

# EEM212 - SAYISAL DEVRE TASARIMI DERS NOTLARI

## *DERS NOTU 13: ZAMAN ANALİZİ*

Dr. İsmail Öztürk \*

<ismail.ozturk@amasya.edu.tr>

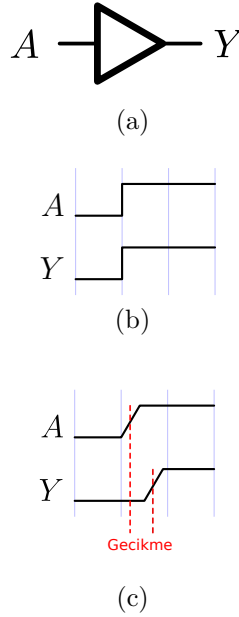
### İçindekiler

<b>1 Giriş</b>	<b>1</b>
<b>2 Kombinasyonel Devrelerde Zamanlama</b>	<b>3</b>
2.1 Glitch'ler . . . . .	5
<b>3 Ardışıl Devrelerde Zamanlama</b>	<b>6</b>
3.1 Çalışma Frekansı . . . . .	7
3.2 Paralleleştirme . . . . .	10

## 1 Giriş

Bundan önceki notlarda tasarım sırasında üzerinde durduğumuz olan en önemli hususlar güç tüketimi ve maliyetti. Kurduğumuz devrelerde güç tüketimi ve maliyeti düşürmek için devre sadeleştirmesinin nasıl yapılacağını görmüş ve tasarladığımız bütün devreleri sadeleştirme yaparak kurmuştuk. Bu kısımda ise, dijital devre tasarımında önemli olan bir diğer kriter olan hız unsurunun nasıl göz önünde bulundurulacağını ve devreyi nasıl daha hızlı çalışacak şekilde tasarlayabileceğimizi göreceğiz. Bunun için tasarladığımız devrelerde zaman analizi yapmamız gerekir.

\* Amasya Üniversitesi Teknoloji Fakültesi EEM Bölümü  
Daha fazla bilgi için: <https://iozturk.com>

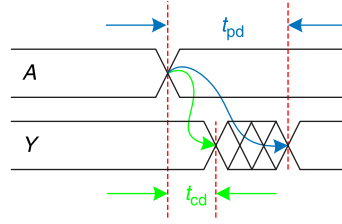


Şekil 1: (a) TAMPON kapısı; (b) ideal zaman diyagramı; (c) gerçek zaman diyagramı.

Zaman analizi temel olarak devre girişlerinde yapılacak değişikliklerin çıkışlara ne kadar sürede etki edeceğini incelemektir. Daha önce görmüş olduğumuz zaman diyagramlarında hep girişler çıkışlara anında etki ediyormuş gibi çizim yapmıştık. Mesela, Şekil 1 (a)'daki TAMPON kapısının ideal zaman diyagramı Şekil 1 (b)'deki gibidir. Görüldüğü üzere  $A$  girişinde yapılan bir değişiklik hemen  $Y$  çıkışına yansımaktadır. Fakat, gerçekte ne lojik seviye anında 0 değerinden 1 değerine çıkmakta; ne de girişteki değişiklik çıkışa anında yansımaktadır. Gerçek zaman diyagramı Şekil 1 (c)'deki gibi olmalıdır.

Şekil 1 (c)'den görülebileceği üzere bir girişi lojik 0 (L) seviyesinden lojik 1 (H) seviyesine geçirmek için bir süre gerekir. Bu süre, zaman diyagramında yapılan değişikliklerin dikey şekilde değil de eğimli bir şekilde gösterilmesini gerektirir. Mantık kapısı, ancak bu eğimli değişimde giriş sinyali lojik 1 kabul edilebilecek seviyeye ulaştığında girişin 1 olduğunu kabul eder. Bu noktadan sonra çıkışın da 1 olması gerekir. Fakat, şekilden görebileceğiniz üzere giriş 1 olsa da çıkışın da 1 olması için belli bir süre geçer. Bu süreye gecikme adı verilir.

Tüm mantık kapıları, latch ve flip-floplar böyle bir gecikmeye sahiptir. Dolayısıyla, bu elemanları kullanarak kurulan dijital devrelerin girişten çıkışa kadar kullanılan tüm elemanların kümülatif bir gecikmesine sahip olacaktır. Bu gecikme ne kadar fazla olursa devre o kadar yavaş çalışacaktır. Sonraki kısımlarda kombinasyonel ve ardışıl devrelerdeki gecikme türlerini, zaman analizinin nasıl yapılacağını ve devreyi nasıl daha hızlı hale getirebileceğimizi göreceğiz.



Şekil 2: Propagasyon ve kontaminasyon gecikmeleri.

