

# 17

## Timing bij data-overdracht

### 17.1 Uitgangspunten

In dit hoofdstuk onderzoeken we de timing van data-overdracht tussen flip-flops en de invloed van de ertussen geplaatste combinatorische logica. Bij dit onderzoek veronderstellen we de eigenschappen van de instelsignalen bekend, onder andere de plaats van de niet-gedefinieerde signaalgebieden en de relevante timing parameters van de componenten. Een korte samenvatting van de eerder afgeleide timing modellen volgt hierna. Aan de hand van deze modellen gaan we na onder welke voorwaarden betrouwbare data-overdracht mogelijk is. In het eerste deel van dit hoofdstuk gaan we uit van gegeven parameterwaarden en stellen we vast aan welke grenzen en eisen de parameters moeten voldoen, opdat betrouwbare data-overdracht mogelijk is. Vervolgens gaan we na hoe we, onder andere door een andere interne structuur van de flip-flops te kiezen, de flip-flop parameters kunnen veranderen. Het verschijnsel clock skew en het creëren van een marge hiervoor krijgt daarbij aandacht. We veronderstellen hierbij dat de ontwerper zich houdt aan de opgeven specificaties. Vaak is dit niet genoeg en moeten de marges voor clock skew vergroot worden, onder meer voor verschijnselen waarmee bij het opstellen van het timing model geen rekening is gehouden. Synchronisatie van externe signalen en de daaraan gekoppelde problemen met metastability bespreken we in hoofdstuk 18.

#### *Samenvatting timing modellen*

*Logische signalen* zijn gemodelleerd met niet-gedefinieerde signaalintervallen en intervallen waarin het signaal beschikbaar (gedefinieerd) is of moet zijn. In een *niet-gedefinieerd signaalinterval* zijn het tijdstip van de eventuele signaalverandering en meestal ook de signaalvorm tijdens de verandering onbekend (clock mode). Het signaal heet dan niet-gedefinieerd. Na het aan elke signaalverandering gekoppelde niet-gedefinieerd interval wordt het signaal verondersteld een waarde te hebben in het toegestane gebied voor H respectievelijk L en wel tot de eerstvolgende signaalverandering. Het signaal is dan *gedefinieerd* en bezit een '*logische waarde*'. Soms is dit eenvoudige signaalmodel niet voldoende. Vergelijk de level mode benadering in hoofdstuk 12. Waar nodig breiden we het model uit, bijvoorbeeld met eisen aan de signaalvorm (single transition).

Voor combinatoriek gebruiken we het *poortmodel* dat in de paragrafen 5.9 t/m 5.11 is opgesteld. Dit model kent een *minimum* en een *maximum propagatietijd* voor signaalovergangen. Voor sommige toepassingen moeten we het signaalmodel verder uitbreiden met het begrip *dominantie*. Dan zijn bepaalde signaalwaarden dominant over (alle) signaalwaarden op de andere ingangen, met name in de dynamische toestand. Zie paragraaf 9.1.

*Latches* en *gated latches* worden in de transparante toestand gemodelleerd met een *minimum* en een *maximum propagatietijd*. Ook zijn er eisen gesteld ten aanzien van de *minimale duur* van een *set- of resetopdracht*, of van de *enable puls* (paragraaf 9.6). Daarnaast zijn bij *gated latches* een *setup time* en een *hold time* voor de instelsignalen gedefinieerd, met als referentiepunt bepaalde niveaus op de flanken van de *enable puls* (paragraaf 9.10).

Op het meester-en-slaaf principe gebaseerde *flip-flops* hebben voor de data-ingangen een *setup time* en een *hold time* en voor de data-uitgangen een *minimum* en *maximum propagatietijd* (referentie t.o.v. de klokflanken). Tussen beide gebieden, de instelling van de flip-flop en de daaropvolgende uitgangsreactie, dient een zekere marge te zitten. Zou er geen marge zitten, dan is een flip-flop in het overlappende gebied transparant (paragrafen 10.3 t/m 10.9).

### **Data-overdracht**

Elke *overdracht van data* tussen flip-flops verloopt dan en slechts dan correct als het bedoelde datasignaal

- tijdig gegenereerd wordt;
- niet te vroeg weggehaald wordt.

Hiervoor zijn de parameters  $t_{su}(\text{data} - \text{to} - \text{clock})$  en  $t_h(\text{data} - \text{to} - \text{clock})$  gedefinieerd. Aan beide eisen moet voldaan zijn. Anders mislukt de data-overdracht of, wat vaker voorkomt, de data-overdracht is niet geheel betrouwbaar. Een voorbeeld van een data-overdracht met problemen hebben we reeds besproken in paragraaf 10.1. De gesignaleerde problemen bij de data-overdracht tussen *gated latches* ontstonden doordat het datasignaal te vroeg veranderde. Flip-flops en de erop gebaseerde clock mode ontwerpstyl maken het mogelijk het logisch ontwerp van een schakeling te scheiden van de timing. Zie hoofdstuk 13 voor de spelregels van de clock mode. Bij data-overdracht tussen flip-flops speelt het *kloksignaal* een essentiële rol. Alle timing parameters refereren naar de flanken ervan.

