



DIGTEC/2021-2022

Jesse op den Brouw

DIGTEC

Tellers

DE HAAGSE
HOGESCHOOL

Tellers

- Tellers worden in digitale systemen voor veel doeleinden gebruikt.
- Een teller kan bijvoorbeeld bepaalde gebeurtenissen tellen.
- Een teller kan ook tijd 'tellen' en tijdintervallen genereren. Zo'n teller wordt een *timer* genoemd.
- Tellers hebben de eigenschap een getelde hoeveelheid te *onthouden*. Dat betekent dat tellers *geheugen* bezitten.

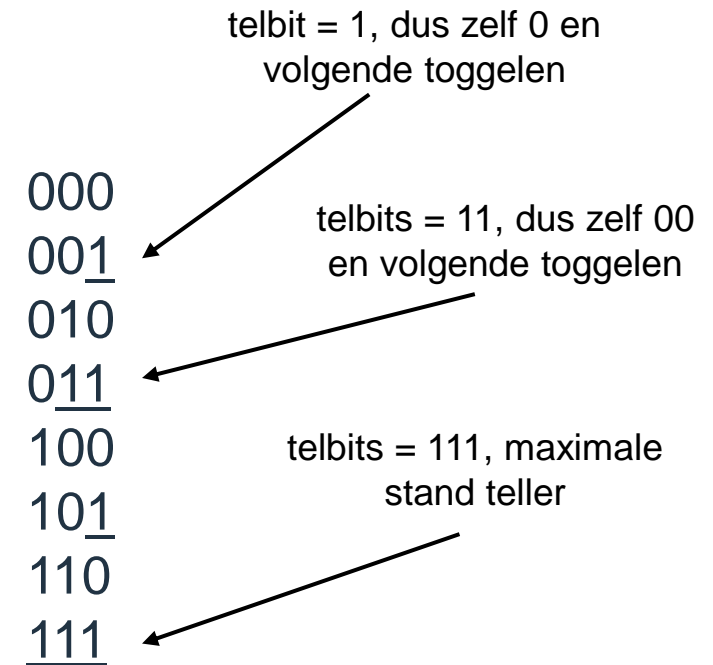
Tellers

- Specifieke eigenschappen van tellers:
 - Omhoog tellen (optellen, up counter)
 - Omlaag tellen (aftellen, down counter)
 - Omhoog/omlaag tellen (optellen/aftellen, up-down counter)
 - Laden (load)
 - Tellen actief (enable)
 - Wissen (clear, zowel asynchroon als synchroon)
 - Telbereik/telcodering (binair, BCD, Gray, One-hot, Johnson, ...)
- Tellers moeten kloksynchroon te worden ontworpen.

Tellers

- Een 1-bit teller, 2-bit teller, 3-teller

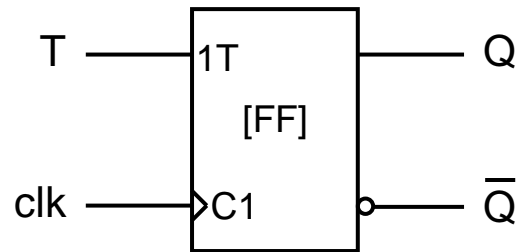
| | |
|----------|-----------|
| 0 | 00 |
| <u>1</u> | <u>01</u> |
| 0 | 10 |
| <u>1</u> | <u>11</u> |
| 0 | 00 |
| <u>1</u> | <u>01</u> |
| 0 | 10 |
| <u>1</u> | <u>11</u> |



- Indien telbits stand 1, 11 of 111 hebben bereikt moeten deze zelf 0, 00 of 000 worden en het volgende telbit moet toggelen.

Tellers

- Het toggelen kan worden gedaan door een T-flipflop.



$$Q^{n+1} = \begin{cases} Q^n & \text{als } T^n = 0 \\ \overline{Q^n} & \text{als } T^n = 1 \end{cases}$$

$$\begin{aligned} Q^{n+1} &= T^n \cdot \overline{Q^n} + \overline{T^n} \cdot Q^n \\ &= T^n \oplus Q^n \end{aligned}$$

Tellers

- Om een telbit te laten toggelen moeten alle voorgaande telbits 1 zijn:

$$T_0 = 1$$

$$T_1 = Q_0$$

$$T_2 = Q_1 \cdot Q_0$$

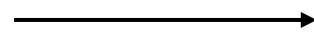
$$T_3 = Q_2 \cdot Q_1 \cdot Q_0$$

...

$$T_n = Q_{n-1} \cdot Q_{n-2} \cdots Q_1 \cdot Q_0$$



AND met veel ingangen



herschrijven

$$T_0 = 1$$

$$T_1 = Q_0$$

$$T_2 = Q_1 \cdot Q_0$$

$$T_3 = Q_2 \cdot Q_1 \cdot Q_0 = Q_2 \cdot (Q_1 \cdot Q_0)$$

...

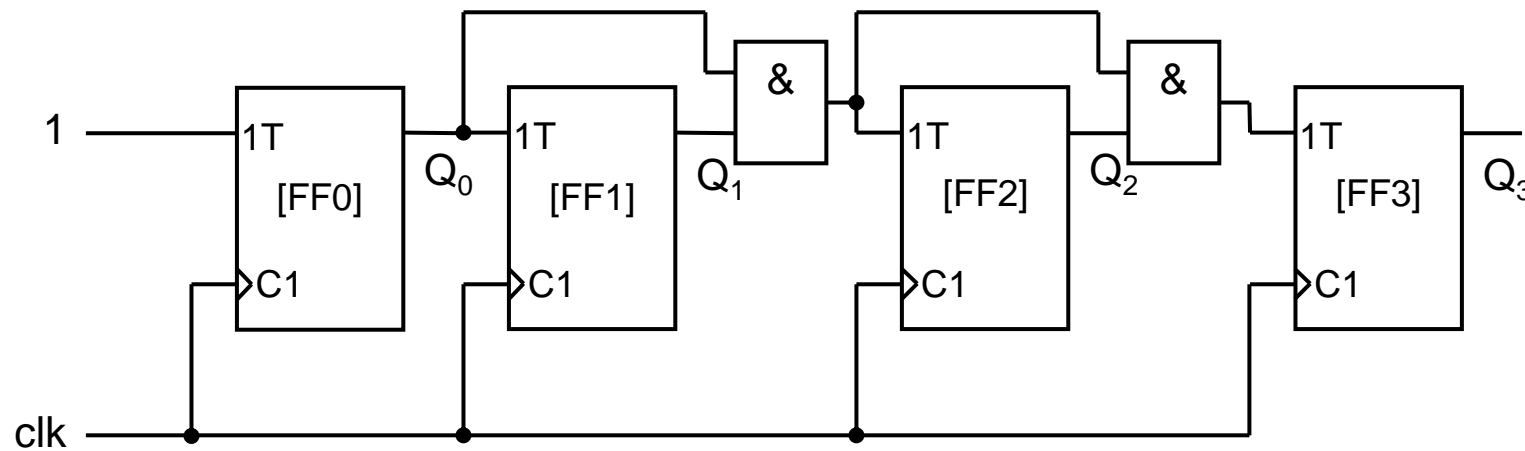
$$T_n = Q_{n-1} \cdot (Q_{n-2} \cdots (Q_1 \cdot Q_0))$$



cascadeschakeling van
2-input ANDs

Tellers

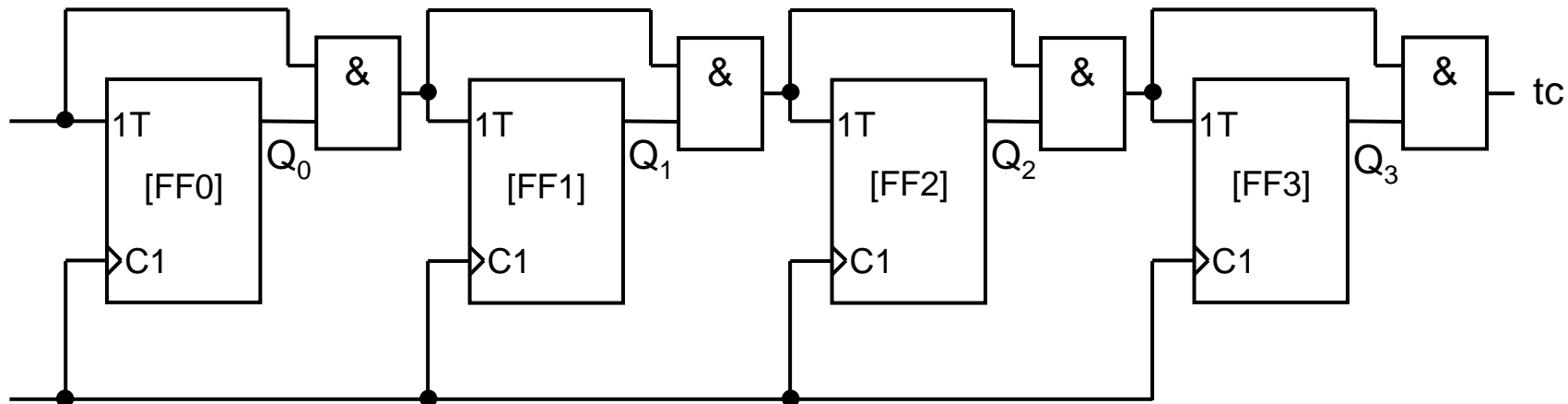
- Hieronder een 4-bit teller op basis van T-flipflops en cascadeschakeling van ANDs.



