DIGTEC/2021-2022

Jesse op den Brouw

DIGTEC

Toestandsmachines

DE HAAGSE HOGESCHOOL

Toestandsmachines

- Logische circuits kunnen verdeeld worden in twee groepen:

- combinatorische logica
  - uitgangen zijn alleen afhankelijk van de huidige ingangswaarden.

- sequentiële logica
  - uitgangen zijn afhankelijk van de huidige ingangswaarden en van een waarden van flipflops.

Toestandsmachines

- Er zijn twee typen sequentiële schakelingen:

- Synchroon
  - Een kloksignaal dient als referentie voor alle acties binnen het systeem.

- Asynchroon
  - Er is geen kloksignaal aanwezig. Acties worden veroorzaakt door (eerdere) acties binnen het systeem.

- Synchrone systemen zijn eenvoudiger te ontwerpen dan asynchrone systemen.

Toestandsmachines

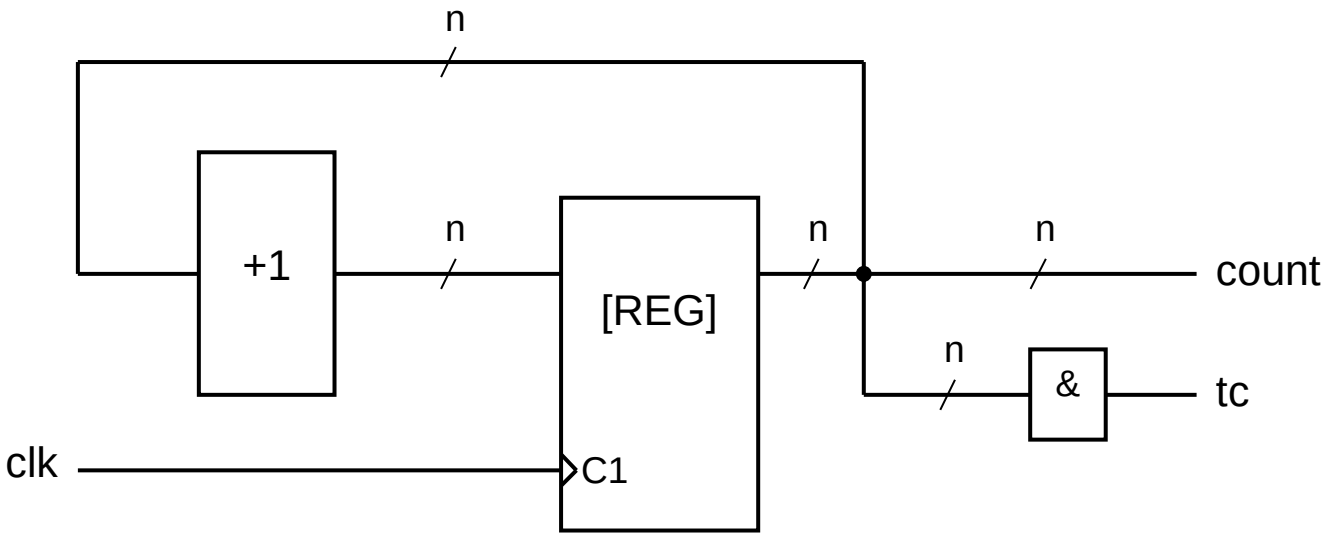
- Waarom kloksynchroon?

- Voordelen:
  - Eenvoudige timingverificatie.
  - Eenvoudige functionele verificatie.
  - Eenvoudige optimalisatie naar de kleinst mogelijke schakeling.
  - Eenvoudig ontwikkeltraject voor het ontwerpen en bouwen van een machine.

- Nadelen:
  - Hulpkloksignaal nodig.
  - Over het algemeen trager dan asynchrone schakelingen (kritieke pad).
  - Verbruikt meer energie dan asynchrone schakelingen

Toestandsmachines

- Een voorbeeld van een synchroon systeem is een teller:



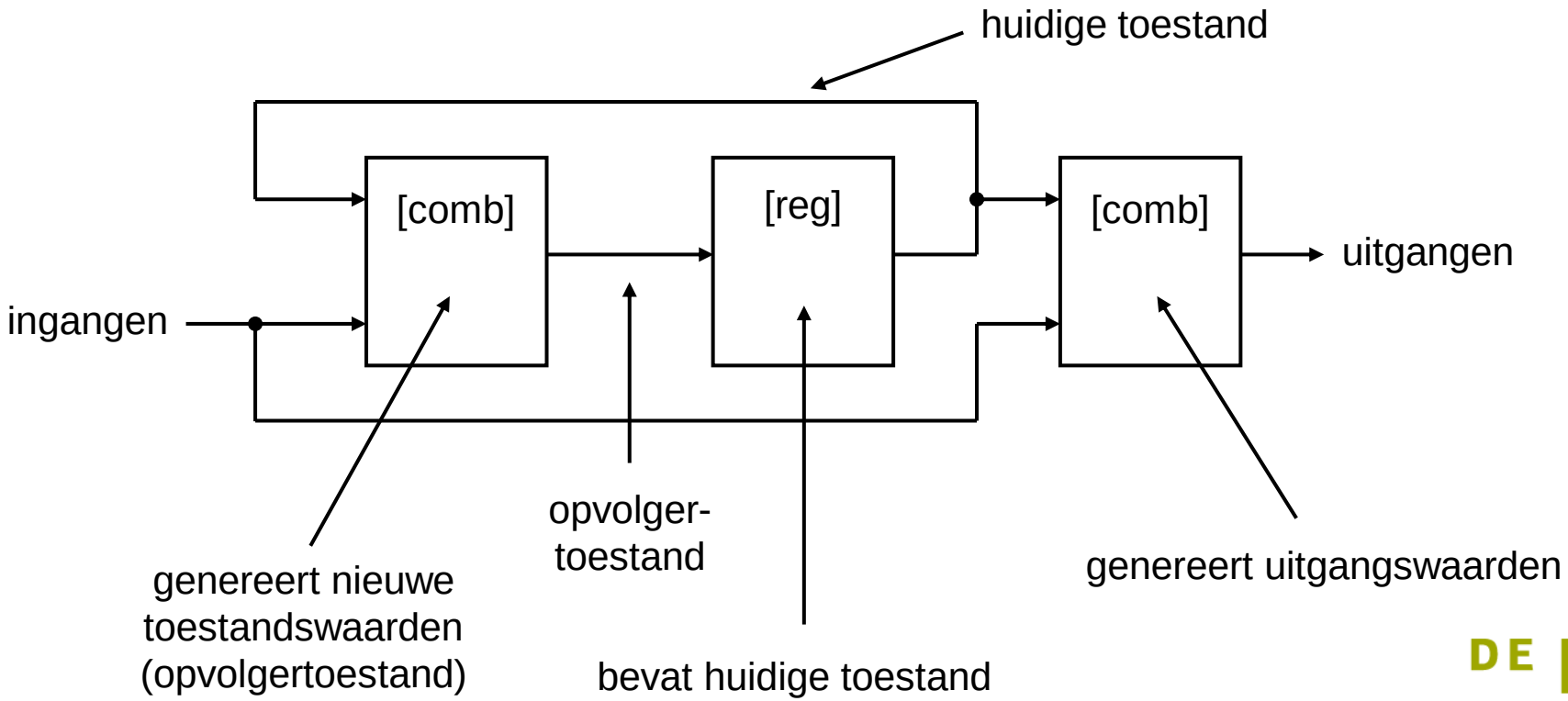
- Het register dient voor opslag van de (huidige) telstand.
- De opteller (+1) genereert de nieuwe telstand.
- tc geeft aan of de teller op de hoogste telstand staat.

Toestandsmachines

- Synchrone systemen worden gerealiseerd door middel van combinatorische logica en één of meer geheugenelementen.
- De geheugenelementen leggen de *huidige toestand* van het synchrone systeem vast.
- De ingangen, samen met de toestand, zorgen ervoor dat de toestand van het systeem verandert.
- De ingangen, samen met de toestand, zorgen ervoor dat de uitgangen een bepaalde waarde krijgen.

Toestandsmachines

- Hieronder de algemene structuur van een synchroon systeem.



DE HAAGSE HOGESCHOOL

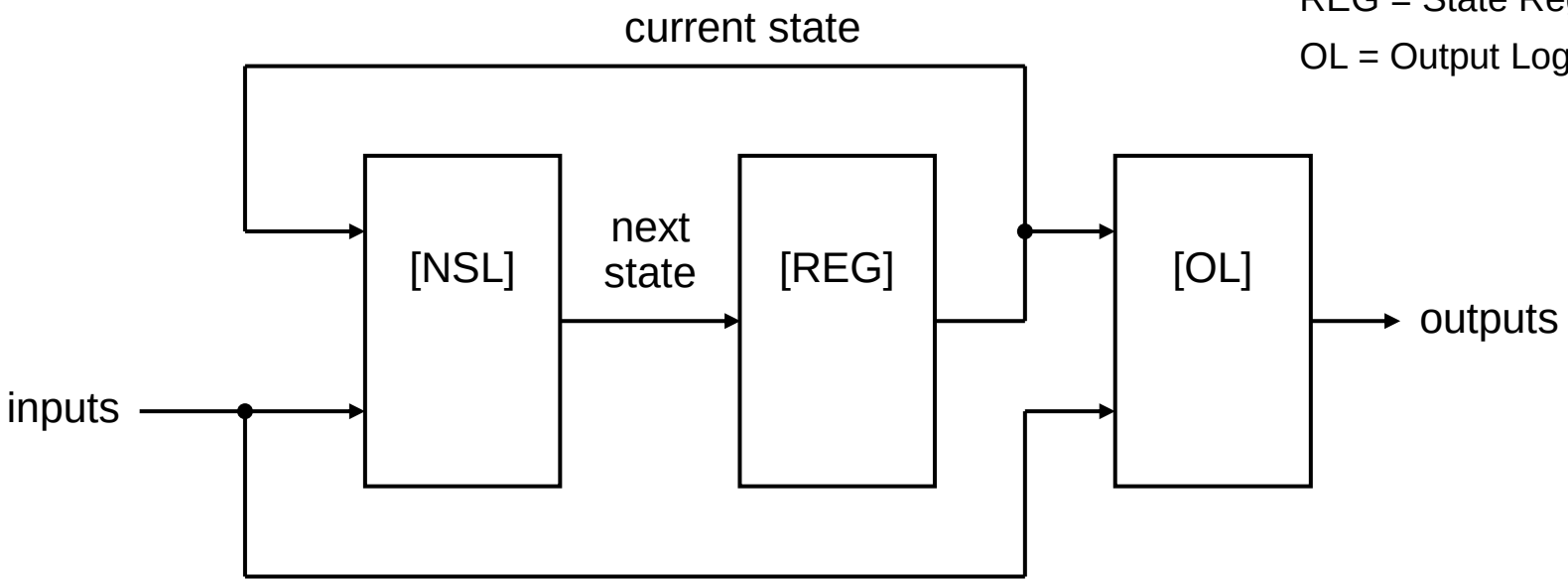
Toestandsmachines

- Algemeen bezitten synchrone systemen toestanden en worden ze *toestandsmachines* genoemd.
- Daarnaast is het aantal toestanden *eindig*.
- Praktische synchrone systemen zijn dus eindige toestandsmachines.
- Een veelgebruikte naam in technische literatuur is *Finite State Machine* (FSM). Een andere term is *automaat* (automaton).
- Er zijn twee typen: Mealy- en Moore-machines.

Toestandsmachines

- Hieronder de structuur van een Mealy-machine.

