumları durum diyagramları olarak incelenebilmelidir.

Saklama ve yazdırma (Save / Print circuit): Tasarlanan devreler kaydedilebilmeli ve yazıcıdan şema olarak baskısı alınabilmelidir.

Önerilen Simülâtörlerin Teknik ve Tasarım Özellikleri

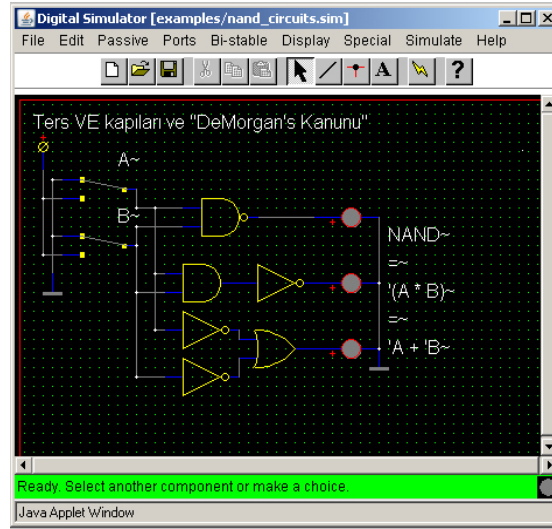
Çalışma içinde önerilen ve eğitsel yaklaşımlar açısından değerlendirilen simülâtörler ilk olarak teknik özellikleri ve yazılım tasarım ilkelerine göre değerlendirilecektir. Teknik karşılaştırma için zaruri olan kavramlar yukarıda verilmiştir, “Shneiderman’ın 8 Altın Kuralı” [14] olarak bilinen yazılım tasarım ilkeleri ise şöyle sıralanmaktadır; (1) Tutarlılık için çabalamak, (2) Sık sık kullanıcılara kısa yol kullanma imkânı verme, (3) Bilgilendirici geri dönüşler sunmak, (4) Tartışmayı

bitiren diyaloglar tasarlamak, (5) Hataları önleme ve basit hataları yönetme, (6) Hareketlerin kolayca geri alınmasına izin vermek, (7) İç kontrol noktalarını desteklemek, (8) Kısa süreli hafıza yüklemesini azaltmak [15].

Burada özellikle bir hususun altının çizilmesi gereklidir; “önerilen simülátörler” sadece teknik kapasiteleri ve tasarım ilkeleri esas alınarak değil, ilerleyen bölümlerde detaylı olarak değineceğimiz eğitimin amaç, kapsam ve içeriği ile bir bütün olarak değerlendirilerek belirlenmiştir.

DigSim (Digital Workshop)

Sayısal devreler giriş derslerinden itibaren kullanılabilir ve tarafımızdan önerilen ilk simülátör DigSim – Digital Workshop v1.10b1 (<http://akademi.okulbilisim.com/elearning/simulator.aspx>) tekil bağlantı yolları ve anahtarlama teknikleri eşdeğer devreleri açısından öğrenmeye en uygun olanlardan biridir. Bu simülátör 1996 yılında Iwan van Rienne tarafından tasarlanmış daha sonra Deborah E. Lynch tarafından da geliştirilmiştir. Simülátör ANSI/IEEE 91–1984 standartlarıyla tanımlanmış geleneksel sayısal devre şekilleri ile gösterilen geniş bir bileşen kütüphanesine sahiptir. Simülátörlerin incelendiği internet sitemizde birçok sayısal tamamlanmış devre örneği bulunmaktadır.



Şekil 3. DigSim simülátörü çalışma penceresi görünümü

DigSim simülátöründe öğrenci “0” veya “1” leri oluşturan işaretlerin nasıl oluşturulduğunu daha kolay kavrar. Bağlantılar, sayısal işaretleri oluşturan elektriksel gerilim seviyeleri ve geçitler gösterim açısından açık ve basittir.

DigSim’in temel eleştirisi alan grafik özelliği tersi beklendiği halde, yeşil rengi mantıksal “0” seviyesi için, kırmızı rengi de mantıksal “1” seviyesi için temsil edilmesidir. Basit bir sayısal devre simülátörü olarak önerilen DigSim sadece sayısal devrelere ait kompozit bileşenlerle dolu geniş bir kütüphaneye sahiptir ve bunun yanı sıra basit bir zamanlama diagramı yeteneği de bulunmaktadır. Ancak kablolar gruplanamaz, alt devre veya kara kutu devre çizimi mümkün değildir. Bu durum simülátörün daha ileri düzeyde bilgisayar bilimleri ders konuları için yetersiz olmasına neden olur. Durum diyagramları ve doğruluk tabloları yoktur, otomatik hata ayıklanamaz [16].

Sayısal devreler başlangıç derslerinde sayı sistemleri ve 2 tabanlı matematik bölümlerinden sonra mantıksal geçitlerin sanal olarak anlatımında DigSim kullanılabilir. DigSim’in basit tel bağlantılı çalışma şekli sayesinde öğrencilerin mantıksal veri seviyeleri “0” ve “1” ile elektriksel devre seviyeleri “yüksek voltaj” veya “toprak seviyesi” arasındaki paralelliği daha doğru kavrayacağını düşünüyoruz.

DigSim ile öğrenci sayısal devrelerin temel konuları geçitler, bileşik devreler, flip-floplar, sayıcılar ve temel yazıcıları simüle edebilir, giriş ve çıkış işaretlerinin zaman içindeki davranışlarını izleyebilir ve karşılaştırabilir. Sayısal

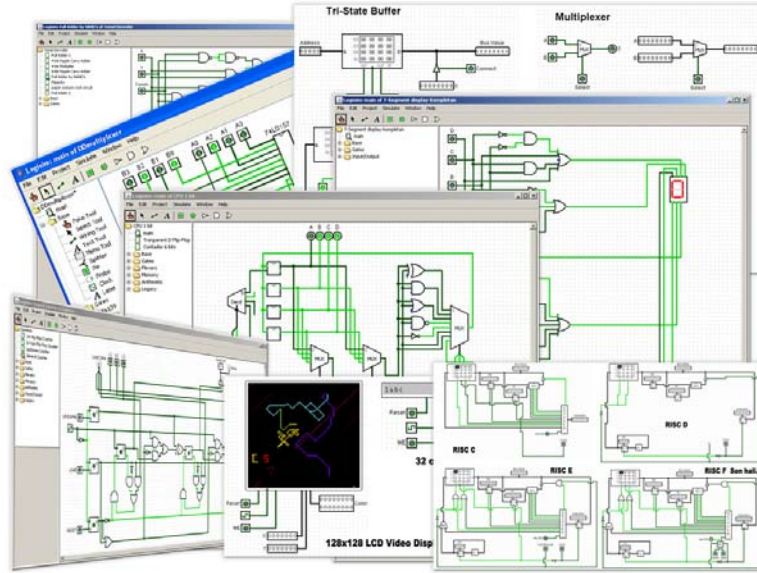
devreler dersleri için son derece gerekli bir öge olan doğrulama tablosu içermediğinden DigSim'in bir "Karnough Haritası" simülasyonu (<http://sourceforge.net/projects/k-map>) ile birlikte kullanılması önerilir.

