

Registers

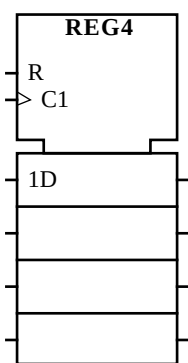
In digitale systemen moet vaak gedurende een zekere tijd een binaire informatie of codewoord worden opgeslagen. Dit gebeurt dan in een register. Het register heeft zoveel flipflops als dat er bits zijn in de binaire informatie.

Soms is het nodig om de informatie in het register één bit te verplaatsen. Dergelijk register noemen we dan een schuifregister.

De informatietoevoer en -afvoer in het register kan serieel of parallel gebeuren. Het algemeen symbool van een 4-bitregister is voorgesteld in figuur 1.1.

De vier gelijke blokjes stellen de vier flipflops voor. Het bovenste blokje stelt de bijkomende logica voor.

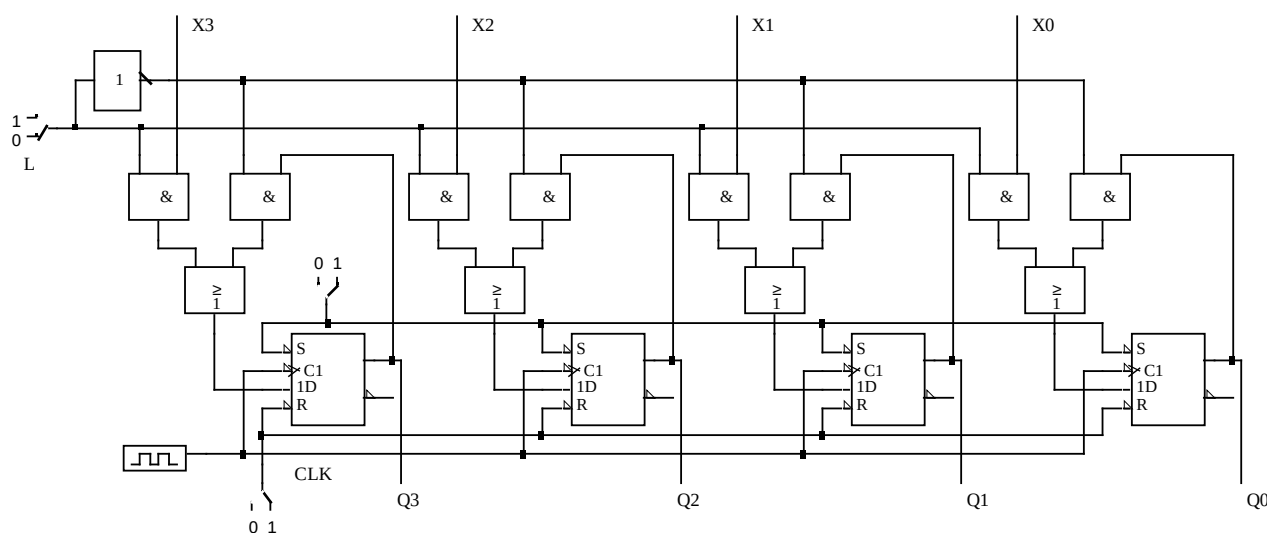
Registers kunnen gebruikt worden als schuifregisters, tellers, serie-parallel omzetter, codegeneratoren, enz.



figuur 1.1

Bufferregisters

Bufferregisters hebben tot doel een binair woord op te nemen en voor een bepaalde tijd in geheugen te houden. De gebruikelijke uitvoering van een bufferregister wordt weergegeven in figuur 1.2.

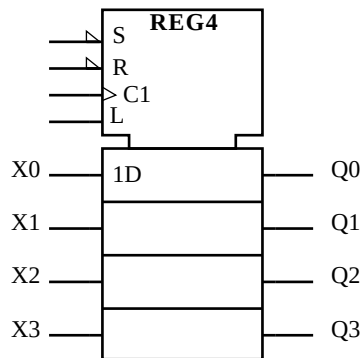


figuur 1.2

Aan de ingangen X0, X1, X2, en, X3 wordt het binaire codewoord aangesloten. Bij het optreden van de klokimpuls wordt het binaire woord in de flipflops opgeslagen zodanig dat $Q3 = X3$, $Q2 = X2$, $Q1 = X1$ en $Q0 = X0$ wordt. Dit is een asynchrone reset omdat deze ingang onafhankelijk is van het kloksignaal C1.

Door de clear-ingang laag te maken worden alle flipflops gereset zodat het ingeschreven woord $Q = 0$ is.

Wordt nu de clear-ingang op hoog gezet, dan is het register klaar om een bepaald woord in geheugen op te nemen.



figuur 2.1

Figuur 2.1 geeft het IEC-symbool voor de schakeling van figuur 1.2. De gemeenschappelijke besturingssignalen (R, S, C1 en L) voor al de D flipflops bevinden zich in het besturingsblok.

Met de load (L) kan het register al of niet geladen worden. De load is hoog actief.

Met de reset (R) kan het volledige register op nul gezet worden. De reset is laag actief.

Met de set (S) kan het volledige register op één gezet worden. De set is laag actief.

Wordt de load-ingang "1" gemaakt, dan wordt de X-informatie doorgespeeld naar de respectievelijke D-ingangen.

Wordt de laod ingang "0" gemaakt dan wordt de Q-informa-

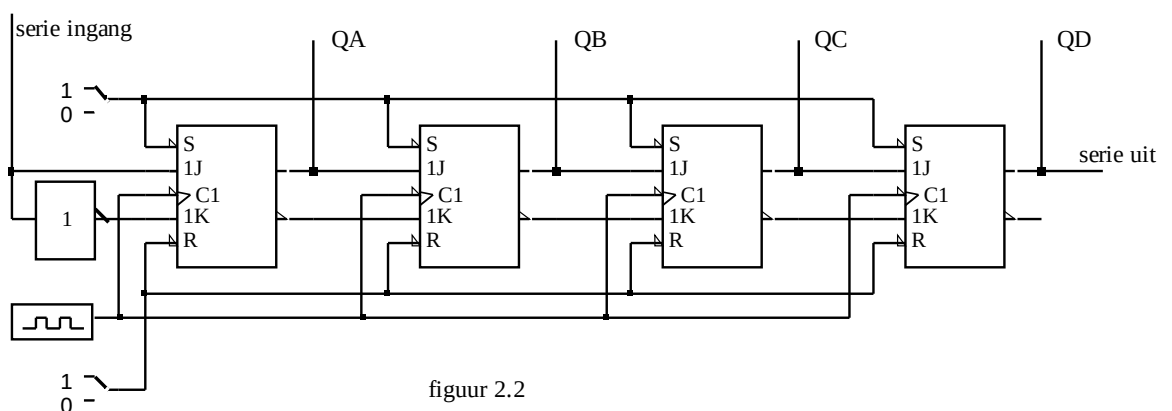
tie doorgespeeld naar de respectievelijke D-ingangen.

De klokimpuls C1 leert ons dat alle D-ingangen reageren op die klokimpuls. Hier is dit op de 0-1 flank of de stijgende flank. Bij de load-ingang (L) is geen polariteitsindicator getekend vandaar dat deze ingang hoog actief is, in tegenstelling tot de reset- en de set-ingang.

Schuifregisters

Een shuifregister dient om informatie dat het bevat één of meerdere posities naar links of naar rechts te verschuiven. De verschuiving gebeurt slechts één positie ineens; bij iedere klokimpuls verschuift de informatie slechts over één naburige flipflop. Het aantal schuifimpulsen of klokimpulsen dat nodig is om de volledige informatie in het geheugen op te nemen is daarom gelijk aan het aantal flipflops dat het register bevat.

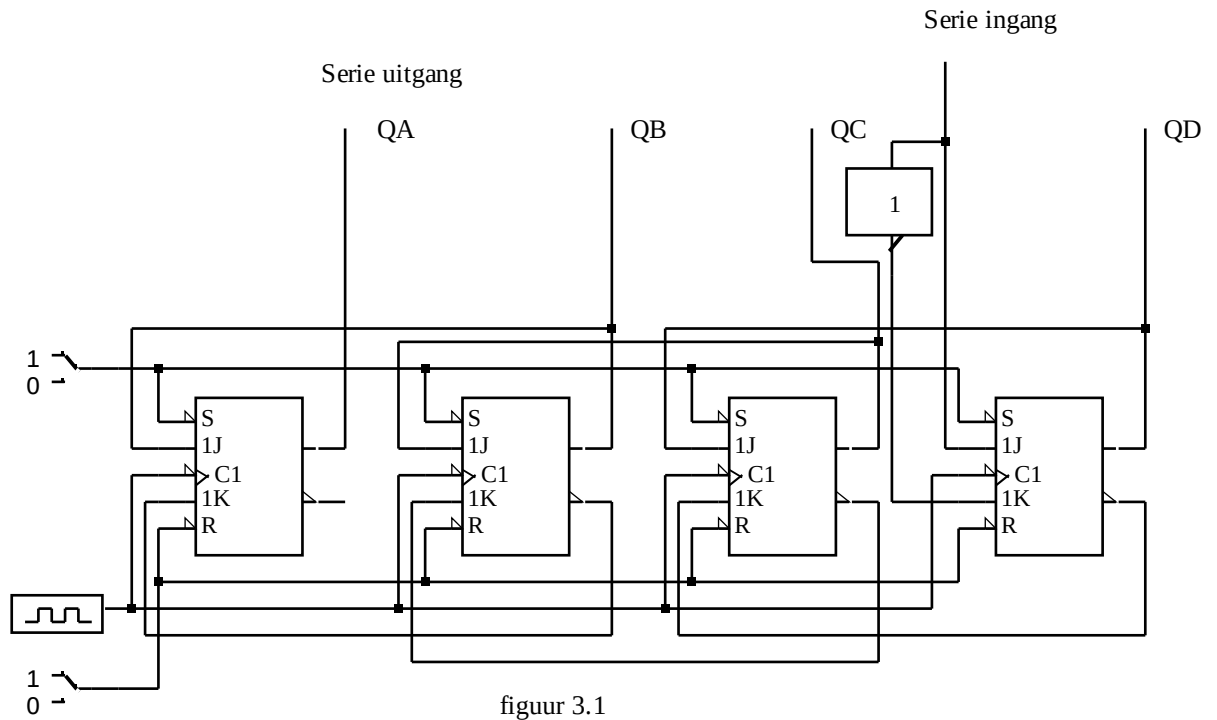
Rechts schuivend register



figuur 2.2

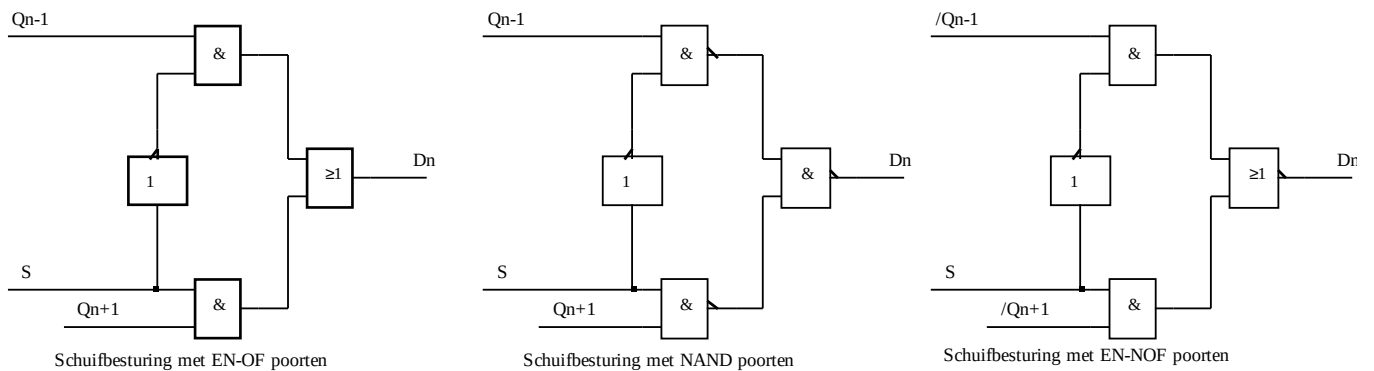
Bij een rechts schuivend register wordt de ingang van de volgende FF. doorverbonden met de uitgang van de vorige FF..

Links schuivend register.



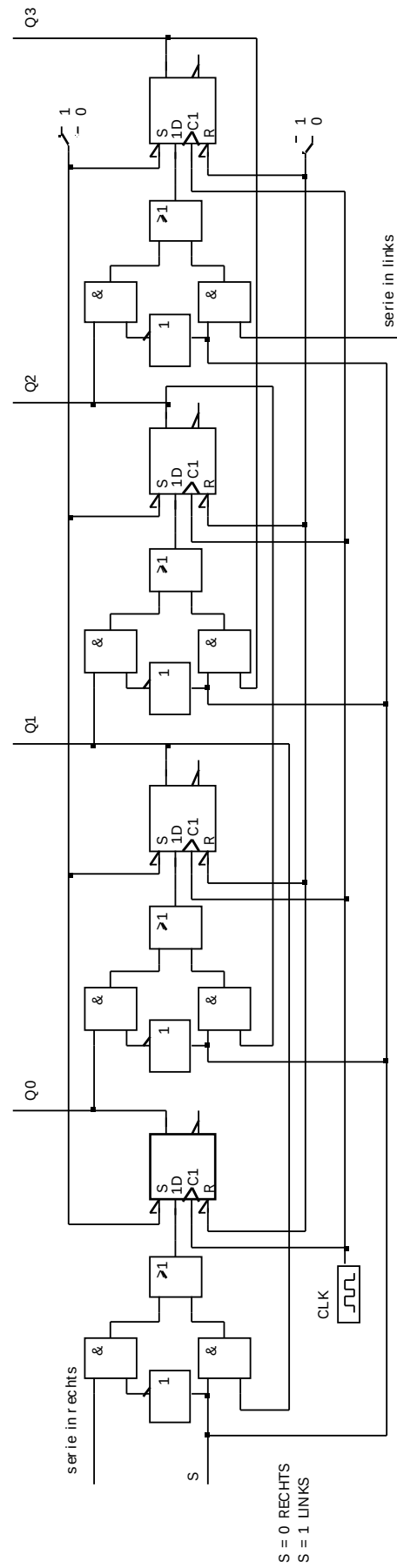
Bij een links schuivend register wordt de uigang van de volgende FF. doorverbonden met de ingang van de vorige FF..

Door gebruik te maken van één van onderstaande schakelingen kan men de schuifbesturing aanpassen.