## 2 Kombinasyonel Devrelerde Zamanlama

Kombinasyonel devrelerdeki gecikmeler **propagasyon** (yayıma) ve **kontaminasyon** gecikmesi olarak iki türe ayrılır. Propagasyon gecikmesi ( $t_{pd}$ )<sup>1</sup> girişlerin değişmesinden çıkışların son değerlerine ulaşmasına kadar geçen maksimum süredir. Kontaminasyon gecikmesi ( $t_{cd}$ )<sup>2</sup> ise girişlerin değişmesinden sonra çıkışların değişiklik göstermeye başladığı ilk ana kadar geçen minimum süredir.

Şekil 1 (a)'daki TAMPON kapısı için propagasyon ve kontaminasyon gecikmeleri Şekil 2'deki gibidir. Şekil 2'de daha önceki zaman diyagramlarından farklı olarak giriş ve çıkışlar aynı anda hem 1 hem de 0 olacak şekilde çizilmiştir. Çünkü bu zaman diyagramında değer ne olduğuyla değil değer ne zaman değişmeye başladığıyla ilgileniyoruz. Görülebileceği üzere A girişi değiştikten sonra Y çıkışının ilk değişmeye başladığı  $t_{cd}$  süresi kontaminasyon gecikmesidir.  $t_{cd}$  anından itibaren Y çıkışı girişte yapılan değişikliği yansıtacak şekilde değişmeye başlar. Fakat, bu değişim de anlık değildir. Çıkışın asıl değeri kararlı bir şekilde göstermesi için de bir süre geçer.  $t_{cd}$  ve çıkışın kararlı olması için geçen sürenin toplamı da  $t_{pd}$  propagasyon gecikmesini verir. Görülebileceği üzere,  $t_{pd}$  girişin değişmesinden çıkışın son kararlı değerine ulaşması için gereken toplam süredir.

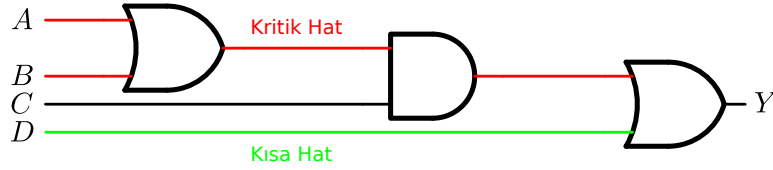
Bir mantık kapısı için  $t_{pd}$  ve  $t_{cd}$  değerleri ortam sıcaklığı, çıkışın 0 iken 1 olması ya da 1 iken 0 olması gibi pek çok duruma göre değişiklik göstermektedir. Bu nedenle, datasheet'lerde bu tür gecikme süreleri minimum, maksimum ve ortalama değerler şeklinde verilir. Gecikmeler genellikle pikosaniyeler ile nanosaniyeler seviyesindedir ( $1 \text{ ps} = 10^{-12} \text{ s}$  ve  $1 \text{ ns} = 10^{-9} \text{ s}$ ).

Birden fazla lojik eleman kullanılarak kurulmuş çok girişli bir devrenin propagasyon gecikmesi giriş ile çıkış arasında kullanılan hatta göre farklılık göstermektedir. Bir hattın propagasyon gecikmesi, hat üzerindeki her bir lojik elemanın propagasyon gecikmelerinin toplamıdır. Örnek olarak Şekil 3'deki devreyi inceleyelim.

A ve B girişlerinde yapılacak bir değişikliğin Y çıkışına ulaşması için toplam 3 adet mantık kapısından geçiş yapılması gerekmektedir. Bu hatların propagasyon gecikmesi 3 mantık kapısının propagasyon gecikmelerinin toplamıdır. C girişindeki değişiklik

<sup>1</sup>ing. *Propagation delay*

<sup>2</sup>ing. *Contamination delay*



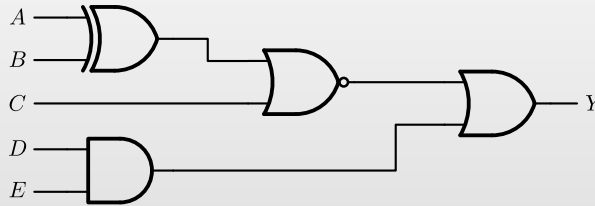
Şekil 3: Kombinyasyonel devrelerde kritik ve kısa hat örneği.

ise 2 mantık kapısından geçtikten sonra  $Y$  çıkışına ulaşmaktadır. Bu hattın propagasyon gecikmesi 2 adet mantık kapısının propagasyon gecikmesinin toplamıdır. Son olarak,  $D$  girişindeki değişiklik ise sadece tek bir mantık kapısından geçerek çıkışa ulaşmaktadır. Bu nedenle,  $D$  giriş hattının propagasyon gecikmesi, tek bir mantık kapısının propagasyon gecikmesine eşdeğerdir<sup>3</sup>. Sonuç olarak  $A$  ve  $B$  giriş hatları en yavaş hatlar iken  $D$  giriş hattı devrenin en hızlı hattıdır.

