

**EEM412 SON SINAVI, 13 SORU, 110 Dakika**

**S.1.** (5p) Bir mikroişlemci tasarımı için Davranışsal (Behavioral), Yapısal (Structural) ve Fiziksel (Physical) alanlara (Domain) örnekler veriniz. Her alan (domain) için İKİ ADET örnek verilmelidir.

**ÇÖZÜM:**

*Davranışsal: Ethernet üzerinde çalışır. Logaritma işlemi yapar.*

*Yapısal: Veri için 128KB'ya Önbellek bulunur. Komut yolu ile Veri yolu ayrılır.*

*Fiziksel: 40nm CMOS 8-metal kullanılır. Domino-logic hücre kütüphanesi kullanılır.*

**S.2.** (10p) İkinci dereceden bir polinomun köklerini bulabilmek için

$$y_1 = -b - (b^2 - 4ac)^{0.5} / 2a$$

$$y_2 = -b + (b^2 - 4ac)^{0.5} / 2a$$

işlemlerini yapacak bir IP çekirdek (core) tasarlanacaktır. Bu tasarım için probleme yapısal tasarım yöntemleri olan Hiyerarşi, Regularity, Locality, Modularity uygulayıp her bir yöntemi BİR ADET örnek ile açıklayınız.

**ÇÖZÜM:**

*Hiyerarşi: İşlemci Bloğu: Çarpma, Bölme, Toplama, Çıkarma, Karekök alma blokları Kontrol bloğu: ROM bellek, FSM*

*Regularity: Çarpma bloğunun birden çok yerde kullanımı.*

*Locality: Çarpma bloğu sembol ile gösterilir. İki adet 32-bit'lik girişler, saat girişi, 64-bit çıkış bulunur.*

*Modularity: Çarpma devresinin giriş ve çıkış sinyalleri, yerleri ve yapısı.*

**S.3.** (5p) Aşağıdaki çizelgede tümdevre tasarım gerçekleştirme seçeneklerine karşılık performanslarını, 1(düşük), 2, 3, ve 4 (yüksek) sayıları ile çizelgede notlandırınız. Bir satırda aynı sayıyı kullanmayınız.

IC seçeneği: Performans:	Özel (Custom)	FPGA	Kapı Dizileri (Gate Arrays)	Hücresel (Cell Based)
Yonga (die) Alanı	1	4	3	2
Çalışma Frekansı	4	1	2	3
Güç Tüketimi	1	4	3	2
Tasarım Süresi	4	1	2	3
Üretilen IC Sayısı	4	1	2	3
Üretim Verimi (Yield)	4	1	2	3
Tasarımda Esneklik	1	4	3	2

**S.4.** (5p) Aşağıda tasarlanacak ARY adlı parçanın giriş ve çıkış sinyal adları tanımlanmıştır. DRM adlı çıkış sinyalinin parça içinde tekrar kullanılacağı bilinmektedir. Buna göre parçanın VHDL ENTITY bölümünü yazınız.



**ÇÖZÜM:**

*Entity s4 is*

```
Port (
    ale : in std_logic;
    clk : in std_logic;
    d : out std_logic_vector(7 downto 0);
    drm : buffer std_logic
```

```
);
```

*End Entity;*

**S.5.** (10p) Eşzamansız sıfırlanabilir, yukarı ve aşağı sayabilen 16-bit sayıcının VHDL ile tasarımını yapınız. VHDL içinde yalnızca ARCHITECTURE bölümünü yazınız ve “For loop” ya da “while” kullanmayınız. Girişler: RESET, UPDOWN, CLK. Çıkış (16-bit): CNT.

