

Mantıksal Sistem Tasarımı – BLM 201

Hafta 11: Bellekler



Fenerbahçe Üniversitesi

Öğretim Elemanları

Öğretim Üyesi: Dr. Vecdi Emre Levent

Ofis: 311

Email: emre.levent@fbu.edu.tr

Asistan: Arş. Gör. Uğur Özbalkan

Ofis: 311

Email: ugur.ozbalkan@fbu.edu.tr

Ders Planı

- Bellek Elemanları
 - Random Access Memory
 - Read Only Memory
 - BRAM
 - DRAM

Bellek Elemanları

- Bir kombinasyonel devrenin çıktısı saklayıcılarda saklanabilir.
- Ancak saklayıcıların sayısı oldukça limitlidir.
- Saklanacak veri miktarı arttığında bunları saklayıcıda tutmak yerine, BRAM(Block RAM) veya DRAM (Dynamic Random Access Memory) gibi saklama alanlarında tutarız.

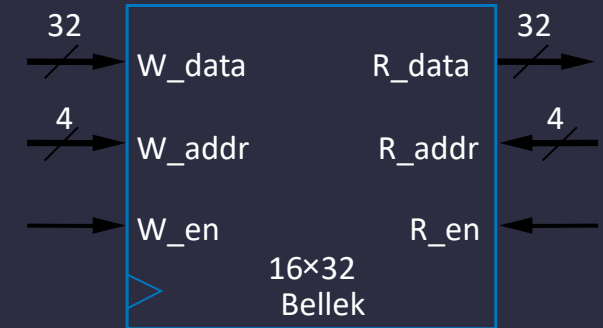
Bellek Elemanları

- ***MxN Bellek***
 - M lokasyon, N bit genişliğinde



Random Access Memory (RAM)

- RAM – Hem yazılabilir hem de okunabilir belleklerdir.
 - Random Access Memory, yani rasgele erişimli bellek
 - Bu isimlendirme teyp türü kayıt elemanları varken türetilmiştir. Bu tür belleklerde sadece ardışık olarak erişim vardı.
 - Rasgele erişimin özelliği, RAM üzerinde istenen adrese doğrudan erişilebilir. Yani ardışık erişimli belleklerde olan bir adrese gitmek için önceki tüm adresleri gezmek gerekmez.
 - Bir saklayıcı grubu ile de bellek yapılabilir. Ancak BRAM'ler ve DRAM'lerde çok daha fazla alan mevcuttur.



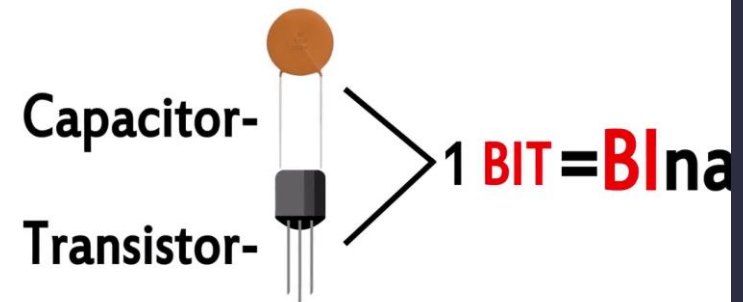
Random Access Memory (RAM)

