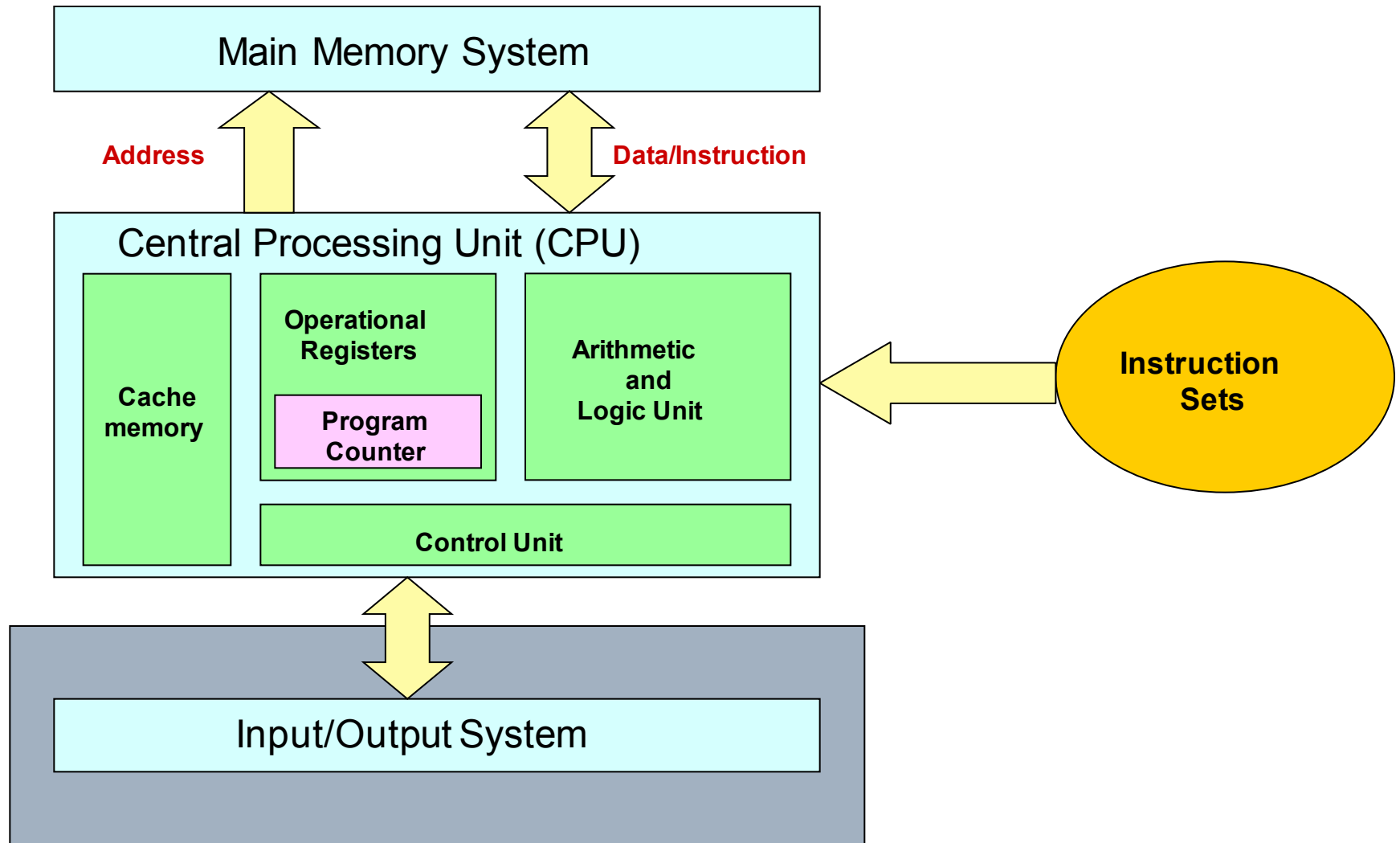


BİLGİSAYAR MİMARİSİ

GİRİŞ-ÇIKIŞ ORGANİZASYONU

Bölüm 8

- Çevresel Elemanlar
- Giriş-Çıkış Arabirimi
- Asenkron Veri Transferi
- Transfer Modları
- Öncelikli Kesme
- Doğrudan Bellek Erişimi
- Arabirim Devreleri
- Standart IO Arabirimleri
- Peripheral Component Interface Bus
- Universal Serial Bus USB



ÇEVRESEL ELEMANLAR

Giriş Elemanları

- Klavye
- Optik Giriş Elemanları
 - Kart Okuyucu
 - Kağıt Band Okuyucu
 - Barkod Okuyucu
 - Digitizer

A digitizer is any device which receives [analog](#) information (such as sound or light) and creates a [digital](#) representation of it (such as a [file](#) on a [computer](#)). This process is called [digitization](#).

• For example, a [digital camera](#) is a digitizer. Light enters the camera through the lens, and the [hardware](#) and [software](#) inside the camera converts that information to [binary](#) data, and stores it in an image file. The user may then transfer the file to a computer,

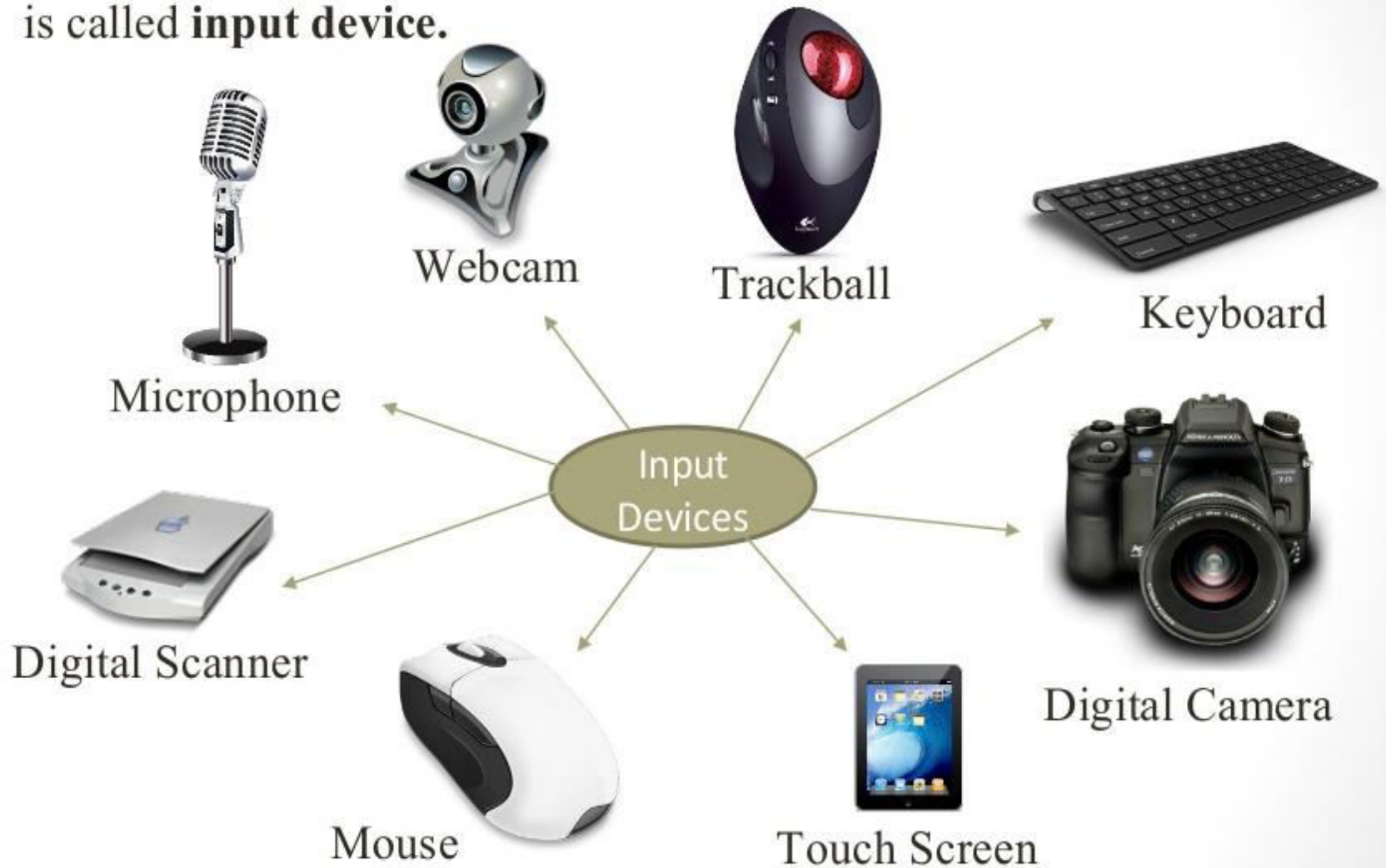
- Optik Okuyucu

Optik formlar üzerindeki kodlanan bilgilerin değişik tekniklerle algılanıp bilgisayar üzerine aktarılmasını gerçekleştiren cihazlardır

- Manyetik Giriş Elemanları
 - Manyetik Kart Okuyucu
- Ekran Giriş Elemanları
 - Dokunmatik ekran
 - Ekran kalem (ışığa duyarlı)
 - Fare
- Analog Giriş Elemanları

Giriş Elemanları

A hardware component used to enter data and instructions into computer is called **input device**.



