

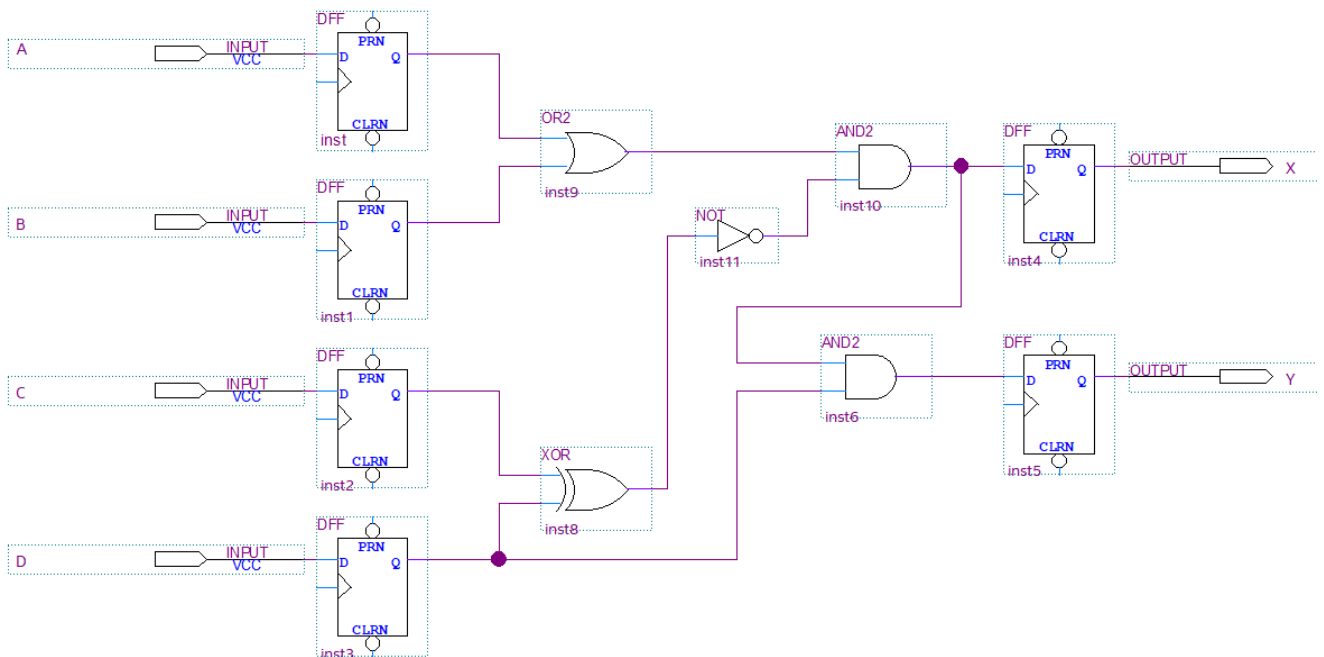
# ELM 234 - Ödev #2



- Problemlerde gidiş yolunuzu ve yönteminizi gösterin.
  - Soruları arkadaşlarınızla tartışın fakat kendiniz çözün ve cevaplarınızı başka öğrencilerle paylaşmayın.
  - Cevaplarınızı **tek bir PDF dosyası** olarak teslim edin. Başka formatlar **kabul edilmeyecektir**.
  - HDL yazmanız istenilen sorularda oluşturduğunuz kodları **düzgün formatlı şekilde** (Consolas veya Courier gibi monospaced fontlar kullanarak 12 pt. şekilde tek satır) ödevinize ekleyin. Örnek format ödevin sonuna mevcuttur. Simülasyon sonuçlarınızın ekran görüntüsünü ekleyin.

**Problem 1. Dinamik Disiplin ve Pipeline** [4 + 10 + 4 + 6 + 12 = 36 puan]

- A. Şekil 1 de verilen devrenin bütün clock girişlerini aynı clocka bağlandığını varsayalım. Tablo 1 de verilen değerlere göre devrenin maximum frekansını bulunuz. Setup ve Hold zamanlamalarına uymuyorsa ona göre gerekli yere ekstra lojik ekleyip, devreyi tekrar analiz ediniz.  $t_{ccq} = 30\text{ps}$  ve  $t_{pcq} = 60\text{ps}$ .
- B. A şıkkındaki devrenizi HDL kullanarak tasarlayın, bir testbench devresi oluşturun ve Modelsim de simüle edin. Simülasyonda propagation zaman atamalarını yapmayı unutmayın. Quartus'da bu devreyi sentezleyerek bu devrede ulaşabileceğiniz max frekansını bulun



