

APPENDIX over LOGISCHE SYMBOLEN
bij

DIGITALE TECHNIEK

van probleemstelling tot realisatie

deel 1



A.P. Thijssen, H.A. Vink, C.H. Eversdijk

THIJSSSEN. APPENDIX OVER LOGISCHE SYMBOLEN BIJ DIGITALE TECHNIEK
VAN PROBLEEMSTELLING TOT REALISATIE, DEEL 1

0784

1607
1607
1607

Appendix



Symbolen voor logische schakelingen (poortniveau)

A.1.	Inleiding	2
A.2.	Vorm van symbolen	3
A.3.	Functie-aanduiding	5
A.4.	De interface met de fysische wereld	6
A.5.	Signaalnamen	8
A.6.	Signaalnamen en logische niveaus	9
A.7.	Inversie en negatie	11
A.8.	Symbolen en formules I	12
A.9.	Symbolen en formules II	15
A.10.	Dualiteit en positieve en negatieve logica	19

Symbolen voor logische schakelingen (MSI-niveau)

A.11.	De afhankelijkheidsnotatie	23
A.12.	Gemeenschappelijke ingangen en uitgangen	25
A.13.	De Z-afhankelijkheid	27
A.14.	G-, V-, en N-afhankelijkheid	28
A.15.	Enkele toepassingen	31
A.16.	EN-afhankelijkheid	33
A.17.	A-afhankelijkheid	36
A.18.	C-afhankelijkheid	39
A.19.	Symbolen voor latches	41
A.20.	Symbolen voor flip-flops	43
A.21.	Symbolen voor flip-flops met direct werkende ingangen	47
A.22.	Symbolen voor flip-flops met ingangselector	48
A.23.	Symbolen voor flip-flops met enable-ingang I	50
A.24.	Symbolen voor flip-flops met enable-ingang II	55
A.25.	Symbolen voor schuifregisters	57
A.26.	M-afhankelijkheid	58
A.27.	Symbolen voor tellers I	62
A.28.	Symbolen voor tellers II	65

Bibliotheek TU Delft



C 1618984

0784
329
3

Symbolen voor logische schakelingen (poortniveau)

A.1. Inleiding

Digitale Techniek is een relatief jong vakgebied, gekenmerkt door een onstuimige groei van de technische mogelijkheden. Het proces van het ontwerpen van digitale schakelingen heeft deze groei van de technische mogelijkheden niet altijd kunnen bijhouden. Het *IEC-symbolensysteem* heeft, op componentenniveau, hieraan zeker positief bijgedragen en heeft als katalysator gewerkt op het ontwikkelen van meer systematiek in het ontwerpen, hoewel het oorspronkelijk bedoeld was als documentatiehulpmiddel.

Tot ca. 1970 was de complexiteit van logische schakelingen beperkt tot poorten en flip-flops. Voor het weergeven van componenten in tekeningen werden *vormsymbolen* toegepast, voor elk type poort een apart symbool. Deze vormsymbolen zijn o.a. gestandaardiseerd in het 76-ANSI/IEEE Y 32E normblad. Tot op de dag van heden worden deze symbolen toegepast. Zie ook figuur 5.4.

Naarmate de complexiteit en de diversiteit van de componenten voor logische schakelingen toenam en bleef toenemen, groeide de behoefte aan meer systematiek bij het weergeven van schakelingen in symbolen. Het was onmogelijk voor elk nieuw circuit een eigen vormsymbool te definiëren.

Binnen de IEC (International Electrotechnical Commission), in 1906 opgericht met als doel coördinatie en uniformering van nationale normen op elektrotechnisch gebied, houdt één van de technische commissies, TC3, zich bezig met het opstellen van internationale normen voor grafische symbolen op elektrotechnisch gebied. De werkgroep WG2 van subcommissie SC 3A bestrijkt het terrein van de tekensymbolen voor logische componenten. Bij het 'construeren' van symbolen voor logische componenten bleek weldra de noodzaak tot het vastleggen van allerlei tot dan toe intuïtief gehanteerde begrippen in de digitale techniek. In de praktijk bestond nl. een grote spraakverwarring op velerlei gebied. De lezer vergelijkte als toelichting hierop bijvoorbeeld de omschrijving van de 'timing' van IC's in oudere catalogi. De werkgroep heeft in deze baanbrekend werk gedaan. Een en ander heeft o.a. geresulteerd in de op dit moment van belang zijnde publicaties:

- publicatie 617-12 'Graphical symbols for diagrams - Part 12: Binary logic elements', over logicasymbolen van de IEC, en de
- nederlandse praktijkrichtlijn NPR 5160, 2e druk, 1984, als toelichting op publicatie 617-12. Hierin is de huidige stand van zaken m.b.t. het tekenen van logicaschema's vastgelegd.

De hierna volgende introductie van het IEC-systeem behandelt het systeem voor zover nodig is voor het interpreteren van de in de tekst toegepaste symbolen. Voor het lezen van alle symbolen in catalogi is kennis van voormelde publicaties noodzakelijk. Zie ook de literatuuropgave.

Het doel van een tekening

Een logische schakeling is een realisatie van een logisch ontwerp. Men zou kunnen stellen dat met een goede documentatie van de schakeling ook het logisch ontwerp gedocu-

menteerd is. De praktijk leert echter anders. Enerzijds bevatten logische schakelingen vaak componenten die geen logische functie bezitten, om redenen van fanin en fanout bijvoorbeeld toegevoegd, zodat de weg terug naar de logische specificatie van het ontwerp vaak onduidelijk is. Anderzijds worden logische schakelingen op technisch geheel verschillende wijzen gerealiseerd, variërend van IC's tot pneumatische logica. Er is dus documentatie nodig op het gehele traject van een ontwerp, en aangepast aan het stadium waarin een ontwerp verkeert.

Om toch zoveel mogelijk *één taal* te kunnen gebruiken heeft men er bij het opstellen van het IEC 617-12 normblad naar gestreefd *intern in symbolen* alle relaties tussen ingangen en uitgangen op *logisch niveau* vast te leggen, d.w.z. alles zoveel mogelijk uit te drukken in 'nullen en enen' en 'logische operaties' hierop, alsmede met hiervan afgeleide operatoren. Dit streven heeft het voordeel van een uniforme beschrijving van de verschillende fysische realisatievormen.

Aanvullend hierop moet de interface naar de fysische realiteit gedefinieerd worden. Immers, een onderhoudstechnicus meet spanningen of andere fysische grootheden, terwijl vaak niet bekend is wat de logische functie ervan is. Ook hierin voorziet IEC 617-12, voor zover het spanningsniveau's betreft. Daarnaast moeten ook open-collector en 3-state uitgangsconstructies worden weergegeven, en vele andere details op het grensvlak logica/fysica.

Tekeningen van logische schakelingen worden onderscheiden in

- theoretische logicaschema's, en
- uitvoeringsschema's.

Op een *theoretisch logicaschema* (logic diagram) worden uitsluitend logische grootheden toegepast, zowel intern in de symbolen (volgens afspraak) als daarbuiten. Een theoretisch logicaschema documenteert een logisch ontwerp. Nu hebben we reeds gezien dat ook hierin ontwerpniveau's kunnen worden onderscheiden, van een theoretisch logisch model van een schakeling tot een op beschikbare componenten *direct afbeeldbaar* logisch model. Zie bijvoorbeeld paragraaf 5.2, figuur 5.8.

Op *uitvoeringsschema's* wordt daarnaast de relatie met fysisch meetbare grootheden vastgelegd. Op uitvoeringsschema's kan aanvullende informatie verstrekt worden, zoals pennummers e.d.

We zullen nu de voornaamste concepten van het 617-12 normblad introduceren.

A.2. Vorm van de symbolen

De IEC-symbolen bestaan uit een *rechthoek van willekeurige afmetingen*. Door de variabele lengte/breedteverhouding is het mogelijk de vorm aan te passen naar gelang het aantal ingangen en/of uitgangen erop. Ook kan men rekening houden met het aantal te vermelden toevoegsymbolen bij een ingang of uitgang. Bij een toenemend aantal ingangen neemt bij 'distinctive shape' symbolen het oppervlak kwadratisch toe, voor rechthoeken is dit ongeveer lineair. Een en ander bevordert de leesbaarheid van een schema.

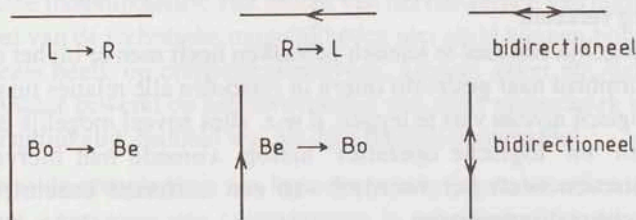
Als *hoofdrichting van de data- en informatiestroom* in tekeningen wordt aanbevolen aan te houden

links → rechts

en/of

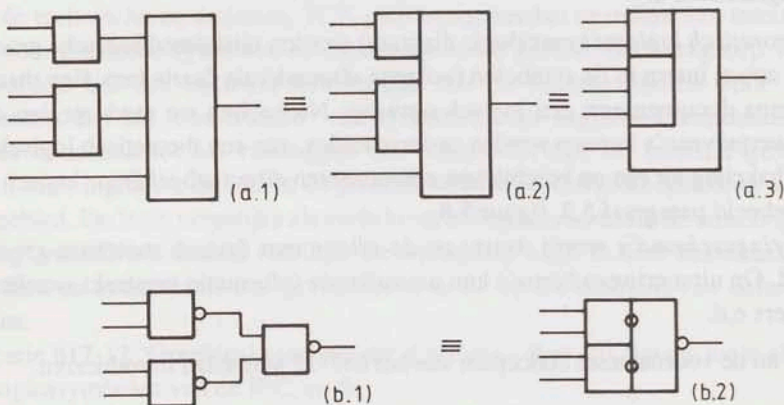
boven \rightarrow beneden.

Er mogen dan geen pijlen in verbindinglijnen gezet worden. Bij afwijkende richtingen kan de duidelijkheid vergen dat er in de verbindinglijnen pijlpunten worden opgenomen, zoals figuur 1 laat zien.



Figuur 1. Richting informatie-opbouw/overdracht.

Ook symbolen mogen gedraaid worden. Daarbij moeten dan de inschriften en bijschriften op de juiste wijze meeroteren. Voor meer informatie hierover zie de NPR 5160, 2e druk. Door hun rechthoekige vorm kunnen symbolen gemakkelijk tegen en eventueel in elkaar getekend worden. Figuur 2 laat enkele van de mogelijkheden zien. Bij aan en in elkaar getekende symbolen wordt verondersteld dat de signaalwaarden gewoon worden doorgegeven. Bij een genegeerde uitgang/ingang moet de negatie-indicator door de verbindinglijn getekend worden.



Figuur 2. Samenvoegen van symbolen.

De in figuur 2 geschetste mogelijkheden maken het mogelijk tekeningen compact uit te voeren. Aldus wordt de overzichtelijkheid bevorderd.

Naast genoemde middelen om symbolen in tekeningen samen te voegen bestaan er enkele andere, waaronder een *gemeenschappelijk ingangsblok* (common input block) en een *gemeenschappelijk uitgangselement* (common output element). Hierop worden gemeenschappelijke ingangen of gemeenschappelijke uitgangen aangegeven. Zie voor details het IEC 617-12 normblad.

De hiervoor genoemde middelen om symbolen aan en in elkaar te tekenen vereisen enige spelregels voor de interpretatie van het aldus samengestelde symbool. Een ervan is dat het

IEC-systeem de algemene regel kent dat er geen *logische verbindingen* (eventueel wel voedingslijnen) aanwezig zijn tussen elementen waarvan de symbolen aan elkaar getekend zijn, indien de grenslijn *evenwijdig is aan de richting van de informatie-overdracht*. Anders wordt het indien de grenslijn *loodrecht* op de richting van de informatie-overdracht staat. In dat geval bestaat er, tenzij anders is aangegeven, precies één logische verbinding in de richting van de informatie-overdracht.

Uiteraard bestaan er op deze regel uitzonderingen, welke in het normblad zijn beschreven. Voor onze toepassingen is de regel voorlopig voldoende.

A.3. Functie-aanduiding

De logische functie van een binair logisch element wordt aangeduid met een zgn. *functiesymbool* (general qualifying symbol). Dit symbool staat bij voorkeur boven in het kader; toegestaan is ook het midden van het kader.

Het functiesymbool heeft altijd betrekking op de interne ingangswaarden en de interne uitgangswaarden van een symbol. Bij het IEC-systeem worden de interne waarden altijd uitgedrukt in de logische 0 (false) en de logische 1 (true). (Hoe deze interne waarden naar 'buiten' worden overgedragen bespreken we hierna.) Figuur 3 geeft een overzicht van enkele gangbare functiesymbolen.

- & Aanduiding logische AND; uitgang is 1 indien alle ingangen 1 zijn.
- ≥ 1 Aanduiding logische OR; uitgang is 1 indien tenminste een ingang 1 is.
- = 1 Aanduiding EXOR-functie; uitgang is 1 indien een ingang 1 is.
- $\geq m$ Uitgang is 1 indien tenminste m ingangen 1 zijn.
- 1 Uitgang is 1 indien de ingang 1 is.
- 2k Uitgang is 1 indien een even aantal ingangen 1 is.
- 2k+1 Uitgang is 1 indien een oneven aantal ingangen 1 is.

Figuur 3. Logische functiesymbolen.

- ▷ Element met versterkte uitgangen (verhoogde fanout)
- ▽ 3-State uitgang
- ◇ Open-circuit uitgang
- ◊ Open-circuit uitgang (vereist pull-up weerstand)
- ◊̄ Open-circuit uitgang (vereist pull-down weerstand)
- ◊̂ Open-circuit uitgang (met interne pull-up weerstand)
- ◊̄̂ Open-circuit uitgang (met interne pull-down weerstand)
- ⏏ Schmitt-trigger ingangen (hysteresis)

Figuur 4. Elektrische/elektronische toevoegsymbolen.

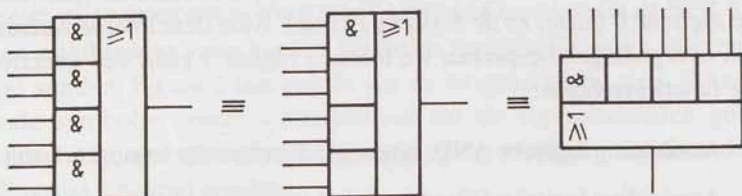
Naast de in figuur 3 aangegeven functiesymbolen kent het systeem nog enkele andere symbolen, waarmee de elektrische/elektronische eigenschappen van een poort kunnen worden aangegeven, of symbolen die nadere informatie geven hoe een logische functie is gerealiseerd (wired AND e.d.). Deze zijn aangegeven in figuur 4. Indien zij in combinatie met het logische functiesymbool voorkomen, boven in het symbool, geldt de aanduiding voor alle ingangen, resp. uitgangen.

Voorbeeld

De combinatie $\& \triangleright$ duidt op de logische AND (intern), terwijl de poort gebufferde uitgangen heeft.

Logische arrays

Om enig tekenwerk te besparen is het toegestaan in een 'array' van elementen met hetzelfde functiesymbool het functiesymbool alleen te vermelden in het bovenste symbool (of meest linkse symbool). Zie figuur 5.



Figuur 5. Aanduiding logische arrays.

Opmerking

In de IEC-publicaties wordt gesproken over de interne logische *toestanden* van ingangen en uitgangen. Buiten de symbolen spreekt men vaak over logische *waarden*. Wij geven er de voorkeur aan het begrip 'toestand' te reserveren voor het beschrijven van *geheugenwerking*. De logische toestand is dan weer zichtbaar/observeerbaar aan de logische waarde van het ermee corresponderende uitgangssignaal. Voor het overige spreken we over logische waarden.

Opmerking

Niet genormaliseerde informatie, bijvoorbeeld de 'naam' van een component in een groter geheel, mag aan de symbolen toegevoegd worden. Afgesproken is deze informatie tussen rechte haken, [...], te plaatsen.

A.4. De interface met de fysische wereld

Bij een ontwerp dat top-down ontstaat hanteert de ontwerper in eerste instantie uitsluitend logische variabelen en operatoren. Na enige tijd ontstaat een *logisch ontwerp* van de te realiseren schakeling. In dit stadium corresponderen de logische nullen en enen intern in de symbolen meestal direct met de logische nullen en enen, welke de waarheidswaarde van externe signaalsnamen aangeven. In een vroeg ontwerpstadium geldt voor de meeste ingangen/uitgangen op symbolen:

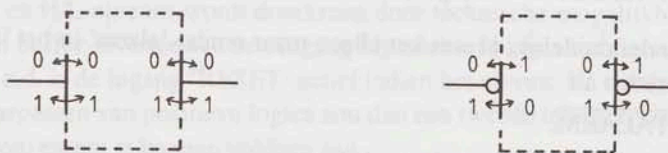
intern 0 \leftrightarrow extern 0;

intern 1 \leftrightarrow extern 1.

Naarmate een ontwerp vordert en men meer te maken krijgt met de eigenschappen van de componenten wordt dit anders. Zo keren de meeste ingangsaanpassingen (fanin invertoren) en uitgangsaanpassingen (totem pole bijvoorbeeld) het signaalniveau om. De logische beschrijving van een signaalinversie is de *negatie*. Het IEC-systeem bevat middelen om een en ander aan de rand van symbolen aan te geven, zowel op logisch niveau als op elektrisch niveau. Voor de gebruiker van logische schema's betekent een en ander dat logische ontwerpschema's en diagrammen kunnen verschillen van uitvoeringsschema's.

De negatie-indicator

Figuur 6 laat zien hoe logische waarden door de grenslijn van een symbool worden overgedragen. Zijn ingangen en uitgangen direct op het symbool getekend, dan worden logische waarden onveranderd overgedragen. Met een cirkeltje ertussen wordt $0 \rightarrow 1$ en $1 \rightarrow 0$ omgezet. Dit cirkeltje heet *negatie-indicator*.

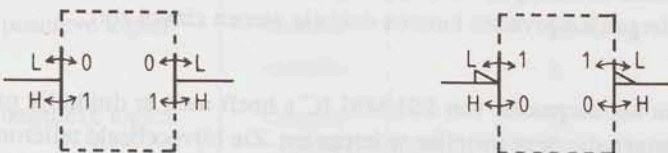


Figuur 6. Negatie-indicator.

De negatie-indicator kan/moet ook worden toegepast bij overgangen van 'embedded symbols'. De negatie-indicator wordt dan door de grenslijn getekend. Zie figuur 2.

De polariteitsindicator

Op uitvoeringsschema's worden externe signaalwaarden uitgedrukt in 'H' en 'L'. Binnen de symboolkaders blijft alles uitgedrukt in logische variabelen en operatoren. De *polariteitsindicator* (met een driehoekje) maakt het mogelijk elk gewenst verband tussen externe signaalwaarden en interne logische waarden weer te geven. Zie figuur 7.



Figuur 7. Polariteitsindicator.

Met de polariteitsindicator is de 'interface' tussen de fysische wereld, zoals men die in een schakeling kan meten en de logica intern in de blokjes, vastgelegd. Voor spanningen is de interface in het IEC 617-12 normblad vastgelegd, voor bijv. pneumatische logica (nog) niet. Wel kan men besluiten ook hier 'H' en 'L' te hanteren, mits duidelijk omschreven is wat er bedoeld wordt.

A.5. Signaalnamen

Signaalnamen geven de betekenis/functie van een signaal weer. Signaalnamen moeten daartoe duidelijk zijn, hetgeen slechts het geval is indien aan bepaalde voorschriften is voldaan. Zo is de naam START/STOP bij een signaalingang niet duidelijk, omdat deze naam een tegenstrijdigheid bevat. Tevens is niet duidelijk of bij 'aanwezigheid' van het signaal gestart of gestopt moet worden.

Het verdient daarom sterke aanbeveling een naam zo mogelijk af te leiden van een propositie, welke laatste dan met de bijbehorende naam (welke er meestal een afkorting van zal zijn) in een overzicht van namen in de documentatie moet worden opgenomen. Voor het vorenstaande geval wordt de naam/propositie dan

START \Leftrightarrow 'Er is een startopdracht'.

Tevens wordt aanbevolen een naam/propositie zo mogelijk in de bevestigende zin te formuleren. De voorkeur gaat dus uit naar

ALARM \Leftrightarrow 'Er is een alarm'

en niet naar

NIETALARM \Leftrightarrow 'Er is geen alarm'.

Dit laatste is minder duidelijk, al was het alleen maar omdat 'alarm' in het laatste geval genoteerd zou worden als

NIETALARM.

Formeel is deze aanduiding echter nog steeds juist.

Soms komt men bij naamgeving in de moeilijkheden. Zo kan het drukken op een knop de functie 'START' hebben indien een apparaat gestopt is, terwijl dezelfde knop de functie 'STOP' heeft indien het apparaat loopt. Men komt dan vrijwel automatisch op de naam START/STOP. Nu is in dit geval het probleem geschapen door de wijze waarop de bediening van het apparaat gerealiseerd is. Over het algemeen is een dergelijke koppeling van functies onoverzichtelijk en dus niet aan te raden.

Een toepassing die wel zinvol is, is het gebruik van de naam 'LEES/SCHRIJF' bij een geheugen bijvoorbeeld. Hier zijn 'lezen' en 'schrijven' ondubbelzinnig aan elkaar gekoppeld omdat het niet actief zijn van de ene opdracht automatisch de andere activeert, en omgekeerd. In dergelijke gevallen kunnen dubbele namen zinvol zijn.

