



DIGTEC/2021-2022

Jesse op den Brouw

DIGTEC

Timing bij dataoverdracht

DE HAAGSE
HOGESCHOOL

Timing bij dataoverdracht

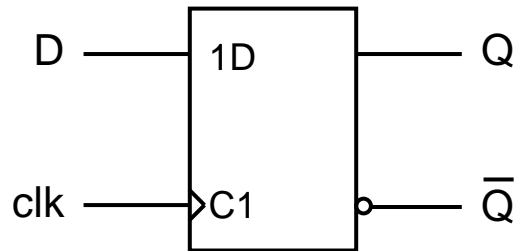
- We hebben nu de logische werking en de elektrische werking besproken.
- We gaan nu de timing bij dataoverdracht tussen flipflops bespreken.
- Er zijn twee varianten: directe en indirecte dataoverdracht.
- We bespreken ook klokskew, het vertragen van het kloksignaal.
- We laten zien hoe de maximale systeemfrequentie berekend kan worden.

Timing bij dataoverdracht

- 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.

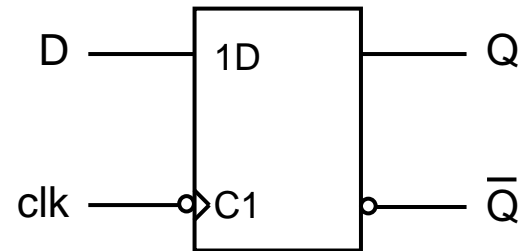
Timing bij dataoverdracht

- 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

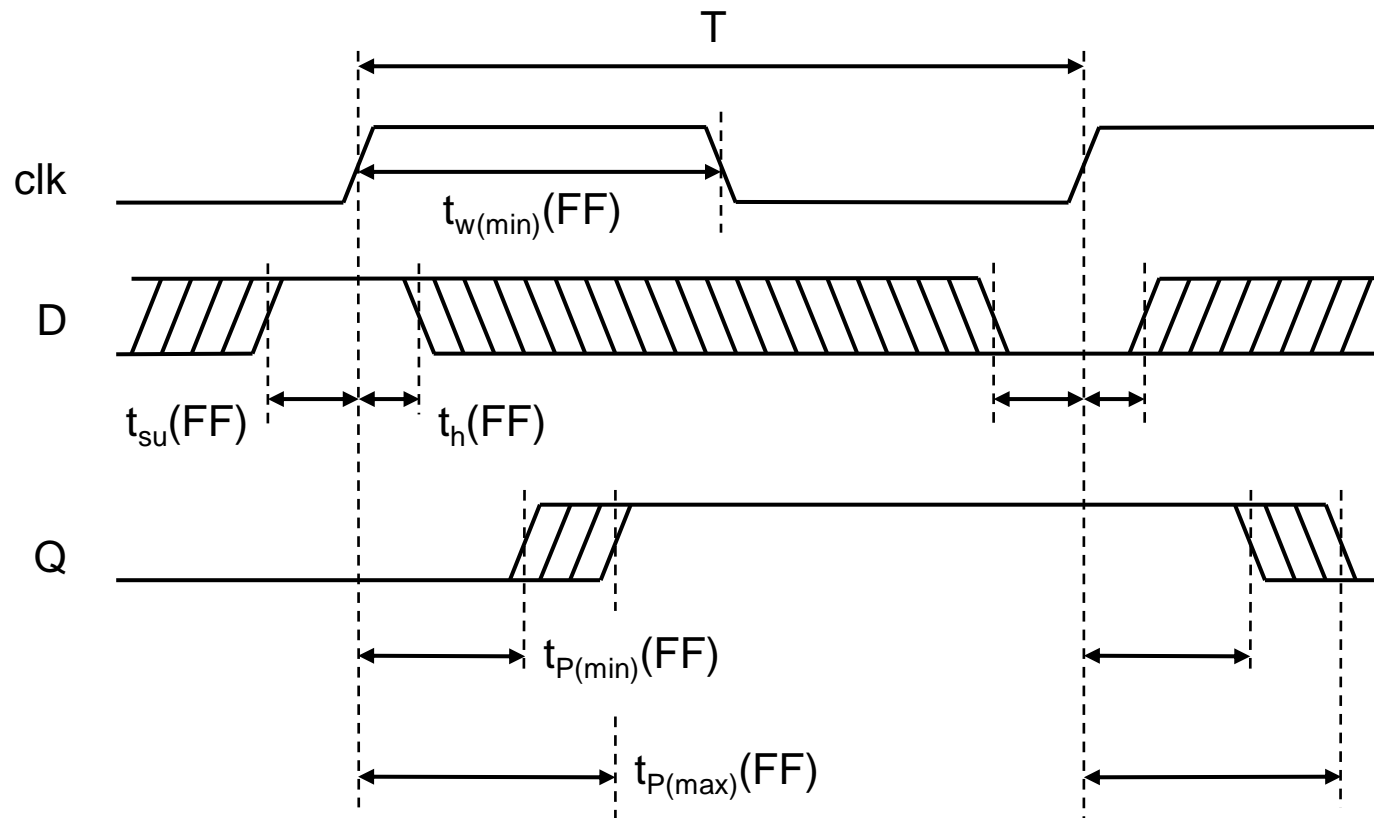


negative edge-triggered
D-flipflop

werking D afhankelijk
van de neergaande flank
van C

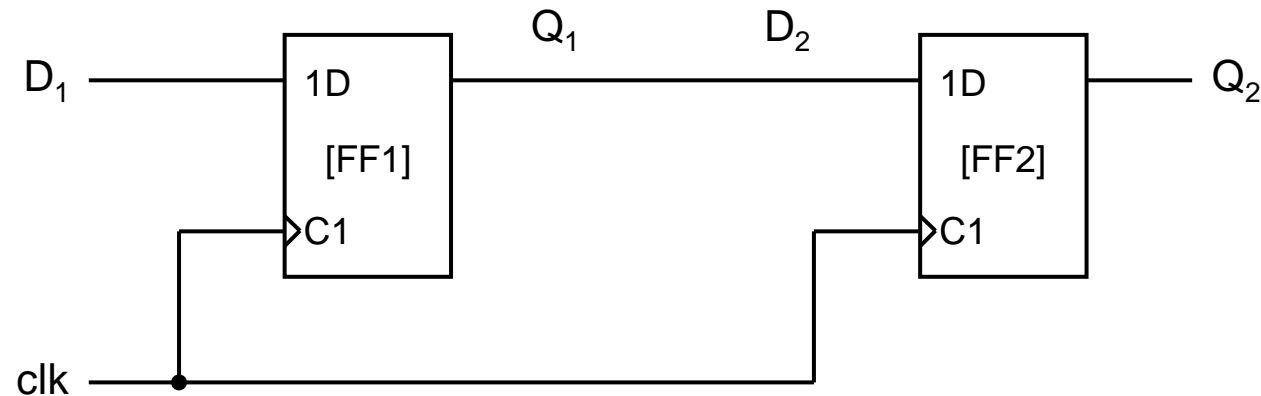
Timing bij dataoverdracht

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



Timing bij 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 timingeigenschappen (bv. op hetzelfde IC).
- Directe overdracht, dus $D_2 = Q_1$.