ÇEVRESEL ELEMANLAR

Çıkış Elemanları

- Kard Delici, Kağıt Band Delici
- CRT
- LCD
- Yazıcı (Impact, Ink Jet, Laser, Dot Matrix)
- Plotter
- Analog
- Ses

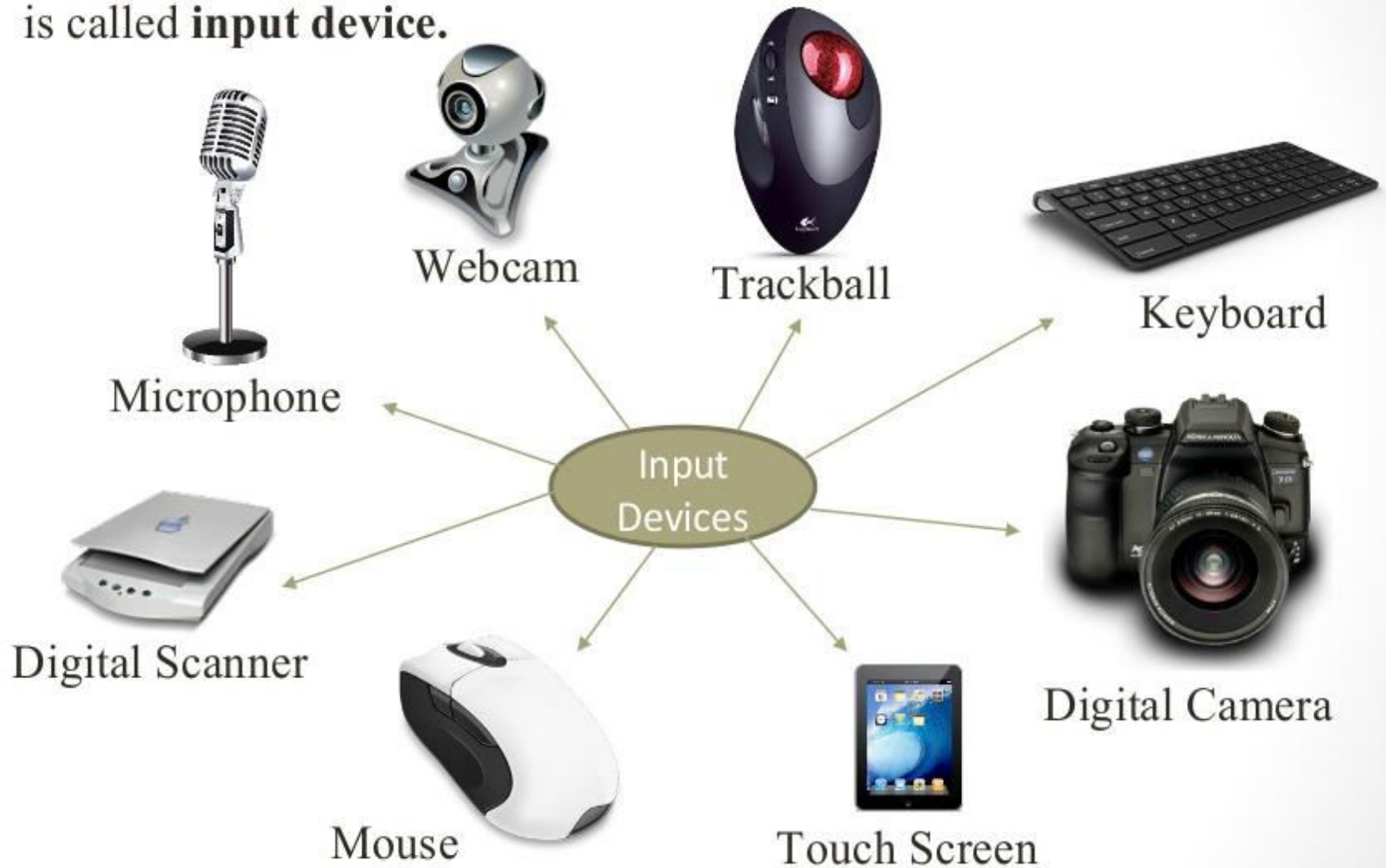
Çıkış Elemanları

A hardware component that are used to receive information from the computer are called **output devices**.



Giriş Elemanları

A hardware component used to enter data and instructions into computer is called **input device**.



Çıkış Elemanları

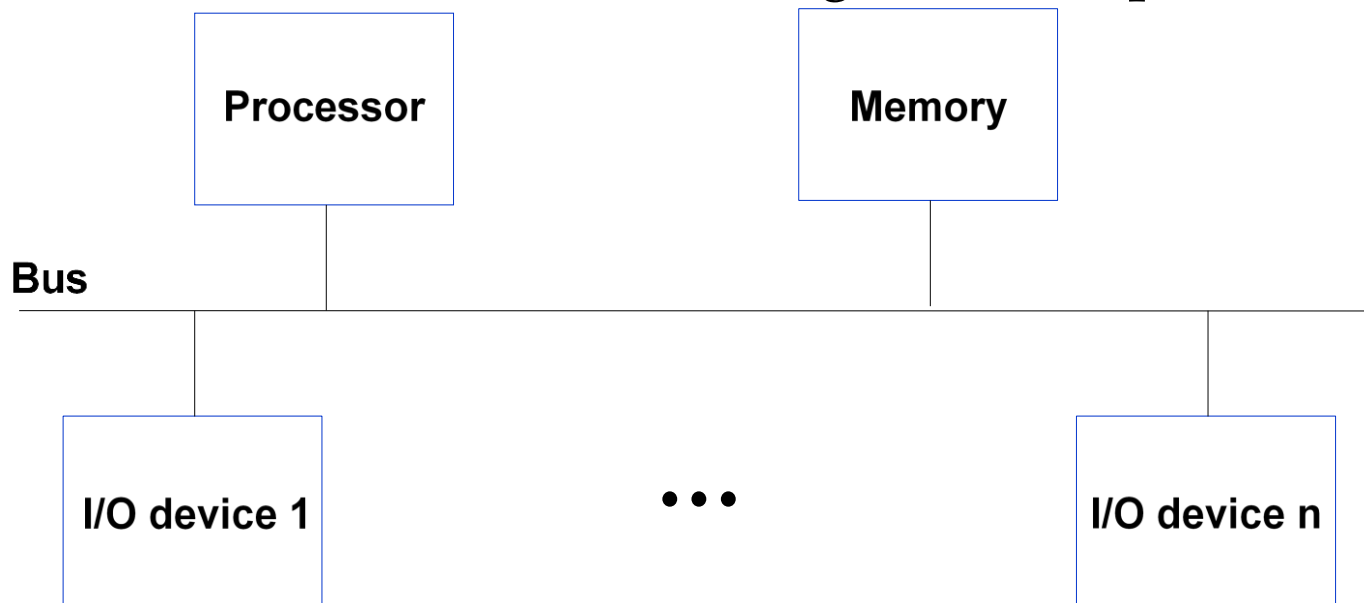
A hardware component that are used to receive information from the computer are called **output devices**.



IO Elemanlarına Erişim

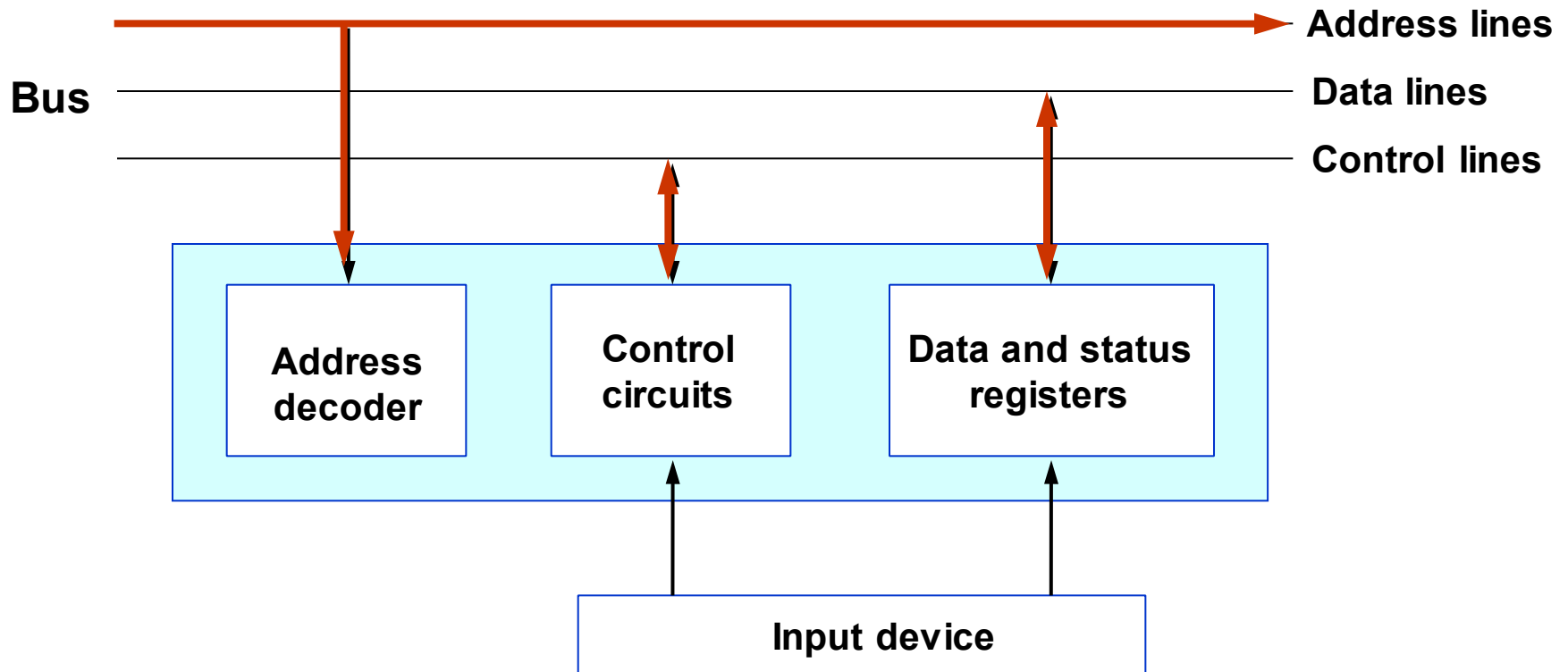
Single-bus structure

- ① The bus enables all the devices connected to it to exchange information
- ① Typically, the bus consists of three sets of lines used to carry address, data, and control signals
- ① Each I/O device is assigned a unique set of addresses