Opmerking

In het begin van het toepassen van SSI/MSI IC's heeft men er duidelijk naar gestreefd functies en ingangen die deze instellen te koppelen. Zie bijvoorbeeld tellerontwerpen met ingangen als LD (load), R (reset) en andere namen. Deze koppeling wordt langzamerhand losgelaten. De in te stellen functies worden *gecodeerd*. De bij de gewenste functie behorende ingangscombinatie moet dan op de instellingen worden aangeboden. De naam bij dergelijke instellingen heeft dan meer een aanduidende dan een functionele betekenis.

A.6. Signaalnamen en logische niveaus

Bij een logisch ontwerp is het verband tussen 'naam' en 'waarheidswaarde' vast. Een naam of de erdoor aangeduide propositie is waar als het ermee corresponderende signaal de logische waarde '1' heeft. Het ligt voor de hand dit in de realisatiefase voort te zetten. Bij een *vast verband* tussen 'naam' en 'signaalwaarde' heeft men de keuze uit twee conventies, *positieve* en *negatieve logica*.

Bij positieve logica wordt het 'waar' zijn van een propositie/naam/variabele voorgesteld door resp. afgebeeld op het met 'H' aangeduide spanningsniveau en het 'niet waar' zijn op 'L'. Wat de gebieden zijn voor 'H' en voor 'L' wordt nader gedefinieerd. De ligging van H/L is o.a. technologie-afhankelijk. In de praktijk volgt men (bij een vaste keuze) meestal de positieve logica conventie.

Bij negatieve logica beeldt men precies omgekeerd af. Dan correspondeert 'waar' met het spanningsniveau 'L'.

Gaan we uit van de *logische specificatie* voor een ontwerp, dan ontstaan er afhankelijk van de keuze voor positieve of voor negatieve logica verschillende *elektrische specificaties* in termen van H/L tabellen. We hebben dit reeds toegelicht in paragraaf 5.1 van de tekst.

Dit streven naar uniformering van de afbeelding van waarheidswaarden van namen/proposities en H/L niveaus wordt doorkruist door technische mogelijkheden en/of voorkeuren. Een buffer-invertor aan een ingang bijvoorbeeld keert het signaalniveau om. Bij veel tellers e.d. is de ingang 'RESET' actief indien het niveau 'L' wordt aangeboden. Bij een strikt toepassen van positieve logica zou dan een tweede invertor voor de resetingang nodig zijn, om extern te kunnen voldoen aan

$$\text{RESET} \Leftrightarrow \text{H.}$$

Deze tweede invertor laat men in de praktijk meestal weg, mede om bijvoorbeeld de setup time $t_{su}(\text{RESET})$ te beperken.

Tabel 1 geeft aan hoe bij positieve en negatieve logica toch aangegeven kan worden met welk niveau het waar zijn van een naam (of het actief zijn van een instelling) correspondeert. Men doet dit door zo nodig de naam te voorzien van een *negatiestreep*.

Conventie	Naam	Naam waar/instelling actief indien het niveau gelijk is aan
positieve logica	<naam>	H
	< $\overline{\text{naam}}$ >	L
negatieve logica	<naam>	L
	< $\overline{\text{naam}}$ >	H

Tabel 1. Naamgeving bij positieve en negatieve logica.

De afspraak positieve of negatieve logica te gebruiken geldt in het algemeen voor een gehele tekening of ontwerp. De afspraak moet wel vermeld worden op de tekening. Hierna ligt de relatie logische specificatie - elektrische specificatie/werking vast.

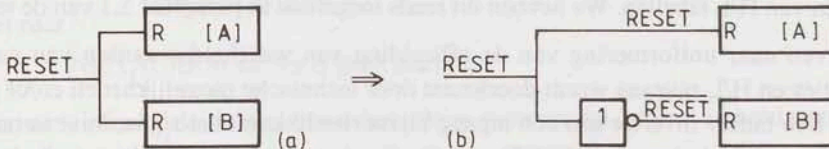
Afspraak

Op schema's met positieve logica en op schema's met negatieve logica past men bij voorkeur de *negatie-indicator* toe om negaties/inversies aan te geven. Een en ander is in overeenstemming met de naamgeving, waarbij het 'andere actieve niveau' ook met een negatiestreek boven de naam wordt aangeduid.

Positieve of negatieve logica is, zeker op onderhoudschema's, niet altijd even duidelijk bij het interpreteren van het schema. Het volgende voorbeeld gaat hier op in.

Voorbeeld

In een *logisch ontwerp* moet het resetsignaal 'RESET' twee modules bedienen. Zie figuur 8.a. Tijdens de realisatie van het ontwerp blijkt dat het handig is om voor A een component te selecteren, waarvan het actieve niveau van de resetingang het niveau 'H' is. Voor module B blijkt 'L' handig te zijn. Het logisch ontwerpschema wordt dus aangepast en omgezet in dat van figuur 8.b. Deze figuur nu is direct afbeeldbaar op de beschikbare componenten.



Figuur 8. Aansluitschema resetingang.

De namen in figuur 8.a en figuur 8.b zijn vermeld overeenkomstig de positieve logica conventie (tabel 1). Op schema (b) komen dan twee namen voor, RESET bij A en $\overline{\text{RESET}}$ bij B. Toch worden beide units *tegelijk* gereset of gereset. De naamgeving suggereert echter een verschil in functie, welke er in dit geval niet is. Hetzelfde bezwaar geldt voor schema's waarop negatieve logica is toegepast.

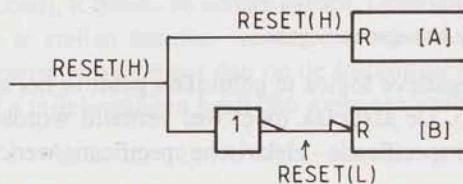
De gesignaleerde mogelijkheden tot misverstanden komen voort uit het feit dat er voor het *aangeven van fysische verschillen*, welk niveau 'actief' is, een *logische operator* wordt toegepast. Op uitvoeringsschema's past men daarom bij voorkeur *individuele logica* toe. (In Nederland zelfs voorgeschreven in de norm.) Bij elke naam wordt dan genoteerd of de bijbehorende functie actief is met het hoge niveau,

naam(H)

of met het lage niveau,

naam(L).

Het schema van figuur 8 wordt in individuele logica dat van figuur 9.



Figuur 9 Aansluitschema in individuele logica.

Individuele logica is op gebruikersniveau in het algemeen wat gemakkelijker te interpreteren dan positieve en negatieve logica. In een bepaalde fase van een ontwerp, het uitvoeringsstadium, gaat men dan ook meestal over van positieve logica naar individuele logica. Wanneer men overstapt hangt van persoonlijke voorkeur en ervaring van de ontwerper af. Op *uitvoeringsschema's* wordt ten alle tijde *individuele logica* aanbevolen! Ook individuele logica geldt voor een hele tekening of ontwerp. Het gebruik van polariteitsindicatoren is voorschreven.

Opmerking

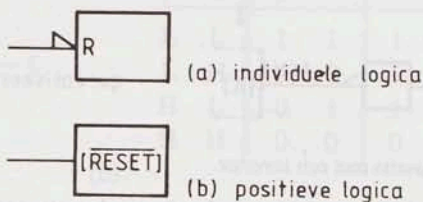
In catalogi van geïntegreerde circuits leest men vaak de aanduiding 'actief Laag' en 'actief Hoog' bij een signaalnaam of ingang van een circuit. Men bedoelt hiermee dan te zeggen dat de propositie

'<naam> is present'

waar is als het niveau op de met de naam corresponderende lijn of signaalpunt Hoog (actief hoog) of Laag (actief laag) is. Uiteraard verdient het de voorkeur met genormaliseerde aanduidingen te werken, met aanduidingen als naam(H), etc.

Opmerking

Het feit dat een instelling actief is met een bepaald logisch niveau kon men op verschillende wijzen aangeven. In figuur 10.a is de genormaliseerde aanduiding toegepast. Het toevoegsymbool R aan de ingang is in de norm gedefinieerd. De polariteitsindicator geeft aan dat het actief zijn ('RESET present') van RESET correspondeert met een uitwendig niveau 'L'.



Figuur 10. Polariteitsaanduiding.

Figuur 10.b toont een niet-genormaliseerde aanduiding, welke hetzelfde aangeeft als het genormaliseerde symbool. Omdat het voor de gangbare logische schakelingen vrijwel altijd mogelijk is de functie in een genormaliseerd symbool uit te drukken verdient de genormaliseerde aanduiding de voorkeur.

A.7. Inversie en negatie

De begrippen *inversie* en *negatie* zijn niet identiek.

Definitie

Een *inverter* is een schakeling, die een hoog *niveau* aan de ingang omzet in een laag niveau aan de uitgang en omgekeerd een laag ingangsniveau in een hoog uitgangsniveau ($H \Rightarrow L$ en $L \Rightarrow H$).

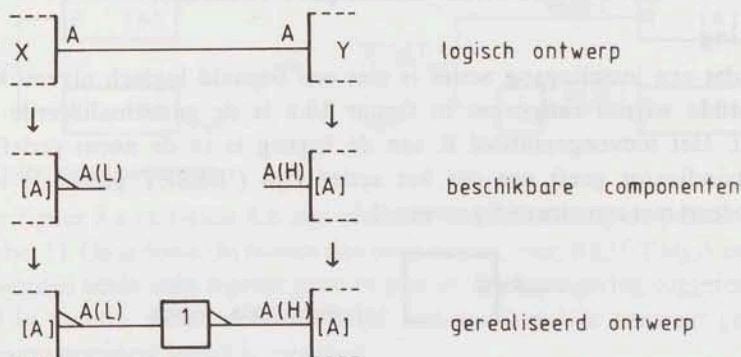
Definitie

Een *negator* zet een aan de ingang aangeboden *logische waarde* om in de tegengestelde waarde aan de uitgang ($0 \Rightarrow 1$ en $1 \Rightarrow 0$ of $T \Rightarrow F$ en $F \Rightarrow T$).

Een invertor behoeft niet noodzakelijk een logische negatie in te houden, zoals we zien aan het volgende voorbeeld.

Voorbeeld

Figuur 11.a toont een gedeelte van een ontwerpschema. Bij het omzetten hiervan in een schakeling blijkt dat voor de realisatie van de module X een bouwsteen beschikbaar is die het signaal A(L) afgeeft. Voor module Y is een bouwsteen beschikbaar die hetingangssignaal A(H) nodig heeft (figuur 11.b). Bij de afbeelding van het logisch ontwerp op de beschikbare modules blijkt het dus nodig te zijn de *signaalniveaus* van de bouwstenen aan te passen. Zie figuur 11.c. De toepaste *invertor* verricht geen logische negatie (dit is immers niet nodig), maar zet alleen een hoog in een laag niveau om en omgekeerd.



Figuur 11. Voorbeeld van realisatie met een invertor.

Het omgekeerde komt ook voor, nl. dat op het logisch ontwerpschema negatoren voorkomen en op het uitvoeringsschema geen expliciet getekende invertoren.

In de praktijk blijken er zeer veel fouten gemaakt te worden bij het afbeelden op 'H' en op 'L'. Het is van groot belang op dit punt nauwkeurig te werken.

Opmerking

Afgeleid van de hiervoor geschetste problematiek is de naamgeving van componenten zoals NAND en NOR. Eigenlijk zou men deze moeten aanduiden als AND/OR met geïnverteerde uitgang'. In de praktijk heeft men dit veelal ondervangen door reeds op logisch niveau de logische operatoren NAND (NOT-AND) en NOR (NOT-OR) te gebruiken. Vergelijk paragraaf 5.3.

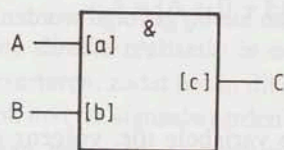
A.8. Symbolen en formules I

Alvorens het symbolensysteem verder te introduceren is het nuttig in te gaan op enkele problemen bij het toepassen ervan. Bespreking van deze problemen vergroot het inzicht.

Een goed top-down gestructureerd ontwerpproces resulteert op zeker moment in een zgn. *logisch ontwerp* van de te realiseren schakeling. Dit logisch ontwerp moet vervolgens worden afgebeeld op componenten welke, in ons geval, werken met H en L spanningsniveaus. Bij de afbeelding moet het logisch ontwerp meestal verder worden uitgewerkt c.q. worden aangepast aan de beschikbare componenten.

Passings/afbeeldingsproblemen lost men meestal op logisch niveau op en niet via uitgebreide manipulaties op H/L-tabellen. Daartoe kent men aan de componenten meestal ook een logische functie toe, via de positieve logica interpretatie bijvoorbeeld. Het logisch ontwerp wordt vervolgens via formulemanipulaties omgezet in een *direct afbeeldbaar logisch ontwerp* (detailed logic design). Een argument te meer om afbeeldingsproblemen op logisch niveau op te lossen is het feit dat symbolen voor componenten niet ondubbelzinnig vastliggen op grond van de genormaliseerde regels. Voor veel bouwstenen zijn verschillende symbolen mogelijk, hetgeen het vergelijken op symbolisch niveau zeer lastig maakt.

Voorbeeld

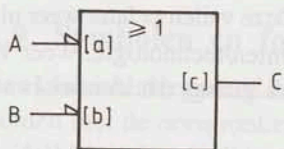


(a)

A	B	a	b	c	C
L	L	0	0	0	H
L	H	0	1	0	H
H	L	1	0	0	H
H	H	1	1	1	L

⇒

A	B	C
L	L	H
L	H	H
H	L	H
H	H	L



(b)

A	B	a	b	c	C
L	L	1	1	1	H
L	H	1	0	1	H
H	L	0	1	1	H
H	H	0	0	0	L

⇒

A	B	C
L	L	H
L	H	H
H	L	H
H	H	L

Figuur 12. Verschillende symbolen met dezelfde H/L-tabel.

In figuur 12.a is het bij een schakeling behorend symbool beschreven. Ernaast is aangegeven hoe hieruit de H/L-tabel wordt afgeleid. We beginnen met in de eerste twee kolommen alle mogelijke combinaties van H/L-waarden van de punten A en B te noteren. Deze waarden worden naar interne logische waarden omgezet (in de kolommen drie en vier) via $H \Rightarrow 1$ en $L \Rightarrow 0$ (geen polariteitsindicator). Voor het gemak zijn interne logische variabelen a t/m c geïntroduceerd. Omdat het geen genormaliseerde aanduidingen betreft worden ze tussen rechte haken geplaatst.

Het functiesymbool & legt het logisch verband vast tussen de interne logische ingangswaarden van a en b enerzijds en de logische waarde van het interne uitgangssignaal c anderzijds. Hiermee ligt ook kolom vijf van de tabel vast.

De interne logische waarde van de variabele c wordt vervolgens naar het uitgangssignaal C vertaald. Op de conversie is de polariteitsindicator van toepassing, zodat $1 \Rightarrow L$ en $0 \Rightarrow H$. Hieruit volgt de tweede tabel, geheel in spanningsniveaus.

Passen we dezelfde procedure toe op het symbool in figuur 12.b, dan ontstaat dezelfde H/L-tabel. De met beide symbolen beschreven componenten blijken identiek te zijn.

In internationaal verband wordt gewerkt aan het opstellen van een *symbolenbibliotheek* voor standaard IC's. Van veel IC's is het symbool reeds vastgesteld. Met betrekking tot de in het voorbeeld geschetste problematiek lost dit echter niets op. Immers, het afbeelden van een logisch ontwerp resulteert zeker niet zonder meer in een 'standaardsymbool' of schema bestaande uit standaardsymbolen.

Conclusie

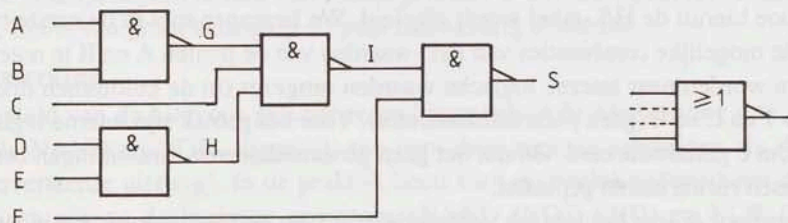
Passingsproblemen moeten bij voorkeur op *logisch niveau* worden uitgezocht.

In de praktijk komt men dan vaak het volgende probleem tegen. Van een gegeven schakeling moet het ontwerp herzien worden (redesign) omdat de schakeling bijvoorbeeld in een andere technologie gerealiseerd gaat worden, met specifieke voorkeuren voor andere logische operatoren. Uiteraard verdient het aanbeveling om een dergelijke aanpassing uit te voeren met als vertrekpunt een specificatie op een *hoger logisch niveau*. In de praktijk is meestal slechts een documentatie van de schakeling beschikbaar, al dan niet in standaardsymbolen, en geen specificatie van de logische werking. Wil men de conversie op logisch niveau kunnen plegen, dan zal eerst een logische formule van de (deel-)schakeling bepaald moeten worden. De volgende procedure kan hierbij gevolgd worden.

Procedure

1. Omrand de te vervangen deelschakeling.
2. Ken aan elke ingang en aan elke uitgang een logische variabele toe, volgens de positieve logica afspraak.
3. Stel vervolgens een formule op voor de schakeling. Zo nodig worden tijdens het opstellen variabelen toegekend aan interne knooppunten. Deze vallen er later weer uit.
4. Beeld de opgestelde formule af op de nieuwe componenten/technologie, weer via positieve logica. Het resultaat is een schakeling met een H/L gedrag dat identiek is aan dat van de oorspronkelijke schakeling.
5. Pas aan de randen eventueel de polariteit van signalen aan. (Indien de 'zender' een uitgangsinvertor heeft en de ontworpen schakeling een ingangsinvertor heeft, dan mogen deze beide geschrapt worden, bijvoorbeeld.)

Voorbeeld



(a) gegeven

(b) beschikbaar

Figuur 13. Conversie van AND-INVERT naar OR-INVERT realisatie.

De schakeling in figuur 13 heeft zes ingangen en één uitgang. In positieve logica geldt:

$$G = \overline{AB},$$

$$H = \overline{DE},$$

$$I = \overline{GCH},$$

$$S = \overline{IF}.$$

Hieruit volgt door substitutie:

$$\begin{aligned} S &= \overline{IF} \\ &= \overline{GCH \cdot F} \\ &= \overline{AB \cdot C \cdot DE \cdot F} \\ &= \overline{(AB + \overline{C} + DE) \cdot F} \\ &= \overline{AB + \overline{C} + DE} + \overline{F} \\ &= (\overline{A} + \overline{B}) \cdot C \cdot (\overline{D} + \overline{E}) + \overline{F} \\ &= \overline{ACD} + \overline{ACE} + \overline{BCD} + \overline{BCE} + \overline{F}. \end{aligned}$$

De eenvoudigste produkt-van-sommen vorm hiervan is:

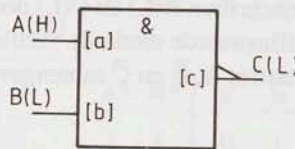
$$S = (\overline{A} + \overline{B} + \overline{F})(C + \overline{F})(\overline{D} + \overline{E} + \overline{F}).$$

Een directe realisatie in een 'two level OR-INVERT' vorm vereist enkele ingangsinvertoren, zodat het in dit stadium aanbeveling verdient naar de omgeving te kijken, of ook inverse signaalwaarden beschikbaar zijn. Bij het afbeelden van de gevonden formules moet, uiteraard, weer positieve logica toegepast worden. Het H/L-gedrag van de schakeling is dan identiek met dat van de oorspronkelijke schakeling.

A.9. Symbolen en formules II

Men kan zich afvragen of de procedure, zoals beschreven in sectie A.8 niet toegepast kan worden met de oorspronkelijke namen/variabelen in het gegeven schema (individuele logica). We werken daartoe de volgende voorbeelden uit.

Voorbeeld



Figuur 14. Naamaanduiding bij individuele logica.

In figuur 14 is een symbool getekend met de namen A(H), B(L) en C(L), in individuele logica. We leiden het logisch verband tussen de namen A en B enerzijds en C anderzijds af met behulp van tabel 2.

A	B	A	B	a	b	c	C	C	A	B	C
F	F	L	H	0	1	0	H	F	F	F	F
F	T	L	L	0	0	0	H	F	F	T	F
T	F	H	H	1	1	1	L	T	T	F	T
T	T	H	L	1	0	0	H	F	T	T	F

⇒

A	B	C
F	F	F
F	T	F
T	F	T
T	T	F

$C = A \cdot \overline{B}$

Tabel 2. Bepaling van een formule bij een symbool.

Deze tabel is als volgt opgesteld. De met de namen corresponderende proposities A en B kunnen de in de eerste twee kolommen vermelde vier combinaties van *waarheidswaarden* aannemen. In de derde en vierde kolom staan de met deze waarheidswaarden corresponderende *niveaus*. Voor A is het verband gedefinieerd door A(H); voor B door B(L).

Er staan geen driehoekjes aan de ingangen. Voor beide ingangen correspondeert uitwendig H met inwendig 1. Dit is weergegeven in de vijfde en zesde kolom van tabel 2 onder a en b. Hieruit volgt, gezien het in het symbool aangegeven functiesymbool &, de zevende kolom onder c.

Aan de uitgang staat de *polariteitsindicator*. Inwendig 1 correspondeert met uitwendig L. Hieruit volgt de kolom in H en L onder C.