Sayısal devre simlatörleri arasında benzer yeteneklere sahip ve gerçek hayata daha yakın örneğin bread-board tarzı ve klasik TTL 74xx entegrelerini görsel olarak daha açık simüle eden araçlar vardır. Ancak sayısal devreler o kadar ilgi çekicidir ki öğrenciler bu sanal deneme panoları üzerinde devre geliştirerek oldukça fazla zaman harcayabilir. Amaç öğrencilerin, geçitlerin daha karmaşık fonksiyonları gerçekleştirmek üzere nasıl bir araya geldiğinin mantığını kavraması olduğundan sanal deneme panolarının gerçeğe yakın görsel sunum araçları bazen gereksiz yere vakit kaybına neden olabilir. Buna rağmen sayısal devreler giriş eğitiminde, kısa bir süre için dahi olsa Java BreadBoard Simulator tarzı bir aracın öğrencilere gösterilmesinin doğru olacağını düşünüyoruz.

Nicholas Glass tarafından sunulan Java Digital Breadboard Simulator (<http://www.cs.york.ac.uk/netpro/bboard/>) gerçek deneysel ortama daha yakın bir görsellik sunmaktadır. Öğrenci laboratuvarlarda görebileceği Bread-Board tarzı sanal panolar üzerinde, TTL entegreler içindeki geçitler ile çalışma sırasındaki elektriksel bağlantıları, gerçek geçit davranışlarını ve sayısal kompozit devreleri izleyebilmektedir [17]. Ancak sayısal devrelerin "oluşturulması ve incelenmesi" amacıyla şematik bir simülasyon kullanılması, öğrenci açısından pedagojik değeri şüpheli olan deneme panolarına göre daha kolay olduğu ve hızlı sonuçlar alınabileceği düşüncesindeyiz.

Logisim

Carl Burch tarafından geliştirilen Logisim (<http://elearning.cizgi.com.tr/logisim.aspx>) pedagojik açıdan üstünde durduğumuz en önemli simülatördür. Logisim'de basit geçitler, kompozit ögeler, alt devreler ve kara kutular ve ayrıca geniş yelpazedeki diğer öğeleri ile Bilgisayar Organizasyonu ve Mimarisi dersleri için idealdir. Çoklu kablo paketleri ve bunların bağlantılarının simülasyon tarafından kontrol edilmesi özelliği, çoklu giriş-çıkış simgeleri ve bileşenlerin yönleri ile giriş adetlerinin tanımlanabiliyor olması temiz bir çalışma ortamı sağlamakta ve mimari dersleri için kullanılacak bir simülasyon olarak Logisim'i öne çıkarmaktadır.



Şekil 4. Logisim simülasyonu çalışma pencerelerinden görüntüler

Logisim'de temel doğruluk çizelgesi, 74xx TTL entegreler alt devre kütüphanesi ve basit saat işareti ile ilk anda çok uygun bir sayısal devre tasarımı ve simülasyonu için grafik bir sistem olarak görülse de sayısal elektronik alanındaki tüm giriş derslerinin önemli bir kısmını oluşturan ardışık devreler konusunda sağladığı destek eksiktir. Zamanlama diyagramları ve dolayısıyla girişlerdeki sayısal işaretlerinin etkisi ile çıkışlarda zaman içinde değişen davranışlar

izlenemekte, sayısal devrelerin incelenmesinde önemli bir yer teşkil eden durum diyagramları ve yayılım gecikmeleri hiçbir şekilde anlatılamamaktadır [18-19].

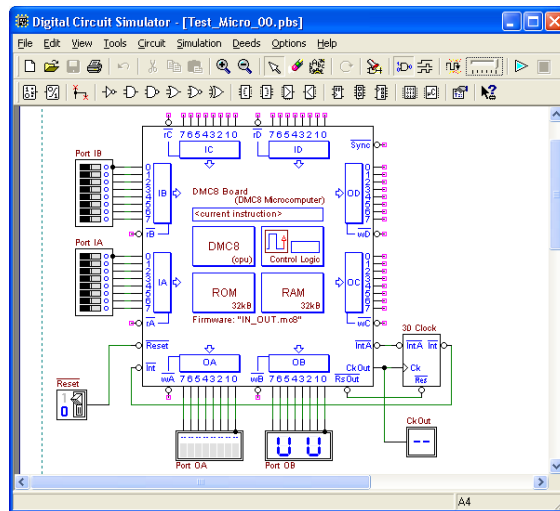
İletişime geçtiğimiz programı geliştiren Carl Burch bu eleştirileri bildiğini ancak gelecek için planlarında zamanlama diyagramları ve durum diyagramları yerine sadece VHDL/Verilog arayüzü planları olduğunu söylemiştir. Bunun üzerine Carl Burch ile iletişim içinde, Çizgi Söğüt Gölgesi olarak zamanlama diyagramları ve diğer ardışık devre araçlarını Logisim programına ekleme projesi oluşturmuş bulunmaktayız.

Tüm eksiklerine rağmen Logisim'i önermemizin temel nedeni programı geliştiren çok farklı gruplar olması ve yapısı gereği programın gelişmeye açık olmasıdır. Alt devreler ve kara kutular ile öğrenci kendi projesinde kullanmak üzere farklı bir kütüphane oluşturabilir. Hiyerarşik tasarım yapısı ile Logisim'de öğrenci örneğin sadece geçitler kullanarak bir ALU (Aritmetik Logic Birim) ve bu alt devreyi kullanarak basit bir işlemciyi sanal olarak tasarlayabilir. Logisim'de Bilgisayar Organizasyonu derslerinde anlatılan yazıcılar, ROM, RAM, aritmetik bileşenler (toplama, çıkartma, çarpma, bölme, vs.), 7-Parçalı gösterge, LCD ekran, Satır göstergesi, Klavye ve bu bileşenlerin kobinasyonu simülatörün dâhili kütüphanesinde veya örnekleri arasında bulunmaktadır. Programa dahil olan alt devrelerin hiyerarşik olarak bir üst devrede gösterimi sadece dikkörtgen olarak mümkünse de Carl Buch bunun çözümünün planları içinde olduğunu bildirmiştir.