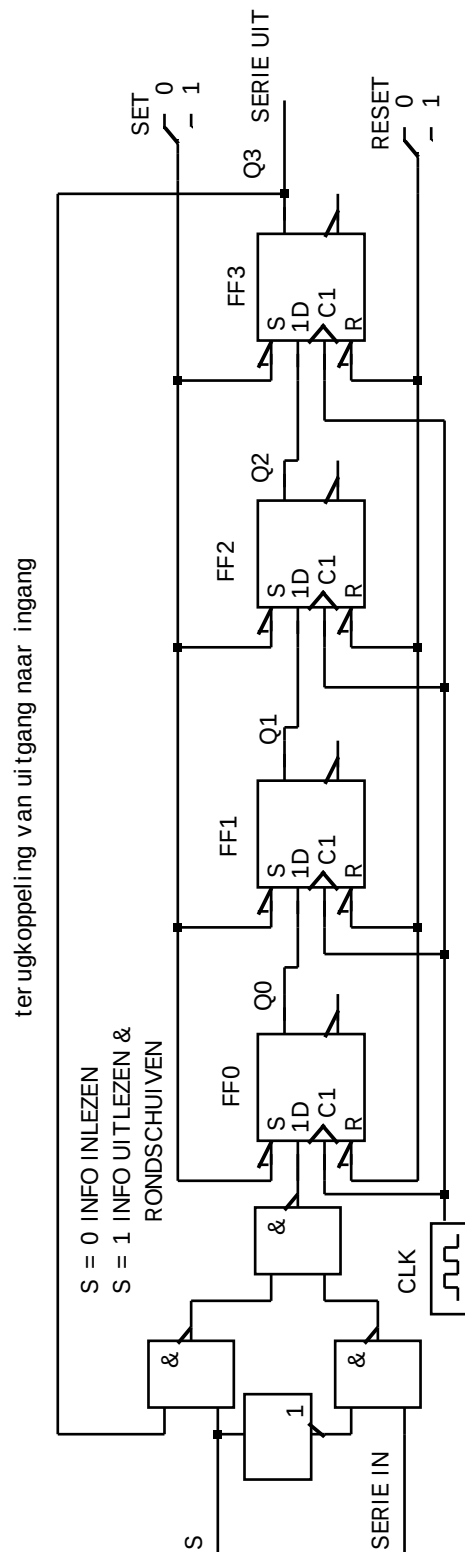


Op de volgende bladzijde zie je een voorbeeld van de links rechts schuivend schuifregister.

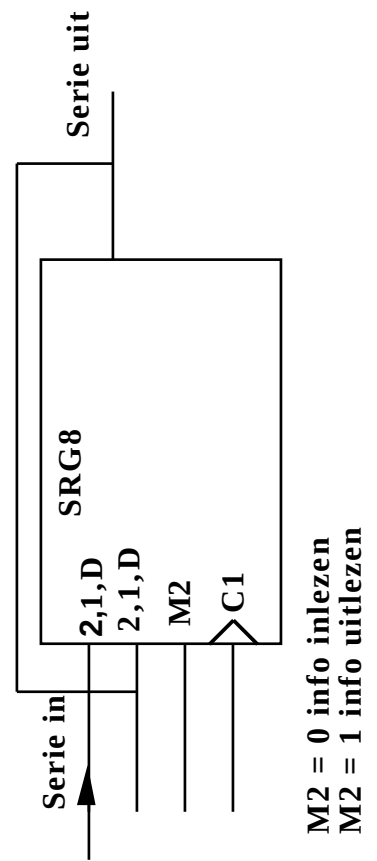
Links rechts schuivend schuifregister



Informatietoevoer bij een rechtsdraaiend schuifregister.



De bedoeling van de terugkoppeling van de uitgang naar de ingang is, dat bij het uitlezen van de informatie; de ingeschoven informatie niet zou verloren gaan.



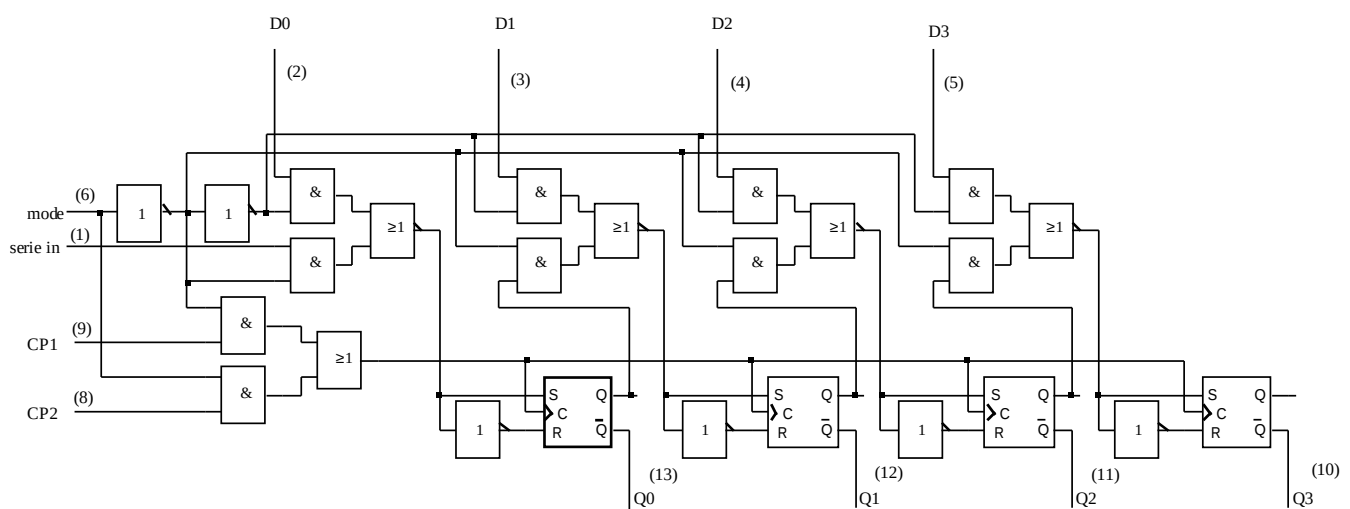
Toepassingen van schuifregisters.

Schuifregisters worden in digitale schakelingen, en vooral in de computertechniek, zeer veel gebruikt. We noemen hier enkele toepassingen op.

- Bufferregisters worden gebruikt om gedurende een bepaalde tijd een binair codewoord te bewaren.
- Schuifregisters in de serie in- serie uit mode kunnen gebruikt worden als vertragsregister. De opgeslagen data zal aan de uitgang verschijnen met een vertraging van N impulsen. In die mode kan het schuifregister ook gebruikt worden als tijdelijke opslag van informatie. Een bijzondere toepassing in de serie in- serie uit mode is de "*Ringteller*" en "*Johnson teller*". Deze worden uitvoerig behandeld in het hoofdstuk van de ringtellers.
- Schuifregisters in de parallel in- parallel uit mode vinden toepassing bij karakter- en codegeneratoren. Voor een herhaling van de code wordt de schakeling gecombineerd met een ringteller die het rondschuiven verzorgt. Uiteraard kan een dergelijk schuifregister gebruikt worden als buffer- of geheugenregister.
- Schuifregisters in de serie in- serie uit mode worden o.a. gebruikt bij data-transmissie. Dit zal het geval zijn waar de bewerking of verwerking van de digitale signalen bit per bit gebeurt (serieel) en bv. het zichtbaar maken van de informatie parallel moet zijn. Denken we hier bijv. aan het uitvoeren van rekenkundige bewerkingen en het zichtbaar maken van het resultaat op een display.
- Schuifregisters in de serie in- serie uit mode worden o.a. gebruikt bij data-transmissie. Dit zal het geval zijn waar de digitale informatie over een lange afstand moet overgebracht worden. Denken we hier o.a. aan de digitale afstandsbediening van ons T.V. toestel en teletext.
- A/D omvormers S.A.R. (Successive approximation register) zie blz 49 hoofdstuk A/D- en D/A omvormers.

Universele schuifregisters

Een rechts schuivend register, links schuivend register, parallel in parallel uit, kunnen gecombineerd worden in een "*Universeel register*". We bestuderen een paar universele schuifregisters aan de hand van voorbeelden uit de TTL-techniek.

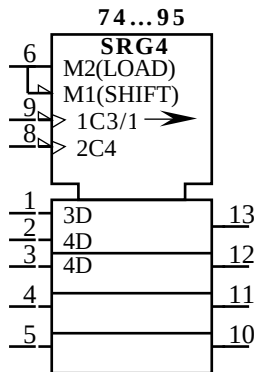


figuur 4.1

In dit schuifregister worden flankgestuurde SR-flipflops gebruikt.

Het IEC-symbool van de 74...95.

Figuur 7.1 toont het IEC-symbool van het universeel schuifregister 74...95.



figuur 7.1

Pin 6 is de mode ingang.

Pin 9 is de clock ingang om naar rechts te schuiven, die afhankelijk is van de mode ingang. Hiervoor moet de mode ingang "0" zijn. De seriële ingang is dan pin 1, dit kan je zien aan de hand van de notatie 3D.

Pin 8 is de clock ingang om de parallelle aangeboden data te vergrendelen. De data ingangen zijn die ingangen aangeduidt met 4D. Hiervoor moet de mode ingang "1" zijn.

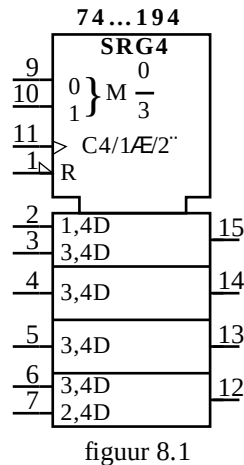
De pinnen 10 tot en met 13 zijn dan de parallelle uitgangen.

Het links schuiven gebeurt voor de mode ingang gelijk aan "1", maar dan moeten de data-ingangen verbonden worden met de parallelle uitgangen.

Opgave 7.1

Teken het schema van een links schuivend schuifregister met behulp van de 74...95. Duidt tevens de serie ingang en de serie uitgang aan. Maak gebruik van het IEC-symbool van de 74...95.

Het IEC-symbool van de 74...194.



figuur 8.1

De 74...194 is een 4-bits bidirectioneel universeel schuifregister. Het schuifregister kan in serie of parallel geladen worden met serie of parallel afvoer; het kan links of rechts schuiven, en heeft een z.g. "Houdmode" waarbij al de schuifbewerkingen geblokkeerd worden.

Figuur 8.1 toont het IEC-symbool van het schuifregister.

De M-ingang is hier een combinatie van de pennen 9 en 10.

SRG4 is het functiesymbool dat aangeeft dat we hier te doen hebben met een 4-bits schuifregister.

De accolade is een bit-groeperingssymbool om twee of meer bits aan een in- of uitgang te groeperen. Door de toepassing van dit symbool in deze situatie, kunnen we het tekenen van een complete code-omzetter vermijden. De ingangen worden aangeduid als bit 2^0 en bit 2^1 . Daarvan vinden we alleen de exponenten terug in de notatie binnen het symbool. Die twee

bits zijn gegroepeerd door de accolade waarachter meteen het resultaat is af

te lezen, n.l. $M_{\frac{0}{3}}$
 $\frac{0}{3}$ Is hierin de verkorte schrijfwijze voor 0 t.m. 3. Met andere woorden: M0, M1, M2, M3.

Onderstaande waarheidstabel geeft de vier mogelijkheden van de mode ingang.

pen 10	pen 9	in.log.waarde		Mode
L	L	0	0	M0
L	H	0	1	M1
H	L	1	0	M2
H	H	1	1	M3

inhibit
 schuif rechts
 schuif links
 laad parallel

Als we in het symbool van figuur 8.1 zoeken naar het volgnummer 0, dan blijkt dat nergens voor te komen. M0 is dus niet gebruikt, in die zin, dat als de pennen 9 en 10 beide laag (L) staan, geen functies zijn geselecteerd en de overige functie zijn *uitgesloten*. Met andere woorden het register is volledig geblokkeerd. M0 blijkt dus een 'disable' stand te zijn voor het hele schuifregister.

Is M1 geselecteerd, dan zien we dat 'schuiven naar rechts' (=naar beneden) is gekozen. De 1 vinden we bovendien bij pen 2 terug, samen met de 4 van C4. De volgorde 1,4 zegt ons dat eerst M1 waar moet zijn en daarna C4, om de informatie van pen2 in het eerste element te kunnen invoeren. Dat betekent tevens dat de daarin al aanwezige informatie wordt doorgeschoven naar het tweede element, enz.. De informatie die in het vierde element aanwezig was, wordt vervangen door die van het derde element.

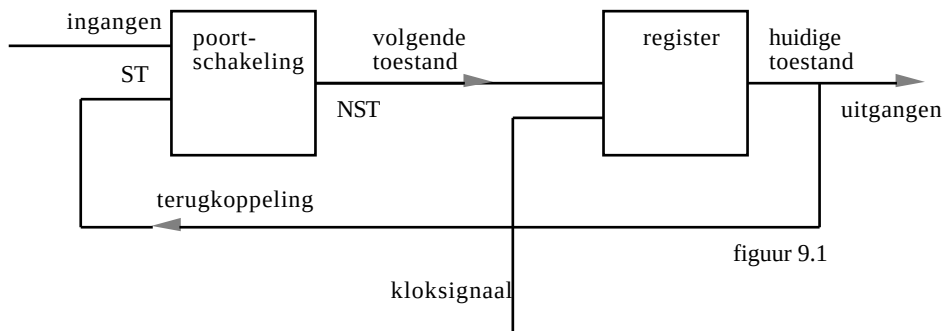
Is M2 geselecteerd, dan is schuiven naar links (=naar boven) gekozen. De informatie wordt dan via pen 7 ingevoerd bij het verschuiven van de L naar H-flank op pen11. En de hele procedure verloopt in omgekeerde volgorde als beschreven in de vorige alinea.

Bij M3, tenslotte, zien we dat de informatie via de pennen 3,4,5, en 6 parallel wordt ingevoerd bij het optreden van de L-H flank aan pen 11.

PARALLEL LOAD -PARALLEL OUT REGISTER MET D-FF.

De structuur van een sequentiële schakeling is gegeven in figuur 9.1. De structuur bestaat uit een aantal geklokte flipflops die één register vormen en een groep poortschakelingen die samen één combinatorische schakeling vormen.

De combinatorische schakeling is opgenomen in een terugkoppelpad waardoor de uitgangen van het register zijn verbonden met de ingangen van de combinatorische schakeling.