Vervolgens wordt de uitgangsspanning via C(L) weer vertaald in T en F. Daarmee is ook de negende kolom ingevuld. In een aparte tabel is het verband tussen de waarheidswaarden van de proposities A en B enerzijds en van C anderzijds nogmaals samengevat (eerste, tweede en negende kolom).

$$C = A \text{ EN } \bar{B} \quad \text{of} \quad C = A \cdot \bar{B}.$$

Deze formule beschrijft het verband tussen de waarheidswaarden van de proposities A, B en C zoals dit door de door het symbool van figuur 14 gerepresenteerde schakeling wordt gelegd.

Opmerking

Het is gemakkelijk in te zien dat men op grond van de geldende afspraken het bijschrift A(H) mag vervangen door $\bar{A}(L)$. Immers, heeft de propositie

‘A is present’

de waarheidswaarde F, dan heeft de propositie

‘NOT-A is present’

de waarheidswaarde T en omgekeerd. Evenzo mag men de bijschriften B(L) en C(L) vervangen door $\bar{B}(H)$ en $\bar{C}(H)$.

Vervangt men in figuur 14 de bijschriften B(L) en C(L) door $\bar{B}(H)$ en $\bar{C}(H)$, dan kan men met behulp van de in tabel 2 geïllustreerde methode verifiëren dat het verband tussen de waarheidswaarden van de proposities A, \bar{B} en \bar{C} samengevat kan worden zoals in tabel 3 is aangegeven (ga dit na!).

A	\bar{B}	\bar{C}
F	F	T
F	T	T
T	F	T
T	T	F

Tabel 3. Verband tussen de proposities A, B en C.

Uit tabel 3 volgt

$$\bar{C} = \bar{A} \text{ OF } \bar{B} \quad \text{of} \quad \bar{C} = \bar{A} + \bar{B},$$

welke formule op grond van de wetten van de logica gelijkwaardig is met de reeds gevonden formule

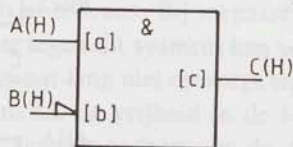
$$C = A \cdot \bar{B}.$$

Voorbeeld

Het symbool in figuur 15 leidt met de bijbehorende (t.o.v. figuur 14 gewijzigde!) namen tot dezelfde formule, namelijk

$$C = A \cdot \bar{B}.$$

(De wijziging bestaat eruit dat per ingang/uitgang naam(L) \Rightarrow naam(H) gaat en dat er een polariteitsindicator is toegevoegd.)



Figuur 15. Naamaanduiding bij individuele logica.

We vinden deze formule zoals in tabel 4 is aangegeven.

A	B	A	B	a	b	c	C	C	A	B	C
F	F	L	L	0	1	0	L	F	F	F	F
F	T	L	H	0	0	0	L	F	F	T	F
T	F	H	L	1	1	1	H	T	T	F	T
T	T	H	H	1	0	0	L	F	T	T	F

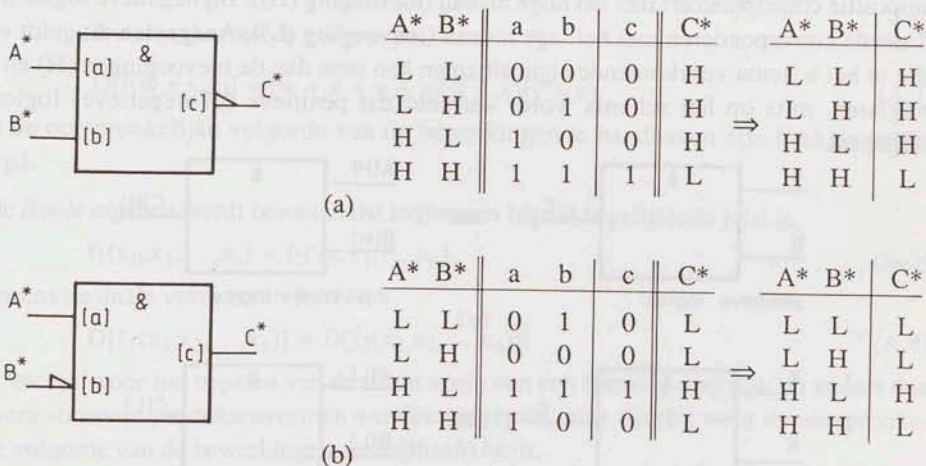
\Rightarrow

A	B	C
F	F	F
F	T	F
T	F	T
T	T	F

$C = A \cdot \bar{B}$

Tabel 4. Bepaling van een formule bij een symbool.

De bij de symbolen/schakelingen behorende H/L-tabellen zijn echter verschillend, zoals figuur 16 aantoont.

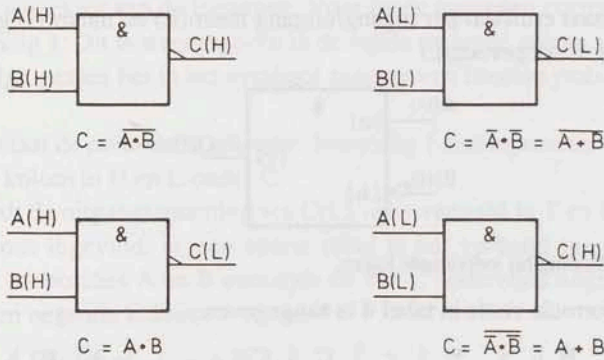


Figuur 16. Afleiding H/L-tabellen.

Hoewel de formules identiek zijn, geldt dit niet voor de bijbehorende H/L-tabellen. De schakelingen kunnen elkaar dus *niet* vervangen. (Het wordt aan de lezer overgelaten te beredeneren waar in het bovenstaande de 'fout' gemaakt is.)

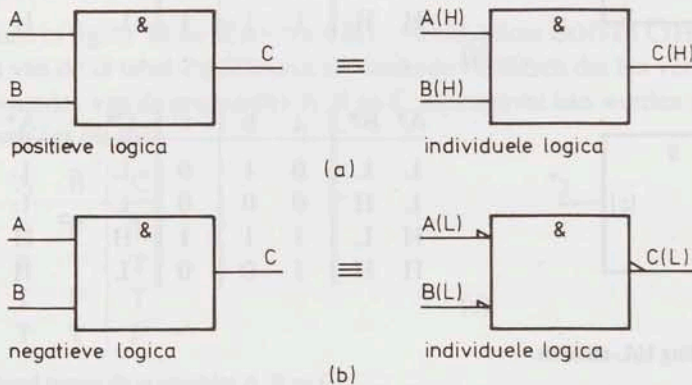
Voorbeeld

Figuur 17 geeft enkele formules, die de functie van een component voor een bepaalde interpretatie van A, B en C specificeren. Een en ander kan met behulp van de in tabel 2 beschreven methode worden afgeleid.



Figuur 17. Verschillende mogelijke functies van eenzelfde bouwsteen, afhankelijk van de interpretatie van A, B en C.

Uit het voorgaande blijkt dat bij toepassing van *individuele logica* ontwerper en gebruiker de vrijheid hebben om de *signaalnamen* zo te kiezen dat deze het best passen bij *hun interpretatie* of *hun beschrijving* van de gerealiseerde schakeling, mits zij de juiste toevoegingen (H) of (L) erachter plaatsen. Deze vrijheid mist men bij toepassing van het systeem van *positieve logica* of dat van *negatieve logica*. Bij positieve logica dient men een signaalnaam steeds zo te kiezen dat de waarheidswaarde T van de bijbehorende propositie correspondeert met het hoge niveau (toevoeging (H)). Bij negatieve logica moet T steeds corresponderen met het lage niveau (toevoeging (L)). Aangezien dit geldt voor alle in het schema voorkomende signaalnamen kan men dan de toevoegingen (H) en (L) weglaten, mits op het schema wordt vermeld dat positieve (of negatieve) logica is toegepast.



Figuur 18. Symbolen in verschillende logische conventies.

De in figuur 18 getoonde symbolen realiseren alle de formule

$$C = A \cdot B.$$

Voorts representeren de beide symbolen van figuur 18.a dezelfde schakeling. Ook de

symbolen van figuur 18.b representeren dezelfde schakeling. De in figuur 18.a gerepresenteerde schakeling verschilt echter van die van figuur 18.b. Zij hebben namelijk verschillende H/L-tabellen.

Conclusie

Uit het voorgaande blijkt dat toepassing van positieve of negatieve logica op uitvoeringsschema's het voordeel heeft dat de door de schakeling gerealiseerde formule gemakkelijk uit het symbool is af te lezen. Bij toepassing van individuele logica is dit moeilijker, hetgeen als bezwaar tegen dit systeem kan worden aangevoerd. De praktijk heeft echter geleerd dat dit bezwaar lang niet opweegt tegen het voordeel dat toepassing van individuele logica biedt m.b.t. de vrijheid in de keuze van signaalnamen en de duidelijke weergave van het fysisch gedrag van de schakeling. Met name bij het onderhoud en het testen van schakelingen weegt dit zwaar.

In deze tekst is daarom, in navolging van de Nederlandse normbladen, op uitvoeringsschema's individuele logica toegepast.

A.10. Dualiteit en positieve/negatieve logica

In de Boole algebra kent men het begrip 'duale vorm' van een formule. De duale vorm van een formule ontstaat uit de formule door de volgende bewerkingen:

- vervang + door \cdot en \cdot door +;
- vervang 1 door 0 en 0 door 1;
- handhaaf de volgorde van de oorspronkelijke bewerkingen.

Voorbeeld

$$f(w,x,y,z) = w\bar{x}yz + \bar{w}x + \bar{y}z. \quad (\text{A.1})$$

$$D[f(w,x,y,z)] = (w + \bar{x} + y + z)(\bar{w} + x)(\bar{y} + \bar{z}). \quad (\text{A.2})$$

Om de oorspronkelijke volgorde van de bewerkingen te handhaven zijn haakjes toegevoegd.

In de *Boole algebra* wordt bewezen dat indien een bepaalde gelijkheid juist is,

$$f_1(x_0, x_1, \dots, x_n) = f_2(x_0, x_1, \dots, x_n), \quad (\text{A.3})$$

eveneens de duale vorm ervan juist is,

$$D[f_1(x_1, x_2, \dots, x_n)] = D[f_2(x_1, x_2, \dots, x_n)]. \quad (\text{A.4})$$

Het 'recept' voor het bepalen van de duale vorm van een formule mag ook op andere dan zuivere som-van-produktenvormen worden toegepast, mits daarbij weer de oorspronkelijke volgorde van de bewerkingen gehandhaafd blijft.

Voorbeeld

$$f(x_0, x_1, x_2, x_3, x_4) = ((x_0 + x_1)x_2 + x_3)x_4.$$

Dan is

$$D[f(x_0, x_1, x_2, x_3, x_4)] = (x_0x_1 + x_2)x_3 + x_4,$$

en niet

$$D[f(x_0, x_1, x_2, x_3, x_4)] \neq x_0x_1 + x_2x_3 + x_4.$$

Toepassing op schakelwetten

Alle schakelwetten in hoofdstuk 2 zijn *in paren* gegeven, welke bij nadere beschouwing elkaars duale vorm blijken te zijn. Op grond van de dualiteit is het dan voldoende om een van de twee wetten te bewijzen. De duale vorm van de wet is dan, op grond van de dualiteit, ook bewezen.

Toepassing op positievelnegatieve logica

Dualiteit kan ook toegepast worden bij het interpreteren van gegeven schakelingen volgens positieve of negatieve logica. Hierop is de volgende stelling van toepassing.

Stelling

$$D[f(x_1, x_2, \dots, x_n)] = \overline{f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)},$$

waarbij $f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ uit $f(x_1, x_2, \dots, x_n)$ ontstaat door x_i in $f(x_1, x_2, \dots, x_n)$ te vervangen door \bar{x}_i en \bar{x}_i door x_i , voor $0 \leq i \leq n$.

Bewijs

We bewijzen de stelling voor functies in twee variabelen. Voor meer variabelen verloopt het bewijs overeenkomstig.

De mintermvorm van een functie in twee variabelen is

$$f(x_0, x_1) = \bar{x}_1 \bar{x}_0 f_0 + \bar{x}_1 x_0 f_1 + x_1 \bar{x}_0 f_2 + x_1 x_0 f_3.$$

Hierin zijn de functiewaarden f_0 t/m f_3 gelijk aan 0 of 1, afhankelijk van de functie.

De duale vorm van de functie $f(x_0, x_1)$ is dan

$$D[f(x_0, x_1)] = (\bar{x}_1 + \bar{x}_0 + \bar{f}_0)(\bar{x}_1 + x_0 + \bar{f}_1)(x_1 + \bar{x}_0 + \bar{f}_2)(x_1 + x_0 + \bar{f}_3).$$

Onder toepassing van de regels

$$(x + y)(x + z) = x + yz$$

en

$$(\bar{x} + y)(x + z) = \bar{x}z + yz$$

volgt hieruit

$$\begin{aligned} D[f(x_0, x_1)] &= [\bar{x}_1 + (\bar{x}_0 + \bar{f}_0)(x_0 + \bar{f}_1)][x_1 + (\bar{x}_0 + \bar{f}_2)(x_0 + \bar{f}_3)] \\ &= \bar{x}_1(\bar{x}_0 + \bar{f}_2)(x_0 + \bar{f}_3) + x_1(\bar{x}_0 + \bar{f}_0)(x_0 + \bar{f}_1) \\ &= \bar{x}_1(\bar{x}_0 \bar{f}_3 + x_0 \bar{f}_2) + x_1(\bar{x}_0 \bar{f}_1 + x_0 \bar{f}_0) \\ &= \bar{x}_1 \bar{x}_0 \bar{f}_3 + \bar{x}_1 x_0 \bar{f}_2 + x_1 \bar{x}_0 \bar{f}_1 + x_1 x_0 \bar{f}_0 \\ &= x_1 x_0 \bar{f}_0 + x_1 \bar{x}_0 \bar{f}_1 + \bar{x}_1 x_0 \bar{f}_2 + \bar{x}_1 \bar{x}_0 \bar{f}_3 \\ &= \bar{x}_1 \bar{x}_0 \bar{f}_0 + \bar{x}_1 \bar{x}_0 \bar{f}_1 + \bar{x}_1 \bar{x}_0 \bar{f}_2 + \bar{x}_1 \bar{x}_0 \bar{f}_3. \end{aligned}$$

Deze laatste vorm is de mintermvorm van $\overline{f(\bar{x}_0, \bar{x}_1)}$. Hiermee is de stelling voor functies in twee variabelen bewezen.

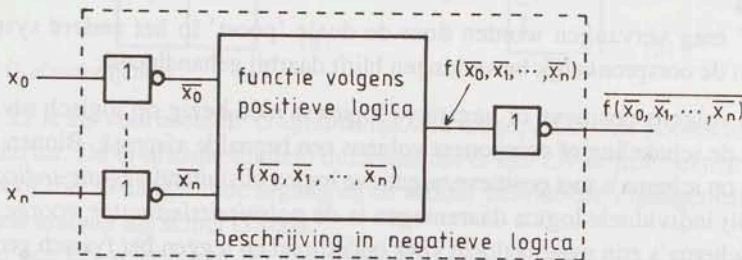
Toepassing

Een gegeven componentbeschrijving, uitgedrukt in H en L, is d.m.v. positieve logica omgezet in een symbool. Binnen dit symbool, we merken dit nogmaals op, is de specificatie van de functie 'logisch'. Bij dit symbool behoort, in positieve logica, een

bepaalde logische formule, welke de externe logische werking beschrijft. We hebben reeds eerder gezien dat de logische formule/functie van een gegeven component afhangt van de wijze van interpreteren, volgens positieve of negatieve logica (paragraaf 5.1) of individuele logica (A.8). *Op grond van de bovenstaande stelling mag nu geconcludeerd worden dat de formules, af te leiden in positieve en negatieve logica, elkaars duale vorm zijn.*

Toelichting

Wanneer in positieve en negatieve logica aan een ingang of uitgang een 'naam' (of variabele) wordt toegekend, dan zijn de met deze namen corresponderende variabelen elkaars complement. Immers, een niveau H correspondeert in het ene systeem met 'waar' en in het andere met 'niet waar'. Aan de ingangen en uitgangen kunnen we dus van het ene systeem naar het andere overgaan door toepassing van een negator aan elke ingang en uitgang. Zie figuur 19.



Figuur 19. Conversie van positieve naar negatieve logica.

Voorbeeld

x_2	x_1	x_0	S
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

x_2	x_1	x_0	S
L	L	L	H
L	L	H	H
L	H	L	H
L	H	H	H
H	L	L	H
H	L	H	H
H	H	L	H
H	H	H	L

beschrijving in H/L.

x_2	x_1	x_0	S
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	1

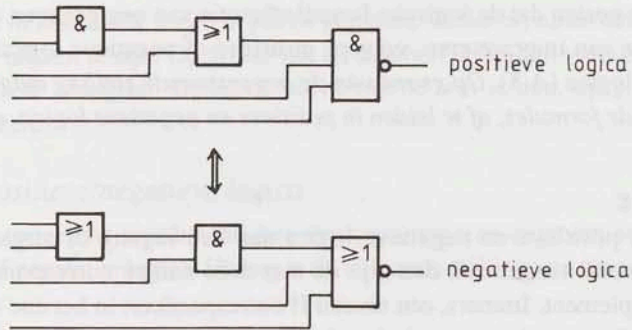
$S_1 = \overline{x_0 x_1 x_2}$

$S_2 = D[S_1]$

$S_2 = \overline{x_0 + x_1 + x_2}$

Figuur 20. Interpretatie volgens positieve en negatieve logica.

Voorbeeld



Figuur 21. Conversie schema's in positieve en negatieve logica.

Opmerking

Elke 'poort' mag vervangen worden door de duale 'poort' in het andere systeem. De volgorde van de oorspronkelijk bewerkingen blijft daarbij gehandhaafd.

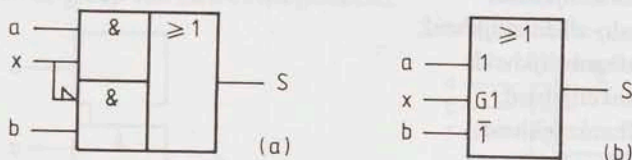
In schema's volgens positieve of negatieve logica is men bezig op logisch niveau, men interpreteert de schakeling of component volgens een bepaalde afspraak. Binnen de IEC is afgesproken op schema's met positieve/negatieve logica uitsluitend *negatie-indicatoren* toe te passen. Bij individuele logica daarentegen is de *polariteitsindicator* voorgeschreven. Dergelijke schema's zijn meer bedoeld voor onderhoud en leggen het fysisch gedrag vast.

Symbolen voor logische schakelingen (MSI-niveau)

A.11. De afhankelijkheidsnotatie

De hiervoor geïntroduceerde elementen van het symbolensysteem zijn geschikt voor de documentatie van eenvoudige poortschakelingen. Voor grotere schakelingen zijn krachtiger hulpmiddelen nodig, de symbolen worden anders te complex. De *afhankelijkheidsnotatie* is er een van. De afhankelijkheidsnotatie is een middel om onderlinge afhankelijkheden tussen ingangen en uitgangen beknopt weer te geven.

Voorbeeld



Figuur 22. G-afhankelijkheid.

In figuur 22 is als voorbeeld de *G-afhankelijkheid* toegepast op het poortschema van een $2 \Rightarrow 1$ selector. De G-afhankelijkheid duidt aan dat er een *EN-relatie* bestaat tussen de in het symbool met G aangeduide ingang en de erdoor beïnvloede ingangen met *hetzelfde identificatienummer* als achter G staat.

Bestaat er een EN-relatie met *het complement van de met G aangeduide ingang* (of uiteindelijk met het inverse signaalniveau) dan wordt dit aangeduid met een negatiestreek boven het identificatienummer. Een en ander is in figuur 22.b aangegeven.

Afhankelijkheidsnotatie

Het systeem van afhankelijkheidsnotatie kent twee soorten ingangen/uitgangen, *beïnvloedende ingangen/uitgangen* en *afhankelijke ingangen/uitgangen* (affecting inputs/outputs respectievelijk affected inputs/outputs). Voor de afhankelijkheidsnotatie geldt een aantal regels, waaronder:

- Een *beïnvloedende ingang/uitgang* draagt een *toevoegsymbool* bestaande uit een of meer hoofdletters, gevolgd door een *identificatienummer* (identifying number) genoteerd in het tientallig stelsel.
- Een *afhankelijke ingang/uitgang* draagt als toevoegsymbool het *identificatienummer* van de beïnvloedende ingang/uitgang waarvan hij afhankelijk is.
- De *letter of lettercombinatie in het toevoegsymbool* van de beïnvloedende ingang/uitgang bepaalt *welke invloed* deze ingang/uitgang op de ervan afhankelijke ingangen/uitgangen uitoefent.
- Als een ingang/uitgang *afhankelijk is van meer dan één beïnvloedende ingang/uitgang*, dan worden in het toevoegsymbool van de afhankelijke ingang/uitgang de identificatienummers vermeld van *alle* beïnvloedende ingangen/uitgangen, gescheiden door komma's, en wel van links naar rechts in de *volgorde* waarin de verschillende invloeden worden uitgeoefend.

- Indien het niet de interne logische waarde zelf van een beïnvloedende ingang/uitgang is die de door de lettercombinatie bepaalde invloed uitoefent, maar de *negatie* daarvan, dan wordt een negatiestreep geplaatst boven het identificatienummer in het toevoegsymbool van de afhankelijke ingangen/uitgangen.
- Behoudens in het geval van de zogenaamde ENABLE-afhankelijkheid is het *effect* van een beïnvloedende ingang/uitgang op de ervan afhankelijke ingangen/uitgangen *uitgedrukt* in termen van hun *interne* logische waarden.