Logisim'in, ders konuları içinde sayısal devrelerin daha az yer tuttuğu, özellikle bilgisayar mimarisi konularında kullanılmasını öneriyoruz. Düzenleme hatalarını önleyen tasarımı, kolay kullanım özellikleri, geniş kütüphanesi ve örnekleri ile Logisim, CPU simülasyonu ve eğitimi için idealdir.

Deeds (Digital Electronics Education and Design Suite)

Deeds (Digital Electronics Education and Design Suite), Giuliano Donzellini ve Domenico Ponta tarafından geliştirilen bir sayısal tasarım proje platformudur (<http://elearning.cizgi.com.tr/deeds.aspx>). Etkileşimli sayısal devre tasarım simülatörü (d-DcS), sonlu durum makinesi modülü (d-FsM), mikrobilgisayar kartı devresi ve emülatörü (d-McE) ile donatılmış bulunan Deeds ile öğrenciler, standart sayısal bileşenler, kompleks öğeler ve kara kutular ile gereğinde mikrobilgisayar kartı aracılığıyla kontrol edildiği sistemleri tasarlayabilmekte ve güncel sayısal tasarımlarda olduğu gibi eşzamanlı olarak simülasyonunu inceleyebilmektedir. Elbette tüm modüller, sistem parçalarının ayrı ayrı test edilebilmesi amacıyla bağımsız olarak da kullanılabilir. Sonlu durum makinesi simülatörü (d-FsM), FSM bileşenlerinin, ASM (Algoritma Durum Makinesi) paradigması kullanılarak grafik olarak düzenlenmesini ve simülasyonunu mümkün kılar. Bu araç kullanıcı tarafından tasarlanan FSM'nin, durum ve zamanlama gelişimi arasındaki ilişkilerinin çalışma gösterimi aracılığıyla lokal fonksiyonel simülasyonunu sağlar. d-FsM tarafından oluşturulan bileşenler doğrudan sayısal devre tasarım simülatörü d-DcS'de kullanılabilir ve herhangi bir sayısal devreye eklenebilir [4,10, 20-22].



Şekil 5. Deeds simülatörü çalışma penceresi görünümü

Deeds sayısal devre derslerine ait birleşik devreler, ardışık devreler, durum diyagramları, son durumlu makinalar konularında örnekleri ve d-Ast (Assistant Browser) yardımcı arayüzü ile öğretmen dostu bir platformdur. Deeds tüm program bileşenlerinin içinden bağlanılabildiği, ders planını ve örnekleri yönlendirildiği bir listeleyici (browser) ekranından çalışmaktadır. Deeds içindeki tüm editörler ve simülatörler bu listeleyici içinden birbirleriyle etkileşimli olarak çalışırken kara kutular ve durum diyagramlarından hareketle oluşturulmuş alt devrelerin tasarım sırasında kullanılan sayısal devreler ile etkileşimi de ayrıca incelenebilmektedir. Sayısal devre tasarım editörü, tasarlanan elektronik devreleri aynı zamanda VHDL süreçleri olarak aktarabilmektedir [4, 21].

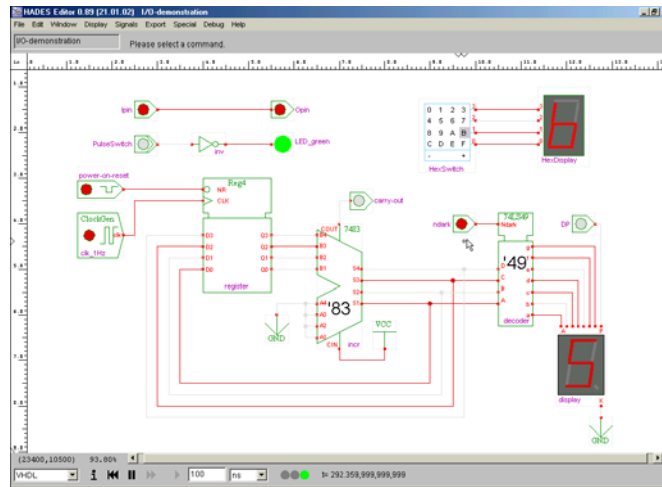
Uygun grafik arayüzü araçları, iyi simülasyon performansı, çok seviyeli iç içe geri al özelliği, interaktif ve zamanlama diyagramlı simülasyon modları Deeds'i öne çıkaran teknik ve tasarım özellikleridir. Ancak giriş ve çıkışların, saat devresinin temel özelliklerinin değiştirilememesi, yollar üzerindeki işaretlerin özellikle simülasyon sırasında karışabilmesi ve eksik olan gerçekleştirme tablosu önemli dezavantajlardır. Özellikle eksik olan çoklu kablo paketleri ve girişleri, tanımlanan sayısal geçitlerin sınırlı olması nedeniyle bir Bilgisayar Mimarisi simülatörü olarak önerilemez. Sınırlı yetenekleri ile sayısal devre tasarım simülatörü d-DcS'de eksik ve yanlış bağlantılar gibi devre hatalarını tespit etmek oldukça zor ve zahmetli olabilmektedir.

Ancak Deeds, kendisinden beklenmeyen şekilde yetenekli bir d-McE "Microcomputer Board Emulator" içermektedir. Bu emülatör bilinen en basit mikroişlemcilerden 8-Bit Z80 uyumlu işlem kodları ile uyumlu olup yine sürpriz bir şekilde d-DcS sayısal devre tasarım editörü içinden de çağırılabilen ve tasarlanan sayısal devreler ile birlikte bir kara kutu olarak (DMC-8 AP) simüle edilebilmektedir. DMC-8 AP mikroişlemci kara kutu devresi aynı zamanda 32K-ROM, 32K-RAM ve diğer bazı giriş çıkış devrelerinin olduğu sanal bir devre olup 2 geçişli bir asambler ve görsel bir hata ayıklayıcı (debugger) ile donatılmıştır [4, 21]. Bu mikro işlemci editörü, yardımcı aracı ve devrenin diğer sayısal devrelere eklenebilmesi özellikleri ile Deeds, sayısal devreler ve mikroişlemciler dersleri için idealdir.

Hades

Hamburg Üniversitesinde Norman Hendrich tarafından geliştirilen Hades, java tabanlı sayısal devre simülasyonu yapabilme yeteneğine sahip, Windows, Linux, Solaris, Macintosh işletim sistemlerinin tamamında ve aynı zamanda internet üzerinden e-öğrenme aracı olarak çalışabilecek gelişmiş bir tasarım/egitim platformudur. Hades, kendi eğitimlerinizi geliştirebileceğiniz bir üst çatıdır (<http://elearning.cizgi.com.tr/hades.aspx>).

