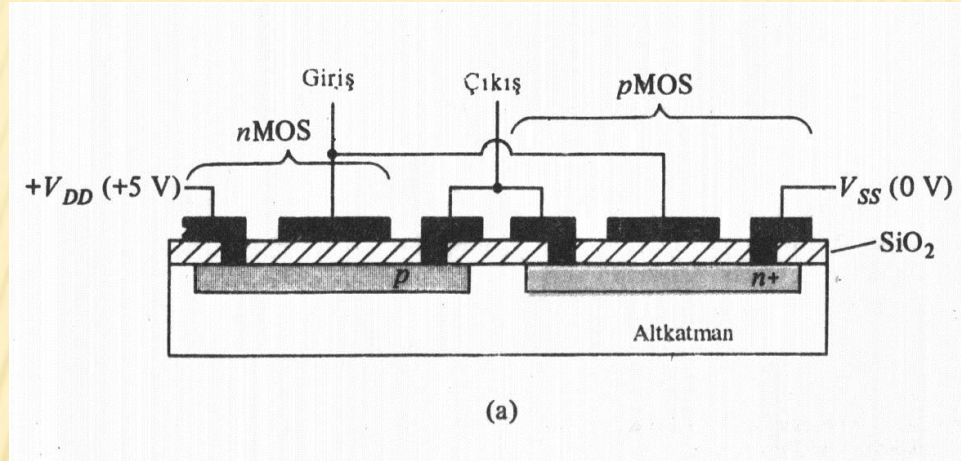
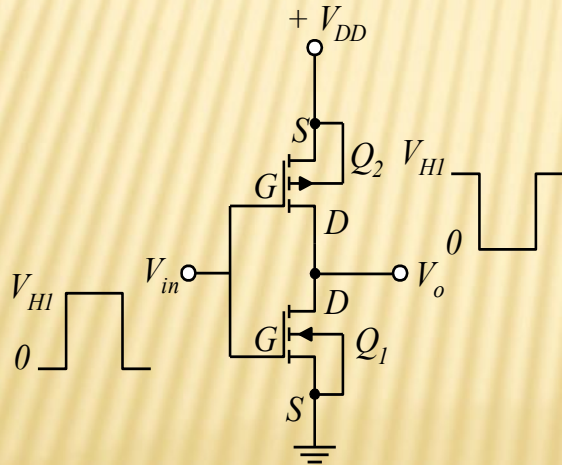


ELM320- ENTEGRE DEVRE TASARIMI

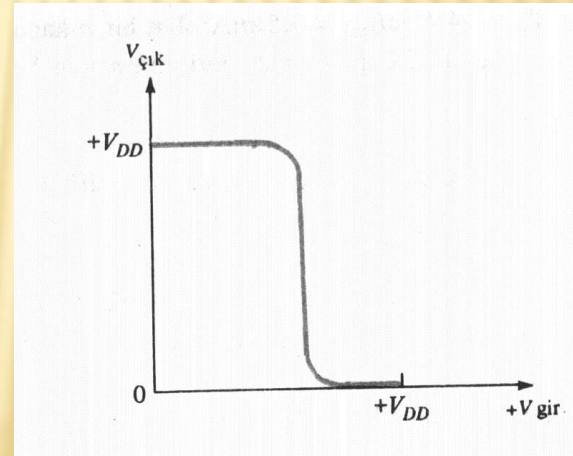
METODOLOJİLER – GİRİŞ



CMOS Yapısı



CMOS Tersleyici



Transfer Karakteristiği

Neden CMOS?

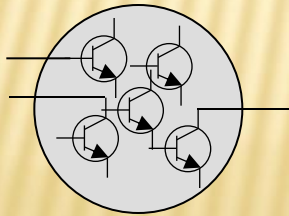
- Gerilimin $+V_{DD}$ den 0V'a düştüğü veya 0V'tan $+V_{DD}$ ye yükseldiği kısa bir süre dışında, PMOS ve NMOS transistörlerinin seri bağlantısındaki transistörlerinden biri kapalıdır ve besleme kaynağından **akım çekilmez**.
- Yüksek ve alçak çıkış düzeyleri arasında kalan kısa anahtarlama süresi dışında (yani biri açık diğeri kapalı duruma geçerken her iki transistörün açık olduğu süre dışında) CMOS devresi besleme kaynağından güç çekmeden yüksek veya alçak çıkış ile çalışmaktadır.
- Aslında bir CMOS devresinin güç tüketimi dc koşullarında **sıfır** olup, uygulanan sinyalin frekansı ile birlikte devre daha sık anahtarlama yaptığından bu tüketim artacaktır.
- Düşük güçlü entegre devrelerin çoğunluğu CMOS transistörleri ile yapılmıştır.

METODOLOJİLER - GİRİŞ

YARIİLETKENLER

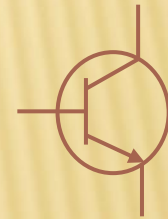
Tümleşik Devreler (IC)

- Birden fazla transistör
- Tek bir yarıiletken üzerinde
- Karmaşık devreler
- Pentium 10 Milyon Transistör üstü



Ayrık Elemanlar (Discrete)

- Diyot
- Transistör
- Tristör
- FET
- UJT
- Güç elektronığı ve OptoElektronik

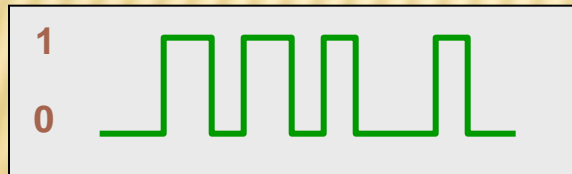


METODOLOJİLER - GİRİŞ

TÜMDEVRELER (Integrated Circuits)

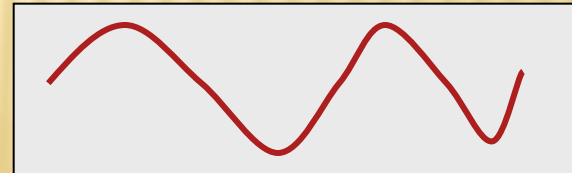
Sayısal Tümdevreler

- Yalnız 2 seviye vardır 0 veya 1
- Kolay bilgi işleme ve depolama
- Gürültüye dayanıklı



Analog Tümdevreler

- Geniş bir gerilim bölgesinde çalışırlar
- Kontrolü sayısal devrelere göre zordur
- Gürültü üretir ve kapabilirler



METODOLOJİLER - GİRİŞ

Sayısal Tümdevreler (Digital ICs)

Mantık Tümdevreleri

- Sayısal mantık işlemlerini gerçekler
- Paralel çalışan devreler

Bellekler

- Sayısal bilgiyi depolayan tümdevreler
- DRAM
- SRAM
- FLASH
- EPROM

Mikro İşlemciler

- Program akış mantığına göre çalışırlar
- uP , uC , DSP

METODOLOJİLER - GİRİŞ

Sayısal Mantık Tümdevreleri (Digital Logic ICs)

Uygulamaya Özgü Tümdevreler (ASIC)

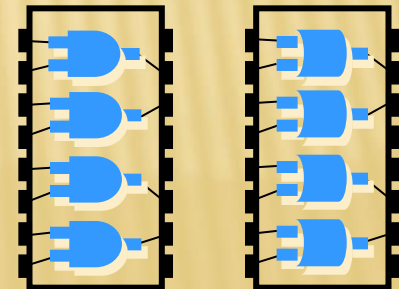
- Sadece tek bir ürün için özel tasarım
- Fabrikadan yarı-mamül veya tam mamül olarak çıkar

Uygulamaya Özgü Standart Ürünler (ASSP)

- Kaydedicileri programla
- DVB tümdevreleri
- MP3 , MPEG tümdevreleri

Standart Ürünler (Standard Products)

- Fonksiyonları tanımlı
- Katalog ürünleri
- 74xxx , CD4000
- TTL , CMOS



METODOLOJİLER - GİRİŞ

Uygulamaya Özgü Tümdevreler (Application Specific Integrated Circuits)



Programlanabilir Mantık Ürünler (PLDs)

- Yarı işlenmiş ürün
- Tasarımcının / Üreticinin prog.
- Kolay prototip geliştirme
- Ön maliyet (NRE) gerekmez

Özel Tümdevreler (Custom ICs)

- Tam bitmiş ürün (tak çalıştır)
- Tasarımcının ön çalışması
- Bir defaya özgü bedel (NRE)
- Uzun üretim süreci
- Kapı Dizileri (Gate Arrays)
- Standard Cells
- Tam Özel (Full Custom)

METODOLOJİLER - GİRİŞ

Programlanabilir Mantık Ürünler (Programmable Logic Devices)

Basit PLDler

- 70 lerde ROM lojik
- Çarpımların Toplamı
- Programlanabilir bağlantı noktası
- PROM
- PLA
- PAL

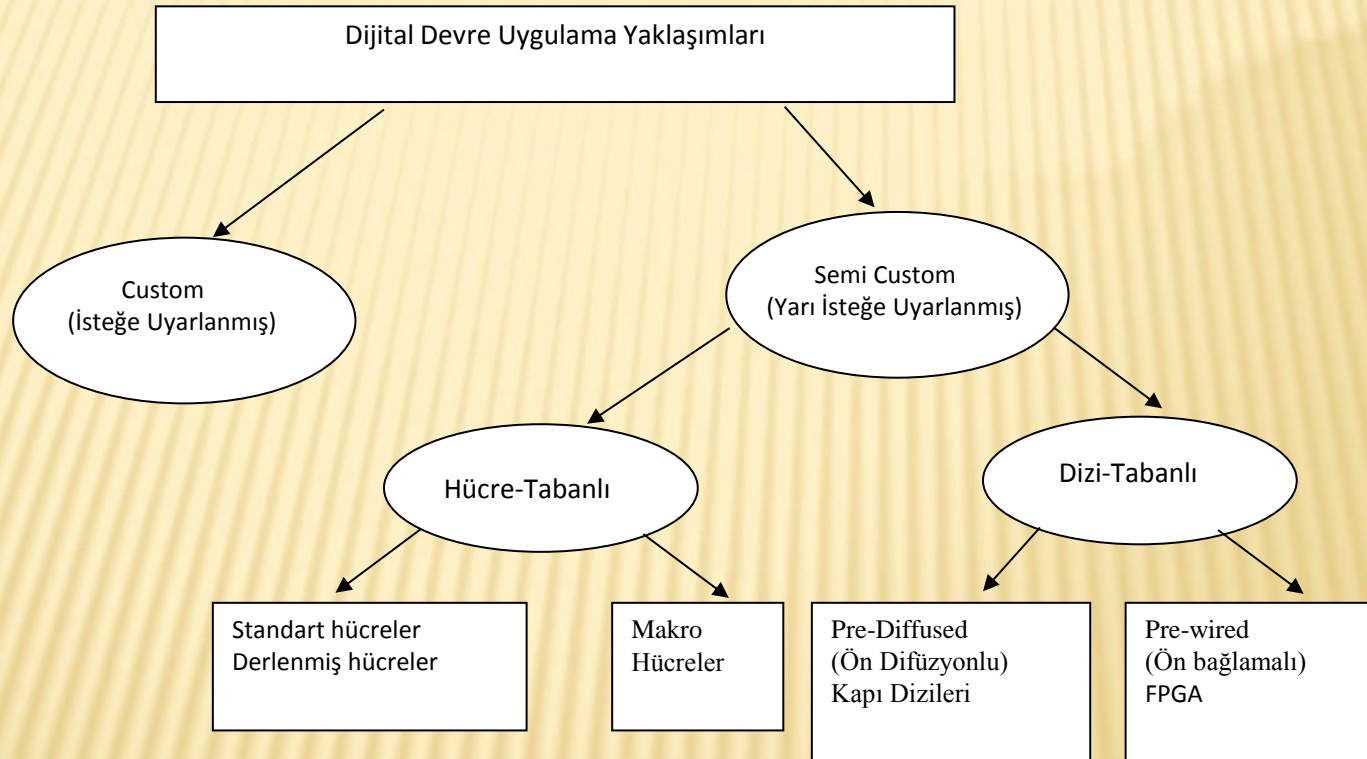
Karmaşık PLDler (CPLD)