Het systeem kent verschillende afhankelijkheden, de letter(combinatie)s ervan zijn de volgende.

- A adresafhankelijkheid;
- C commando-afhankelijkheid;
- EN enable-afhankelijkheid;
- G EN-afhankelijkheid;
- M mode-afhankelijkheid;
- N negatie-afhankelijkheid;
- R resetafhankelijkheid;
- S setafhankelijkheid;
- V OF-afhankelijkheid;
- Z hulpmiddel voor het aangeven van interne verbindingen.

Enkele ervan zullen we hierna kort behandelen. Binnen een symbool kunnen verschillende types afhankelijkheden voorkomen. Wanneer er meer dan een afhankelijkheid voorkomt, dan gelden aanvullende regels.

- Twee beïnvloedende ingangen/uitgangen met *verschillende letter(combinatie)s* hebben altijd *verschillende identificatienummers*.
- Beïnvloedende ingangen/uitgangen met *dezelfde letter(combinatie)* en hetzelfde *identificatienummer* staan in een OF-afhankelijkheid t.o.v. elkaar.
- *Identificatienummers* van beïnvloedende ingangen/uitgangen behoeven *niet opeenvolgend* gekozen te worden. Zij dienen echter zo gekozen te worden dat *geen verwarring* kan ontstaan met andere in hetzelfde symbool voorkomende getallen zoals gewichten bij ingangen en uitgangen van code-omzeters.

Met deze regels kan de onderlinge hiërarchie tussen verschillende afhankelijkheden bepaald worden.

Tot slot nog een regel die enige toelichting behoeft.

- Een beïnvloedende ingang/uitgang heeft uitsluitend een effect op de ervan afhankelijke ingangen/uitgangen. Het is geen 'normale' ingang van het element waaraan hij is getekend.

In figuur 22.b heeft de G1-ingang geen invloed op de OF-functie, zoals die in het symbool is voorgesteld, althans niet direct. G1 doet dit uitsluitend bij aanwezigheid van signaal a of van signaal b.

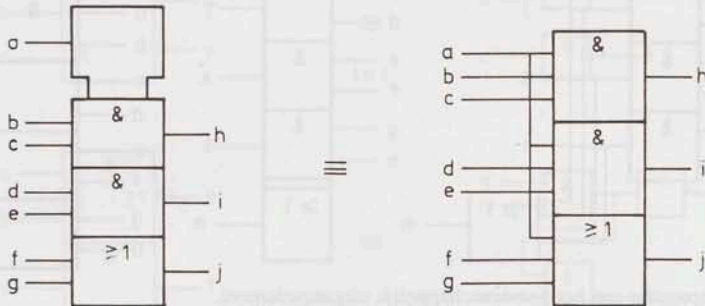
Opmerking

De afhankelijkheidsnotatie wordt bij voorkeur niet gebruikt bij zeer kleine schakelingen, op het poortniveau bijvoorbeeld.

A.12. Gemeenschappelijke ingangen en uitgangen

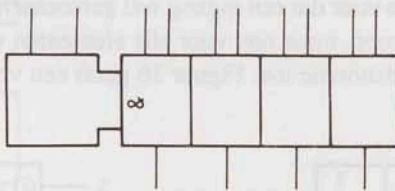
Het gemeenschappelijk ingangsblok

Indien een groep (array) van aan elkaar getekende, doch niet noodzakelijk gelijke, elementen een of meer gemeenschappelijke ingangen bezit, dan kunnen deze ingangen getekend worden aan een *gemeenschappelijk ingangsblok*. Dit blok wordt aan één der uiteinden van de groep van elementen getekend.



Figuur 23. Toepassing van het gemeenschappelijk ingangsblok.

Ingang a in figuur 23 is een gemeenschappelijke ingang van alle elementen van de groep. Vooral als er meer gemeenschappelijke ingangen zijn biedt toepassing van het gemeenschappelijk ingangsblok voordelen, omdat een groot aantal lijnkruisingen hierdoor kan worden vermeden. Figuur 24 toont de toepassing van een gemeenschappelijk ingangsblok met twee ingangen op een horizontaal georiënteerde groep van elementen.



Figuur 24. Toepassing van het gemeenschappelijk ingangsblok.

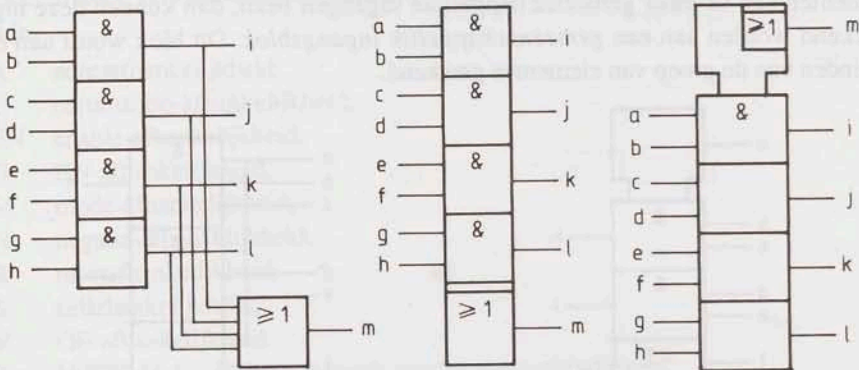
Vaak zijn gemeenschappelijke ingangen afkomstig van de *besturing* (control) van de schakeling. Het gemeenschappelijk ingangsblok heet daarom ook wel *gemeenschappelijk besturingsblok* (common control block). Door consequent alle besturingssignalen hierop aan te sluiten kan een overzichtelijke tekening ontstaan, met een duidelijke scheiding tussen datapad en besturing.

Opmerking

Wanneer een gemeenschappelijk ingangsblok wordt toegepast bij een groep van elementen waarin een bepaalde rangorde schuilt, gewichten van uitgangen bijvoorbeeld, dan wordt het element met de uitgang met de laagste rangorde het dichtst op het ingangsblok getekend.

Het gemeenschappelijk uitgangselement

Vaak bezitten schakelingen die opgebouwd zijn uit een groep van elementen behalve de uitgangen van de elementen zelf ook uitgangen die een functie zijn van de uitgangen van *alle* elementen. In dat geval kan men gebruik maken van een of meer *gemeenschappelijke uitgangselementen* (common output elements). Dit element dat bestaat uit een rechthoek waarvan een zijde dubbel getekend is, wordt aan één van de uiteinden van de groep van symbolen getekend, daarvan gescheiden door genoemde dubbele streep, of in het gemeenschappelijk ingangsblok.

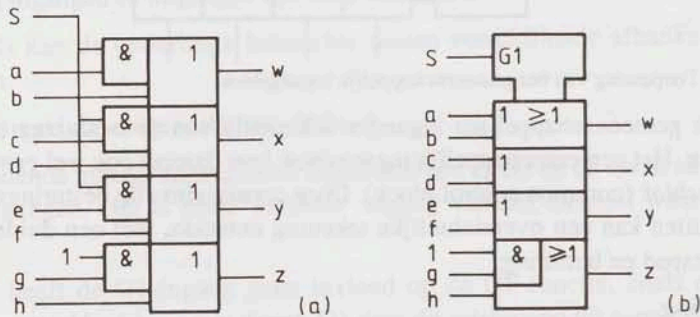


Figuur 25. Toepassing van het gemeenschappelijk uitgangselement.

Het gemeenschappelijk uitgangselement bespaart veel kruisende lijnen in tekeningen. Bedenken we daarnaast dat een gemeenschappelijk uitgangselement ook externe ingangen mag bezitten, dan is het een middel dat de overzichtelijkheid van een tekening sterk kan bevorderen.

Toepassing van de afhankelijkheidsnotatie op het gemeenschappelijk ingangsblok of gemeenschappelijk uitgangselement.

Vaak doet zich het probleem voor dat een ingang wel gemeenschappelijk is voor een groot aantal elementen van een groep, maar niet voor alle elementen van de groep. In dat geval past men de afhankelijkheidsnotatie toe. Figuur 26 geeft een voorbeeld voor G-afhankelijkheid.



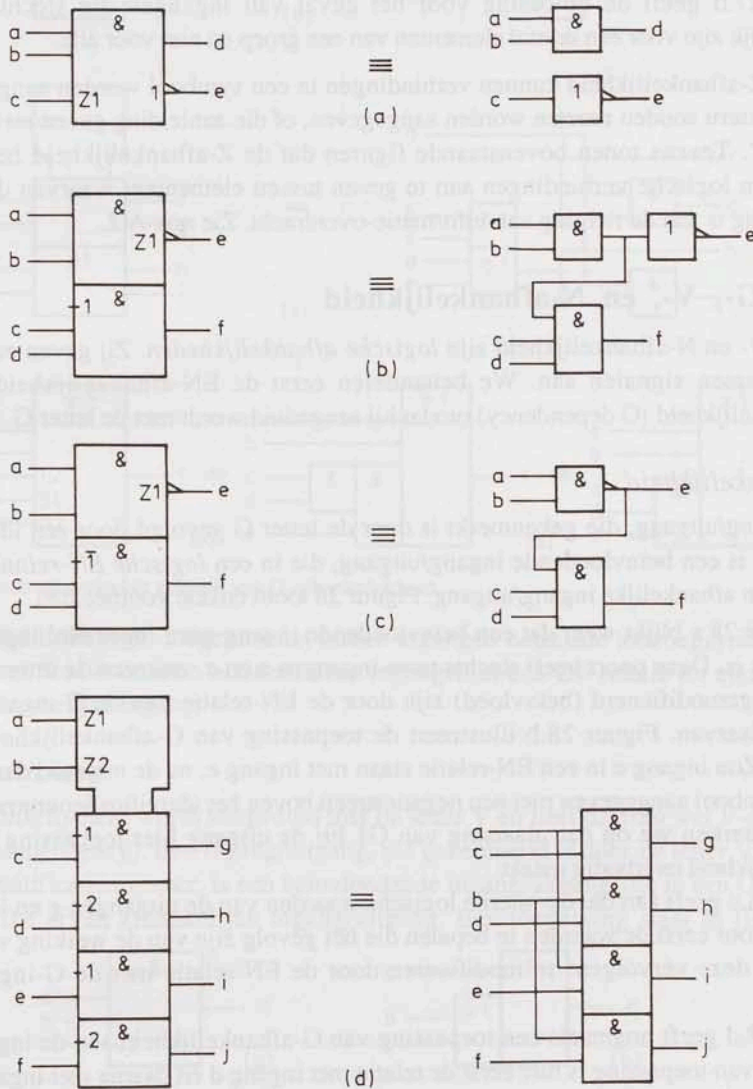
Figuur 26. Afhankelijkheidsnotatie in gemeenschappelijk ingangsblok.

Afhankelijkheidsnotatie, in combinatie met het gemeenschappelijk ingangsblok en het gemeenschappelijk uitgangselement, vormt een machtig hulpmiddel om schakelingen

compact te kunnen weergeven in tekeningen.

A.13. Z-afhankelijkheid

De eerste soort afhankelijkheid die we zullen bekijken is de *verbindingsafhankelijkheid* (*interconnection dependency*), ook wel *Z-afhankelijkheid* (*Z dependency*) genoemd omdat deze wordt aangeduid met de letter Z. Een ingang/uitgang die gekenmerkt is door de letter Z gevolgd door een identificatienummer, is een beïnvloedende ingang/uitgang die zijn interne logische waarde *opdrukt* op de ervan afhankelijke ingangen/uitgangen. Figuur 27 toont enkele voorbeelden.



Figuur 27. Illustraties van het gebruik van Z-afhankelijkheid.

Figuur 27.a illustreert dat de beïnvloedende ingang c geen 'normale' ingang van de AND-poort is, maar uitsluitend zijn inwendige logische waarde oplegt aan de ervan afhankelijke uitgang e. Daarmee is uitgang e tevens onafhankelijk van de ingangen a en b; c is onafhankelijk van de werking van de AND-poort.

In de figuren 27.b t/m 27.d is gebruik gemaakt van een kort streepje om een zogenaamde *inwendige ingang* aan te duiden. Hoewel een dergelijke inwendige ingang volgens de IEC-afspraken geacht wordt steeds de interne logische waarde 1 te bezitten, wordt deze 1 hier overheerst door de invloed van de Z-afhankelijkheid.

De figuren 27.b en 27.c tonen duidelijk dat het de *interne logische waarde* van de beïnvloedende Z-uitgang is die op de afhankelijke ingang wordt overgedragen.

Figuur 27.d geeft de oplossing voor het geval van ingangen die slechts gemeenschappelijk zijn voor een *aantal* elementen van een groep en niet voor alle.

Met de Z-afhankelijkheid kunnen verbindingen in een symbool worden aangegeven die anders extern zouden moeten worden aangegeven, of die aanleiding geven tot 'embedded symbols'. Tevens tonen bovenstaande figuren dat de Z-afhankelijkheid het mogelijk maakt om logische verbindingen aan te geven tussen elementen, waarvan de grenslijn evenwijdig is aan de richting van informatie-overdracht. Zie ook A.2.

A.14. G-, V-, en N-afhankelijkheid

De G-, V- en N-afhankelijkheid zijn *logische afhankelijkheden*. Zij geven een logische relatie tussen signalen aan. We behandelen eerst de EN-afhankelijkheid, ook wel G-afhankelijkheid (G dependency) omdat hij aangeduid wordt met de letter G.

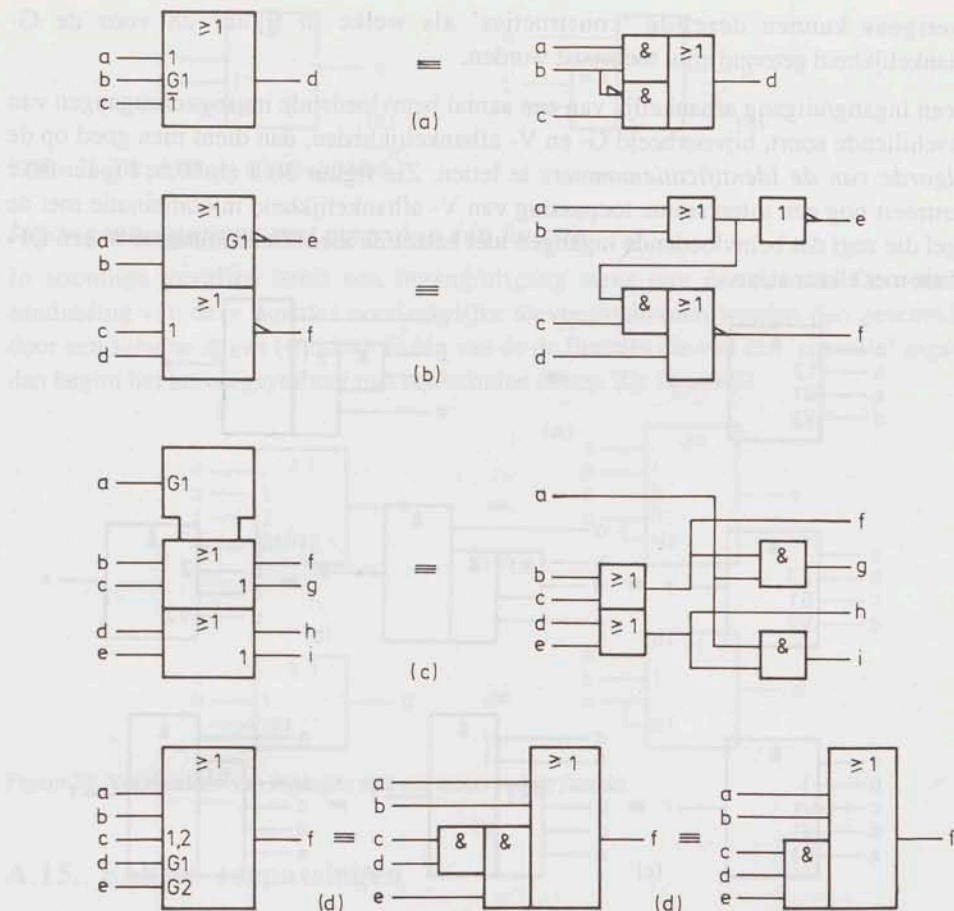
G-afhankelijkheid

Een ingang/uitgang, die gekenmerkt is door de letter G gevolgd door een identificatienummer, is een beïnvloedende ingang/uitgang, die in een *logische EN-relatie* staat met elke ervan afhankelijke ingang/uitgang. Figuur 28 toont enkele voorbeelden.

Uit figuur 28.a blijkt weer dat een beïnvloedende ingang geen 'normale' ingang van de OR-poort is. Deze poort heeft slechts twee ingangen a en c, waarvan de interne logische waarden gemodificeerd (beïnvloed) zijn door de EN-relatie met de G-ingang resp. de negatie daarvan. Figuur 28.b illustreert de toepassing van G-afhankelijkheid aan een uitgang. Zou ingang c in een EN-relatie staan met ingang e, na de negator, dan wordt dit in het symbool aangegeven met een negatiestreep boven het identificatienummer ($1 \Rightarrow 1$). Tevens merken we op dat plaatsing van G1 bij de uitgang hier toepassing van de Z-afhankelijkheid overbodig maakt.

Figuur 28.c geeft aan dat de interne logische waarden van de uitgangen g en i verkregen worden door eerst de waarden te bepalen die het gevolg zijn van de werking van de OR-poort en deze vervolgens te modifieren door de EN-relatie met de G-ingang toe te passen.

Figuur 28.d geeft nogmaals een toepassing van G-afhankelijkheid aan de ingangen. De volgorde van toepassing is hier eerst de relatie met ingang d en daarna met ingang e. In de volgorde 1,2 van het toevoegsymbool wordt dit tot uitdrukking gebracht, volgens een eerdere afspraak. Overigens maakt in dit voorbeeld de volgorde voor het beschrijven van de logische functie van de schakeling niets uit.

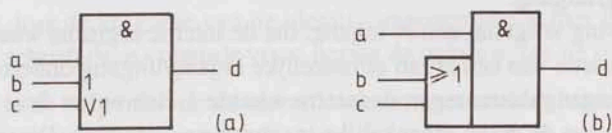


Figuur 28. Illustraties van het gebruik van G-afhankelijkheid.

In figuur 28.d is de regel toegepast dat indien ingangen hetzelfde toevoegsymbool met identiek identificatienummer bezitten, deze ingangen in een OF-relatie tot elkaar staan m.b.t. de afhankelijke ingang.

V-afhankelijkheid

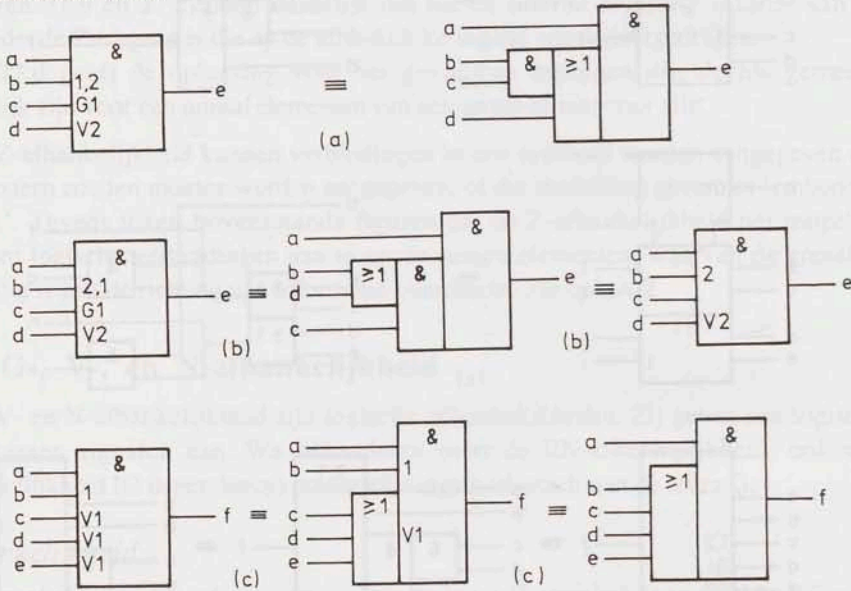
De *OF-afhankelijkheid* wordt aangeduid met de letter V en heet daarom wel *V-afhankelijkheid* (V dependency). Een ingang/uitgang, die gekenmerkt is door de letter V gevolgd door een identificatienummer, is een beïnvloedende ingang/uitgang, die in een OF-relatie staat met elke ervan afhankelijke ingang/uitgang. Een toepassing staat in figuur 29.



Figuur 29. Toepassing V-afhankelijkheid.

Overigens kunnen dezelfde 'constructies' als welke in figuur 28 voor de G-afhankelijkheid getoond zijn, toegepast worden.

Is een ingang/uitgang afhankelijk van een aantal beïnvloedende ingangen/uitgangen van verschillende soort, bijvoorbeeld G- en V- afhankelijkheden, dan dient men goed op de *volgorde van de identificatienummers* te letten. Zie figuur 30.a en 30.b. Figuur 30.c illustreert nog een interessante toepassing van V-afhankelijkheid in combinatie met de regel die zegt dat beïnvloedende ingangen met hetzelfde identificatienummer in een OF-relatie met elkaar staan.



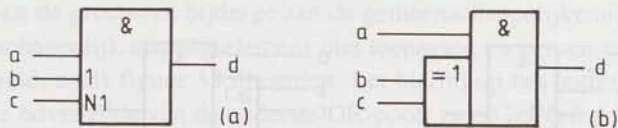
Figuur 30. Voorbeelden van meervoudige afhankelijkheid.