Bir devredeki en yavaş hat **kritik hat** olarak adlandırılır. Şekil 3'de  $A$  ve  $B$  giriş hatları en yavaş hatlar olduğundan, kırmızı ile gösterilen hatlar bu devrenin kritik hatlarıdır. Tam tersi, bir devredeki en hızlı hat ise **kısa hat** olarak adlandırılmaktadır. Bu nedenle, Şekil 3'de yeşil ile gösterilen  $D$  giriş hattı devrenin kısa hattıdır. **Bir devrenin propagasyon gecikmesi belirlenirken kritik hattın gecikmesi kullanılır. Kısa hat ise kontaminasyon gecikmesinin belirlenmesi için kullanılır.**

Kısa hat ifadesi ile kısa devre ifadesi farklı terimlerdir. Bunları birbirine karıştırmamanız gerekir. Devrenizde kısa devre varsa, devrenin zaman analizinden daha ciddi problemleriniz var demektir.

### Örnek 2.1:



Yukarıdaki devredeki her bir eleman için  $t_{pd} = 100$  ps ve  $t_{cd} = 60$  ps ise devrenin propagasyon ve kontaminasyon gecikmelerini bulunuz.

Hatırlayacağımız üzere propagasyon gecikmesi girişlerdeki değişiklik ile çıkışlar kararlı hale gelene kadar geçen **maksimum** süredir. Bu maksimum süreyi yukarıda bahsetmiş olduğumuz üzere kritik hat belirler. Devreye baktığımızda  $A$

<sup>3</sup>Düz hat üzerinde sinyalin ilerlemesi için gereken süre çok kısa olduğundan ihmal edilmektedir.

ve  $B$  giriş hatları toplam 3 mantık kapısından geçtiği için kritik hatlardır. Kritik hattın toplam propagasyon gecikmesi de devrenin propagasyon gecikmesi olacaktır. Buna göre, verilen devrenin propagasyon gecikmesi  $3 \times 100 \text{ ps} = 300 \text{ ps}$ 'dir.

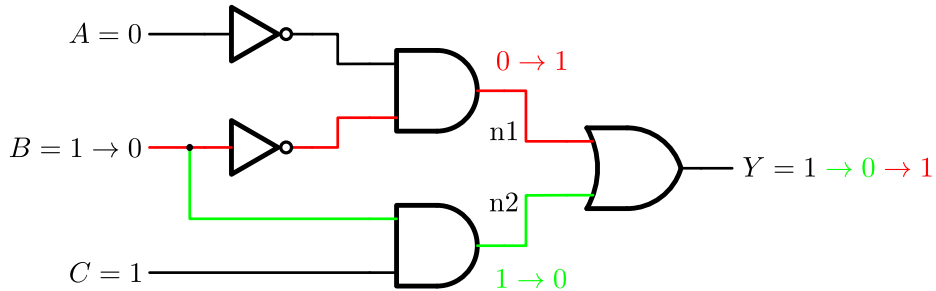
Kontaminasyon gecikmesi ise girişlerde yapılan değişiklik ile çıkışların değişmeye ilk başladığı ana kadar geçen **minimum** süredir. Bu tanıma göre minimum süreyi kısa hat belirleyecek olup devrenin kontaminasyon gecikmesi kısa hat üzerindeki mantık kapılarının kontaminasyon gecikmelerinin toplamı olacaktır.  $C$ ,  $D$  ve  $E$  giriş hatlarının her biri toplam 2 mantık kapısından geçtiğinden bu hatların her biri kısa hattır. Dolayısıyla, devrenin kontaminasyon gecikmesi  $2 \times 60 \text{ ps} = 120 \text{ ps}$  olmalıdır.

## 2.1 Glitch'ler

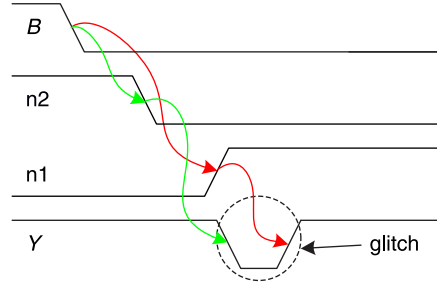
Yukarıda incelediğimiz devre örneklerinde her bir giriş ayrı bir hat üzerinde bulunmaktaydı. Fakat, dijital devrelerde çoğunlukla aynı giriş birden fazla hat tarafından kullanılır. Bu gibi bir durumda ortak girişte yapılacak değişiklik, hatların farklı sürelerde çıkışa etki etmesinden dolayı çıkışta geçici "hatalı" değerlerin gözükmesine neden olacaktır. İşte çıkışta kısa süreli görülen bu hatalı çıkışlara **glitch** adı verilir.