ÇÖZÜM:

*Architecture beh of s5 is*

*Begin*

*counter: process(clk, reset, updown)*

*variable count : std\_logic\_vector(15 downto 0);*

*Begin*

*if reset = '1' then count := (others => '0');*

*elsif clk'event and clk = '1' then*

*if updown = '1' then count := count + 1;*

*else count := count - 1;*

*end if;*

*end if;*

*cnt <= count;*

*End process;*

*End Architecture;*

**S.6.** (10p) 15 adet DFF'ten oluşan, 8. ve 15. DFF çıkışları ExOR kapısından geçirilen bir PRBS üretici VHDL ile tasarlayınız. Sağdan sola öteleme için aşağıdaki örnek VHDL satırını kullanınız:

*shift\_reg <= shift\_reg(13 downto 0) & LSB\_in*

ÇÖZÜM:

*Library ieee;*

*Use ieee.std\_logic\_1164.all;*

*Use ieee.std\_logic\_unsigned.all;*

*Entity s6 is*

*Port ( clk : in std\_logic;*

*reset : in std\_logic;*

*prbs : buffer std\_logic*

*);*

*End Entity;*

*Architecture beh of s6 is*

*signal xorgate : std\_logic;*

*signal shift\_reg : std\_logic\_vector(14 downto 0);*

*Begin*

*xorgate <= shift\_reg(7) XOR shift\_reg(14);*

*prbs <= shift\_reg(14);*

*shifter: process(clk, reset, shift\_reg)*

*Begin*

*if reset = '1' then shift\_reg <= (others => '1');*

*elsif clk'event and clk = '1' then*

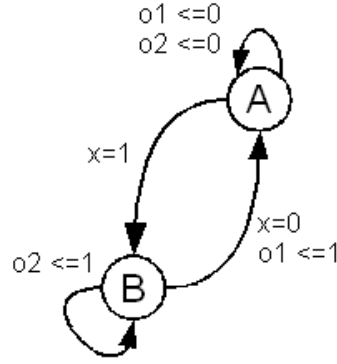
*shift\_reg <= shift\_reg(13 downto 0) & xorgate;*

*end if;*

*End process;*

*End Architecture;*

S.7. (10p) Aşağıdaki FSM için VHDL ile ARCHITECTURE tasarımını yapınız.



ÇÖZÜM:

Architecture beh of s7 is

type state is (A, B);  
signal states : state;

Begin

FSM: process(clk, reset, x, states)

Begin

```

if reset = '1' then states <= A;
elsif clk'event and clk = '1' then
  case states is
    when A =>
      o1 <= '0';
      o2 <= '0';
      if x = '1' then states <= B;
      else states <= A;
      end if;

    when B =>
      o2 <= '1';
      if x = '0' then states <= A;
      else states <= B;
      end if;
  end case;
end if;

```

when others => states <= A;  
end case;

end if;

End process;

End Architecture;

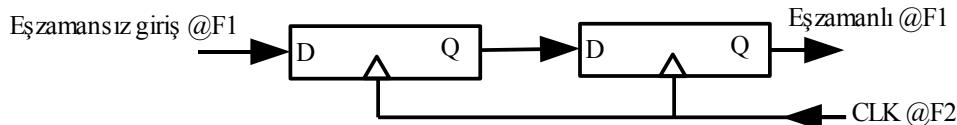
S.8. (5p) “MTBF” neyin kısaltmasıdır? F1 frekanslı eşzamansız sinyal girişini F2 frekansında çalışan devre için eşzamanlı sinyal durumuna getirmek için bir devre çiziniz. MTBF değerinin F1 ve F2 ile ilişkisini belirtiniz.

ÇÖZÜM:

“MTBF” = “Mean Time Between Failure”.

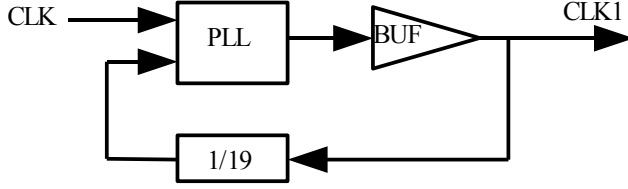
MTBF düşük olması için F1 ve F2 düşük olmalı. Ancak  $F2 > F1$  de olmalı.