NSL = Next State Logic
REG = State Register
OL = Output Logic



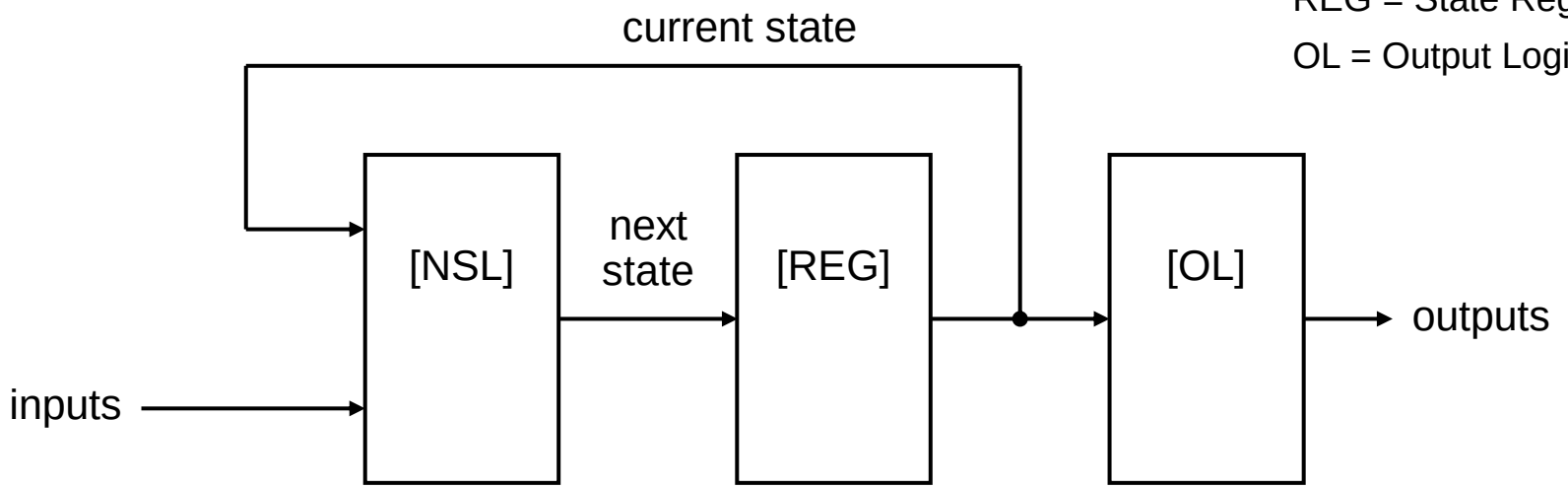
- Bij een Mealy-machine zijn de uitgangen afhankelijk van de toestand van de machine én de ingangen.

DE HAAGSE HOGESCHOOL

Toestandsmachines

- Hieronder de structuur van een Moore-machine.

NSL = Next State Logic
REG = State Register
OL = Output Logic



- Bij een Moore-machine zijn de uitgangen alleen afhankelijk van de toestand van de machine. De Moore-machine is een vereenvoudiging van de Mealy-machine.

DE HAAGSE HOGESCHOOL

Toestandsmachines

- Om tot een toestandsmachine te komen, moeten de volgende stappen genomen te worden:

  Opstellen van het toestandsdiagram

  Opstellen van de toestandstabel

  Kiezen van toestandscodering

  Opstellen van de opvolgertoestand- en uitgangstabellen

  Functies voor opvolgertoestand en uitgangen opstellen

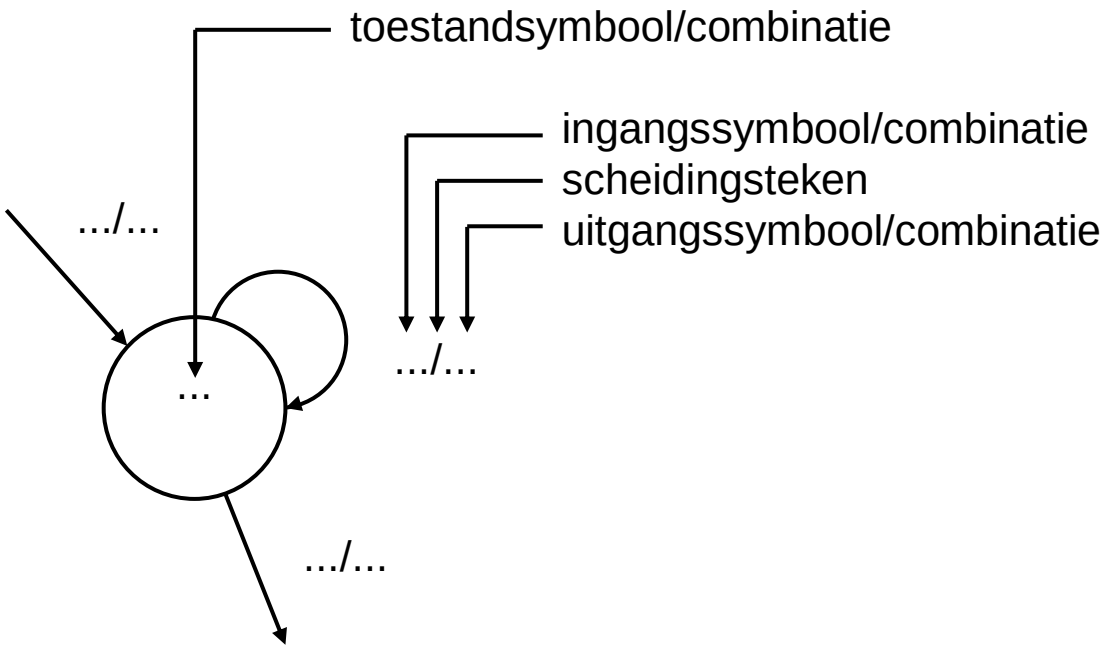
  Realiseren van de schakeling met logica

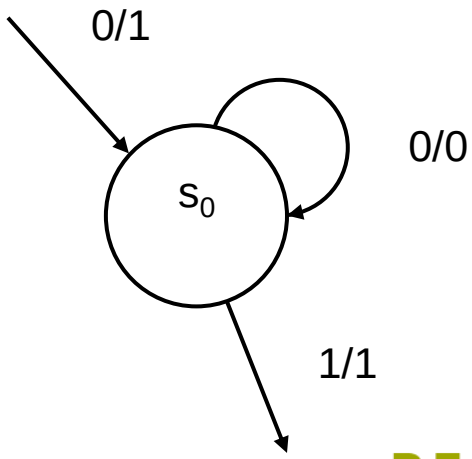
Toestandsmachines

- De werking van een machine kan eenvoudig weergegeven worden met een toestandsdiagram.
- Het geeft aan wat de machine onder welke conditie gaat (moet gaan) doen.
- Een toestandsdiagram bestaat uit een aantal cirkels of bollen die de *toestanden* weergeven en pijlen die de overgangen en de *overgangscondities* (*transities*) van de ene naar de andere toestand aangeven.
- Er zijn twee beschrijvingswijzen, één voor Mealy-machines en één voor Moore-machines.

Toestandsmachines

- Bij Mealy worden bij de overgangscondities de uitgangswaarden naast de ingangswaarden geschreven, de toestand in de cirkel.

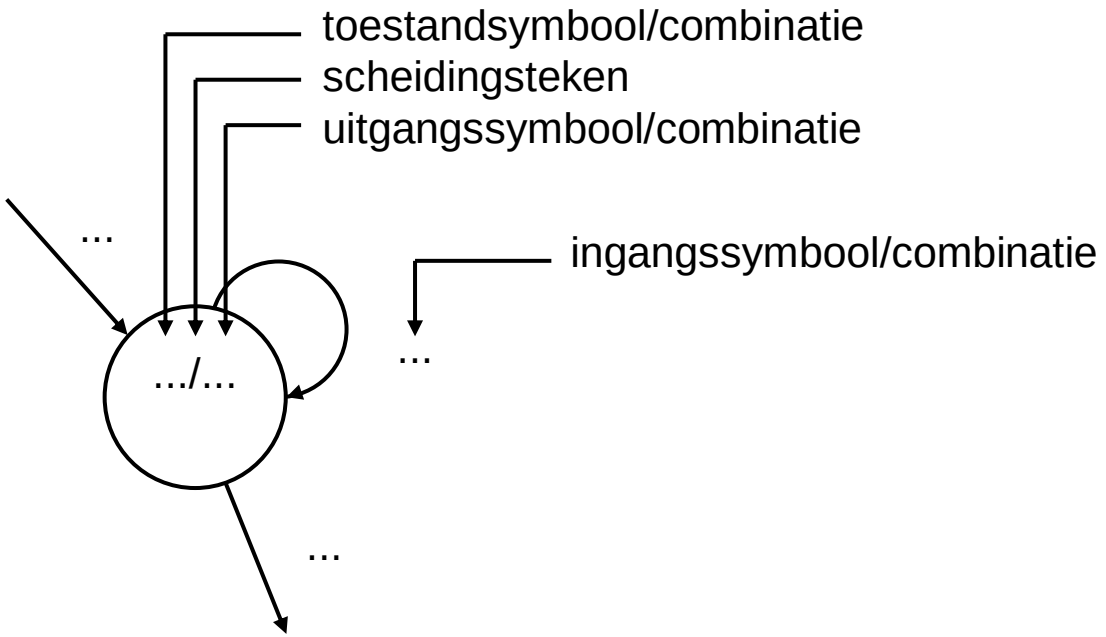


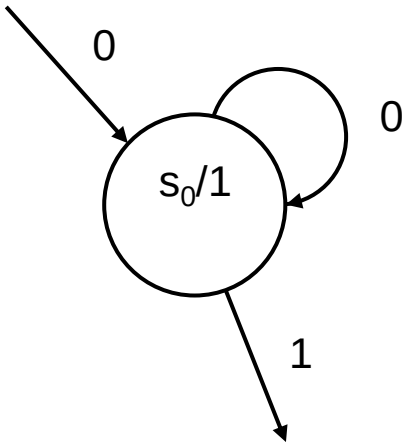


DE HAAGSE HOGESCHOOL

Toestandsmachines

- Bij Moore worden de uitgangswaarden naast de toestand geschreven, de ingangswaarden bij de overgangscondities.





DE HAAGSE HOGESCHOOL

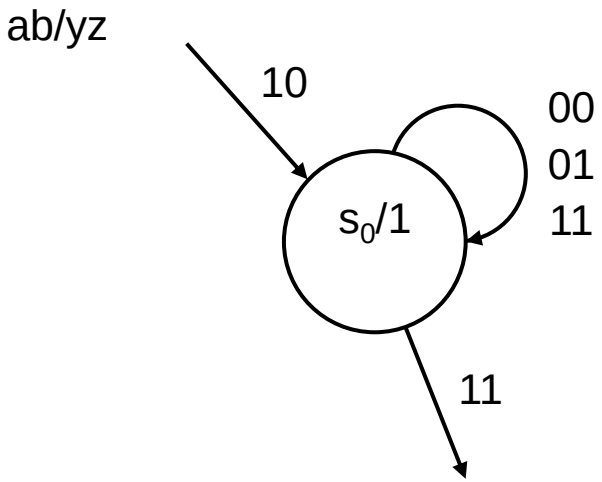
Toestandsmachines

- Het ontwerpen een van toestandsdiagram (en dus de machine) begint bij het bepalen van het aantal toestanden dat nodig is en welke overgangsvoorwaarden nodig zijn.
- Dit volgt uit een (meestal) geschreven beschrijving.
- Hiervoor is geen vaste procedure. De ontwerper moet zelf bedenken wat de machine moet doen op basis van de beschrijving en goed nadenken over eventuele situaties die niet beschreven zijn.
- Er is altijd een begintoestand waar de machine in terecht komt na een reset-opdracht.

