



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

Digitale System Engineering 1

Week 4 – Toepassing: Pulse Width Modulation
Jesse op den Brouw
DIGSE1/2018-2019

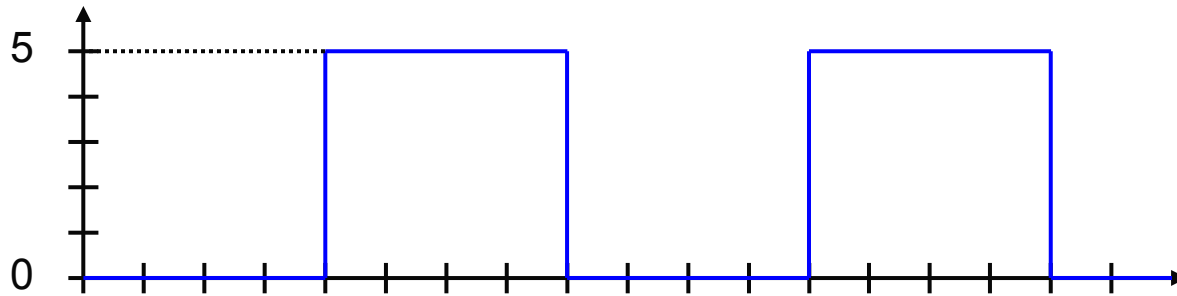
DE HAAGSE
HOGESCHOOL

PWM basics

- 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*.

Pulshoogte

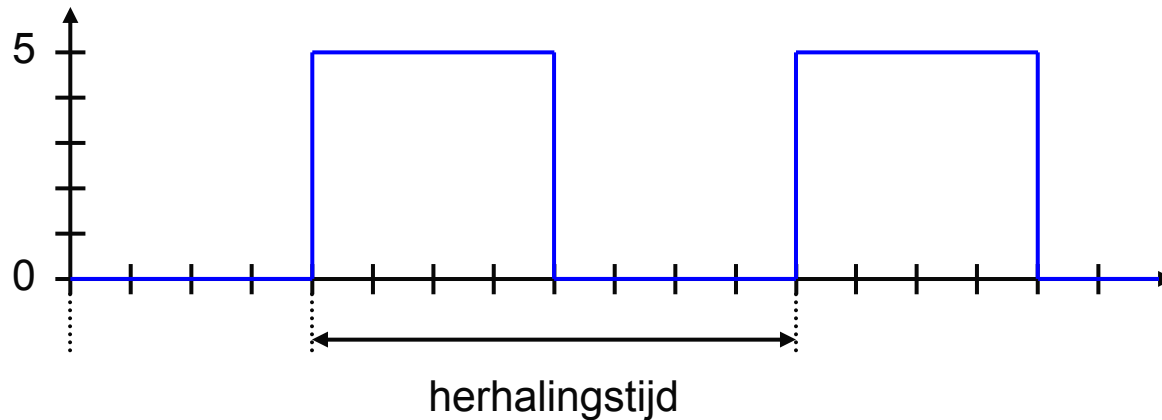
- De puls heeft een zekere maximum en minimum en kan gezien worden als een binair signaal (een signaal met slechts twee waarden).



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

Herhalingstijd

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



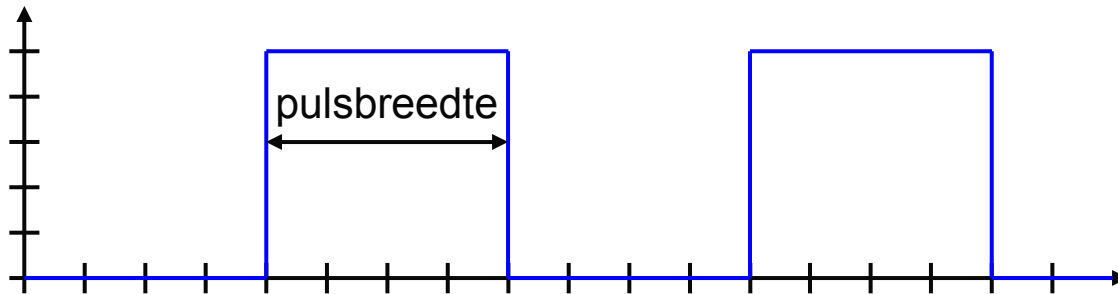
- Deze herhalingstijd moet veel kleiner zijn dan de reactiesnelheid van de motor.

Schakelfrequentie

- 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.

