Bilgisayar Mimarileri
Giriş - Sayısal Tasarım
Yazaç Aktarım Dili (Register Transfer Language (RTL))

Kevser Nur Çoğalmış

Bilgisayar Mühendisliği Bölümü

2025

RECEP TAYYİP ERDOĞAN ÜNİVERSİTESİ

Kaydediciler/Yazaçlar (Registers)

- Bir kaydedici, bir bitlik bilgiyi saklama kapasitesine sahip flip floplardan oluşan devredir.
- n-bit kaydedici n adet flip flop ile oluşturulur ve n bit boyutunda ikili bir veriyi saklama kapasitesine sahiptir.
- Kaydediciler flip floplara ek olarak denetleme yapmak üzere mantık kapıları da içerebilirler.

Kaydediciler/Yazaçlar (Registers)

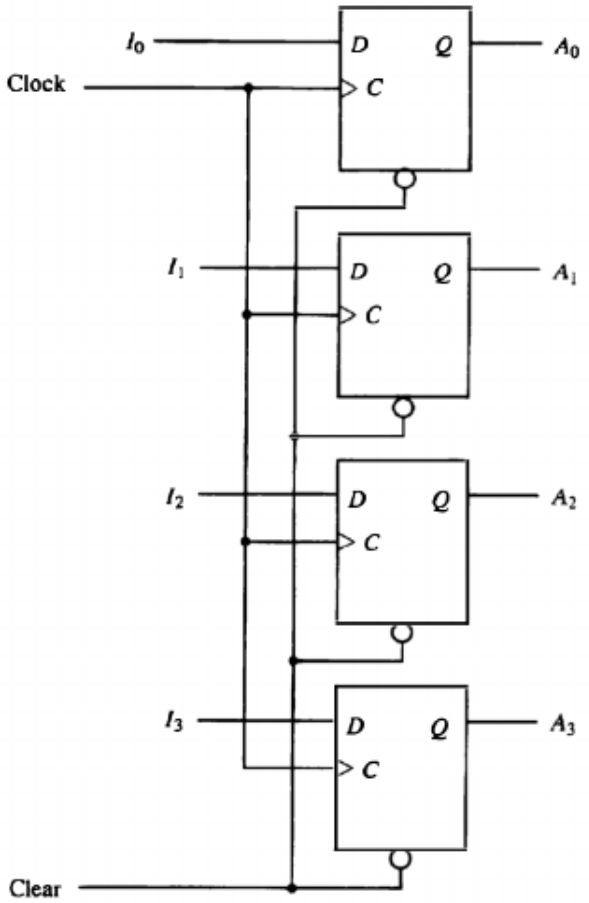


Figure 2-6 4-bit register.

Kaydediciler/Yazaçlar (Registers)

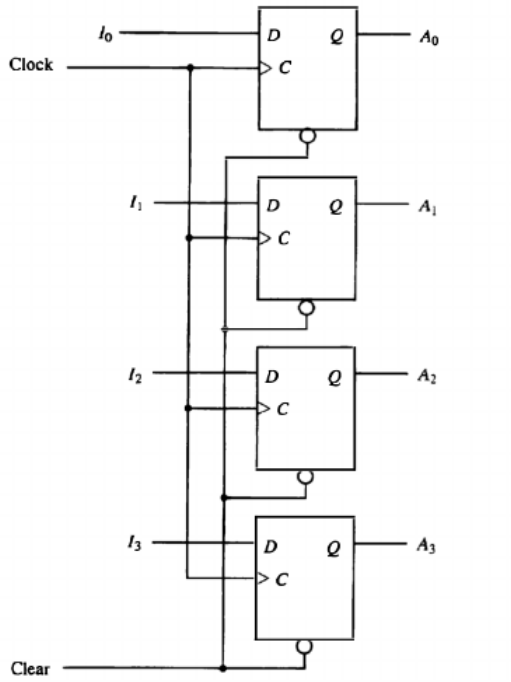


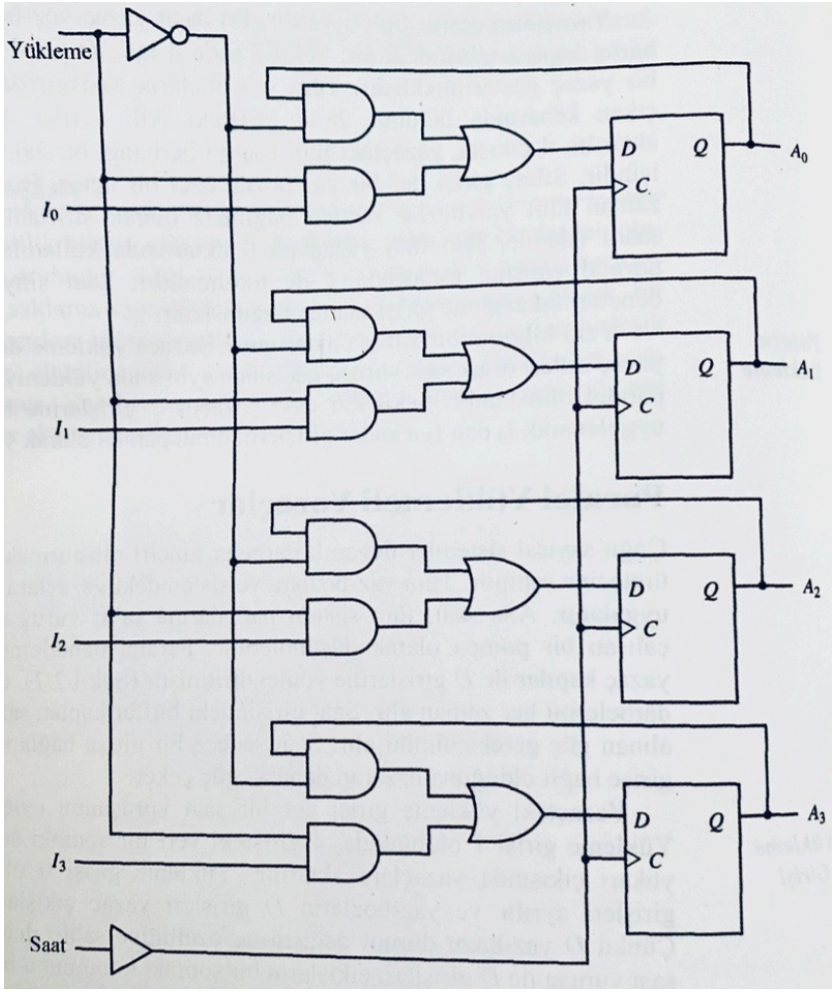
Figure 2-6 4-bit register.

- Bir veriyi kaydediciye aktarma/yazma işlemine yükleme/load denilir.
- Bu register görüldüğü üzere paralel load (in/out) yapmaktadır.

