

-152116025- TASARIM SÜREÇLERİ

Ders 8: Tasarım Evresi-3

Dr. Yıldıray Anagün,

yanagun@ogu.edu.tr

Eskişehir Osmangazi University

Computer Engineering Department

Tasarım Evresinde Neler Yapılır?

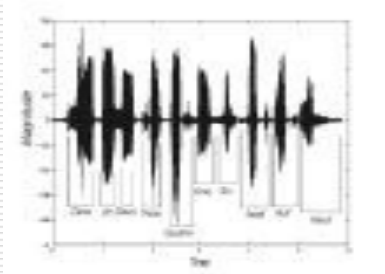
- ☐ Kullanıcı-Sistem ara yüzü tasarımı
 - ☐ Veri tabanı tasarımı
 - ☐ Gömülü Donanım tasarımı
 - ☐ Yazılım tasarımı
 - ☐ Sistem kontrolleri/test tasarımı.
 - ☐ Ağ tasarımı ve güvenliği
-

Gömülü Sistem Tasarımı

Gömülü Sistem genelde iki kısımdan oluşur

- ❑ Elektronik Kısım
- ❑ Gömülü Yazılım (veya firmware)

Biyomedikal Görüntüleme



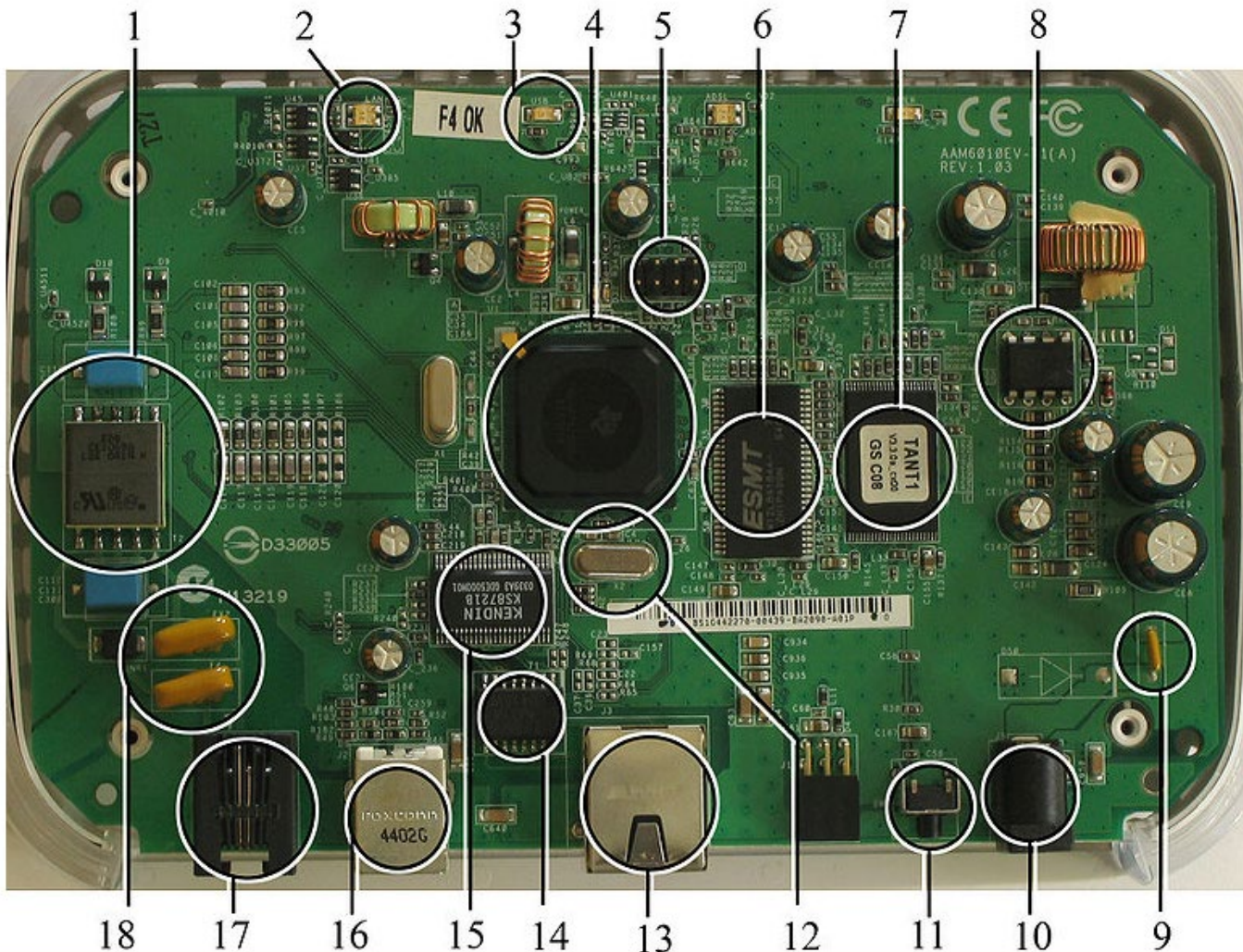
Fotoğraf Makineleri



Cep Telefonları



Netgear DG632 ADSL Modem/Yöneltili



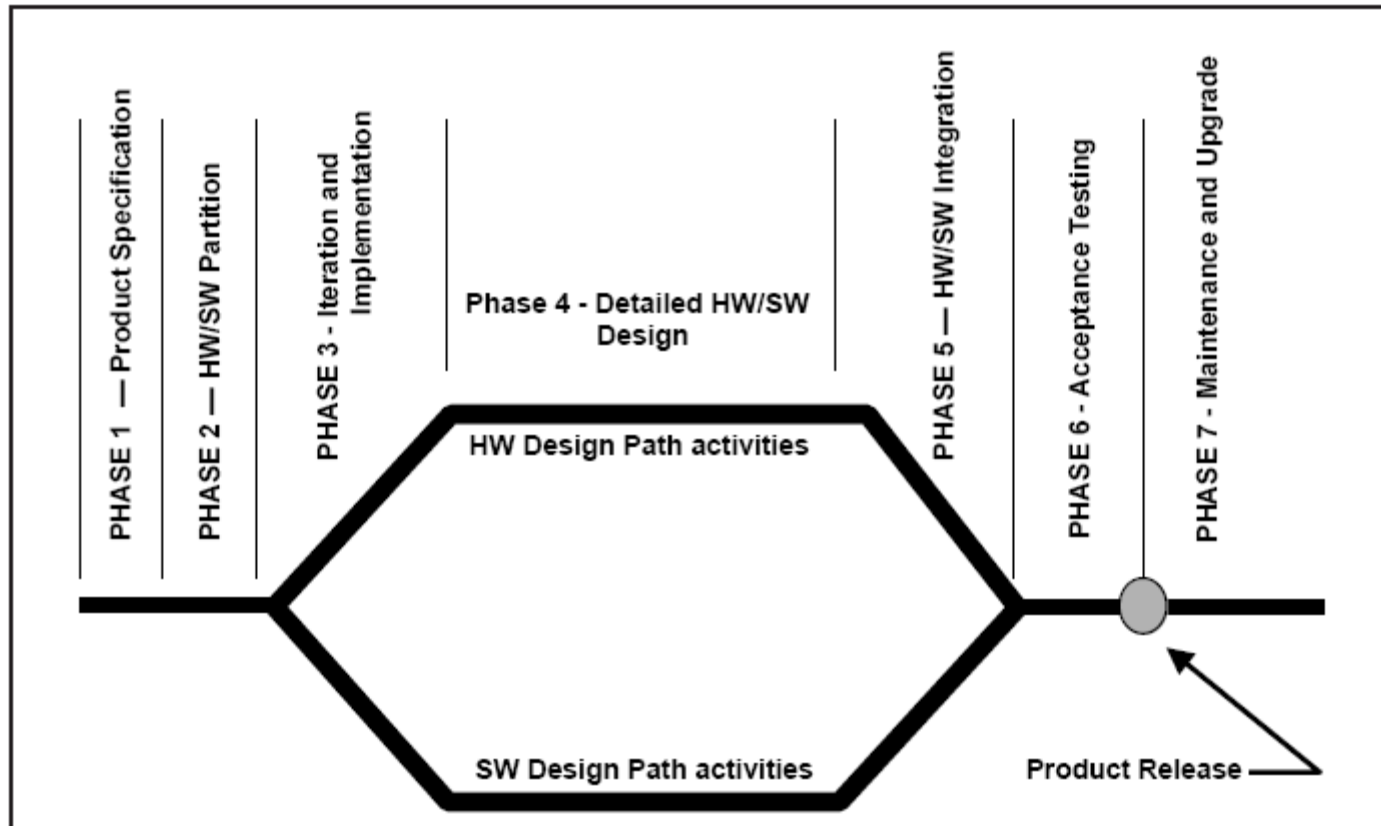
- (4) Mikroişlemci