- RAM'ler 2 gruba ayrılabilir
- Statik
- Dinamik

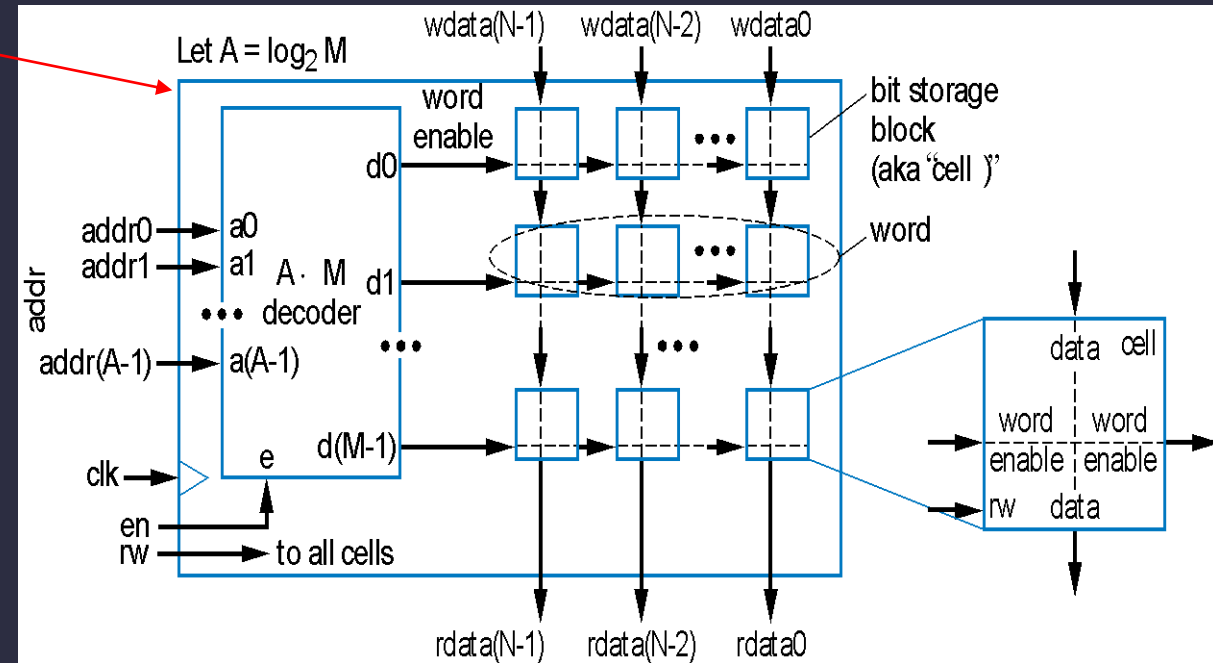
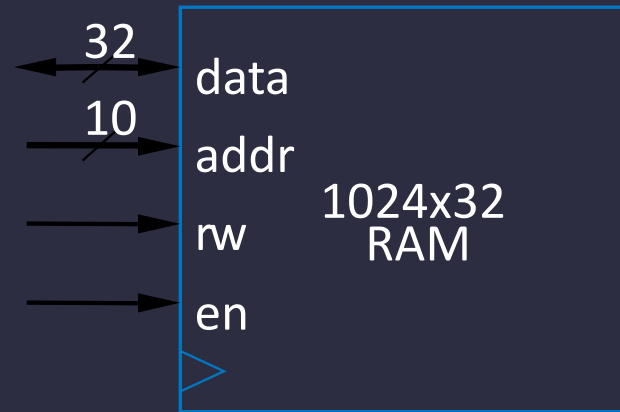
Static RAM



DYNAMIC RAM



Statik RAM (SRAM)