Kaydediciler/Yazaçlar (Registers)

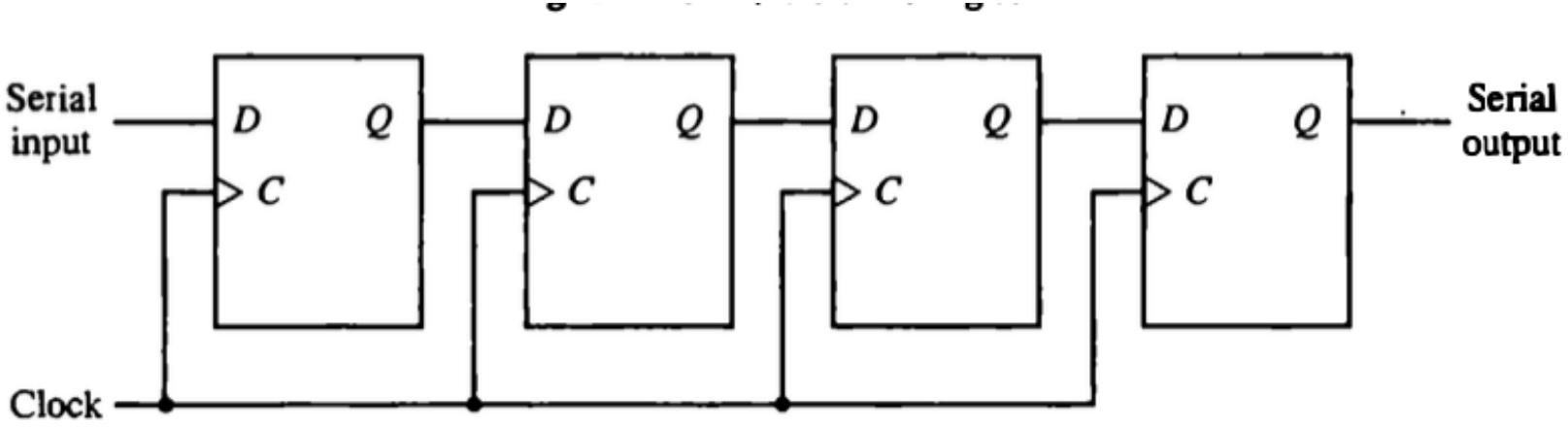
- Clock sinyali üretimi ana saat ile oluşur.
- Ana saat tüm sistemler içindir.
- Saat darbesi sürekli vardır.

Kaydediciler/Yazaçlar (Registers)



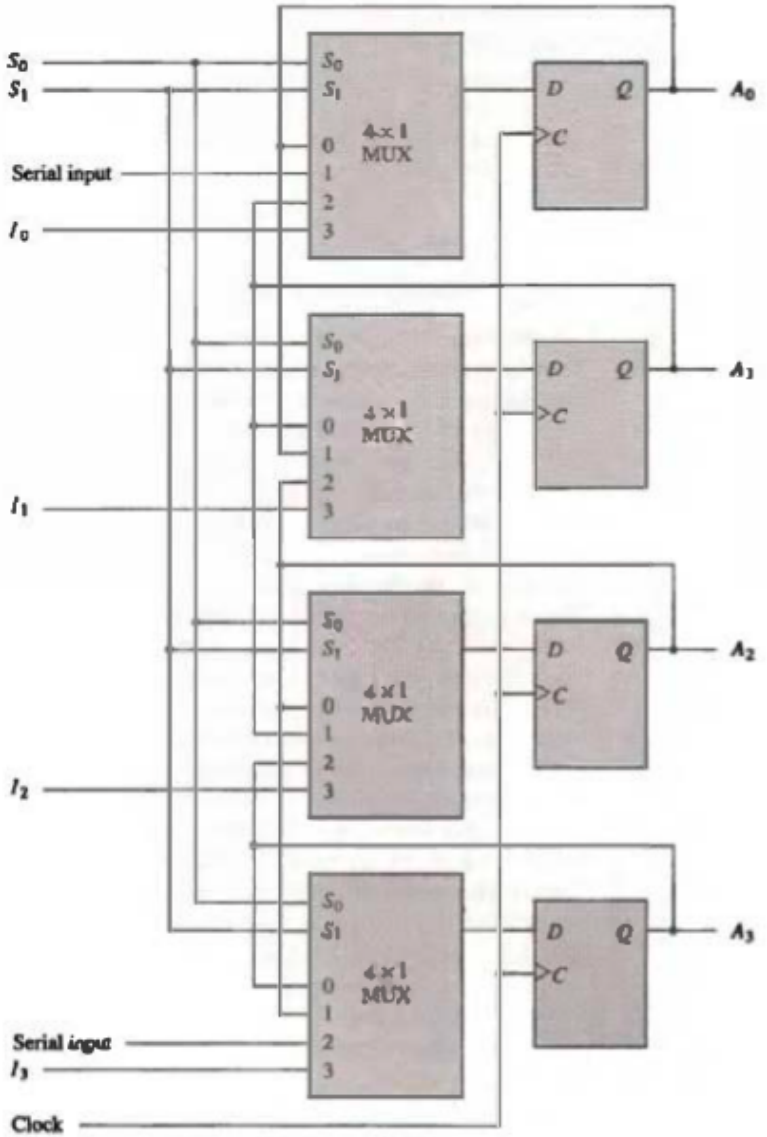
Kaydırma Yazaçları (Shift Registers)

- Girilen veriyi sağa ya da sola kaydıran devrelerdir. Bir flip flopun çıkışı diğerinin girişine bağlıdır.

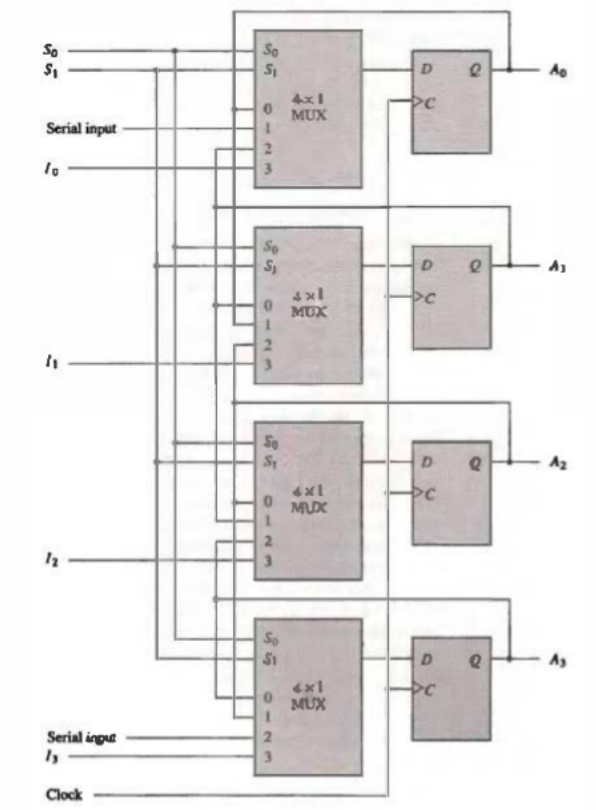


Paralel yüklemeli çift yönlü kaydırma yazaçları

- Her iki yöne de veriyi kaydırma özelliği olan yazaçlardır. Sahip olması gereken özellikler;
  - Tüm işlemlerde uyum sağlamak için ortak bir saat girişi,
  - Sağa kaydırma işlemi için sağ tarafta bağlı bir seri veri girişi,
  - Sola kaydırma işlemi için sol tarafta bağlı bir seri veri girişi,
  - Paralel yükleme işlemi için n adet veri giriş hattı (tipe göre değişebilir.),
  - Paralel veri çıkışı için n adet veri çıkış hattı (tipe göre değişebilir.),
  - Devamlı saat darbesi uygulandığında yazacın içeriğinin değişip değişmediğini denetleyen bir durum kontrolü.