Toestandsmachines

- Het ontwikkelde toestandsdiagram moet *volledig* en *niet-tegenstrijdig* zijn.

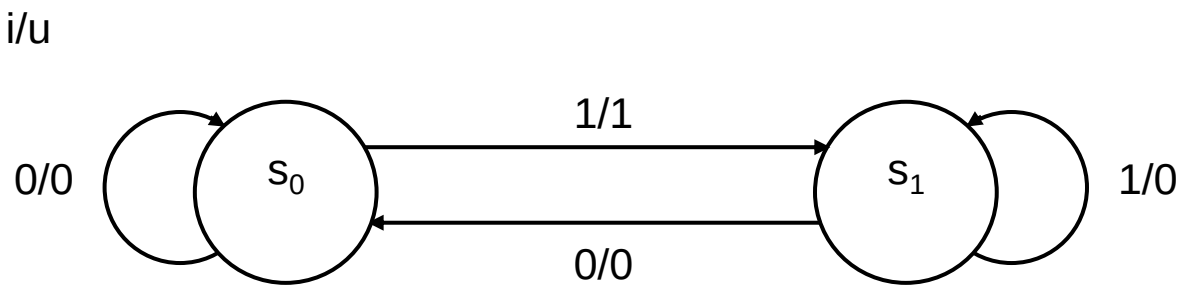
- Voor elke ingangscombinatie moet een overgang beschreven zijn, voor elke uitgang moet een waarde gedefinieerd worden (don't cares zijn mogelijk).



wat moet er gebeuren als
ab = 10 of ab = 11?
wat is de waarde van z?

Toestandsmachines

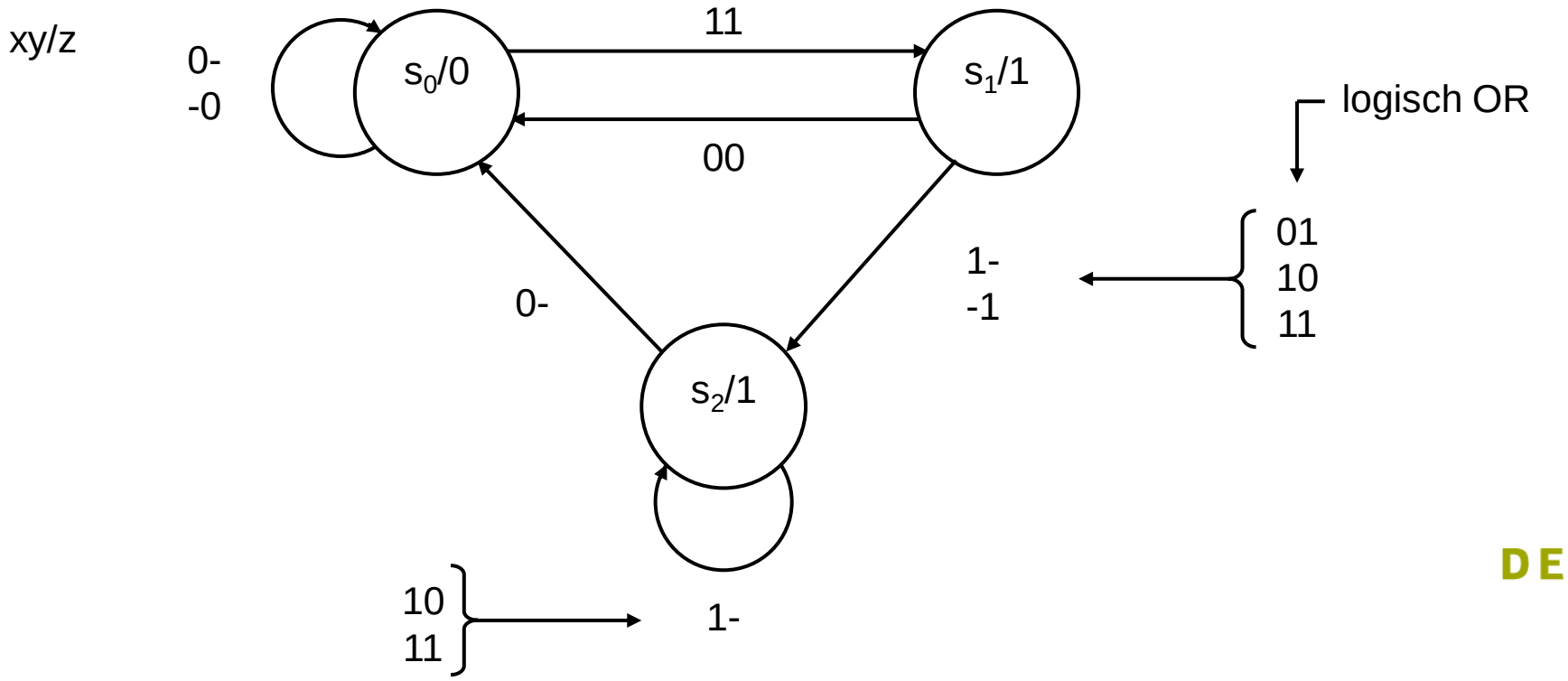
- Als voorbeelden volgen nu een Mealy en een Moore-machine. Deze machines worden gebruikt tot en met de synthese.
- De Mealy-machine heeft één ingang, één uitgang en twee toestanden.



DE HAAGSE HOGESCHOOL

Toestandsmachines

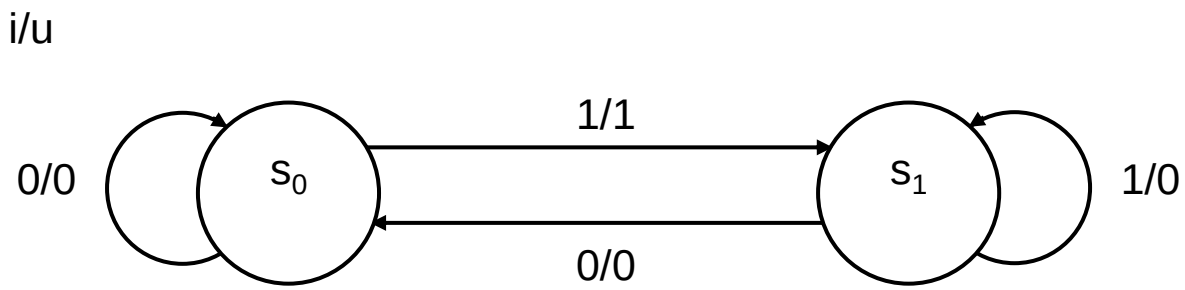
- De Moore-machine heeft twee ingangen, één uitgang en 3 toestanden. Merk op dat bij sommige overgangscondities don't cares gegeven zijn.



DE HAAGSE HOGESCHOOL

Toestandsmachines

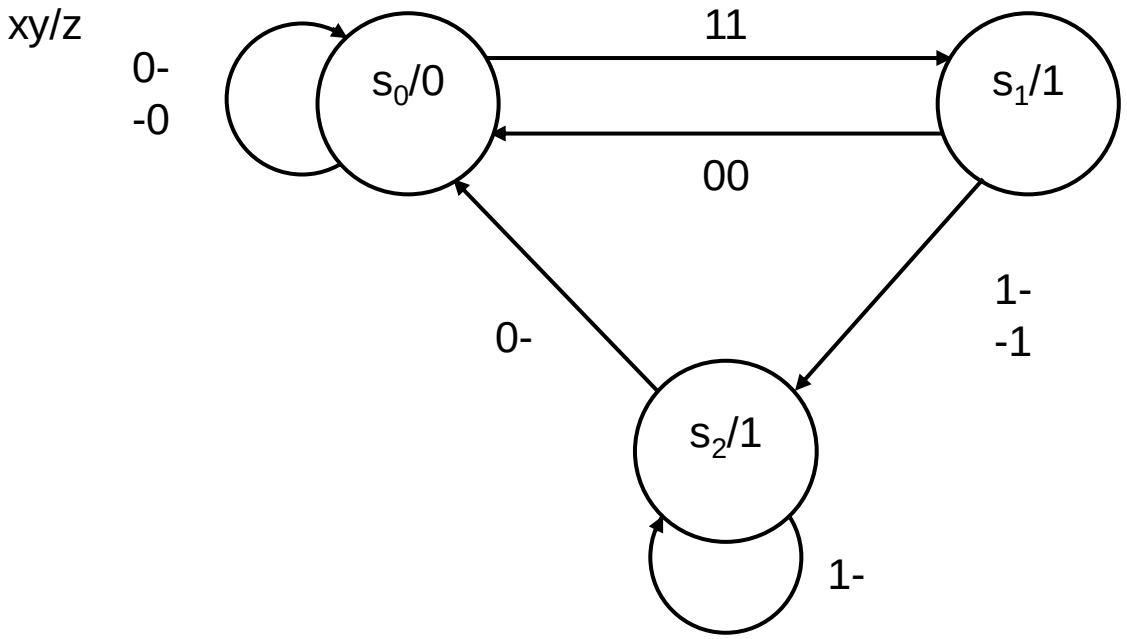
- De toestandstabel van de Mealy-machine. Wordt veel gebruikt in literatuur.



toestand	opvolger		uitgang	
	0	1	0	1
s_0	s_0	s_1	0	1
s_1	s_0	s_1	0	0

DE HAAGSE HOGESCHOOL

Toestandsmachines



toestand	opvolger				uitgang
	00	01	10	11	
s_0	s_0	s_0	s_0	s_1	0
s_1	s_0	s_2	s_2	s_2	1
s_2	s_0	s_0	s_2	s_2	1

Toestandsmachines

- De volgende stap in het ontwerpproces is het toekennen van codecombinaties aan de toestanden.
- De toestanden moeten uniek gecodeerd worden dus bij n toestanden zijn tenminste $^{2}\log n$ *toestandsbits* nodig.
- Voor elke toestandsbit is een flipflop nodig.
- De keuze van de codering is vrij te kiezen zolang deze voor elke toestand uniek is.
- Er kunnen meer flipflops gebruikt worden dan strikt noodzakelijk zijn.

Toestandsmachines

- Meest gebruikte coderingen zijn:
  - (zuiver) binaire telcode - voor tellers
  - Gray code - voor tellers (extra output logic)
  - One Hot - snelle systemen / systemen met veel flipflops (FPGA)

- Binaire telcode en Gray code gebruiken de minst aantal flipflops.

- One Hot code gebruikt voor elke toestand één flipflop.
  - De achterliggende gedachte is dat de omvang van de producttermen in de next state logic afneemt en hierdoor de maximale propagatietijd van de combinatoriek kleiner wordt.
  - Voordeel: veel don’t cares, dus minimale logica.
  - Nadeel: veel niet-gebruikte codecombinaties.

Toestandsmachines

- Het aantal verschillende toestandscoderingen wordt gegeven door

$$N_D = \frac{2^k!}{(2^k - n)!k!}$$

N_D: het aantal verschillende coderingen
n: het aantal toestanden
k: het aantal toestandsbits

- Als voorbeeld: voor een machine met drie toestandbits en vijf toestanden zijn 1120 verschillende coderingen mogelijk.

- Om de meest optimale codering te vinden, moeten alle combinaties onderzocht worden.

Toestandsmachines

- Nadat de toestandscodering bepaald is, worden de functies voor de opvolgertoestand (*next state equations*) en de uitgangen (*output equations*) afgeleid.