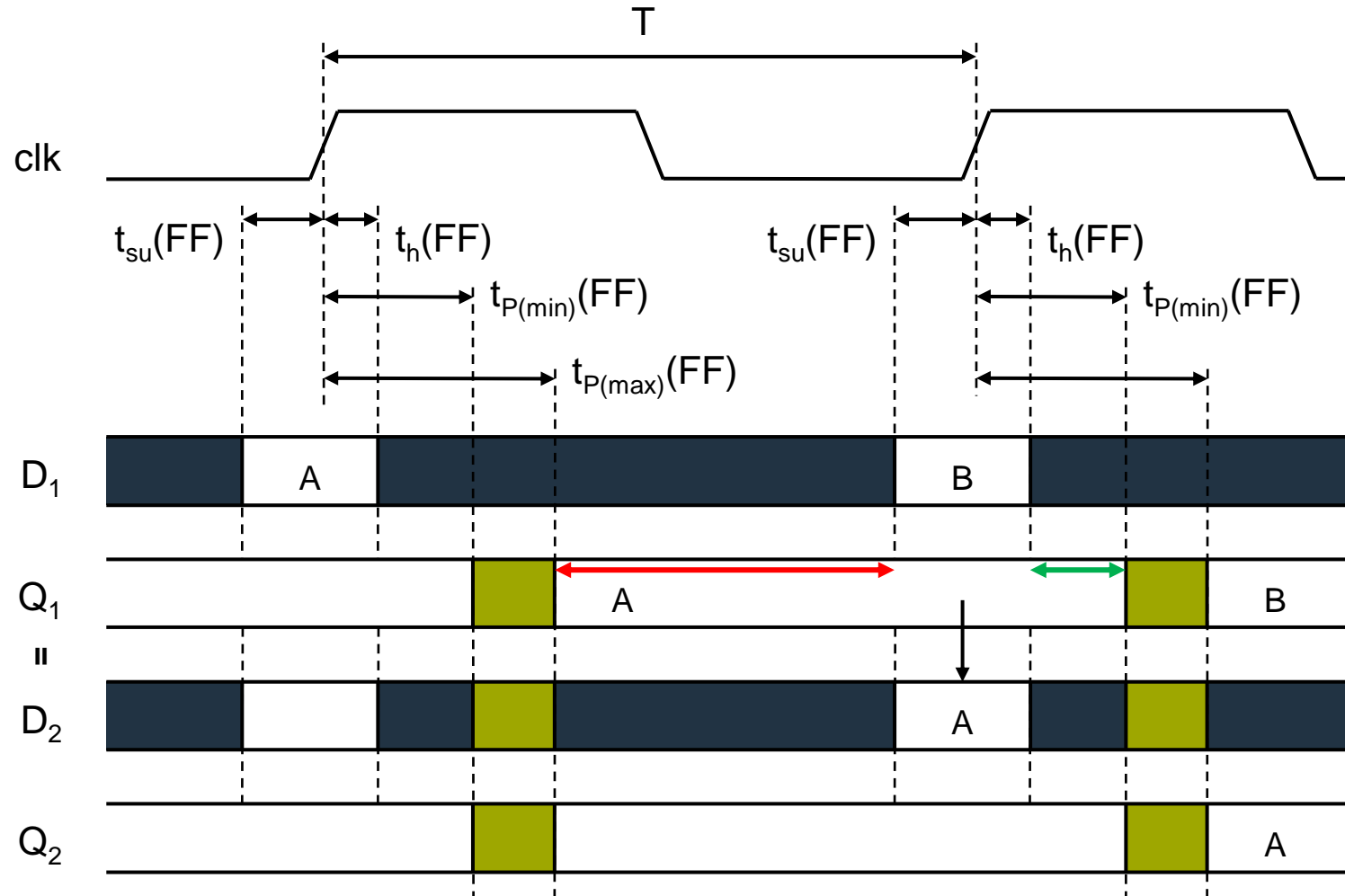
Timing bij dataoverdracht

- 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 $t_{P(min)}(FF) - t_{P(max)}(FF)$.

Timing bij 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 $t_{P(min)}(FF) - t_{P(max)}(FF)$.
- Het geheel kan worden getekend in een timingdiagram.

