

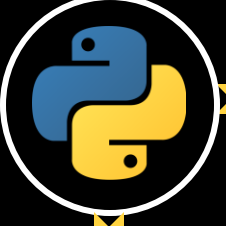


# Bilgisayar Programlamaya Giriş

---

Murat Uğur KIRAZ  
Online Python Eğitimi

# Bilgisayar Nasıl Çalışır?



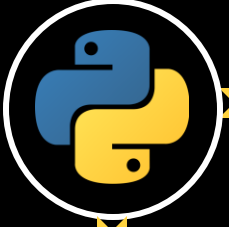
- Program, bilgisayarı kullanılabilir hale getirir.
- Bilgisayarlar çok karmaşık görevleri yerine getirebilirler.
- Yalnızca son derece basit işlemleri gerçekleştirebilir.
- Çağdaş bilgisayarlar, yalnızca toplama veya bölme gibi çok temel işlemlerin sonuçlarını değerlendirebilir.

# Doğal diller ve programlama dilleri



- Bilgisayarların da makine dili adı verilen, oldukça ilkel bir dili vardır.
- Bir bilgisayar, teknik olarak en karmaşık olanı bile, en ufak bir zeka izinden dahi yoksundur.
- Bilinen komutların tam bir kümesine talimat listesi adı verilir (instruction list) ve bazen IL olarak kısaltılır.
- Şu anda hiçbir bilgisayar yeni bir dil oluşturabilecek kapasitede değil.

# Bir Dilin «Dil» Olması İçin Ne Gereklidir?

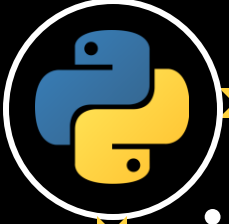


- Her dil aşağıdaki unsurlardan oluşur:
  - Alfabe
  - Sembol
  - Söz dizimi
  - Anlambilim
- IL (Talimat Listesi) aslında bir makine dilinin alfabesidir. Bilgisayarın ana dilidir.
- Temel seviye diller
- Yüksek Seviye diller

# Tarihçe

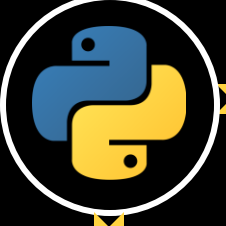


# Teknolojinin Doğuşu



- Mekanik Dönem
- Elektro-Mekanik Dönem
- Elektronik Dönem
- Birinci Nesil Bilgisayarlar (1940 – 1956)
- İkinci Nesil Bilgisayarlar (1956 – 1963)
- Üçüncü Nesil Bilgisayarlar (1946 -1971)
- Dördüncü Nesil Bilgisayarlar (1971 – Günümüz)
- Beşinci Nesil Bilgisayarlar (Günümüz – Yakın Gelecek)

# ilk Bilgisayar



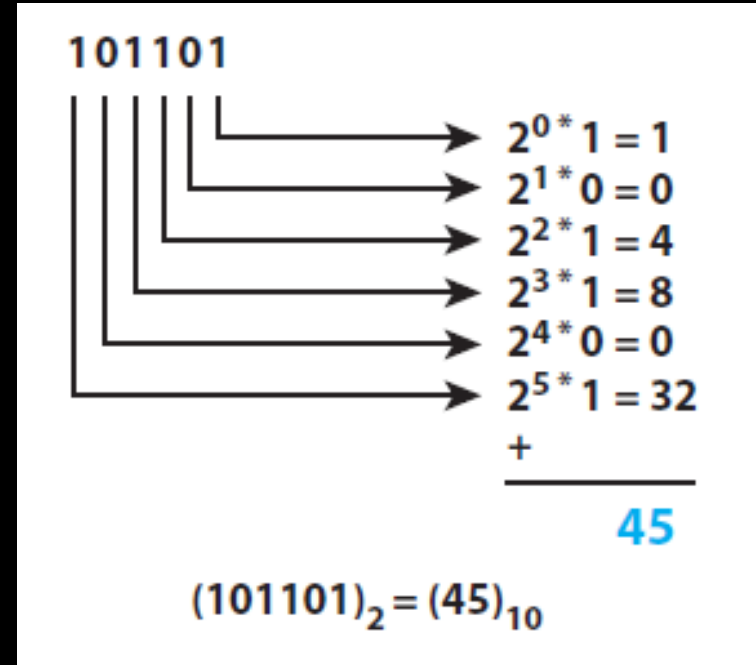
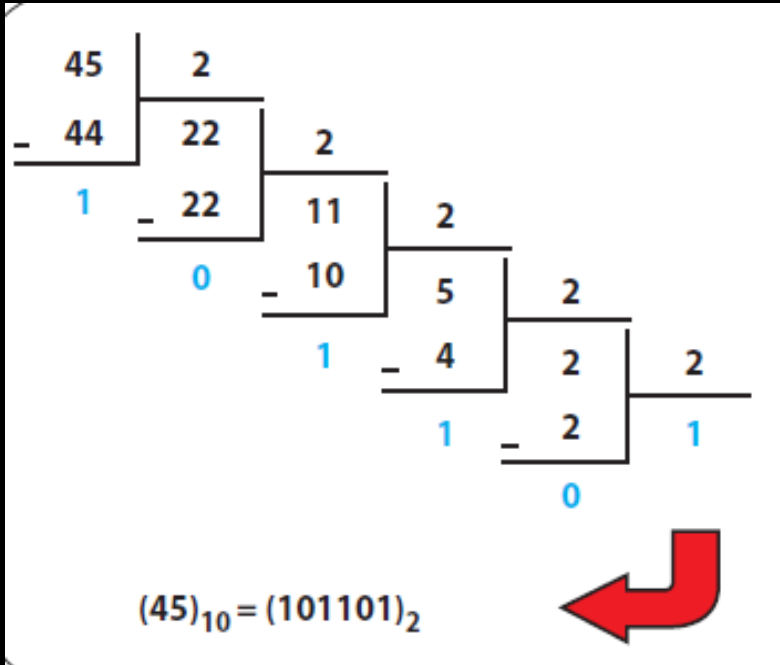
# İkili Sayı Sistemi







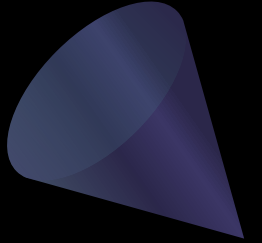
# İkili, Sayı Sistemi



# Kavramların İkili (Binary) İfadesi



- Sayılar,
  - Metinler,
  - Renkler, Fotoğraflar, Video
  - Ses,
- 8 Bit = 1 byte
  - 1000 byte = 1 Kilobyte
  - 1000 Kilobyte = 1 Megabyte
  - 1000 Megabyte = 1 Gigabyte
  - 1000 Gigabyte = 1 Terrabyte



# Kesirli Sayılar



$1 \times 2^3$	$1 \times 2^2$	$0 \times 2^1$	$1 \times 2^0$	.	$1 \times 2^{-1}$	$0 \times 2^{-2}$	$1 \times 2^{-3}$	$1 \times 2^{-4}$
<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	.	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>
8	4	0	1		0.5	0	0.125	0.0625

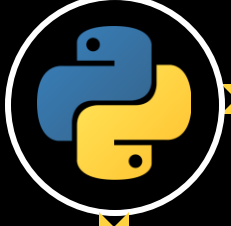
↑  
Binary point

$8 + 4 + 0 + 1 + 0.5 + 0 + 0.125 + 0.0625 = 13.6875$  (Base 10)