- 80 sonları EPROM
- Birden fazla PAL bloğu
- Esnek Anahtarlama matrisi
- Makro hücre
- Öngörülebilir zamanlama
- FLASH teknolojisi

Programlanabilir Kapı Dizileri (FPGA)

- Kapı dizisi yapısında
- Programlanabilir arabağlantılar
- Esnek Giriş/Çıkış blokları
- Programlanabilir Lojik Hücreler
- Lojik fonksiyona bağlı zamanlama

UYGULAMA YAKLAŞIMLARI



Hangi Metot?

Entegre devrelerin uygulanabilirliği ve kullanılabilirliği 5 temel faktöre bağlıdır:

- Performans,
- Hız,
- Güç tüketimi,
- Fiyat,
- Üretim hacmi,

Sonuç: Dijital tasarımların çoğu performansta değil, birleşim yoğunluğunda (kapasite) rekabet ederler

Dizi – Tabanlı Uygulama Yaklaşımları

Avantajları:

- düşük imalat süresi
- düşük tasarım maliyeti
- küçük hacimli serilere uygun

Dezavantajları:

- düşük performans
- düşük birleştirme yoğunluğu
- yüksek güç tüketimi

Öndifüzyonlu (Maske Programlanabilir) Diziler

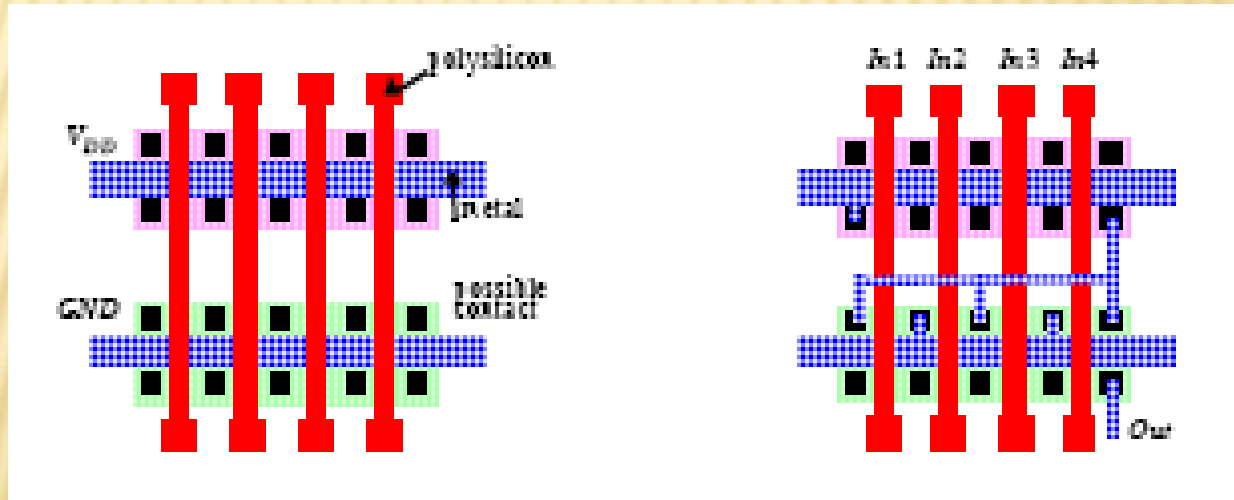
Adım 1. Basit hücre veya transistör dizileri içeren ince tabaka (wafer) grupları üretici tarafından imal edilir ve depolanır.

Adım 2. Bu işlenmemiş ince tabakaları (wafers) gerçek bir tasarıma dönüştürmek için, arzulanan arabağlantılar ilave edilir (metalizasyon adımı).

Öndifüzyonlu Diziler (Devam)

Öndifüzyonlu yaklaşım ince tabakanın (wafer) tipine bağlı olarak iki şekilde oluşturulabilir:

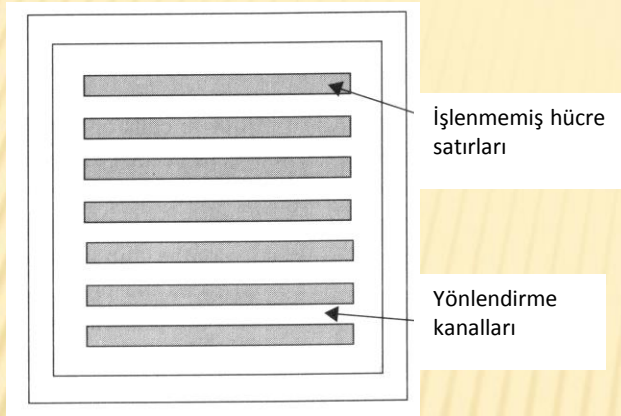
- gate-array (kapı-dizisi)
- sea-of-gates (kapılar denizi) yaklaşımı



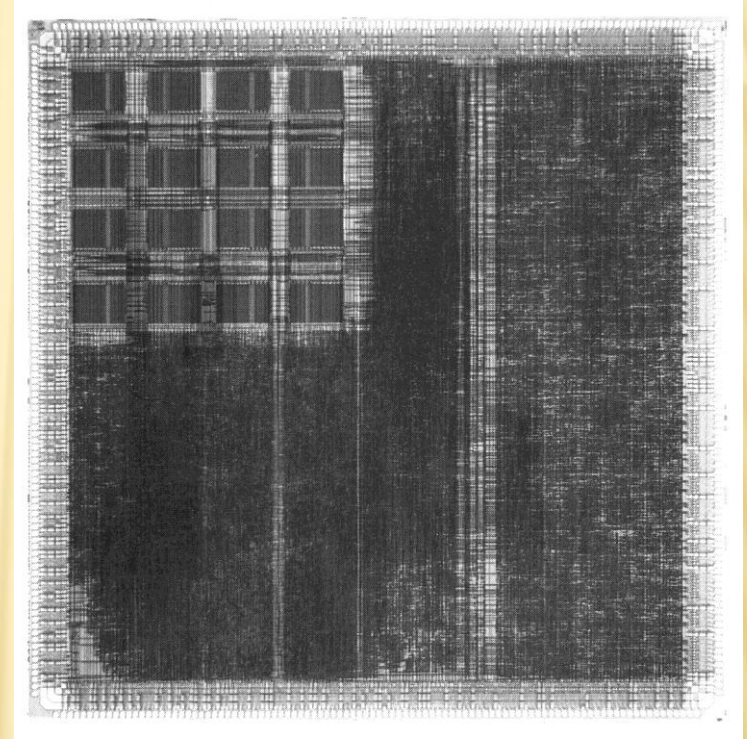
(a) İşlenmemiş gate-array hücresi

(b) Dört girişli NOR kapısı olarak programlanmış hücre

Öndifüzyonlu Diziler (Devam)



Gate-array mimarisi



Sea of Gate kalıbı mikrofotoğrafı

Önbağlamalı Diziler (Prewired Arrays)

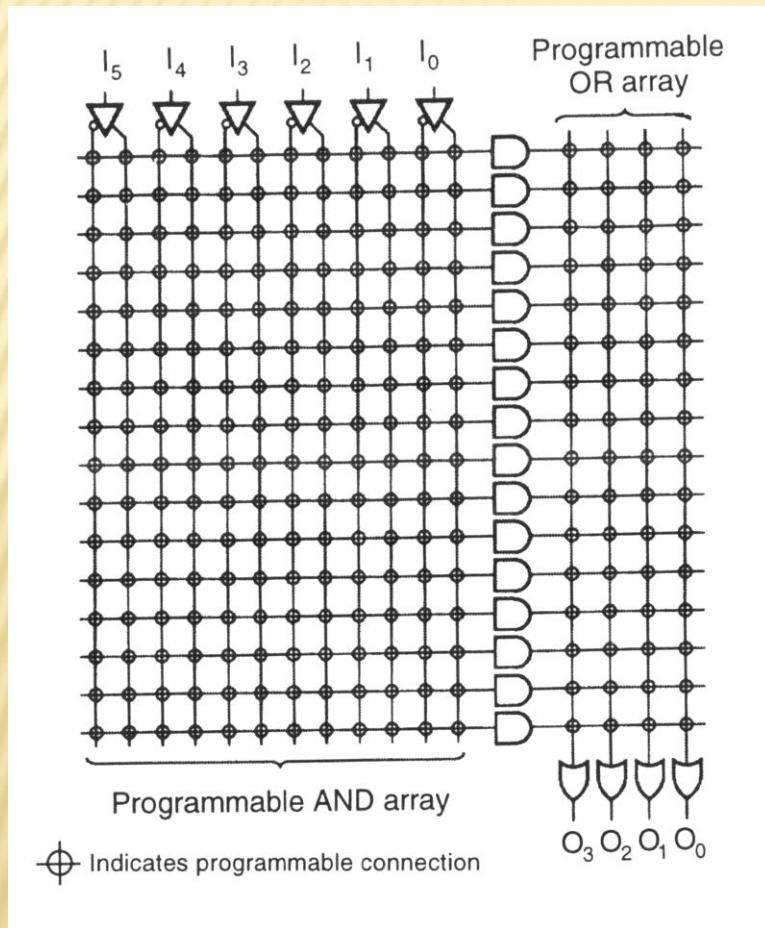
- Verilen bir Boolean cebiri ifadesini yürüten, üreticinin dışında kullanıcı tarafından tasarlanabilen önceden işlenmiş mimari.
- Avantajı tasarım işleminin üretim işleminden tamamen ayrı olmasıdır.
- En yaygın olanı Alan Programlanabilir Kapı Dizileri (FPGA) olarak adlandırılır.
- Dezavantajı performanstaki ve tasarım yoğunluğundaki kayıplardır.