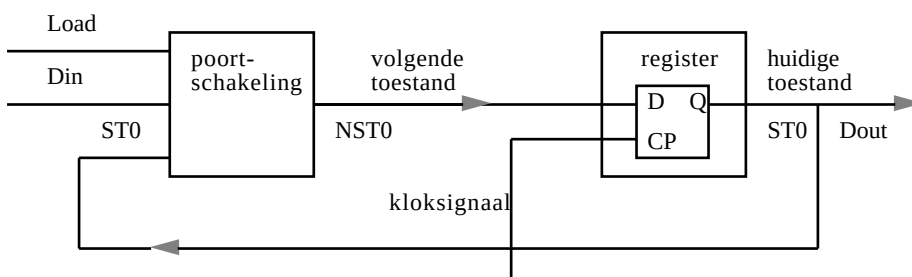


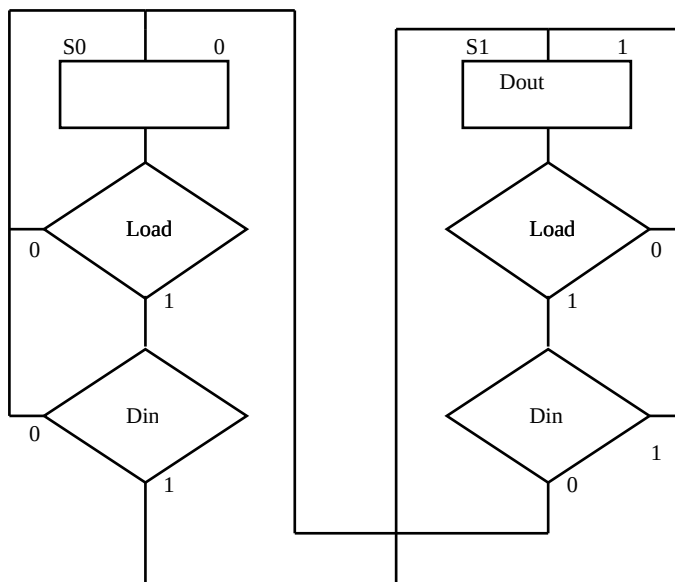
Als de waarden van alle ingangen op dezelfde positieve flank van het kloksignaal CP in het register geladen worden, spreekt men van een parallel laadbaar register.

Elk logisch systeem is uitgerust met een kloksignaalgenerator. De klokgenerator is een oscillator, die een continue blokspanning met een vaste frequentie opwekt. Deze continue blokspanning wordt het kloksignaal CP genoemd en vormt de hartslag van het totale logische systeem.

Het is meestal niet gewenst dat een register, toegepast in een logisch systeem, bij elke positieve flank van het signaal CP geladen wordt. Daarom is aan het parallel laadbare register een besturingsingang met de naam LOAD toegevoegd. Met de besturing van deze ingang LOAD kan de ontwerper kiezen met welke klokimpuls en op welk tijdstip het register geladen zal worden.

Om een register met deze eigenschap te realiseren, is het voldoende de schakeling van slechts één bit te ontwerpen. Tijdens de realisatie kan dan de gewenste registerbreedte verkregen worden door de schakeling van dit éne bit te herhalen. Figuur 9.2 geeft het blokschema van het 1-bit register.





Figuur 10.1 toestandsdiagram van een 1-bit register

INGANGEN			UITGANGEN	
Load	Din	ST0	NST0	Dout
0	X	0	0	0
1	0	0	0	0
1	1	0	1	0
0	X	1	1	1
1	0	1	0	1
1	1	1	1	1

Figuur 10.2 Waarheidstabel

Din ST0					
Load		00	01	11	10
0		0	1	1	0
1		0	0	1	1

$$NST = Load.Din + NOTLoad.ST0$$

De waarheidstabel figuur 10.2 van de combinatorische schakeling kan nu opgesteld worden door het volgen van alle paden van het toestandsdiagram figuur 10.1.

Elk pad levert één regel van de waarheidstabel op. Een pad begint altijd met het uitgangspad van een toestandsblok en eindigt op het ingangspad van een toestandsblok.

Het toestandsblok waar het pad begint, symboliseert de huidige toestand of 'state'. Het toestandsblok waar het pad eindigt geeft de volgende toestand of 'next state' aan.

De waarde van het signaal ST0 is gelijk aan de waarde van de huidige toestand. De waarde van het signaal NST0 is gelijk aan de waarde van de volgende toestand.

In de te realiseren schakeling komt het signaal NST0 overeen met de ingang D, en het signaal ST0 met de uitgang Q van de DFF, die het 1-bit parallel load register bevat.

Het toestandsdiagram figuur 10.1 heeft zes verschillende paden. Ga dit zelf na aan de hand van de op deze wijze gevonden waarheidstabel van figuur 10.2!

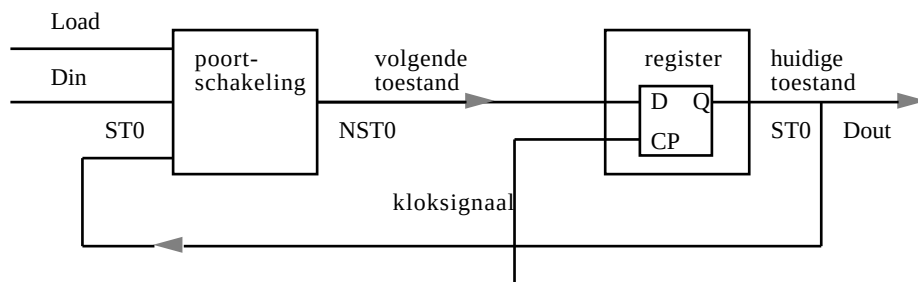
De eerste regel van de tabel kan als volgt gelezen worden:

Als de ingang Load=0 en de ingang Din=0 of 1 is en de toestand van het register is 0, dan is de volgende toestand van het register=0 en de uitgang Dout=0.

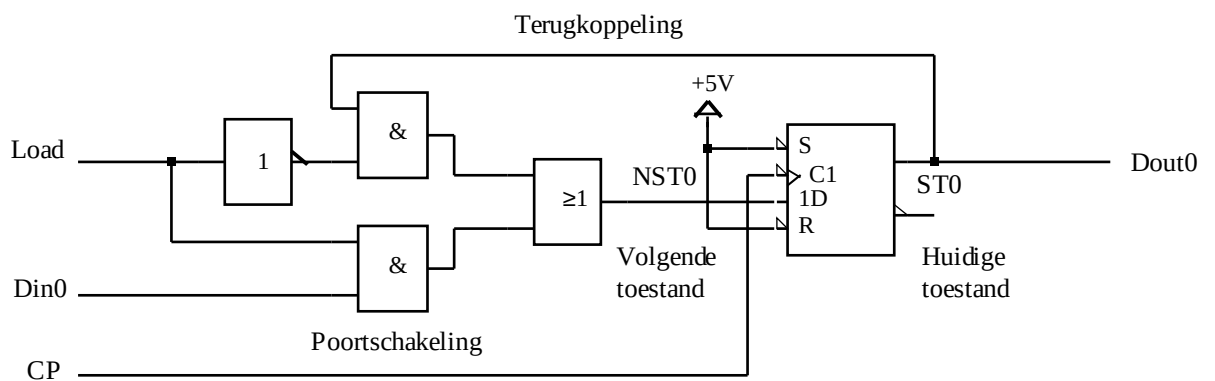
De kolom waarden van de uitgang Dout zijn gelijk aan de kolom waarden van de ingang ST0. Omdat het signaal ST0 aanwezig is als uitgangssingnaal Q van het register, is de uitgang Dout=Q.

Met behulp van het Karnaughdiagram van de waarheidstabel wordt de geminimaliseerde Booleaanse vergelijking NST0 van de combinatorische schakeling gevonden.

De gevonden combinatorische schakeling vinden we terug in figuur 11.2.



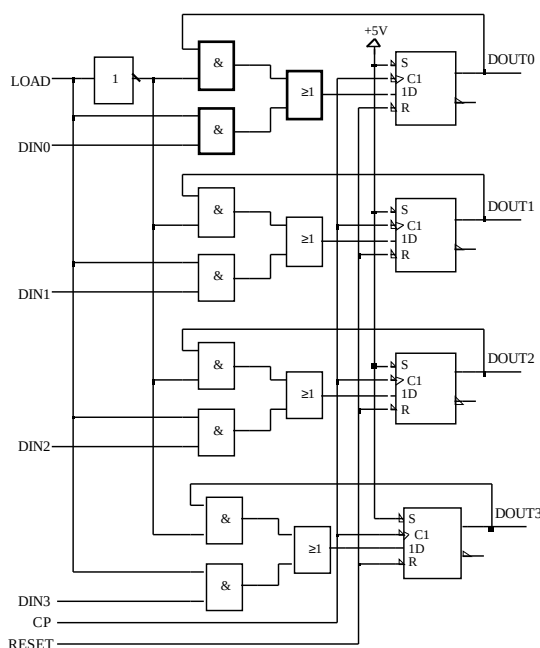
figuur 11.1



figuur 11.2

Door nu gebruik te maken van het blokschema van figuur 11.1 en de geminimaliseerde vergelijking van de uitgang NST0 kan het schema van het 1-bit register getekend worden.

4-BIT PARALLEL LOAD - PARALLEL OUT REGISTER



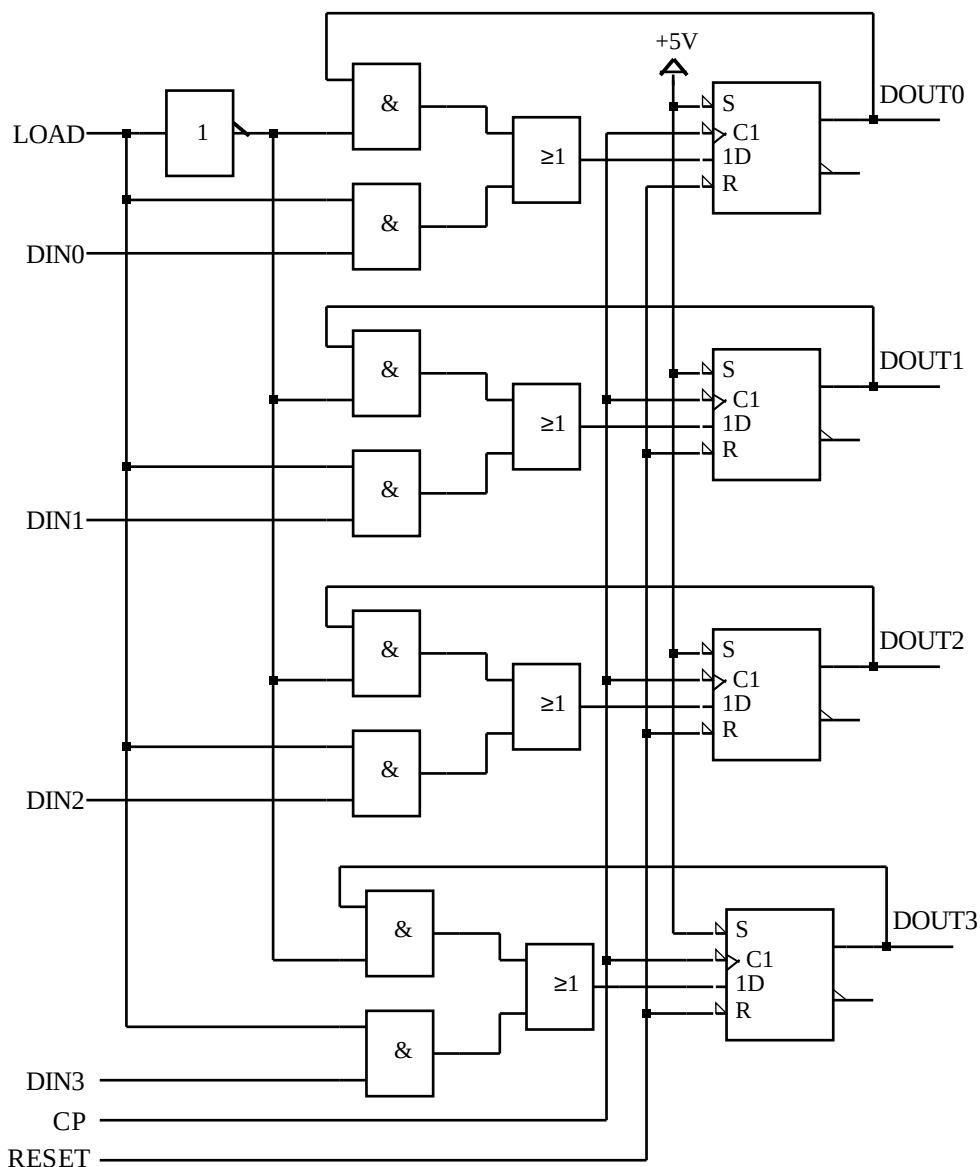
FIGUUR 11.3 4-BIT PARALLEL LOAD - PARALLEL OUT REGISTER

4-BIT PARALLEL LOAD - PARALLEL OUT REGISTER

Het schema van een 4-bit parallel load - parallel out register is samengesteld uit de parallelschakeling van vier 1-bit registers. De gemeenschappelijke signalen zijn LOAD, CP en RESET. Niet gemeenschappelijke zijn de ingangen DIN0...DIN3 en de uitgangen DOUT0...DOUT3.

De signalen LOAD, CP, en RESET zijn verbonden met de vier 1-bit registers. Daarom worden deze signalen ook wel globale signalen genoemd.

De ingangssignalen DIN0...DIN3 en de uitgangssignalen DOUT0...DOUT3 betreffen elk één specifiek 1-bit register en worden daarom vaak lokale signalen genoemd.



FIGUUR 12.1 4-BIT PARALLEL LOAD - PARALLEL OUT REGISTER

ANALYSE VAN EEN RECHTS SCHUIVEND REGISTER MET D-FF'S.

Het rechtsschuivend register in zijn eenvoudigste uitvoering, kan de 'schuif rechts' of 'shift right' operatie en de 'niet verander' of 'hold' operatie uitvoeren.

De besturing van de operatie wordt uitgevoerd door de besturingsingang SH/HOLD.

Als het signaal SH/HOLD=1, wordt op iedere positieve flank van het signaal CP de schuifoperatie uitgevoerd.

Is het signaal SH/HOLD=0, dan verandert de inhoud van het register niet.

Tijdens de schuifoperatie wordt de binaire informatie aanwezig op de ingang DIN, in het ritme van het kloksignaal CP serieel in het register geschoven. Tegelijkertijd verschijnt de binaire informatie die in het register aanwezig is, bit voor bit, op de uitgang van de laatste flipflop

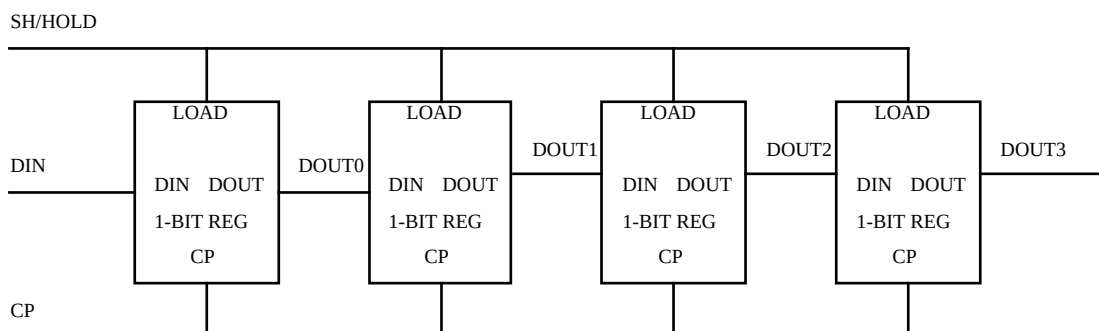
De laatst ingeschoven informatie wordt ook als laatst uitgeschoven. Dit proces wordt ook wel 'last in, last out' genoemd.

