



Academie voor Technology, Innovation &
Society Delft
Academie voor ICT & Media

Digitale System Engineering 1

Week 5 – Timing, dataoverdracht
Jesse op den Brouw
DIGSE1/2018-2019

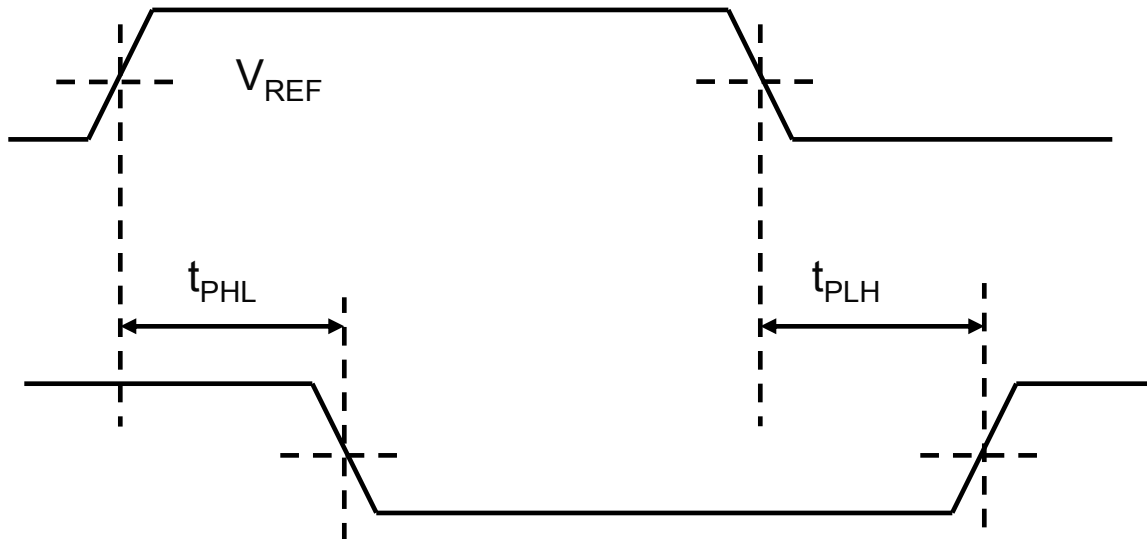
DE HAAGSE
HOGESCHOOL

Timing (revisited)

- Een verandering op de ingang van een component geeft als resultaat een verandering op de uitgang.
- Dit gaat niet oneindig snel; de transistoren in een IC hebben parasitaire capaciteiten en die moeten worden opgeladen of ontladen.
- Het duurt dus even voordat de uitgangsverandering zichtbaar wordt.
- Deze tijd wordt vertragingstijd (*propagation delay*) genoemd.

Timing

- De fabrikant geeft in de *datasheet* hoeveel deze tijden bedragen, met meestal daarbij de meetopstelling en -instellingen.
- De relaties tussen ingangen en uitgangen worden getekend in een tijdvolgordediagram (timing diagram).



V_{REF} = de referentiespanning voor de tijdmeting (1,5 V)

t_{PHL} = vertragingstijd voor uitgang hoog naar laag

t_{PLH} = vertragingstijd voor uitgang laag naar hoog

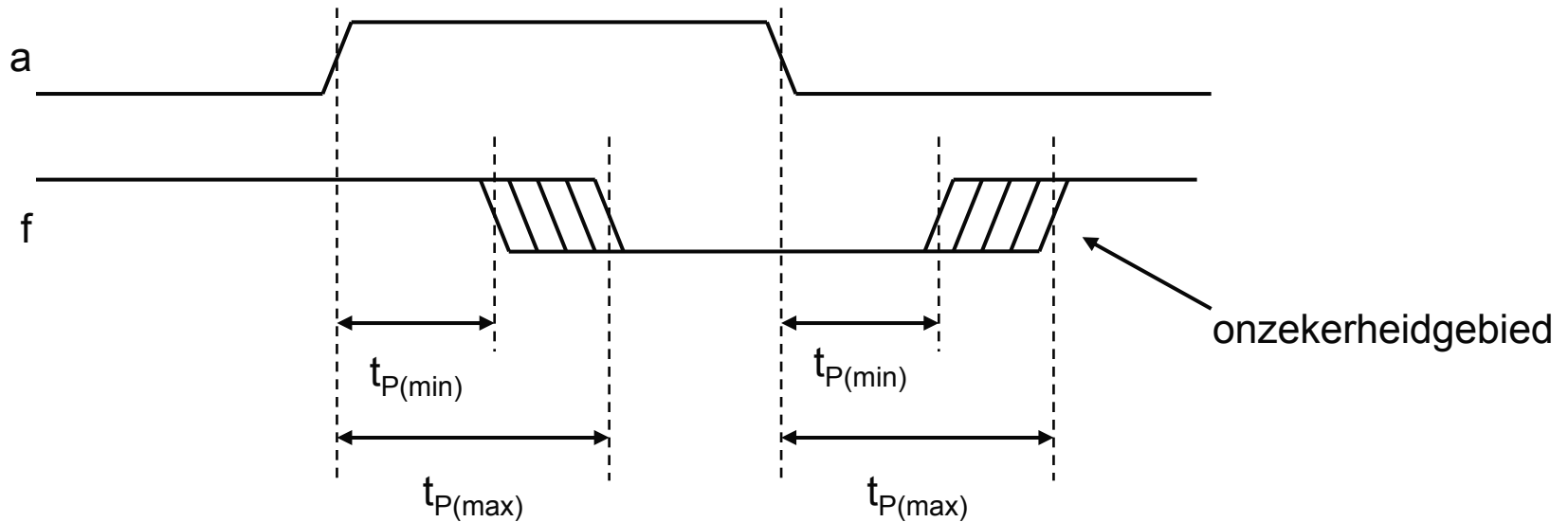
Timing

- Bij componenten (losse poorten, library-modules) worden de volgende tijden vermeld:
- $t_{P(\min)}$ De fabrikant garandeert dat de uitgang nooit eerder verandert dan de minimale vertragingstijd.
- $t_{P(\max)}$ De fabrikant garandeert dat de uitgang nooit later verandert dan de maximale vertragingstijd.
- $t_{P(\text{typ})}$ De typische vertragingstijd, “nice to know”, maar er kan niet mee ontworpen worden.

nb: in plaats van t_p ook wel t_{pD}

Timing

- Hieronder het timingdiagram van een NOT-poort.



- In het gearceerde gedeelte is de definitieve waarde van de uitgang nog onzeker.

Minimale pulsbreedte

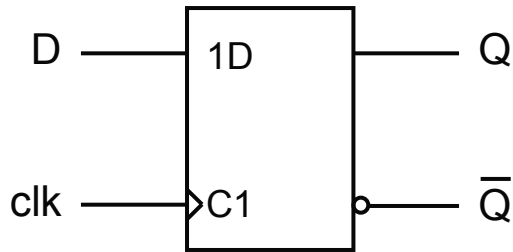
- De transistoren en verbindingen op een IC hebben allemaal last van *paracitaire capaciteiten*.
- Deze paracitaire capaciteiten moeten bij het schakelen geladen en ontladen worden en dat kost enige tijd.
- De sturende puls levert de energie op de capaciteiten te laden en moet een *minimale pulsbreedte* hebben.
- Pulsen korter dan deze minimale pulsbreedte worden door de logische poort niet 'gezien'.

Edge triggered D-flipflop

- De digitale component die verreweg het meest wordt gebruikt voor het opslaan van logische waarden is de *edge triggered D-flipflop*.
- Deze component neemt de data op de D-ingang over op een flank van het *kloksignaal*.
- Het kloksignaal is een hulpsignaal en heeft geen logische waarde.
- Dit overnemen wordt ook wel inklokken genoemd.

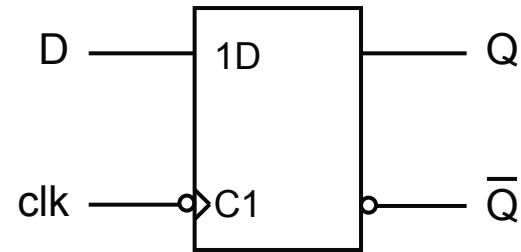
Symbolen edge triggered D-flipflops

- Er zijn twee mogelijkheden: de opgaande flank (positive edge) en neergaande flank (negative edge).



positive edge-triggered
D-flipflop

werking D afhankelijk
van de opgaande flank
van C



negative edge-triggered
D-flipflop

werking D afhankelijk
van de neergaande flank
van C

Timing positive edge triggered D-flipflop

- De *setuptijd* t_{su} is de tijd waarbij de waarde van de D-ingang stabiel moet zijn vóórdat de klok van 0 naar 1 gaat.
- De *holdtijd* t_h is de tijd waarbij de waarde van de D-ingang stabiel moet blijven nádat de klok van 0 naar 1 gaat.
- In het gebied $t_{su} - t_h$ moet D dus stabiel blijven.
- Verandering op de D-ingang tijdens $t_{su} - t_h$ kan leiden tot het correct overnemen van de data, oscilleren of tot *metastabiliteit*.