In het *tijdstip van de klokflanken* zit altijd enige *onzekerheid*. Deze onzekerheid komt voort uit de eigenschappen van de klokgenerator, de klok distributie inclusief eventuele klok-buffers (dat zijn versterkers voor de kloksignalen) en de (capacitieve) belasting van de kloklijnen, alsmede uit de herkenning van de klok door de flip-flops. In een klokgenerator worden vaak verschillende kloksignalen opgewekt, al dan niet in fase met elkaar. Het is niet mogelijk dit exact tegelijk te doen. Dit resulteert ook in enige onzekerheid ten aanzien van de tijdstippen van de klokflanken. Daarnaast hebben de verschillende klokuitgangen meestal een ongelijke (capacitieve) belasting. Dit leidt tot een zekere vervorming van de klokflanken onderling. Hoe hoger de belasting van een kloklijn, des te lager is de flanksteilheid van het kloksignaal erop. De belasting hangt ook samen met de distributie van de kloksignalen, onder andere wegens verschil in lengte van de klokaansluitingen. Buffers in kloklijnen veroorzaken een relatief aanzienlijke signaalvertraging. Daarom moeten alle klokbuffers in een kloknetwerk identieke eigenschappen hebben. Het is vaak beter uit te gaan van één klokbuffer voor de hele schakeling dan van veel lokale klokbuffers, of de

invloed van klokbuffers te compenseren met behulp van phase locked loops (PLL's) [den Dulk, 1989]. Tenslotte heeft niet iedere flip-flop hetzelfde drempelniveau, waarop de ingang een signaalovergang herkent. Wanneer flip-flops op verschillende niveaus van hun klokflanken reageren, dan heeft dit invloed op het tijdstip van het herkennen van de klokpuls. Genoemde verschijnselen hebben we eerder besproken. Elke timing referentie naar klokflanken is dus behept met enige onnauwkeurigheid. We duiden dit verschijnsel aan als *clock skew*. Gezien de geconstateerde onzekerheden zal een clock mode schakeling op een aantal punten voldoende grote marges moeten bezitten om alle gevolgen van 'onnauwkeurigheid' in de referentie naar de klok te kunnen opvangen. Deze marges zijn essentieel voor de betrouwbaarheid en voor de reproduceerbaarheid van een schakeling. Kortom, het onderwerp van dit hoofdstuk is: ***verstandig omgaan met onzekerheid***.

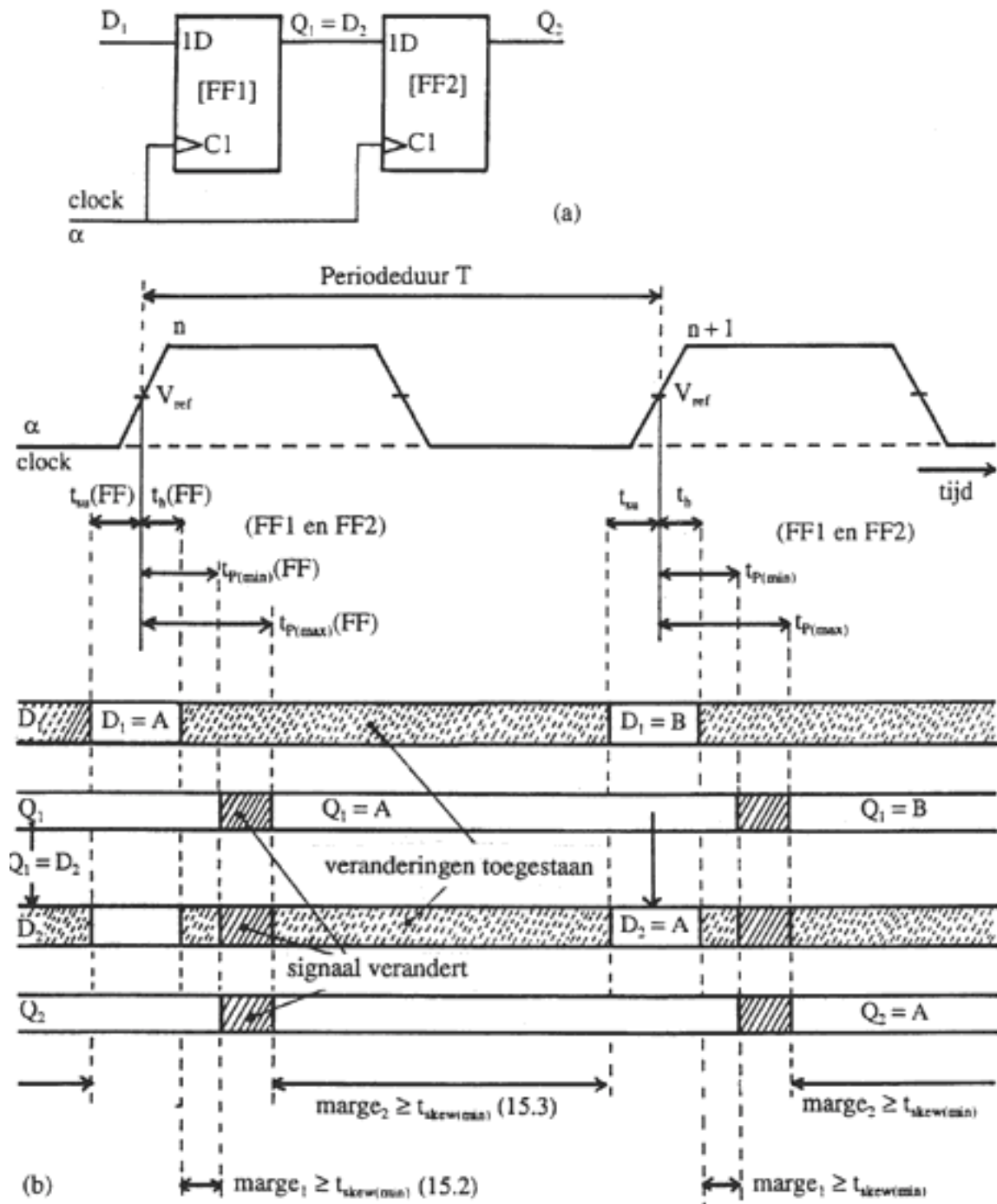
## 17.2 Directe data-overdracht tussen identieke flip-flops

We onderscheiden bij data-overdracht directe en indirecte data-overdracht. Bij *directe data-overdracht* stuurt de Q uitgang van de sturende flip-flop direct de D ingangen van alle erdoor ingestelde flip-flops. Data-overdracht met combinatoriek tussen de flip-flops, de *indirecte data-overdracht*, bespreken we in paragraaf 17.4.