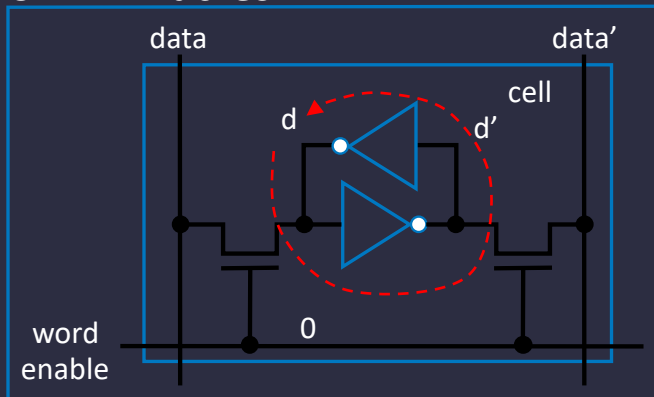


- Statik Hücreli RAM

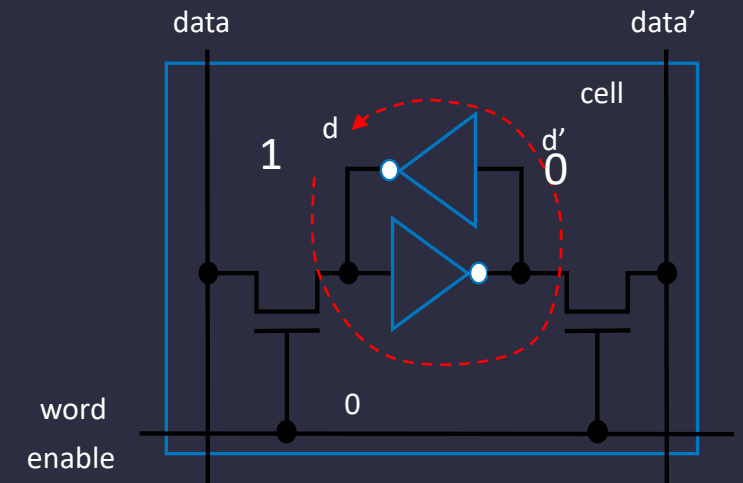
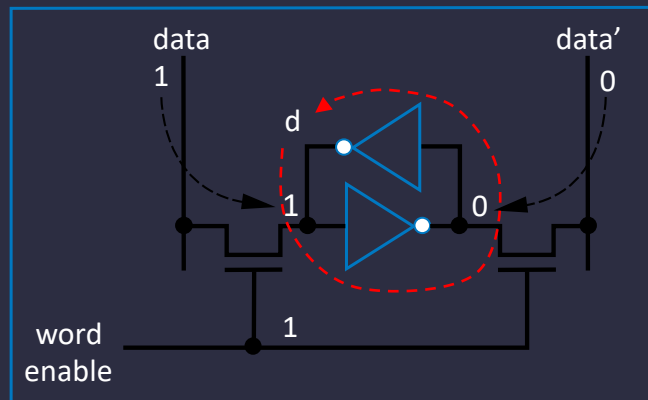
- 6 transistör'den oluşur (Inverter 2 transistör kullanmaktadır)
- Statik RAM'lerin özelliği tamamı transistörlerden oluşmasıdır.

Statik RAM (SRAM)