- Nadeel: maximale frequentie wordt lager naar mate er meer telbits zijn.

Tellers

- We breiden de teller uit met een enable (en) en een terminal count (tc)
 - Indien en = 1 dan telt de teller op een klokflank
 - Indien de teller in de hoogste stand en enable actief, dan tc = 1 anders 0

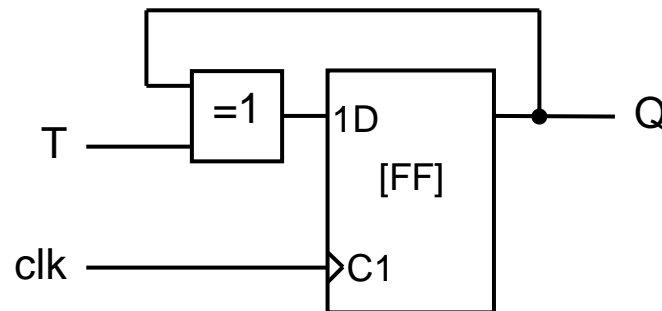


- De teller is nu te cascaderen tot een grotere teller

Tellers

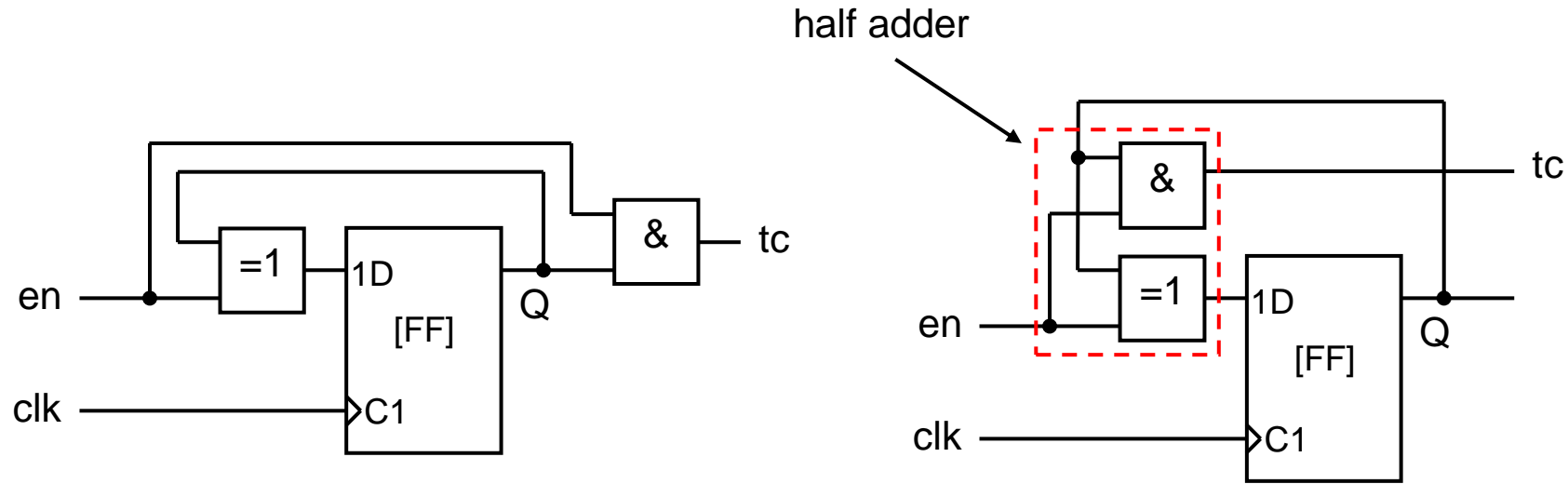
- Een nadeel van een T-flipflop is dat deze niet makkelijk synchroon te clearen is en ook niet makkelijk te laden met een '0' of '1'.
- Een D-flipflop kan dit wel (er is wel voorzetlogica nodig).
- Een T-flipflop kan eenvoudig worden ontworpen uit een D-flipflop:

$$\left. \begin{array}{l} Q^{n+1} = T^n \oplus Q^n \\ Q^{n+1} = D^n \end{array} \right\} D^n = T^n \oplus Q^n$$



Tellers

- Een telsectie kan eenvoudig worden ontworpen uit een D-flipflop, een EXOR en een AND.

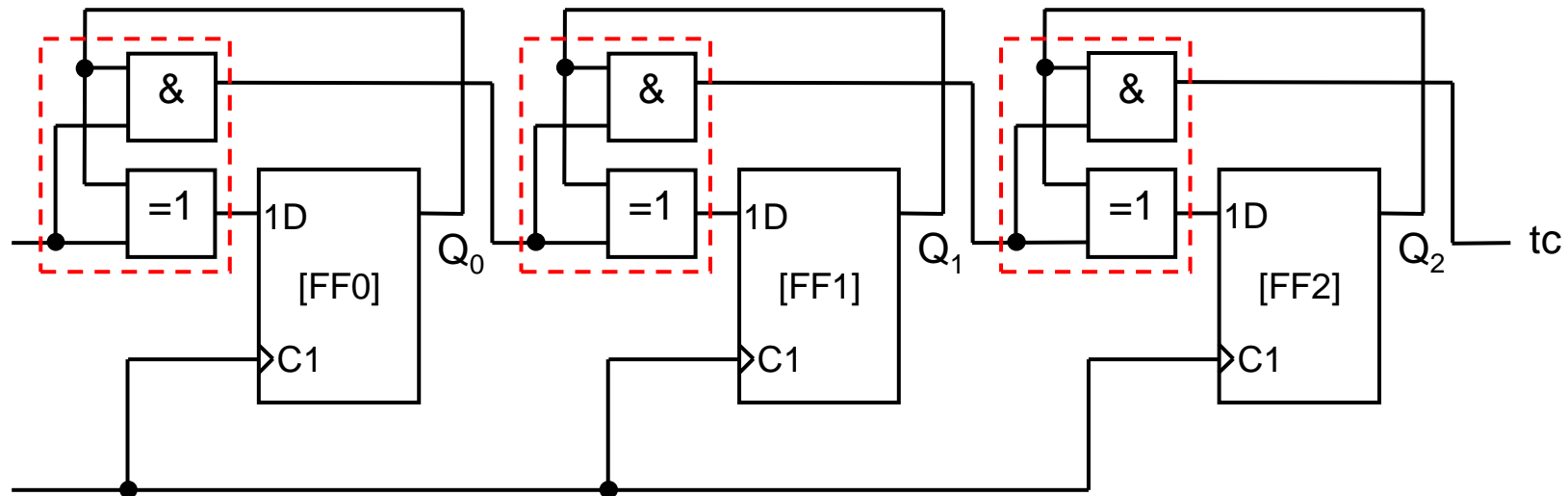


$$Q^{n+1} = en^n \oplus Q^n$$

$$tc^n = en^n \cdot Q^n$$

Tellers

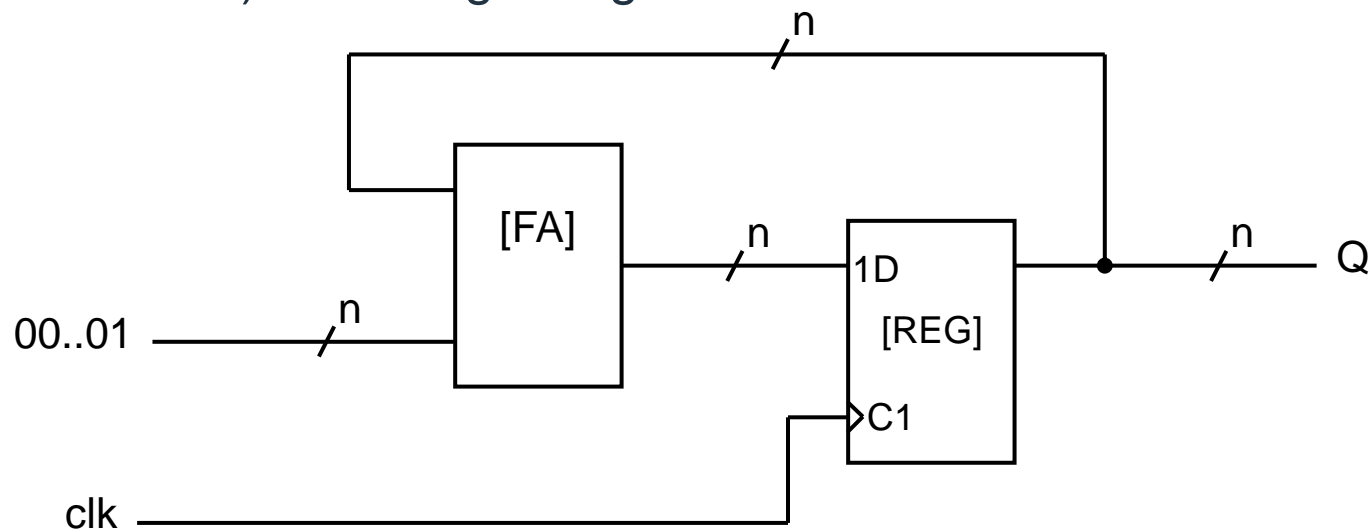
- Hieronder een voorbeeld van een 3-bit teller.