# Metinler



- [ASCII tablosu](#)
- UTF Kodlama



# Renkler, Fotoğraflar, Video

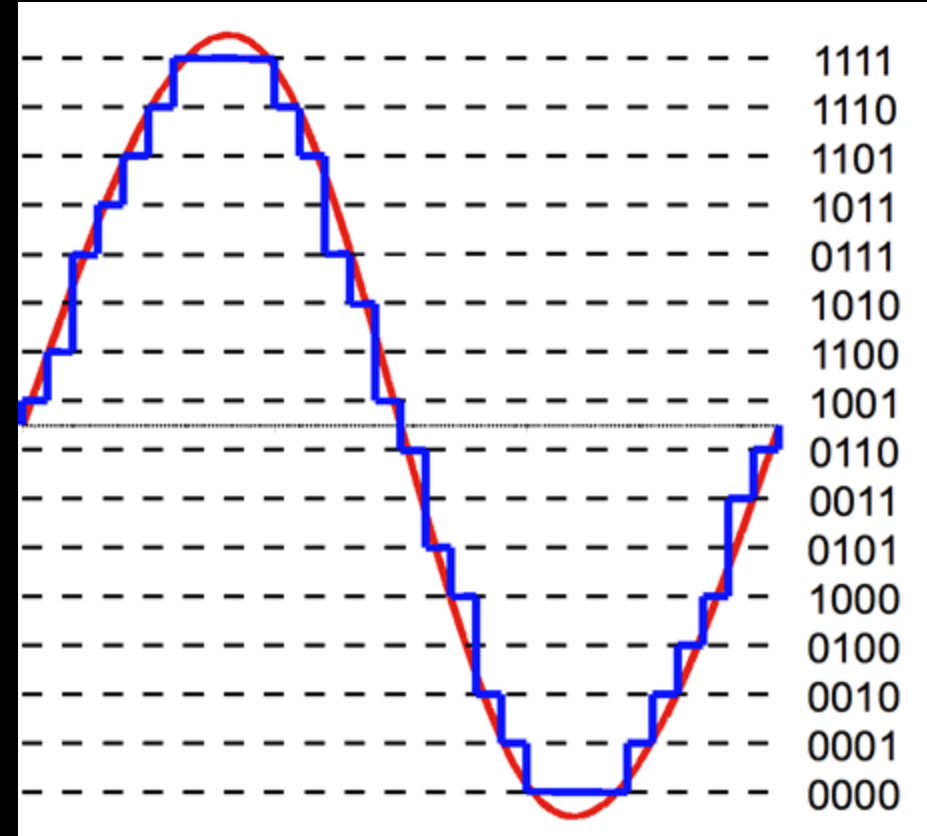
- RGB Değeri
- Piksel



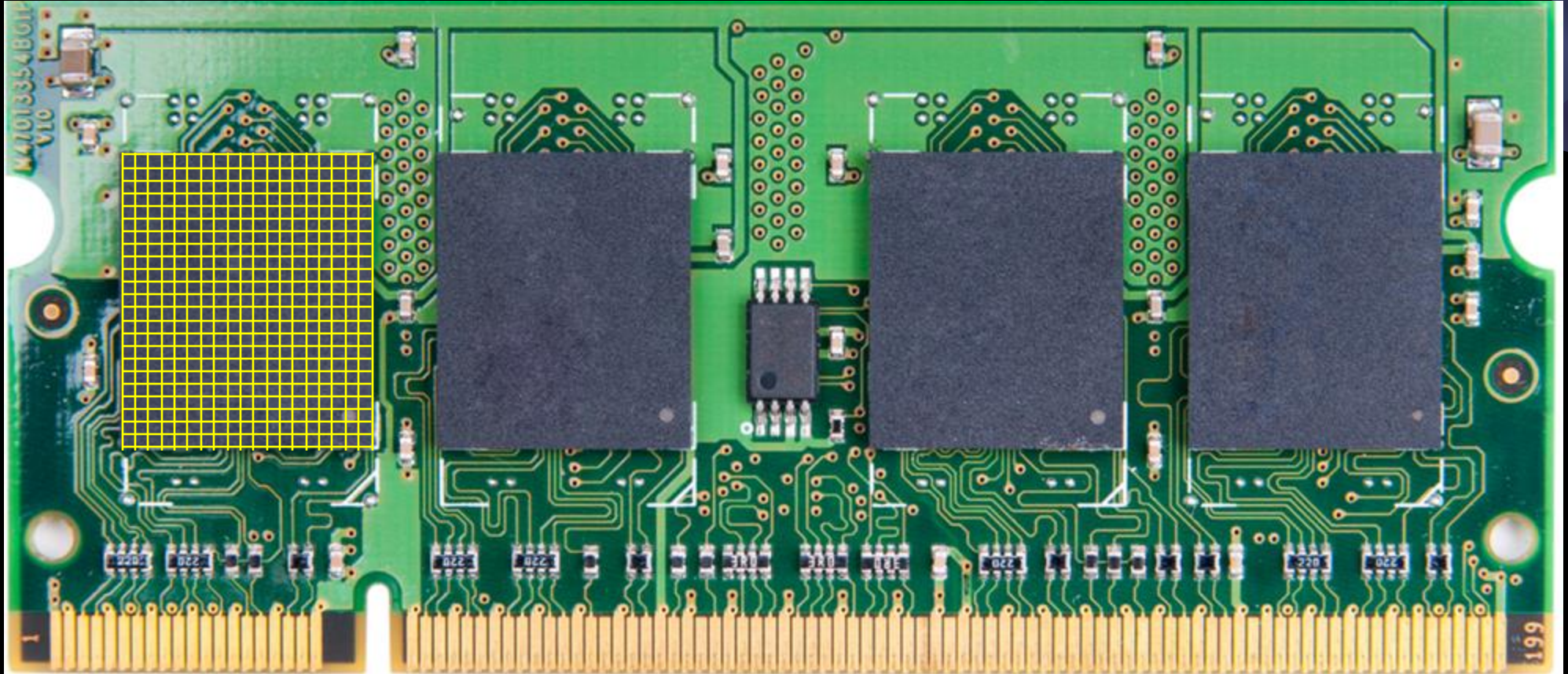
# Ses



- Analog-Dijital dönüşüm