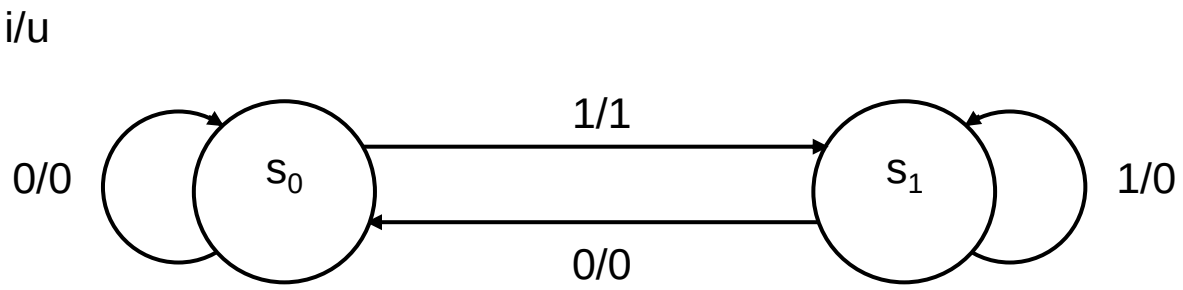
- Uit de toestandstabel wordt een waarheidstabel opgesteld (met van links naar rechts):

  - de toestandscoderingen gecombineerd met de ingangscombinaties
  - de opvolgertoestanden
  - de instelsignalen voor de gebruikte flipflops (D, JK, SR, T)
  - de uitgangscombinaties

- Hier wordt alleen het gebruik van D-flipflops toegelicht.

Toestandsmachines

- Er zijn slechts twee toestanden dus één flipflop is genoeg om unieke codes toe te kennen. Toestand s_0 wordt gecodeerd met 0 en s_1 met 1.



Q^n	i^n	Q^{n+1}	D^n	u^n
0	0	0	0	0
0	1	1	1	1
1	0	0	0	0
1	1	1	1	0

Toestandsmachines

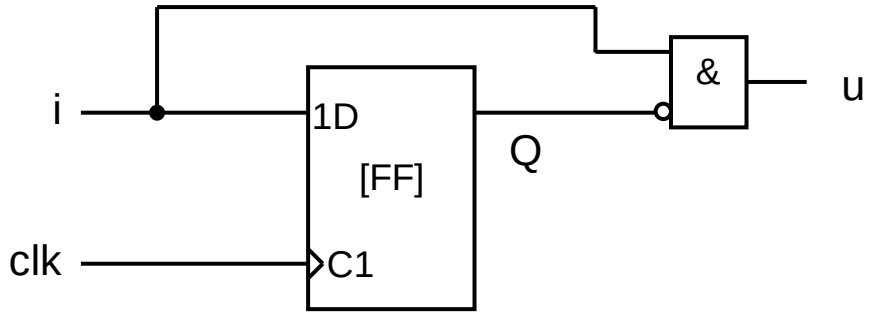
- Uit de waarheidstabel volgt direct de functie voor de toestanden:

$$D^n = i^n$$

- Ook is direct de functie voor de uitgang op te schrijven:

$$u^n = \overline{Q^n} \cdot i^n = [\overline{Q} \cdot i]^n$$

- Het schema is:



DE HAAGSE HOGESCHOOL

Toestandsmachines

- De Moore-machine heeft drie toestanden. Er zijn twee flipflops nodig om de toestanden te coderen. Een voor de hand liggende code is:

  $s_0 = 00$
  $s_1 = 01$
  $s_2 = 10$

- De combinatie 11 wordt niet gebruikt. *De machine heeft ongebruikte toestanden*.

- Aangezien de machine twee ingangen en twee toestandsbits heeft, zal de waarheidstabel uit 16 regels bestaan.

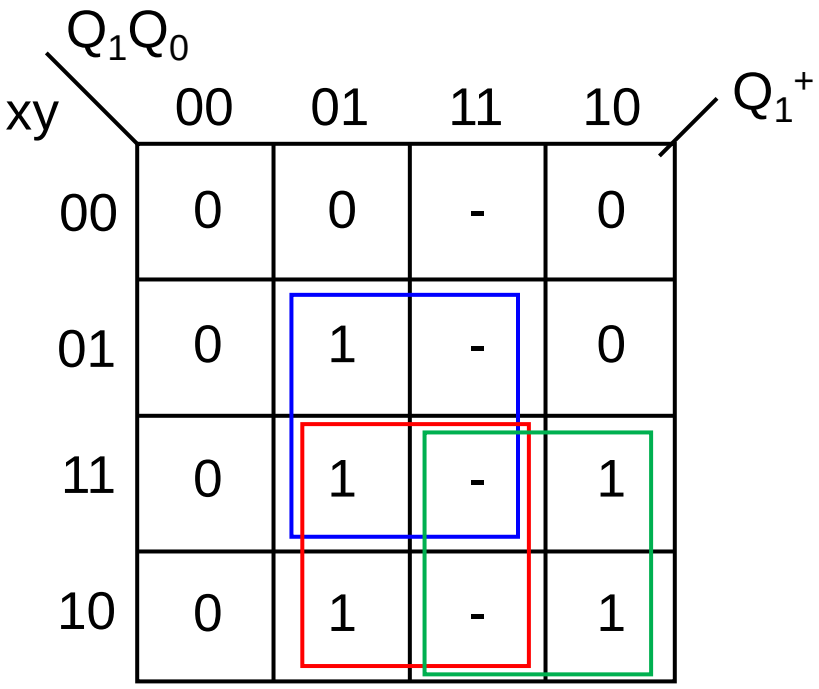
Toestandsmachines

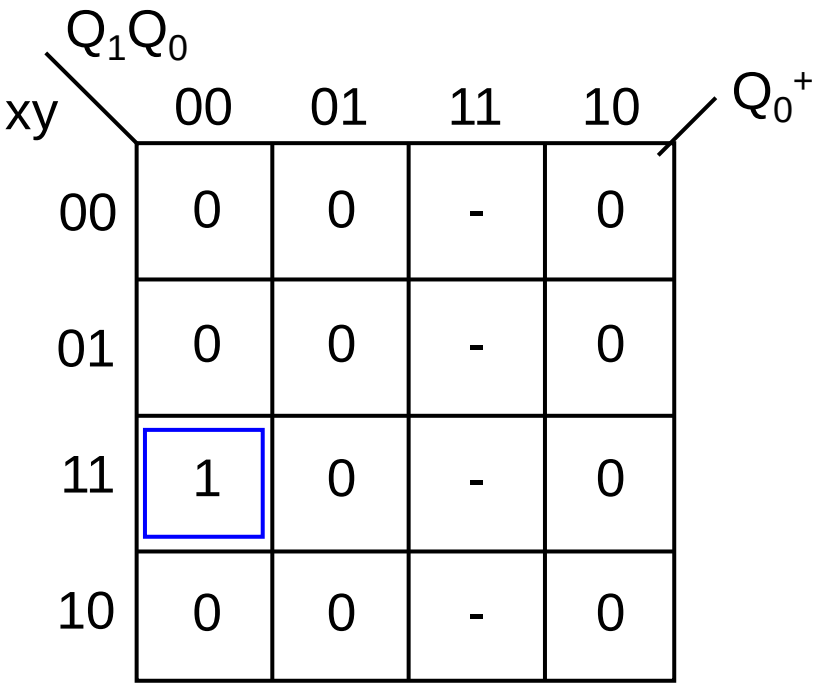
	Q_1^n	Q_0^n	x^n	y^n	Q_1^{n+1}	Q_0^{n+1}	D_1^n	D_0^n	z^n
s_0	0	0	0	0	0	0	0	0	0
	0	0	0	1	0	0	0	0	0
	0	0	1	0	0	0	0	0	0
	0	0	1	1	0	1	0	1	0
s_1	0	1	0	0	0	0	0	0	1
	0	1	0	1	1	0	1	0	1
	0	1	1	0	1	0	1	0	1
	0	1	1	1	1	0	1	0	1
s_2	1	0	0	0	0	0	0	0	1
	1	0	0	1	0	0	0	0	1
	1	0	1	0	1	0	1	0	1
	1	0	1	1	1	0	1	0	1
	1	1	0	0	-	-	-	-	-
	1	1	0	1	-	-	-	-	-
	1	1	1	0	-	-	-	-	-
	1	1	1	1	-	-	-	-	-

niet gebruikt

Toestandsmachines

- De Karnaughdiagrammen:





DE HAAGSE HOGESCHOOL

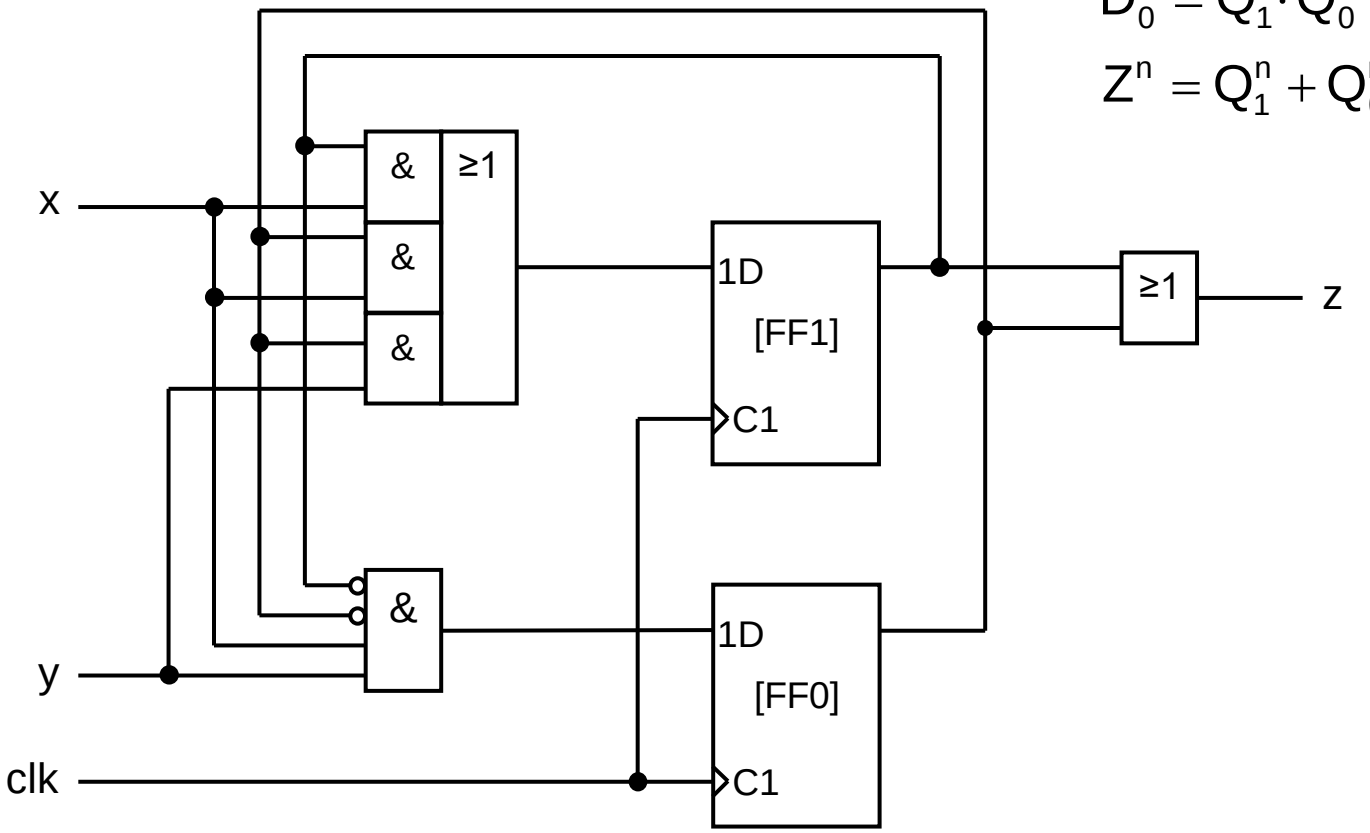
Toestandsmachines

- De volgende functies worden gevonden:

$$D_1^n = Q_1^n \cdot x^n + Q_0^n \cdot y^n + Q_0^n \cdot x^n$$

$$D_0^n = \overline{Q_1^n} \cdot \overline{Q_0^n} \cdot x^n \cdot y^n$$

$$Z^n = Q_1^n + Q_0^n$$



DE HAAGSE HOGESCHOOL

Toestandsmachines

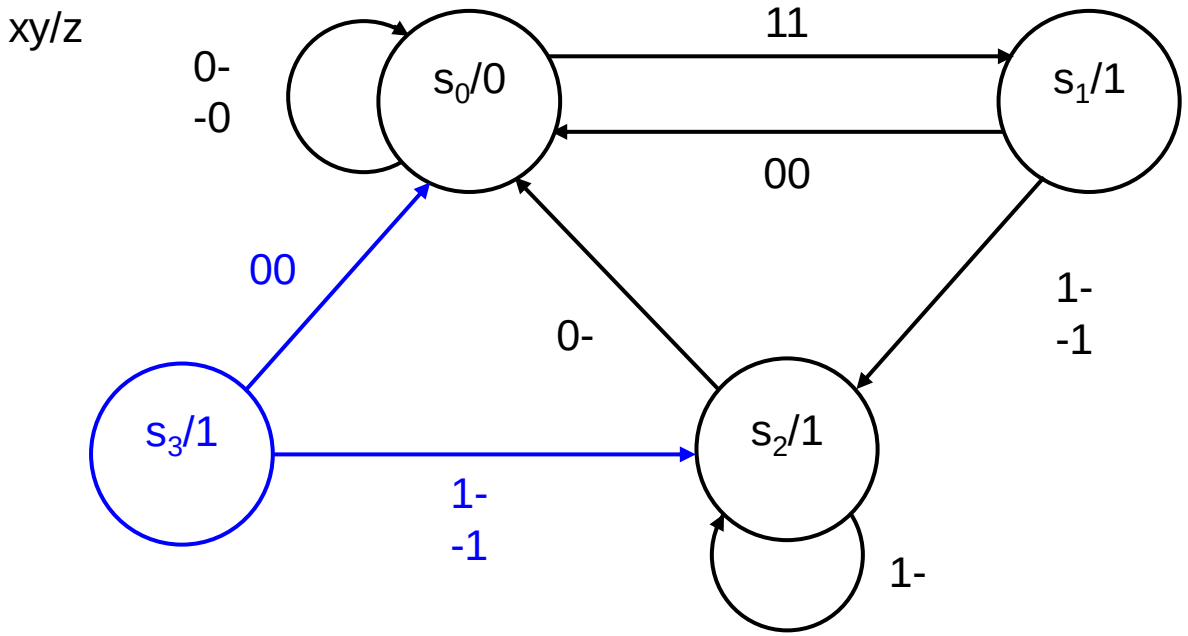
- Eén toestand is niet gebruikt: $Q_1Q_0 = 11$.
- Wat doet de machine als deze per ongeluk in de ongebruikte toestand terecht komt?
- Na onderzoek van de toestandsfuncties komt er de volgende functietabel uit (als $Q_1Q_0 = 11$):

x	y	naar	z
0	0	s_0	1
0	1	s_2	1
1	0	s_2	1
1	1	s_2	1

De uitgang van de schakeling blijft 1!

Toestandsmachines

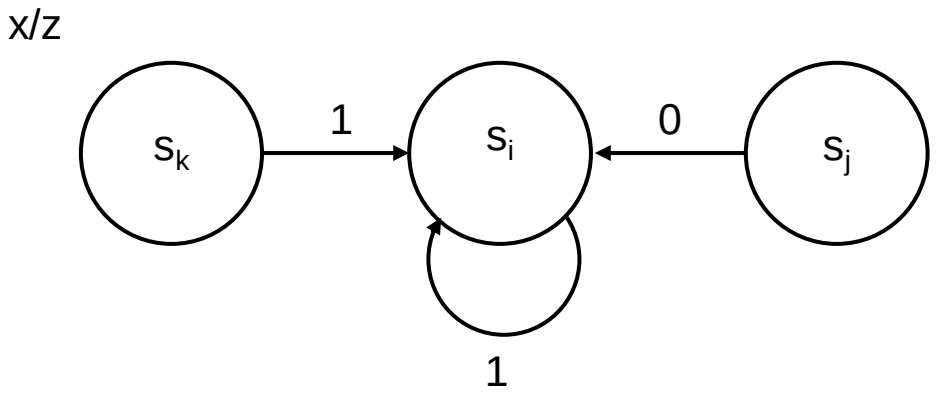
- Het volledige toestandsdiagram is (na uitwerking toestandsfuncties):



- Het is dus te voorspellen wat de machine gaat doen in ongebruikte toestanden.

Toestandsmachines

- Het vinden van de toestandsfuncties bij one-hot is eenvoudig.
- In toestand s_i: bekijk alle overgangen van toestanden naar s_i. Elke overgang levert een productterm voor de functie van s_i. De som van deze producttermen is de functie voor s_i.



$$D_i^n = Q_k \cdot x + Q_i \cdot x + Q_j \cdot \overline{x}$$

DE HAAGSE HOGESCHOOL

Toestandsmachines

- De keuze van de codering heeft directe consequenties voor de omvang van de combinatorische logica.
- De praktijk leert echter een aantal “optimale” coderingen voor bepaalde klassen van machines.
- Thijssen heeft onderzocht dat wanneer een groot aantal willekeurig gekozen coderingen wordt gebruikt, de verhouding tussen de codering met het minst aantal poorten (*gate equivalent*) en de codering met het meest aantal poorten een factor 1/3 – 1/4 bedraagt.
- Dus: probeer een tiental willekeurige combinaties en selecteer die met de minst aantal poorten.

Toestandsmachines

- Eerder zijn de functies uitgewerkt voor de codering $S_0 = 00$, $S_1 = 01$ en $S_2 = 10$. Dit levert de volgende functies op:

$$D_1^n = Q_1^n \cdot x^n + Q_0^n \cdot y^n + Q_0^n \cdot x^n$$

$$D_0^n = \overline{Q_1^n} \cdot \overline{Q_0^n} \cdot x^n \cdot y^n$$

$$Z^n = Q_1^n + Q_0^n$$

- Bij een codering van $S_0 = 11$, $S_1 = 10$ en $S_2 = 00$ worden de functies

$$D_1^n = Q_0^n + \overline{Q_1^n} \cdot \overline{x^n} + \overline{x^n} \cdot \overline{y^n}$$

$$D_0^n = Q_0^n \cdot \overline{x^n} + Q_0^n \cdot \overline{y^n} + \overline{Q_1^n} \cdot \overline{x^n} + \overline{x^n} \cdot \overline{y^n}$$

$$Z^n = \overline{Q_1^n}$$

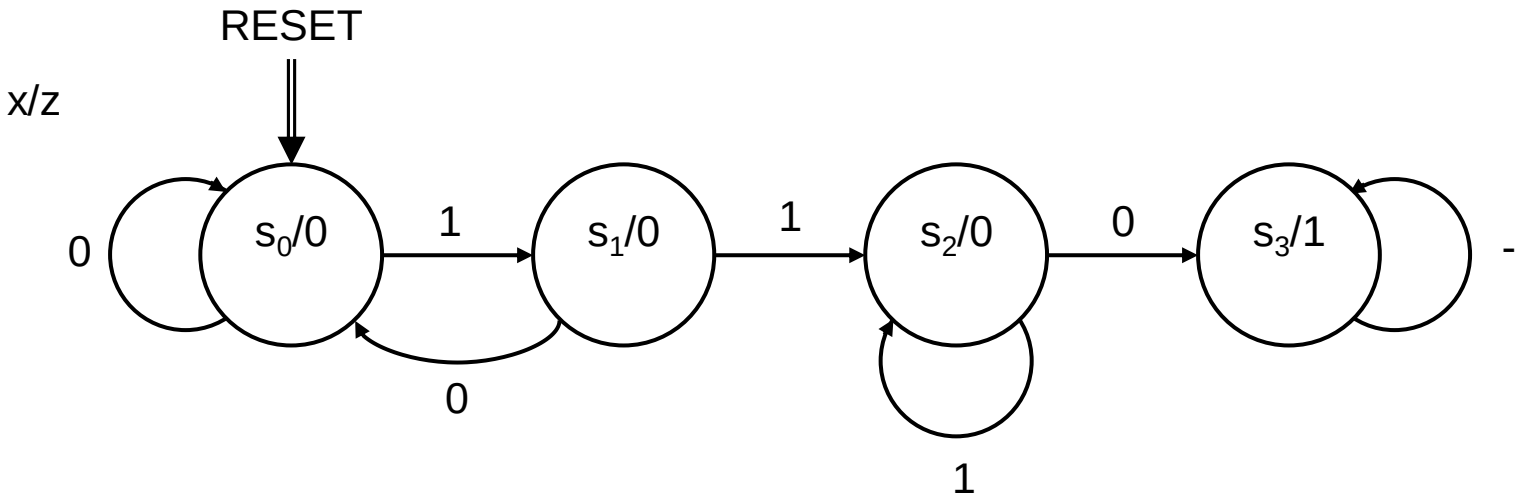
- De tweede codering levert meer hardware (= transistoren) op.

Toestandsmachines

- Een typische sequentiële machine is een *herkenningsautomaat* of *patroonherkenner*.
- In een serieel aangeboden *bitstring* wordt gezocht naar een patroon.
- De bitstring wordt aangeboden op een ingang.
- Als het patroon gevonden is wordt de uitgang 1, anders is de uitgang 0.

Toestandsmachines

- Ontwerp een machine voor het herkennen van het patroon (of reeks) 110. De machine geeft een 1 af en stopt met herkennen.

