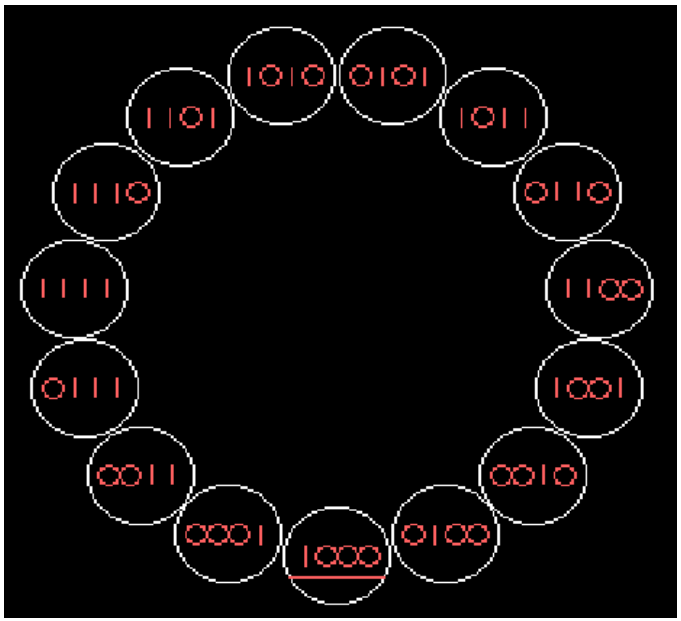
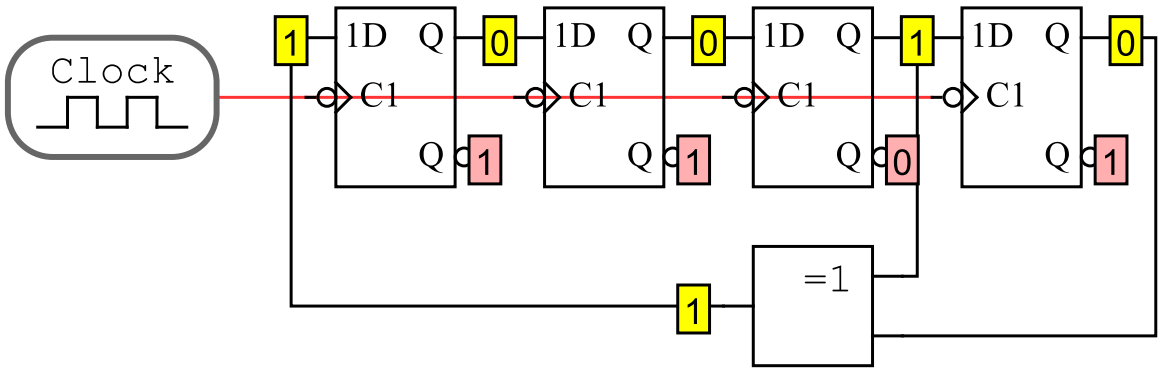
Reconstruction: NLT-module Digitale techniek
Context: Disco

Context

- If humidity is > 80% or the temperature > 26 ℃ a fan is blowing;
- If there are ≥ 31 persons inside the disco a red light switches on;
- If the noise level > ..dB the volume knob of the sound amplifier turn to the left.
- A disco light must be designed and build;

A pseudo random generator to steer the disco lights (linear feed back shift register)



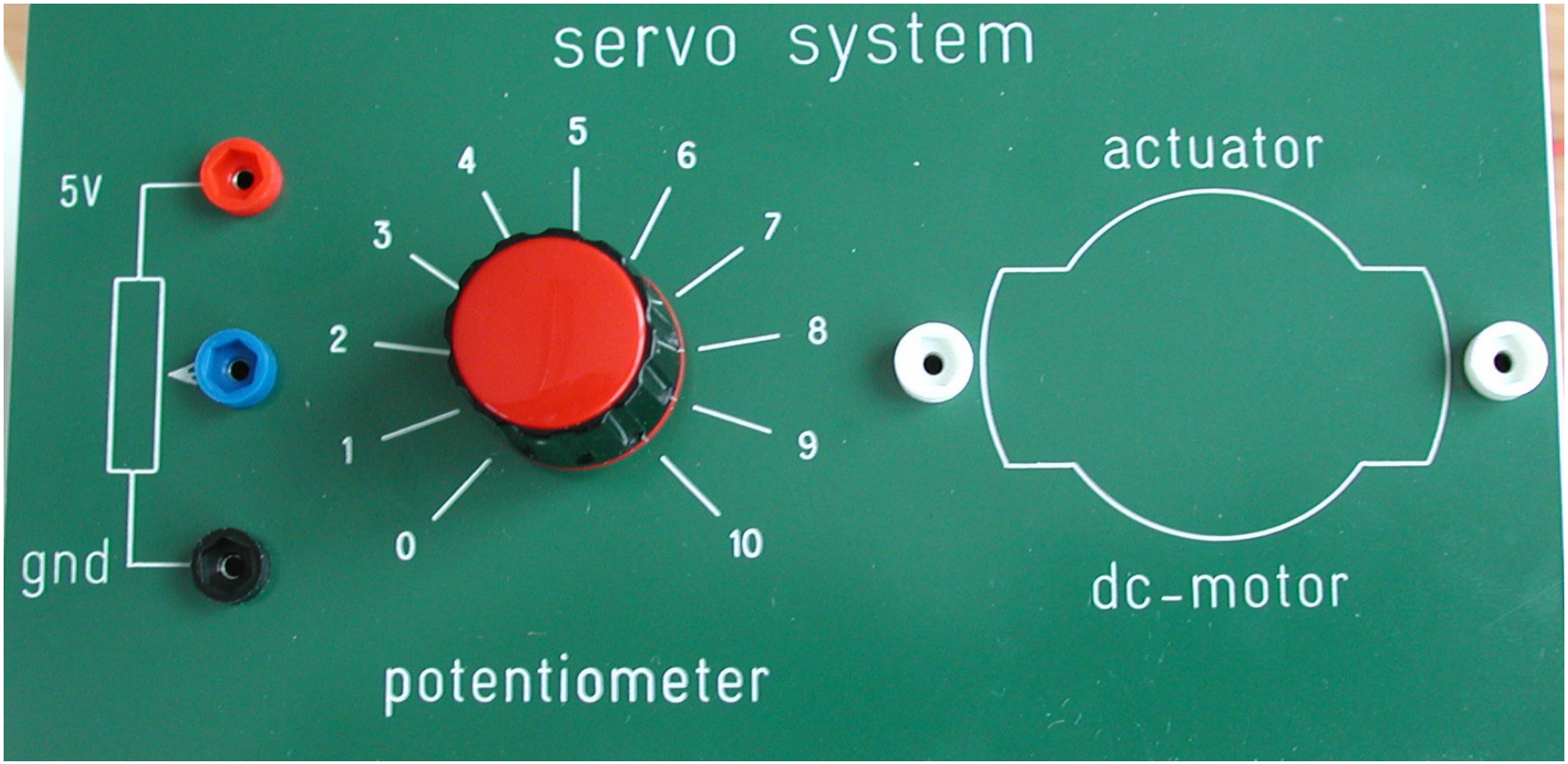


Required sensors:

- Humidity sensor,
- Temperature sensor,
- Photogate
- Sound sensor
- Position of a volume regulator

Required actuators:

- Disco light
- Fan
- Volume regulator to prevent ear-damage



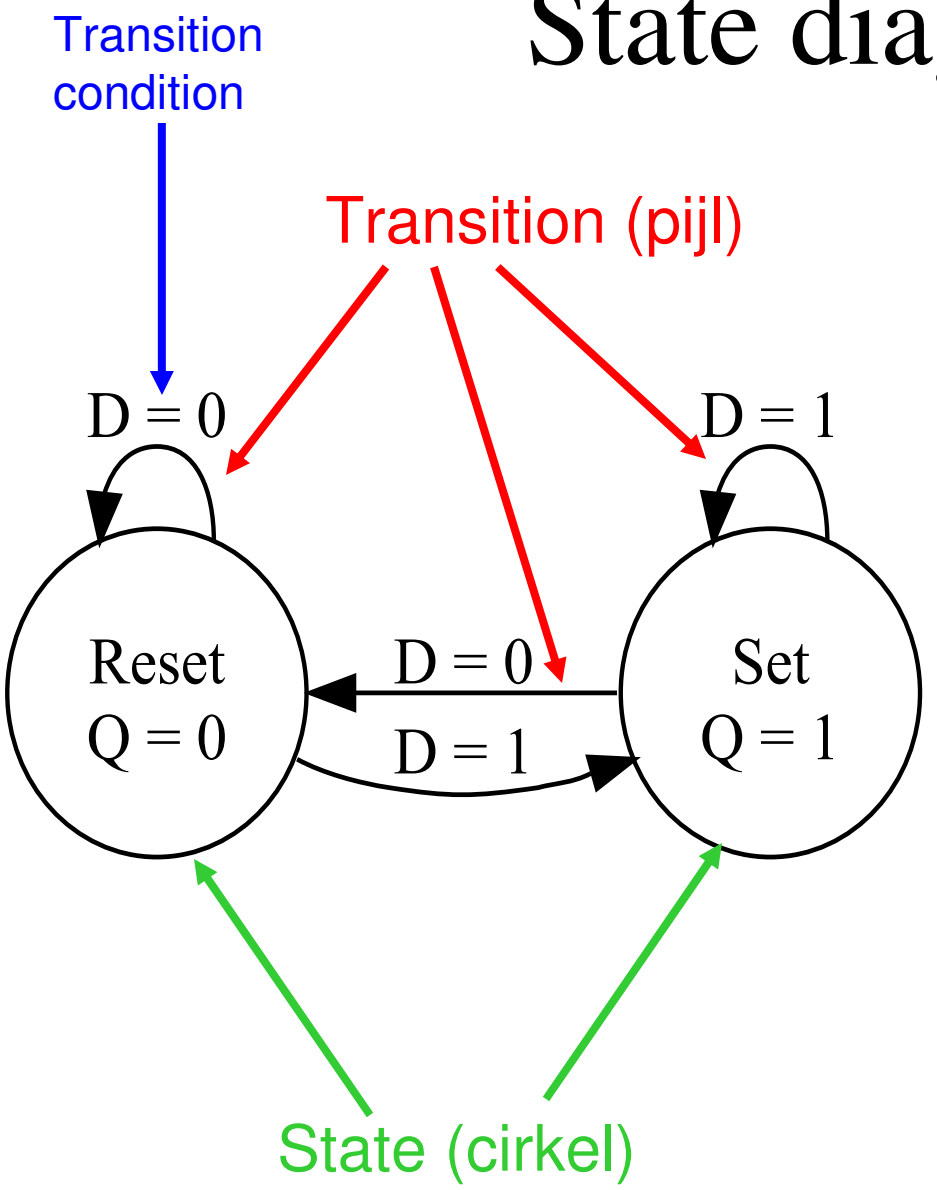
Ontwerpen van een eindige toestandsautomaat

- Flankgetriggerde D-flipflop
- Toestandsdiagram
- Toestandstabel/waarheidstabel
- Som van mintermen
- Vereenvoudigen van deze som
- Implementatie
- Demonstratie

Wat is een eindige toestandsautomaat?

- Een **eindige toestandsautomaat of Finite-state machine** is een model voor het gedrag voor een systeem, bestaande uit een eindig aantal toestanden, overgangen tussen die toestanden en acties.