Timing bij dataoverdracht

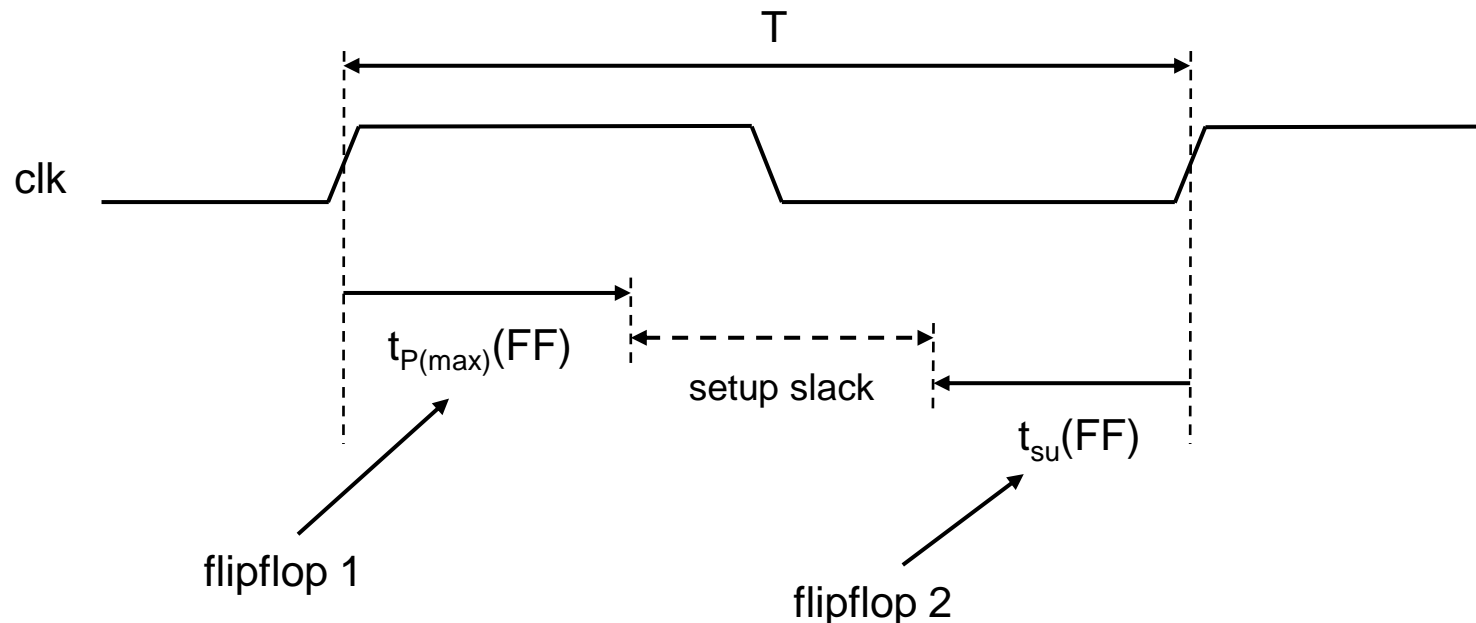


Timing bij dataoverdracht

- 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.

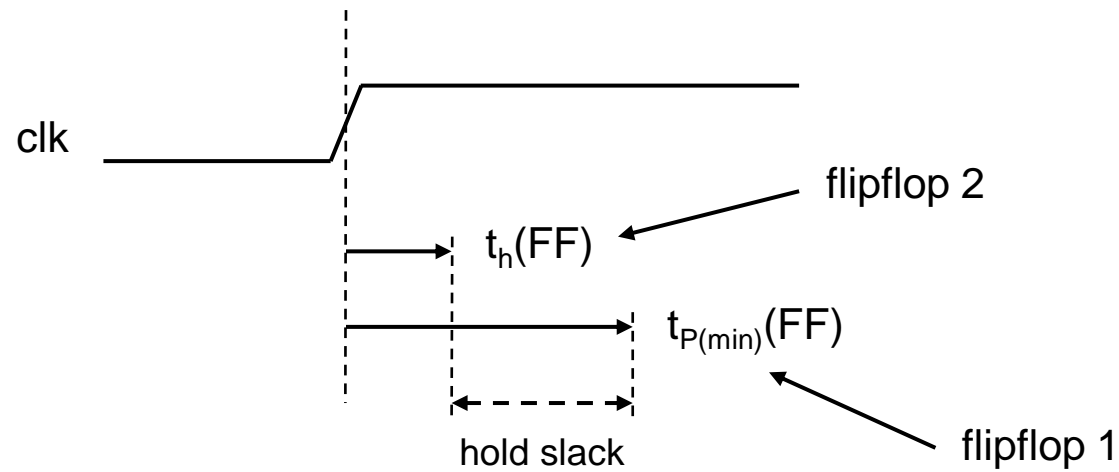
Timing bij dataoverdracht

- 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.



Timing bij 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.



Timing bij dataoverdracht

- 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

Timing bij 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.

Timing bij 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:

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

Timing bij 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:

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

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

Timing bij dataoverdracht

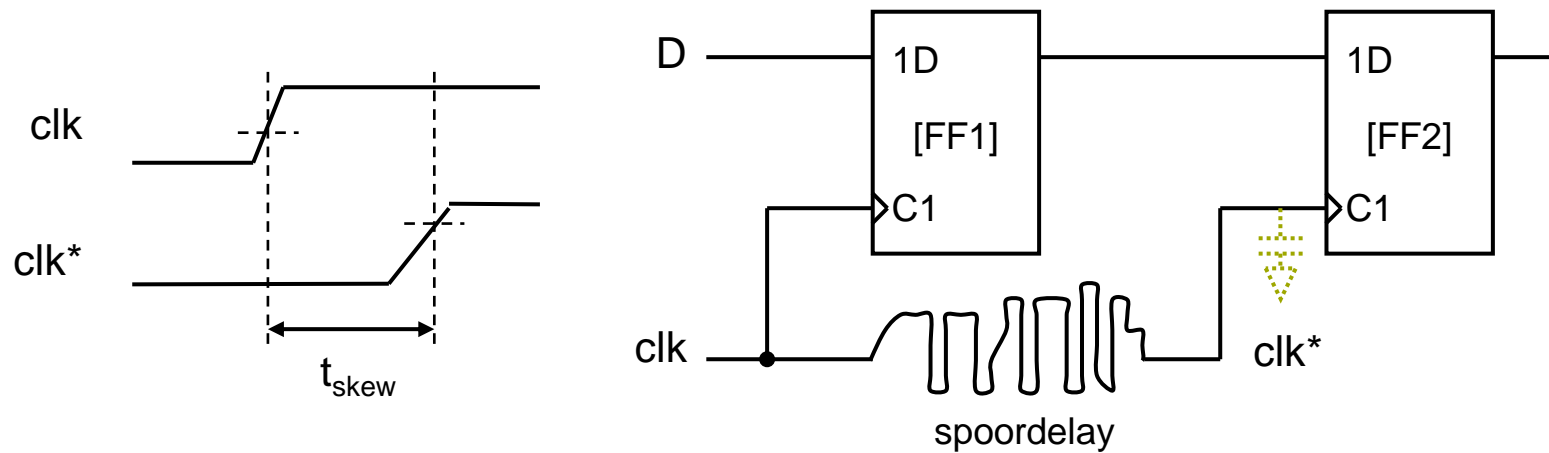
- Er zijn nog aanvullende eisen met betrekking op de vorm van het kloksignaal en de interne werking van de flipflop.
- Het kloksignaal moet *zonder vertraging* worden aangeboden aan beide flipflops.
- De klokflanken die aan de flipflops worden aangeboden moeten *identieke vorm* hebben (niet vervormd).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

Timing bij 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.