Timing positive edge triggered D-flipflop

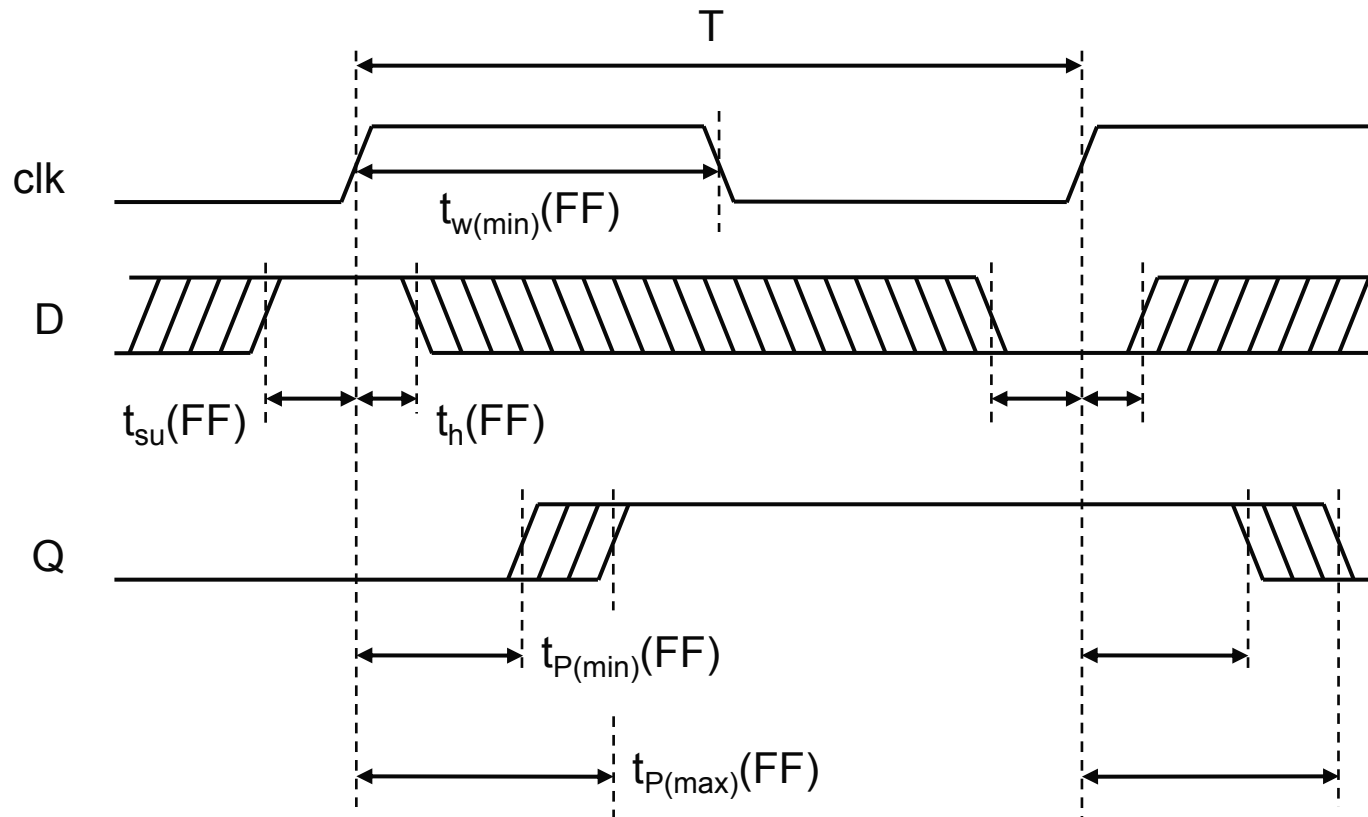
- De *minimale propagatietijd* $t_{P(\min)}(\text{FF})$ is de tijd na de actieve klokflank waarbij de waarde van de uitgang nog niet verandert.
- De *maximale propagatietijd* $t_{P(\max)}(\text{FF})$ is de tijd na de actieve klokflank waarbij de waarde van de uitgang de definitieve waarde heeft aangenomen.
- In het gebied $t_{P(\min)}(\text{FF}) - t_{P(\max)}(\text{FF})$ is de definitieve waarde van de uitgang nog onzeker.
- Noot: in veel engelstalige documentatie wordt gesproken over t_{co} (clock-to-output). Dit is (bijna) altijd de maximale propagatietijd.

Timing positive edge triggered D-flipflop

- De *minimale pulsbreedteduur* $t_{w(\min)}(\text{FF})$ is de tijd die het kloksignaal logisch 1 moet zijn zodat de flipflop betrouwbaar data kan overnemen.
- De *periodeduur* T is de tijd tussen twee actieve klokflanken.
- Er zijn ook timingeigenschappen voor asynchrone reset en preset. Die worden hier niet besproken.
- Het geheel aan timingparameters kan worden getekend in een timingdiagram.

Timing positive edge triggered D-flipflop

- Hieronder de timing van de positive edge triggered D-flipflop.



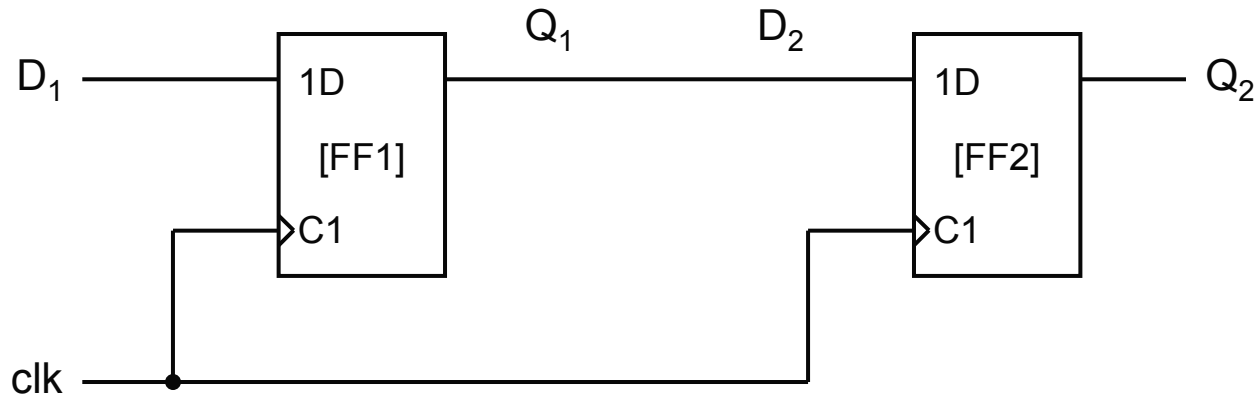
Timing edge triggered D-flipflop

- De tijden op een rijtje

$t_{P(\min)}(\text{FF})$	De minimale vertragingstijd van de flipflop t.o.v. de actieve klokflank.
$t_{P(\max)}(\text{FF})$	De maximale vertragingstijd van de flipflop t.o.v. de actieve klokflank.
$t_{\text{su}}(\text{FF})$	De setuptijd van de D-ingang van de flipflop t.o.v. de actieve klokflank.
$t_{\text{h}}(\text{FF})$	De holdtijd van de D-ingang van de flipflop t.o.v. de actieve klokflank.
$t_{w(\min)}(\text{FF})$	De minimale pulsbreedteduur van het kloksignaal.
T	De periodeduur van het kloksignaal.

Directe dataoverdracht

- Bij directe dataoverdracht tussen twee flipflops is de uitgang van de eerste flipflop direct verbonden met de ingang van de tweede flipflop.



- De flipflops hebben dezelfde timing eigenschappen (bv. op hetzelfde IC).
- Directe overdracht, dus $D_2 = Q_1$.

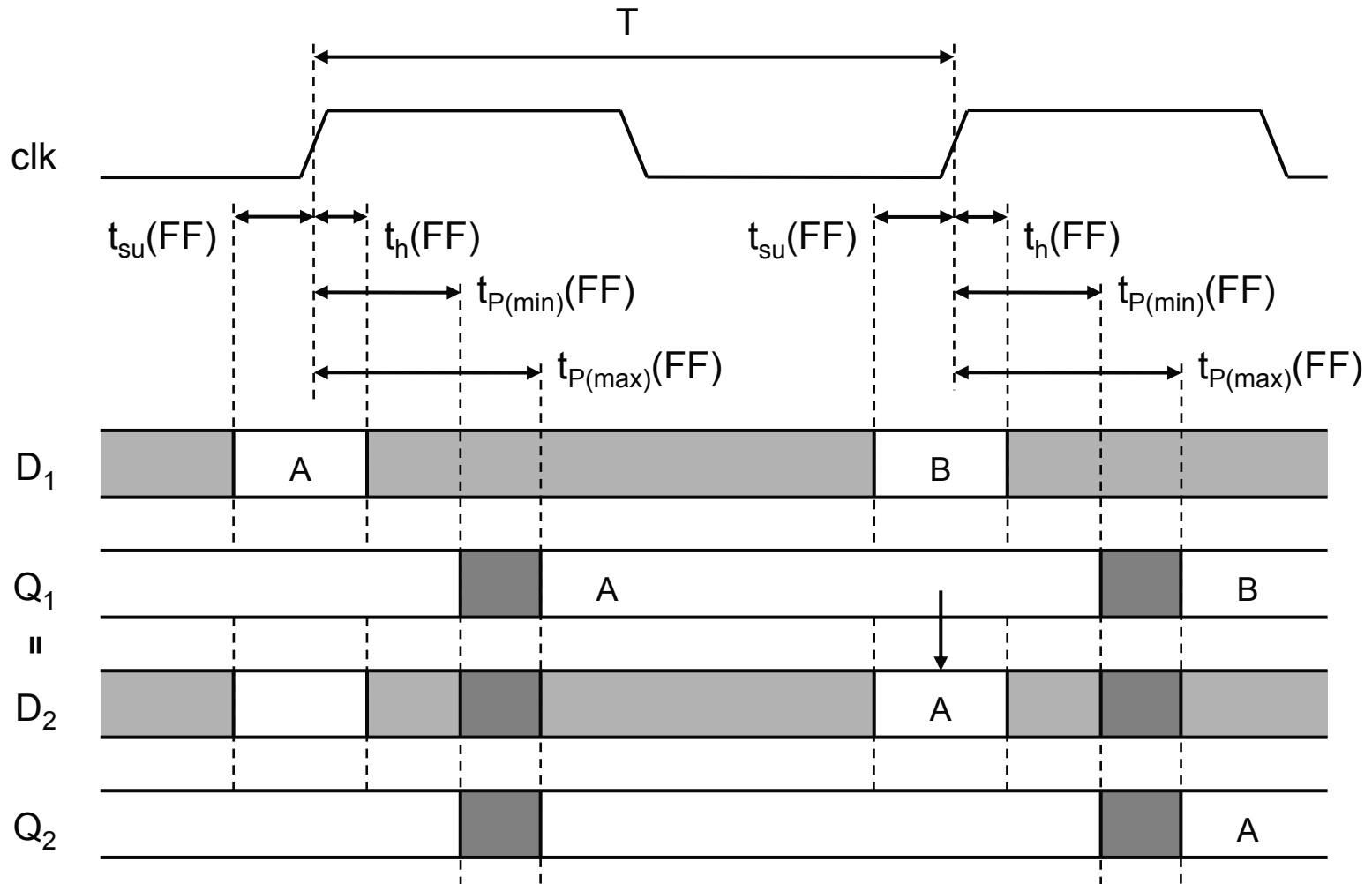
Directe dataoverdracht

- Voor de timing bij directe dataoverdracht kan het volgende vermeld worden.
- *Alle timingparameters worden gegeven ten opzichte van de actieve klokflank.*
- Voor flipflop 1 geldt dat D_1 stabiel moet zijn rond de actieve flank, tijdens het interval $t_{su} - t_h$.
- Uitgang Q_1 van flipflop 1 verandert in het gebied $t_{P(\min)}(FF) - t_{P(\max)}(FF)$.