Pulsduur, pulsbreedte

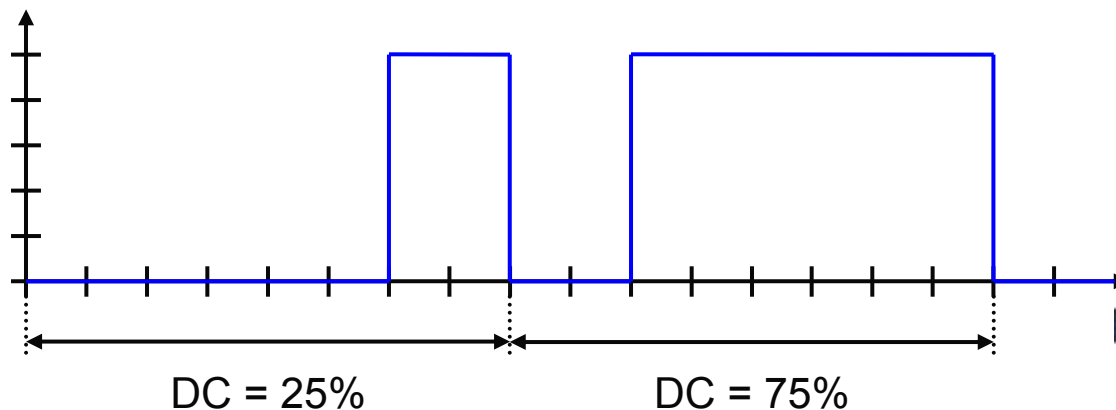
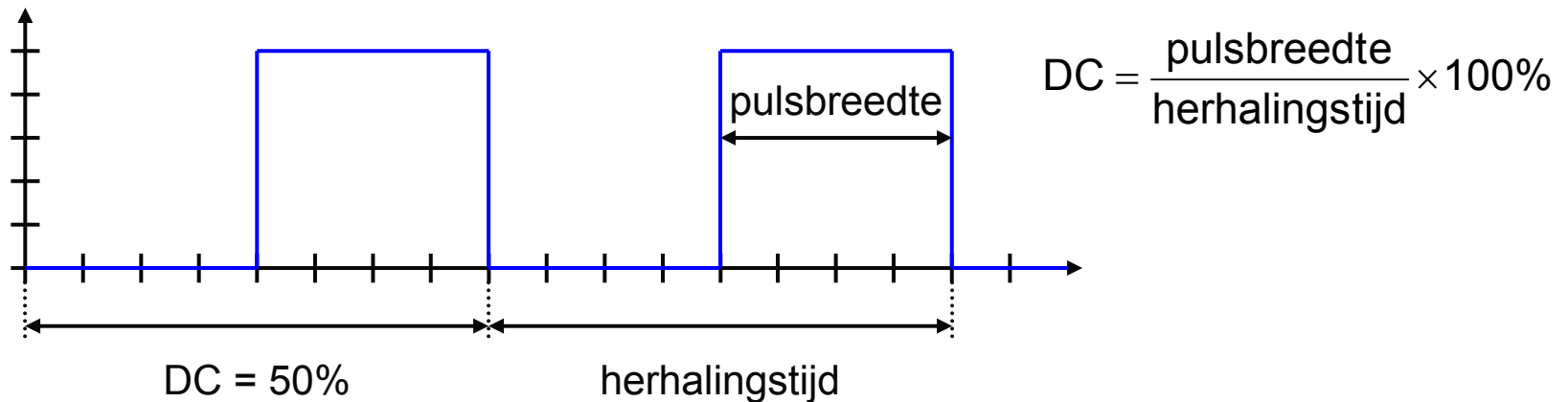
- De *pulsduur* of *pulsbreedte* van de puls is de tijdsduur dat de puls 'aan' is.



- De pulsbreedte ligt tussen 0 en de herhalingsstijd.

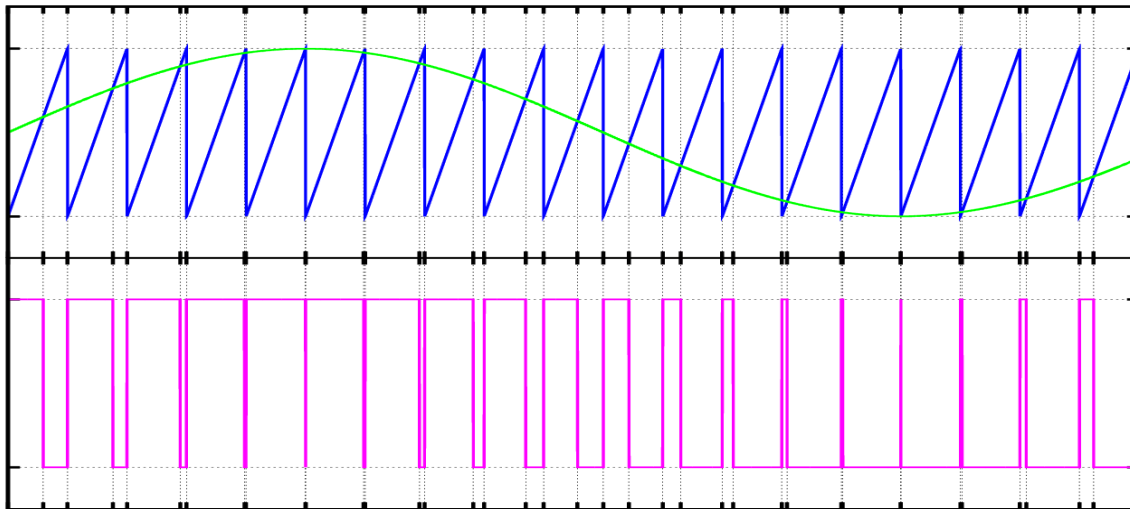
Duty Cycle

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



PWM-siginaal generatie

- 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-siginaal (magenta) maximaal.