Önbağlamalı Diziler (Devam)

FPGA'ler içlerinde kullanılan **programlama teknolojisine** göre gruplandırılır:

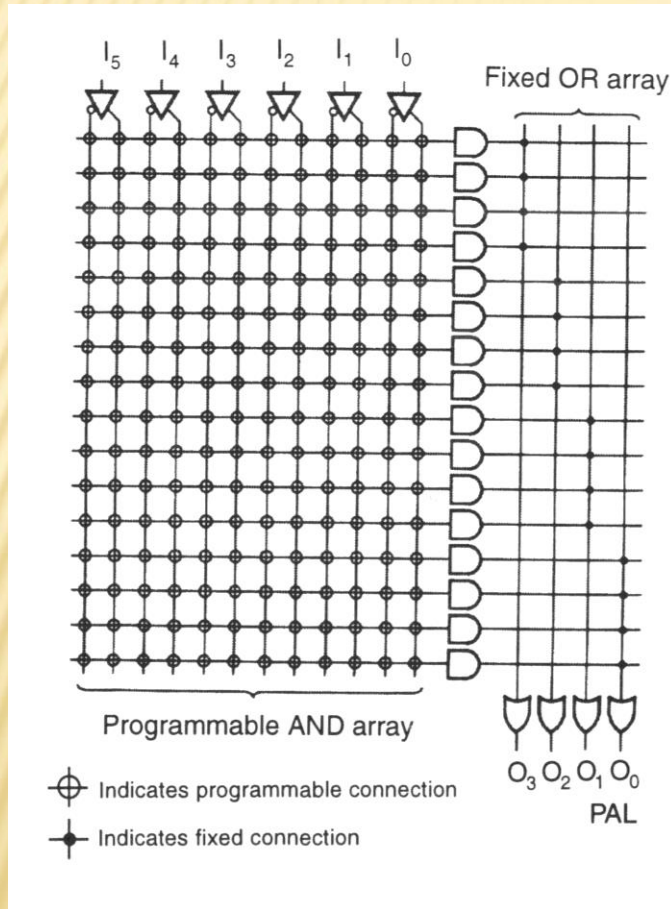
- Sigorta-Tabanlı FPGA
- Nonvolatile (Uçucu olmayan) FPGA
- RAM-Tabanlı (Uçucu) FPGA

PLA - Programmable Logic Array



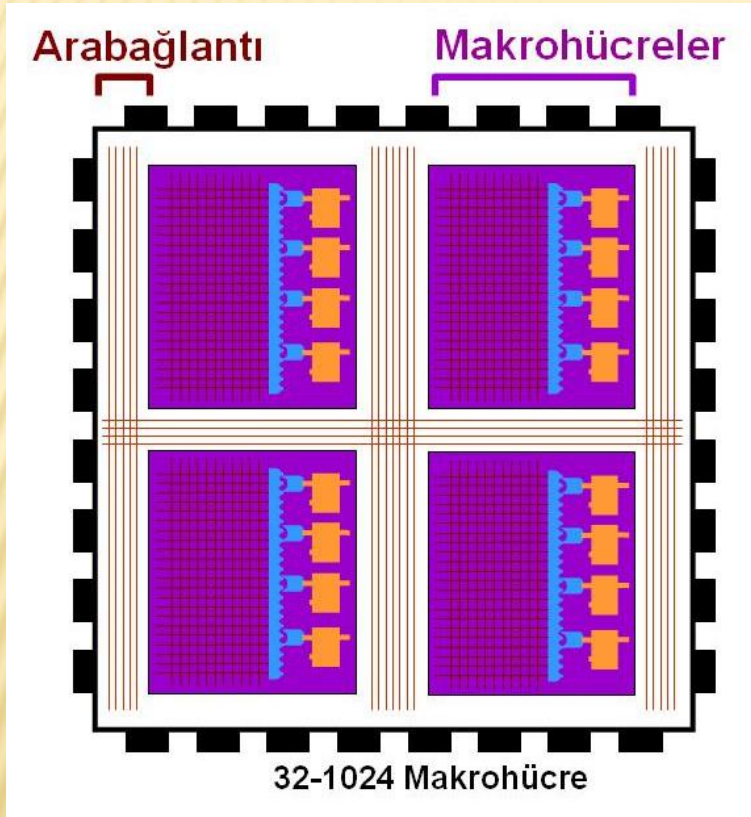
- İki programlanabilir düzlem
- VE / VEYA kapılarının sınırsız kombinasyonu
- Yüksek lojik yoğunluk
- Çok fazla sigorta, PAL'e göre daha yavaş

PAL - Programmable Array Logic



- Tek programlanabilir düzlem – VE düzlemi
- Sabit VEYA düzlemi
- VE / VEYA kapılarının sınırlı kombinasyonu
- Orta seviyede lojik yoğunluk
- Az sayıda sigorta, PLA'dan daha hızlı

CPLD - Complex Programmable Logic Device



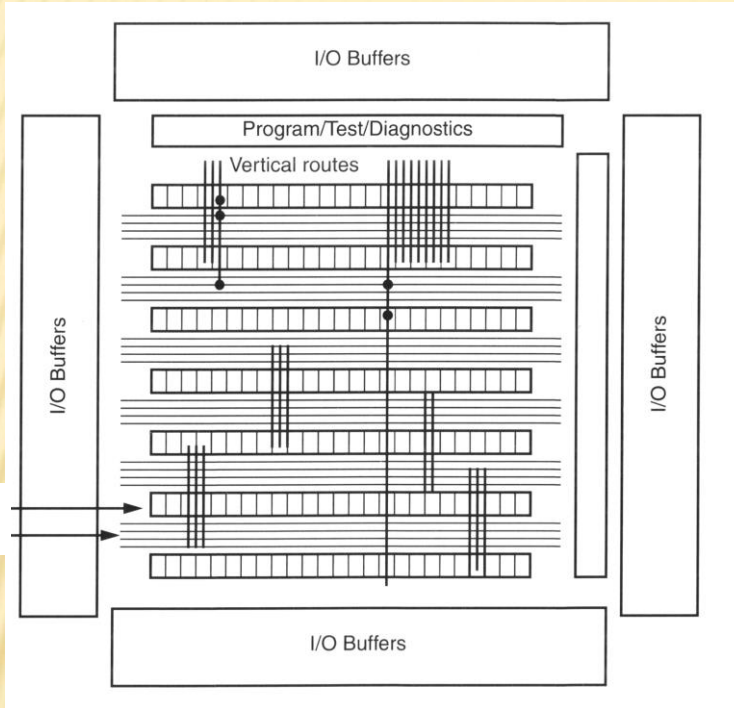
- 80 sonları EPROM
- Birden fazla PAL bloğu
- Esnek Anahtarlama matrisi
- Makro hücre
- Öngörülebilir zamanlama
- FLASH teknolojisi

FPGA – Field Programmable Gate Array



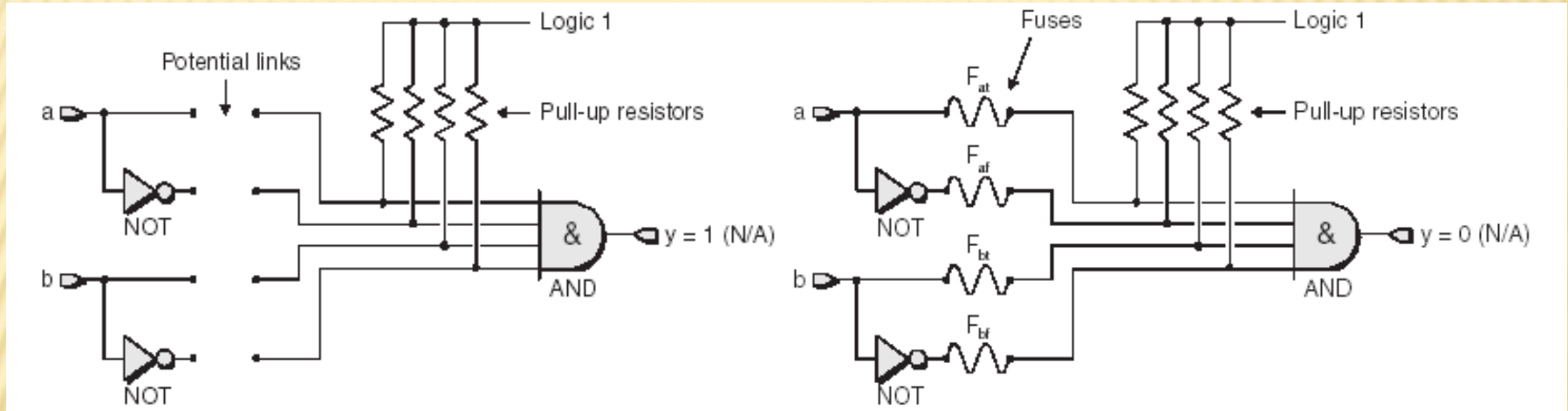
- Kapı dizisi yapısında
- Programlanabilir arabağlantılar
- Esnek Giriş/Çıkış blokları
- Programlanabilir Lojik Hücreler
- Lojik fonksiyona bağlı zamanlama

Sigorta-Tabanlı FPGA



- PLD (PAL, PLA, PROM)
- PLD'ler düzenli yapılardır ama düşük birleşim yoğunlukludur ve kötü performanslı çalışma eğilimindedirler
- PLD'lerin mimarisi göz önünde tutularak, diğer mimarilerle (sea-of-gate, standart-hücre) harmanlanarak FPGA'ler oluşturulur.