Directe dataoverdracht

- Voor flipflop 2 geldt dat D_2 stabiel moet zijn rond de actieve klokflank, dus tijdens het interval $t_{su} - t_h$.
- Uitgang Q_2 van flipflop 2 verandert in het gebied $t_{P(\min)}(\text{FF}) - t_{P(\max)}(\text{FF})$.
- Het geheel kan worden getekend in een timingdiagram.

Timing directe dataoverdracht

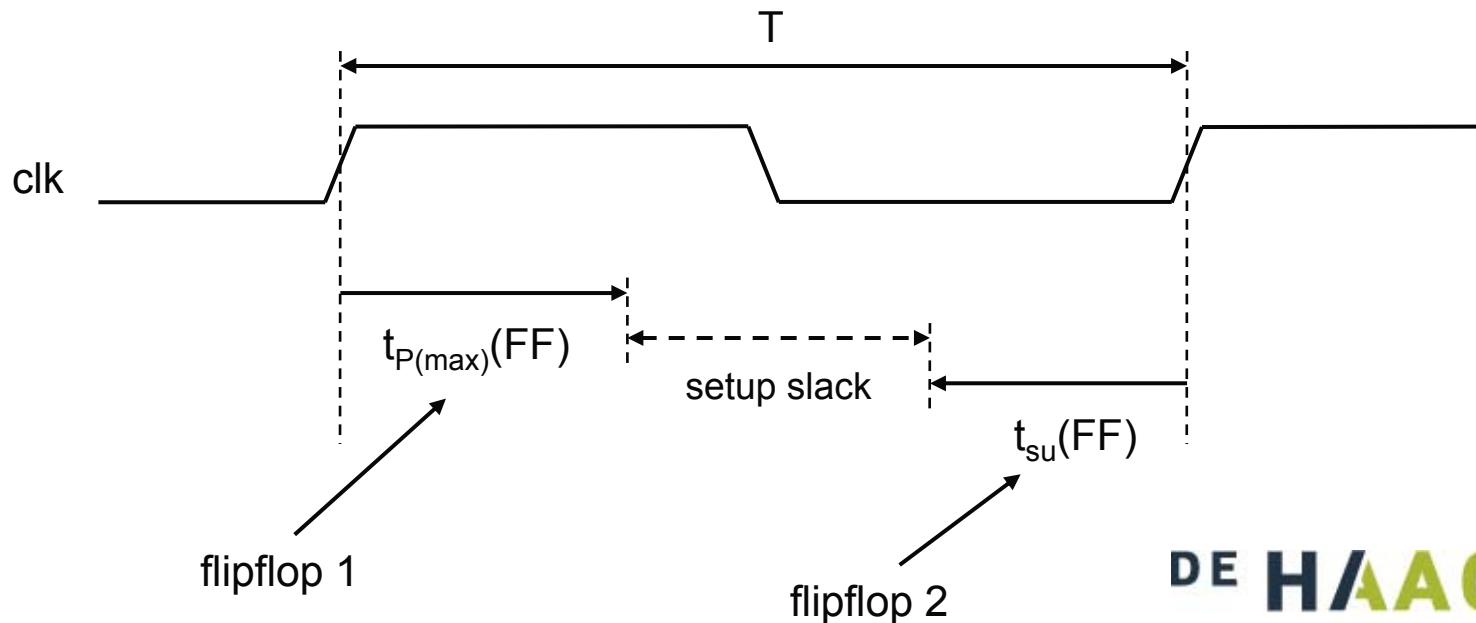


Voorwaarden betrouwbare dataoverdracht

- We willen graag *betrouwbare dataoverdracht* tussen de twee flipflops hebben.
- Voor flipflop 2 geldt dat D_2 stabiel moet zijn tijdens het interval $t_{su} - t_h$.
- Dat betekent dat de uitgang Q_1 van flipflop 1 dus stabiel moet zijn tijdens het gebied $t_{su} - t_h$.
- Hieruit volgen *twee* voorwaarden.

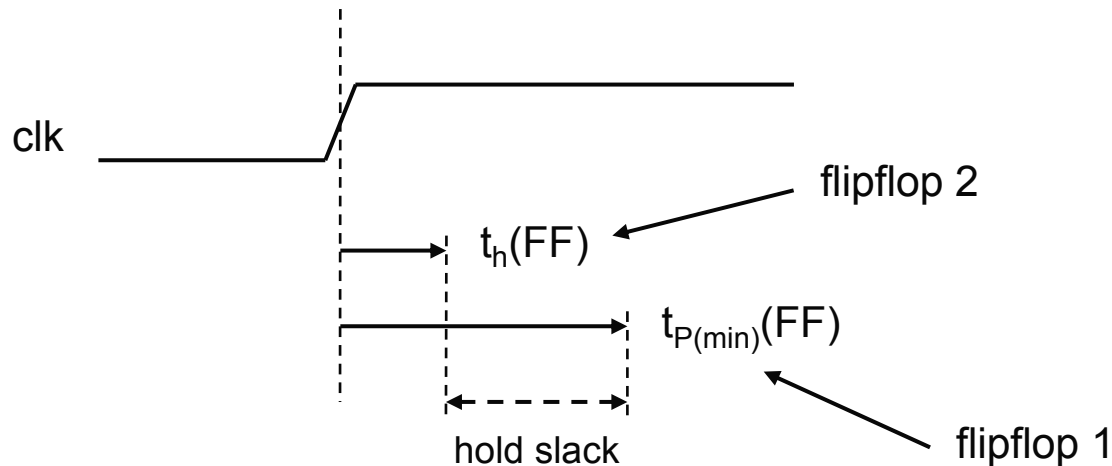
Voorwaarden betrouwbare dataoverdracht

- Uitgang Q_1 van flipflop 1 moet stabiel zijn *voordat* de setuptijd t_{su} van flipflop 2 ingaat.
- Dat houdt in dat de setuptijd $t_{su}(FF)$ van flipflop 2 pas mag ingaan *na* de maximale propagatietijd $t_{P(max)}(FF)$ van flipflop 1.



Voorwaarden betrouwbare dataoverdracht

- Uitgang Q_1 van flipflop 1 mag pas weer veranderen (niet stabiel zijn) *nadat* de holdtijd t_h van flipflop 2 is verstreken.
- Daaruit volgt dat de minimale propagatietijd $t_{P(\min)}(\text{FF})$ van flipflop 1 *groter* moet zijn dan de holdtijd $t_h(\text{FF})$ van flipflop 2.



Setup en hold slack

- In de voorgaande slides zijn twee termen gebruikt die nadere toelichting vereisen:
- *Setup slack*[†]: de tijd die “over” is na het verstrijken van de maximale propagatietijd van de eerste flipflop en voordat de setuptijd van de tweede flipflop ingaat.
- *Hold slack*[†]: de tijd die “over” is na het verstrijken van de holdtijd van de tweede flipflop en het verstrijken van de minimale propagatietijd van de eerste flipflop.

[†] De termen worden gebruikt in de Quartus Timing Analyzer

Voorwaarden betrouwbare dataoverdracht

- De eerste voorwaarde is *frequentie-afhankelijk* en kan worden uitgedrukt in de maximale propagatietijd van eerste flipflop, de setup slack en de setuptijd van de tweede flipflop:

$$T = t_{P(\max)}(\text{FF}) + t_{\text{slack,su}} + t_{\text{su}}(\text{FF}) \quad [\text{s}]$$

$$f = \frac{1}{t_{P(\max)}(\text{FF}) + t_{\text{slack,su}} + t_{\text{su}}(\text{FF})} \quad [\text{Hz}]$$

- Dit is de *standalone systeemfrequentie*.
- Als de setup slack 0 is, wordt dit de *maximale standalone systeemfrequentie* genoemd.

Voorwaarden betrouwbare dataoverdracht

- De tweede voorwaarde is *frequentie-onafhankelijk* en kan worden uitgedrukt in de minimale propagatietijd van eerste flipflop, de hold slack en de holdtijd van de tweede flipflop:

$$t_{P(\min)}(\text{FF}) = t_h(\text{FF}) + t_{\text{slack,h}} \quad [\text{s}]$$

Voorwaarden betrouwbare dataoverdracht

- Aan de voorwaarden voor betrouwbare dataoverdracht wordt voldaan als de tijden van de setup slack en hold slack groter zijn dan of gelijk zijn aan 0:

setup slack: $t_{\text{slack,su}} = T - t_{P(\text{max})}(\text{FF}) - t_{\text{su}}(\text{FF}) \geq 0$

hold slack: $t_{\text{slack,h}} = t_{P(\text{min})}(\text{FF}) - t_{\text{h}}(\text{FF}) \geq 0$

Opgave

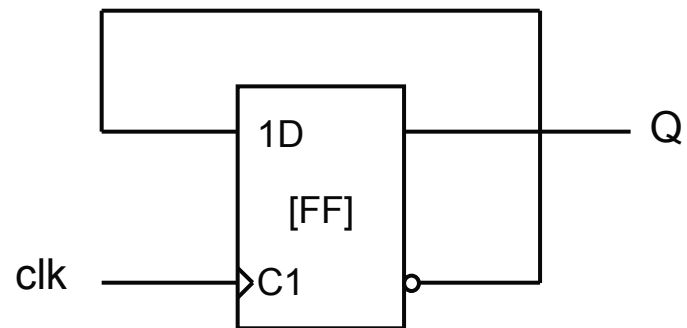
- Van onderstaande tweedeler zijn de volgende timingparameters gegeven.

$$t_{su}(FF) = 15 \text{ ns}$$

$$t_h(FF) = 5 \text{ ns}$$

$$t_{P(\min)}(FF) = 10 \text{ ns}$$

$$t_{P(\max)}(FF) = 35 \text{ ns}$$



Bepaal de maximale frequentie waarop dit systeem kan werken.
Is betrouwbare dataoverdracht mogelijk?