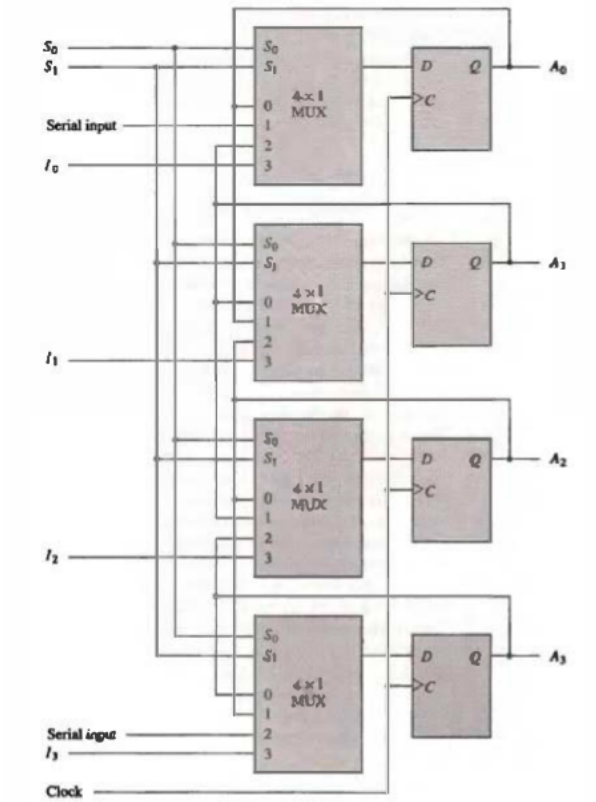

S_0
S_1
Serial input
I_0
I_1
I_2
Serial input
I_3
Clock
4×1 MUX
D
Q
C
A_0
A_1
A_2
A_3

Paralel yüklemeli çift yönlü kaydırma yazaçlar



S_1	S_0	G
0	0	0
0	1	1
1	0	2
1	1	3

Paralel yüklemeli çift yönlü kaydırma yazaçlar



S_1	S_0	Register operation
0	0	No change
0	1	Shift right (down)
1	0	Shift left (up)
1	1	Parallel load

Kaydırma Yazaçları (Shift Registers)

- Kaydırma yazaçları sıklıkla sayısal sistemleri birbirine bağlamak için kullanılır.
- Örneğin; Diyelim ki n bitlik bir bilgiyi saklamak istiyoruz.
  - Eğer ki veri kaynağı ile bilginin saklanacağı yer arasındaki mesafe uzaksa n adet sayısal veri hattı kullanmak oldukça pahalıdır.
  - Bu nedenle tek bir veri hattı kullanılarak veri seri olarak aktarılır ve daha ekonomik bir yol kullanılmış olur.
  - Göndericiler n bitlik veriyi paralel olarak flip floplara yükler, seri çıkış hattından veriyi iletirler.
  - Alıcılar ise veriyi seri olarak kaydırma yazacı içerisine alırlar.
  - n sayıda veri geldiğinde (n bit) bilgiler toplu olarak flip flopların çıkışından paralel olarak aynı anda okunur.
  - Yani, veriyi gönderen kaynak paralelden seriye, alıcı kaynak ise seriden paralele çevirim yapar.

Sayıcılar (Counters)

- Giriş vuruşlarının uygulanmasıyla önceden belirlenmiş değerler alan yazaçlara sayıcı denir.
- Giriş vuruşları saat darbeleri olabileceği gibi başka girişler de olabilir.
- Sayıcılar sayısal mantık içeren neredeyse tüm cihazlarda bulunur.
- Bir olayın kaç defa tekrar ettiğini saymakta kullanılır, zaman sinyali üretirler.
- Böylece aslında işlemlerin sırasını denetlerler.

Sayıcılar (Counters)

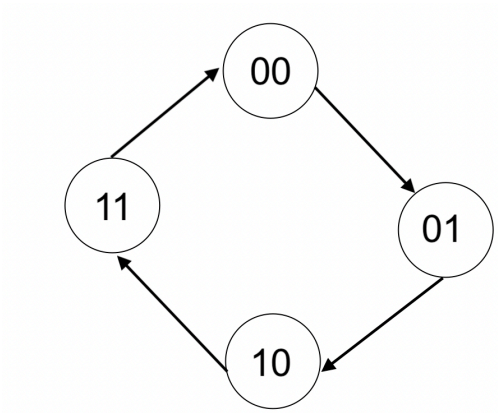
- n bit bir ikili sayıcı n bitlik bir yazaçtır.
- n adet flip floptan oluşur, binary sayım yapar.
- n bit sayıcı 0'dan 2^n - 1'e kadar sayar.
- Düzenli sayma işleminde genellikle herhangi bir bitten daha düşük mertebeli bitler tüm durumları gerçeklediğinde tümleyeni alınır.

Sayıcılar (Counters)

0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Sayıcılar (Counters)

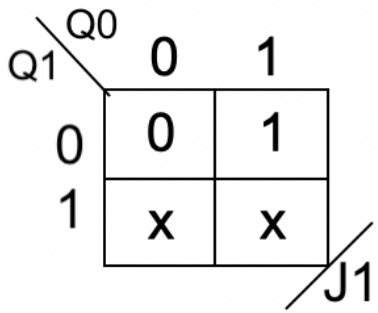
- 00,01,10,11 şeklinde sayan ikili sayıcı nasıl tasarlanır?

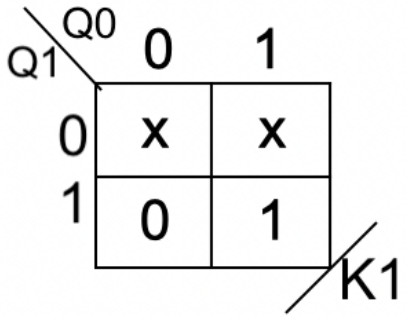


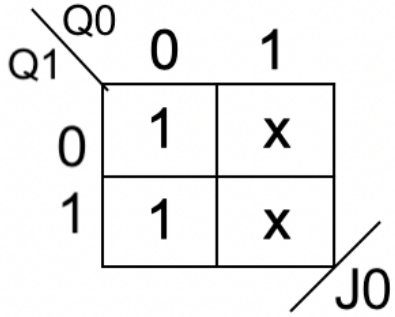
Transition Table / Excitation Table

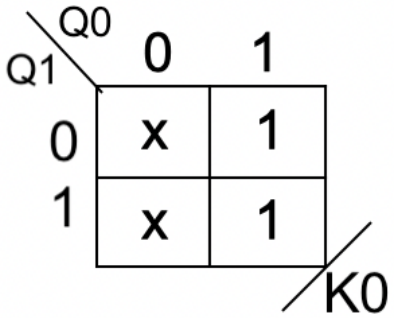
şimdiki durum		sonraki durum		FF inputs			
Q1	**Q0**	**Q1**	**Q0**	**J1**	**K1**	**J0**	**K0**
0	0	0	1	0	x	1	x
0	1	1	0	1	x	x	1
1	0	1	1	x	0	1	x
1	1	0	0	x	1	x	1

