



**KARADENİZ TEKNİK ÜNİVERSİTESİ**  
**Bilgisayar Mühendisliği Bölümü**  
**Sayısal Tasarım Laboratuvarı**



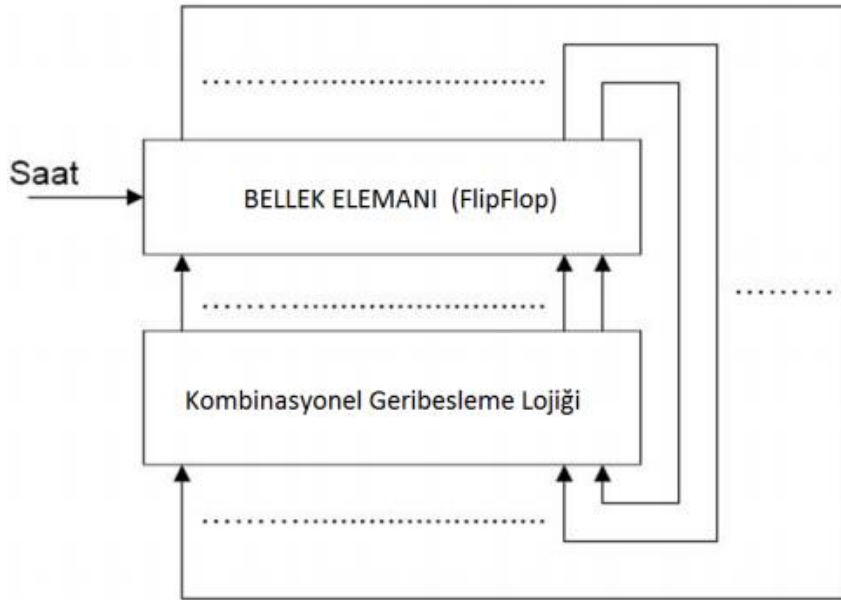
## VHDL ile SAYICILAR ve KAYDIRMALI KAYDEDİCİLER

### 1. Giriş

Geniş uygulama alanı olan sayıcılar, bir fiziksel olayda meydana gelen değişimleri veya geçişleri sayar. Mesela dönen bir cismin her dönüşünde oluşturduğu mekanik veya elektriksel darbeler uygun bir sayıcı yardımıyla sayılabilir. Elektriksel darbeleri sayan sayıcılar asenkron ve senkron olmak üzere genellikle iki sınıfa ayrılırlar. Bir FlipFlop dizisinden oluşan asenkron sayıcıda “taşımaya zamanı”, bütün FlipFlop’ların geçiş sürelerinin toplamı kadardır. Eğer sayma zinciri çok uzun ve taşımaya zamanı giriş darbeleri arasındaki süreden daha uzun ise, o zaman darbe aralıklarında sayıcıyı okumak mümkün olmayacaktır. Bu sorun senkron sayıcı kullanarak yok edilebilir. Senkron sayıcıda bütün çıkışlar saatle senkronlu olarak aynı anda durum değiştirir.

Şekil 1 ’de görüldüğü gibi, senkron ardışıl devre olan senkron sayıcı, bellek ve kombinyonel anahtarlama devresi olmak üzere iki kısımdan meydana gelir. Burada bellek, saatlenebilen Flip Flop’lardır. Kombinyonel devre ise sayıcının arzulanan sırada saymasını sağlayan geri besleme lojiğini meydana getirir.

İkili bellek elemanlarının sayısı  $n$  ise bu sayıcı ile en fazla  $2^n$  kadar durum sayılabilir. O halde  $m$  durum sayabilen bir sayıcı için  $2^{n-1} < m < 2^n$  olmak üzere  $n$  adet FlipFlop gerekir.  $m \neq 2$  olduğu zaman  $2^n$ ’nin kuvveti olmayan sayıcılar elde edilir, örneğin onlu sayıcı gibi.



Şekil 1: Senkron Ardışıl Devre Blok Diyagramı

## 2. SENKRON SAYICILARIN TASARIMI

Senkron sayıcılar saat modlu ardışıl devre tasarım yöntemleri kullanılarak tasarlanabilir. Tasarımda karşılaşılan en önemli sorun, ikinin kuvveti olmayan saymalarda, besleme gerilimi uygulandığı zaman FlipFlop'larda meydana gelen rastgele değerlerden sayma halkasına geçilememesidir. Kombinasyonel geri besleme lojiği uygun seçilerek bu sorun yok edilebilir.