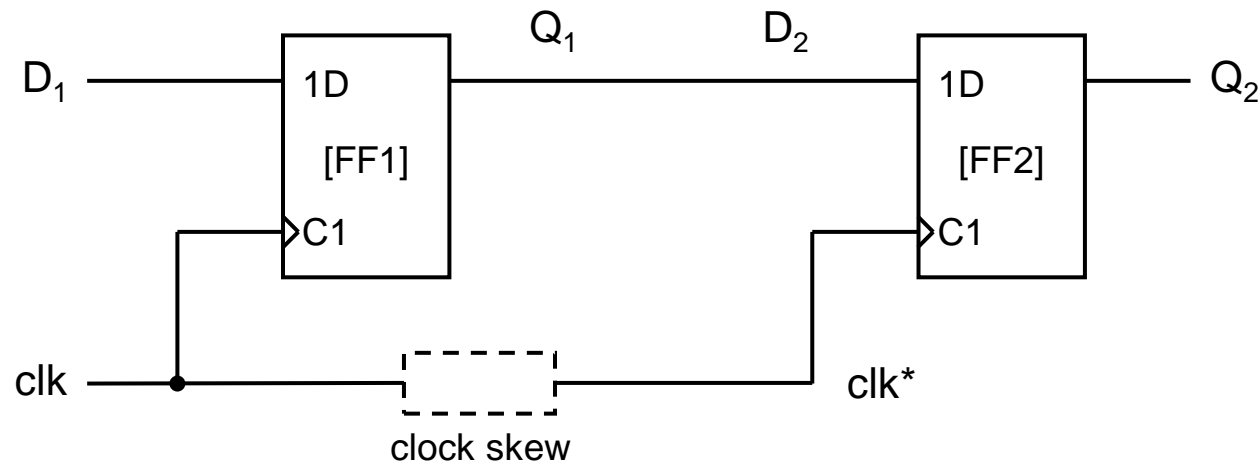
Timing bij dataoverdracht

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



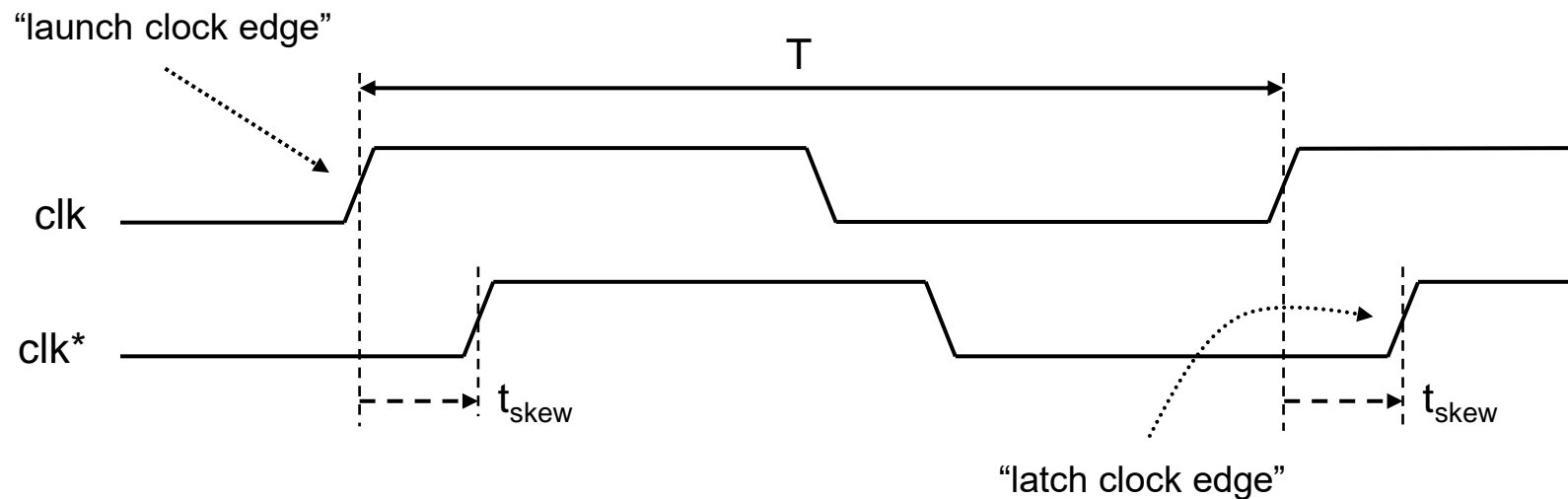
Timing bij dataoverdracht

- 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.