Hades'in, diğer profesyonel tasarım araçlarının çoğunda olmayan basit fakat etkin tasarımı, eğitmenin teorik ders anlatımını azaltabilmesini bunun yerine simülasyon ile öğrencilerin sanki bir "oyun" oynar gibi çalışabilmelerini sağlar. Bunun yanında, en alt seviyeden başlayarak yukarıya doğru büyüyen hiyerarşik tasarım imkânları, Hades'i sadece öğrenciler için değil, yazılım ile uğraşmak istemeyen ancak daha kompleks devreler ve simülasyonlar hazırlamak isteyen ve bunların anlatımlarını hedefleyen eğitimciler için de ideal kılar.



Şekil 6. Hades simülatörü çalışma penceresi görünümü

Tüm applet'leri Java dilinde yazılmış Hades güçlü yazılım dili özelliği, değişik süreçlerde ele alınabilen simülasyon ve hata ayıklama teknikleri ve görsel tasarım özellikleri sayesinde Hades ile komple sistem seviyesinde test değerlendirmeleri yapılabilir. Hades'in çekirdeğine erişip kendi simülasyon modelinizi geliştirme imkânı da mevcuttur, bir örnek olarak bu şekilde hazırlanan RaVi (Visualization of Computer Architecture) incelenebilir (<http://ls12-www.cs.uni-dortmund.de/ravi/ravi-documentation.html>).

Nesne yönelimli yazılımı ile Hades simülasyon çatısı altında birçok temel tasarım süreçleri gerçekleştirilebilir. Std logic ve std logic vector çoklu seviyeli mantıksal tasarımları, geçit seviyesinde, RTL (Register Transfer Language) seviyesinde ve sistem seviyesinde sayısal devreler ile gerçekleştirip, simülasyonu yapılabilir. Tasarımlar tümleşik devrelere dönüştürülebilir yeniden konfigüre edilebilir, davranış izleme ve hatta sistem izleme seviyesinde simüle edilebilir. Native Java içeren veya gömülü sistemlerde herhangi bir kod içeren donanım ve buna bağımlı yazılım birlikte simüle edilebilir ve yazılımı değiştirilerek simülasyon tekrarlanabilir. Kendi modeller oluşturulabilir, bu modelleri diğer tasarım bileşenleri ile birleştirilebilir. Tasarımlar RTL veya geçit seviyesinde Synopsys uyumlu VHDL [IEEE-93a] diline dönüştürülebilir. Tasarımlar kaydedilerek saklanabilir, diğer xfig veya jfig grafik editörlerinde kullanılabilir yüksek kalitede Poscript formatında paylaşılabilir.

Hades'te çok geniş bir bileşen kütüphanesi bulunmaktadır. Temel ve 6 girişe kadar kompleks geçitler, tüm standart flip-flop'lar, interaktif anahtarlar, tekil ve 7'li LED'li göstergeleri içeren çok değişik yelpazede giriş çıkış devreleri, klasik 74xx TTL devrelerin davranış modelleri, std logic vector'e dayalı geniş RTL bileşenler kütüphanesi, grafik editöre sahip RAM, ROM ve MicroCode devreler, LCD-göstergeleri, VT52/VT100 standardında bir metin terminali, UART/USART, FSM (Finite State Machine) sonlu durum makineleri düzenlenebilecek ve durum diyagramları çıkarılabilecek bir editör uygulaması, PIC16Cxxx ve MIPS R3000 ailesine ait, hem yazılım hem de donanım olarak incelenebilecek mikro kontrollü devreler, öğrencilerin kendi kendine analizlerini değerlendirebilmeleri için LFSR- ve BILBO-yazıcıları ve özel imza analizi bileşenleri [23].

Hades, Elektronik Mühendisliği ve Bilgisayar Mühendisliği'nin her ikisini de kapsayacak geniş bir müfredat içinde tüm simülasyon modellerini içerdiği gibi "Linear-feedback shift registers" doğrusal geri besleme ve BILBO yazıcıları seçeneği ile yapılan tasarımların öğrencinin kendi başına doğruluğunu test etmesine de imkân tanır.

Hades'in etkileşimli simülasyon yetenekleri; "glow-mode" denilen yolların, anahtarların, giriş ve çıkış uçlarının veya göstergelerin değişik durumlarda dirençlerin üzerindeki benzer DIN/IEC formatında 16 değişik renk olarak simülasyonundan, saat işaretinde sadece frekansın değil darbe/boşluk sürelerinin bile tanımlanabilmesine kadar tüm detaylar oldukça etkileyicidir. Yazıcı transfer tasarım seviyesinde (Register-transfer level) HADES çatısı ile geleneksel olarak laboratuvar, ev ödevleri ve projelerde kullanılan simülâtörün ders anlatırken de kullanılabilmesi düşünülmektedir. RTLIB kütüphanesindeki bir ALU'nun özellikleri tanımlanabilmekte ve Hades çatısı sanki yeniden düzenlenebilir bir animatöre dönüşmektedir.

Temel ALU fonksiyonlarından karmaşık komple mikro kodlu bir işlemci makinesinin geçitler ve register-transfer seviyesinde bu görsel anlatım yeteneği ile Hades, e-öğrenme platformu olarak idealdir. Hades ile yollar üzerindeki verilerin yayılımı, geçit gecikmeleri, hafıza birimlerindeki benzer gecikmeler bile simüle edilebilir hatta hız aşımı simülasyonları yapılabilir, çevrim bazında simülasyonlar incelenebilir. Çok seviyeli yakınlaştırma özelliği, yine çok seviyeli iç içe geri al ve yinele yeteneği, taşınan bir bileşenin yolları ile birlikte taşınabilmesi ile Hades, tasarım açısından da öne diğerlerinden daha öne çıkmaktadır.

Hades'e getirilebilecek tek kritik yazılım tasarımın 8 temel ilkesine çok sadık kalamamasıdır. Yazılım kısıtlamalarından doğan tasarım çizgileri (grid) ve iki imleç kafa karıştırmaktadır. Bazı eleştirmenlerin Hades'i hantal veya kullanılmasını zor bulmaları bundan kaynaklanmaktadır. Hades'i sadece üniversite seviyesinde özellikle elektronik mühendisliği müfredatı için öneriyoruz.