Bij data-overdracht moeten de eigenschappen van het uitgangssignaal van de sturende flip-flop passen bij de eisen die de ingangen van alle erdoor ingestelde flip-flops stellen. Deze eisen kunnen in verschillende categorieën ingedeeld worden, waaronder elektrische compatibiliteit, de juiste timing van de signalen en, uiteraard, de juiste logische waarde van het signaal. Onder elektrische compatibiliteit verstaan we allereerst de gedefinieerde intervallen voor de H/L niveaus met de daaraan gekoppelde DC noise margins. Zonder DC noise margins kan geen enkele variatie in de voedingsspanning van de verschillende IC's opgevangen worden. Verder moeten de impedantieniveaus tussen IC's van verschillende logische families redelijk overeenkomen. Anders mogen de gespecificeerde (timing) parameters niet zonder meer toegepast worden en zijn er correcties op nodig. Parameterwaarden zijn technologie-afhankelijk en niet uitwisselbaar tussen verschillende technologieën. Zijn twee IC's elektrisch compatibel, dan hangt het van de timing parameters af of directe data-overdracht mogelijk is. Dit is lang niet altijd het geval. Voor identieke flip-flops, binnen één IC en in dezelfde technologie uitgevoerd, is dit uiteraard wel zo. Anders zou het flip-flop ontwerp niet deugen. Bij de timing verificatie moet altijd met de nodige marges gerekend worden. Deze zijn nodig voor het opvangen van clock skew, maar ook voor het opvangen van parametervariaties als gevolg van schommelingen in de temperatuur, variaties in de voedingsspanning, enz.

### ***Directe data-overdracht***

*We gaan er hierna van uit dat de elektrische compatibiliteit geverifieerd is en dat er relevante schattingen voor de benodigde marges in de timing gedaan zijn.* Dit laatste is in de praktijk nog een heel probleem, omdat bij standaard IC's in de databoeken de noodzakelijke gegevens om dit te doen vaak ontbreken of incompleet zijn.



**Figuur 17.1.** Directe data-overdracht tussen identieke flip-flops.

Figuur 17.1 schetst de data-overdracht tussen identieke flip-flops. De uitgang van flip-flop FF1 in figuur 17.1.a stuurt direct de data-ingang van flip-flop FF2. Beide flip-flops zijn van hetzelfde logische type, hebben identieke timing parameters en zijn op dezelfde klokpuls  $\alpha$  aangesloten. Voor een correcte data-overdracht is het noodzakelijk dat de uitgang  $Q_1$  van flip-flop FF1 niet verandert gedurende het  $t_{su} - t_h$  interval van ingang  $D_2$  van flip-flop FF2. Het complete timing diagram met de bijbehorende gedefinieerde en niet gedefinieerde signaalintervallen is beschreven in figuur 17.1.b. We zien dat de veranderingen

van  $Q_1$  ruim buiten het  $t_{su} - t_h$  instelinterval van  $D_2$  vallen, bij de gespecificeerde (verhouding van de) parameterwaarden. De bedoelde data-overdracht verloopt correct. (Ga na hoe het signaal 'A' door de schakeling loopt. Kleur de intervallen in.)

In figuur 17.1 is de opgaande klokflank de referentie voor alle timing parameters. Eerder hebben we beredeneerd dat op de timing referentie enige marges toegepast moeten worden, vanwege verschijnselen die we hebben samengevat onder de naam *clock skew*. We drukken de minimaal benodigde marge uit in de parameter  $t_{skew(min)}$ , waaruit de eis volgt:

$$\text{marges in de timing} \geq t_{skew(min)} \quad (17.1)$$

De parameter  $t_{skew(min)}$  geeft de marge aan, waarbinnen we de tijdreferentie niet nauwkeuriger kunnen vaststellen of inschatten. Deze marge zal bij data-overdracht dus altijd moeten overblijven. Gemakshalve veronderstellen we dat (verschillen in de maximale) *looptijden van signalen door de bedrading* in de marge  $t_{skew(min)}$  opgenomen zijn. In de volgende formules komt derhalve geen bijdrage voor looptijden voor!

In figuur 17.1 zijn de plaatsen aangegeven, waar een marge voor clock skew nodig is. De marges moeten zitten tussen veranderingen van de uitgang  $Q_1$  van de sturende flip-flop FF1 en het  $t_{su} - t_h$  interval van de ingang  $D_2$  van flip-flop FF2. We zien uit de figuur dat

$$\text{marge}_1 = t_{P(min)}(FF1) - t_h(FF2) \geq t_{skew(min)} \quad (17.2)$$

(drukt uit dat  $Q_1$  niet mag veranderen voordat  $D_2$  dit toestaat) en

$$\text{marge}_2 = T_{clock} - t_{P(max)}(FF1) - t_{su}(FF2) \geq t_{skew(min)} \quad (17.3)$$

(drukt uit dat de volgende klokpuls niet te vroeg mag komen). Marge (17.2) heeft te maken met de hold time van het signaal op de D ingang van flip-flop FF2 en wel ten opzichte van de (minimum) propagatietijd van het uitgangssignaal van flip-flop FF1. Deze marge is *frequentie-onafhankelijk*. De marge (17.3) is wel *frequentie-afhankelijk* en geeft aan dat het D signaal van FF2 niet te laat mag arriveren.

Uit een vergelijking van de marges in de figuur volgt dat de klokflank voor flip-flop FF2 slechts een weinig later gegeven mag worden maar, afhankelijk van de periode van de klok, wel een hele tijd eerder. Bij hinder van clock skew probeert men wel om het spoor (bedrading) voor de klokpuls van FF2 in de layout in de richting van flip-flop FF1 te laten lopen en niet, zoals in het schema, van FF1 naar FF2. Het behoeft geen toelichting dat het timing plan van de schakeling dan eigenlijk fundamenteel fout is opgezet.