- Een teller is dus te maken op basis van D-flipflops en half adders.

Tellers

- In feite is een teller te realiseren met een *full adder* en een *register*. De huidige stand van het register (de tellerwaarde) wordt teruggevoerd naar de opteller. Op de andere ingang van de teller wordt de constante 00..01 aangeboden. De nieuwe berekende waarde wordt (op een klokflank) in het register geklokt.

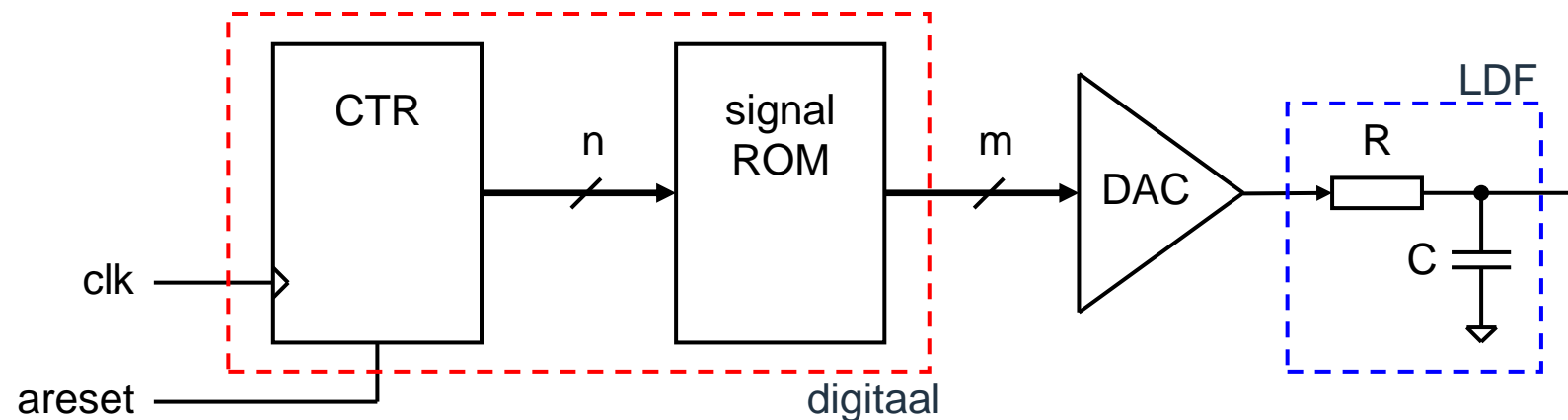


Tellers

- Een mooi voorbeeld waar een teller wordt gebruikt is een digitale signaalvorm generator.
- Het te genereren signaal is sinusvormig (als voorbeeld).
- De teller telt cyclisch (begint weer bij 0 als hoogste telstand is bereikt).
- De teller stuurt een ROM-tabel aan. De tabel bevat een gecodeerde sinus. Een ROM is in feite een elektronische waarheidstabel.
- De ROM-tabel stuurt een digitaal-analoog converter aan. Deze genereert het analoge signaal.
- Aan de uitgang van de digitaal-analoog converter is een laagdoorlaat-filter (LDF) gekoppeld die het signaal reconstrueert.

Tellers

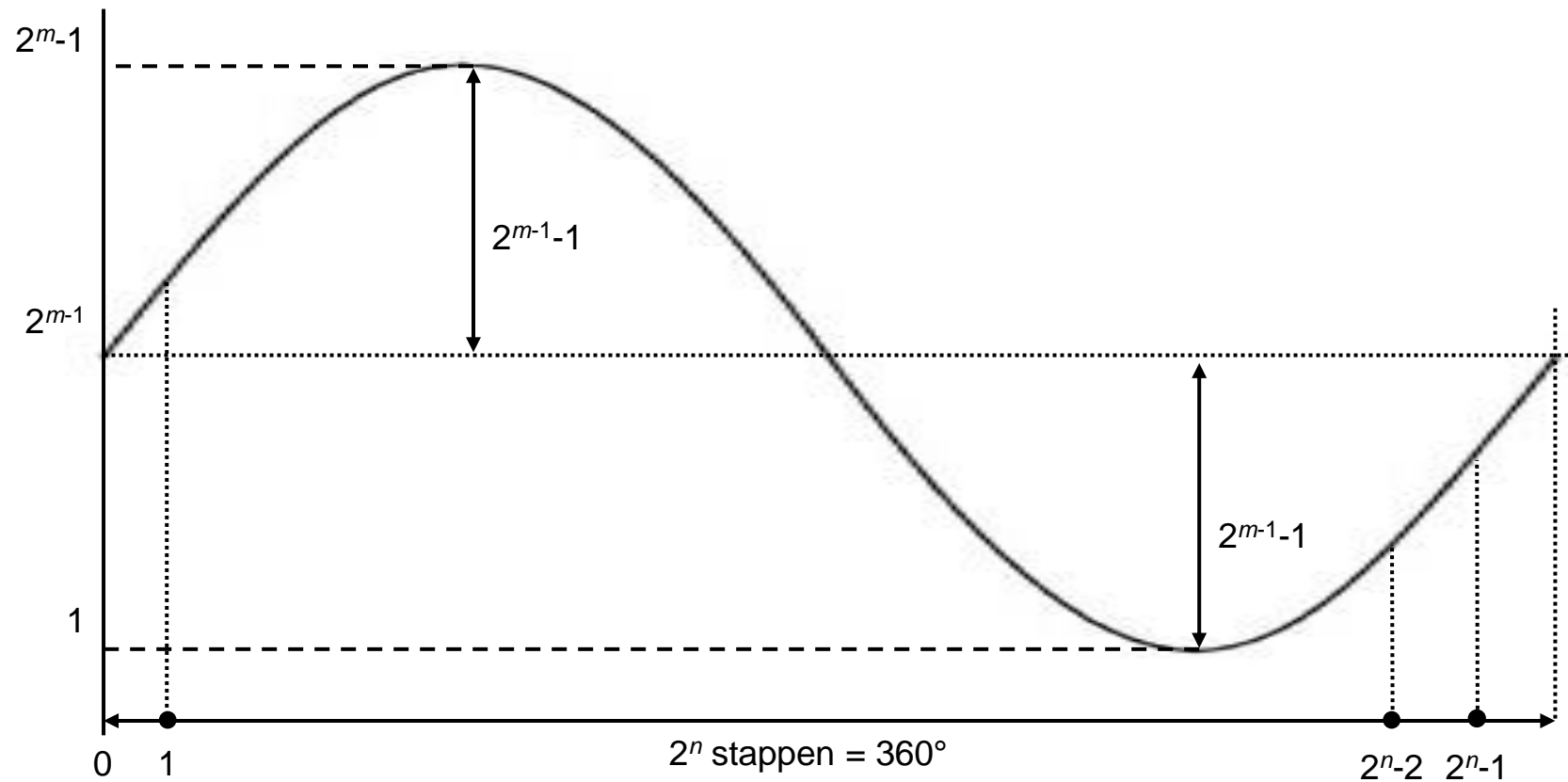
- Hieronder is het blokschema gegeven.



- De teller loopt van 0 t/m $2^n - 1$. De signal ROM heeft *n* adresbits en *m* databits. Op een adres kan één van 2^m verschillende waarden geplaatst worden.

Tellers

- Omzetten van analoge sinus naar digitale representatie.