Simülâtörlerin Değerlendirilmesi

Simülâtörlerin değerlendirilmesinde esas alınacak kriterler için belirleyici faktör hiç şüphesiz öğretmen ve öğrencilerin ihtiyaçlarıdır. Poplawski tarafından, özeld bir sayısal devre simülâtörü olan JSL üzerine yapılan çalışmada öğrenci ve öğretmen ihtiyaçlarına yönelik olarak şu değerlendirmeler yapılmıştır [24]:

Öğrenciler açısından her şey mümkün olduğunca sezgisel ve öğrenmesi kolay olmalı, kademeli bir öğrenme stratejisi izlenmelidir. Kullanılan simülâtör giderek daha karmaşık devreleri oluşturmaya yönlendirmeli, kapsamlı bir çevrim içi yardım desteği bulunmalı ve sistem mantık dışı seçimler gibi basit kullanıcı hatalarına karşı uyarı sistemine sahip olmalıdır. “Geri al” ve “yinele” gibi özelliklere sahip olmalı, bu sayede hataya sebep olan uygulamalar kolaylıkla elenebilmelidir. Kes/yapıştır komutları desteklenmeli, bu sayede devreler daha kolay bir şekilde hazırlanabilmelidir. Hata ayıklama son derece kolay olmalı, özellikle belirli bir devrenin yavaş simülasyonu ile sinyal ve hafıza değişkenlerinin değerleri belirlenebilmeli ve görüntülenebilmelidir. Bunların dışında, geliştirilen araç son derece portatif olmalı, bu sayede öğrenci tarafından istenilen her platformda rahatlıkla kullanılabilirdir.

Öğretmen açısından yine öğrencilerin beklentilerine uyumlu şekilde devre öğeleri rahatlıkla yerleştirilebilir olmalı (sürükle-bırak, kes-yapıştır), kablo çizimi gibi öğelerin bağlanması işlemleri basit bir şekilde (örneğin fare hareketleriyle) yapılabilir. Öğeleri ve bağlantıları yeniden düzenlemek kolay olmalı, ders öncesi oluşturulan alt devreler minimum çaba ile alınıp sisteme entegre edilebilmelidir. Yine aynı şekilde ders öncesi oluşturulan giriş sinyal değerleri ve/veya sıraları, bellek ögesi değerleri desteklenebilmelidir. İstenilen devre bileşenin simülasyonu son durumun rahat bir şekilde görüntülenebilmesi amacıyla aşamalı olarak yavaş veya hızlı bir şekilde ilerletilebilmeli, yani eğitimcinin kontrolünde olabilmelidir. Sinyal değerleri ve hafıza ögesi değerleri kolaylıkla görüntülenebilmeli ve gerçek zamanlı olarak izlenebilmelidir.

Bu ihtiyaçlara cevap verebilecek nitelikte eğitim alanlarına yönelik ticari ürünlerin bir bölümü: *Xilinx* (<http://www.xilinx.com>), *Altera* (<http://www.altera.com>), *OrCAD* (<http://www.cadence.com/products/orcad>), *NI MultiSim* (öncesinde Electronics Workbench) (<http://www.ni.com/academic/multisim>), *Tina Design Suite* (<http://www.designsoftware.com>), *Digital Works* (<http://www.spsu.edu/cs/faculty/bbrowncircuits/howto.html>), *MacroSim* (<http://www.engineeringsoftware.com/pr/addProd106.htm>), *Proteus* (http://www.labcenter.co.uk/index_uk.htm) şeklinde sıralanabilir. Ancak biz hem bu öğretmen ve öğrenci ihtiyaçları, hem de yukarıda detaylı bir şekilde üzerinde durduğumuz teknik özellikleri ve tasarım ilkelerine göre şu ana kadar adından söz ettiğimiz tüm simülâtörleri değerlendirdiğimizde bizim için 4 simülâtör öne çıkmaktadır: **DigSim**, **Logisim**, **Deeds** ve **Hades**.

Önerdiğimiz simülâtörler gerek grafik özellikleri, devre inceleme ve biçimlendirme metotları, kullanım kolaylığı, kapsamlı hata kontrolü ve simülasyonu görüntüleme özellikleri gibi işlevleri bakımından ticari ürünlerden geri kalmadıkları gibi açık kaynak kodlu, hâlihazırda ilgililer tarafından geliştirilmesine devam edilen ve internet ortamında ücretsiz olarak kullanıma sunulan araçlar olmaları nedeniyle bizce diğerlerinden daha üstündür.

Öğrenci ve öğretmen ihtiyaçlarını ortak bir şekilde ele alınmasıyla oluşan kriterlere göre çalışma içinde tarafımızdan önerilen simülâtör araçları değerlendirildiğinde Tablo 1’deki sonuçlar elde edilmektedir.

Tablo 1. Önerilen simülâtörlerin öğrenci ve öğrenmen ihtiyaçlarına göre değerlendirilmesi

Kriterler	DigSim	Logisim	Deeds	Hades
Kullanım Kolaylığı	😊😊😊	😊😊😊	😊😊	😊
Grafik Arabirimi	😊	😊😊	😊😊😊	😊😊😊
Platform Esnekliği	😊	😊😊😊	😞 (Windows)	😊😊😊
Sayısal Devreler	😊😊😊	😊😊😊	😊😊😊	😊😊😊
FSM	😞	😞	😊😊😊	😊😊😊
Zamanlama Diyagramları	😊	😞	😊😊😊	😊😊😊
Mikro denetleyiciler ve CPU	😞	😊😊😊	😊😊😊	😊😊😊
Kodlama ve Assembler Dili	😞	😞	😊😊😊	😊😊😊
VHDL Desteği	😞	😞	😊😊😊	😊😊😊
Popüler Kullanım	😊😊😊	😊😊😊	😞	😞
Diğer		Java (+)	Education Suite (+)	Java (+)