İstenilen devre eşzamanlayıcı devresidir:



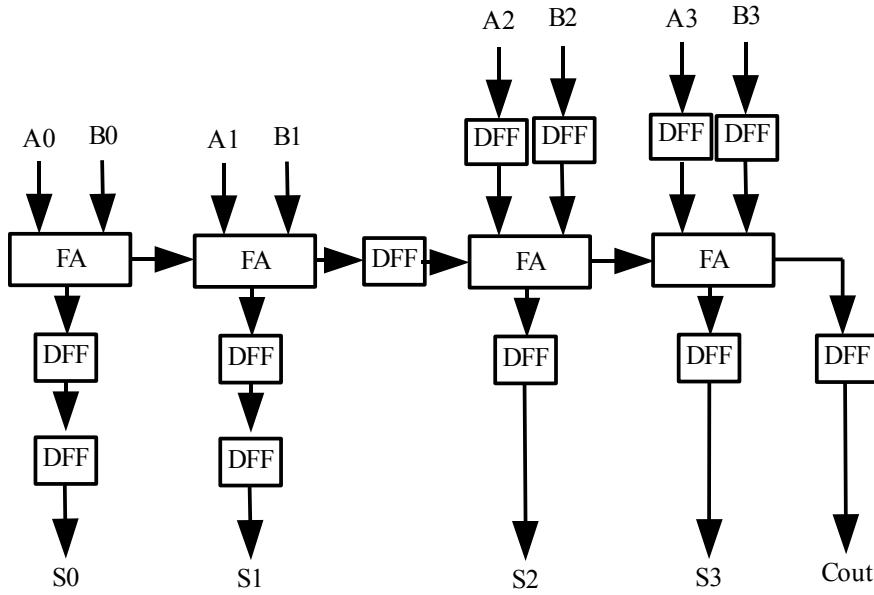
**S.9.** (10p) Bir tümdevreye CLK adlı saat sinyali gelmekte ve tümdevre içinde CLK1 adlı saat “clock buffer” ile dağıtılıp kullanılmak isteniyor. CLK1 frekansı CLK'nın 19 katı olmalı ve tümdevre çıkış sinyalleri CLK1 saat sinyali ile faz farksız olmalıdır. Bu koşulları sağlayacak devre parçalarını tümdevre içine yerleştirip bağlantılarını yönleri ile birlikte gösteriniz.

**ÇÖZÜM:**



**S.10.** (10p) 4-bit A ve 4-bit B sayılarını toplayan bir “ripple adder” yapısına 2 kademeli (2 periyod gecikmeli) “pipeline” yapı uygulayıp FA ve DFF blokları kullanarak çiziniz, bloklar arası giriş ve çıkış sinyallerinin yönlerini belirtiniz.

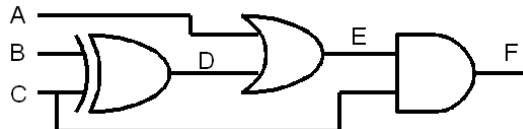
**ÇÖZÜM:**



**S.11.** (5p) “3-port” üzerinden erişim yapılabilen bir bellek hücresi çizip her sinyali adlandırın.

**ÇÖZÜM:**

**S.12.** (5p) Aşağıdaki devrenin D teli üzerinde SA0 ve SA1 hatalarını belirleyebilecek iki adet test vektörü bulunuz. Vektör elemanlarında gerekirse X (don't care) kullanınız.



**ÇÖZÜM:**

*SA0 için:  $[A,B,C] = [0\ 0\ 1]$*

*SA1 için:  $[A,B,C] = [0\ 1\ 1]$*

**S.13.** (10p) 32-bit A sayısı ile 32-bit B sayısını çarpan bir çarpıcı bloğunu test etmek için PRBS üreteç ve “signature analyzer” kullanıp bağlantılarını yönleri ve hat sayılarıyla birlikte gösteriniz. Kullanılan PRBS ve “signature analyzer” blok içlerini RTL düzeyinde çiziniz.