- (6) RAM
- (7) Flash Bellek
- (8) Regülatör
- (12) Kristal osilatör
- (13) Ethernet girişi
- (15) Ethernet PHY alıcı-verici
- (16) USB

Gömülü Sistem Tasarımı

Tasarım Adımları

- ☐ Ürün Özelliklerini belirleme (ANALİZ!!)
 - ☐ Donanım/Yazılım Ayrıştırması Yapma
 - ☐ Son Kontrol
 - ☐ Bağımsız donanım ve yazılım tasarımı
 - ☐ Donanım/yazılım entegrasyonu
 - ☐ Testler
 - ☐ Bakım ve Güncelleme
-

Gömülü Sistem Tasarım Adımları

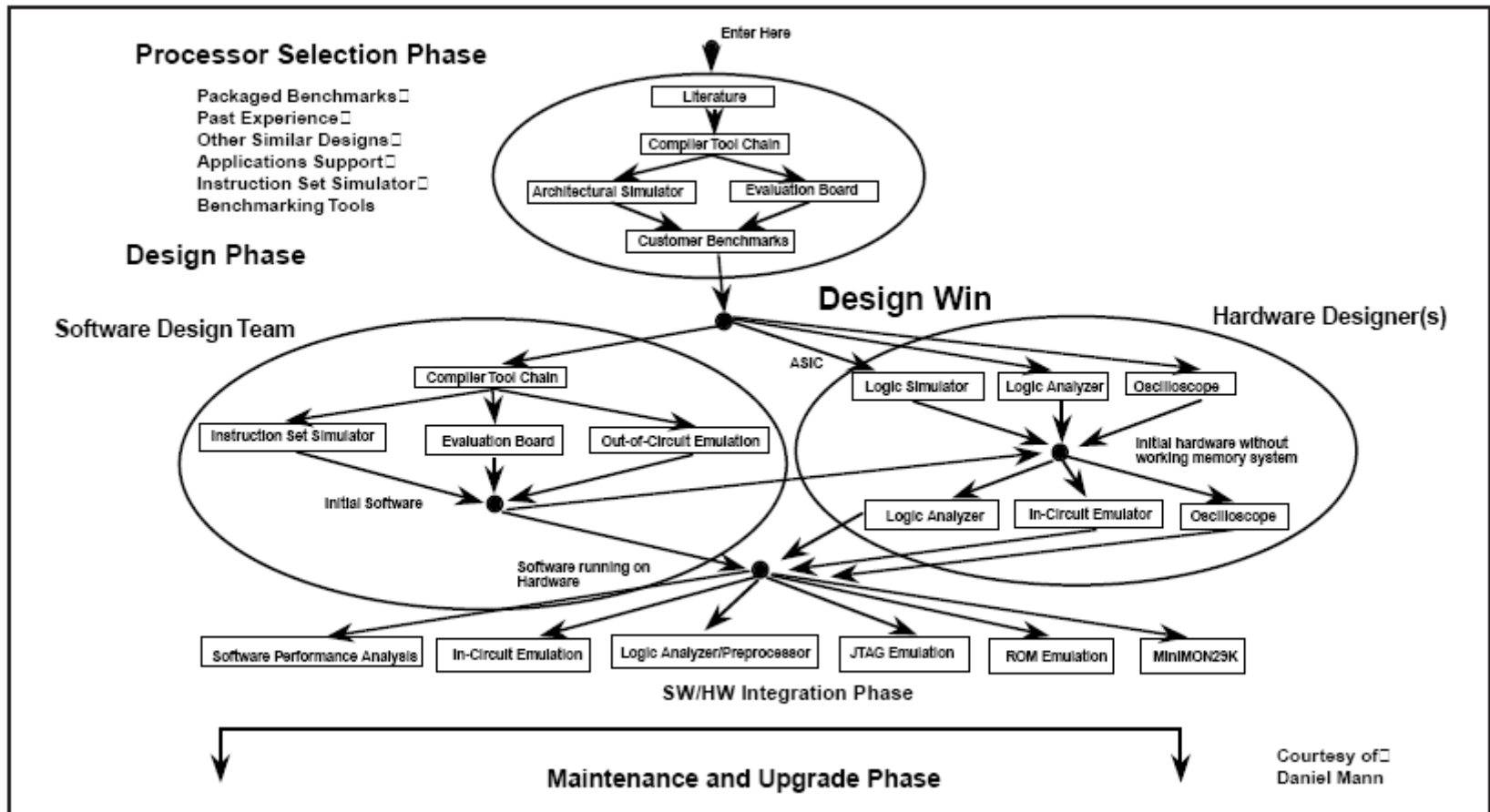


Alternatif Tasarım: Araç-tabanlı geliştirme

□ Araçlar

- Mevcut işlemciler
 - Mevcut Yazılım geliştirme araçları
 - Mevcut Donanım geliştirme araçları
 - Mevcut test araçları
 - Önceki projelerdeki oluşan donanım/yazılım tasarımları
 - Tasarım takımının ihtiyacı olan araçlar hazır mı?
-