Örnek olarak Şekil 4'deki devreyi inceleyelim. Görebileceğiniz üzere  $B$  girişi hem kırmızı ile gösterilen kritik hatta hem de yeşil ile gösterilen kısa hatta bağlıdır. İlk durumda  $A = 0$ ,  $B = C = 1$  olduğu için  $n1 = 0$ ,  $n2 = 1$  ve  $Y = 1$ 'dir.  $B = 0$  yapıldığında, değişiklik kritik hattaki DEĞİL kapısından geçtiğinde kısa hattaki VE kapısından da geçmiş olacak ve  $n2 = 0$  olacaktır. Değişiklik kısa hat üzerinden VEYA kapısına ulaşıp çıkışı  $Y = 0$  yaptığında ise  $n1$  değeri daha yeni 1 olacaktır.  $n1$  değeri 1 olduktan sonra ise VEYA kapısının propagasyon gecikmesi kadar süre geçtikten sonra  $Y$  çıkışı güncellenerek tekrar 1 olacaktır. Görmüş olduğunuz üzere çıkış kısa bir süreliğine  $Y = 0$  olarak kalmaktadır. İşte bu hatalı çıkışa glitch adı verilir.

Yukarıda anlattıklarımızı zaman diyagramında gösterecek olursak Şekil 5'deki zaman diyagramını elde ederiz. Şekildeki kırmızı oklar kritik hattın etkisini gösterirken, yeşil



Şekil 4: Glitch devresi örneği.



Şekil 5: Glitch devresinin zaman diyagramı.

oklar kısa hattın etkisini göstermektedir. Görülebileceği üzere kritik hattın kısa hatta göre geç kalması nedeniyle girişler  $A = B = 0$ ,  $C = 1$  olsa da çıkış çok kısa bir süreliğine hatalı bir şekilde 0 olmaktadır.

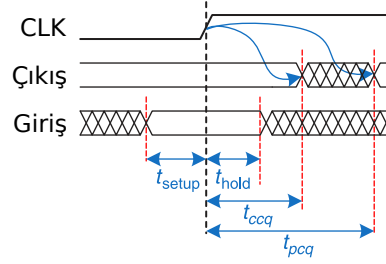
Vermiş olduğumuz örnekte devreye etkisiz elemanlar ekleyerek glitch oluşmasını engellemek mümkündür. Fakat dijital devrelerde birden fazla giriş aynı anda değiştirildiğinde de glitch'ler oluşmakta ve bu tür glitch'leri engellemek mümkün olmamaktadır. Dolayısıyla, dijital devre tasarlarken glitch'leri önlemek için ekstra bir çaba sarf edilmez. Fakat, devrenizde glitch'ler olduğunu bilmeniz gerekir. Devrenin propagasyon ve kontaminasyon gecikmelerini belirlemek bu nedenle önemlidir. Kontaminasyon süresinden hemen sonra çıkışta glitch'ler görülebilecek ve bu durum propagasyon süresi sonuna kadar devam edecektir. Bu nedenle, devredeki çıkışları kullanmadan önce propagasyon süresinin geçmesi beklenmelidir.

### 3 Ardışıl Devrelerde Zamanlama

Yükselen kenarda tetiklenen bir D flip flopu göz önünde bulunduralım. Yükselen kenarda  $D$  girişinin değeri kopyalanarak  $Q$  çıkışına aktarılır. Bu işleme örnekleme adı verilir. Saat darbesi yükselirken  $D$  değeri sabitse bu örnekleme işlemi gayet basit bir şekilde yapılır. Fakat, saat darbesi yükselirken aynı zamanda  $D$  değeri de değişirse ne olur?

Bu problem fotoğraf makinesi ile çekim yapmaya benzerdir. Bir ağaç üzerinde uçmak üzere olan bir kuşun fotoğrafını çektiğinizi varsayın. Eğer deklanşöre önce basarsanız kuşu ağacın üzerinde görüntülersiniz. Eğer deklanşöre geç basarsanız kuş uçmuş olur ve sadece ağacı görüntülersiniz. Tam kuş havalandığı anda deklanşöre basarsanız kuşu belli belirsiz bir flu bir şekilde görüntülemiş olursunuz. Bunun nedeni fotoğraf çekiminde net bir görüntü için objelerin belli bir süre boyunca sabit kalması gerekliliğidir. Bu süreye apertür süresi adı verilir.

Tıpkı fotoğraf makinesinde olduğu gibi flip flopların da sağlıklı bir örnekleme işlemi yapabilmesi için bir **apertür süresine** ihtiyaçları vardır.  $D$  girişi yükselen kenarın hemen öncesinde ve hemen sonrasında belli bir süre boyunca sabit kalmalıdır. Yükselen kenardan hemen önce gerekli olan süreye **setup süresi** adı verilir ve  $t_{setup}$  ile



Şekil 6: Flip flop örnekleme için gereken süreler.

gösterilir. Yükselen kenardan hemen sonra gereken süreye ise **hold süresi** adı verilir ve  $t_{hold}$  ile ifade edilir. İşte flip flopun apertür süresi  $t_{setup} + t_{hold}$  kadardır.