Voorwaarden betrouwbare dataoverdracht

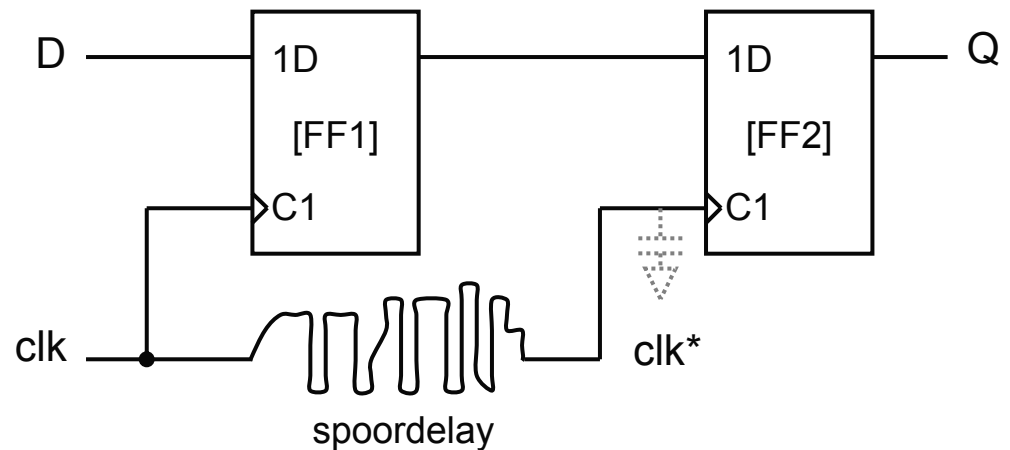
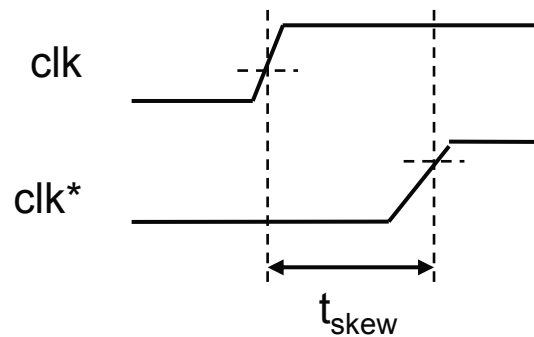
- Er zijn nog aanvullende eisen met betrekking op de vorm van het kloksignaal en de interne werking van de flipflop.
- Het kloksignaal moet *zonder vertraging* worden aangeboden aan beide flipflops.
- De klokflanken die aan de flipflops worden aangeboden moeten *identieke vorm* hebben (niet vervormd). Het vervormen gebeurt onder andere door *capacitieve belasting*.
- Beide flipflops moeten op *hetzelfde (spannings)niveau* van de actieve klokflank data inklokken.

Voorwaarden betrouwbare dataoverdracht

- In de praktijk wordt hier nooit aan voldaan.
- Vooral op grote IC's is vertraging in de kloklijn een lastig probleem.
- Op grote IC's wordt een speciaal klokdistributiesysteem aangelegd om vertragingstijden en belasting te minimaliseren.
- Daarnaast worden Phase Lock Loops (PLL) gebruikt om klokvertragingen te minimaliseren.
- Het geheel van vertraging, vervorming door capacitieve belasting en triggerpunt wordt *clock skew* genoemd.

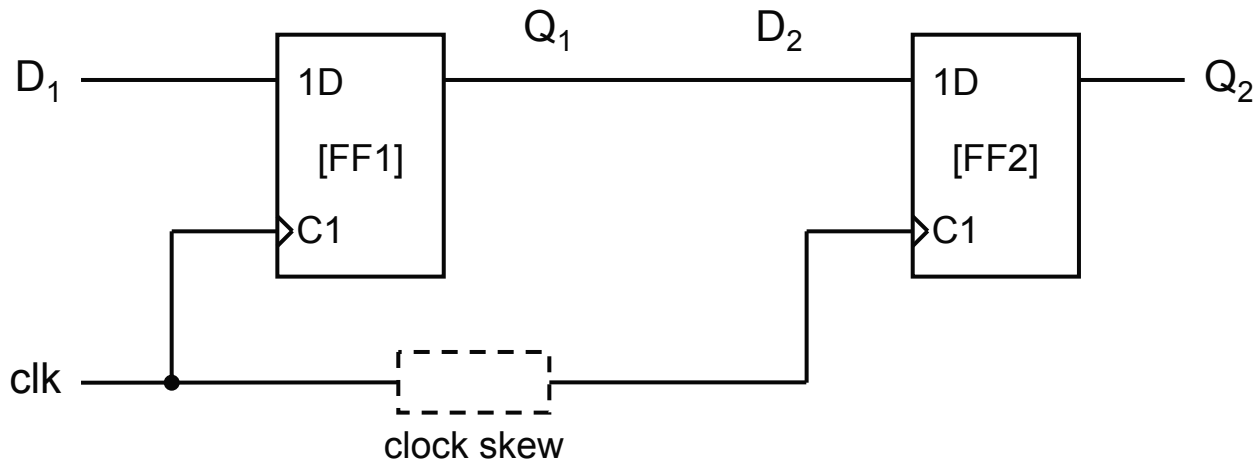
Clock skew

- *In elk digitaal systeem is dus sprake van clock skew.*
- De timing moet worden uitgebreid met een extra parameter t_{skew} .



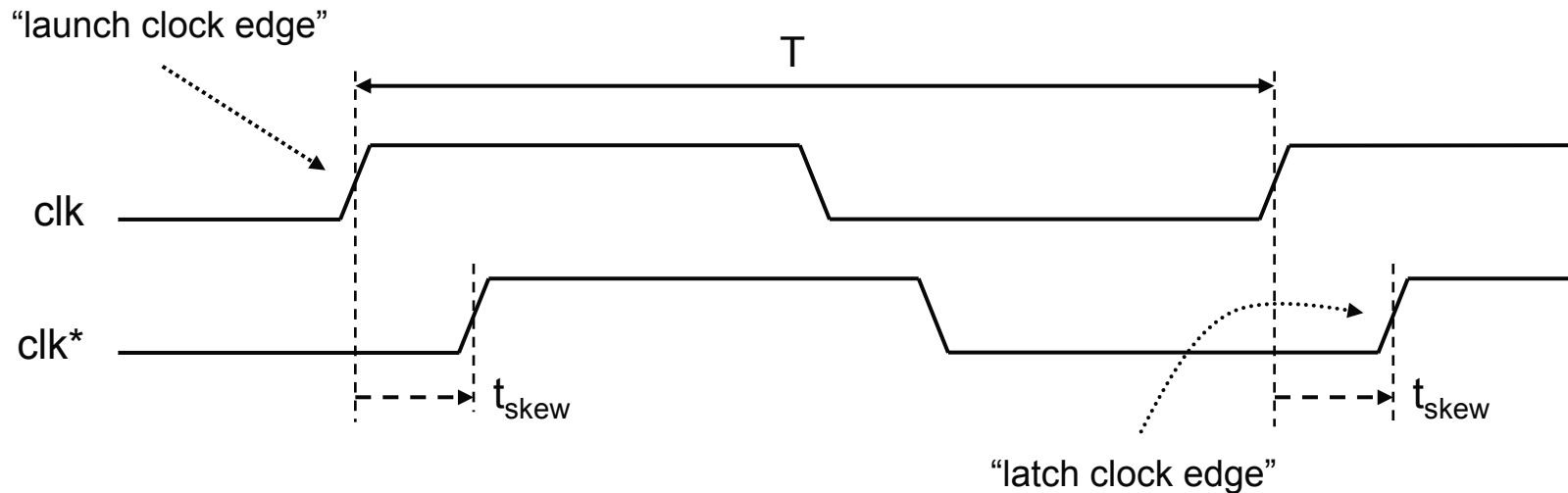
Clock skew

- Gegeven onderstaand schema. Het kloksignaal van FF2 is iets vertraagd. We kunnen de frequentie uitdrukken in de timing-parameters van de flipflops en de clock skew.