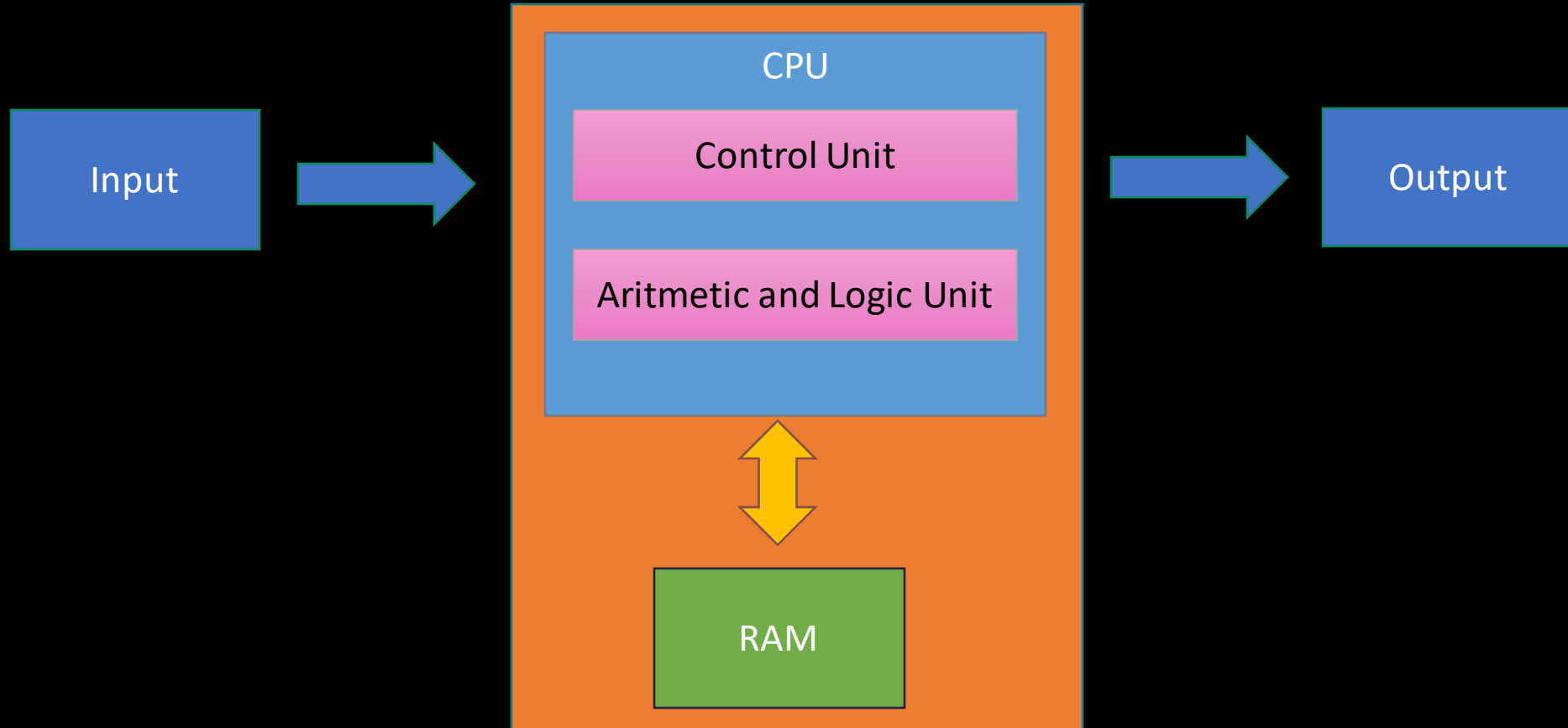


# RAM



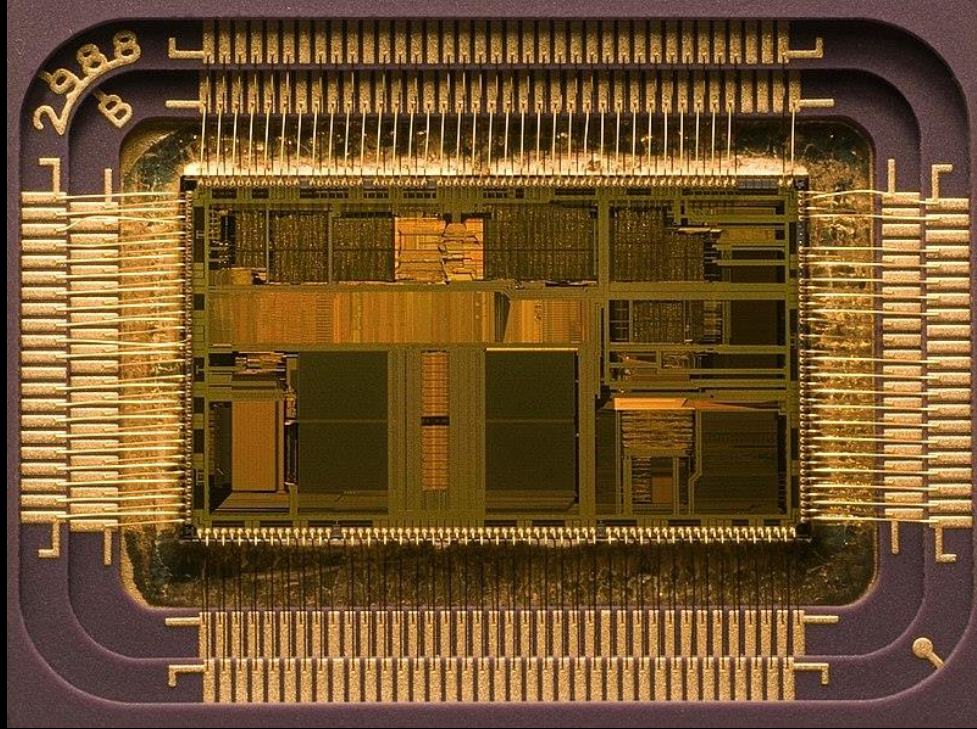
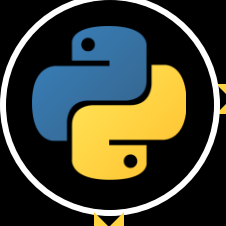


# Bilgisayar Mimarisi

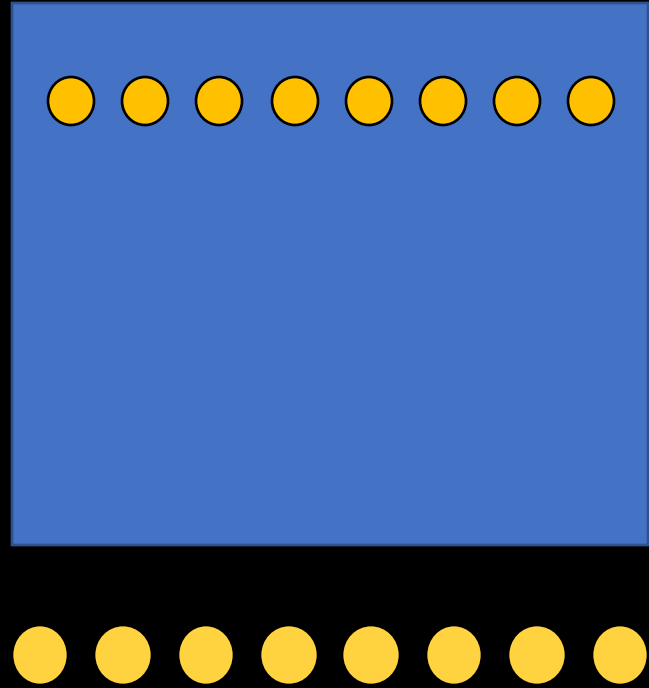
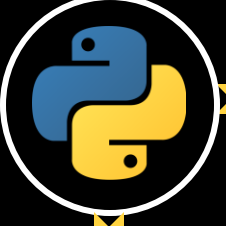