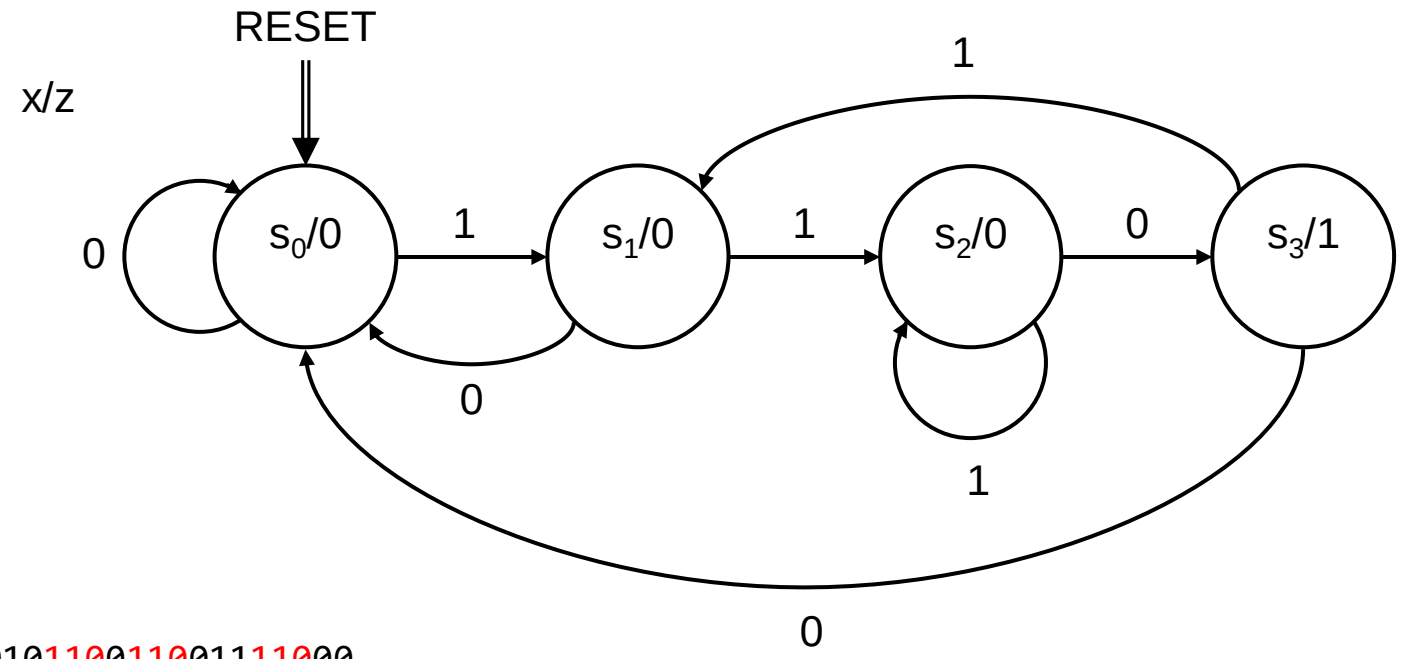



```
0010110011001111000
0000000111111111111
```


DE HAAGSE HOGESCHOOL

Toestandsmachines

- Ontwerp een machine voor het doorlopend herkennen van het patroon (of reeks) 110. De machine moet bij herkenning een 1 afgeven.

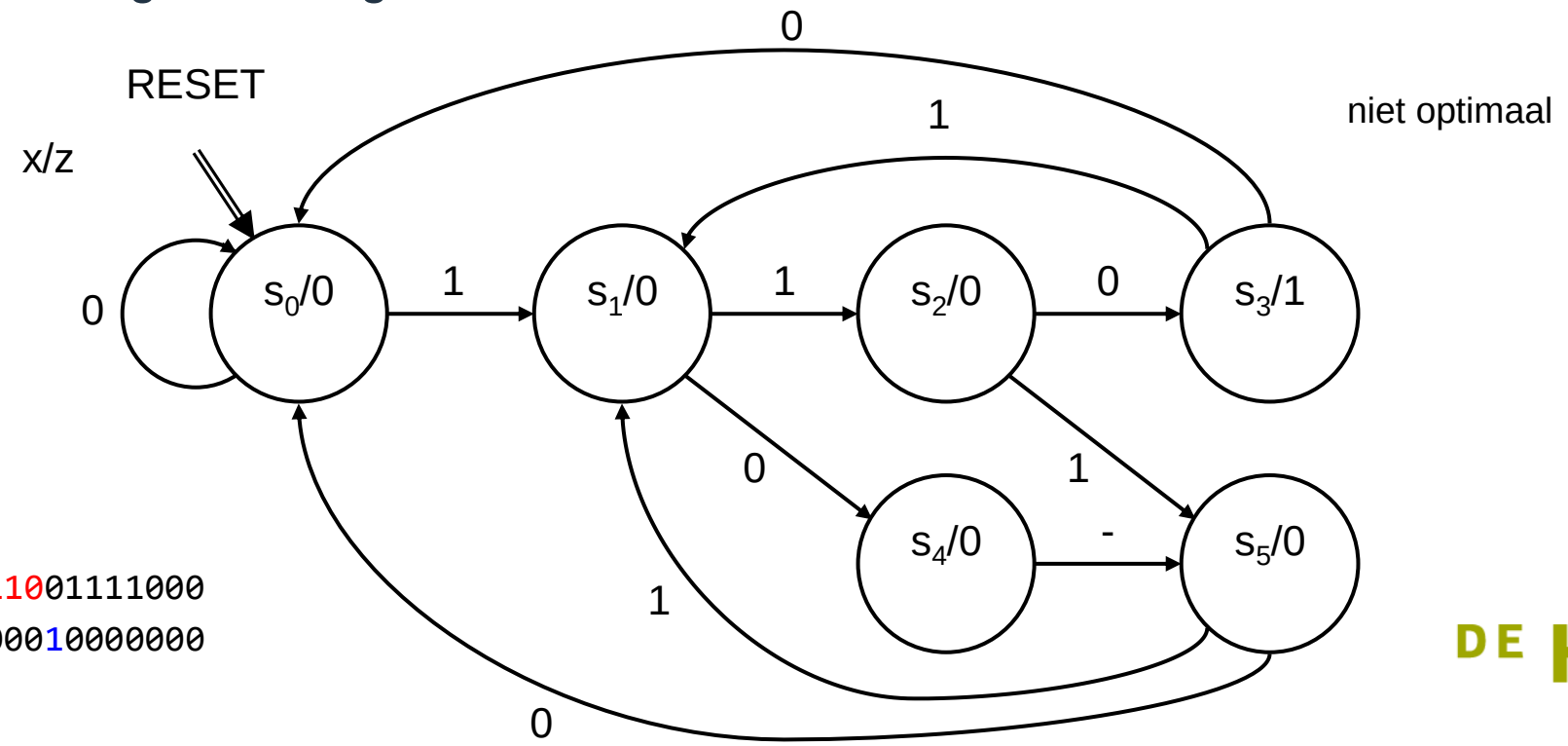



```
00101100110011111000
00000001000100000010
```


DE HAAGSE HOGESCHOOL

Toestandsmachines

- Ontwerp een machine voor het bloksgewijs herkennen van het patroon (of reeks) 110. De machine begint met herkennen bij een 1 en moet bij herkenning een 1 afgeven.

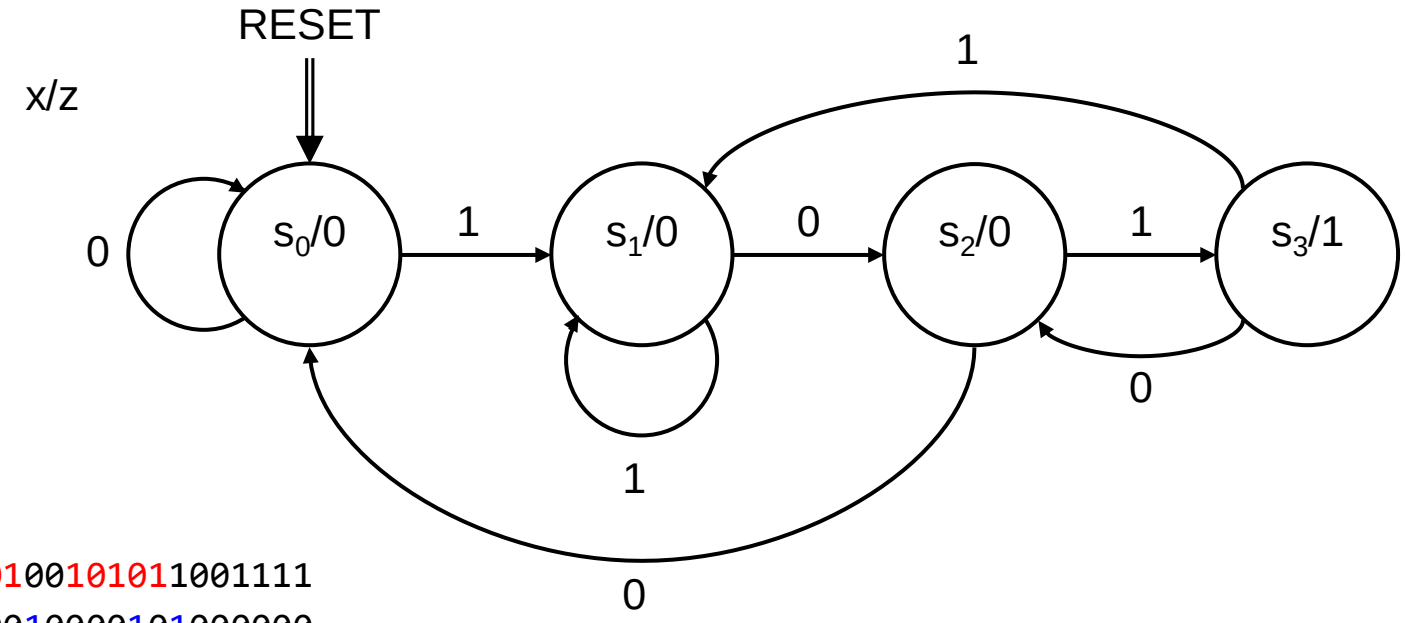



```
00101100110011111000
00000000000100000000
```


DE HAAGSE HOGESCHOOL

Toestandsmachines

- Ontwerp een machine voor het doorlopend herkennen van het patroon (of reeks) 101. Overlappingen zijn mogelijk. De machine moet bij herkenning een 1 afgeven.

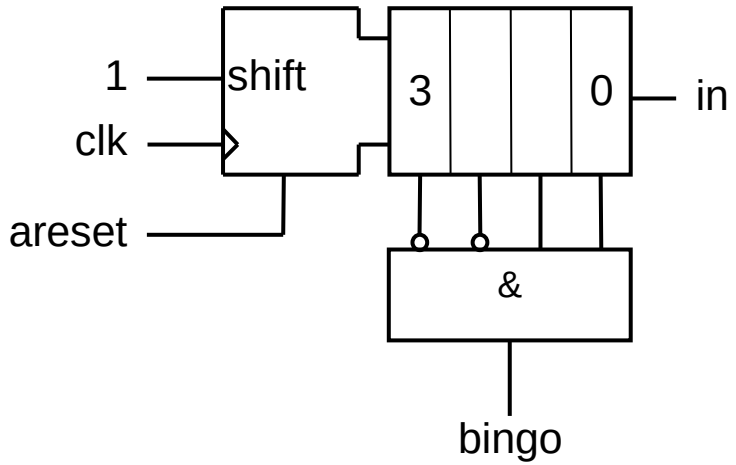



```
00101001010111001111
00000100001010000000
```


DE HAAGSE HOGESCHOOL

Toestandsmachines

- Een herkenningsautomaat kan gemaakt worden met behulp van een *schuifregister* en een AND-poort.
- Als voorbeeld het herkennen van 0011:

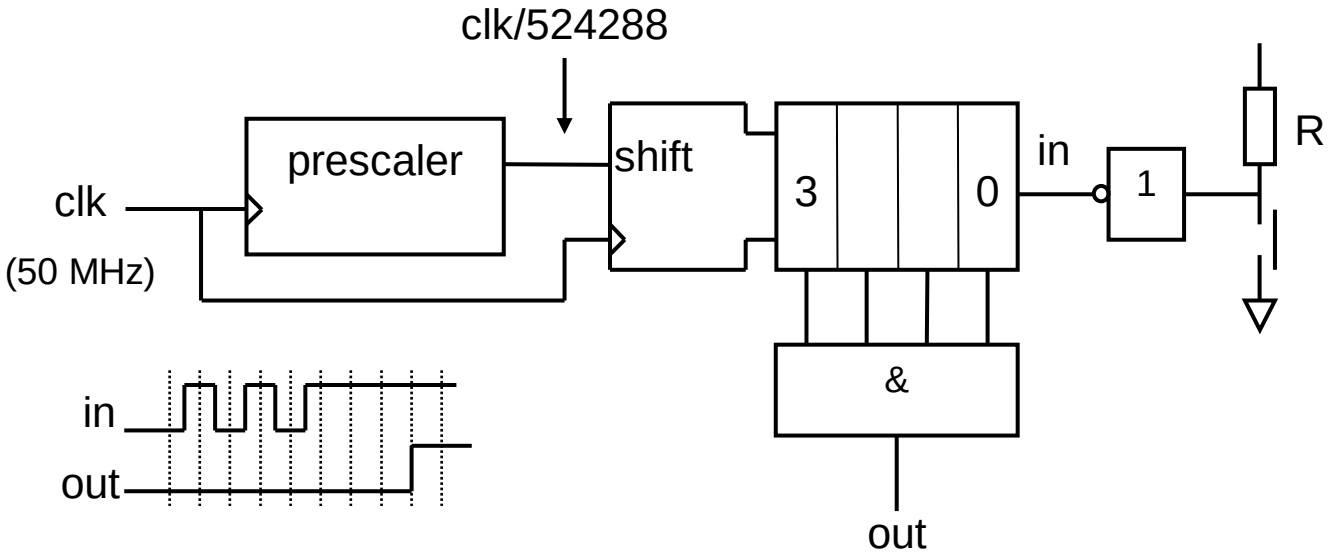


- Wat kan er mis gaan bij het aanbieden van 1100?