Araç-Tabanlı Bakış



Ürün Özelliklerini belirleme

- ❑ Analiz kapsamında izlenen yöntemler kullanılır.
 - ❑ Genel İsterler;
 - Her işi yapan bir donanım
 - Müşteri ihtiyaçlarına uygun bir donanım
 - Pazarlama takımı ile görüş paylaşımı- Markete uygunluk
 - ❑ Teknik detaylar
 - Kablosuz haberleşmeli tasarımlar
 - CAN/RS485/RS232 'li platformlar
 - Dokunmatik arayüzler
 - İşlemci tipi: PIC, DSP, FPGA
 - ADC/DAC 'lı platformlar
 - GSM/GPRS haberleşmeli platformlar
 - Ultra düşük güç tüketimli mobil cihazlar
-

Takım Özelliklerini Belirleme

- ☐ Çevik Yazılım Geliştirme
 - Extreme programming
 - ☐ Communication
 - ☐ Simplicity
 - ☐ Feedback
 - ☐ Courage
 - ☐ Bu önemli çünkü birbirine bağımlı donanım/yazılım özellikleri bulunmaktadır.
-

Donanım/Yazılım Ayırıştırma

☐ Ayırıştırma Kararı

- Hangi Kısım Donanımda/Hangi Kısım yazılımda?
- İhtiyaç olan algoritmalar/işlemler neler?
 - ☐ Örnek sinyal filtreleme: Donanım/Yazılım
- İşlemleri donanım veya Yazılıma ata

Donanım/Yazılım Ayrıştırma

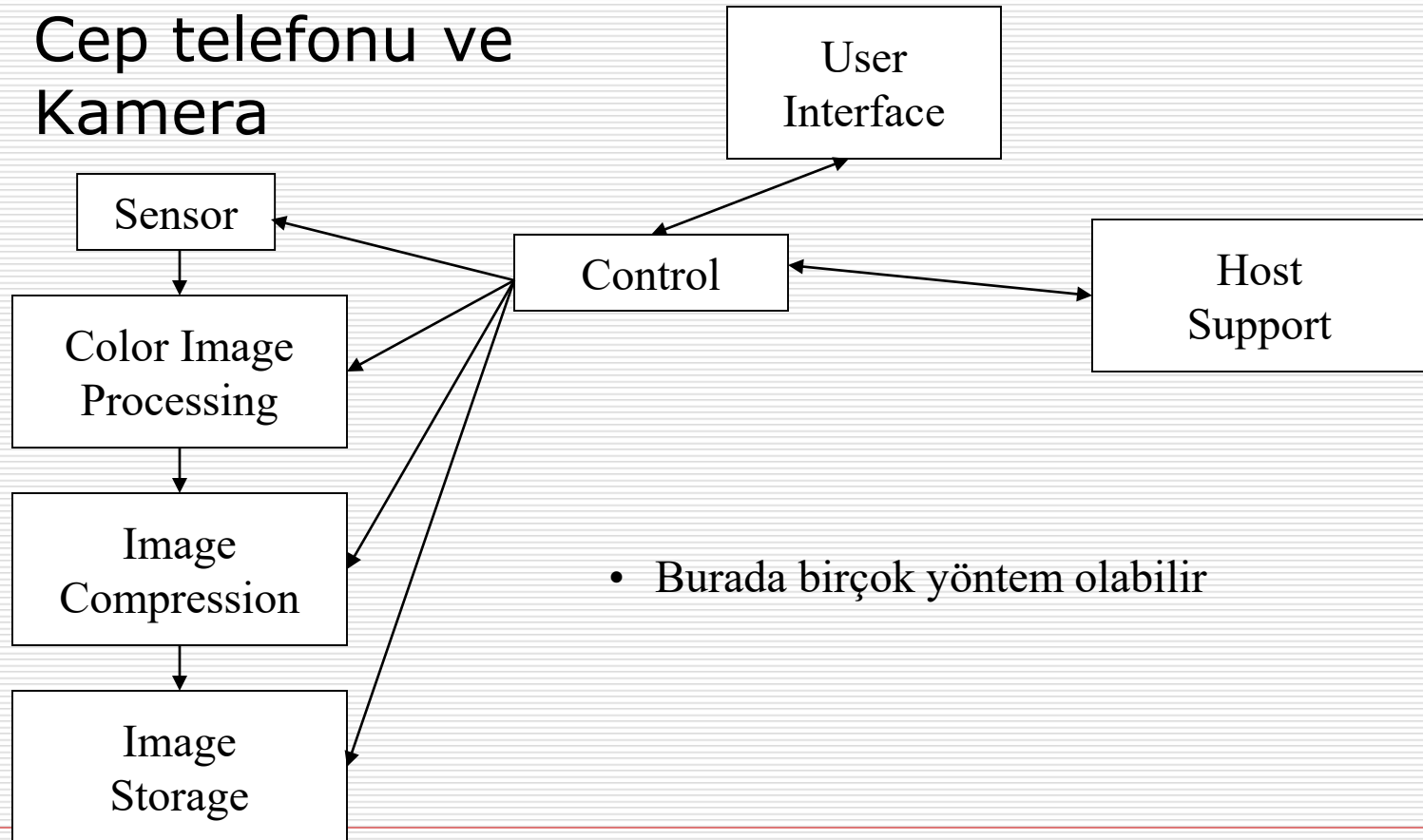
- Karar Kriterleri
 - Sistem Performans İsterleri
 - Gelecek Güncellemeler
 - Geliştirme Maliyeti
 - Son ürün maliyeti
 - Markete Çıkma Zamanı
 - Fikri/Sinai haklar koruması
 - Standard olan veya olmayan bileşenler
 - Market Öncüsü veya takipçisi olma
 - Bunlar birbirleri ile çelişebilir.
-

Donanım/Yazılım Ayrıştırma

- Tasarımcılar burada bir geliştirme süreci işletebilir.
 - Yazılım gerçeklemeleri ile başla
 - Takip eden kısımda donanıma geçiş yap(fpga)
 - Bu
 - Pazara çıkma süresi
 - Geliştirme Maliyetleri
 - Sonraki adımlarda maliyet azaltma vb. katkısı olur
 - Büyük Ölçüde tecrübeye bağlı
-