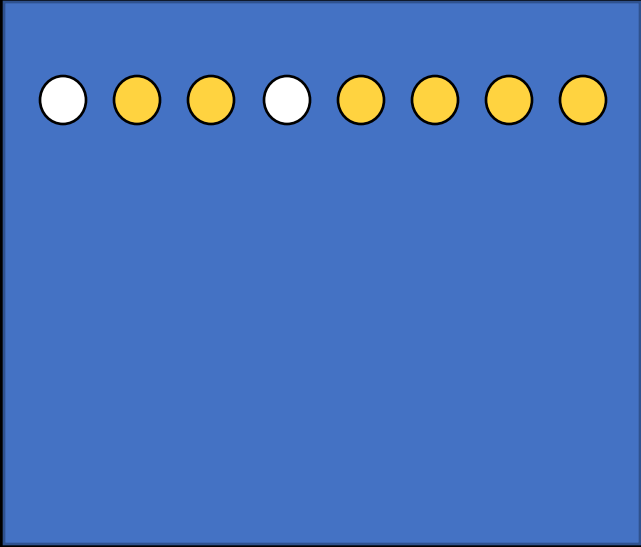
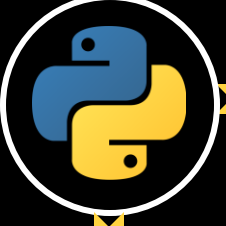
# Merkezi İşlem Birimi



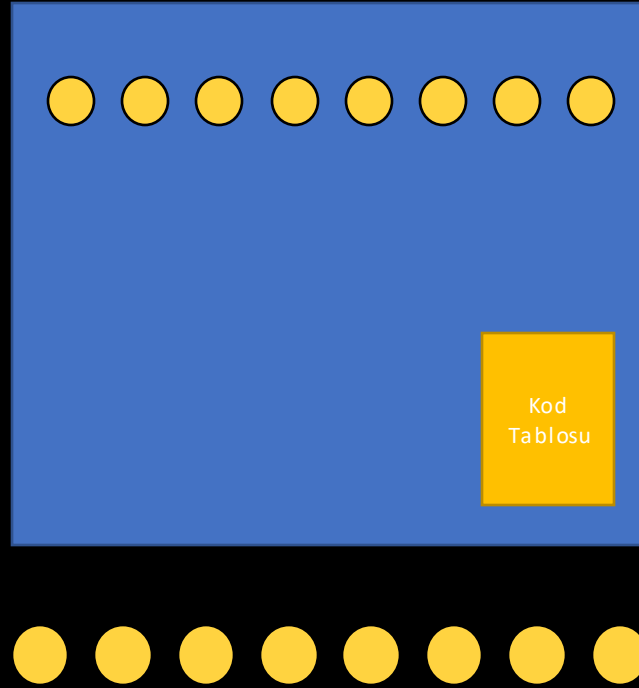
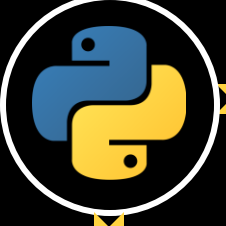
# Merkezi İşlem Birimi



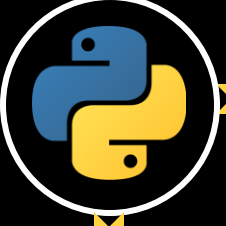
# Merkezi İşlem Birimi



# Merkezi İşlem Birimi



# Merkezi İşlem Birimi – Kod Tablosu



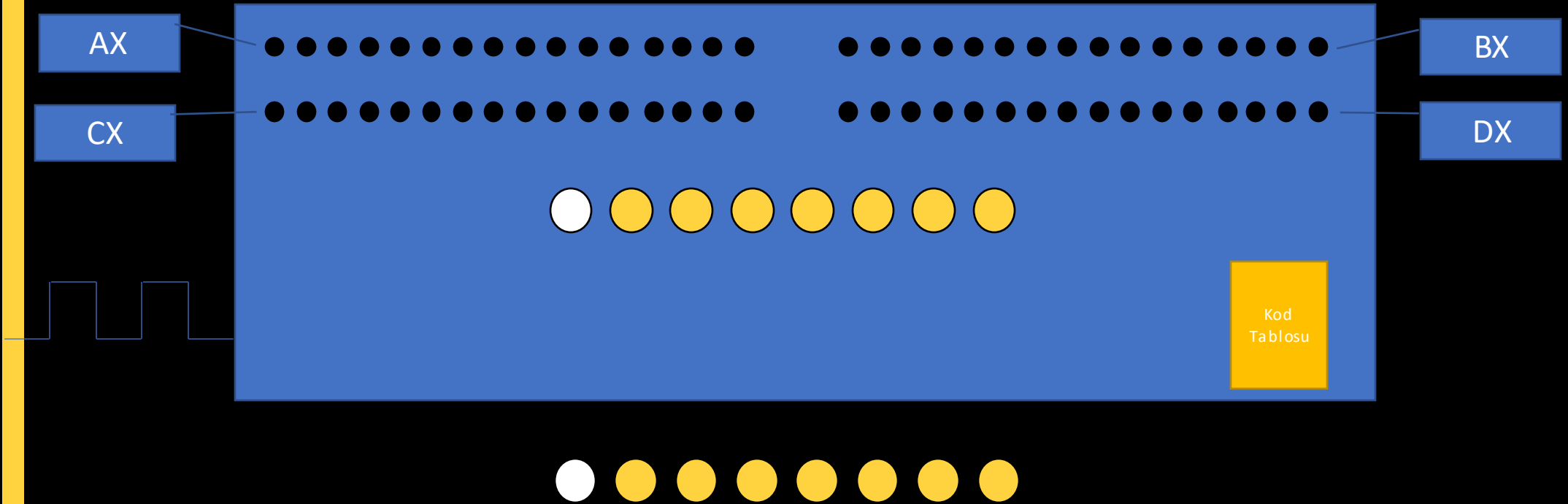
1000 0000	Bir sonraki sırada bir sayı var. AX kayıtçısına yerleştir.
1001 0000	Bir sonraki sırada bir sayı var. BX kayıtçısına yerleştir.
1011 0000	AX ve BX i topla, işlemi AX kayıtçısına yaz
1100 0000	Sonucu harici veri yoluna yaz
0000 0000	0
0000 0001	1
0000 0010	2
0000 0011	3