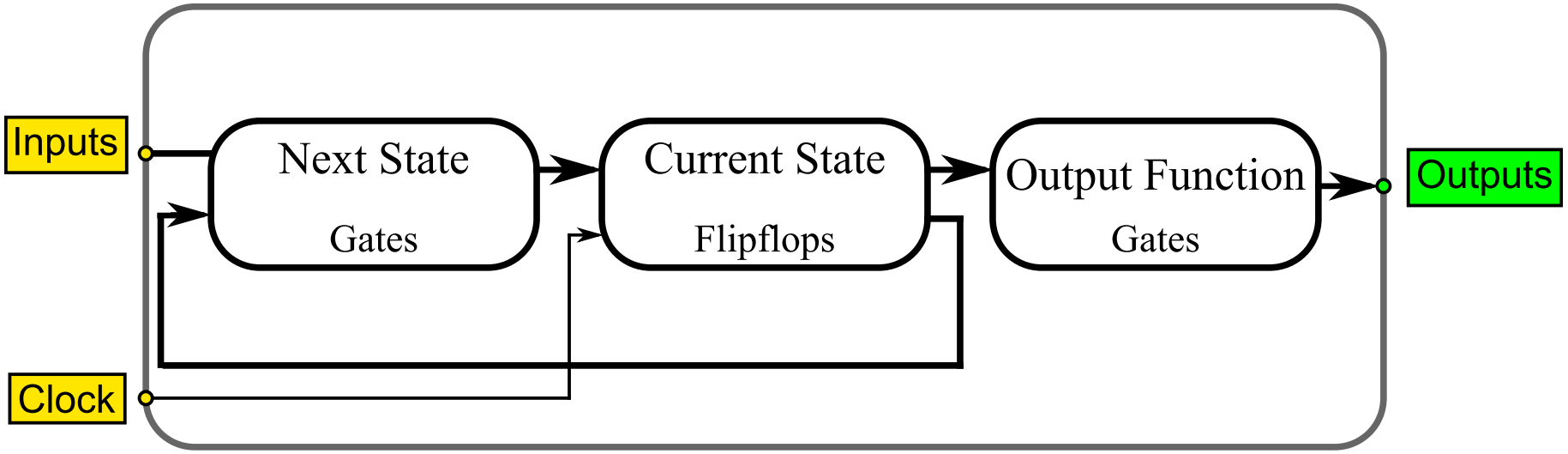
State diagram



D = 0 (close door)
D = 1 (open door)
Q = 0 (door is closed)
Q = 1 (door is open)

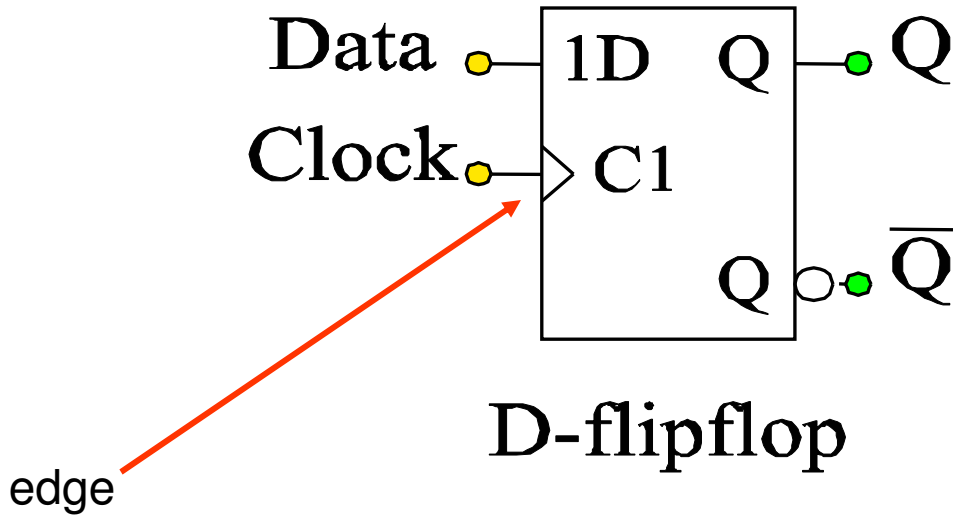
Q = {0,1} verzameling toestanden
D = {0,1} invoeralfabet
0 ϵ Q is de initiële toestand
δ = is de overgangsfunctie

Finite state machines



Moore Machine

Current state: D-flipflop



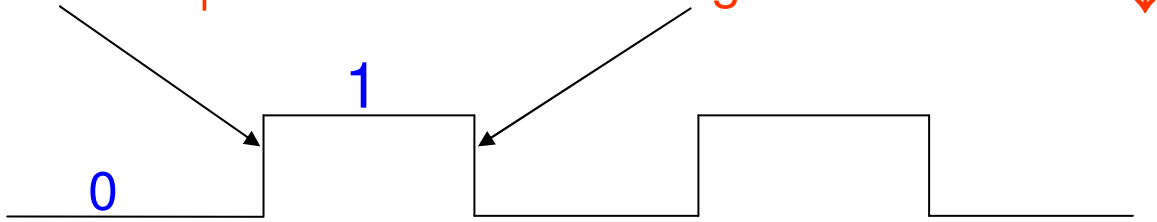
Level triggerered & Edge triggered

0 → 1 -overgang

opgaande klokflank ↑

1 → 0 -overgang

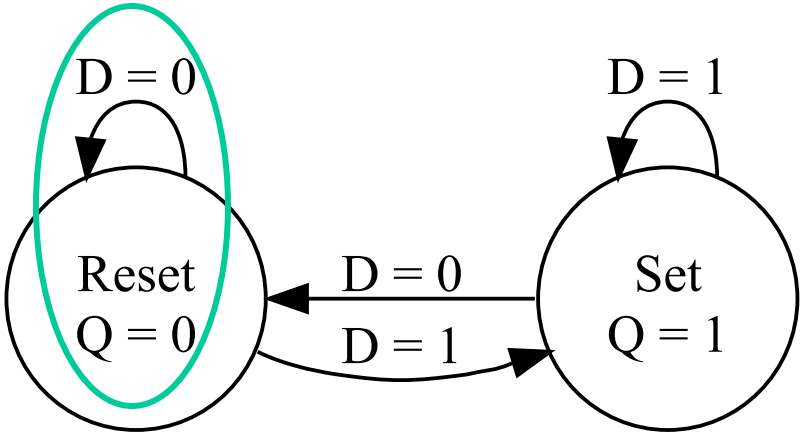
neergaande klokflank ↓



Truth table D-flipflop

D	klok	$Q_{current}$	Q_{next}	Functie
0	↑	0	0	Load 0 Reset
0	↑	1	0	Load 0 Reset
1	↑	0	1	Load 1 Set
1	↑	1	1	Load 1 Set

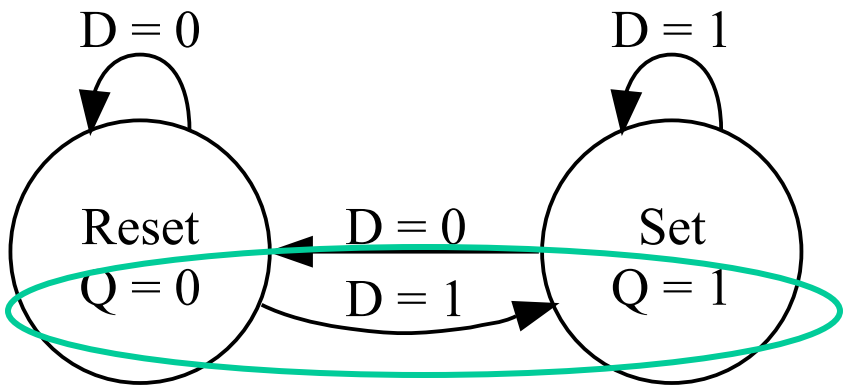
State diagram & Truth table D-flipflop



D_n	klok	Q_n	Q_{n+1}	Functie
0	↑	0	0	Load 0 Reset
0	↑	1	0	Load 0 Reset
1	↑	0	1	Load 1 Set
1	↑	1	1	Load 1 Set