Tellers

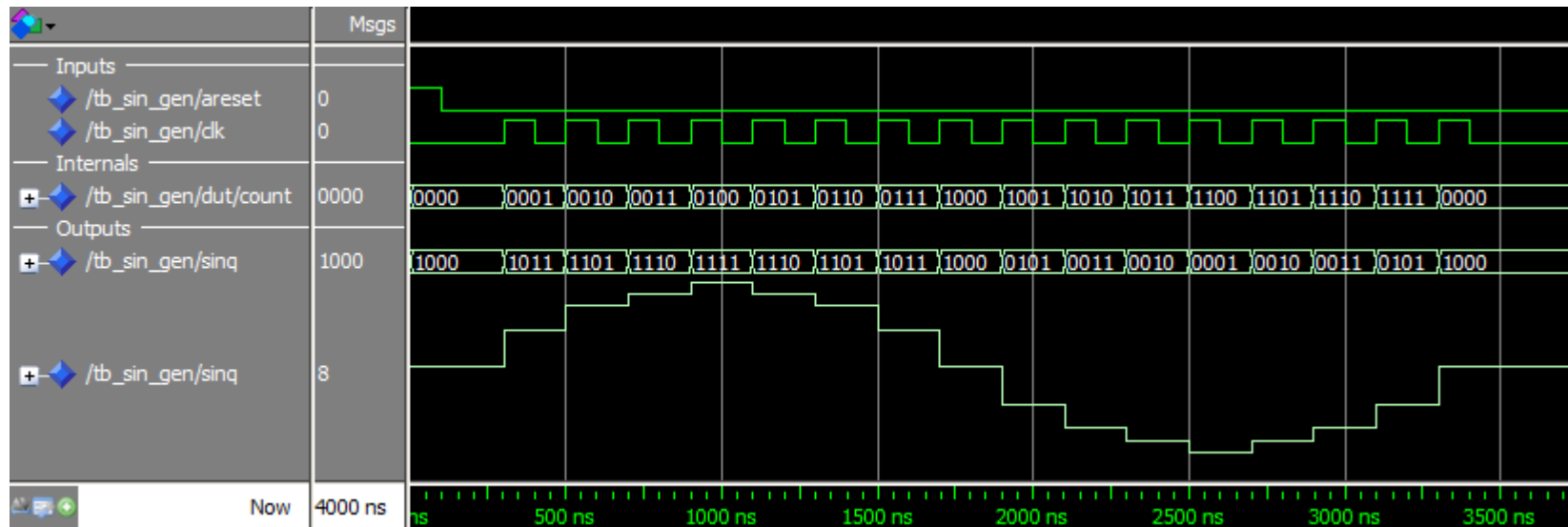
- Het uitrekenen van de tabel gaat als volgt

$$\text{tabelwaarde}_i = \sin\left(i \cdot \frac{360^\circ}{2^n}\right) \cdot (2^{m-1} - 1) + 2^{m-1} \quad i = 0 \dots 2^n - 1$$

- In één rondgang langs een cirkel worden 2^n waarden berekend.
- Elke sinuswaarde wordt geschaald naar helft-1 van het bereik $(2^{m-1}-1)$ met een offset van de helft van het bereik (2^{m-1}) .
- De minimale waarde is 1 en de maximale waarde is 2^{m-1} .

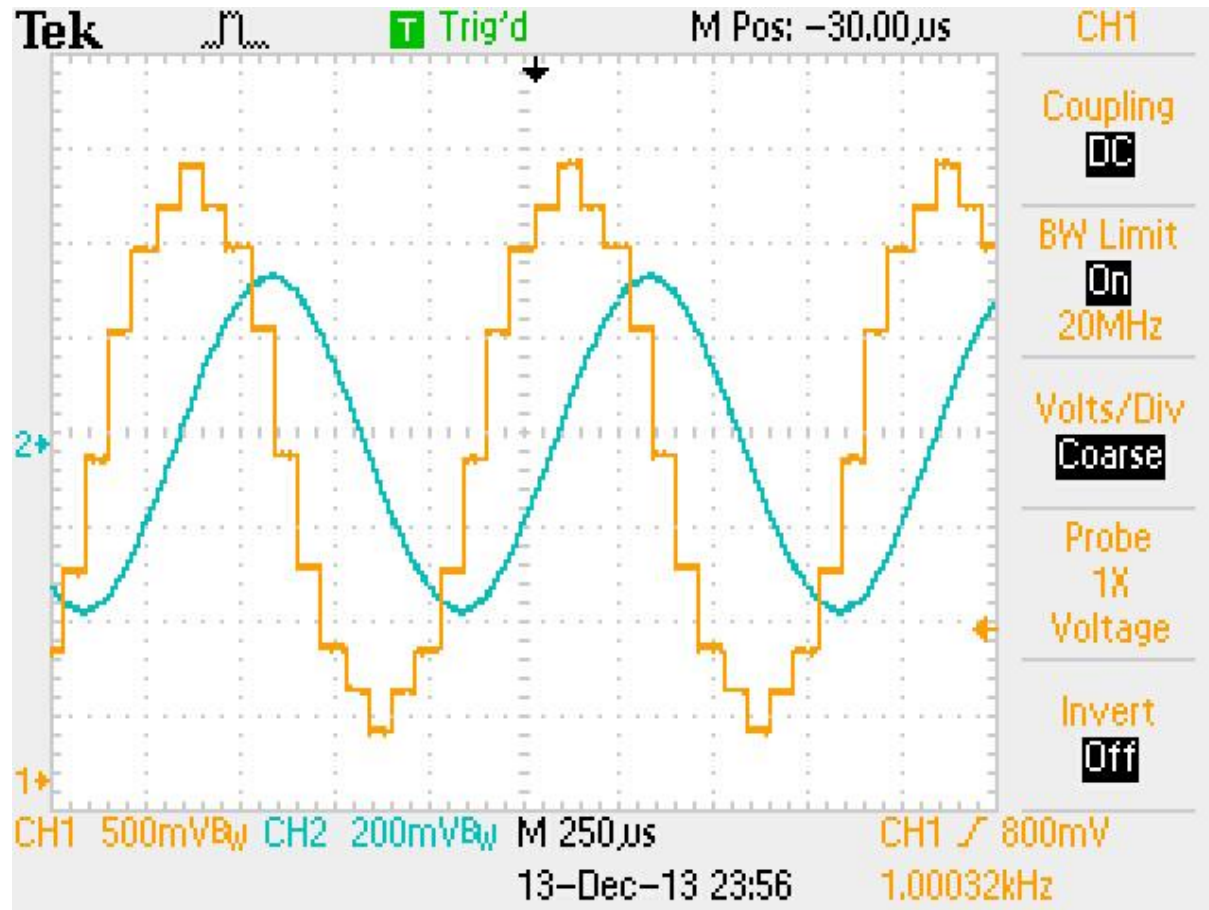
Tellers

- Hieronder het simulatieresultaat.



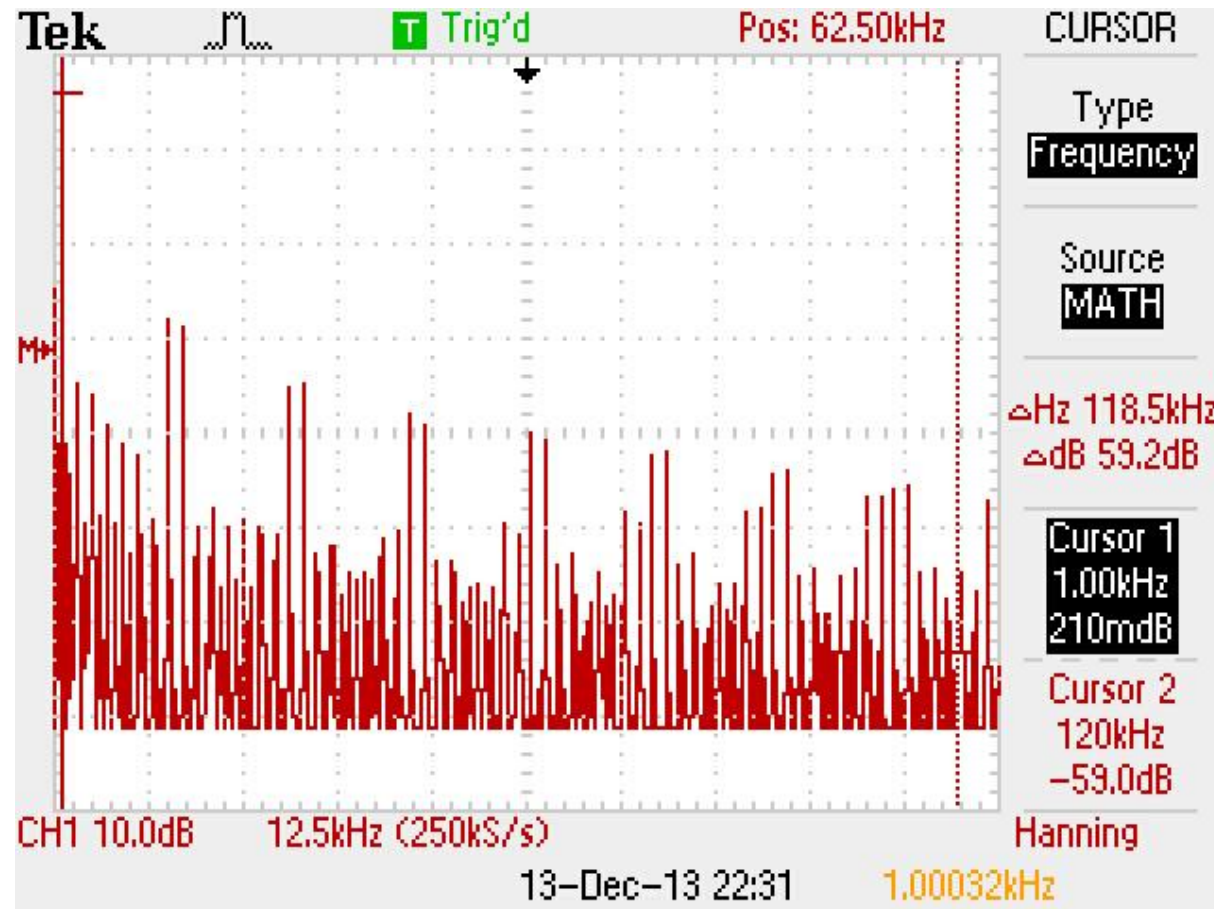
- Noot: interne tellerwaarde is afgebeeld als dut/count. Signaal sinq is ook als “analoog” signaal afgebeeld.