N-afhankelijkheid

Het IEC-systeem kent twee manieren om een negatie aan te geven. Allereerst kan een identificatienummer bij een afhankelijke ingang/uitgang voorzien worden van een negatiestreep. De ingang/uitgang is dan afhankelijk van het complement van de beïnvloedende ingang/uitgang.

Daarnaast kent het systeem ook de *negatie-afhankelijkheid* (**negate dependency**), ook wel *N-afhankelijkheid* genoemd omdat hij aangeduid wordt met de letter N. Een ingang/uitgang, die gekenmerkt is door de letter N gevolgd door een identificatienummer, is een beïnvloedende ingang/uitgang, die in een EXCLUSIEVE OF-relatie staat met elke ervan afhankelijke ingang/uitgang.

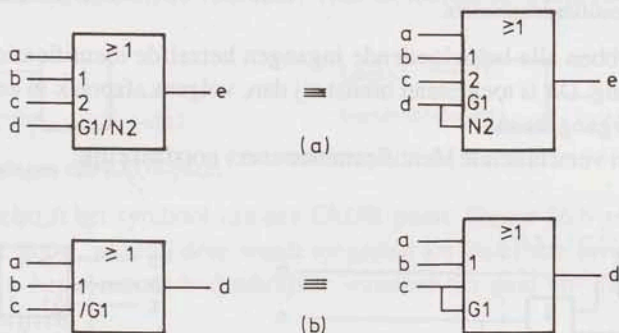
Uit deze omschrijving volgt dat een N-ingang, die de interne logische waarde 0 heeft, de interne logische waarde van een ervan afhankelijke ingang/uitgang onberoerd laat. Heeft de beïnvloedende ingang daarentegen de interne waarde 1, dan vormt deze de negatie van de logische waarde van de ervan afhankelijke ingangen en uitgangen. Dit verklaart ook de naam. Zie figuur 31.



Figuur 31. Toepassing N-afhankelijkheid.

Ingangen/uitgangen met meer dan één functie

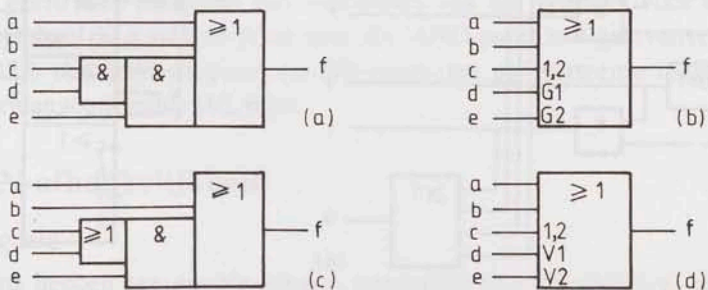
In sommige gevallen heeft een ingang/uitgang *meer dan één functie*. De voor de aanduiding van deze functies noodzakelijke toevoegsymbolen worden dan gescheiden door een *schuine streep* (solidus). Is één van de de functies die van een 'normale' ingang, dan begint het toevoegsymbool met een schuine streep. Zie figuur 32.



Figuur 32. Voorbeelden van ingangen met een meervoudige functie.

A.15. Enkele toepassingen

Voorbeeld

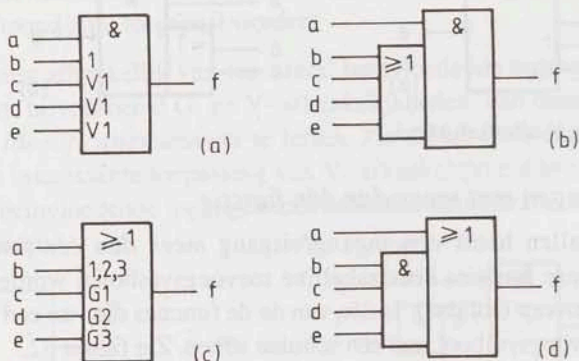


Figuur 33. Volgorde identificatienummers.

In figuur 33.b doet de volgorde van de identificatienummers achter ingang c er niet toe. Het normblad schrijft deze volgorde voor, gezien de opbouw van de schakeling.

In figuur 33.d is de volgorde van de identificatienummers achter ingang c essentieel voor een juiste interpretatie van het symbool.

Voorbeeld

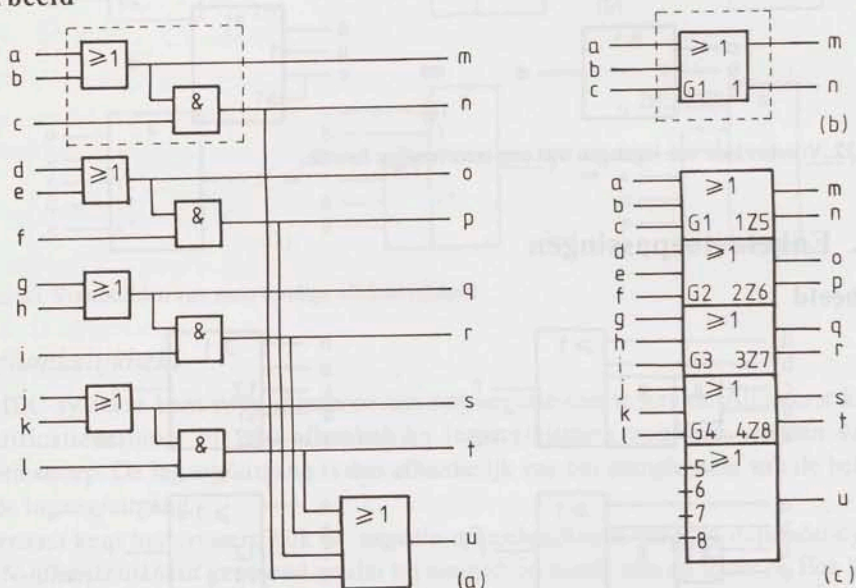


Figuur 34. Keuze identificatienummers.

In figuur 34.a hebben alle beïnvloedende ingangen hetzelfde identificatienummer als de afhankelijke ingang. Dit is toegestaan omdat zij dan, volgens afspraak in een OF-relatie tot de afhankelijke ingang staan.

In figuur 34.c zijn verschillende identificatienummers noodzakelijk.

Voorbeeld



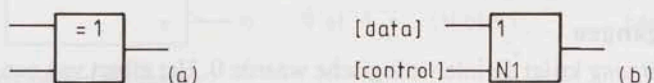
Figuur 35. Toepassing Z-afhankelijkheid op een gemeenschappelijke uitgang.

Voor het gemeenschappelijk uitgangselement geldt de regel dat van elk element van de groep precies één uitgang als ingang van het gemeenschappelijk uitgangselement fungeert. De interne logische waarde van deze ingang is dan gelijk aan die van de betreffende uitgang. Uit deze regel volgt dat het gebruik van het gemeenschappelijk uitgangselement alleen dan is toegestaan als van *elk* element van de groep *alle* uitgangen steeds dezelfde interne logische waarde hebben. Anders zou er immers onzekerheid bestaan van welke uitgang van een element de logische waarde op de ingang van het gemeenschappelijk uitgangselement wordt overgedragen. Is aan deze eis niet voldaan of leveren niet alle

elementen van de groep een bijdrage aan de gemeenschappelijke uitgang, dan mag men het gemeenschappelijk uitgangselement niet toepassen en is men aangewezen op de Z-afhankelijkheid, zoals figuur 35 illustreert. Let hierbij op het ontbreken van de dubbele streep aan de bovenzijde van de onderste OR-poort en op het feit dat de interne logische waarde, die een Z-uitgang overdraagt, de waarde is die ontstaat na toepassing van de EN-relatie met de betreffende G-ingang.

Voorbeeld

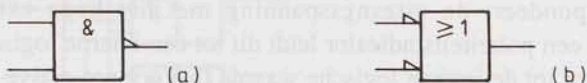
Afhankelijk van het ontwerpstadium worden aan symbolen in tekeningen verschillende eisen gesteld. Ook kunnen schakelingen voor meer dan één toepassing gebruikt worden. Men wordt dan geconfronteerd met de vraag welke eisen aan het symbool prevaleren, standaardisatie of duidelijkheid bijvoorbeeld. Voor documentatie van realisatieschema's verdienen standaardsymbolen de voorkeur. Voor de meeste IC's zijn standaardsymbolen voorgesteld.



Figuur 36. Toepassingen van EXOR-poort.

Figuur 36.a beschrijft het symbool van een EXOR-poort. Figuur 36.b geeft een symbool voor een EXOR-poort, waarbij deze wordt toegepast als instelbare inverter/negator. De laatste vorm van het symbool is duidelijker wanneer het gaat om de *functie* van het element te beschrijven.

Voorbeeld



Figuur 37. Symbolen voor een poort van het circuit SN7400.

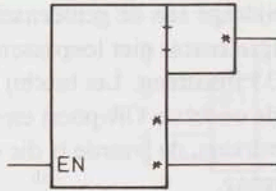
Figuur 37 geeft twee manieren van voorstellen van het symbool voor een '2- input NAND'. Symbool 37.a stelt de poort voor als 'AND-poort met geïnverteerde uitgang'. Symbool 37.b beschrijft de poort als OR-poort met geïnverteerde ingangen'. Beide symbolen leiden tot dezelfde H/L tabel.

A.16. EN-afhankelijkheid

De EN-ingang

Vele circuits hebben een *enable-ingang*, aangeduid met EN. Een EN-ingang stelt de uitgangen in op hun normale werking, met uitgangsniveau H of L, of stuurt ze naar bijvoorbeeld een toestand van zeer hoge impedantie, zoals bij 3-state uitgangen. Op de EN-ingang zijn de volgende afspraken van toepassing.

Een enable-ingang, aangeduid door het toevoegsymbool EN, heeft uitsluitend invloed op de uitgangen van het element en wel op *alle* uitgangen. Bij een samengesteld element zijn dit alle *naar buiten gevoerde* uitgangen en niet de eventuele inwendige uitgangen van de rechthoek die naar interne ingangen gaan. In figuur 38 bijvoorbeeld staan slechts de met * gemerkte uitgangen onder besturing van de EN-ingang.



Figuur 38. Uitgangen, beïnvloed door een EN-ingang.

Het effect van een EN-ingang op een uitgang is als volgt gedefinieerd.

Een EN-ingang die de interne logische waarde 1 heeft, laat alle uitgangen *onberoerd*. Zij bezitten dan hun normaal gedefinieerde logische waarde en hebben het normale effect op de erdoor gestuurde elementen.

Een EN-ingang die de interne logische waarde 0 heeft, oefent de volgende uitwerking op de erdoor beïnvloede uitgangen uit.

Normale uitgangen

Een *normale uitgang* krijgt de interne logische waarde 0. Het effect van een EN-ingang op een normale uitgang is gelijk aan dat van een G-ingang.

Open-circuit uitgangen

Een *open-circuit uitgang* wordt naar een toestand van zeer hoge impedantie gestuurd. Een open-circuit uitgang krijgt de interne logische waarde, die overeenkomt met het externe niveau met de zeer hoge impedantie.

Nemen we als voorbeeld een active low/passive pull-up uitgang. Wanneer de impedantie hoog is, correspondeert de uitgangsspanning met het hoge externe niveau. Bij afwezigheid van een polariteitsindicator leidt dit tot een interne logische waarde 1, met polariteitsindicator tot de interne logische waarde 0. Voor een active high/passive pull-down gelden overeenkomstige redeneringen.

3-State uitgangen

Een 3-state uitgang wordt naar een externe toestand van zeer hoge impedantie gestuurd. *Intern* behoudt een 3-state uitgang zijn normale interne logische waarde, deze waarde wordt *niet* beïnvloed door de EN-ingang.

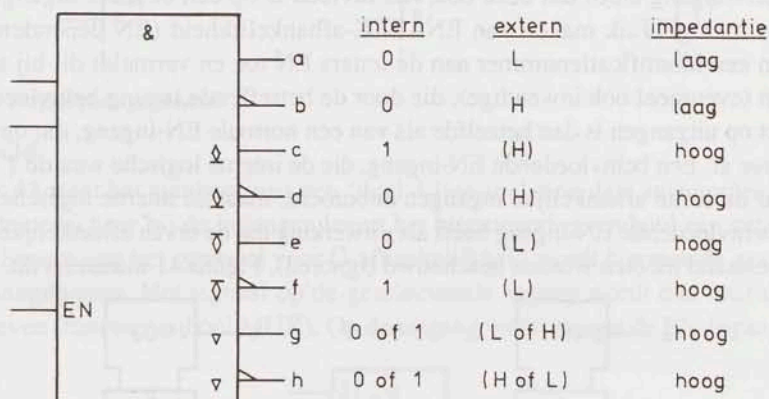
Opmerking

Open-circuit uitgangen en 3-state uitgangen verschillen dus duidelijk m.b.t. de definitie van de interne logische waarde van de uitgang. Bij open-circuit uitgangen bepaalt de uitgangsconstructie bij $EN = 0$ de interne logische waarde, bij 3-state uitgangen is het de logische functie van het element die de interne logische waarde bepaalt.

Figuur 39 licht een en ander toe. Hierbij is aangenomen dat de enable-ingang EN intern 0 is. Alle open-circuit uitgangen zijn *disabled*. Het is dan eigenlijk niet van belang welk extern logisch niveau zij aannemen, omdat dit wegens de toestand van zeer hoge impedantie toch geen effect heeft op de elementen waarmee zij verbonden zijn. Deze niveaus zijn daarom tussen haken geplaatst.

Men kan zich afvragen waarom de interne logische waarden van de uitgangen c t/m h van belang zijn. Hun uitgangsimpedantie is immers te hoog om enig effect te hebben. Men

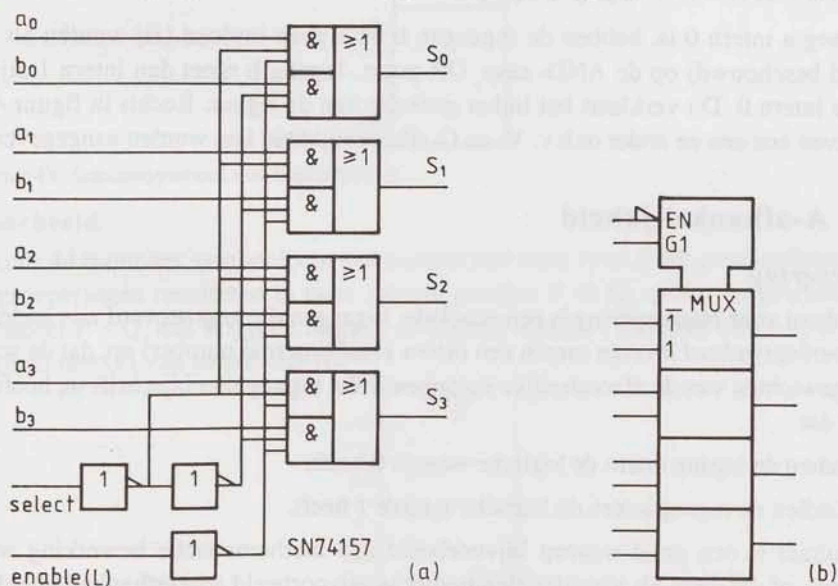
bedenke echter dat deze uitgangen eventueel ook intern als beïnvloedende uitgangen/ingangen kunnen optreden en dan is juist hun interne logische waarde bepalend voor hun invloed.



Figuur 39. Invloed van een EN-ingang.

Voorbeeld

Figuur 40 toont het uitvoeringsschema van een viervoudige $2 \Rightarrow 1$ selector/multiplexer van het type SN74157. In het bijbehorende symbool is gebruik gemaakt van G-afhankelijkheid aan de ingangen en EN-afhankelijkheid aan de uitgangen.

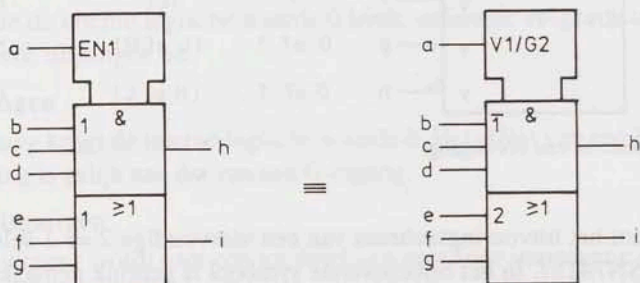


Figuur 40. Toepassing G/EN-afhankelijkheid.

Hoewel intern beide sturingangen op hetzelfde niveau en op dezelfde poorten ingrijpen is vanwege een *gemakkelijke interpretatie* van het symbool gekozen voor verschillende afhankelijkheidsnotaties.

EN-afhankelijkheid

Uiteraard kan het voorkomen dat niet alle uitgangen van een element onder besturing staan van een EN-ingang en/of dat deze ook van invloed is op een of meer ingangen. In dat geval kan men gebruik maken van ENABLE-afhankelijkheid (EN dependency). Men voegt dan een identificatienummer aan de letters EN toe en vermeldt dit bij alle in- en uitgangen (eventueel ook inwendige), die door de betreffende ingang beïnvloed worden. Het effect op uitgangen is dan hetzelfde als van een normale EN-ingang, dat op ingangen wijkt echter af. Een beïnvloedende EN-ingang, die de interne logische waarde 1 heeft, laat weliswaar de ervan afhankelijke ingangen onberoerd, maar de interne logische waarde 0 van een beïnvloedende EN-ingang heeft als uitwerking dat de ervan afhankelijke ingangen als niet bestaand moeten worden beschouwd (ignored). Figuur 41 illustreert dit.



Figuur 41. Invloed van een EN-ingang op ingangen.

Als ingang *a* intern 0 is, hebben de ingangen *b* en *e* geen invloed (zij worden als niet bestaand beschouwd) op de AND- resp. OR-poort. Ingang *b* moet dan intern 1 zijn en ingang *e* intern 0. Dit verklaart het linker gedeelte van de figuur. Rechts in figuur 41 is aangegeven hoe een en ander m.b.v. V- en G-afhankelijkheid kan worden aangegeven.

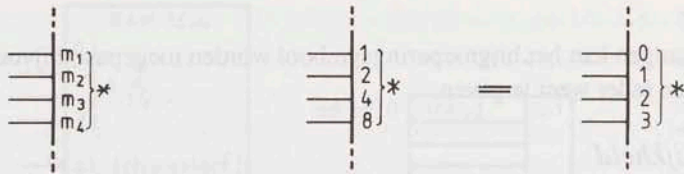
A.17. A-afhankelijkheid

Bitgroepering

Het symbool voor bitgroepering is een accolade. Ingangen die gegroepeerd zijn onder het *bitgroeperingsymbool* leveren samen een *intern getal* (internal number) op, dat de som is van de gewichten van de afzonderlijke ingangen. Elke ingang met bijschrift m_i heeft een gewicht dat

- 0 is indien de ingang intern de logische waarde 0 heeft;
- m_i is indien de ingang intern de logische waarde 1 heeft.

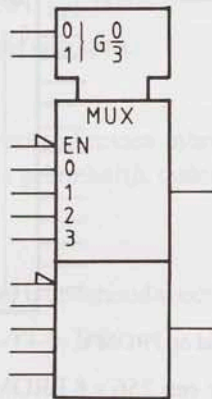
Het resultaat is een getal waarop bijvoorbeeld een mathematische bewerking wordt uitgevoerd, of dat dient als identificatienummer in bijvoorbeeld adresafhankelijkheid. Bij het bitgroeperingsymbool moeten de gewichten der ingangen, de m_i 's, in opklimmende of afnemende volgorde genoteerd worden. Zijn alle gewichten binair, dan kan volstaan worden met het noteren van de exponent van het grondtal 2. Anders moet het decimale equivalent van het gewicht vermeld worden. Zie figuur 42.



Figuur 42. Bitgroeperingsymbool.

Voorbeeld

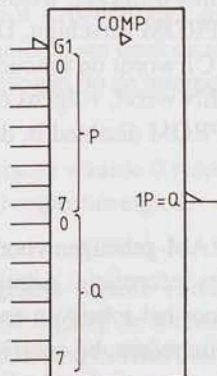
In figuur 43 staat het symbool van een 'dual 4-line-to-1-line data selector/multiplexer'. Gezien de gewichten bij de ingangen levert het bitgroeperingsymbool een getal tussen 0 en 3 op. Samen met het symbool voor G-afhankelijkheid wordt hiermee de geselecteerde ingang aangewezen. Het signaal op de geselecteerde ingang wordt dan naar de uitgang doorgegeven (functiesymbool MUX). Op de uitgang werkt tevens de EN-ingang.



Figuur 43. Selectorsymbool met bitgroepering.

Voorbeeld

Figuur 44 toont het symbool van een comparator voor twee 8-bit getallen/bitstrings. De bitgroeperingen resulteren in twee interne getallen P en Q, welke vergeleken worden. Wanneer $P = Q$, dan krijgt de uitgang de interne waarde 1 (waarop ook nog G-afhankelijkheid met G1 van toepassing is).



Figuur 44. Comparatorsymbool met bitgroepering.