### Şekil 1

- C. Şekil 1 de verilen devreyi Clock Skew = 15ps olduğunu varsayarak tekrar analiz ediniz. maximum çalışma frekansını bulunuz. Setup ve Hold zamanlamalarına uymuyorsa ona göre gerekli yere ekstra lojik ekleyip, devreyi tekrar analiz ediniz.
- D. Şekil 1 de verilen devrenin **C şıkkındaki verilen clock skew değerine ve çözümünüze göre**, latency ve throughputunu hesaplayınız. (Eğer C şıkkını çözemedi iseniz bunu açıkça belirtip clock skew değerini bu soru için 10ps alabilirsiniz) Max frekansını arttırmak için en uygun yere pipeline ekleyip, tekrar hesaplamalarınızı yapınız. Throughput, latency, max freq, ve setup ve hold zamanlamalarına uyduğunu gösterin. Pipeline sayısı kendiniz seçebilirsiniz.
- E. D şıkkındaki devrenizi HDL kullanarak tasarlayın, bir testbench devresi oluşturun ve Modelsim de simüle edin. Simülasyonda propagation zaman atamalarını yapmayı unutmayın. Quartus'da bu devreyi sentezleyerek bu devrede ulaşabileceğiniz max frekansını bulun. B şıkkındaki devrenizle karşılaştırın.

Kapı	$t_{pd}$ (ps)	$t_{cd}$ (ps)
NOT	35	30
BUFFER	40	35
2-giriş AND	50	50
2-giriş XOR	80	80
2-giriş OR	55	55
3-giriş OR	70	70
all others	90	90
SETUP	45	
HOLD	90	

Tablo 1

## Problem 2. Synchronizer [5 + 5 = 10 puan]

Dışarı bağlanan bir butonu örneklemek için iki FF li bir synchronizer tasarladınız. Gerekliyse Setup ve Hold zamanlarını Tablo 1 deki değerlere göre alabilirsiniz.  $T_0 = 15$  ps,  $t_{hao} = 25$  ps

- A. 10 yıllık bir MTFB (Mean Time Between Failures) için devrenin maximum çalışabileceği frekansı bulunuz.
- B. Frekansı artırmak için **bir FF daha** eklediniz. Aynı MTFB için yeni frekans değerini hesaplayınız.

### Problem 3. NZVC Hesapları [5 + 10 + 15 = 30 puan]

A. Tablo 2 de verilen 16-bit operasyonlara göre sonuçları ve NZVC bayraklarını hesaplayınız.

	operasyon
a	0x1F32 - 0xFF02
b	0x0010 - 0x1F01
c	0xF801 + 0x7000
d	0xFE02 - 0x00DF
e	0x3D20 + 0xF5D0

Tablo 2

B. ALU devresi sonuçlarını gösteren **bu NZVC bayrakları baz alarak unsigned** HS, LS, HI, ve LO çıkışlarını gösteren devrelerin boolean denklemlerini bulun. Devre girişlerinizi 16-bit olarak alın.

- HS =  $A \geq B$  (A, B ye eşit veya daha büyük. High/Same)
- LS =  $A \leq B$  (A, B ye eşit veya daha küçük. Low/Same)
- HI =  $A > B$  (A, B den daha büyük. High)
- LO =  $A < B$  (A, B den daha küçük. Low)

C. HDL kullanarak tasarlayın, bir testbench devresi oluşturun ve Modelsim de simüle edin. Bunun için A,B girişlerinden NZVC bayraklarını generate eden bir devre tasarlayıp, testbench devrenizde bu A, B girişlerine değer gönderin.

### Problem 4. Multiplier tasarımı [12 + 12 = 24 puan]

A. AND kapıları ve full adder lar kullanarak iki unsigned 5-bitlik binary sayıyı çarpan bir devre tasarlayınız. Bu devrenin propagation ve contamination gecikmelerini Tablo 1 de verilen değerlere göre hesaplayınız.

B. HDL kullanarak bu devreyi tasarlayın, bir testbench devresi oluşturun ve Modelsim de simüle edin. Doğrulamayı otomatik olarak \* operatörü kullanan ayrı bir devre ile karşılaştırın.

- Quartus'da bu devreyi sentezleyerek bu devrede ulaşabileceğiniz max frekansını bulun. Bunun için giriş ve çıkışlara register bağlayabilirsiniz.
- Devrenizi, \* operatörü kullanan ayrı bir devre ile sonuçlarınızı karşılaştırın. (i.e. assign c = a \* b;), max freq, utilization.

## Örnek kod formatı:

Tablo içerisinde monospace font kullanarak (12pt, Consolas)

```
/*  
 * mult.sv  
 */  
  
module mult #(parameter N = 5) (  
    input  logic [N-1:0]  a, b,  
    output logic [2*N-1:0] s  
);  
  
assign s = a * b;  
  
endmodule
```