Bron: wikipedia

Gemiddelde waarde

- De gemiddelde waarde van een PWM signaal is

$$\begin{aligned}U_{\text{gem}} &= \text{DC} \cdot (U_{\text{max}} - U_{\text{min}}) + U_{\text{min}} \\ &= \text{DC} \cdot U_{\text{max}} + (1 - \text{DC}) \cdot U_{\text{min}}\end{aligned}$$

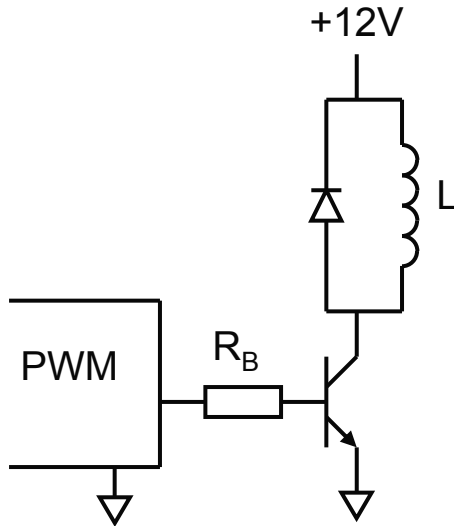
- Bij een U_{min} van 0V wordt dit

$$U_{\text{gem}} = \text{DC} \cdot U_{\text{max}}$$

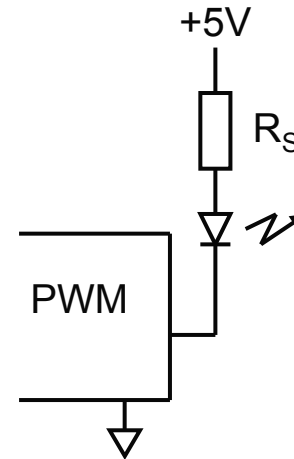
- PWM 0V – 5V, DC = 50%: $U_{\text{gem}} = 2,5\text{V}$
- PWM 0V – 12V, DC = 25%: $U_{\text{gem}} = 3,0\text{V}$

Voorbeelden

- Hieronder twee voorbeelden van sturing van elektronische componenten door middel van PWM.



sturing van een spoel (motor)



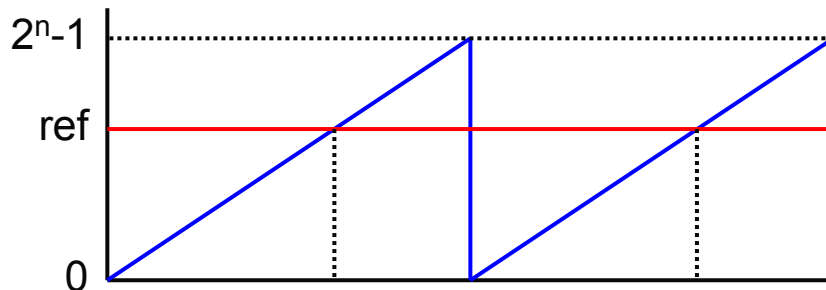
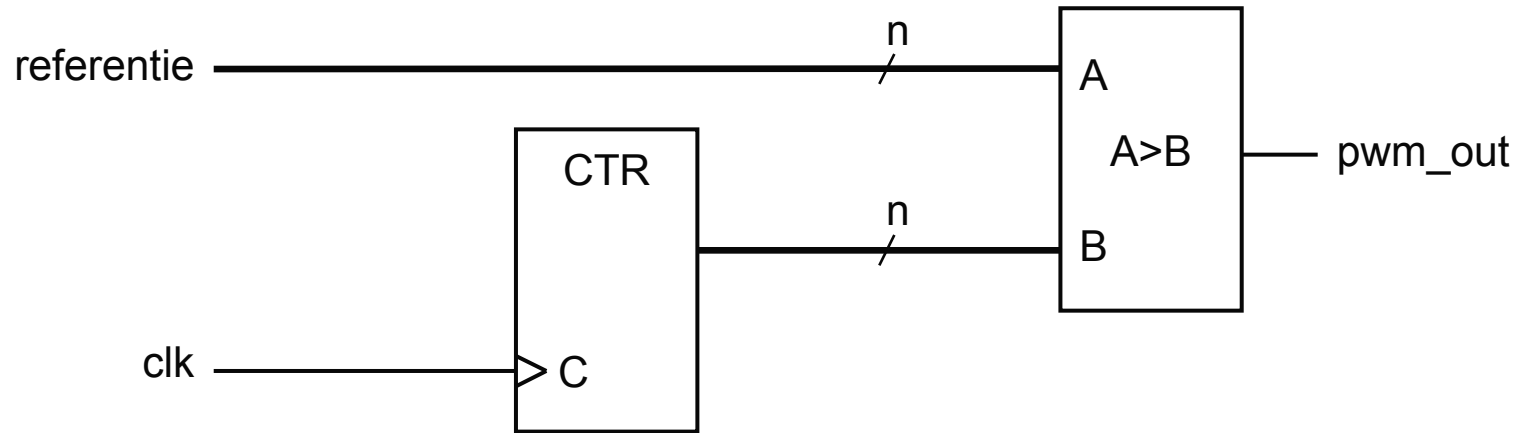
sturing van een led

Digitale PWM-generatie

- 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.