# Basit Bir Toplama

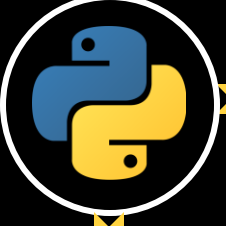


- $3+5 = ?$
- $3 = 0000\ 0011$
- $5 = 0000\ 0101$
- $8 = 0000\ 1000$

# Merkezi İşlem Birimi



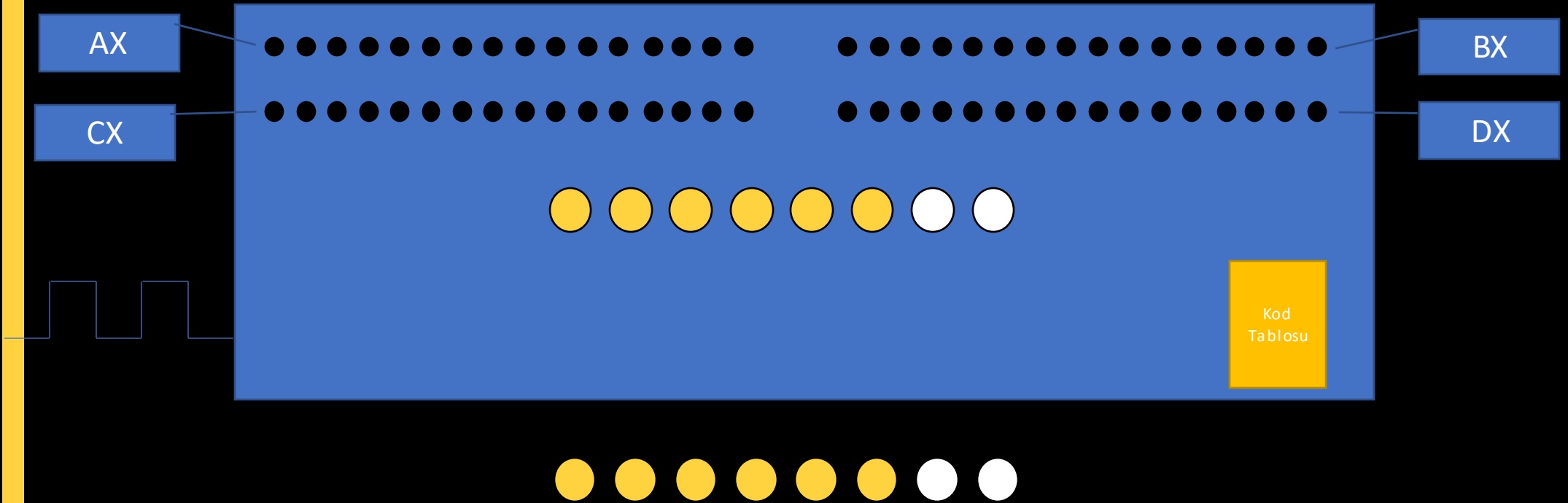
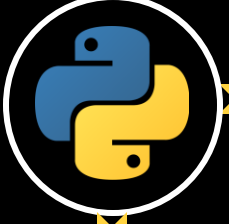
# Merkezi İşlem Birimi – Kod Tablosu



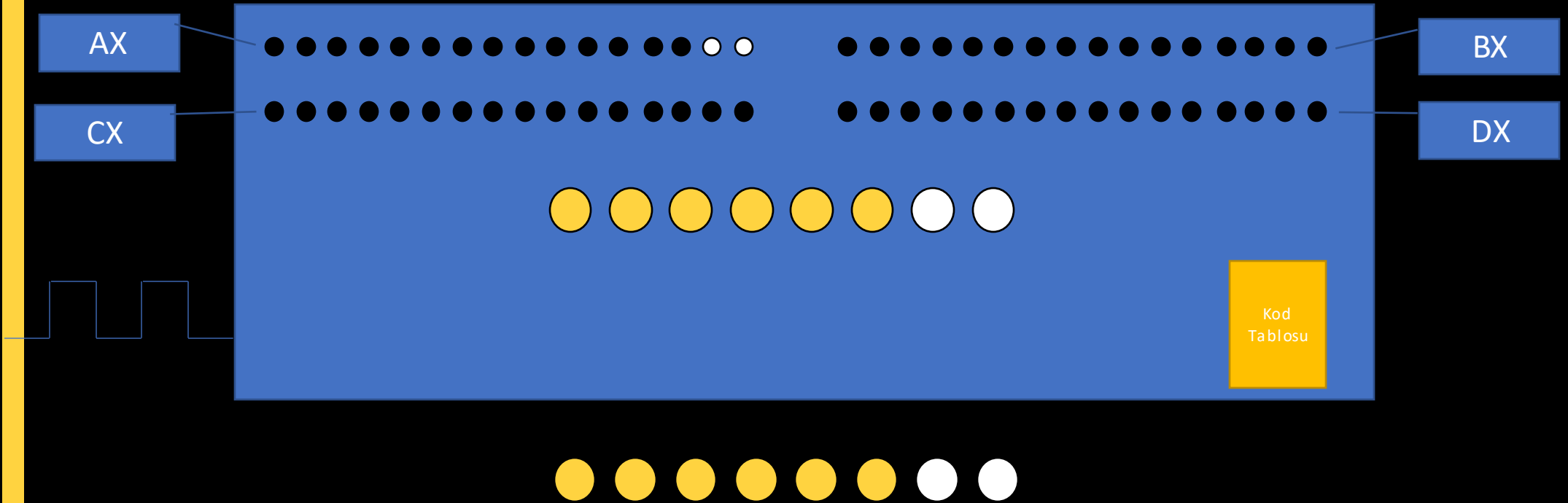
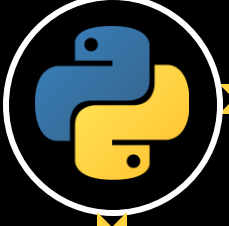
1000 0000	Bir sonraki sırada bir sayı var. AX kayıtçısına yerleştir.
1001 0000	Bir sonraki sırada bir sayı var. BX kayıtçısına yerleştir.
1011 0000	AX ve BX i topla, işlemi AX kayıtçısına yaz
1100 0000	Sonucu harici veri yoluna yaz
0000 0000	0
0000 0001	1
0000 0010	2
0000 0011	3