Sayıcılar (Counters)



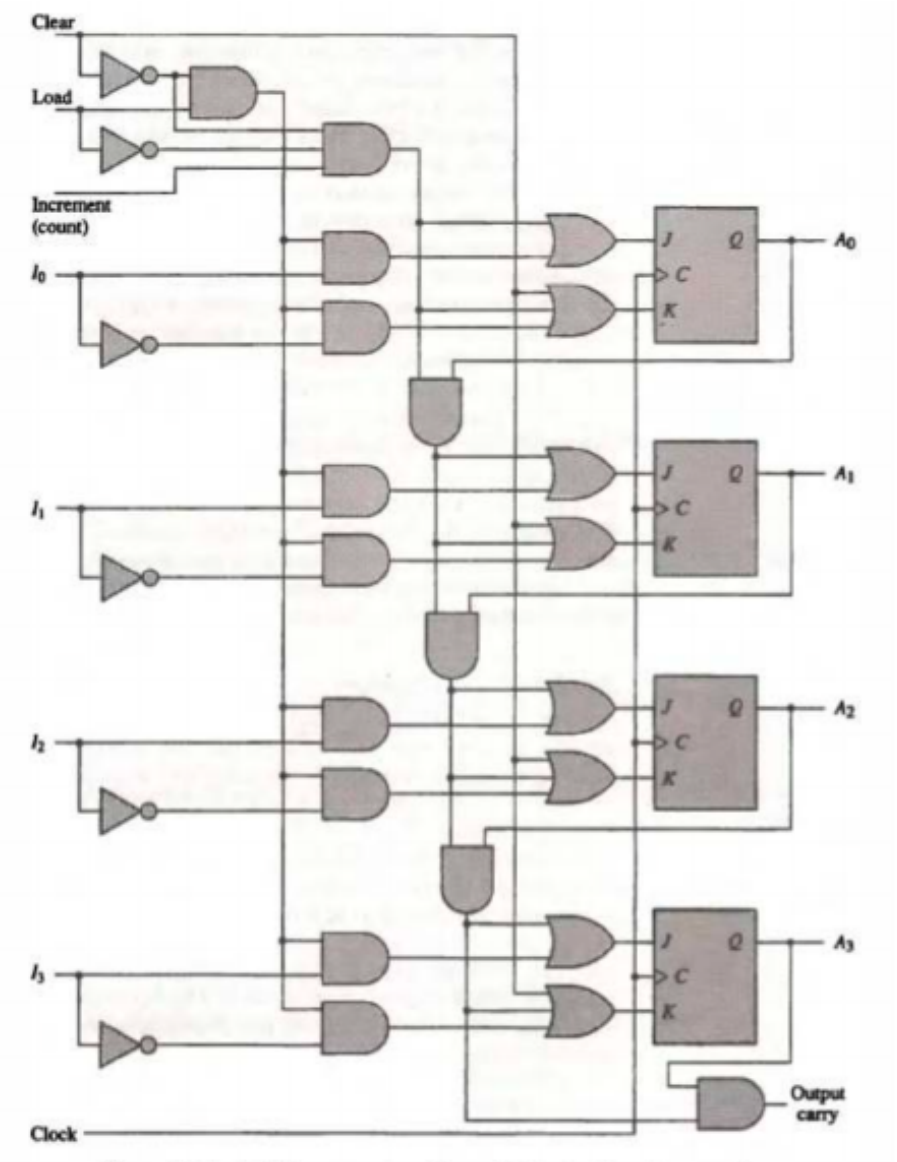




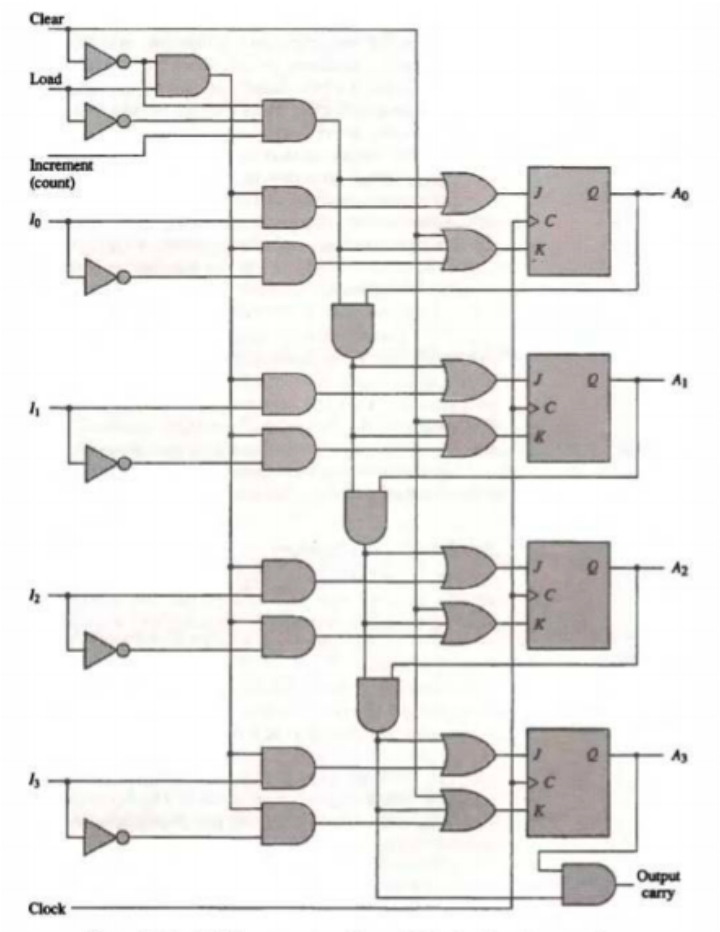


Paralel Yüklemeli İkili Sayıcı

- Sayısal sistemlerde saymanın her zaman o'dan değil, belli bir sayıdan başlaması gerekebilir.
- Belli bir başlangıç değeri, yükleme ile elde edilir.
- Bunun için paralel yükleme yapabilen sayıcılar kullanılır.


Clear
Load
Increment
(count)
I_0
I_1
I_2
I_3
J
Q
C
K
A_0
A_1
A_2
A_3
Output
carry
Clock

Paralel Yüklemeli İkili Sayıcı



Clock	Clear	Load	Increment	Operation
↑	0	0	0	No change
↑	0	0	1	Increment count by 1
↑	0	1	×	Load inputs I_0 through I_3
↑	1	×	×	Clear outputs to 0

Paralel Yüklemeli İkili Sayıcı - Ödev

- Bu sayıcıyı LOGISIM ile simülasyonda kurun.
- Herkes öğrenci numarasının son hanesinden başlayarak, sayıcının maksimumuna kadar sayma işlemi gerçekleyecek.
- Bu işlem için kaç adet clock sinyal gerekir?
- Bu 4 bitlik sayıcıyı entegre olarak kullanıp 16 bitlik sayıcı elde edin.