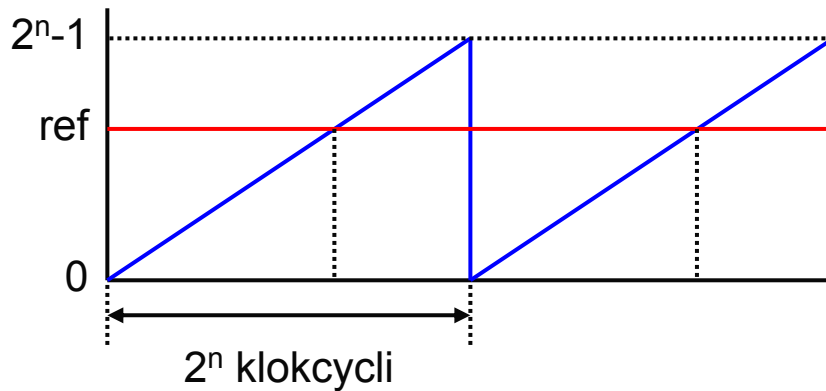
Digitale PWM-generatie

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



Digitale PWM-generatie

- 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!

Opgaven

- Bepaal de resolutie van de DC voor een n-bit digitaal PWM-systeem.
- Het PWM-signaal is hoog als de referentiewaarde groter is dan de tellerwaarde (op enig moment). Daardoor is een DC van 100% niet haalbaar. Een student past de schakeling aan zodat het PWM-signaal is hoog als de referentiewaarde groter *of gelijk* is dan de tellerwaarde (op enig moment). Is dit een slimme keuze? Zijn er andere problemen te verwachten? Motiveer het antwoord.
- Bedenk een andere manier om tot een DC van 100% te komen zonder de vergelijkschakeling aan te passen.

PWM hardware

- 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.

PWM hardware

- De betekenis van de signalen:

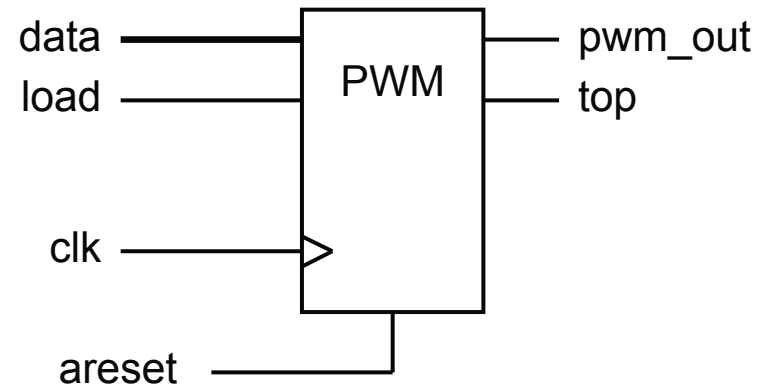
data : het 8-bit PWM getal

load: laden nieuwe waarde

pwm_out : het PWM output signaal

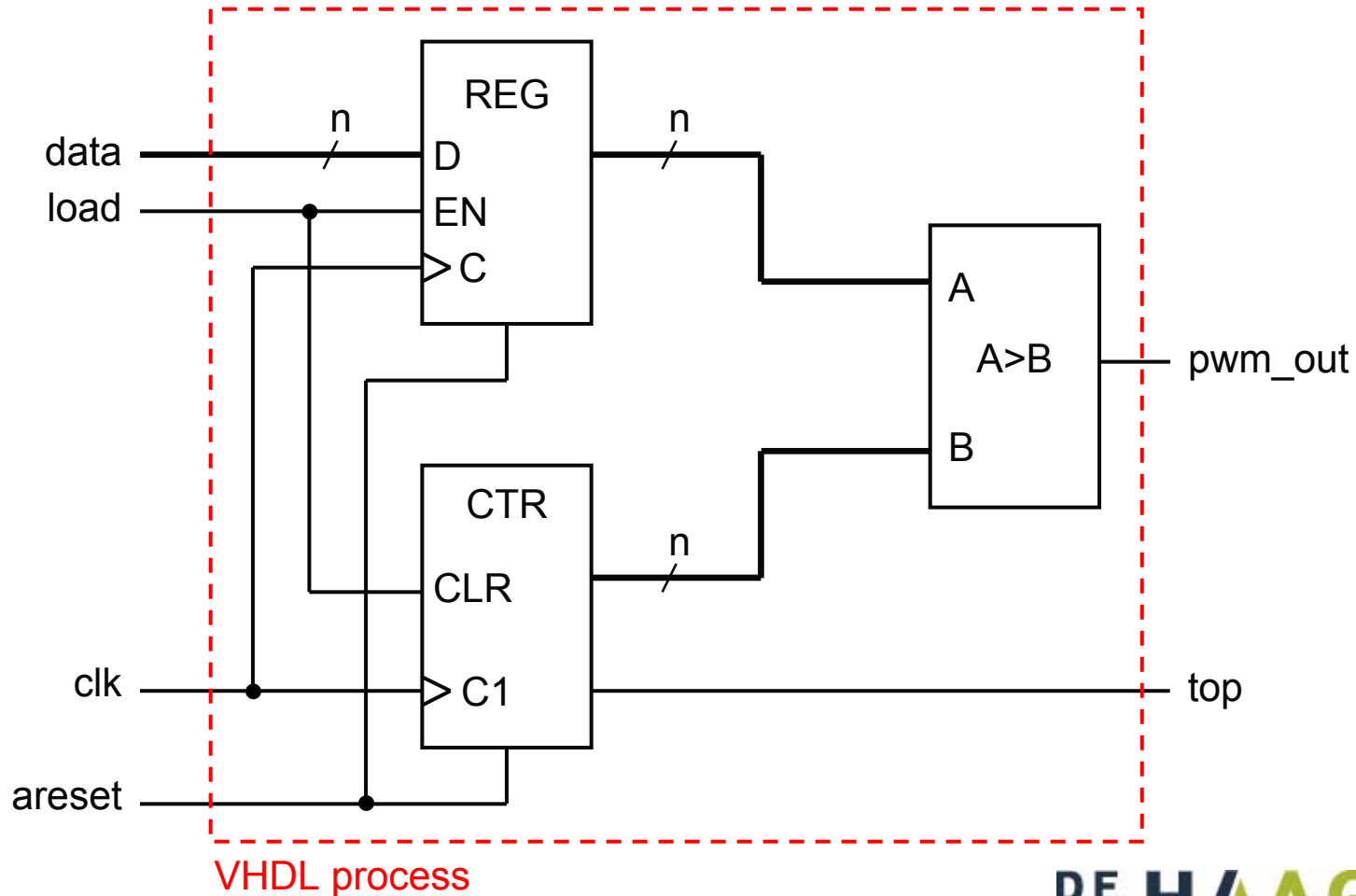
top: einde PWM-cyclus

areset: asynchrone reset



- Natuurlijk kan de beschrijving helemaal structural worden opgesteld met op de onderste laag de gedragsbeschrijvingen, maar hier is gekozen voor een meer algoritmische benadering. Het toont goed de kracht van VHDL.

PWM hardware



VHDL-code entity

- De entity-beschrijving.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;           -- for '+' and '>'

entity pwm_8bit is
  port (clk      : in std_logic;
        areset   : in std_logic;
        load     : in std_logic;    -- Load new data
        data     : in unsigned (7 downto 0);
        top      : out std_logic;   -- counter at top
        pwm_out  : out std_logic    -- PWM output
  );
end pwm_8bit;
```

VHDL-code architecture

- Er is een *teller* nodig die de PWM-cyclus bijhoudt.
- De aangeboden referentiewaarde moet intern opgeslagen worden in een *register*.

```
architecture rtl of pwm_8bit is
  signal counter : unsigned (7 downto 0);
  signal reg : unsigned (7 downto 0);
begin
  ...
end rtl;
```

VHDL-code process en asynchrone reset

- Het PWM-proces is gevoelig voor de klok, de asynchrone reset, het register en de teller (later zal blijken waarom).

```
process (areset, clk, reg, counter) is
begin
  if (areset = '1') then
    counter <= "00000000";    -- reset the PWM-counter
    reg <= "00000000";       -- reset PWM data register
  elsif rising_edge(clk) then
    ...
end process;
```

VHDL-code laaddetectie

- Alle andere acties gaan onder besturing van de opgaande flank. Als load actief wordt moet het systeem geladen worden met beginwaarden.
- Zo niet, dan moet de teller doorgaan met de PWM-cyclus.


```
elsif rising_edge(clk) then      -- on positive edge
  if load = '1' then           -- if load
    reg <= data;                -- load new PWM value
    counter <= "00000000";     -- restart PWM cycle
  else
    counter <= counter + 1;    -- update counter
  end if;
end if;
```

VHDL-code top

- Als de teller aan het eind van de PWM-cyclus is (top), dan moet een puls worden afgegeven.
- Dit is een combinatorische schakeling.