Donanım/Yazılım Ayırıştırma

□ Cep telefonu ve Kamera



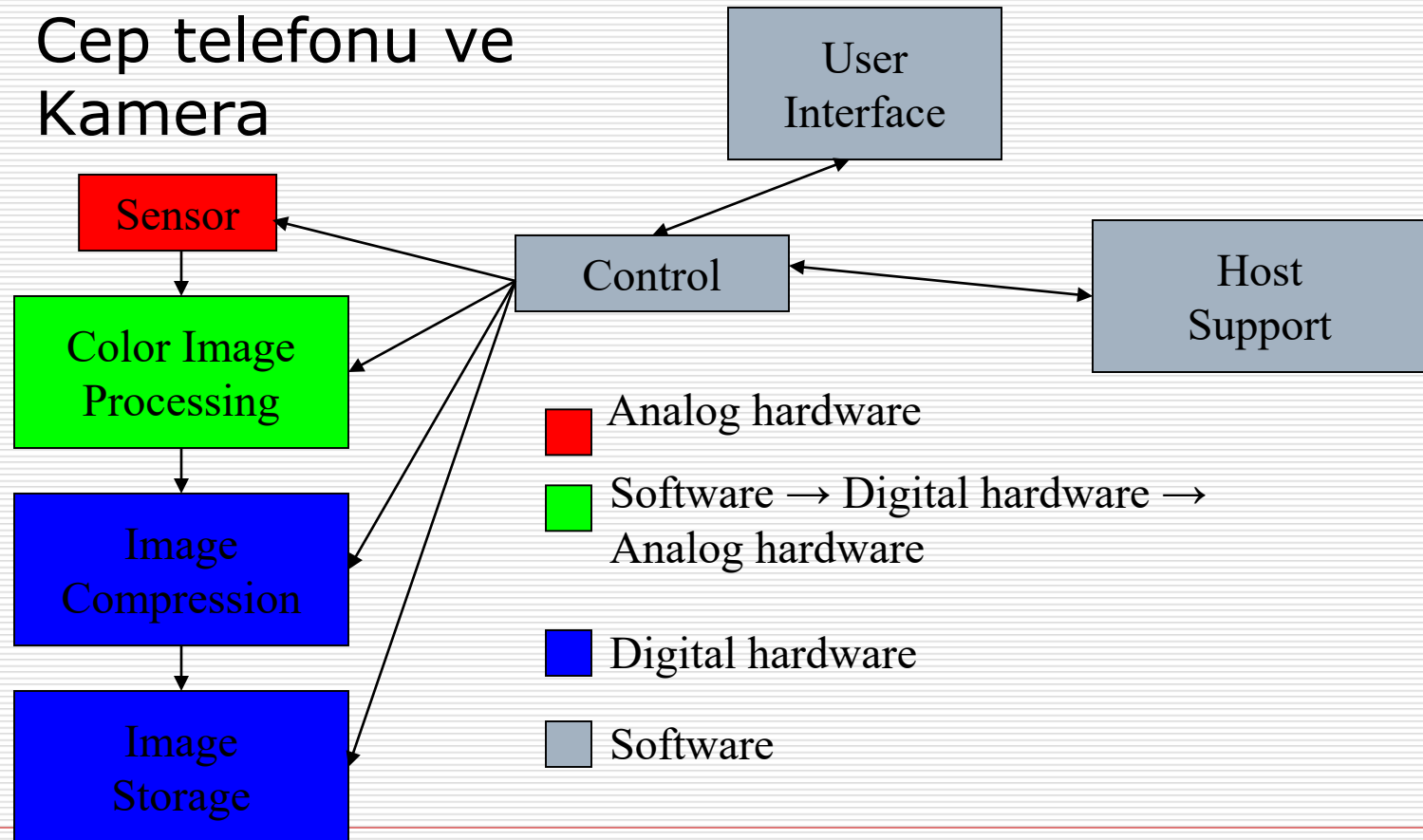
- Burada birçok yöntem olabilir

Son Kontrol

- Hangi ayrışımın yapılacağıнын test ile karar verilmesi
 - Donanım tasarımcıları benzetimlerle performans testi yapması
 - Yazılımcıların deneme kitleri kullanarak benchmark kodları geliştirmesi
 - Tasarım yapmadan beklenen sistem performansı hakkında bilgi edinmek mümkün olabilir.
-

Örnek

□ Cep telefonu ve Kamera



Bağımsız Donanım/Yazılım Tasarımı

- Eldeki işler
 - Hangi kapsamda yapılabilir?
 - Bilgisayar Mühendisi
 - Yazılım Tasarımı
 - Digital devre Tasarımı
 - Elektronik Mühendisi
 - Analog devre tasarımı
 - Digital devre Tasarımı
 - Sistem Mühendisi
 - Makine Mühendisi
 - Vb.
-

Donanım Tasarımı Bileşenleri:

Mikroişlemciler

- Robert Noyce ve Gordon Moore Intel'i kurdular.
- Intel ilk programlanabilir hesap makinesini üretti.
- Intel ilk mikroişlemciyi 1971'de tasarladı.
 - Model 4004
 - 4-bit; 2300 transistor, 640 byte hafıza, 108 KHz saat frekansı

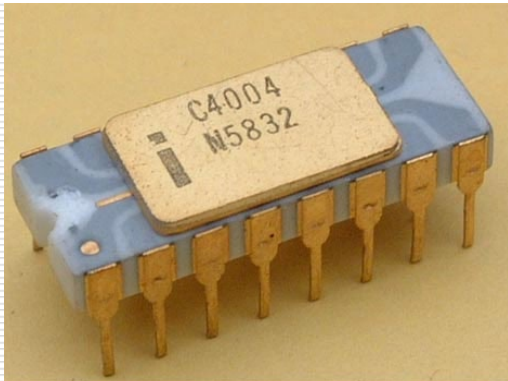
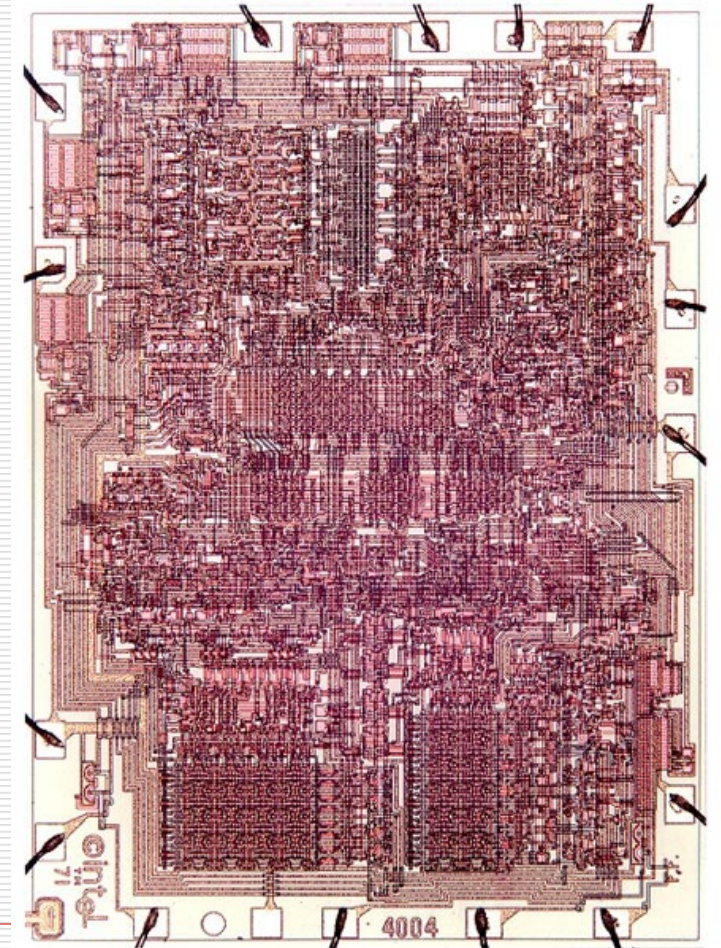


Image courtesy of CPU-Zone.com. Used with permission.