Bellek Birimi - Memory Unit

- Bellek birimi saklama yapabilen yazaçlardan oluşan topluluktur.
- Bellek kelime-word şeklinde ikili verileri saklar.
- 8 bit - 1 byte, 16 bit - 2 byte, 32 bit - 4 byte.
- Belleklerin boyutları saklayabilecekleri kelime kapasiteleri ile belirtilir.
- Standart bir bellekte her adres bir kelime saklar.
- k adres veri hattının bit sayısı olmak üzere, 0'dan 2^k - 1'e kadar adres hücresi oluşur.
- Bu bellekte saklanabilecek maksimum kelime sayısı 2^k'dır.
- Bellekte tutulan bir kelimeye erişim k bitlik adres bilgisi ile gerçekleşir.
- Bellekteki bir kod çözücü (dekoder) adresi alır ve kelimeye erişir.

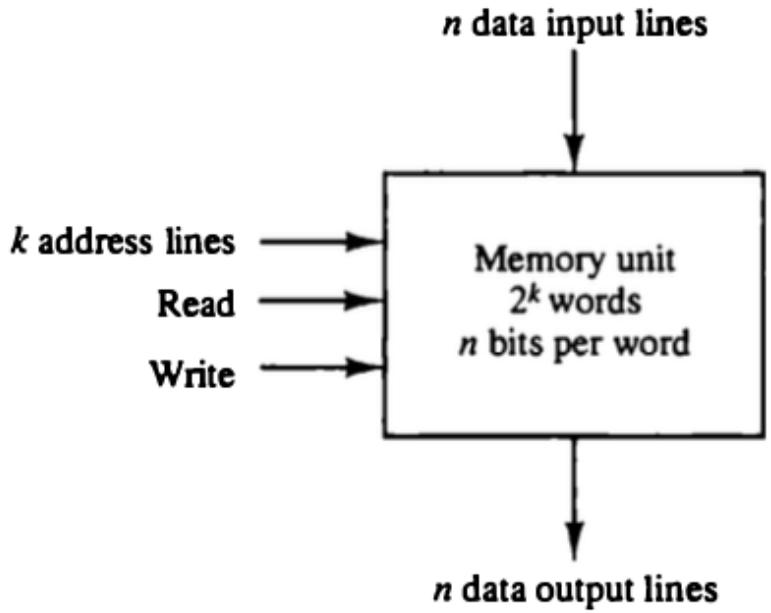
Bellek Birimi - Memory Unit

- Örn; 1024 kelimelik bir hafıza birimi için en az 10 bitlik adres veri hattı gerekir.
- 2^{32} kelimelik bir hafıza birimi için en az 32 bitlik adres veri hattı gerekir.
- $K = 2^{10}$, $M = 2^{20}$, $G = 2^{30}$ olduğuna göre;
- $64K = 2^6.2^{10} = 2^{16}$,
- $2M = 2^1.2^{20} = 2^{21}$,
- $4G = 2^2.2^{30} = 2^{32}$.

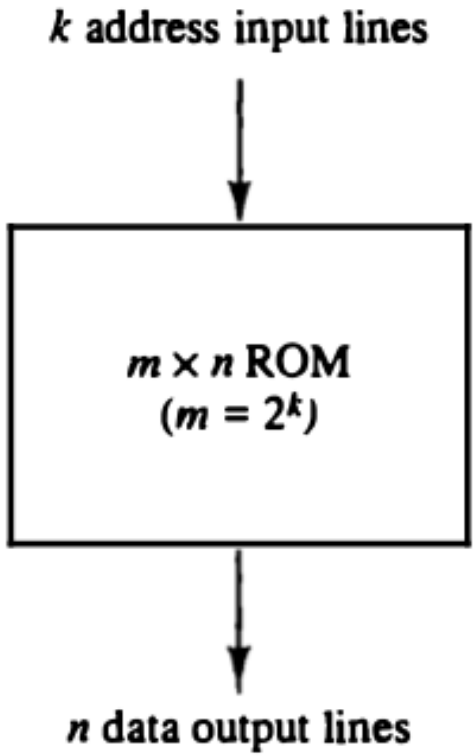
Bellek Birimi - Memory Unit

- Temelde 2 tip bellek vardır:
  - RAM (Random Access Memory): Rastgele Erişimli Bellek
  - ROM (Read Only Memory): Salt-okunur Bellek

Rastgele Erişimli Bellek - RAM



Salt-okunur Bellek - ROM



RAM - Örnek

- Her biri 8 bitten oluşan 850 tane word (kelime) saklamak için;
  - Gerekli adres satırı (k): $850>512(2^9)$, $850<1024(2^{10})$ $\Rightarrow$ k=10
  - Her bir hücrenin tutması gereken bit sayısı: 8
- Her biri 12 bitten oluşan 2500 tane word (kelime) saklamak için;
  - Gerekli adres satırı (k): $2500>2048(2^{11})$, $2500<4096(2^{12})$ $\Rightarrow$ k=12
  - Her bir hücrenin tutması gereken bit sayısı: 12

Sayı Sistemleri - Tümleyen

- Tümleyen sayılar bilgisayarların çıkarma işlemi yapmalarını kolaylaştırır.
- r tabanındaki n bitlik N sayısının;
- (r - 1)'e göre tümleyeni => (r^n - 1) - N ile hesaplanır.
- Örn: r=2 için, r-1 = 1'dir. Yani (2^n - 1) - N ile ikili tabanda tümleyen hesaplanır.
- Örn: N=1011001b için;
- (r^n - 1) - N => (2^7 - 1) - 1011001b = 127 - 1011001b = 1111111b - 1011001b = 0100110b
- r'ye göre tümleyeni => r^n - N ile hesaplanır.
- r'ye göre tümleyen, r-1'e göre tümleyene 1 eklenerek hesaplanır.
- Örn: N=1011001b için;
- r^n - N => (2^7 - 1011001b) = 128 - 1011001b = 10000000b - 1011001b = 0100111b

Sayı Sistemleri - Tümleyen

- Neden sayısal sistemler 2'ye tümleyene ihtiyaç duyuyor?
- M ve N ikili tabanda sayılar olmak üzere M - N için;
- N'nin 2'ye göre tümleyeni alınıp, toplama işlemi yapılır.
- M = 1010100b, N = 1000011b => M> N
- M - N = 1010100b + 0111101b = 10010001b
- N - M = 1000011b + 0101100b = 1101111b

Sayı Sistemleri - Tümleyen

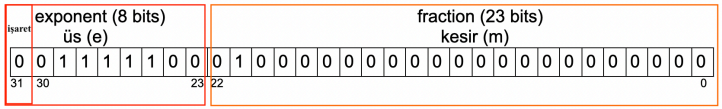
- İşaretli sayı sistemlerinde MSB = 0 ise pozitif, MSB = 1 ise negatif sayıdır.
- Bir sayı 3 farklı şekilde gösterilebilir:
  - İşaretli normal ikili (normal form) (Pozitif sayılar sadece bu formda)
  - İşaretli 1'e göre tümleyen (1st complement)
  - İşaretli 2'ye göre tümleyen (2nd complement)
- 8 bitlik bir yazaçta 14 sayısını depolamak istiyorsak;
  - +14 => MSB = 0 ve 00001110
  - -14 =>
    1. 10001110b <= işaretli sayı sistemi gösterimi
    2. 11110001b <= işaretli 1'e göre tümeleyen gösterim
    3. 11110010b <= işaretli 2'ye göre tümeleyen gösterim