Merk op dat van de inhoud van het register één plaats naar rechts verschoven wordt, de binaire waarde van de inhoud vermenigvuldigd wordt met twee.

Om een 4-bit rechtsschuivend register te realiseren, wordt uitgegaan van het 1-bit register.

Het blokschema van figuur 13.1 is getekend hoe de 1-bit registers serieel, als een ketting, met elkaar verbonden zijn.

De binaire informatie wordt bij de operatie 'schuif rechts' op elke positief gaande flank van het signaal CP één plaats naar rechts geschoven. Immers alle flipflops zijn met hetzelfde kloksignaal CP verbonden. De uitgang van elke flipflop is verbonden met elke volgende flipflop waardoor de toestand van elke flipflop de volgende toestand van de volgende flipflop bepaalt. Het informatiebit van de laatste flipflop staat gedurende één klokperiode ter beschikking op de uitgang DOUT3. Zodra de periodetijd verstreken is, gaat dit informatiebit verloren.

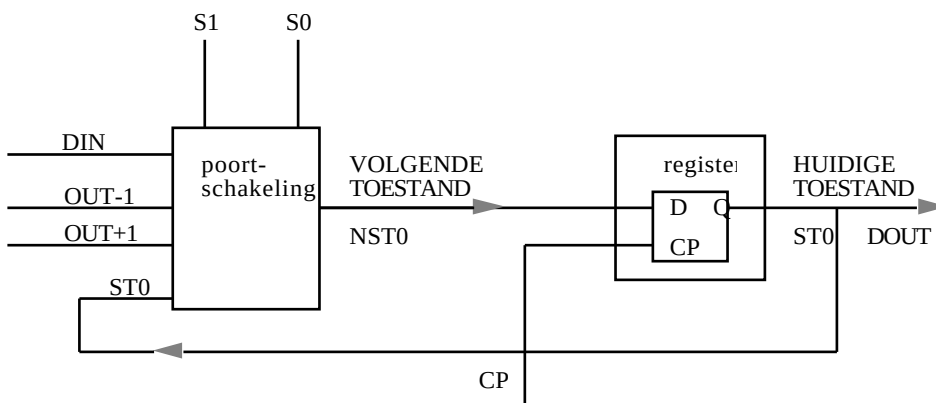


Opgave 13.1

Teken het schema van een rechts schuivend register door gebruik te maken van bovenstaand blokschema.

Ontwerp van een universeel bidirectioneel schuifregister.

Het universeel bidirectioneel schuifregister combineert de mogelijkheden van het parallelle register, en van het seriële register. Het blokschema is gegeven in figuur 14.1.



figuur 14.1

Een bidirectioneel register beschikt over de mogelijkheden om de informatie in het register, zowel naar rechts als naar links te schuiven. Merk op dat de operatie van het één plaats naar links schuiven, identiek is aan de rekenkundige operatie van het delen van het binaire getal in het register met de factor 2.

Het rechtsschuivend register werd verkregen door de uitgang van het 1-bit register met het volgnummer N, te verbinden met de ingang van de 1-bit register met het volgnummer N+1.

Op gelijksoortige wijze wordt een linksschuivend register verkregen door de uitgang van 1-bit register module N+1 te verbinden met de ingang van de 1-bitregister module N.

De schuiflinks, schuifrechts, parallel laad en verander niet operaties moeten worden bestuurd door besturingsingangen. Om de vier operaties te kunnen selecteren, zijn minimaal twee ingangen S0 en S1 nodig.

De overige ingangen zijn:

- de ingang CP waarop het kloksignaal wordt aangesloten en waarmee alle operaties gesynchroniseerd worden.
- asynchrone RESET die het mogelijk maakt om alle bits van het register op 0 te zetten, onafhankelijk van het signaal CP.

S1	S0	in.log.waarde		Mode	OPERATIE	RESET
L	L	0	0	M0	inhibit	1
L	H	0	1	M1	schuif rechts	1
H	L	1	0	M2	schuif links	1
H	H	1	1	M3	laad parallel	1
X	X	X	X	X	op nul zetten	0

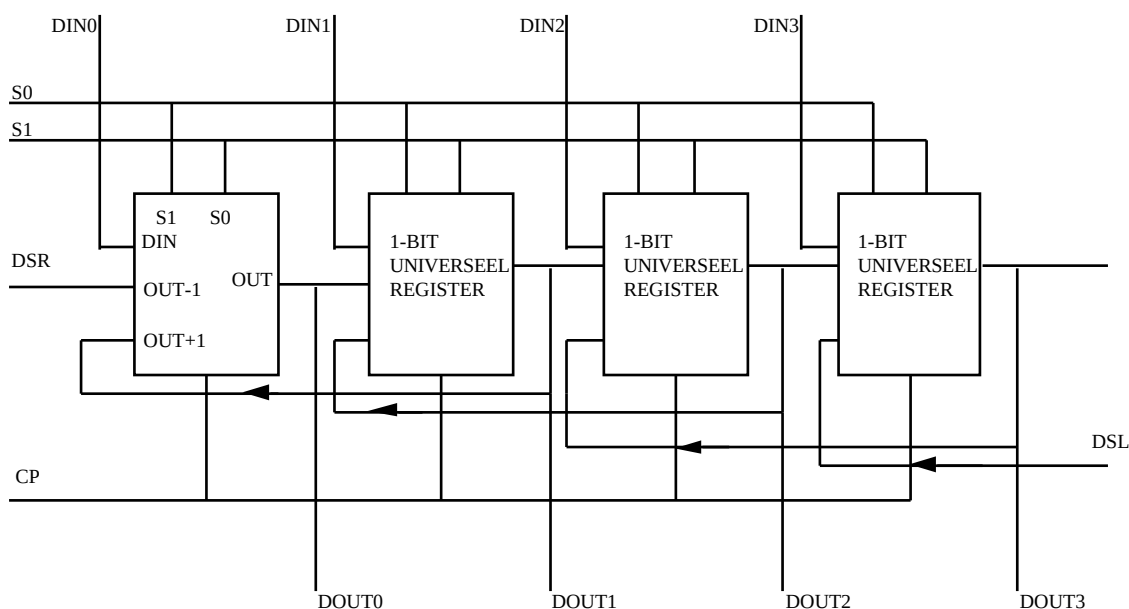
Bovenstaande functietabel toont de operaties en hun relaties. Merk op dat we hier te doen hebben met een asynchrone RESET.

De vijf synchrone ingangen van figuur 14.1 hebben de volgende functies:

- de ingang S0 en S1 selecteren de operatie
- de OUT-1 is verbonden met de uitgang van het voorgaande bit en maakt de operatie 'schuif rechts' mogelijk
- de ingang OUT+1 is verbonden met de uitgang van het volgende bit en maakt de operatie 'schuif links' mogelijk
- de ingang DIN maakt het mogelijk de operatie 'laad parallel' uit te voeren
- de operatie 'verander niet' wordt mogelijk door de terugkoppeling van het signaal STO.

De asynchrone RESET blijft buiten beschouwing gedurende de synthese van het synchrone deel van de functie en zal later tijdens de realisatiefase aan de schakeling worden toegevoegd.

Figuur 15.1 geeft het gedetailleerde blokschema van een vier-bit universeel schuifregister.



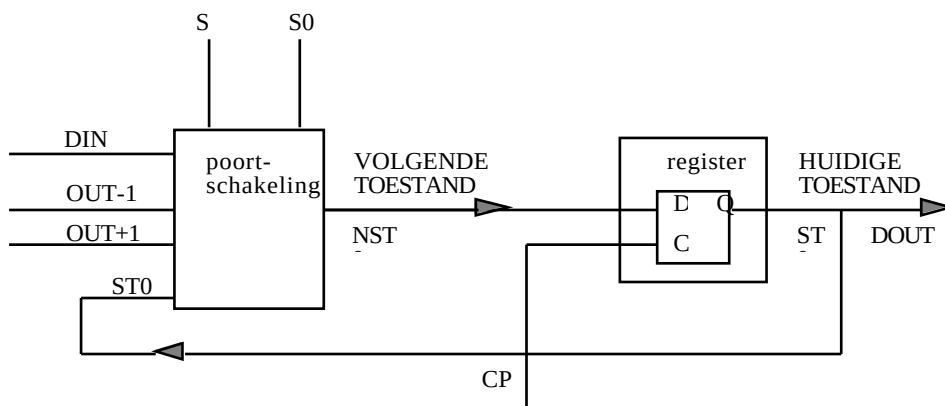
FIGUUR 15.1

De synchrone ingangen van dit 4-bits universeel schuifregister zijn:

- de ingangen S0 en S1 voor het selecteren van de operatie
- de ingang DSR (Data Shift Right) is de seriële data ingang en kan verbonden worden met de laatste uitgang DOUT van een voorgaande 4-bits universeel register, deze maakt de operatie 'schuif rechts' mogelijk
- De ingang DSL (Data Shift Left) is de seriële data ingang. Deze ingang kan verbonden worden met de eerste uitgang DOUT1 van een volgend 4-bits universeel register en maakt de operatie 'schuif links' mogelijk
- de ingang DIN0..DIN3 maakt het mogelijk de operatie 'laad parallel' uit te voeren.

Op de uitgangen DOUT0..DOUT3 is de data van het vier bits register opgeslagen.

TOESTAND	INGANGEN						UITGANGEN	
	S1	S0	OUT-1	OUT+1	DIN	STO	NSTO	OUT
T0	0	0	X	X	X	0	0	0
T0	0	1	0	X	X	0	0	0
T0	0	1	1	X	X	0	1	0
T0	1	0	X	0	X	0	0	0
T0	1	0	X	1	X	0	1	0
T0	1	1	X	X	0	0	0	0
T0	1	1	X	X	1	0	1	0
T1	0	0	X	X	X	1	1	1
T1	0	1	0	X	X	1	0	1
T1	0	1	1	X	X	1	1	1
T1	1	0	X	0	X	1	0	1
T1	1	0	X	1	X	1	1	1
T1	1	1	X	X	0	1	0	1
T1	1	1	X	X	1	1	1	1



figuur 16.1

Als **S1=0 en S0=0**, dan is de operatie '**verander niet**' gekozen en mag de toestandswaarde niet veranderen. Anders gezegd: de volgende toestandswaarde moet gelijk zijn aan de huidige toestandswaarde. Dit wordt bereikt door de waarde STO van de eigen uitgang OUT de volgende toestand te laten bepalen.

Als **S1=0 e, S0=1**, dan is de operatie '**schuif rechts**' gekozen en moet de volgende toestandswaarde gelijk worden aan de waarde van de ingang OUT-1.

Als **S1=1 en S0=0**, dan is de mode '**schuif links**' gekozen en moet de volgende toestandswaarde gelijk worden aan de waarde van de ingang OUT+1.

Als **S1=1 en S0=1**, dan is de mode '**laad parallel**' gekozen en moet de toestandswaarde gelijk worden aan de waarde van de ingang DIN.

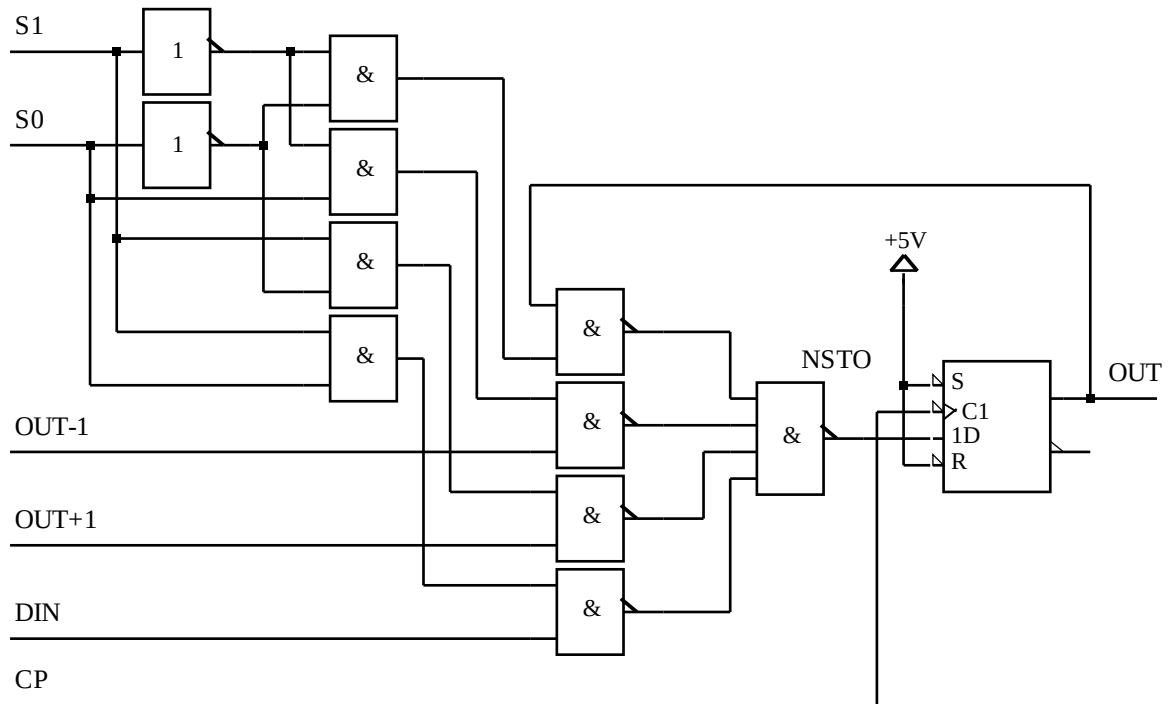
Al deze mogelijkheden zijn dan weergegeven in bovenstaande waarheidstabel.

Uit de waarheidstabel blijkt dat de Booleaanse vergelijking van de uitgang OUT gelijk is aan de huidige toestand STO. Dit betekent dat de uitgang van de DFF., die de toestand STO bevat, direct het uitgangssignaal OUT vormt. Na minimalisatie van de Karnaughkaart is

$$NSTO = S1.S0.DIN + S1./S0.OUT+1 + /S1.S0.OUT-1 + /S1./S0.STO$$

$$NSTO = S1.S0.DIN + S1./S0.OUT+1 + /S1.S0.OUT-1 + /S1./S0.STO$$

Het schema van het 1-bit universeel register van figuur 17.1 bevat bovenstaande Booleaanse vergelijking. Ga dit zelf na door gebruik te maken van de regel van de Morgan.