Timing bij dataoverdracht

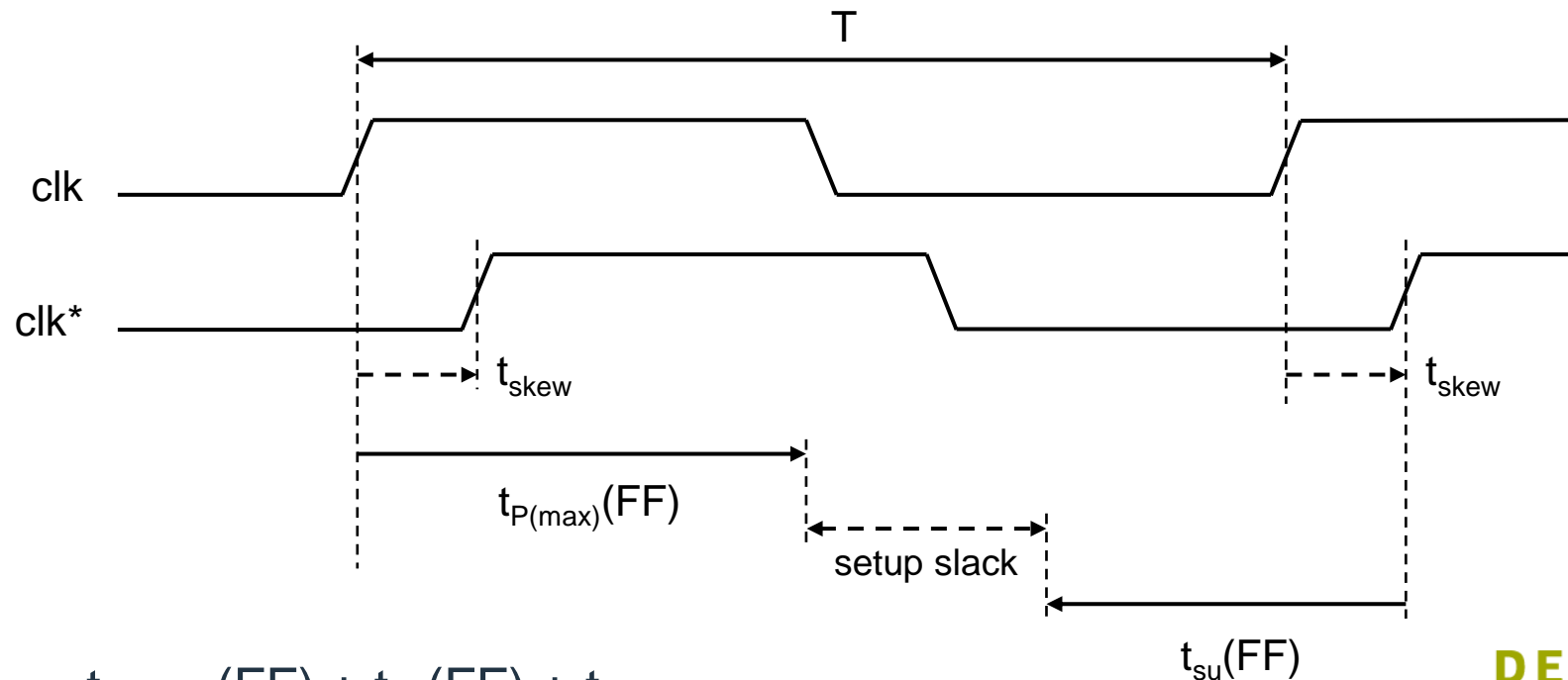
- Om de twee klokflanken uit elkaar te houden wordt gesproken van een *launch clock edge*[†] en een *latch clock edge*[†]. Onderstaande tekening verduidelijkt de twee begrippen.



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

Timing bij dataoverdracht

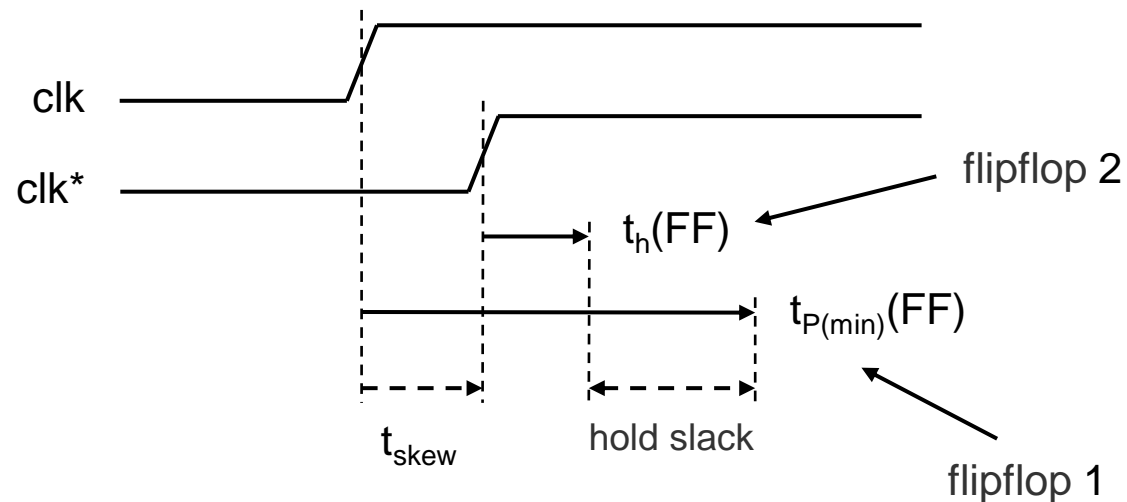
- We tekenen een timingdiagram en leiden de formule voor T af. Clock skew heeft (in dit geval) *positieve* invloed op de *setup slack*.



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

Timing bij dataoverdracht

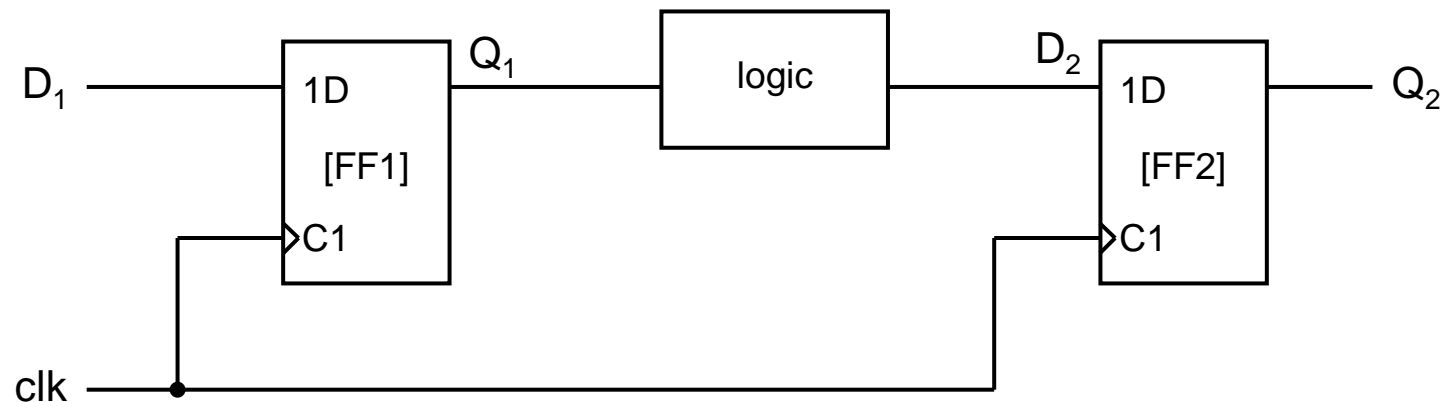
- Clock skew heeft (in dit geval) *negatieve* invloed op de hold slack. De uitgang van de eerste flipflop mag pas veranderen na de holdtijd van de tweede flipflop. Deze holdtijd wordt “verlengd” door de clock skew.