Kayan Sayı Gösterimi - Floating Numbers

- $+ 6132.789 = + 0.6132789 + 04 \Rightarrow 0.6132789 \times 10^{+4}$
- $m \times r^e$
- $+ 1. \times 2^e$
- $+ 1001.11b \Rightarrow 1.00111b \times 2^3$
- $m \times 2^e = + .1001110b \times 2^4$

IEEE Standard 754 Floating Point Representation

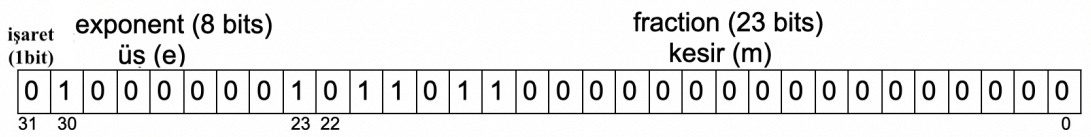
- Genel olarak 32 bitlik bir bellek alanından, 3 parça şeklinde oluşturulur.



- 1. işaret biti : + => 0, - => 1
- 2. üs (exponent) : Biased exponent (sapma) olarak gösterilir.
- Bias: k üssün bit sayısı olmak üzere $2^{k-1} - 1$ olarak hesaplanır. 32 bit standartta üs 8 bit olduğuna göre bias 127.
- Biased exponent: bias + üs ile hesaplanmıştır.
- Fraction (kesir): olduğu gibi yazılır.
- O zaman şekilde;
  - işaret biti = 0 sayı pozitif,
  - biased exponent: 01111100b ise 01111100b - 1111111b (127) = -3
  - kesir: 01 0 halde sayı: + 1.01b $\times$ 2^{-3}

IEEE Standard 754 Floating Point Representation

- Örnek: 101.1011b
- $m \times 2^e$ olarak ifade edelim.
- $1.011011b \times 2^2$
- işaret biti: 0
- biased exponent: 127 + üs = 1111111b + 10b = 10000001b
- kesir: 011011b



Yazaç Aktarım Dili

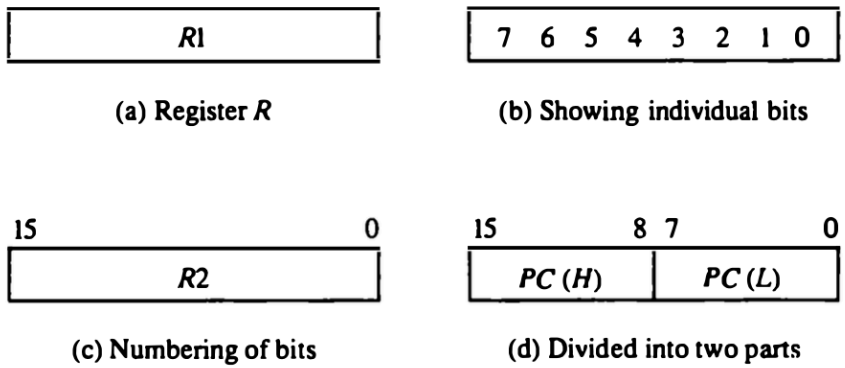
- Sayısal bilgisayarların tasarımında modüler yaklaşım kullanılır.
- Bu modüller yazaçlar, dekoderler, aritmetik elemanlar ve denetim mantığından oluşur.
- Her bir sayısal modül bunları oluşturan yazaç ve yazaçlarda mevcut bilgiler üzerinde icra edilen işlemlerle tanımlanır.
- Yazaçlarda yüklü veriler üzerinde icra edilen işlemlere mikro işlem denir.
- Mikro işlemin sonucunda yazacın içindeki ikili veri değişebilir veya başka bir yazaca aktarılabilir.
- Kaydırma, sayma, yükleme ve silme mikro işlemleri örnek olarak verilebilir.

Yazaç Aktarım Dili

- RTL: Sayısal modüllerin yazaçları arasında meydana gelecek mikro işlemleri sembolik olarak ifade eden bir sistemdir.
- Bu sistemde yazaçlar büyük harflerle ifade edilir.
- Örn: Bellek biriminin bir adresini gösteren yazaç bellek yazacı olarak isimlendirilir, MAR diye gösterilir.
- PC: Program Counter (Program Sayacı)
- IR: Instrucion Register (Buyruk Yazacı)
- R1: R1 işlem yazacı

Yazaç Aktarım Dili

- R1←R2

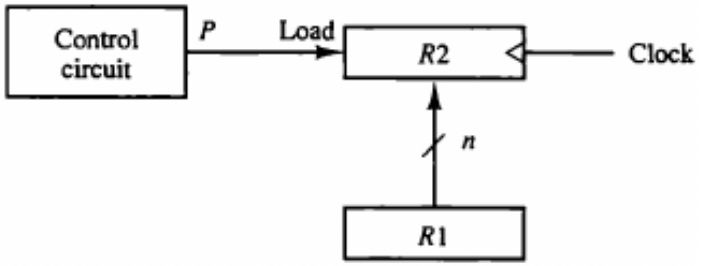


Yazaç Aktarım Dili

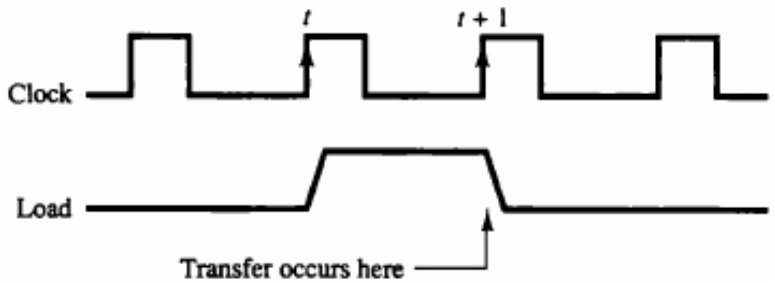
- R2 ← R1, bilgi aktarım mikro işlemi. R1'in içeriği R2'ye kopyalanır. R1 değişmez.
- if P = 1 (boolean) then R2 ← R1, denetim fonksiyonu ,
- Denetim değişkeni P=1 olduğunda, R1 kaydedicisinin içeriğini R2'ye aktarır.

Mikro işlemin Blok Şeması

Figure 4-2 Transfer from R1 to R2 when P = 1.