Figuur 17.1

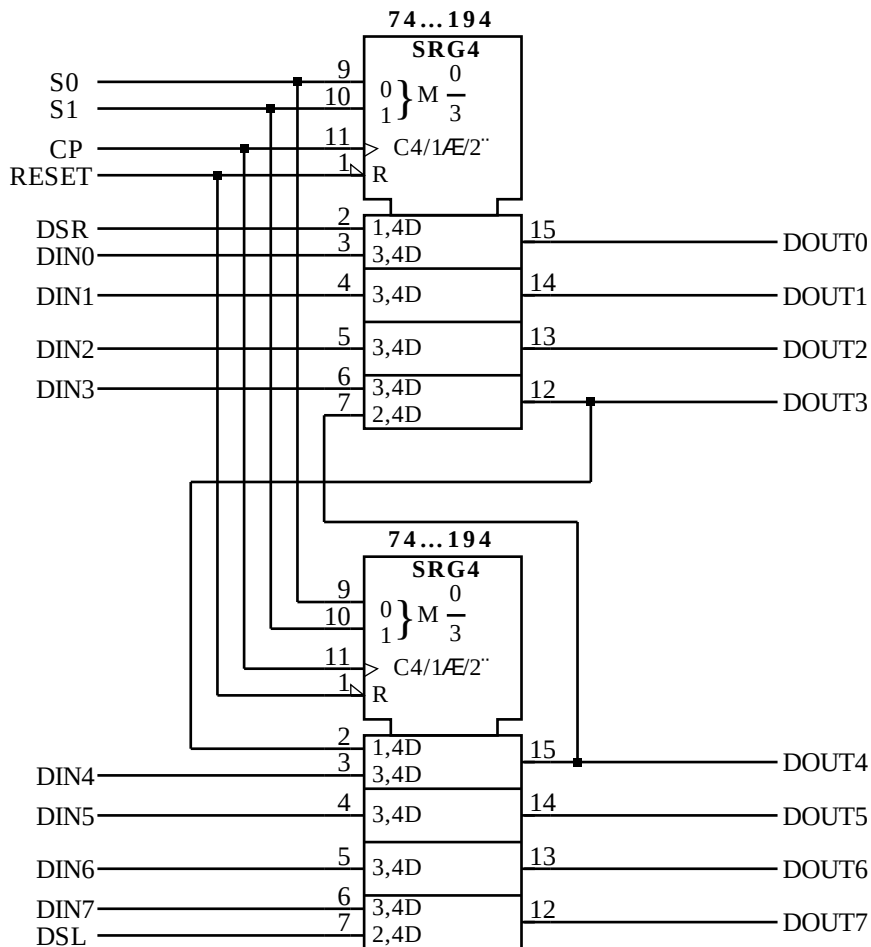
In het schema zijn de vier logische deelprodukten van S0 en S1 weergegeven door een volledige 2 naar 4 decoder. Deze vier deelprodukten bepalen de stand van de 4 naar 1 multiplexer. De stand van de multiplexer bepaalt welke ingang verbonden is met de D ingang van de DFF..

Oefening 17.1

Teken het schema van een 4-bits universeel bidirectioneel schuifregister met asynchrone reset door gebruik te maken van bovenstaande Booleaanse vergelijking.

8-Bits universeel bidirectioneel schuifregister met de 74...194.

Figuur 18.1 is het schema van twee 4-bits universeel schuifregister die samen één 8-bits universeel schuifregister. In plaats van de uit losse poorten en flipflops opgebouwde schakeling is gebruikt, is gebruik gemaakt van twee geïntegreerde schakelingen die elk één 4-bits universeel register bevat. Afgezien van de registerbreedte beschikt deze schakeling over dezelfde eigenschappen als het 4-bits universeel schuifregister. Door de structuur van het 1-bit universeel register kunnen universele schuifregisters samengesteld worden van iedere gewenste breedte.



figuur 18.1

In voorgaande ontwerpmethoden zijn drie niveau's te herkennen.

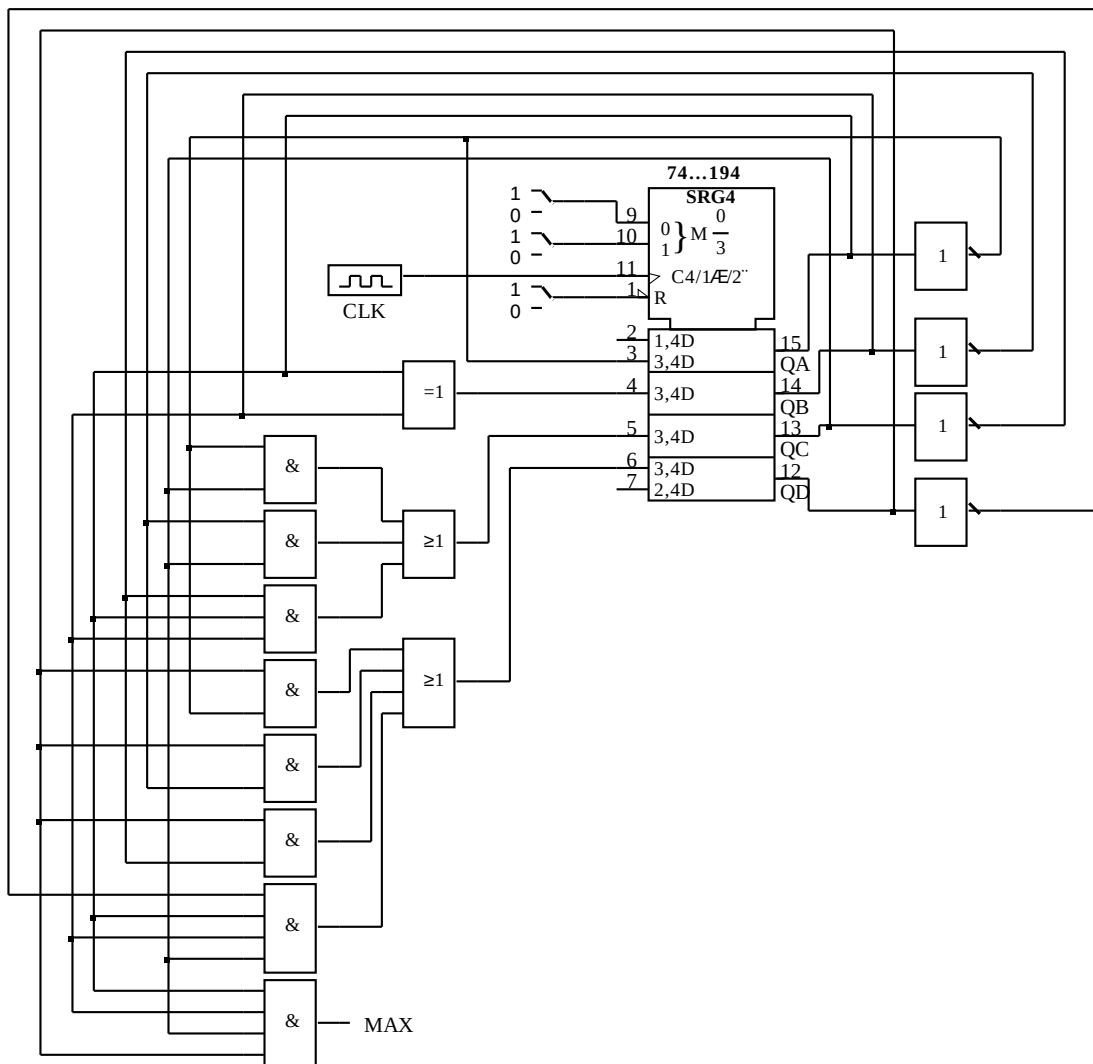
Het **eerste niveau** geeft een beschrijving van de functionaliteit van een universeel schuifregister, ongeacht het aantal bits.

Het **tweede niveau** is het 4-bits universeel schuifregister zoals in opgave 17.1.

Het **derde niveau** is het ontwerp van het 1-bit universeel register gerealiseerd in figuur 17.1.

Het derde niveau is het laagste niveau. Het gebruik van onderling afhankelijk en onder elkaar geplaatste niveaus waarbij elk lager niveau een kleinere subfunctie met meer details bevat, wordt de hiërarchische ontwerpmethodiek genoemd. Soms wordt deze ontwerpmethodiek zeer intensief toegepast om grote complexe digitale systemen, snel, betrouwbaar en zo eenvoudig mogelijk te ontwerpen.

Voorbeeld van een synchrone 16 deler opgebouwd met een 4-bits bidirectioneel schuifregister in de laad parallel-mode.



Vergelijkingen van het combinatorisch circuit.

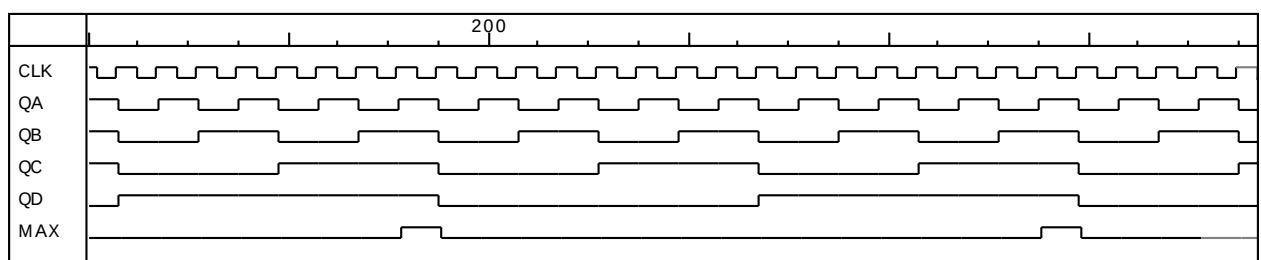
$$DA = \sqrt{QA}$$

$$DB = \sqrt{QA \cdot QB} + QA \cdot \sqrt{QB}$$

$$DC = \frac{1}{Q_A} \cdot Q_C + \frac{1}{Q_B} \cdot Q_C + Q_A \cdot Q_B \cdot \frac{1}{Q_C}$$

$$DD = \frac{1}{Q_A} \cdot Q_D + \frac{1}{Q_B} \cdot Q_D + \frac{1}{Q_C} \cdot Q_D + \frac{Q_A \cdot Q_B \cdot Q_C}{Q_D} \text{ en}$$

$$\text{MAX} = Q_A \cdot Q_B \cdot Q_C \cdot Q_D$$



ONTWERP VAN EEN SERIELE OPTELLER.

Bij deze opteller is het noodzakelijk dat de bits van de getallen A en B paarsgewijs en synchroon op ritme van het kloksignaal, aan de ingangen worden aangeboden.

We gebruiken twee rechtsschuivende registers A en B. Op het ritme van het kloksignaal worden de bits van de getallen A en B, opgeslagen in deze registers, en paarsgewijs bit voor bit aan de seriële opteller aangeboden.

Doordat de overdracht van het voorafgaande bitpaar overgedragen moet worden naar het volgende bitpaar, is een geheugenelement of "carry store" nodig om deze overdracht gedurende een klokperiode te onthouden.

Analyse van de seriële opteller.

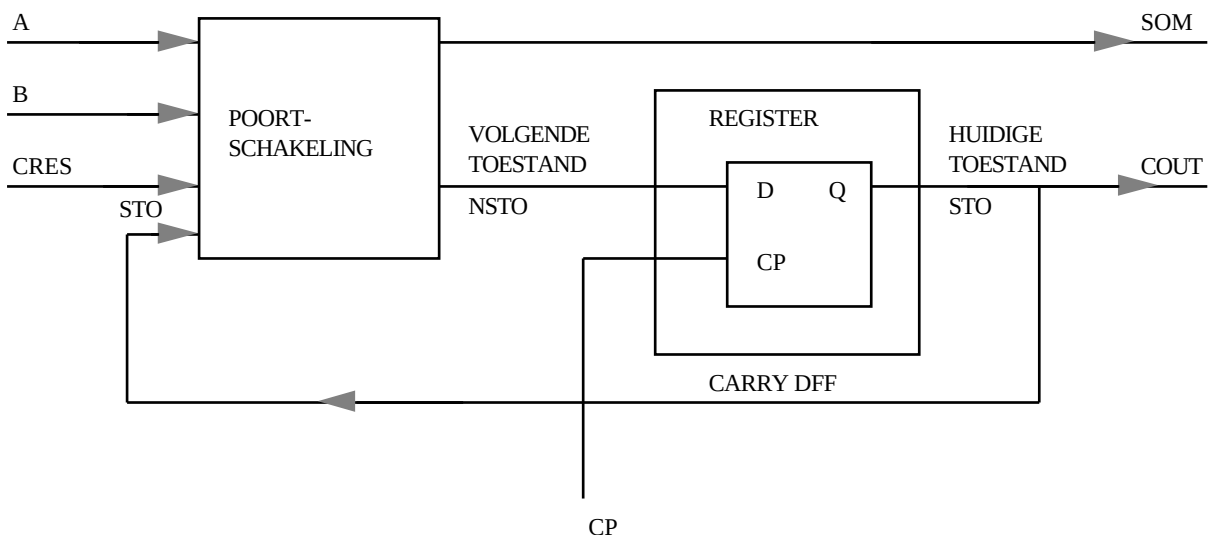
Het optelproces moet beginnen met het minst belangrijke bitpaar en de stand 0 van het overdrachtgeheugenelement.

Het optelproces eindigt na de optelling van het meest belangrijke bitpaar. De stand van het overdrachtgeheugenelement is dan gelijk aan de waarde van het overdracht van de laatste optelling.

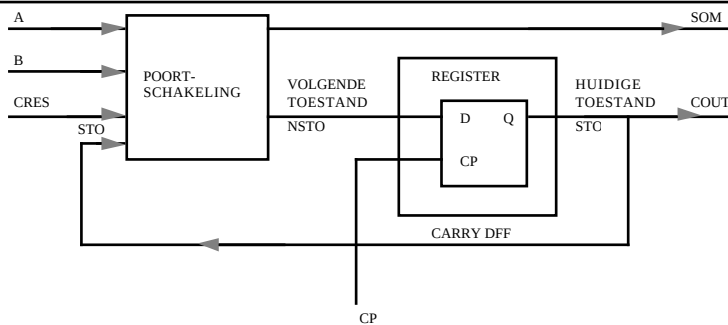
De seriële opteller heeft twee ingangen A en B waarop de uitgangen van registers A en B worden aangesloten, één ingang CRES en één uitgang SOM.