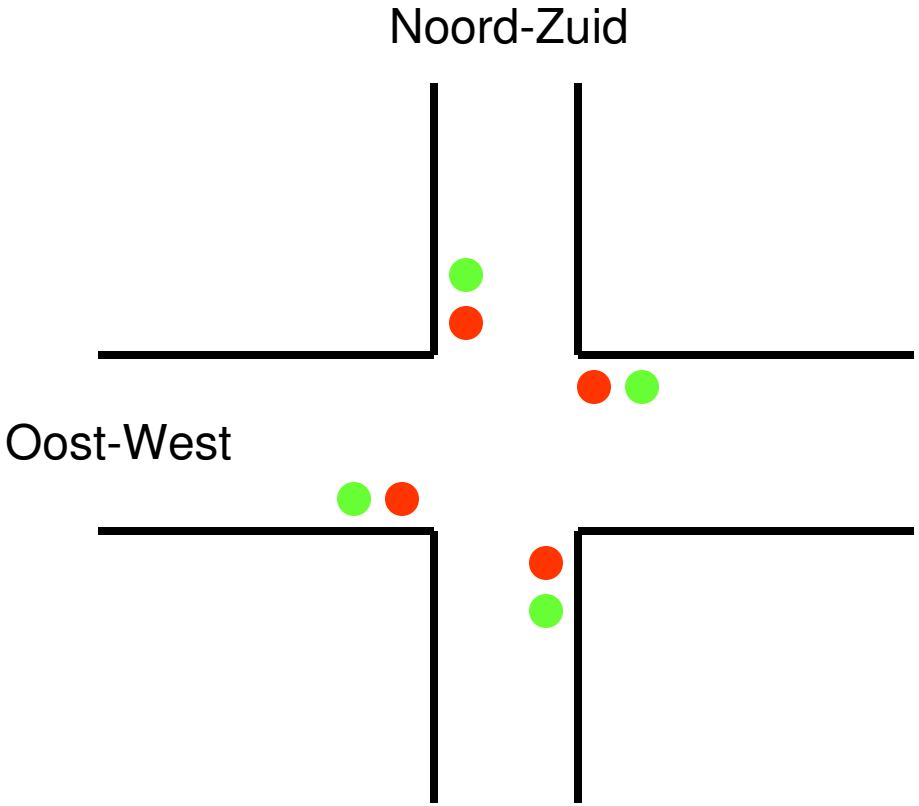
Tabel1

State diagram D-flipflop



D_n	klok	Q_n	Q_{n+1}	Functie
0	↑	0	0	Load 0 Reset
0	↑	1	0	Load 0 Reset
1	↑	0	1	Load 1 Set
1	↑	1	1	Load 1 Set

Voorbeeld: Verkeerslicht



Stoplichten zijn gekoppeld
Twee toestanden:

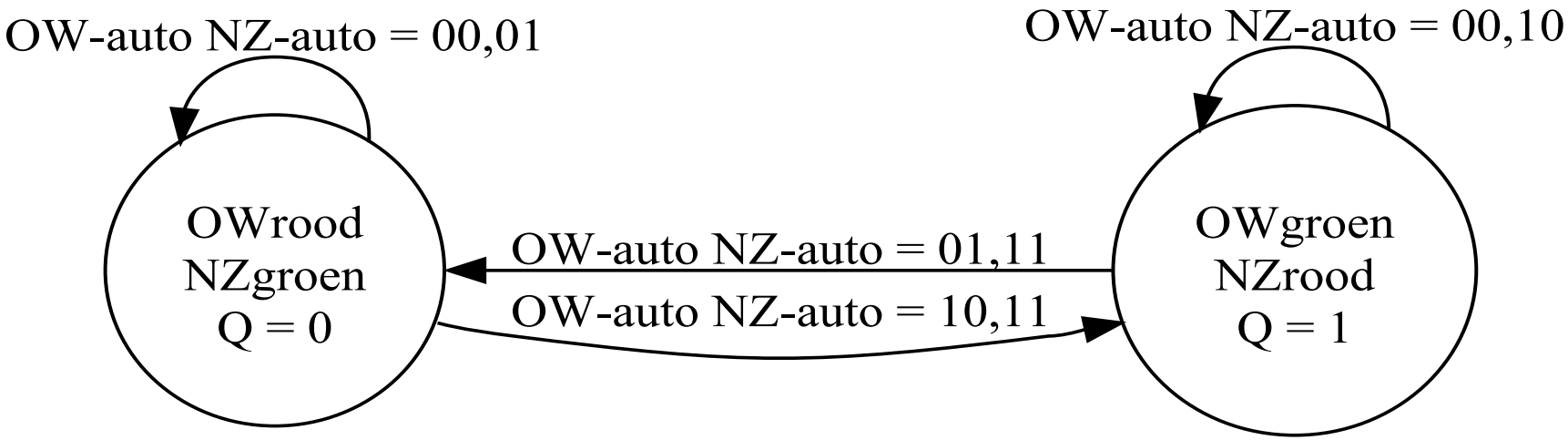
1. OW rood & NZ groen
2. OW groen & NZ rood

Verkeerslicht

NZ_{auto}	OW_{auto}	$Q_{current}$	Q_{next}	Toestand licht oost-west route 1 = groen
0	0	0	0	Geen auto, licht blijft rood
0	0	1	1	Geen auto, licht blijft groen
0	1	0	1	OW_{auto}, licht wordt groen
0	1	1	1	OW_{auto}, licht blijft groen
1	0	0	0	NZ_{auto}, licht blijft rood
1	0	1	0	NZ_{auto}, licht wordt rood
1	1	0	1	2 auto's, licht wordt groen
1	1	1	0	2 auto's, licht wordt rood

Next State function voor een stoplicht

State Diagram



Van State diagram → tabel

NZ_{auto}	OW_{auto}	$Q_{current}$	Q_{next}	Toestand licht oost-west route 1 = groen
0	0	0	0	Geen auto, licht blijft rood
0	0	1	1	Geen auto, licht blijft groen
0	1	0		OW_{auto}, licht wordt groen
0	1	1		OW_{auto}, licht blijft groen
1	0	0		NZ_{auto}, licht blijft rood
1	0	1		NZ_{auto}, licht wordt rood
1	1	0		2 auto's, licht wordt groen
1	1	1		2 auto's, licht wordt rood
Next State function voor een stoplicht				