Tablo 2. IEEE/ACM Bilgisayar Müfredatı üzerinde simülatorlerin değerlendirilmesi

Digital Logic (CE-DIG), Computer engineering - Computer Architecture and Organization (CE-CAO), Computer Science - Architecture and Organization (CS-AR)		DigSim	Logisim	Deeds	Hades
Digital logic	Basic concepts, number systems, codes, binary arithmetic, Boolean algebra, basic gates,	✓	✓	✓	✓
	Combinational logic systems, multiplexers, demultiplexers, decoders, encoders, comparators	✓	✓	✓	✓
Arithmetic- Logic unit	Integer arithmetic unit: Adders, subtracters, carry lookahead	✓	✓	✓	✓
	Arithmetic and logic units (ALUs)	✓	✓	✓	✓
	Integer arithmetic unit: Multiplication and division		✓	✓	✓
	Floating-point arithmetic unit				✓
Memory elements	Sequential systems: Flipflops, latches, Registers	✓	✓	✓	✓
	Memory systems, RAM & ROM		✓	✓	✓
	Memory system design, Memory hierarchy		✓	✓	✓
	Sequential logic circuits: Finite state machines (FSMs), synchronous and asynchronous circuits,	✓		✓	✓
	Sequential units: Data&shift registers, counters, sequence detectors, synchronizers, debouncers, controllers	✓	✓	✓	✓
Fundamentals of computer architecture	Instruction formats, instruction types, cycles, register files, addressing modes		✓	✓	✓
	Basic organization of the von Neumann machine		✓		✓
	Control unit; instruction fetch, decode, and execution		✓		✓
	Simple processors: Datapath		✓		✓
	Simple processors Control unit: Hardwired realization		✓		✓
	Simple processors Control unit: Microprogrammed realization		✓		✓
Assembly language	Introduction to assembly language			✓	✓
	Assembly/machine language programming			✓	✓
	Programming in assembly language: Instruction formats, Addressing modes			✓	✓
	Procedures, Subroutine call and return mechanisms, I/O and interrupts			✓	✓
Input/Output	I/O fundamentals: handshaking, buffering,			✓	✓
	I/O techniques: programmed I/O, interrupt-driven I/O, DMA		✓	✓	✓
	Interrupt structures: vectored and prioritized, interrupt overhead,		✓	✓	✓
	Memory system design and interfacing		✓	✓	✓
	Buses: bus protocols, local and geographic arbitration		✓	✓	✓
Processor systems design	The CPU interface: clock, control, data and address buses		✓	✓	✓
	Address decoding and memory interfacing		✓	✓	✓
	Basic parallel and serial interfaces		✓	✓	✓
	Timers		✓	✓	✓
	System firmware		✓	✓	✓
Organization of the CPU	Implementation of the von Neumann machine		✓		✓
	Single vs. multiple bus datapaths		✓		✓
	ISA; machine architecture as a framework for encapsulating design decisions		✓		✓
	Relationship between the architecture and the compiler			✓	✓
	Implementing instructions		✓		✓
	Control unit: hardwired realization vs. microprogrammed realization		✓		✓
	Arithmetic units, for multiplication and division		✓		✓
Basic pipelining & Advanced processors	Instruction pipelining		✓		✓
	Trends in computer architecture: CISC, RISC, VLIW, Pipelined processor				✓
	Introduction to instruction-level parallelism (ILP)				✓
	Pipeline hazards: structural, data and control		✓		✓
	Reducing the effects of hazards		✓		✓
	Examples of contemporary processors		✓	✓	✓
	Introduction to multiprocessor systems				✓
	Microprocessors			✓	✓

Bilgisayar Bilimleri müfredatı açısından değerlendirme

Çalışma kapsamında Türkiye’de bilgisayar ve bağlı bilim alanlarını ilgilendiren orta ve yüksek öğretim müfredatı ile uygulama örnekleri incelenmiş, simülâtör kullanımının yaygın olmadığı veya kullanılan simülâtörlerin müfredat ile doğrudan uyumlu olmadığı görülmüştür. Ayrıca mevcut araçların pedagojik açıdan uygun bir yapıdan uzak oldukları, eğitimin hedefleriyle örtüşmedikleri görülmüştür. Tarafımızdan önerilen simülâtörler, lise seviyesi veya analitik düşünce yeteneklerinin henüz olgunlaşmadığı giriş seviyesinden mühendislik programlarının uygulandığı lisans ve lisansüstü seviyelerine kadar pedagojik ihtiyaçlara cevap verebilmektedir. Hatta daha ötesinde eğitimci ve program geliştiricilerin ihtiyaçlarını da karşılayabilecek düzeyde pedagojik bilgiler içeren araçlardır.

Bilgisayar mimarisi ve bağlı bilim alanlarını ilgilendiren eğitim müfredatı çok geniş bir yelpazeye sahip olduğundan, çalışmamız dâhilinde hakkında bilgi verdiğimiz veya tarafımızdan önerilen simülâtörler genel olarak değerlendirilmiştir. Eğitim konusu, alanı ve eğitimin amacına göre simülâtörlerin çalışma içinde detaylı olarak vermeye çalıştığımız işlev ve kapasiteleri ayrıca değerlendirilmelidir. Bu şekilde bir değerlendirmeye örnek oluşturması için mümkün olduğunca kapsamlı ve standardize edilmiş bir örneği olarak IEEE (Institute of Electrical and Electronics Engineers) bilgisayar müfredatı [25-26] üzerinde önerdiğimiz simülâtörler değerlendirilmiştir (Tablo 2).

Dünyada ve Türkiye’de takip edilen eğitim stratejileri

Teknolojinin gelişimi, internet kullanımının yaygınlaşması ve eğitim alanındaki tüm bilimsel veriler dikkate alınarak geliştirilen ticari (WebCT ve BlackBoard) veya ücretsiz (Moodle) uzaktan eğitim (e-Learning) sistemleri mevcuttur. Öğrenci ve eğitimcinin anı zaman ve aynı mekânda buluşma gerekliliğini ortadan kaldıran, “öğrenen merkezli” bu uzaktan eğitim sistemleri yukarıda sayıldığı gibi ticari veya ücretsiz programlar kullanılmaksızın mevcut ve yaygın kullanımlı forum/portal platformlarında da sağlanabilir. Özellikle bilgisayar mimarisi ve organizasyonu ile ilgili eğitim alanları söz konusu olduğunda çalışmamız içinde anlatılan simülâtörler ve diğer e-öğrenme araçlarının aktif bir şekilde kullanıma sunulduğu Çizgi Söğüt Gölgesi (<http://csg.cizgi.com.tr/portal.php>) ve Tübider Okul Bilişim (<http://www.okulbilisim.com/>) internet siteleri sözünü ettiğimiz bu uygulamalara birer örnektir.