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

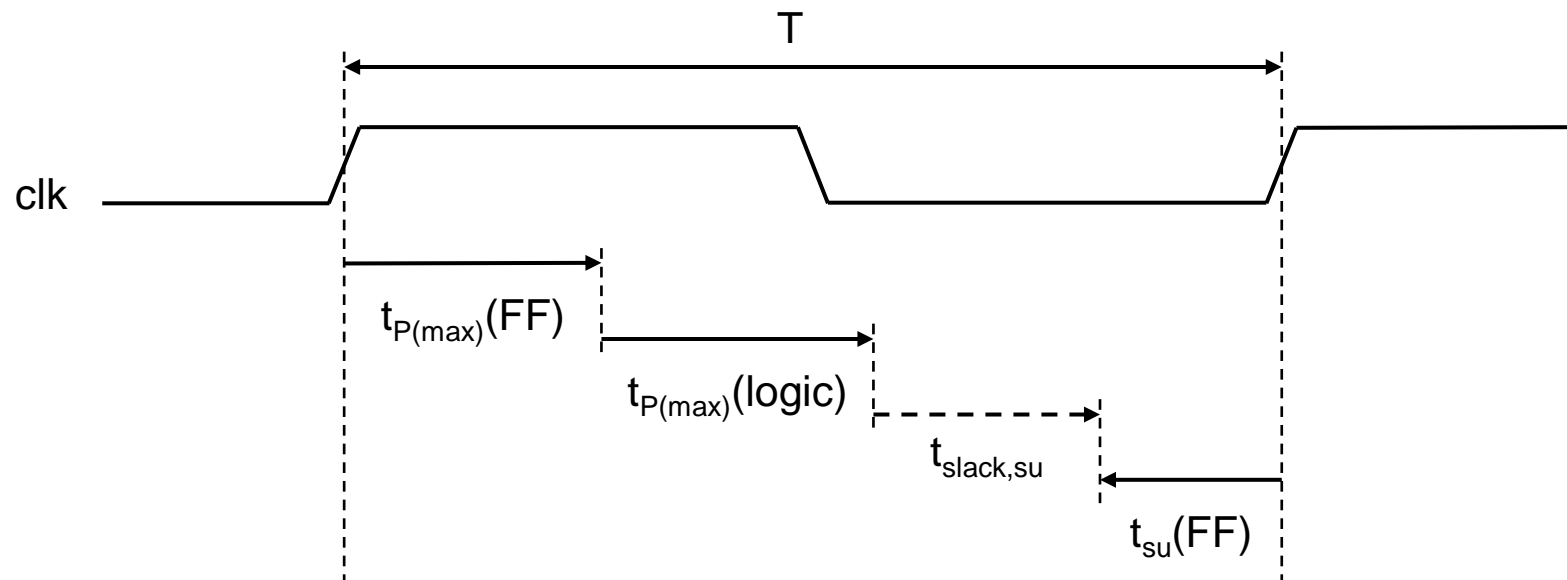
Timing bij dataoverdracht

- 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.



Timing bij dataoverdracht

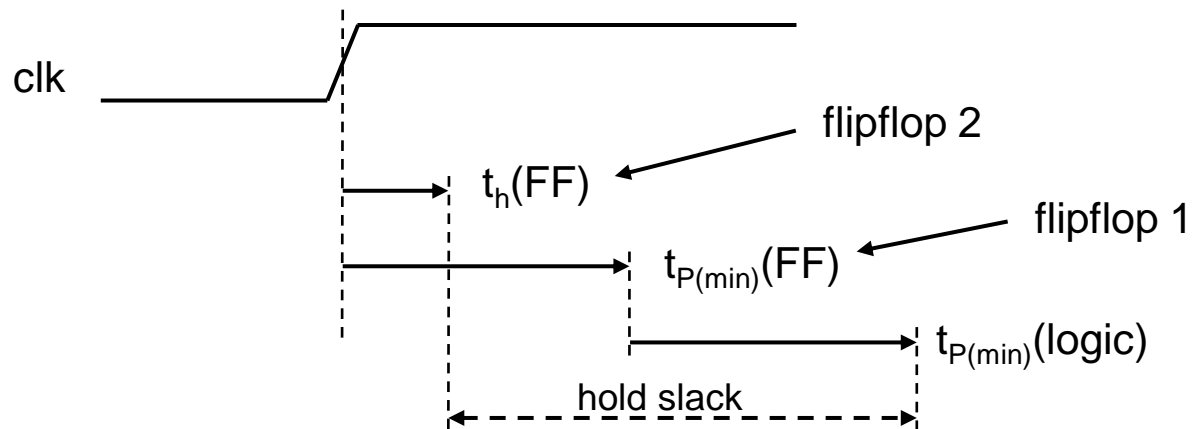
- Voor het berekenen van de periodeduur T moet nu de maximale propagatietijd van de logica $t_{P(\max)}(\text{logic})$ meegenomen worden.



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

Timing bij dataoverdracht

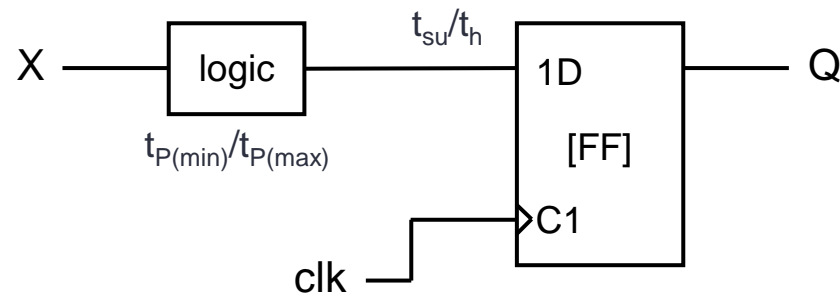
- De waarde van de uitgang van de eerste flipflop verandert op z'n vroegst na de minimale vertragingstijd van de flipflop. Daarna gaat het signaal door de logica. Op z'n vroegst verandert de uitgang van de logica na de minimale vertragingstijd van de logica. Deze twee tijden opgeteld moet groter zijn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



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

Timing bij dataoverdracht

- 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.