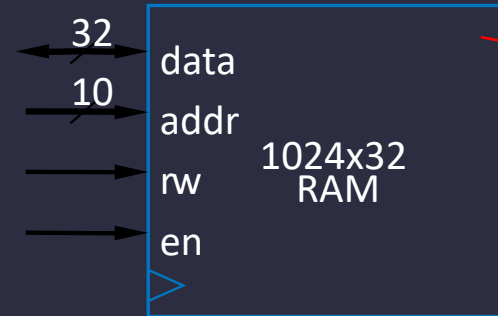
SRAM Hücresi



SRAM Hücresi

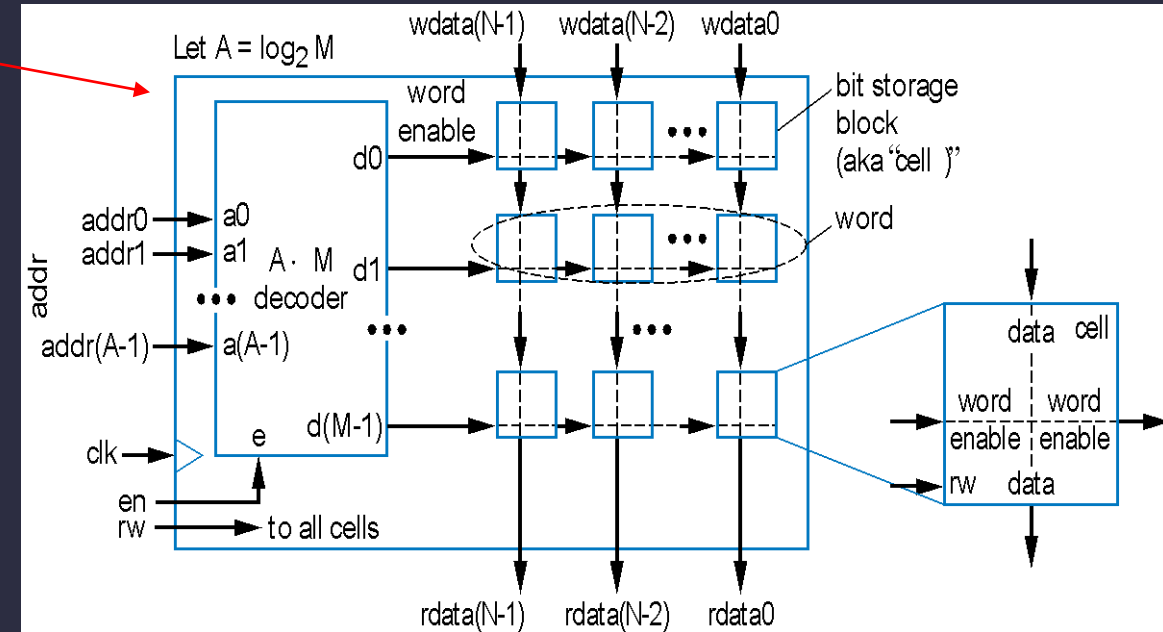


Dinamik RAM (DRAM)



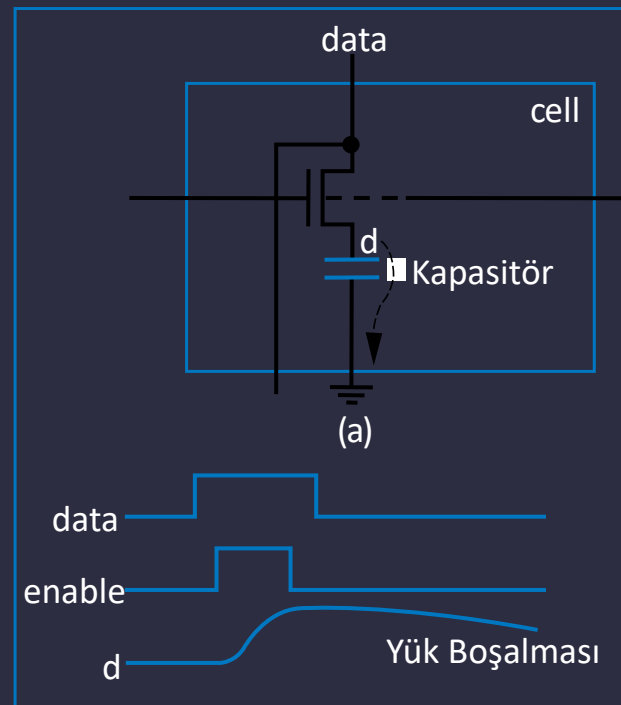
- Dinamik RAM Hücresi

- 1 transistor
- Saklamak için kapasitör kullanmaktadır.



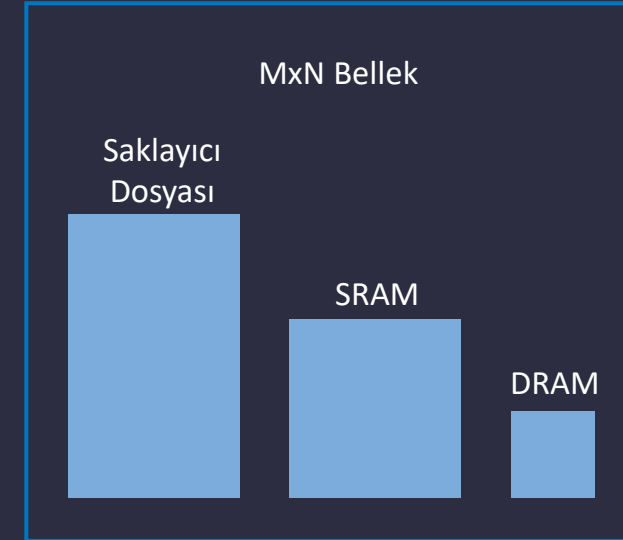
Dinamik RAM (DRAM)