Giriş Elemanı İçin IO Arabirimi

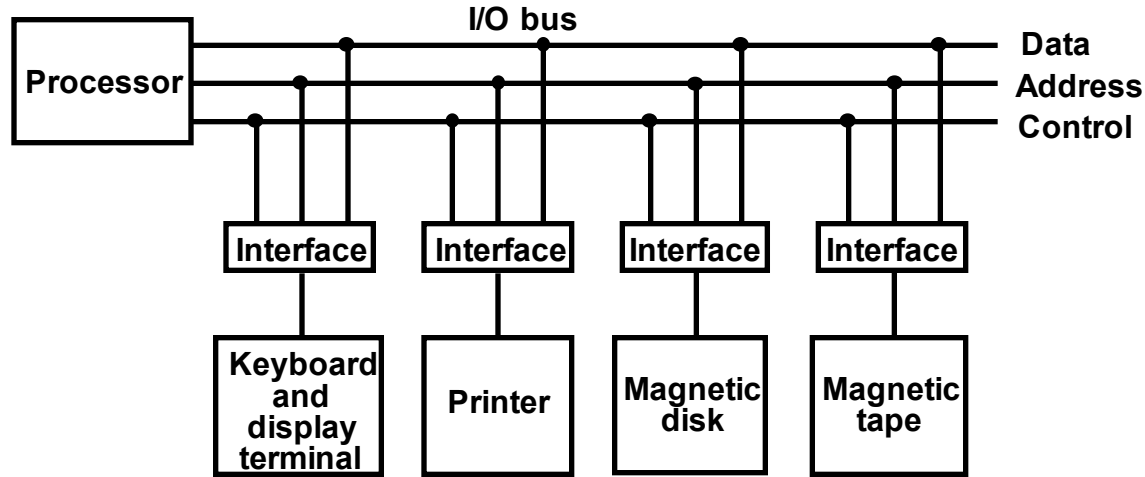
- The address decoder, the data and status registers, and the control circuitry required to coordinate I/O transfers constitute the device's *interface circuit*



GİRİŞ-ÇIKIŞ ARABİRİMİ

- Saklama üniteleri (Bellek veya CPU register) ve çevresel I/O elemanları arasında bilgi transfer metodu sağlar.
- Bilgisayar ve çevresel elemanlar arasındaki farklılıkları giderir.
 - Çevresel elemanlar- Elektromekanik Elemanlar
 - CPU veya Bellek - Elektronik Eleman
- Data Transfer Hızı
 - Çevresel Elemanlar – Genellikle daha yavaş
 - CPU veya Bellek – Genellikle çevresel elemanlardan daha hızlıdır.
 - Senkronizasyon mekanizmasının bazı türlerine ihtiyaç olabilir.
- Bilgi Birimi
 - Çevresel – Byte, Blok, ...
 - CPU veya Bellek – Kelime (Word=
- Bilgi simgeleme farklı olabilir.

I/O Ortak Yolu ve Arabirim Modülleri



Her çevresel eleman arabirimi ile ilişkili bir arabirim modülüne sahiptir.

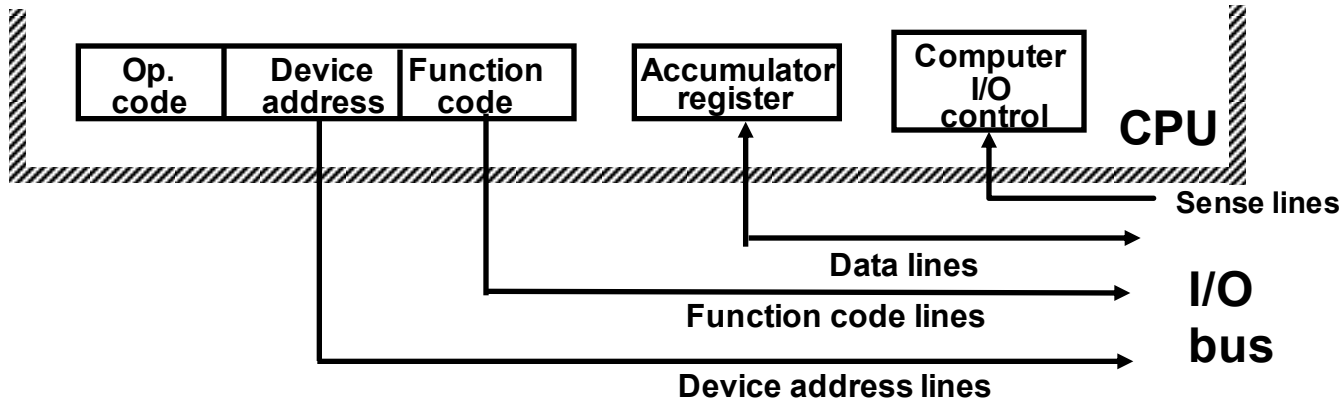
- Eleman adreslerinin kodçözümü (eleman kodu)
 - Komutların kodçözümü (işlem)
 - Çevresel kontrolör için gerekli işaretleri sağlar.
 - Data akışını senkronize eder ve yönlendirir.
- Çevresel elemanlar ve CPU veya Bellek arasında transfer hızını ayarlar.

Tipik I/O komutları

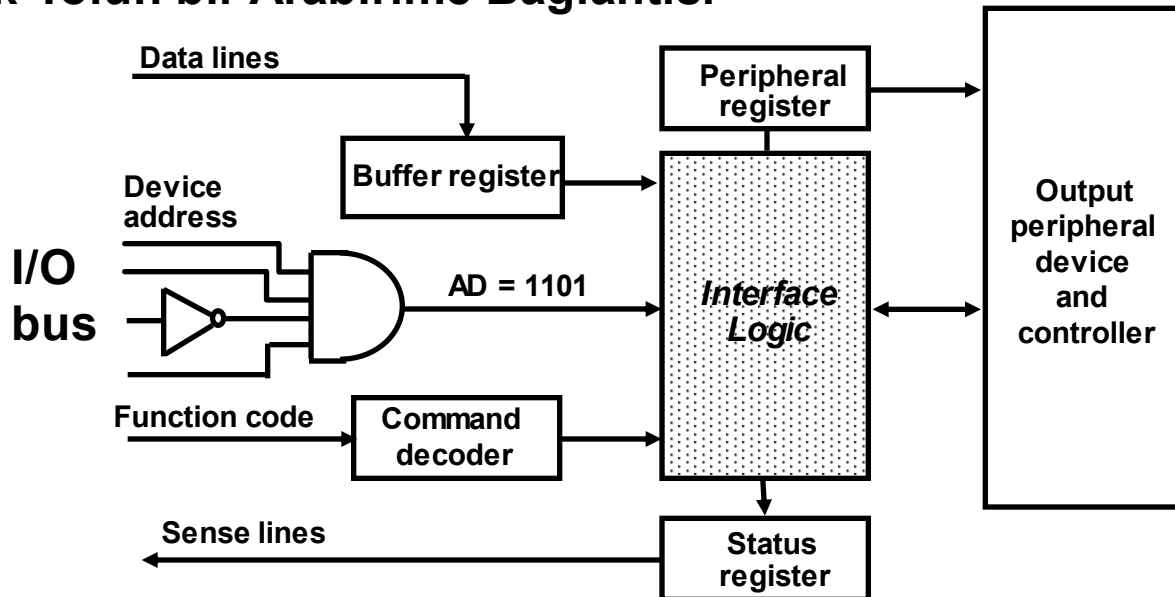
Opkodu	Eleman adresi	Fonksiyon Kodu
(Komut)		

I/O ORTAK YOLU BAĞLANTISI

I/O Ortak Yolun CPU bağlantısı



I/O Ortak Yolun bir Arabirime Bağlantısı



I/O ve BELLEK ORTAK YOLU

Ortak Yol Fonksiyonları

Bellek Ortak Yolu : CPU ve Bellek arasındaki veri transferi

I/O Ortak Yolu : I/O arabirim üzerinden I/O elemanları ile CPU arası veri transferi

Fiziksel Organizasyonlar:

Bir bilgisayar Sisteminde Bellek ve I/O ortak yolu 3 şekilde organize edilebilir:

1. Heriki fonksiyon için ayrık kontrol hatlı bir ortak yol kullanımı
2. Heriki fonksiyon için ortak kontrol hatlı bir ortak yol kullanımı
3. Bellek ve I/O için iki ayrık ortak yol kullanımı

ilki Bellekle iletişim , diğeri ise I/O arabirimi ile iletişim sağlar.

- CPU ve tüm arabirim üniteleri arası iletişim bir ortak I/O Bus ile sağlanır.
I/O Bus
- Bir çevresel elemana bağlı bir arabirim
data registers , *kontrol register*, ve bir *status register* içerebilir.

Bir komut, çevresel elemana en uygun arabirim register üzerinden gönderilebilir.

Fonksiyon kodu ve sense hatları gerekli değildir. (Data Transferi, kontrol, ve status bilgisi daima bir ortak I/O yolu üzerinden iletilir.)