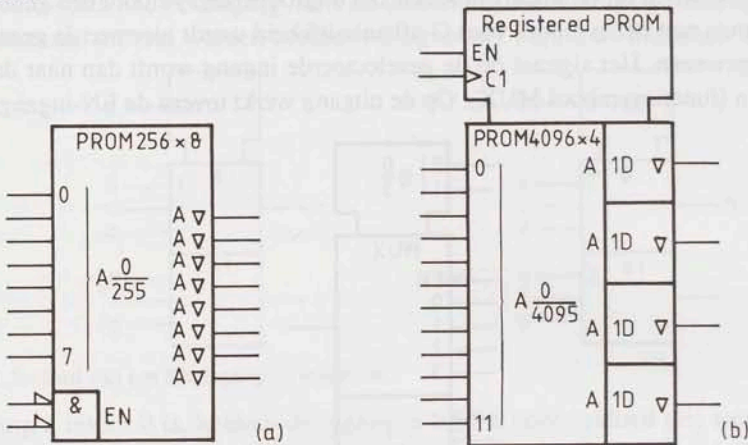
Opmerking

Ook aan uitgangen kan het bitgroeperingsymbool worden toegepast, bijvoorbeeld om de inhoud van een teller weer te geven.

A-afhankelijkheid

Met adresafhankelijkheid in combinatie met bitgroepering kan men o.a. geheugenarrays beknopt weergeven. Adresafhankelijkheid wordt weergegeven met de letter A. We introduceren adresafhankelijkheid aan de hand van enkele toepassingen.

Voorbeeld



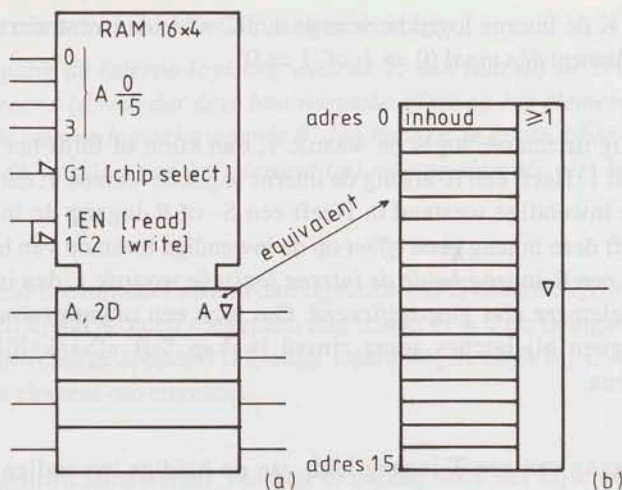
Figuur 45. Toepassing adresafhankelijkheid op PROM's.

Figuur 45.a toont het symbool voor een 256×8 PROM-circuit. Het interne getal van het bitgroeperingsymbool wordt met adresafhankelijkheid omgezet in een adres, lopend van 0 t/m 255. De inhoud van het aangewezen adres verschijnt dan op de acht uitgangen. Wanneer adresafhankelijkheid wordt toegepast, dan behoeven niet alle 256 adressen getekend te worden. Bij toepassing van G-afhankelijkheid bijvoorbeeld zou dit wel nodig zijn.

In figuur 45.b is het symbool van een 4096×4 registered PROM getekend. Het aangeboden adres wordt in een intern adres omgezet, waarna de 4-bit inhoud van het aangewezen adres aan de uitgang van de PROM verschijnt. Daar bevindt zich een 4-bit register, met 3-state uitgangen. Op de klok C1 wordt de inhoud van het aangewezen adres in het register geklokt. De enable-ingang EN werkt, volgens een eerdere afspraak, uitsluitend op de externe uitgangen. Wanneer de PROM disabled is, dan is het toch mogelijk het register opnieuw te laden.

Voorbeeld

Het symbool in figuur 46 stelt een RAM-geheugen voor. De vier adresingangen wijzen 16 adressen aan, elk bevattend vier bits informatie. Adresafhankelijkheid wordt hier tweemaal toegepast, aan de ingangen voor het schrijven en aan de uitgangen voor het lezen. Bij lezen is de 3-state uitgang laag-impedant, bij schrijven hoog-impedant. (De betekenis van de aanduiding C2 bij de schrijfingang wordt toegelicht bij C-afhankelijkheid hierna.)



Figuur 46. Toepassing adresafhankelijkheid op RAM's.

Opmerking

Bij een ROM of PROM kan het verband tussen inhoud en uitgangswaarden niet in het symbool worden beschreven. Het is gebruikelijk onder de functie-aanduiding een verwijzing op te nemen naar een tabel,

[Tn],

waarmee wordt aangegeven dat tabel n de inhoud specificeert. Alternatieven zijn

[PROM Tn] of [PAL/PLA Tn].

A.18. C-afhankelijkheid

Geheugenwerking van latches en flip-flops wordt vastgelegd met behulp van hun *interne toestand*. Verschillende instellingen hebben, op hun eigen manier, invloed op de interne toestand. Daarnaast verschilt deze invloed voor latches, gated latches en flip-flops, hetgeen o.a. tot uitdrukking komt in de concepten *voorbereidende* en *directe instellingen* (preparatory en direct acting inputs). Een en ander wordt aangegeven met *besturingsafhankelijkheid* (control dependency), aangeduid met *C-afhankelijkheid*. Alvorens deze te introduceren definiëren we eerst de invloed van D-ingangen, J- en K-ingangen, S- en R-ingangen en T-ingangen op de interne toestand van een element.

D-ingang

Heeft een D-ingang de interne logische waarde 0 respectievelijk 1, dan komt of blijft het element in de inwendige toestand 0 respectievelijk 1.

J- en K-ingang

Heeft een J-ingang de interne logische waarde 1, dan komt of blijft het element in de inwendige toestand 1. Heeft een K-ingang de interne logische waarde 1, dan komt of blijft het element in de inwendige toestand 0. Heeft een J- of K-ingang de interne logische waarde 0, dan heeft deze ingang *geen effect* op de inwendige toestand van het element.

Heeft zowel J als K de interne logische waarde 1, $JK = 11$, dan verandert de inwendige toestand van het element *één maal* ($0 \Rightarrow 1$ of $1 \Rightarrow 0$).

S- en R-ingang

Heeft een S-ingang de interne logische waarde 1, dan komt of blijft het element in de inwendige toestand 1. Heeft een R-ingang de interne logische waarde 1, dan komt of blijft het element in de inwendige toestand 0. Heeft een S- of R-ingang de interne logische waarde 0, dan heeft deze ingang *geen effect* op de inwendige toestand van het element.

Hebben een S- en een R-ingang *beide de interne logische waarde 1*, dan is de inwendige toestand van het element *niet gespecificeerd*. Om toch een uitgangswaarde te kunnen specificeren, hetgeen bij latches soms zinvol is, kan S-R afhankelijkheid worden toegepast. Zie hierna.

T-ingang

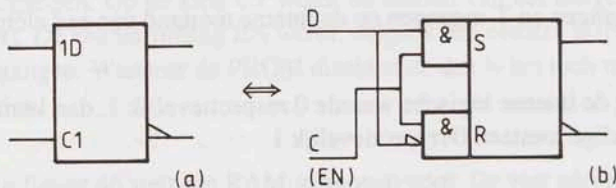
Elke $0 \Rightarrow 1$ overgang van een T-ingang leidt van de huidige inwendige toestand naar diens complement. De overige waarden/overgang van T hebben geen invloed op de inwendige toestand van het element.

Volgens bovenstaande omschrijvingen zijn de ingangen gedefinieerd als direct werkende ingangen. Zij bepalen niet alleen wat er gebeurt, maar ook wanneer het gebeurt. Bij flip-flops hebben J- en K-ingangen en D-ingangen daarentegen altijd het karakter van een voorbereidende ingang en S- en R-ingangen soms. Het al dan niet actief zijn ervan wordt dan bepaald door een klokingang, of algemener een *commando-ingang*.

C-afhankelijkheid

Wanneer het wel/niet actief zijn van ingangen afhankelijk is van andere ingangen, dan kan dit worden aangegeven met *besturingsafhankelijkheid*, aangeduid met C-afhankelijkheid. De klokingang bij flip-flops en de enable-ingang bij gated latches wordt dan voorzien van de letter C, gevolgd door een *identificatienummer*. Deze beïnvloedende C-ingang heeft geen direct effect op de werking/toestand van het element. Een C-ingang beïnvloedt deze werking slechts via de ervan afhankelijke ingangen S en R, J en K, D of T en wel door deze al dan niet actief te maken. Bij de afhankelijke ingangen wordt dan het identificatienummer van de hen beïnvloedende C-ingang geplaatst en wel links van de letters S, R, J, K, D of T.

Voorbeeld



Figuur 47. Gated D-latch.

Wanneer in figuur 47 de interne logische waarde van de met C1 aangeduide ingang 1 is, dan is de inwendige toestand gelijk aan de interne logische waarde van de D-ingang. Is de interne logische waarde van C1 gelijk aan 0, dan heeft D geen effect. De inwendige toestand blijft onveranderd, het element onthoudt.

De werking van een beïnvloedende C-ingang is als volgt:

Heeft een C-ingang de interne logische waarde 1, dan laat hij de ervan afhankelijke ingangen onberoerd (d.w.z. dat deze hun normale effect op het element hebben). Heeft een C-ingang de interne logische waarde 0, dan hebben de ervan afhankelijke ingangen geen effect op de werking van het element (zij moeten dan als niet bestaand worden beschouwd).

Opmerking

C-afhankelijkheid is essentieel anders dan bijvoorbeeld G-afhankelijkheid. Zou in figuur 47.a G-afhankelijkheid worden toegepast, dan maakt $G = 0$ de D-ingang 0, hetgeen het element in de inwendige toestand 0 brengt. Daarentegen blijft bij $C = 0$ de inwendige toestand van het element onveranderd.

Opmerking

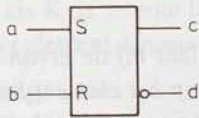
Wanneer het identificatienummer van een C-ingang voor het bijschrift van een ervan afhankelijke ingang voorzien is van een *negatiestreek*, dan activeert $C = 0$ de betreffende ingang juist.

A.19. Symbolen voor latches

Bij de hiervoor gegeven definities van het effect van een S- en een R-ingang is gesteld dat de inwendige toestand van een element, waarvan zowel een S- als een R-ingang de interne logische waarde 1 heeft, niet gespecificeerd is. Dit is in overeenstemming met het feit dat de S-R latch en de $\bar{S}\bar{R}$ latch, die meestal gerealiseerd zijn met twee NOR's resp. twee NAND's, bij de ingangscombinatie $SR = 11$ (of $\bar{S}\bar{R} = 00$) geen ondubbelzinnig bepaalde inwendige toestand bezitten. Wordt de combinatie $SR = 11$ gevolgd door $SR = 10$ of $SR = 01$, dan is de inwendige toestand na deze ingangsverandering weer volledig bepaald. Wordt $SR = 11$ echter gevolgd door $SR = 00$, dan hangt de nieuwe inwendige toestand af van kleine verschillen in de fysische eigenschappen van de individuele componenten, waaruit het element is opgebouwd. Dit is de reden waarom toepassing van de overgang van $SR = 11$ naar $\bar{S}\bar{R} = 00$ in hoofdstuk 9 is uitgesloten.

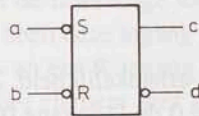
Hoewel een S-R latch en een $\bar{S}\bar{R}$ latch bij de ingangscombinatie $SR = 11$ geen ondubbelzinnig bepaalde inwendige toestand bezitten, zijn de interne en dus ook de externe logische waarden van hun uitgangen meestal wel ondubbelzinnig bepaald. Wanneer we dit in het symbool willen aangeven, dan kan dit door van *set-* en *resetafhankelijkheid* gebruik te maken (S/R-afhankelijkheid). De S- en/of de R-ingang is dan een beïnvloedende ingang die, als deze intern 1 is, de interne logische waarden van de ervan afhankelijke uitgangen 1 respectievelijk 0 maakt, ongeacht de interne toestand en andere S- en R-ingangen. Zie figuur 48.

We demonstreren het gebruik van S/R-afhankelijkheid aan de hand van figuur 48.c. Indien S en R beide intern 1 zijn (extern $ab = 11$), dan worden bij een uit twee NOR's opgebouwde latch beide uitgangen extern 0. Dit beschrijven we met een resetafhankelijkheid van de ingang aangeduid met R2 aan de bovenste uitgang (c-uitgang). Om de d-uitgang extern 0 te maken, moet deze intern 1 zijn. Hierop is het signaal op de setingang dominant, hetgeen aangegeven wordt met een afhankelijkheid van de setingang S1.

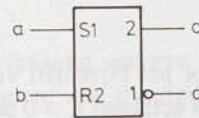


a. S-R Latch

a	b	S	R	c	d
0	0	0	0	onthouden	
0	1	0	1	0 1	
1	0	1	0	1 0	
1	1	1	1	niet gespecificeerd	

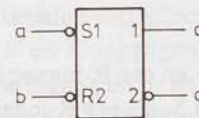
b. \bar{S} -R Latch

a	b	S	R	c	d
1	1	0	0	onthouden	
1	0	0	1	0 1	
0	1	1	0	1 0	
0	0	1	1	niet gespecificeerd	

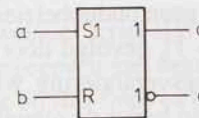


c. S-R Latch, met twee NOR's.

a	b	S	R	c	d
0	0	0	0	onthouden	
0	1	0	1	0 1	
1	0	1	0	1 0	
1	1	1	1	0 0	

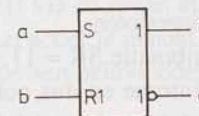
d. \bar{S} -R Latch, met twee NAND's.

a	b	S	R	c	d
1	1	0	0	onthouden	
1	0	0	1	0 1	
0	1	1	0	1 0	
0	0	1	1	1 1	



e. S-R Latch, met overheersende SET.

a	b	S	R	c	d
0	0	0	0	onthouden	
0	1	0	1	0 1	
1	0	1	0	1 0	
1	1	1	1	1 0	



f. S-R Latch, met overheersende RESET.

a	b	S	R	c	d
0	0	0	0	onthouden	
0	1	0	1	0 1	
1	0	1	0	1 0	
1	1	1	1	0 1	

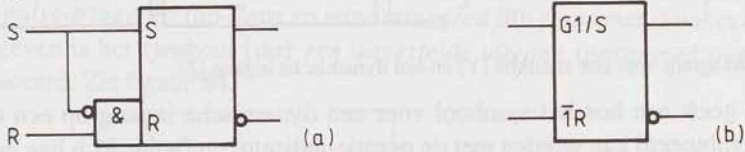
Figuur 48. Symbolen voor latches.

Opmerking

Bij de introductie van de afhankelijkheidsnotatie is gesteld dat een beïnvloedende ingang geen 'normale' ingang is van het element en alleen een effect heeft op de ervan afhankelijke ingangen en uitgangen. De S/R-afhankelijkheid vormt hierop een uitzondering. Een beïnvloedende S-ingang heeft, indien alle R-ingangen intern 0 zijn, wel degelijk een normaal effect op de inwendige toestand en derhalve op alle uitgangen van het element en niet alleen op de ervan afhankelijke uitgangen. (Naar de mening van de auteurs had men beter de definitie van S/R-afhankelijkheid streng kunnen toepassen. Aan de S/R-ingangen kan door plaatsing van twee aanduidingen, S/S1 bijvoorbeeld, het verschil worden aangegeven.)

Toepassing

Latches kunnen zo gerealiseerd worden dat zij een set- respectievelijk een resetingang bezitten die de andere ingang *overheerst*. Als eerste opzet van een schakeling denken we dan aan figuur 49.a, waarin sprake is van een overheersende set. De $SR = 11$ combinatie is dan vergrendeld.



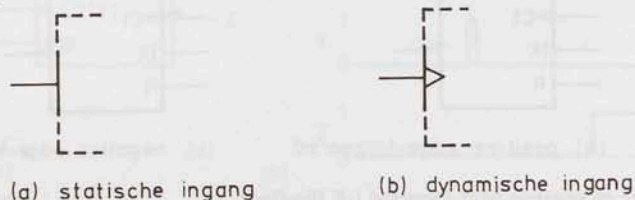
Figuur 49. Latch met overheersende set.

Een en ander kan met G-afhankelijkheid worden weergegeven. Bij dit type latch is de interne toestand altijd bepaald ($SR = 11$ is vergrendeld) en ook zijn de uitgangen altijd elkaars complement. Het probleem van de $SR = 11$ naar de $SR = 00$ overgang (externe signalen) bestaat echter nog steeds!

A.20. Symbolen voor flip-flops

In de voorafgaande beschouwingen is steeds gesproken over 'de interne toestand' van een element. Bij latches en gated latches is duidelijk wat hiermee bedoeld wordt, bij flip-flops niet. Afhankelijk van de beschouwingwijze heeft een flip-flop vier of soms meer interne toestanden (level mode) of twee interne toestanden (clock mode). Intern bestaat een flip-flop uit twee latches, of bestaat de flip-flop uit een constructie die effectief hetzelfde gedrag realiseert. Volgens de clock mode beschouwingwijze is de interne toestand van een flip-flop de toestand van de latch waarmee de uitgangswaarde(n) corresponderen. Deze gedachtengang wordt min of meer ook gevolgd bij het opstellen van de IEC-symbolen.

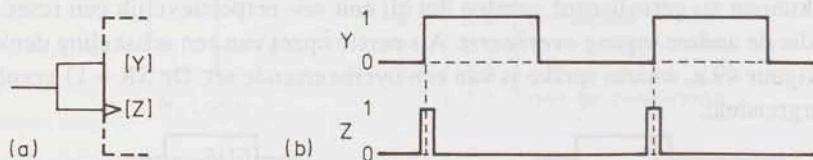
Met C-afhankelijkheid zoals deze eerder is geïntroduceerd kan nog geen onderscheid gemaakt worden ten aanzien van de manier waarop flip-flops reageren op de klokpuls. Om de timing te kunnen karakteriseren als edge-triggered, pulse-triggered of edge-triggered met data lockout zijn twee extra symbolen nodig. Een hiervan is het in figuur 50 getekende symbool van een *dynamische ingang* (dynamic input). In tegenstelling tot de dynamische ingang wordt een normale besturingsingang wel een *statische ingang* (static input) genoemd.



Figuur 50. Symbolen voor een statische en een dynamische ingang.

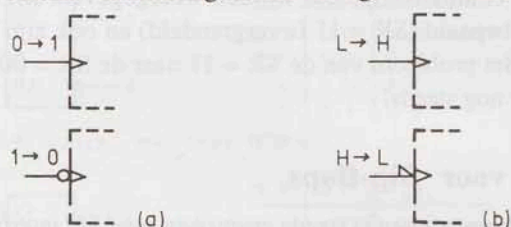
Een dynamische ingang heeft de interne logische waarde 1 gedurende een korte tijd voor en na de $0 \Rightarrow 1$ overgang die de interne logische waarde vertoond zou hebben, indien deze ingang statische geweest zou zijn. Op alle overige momenten is de interne logische waarde 0.

In het tijddiagram van figuur 51 is dit nader toegelicht.



Figuur 51. Tijddiagram voor een statische [Y] en een dynamische ingang [Z].

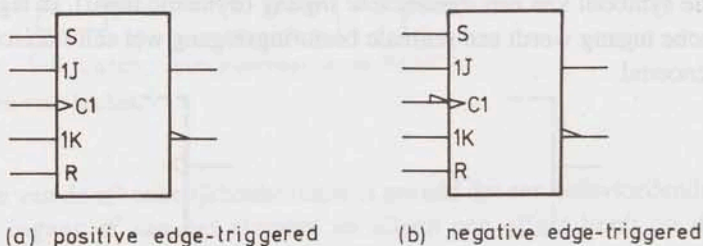
Figuur 52.a geeft aan hoe het symbool voor een dynamische ingang op een ontwerp-schema gecombineerd kan worden met de negatie-indicator en figuur 52.b hoe men dit op een uitvoeringsschema met de polariteitsindicator kan combineren. Bij de ingangslijnen is vermeld met welke externe verandering de interne logische waarde 1 correspondeert.



Figuur 52. Dynamische ingang met negatie-indicator en met polariteitsindicator.

Op grond van het voorgaande is het duidelijk dat het symbool voor een dynamische ingang gebruikt kan worden om een flip-flop met edge-triggered timing te symboliseren. Bij een edge-triggered flip-flop liggen de setup time t_{su} en de hold time t_h voor een ingangssignaal rond één flank. Gedurende het $t_{su}-t_h$ interval bepaalt het ingangssignaal welke de nieuwe uitgangswaarde gaat worden, m.a.w. tevens welke de interne toestand volgens de clock mode beschouwing wordt.

Een en ander laat onverlet wat voor invloed een instelsignaal op de ingangslatch heeft. Meestal is het zo dat bij een edge-triggered flip-flop de ingangslatch het ingangssignaal continu volgt gedurende de klokfase welke voorafgaat aan de flank waarop/waarna de uitgang reageert. Een dynamische ingang impliceert dus niet automatisch dat van een kloksignaal een kort pulsje wordt afgeleid, zoals vaak in de literatuur wordt gesuggereerd.



Figuur 53. Positive en negative edge-triggered J-K flip-flops.

Figuur 53 toont hoe op een uitvoeringsschema een positive en een negative edge-triggered J-K flip-flop kunnen worden gerepresenteerd. Hierbij wordt verondersteld dat de tijd gedurende welke de dynamische ingang intern 1 is overeenkomt met de setup en de hold time van de flip-flop. Omdat de J- en K-ingangen afhankelijk zijn van de beïnvloedende

C-ingang hebben zij een *voorbereidend karakter*. Zij mogen niet veranderen gedurende de setup en de hold time, d.w.z. zolang de C-ingang intern 1 is. De S- en R-ingangen zijn niet afhankelijk van de C-ingang. Zij zijn daarom direct werkend.