```
if counter = "11111111" then    -- all 1's, counter at top
    top <= '1';
else
    top <= '0';
end if;
```

counter wordt
"gelezen", dus in
de sensitivity list



VHDL-code PWM output

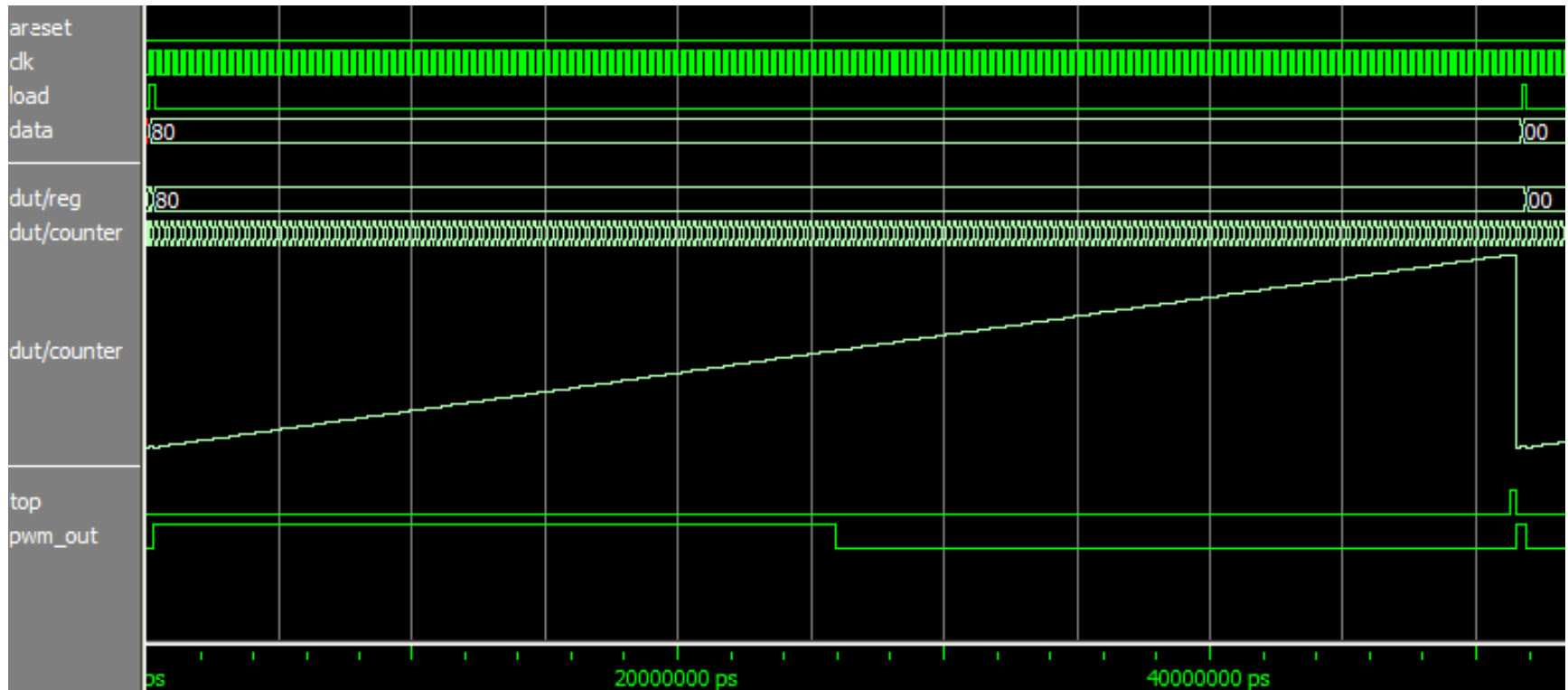
- Het daadwerkelijk genereren van het PWM-signaal gebeurt aan de hand van een *vergelijker* (comparator).
- Dit is een combinatorische schakeling.