İzole I/O ve Bellek Haritalı I/O

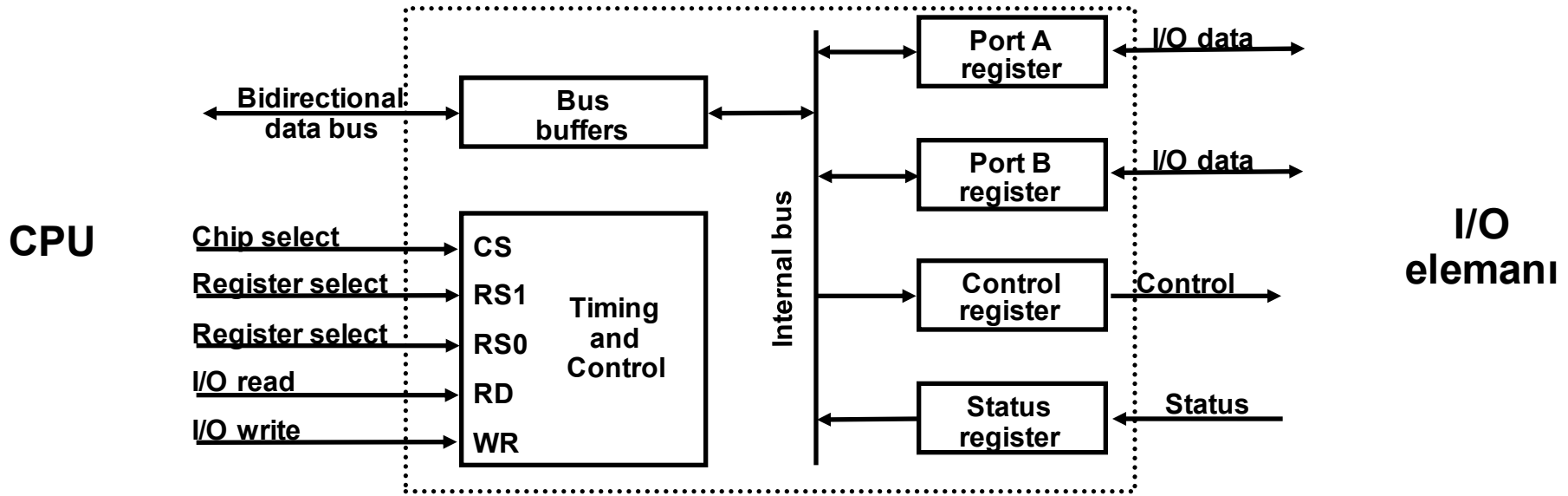
İzole I/O

- Bellek oku/yaz kontrol hatlarına ilaveten Ayrık I/O oku/yaz kontrol hatları
- Ayrık (izole) bellek ve I/O adres alanı
- Ayrık giriş ve çıkış komutları

Bellek Haritalı I/O

- Tek bir oku/yaz bellek kontrol hatları
(bellek ve I/O transfer arasında farklılık yok)
- Bellek ve I/O adresleri ortak adres alanını paylaşır.
 - > mevcut bellek adres bölgesini küçültür.
- Hiçbir belirli giriş veya çıkış komutu bulunmaz.
 - > Aynı bellek referanslı komutlar I/O transferleri için de kullanılır.
- I/O işlemlerinin yapılmasında esneklik sağlar.

I/O ARABİRİMİ



Programlanabilir Arabirim

CS	RS1	RS0	Register selected
0	x	x	None - data bus in high-impedance
1	0	0	Port A register
1	0	1	Port B register
1	1	0	Control register
1	1	1	Status register

- Herbir porttaki bilgi için I/O elemanının çalışma moduna bağlı bir anlam tahsis edilebilir

Port A = Data; Port B = Komut; Port C = Durumlar (Status)

- CPU herbir portu Kontrol register a bir byte transfer ederek yükler.
- CPU herbir portun çalışma modunu tanımlar.

→ **Programlanabilir Port:** Kontrol register bitlerini değiştirerek, arabirim karakteristiklerini değiştirmek mümkündür.

ASENKRON DATA TRANSFERİ

SENKRON VE ASENKRON İŞLEMLER

Senkron- Tüm elemanlar ortak bir saat üreticinden zamanlama bilgisini üretir.

Asenkron- Ortak saat işareti bulunmaz.

Asenkron Data Transferi

2 bağımsız birim arasındaki Asenkron data transferi, datanın iletildiği zamanı göstermek için iletişim halindeki birimlerin arasında kontrol işaretlerinin olmasını gerektirir.

2 Asenkron Data Transfer Metodu

Strobe darbesi :

- Bir strobe darbesi, transfer meydana gelmek zorunda olduğunda diğer bir birimi göstermek için bir birimin ürettiği işarettir.

El Sıkışma :

- Bir kontrol işareti datanın varlığını göstermek için iletilmekte olan herbir dataya eşlik etmesidir.
- Alıcı birim datayı kabul ettiğini onaylamak için diğer bir kontrol işareti gönderir.

STROBE KONTROL

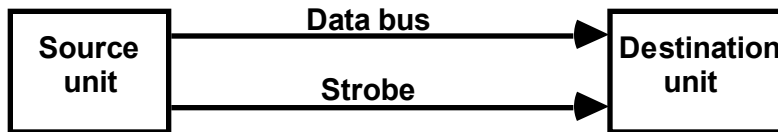
- * Herbir transferi zamanlamada, tek bir kontrol işareti kullanır.
- Strobe işaretini kaynak veya hedef ünite gönderebilir.

Data Transfer için

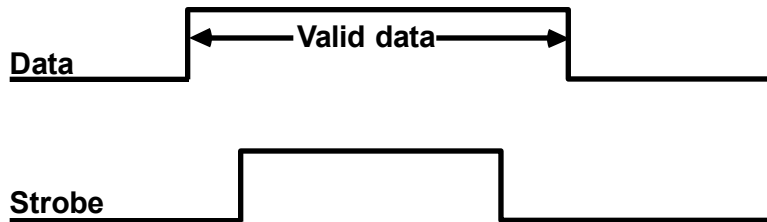
Kaynak Strobe işaretini gönderiyor

Hedefe data bus üzerindeki bilginin geçerli bilgi olduğunu bildirir. Yeterli bir süre aktif kalarak hedefe datayı alma olanağı tanır.

Blok Diyagram



Zamanlama Diyagramı

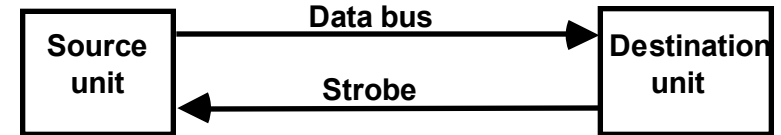


Data Transfer için

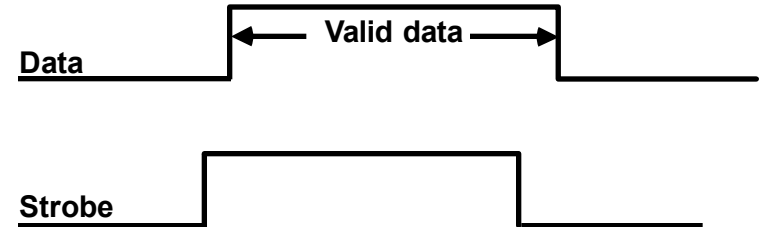
Hedef Strobe işaretini gönderiyor

Hedef, Kaynağa datayı göndermesi için strobe işaretini yollar. Yeterli bir süre data geçerli kalmalı ki hedef datayı kabul edebilsin.

Blok Diyagram



Zamanlama Diyagramı



EL SIKIŞMA

Strobe Metodları

Kaynak-Etkili iletişim

Transferi başlatan Kaynak Ünitesinin hedef ünitesinin gerçekten datayı alıp almadığından bilgisi yoktur.

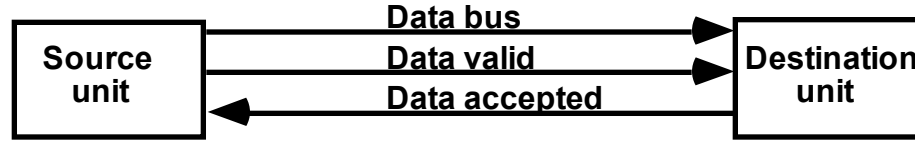
Hedef-Etkili iletişim

Transferi başlatan Hedef Ünitesinin kaynağın datayı bus üzerine yerleştirip yerleştirmedeği bilgisi yoktur.

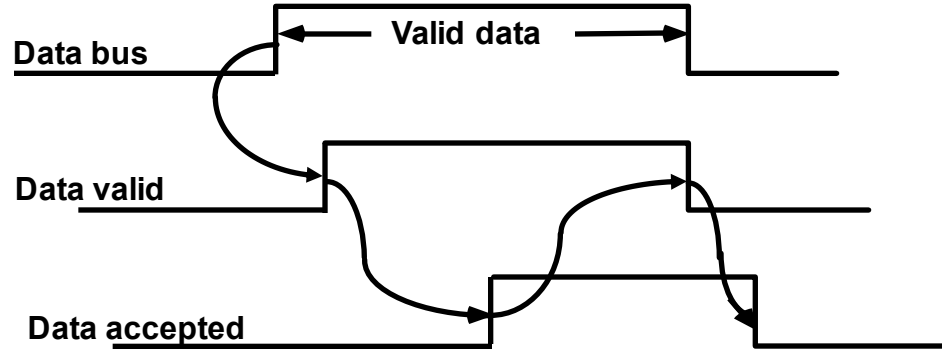
Bu problemi çözmek için, **ELSIKİŞMA Metodu** kullanılır.
Transferi başlatan birime bir cevap sağlamak için
2. bir kontrol işareti gönderilir.