Bepalend voor de toegestane clock skew van de klokflanken zijn de marges die overschieten van (17.2) en (17.3), nadat hier  $t_{skew(min)}$  van afgetrokken is. Daarbij is  $t_{skew(min)}$  de minimaal benodigde marge voor clock skew in een bepaalde schakeling. De parameter  $t_{skew(min)}$  is *toepassingsafhankelijk*. Elke marge die boven  $t_{skew(min)}$  beschikbaar is, kan gebruikt worden om bijvoorbeeld de (extra toegevoegde) propagatietijd van klokbuffers op te vangen resp. te compenseren. Omgekeerd, als de beschikbare marges kleiner zijn dan  $t_{skew(min)}$ , dan is betrouwbare data-overdracht niet gegarandeerd. In de praktijk doen proefschakelingen het dan vaak nog wel, maar de data-overdracht is niet 100% foutloos of de schakeling doet het niet tegen de grenzen van zijn specificatie (temperatuurbereik

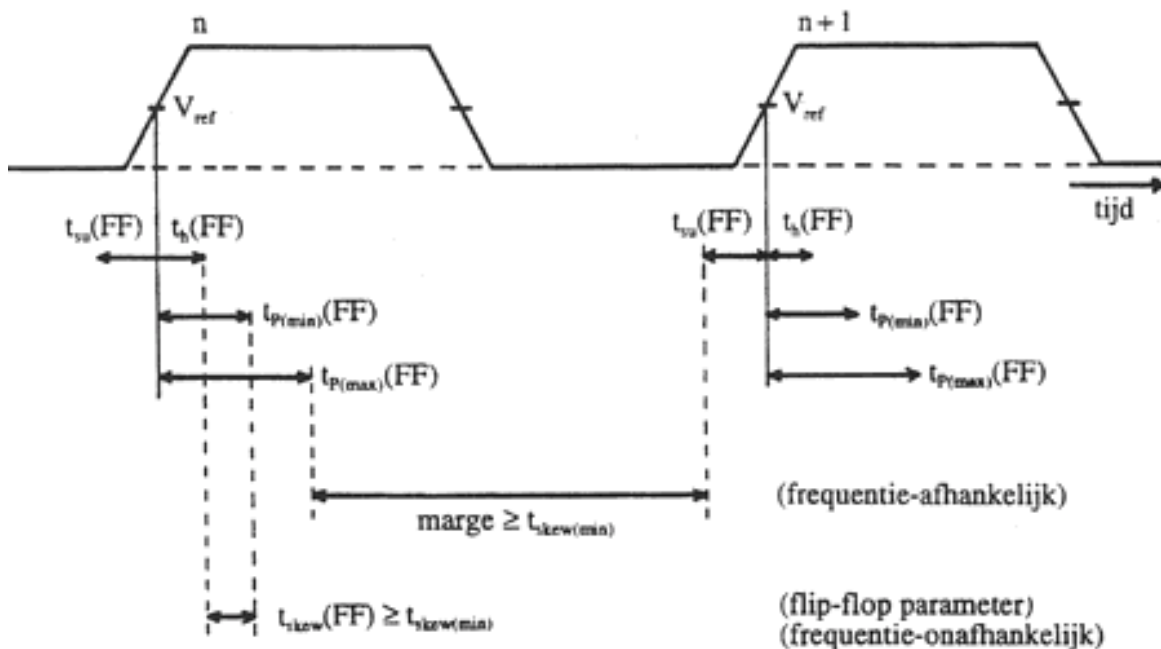
bijvoorbeeld). Ook kan in productie het uitvalpercentage na het testen beduidend hoger zijn dan verwacht.

### Kwaliteit van een flip-flop ontwerp

In figuur 17.2 is aangegeven hoe de beschikbare marges voor clock skew uit de gegeven flip-flop parameters en de klokperiode kunnen worden afgeleid. De kleinste marge, die tussen  $t_h$  en  $t_{P(\min)}$  van de flip-flop, hangt uitsluitend van de flip-flop parameters zelf af. We noemen deze marge  $t_{\text{skew}}(\text{FF})$ :

$$t_{\text{skew}}(\text{FF}) = t_{P(\min)}(\text{FF}) - t_h(\text{FF}) \quad (17.4)$$

De parameter  $t_{\text{skew}}(\text{FF}) = t_{P(\min)}(\text{FF}) - t_h(\text{FF})$  is een belangrijke maat voor de kwaliteit van een flip-flop ontwerp. Hoe kleiner de waarde ervan is, des te kritischer is de timing van de data-overdracht wanneer rekening gehouden wordt met clock skew