Bunun dışında yukarıda bahsetmiş olduğumuz üzere, diğer tüm lojik elemanlar gibi **flip floplar da kendilerine ait kontaminasyon ve propagasyon gecikmelerine sahiptir.** Fakat, kombinasyonel elemanlardan farklı olarak flip floplar için bu süreler girişte yapılan bir değişiklikten itibaren değil, saat darbesinden (yükselen kenardan) itibaren ölçülür. Flip floplardaki kontaminasyon gecikmesi  $t_{ccq}$  ile gösterilir. Buradaki  $ccq$  “clock to Q contamination delay” yani “saat darbesi ile flip flopun Q çıkışı arasındaki kontaminasyon gecikmesi” ifadesinin kısaltmasıdır. Flip floplardaki propagasyon gecikmesi ise  $t_{pcq}$  ile gösterilir. Buradaki  $pcq$  “clock to Q propagation delay” yani “saat darbesi ile flip flopun Q çıkışı arasındaki propagasyon gecikmesi” ifadesinin kısaltmasıdır.

Buna göre, flip flopun örnekleme için gereken süreler Şekil 6’daki gibi olacaktır. Zaman diyagramında giriş ve çıkışların değeri bizi ilgilendirmediği için hem 1 hem de 0 olacak şekilde çizilmişlerdir. Görüleceği üzere yükselen kenardan hemen önceki  $t_{setup}$  ve hemen sonraki  $t_{hold}$  süreleri boyunca giriş sabit kalmaktadır. Bu flip flopun örnekleme yapabilmesi için şarttır. Yükselen kenardan sonra Q çıkışının ilk değişmeye başladığı minimum süre, zaman diyagramından görülebileceği üzere kontaminasyon gecikmesi  $t_{ccq}$ ’dir. Yükselen kenardan sonra Q çıkışının D değerini kararlı bir şekilde göstermeye başladığı süre ise  $t_{pcq}$  propagasyon gecikmesidir.

Yukarıda verdiğimiz örnek yükselen kenar tetiklemeli flip flop içindir. Düşen kenar tetiklemeli flip floplarda aynı süreler düşen kenara göre belirlenir.

### 3.1 Çalışma Frekansı

Dijital sistemlerdeki ardışıl devreler belli bir saat frekansına göre çalışır. Bu saat frekansı  $f_c$  ile gösterilir ve çalışma frekansı olarak adlandırılır. **Çalışma frekansı ne kadar yüksek olursa ardışıl devre o kadar hızlı olacaktır.**  $T_c = 1/f_c$  ise saat sinyalinin periyodudur.

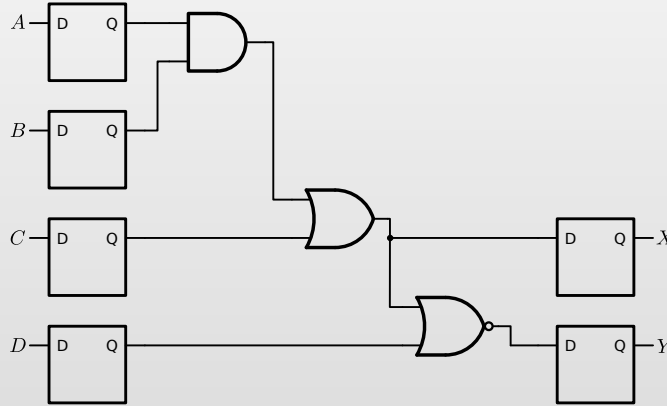
Ardışıl devrenin kombinyonel devreye bağlı bir flip flop olduğunu varsayalım. Kombinyonel devrenin girişlerinin flip fropa ulaşması için kombinyonel devrenin propagasyon gecikmesi  $t_{pd}$  kadar süre geçmesi gerekir. Daha sonra flip flopun örnekleme yapabilmesi için  $t_{setup}$  süresi boyunca flip flop girişi sabit kalmalıdır. Ardından ise  $Q$  çıkışının hazır olabilmesi için  $t_{pcq}$  kadar süre geçmesi gerekir. Bu nedenle, ardışıl devrenin düzgün çalışabilmesi için  $t_{pd} + t_{setup} + t_{pcq}$  süresi boyunca ikinci bir kenar tetikleme olmamalıdır. Bu şart ise ancak saat periyodu

$$T_c \geq t_{pd} + t_{setup} + t_{pcq}$$

olursa sağlanır. (Üreticiler tarafından  $t_{hold} < t_{ccq}$  şartı sağlandığı ve  $t_{ccq} < t_{pcq}$  olduğu için  $t_{hold}$  ve  $t_{ccq}$  bu toplama katılmaz. Başka bir deyişle  $t_{pcq}$  zaten  $t_{hold}$  ve  $t_{ccq}$  sürelerini içermektedir.)