Tabel→ som van mintermen

NZ_{auto}	OW_{auto}	$Q_{current}$	Q_{next}	Toestand licht oost-west route 1 = groen
0	0	0	0	Geen auto, licht blijft rood
0	0	1	1	Geen auto, licht blijft groen
0	1	0	1	OW_{auto}, licht wordt groen
0	1	1	1	OW_{auto}, licht blijft groen
1	0	0	0	NZ_{auto}, licht blijft rood
1	0	1	0	NZ_{auto}, licht wordt rood
1	1	0	1	2 auto's, licht wordt groen
1	1	1	0	2 auto's, licht wordt rood
Next State function voor een stoplicht				

$$Q_{next} = \overline{NZ}.\overline{OW}.Q + \quad + \quad +$$

Som van mintermen

$$Q_{next} = f(NZ, OW, Q_{current})$$

$$Q_{next} = \overline{NZ}.\overline{OW}.Q + \overline{NZ}.OW.\overline{Q} + \overline{NZ}.OW.Q + NZ.OW.\overline{Q}.$$

Vereenvoudigen van deze som

$$Q_{next} = \overline{NZ}.\overline{OW}.Q + \overline{NZ}.OW.\overline{Q} + \overline{NZ}.OW.Q + NZ.OW.\overline{Q}.$$

$$\overline{NZ}.\overline{OW}.Q + \overline{NZ}.OW.Q = \overline{NZ}.Q.(\overline{OW} + OW) = \overline{NZ}.Q.1 = \overline{NZ}.Q.$$

Distributieve wet

A OR NOT(A) = 1

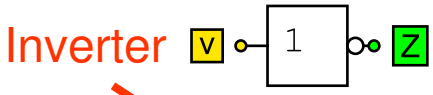
Q AND 1 = Q

Vereenvoudigen van deze som

$$Q_{next} = \overline{NZ}.\overline{OW}.Q + \overline{NZ}.OW.\overline{Q} + \overline{NZ}.OW.Q + NZ.OW.\overline{Q}.$$

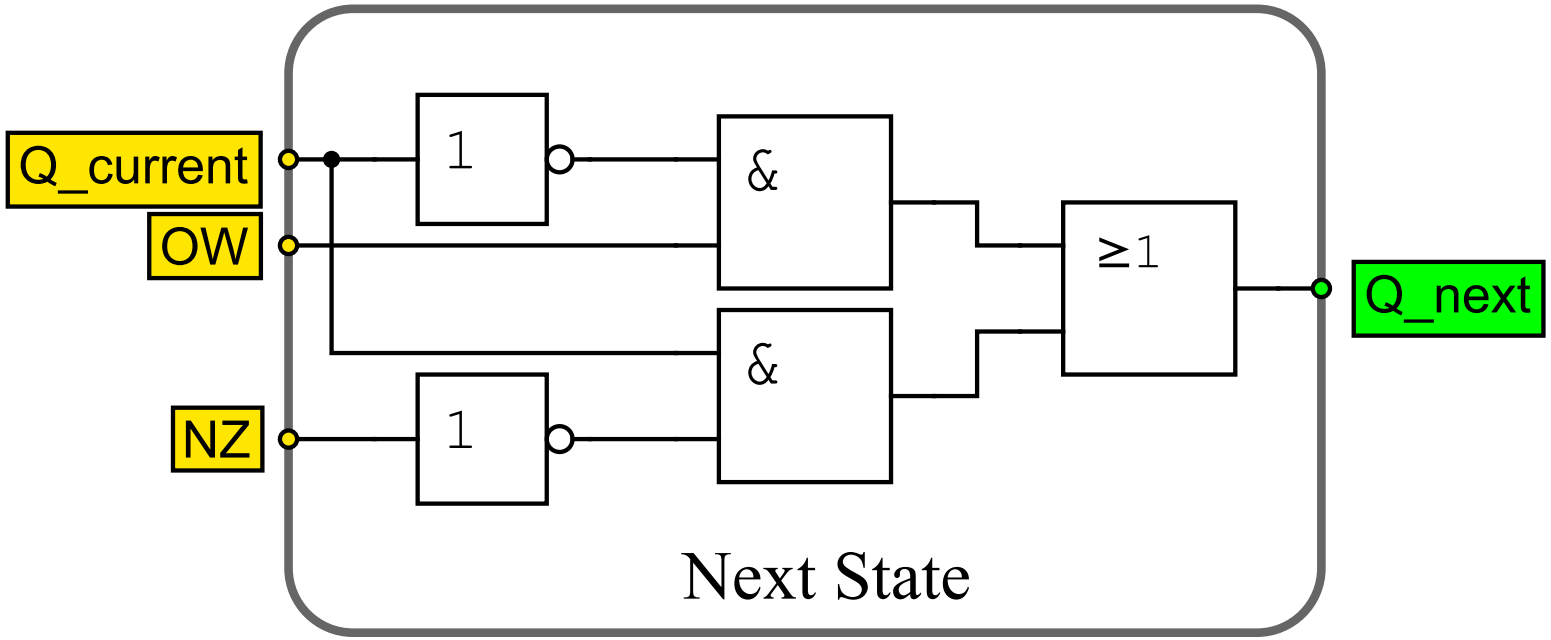
$$Q_{next} = \overline{NZ}.Q + OW.\overline{Q}.$$