**Figuur 17.2.** Marges voor clock skew.

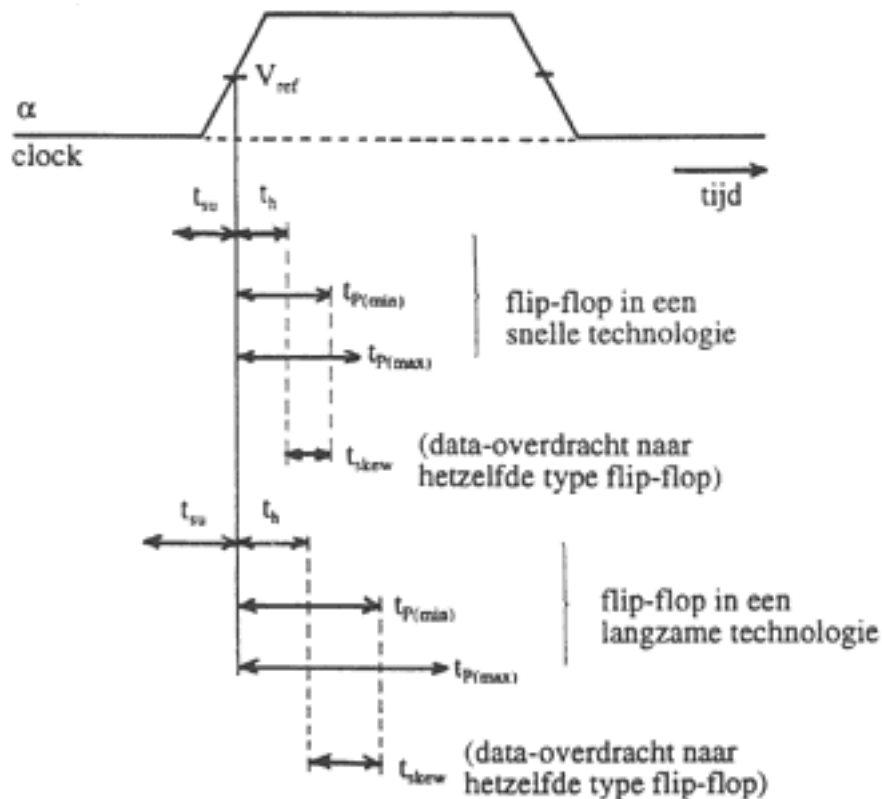
In de andere marge, die tussen  $t_{P(\max)}(\text{FF})$  na klokpuls no.  $n$  en  $t_{su}(\text{FF})$  voor klokpuls no.  $n+1$ , zit meestal voldoende ruimte. De marge is bijna de gehele periodeduur van de klok. Behalve bij hoogfrequent schakelingen is deze marge nooit kritisch. Bij indirecte data-overdracht kan dit anders uitwerken. Zie paragraaf 17.4 voor een verdere toelichting.

### Keuze voor een snelle of een langzame technologie

Flip-flop parameters hangen af van de interne opbouw van de flip-flop. Zie paragraaf 17.7. Ook de technologiekeuze heeft invloed. In het geval dat men een snellere technologie kiest worden de beschikbare marges voor clock skew als regel kleiner. Blijven alle parameters ten opzichte van elkaar in dezelfde verhouding staan, dan is de marge  $t_{\text{skew}}(\text{FF})$ ,

$$t_{\text{skew}}(\text{FF}) = t_{P(\text{min})}(\text{FF}) - t_h(\text{FF})$$

beduidend kleiner voor flip-flops in een snellere technologie. Zie figuur 17.3.



**Figuur 17.3.** Clock skew bij snelle en langzame flip-flops.

Deze conclusie heeft consequenties voor de keuze van de technologie: **kies deze niet sneller dan strikt noodzakelijk is**. Bij een snelle technologie moet (de layout van) de klok distributie heel zorgvuldig worden uitgevoerd. Iedere aanleiding tot clock skew dient te worden vermeden. Een langzamere technologie resulteert in robuustere schakelingen.

#### Opmerking

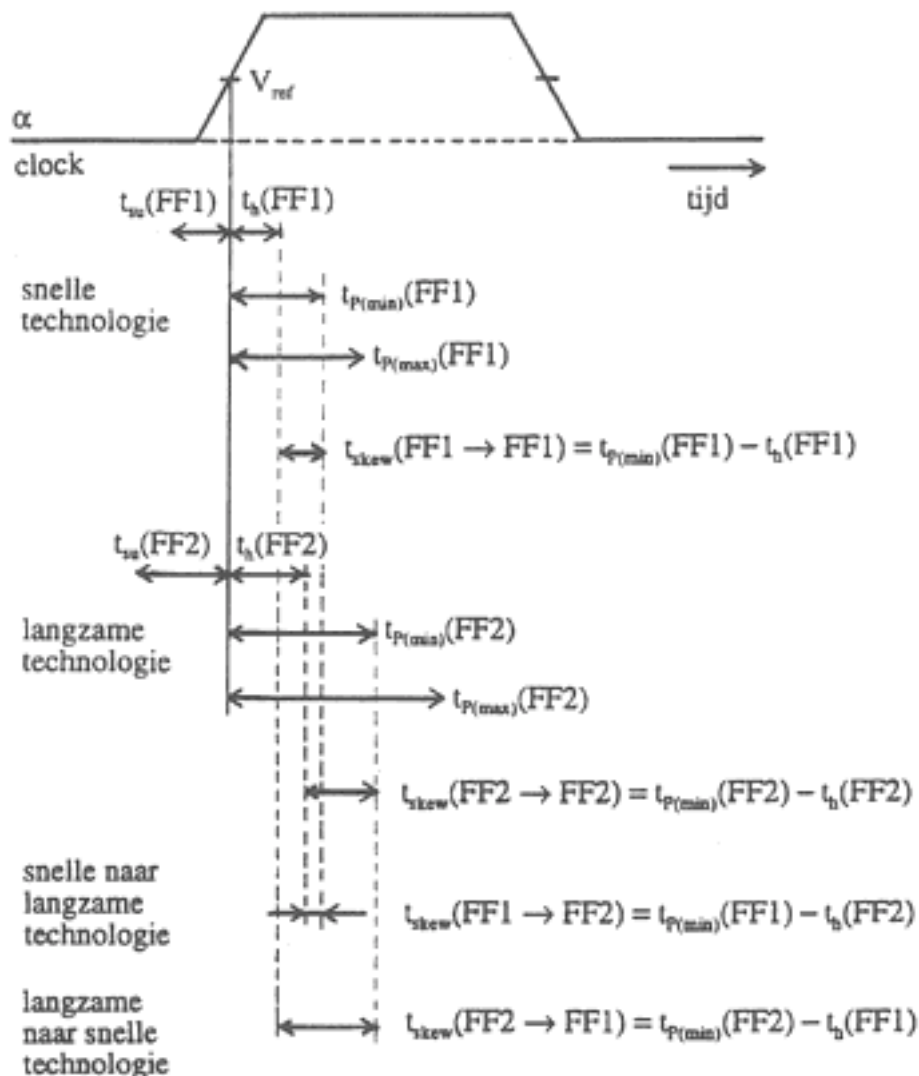
In deze paragraaf hebben we ons beperkt tot (positive) edge-triggered flip-flops. De uitbreiding naar andere timing systemen behandelen we later. In principe is de gedachtengang daarbij hetzelfde. □