Tellers



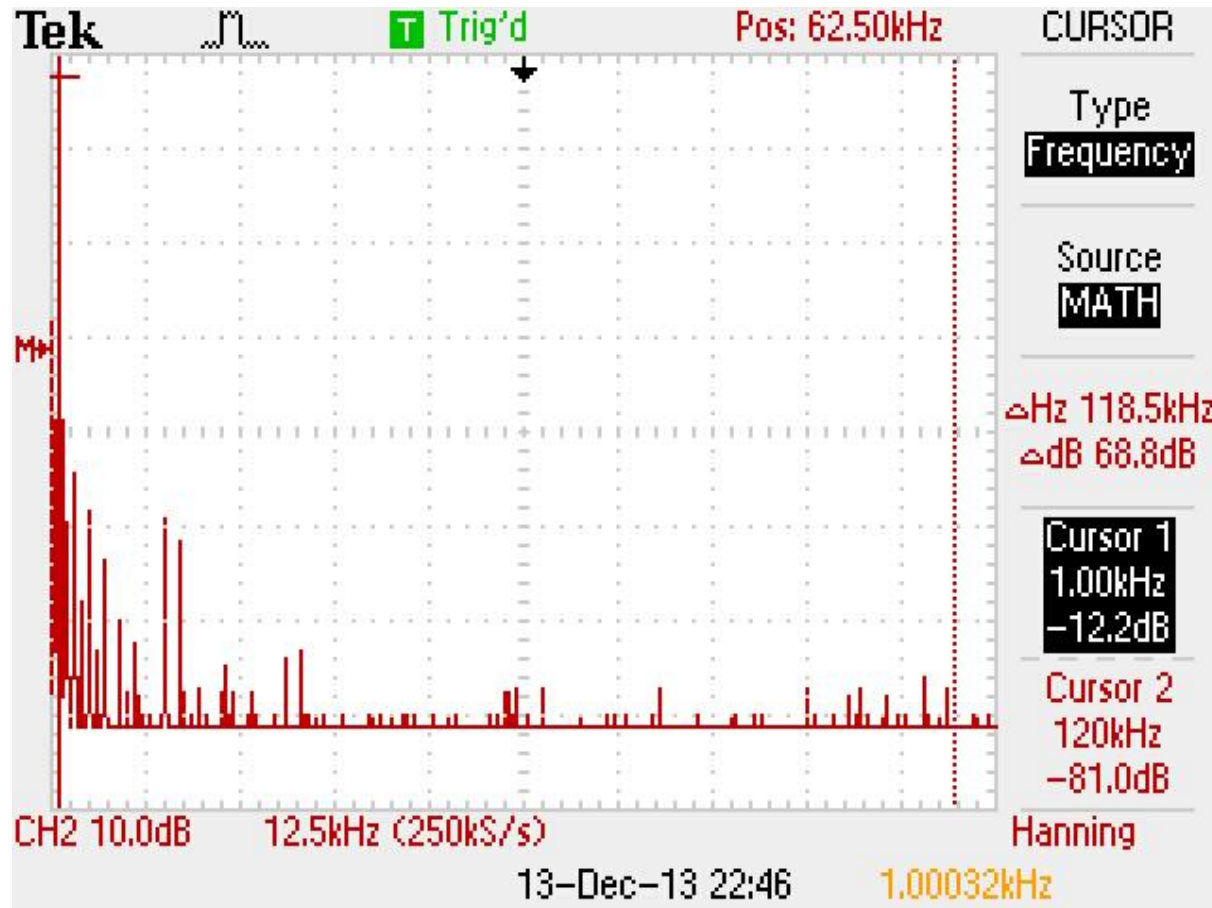
- Het gele, geblokte signaal komt direct uit de ADC.
- Het blauwe signaal is na LD-filtering.
- Merk op dat de Volts/div niet gelijk staan.
- De fasedraaiing komt door het LD-filter.

Tellers



- Het frequentie-spectrum laat een paal zien op 1 kHz.
- Maar ook op 15, 17 kHz, 31 kHz, 33 kHz, 47 kHz, 49 kHz

Tellers



- Het gefilterde signaal is al beter, er zijn nog maar een paar palen over.

Tellers

- We gaan nu een BCD-teller ontwikkelen.
- De BCD-teller telt van 0000 t/m 1001 en begint dan weer op 0000.
- De BCD-teller gebruikt dus 1000 “standen” van de 16 mogelijke standen.
- We stellen een tabel op van alle gebruikte standen en ontwikkelen de functies van de T-flipflops.

Tellers

- In de tabel links is de telcyclus te zien
- Q_0 toggelt altijd.
- Q_1 toggelt altijd op $Q_0 = 1$, maar alleen als $Q_3 = 0$.
- Q_2 toggelt bij $Q_3Q_2Q_1 = 0011$ en $Q_3Q_2Q_1 = 0111$. Q_3 kan verwijderd worden want die is dan nooit 1. Q_2 is zowel 0 als 1 dus kan verwijderd worden.
- Q_3 toggelt bij 0111 en bij 1001. Nu is 1001 de enige stand met $Q_3Q_0 = 11$ dus kunnen Q_2 en Q_1 verwijderd worden.

| Huidig | Volgend | Opmerkingen |
|--------|---------|---------------------|
| 0000 | 0001 | Begin cyclus |
| 0001 | 0010 | |
| 0010 | 0011 | Q1 toggelt |
| 0011 | 0100 | |
| 0100 | 0101 | Q2, Q1 toggelen |
| 0101 | 0110 | |
| 0110 | 0111 | Q1 toggelt |
| 0111 | 1000 | |
| 1000 | 1001 | Q3, Q2, Q1 toggelen |
| 1001 | 0000 | |
| 0000 | 0001 | Q3 toggelt |

Tellers

- De functies voor de T-ingangen is dan als volgt:

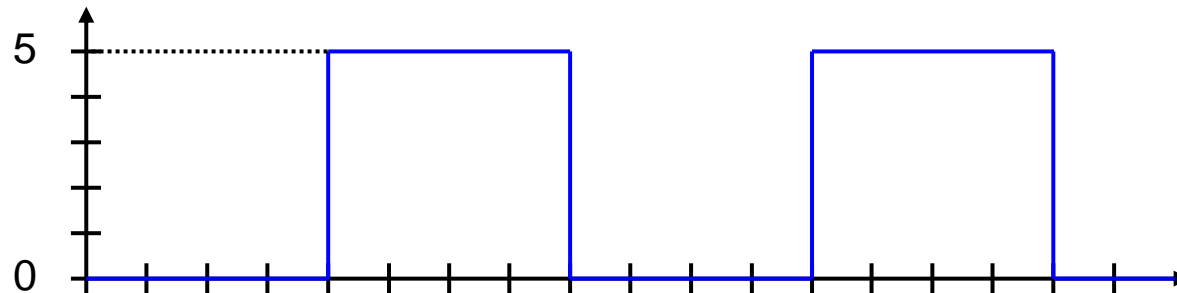
$$\begin{aligned}T_0 &= 1 \\T_1 &= \overline{Q_3} \cdot Q_1 \\T_2 &= Q_1 \cdot Q_0 \\T_3 &= Q_2 \cdot Q_1 \cdot Q_0 + Q_3 \cdot Q_0\end{aligned}$$

Tellers

- Het regelen van het toerental van een elektromotor kan eenvoudig worden gedaan door gebruik te maken van *Pulse Width Modulation* (PWM).
- Een elektromotor heeft een zekere *massatraagheid*. De motor reageert met een zekere traagheid als het stuursignaal (spanning of stroom) sterk verandert.
- Daar maakt PWM gebruik van. Door de motor met een *herhalende puls* (pulstrein) aan te sturen zal deze sneller of langzamer willen draaien.
- De pulstrein wordt door de motor *uitgemiddeld*.