```
if reg > counter then -- ref. waarde higher then counter
    pwm_out <= '1';
else
    pwm_out <= '0';
end if;
end process;
```

← counter en reg worden “gelezen”,
dus in de sensitivity list

Simulatieresultaat

- Hieronder het simulatieresultaat voor één cyclus met referentie 80_{16} .



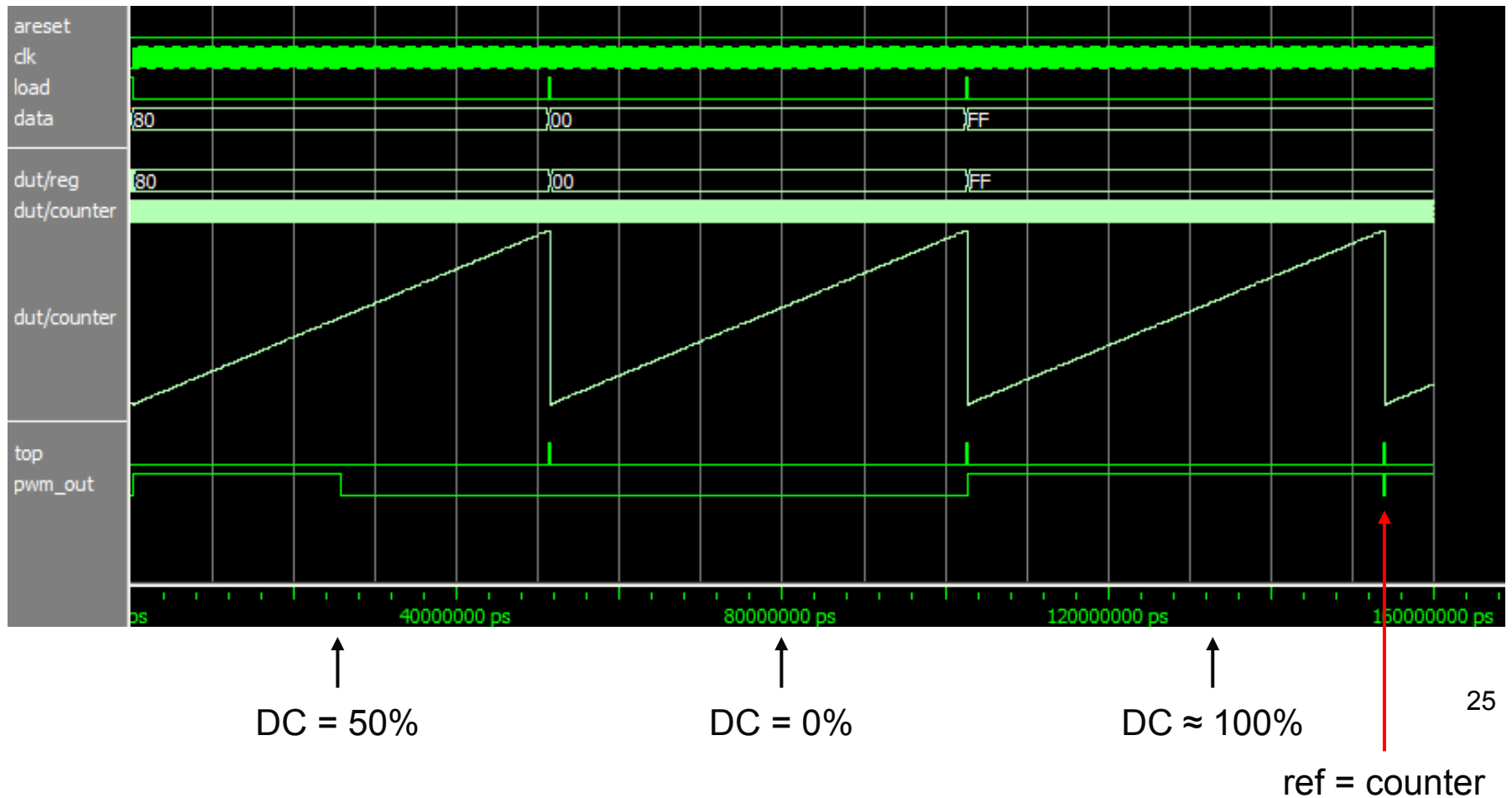
↑
load actief op flank, register
wordt geladen met 8-bit
referentiewaarde, teller wordt
herstart.

↑
teller groter dan
referentiewaarde,
pwm-uitgang laag

↑
einde cyclus 24

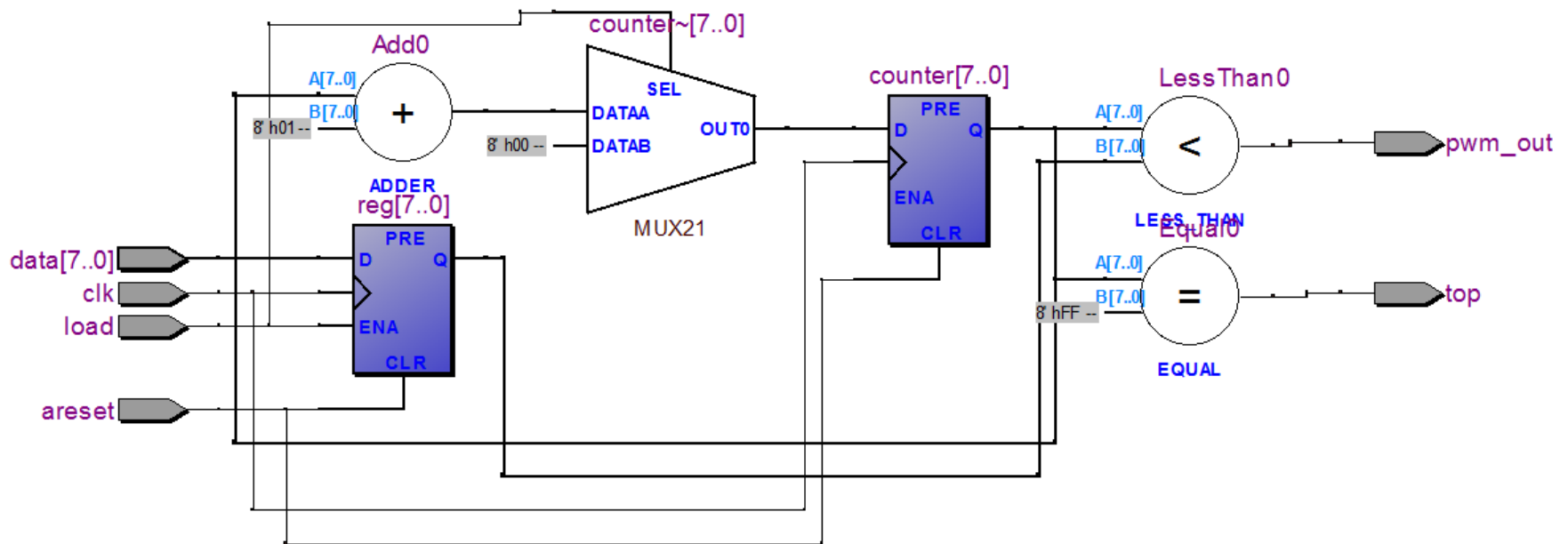
Simulatieresultaat

- Hieronder het simulatieresultaat voor drie cycli met verschillende referentiewaarden.



Synthese van PWM hardware

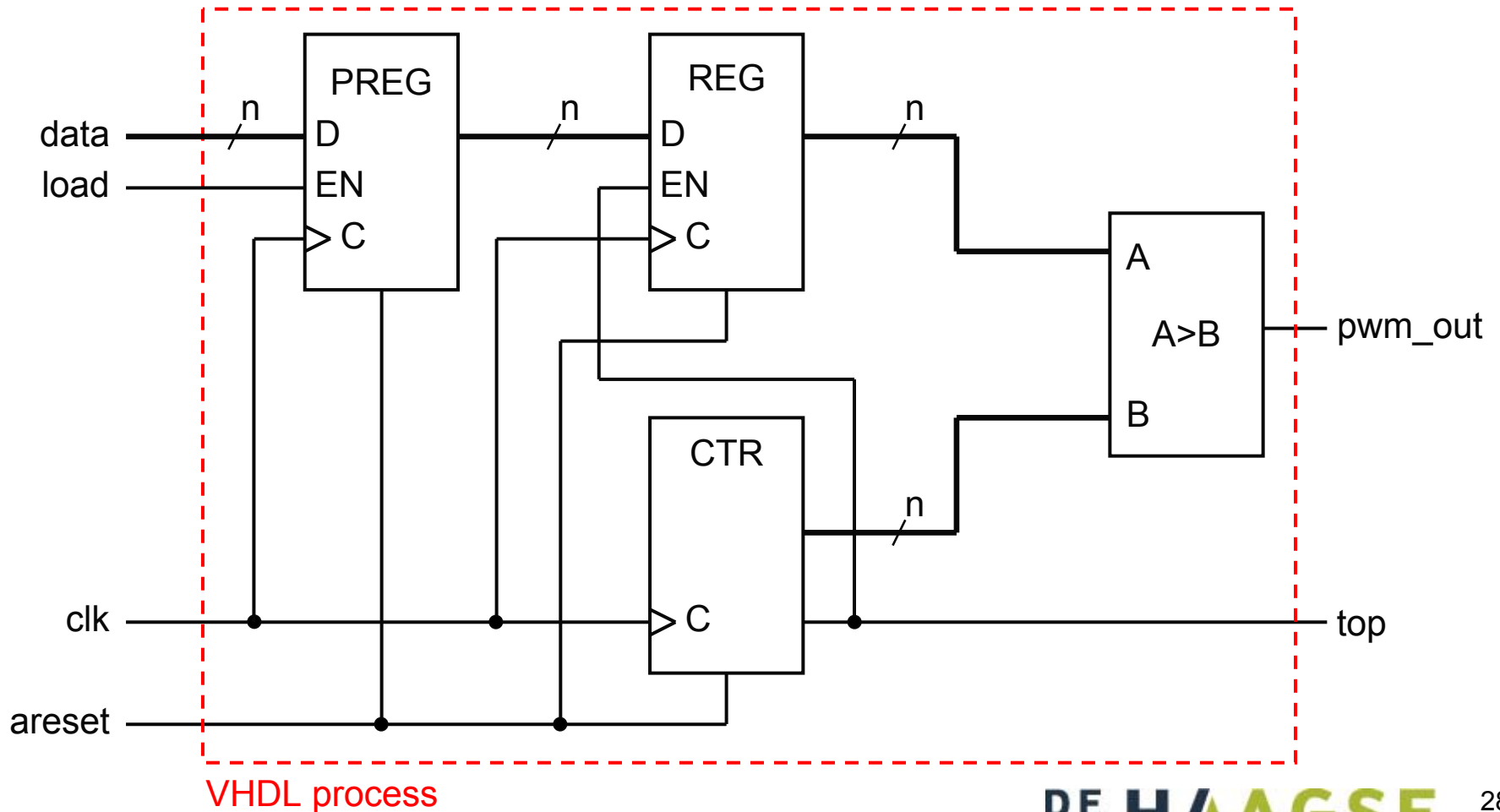
- Hieronder de synthese van de VHDL-beschrijving in *primitives*.



PWM hardware met gesynchroniseerde load

- Bij het laden van een nieuwe referentiewaarde wordt de PWM-cyclus afgebroken.
- Dat kan voorkomen worden door te wachten tot de PWM-cyclus is afgelopen (via het top-signaal) en dan een nieuwe waarde te laten.
- Met een extra register (preload register) is dit probleem eenvoudig te verhelpen.
- Het preload register kan op elk moment geladen worden, maar wordt pas doorgesluisd als de PWM-cyclus is afgelopen.

PWM hardware met gesynchroneerde load



Referenties

- <http://www.netrino.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation>
- <http://www.discovercircuits.com/H-Corner/PWM.htm>
- http://en.wikipedia.org/wiki/Pulse-width_modulation
- <http://www.ti.com/lit/an/slvae05/slvae05.pdf>
- <http://en.wikipedia.org/wiki/Dimmer>



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