Timing bij dataoverdracht

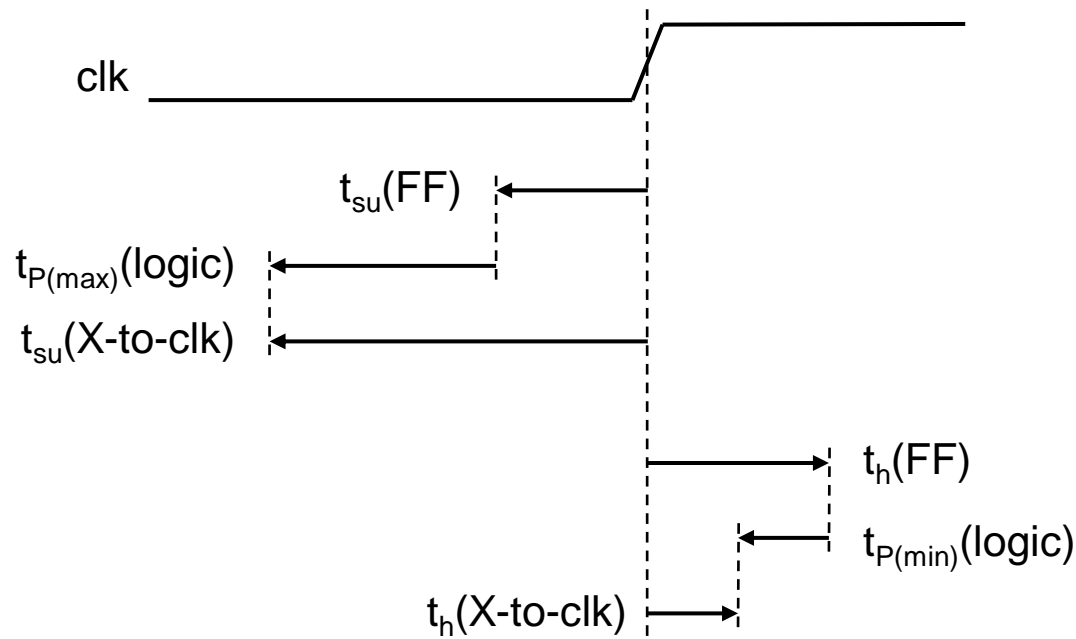
- Voorwaarde is natuurlijk dat de setup- en holdtijden van de D-ingang gerespecteerd worden.
- Een verandering op signaal X heeft maximaal $t_{P(\max)}(\text{logic})$ nodig.
- De laatst mogelijke verandering op X moet dus $t_{P(\max)}(\text{logic})$ vóór $t_{\text{su}}(\text{FF})$ gebeuren.
- Dus $t_{\text{su}}(\text{X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

Timing bij dataoverdracht

- 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 $t_h(\text{X - to - clk}) = t_h(\text{FF}) - t_{P(\text{min})}(\text{logic})$
- Let op: min-teken in formule!

Timing bij dataoverdracht

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

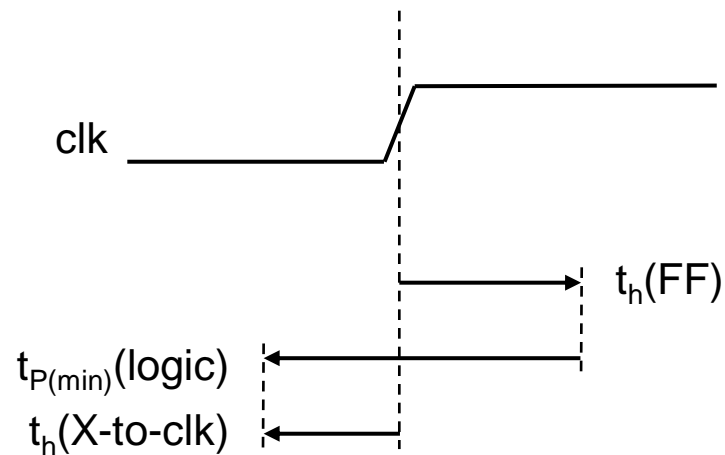


$$t_{su}(X - to - clk) = t_{su}(FF) + t_{P(max)}(logic)$$

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

Timing bij dataoverdracht

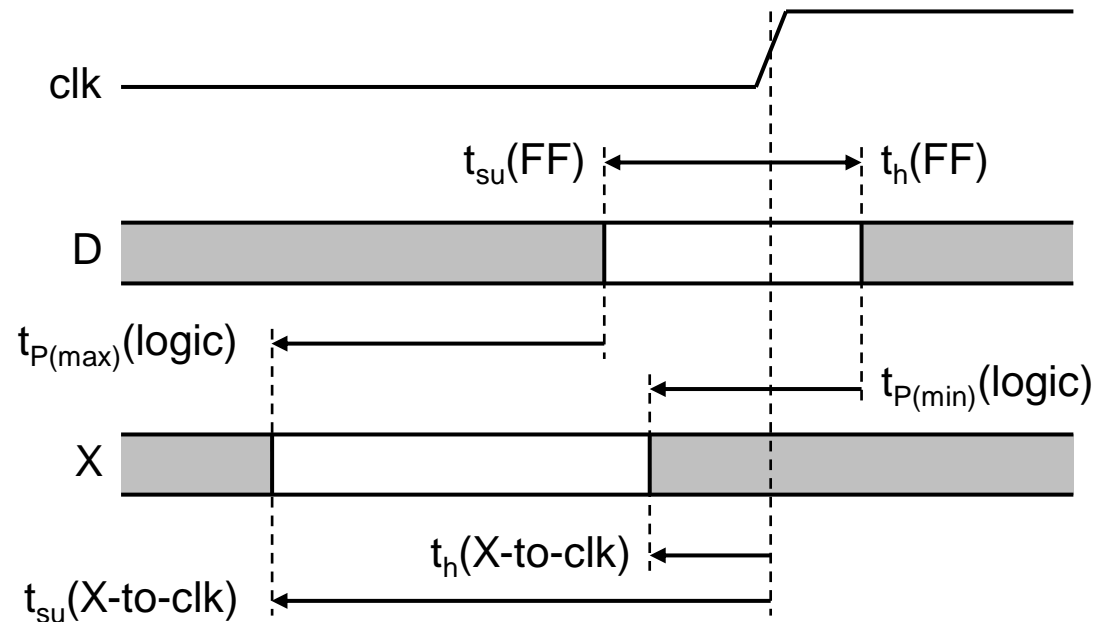
- Een negatief antwoord voor de holdtijd $t_h(X\text{-to-clk})$ betekent dus dat het signaal nog voor de klokflank mag veranderen.