DRAM cell



Bellek Karşılaştırmaları

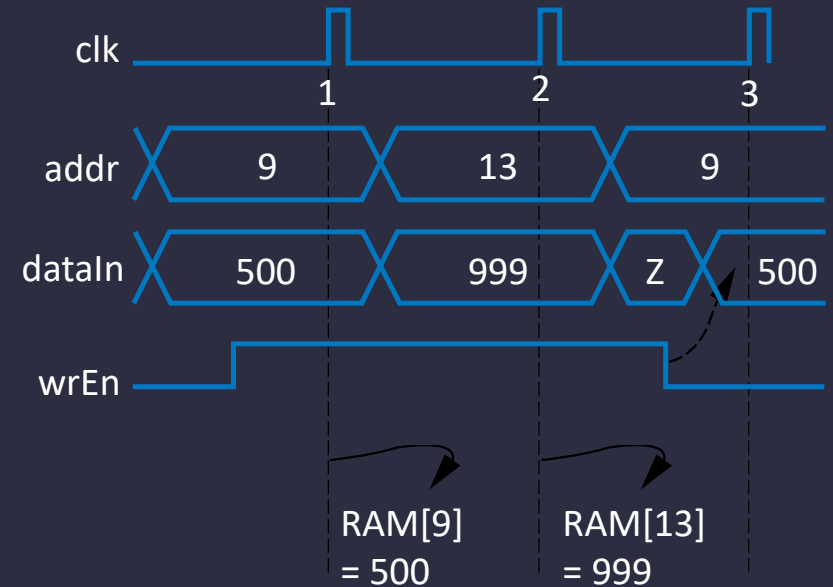
- Saklayıcı Dosyası (Register File)
 - Saklayıcılardan oluşur
 - En hızlısıdır
 - Ancak en pahalı çözümdür
 - Birim saklama alanı olarak en büyük fiziksel alana sahiptir
- SRAM
 - Hızlıdır
 - Saklayıcılara göre daha kompakt yapıdadır.
 - FPGA'lerdeki BRAM'ler SRAM yapısındadır
- DRAM
 - En erişim hızına sahiptir
 - Yenileme (Refreshing) zamanlarında erişilemezler
 - Ancak diğer belleklere göre en kompakt yapıdır
 - DRAM'lerin adreslerine erişim yapılırken ilk defa bir adrese erişilecekse belirli bir gecikme ile erişilir. Bu adresi takip eden adreslerin erişiminde ise tekrar bir gecikme yaşanmaz.



Belleklerin Birim Alan Açısından Kapladıkları Fiziksel Alan

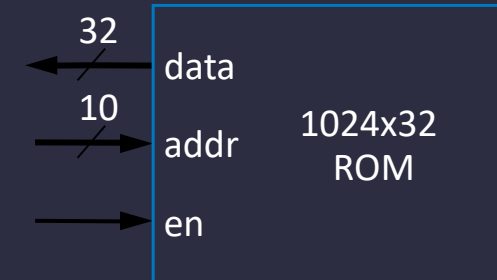
RAM'e Yazma ve Okuma

- Yazma
 - *addr* portuna RAM'in yazılmasını istediğiniz adresi yazın, bununla birlikte paralel olarak *wrEn* sinyalini 1'e çekin
- Okuma
 - *addr* portuna okunmasını istediğiniz adresi verip, 1 clock cycle sonra *dataOut* portundan veriyi okuyunuz.