Tellers

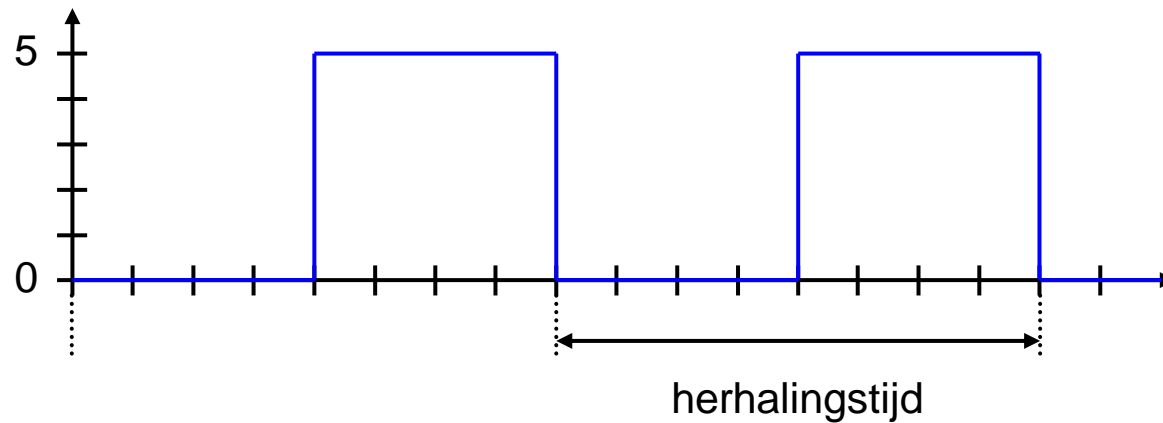
- De puls heeft een zekere maximum en minimum en kan gezien worden als een binair signaal (een signaal met slechts twee waarden).



- In de digitale techniek kan dit bijvoorbeeld tussen 0 V en 5 V zijn.
- Het minimum kan ook negatief zijn.

Tellers

- De *herhalingstijd* van de puls is de tijdsduur tussen het begin en eind van de pulscyclus.



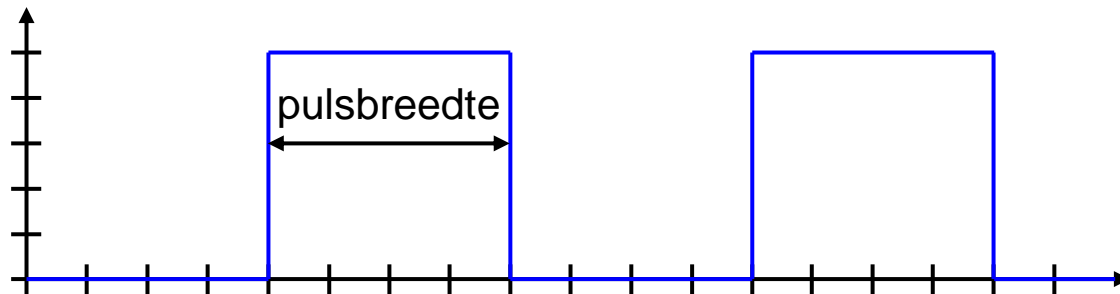
- Deze herhalingstijd moet veel kleiner zijn dan de reactiesnelheid van de motor.

Tellers

- De inverse van de herhalingsstijd is de PWM-schakelfrequentie.
- Voor een elektromotor ligt de schakelfrequentie tussen enkele kilohertz (kHz) en enkele tientallen kilohertz.
- Voor een dimmer ligt dit op 100 Hz.
- Voor een oven ligt dit rond 30 mHz (5x per minuut).
- Recente ontwikkelingen zijn *klasse-D* audioversterkers.

Tellers

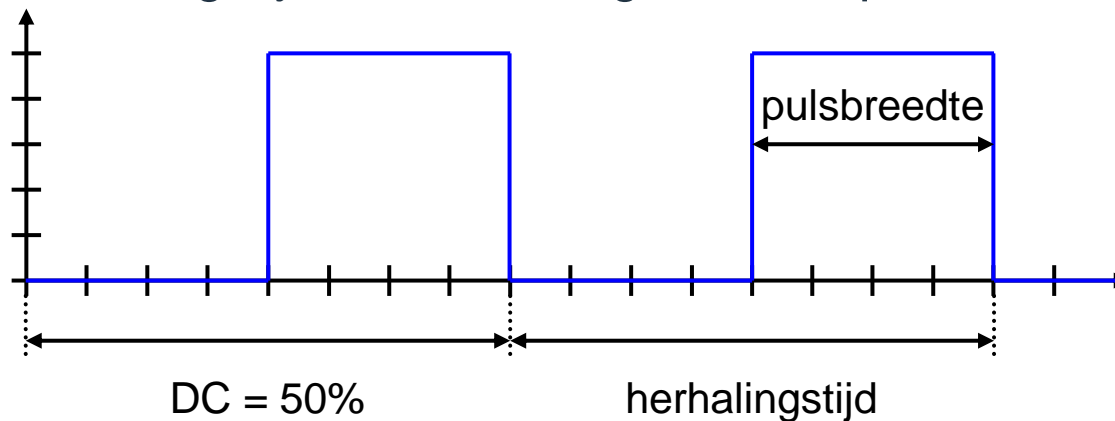
- De *pulsduur* of *pulsbreedte* van de puls is de tijdsduur dat de puls 'aan' is.



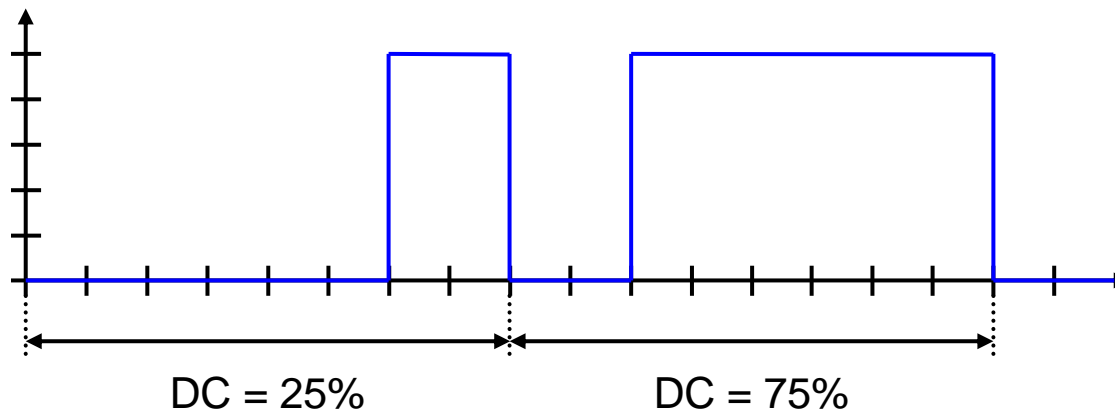
- De pulsbreedte ligt tussen 0 en de herhalingsstijd.

Tellers

- De Duty Cycle (DC) is de verhouding tussen de pulsbreedte en de herhalingsstijd en wordt uitgedrukt in procenten:



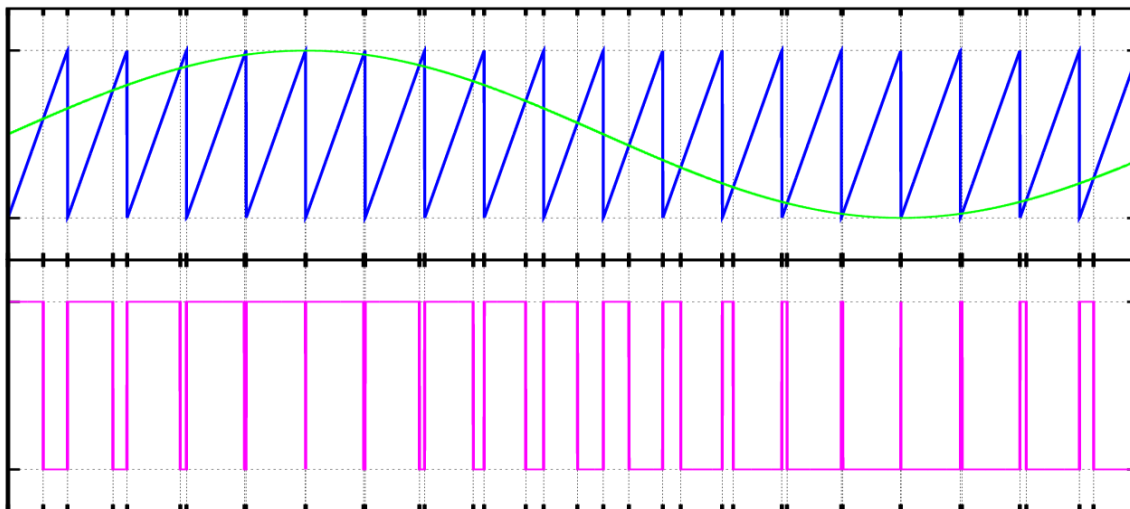
$$DC = \frac{\text{pulsbreedte}}{\text{herhalingsstijd}} \times 100\%$$



DC = 75%

Tellers

- De eenvoudigste manier om PWM-signalen te genereren is met behulp van een *zaagtand-* of *driehoeksignaal* en een vergelijkschakeling (comparator).
- Als de waarde van een referentiesignaal (groen) groter is dan dat van zaagtandsignaal (blauw) dan is het PWM-signaal (magenta) maximaal.