ELSIKİŞMA KULLANAN KAYNAK ETKİLİ TRANSFER

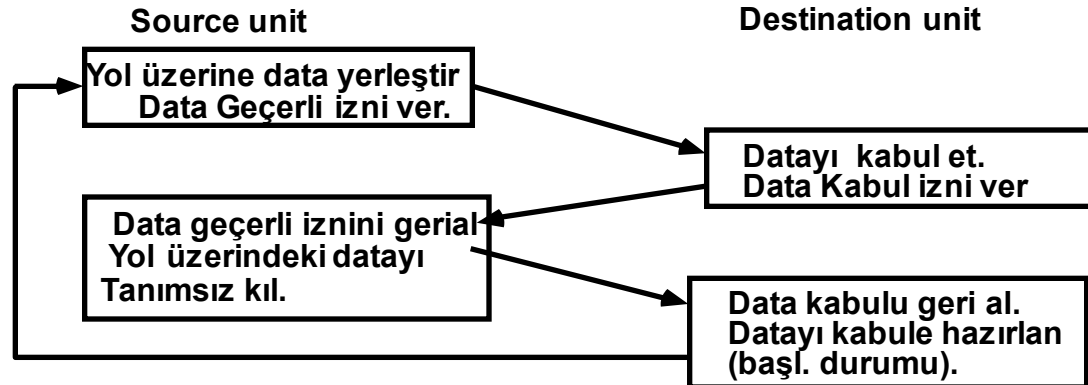
Blok Diyagram



Zamanlama Diyagramı



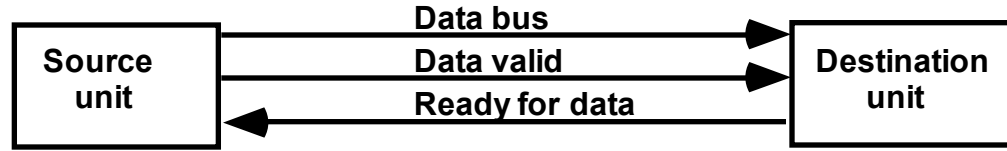
Olay Dizisi



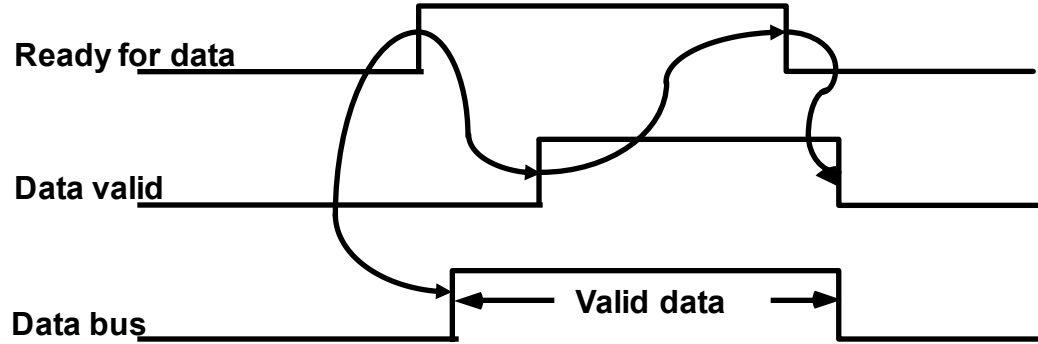
- * Bir durumdan diğerine keyfi gecikmelere esnektir
- * Kendi data transfer hızında cevap vermek için her üniteye izin ver.
- * Transfer hızı daha yavaş ünite tarafından belirlenir.

ELSIKİŞMA KULLANAN HEDEF ETKİLİ TRANSFER

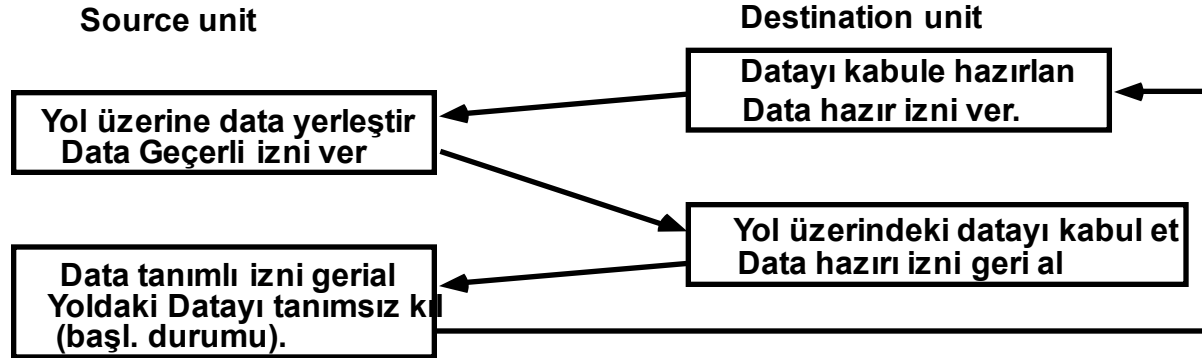
Blok Diyagram



Zamanlama Diyagramı



Olay Dizisi



- El sıkışma yüksek derecede bir esneklik ve güvenilirlik sağlar bir data transferinin başarılı tamamlanması heriki ünitenin de aktif katılımına bağlıdır.
- * Bir ünite hatalı ise, data transferi tamamlanamaz.
- Bir iç saat üretici ile el sıkışma kontrol işareti geldiğinde sayma yaptırılıp bir süre sonra dönüş kontrol işareti alınamazsa sayma durdurularak hata işareti üretilir.

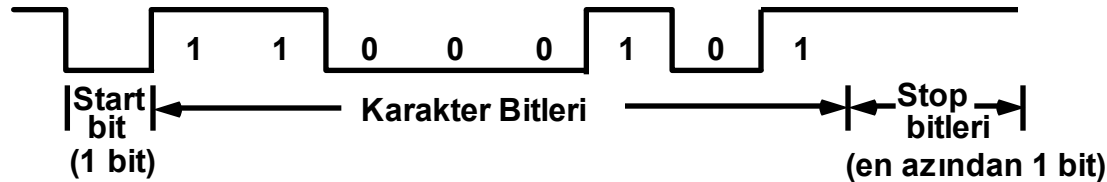
ASENKRON SERİ TRANSFER

Veri Transferi 4 şekilde yapılabilir:

Asenkron Seri Transfer
Senkron Seri Transfer
Asenkron Paralel Transfer
Senkron Paralel Transfer

Asenkron Seri Transfer

- Karakter kodlarının her iki ucuna yerleştirilen özel bitler kullanır.
- Her karakter 3 kısımdan oluşur.: Start bit; Data bitleri; Stop bitleri



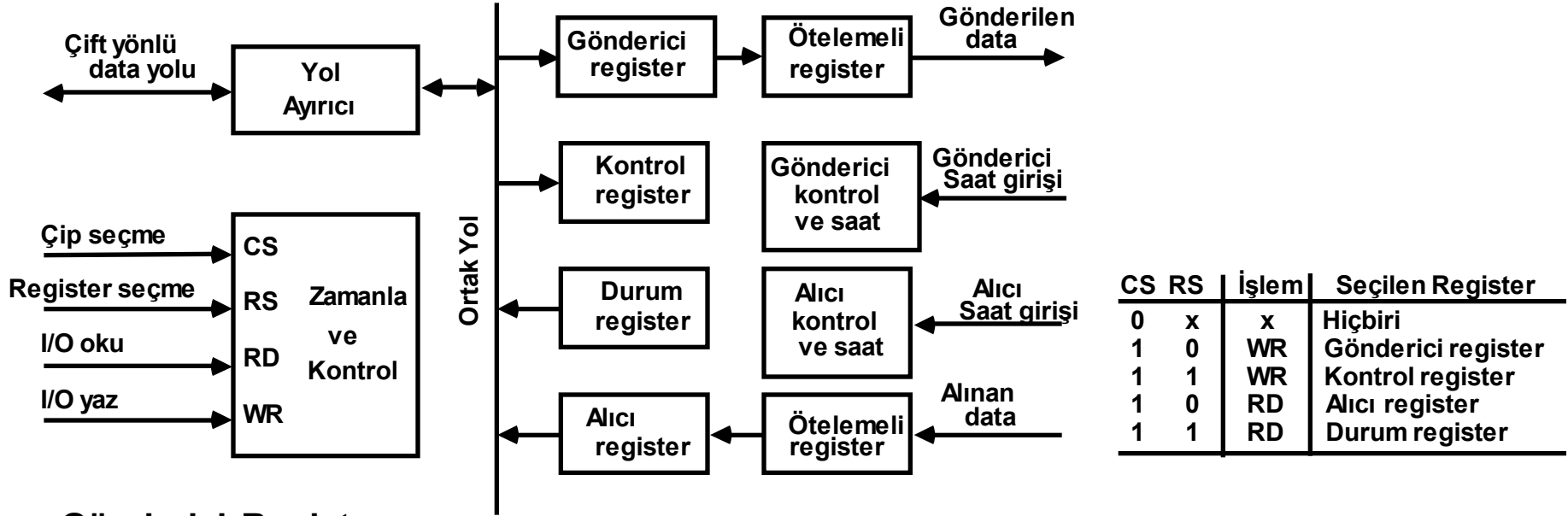
Bir Karakter alıcı tarafından 4 kural bilgisi ile belirlenir :

- Veri gönderilmediğinde, hat 1 durumunda tutulur (boştaki durum)
- Bir karakter iletiminin başlatıldığı *Start Bit* ile bilinir (daima 0 olan)
- Karakter bitleri daima *Start Biti* takip eder.
- Enson karakterden sonra, *Stop Biti* yakalanır, hat en azından 1 bit süresi için 1 durumuna geri döner.

Alıcı transfer hızını ve bilgi içeren bitlerin sayısını önceden bilir.

UNİVERSAL ASENKRON ALICI-VERİCİ - UART

Entegre Devre yapısındaki mevcut bir asenkron iletişim arabirimi



Gönderici Register

- Data yolu üzerinden (CPU dan) bir veri kabul eder.
- Seri iletişim için ötelemeli register üzerinden veri transferi yapar.

Alıcı

- Seri bilgiyi başka bir ötelemeli register içine kaydeder.
- Bütün data (byte) alıcı register a yollanır.