De uitgestelde uitgang

Om ook *pulse-triggered flip-flops* en *edge-triggered flip-flops met data lockout* weer te kunnen geven is het *symbol voor een uitgestelde uitgang (postponed output symbol)* geïntroduceerd. Zie figuur 54.

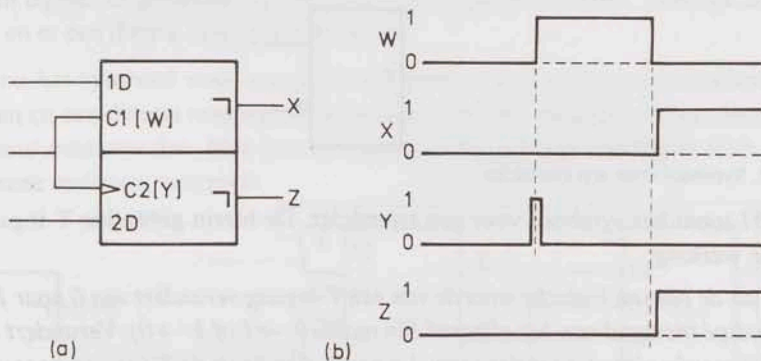


Figuur 54. Uitgestelde uitgang.

Bij een uitgestelde uitgang wordt een verandering van de interne logische waarde die het gevolg is van het intern 1 worden van een statische C-ingang uitgesteld tot het moment waarop de interne logische waarde van deze C-ingang weer 0 wordt. Is de verandering het gevolg van het intern 1 worden van een dynamische C-ingang, dan wordt deze uitgesteld tot het moment waarop de interne logische waarde van deze C-ingang weer 0 geworden zou zijn, indien deze ingang statische geweest zou zijn. In beide gevallen mogen de van de C-ingang afhankelijke ingangen niet veranderen zolang de C-ingang intern 1 is.

In het tijddiagram van figuur 55 is dit nader toegelicht. Hierbij is aangenomen dat de D-ingangen beide 1 zijn en dat de uitgangen aanvankelijk 0 zijn.

Het bovenste symbool van figuur 55.a representeert een *pulse-triggered D flip-flop* en het onderste een *edge-triggered D flip-flop met data lockout*. Het verschil in uitwendig gedrag bestaat hierin, dat bij het bovenste element de D-ingang stabiel moet zijn gedurende de tijd dat C1 intern 1 is ($t_{su} + t_{puls} + t_h$) en bij het onderste element zolang C2 intern 1 is (alleen $t_{su} + t_h$).

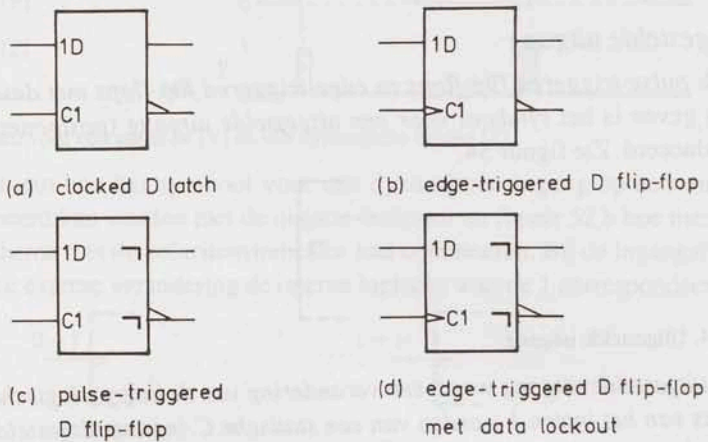


Figuur 55. Tijddiagram.

Opmerking

Hoewel de definitie van de uitgestelde uitgang geformuleerd is met betrekking tot een C-ingang geldt deze, zoals later zal blijken, ook indien men hierin 'C-ingang' door 'T-ingang', 'schuifingang' of 'telingang' vervangt.

Figuur 56 geeft een overzicht van de vier mogelijkheden die er voor geheugenelementen met een klokingang met betrekking tot hun timing bestaan. Als voorbeeld is als afhankelijke ingang een D-ingang gekozen.



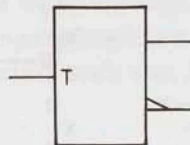
Figuur 56. Vier typen geklokte D-elementen.

Opmerking

Aangezien, zoals reeds is gesteld, de inwendige logische waarden van de uitgangen gelijk zijn aan de inwendige toestand van een element, wordt de verandering van de inwendige toestand van het element volgens figuur 56.c en dat volgens figuur 56.d eveneens uitgesteld.

Opmerking

Zoals eveneens reeds eerder is gesteld specificeren de symbolen voor flip-flops (figuur 56.b, c en d) de werking van de door hen gerepresenteerde elementen *niet*, indien de van de C-ingang afhankelijke ingangen veranderen wanneer deze C-ingang intern 1 is. Bij het symbool voor de latch (figuur 56.a) is dit wel het geval.



Figuur 57. Symbool voor een tweedeler.

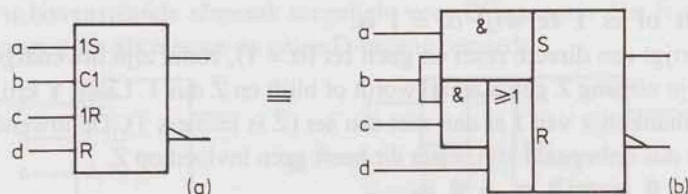
Figuur 57 toont het symbool voor een tweedeler. De hierin gebruikte T-ingang heeft de volgende werking:

Telkens als de interne logische waarde van een T-ingang verandert van 0 naar 1, verandert de inwendige toestand van het element één maal ($0 \rightarrow 1$ of $1 \rightarrow 0$). Verandert zijn interne logische waarde niet of gaat deze van 1 naar 0, dan heeft de T-ingang geen effect op de werking van het element, tenzij dit een of meer uitgestelde uitgangen heeft.

Deze definitie is zodanig, dat het niet nodig is om in figuur 57 het symbool voor een dynamische ingang toe te passen.

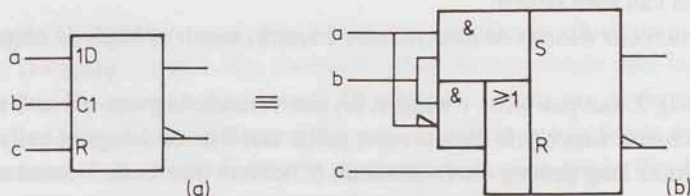
A.21. Symbolen voor flip-flops met direct werkende ingangen

Veel flip-flops zijn uitgerust met direct werkende S- en R-ingangen, vaak asynchrone preset en clear genoemd. Om de werking en de timing hiervan toe te lichten wordt eerst de gated latch met een extra directe reset bekeken. Hoewel deze schakeling als zodanig niet vaak afzonderlijk wordt toegepast, is inzicht in het gedrag ervan toch nodig, omdat hij veelvuldig als onderdeel van flip-flops wordt gebruikt. Figuur 58 geeft een voorbeeld.



Figuur 58. Gated S-R latch met directe resetingang.

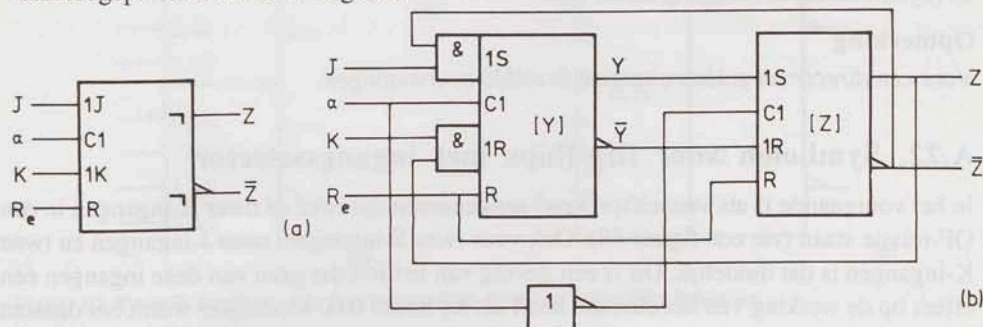
Indien ingang b de waarde 1 heeft, heeft de schakeling twee R-ingangen, die ieder voor zich de inwendige toestand 0 kunnen maken. Zij staan derhalve in een OF-relatie, zoals figuur 58.b aangeeft. De directe R-ingang biedt de mogelijkheid om de schakeling buiten de C-ingang om (d.w.z. ook als deze 0 is) een reset te geven en dus op 0 te zetten. Zijn a en b beide 1, dan leidt het geven van een directe reset tot een *niet ondubbelzinnig bepaalde inwendige toestand* van de latch. Immers, dan zijn de interne S en R beide 1.



Figuur 59. Gated D latch met directe resetingang.

Ook bij de in figuur 59 getoonde D latch doet zich dit probleem voor, namelijk als a en b beide 1 zijn en er een directe reset gegeven wordt.

Beschouw nu het symbool voor een pulse-triggered J-K flip-flop met voorbereidende J/K-ingangen en een directe resetingang R in figuur 60. Het probleem is hoe dit symbool geïnterpreteerd moet worden. Men kan dit nagaan in het schema van figuur 60.b, dat een vaak toegepaste realisatie weergeeft.



Figuur 60. Pulse triggered J-K flip-flop met directe R-ingang.

Om de werking van de directe resetingang R_e op de schakeling te kunnen analyseren moeten vier gevallen onderscheiden worden.

1. R_e wordt of is 1 terwijl $\alpha = 0$ is.

Latch Y krijgt een directe reset en geen set ($\alpha = 0$), zodat de uitgang Y de waarde 0 krijgt of behoudt. Hierdoor krijgt latch Z geen set, of niet langer een set, maar wel een (dubbele) reset, zodat de inwendige toestand en dus ook de uitgang Z gelijk aan 0 wordt of blijft.

2. R_e wordt of is 1 terwijl $\alpha = 1$ is.

Latch Z krijgt een directe reset en geen set ($\alpha = 1$), zodat zijn inwendige toestand en dus ook zijn uitgang Z gelijk aan 0 wordt of blijft en \bar{Z} dus 1. Latch Y krijgt een directe reset en afhankelijk van J al dan niet een set (\bar{Z} is immers 1). De inwendige toestand van Y kan dus onbepaald zijn, maar dit heeft geen invloed op Z.

3. R_e wordt 0 terwijl $\alpha = 0$ is.

Uit geval 1 volgt dat, zolang R_e nog 1 is, beide latches in de 0-stand verkeren. Wordt R_e nu 0, dan blijft $Y = 0$ en omdat Z nu weer de stand van Y overneemt, blijft Z ook 0.

4. R_e wordt 0 terwijl $\alpha = 1$ is.

Uit geval 2 volgt dat, zolang R_e nog 1 is, latch Z in de 0-stand verkeert, maar dat de inwendige toestand van Y gelijk aan 0 (voor $J = 0$) of onbepaald (voor $J = 1$) is. K heeft geen invloed omdat $Z = 0$. Wordt R_e nu 0, dan verdwijnt de reset van Y, zodat Y gelijk aan 0 blijft bij $J = 0$ of gelijk aan 1 wordt bij $J = 1$.

Resumerend kan men stellen:

- Op het moment waarop de directe reset 1 wordt, wordt of blijft de uitgang Z gelijk aan 0.
- De uitgang Z kan pas weer 1 worden bij een verandering van α van 1 naar 0, die volgt op een situatie waarbij de directe reset gelijk aan 0 en de J-ingang gelijk aan 1 is. Deze situatie moet lang genoeg duren om latch Y betrouwbaar in de 1-stand te brengen.

Een *directe reset* van een pulse-triggered flip-flop is dus duidelijk *dominant*. Zodra hij wegvalt werkt de schakeling weer *normaal*, tenzij dit te kort voor of tijdens een interne $1 \Rightarrow 0$ overgang van de klokingang plaatsvindt, in welk geval het niet zeker is wat er gebeurt. *Aan het tijdstip van weghalen van een directe reset moeten derhalve eisen gesteld worden!*

Deze uitspraak geldt ook voor een edge-triggered flip-flop, al dan niet met data lockout, met dien verstande dat het niet zeker is wat er gebeurt indien de directe reset wegvalt als de (dynamische) klokingang intern 1 is.

Opmerking

Voor een *directe set* gelden uiteraard dezelfde overwegingen.

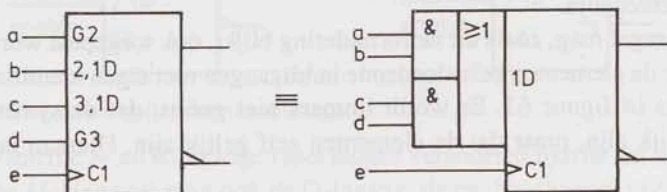
A.22. Symbolen voor flip-flops met ingangselector

In het voorgaande is als vanzelfsprekend aangenomen dat twee of meer R-ingangen in een OF-relatie staan (zie o.a. figuur 58). Ook voor twee S-ingangen, twee J-ingangen en twee K-ingangen is dat duidelijk. Dit is een gevolg van het feit dat geen van deze ingangen een effect op de werking van het element heeft als hij intern 0 is. Moeilijker wordt het daarom voor twee of meer D-ingangen. Immers, hebben twee D-ingangen verschillende interne

logische waarden, dan wil de één het element in de 1-stand en de ander in de 0-stand brengen. Om de symbolen zo eenvoudig mogelijk te houden is in het IEC-systeem nu de volgende afspraak gemaakt:

Twee of meer ingangen, die met dezelfde letter zijn aangeduid, staat in een OF-relatie. Zijn het echter beïnvloedende ingangen in de zin van de afhankelijkheidsnotatie, dan is dit alleen het geval als zij ook hetzelfde identificatienummer bezitten.

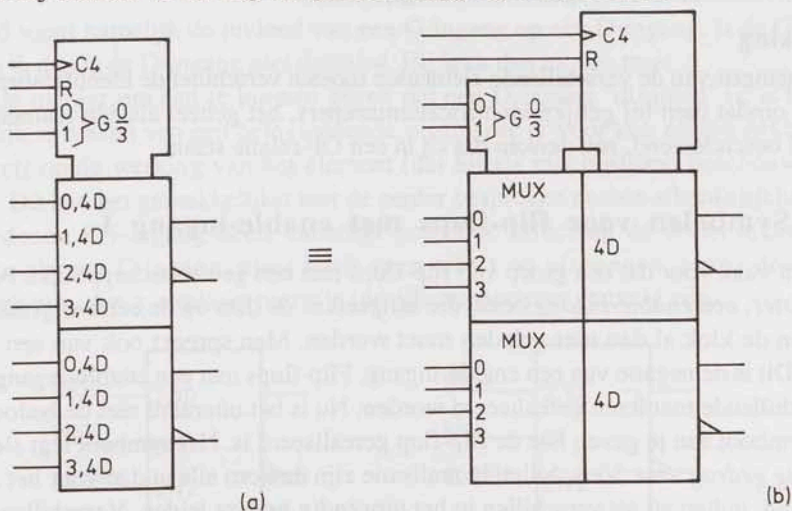
In figuur 61 is bovenstaande afspraak toegelicht voor D-ingangen. Het is de OF-functie van de D-ingangen die als nieuwe en enige D-ingang optreedt.



Figuur 61. Element met o.a. twee D-ingangen.

Men kan hiervan gebruik maken bij het vormen van een symbool voor een groep van D flip-flops, die elk voorzien zijn van een *data selector*, terwijl de instelling van de selectoren en de besturing van de flip-flops gemeenschappelijk is. Figuur 62 geeft een voorbeeld voor een groep van twee D flip-flops, elk met een 1-uit-4 selector voor de D-ingang.

Omdat van de vier G-ingangen er precies één intern 1 is, kan in elk element van figuur 62.a slechts één D-ingang intern 1 zijn, namelijk indien de betreffende data-ingang intern 1 is. De overige drie D-ingangen zijn intern 0. De logische waarde van de (impliciete) OF-functie van de vier D-ingangen is daarom gelijk aan die van de geselecteerde data-ingang. Omdat hun gemeenschappelijke C-ingang dynamisch is, zijn beide flip-flops edge-triggered. Zij bezitten voorts nog een gemeenschappelijke directe reset.



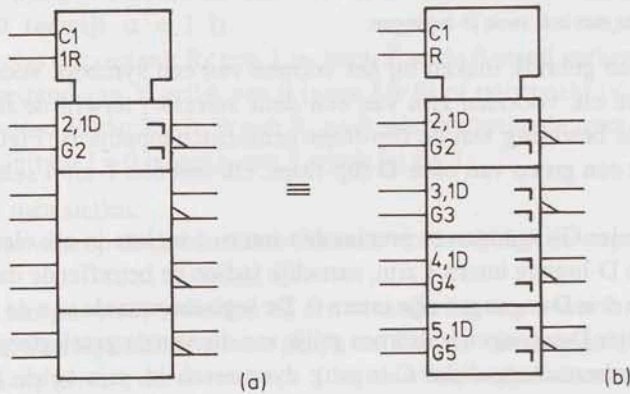
Figuur 62. Twee D flip-flops met data selectors en gemeenschappelijke besturing.

Figuur 62.b toont een ander symbool voor dezelfde schakeling waarin de data selectors expliciet zijn aangegeven. Op beide symbolen kan nog de volgende regel worden toegepast:

Bij een groep van gelijke elementen, waarvan de symbolen aan elkaar getekend zijn, behoeven het functiesymbool en de toevoegsymbolen bij de in- en uitgangen slechts in het eerste symbool te worden aangegeven.

Past men deze regel toe op figuur 62, dan zijn de onderste rechthoeken dus leeg. In dat geval verdient de tekenwijze volgens figuur 62.a duidelijk de voorkeur, zeker als er meer dan twee elementen zijn.

Bovenstaande regel mag, zoals uit de formulering blijkt, ook toegepast worden indien de symbolen voor de elementen beïnvloedende in/uitgangen met eigen identificatienummers bevatten, zoals in figuur 63. Er wordt immers niet geëist, dat de symbolen voor de elementen gelijk zijn, maar dat de elementen zelf gelijk zijn. Hieraan is in figuur 63 voldaan.



Figuur 63. Voorbeeld van het weglaten van toevoegsymbolen.

Opmerking

De G-ingangen van de verschillende elementen moeten verschillende identificatienummers hebben, omdat men bij gelijke identificatienummers, het geheel als één (samengesteld) symbool beschouwend, zou denken dat zij in een OF-relatie staan.

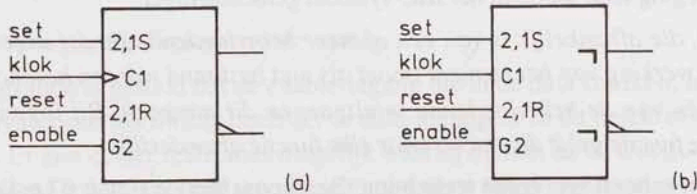
A.23. Symbolen voor flip-flops met enable-ingang I

Het komt vaak voor dat een groep van flip-flops met een gemeenschappelijke besturing, een *register*, een *enable-ingang* bezit, die aangeeft of de data op de eerstvolgende actieve flank van de klok al dan niet geladen moet worden. Men spreekt ook van een *inhibit-ingang*. Dit is de negatie van een enable-ingang. Flip-flops met een enable-ingang kunnen op verschillende manieren gerealiseerd worden. Nu is het uiteraard niet de bedoeling om in het symbool aan te geven hoe de flip-flop gerealiseerd is. Het symbool legt slechts het *uitwendig gedrag* vast. Verschillen in realisatie zijn daarom alleen dan voor het symbool van belang, indien zij tot verschillen in het *uitwendig gedrag* leiden. Verschillen t.a.v. de *timing van een enable-ingang*, d.w.z. de momenten waarop deze mag veranderen, spelen hierbij een grote rol.

Schakelen in de data

De beste manier om het effect van een enable-ingang te realiseren is om hem de data-ingangen al dan niet te laten activeren, zoals in figuur 64 is aangegeven.

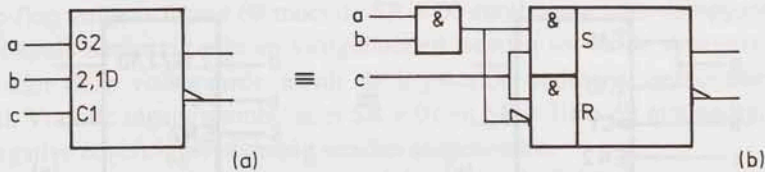
Omdat dit vaak d.m.v. AND-poorten gebeurt, is in de figuren 64.a en b gebruik gemaakt van de G-afhankelijkheid.



Figuur 64. Symbolen voor flip-flops met enable-ingang.

Aangezien de interne S- en R-ingangen niet mogen veranderen tijdens het intern 1 zijn van de *dynamische klokingang*, mag ook de G-ingang, de enable-ingang, gedurende die tijd niet veranderen. In figuur 64.a heeft deze derhalve evenals de set- en de resetingang een edge-triggered timing en in figuur 64.b een pulse-triggered timing.

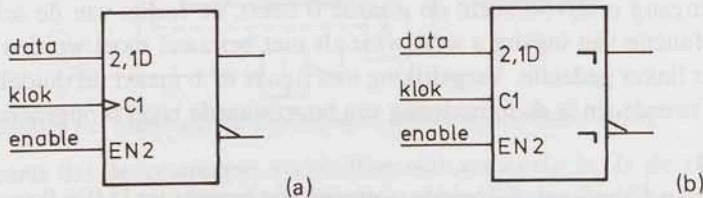
De mogelijkheid om G-afhankelijkheid toe te passen berust op het feit dat S- en R-ingangen geen effect op de werking van het element hebben als zij intern 0 zijn. Zij zijn dan *disabled*. Dit geldt uiteraard ook voor J- en K-ingangen, maar niet voor D-ingangen.