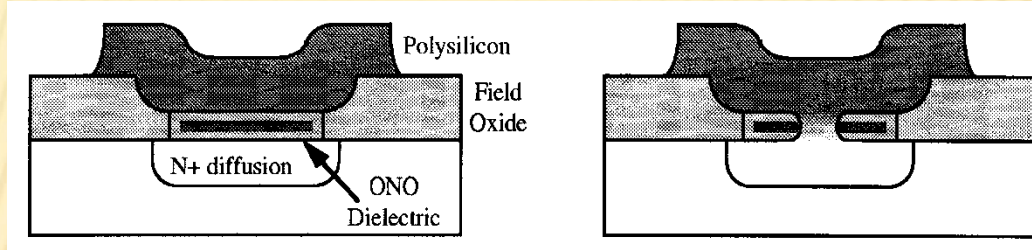
Sigorta Mantığı



Sigorta elemanının devrede kullanılması

Antisigorta Mantığı

ACTEL firmasının Programlanabilir Düşük Empedanslı
Devre Elemanı (PLICE)

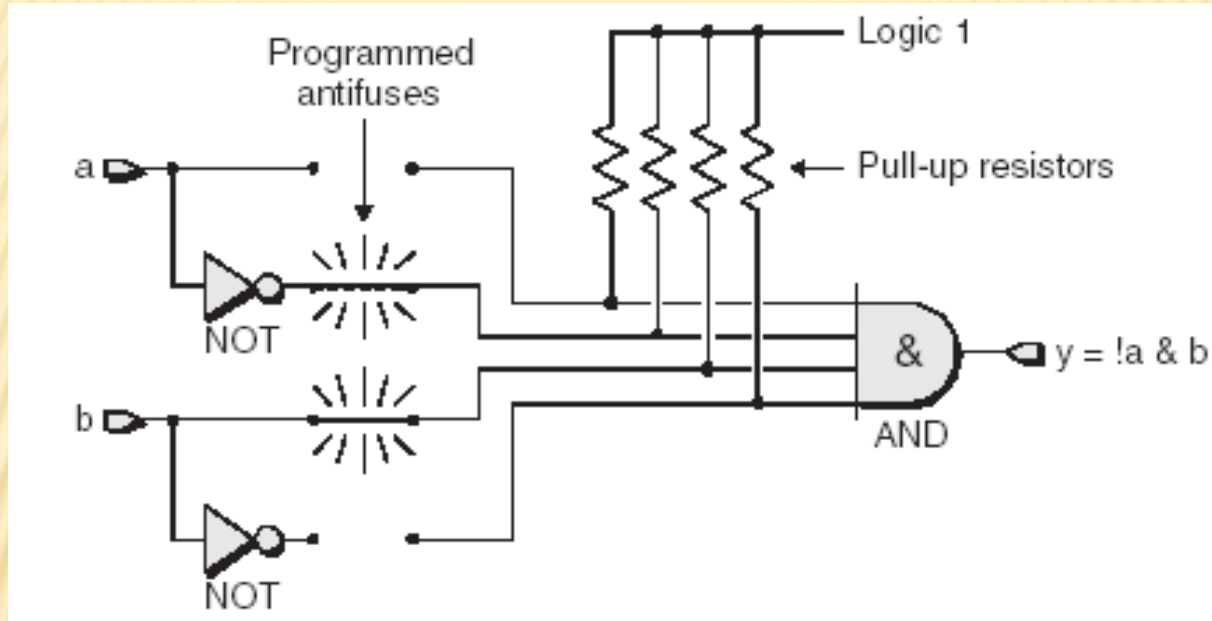


Patlatılmamış Antifuse

Patlatılmış Antifuse

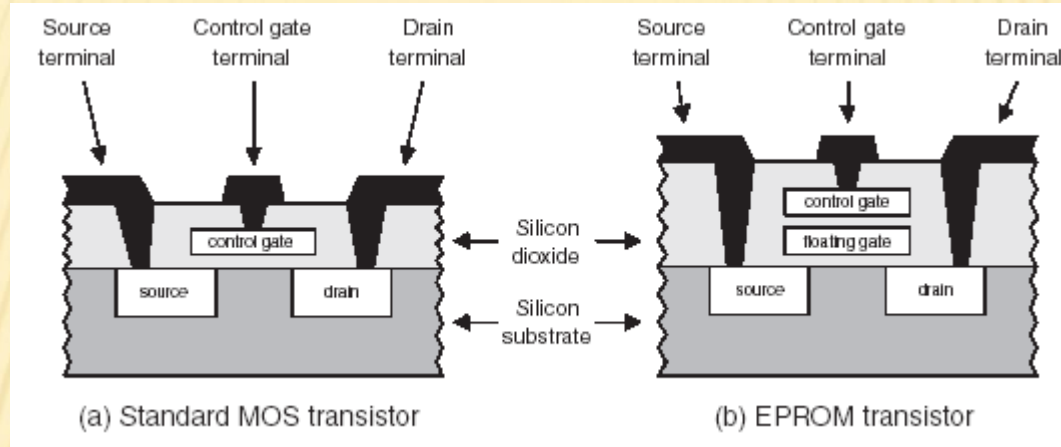
- Antisigorta da programlanmamış durumda tüm bağlantılar açık anahtar gibidir (yüksek direnç -100M Ω)
- Dielektrik maddeye yüksek voltaj verilerek kapalı anahtar (200-500 Ω) konumuna getirilir.
- Programlama işlemi bir kereye mahsustur, elemanlar tekrar tekrar programlanamaz.

Antisigorta Mantığı (devam)



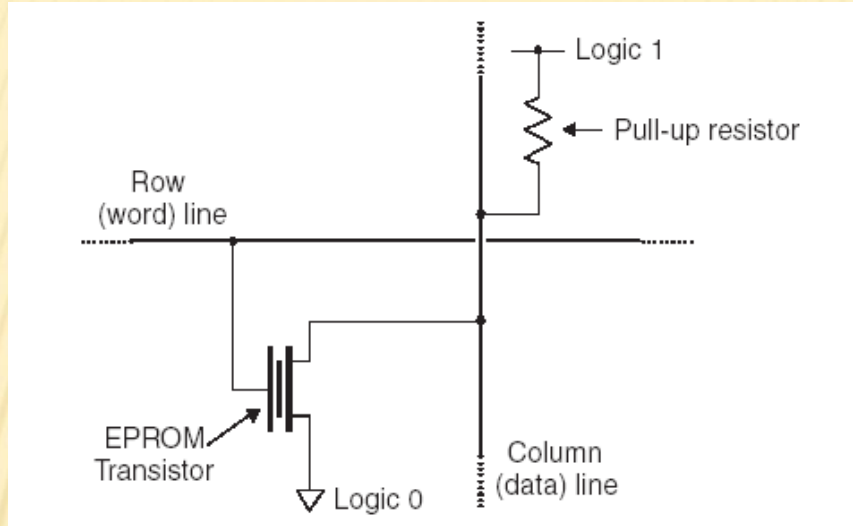
Antifuse elemanı kullanılan bir devre

Nonvolatile FPGA



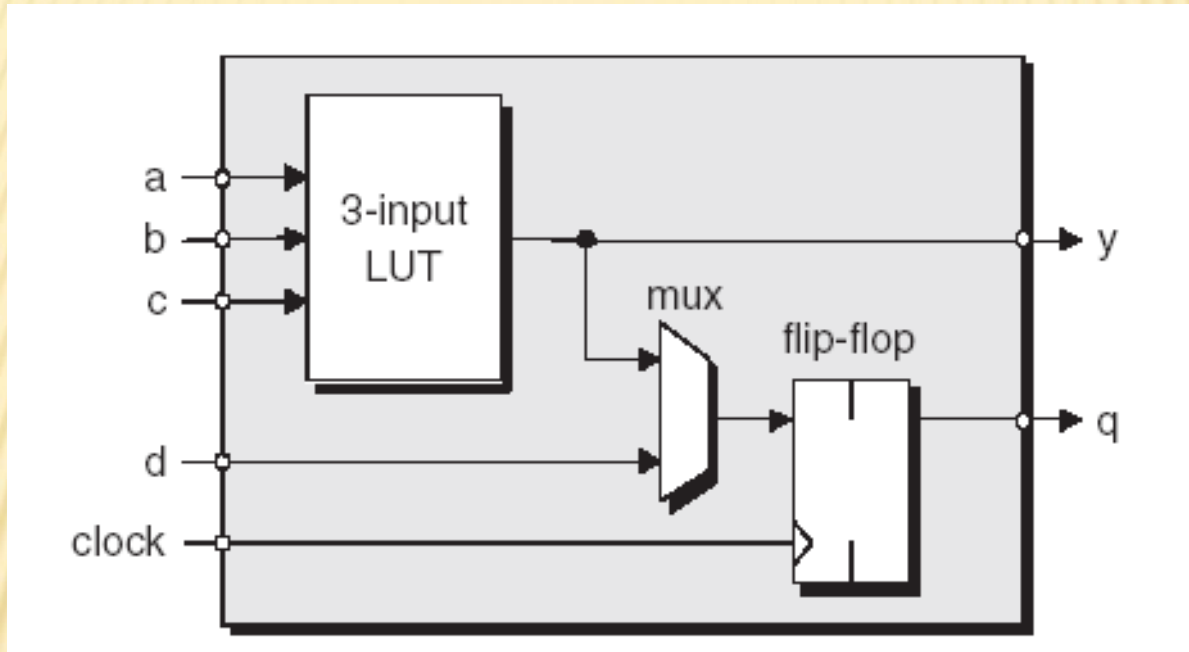
- Kayan gate terminali düzenli olarak MOS transistör gate terminali ve kanal arasında yer alır.
- UV ışık ile bilgiler silinebilir.
- İçerisinde bulunan EPROM transistörden dolayı EPLD adı da verilir.

Nonvolatile FPGA (devam)



- Hücreyi programlamak için transistörün drain terminali 12 voltta tutulurken, kontrol gate'e yaklaşık 13-14 volt uygulanır. Bu uygulama kayan gate'in negatif yüklenmesiyle sonuçlanır ve transistörün eşik voltajı artar (yaklaşık 7 volt).
- Böylece tüm normal devre voltajlarında (maksimum 5-6 volt) transistörler kalıcı olarak "off" kılınır.
- Kalıcı olarak "off" yapılan transistörler 125C°'de en az 10 yıl konumlarını korurlar.