Donanım Tasarımı Bileşenleri: Mikrodenetleyiciler

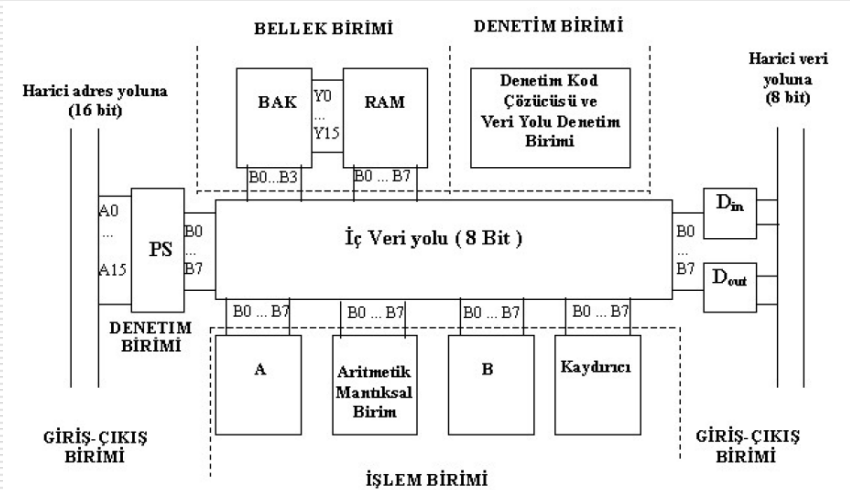
- Bir mikroişlemcinin, hafıza ve giriş - çıkışlar, kristal osilatör, zamanlayıcılar (timers), seri ve analog giriş çıkışlar, programlanabilir hafıza (Flash, ROM) gibi bileşenlerle tek bir **tümleşik devre** üzerinde üretilmiş halidir.



Infineon tarafından üretilmiş 8051 tabanlı bir mikrodenetleyici

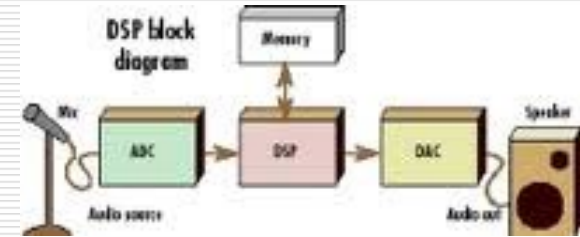
Mikroişlemcilere göre

- oldukça küçük boyutludurlar,
- çok düşük güç tüketimine sahiptirler,
- düşük maliyetlidirler,
- yüksek performansa sahiptirler.



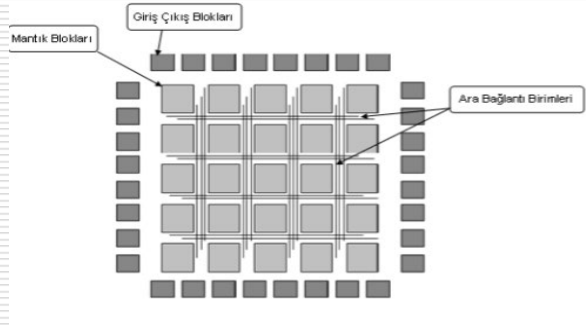
DonanımTasarımı Bileşenleri: Sayısal İşaret İşlemcisi (DSP)

- Sayısal işaret işleme konusunda yüksek hızda yüksek doğruluk gerektiren uygulamalar için optimize edilmiş mimariye sahip özel bir mikroişlemcidir.
- Ses veya görüntü sensörlerinden elde edilen işaretler analogdan sayısala çevrilerek, sayısal olarak işlenir ve daha sonra elde edilen sayısal veri, tekrardan analog işaret haline getirilir.DSP uygulaması belirli bir süre içinde tamamlanmalıdır, ertelenmiş (batch)işlem yapmak mümkün değildir.
- Özel olarak tasarlanmış DSP işlemciler büyük batarya veya özel bir soğutma sistemi gerektirmeksizin daha yüksek performanslı, düşük gecikmeli ve düşük maliyetli çözüm sağlarlar.



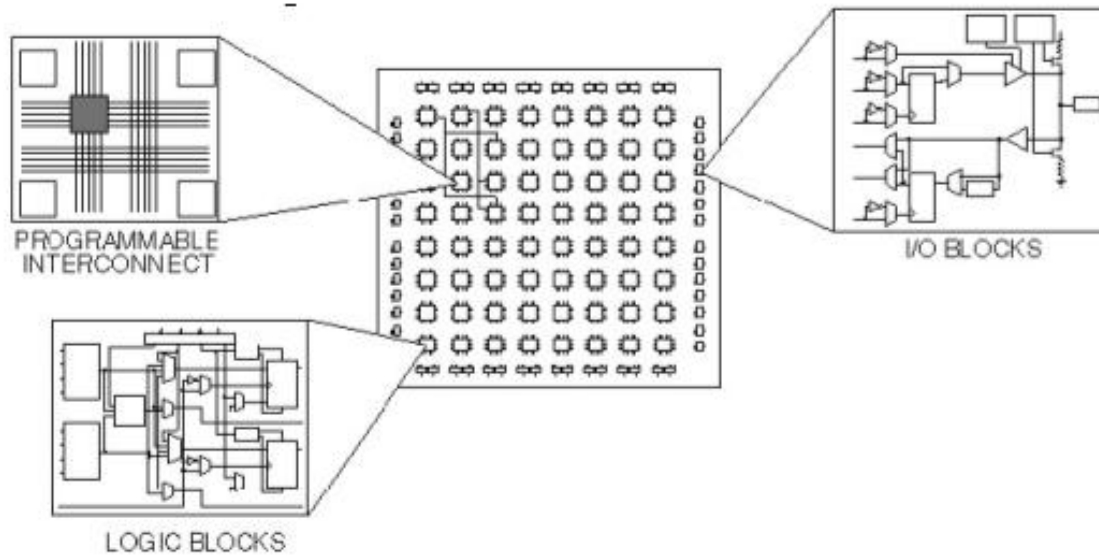
DonanımTasarımı Bileşenleri: Field Programmable Gate Array-FPGA

- ❑ Xilinx firması FPGA adını verdiği yeni bir IC sınıfı geliştirdi ve 1984 yılında pazara sunulacak hale getirdi.
- ❑ Programlanabilir mantık blokları ve bu bloklar arasındaki ara bağlantılardan oluşan ve geniş uygulama alanlarına sahip olan sayısal **tümleşik devrelerdir**
- ❑ Alanda programlanabilir ismi verilmesinin nedeni, mantık bloklarının ve ara bağlantıların imalat sürecinden sonra programlanabilmesidir.
- ❑ **Donanım Tanımlama Dilleri(Hardware Description Languages, HDL):** En yaygın olanlar; VHDL, Verilog ve Abel'dir