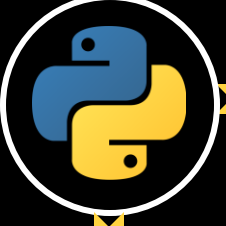
# Merkezi İşlem Birimi



# Merkezi İşlem Birimi

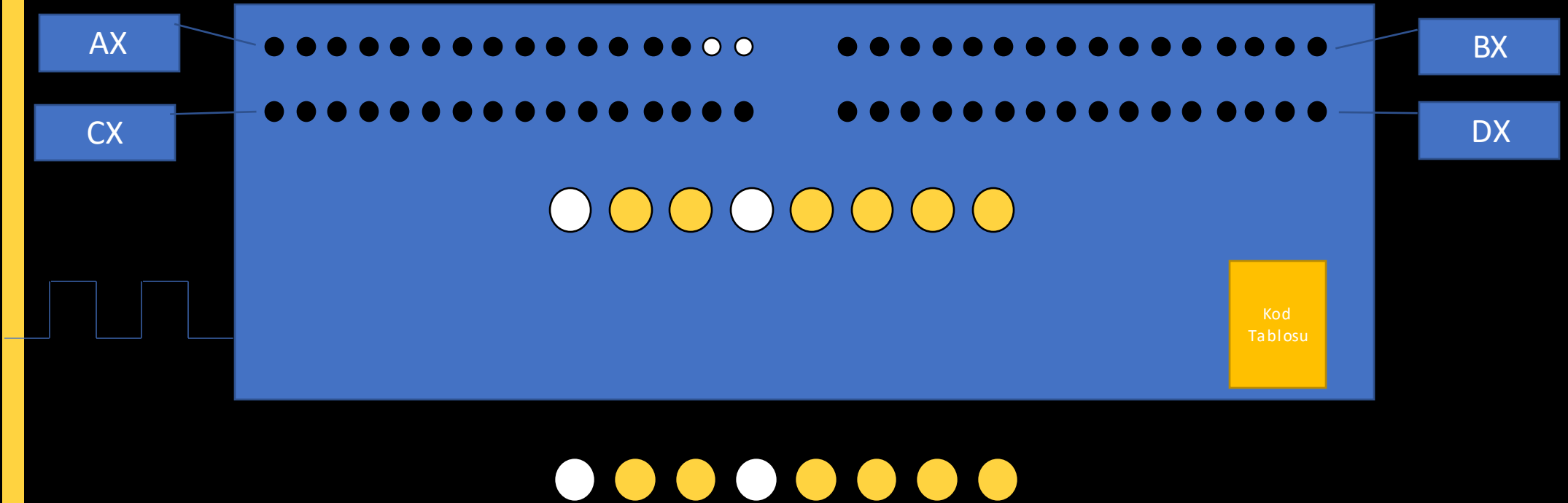


# Merkezi İşlem Birimi – Kod Tablosu

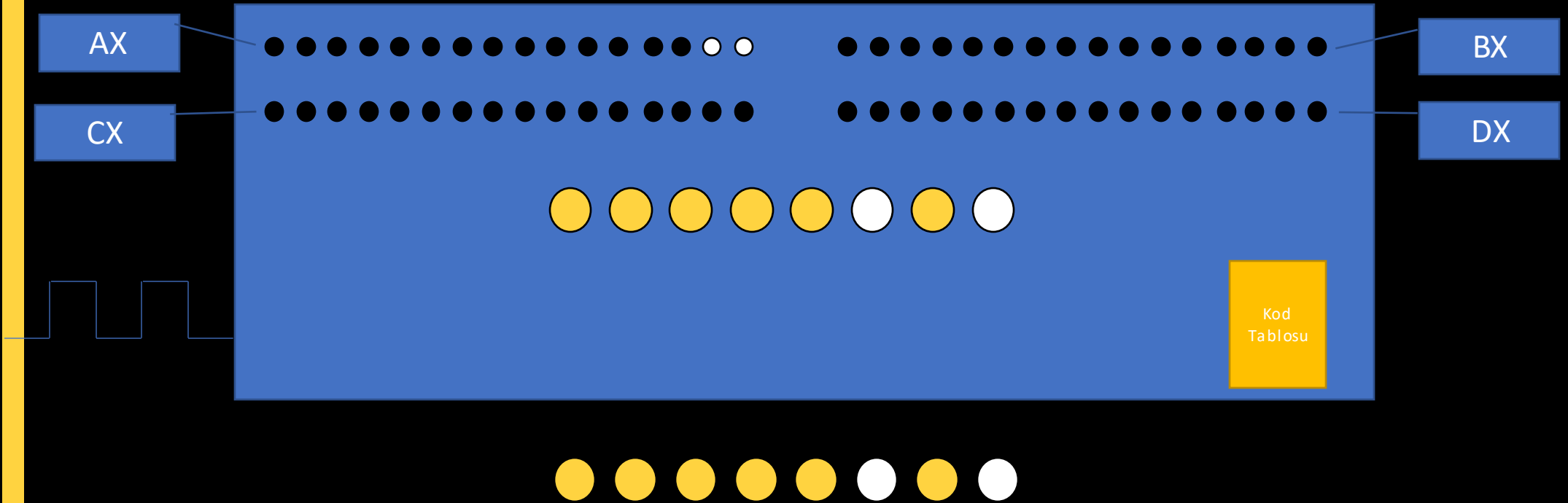


1000 0000	Bir sonraki sırada bir sayı var. AX kayıtçısına yerleştir.
1001 0000	Bir sonraki sırada bir sayı var. BX kayıtçısına yerleştir.
1011 0000	AX ve BX i topla, işlemi AX kayıtçısına yaz
1100 0000	Sonucu harici veri yoluna yaz
0000 0000	0
0000 0001	1
0000 0010	2
0000 0011	3