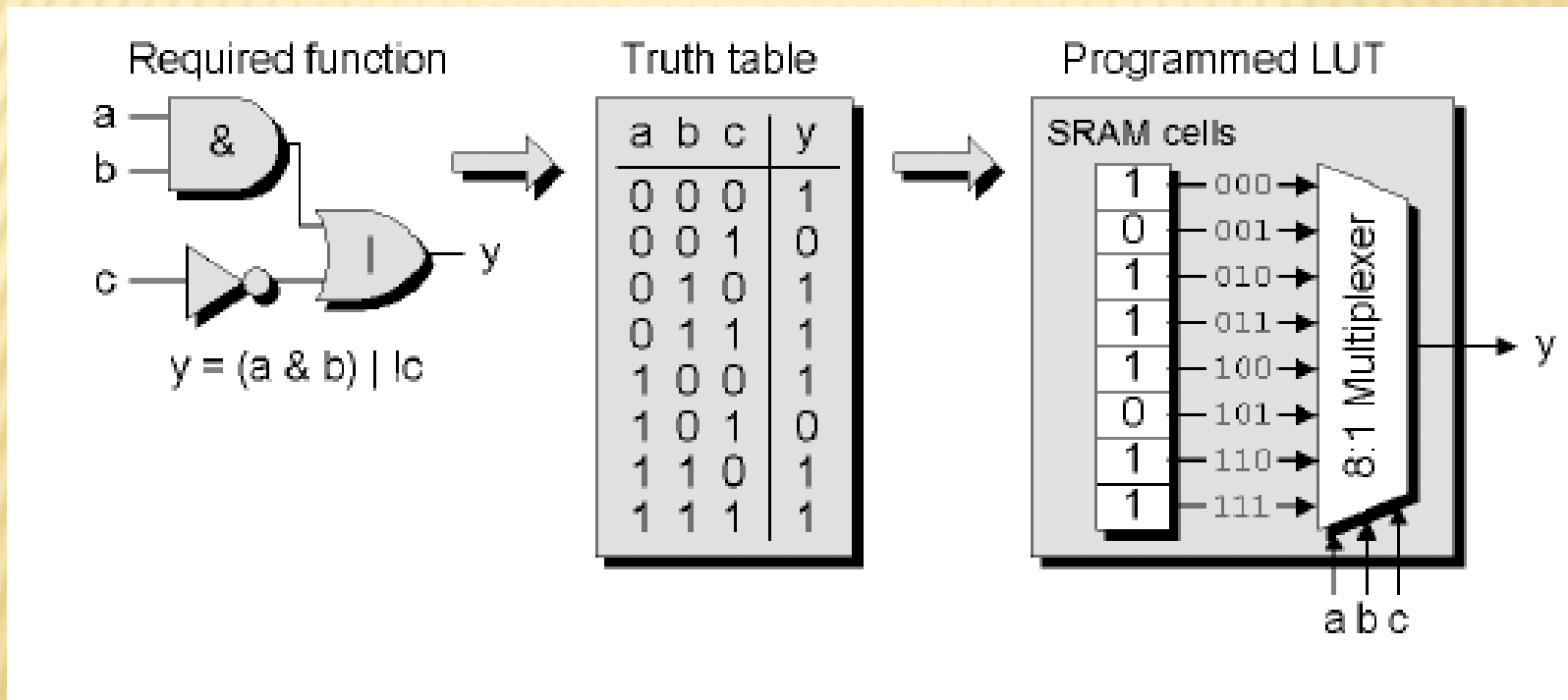
RAM – Tabanlı FPGA



RAM Tabanlı FPGA Lojik birimi (CLB –Configurable Logic Block)

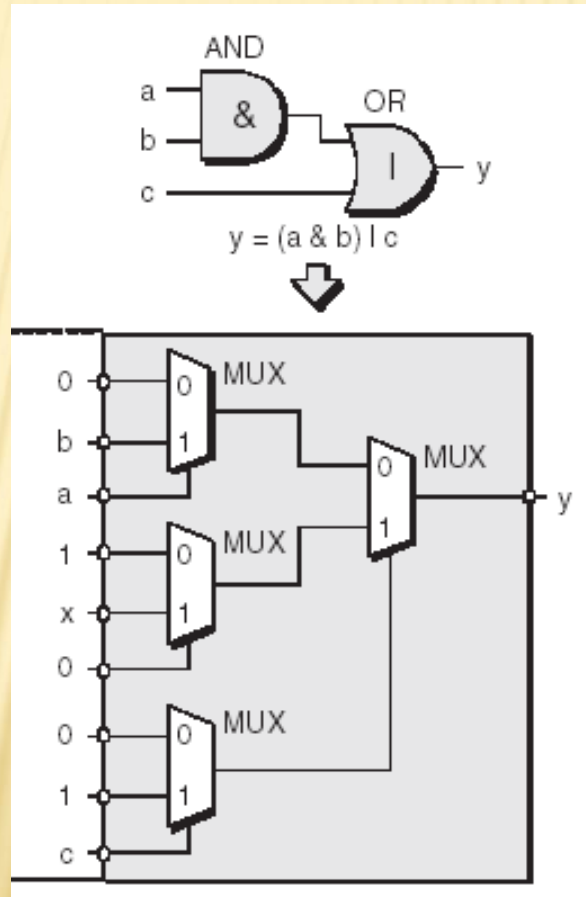
RAM – Tabanlı FPGA (devam)

Örnek Tasarım 1

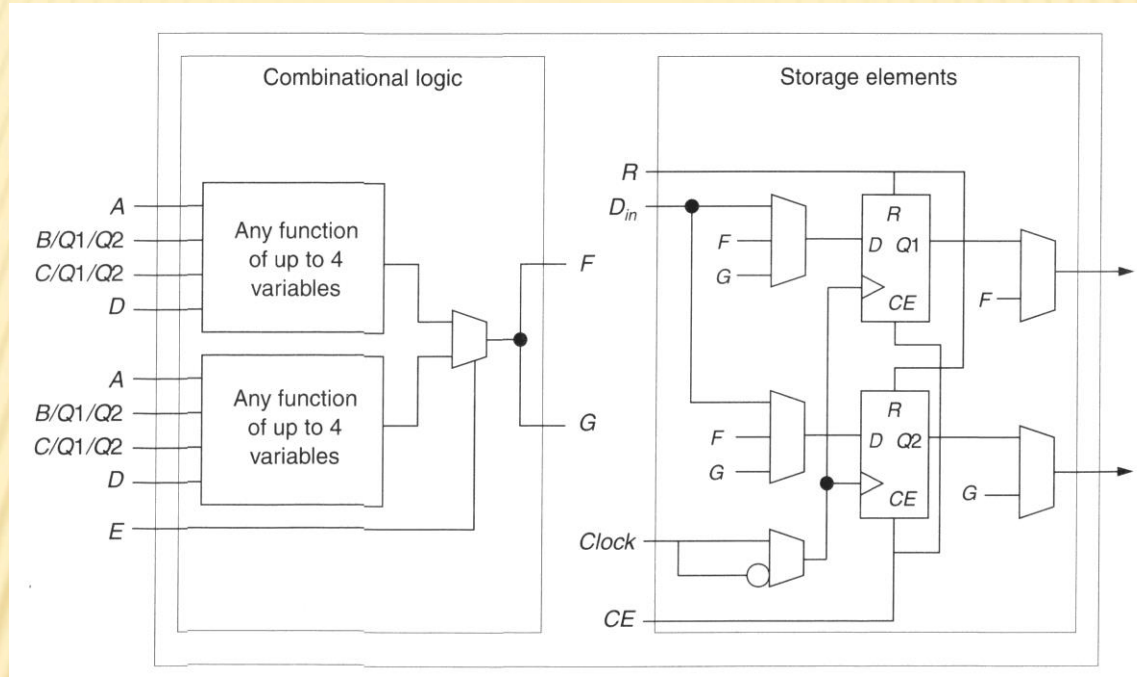


RAM – Tabanlı FPGA (devam)

Örnek Tasarım 2

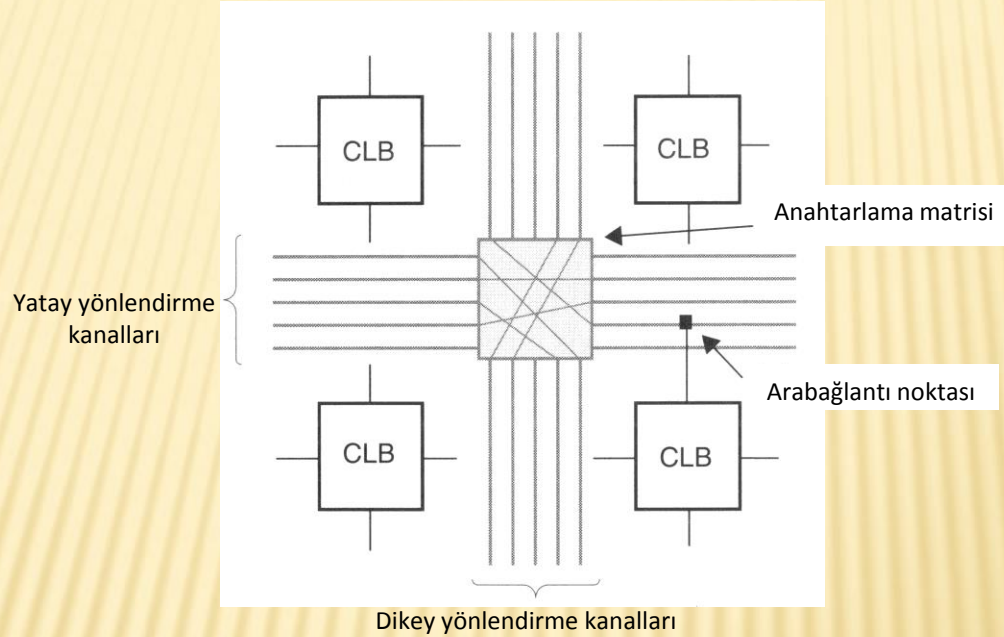


RAM – Tabanlı FPGA (devam)



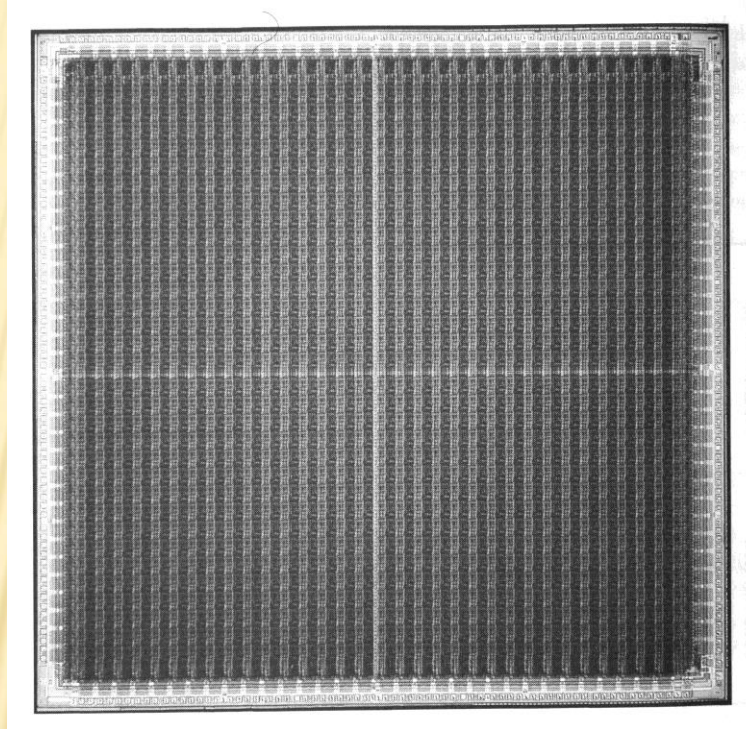
Gelişmiş bir CLB yapısı

RAM – Tabanlı FPGA (devam)