Clock skew

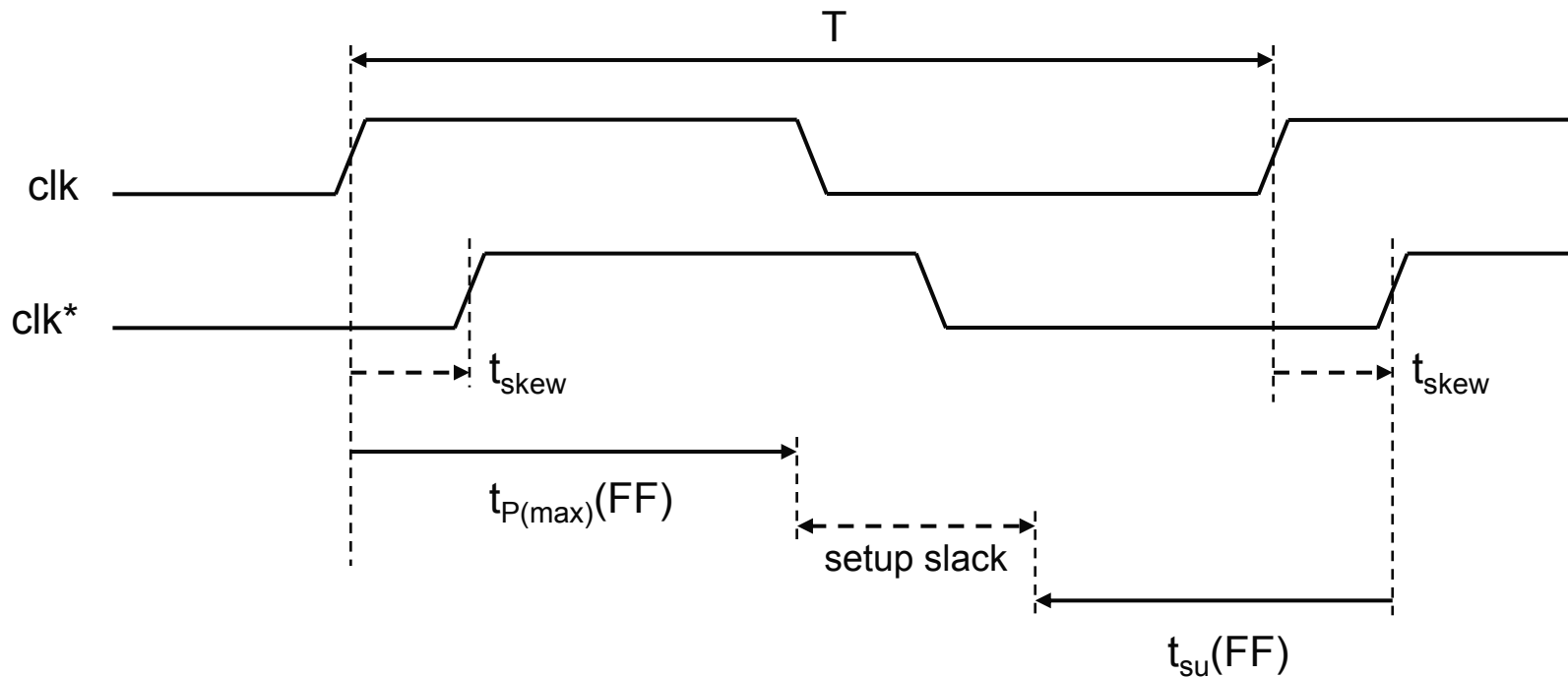
- Om de twee klokflanken uit elkaar te houden wordt gesproken van een *launch clock edge*[†] en een *latch clock edge*[†]. Onderstaande tekening verduidelijkt de twee begrippen.



[†] De termen worden gebruikt in de Quartus Timing Analyzer

Clock skew

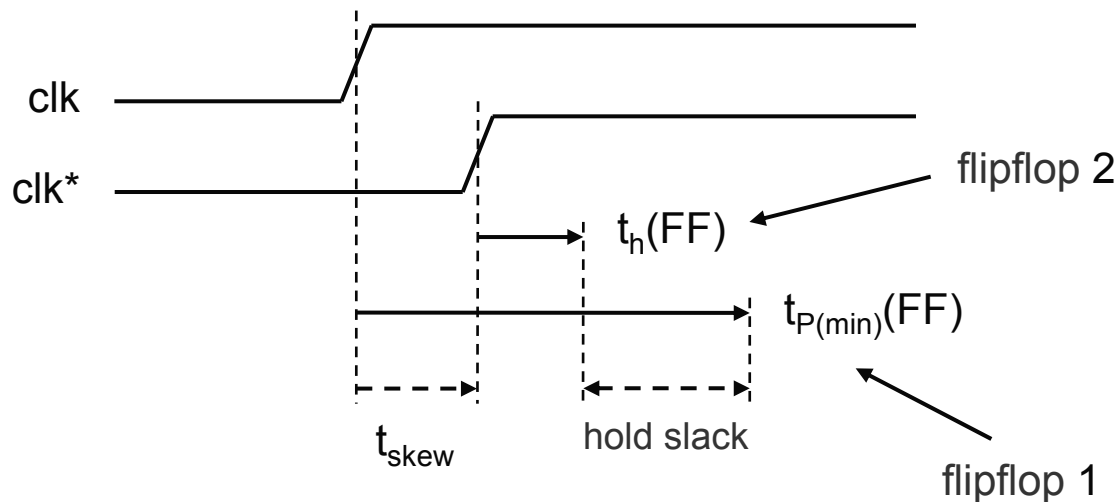
- We tekenen een timingdiagram en leiden de formule voor T af. Clock skew heeft (in dit geval) *positieve* invloed op de *setup slack*.



$$T = t_{P(max)}(FF) + t_{su}(FF) + t_{slack,su} - t_{skew}$$

Clock skew

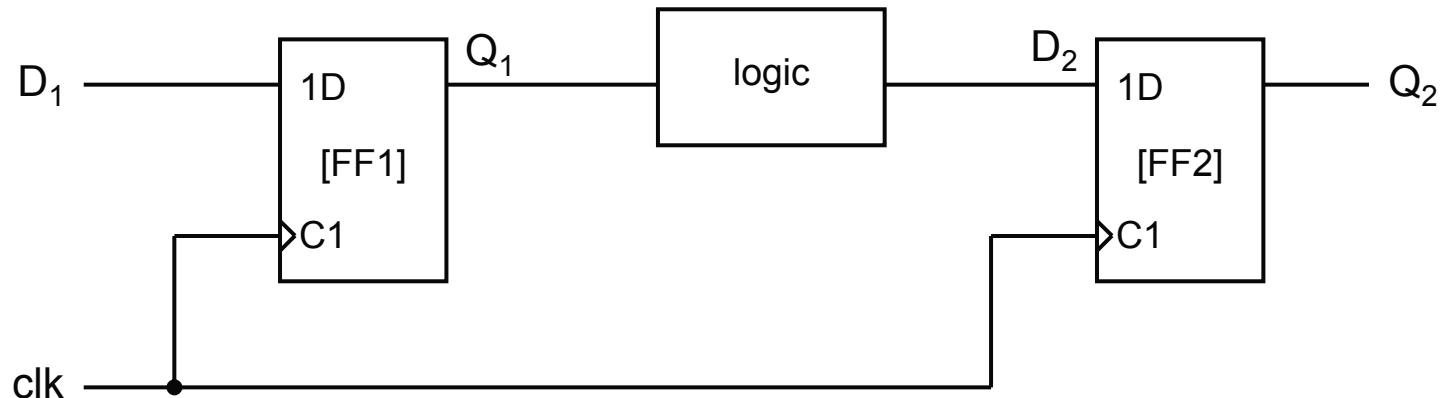
- Clock skew heeft (in dit geval) *negatieve* invloed op de hold slack. De uitgang van de eerste flipflop mag pas veranderen na de holdtijd van de tweede flipflop. Deze holdtijd wordt “verlengd” door de clock skew.



$$t_{P(min)}(FF) = t_{skew} + t_h(FF) + t_{slack,h}$$

Indirecte dataoverdracht

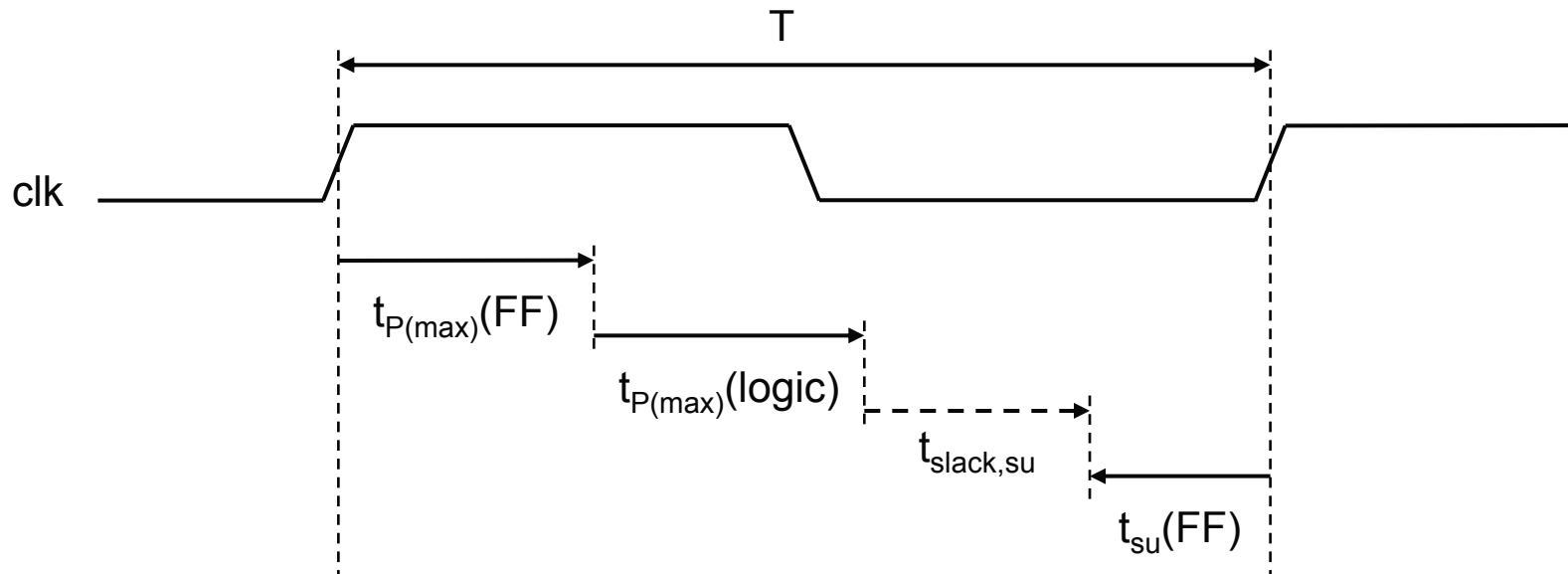
- Bij indirecte dataoverdracht ligt tussen de twee flipflops nog een stuk combinatorische logica.
- De uitgang van de eerste flipflop is verbonden met de ingang van de logica en de uitgang van de logica is verbonden met de ingang van de tweede flipflop.