FPGA Mimarisi

- Programlanabilir mantık blokları, ara bağlantılar içerisine gömülü şekilde bulunur. Programlanabilir mantık bloklarının yapılandırılması ve bu bloklar arasındaki iletişim ara bağlantılar sayesinde gerçekleşir. Giriş çıkış blokları, ara bağlantılar ile bütünleşmiş devrenin paket bacakları arasındaki ilişkiyi sağlar



ELEKTRONİK DEVRE TASARIM PROBLEMİ

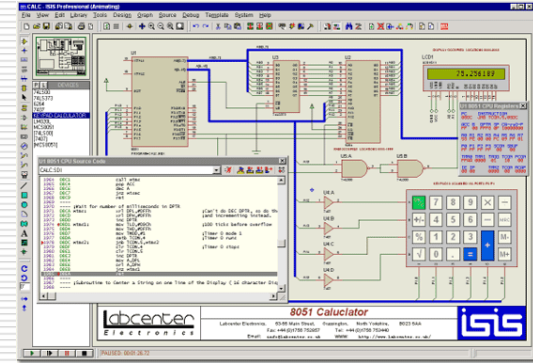
- ❑ Elektronik devre tasarımı maddi yükümlülüğü yüksek bir bilimsel çalışma sonucu ortaya çıkabilmektedir.
 - ❑ Devre tasarımının en büyük ikinci zorluğu ise tasarım için ayrılması gereken zamandır. Yüksek sayıda eleman içeren bir sistemin tasarlanması yıllar alabilmektedir.
 - ❑ Lineer olmayan sistemleri tanımlamak üzere kullanılan devre takımları da lineer olmayacağından, ağır hesap yükü gerektirmektedir.
 - ❑ Her tasarımcı tasarlayacağı devrenin hangi koşullarda çalışacağına dair bir takım kriterler belirler. Devre bu "spec" değerlerini minimum hata ile sağlamalı ve minimum alan kullanarak devre tasarlanmalıdır.
-

Elektronik Devre Tasarım Araçları

- ❑ Devre Analizi :SPICE®, Electronic Workbench®, Multisim ®
 - ❑ Sistem Modelleme : MATLAB Simulink®,
 - ❑ Sistem Optimizasyonu: PSPICE Optimizer ®, Neocircuit ®
 - ❑ FPGA Sentezleyici: Xilinx ISE ®, Altera Quartus®
 - ❑ Gömülü Kod üretimi: Matlab Simulink ®, C/C++
 - ❑ Serim Editörleri: Synopsys®, Magic®,
-

BİLGİSAYAR DESTEKLİ DİJİTAL SİSTEM TASARIMI

- Bilgisayarların yaygın bir biçimde kullanılmaya başlamasıyla birlikte, teknik alanlarda da çizim programları kullanılmaya başlamıştır.
- Bilgisayar Destekli Tasarım üzerine çizim - simülasyon ve analiz (üçünü) gerçekleştirebilen dünyada sayılı yazılımlar bulunmaktadır. **Proteus** (İsis-Ares), **Micro-Cap**, **MicroSim-Pspice**, **Orcad-Pspice**, **DesignLab**, **Electronics Workbench(EWB)**, **multiSIM**, **ULTIboard**, **ULTIcap**, **BoardMaker** v.b. gibi programları verebiliriz.



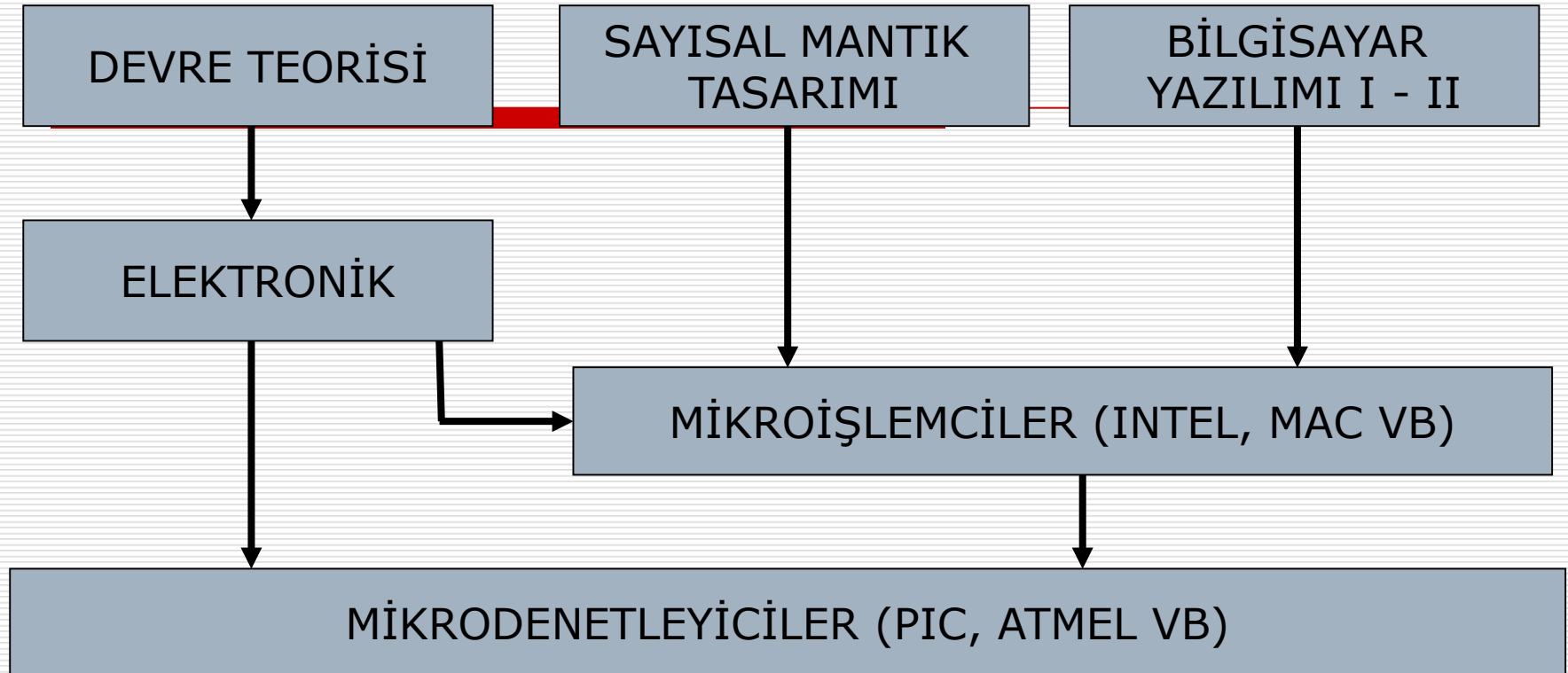
Proteus (İSİS/ARES)