$f_c = 1/T_c$  olduğu için ardışıl devrenin çalışma frekansı  $T_c$  değeri tarafından sınırlanır. Devreyi daha yüksek frekanslarda çalıştırmak için  $T_c$  periyodunun düşürülmesi gerekir.  $t_{setup}$  ve  $t_{pcq}$  flip flop üreticisi tarafından belirlenen değerler olduğundan tasarımcının daha yüksek bir frekansa ulaşabilmesi için yapabileceği şey  $t_{pd}$  yani kombinyonel devrenin propagasyon gecikmesini daha düşük olacak şekilde yeniden tasarlamaktır. Bunun dışında, devrenin daha hızlı çalışması için bir diğer alternatif olarak paralelleştirmeden faydalanılır. Bunu ise daha sonra göreceğiz.

### Örnek 3.1:



Yukarıdaki devrede flip floplar için  $t_{ccq} = 30$  ps,  $t_{pcq} = 80$  ps,  $t_{setup} = 50$  ps ve  $t_{hold} = 60$  ps'dir. Her bir mantık kapısı için propagasyon gecikmesi  $t_{pd} = 40$  ps ve kontaminasyon gecikmesi  $t_{cd} = 25$  ps'dir.

Buna göre devrenin maksimum çalışma frekansını hesaplayıp devrede herhangi bir zamanlama ihlali olup olmadığını belirleyin. (Not: Basitlik olması açısından flip flopların saat girişleri gösterilmemiştir.)



Devrenin kritik hattı  $A$  girişi ile  $Y$  çıkışı arasındadır. Kritik hatta 3 adet mantık kapısı bulunduğu için kombinasyonel devrenin toplam propagasyon gecikmesi  $3t_{pd} = 120$  ps olacaktır. Buna göre, çalışma periyodu bağıntısı aşağıdaki gibi olacaktır:

$$T_c \geq 3t_{pd} + t_{setup} + t_{pcq} = 120 + 50 + 80 = 250 \text{ ps}$$

$T_c = 250$  ps seçilirse maksimum çalışma frekansı  $f_c = 1/(250 \text{ ps}) = 4$  GHz olarak bulunur.

Devrede zamanlama ihlali olup olmadığını belirlemek için  $t_{hold}$  süresi içerisinde  $X$  ve  $Y$  flip flopların giriş değerlerinin değişip değişmediğinin kontrol edilmesi gerekir. Bunun için kısa hatlardaki kontaminasyon gecikmelerine bakılır. (Hatırlayacağınız üzere kontaminasyon süresi çıkışların değişmeye ilk başladığı süredir). Devrenin kısa hatları  $C$  ile  $X$  ve  $D$  ile  $Y$  arasındaki hatlar olduğu için bu hatlar incelenir. Yükselen kenardan sonra  $C$  ve  $D$  flip floplarının çıkışlarının ilk değişmeye başladığı süre  $t_{ccq} = 30$  ps olacaktır. Ardından mantık kapıları için  $t_{cd} = 25$  ps süre geçtikten sonra  $X$  ve  $Y$  flip floplarının girişleri değişmeye başlayacaktır. Yani, kısa hatlarda yükselen kenardan  $t_{ccq} + t_{cd} = 55$  ps sonra  $X$  ve  $Y$  flip floplarının girişleri değişmeye başlayacaktır. Oysa ki, flip floplar için  $t_{hold} = 60$  ps olduğundan yükselen kenardan 60 ps sonrasına kadar bu girişlerin değişmemesi gerekir. Dolayısıyla, **devrede  $t_{hold}$  ihlali vardır ve devre düzgün çalışmayacaktır.**

### Örnek 3.2:

*Bir önceki örnekteki  $t_{hold}$  ihlalinin gidermek için ne yapılmalıdır?*

Kısa hatlardaki kontaminasyon gecikmelerinin  $t_{hold} = 60$  ps değerinden daha fazla olması sağlanmalıdır. Bunun için  $C$  ile  $X$  ve  $D$  ile  $Y$  arasındaki hatlara birer tane TAMPON kapısının bağlanması yeterlidir. Bu durumda bu hatlardaki kontaminasyon gecikmesi  $t_{ccq} + 2t_{cd} = 30 + 50 = 80$  ps olacak ve 60 ps boyunca  $X$  ve  $Y$  flip floplarının girişlerinin değişmemesi garanti altına alınacaktır. (Maksimum çalışma frekansı da değişmeyecektir).