Toestandsmachines

- Een mechanische drukknop of schakelaar heeft last van *denderen*. Bij overgaan stuitert de kontaktveer. Dit levert meerdere pulsen op aan een ingang. Een herkenninsautomaat biedt uitkomst.

- Het denderen duurt 0,1 ms – 10 ms. Een systeemklok is erg snel, dat zou een te groot schuifregister opleveren. Een *prescaler* biedt uitkomst.



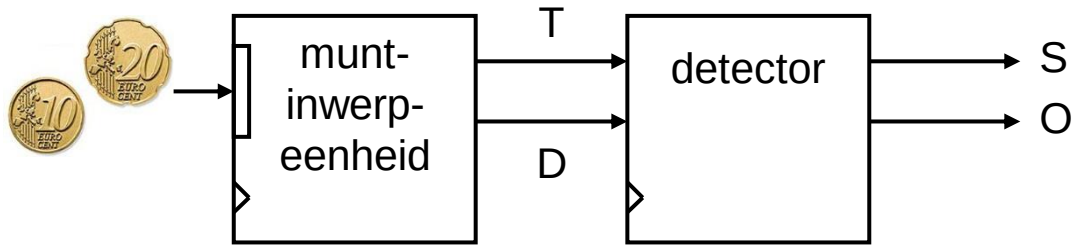
DE HAAGSE HOGESCHOOL

Toestandsmachines

- In het volgende voorbeeld wordt een machine ontwikkeld die een snoepautomaat bestuurt.
- De snoepautomaat accepteert alleen muntstukken van 20 cent (T) en 10 cent (D).
- Er is 30 cent nodig om snoepgoed vrij te geven.
- Als er meer dan 30 cent wordt ingeworpen moet de machine geen snoepgoed vrijgeven maar aanduiden dat er teveel geld is ingeworpen.

Toestandsmachines

- De munt-inwerpeenheid geeft twee signalen af, T voor een muntstuk van 20 cent en D voor een muntstuk van 10 cent.

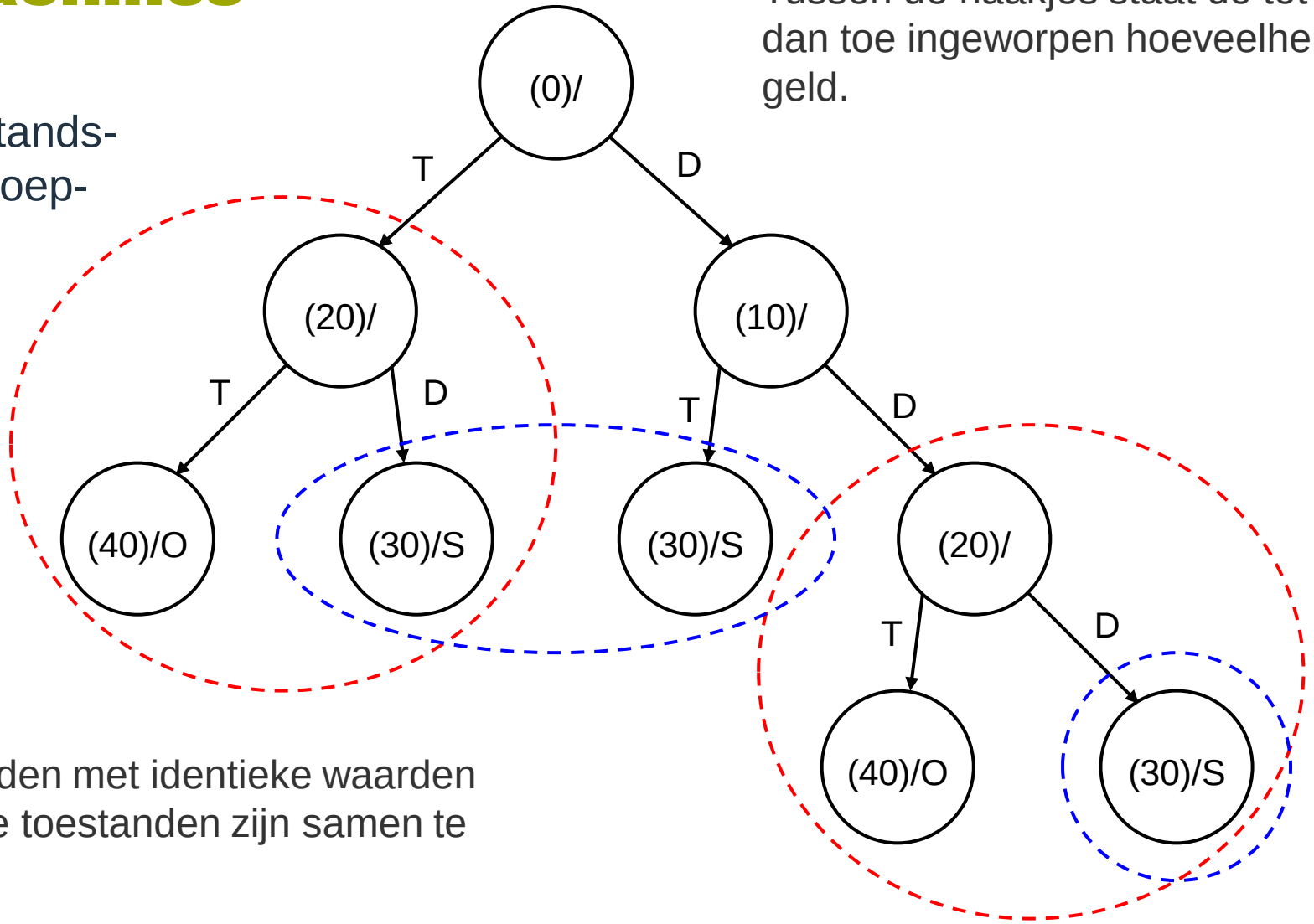


- De eenheid kan slechts één munt te gelijk accepteren. Na detectie van een muntstuk wordt het betreffende signaal één klokcyclus geactiveerd. De signalen T en D kunnen dus nooit tegelijk geactiveerd worden.

- Het signaal S wordt geactiveerd als 30 cent is ingeworpen. Het signaal O wordt geactiveerd als meer dan 30 cent is ingeworpen.

Toestandsmachines

- Hiernaast het toestands-diagram van de snoep-automaat.



Tussen de haakjes staat de tot dan toe ingeworpen hoeveelheid geld.

Merk op dat toestanden met identieke waarden equivalent zijn! Deze toestanden zijn samen te nemen.

Toestandsmachines

- Het toestandsdiagram kan met minder toestanden getekend worden door het hanteren van de volgende regels:

  T + T = 40
  T + D = 30
  D + T = 30
  D + D + T = 40
  D + D + D = 30

  D + D = T

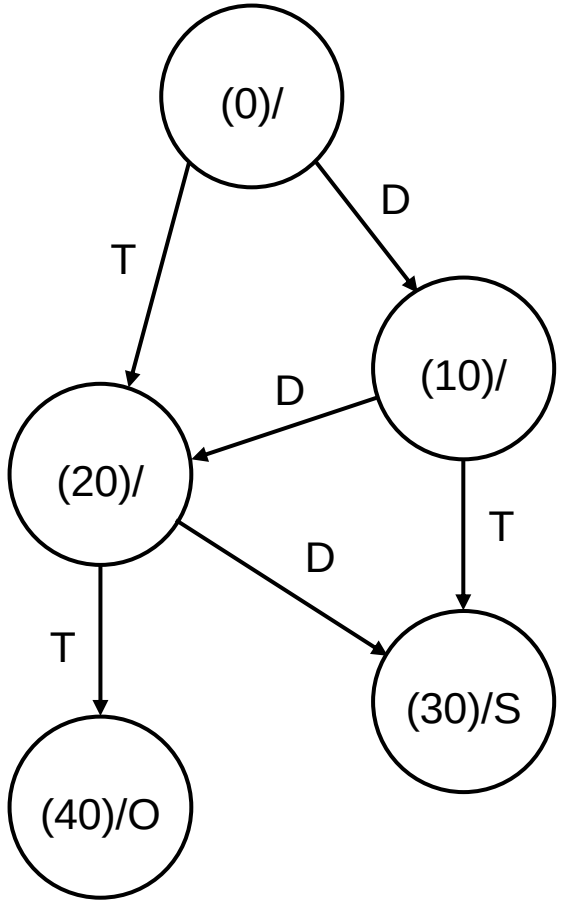
- Combinaties die leiden tot de hoeveelheid van 30 zijn dus identiek, net als voor 40. Verder is te zien dat 10+10 = 20. Meer dan 40 komt niet voor, want dan is minstens al een hoeveelheid van 30 ingeworpen.

Toestandsmachines

- Er wordt gebruik gemaakt van het feit dat D + D = T
  en dat D +T = T + D

  D = 10
  T = D + D = 20
  T + D = D + T = 30
  T + T = 40

- Het aantal toestanden is verminderd tot 5.



DE HAAGSE HOGESCHOOL

Toestandsmachines

- Net als bij combinatoriek is het mogelijk het aantal toestanden te minimaliseren.

- Het idee is te zoeken naar *equivalente toestanden*.
  - Twee toestanden zijn equivalent indien voor elke mogelijke reeks ingangswaarden dezelfde reeks uitgangswaarden wordt geproduceerd.

- Formele procedure: Paull - Unger (1959)

- In de praktijk wordt dit weinig gebruikt, zeker als het aantal toestanden gering is. Een goede ontwerper zorgt meestal voor een minimale of bijna-minimale oplossing.

Toestandsmachines

- De uitgangen van de Mealy-machine veranderen op de actieve klokflank én de ingangen.