Programlanabilir Çevresel Arabağlantı Ağı

RAM – Tabanlı FPGA (devam)



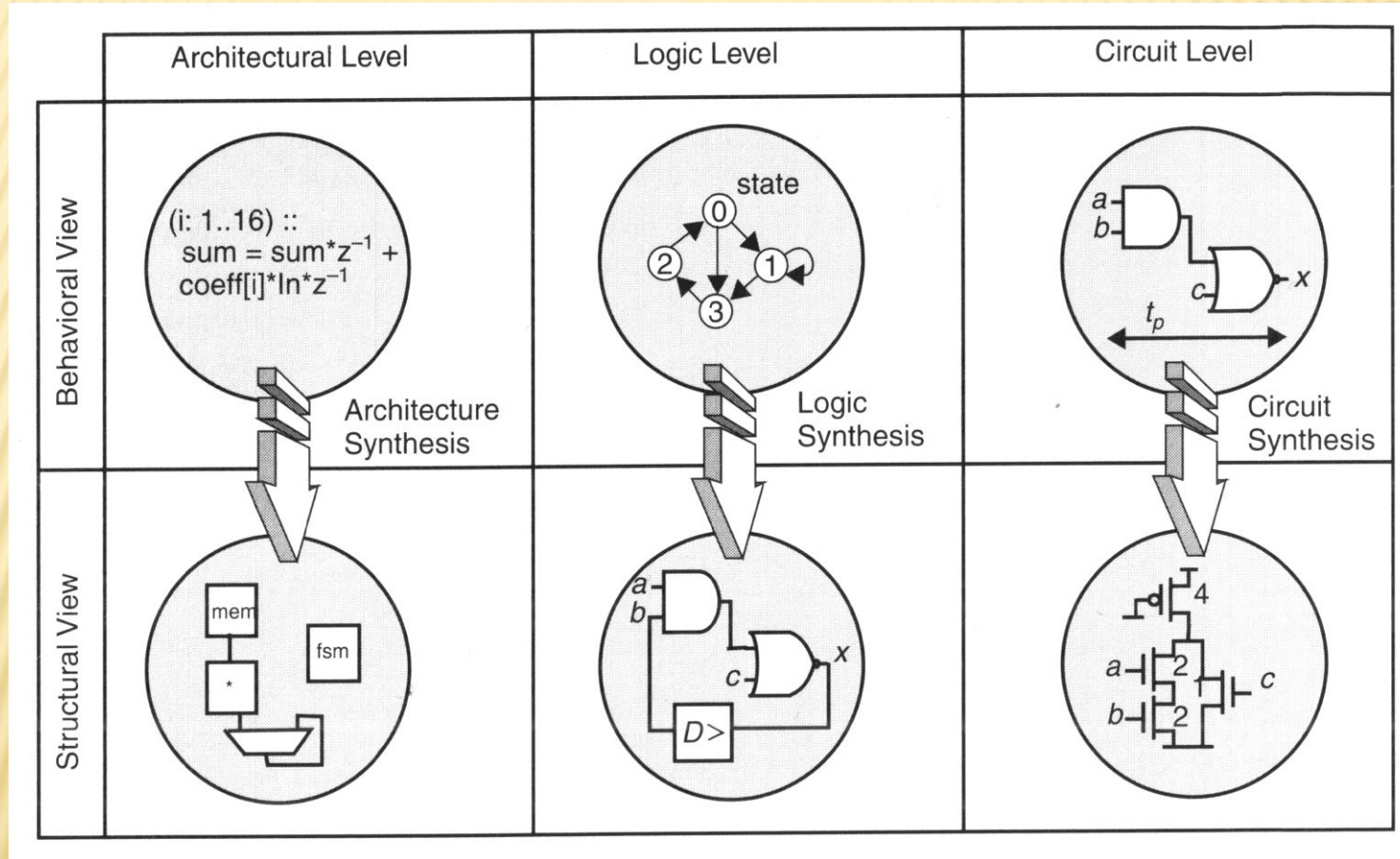
XC4025 FPGA ip mikrofotoğrafı

TASARIM SENTEZİ

Sentez: İki farklı tasarım durumunun aralarında dönüşümü.

- Dönüşüm tipik olarak davranışsal (behavioral) durumdan yapısal (structural) duruma olur.
- **Davranışsal durum** - teorik durum,
- **Yapısal durum** - pratik durum
- Teorinin (matematiksel gerçeklerin) pratiğe (elektronik devreye) çevrilmesi işlemi.

TASARIM SENTEZİ



Devre Sentezi

- Lojik devreyi transistör ağına konulmuş çeşitli kısıtlamalara uyarak dönüştürme işlemi.
- Bu işlem iki aşamada gerçekleştirilir:
 - 1.** Lojik denklemden (devreden) transistör bağlantı şemalarının çıkarılması : complementary static, pass-transistörler, dinamik, DCVSL vb. gibi devre sitilinin seçilmesi ve lojik ağın inşa edilmesi.
 - 2.** Transistöre dönüştürülen devredeki transistörleri boyutlandırma: performans, sensitivite, parazitler, alan, fan-out ve iç kapasitans gibi değişkenlerin belirlenmesi.

Lojik Sentez

- Lojik-seviye modeli durumundan yapısal duruma dönüştürme işlemidir.
- Lojik model durum geçiş diyagramları, sonlu durum makineleri, şematik diyagramlar, Boolean eşitlikleri, doğruluk tabloları ve HDL açıklamaları gibi birçok farklı şekil kullanılabilir.
- Devrenin çalışmasına (combinational veya ardışıl) ve uygulanacak mimariye (çoklu seviye lojik, PLA veya FPGA) göre değişir.
- Sentez işleminde alan, hız, güç gibi tasarım değişkenleri ayarlandıktan sonra iki farklı lojik optimizasyon sistemi düşünülebilir.
 1. Çeşitli Boolean veya cebirsel manipülasyonların kullanıldığı teknolojiden bağımsız evre.
 2. Teknolojiden bağımsız sonuçların arzu edilen uygulama mimarisine karmaşıklıklar ve özellikler göz önünde bulundurularak teknoloji bağımlı şemalandırma.

Mimari Sentez

- Sentez alanında en son gelişmedir.
- Davranışsal veya yüksek seviye sentez olarak da anılır.
- Görevi performans, güç, alan gibi kısıtlamaları ayarlayıp, icra edilecek görevin davranışsal açıklamasını yapısal mimari tasarımına dönüştürmektir.
- Yazılımlarda programları derleyen derleyicilerin (compiler) donanım versiyonuna benzetilebilir.

TEST İŞLEMİ

- Test düğüm noktalarındaki karakteristiklerin davranışsal analizi ile yapılır.
- Tasarımcı hangi düğümlerin cevaplarını ve I/O paternlerini gözleme konusunda serbesttir.
- Mikroişlemciler gibi milyonlarca transistörden oluşan karmaşık elemanların birçok test yöntemi vardır.
- Test sırasında izlenen temel prosedür şöyledir:
 - Diagnostic (teşhis) test** board'un veya çipin debug işlemi sırasında kullanılır. Hatalı parçayı, parçanın hata tanımını ve hatalı parçanın konumunu tespit etmek bu testin amacıdır.
 - İşlevsellik testi** bir elemanın çalışıp çalışmadığını belirler. Test sonucu iki ihtimaldir. Evet yada hayır. Bu yüzden diagnostic testten daha kolay bir testtir. Her parçaya uygulanmasından dolayı test hızlı olmalıdır.
 - Parametrik test** gürültü sınırı, yayılım gecikmesi ve maksimum clock frekansı gibi birbirinden bağımsız düşünülemeyen parametreleri kontrol eder. Bu test yöntemi sadece evet/hayır yanıtı veren işlevsellik testinden daha değişik kurgulara sahiptir. Parametrik test genellikle statik (dc) ve dinamik (ac) test olarak ikiye ayrılır.

TASARIM BÜTÜNLÜĞÜ

- 74 serisi standart lojik kapıları düşük maliyetli CPLD'lere entegre etmek çok etkili bir yöntemdir. Sadece PCB alanından ve bakır tabakalardan yani toplam sistem maliyetinden tasarruf etmeyecek aynı zamanda 20 farklı tanımlanmış lojik devre ailesini satın almak ve depolamak yerine genel bir parça satın alınmış olacaktır.
- Üretim aşamasında cihaz sadece bir eleman alıp onu yerleştirecek ve üretim hızı artacaktır.
- Daha az eleman daha yüksek kalite ve daha iyi bir Hata/Zaman (Failure In Time-FIT) faktörü sağlayacaktır.
- Gelişmiş ürünlerde düşük güç harcaması ve azaltılmış sıcaklık emisyonu sağlanarak soğutucu kullanımının önüne geçilecek ve yüksek güvenilirlikte ürünler ortaya çıkmış olacaktır.

TASARIM SÜRECİ

- Geliştirilmiş özel tasarım programları sayesinde programlanabilir mantık devreleriyle tasarım oldukça kolaylaşmıştır.
- Tasarımlar herhangi bir tanımlama dili örneğin ABEL (Advanced Boolean Expression Language), VHDL (VHSIC Hardware Description Language), Verilog ile yada bir şematik çizim programı kullanılarak kolay ve çabuk tanımlanabilir.
- Şematik çizim programları (Schematic capture) tasarımcıların yıllardan beri programlanabilir lojik elemanları ve kapı dizilerini tanımlamak için kullandığı alışlagelmiş bir yöntemdir. Bu tasarımcının istediği kapıyı ve istediği bağlantıları belirtebileceği grafik tabanlı bir programdır.