Çalışmamız boyunca üzerinde durduğumuz yaklaşımlar değerlendirildiğinde bilgisayar ve mühendislik bilimlerinde etkin öğrenme için *öğretmen-ders organizasyonu*’nun aşağıdaki şekilde yapılmasının uygun olacağı düşünülmektedir:

- Öğretmenin web sayfası (kişisel bilgiler, dersler, bağlantılar, zaman çizelgesi)
- Ders notları organizasyonu (pdf, ppt, video, animasyon vs.)
- Laboratuvar / çözülmüş örnekler / yarı çözülmüş örnekler
- Ödevler / yarı çözülmüş ödevler
- Tartışma forumları
- Yükleme alanı
- Proje tabanlı ve iş birliğine dayalı eğitim uygulamaları

Dünyadaki birçok saygın öğretim kurumunda bilgisayar sistemlerinin alt yapısı ve mühendislik eğitimleriyle ilgili eğitim programlarında (sayısal devreler, bilgisayar organizasyonu ve mimarisi, mikro denetleyiciler ve mikroişlemciler) simülâtörlerin, animasyonların ve video derslerin yaygın bir şekilde kullanıldığını ve üstelik bunların büyük bir bölümünün internette ücretsiz olarak genel kullanıma sunulduğunu görmekteyiz. Bunlara, Berkley (<http://www-inst.eecs.berkeley.edu/~cs61c/fa07/Calendar.php>), Harvard (<http://www.eecs.harvard.edu/cs141/lectures.html>) ve MIT (<http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/index.htm>) gibi saygın öğretim kurumlarından örnekler vermek mümkündür. Ülkemizde uzaktan eğitim programı olan birkaç eğitim kurumumuz ve bazı kişisel gayretler dışında bu tür kaynakların etkin bir şekilde kullanıldığı örnekleri vermek pek mümkün görünmemektedir. Yaptığımız araştırmalar neticesinde ulaşılabildiğimiz kadarıyla TOBB Ekonomi ve Teknoloji Üniversitesi, Bilgisayar Mühendisliği öğretim üyesi Prof Dr. Ali Yazıcı (<http://aliyazici.etu.edu.tr/>), İTÜ Elektrik Elektronik Fakültesi Bilgisayar Mühendisliği Bölümü öğretim üyesi Yrd. Doç. Dr. Feza Buzluca (<http://www3.itu.edu.tr/~buzluca/>) ve YTÜ Elektronik ve Haberleşme Mühendisliği Bölümü öğretim üyesi Yrd. Doç. Dr. Tuncay Uzun’un (<http://www.yildiz.edu.tr/~uzun/>) kişisel sayfaları bu konuda örnek gösterebileceğimiz diğer çalışmalardır.

Çalışmamızda bilgisayar mimarisi ve ilgili eğitim alanlarına yönelik eğitsel araç ve uygulamalar pedagojik ihtiyaçlardan müfredat içindeki uygulama alanlarına kadar kapsamlı bir şekilde anlatılmıştır. Önerilen araçların diğer araç ve servislerle birlikte verilen örneklere benzer şekilde uygulanmasıyla dünyadaki örneklere yakın platformların oluşturulması mümkün olabilecektir.

Sonuç

Bilgisayar mimarisi ve sayısal devreler eğitimi gibi soyutluluk derecesinin giderek arttığı, anlaşılabilirliğin azaldığı, hızlı gelişen teknolojilerle birlikte bunlara uygun eğitsel araçların bulunma güçlüğü, mevcut eğitsel araçların çok kısa sürelerde geçerliliğini yitirdiği bir alanda verilmesi gereken eğitim, ders araçları ve pedagojik yaklaşımlar açısından değerlendirilmiştir. Öğrencilerin sayısal devre tasarımın temellerini anlaması, analiz ve problem çözme kapasitelerini ve tasarım becerilerini geliştirmesi nedeniyle bu alanda simülasyonların vazgeçilmez bir eğitim aracı olduğu vurgulanmıştır. İnternet ortamında yaygın olarak bulunan ve bu eğitim alanında kullanılacak simülasyonlar kapsamlı olarak incelenmiş ve bunların işleyiş, avantaj ve dezavantajları konusunda bilgiler verilmiştir.

Simülasyonların teknik özellikleri tasarım kriterleri beraberinde karşılaştırılarak öğretmen ve öğrenci ihtiyaçlarına göre değerlendirilmiştir. Simülasyonların grafik özellikleri, devre inceleme ve biçimlendirme metotları, kullanım kolaylığı, kapsamlı hata kontrolü ve simülasyonu görüntüleme özellikleri gibi değerlendirme kriterleri ele alındığında tarafımızdan önerilen açık kaynak kodlu ve ücretsiz araçların (DigSim, Logisim, Deeds ve Hades) ticari bir ürün olarak tasarlanmış diğer araçlara göre pedagojik olarak daha etkin olduğu sonucuna varılmıştır.

Dünyada ve Türkiye’de bilgisayar mimarisi ve sayısal sistemler üzerine tasarlanan eğitim platformları, kaynak kullanımı, uygulanan ve geliştirilen yöntemler karşılaştırıldığında ülkemizde bu alandaki mevcut kaynakların yeterince etkin kullanılmadığı sonucuna varılmıştır. İncelenen genel erişimli kaynaklar etkin öğrenme yöntemleri dikkate alınarak ayrı bir çalışma ile ülkemizdeki eğitmen ve öğrencilerin kullanımına ve değerlendirmesine sunulmuştur.

Çalışma boyunca sıraladığımız değerlendirme kriterlerinin hepsini, farklı eğitim seviyelerindeki öğrenciler için tek bir simülasyon ile sağlanamayacağı görüşündeyiz. Bu sebeple kapsamlı bir müfredat için basitten karmaşığa doğru bir dizi simülasyon önerilmiş ve bunların öğrenci ihtiyaçları ve öğretmen isteklerine göre avantaj ve dezavantajları irdelenmiştir.

Teknik ve mühendislik eğitimlerinde laboratuvar dersleri için günümüzde kullanılan araçlar ve ekipmanlar daha çok komple deney setleri olup oldukça pahalıdır. Bazı üniversitelerde ise laboratuvar dersleri yalnızca belirli konular için verilmekte (genellikle laboratuvar uygulamaları daha kolay olan kısım), teorik içeriğin tamamını kapsamamakta ve sonuç olarak eğitimin hedeflerine ulaşamamaktadır. Tarafımızdan önerilen simülasyonların kullanımıyla bu sorunların önüne geçileceği düşünülmektedir. Ancak belirtilmesi gereken önemli bir husus daha vardır; simülasyonların bazıları (özellikle çok didaktik olanlar), gerçek dünyadan oldukça uzak olduğundan dolayı bunların gelişigüzel ve çok fazla kullanılması risklidir. Ayrıca bilgisayar organizasyonu konusunun içeriği bir takım teknik bilgiler olmaksızın pratik olarak uygulamaya geçirilmesi oldukça zordur. Bu nedenle özellikle eğitimcilerin ders içeriği, seviyesi ve hedefleri doğrultusunda uygun araç ve servisleri dikkatli bir şekilde seçmesinin ve etkin öğrenme hedeflerine uygun yaklaşımlarla ele almaları zorunludur.

Konuyla ilgili projelerimizin ve bu çalışmanın ortaya çıkmasında destek ve teşvikleriyle bizleri motive eden Çizgi Söğüt Gölgesi ve Tübider Okul Bilişim platformlarında eş zamanlı yürütülen e-Learning projesi üyelerine, ayrıca görüş ve önerilerinden yararlandığımız değerli akademisyenlerimize teşekkür ederiz.