Het signaal op de ingang CRES brengt de stand van de "carry store" flipflop in de stand 0, zodat de optelling van het eerste bitpaar uitgevoerd wordt met de overdracht 0. Bit voor bit verschijnt dan de som van de twee binaire getallen aanwezig in register A en register B op de uitgang SOM. De som van elke bitpaar is gevormd door de waarden van de op de ingang A en B aanwezige bits en de CARRY van de voorafgaande optelling.

De overdracht van de optelling van elk bitpaar wordt bij een seriële opteller opgeslagen in een flipflop. Op deze manier kan de volgende bitpaaroptelling beschikken over de overdracht van de vorige optelling. Hierdoor is de seriële opteller een sequentiële schakeling.



figuur 20.1

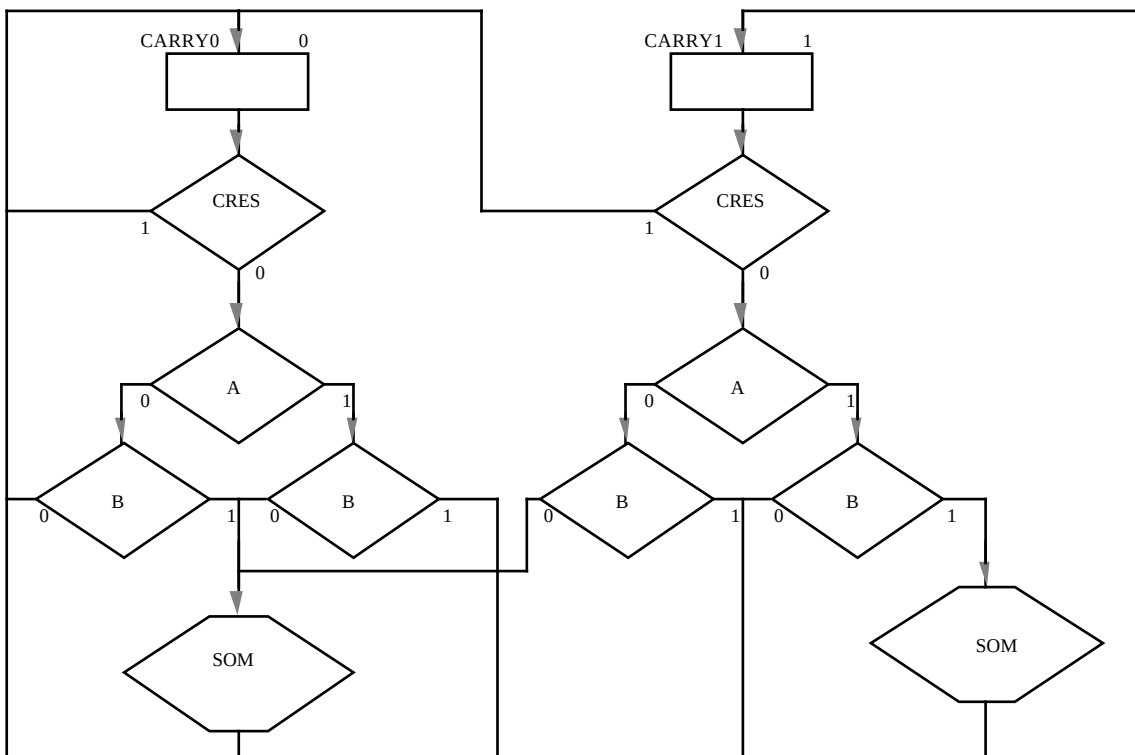


figuur 21.1

In het blokschema van figuur 21.1 is de structuur van de sequentiële schakeling die de seriële opteller vormt getekend.

De schakeling is opgebouwd uit een combinatorische functie, die met uitsluitend poorten gerealiseerd kan worden, en een flipflop als geheuelement waarvan de uitgang teruggekoppeld wordt naar een ingang van de combinatorische schakeling.

De combinatorische functie bepaalt de SOM en de overdracht van het op te tellen bitpaar en de overdracht van de vorige optelling. De overdracht van de vorige optelling is opgeslagen in het register als het signaal STO. De nieuwe overdracht, nodig voor de volgende optelling, is beschikbaar als het signaal NSTO



FIGUUR 21.2

Het toestandsdiagram van figuur 21.2 is zo opgesteld dat aan de volledige functionaliteit van de seriële opteller voldaan wordt.

De toestand van de overdracht van de vorige optelling = 0 wordt symbolisch weergegeven door het toestandblok met de naam CARRY0 en de bijbehorende waarde 0. Is er een overdracht gegenereerd tijdens de vorige optelling dan bevindt de serial adder zich in het toestandblok CARRY1 met de waarde 1.

Als de ingang CRES=1 (carry reset) zal ongeacht met welk toestandblok begonnen wordt het pas eindigen in toestand CARRY0. Door deze constructie zal de ingang CRES zich gedragen als een **synchrono nulzet commando** van de opgeslagen overdracht. Dit commando wordt gegeven telkens als een volgende optelling van twee vier-bit getallen wordt gestart.

Als tijdens de toestand CARRY0 de ingangen A en B de waarde 1 hebben, loopt het pad van toestand CARRY0 naar de volgende toestand CARRY1 met de bijbehorende waarde 1. Doordat het conditionele blok SOM niet in dit pad is opgenomen verkrijgt uitgang SOM de waarde 0. Bovendien heeft door het veranderen van toestand de gegenereerde carry voor de optelling van het volgende bit de waarde 1 verkregen. In onderstaande waarheidstabel is dit pad aangeduid in regel 4.

Als de toestand CARRY1 aanwezig is, zal als de ingangen A en B de waarde 1 hebben, het pad beginnen in toestand CARRY1 en via het conditionele blok SOM eindigen in de volgende toestand CARRY1.

Hiermee is aangegeven dat de SOM de waarde 1 verkrijgt en de gegenereerde carry voor de volgende optelling de waarde 1 heeft.

Dit pad is in laatste regel van de waarheidstabel beschreven.

Merk op dat de uitgang welke beschreven is door een conditioneel blok WAAR wordt onafhankelijk van de waarde van de toestand en de ingangscondities. Daar de toestand synchroon met het kloksignaal CP verandert maar de ingangscondities onafhankelijk van het kloksignaal en daardoor asynchroon zijn is ook het uitgangssignaal aangegeven door een conditioneel blok **asynchroon**.

Elke regel in de waarheidstabel van onderstaande figuur geeft één mogelijk pad aan. Alle regels samen vormen daardoor de volledige beschrijving van in de seriële opteller opgenomen combinatorische functie.

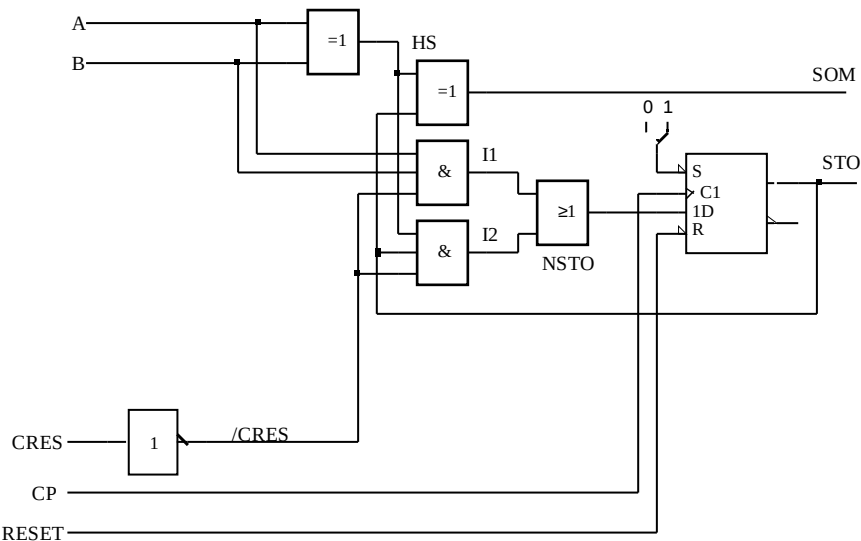
TOESTAND	INGANGEN				UITGANGEN	
	A	B	CRES	STO	NSTO	SOM
CARRY0	0	0	0	0	0	0
CARRY0	0	1	0	0	0	1
CARRY0	1	0	0	0	0	1
CARRY0	1	1	0	0	1	0
CARRY0	X	X	1	0	0	0
CARRY1	0	0	0	1	0	1
CARRY1	0	1	0	1	1	0
CARRY1	1	0	0	1	1	0
CARRY1	1	1	0	1	1	1
CARRY1	X	X	1	1	0	0

NSTO	AB	00	01	11	10
	CRES/STO				
	00	0	0	1	0
	01	0	1	1	1
	11	0	0	0	0
	10	0	0	0	0

$$NSTO = (A \text{ EXOR } B) ./ CRES.STO + A.B ./ CRES$$

SOM	AB	00	01	11	10
	CRES/STO				
	00	0	1	0	1
	01	1	0	1	0
	11	0	0	0	0
	10	0	0	0	0

$$SOM = /CRES(A \text{ EXOR } B \text{ EXOR } STO)$$



Figuur 23.1

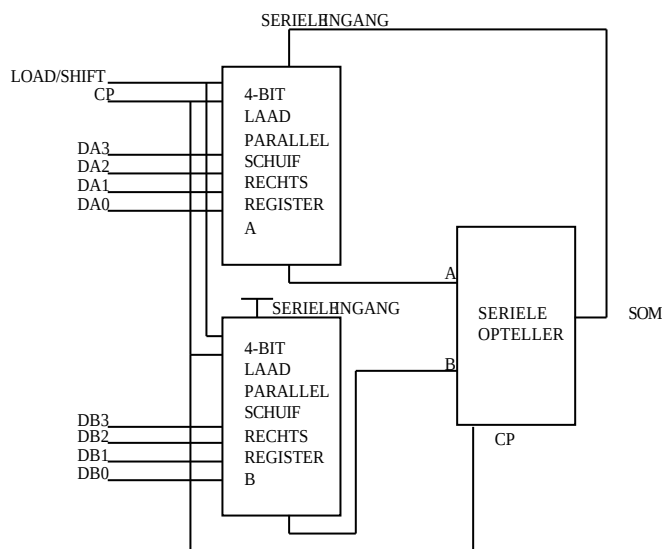
De synchrone CRES zorgt ervoor dat als elke keer begonnen wordt met een volgende seriële optelling van twee getallen, de inhoud van het overdrachtgeheugenelement 0 is. Bij de optelling van de LSD van het volgende getallenpaar, is dan verzekerd dat het "carry in" signaal een waarde 0 heeft.

Het asynchrone signaal RESET wordt alleen gebruikt om op het moment van het inschakelen van de voedingsspanning, de overdracht flipflop in de beginstand CARRY0 te brengen.

Een toepassing van de seriële opteller is weergegeven in figuur 23.2, waar de seriële opteller is opgenomen in een schakeling met twee 4-bits universeel schuifregister de 74...194.

Aan de seriële opteller worden paarsgewijs de bits van de getallen A en B aangeboden. De 4-bits getallen zijn in de registers A en B aanwezig.

De getallen A en B worden parallel geladen in de registers A en B. De registers A en B voeren daarom gedurende een periode van het klok-sig-naal CP de operatie "laad parallel" uit. Tegelijk met het laden van de getallen A en B, wordt ook de "carry store" flipflop in de stand 0 gezet.

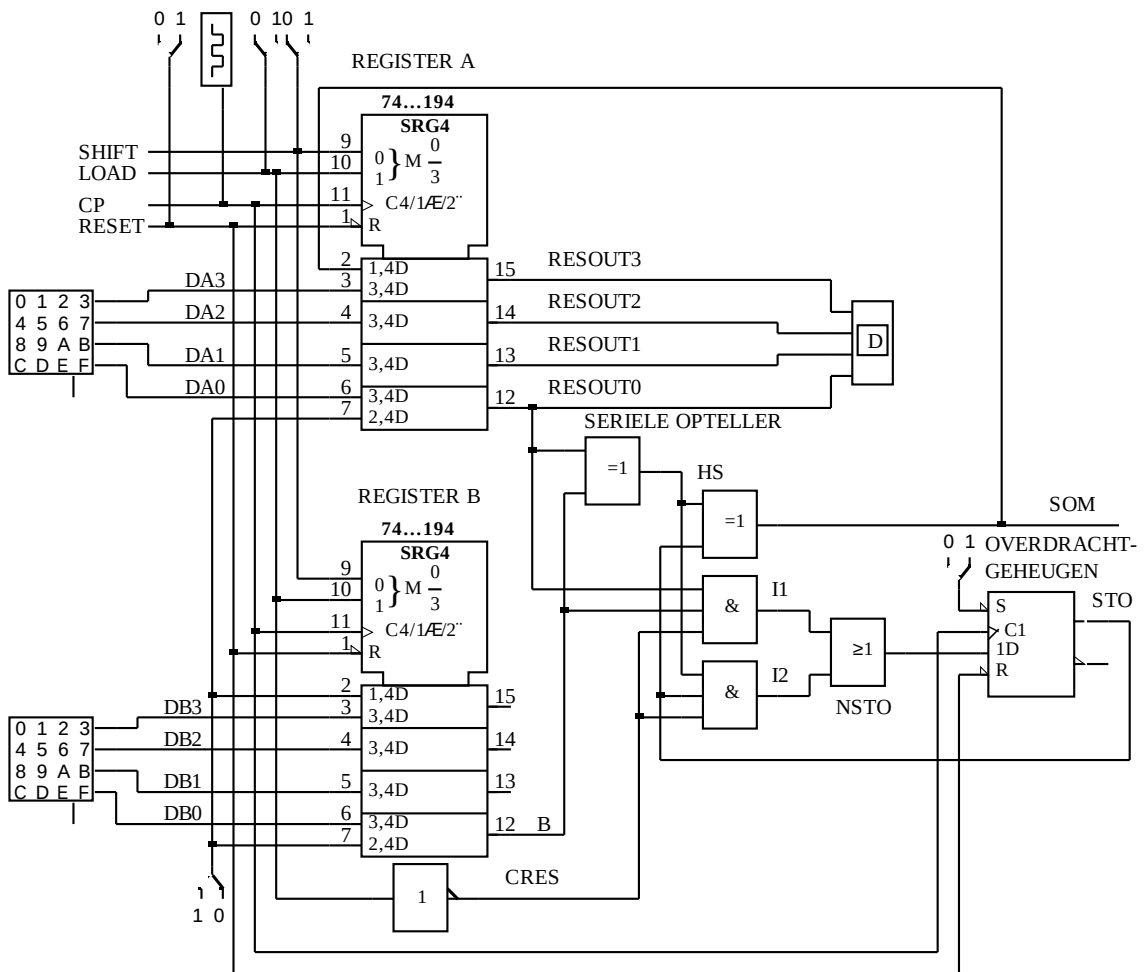


FIGUR 23.2

De uitgang SOM van de seriële opteller is verbonden met de seriële ingang van register A. Van zodra één bit van het getal A uit het register A schuift, wordt het bit van de geschoven SOM op de juist vrijgekomen plaats in het register A geschoven.