Şematik Çizim

➤ Şematik çizimde dört adım vardır:

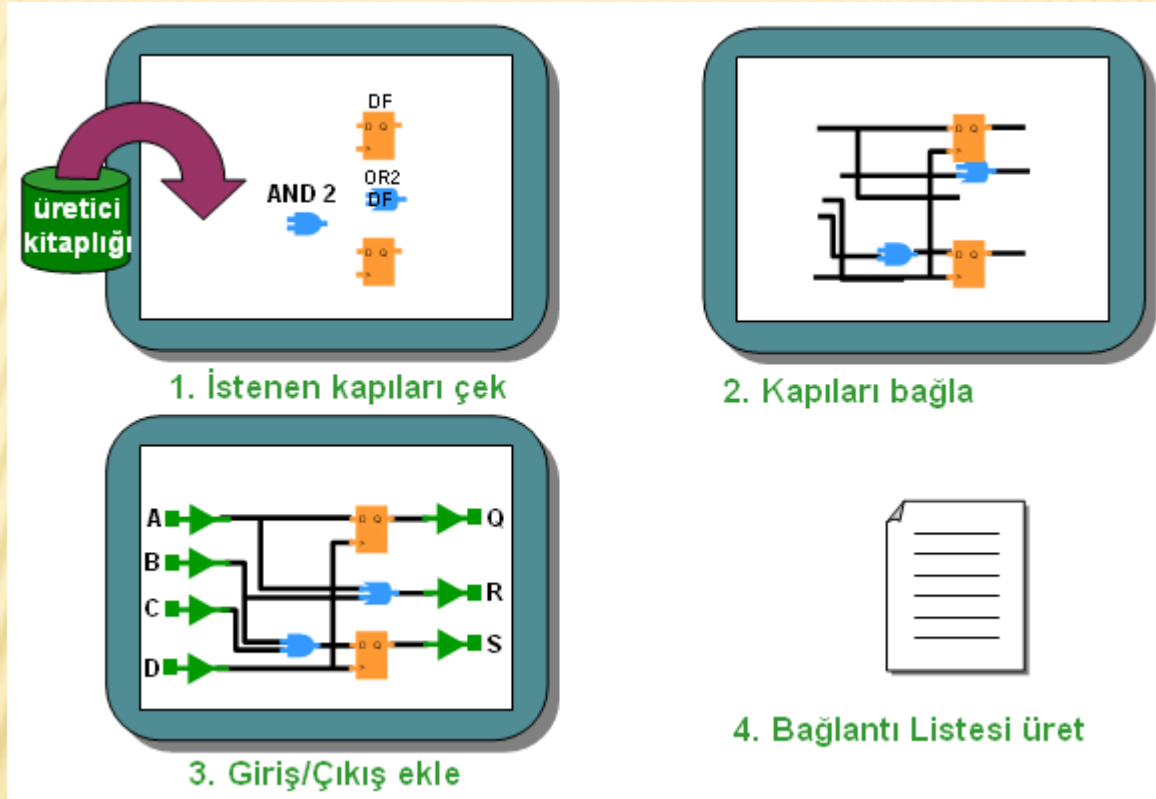
1.Adım: Özel bir şematik çizim programı ve entegre kütüphanesi seçildikten sonra, tasarımcı seçilen kütüphaneden istediği lojik kapıları yükleyerek devreyi oluşturmaya başlar. İstenilen kapı kombinasyonları oluşturulabilir. Bu amaçla seçilecek programın ait olduğu firma ve entegre ailesinin kütüphanesi bu aşamada çok önemlidir. Bunun yanında kullanılacak entegrenin kılıf yapısı ve hızı çok fazla önemsenmemektedir.

2.Adım: Netlist veya hatları kullanarak kapıları bağlamak. Uygulama için istenilen konfigürasyonda kapıları bağlamak tamamen tasarımcının kontrolü altındadır.

3.Adım: Giriş ve çıkış buffer'ları eklenir ve adlandırılır. Böylece entegrenin I/O pinleri tanımlanmış olur.

4.Adım: Son işlemde bir netlist dosyası oluşturulur.

Şematik Çizim (devam)



➤ Tasarımcı, devreyi grafik olarak tanımlayabilir.

Şematik Çizim - Netlist

- Netlist, tasarım programı tarafından örneğin şematik çizim programı ile üretilen, devrenin metin eşdeğeridir.
- Netlist, diğer programların, devrede hangi kapıların kullanıldığı, nasıl bağlandığı ve I/O pinlerinin isimleri gibi bilgileri anlayabilmesi için oluşturulmuş özlü bir yoldur.
- Yandaki örnekte, netlist şematik programda devrenin gerçek eşdeğerini yansıtır.
- Her bir eleman için ve her bir net için birer satır kullanılır. Bilgisayar elemanlara (G1-G4) ve netlere (N1-N8) isimler atar.
- Bu tasarım gerçekleştirilirse, giriş uçları A,B,C,D ve çıkış uçları Q,R,S olacaktır.
- Üretici firma tabanlı dosyalar (XNF - Xilinx Netlist Format) ve Endüstri tabanlı standart netlist formatı (EDIF - Electronic Digital Interchange Format) kullanılır.
- Eğer bir tasarımın netlist dosyası varsa, o devrenin işlevini tespit etmek için gerekli her şey hazırdır.

design netlist

```
Component AND2 G1;  
Component OR2 G2;  
Component DF1 G3;  
Component DF1 G4;
```

```
Net N1: A, G2:a, G3:d;  
Net N2: B, G2:b, G1:a;  
Net N3: C, G1:b;  
Net N4: D, G3:c, G4:c;  
Net N5: G3:q, Q;  
Net N6: G1:y, R;  
Net N7: G2:y, G4:d;  
Net N8: G4:q, S;
```


Neden Donanım Tanımlama Dilleri?

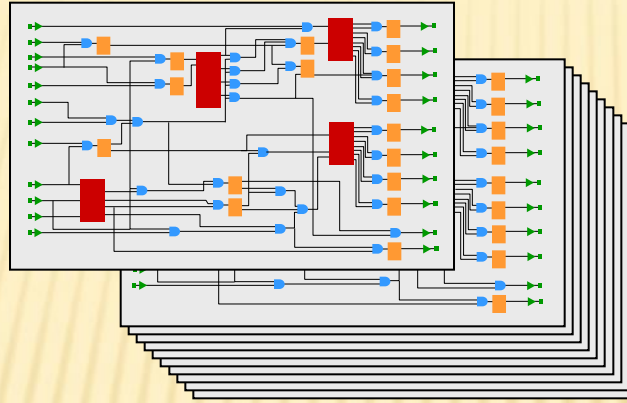
- Daha gerçekçi bir örnek vermek gerekirse 10.000 eşdeğer kapıyla oluşturulan bir tasarımı göz önüne alalım.
- Tipik bir şematik çizim sayfası, makro lojik bloklarla beraber yaklaşık 200 kapı içermektedir. Bundan dolayı 10.000 kapılık tasarımı gerçekleştirmek için 50 çizim sayfası gerekmektedir.
- Her bir sayfa için yukarıda belirtilen adımlar tekrar edilmelidir: elemanları ekleme, kapılar arasındaki bağlantıları gerçekleştirme, I/O pinlerini ekleme ve netlist oluşturma.
- Bu işlem oldukça uzun süren ve zahmetli bir yoldur özellikle 20k, 50k ve daha yüksek kapasiteli tasarımlar için.

Neden Donanım Tanımlama Dilleri? (devam)

- Şematik çizim programını kullanmanın bir diğer problemi ise firmalar ve teknolojiler arasında geçiş yapmanın zorluğudur.
- Şöyle ki, bir A firmasının ürettiği FPGA ile 10.000 kapı kapasiteli bir tasarım gerçekleştirdiniz ve bunu başka bir kapı dizisine transfer etmek istediniz.
- O zaman yapmış olduğunuz tasarımın 50 sayfasını tek tek, diğer bir firmanın ürettiği kapı dizisinin eleman kütüphanesini kullanarak değiştirmek zorunda kalacaksınız.
- Bunun için daha iyi bir yöntem vardır. Bu yöntem High Level Design-HLD (Yüksek Seviyeli Tasarım), Behavioural (Davranışsal) veya Hardware Description Language – HDL (Donanım Tanımlama Dili) olarak adlandırılır.

Donanım Tanımlama Dilleri (devam)

Ortalama her bir
sayfada 200 kapı
yer alır



50 sayfa şema !

$$200 \text{ kapı/sayfa} \times 50 \text{ sayfa} = 10,000 \text{ eşdeğer kapılık tasarım}$$

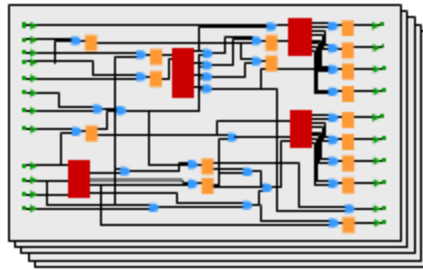
Donanım Tanımlama Dilleri

- Grafiksel bir alçak seviyeli kapı tanımlaması yerine yüksek seviyeli bir dil kullanarak devreyi bir metin dosyası olarak tanımlamaktır.
- *Behavioural* (Davranışsal) teriminin bu dilde kullanımının nedeni, tasarımcının bir uygulama oluşturmak için gerekli olan kapıları çizim yerine devrenin fonksiyonunu veya davranışını kelimelerle tanımlayabilmesidir.
- İki farklı HDL formatı vardır: **VHDL** (Very High Speed Integrated Circuit High-level Design Language) ve **Verilog**.

Donanım Tanımlama Dilleri (devam)

- HDL tanımlama ile yapılan bir tasarım tamamen firmalara bağımsız olarak ve de daha kısa bir sürede gerçekleştirilebilir.
- Bu ise oluşturulan aynı kodun hem bir FPGA hem de bir Lojik Kapı Dizisi gerçekleştirmede kullanılabilmesi demektir.
- Örneğin 32x32 çarpma ünitesi gerçekleştirmek için, şematik yaklaşımda önceden oluşturulmuş bir çarpıcının 30 sayfasının 3 kez daha ve daha büyük bir bus (hat) genişliği için tekrar çizilmesi demektir.
- Bu yaklaşık olarak 4 saatlik bir grafik düzeltme işlemi demektir.
- HDL tanımlama için ise bu sadece iki satırda hat genişliklerini değiştirmekle 4 saniye içinde halledilebilir.

Donanım Tanımlama Dilleri (devam)



Şematik tasarım

- 6,000 kapı
- 30 şematik sayfa
- 3 gün tasarım süresi
- Üreticiye bağımlı tasarım

VEYA

HDL de yazılmış dosya

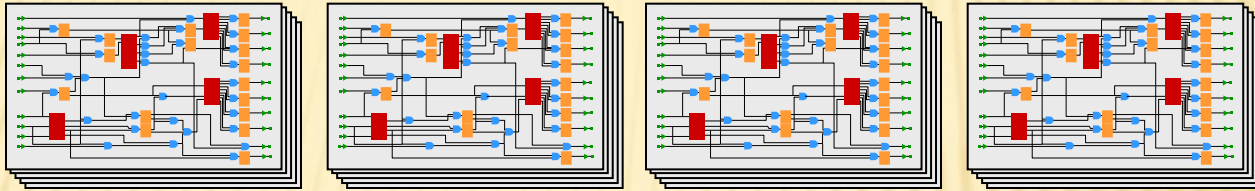
```
entity MULT is
  port(A,B:in std_logic(15 downto 0);
        Y:out std_logic(31 downto 0));
end MULT;

architecture BEHAVE of MULT is
begin
  Y <= A * B;
end BEHAVE;
```

- 6,000 kapı
- 1 text dosya
- 8 satırlık dosya
- 3 dakikada yazım
- Tamamen üreticiden bağımsız!!