(a) Block diagram



(b) Timing diagram

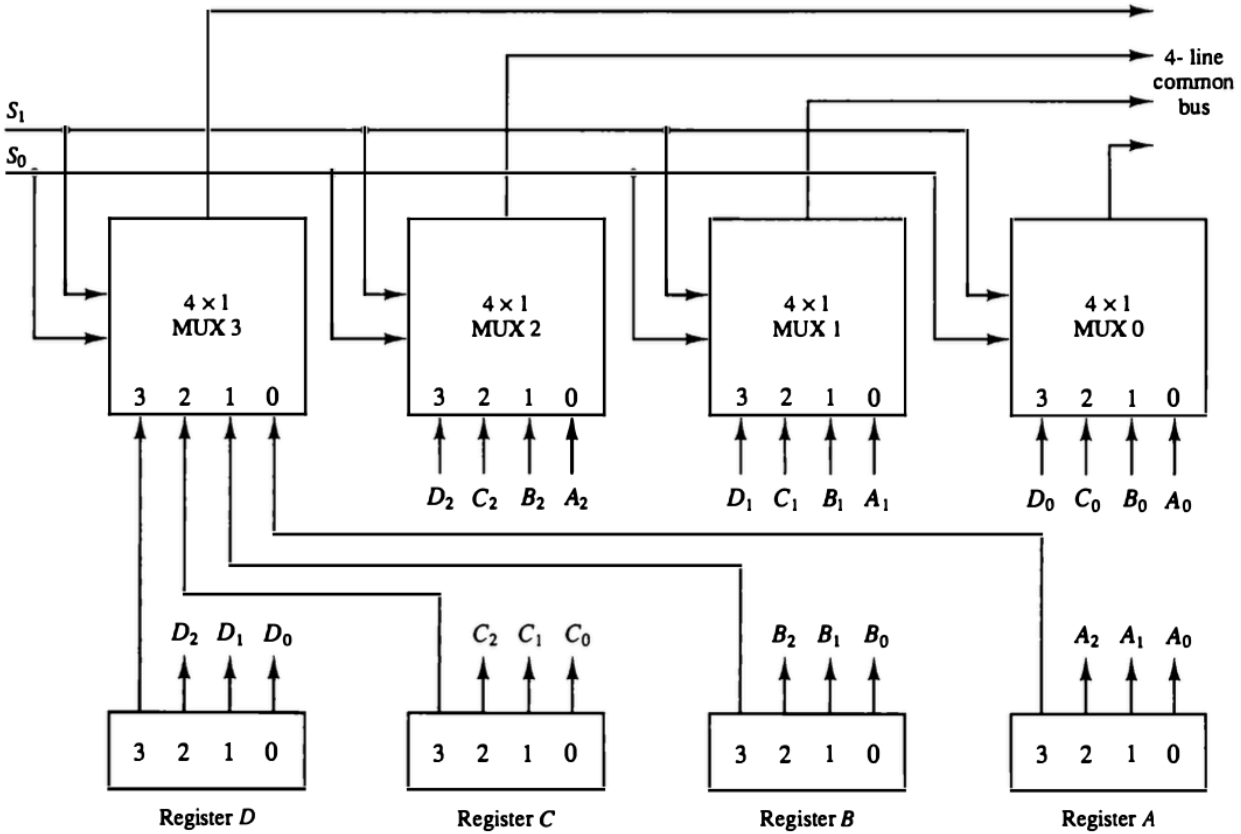
Yazaç Aktarım Dili - Kurallar

- Yazaçlar büyük harflerle ve harfi takip eden rakamlarla gösterilir. Örn: MAR, R1, PC
- Parantez ( ) yazaçların bölümlerini gösterir. Örn: R2(0-7), R2(L)
- Ok işareti (←) bilginin gideceği yönü göstermek için kullanılır.
- Virgül (,) aynı anda birden fazla mikro işlemi ifade için kullanılır. Örn: R2 ← R1, R1 ← R2 (swap)

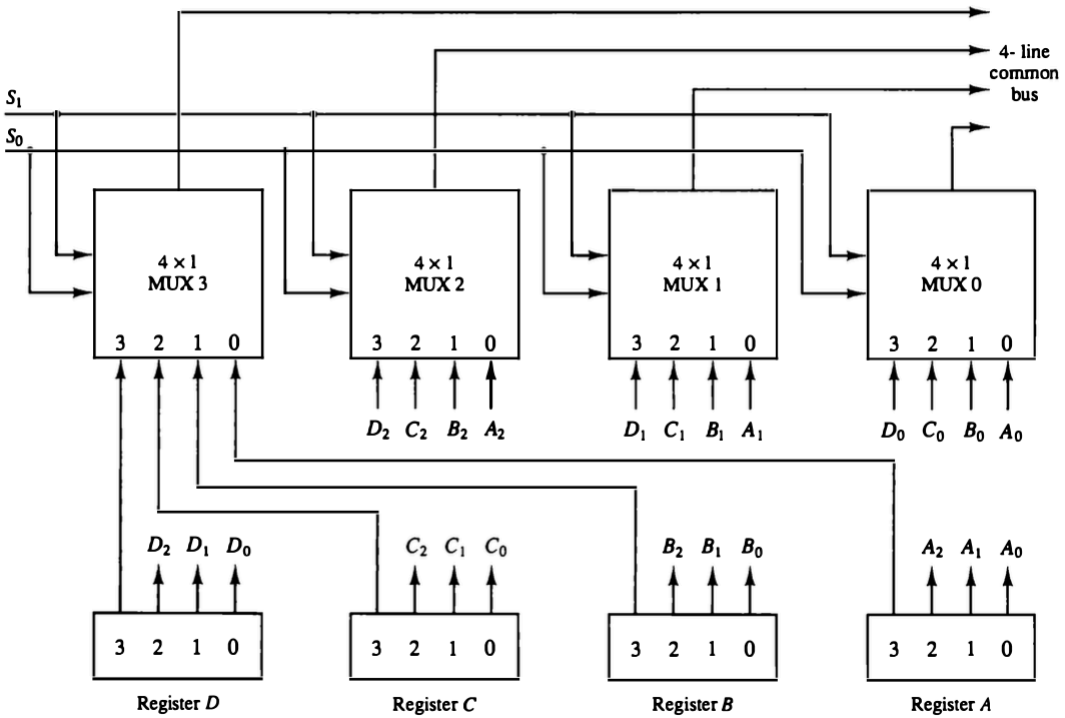
Veri Yolu ve Bellek Aktarımı

- Herhangi bir sayısal bilgisayarın birçok yazaca ihtiyacı vardır.
- Bu yazaçlar arasında birçok transfer işlemi gerçekleştirilmesi gerekir ve bu birçok veri yolu gerektirir.
- Bu nedenle ortak bir yolu kurulması sistemin etkin bir şekilde çalışmasını sağlar
- Multiplexer böyle bir sistemin kurulması için tercih edilebilecek devrelerdendir.