Bilgisayar Destekli Tasarımın Avantajları

- Gerçek bir laboratuvar uygulamasında, elemanların temini, istenilen özellikte elemanların bulunamayışı, arızalı eleman kullanma olasılığı, uygun test ve ölçü cihazların temini, kaza, arıza esnasında cihaz ve kullanıcılara verilebilecek zararlar her zaman mevcuttur. Bilgisayar ile gerçekleştirilen sanal laboratuvar, tüm bu olumsuzlukların ortadan kaldırılmasının yanında, gerçek ortamda yapamayacağımız elektriksel hesaplama ve ölçümleri kolayca çok hassa bir şekilde sağlayabiliriz.
Maliyet : Gerçek bir elektronik laboratuvarı donatmak oldukça pahalıdır.
 - **Zaman :** Zamandan tasarruf edilir. Bir devreyi çizmek, kurmak ve test etmek, malzeme listesi çıkarmak, kağıda aktarmak, baskı devresine aktarmak çok kısa sürede gerçekleştirilebilir
-

MİKROİŞLEMCİLER/MİKRODENETLEYİCİLER

DERSİ İLE İLGİLİ DERSLER



Yazılım Tasarımı:Gerçek Zamanlı İşletim Sistemleri(RTOS)

□ Gerçek zamanlı sistemler

- Mantıksal ve İşlevsel

- Zamansal

doğruluğa sahip olması gereken sistemlerdir.

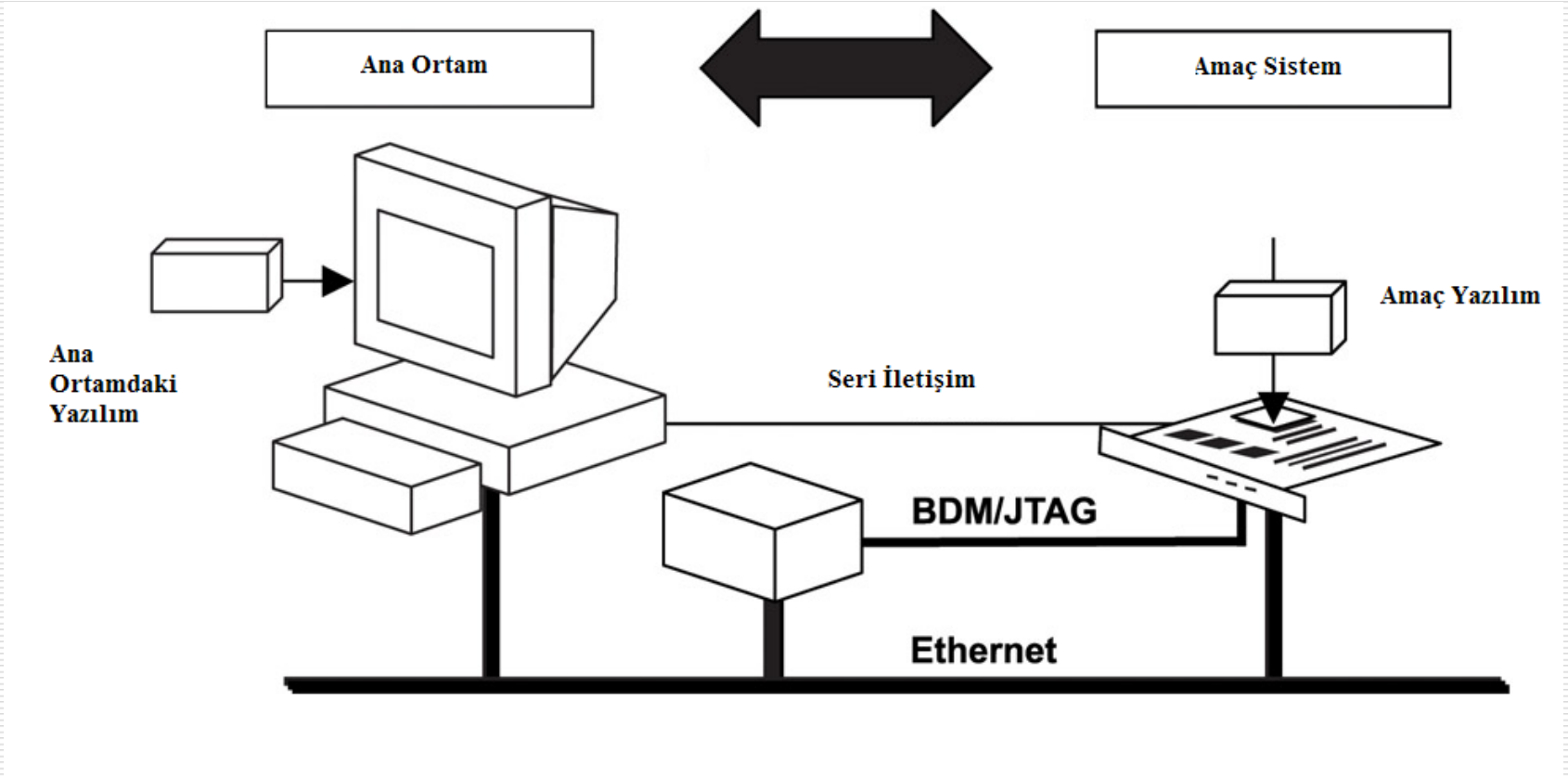
- çok keskin zaman kesin kısıtlarına sahip olan (*hard real time*)

- daha esnek zaman kısıtlarına sahip olan (*soft real time*)

Çapraz Geliştirme Ortamı

- ❑ Gömülü sistemlerde **yazılım çapraz geliştirme ortamı** yöntemi ile geliştirilmektedir.
 - ❑ Yazılım *ana ortamda* geliştirilir.
 - yazılım geliştirildiği donanım
 - işletim sistemi
 - yazılım geliştirme araçları
 - ❑ Geliştirilen yazılım, ana ortamdan *amaç ortama* taşınır.
 - ❑ Ana ortamdan amaç ortama taşınma işleminde esas önem **çapraz derleyici** sistem aracındadır.
 - ❑ Çapraz derleyici, bir işlemci mimarisi üzerinde çalışan ve değişik işlemci mimarileri için kod üretebilen sistem yazılımıdır.
-