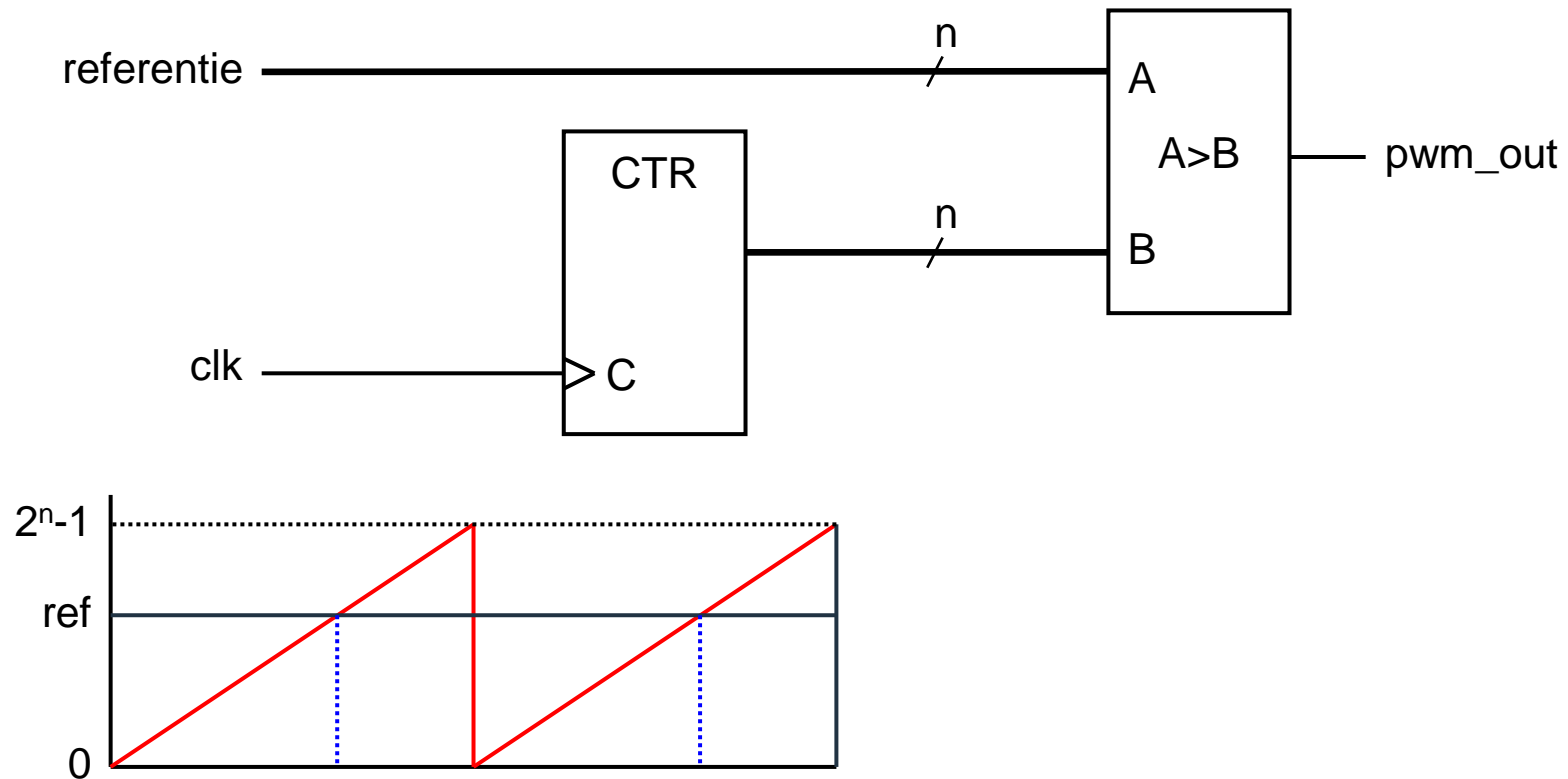


Tellers

- Deze techniek kan eenvoudig worden uitgewerkt in een digitale schakeling.
- De signalen worden *gedigitaliseerd* (omgezet naar unsigned getallen).
- Het zaagtandsignaal wordt gemaakt door een teller cyclisch te laten tellen van 0 tot een maximum.
- Het referentiesignaal is gewoon een binair getal.
- Beide worden aangeboden aan een vergelijkschakeling.

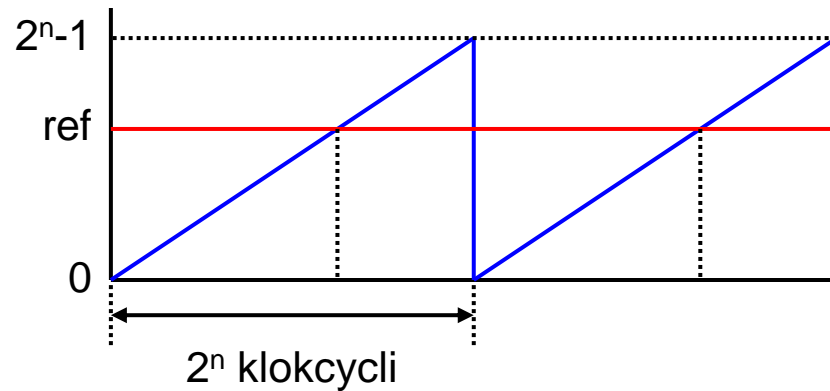
Tellers

- Hieronder het basisschema voor PWM-generatie. Als de referentie groter is dan de tellerwaarde is de PWM-uitgang hoog.



Tellers

- De schakelfrequentie is $f_{\text{PWM}} = \frac{f_{\text{clk}}}{2^n}$



- De Duty Cycle is $DC = \frac{\text{ref}}{2^n} \times 100\%$
- Merk op: de DC kan nooit 100% zijn!

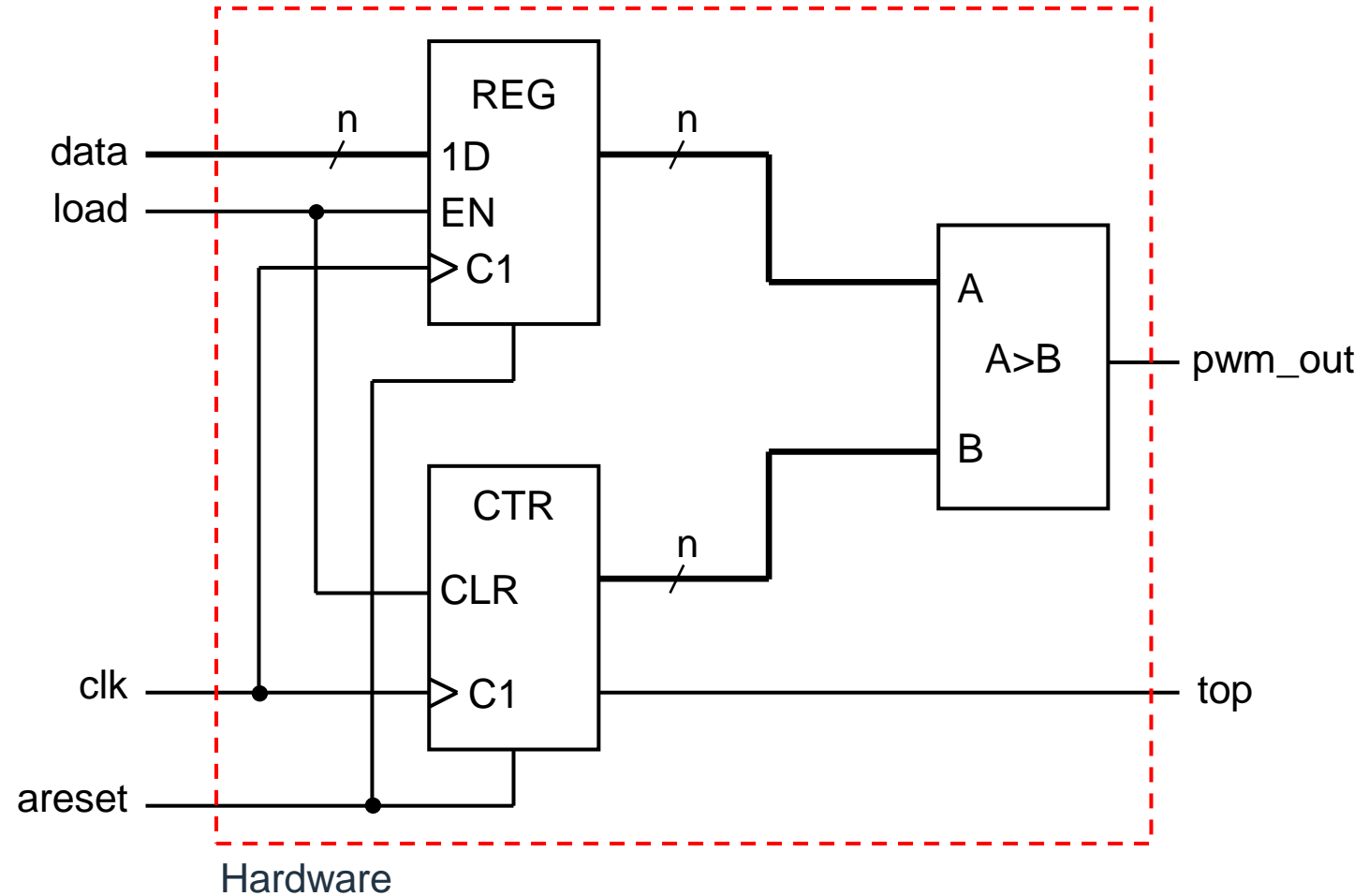
Een DC van 100% is een gelijkspanning met maximale waarde!