Veri Yolu ve Bellek Aktarımı



Veri Yolu ve Bellek Aktarımı



Dört yazaçlı veri yolu sistemi

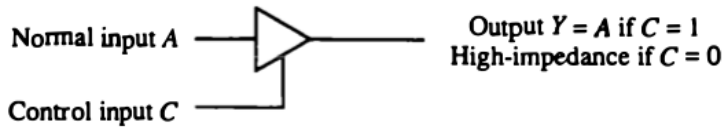
S_1	S_0	Register selected
0	0	A
0	1	B
1	0	C
1	1	D

Veri Yolu ve Bellek Aktarımı

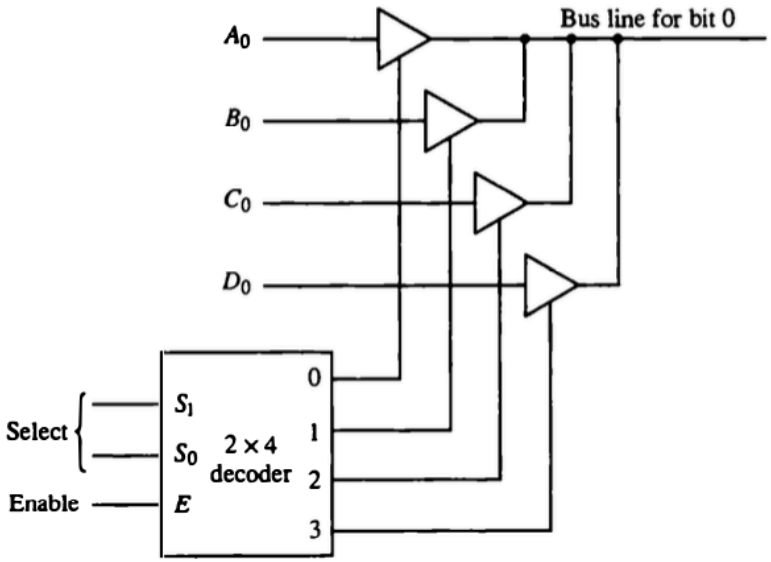
- Genel olarak bir veri yolu sistemi n bitlik k tane yazaçtan, n bit hatlı ortak bir veri yolundan ve kx1 boyutunda n tane MUC'tan oluşur.
- Örn: 16 bitlik 8 adet yazaçlı bir bilgisayarın ortak veri yolu 16 tane 8x1 boyutunda MUX gerektirir. Ortak veri yolu 16 bitlik bir yoldur.
- RTL: VERİ YOLU ← C, R1← VERİ YOLU
- C yazacının içeriği ortak veri yoluna aktarılır. R1 yazacının yükleme denetim girişi (load) aktif hale geldiğinde, o andaki VERİ YOLU içeriği R1 yazacına aktarılır.

3 Durumlu veri Yolu Bufferları (3-state Bus Buffers)

- Veri yolu MUX yerine 3 durumlu kapılarla da oluşturulabilir.
- 3 durumlu kapı (3-state buffer): 3 ayrı durumu gösteren sayısal bir devredir.
  - 1
  - 0
  - Yüksek direnç durumu



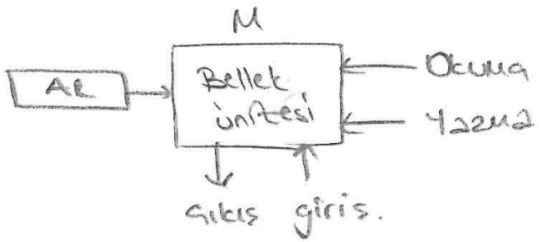
3 Durumlu veri Yolu Bufferları (3-state Bus Buffers)



- n bitlik k yazaçlı bir sistem için veri yolu oluşturmak isteniyorsa, k adet bufferdan oluşan n adet yukarıdaki devreden gereklidir.

Bellek Aktarımı

- Bilginin bellekten dışarı aktarılması okuma işlemi,
- Yeni bilgilerin belleğe aktarılması yazma işlemi.
- Seçilen bellek kelimesi adres yardımıyla belirlenir.
- Bellek M harfiyle gösterilir.
- AR isimli özel yazaç bellekteki hedef adres hücresinin bilgisini saklar.



Bellek Aktarımı

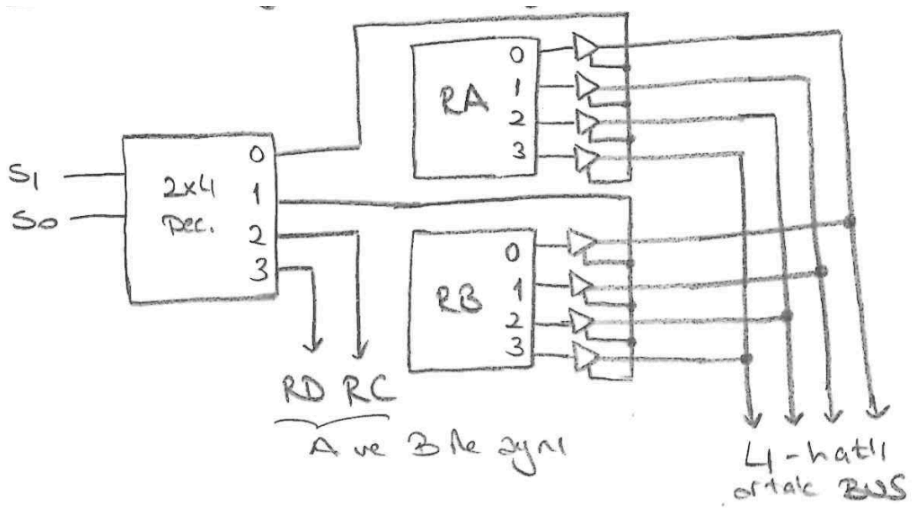
- Okuma: DR $\leftarrow$ M[*AR*]
  - AR içeriği bellek adres hattına gönderilir.
  - READ (=1) işareti bellek ünitesine gönderilir.
  - Belirlenen adresteki veri bellek çıkış hattına gönderilir.
  - Bu veri DR (data register) sayacına yüklenmek üzere ortak veri yolu (BUS) üzerinden gönderilir.
- Yazma: M[*AR*] $\leftarrow$ DR
  - AR içeriği bellek adres hattına gönderilir.
  - WRITE (=1) işareti bellek ünitesine gönderilir.
  - DR yazacındaki veri ortak veri yolu üzerinden bellek veri girişine gönderilir.
  - Bu veri bellekte belirlenen hücreye gönderilir.

Bellek Aktarımı - Örnek

- Aşağıdaki işlemi 2 yazaç aktarım işlemi yapan denetim fonksiyonu ile gerçekleyin.
- if (P=1) then (R1 ← R2) else if (Q=1) then (R1 ← R3)
- P: (R1 ← R2)
- P'Q: (R1 ← R3)

Bellek Aktarımı - Örnek

- Daha önce gördüğümüz MUX ile kurulan veri yolunu 3 durumlu buffer ve dekoderle kurunuz. Şeklini çiziniz.



Bellek Aktarımı - Örnek

- Bellekte gerçekleşen aşağıdaki işlemleri açıklayınız.
- R2 ← M[*AR*] - AR yazacı adresi belirlenen veri bellekten okunup, R2'ye aktarılır.
- AR = 100, M[*AR*] ← R3 - Belleğin 100 numaralı adresine R3 yazacı içerisinde veri yazılır.

Bellek Aktarımı - Örnek

- MUX ile oluşturulmuş 32 bitlik 16 yazaçlı bir veri yolu olan bir bilgisayar sisteminde;
  - Her bir seçici için gerekli olan seçim ucu sayısı: 4
  - Her bir seçicinin boyutu: 16x1
  - Veri yolunda kullanılan seçici adedi: 32