Van expressie naar schakeling



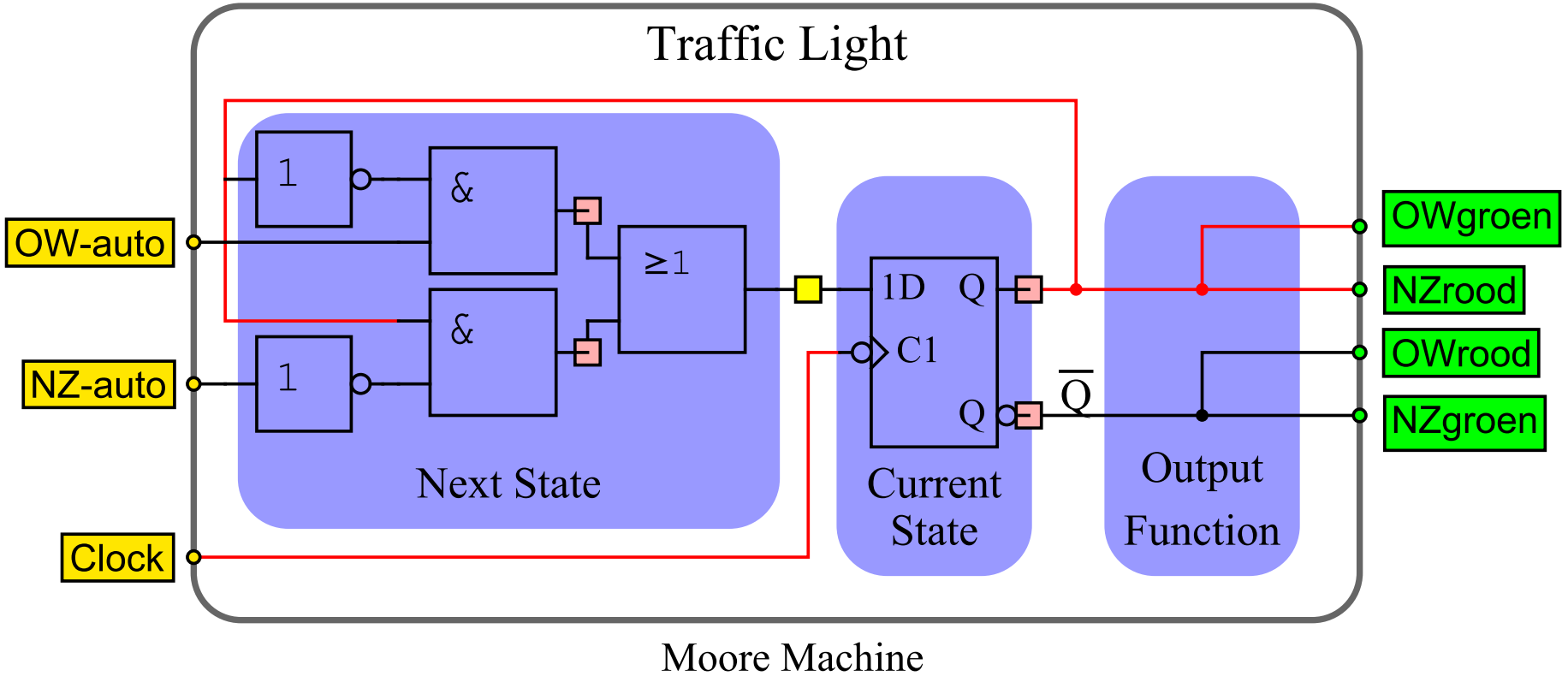
$$Q_{next} = (\overline{NZ}.Q) + (OW.\overline{Q}).$$

Implementatie "Next State"met poorten



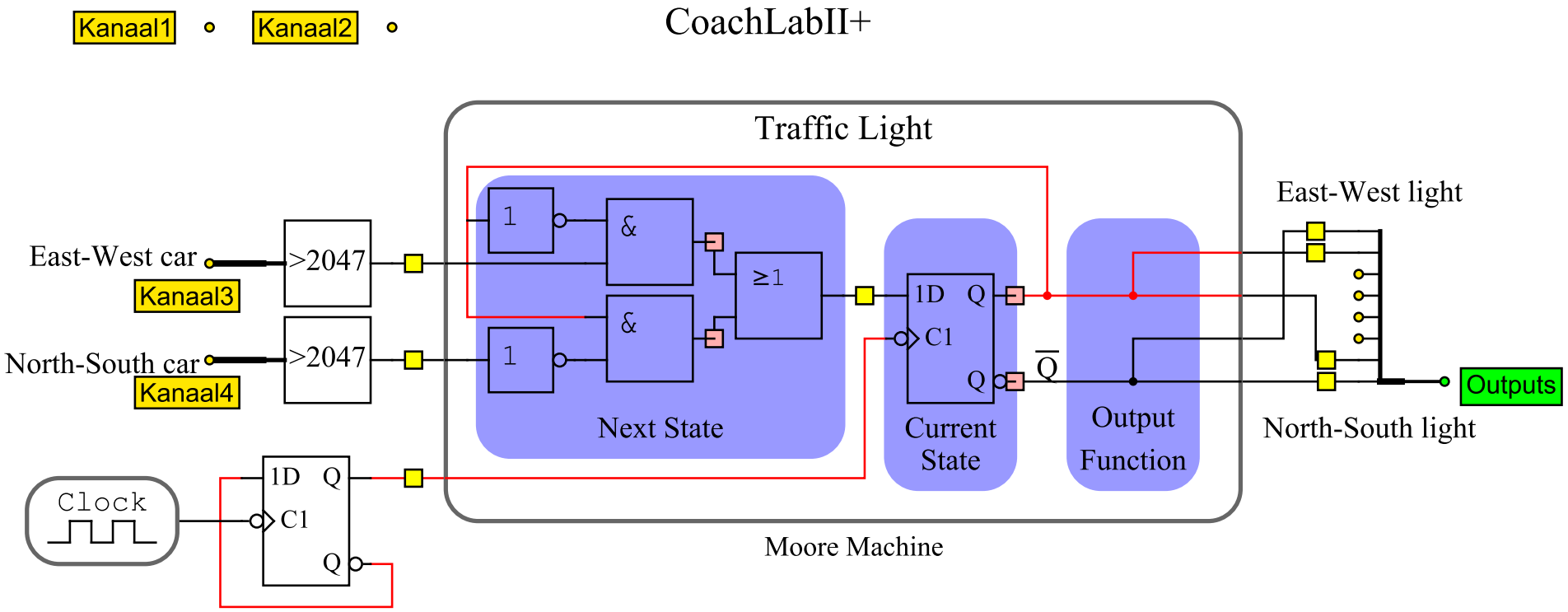
$$Q_{next} = \overline{NZ}.Q + OW.\overline{Q}.$$