Durum Register Bitleri

- I/O bayrakları ve hatalar için kullanılır.

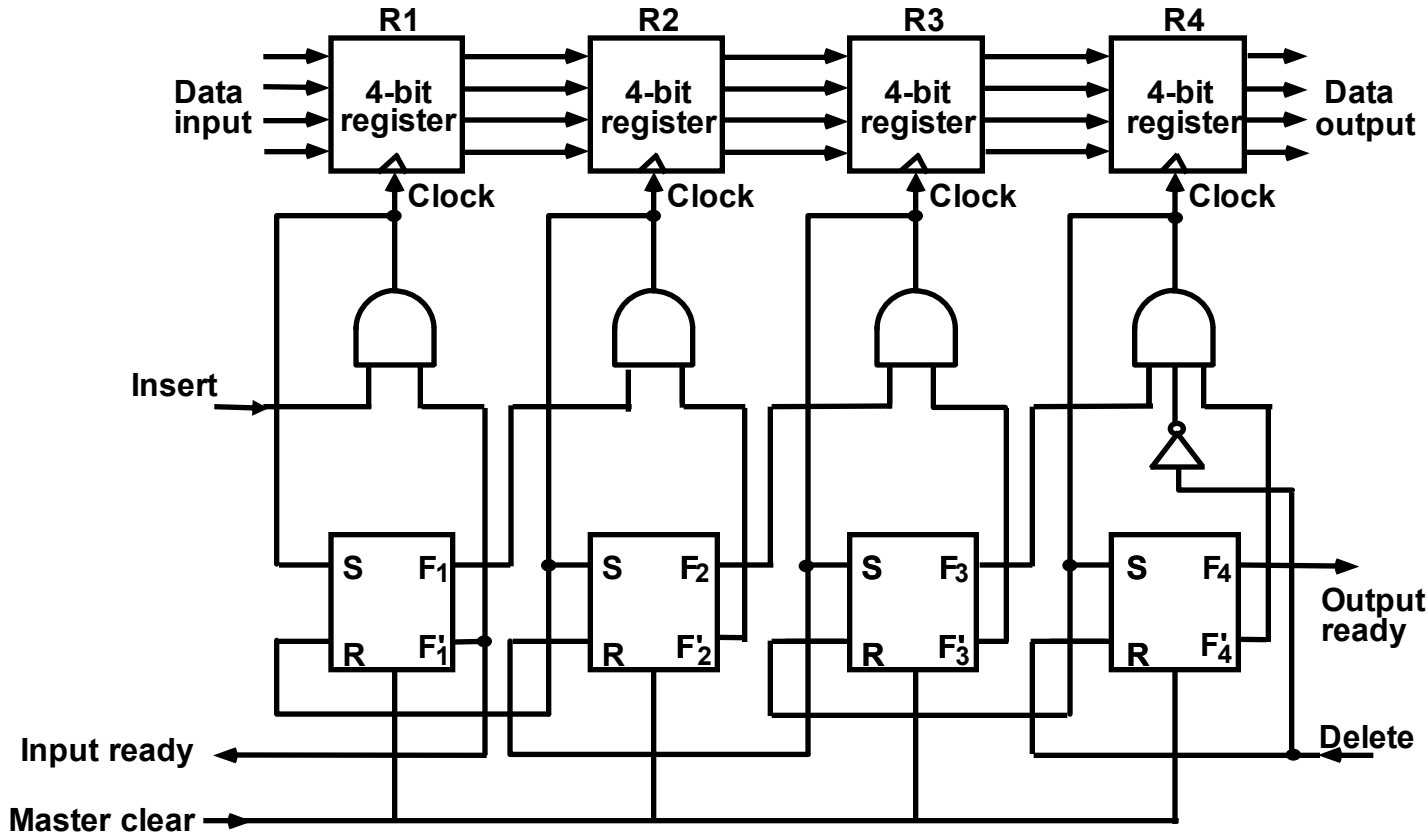
Kontrol Register Bitleri

- İletişim hızı (baud rate), karakterdeki bit sayısı, eşlik kontrolü ve üretimi olup olmadığı, ve stop bitlerin sayısı.

İLK GİREN İLK ÇIKAR (FIFO) TAMPON

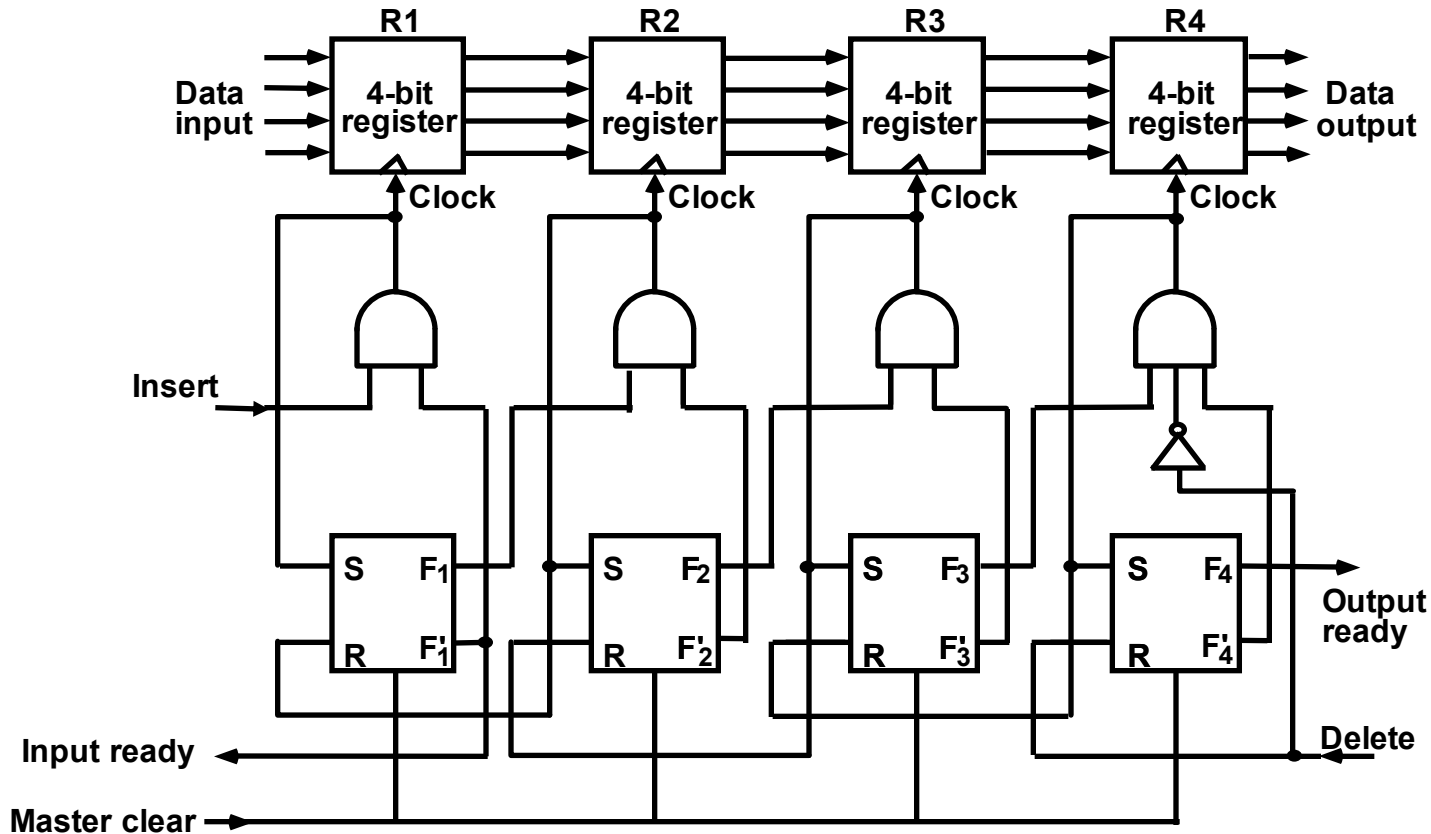
- * Data giriş ve çıkışı iki farklı hızda yapılabilir.
- * Data çıkışı daima tampona data giriş sırasına göre yapılır.
- Asenkron iletişim kullanan uygulamalarda yararlıdır.

Örnek Tasarım: 4 x 4 FIFO Tampon (4 adet 4-bit registers R_i),
4 Kontrol Register (flip-floplar F_i , herbiri R_i ile ilişkilidir)



İLK GİREN İLK ÇIKAR (FIFO) TAMPON

- Input Ready=1 ve Insert=1 ise $R1 \leftarrow \text{Data 1}$, $F1=1$ (Input ready=0)
 $F1=1$ ise $R2 \leftarrow \text{Data 1}$, $F2=1$, $F1=0$, Input Ready=1, $R1 \leftarrow \text{Data 2}$
- $F2=1$ ise $R3 \leftarrow \text{Data 1}$, $F3=1$, $F2=0$, $R2 \leftarrow \text{Data 2}$, $R1 \leftarrow \text{Data 3}$
- $F3=1$ ise $R4 \leftarrow \text{Data 1}$, $F4=1$, $F3=0$, $R3 \leftarrow \text{Data 2}$, $R2 \leftarrow \text{Data 3}$, $R1 \leftarrow \text{Data 4}$,
Output Ready=1



DATA TRANSFER MODLARI

CPU veya Bellek ile Çevresel Elemanların arasındaki

Data Transferi 3 farklı şekilde gerçekleştirilebilir:

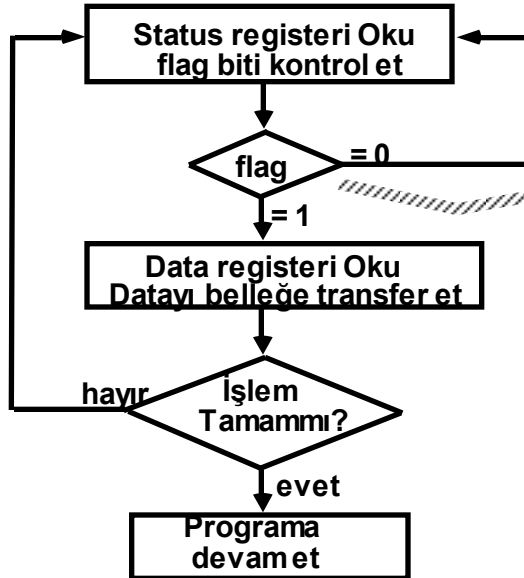
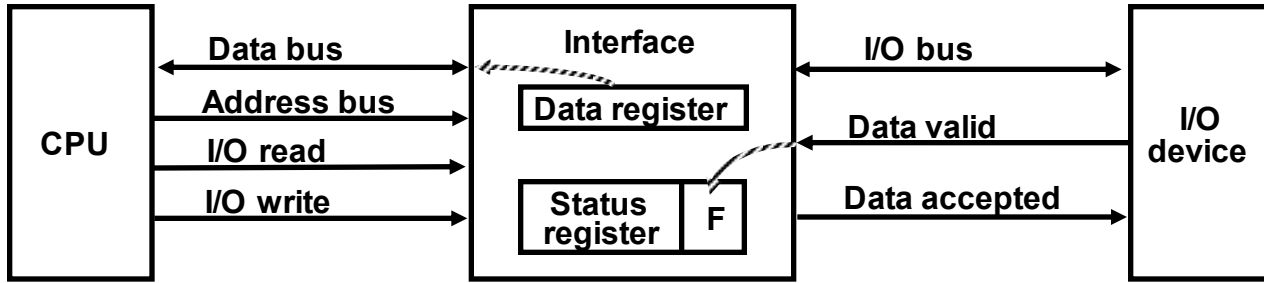
- 1. Yazılımsal Kontrollü I/O**
- 2. Kesme Kontrollü I/O**
- 3. Direk Bellek Erişimli (DMA) I/O**

PROGRAM-KONTROLLÜ I/O

I/O elemanı bellekle doğrudan erişimli değildir. CPU komutlarıyla I/O ile bellek arasında data transferi sağlanır. *Input* komutu data CPU register, *Store* komutu ile de belleğe gönderilir.

Giriş Elemanı ile CPU arasındaki Data Transferi şekilde görülmektedir:

I/O elemanı, Datayı I/O yoluna çıkarır ve Data valid işareti aktifleştirir. Arabirim ise datayı Data register a alır ve Data Accepted işareti gönderir, status register içinde bir biti set eder. CPU ile arabirim arasındaki iletişim aşağıdaki akış şemasından izlenebilir.



Status Register Kontrolü

- CPU hızını I/O eleman hızına göre yavaşlatır
- I/O elemanın iletim hızı 100 byte/sn
CPU komut çevrimi 1 mikrosaniye
Yani, 1 byte data transferi için
10000 status bit kontrolü gerekir,
- Basit tasarımlı
- Donanımı az

KESME ETKİLİ I/O

Kesme İsteği ile başlatılan I/O Data Transferi :

CPU yerine, I/O arabirimi I/O elemanının durumunu görüntüler.

- Arabirim I/O elemanın data transferi için hazır olduğu bilgisini alırsa, CPU ya giden bir kesme isteği üretir.**
- Bu isteğe bağlı olarak, CPU geçici olarak yaptığı işi keser , data transferini yürütmek için servis rutinine dallanır ve daha sonra kesme gelmeden önce yapmış olduğu göreve geri döner.**
- Kesme servis rutini başlangıç adresini seçmek için bir yol, vektörel kesme (bilgisayara bağlı bir çevresel elemandan gelen kesme istekleri- Örneğin, öncelikli kesme), diğeri vektörel olmayan kesme (kesme isteği bellekte sabit bir alana dallanma ile karşılanır.)**

ÖNCELİKLİ KESME

Öncelik

- 2 veya daha fazla kesme isteği olduğunda, ilk önce servis edilecek kesme isteğini belirler.
- Ayrıca, bir kesme isteği karşılanırken, bilgisayar kesme işlemini sağlayacak şartları da belirler.
- Düşük öncelikli kesme istekleri yürütülürken, yüksek öncelikli kesme isteklerine cevap verebilir.

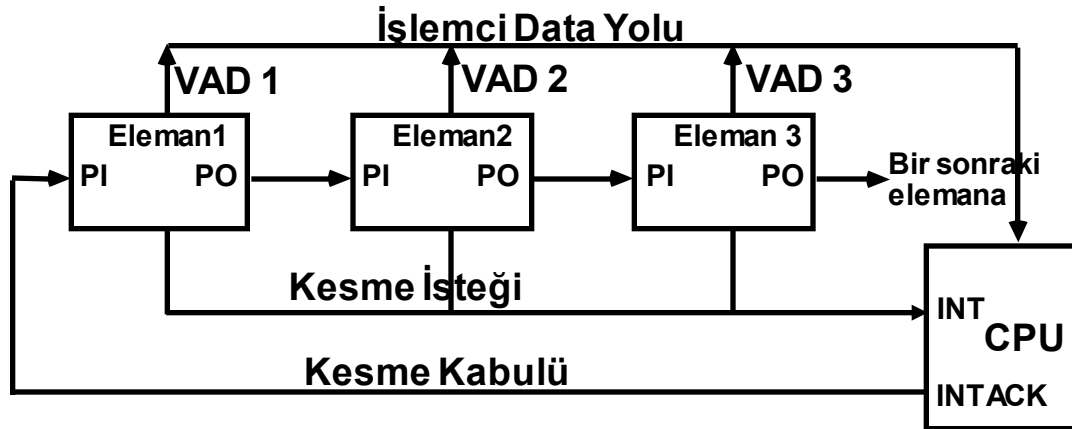
Yazılımsal Öncelikli Kesme

- Öncelik, çevresel elemanlardan oluşan kesme kaynaklarının test sırasıyla belirlenir.
- Yazılımsal olduğu için esnektir.
- Donanımsal yapısı küçük olduğundan düşük maliyetli
- Çok yavaş

Donanımsal Öncelikli Kesme

- Yüksek öncelikli istekleri belirlemek için, bütün kesme kaynaklarını kabul eden bir öncelikli kesme yöneticisi gerektirir.
- Donanımsal öncelik tanıma işlemi hızlıdır.
- Her kesme kaynağına doğrudan ulaşmak için kendine has kesme vektörüne sahip olduğundan hızlıdır.

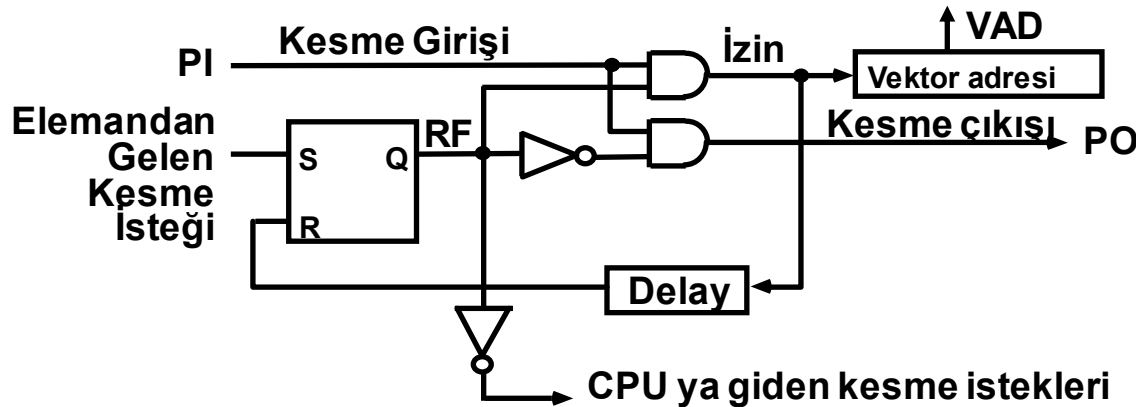
ÖNCELİKLİ SERİ KESME - DAISY-CHAIN -



- Seri donanımsal kesme fonksiyonu
- Kesme İstek hattı
 - Tek bir ortak hat
- * Kesme Kabul hattı