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

Timing bij dataoverdracht

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



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

Timing bij dataoverdracht

- 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:

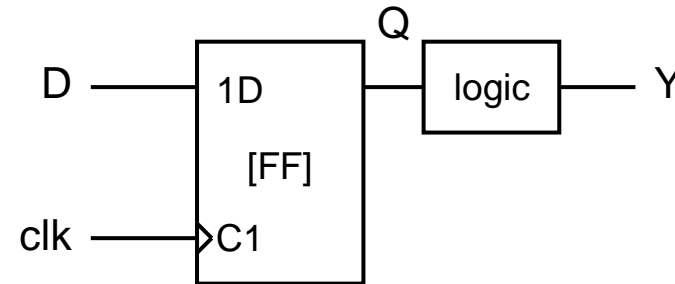
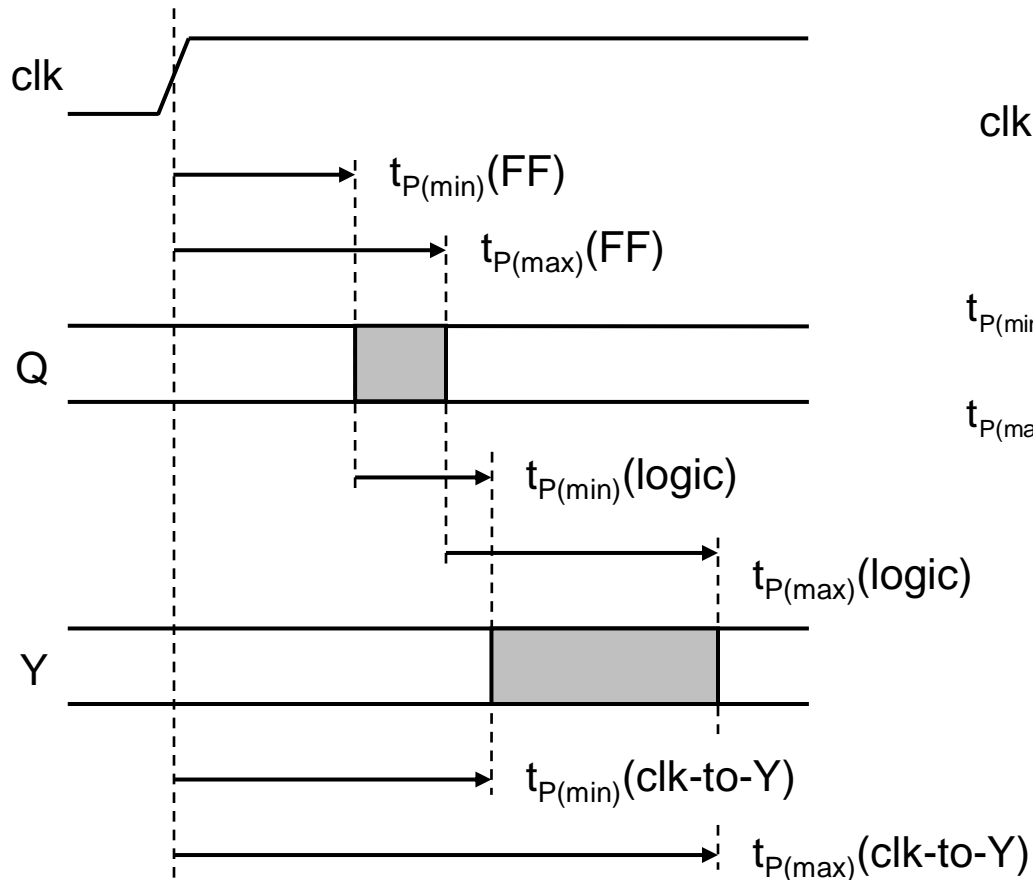
$$t_{P(\min)}(\text{out}) = t_{P(\min)}(\text{FF}) + t_{P(\min)}(\text{logic})$$

- 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 bij dataoverdracht

- Hieronder het timingdiagram.

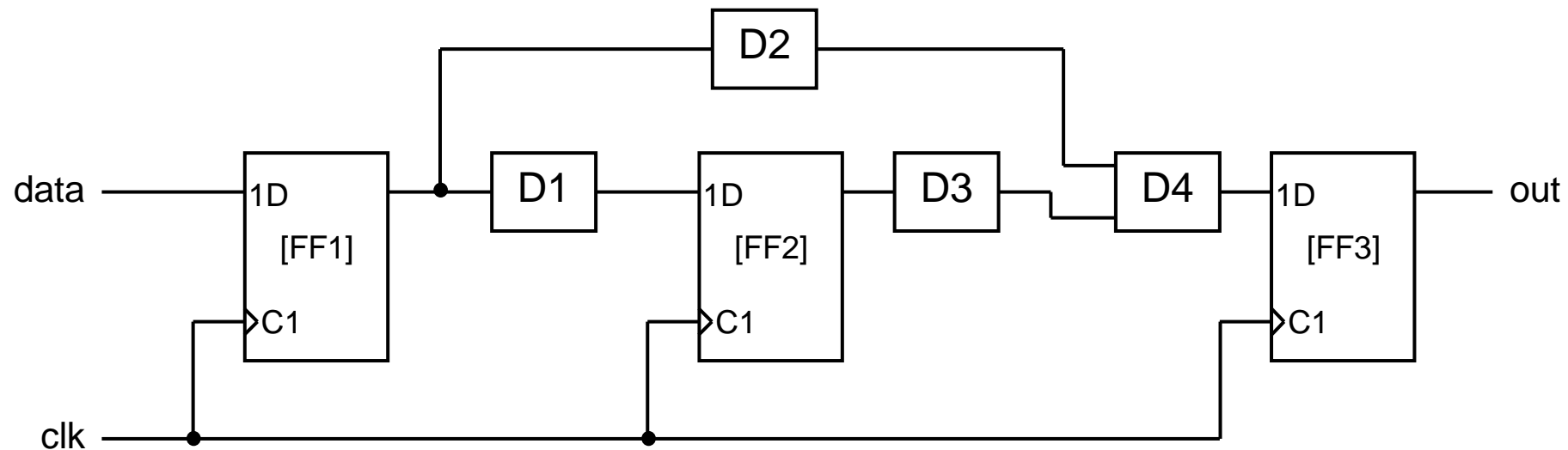


$$t_{P(\min)}(\text{clk} - \text{to} - \text{Y}) = t_{P(\min)}(\text{FF}) + t_{P(\min)}(\text{logic})$$

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

Timing bij dataoverdracht

- 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.



Timing bij dataoverdracht

- 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.

Timing bij dataoverdracht

- 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:

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

Timing bij 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

let's change