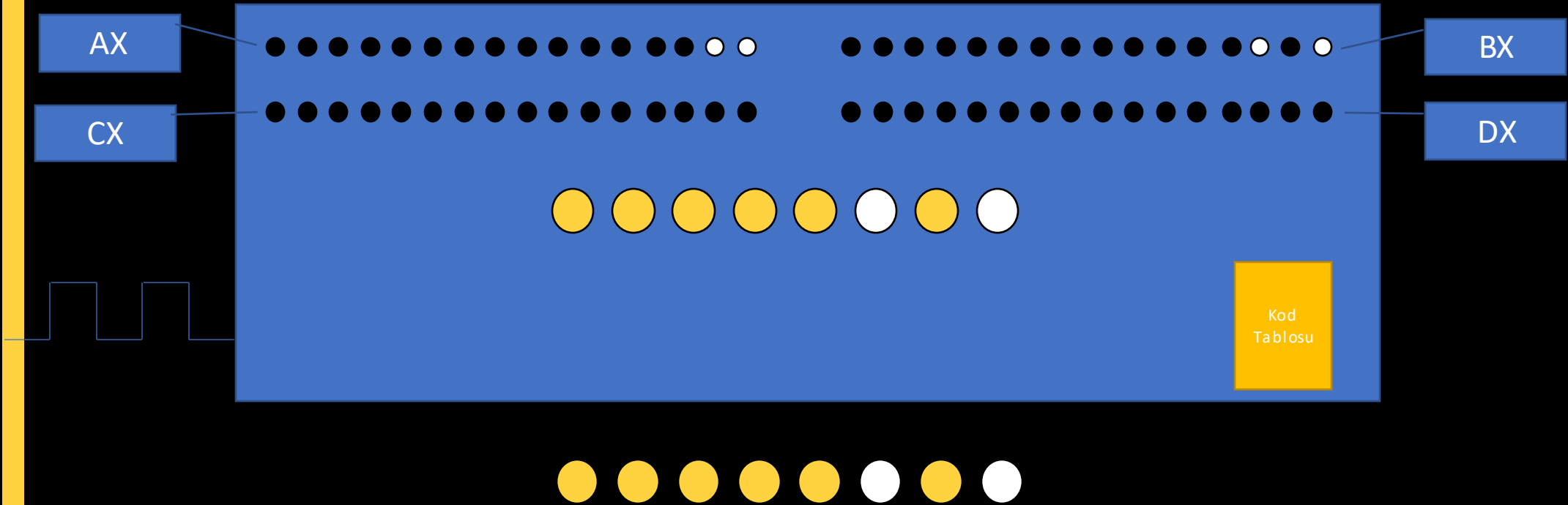
# Merkezi İşlem Birimi



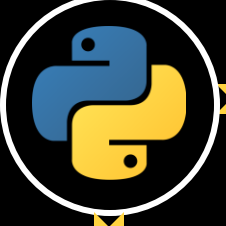
# Merkezi İşlem Birimi



# Merkezi İşlem Birimi

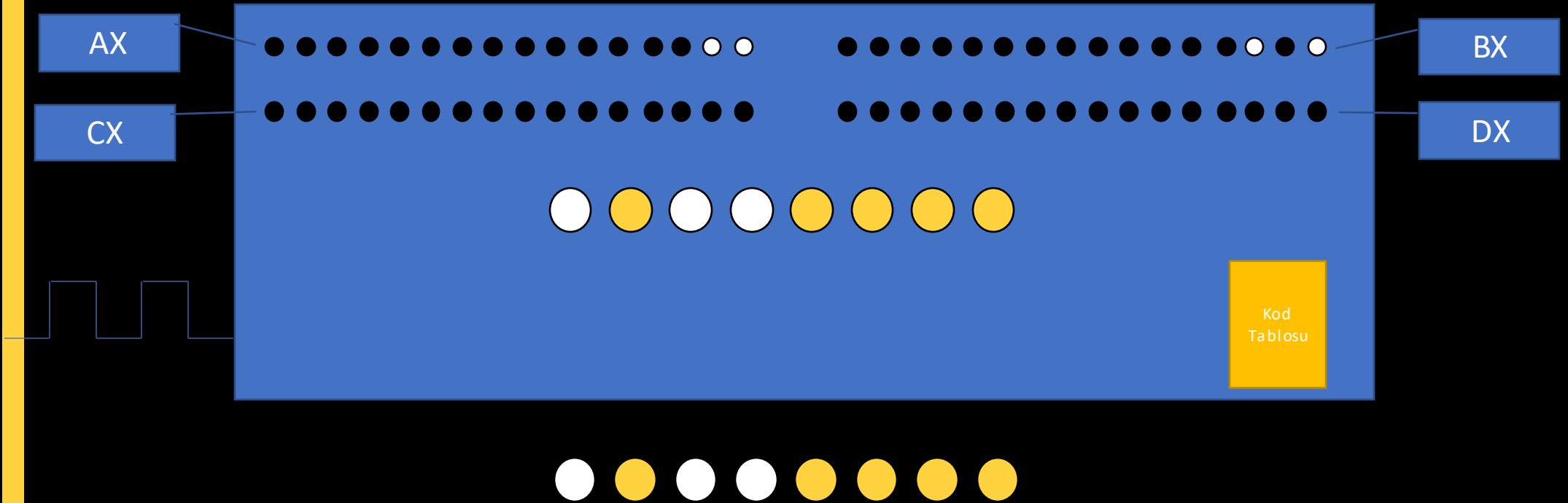


# Merkezi İşlem Birimi – Kod Tablosu



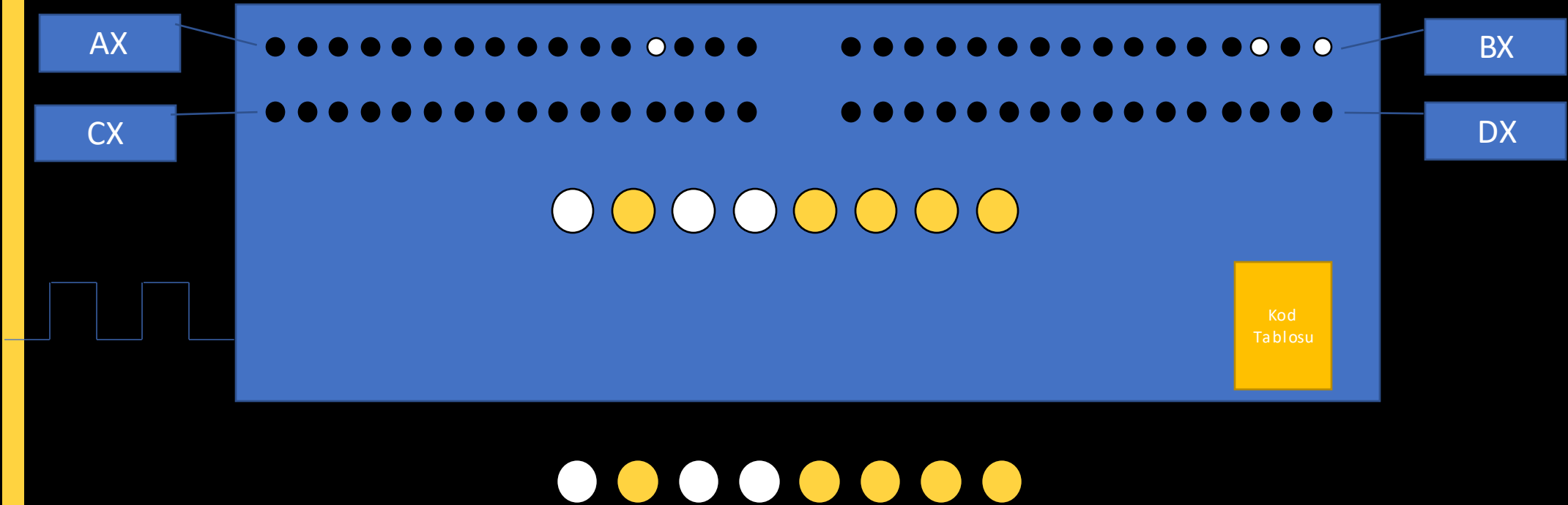
1000 0000	Bir sonraki sırada bir sayı var. AX kayıtçısına yerleştir.
1001 0000	Bir sonraki sırada bir sayı var. BX kayıtçısına yerleştir.
1011 0000	AX ve BX i topla, işlemi AX kayıtçısına yaz
1100 0000	Sonucu harici veri yoluna yaz
0000 0000	0
0000 0001	1
0000 0010	2
0000 0011	3