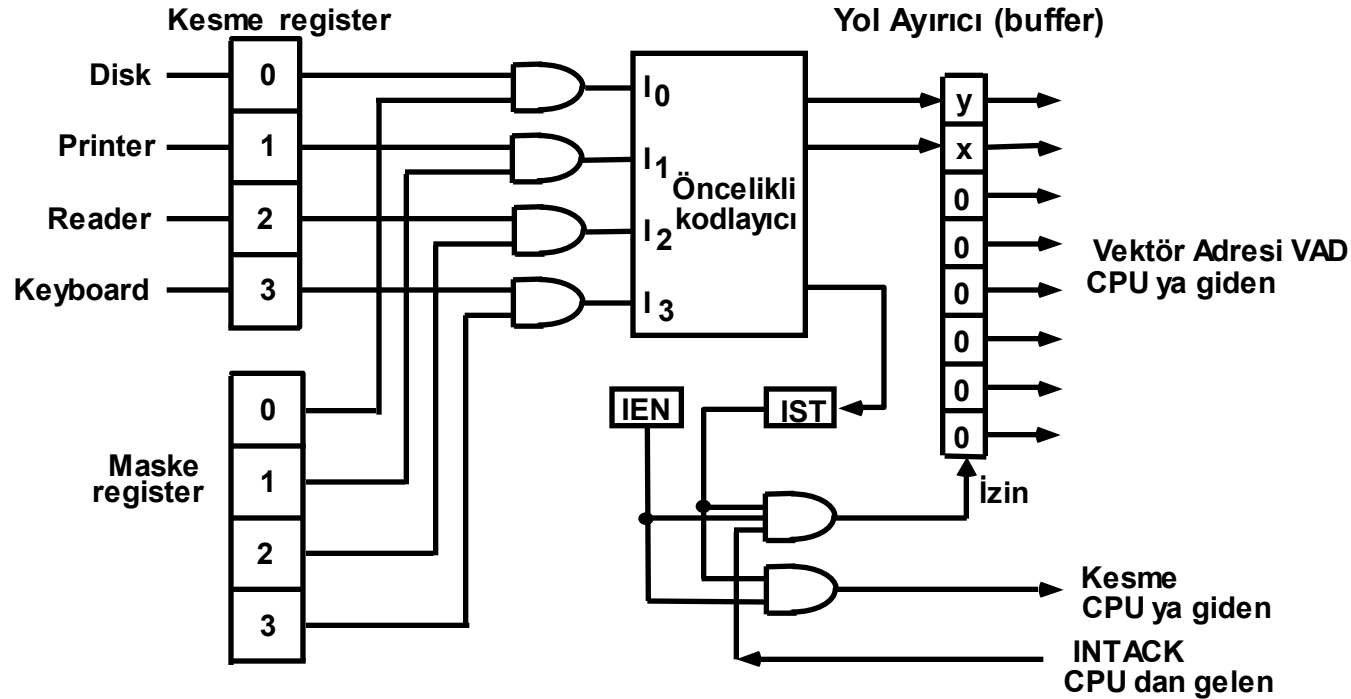
Herhangi bir elemandan Kesme İsteği gelirse, (INT=1) CPU , INTACK işareti göndererek kesmeyi kabul eder (INTACK=1) Elemanın PI girişi 1 olduğunda Data yoluna VAD adresini çıkarır. Kesme isteyen elemanlardan CPU ya en yakın olanın isteği karşılanır Ve o eleman sağındaki elemanın PI girişini bloke eder. (PI=0)

Öncelikli Kesme Lojiği



PI	RF	PO	İzin
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

ÖNCELİKLİ PARALEL KESME



IEN: ION veya IOF komutları ile 1 veya 0 değeri yüklenir.

IST: Maskelenemeyen kesme olduğunu gösterir.

INTACK : Kesme lojigi tarafından üretilen VAD adresini yüklemek için Üç Durumlu Yol Ayırıcısına izin verir.

Kesme Register:

- Herbir biti farklı bir kesme kaynağından gelen kesme istekleri ile ilişkilidir. Kaynaklar farklı kesme önceliklerine sahip olabilir.
- Herbir bit program komutları ile silinabilir.

Maske Register:

- Maske Register bir Kesme Register ile ilişkilidir.
- Herbir bite bir komutla 1 veya 0 değer verilebilir.

ÖNCELİKLİ KESME KODLAYICI

Bir kesmeden daha fazla kesme geldiğinde, en yüksek öncelikli kesmeyi belirler.

Öncelikli Kesme Kodlayıcı Doğruluk Tablosu

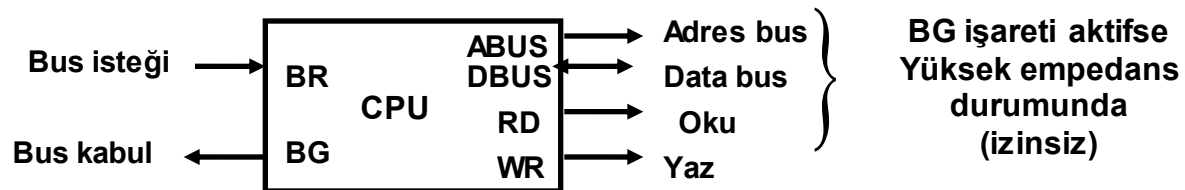
Inputs				Outputs			Boolean functions
I_0	I_1	I_2	I_3	x	y	IST	
1	d	d	d	0	0	1	$x = I_0' \cdot I_1'$ $y = I_0' \cdot I_1 + I_0' \cdot I_2'$ $(IST) = I_0 + I_1 + I_2 + I_3$
0	1	d	d	0	1	1	
0	0	1	d	1	0	1	
0	0	0	1	1	1	1	
0	0	0	0	d	d	0	

DOĞRUDAN BELLEK ERİŞİMLİ I/O

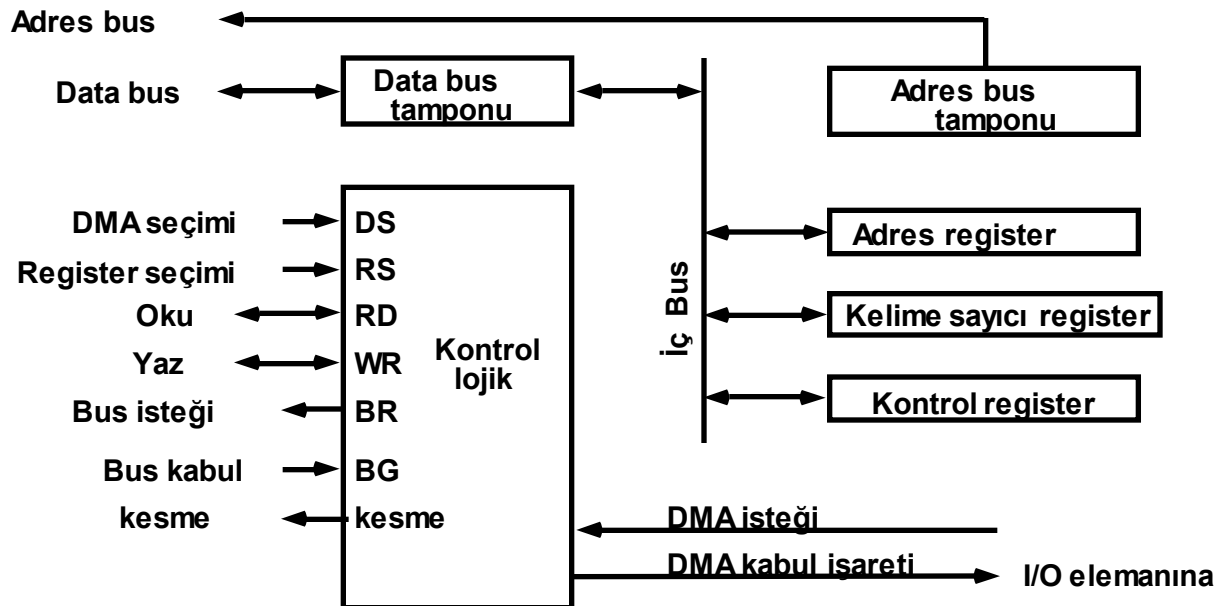
- Yüksek hızda büyük data bloklarının transferi için kullanılan bir yöntemdir. Büyük data blokları hard disk, manyetik disk gibi elemanlarda tutulur.
- CPU üzerinden transfer sağlamadan I/O elemanları ile bellek arası data transferi doğrudan erişimle daha hızlıdır.
- DMA kontrolör devresi bu iletişimde bir arabirim görevi yapar.
- CPU, transfer edilecek kelime sayısı ve bellek adreslerini göndererek DMA kontrolör devresini çalıştırır.
- Diğer görevler için CPU serbest kalır.

DOĞRUDAN BELLEK ERİŞİMİ

CPU bus signals for DMA transfer



Block diagram of DMA controller



DMA I/O İŞLEMİ

I/O işlemini başlatmak için:

- CPU aşağıdaki komutları icra etmesi gerekir.
Bellek Adres Registere adres yükle
Kelime sayıcıya data adedini yükle
Oku veya Yaz Fonksiyonu gerçekleştir
DMA transferi için bir kontrol komutu

Bir kontrol komutu alındığında, DMA CPU dan bağımsız olarak aşağıdaki gibi I/O işlemini yerine getirir.

Giriş İşlemi

- [1] Giriş Elemanı $e \leftarrow RD$ Oku kontrol işareti
- [2] Buffer(DMA kontrol devresi) \leftarrow Giriş Byte datası;

assembles the byte into a word until word is full
- [4] $M \leftarrow$ bellek adresleri, W (Yaz kontrol işareti)
- [5] Adres Reg \leftarrow Adres Reg + 1; WC(Kelime Sayıcı) \leftarrow WC - 1
- [6] Eğer WC = 0 ise Interrupt to acknowledge done, aksi halde [1]e git

Çıkış İşlemi

- [1] $M \leftarrow$ M Adres, R
 $M \text{ Adres } R \leftarrow M \text{ Adres } R + 1, WC \leftarrow WC - 1$
- [2] Disassemble the word
- [3] Buffer \leftarrow One byte; Çıkış Elemanı \leftarrow W, for all disassembled bytes
- [4] If WC = 0, then Interrupt to acknowledge done, else go to [1]

ÇEVİRİM ÇALMA

DMA I/O işlemi meydana gelirken, CPU ayrıca komutları icra eder.

DMA Kontrol devresi ve CPU herikiside belleğe erişir bu durumda Bellek Erişimde bir zıtlık oluşabilir.

Bellek Yolu Kontrolü

- Belleğe erişim isteyen tüm elemanların aktivitelerini koordine eder.**
- Öncelik Sistemi**

Çevrim Çalma (Cycle stealing):

- CPU genellikle I/O (DMA) işleminden daha hızlıdır.
CPU bellek çevrimlerinin çoğunu kullanır.**
- DMA Kontrolü, CPU nun kullandığı bellek çevrimlerini kontrolünü ele alır.**
- Bu çevrimlerde, CPU etkisiz kalır (hiçbir işlem yapmaz)**
- Yavaş CPU için, DMA Kontrolcü bellek çevrimlerinin çoğunu kontrol eder.**

DMA TRANSFER