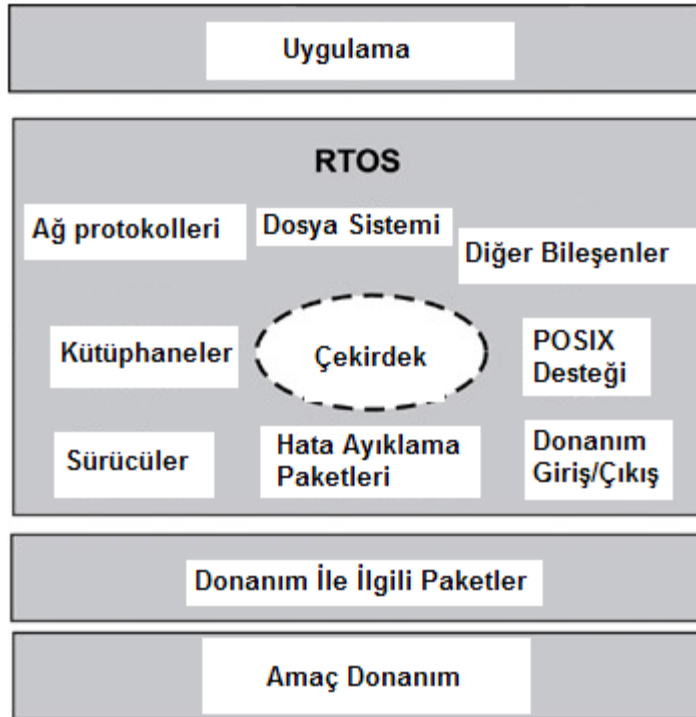
Çapraz Geliştirme Ortamı



Gerçek Zamanlı İşletim Sistemleri

- ❑ Gerçek zamanlı işletim sistemi, arkaplan/önplan gibi basit bir şekilde gerçekleştirilemeyecek gerçek zamanlı uygulamaların geliştirilebilmesi için uygun bir yazılım ortamı sağlamaktadır.
 - ❑ Tüm gömülü sistemler bir gerçek zamanlı işletim sistemi kullanılarak hazırlanmazlar.
 - ❑ Gerçek zamanlı işletim sisteminin merkezinde **çekirdek** vardır.
 - süreçlerin yönetim
 - süreçler arası iletişim
 - semafor, kilit, mesaj kutuları, kuyruklar
 - zamanlayıcı yönetimivb servisler
-

Gerçek Zamanlı İşletim Sistemleri



- ☐ Gömülü Linux
- ☐ [Windows CE](#)
- ☐ [ECos](#)
- ☐ [VxWorks](#)
- ☐ [FreeRTOS](#)
- ☐ [LynxOS](#)
- ☐ QNX

Gerçek Zamanlı İşletim Sistemleri – Genel Amaçlı İşletim Sistemleri

- ❑ Çoklu programlamaya olanak sağlarlar
 - ❑ Yazılımsal ve donanımsal kaynakları yönetirler
 - ❑ Donanımsal ortamı uygulamalardan soyutlarlar
-

Gerçek Zamanlı İşletim Sistemleri – Genel Amaçlı İşletim Sistemleri

- ❑ Gömülü uygulamalar için daha güvenilirlerdir.
 - ❑ Uygulama gereksinimlerine göre genişleyebilir ya da daralabilirler.
 - ❑ Daha iyi başarıma sahiptirler.
 - ❑ Daha az bellek gereksinimlerine sahiptirler.
 - ❑ İşlemci yönetim algoritmaları gerçek zamanlı sistemlere yöneliktir.
 - ❑ Değişik donanım ortamlarına daha kolay taşınabilme özelliklerine sahiptirler.
-

Gerçek Zamanlı İşletim Sistemlerinin Kıyaslanması

- ☐ Nasıl bir çekirdeğe sahip olduğu
 - ☐ Kesme gecikme süreleri
 - ☐ Kesme dönüş süresi
 - ☐ Zamanlayıcı yönetimi
 - ☐ Bellek gereksinimleri
 - ☐ Güvenilirlik
 - ☐ Tahmin edilebilirlik
 - ☐ Genişleyebilirlik
-

Tasarım Desenleri

- ❑ Bir yazılım mühendisliği kavramı olarak tasarım desenleri, yazılım tasarımında ortaya çıkan ortak problemlerin genel çözümleridir.
 - ❑ Tamamiyle bitmiş ve doğrudan kodlanabilen tasarımlar değildir.
 - ❑ Ortaya koymuş oldukları genel çözüm yolları uygulamaya göre şekillendirilerek ve değiştirilerek kullanılırlar.
 - ❑ Uygulamaya yönelik gösterimleri ve ilişkileri barındırmazlar, problemin genel çözüm yoluna ilişkin tasarımsal gösterimleri içerirler.
 - ❑ Algoritmik ifadeler de tasarım desenleri içerisinde yer almazlar.
 - ❑ Tasarım desenleri uygulama geliştirme süresini kısaltırlar.
 - ❑ Tasarımcılar için ortak bir dil oluşturmaktadırlar.
 - ❑ Kullanılmış ve başarıya ulaşmış tasarım ve mimarilerin yeniden kullanılabilmesini sağlarlar.
 - ❑ Yeni gereksinimleri önceden sezerek sistemi genişleyebilir olarak tasarlamak için kullanılırlar.
-

Tasarım Desenlerinin Sınıflandırılması

- ❑ *Mimarisel Tasarım Desenleri* : Sistemin alt sistemlere nasıl ayrıştırılması gerektiğini ve alt sistemler arasındaki ilişkilerin nasıl olması gerektiğini belirtirler.
 - ❑ *Yaratımsal Tasarım Desenleri* : Sistemdeki nesnelerin somut sınıfları belirtilerek nasıl yaratılması gerektiğini belirtirler.
 - ❑ *Yapısal Tasarım Desenleri*: Nesneler ve sınıflar arasındaki kalıtım ve içermeye ilişkilerini belirtirler
 - ❑ *Davranışsal Tasarım Desenleri* : Nesnelerin sorumluluklarını aralarındaki etkileşimleri belirtirler.
-

Desenlerin Çözdüğü Temel Problemler

- ❑ Nesneleri sınıfları açıkça belirterek yaratmak
 - ❑ Belirli operasyonlara bağımlılık
 - ❑ Yazılım ve donanım ortamlarına bağımlılık
 - ❑ Nesnelerin gösterimine ve gerçekleştirimlerine bağımlılık
 - ❑ Algoritmik bağımlılıklar
 - ❑ Nesnelerin birbirlerine bağımlılığı
 - ❑ Kalıtım mekanizmasının getirdiği yan etkiler
 - ❑ Nesnelerin değişiminin zor olması
-

Modern Gömülü İşletim Sistemleri

- ❑ Günümüz gerçek zamanlı sistemleri büyük ve karmaşık yazılımlardır.
 - ❑ Birbirinden farklı kısıtlamalara ve isteklere sahiptirler.
 - Yeni mimariler
 - Değişen market koşulları
 - ❑ Artık gömülü sistemler birleştirici dili ile kolayca yazılabilen ve tasarlanabilen sistemler değildir.
-

Modern Gömülü İşletim Sistemleri

- ❑ İyi bir yazılım mimarisi
 - ❑ Sistem bileşenlerinin yeniden kullanılabilir olması
 - ❑ Dağıtıklık (Alt sistemlerin dağıtılabılır olarak tasarlanması)
 - ❑ Ortamdan bağımsız olma ve değişik sistem mimarilerine ve donanımsal ortamlara taşınabilme
 - ❑ Genişleyebilir ve isteğe göre değiştirilebilir alt sistemlere sahip olma
 - ❑ Kaynakların verimli kullanılması
 - ❑ Uygulama ihtiyaçlarına göre değişebilme
-