## 17.3 Directe data-overdracht tussen flip-flops met verschillende timing parameters

Op PCB's (printed circuit boards) zit vaak een hele mix van IC's en wel vanwege snelheids- en technologieverschillen. Binnen één technologie dissiperen snelle IC's als regel meer. Snelheid wordt bij (bipolaire) IC's onder andere bereikt door over te gaan op lagere impedanties, hetgeen tot hogere stromen leidt. Om de dissipatie te beperken wordt dan niet

het gehele ontwerp in snelle IC's uitgevoerd, maar slechts die delen waarvan de gewenste 'performance' dit vereist. Dit heeft gevolgen voor de marges bij data-overdracht. Bij technologieverschillen treffen we een overeenkomstige situatie aan. De technologie om een 1Mbit RAM IC te maken verschilt sterk van de technologie waarin kleine registers en tellers gemaakt worden. Bibliotheken voor IC ontwerpsystemen bevatten meer dan één type flip-flop. Afhankelijk van o.a. de afmetingen van de transistoren, de layout en de ingebouwde functies zijn de parameters van de opgenomen flip-flops verschillend. Allereerst moeten we in deze gevallen letten op de al eerder genoemde elektrische compatibiliteit. Zijn de parameterwaarden bij een mix van IC's en/of van technologieën nog wel geldig? Daarnaast kan ook de onderlinge verhouding van de parameters tot conflicten aanleiding geven.

### Voorbeeld



**Figuur 17.4.** Berekening marge  $t_{skew}$  bij data-overdracht tussen flip-flops met verschillende timing parameters.



In figuur 17.4 is de data-overdracht geschetst tussen een snel en een langzaam type IC. We nemen aan dat de interne structuren van de flip-flops identiek zijn en dat de signalen elektrisch compatibel zijn.

Bij deze wijze van data-overdracht zien we vier verschillende waarden voor de marge  $t_{\text{skew}}$  optreden:

$$t_{\text{skew}}(\text{FF1} \rightarrow \text{FF1}) = t_{\text{P}(\text{min})}(\text{FF1}) - t_{\text{h}}(\text{FF1}) \quad (17.5.a)$$

$$t_{\text{skew}}(\text{FF2} \rightarrow \text{FF2}) = t_{\text{P}(\text{min})}(\text{FF2}) - t_{\text{h}}(\text{FF2}) \quad (17.5.b)$$

$$t_{\text{skew}}(\text{FF1} \rightarrow \text{FF2}) = t_{\text{P}(\text{min})}(\text{FF1}) - t_{\text{h}}(\text{FF2}) \quad (17.5.c)$$

$$t_{\text{skew}}(\text{FF2} \rightarrow \text{FF1}) = t_{\text{P}(\text{min})}(\text{FF2}) - t_{\text{h}}(\text{FF1}) \quad (17.5.d)$$

Van deze marges is  $t_{\text{skew}}(\text{FF1} \rightarrow \text{FF2})$ , de beschikbare marge bij data-overdracht van het snelle naar het langzame type, de kleinste. *De combinatie snel/langzaam is dus slechter dan een schakeling die geheel uit snelle IC's bestaat. Onder omstandigheden kan het snelheidsverschil te groot zijn en is betrouwbare data-overdracht niet meer mogelijk. Een verrassende conclusie!*  $\square$

In de praktijk komen identieke situaties ook voor bij het ontwerpen van zeer grote IC's, IC's voor onder andere digitale signaalbewerkingen. Aan zo'n IC wordt door verschillende design teams gewerkt. Wanneer niet alle teams uitgaan van dezelfde bibliotheek, of wanneer er in dezelfde bibliotheek verschillende types flip-flops zitten, dan kunnen er serieuze timing problemen ontstaan. Hetzelfde geldt als bij de modules verschillende layout stijlen worden toegepast. Meestal verifieert elk team het eigen ontwerp door een uitgebreide simulatie. Tenslotte wordt het geheel samengevoegd, waarbij de interface opnieuw dient te worden gecontroleerd. Behalve timing fouten zitten er in de interface vaak ook 'klokfase verschillen'. Een datasignaal is er dan een klokpuls te vroeg of te laat, maar ten opzichte van de (foute) klokflank wel op het juiste tijdstip.

### ***Conclusies directe data-overdracht***

Het bovenstaande leidt tot de volgende conclusies.

- Bij *directe data-overdracht* tussen flip-flop FF1 en flip-flop FF2 geldt een *marge bij de timing* van

$$\text{marge timing} = t_{\text{P}(\text{min})}(\text{FF1}) - t_{\text{h}}(\text{FF2}) \quad (17.6)$$

- Deze *marge is nodig om de gevolgen van clock skew*, dit is de som van alle onzekerheden in de timing referentie, *op te vangen*. De marge (17.6) is *frequentie-onafhankelijk*.
- Bij directe data-overdracht in snelle logica is deze marge als regel (zeer) klein.
- De parameter  $t_{\text{skew}}(\text{FF})$ ,

$$t_{\text{skew}}(\text{FF}) = t_{\text{P}(\text{min})}(\text{FF}) - t_{\text{h}}(\text{FF}) \quad (17.7)$$

is een belangrijke maat voor de kwaliteit van een flip-flop ontwerp.