Donanım Tanımlama Dilleri (devam)

32 x 32 Çarpma Devresine Dönüştürme



30 sayfayı 3 kere kopyala ve 90 sayfada düzeltme: 4 saat

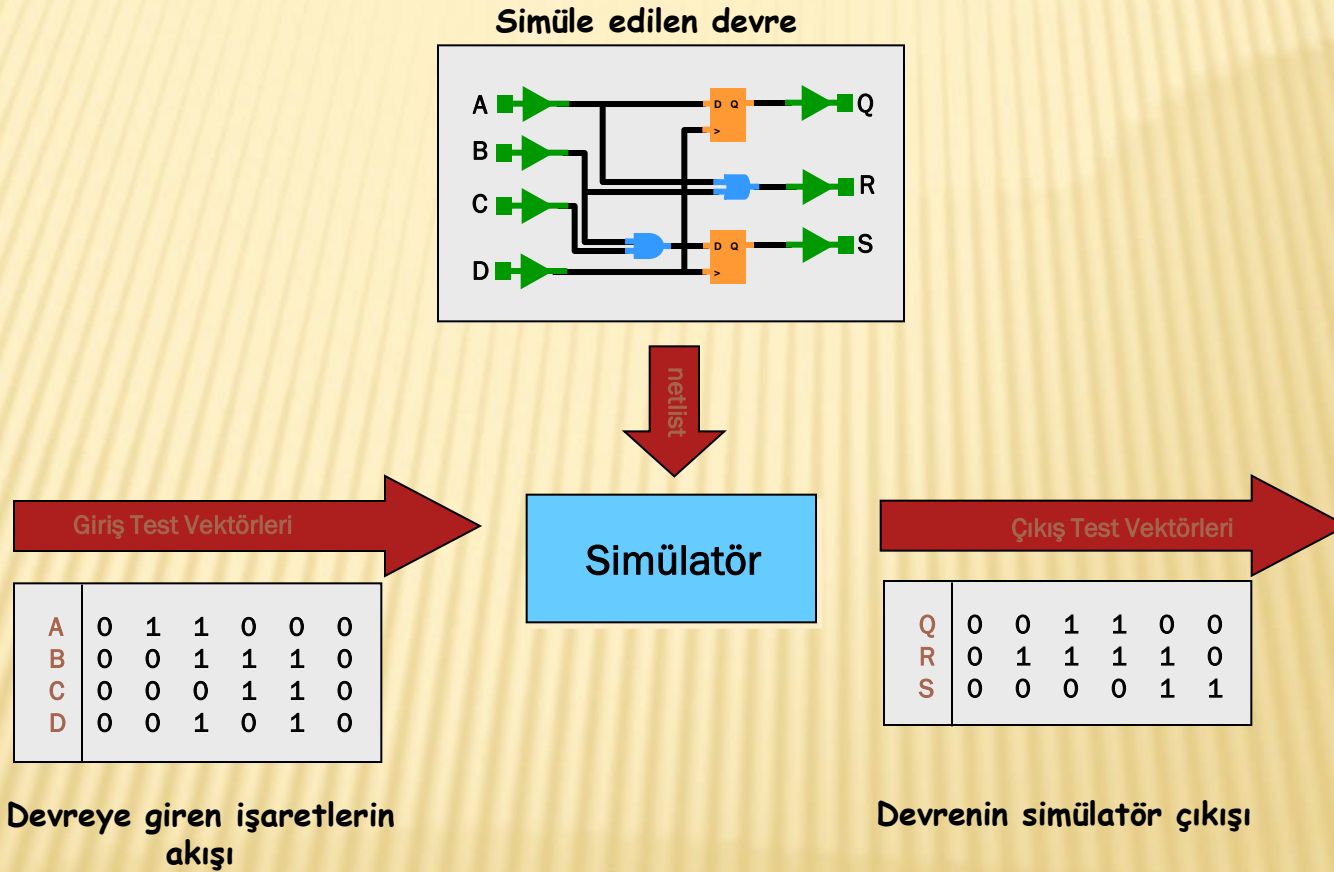
VEYA

15 yerine 31,
31 terine 63:
4 saniye

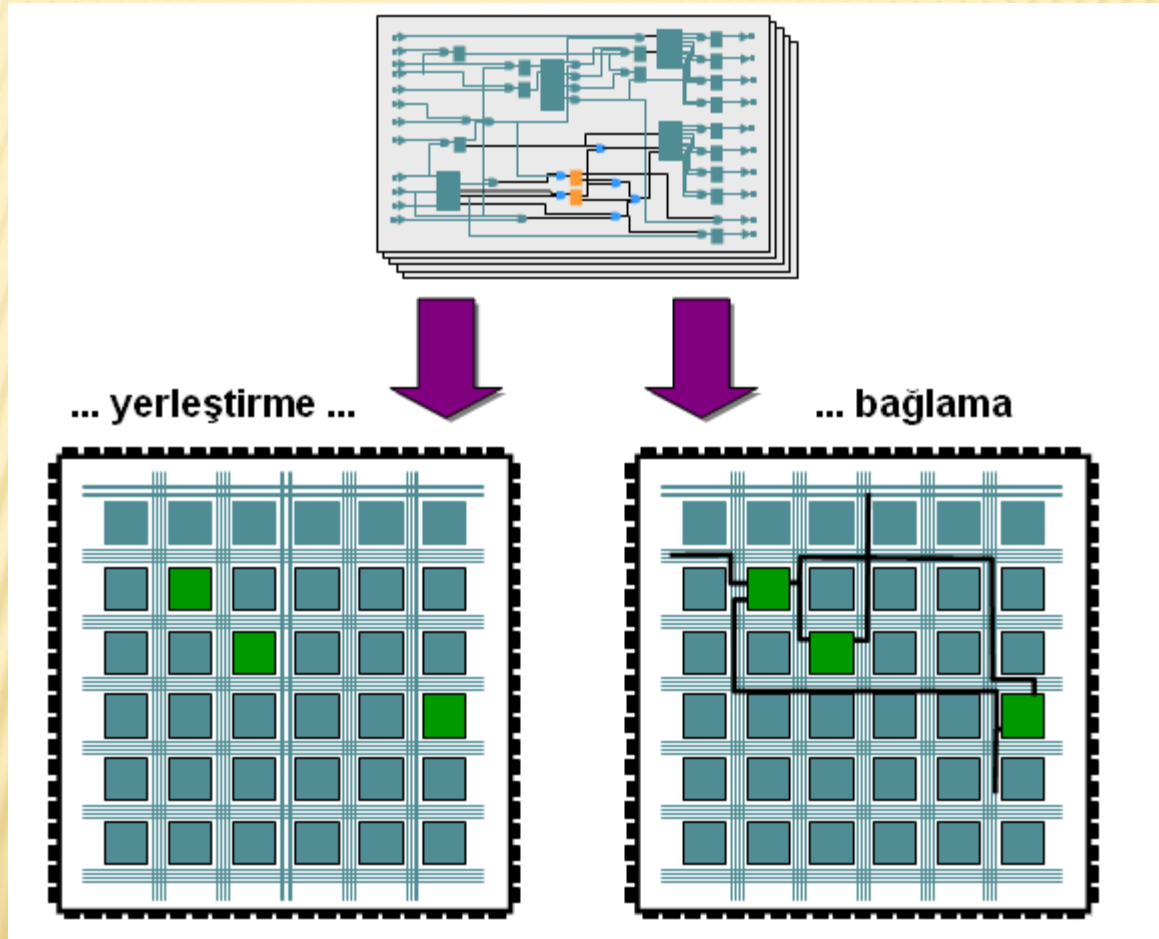
```
entity MULT is
  port(A,B:in std_logic(1531 downto 0);
        Y:out std_logic(3163 downto 0));
end MULT;
```

```
architecture BEHAVE of MULT is
begin
  Y <= A * B;
end BEHAVE;
```

Simülasyon Aşaması

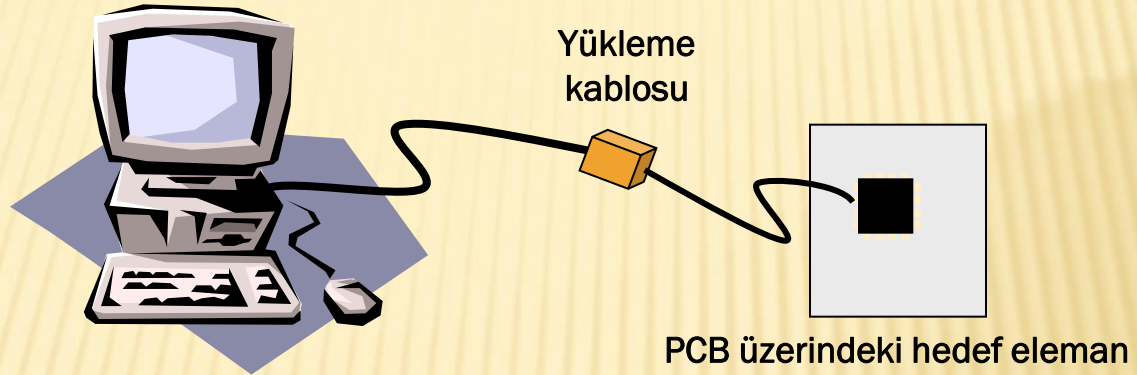


Yerleştirme ve Bağlama Aşaması

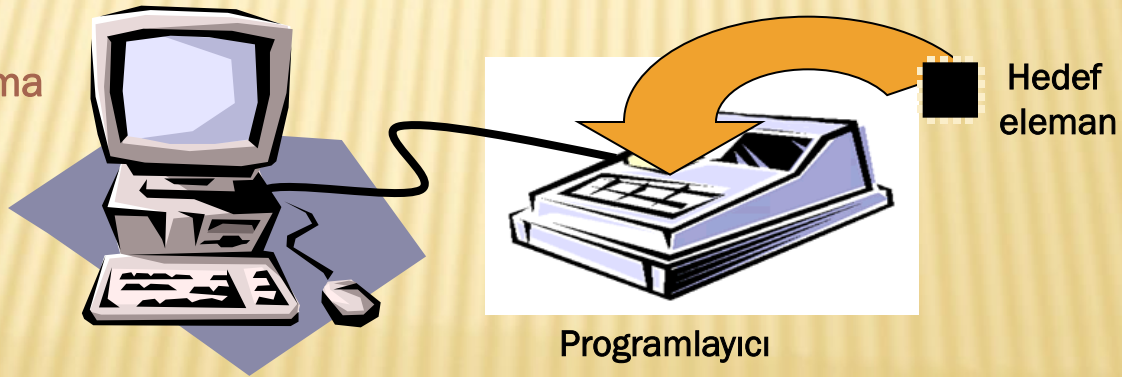


Yükleme / Programlama Aşaması

Yükleme



Programlama



Tasarım Akışı Özeti