nb: timing flipflops zijn identiek

Indirecte dataoverdracht

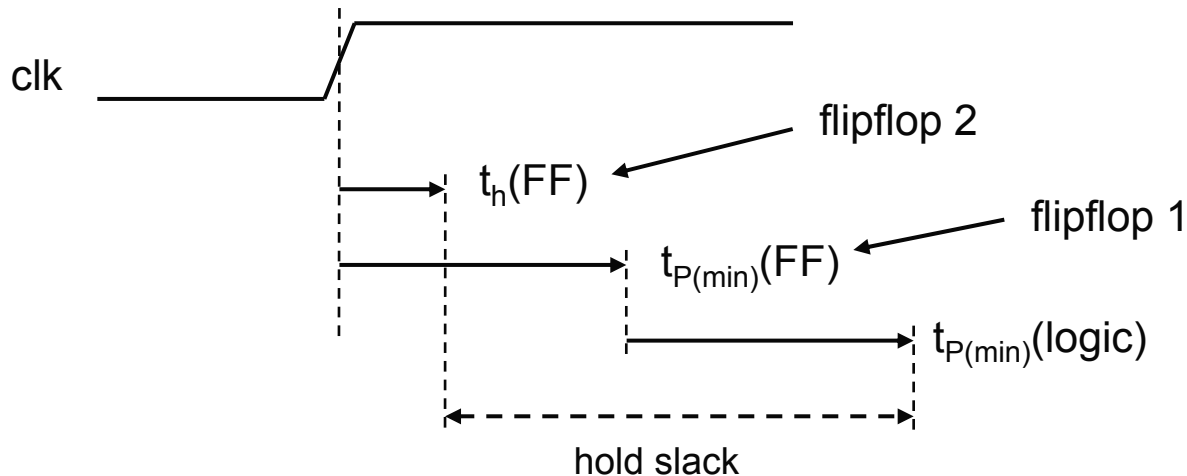
- Voor het berekenen van de periodeduur T moet nu de maximale propagatietijd van de logica $t_{P(\max)}(\text{logic})$ meegenomen worden.



$$T = t_{P(\max)}(\text{FF}) + t_{P(\max)}(\text{logic}) + t_{\text{slack,su}} + t_{\text{su}}(\text{FF})$$

Indirecte dataoverdracht

- De waarde van de uitgang van de eerste flipflop verandert op z'n vroegst na de minimale vertragingstijd van de flipflop. Daarna gaat het signaal door de logica. Op z'n vroegst verandert de uitgang van de logica na de minimale vertragingstijd van de logica. Deze twee tijden opgeteld moet groter zijn dan de holdtijd.



$$t_{p(\text{min})}(\text{FF}) + t_{p(\text{min})}(\text{logic}) = t_h(\text{FF}) + t_{\text{slack},h}$$

Maximale frequentie indirecte dataoverdracht

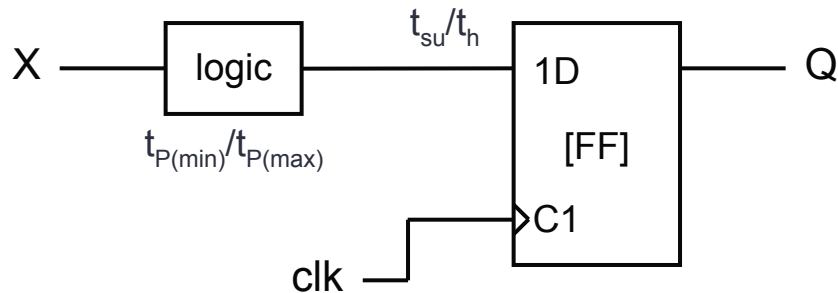
- Een belangrijke parameter van een digitale systeem is de *maximale frequentie* waarop het systeem nog betrouwbaar werkt.
- Hierbij wordt setup slack op 0 gesteld.
- De inverse van de minimale periodetijd is de maximale frequentie.

$$f_{\max} = \frac{1}{T_{\min}} = \frac{1}{t_{P(\max)}(\text{FF}) + t_{P(\max)}(\text{logic}) + t_{\text{su}}(\text{FF})} \quad [\text{Hz}]$$

Noot: clock skew is niet in de formule verwerkt.

Setup- en holdtijd van voorzetlogica

- Als er logica geplaatst wordt voor de ingang van de flipflop, heeft dit gevolgen voor de timingparameters van de sturende ingang.



- De logica heeft een zekere vertragingstijd $t_{P(min)}$ (logic) en $t_{P(max)}$ (logic).
- Voor signaal X gelden andere setup- en holdtijden dan voor de D-ingang van de flipflop.

Setup- en holdtijd van voorzetlogica

- Voorwaarde is natuurlijk dat de setup- en holdtijden van de D-ingang gerespecteerd worden.
- Een verandering op signaal X heeft maximaal $t_{P(\max)}(\text{logic})$ nodig.
- De laatst mogelijke verandering op X moet dus $t_{P(\max)}(\text{logic})$ vóór $t_{su}(\text{FF})$ gebeuren.
- Dus $t_{su}(X - \text{to} - \text{clk}) = t_{su}(\text{FF}) + t_{P(\max)}(\text{logic})$

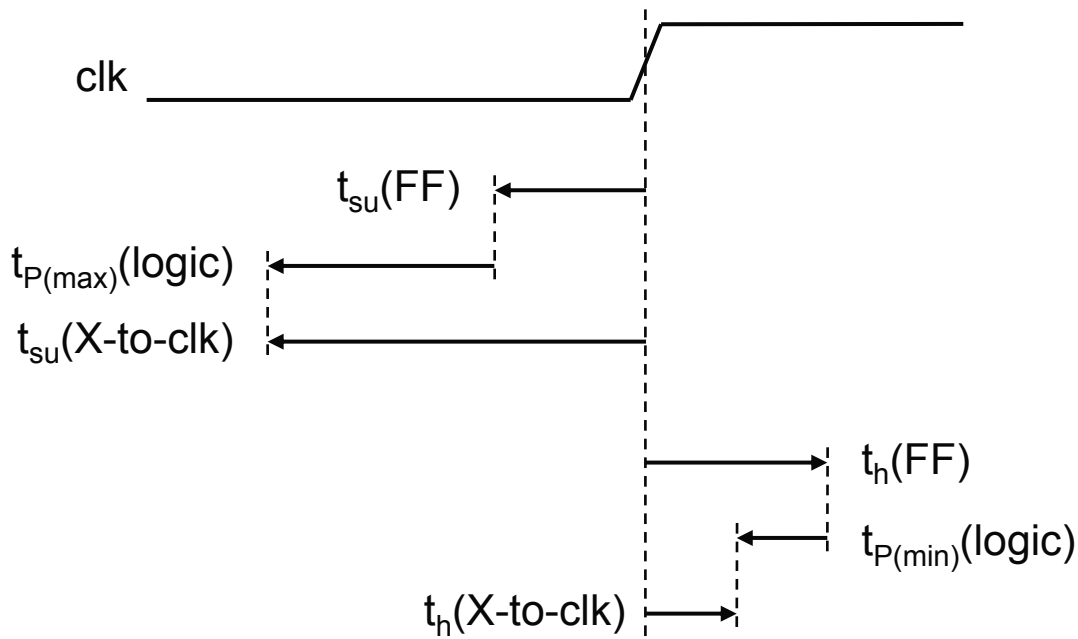
Setup- en holdtijd van voorzetlogica

- De waarde op de D-ingang van de flipflop na de actieve flank nog even stabiel blijven, de holdtijd $t_h(\text{FF})$.
- Een verandering op signaal X heeft minimaal $t_{P(\text{min})}(\text{logic})$ nodig. Dan pas mag de holdtijd $t_h(\text{FF})$ van de flipflop verstreken zijn.
- De eerst mogelijke (nieuwe) verandering op X mag dus $t_{P(\text{min})}(\text{logic})$ vóór $t_h(\text{FF})$ gebeuren.
- Dus $t_h(X - \text{to} - \text{clk}) = t_h(\text{FF}) - t_{P(\text{min})}(\text{logic})$

Let op: min-teken in formule!

Setup- en holdtijd van voorzetlogica

- In onderstaande timingdiagram worden de setup- en holdtijden van de logica t.o.v de klok weergegeven.



$$t_{su}(X - to - clk) = t_{su}(FF) + t_{P(max)}(logic)$$

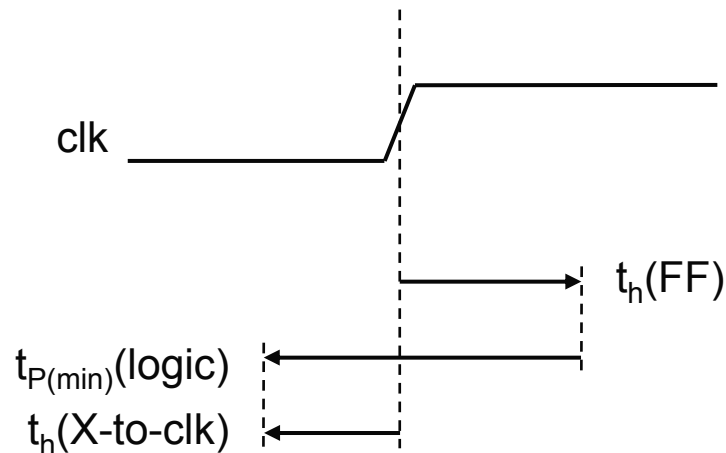
$$t_h(X - to - clk) = t_h(FF) - t_{P(min)}(logic)$$

Setup- en holdtijd van voorzetlogica

- Voor de holdtijd kan nog een opmerking worden gemaakt.
- In veel gevallen is de minimale propagatietijd $t_{P(\min)}(\text{logic})$ *groter* dan de holdtijd $t_h(\text{FF})$.
- De berekening van de holdtijd $t_h(\text{X-to-clk})$ levert dan een *negatief* antwoord.
- Dit houdt in dat het signaal op de ingang van de logica *voor* de klokflank al weer mag veranderen.