Önceki örneklerdeki  $t_{hold}$  değerleri alışılmadık derecede uzundur. Normalde flip flop üreticileri  $t_{hold} < t_{ccq}$  şartını sağlayacak şekilde üretim yaparlar. Bu sayede flip flopları art arda bağlamak herhangi bir sorun teşkil etmez.

## 3.2 Paralleleştirme

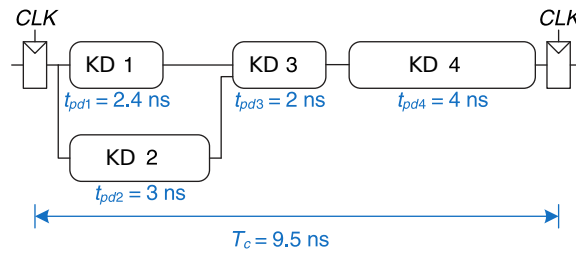
Dijital devrelerin hızının ve verimliliğın ölçüsü olarak **latency** ve **throughput** değerlerine bakılır. Latency dijital devrenin bir veriyi işlemesi için baştan sona kadar geçen toplam süreyi ifade eder. Throughput ise birim zaman başına işlenen veri miktarıdır. Dijital devre yerine örnek olarak kurabiye pişirdiğimizi varsayalım. Bir tepsi kurabiye yapmak için 5 dakikada tepsiyi hazırladığımızı ve ardından 15 dakikada bir tepsi kurabiye pişirdiğimizi varsayalım. Bu işlem için latency 20 dakika veya 1/3 saattir. 1 saatte 3 tepsi kurabiye pişirebildiğimiz için throughput miktarı ise 3 tepsi/saat olacaktır.

Devre verimliliğini arttırmak için çalışma frekansını arttırmayı deneyebiliriz. Yani kurabiye pişirirken tepsi hazırlama ve sürelerini daha da kısaltabiliriz. Fakat, frekans ancak bir noktaya kadar arttırılabilir. Bu nedenle, çalışma verimliliğini arttırmak için frekansını arttırmak dışında paralel çalışma prensibinden faydalanılır. Paralleleştirme iki türlü yapılabilir: Ya aynı donanımı tekrar kullanarak aynı anda birden fazla iş yaparız ya da işi parçalara bölerek yaparız. İkinci tür paraleleştirmeye **pipelining** adı verilir.

Kurabiye pişirme işini ilk türde paraleleştirmek istersek kendi tepsisi ve fırını olan bir arkadaşımızdan yardım istememiz gerekir (aynı donanım). Böylece, iki kişi 20 dakikada 2 tepsi kurabiye yapar. Bu durumda latency (iş yapma için gereken süre) halen 20 dakikadır fakat throughput saatte artık 6 tepsi kurabiye yapıldığından 6 tepsi/saat şeklinde iki katına çıkacaktır.

Pipelining yapmak içinse işi parçalara bölmemiz gerekir. Eğer ikinci bir tepsi kullanırsak tepsi hazırlama ve fırında pişirme işlerini iki ayrı iş olarak yapabiliriz. Bunun için sadece ikinci bir tepsi kullanmamız yeterlidir. Böylece fırında bir tepsi pişerken diğer tepsiyi önceden hazırlamış oluruz. Yani pişirmeden sonra tepsi hazırlamak için 5 dk daha harcanmaz ve her 15 dakikada bir yeni bir tepsi pişirilir. Bu sayede, saatte 4 tepsi pişirilir ve throughput 4 tepsi/saat olur; verimlilik artar. Pipelining kullanmanın en önemli avantajı aynı donanımı tekrar kullanmaksızın verimliliğın arttırılabilmesidir.

Paralleleştirme için aşağıdaki devre blok diyagramını inceleyelim:



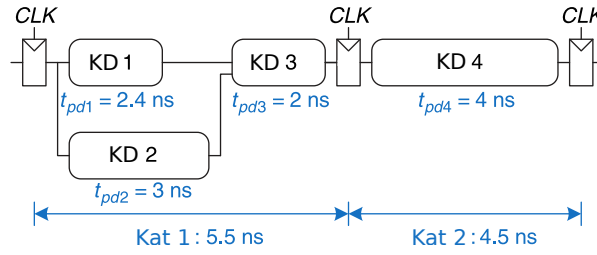
Blok diyagramdaki KD blokları kombinasyonel devreleri ifade ederken saat (CLK) girişli bloklar flip flopları temsil etmektedir. Flip floplar için  $t_{pcq} = 0.3 \text{ ns}$  ve  $t_{setup} = 0.2 \text{ ns}$ 'dir. Kombinasyonel devreler için kritik hattın gecikmesi  $t_{pd2} + t_{pd3} + t_{pd4} =$

$3 + 2 + 4 = 9$  ns'dir. Dolayısıyla, verilen devrenin maksimum çalışma frekansında çalışması için  $T_c = 0.3 + 0.2 + 9 = 9.5$  ns olmalıdır. Bu değer aynı zamanda devrenin latency değeridir.