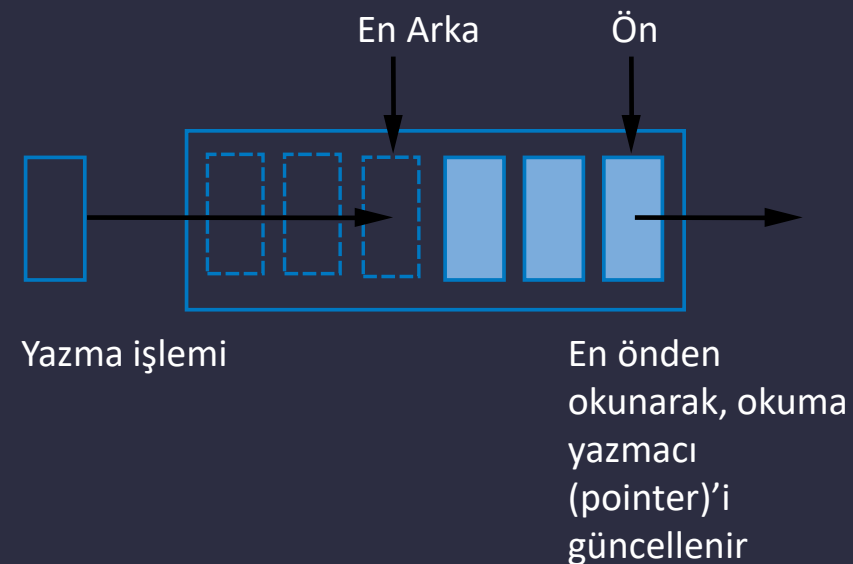
Read-Only Memory – ROM

- Bu tür belleklerden sadece veri okunabilir, yazma yapılamaz
 - Sadece çıkış portları vardır
- RAM'e göre avantajları
 - Belleğinin değiştirilemez olmasını istiyorsanız bu yapı tercih edilebilir.