Tijdens de seriële optelling voeren beide registers de operatie "schuif rechts" uit. De seriële optelling is na vier positieve flanken van het kloksignaal CP beëindigd. De serieel gevormde SOM is parallel ter beschikking op de uitgangen RESOUT3....RESOUT0 zie figuur 24.1.

De overdracht van elke optelling wordt gedurende één klokperiode opgeslagen in de "carry" flipflop van de seriële opteller en daardoor bewaard voor de volgende optelling.



Figuur 24.1 schema 4-seriële opteller

Het schema van figuur 24.1 geeft de gedetailleerde uitwerking van de 4-bits seriële opteller, samen met de schuifregisters A en B.

Om deze optelmachine juist te laten functioneren, is een juiste besturing nodig van de signalen SHIFT en LOAD.

Deze signalen worden in een logisch systeem geleverd door synchrone besturingsschakelingen.

De eerste besturingsactie tijdens het inschakelen van de voedingsspanning, is het nulzetten van de "carry" flipflop van de seriële opteller met het signaal RESET.

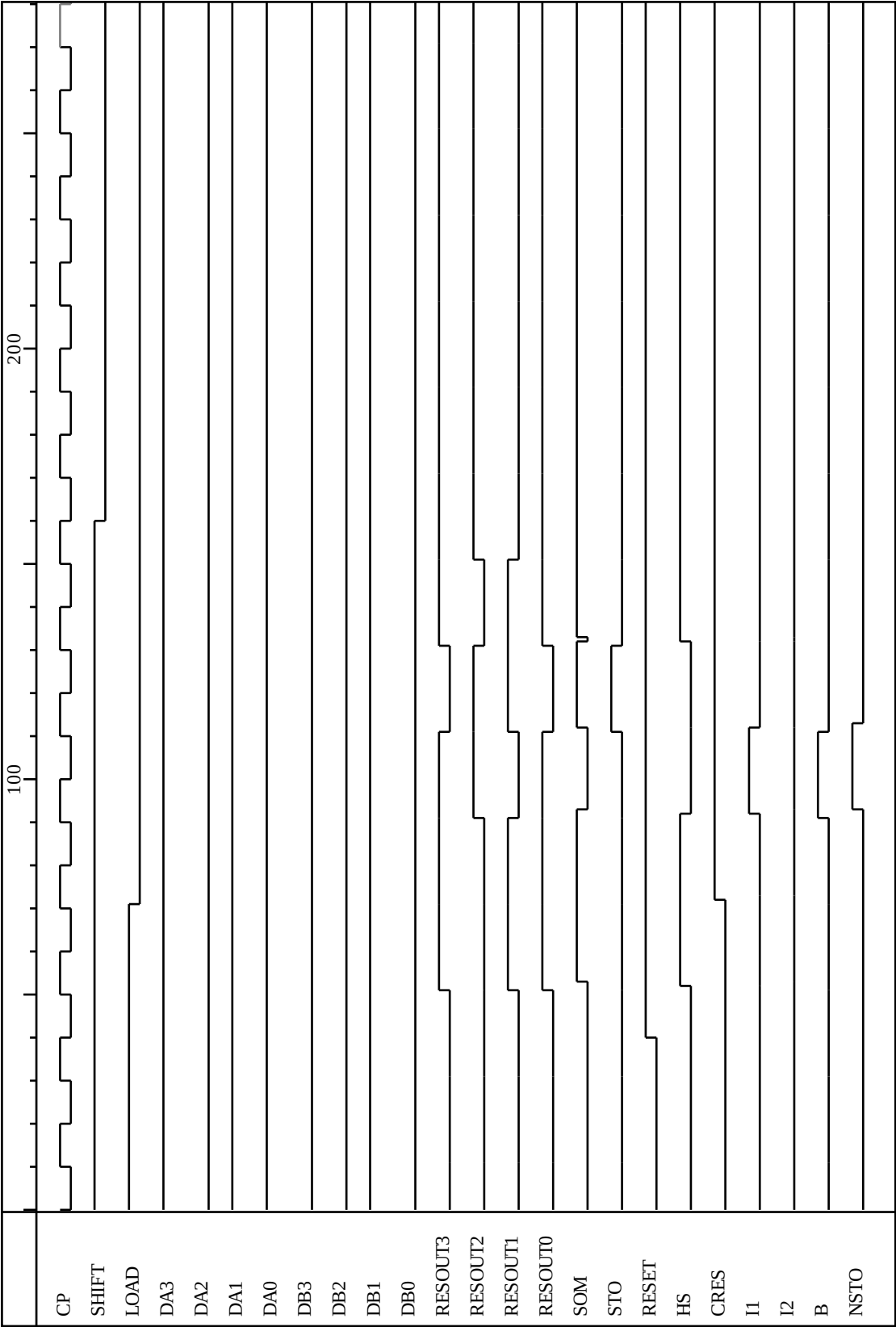
Vervolgens worden de registers A en B parallel geladen met de getallen A en B welke opgeteld worden.

Hierna voeren de registers A en B de operatie "schuif rechts" uit gedurende vier positieve flanken van het klopsignaal CP.

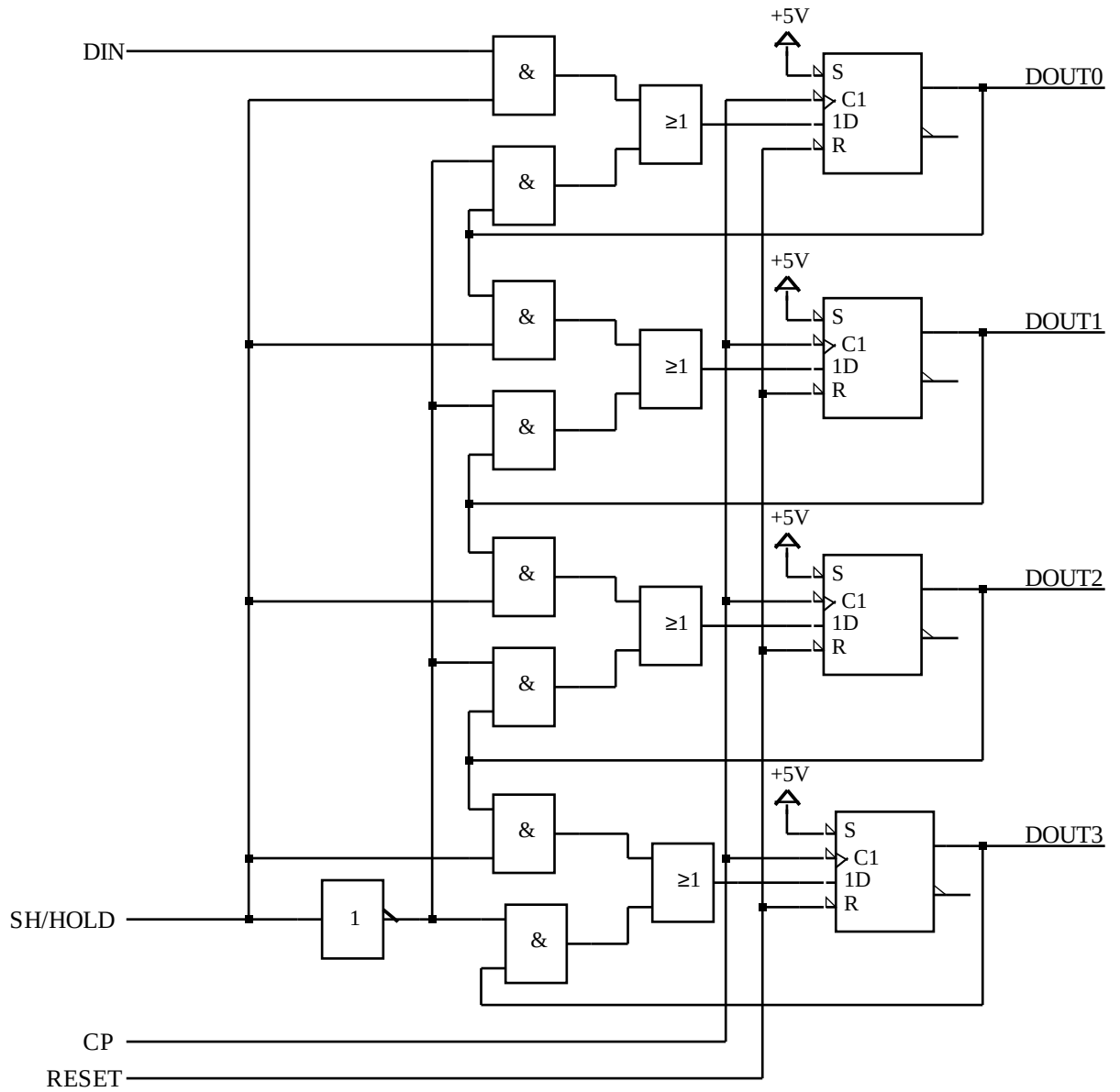
Om na de optelling van alle bits gedurende één periode van het signaal CP het resultaat van de optelling ter beschikking te hebben, voeren de registers A en B de operatie "verander niet" uit.

Hierna kan een tweede optelling plaatsvinden met dezelfde besturingscyclus. In de simulatie van figuur 25.1 wordt de volledige cyclus van de seriële optelling van de twee 4-bits getallen weergegeven.