Frekans birimi olan "Hertz" saniye başına düşen devir (periyot) sayısını ifade eder. 105 MHz 1 saniyede 105 milyon saat darbesi demektir.

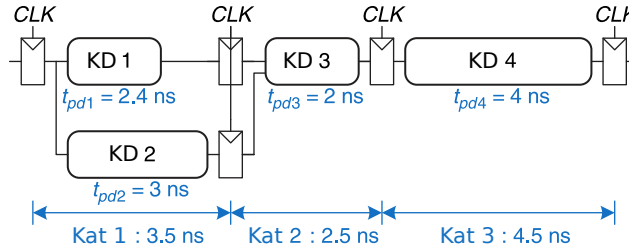
Throughput ise 1 saniyede üretilen bit miktarıdır. Devre her bir saat periyodunda (darbesinde) 1 bit veri ürettiği ve frekans  $f_c = 1/(9.5 \text{ ns}) = 105$  MHz olduğu için **devre 1 saniyede 105 milyon bit üretir**. Bunu 105 Mbps (Megabit per second <sup>4</sup>) ile ifade ederiz.

Şimdi bu devreye pipelining uygulayacağız. Pipelining uygulamak için KD 3 ve KD 4 kombinasyonel devreleri arasında bir flip flop bağlayarak devreyi aşağıdaki gibi iki katlı bir hale getiririz:



Görülebileceği üzere ilk kat için minimum saat periyodu  $T_c = 3 + 2 + 0.3 + 0.2 = 5.5$  ns ve ikinci kat için minimum saat periyodu  $T_c = 4 + 0.3 + 0.2 = 4.5$  ns olmaktadır. Flip floplar ortak saat frekansıyla çalışacağı için bu iki  $T_c$  değerinden büyük olan yani 5.5 ns tüm devrenin minimum saat periyodu olarak seçilmelidir. Bu durumda, devre aynı işi tek bir saat periyodu yerine iki tane saat periyodunda yapacaktır. Bu nedenle latency  $2 \times 5.5 = 11$  ns olacaktır. Pipeline ile latency artmış olsa da her  $T_c = 5.5$  ns'de devre çıkış vereceği için throughput  $1/(5.5 \text{ ns}) = 182$  Mbps olacaktır. Yani pipelining ile throughput ve çalışma frekansı ( $f_c = 1/T_c = 182$  MHz) neredeyse iki katına çıkmıştır. Böylece devre hızlandırılmıştır.

KD 3 ile önceki katlar arasında da pipelining uygulamak mümkündür. KD 1 ve KD 2 paralel devreler olduğu için pipelining yapmak için aşağıdaki gibi hem KD 1 hem de KD 2 çıkışına flip flop bağlanır:



Bu durumda ilk kat için minimum saat periyodu 3.5 ns; ikinci kat için 2.5 ns; üçüncü kat için 4.5 ns olacaktır. Yine bu değerlerden en büyük olan devrenin saat periyodu

<sup>4</sup>tr. Saniye başına megabit

olarak seçilir. Yani  $T_c = 4.5$  ns olur. Bu sefer tek bir işin yapılması için üç saat periyodu harcanacağı için latency yine artarak  $3 \times 4.5 = 13.5$  ns olur. Fakat, her  $T_c = 4.5$  ns'de devre bir çıkış üreteceği için throughput da artarak  $1/(4.5 \text{ ns}) = 222$  Mbps olacaktır. Görülebileceği üzere pipelining ile aynı devreye sadece flip flop ekleyerek sistemin verimliliğini ve hızını arttırmak mümkündür.

### Örnek 3.3:

*Yukarıda verilen pipeline örneği devrelerde throughput değerleri sırasıyla 105 Mbps, 182 Mbps ve 222 Mbps olarak bulunmuştu. Bu devreler 1 bit yerine her bir saat darbesinde  $n$  bitlik veri üretseydi throughput değerleri sırasıyla ne olurdu?*

1 bit için throughput değerleri aynı zamanda devrenin çalışma frekanslarına karşılık gelir. Dolayısıyla, bu devrelerin çalışma frekansları sırasıyla 105 MHz, 182 MHz ve 222 MHz olmalıdır. 105 MHz 105 milyon saat darbesi anlamına gelmektedir. Her bir saat darbesi  $n$ -bitlik çıkış üreteceği için throughput saniye başına  $n \times 105$  milyon bit ya da kısaca  $n \times 105$  Mbps olmalıdır. Diğer örneklerin throughput değerleri de benzer şekilde  $n \times 182$  Mbps ve  $n \times 222$  Mbps olarak bulunur.

## İlave Okuma

- Harris D., Harris S., Digital Design and Computer Architecture, Morgan Kaufmann, 2nd Ed., 720 pp, 2012.