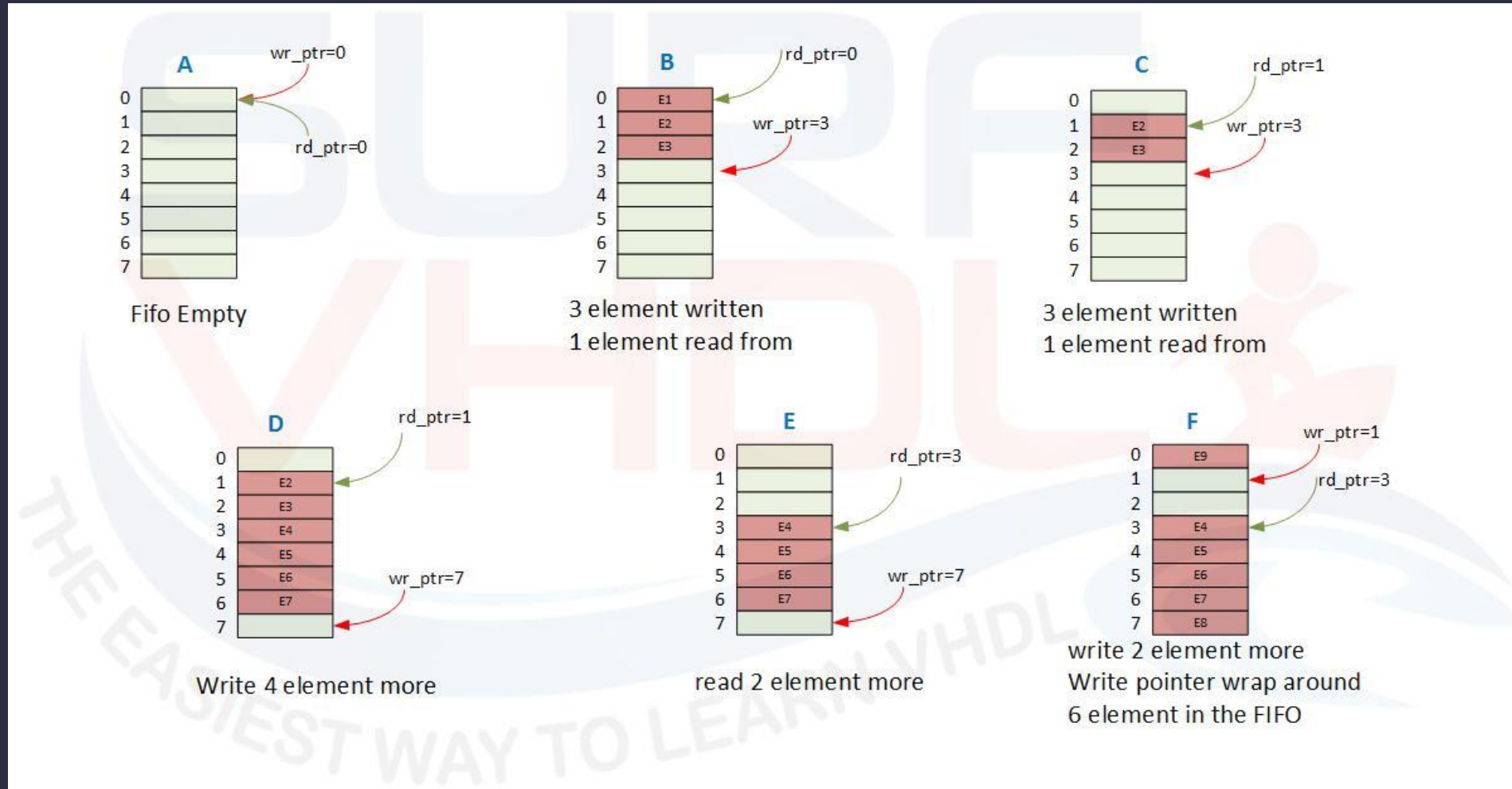


Kuyruklar (Queues) – FIFO (First In First Out)

- *Kuyruklar, ilk yazılanın ilk okunduğu saklama alanlarıdır*
- Push ve pop isminde iki ana işlemi bulunur.
- Push işlemi saklama alanına veriyi kaydetmeye, pop işlemi ise alandan veri okumak için kullanılır.



Kuyruklar (Queues) – FIFO (First In First Out)



Kuyruklar (Queues) – FIFO (First In First Out)

Örnekler:

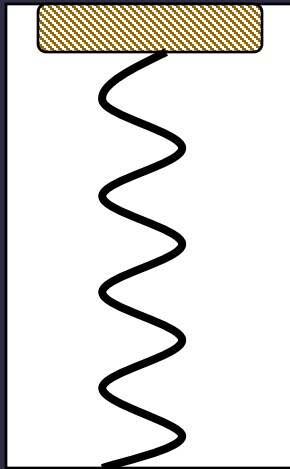
- Klavye
 - Klavye bir FIFO'a basılan tuşları doldururken, bilgisayar FIFO boş olmadıkça okuyabilir.
- Ağ yönlendiricileri (Router)
 - Gelen paketleri bir FIFO'a basar, ardından paketin nereye yönlendireceği bilgisini içeren byte gelince, önceki FIFO'da tutulan byte'lar ile birlikte istenen adrese yönlendirir.

Yığınlar

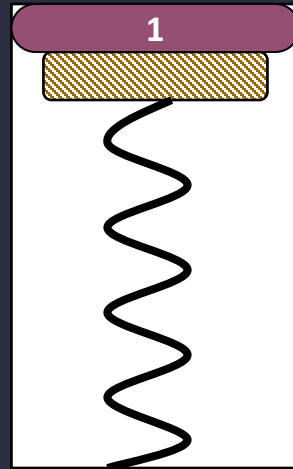
- LIFO (last-in first-out, Son Giren İlk Çıkar) depolama yapısıdır.
 - Kaydedilen ilk veri, en son çıkar.
 - Yığına kaydedilen son veri ise, ilk çıkar

- İki ana fonksiyonu vardır:
- PUSH: Yığına veri ekler
- POP: Yığından veri alır

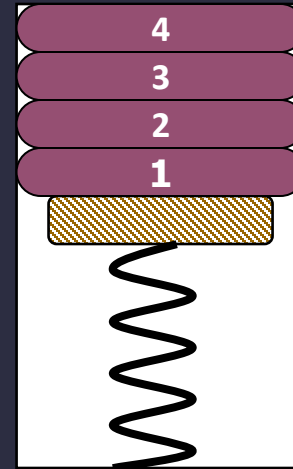
Yığın Örneği



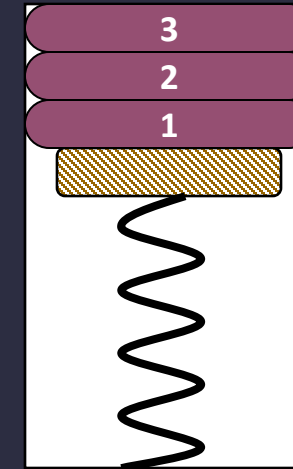
Başlangıç Durumu



Bir veri eklendikten sonra



3 veri daha eklendikten sonra



Bir veri alındığında

Yazılım Gerçeklemesi

- Veriler hareket etmez, yığının başlangıç adresi değişir