- De parameter  $t_{\text{skew}}(\text{FF})$  is bij een *snelle flip-flop* in het algemeen kleiner dan bij een langzamer type.
- Door de flip-flop *intern anders op te bouwen* kan de waarde van de parameter  $t_{\text{skew}}(\text{FF})$  ook beïnvloed worden, zowel in gunstige als in ongunstige zin. Zie paragraaf 17.7.
- Een mix van types pakt meestal slechter uit dan één type flip-flop.
- Voor de correcte werking van een schakeling moet op alle verbindingen gelden dat

$$t_{\text{skew}}(\text{FF1} \rightarrow \text{FF2}) = t_{\text{P}(\text{min})}(\text{FF1}) - t_{\text{h}}(\text{FF2}) \geq t_{\text{skew}(\text{min})} \quad (17.8)$$

waarbij  $t_{\text{skew}(\text{min})}$  de minimaal benodigde marge voor het opvangen van onnauwkeurigheid in de tijdreferentie is. Hierbij moeten alle relevante effecten gesommeerd worden!

- De maximale klokfrequentie wordt bepaald door hetgeen er overschiet van de klokperiode  $T$ , wanneer hiervan  $t_{\text{P}(\text{max})}(\text{FF})$  van de sturende flip-flop en  $t_{\text{su}}(\text{FF})$  van de gestuurde flip-flop afgetrokken zijn en tevens  $t_{\text{skew}(\text{min})}$ . Deze conclusie worst case te interpreteren, over alle paden voor data-overdracht van de gehele schakeling.

In veel databoeken wordt er geen specificatie van de parameter  $t_{\text{P}(\text{min})}(\text{FF})$  gegeven. Of  $t_{\text{P}(\text{min})}(\text{FF})$  is een vast percentage van  $t_{\text{P}(\text{max})}(\text{FF})$ . In dat geval is het niet mogelijk de risico's bij de timing na te gaan. Eigenlijk een absurde situatie. Zie bijna alle databoeken.

In de volgende paragraaf tonen we aan dat logica tussen de flip-flops tot een aanzienlijke vergroting van de marges voor clock skew leidt. Bij directe data-overdracht kunnen we bij onvoldoende marges voor clock skew de oplossing slechts zoeken in een aangepast flip-flop ontwerp (paragraaf 17.7), of door over te gaan op een data lockout timing (paragraaf 17.9). Als laatste en meest krachtige middel noemen we de overgang op een tweefasige klok (paragraaf 17.10).

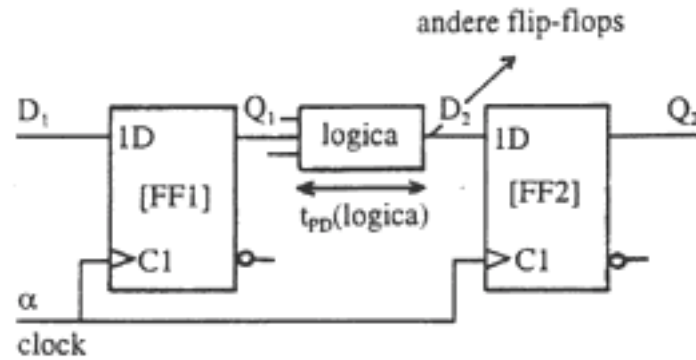
## 17.4 Indirecte data-overdracht

Bij *indirecte data-overdracht* zit er combinatoriek tussen de sturende flip-flop FF1 en de erdoor ingestelde flip-flops FF2. Zie figuur 17.5. Deze vorm van data-overdracht verloopt correct indien voldaan is aan een drietal voorwaarden:

- Het signaal op de ingang  $D_2$  mag niet eerder weggehaald worden dan nadat flip-flop FF2 het heeft ingelezen. Deze eis is *onafhankelijk van de klokfrequentie*.
- Het signaal op de ingang  $D_2$  mag niet te laat arriveren. Deze eis stelt een *boven grens aan de signaalpropagatie* door de combinatoriek, of stelt een *boven grens aan de klokfrequentie* bij gegeven combinatoriek.
- Op de nodige plaatsen moet rekening gehouden worden met  $t_{\text{skew}(\text{min})}$ , de (maximale) *onnauwkeurigheid in de tijdreferentie*.

Figuur 17.6 geeft een toelichting. Na de actieve klokflank past de  $Q_1$  uitgang van flip-flop FF1 zich aan diens nieuwe inwendige toestand aan en wel binnen het door  $t_{\text{P}(\text{min})}(\text{FF1})$  en

$t_{P(\max)}(\text{FF1})$  bepaalde interval. Na  $t_{P(\max)}(\text{FF1})$  heeft  $Q_1$  zijn nieuwe waarde. Hierna stelt de combinatorische logica zich definitief in. Deze heeft hiervoor maximaal  $t_{P(\max)}(\text{logic})$  nodig. Daarna stelt de ingang  $D_2$  van flip-flop FF2 zich in. Hiervoor is een tijd  $t_{su}(\text{FF2})$  nodig, gerekend ten opzichte van de eerstvolgende actieve klokflank. Alle instellingen moeten gereed zijn voor de eerstvolgende actieve klokflank. Afhankelijk van de *periode-duur*  $T$  van de klok is er dan nog een zekere tijd over.



**Figuur 17.5.** Data-overdracht via combinatorische logica.

Betrouwbare data-overdracht via tussenliggende logica is slechts mogelijk indien geldt

$$t_{P(\min)}(\text{FF1}) + t_{P(\min)}(\text{logic}) - t_h(\text{FF2}) \geq t_{\text{skew}(\min)} \quad (\text{data niet te vroeg weg}) \quad (17.9)$$

(voor edge-triggered flip-flops) en tevens

$$T - t_{P(\max)}(\text{FF1}) - t_{P(\max)}(\text{logic}) - t_{su}(\text{FF2}) \geq t_{\text{skew}(\min)} \quad (\text{data te laat aangeboden}) \quad (17.10)$$

De formules (17.9) en (17.10) moeten voor *alle signaalwegen* van flip-flop uitgangen naar flip-flop ingangen gelden en wel *worst case*. (Het bovenstaande geldt onder de stilzwijgende aanname van *elektrische compatibiliteit*!)