Kaynaklar

- [1] B. Gültekin Çetiner, Nedim Türkmen ve Oğuz Borat, “Yüksek Öğretimde Paradigma Değişimi”, http://www.drcetiner.org/yuksek_ogretimde_paradigma_degisimi_paradigma_kaymasi.htm
- [2] Wolffe, G.S., Yurcik, W., Osborne, H., Holliday, M.A., 2002. “Teaching Computer Organization/Architecture with limited Resources using Simulators”, Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education, ACM Press, pp. 176-180

- [3] Poplawski, D.A., 2007. "Simulation and animation: A pedagogically targeted logic design and simulation tool", June 2007 Proceedings of the 2007 workshop on Computer architecture education WCAE '07, ACM Pres, pp.1-7
- [4] Donzellini, G., Ponta, D., 2007. "A Simulation Environment for e-Learning in Digital Design", Trans. on Industrial Electronics, vol. 54, no. 6
- [5] Kim, B. (2001). Social constructivism. In M. Orey (Ed.), Emerging perspectives on learning, teaching, and technology. E-book available at <http://itstudio.coe.uga.edu/ebook/>
- [6] NTL Institute for Applied Behavioral Science, 300 N. Lee Street, Suite 300, Alexandria, VA 22314. 1-800-777-5227
- [7] Shatilla, Y., 2004. "University-Industry Relations: A Step Closer," 2004 Conference for Industry and Education Collaboration (2004 CIEC), CIP-323, Mississippi, USA
- [8] Filiz Eyüboğlu, 2004. "e-Öğrenme Nedir-5; Karma / Harmanlanmış Öğrenim (Blended Learning)", TBD Dergi, Sayı:148 08.03.2004, <http://dergi.tbd.org.tr/>
- [9] Ponta, D., Donzellini, G., Markkanen, H., 2003. "Project Based Learning In Internet", 33rd ASEE/IEEE Frontiers in Education Conference, November 5-8, S2F-7, Boulder-Colorado, USA
- [10] Ponta, D., Donzellini, G., 2004. "Active Learning With Network Based Projects", Fourth international workshop on Active Learning in Engineering Education, June 6-9 2004, p.45-46, Nantes, France
- [11] Coutinho, L.M.N., Mendes, J.L.D., Martins, C.A.P.S., 2006. "MSCSim –Multilevel and Split Cache Simulator", 36th ASEE/IEEE Frontiers in Education Conference, October 28 – 31, 2006, p. T1F-7, San Diego, USA
- [12] Freitas, H.C., Martins, C.A.P.S., 2003. "Didactic Architectures and Simulator for Network Processor Learning", 30th International Symposium on Computer Architecture, June 8, 2003, San Diego, USA
- [13] Bečvář, M., Kahánek, S., 2007. "VLIW-DLX Simulator for Educational Purposes", 34th International Symposium on Computer Architecture, June 9, 2007, San Diego, USA
- [14] Shneiderman, B., 2002. "Designing the User Interface - Strategies for Effitive Human-Computer Interaction", second edition, Reading, MA: Addison-Wesley Publishing Company
- [15] Halstead, S., "Circuit Design Tools for a Breadboard Simulator", University of York, http://www.stephen-halstead.co.uk/breadboard_simulator/index.php
- [16] Douglas, A., 2007. "Animation of a Digital Circuit Simulator", University of Glasgow, <http://www.dcs.gla.ac.uk/~daw/masters-projects/dissertations/Douglas.2007.pdf>
- [17] Glass, N., "Java Digital Breadboard Simulator", University in Prague, <http://www.cs.york.ac.uk/netpro/bboard/>
- [18] Burch C., 2002. "Logisim: A graphical system for logic circuit design and simulation", Journal of Educational Resources in Computing, Vol. 2 (March 2002), p. 5-16
- [19] Burch C., 2004. "Science of Computing Suite (SOCS): Resources for a BreadthFirst Introduction", SIGCSE'04, March 3–7, 2004, Norfolk, Virginia, USA.
- [20] Donzellini, G., Ponta, D., 2004. "NetPro for Information Engineering", DIBE - University of Genoa, Italy, http://netpro.evtek.fi/team/wp6/presentations/Learntec_2004_Ponta.ppt
- [21] Donzellini, G., Ponta, D., 2004. "A Learning Enviroment For digital electronics", DIBE - University of Genoa, Italy, http://epsc.upc.edu/projectes/ed/programari/deeds/10869_DEEDS_TAEE2004.pdf
- [22] Donzellini, G., Ponta, D., 2003. "DEEDS: E-Learning Environment for Digital Design", EUNITE - European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems, 10-12 July 2003, Oulu, Finland
- [23] Hendrich, N., 2002. "Automatic Checking of Students' Designs Using Built-In Selftest Methods", University of Hamburg, <http://tams-www.informatik.uni-hamburg.de/publications/2002/ewme2002/bist.pdf>
- [24] Poplawski, D.A., 2007. "A Pedagogically Targeted Logic Design and Simulation Tool", Proceedings of the 2007 workshop on Computer architecture education, p. 1-7, San Diego, USA
- [25] Computing Curricula 2001 Computer Science Final Report (December 15, 2001), http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/cc2001.pdf
- [26] Computing Curricula 2005 – The Overview Report (September 30, 2005), http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf



M. Niyazi Saral

1957 doğumlu M. Niyazi Saral, 1981 yılında İTÜ'den Elektronik Yüksek Mühendisi olarak mezun oldu. Aynı üniversitede 2 yıl öğretim görevlisi olarak çalışan Niyazi Saral daha sonra sektörde mühendis ve yönetici olarak çalıştı. Evli ve iki çocuk babası Niyazi Saral kendi mesleği dışında e-öğrenme, etik kuramlar, tarih ve ekonomi üzerine hobi çalışmaları yapmaktadır. Niyazi Saral Tübider (Türkiye Bilişim Sektörü Derneği) Okul Bilişim projesinde gönüllü olarak çalışmaktadır.



Yıldıray Topcu

1973 Bafra doğumlu. 1995 yılında OMÜ, Fen-Ed. Fak. Kimya Bölümünden mezun oldu. Aynı bölümde 1998 yılında Yüksek Lisans ve 2002 yılında Doktorasını tamamladı. Kariyerine öğretim görevlisi olarak devam eden Yıldıray Topcu, evli ve iki çocuk babasıdır. Mesleği dışında Adli Bilimler ve Kriminoloji alanında hobi çalışmaları yapmakta ve Çizgi Söğüt Gölgesi internet topluluğu tarafından yürütülen sosyal projelere gönüllü olarak katkıda bulunmaktadır.