Setup- en holdtijd van voorzetlogica

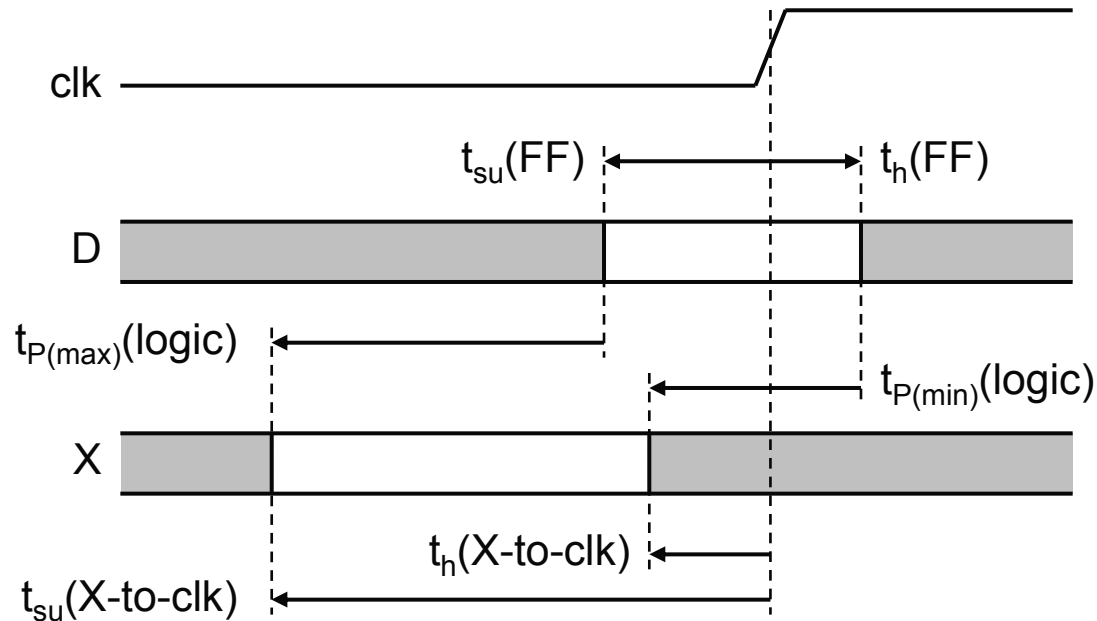
- Een negatief antwoord voor de holdtijd $t_h(X\text{-to-clk})$ betekent dus dat het signaal nog voor de klokflank mag veranderen.



$$t_h(X - \text{to} - \text{clk}) = t_h(\text{FF}) - t_{P(\text{min})}(\text{logic})$$

Setup- en holdtijd van voorzetlogica

- Het gebied waarin X stabiel moet zijn is groter dan dat van D.



Let op: holdtijd X t.o.v. clk is negatief!

Timing uitgangslogica

- Aan de uitgang van een flipflop kan logica worden toegevoegd.
- De minimale propagatietijd van de uitgang is de som van de minimale propagatietijden van de flipflop en de logica:

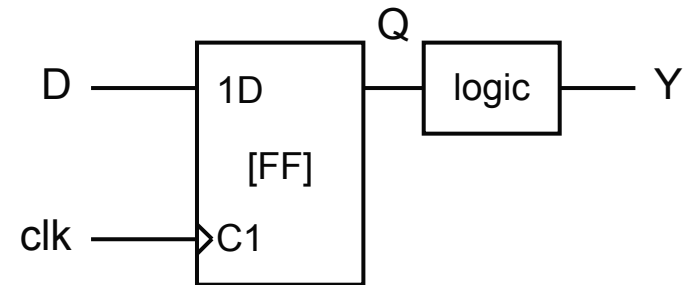
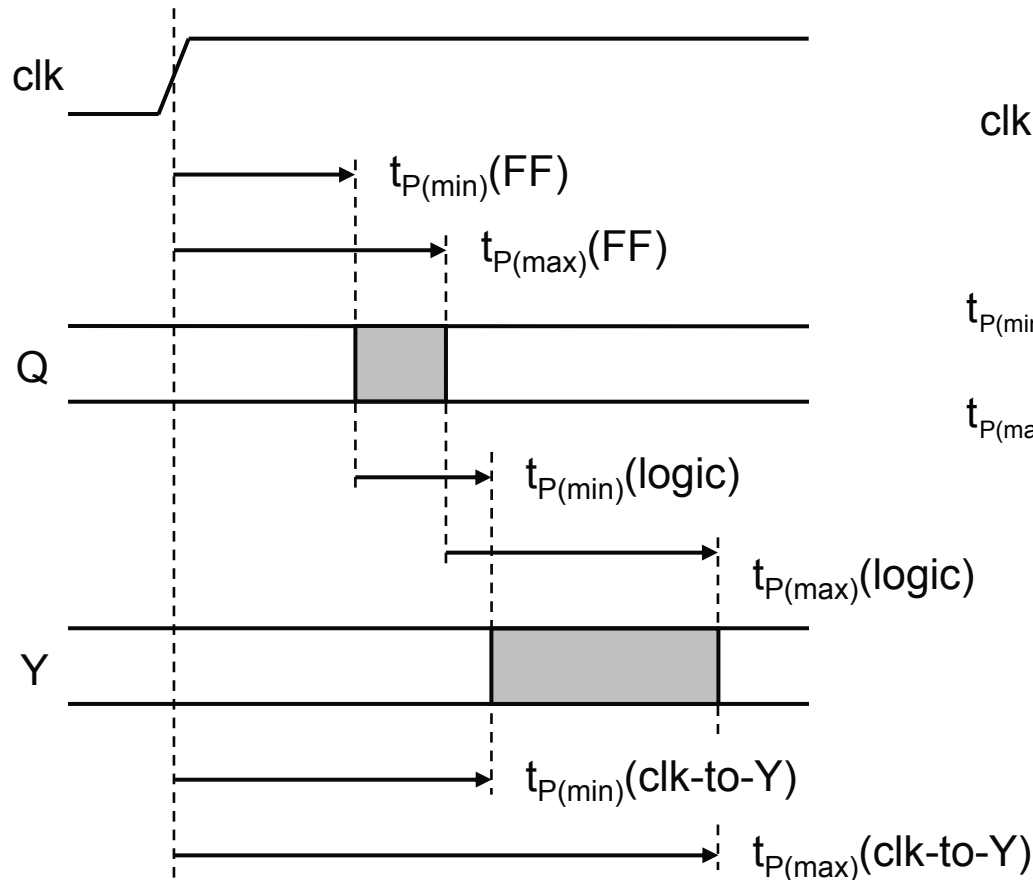
$$t_{P(\min)}(\text{out}) = t_{P(\min)}(\text{FF}) + t_{P(\min)}(\text{logic})$$

- De maximale propagatietijd van de uitgang is de som van de maximale propagatietijden van de flipflop en de logica:

$$t_{P(\max)}(\text{out}) = t_{P(\max)}(\text{FF}) + t_{P(\max)}(\text{logic})$$

Timing uitgangsl logica

- Hieronder het timingdiagram.

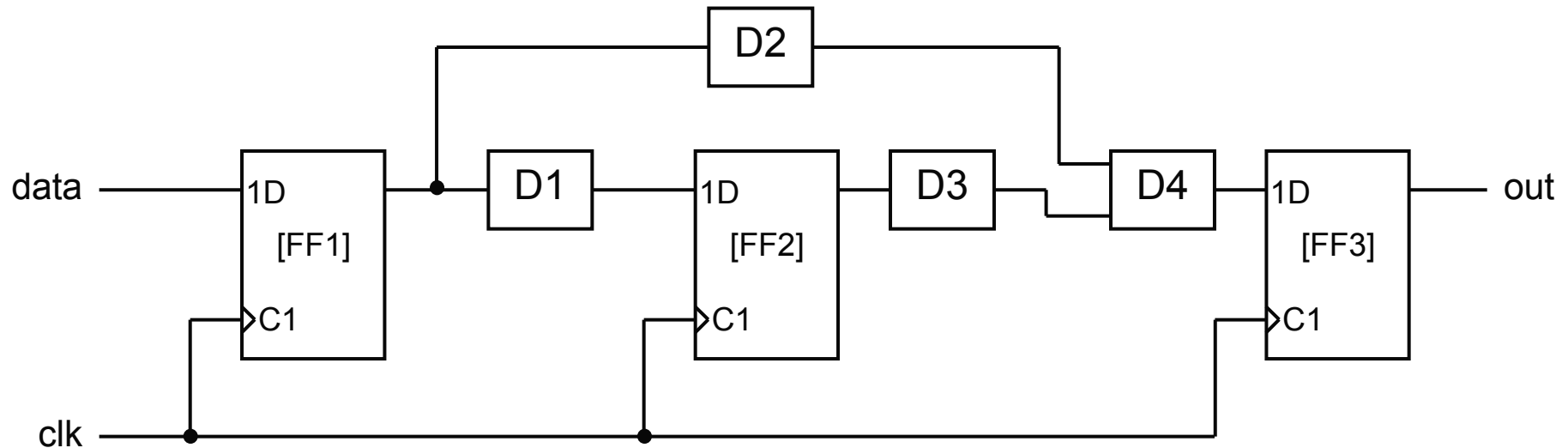


$$t_{P(\min)}(clk - to - Y) = t_{P(\min)}(FF) + t_{P(\min)}(logic)$$

$$t_{P(\max)}(clk - to - Y) = t_{P(\max)}(FF) + t_{P(\max)}(logic)$$

Bepalen maximale frequentie

- In een digitaal systeem zijn er meerdere paden tussen de flipflops. Voor het bepalen van de maximale frequentie (minimale periodetijd) moeten alle paden doorgerekend worden. De timing van de flipflops is identiek.