Eindige toestandsautomaat

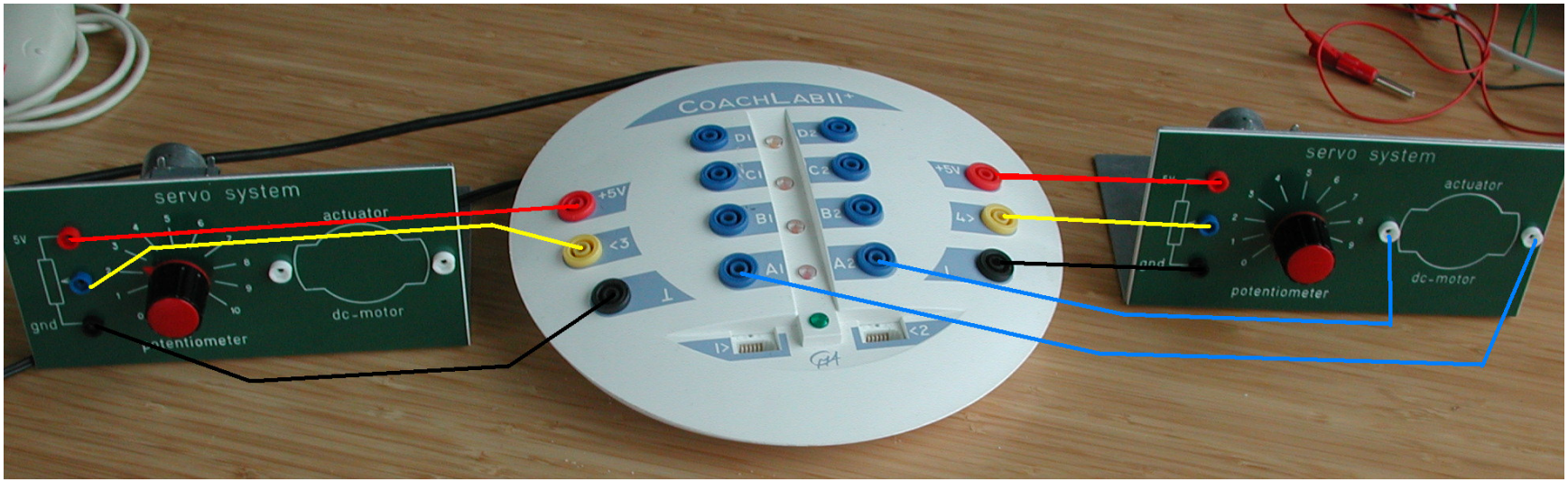


Moore Machine

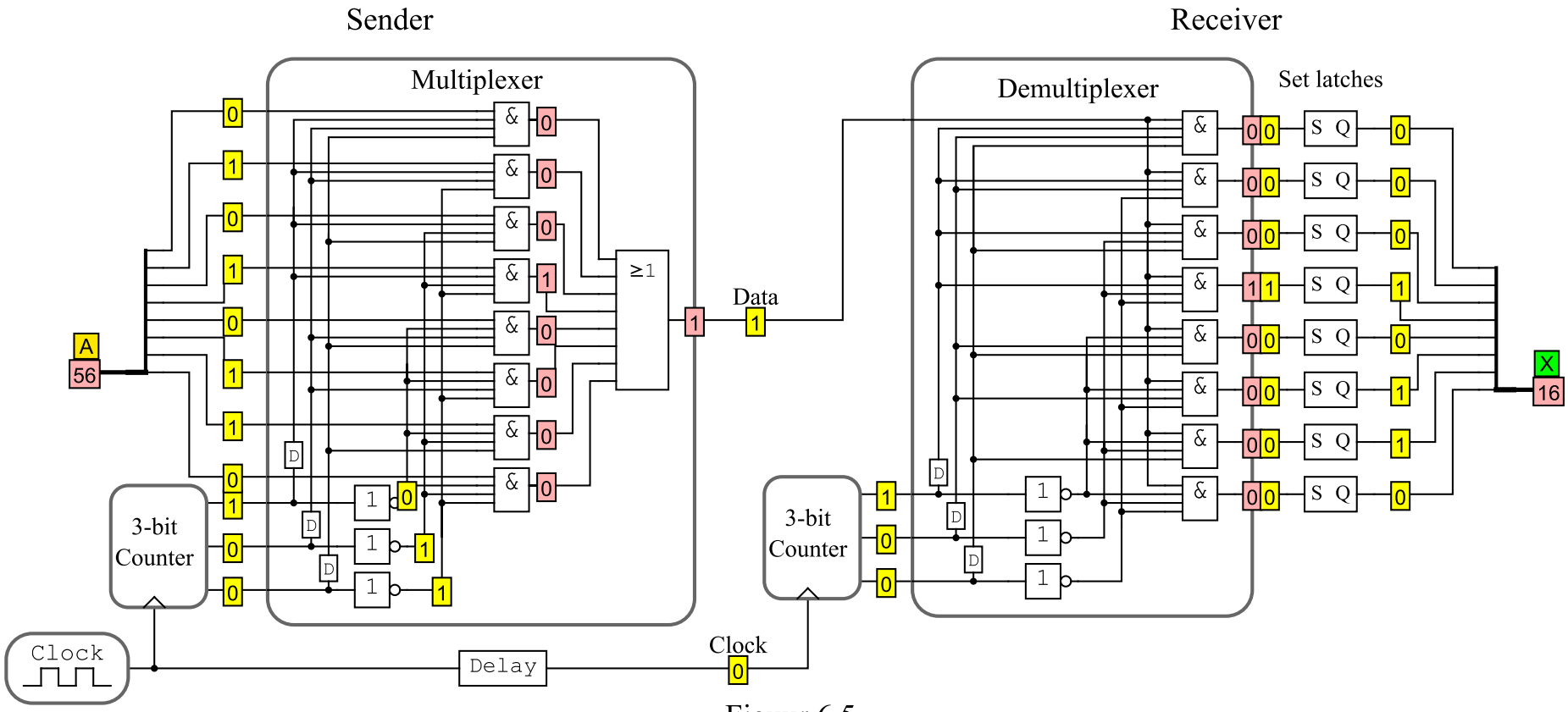
Eindige toestandsautomaat
Sensoren en actuatoren via CoachLbII+