Uygulamada daha ziyade sayma sırası kontrol edilebilen sayıcılar kullanılmaktadır. Kontrol girişindeki lojik seviyeye bağlı olarak sayma halkasında bulunan durumlardan bazıları atlanabilir veya ileri-geri sayıcılarda olduğu gibi sayma sırası değiştirilebilir.

Sayıcılarda devre çıkışları genellikle FlipFlop'ların çıkışlarıdır, yani çıkışlar seviyedir. O halde farklı çıkışların sayısı durumların sayısını belirleyeceğinden bu devrelerde durum sayısı veya diğer bir deyişle bellek elaman sayısı azaltılamaz. Bu yüzden ekonomiklik daha çok kombinasyonel kısımda düşünülebilir. Tasarım işleminde aşağıdaki adımların kullanılması tasarımı sistematik hale getirir: 1. Sayıcının verilen özelliklerinden faydalanarak durum diyagramı çizilir. 2. Durum tablosu kurulur. Mümkünse durum indirgemesi yapıldıktan sonra bellek elemanlarının sayısı belirlenir. 3. Durum geçiş tablosu kurulur ve Flip Flop türü belirlenir. 4. Flip Flop'ların veri girişlerine uygulanacak işlevler en az biçimde elde edilir.

## 3. DENEYE HAZIRLIK

- 4-Bitlik Gray kod sayıcıyı D FlipFlop'larıyla tasarlayınız.
- 8-bitlik bir sayıyı 3 bit sola kaydırmakla ne yapmış olunur.

## 4. DENEYİN YAPILIŞI

- Aşağıda verilen 8-bit sayıcı devresini kodlayınız ve bu devreye Reset girişini ekleyiniz?

```
-----  
-- 8- Bit sayıcı devresi  
-- senkron reset girişi  
-- process'te degisken kullanımı  
-----  
Library IEEE;  
Use IEEE.std-logic-1164.all;  
Use IEEE.STD-LOGIC-UNSIGNED.ALL;  
Use IEEE.STD-LOGIC-ARITH.ALL;  
Entity eSayıcı is  
    Port( s: in std-logic;  
          F: out std-logic-vector(7 downto 0) );  
End eSayıcı;  
  
Architecture behv of eSayıcı is  
Begin  
  
-- Sayac degiskenini artıran işlev  
pSayıcı: Process( s )  
    Variable Sayac:
```

```

std_logic_vector(7 downto 0):="00000000";
Begin
  If ( s'event AND s = '1' ) then -- yükselen kenr tespiti
    Sayac := Sayac + 1;
  End if;
End Process;
  F <= Sayac; -- cikisa aktar
End behv;
-----

```

- Bu devreyi 50MHz olan FPGA kartında her saniyede bir artacak şekilde ayarlayınız.
  - Sayac bilgisini FPGA kartındaki LED lere aktarınız.
  - VHDL dilinde ulama & ekleme sembolü kullanılarak kaydırmalı kaydediciler tasarlanabilir
- Örneğin, 8-bitlik Y8 saklayıcısını bir bit sola kaydırmak için yazmamız gereken VHDL ifadesi aşağıda verilmiştir:

```

Y8 <= Y8(6 downto 0) & "0";

```

```

-----
-- Sola kaydırmalı kaydedici
-- Savran
-----
Library IEEE;
Use IEEE.std_logic_1164.all;
Entity eSolaKaydir is
Port ( s, y : in std_logic; --- y yukle
      A: in std_logic_vector(7 downto 0);
      F: out std_logic_vector(7 downto 0) );
End eSolaKaydir;
Architecture Behv of eSolaKaydir is
Signal Y8 : std_logic_vector(7 downto 0) := "ZZZZZZZZ";
Begin
  Process( s, y )
  Begin
    If ( y = '1' ) then
      Y8 <= A;
    ElseIf ( Rising_edge (s)) then
      Y8 <= Y8(6 downto 0) & "0";
    End if;
  End Process;
  F <= Y8; -- eszamanli atama
End Behv;

```