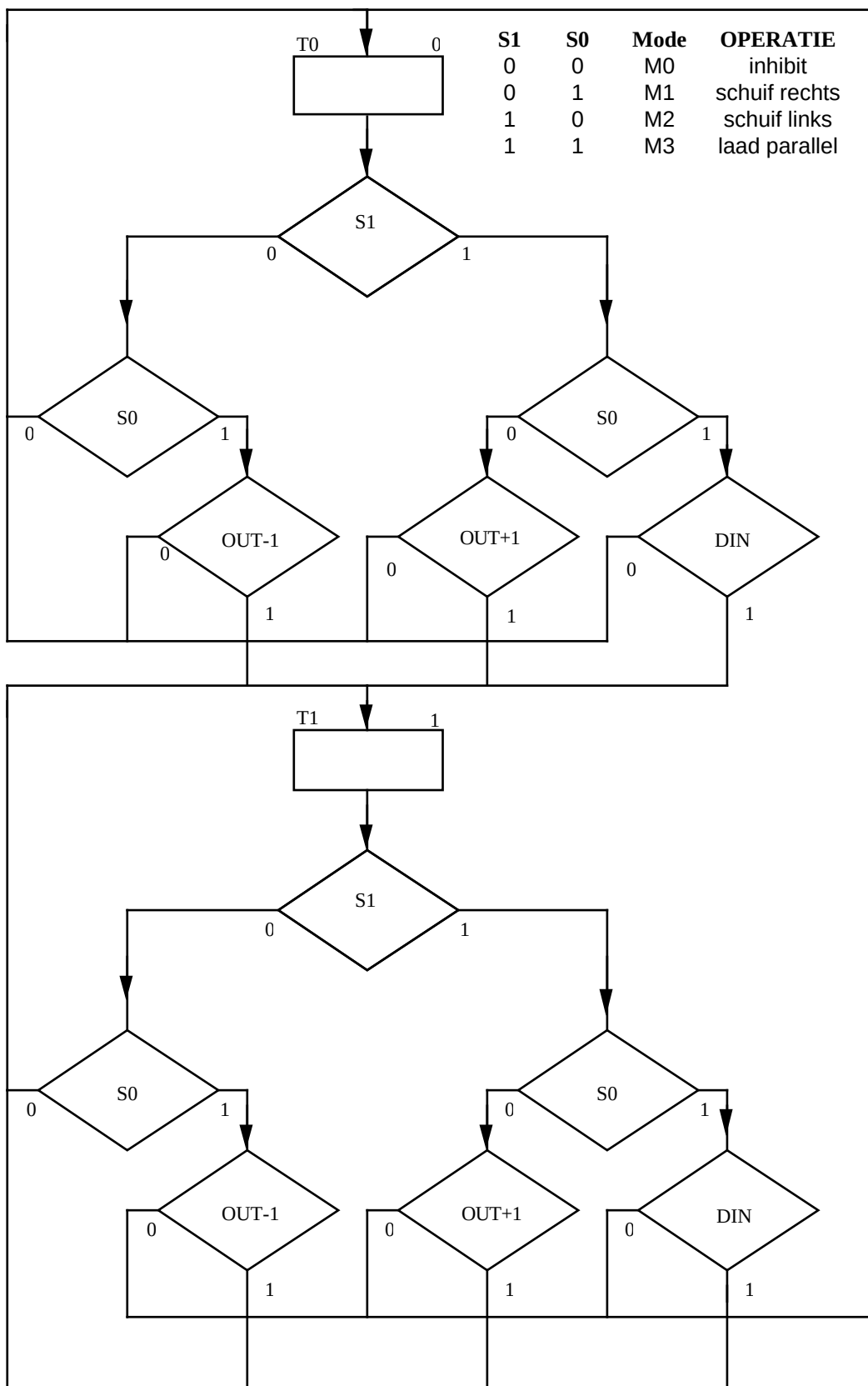
Tijdsvolgorde 4-bits seriële opteller



SCHEMA 4-BIT RECHTSSCHUIVEND REGISTER

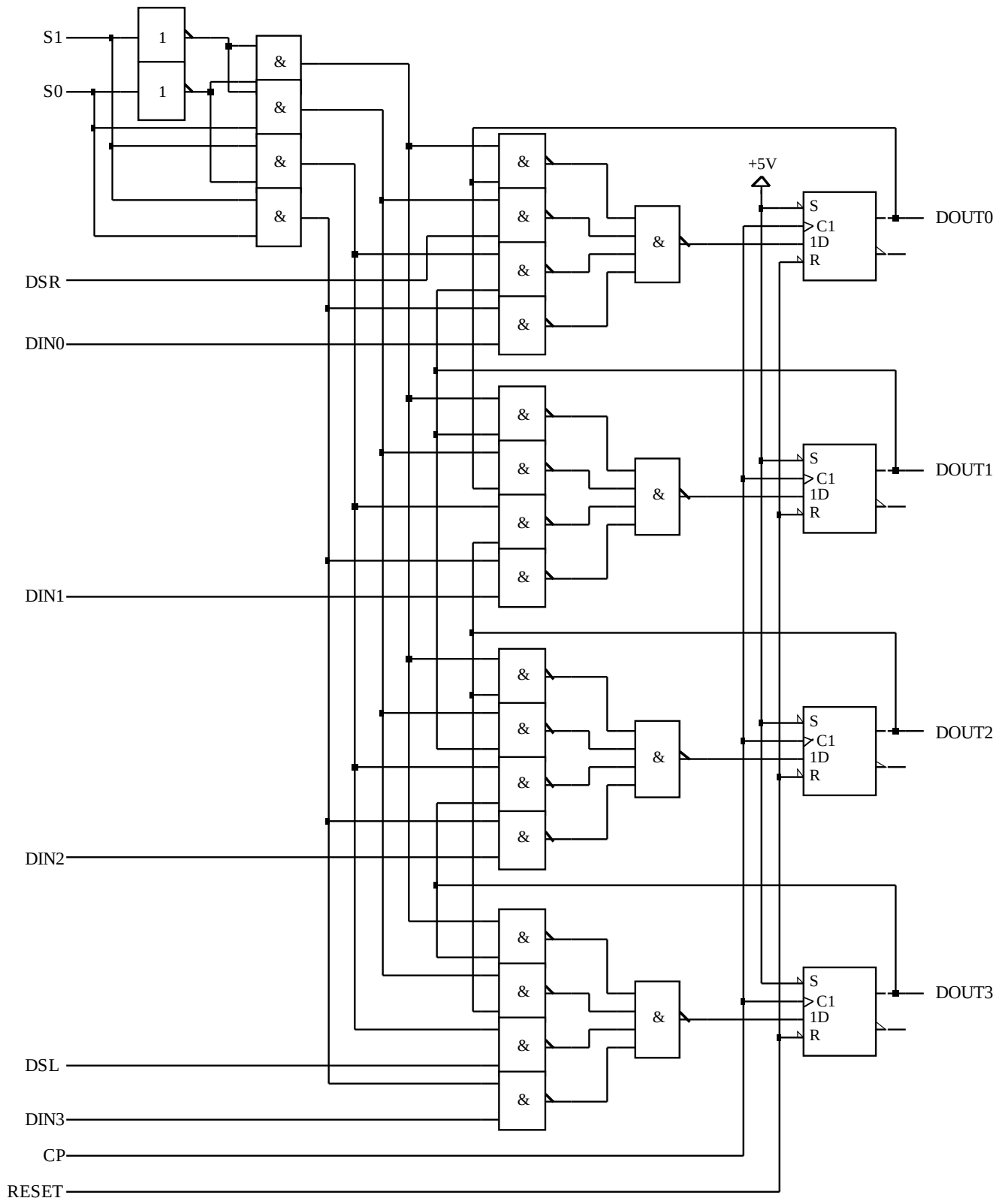


OPLOSSING 13.1



FIGUUR 27.1 TOESTANDSDIAGRAM 1-BIT UNIVERSEEL REGISTER

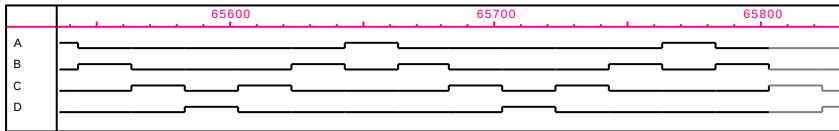
SCHEMA 4-BITS UNIVERSEEL SCHUIFREGISTER



OPLOSSING 17.1

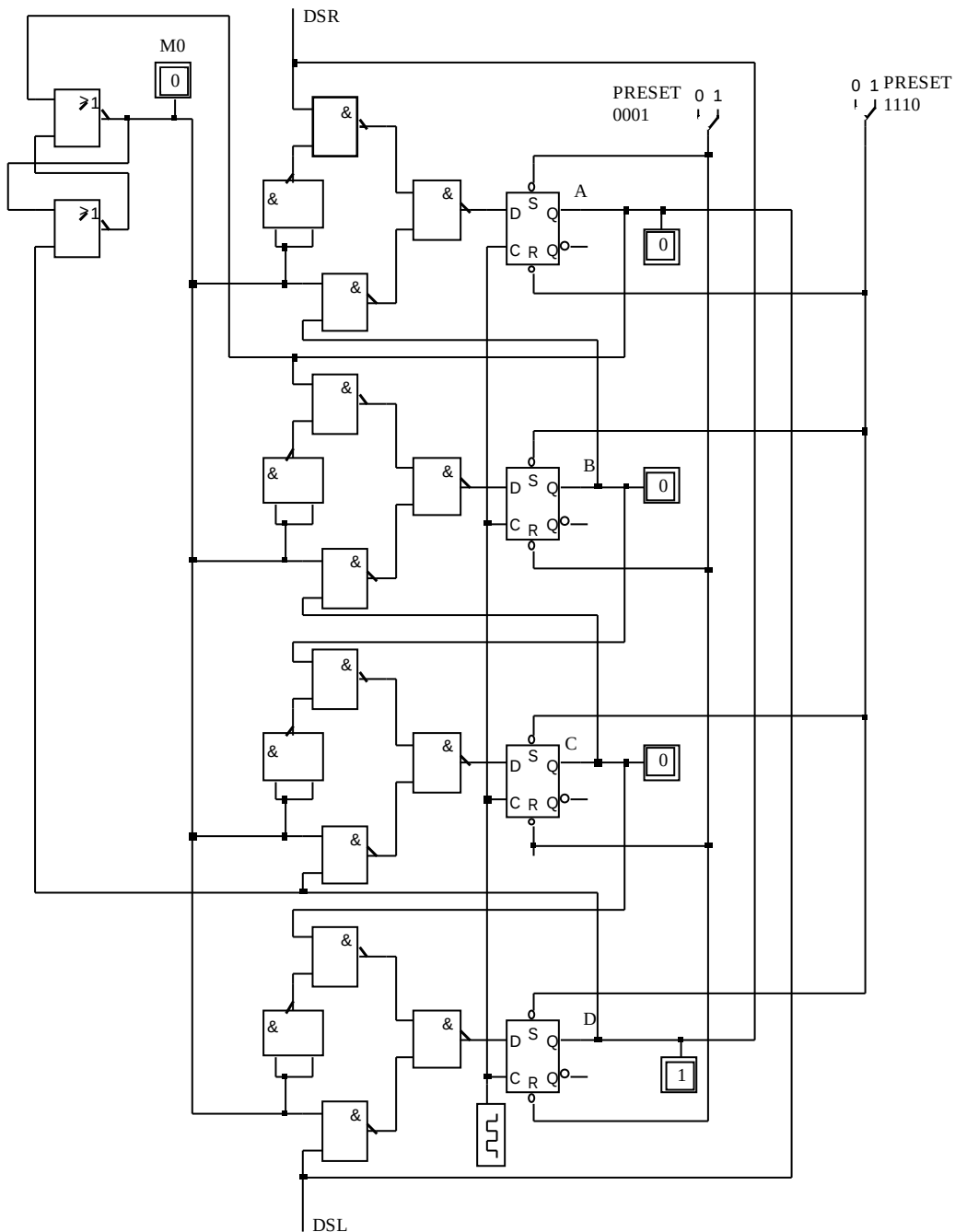
Oefening 29.1

Bouw een sequentiële schakeling die voldoet aan onderstaand timingdiagram.
Maak gebruik van een universeel schuifregister opgebouwd met D-flipflops. (7474)



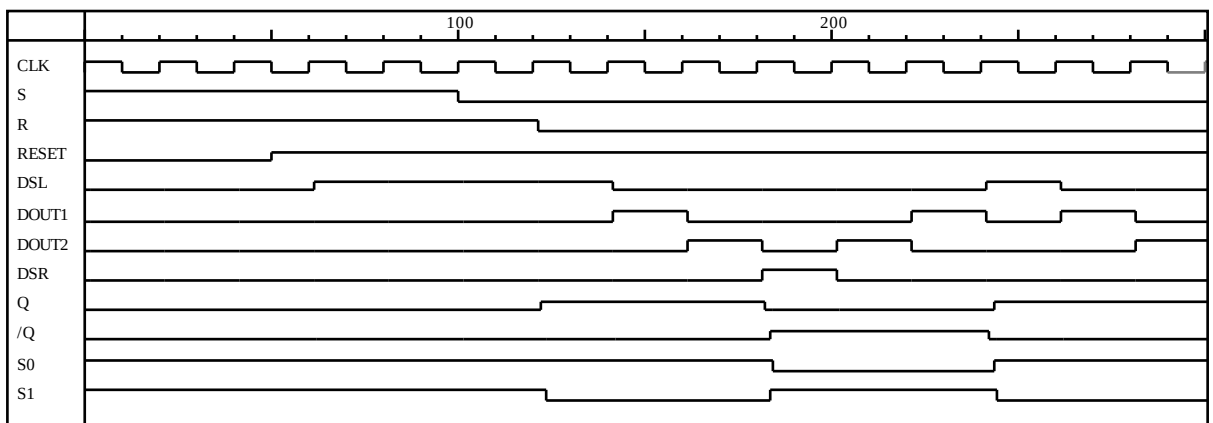
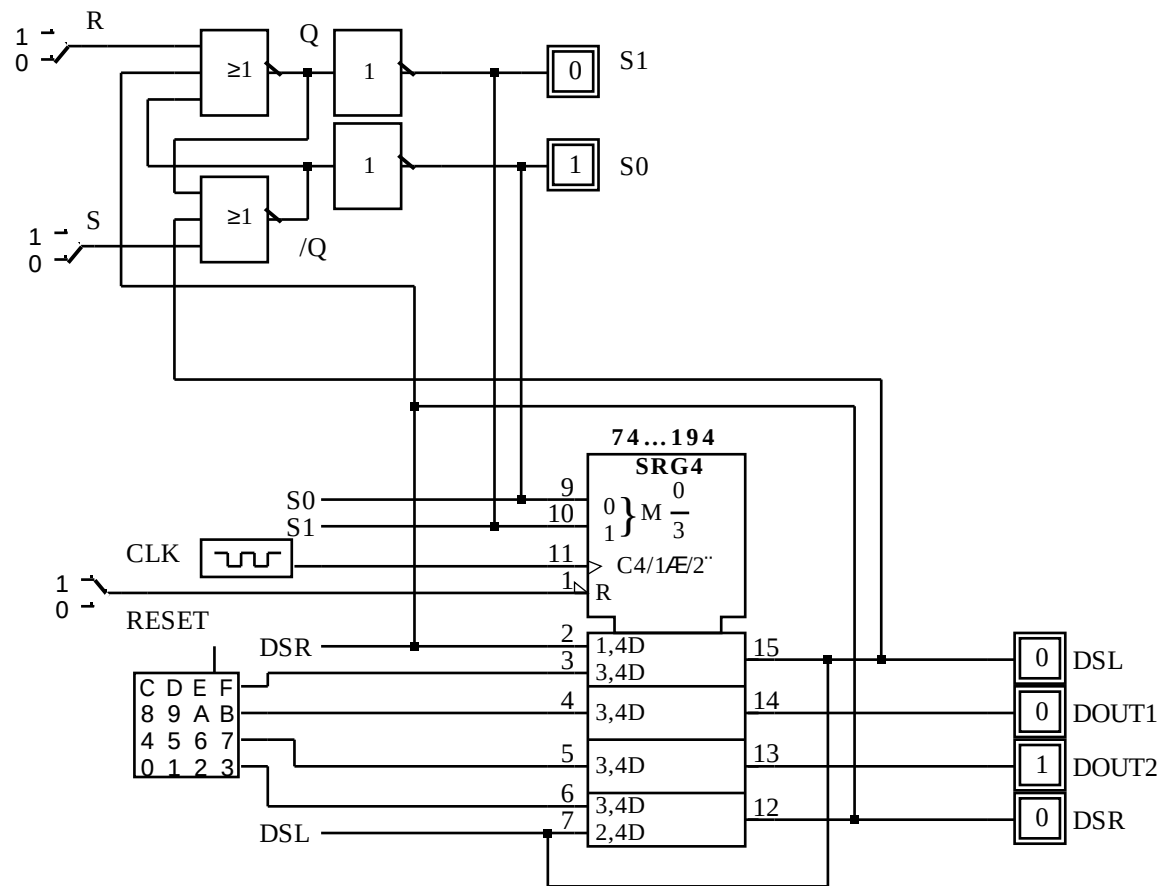
Maak gebruik van een links rechts schuivend register zoals op blz 4.

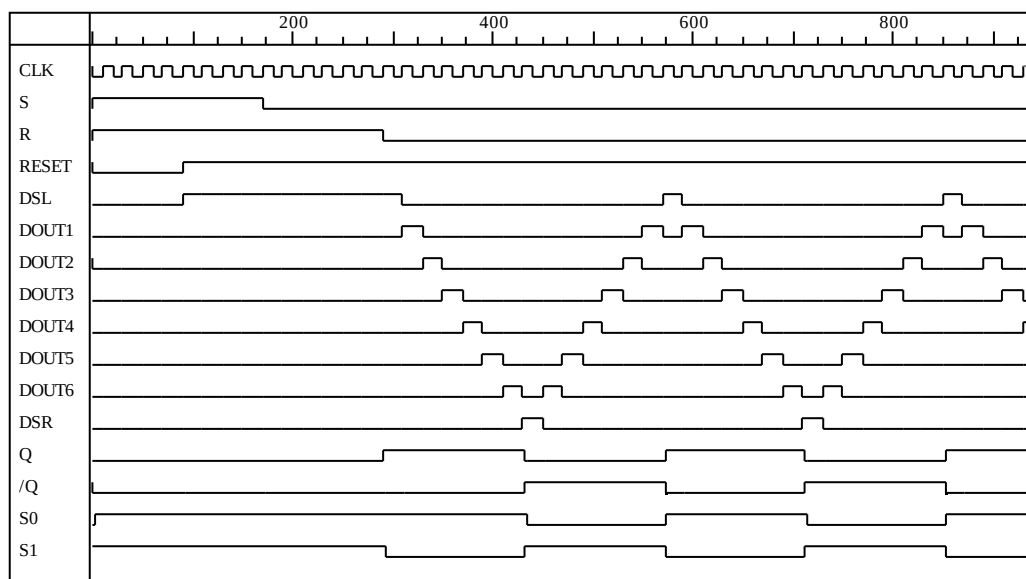
Oplossing



OPLOSSING 29.1

Dit is een toepassing op een 4-bits universeel bidirectioneel schuifregister met de 74...194, waarbij één bit voordurend van links naar rechts schuift.





Registers

Inhoud

Bufferregisters

Schuifregisters

Rechts schuivend register

Links schuivend register

Informatietoevoer

Informatie afvoer

Schuifbesturing

Toepassing van schuifregisters

Universele schuifregisters

De 7495

De 74194

Parallel load - parallel out register met D-FF's

4-Bit parallel load - parallel out register.

Analyse van een rechts schuivend register

Universeel bidirectioneel schuifregisters

Toepassingen op schuifregisters

Oefeningen