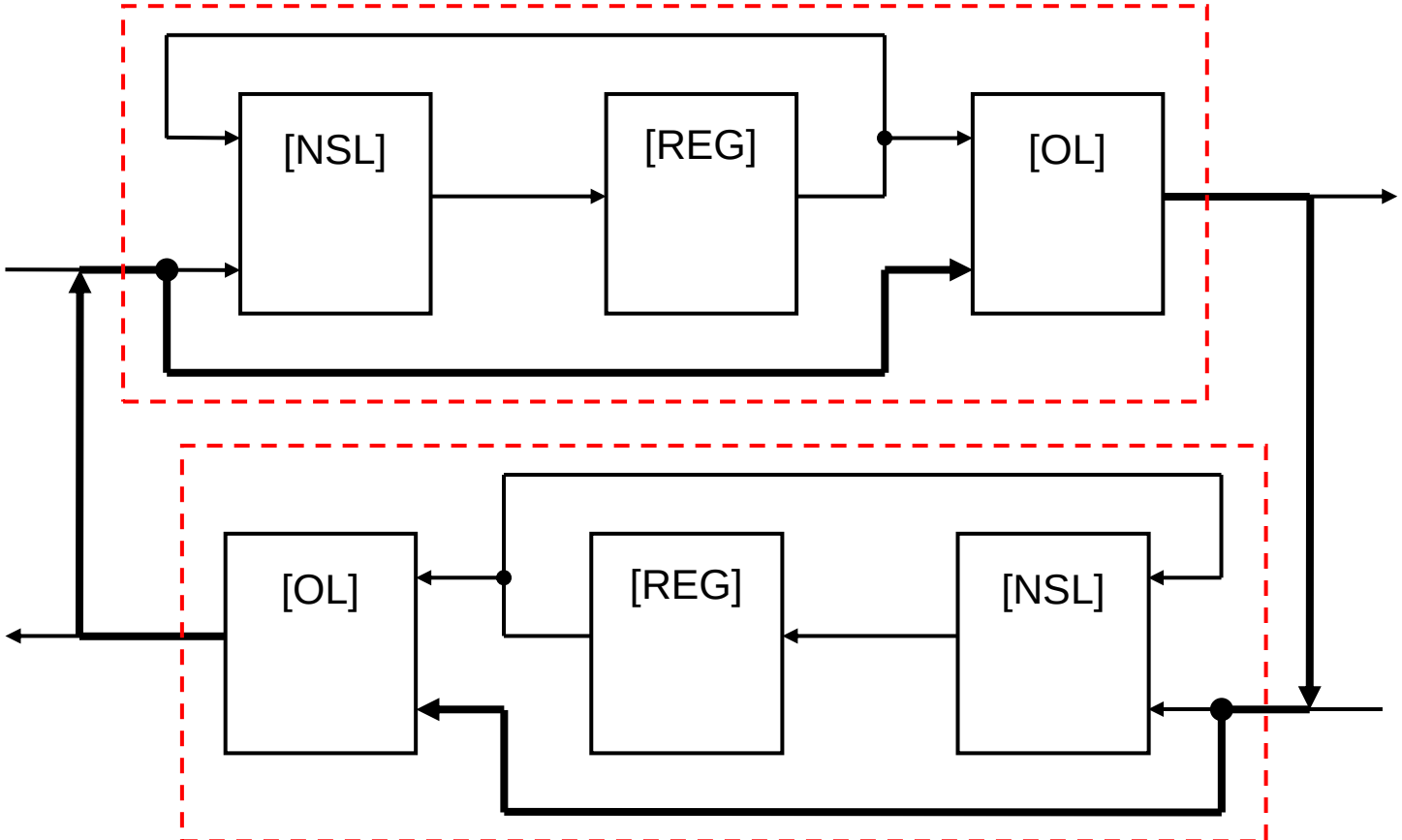
- Dat houdt in dat de uitgangen twee sets vertragingen hebben.
  - Ten opzichte van de actieve klokflank (synchroon)
  - Ten opzichte van de ingangen (asynchroon)

- Moore-machines hebben dat probleem niet, de uitgangen veranderen altijd synchroon met de klokflank.

- Het tijdgedrag van synchrone machines kunnen d.m.v. de bekende procedures doorgerekend worden.

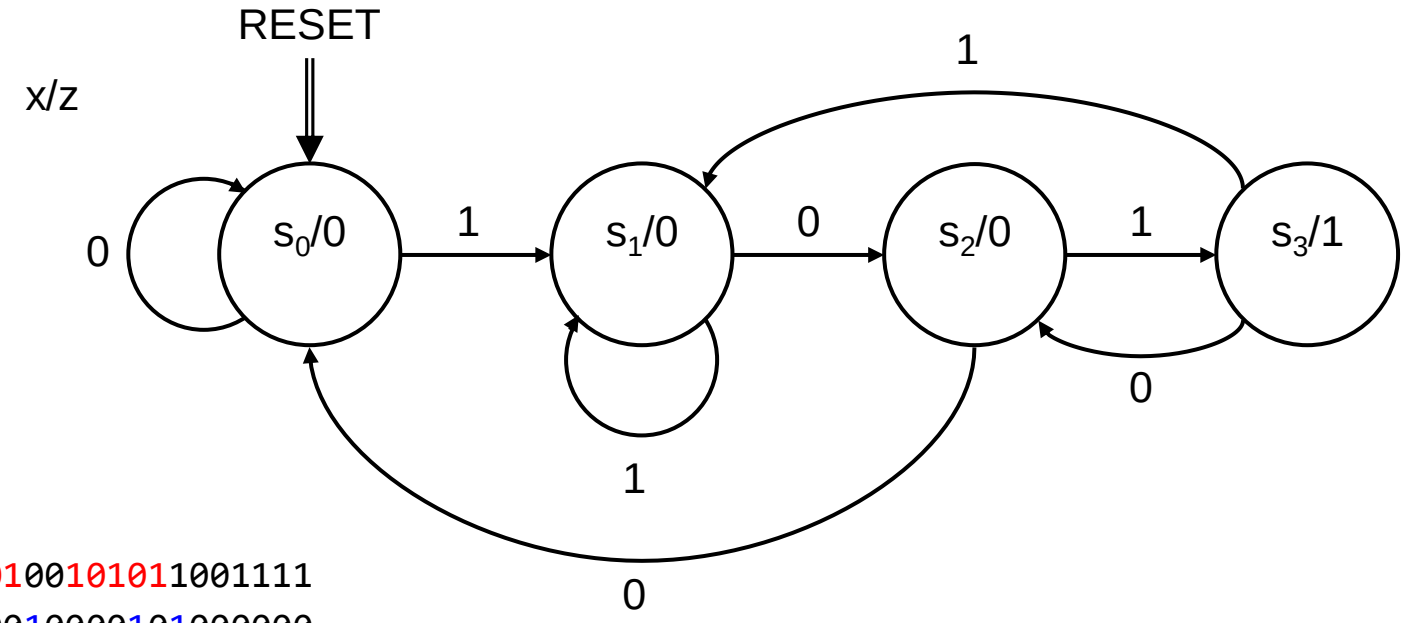
Toestandsmachines

- Rondgekoppelde Mealy-machines kunnen last hebben van asynchrone terugkoppeling. Moore-machines hebben dat probleem niet.



Toestandsmachines

- Ontwerp een machine voor het doorlopend herkennen van het patroon (of reeks) 101. Overlappingen zijn mogelijk. De machine moet bij herkenning een 1 afgeven.



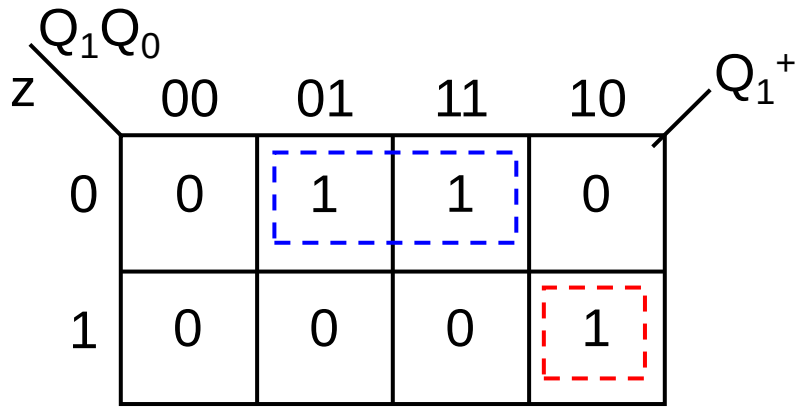

```
00101001010111001111
00000100001010000000
```


DE HAAGSE HOGESCHOOL

Toestandsmachines

- Toestandscodering: $s_0 = 00$, $s_1 = 01$, $s_2 = 10$ en $s_3 = 11$.

Q_1^n	Q_0^n	x^n	Q_1^{n+1}	Q_0^{n+1}	z^n
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	0	1	1



$$Q_1^{n+1} = Q_0^n \cdot \overline{z^n} + Q_1^n \cdot \overline{Q_0^n} \cdot z^n$$

DE HAAGSE HOGESCHOOL

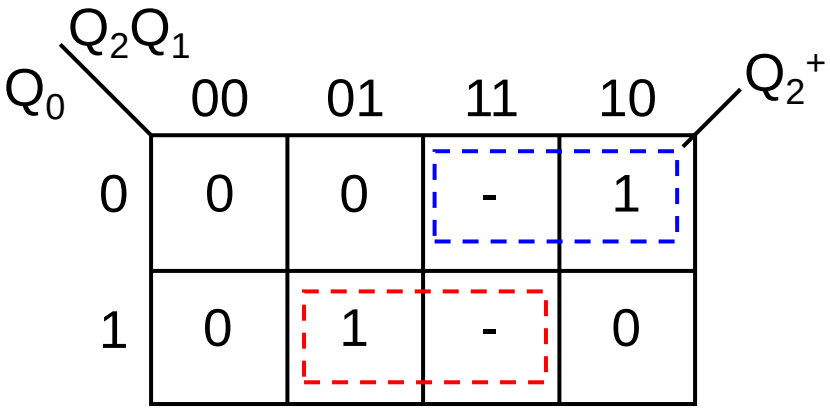
Toestandsmachines

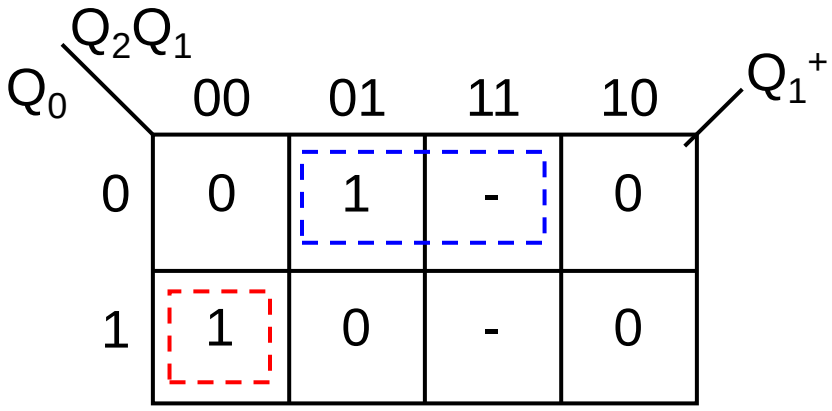
- Ontwerpen van een 6-teller kan op de bekende wijze.
- De waarheidstabel van de 6-teller:

Q_2^n	Q_1^n	Q_0^n	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	-	-	-
1	1	1	-	-	-

Toestandsmachines

- Karnaughdiagrammen en functies:





$$Q_2^{n+1} = Q_2^n \cdot \overline{Q_0^n} + Q_1^n \cdot Q_0^n$$

$$Q_1^{n+1} = \overline{Q_2^n} \cdot \overline{Q_1^n} \cdot Q_0^n + Q_1^n \cdot \overline{Q_0^n}$$

$$Q_0^{n+1} = \overline{Q_0^n}$$

DE HAAGSE HOGESCHOOL

let’s change