Bepalen maximale frequentie

- In de schakeling zijn drie paden te ontdekken waarlangs data van flipflop naar flipflop wordt getransporteerd.

FF1 → D1 → FF2

FF1 → D2 → D4 → FF3

FF1 → D3 → D4 → FF3

- Elk pad levert allemaal een minimale periodetijd behorende bij het pad.
- De *grootste* minimale periodetijd bepaalt de maximale frequentie van het hele systeem.

Bepalen maximale frequentie

- De paden:

$$T_{\min_1} = t_{P(\max)}(FF) + t_{P(\max)}(D1) + t_{su}(FF)$$

$$T_{\min_2} = t_{P(\max)}(FF) + t_{P(\max)}(D2) + t_{P(\max)}(D4) + t_{su}(FF)$$

$$T_{\min_3} = t_{P(\max)}(FF) + t_{P(\max)}(D3) + t_{P(\max)}(D4) + t_{su}(FF)$$

- De minimale periodetijd van het hele systeem:

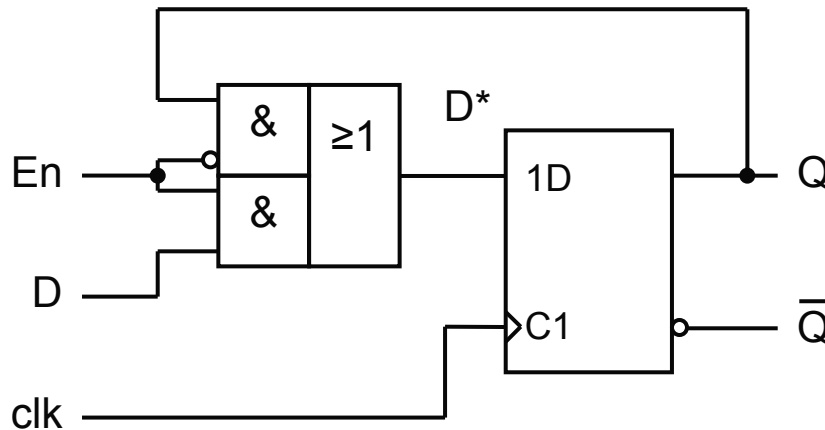
$$T_{\min_{\text{stroom}}} = \max(T_{\min_1}, T_{\min_2}, T_{\min_3})$$

Voorwaarden betrouwbare dataoverdracht

- Het langzaamste pad bepaalt de maximale frequentie. Dit wordt het kritieke pad genoemd. De *setup slack* van dit pad is 0.
- In geen enkel pad mag een *hold time violation* zijn, de *hold slack* moet ≥ 0 zijn.
 - Dit kan eventueel opgelost worden door een vertraging in het datasignaal te plaatsen.
- Er zijn technieken om de timing van paden te optimaliseren
 - Herontwerp kritieke pad
 - Retiming
 - Pipelining
 - Multi-cycle pad

Opgaven

- Gegeven onderstaande flipflop met enable.



timingparameters (in ns):

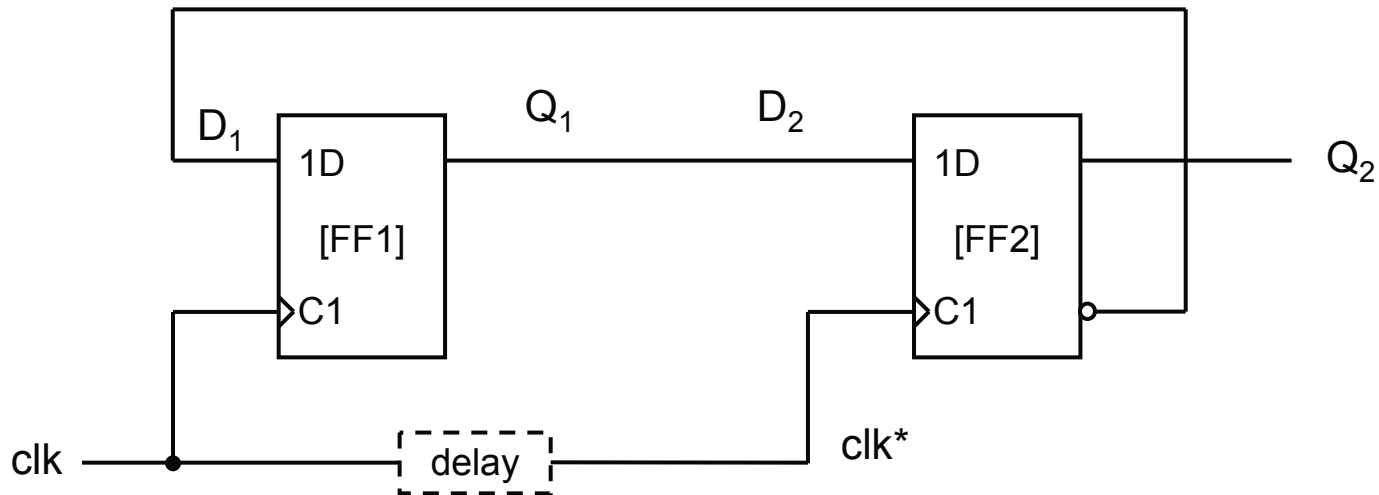
$$\text{FF: } t_{\text{su}}/t_{\text{h}}/t_{\text{P}(\text{min})}/t_{\text{P}(\text{max})} = 14/4/7/17$$

$$\text{logic: } t_{\text{p}(\text{min})}/t_{\text{P}(\text{max})} = 8/19$$

- Bepaal de setup- en holdtijd voor signaal D t.o.v. van de actieve klokflank. Bepaal de maximale frequentie waarop dit systeem betrouwbare kan werken. Teken tijddiagrammen waaruit de berekeningen kunnen worden gemaakt.

Opgaven

- Gegeven onderstaand schema. Het kloksignaal van FF2 is iets vertraagd t.o.v. de klok van FF1. Druk de maximale frequentie uit in de timing-parameters van de flipflops en de clock skew (delay).



Analyseer de logische werking van dit systeem.

Opgaven

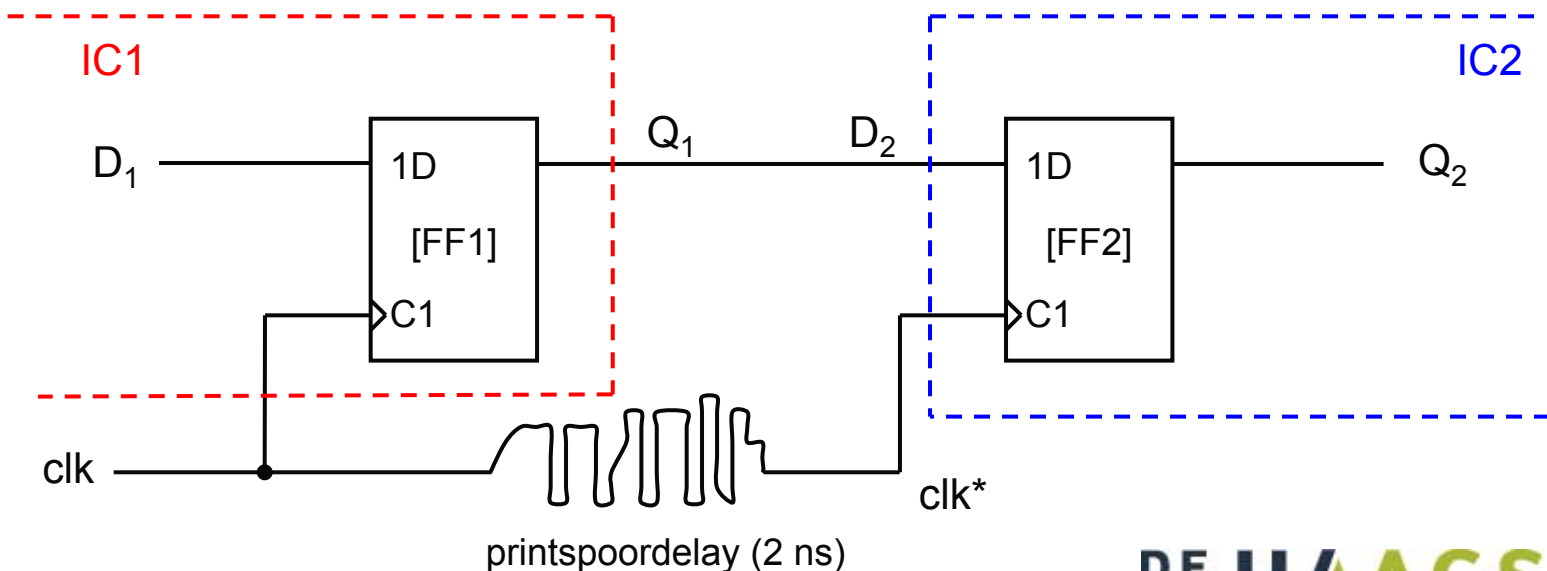
- Gegeven onderstaand schema. Het laat dataoverdracht tussen twee flipflops van verschillende IC's zien. De timing van de flipflops is verschillend. Is betrouwbare dataoverdracht mogelijk?

timingparameters (in ns):

$$\text{FF1: } t_{\text{su}}/t_{\text{h}}/t_{\text{P}(\text{min})}/t_{\text{P}(\text{max})} = 10/3/7/12$$

timingparameters (in ns):

$$\text{FF2: } t_{\text{su}}/t_{\text{h}}/t_{\text{P}(\text{min})}/t_{\text{P}(\text{max})} = 20/6/10/25$$





Academie voor Technology, Innovation &
Society Delft
Academie voor ICT & Media

De Haagse Hogeschool, Delft
+31-15-2606311
J.E.J.opdenBrouw@hhs.nl
www.dehaagsehogeschool.nl

DE HAAGSE
HOGESCHOOL