Uit figuur 17.6 zien we dat op twee plaatsen de referentie naar de klok een rol speelt. De in formule (17.9) geformuleerde eis is *frequentie-onafhankelijk*, maar heeft wel te maken met clock skew. Een vergelijking met de eisen bij *directe data-overdracht* (17.2),

$$t_{P(\min)}(\text{FF1}) - t_h(\text{FF2}) \geq t_{\text{skew}(\min)}$$

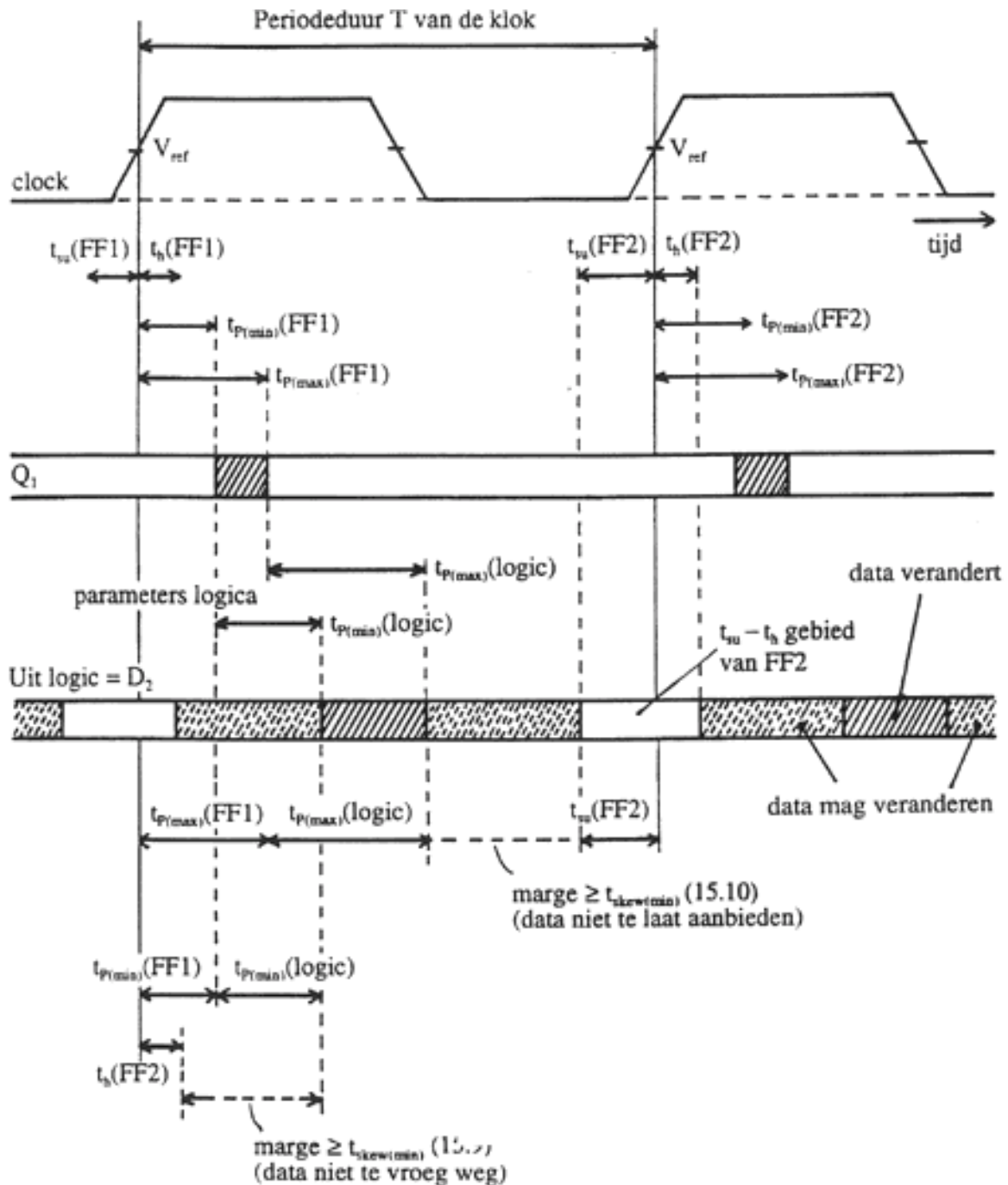
leert dat bij *indirecte data-overdracht* de marge voor het opvangen van clock skew met  $t_{P(\min)}(\text{logic})$  vergroot is. Bij indirecte data-overdracht zal deze marge voor clock skew zelden een probleem zijn. De met formule (17.10) geformuleerde eis stelt een grens aan de *maximale klokfrequentie*. De klokperiode moet voldoen aan

$$T \geq t_{P(\max)}(\text{FF1}) + t_{P(\max)}(\text{logic}) + t_{su}(\text{FF2}) + t_{\text{skew}(\min)} \quad (17.11)$$

### Opmerking

In de bovenstaande formules is de *looptijd van de signalen* over de verbindingen niet meegerekend. Signalen hebben een eindige looptijd en bij langere verbindingen zouden we deze dus moeten meenemen. Bij snelle logica moet dit aspect zeker bekeken worden. Meestal is er een correctieterm in (17.11) nodig van enkele ns. Wij hebben eerder verondersteld dat looptijden bevat zijn in de benodigde marges voor clock skew. Daarnaast

vormen verbindingen ook een (capacitieve) belasting voor de sturende uitgangen. Stilzwijgend hebben we verondersteld dat de invloed hiervan op de propagatietijd is meegerekend in de propagatietijden van de flip-flops en de logica. Intern in IC's werkt men met een *layout afhankelijk tijdsmodel*. Dan kan de invloed van de layout nauwkeuriger verdisconteerd worden. Op deze details gaan we hier niet verder in. Deze problematiek is voor de technoloog  $\square$



**Figuur 17.6.** Data-overdracht van flip-flop FF1 via logica naar flip-flop FF2.