# Merkezi İşlem Birimi

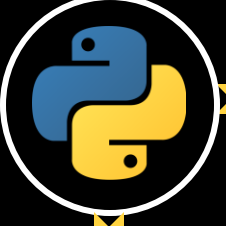




# Merkezi İşlem Birimi

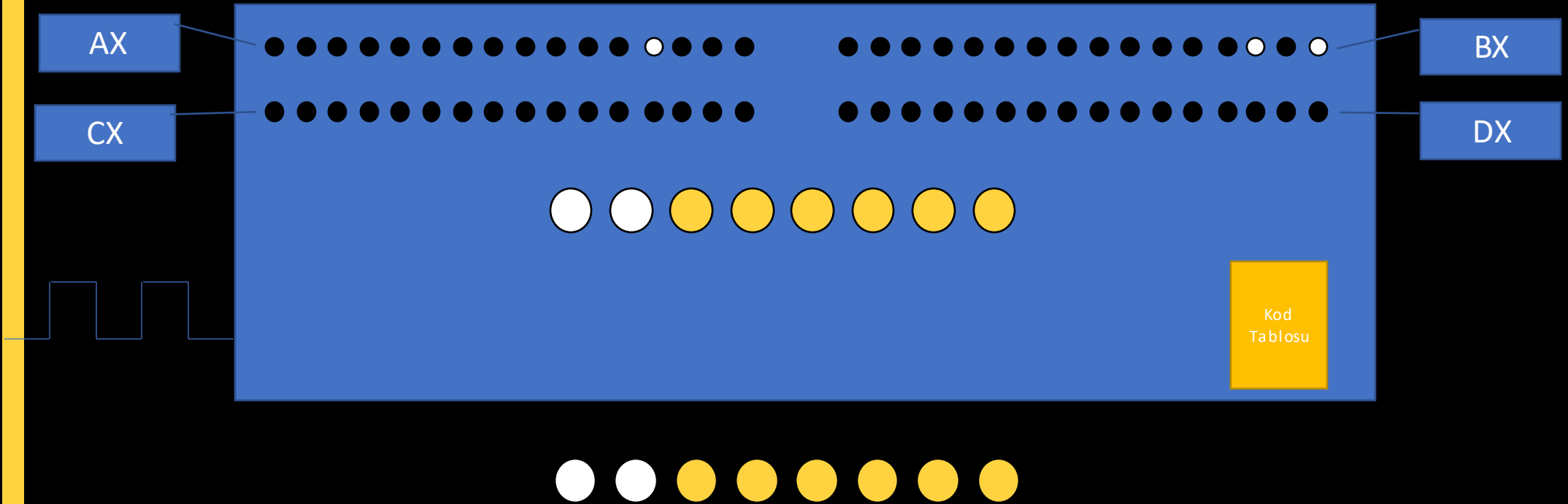


# Merkezi İşlem Birimi – Kod Tablosu

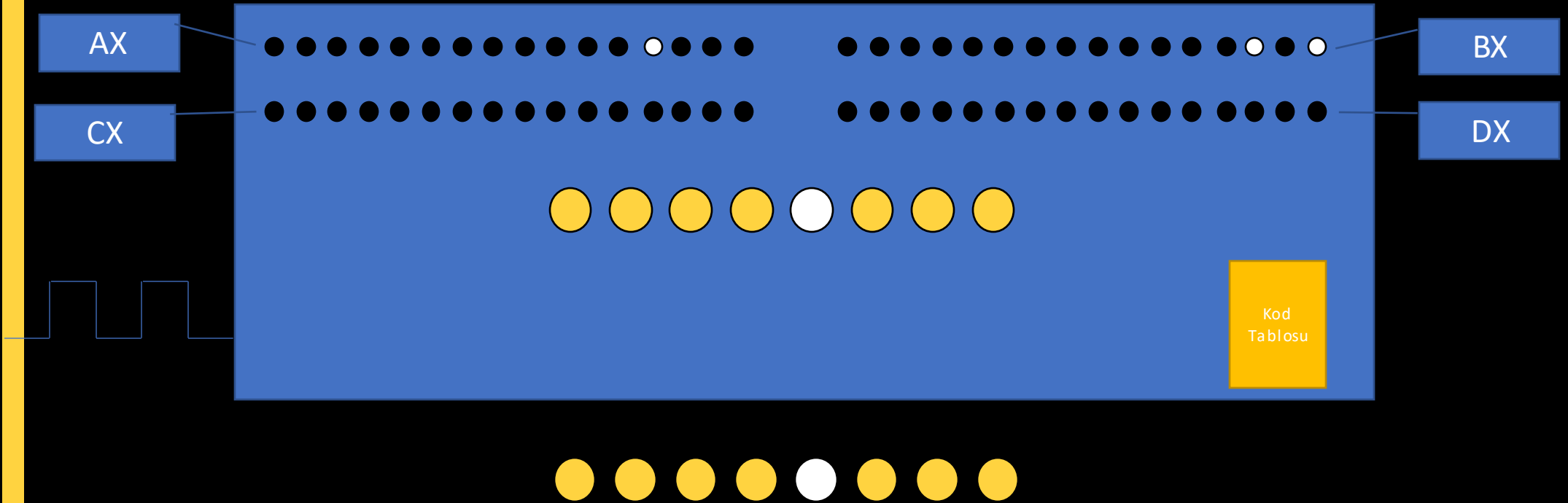


1000 0000	Bir sonraki sırada bir sayı var. AX kayıtçısına yerleştir.
1001 0000	Bir sonraki sırada bir sayı var. BX kayıtçısına yerleştir.
1011 0000	AX ve BX i topla, işlemi AX kayıtçısına yaz
1100 0000	Sonucu harici veri yoluna yaz
0000 0000	0
0000 0001	1
0000 0010	2
0000 0011	3

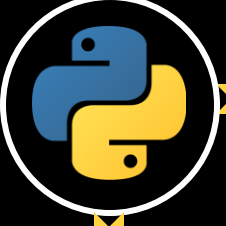
# Merkezi İşlem Birimi



# Merkezi İşlem Birimi

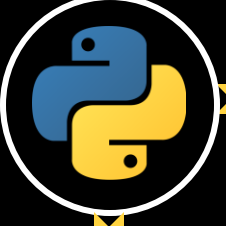


# Derleme - Yorumlama



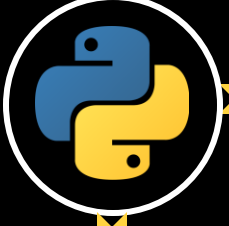
- DERLEME - kaynak program, makineyi içeren bir dosya (örneğin, kodun MS Windows altında çalıştırılması amaçlanıyorsa bir .exe dosyası) alınarak bir kez çevrilir (ancak, kaynak kodunu her değiştirdiğinizde bu işlem tekrarlanmalıdır) kod; artık dosyayı dünya çapında dağıtabilirsiniz; bu çeviriyi yapan programa derleyici veya çevirmen denir;

# Derleme - Yorumlama



- YORUMLAMA - siz (veya kodun herhangi bir kullanıcısı), her çalıştırılması gerektiğinde kaynak programı çevirebilirsiniz; Bu tür bir dönüşümü gerçekleştiren programa yorumlayıcı denir, çünkü her çalıştırılması amaçlandığında kodu yorumlar; bu aynı zamanda kaynak kodunu olduğu gibi dağıtamayacağınız anlamına da gelir, çünkü son kullanıcının onu yürütmek için yorumlayıcıya da ihtiyacı vardır.

# Yorumlayıcı Ne Yapar?



- Bir bilgisayar programı aslında bir metin parçasıdır, dolayısıyla kaynak kodu genellikle metin dosyalarına yerleştirilir.
- Tercüman kaynak kodunu Batı kültüründe yaygın olan bir şekilde okur: yukarıdan aşağıya ve soldan sağa.
- Yorumlayıcı, sonraki tüm satırların doğru olup olmadığını kontrol eder.
- Yorumlayıcı bir hata bulduğunda işini hemen bitirir. Bu durumda tek sonuç bir hata mesajıdır.
- Satır iyi görünüyorsa, yorumlayıcı onu çalıştırmaya çalışır.

# Derleyici ve Yorumlayıcı Arasındaki Farklar

	Derleyici	Yorumlayıcı
Avantaj	<ul style="list-style-type: none"><li>• çevrilmiş kodun yürütülmesi genellikle daha hızlıdır;</li><li>• yalnızca kullanıcının derleyiciye sahip olması gerekir - son kullanıcı kodu onsuz kullanabilir;</li><li>• çevrilen kod makine dili kullanılarak saklanır; anlaşılması çok zor olduğundan, kendi icatlarınız ve programlama hileleriniz büyük olasılıkla sırrınız olarak kalacaktır.</li></ul>	<ul style="list-style-type: none"><li>• kodu tamamlar tamamlamaz çalıştırabilirsiniz; çevirinin ek aşamaları yoktur;</li><li>• kod, makine dili değil, programlama dili kullanılarak saklanır; bu, kodun farklı makine dilleri kullanan bilgisayarlarda çalıştırılabileceği anlamına gelir; kodunuzu her farklı mimari için ayrı ayrı derlemezsiniz.</li></ul>
Dezavantaj	<ul style="list-style-type: none"><li>• derlemenin kendisi çok zaman alıcı bir süreç olabilir; değişiklik yaptıktan hemen sonra kodunuzu çalıştıramayabilirsiniz;</li><li>• kodunuzun çalıştırılmasını istediğiniz donanım platformu sayısı kadar derleyiciye sahip olmanız gerekir.</li></ul>	<ul style="list-style-type: none"><li>• Yorumlamanın kodunuzu yüksek hıza çıkarmasını beklemeyin; kodunuz bilgisayarın gücünü yorumlayıcıyla paylaşacaktır, dolayısıyla gerçekten hızlı olamaz;</li><li>• hem sizin hem de son kullanıcının kodunuzu çalıştıracak tercümana sahip olması gerekir.</li></ul>