Tweede voorbeeld: Steering a servo-system



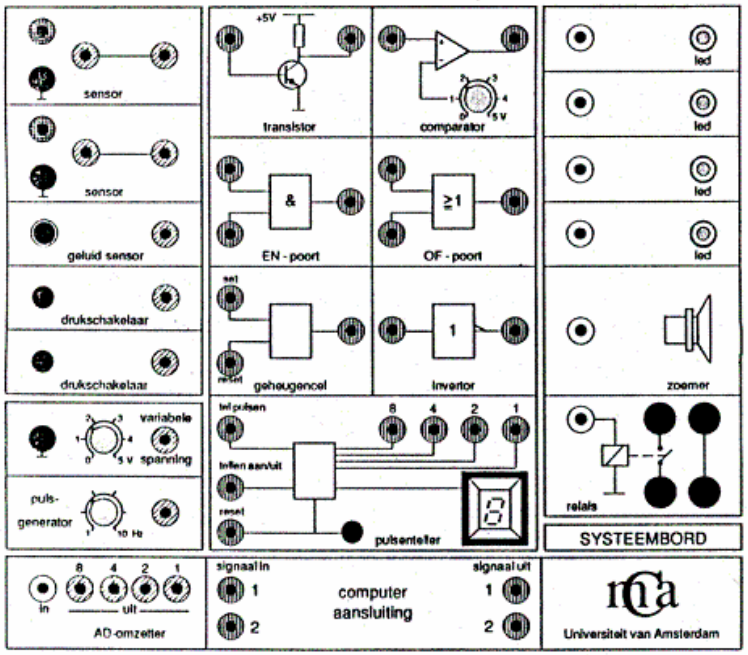
Synchrone seriële datatransmissie



Figuur 6.5

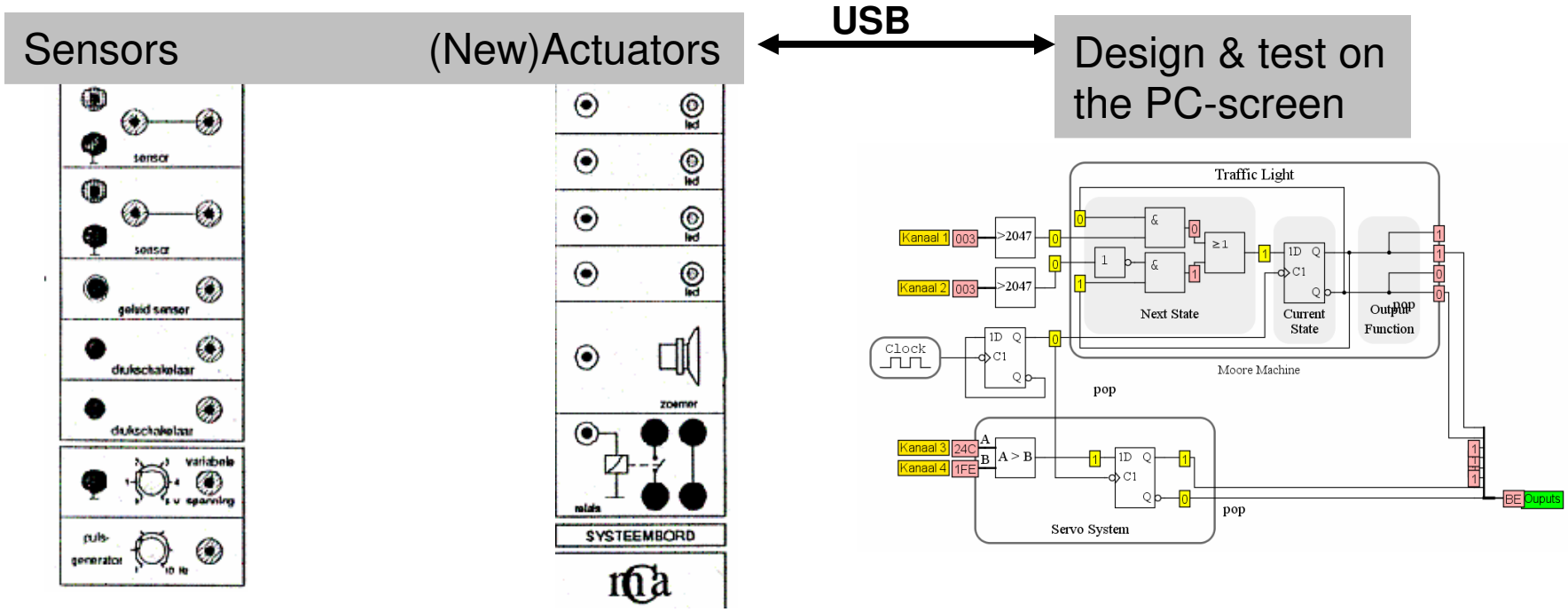
Gewenste ontwikkeling: Nieuw systeembord

Sensors Processing Actuators



A new System Board

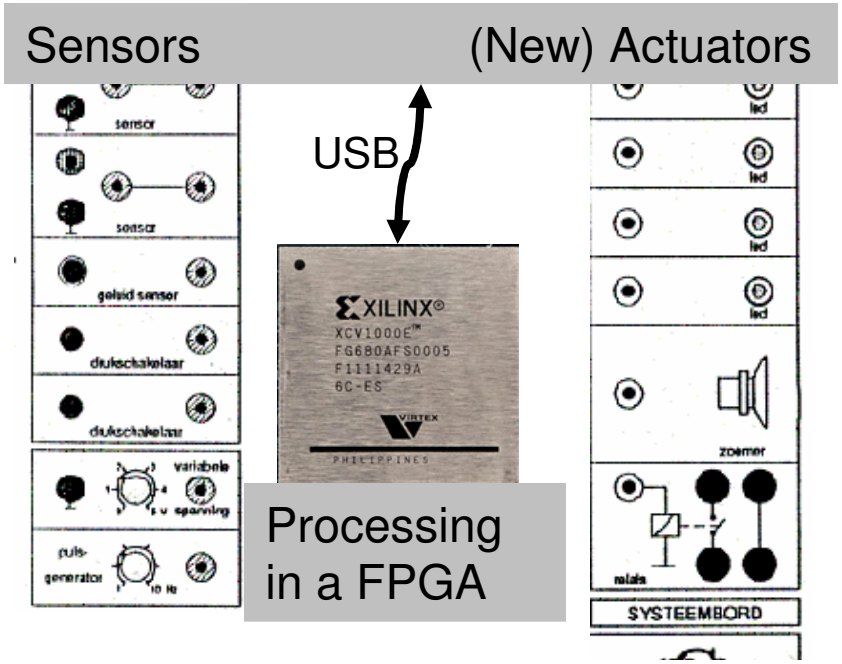
Design & test on the PC-screen



System Board II

A new System Board

Compile & Load to a FPGA



System Board II

SIM-PL is gemaakt door Wouter Koolen-Wijkstra

Wouter heeft vorige week zijn Master of Logic behaald en promoveert
In februari 2011 bij het Centrum voor Wiskunde en Informatica.

Website:www.science.uva.nl/amstel/SIM-PL/

- Software
- Componenten
- Tutorial
- Onderwijsmateriaal

Mail to: .Bruidegom@uva.nl

Vragen?