Tellers

- De functionele beschrijving van de PWM-schakeling:
- Het systeem moet een nieuw referentiegetal kunnen laden. De PWM-generatie wordt dan opnieuw gestart (de cyclus start opnieuw).
- De teller telt van 0 t/m 2^n-1 en begint dan weer op 0.
- Op enig moment is de PWM-uitgang hoog als de waarde van het referentiegetal groter is dan de waarde van de teller.
- Als de teller op de maximale waarde staat moet een puls worden afgegeven (handig voor synchronisatie met externe componenten).
- Het systeem heeft een asynchrone reset.

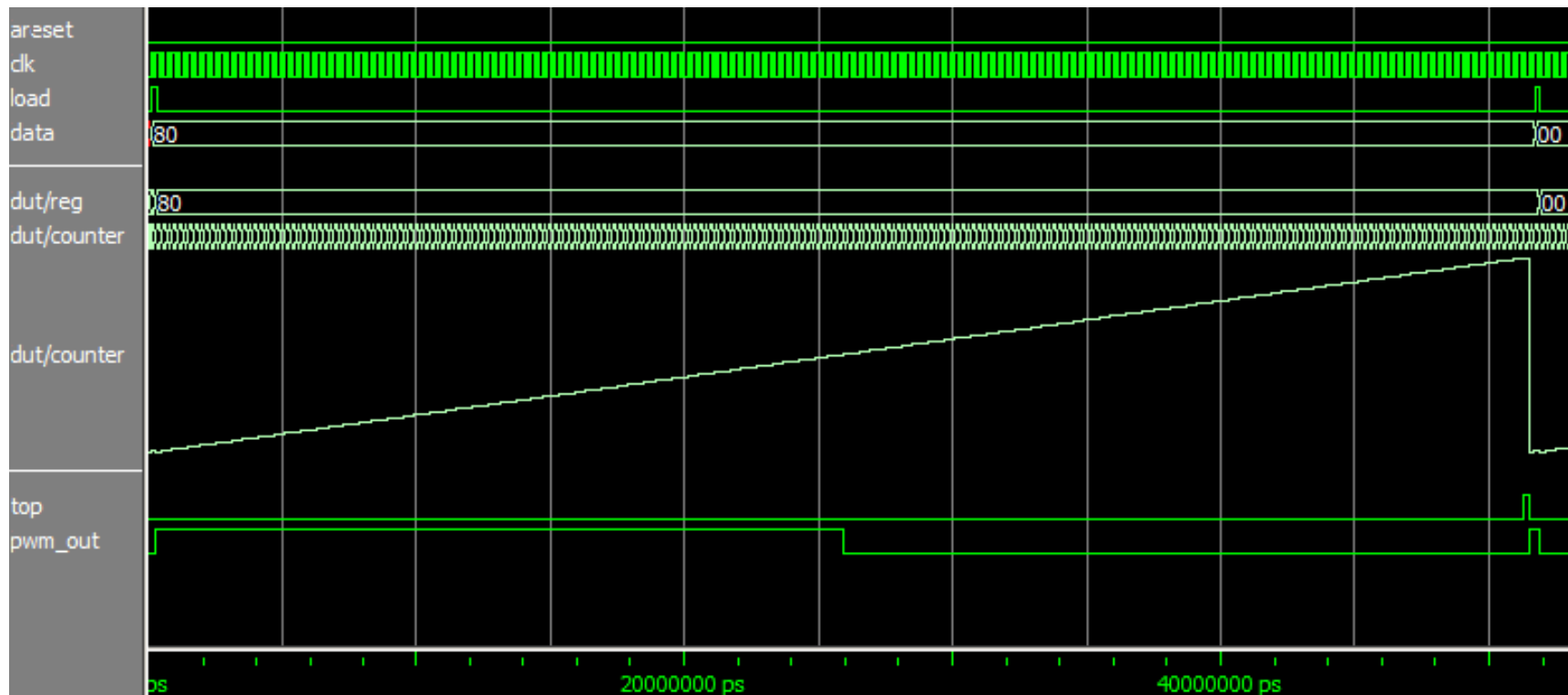
Tellers

- De schakeling bestaat uit een register om de referentiewaarde te laden.
- De teller telt tussen 0 en 2^n-1 .
- De vergelijker vergelijkt de tellerstand met de referentiewaarde.
- Top geeft aan wanneer de teller op de eindstand staat.



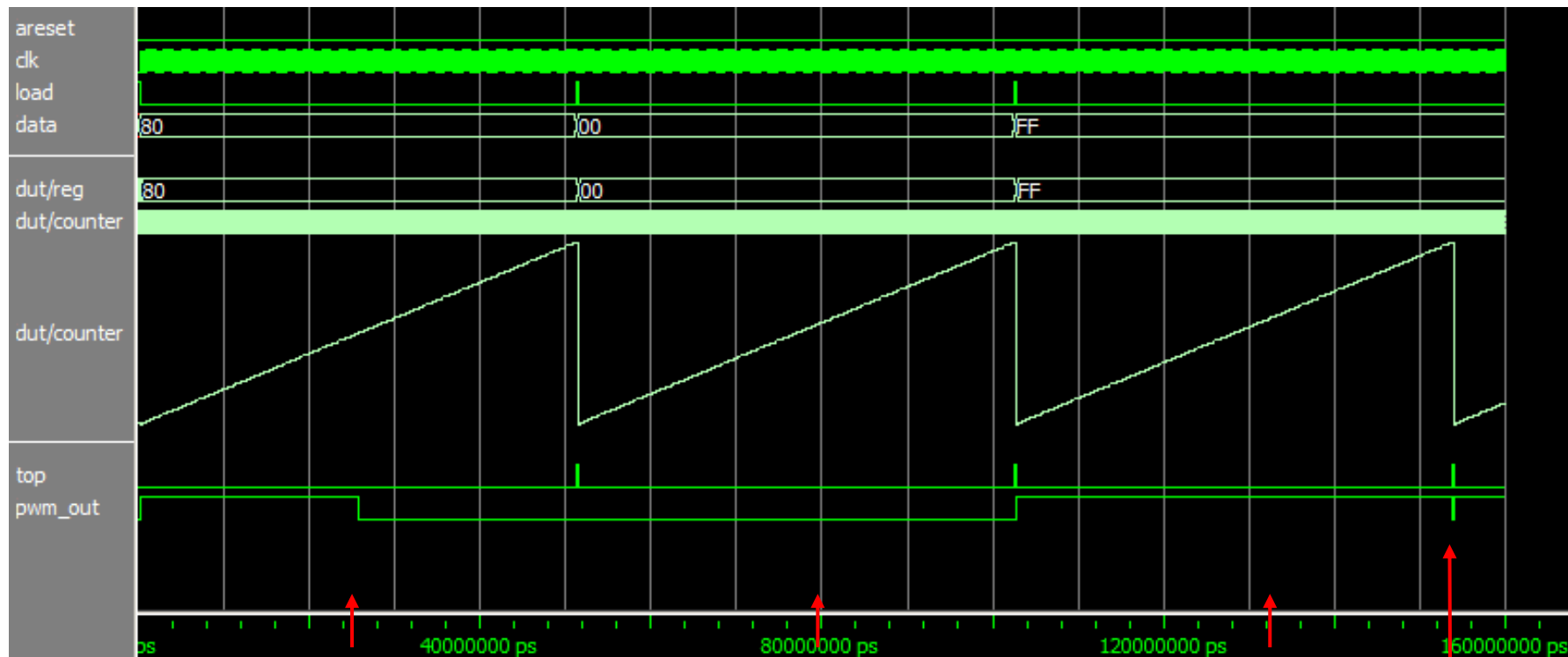
Tellers

- Hieronder een simulatie van een 8-bits systeem met referentie 128 (80_{16}).



Tellers

- Hieronder het simulatieresultaat voor drie cycli met verschillende referentiewaarden.



DC = 50%

DC = 0%

DC ≈ 100%

ref = counter

let's change