Figuur 65. Invloed van een G-ingang op een D-ingang.

Figuur 65 toont namelijk de invloed van een G-ingang op een D-ingang. Is de G-ingang (a) intern 0, dan is de D-ingang *niet disabled*. Hij leidt dan tot een reset.

Een goede manier om aan te kunnen geven dat een D-ingang 'disabled' is, is om hem afhankelijk te maken van een beïnvloedende ingang die er voor kan zorgen dat hij *geen effect* heeft op de werking van het element (dat hij als niet bestaand beschouwd moet worden). Dit kan het gemakkelijkst met de eerder besproken enable-afhankelijkheid. Een beïnvloedende EN-ingang heeft namelijk hetzelfde effect op de ervan afhankelijke *ingangen* als een C-ingang, maar heeft geen effect op *uitgangen*, tenzij deze ervan afhankelijk zijn, d.w.z. expliciet met zijn identificatienummer gemerkt zijn.



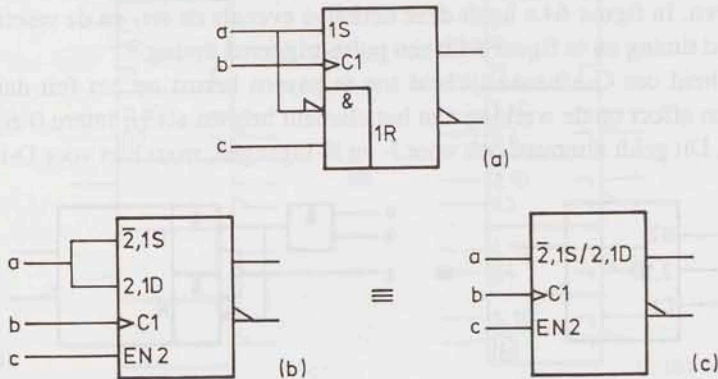
Figuur 66. Symbolen voor flip-flops met enable-ingang.

Is de EN-ingang in figuur 66.a intern 0, dan heeft de ervan afhankelijke D-ingang geen effect op de flip-flop, ook al zou de C-ingang intern 1 zijn. Immers, is de C-ingang intern 1, dan heeft de ervan afhankelijke D-ingang weliswaar zijn 'normale' effect, maar als dit door de EN-ingang gemodificeerd is tot 'geen effect' is het resultaat: *geen effect*. De D-ingang kan dus alleen iets doen wanneer de EN- en de C-ingang beide 1 zijn.

Deze overweging leidt tot de in het IEC-systeem geldende regel:

Een ingang, die afhankelijk is van een of meer beïnvloedende in/uitgangen, heeft geen effect op de werking van het element (moet als niet bestaand worden beschouwd), indien tenminste één van de beïnvloedende in/uitgangen dit aangeeft. Bij ingangen met een meervoudige functie geldt dit slechts voor elke functie afzonderlijk.

De laatste zin heeft nog enige toelichting. Beschouw hiertoe figuur 67.a. Deze toont een symbool dat de erdoor gerepresenteerde schakeling nauwkeurig weergeeft. Niettemin is het voor de ontwerper, die deze schakeling wil toepassen, niet eenvoudig om het uitwendig gedrag ervan snel te doorzien.



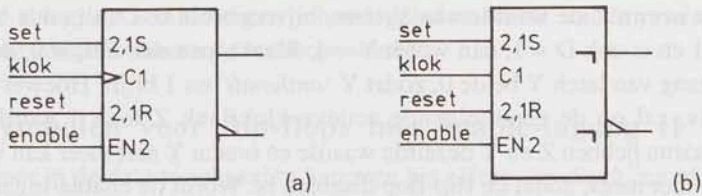
Figuur 67. Flip-flop met een ingang met een meervoudige functie.

Een andere wijze van representeren van de schakeling kan hierbij behulpzaam zijn. Het symbool van figuur 67.a leert dat, als $c = 0$ is, er geen reset gegeven kan worden. Ingang a is dan een S-ingang. Heeft c daarentegen de waarde 1, dan veroorzaakt ingang a, als hij 1 is een set en als hij 0 is een reset, d.w.z. a fungeert dan als D-ingang. Dit is weergegeven in figuur 67.b. Hoewel ogenschijnlijk complexer dan figuur 67.a, geeft figuur 67.b duidelijk aan dat ingang a twee functies heeft en dat ingang c bepaalt welke van deze functies wordt uitgeoefend. De gebruiker van de schakeling zal daarom het symbool van figuur 67.b prefereren boven dat van figuur 67.a.

Figuur 67.c toont een alternatief voor het symbool van figuur 67.b. Het laat tevens zien dat, indien ingang c bijvoorbeeld de waarde 0 heeft, de rechts van de schuine streep aangeduide functie van ingang a weliswaar als niet bestaand moet worden beschouwd, maar niet het linker gedeelte. Vergelijking met figuur 67.b maakt dit duidelijk. Dit is de reden dat de tweede zin in de formulering van bovenstaande *regel* is opgenomen.

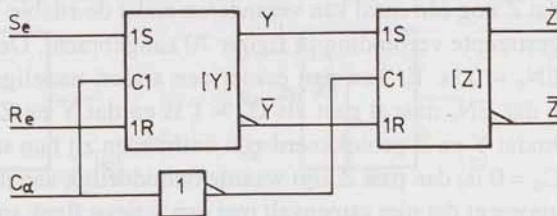
Opmerking

Het gebruik van *EN-afhankelijkheid* is uiteraard niet beperkt tot D flip-flops. Ook andere typen flip-flops met enable-ingang kunnen ermee worden weergegeven, zoals de figuren 68.a en b laten zien.



Figuur 68. Symbolen voor flip-flops met enable-ingang.

In het voorafgaande is gesteld dat de enable-ingang die in de data schakelt, het tot nu toe besproken geval, dezelfde timing heeft als de data-ingangen. In de beschreven symbolen is dat ook zo. Er zijn echter realisaties mogelijk waarbij dit niet zo is. Om dit probleem in zijn juiste context te kunnen plaatsen refereren we eerst aan de pulse-triggered S-R flip-flop in figuur 69.

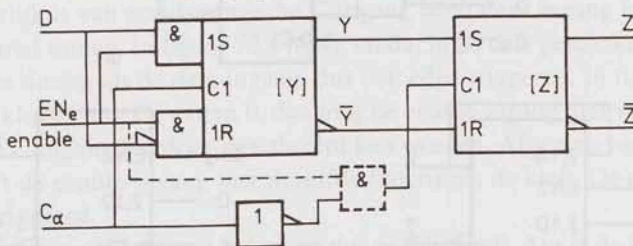


Figuur 69. Pulse-triggered S-R flip-flop.

Bij de flip-flop volgens figuur 69 moet de $SR = 00$ combinatie vóór de opgaande flank van de klokpuls aanwezig zijn en vastgehouden worden tot na de neergaande flank. Slechts onder deze voorwaarde wordt de ingestelde opdracht, onthouden, correct uitgevoerd. Voor de ingangscombinaties $SR = 01$ en $SR = 10$ is dit niet nodig, hiervoor kan een negatieve edge-triggered timing worden aangehouden.

In het algemeen wordt het sterk ontraden om bij een pulse-triggered timing instelsignalen te veranderen gedurende de klokpuls. Men kan het de ontwerper echter niet verbieden hiervan gebruik te maken. De symbolen moeten derhalve dit soort gebruik voorzien.

Tussen enable-ingangen en data-ingangen bestaan vaak verschillen in de timing ervan. Meestal definieert men de timing dan 'worst case', hoewel het ook hier voorkomt dat men slechts let op de timing van elk signaal afzonderlijk. Als voorbeeld onderzoeken we de implementatie van de enable-ingang op een D flip-flop volgens figuur 70.



Figuur 70. D flip-flop met edge-triggered data-ingang en pulse-triggered enable-ingang.

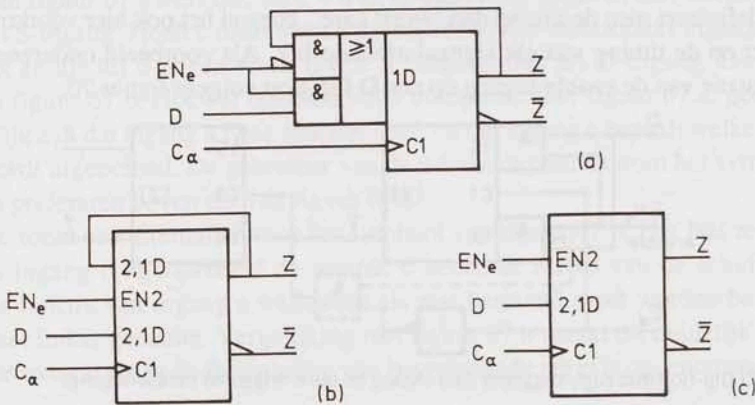
Onderstel eerst dat de gestreepte verbinding niet aanwezig is. Is de enable-ingang $EN_e = 1$, dan heeft de schakeling de normale werking van een negatieve edge-triggered D flip-flop. Zie paragraaf 11.1 en 11.2. De D-ingang heeft dus een edge-triggered timing.

Is $C_\alpha = 0$, dan neemt Z de waarde van Y over, bijvoorbeeld 0. Dan is dus $Y = Z = 0$. Wordt C_α nu 1 en is ook $D = 1$, dan wordt $Y = 1$. Maakt men dan $EN_e = 0$, dan worden de S- en R-ingang van latch Y beide 0, zodat Y 'onthoudt' en 1 blijft. Hoewel de enable-ingang nu 0 is, zal op de eerstvolgende actieve klokflank Z toch 1 worden, d.w.z. veranderen. Daarna hebben Z en Y dezelfde waarde en omdat Y niet meer kan veranderen doet Z dit ook niet meer, zodat de flip-flop disabled is. Wordt de enable-ingang weer 1, dan gaat de schakeling weer normaal werken. De 'fout' in deze schakeling is dat Z nog één maal kan veranderen nadat de enable-ingang 0 geworden is. Dit gebeurt alleen als Y en Z verschillende waarden hebben, d.w.z. als $C_\alpha = 0$ is. De enable-ingang heeft daarom een *pulse-triggered timing*. Om dit correct weer te geven moet het symbool dan ook getekend worden als een pulse-triggered D flip-flop met enable-ingang, ook al heeft de data-ingang een edge-triggered timing.

Om te voorkomen dat Z nog één maal kan veranderen nadat de enable-ingang 0 geworden is, wordt soms de gestreepte verbinding in figuur 70 aangebracht. Deze blokkeert de C-ingang van Z als $EN_e = 0$ is. Er kan dan echter een ander, nadeliger effect optreden. Onderstel namelijk dat EN_e naar 0 gaat als $C_\alpha = 1$ is en dat Y en Z dan verschillende waarden hebben. Omdat Y en Z geblokkeerd zijn onthouden zij hun standen. Wordt EN_e dan weer 1 terwijl $C_\alpha = 0$ is, dan past Z zijn waarde onmiddellijk aan die van Y aan, zodat Z verandert op een moment dat niet samenvalt met een actieve flank van C_α . Dit is uit een oogpunt van *systemontwerp ontoelaatbaar*.

Om een D flip-flop te realiseren die zowel een edge-triggered data-ingang als een edge-triggered enable-ingang heeft moet men er voor zorgen dat, als de enable-ingang 0 wordt, Y en Z steeds dezelfde waarde hebben. Is $C_\alpha = 0$, dan is hier zonder meer aan voldaan. Is $C_\alpha = 1$, dan moet Y direct aangepast worden aan Z .

Dit kan op de in figuur 71.a geschetste wijze. Figuur 71.b toont hoe dit aangegeven kan worden door van EN-afhankelijkheid gebruik te maken. Zo men wenst kan de terugkoppeling dan nog vermeden worden door Z-afhankelijkheid toe te passen. Aan de gebruiker die niet in de realisatie maar uitsluitend in het uitwendig gedrag geïnteresseerd is geeft het symbool van figuur 71.c echter alle informatie die nodig is.



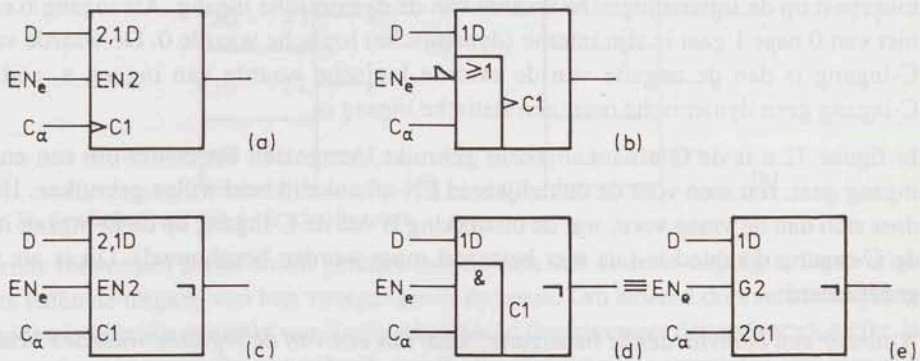
Figuur 71. Realisatie van een edge-triggered D flip-flop met enable-ingang.

Uit figuur 71.b blijkt duidelijk dat de disable-functie gerealiseerd is door de flip-flop met zijn eigen inhoud te laden. De schakeling is een D flip-flop met een selector, die bepaalt of

de externe data dan wel de eigen inhoud geladen wordt. In de praktijk heeft deze oplossing de voorkeur boven de eerdergenoemde.

A.24. Symbolen voor flip-flops met enable-ingang II

Behalve door in de data te schakelen kan men het effect van *disabling* of *inhibiting* ook realiseren door de actieve flank van het kloksignaal te onderdrukken. Dit gebeurt dan door het kloksignaal via een poort aan het element toe te voeren en op deze poort tevens het enable-sigitaal aan te sluiten. In de linker kolom van figuur 72 zijn de symbolen gegeven voor een edge- en een pulse-triggered D flip-flop, waarbij in de data geschakeld is. De tweede kolom toont hun *logische equivalenten* waarbij in de klok geschakeld is. Hierbij wordt met logisch equivalent bedoeld dat de externe logische waarden van de ingangen van beide elementen overeenkomen. Hun verschil ligt in de *timing van de enable-ingang*.



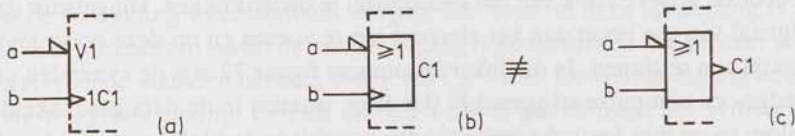
Figuur 72. In de data en in de klok geschakelde D flip-flops.

Als in figuur 72.a de enable-ingang extern en dus ook intern 0 is, is de D-ingang en dus ook de flip-flop disabled. Als in figuur 72.b de enable-ingang extern 0 is, is deze intern 1. De uitgang van de OR-poort is dan 1, onafhankelijk van de klokingang. De klokingang kan de *dynamische* C-ingang daarom niet intern 1 maken, omdat hiervoor een overgang van de uitgang van de OR-poort van 0 naar 1 nodig is. De dynamische C-ingang en dus ook de flip-flop zijn zodoende disabled. Is de enable-ingang extern 1, dan kunnen in beide figuren de klok- en de data-ingang hun werk doen. Omdat in beide gevallen de data-ingang afhankelijk is van een dynamische C-ingang heeft deze ingang in beide gevallen een edge-triggered timing. In figuur 72.a heeft, omdat in de data geschakeld is, de enable-ingang dezelfde timing als de data-ingang, dus ook edge-triggered. In figuur 72.b echter niet. Is hier de klokingang C_α extern 0, dan mag de enable-ingang niet veranderen omdat de dynamische C-ingang hierdoor geactiveerd kan worden. Afgezien van de negatie aan de ingang heeft de enable-ingang hier dezelfde functie als de klok. De enable-ingang is dus niet edge-triggered.

Is in figuur 72.c de enable-ingang extern en dus ook intern 0, dan is de D-ingang en dus ook de flip-flop disabled. Als in figuur 72.d de enable-ingang extern 0 is, is de uitgang van de AND-poort 0, onafhankelijk van de klokingang. De C-ingang en dus ook de flip-flop zijn zodoende disabled. Is de enable-ingang extern 1, dan kunnen in beide figuren de klok- en de data-ingang hun werk doen. Voorts heeft in beide figuren de enable-ingang, evenals de data-ingang een pulse-triggered timing. Hij mag niet veranderen als de

klokingang 1 is. In figuur 72.c niet omdat de D-ingang dan stabiel moet zijn en in figuur 72.d niet om te voorkomen dat de C-ingang van de flip-flop voortijdig 0 wordt, waardoor de uitgang op een ongewenst tijdstip zou kunnen veranderen. Evenals in figuur 72.b heeft de enable-ingang hier weer dezelfde functie als de klok.

In de derde kolom van figuur 72 is de poort voor de C-ingang door middel van de afhankelijkheidsnotatie weergegeven (figuur 72.e). In de bovenste regel kan dit niet.



Figuur 73. Toepassing van de afhankelijkheidsnotatie op een dynamische ingang.

Zoals figuur 73.b laat zien moet bij de notatie volgens figuur 73.a de OF-relatie worden toegepast op de interne logische waarde van de dynamische ingang. Als ingang b extern niet van 0 naar 1 gaat is zijn interne (dynamische) logische waarde 0. De waarde van de C-ingang is dan de negatie van de externe logische waarde van ingang a, zodat de C-ingang geen dynamische maar een statische ingang is.

In figuur 72.e is de G-afhankelijkheid gebruikt. Aangezien het echter om een enable-ingang gaat, zou men voor de duidelijkheid EN-afhankelijkheid willen gebruiken. Hierbij doet zich dan de vraag voor, wat de uitwerking is van de C-ingang op de D-ingang indien de C-ingang disabled is (als niet bestaand moet worden beschouwd). Dit is als volgt gedefinieerd:

Wanneer een beïnvloedende in/uitgang, waarvan een van de logische waarden betekent dat de ervan afhankelijke ingangen disabled zijn (als niet bestaand moeten worden beschouwd), zelf disabled is, zijn ook de ervan afhankelijke ingangen disabled.

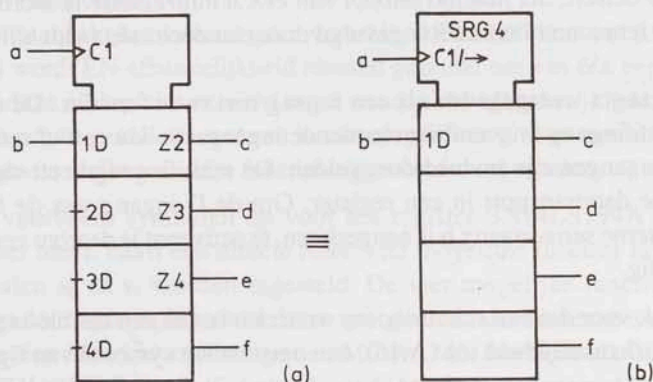
Hoewel EN-afhankelijkheid gebruikt zou kunnen worden om de C-ingang te beïnvloeden is dit sterk te ontraden. Dit zou namelijk het feit verdoezelen dat de enable-ingang onder bepaalde omstandigheden zelf een (ongewenste) actieve klokflank kan veroorzaken, zoals in figuur 72.d ook het geval is. Om de gebruiker hierop te attenderen verdient toepassing van G- of V-afhankelijkheid een sterke voorkeur. Zij duiden EN- en OF-relaties aan, zodat de gebruiker kan zien hoe de schakeling gerealiseerd is en kan nagaan welke gevaren hij loopt.

In figuur 72 is bij de 'klokgeschakelde' edge-triggered flip-flop een OR-poort en bij de pulse-triggered flip-flop een AND-poort gebruikt. Dit kan natuurlijk ook anders. In grotere systemen worden de enable-ingangen echter meestal 'gestuurd' door uitgangen van flip-flops die op dezelfde actieve klokflank reageren als de flip-flop waarop de enable-ingang is aangesloten. Dit betekent dat deze ingang moet kunnen veranderen als de klok de externe logische waarde heeft die volgt op de actieve flank. In figuur 72.b is dit de waarde 1 en in figuur 72.d de waarde 0. Opdat een verandering van de enable-ingang geen verandering van de C-ingang teweegbrengt, is daarom in figuur 72.b een OR-poort en in figuur 72.d een AND-poort gebruikt.

Omdat bij schakelen in de klok een storing op een enable-ingang een niet gewenste flank van de inwendige klok (de C-ingang) kan veroorzaken en bij schakelen in de data niet, verdient het aanbeveling om bij de componentenkeuze bij voorkeur voor 'data-geschakelde' circuits te opteren.

A.25. Symbolen voor schuifregisters

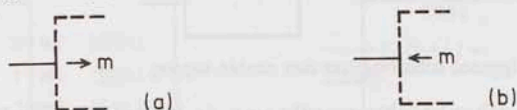
Zoals in paragraaf 12.1 is gesteld is een register een groep van flip-flops die een gemeenschappelijke instelling (besturing) bezitten. Het vormen van een symbool voor schuifregisters met verschillende instellingen, zoals bijvoorbeeld het type SN74194A, is met de tot nu toe geïntroduceerde middelen lastig. Stellen we bijvoorbeeld een symbool samen voor een 4-bit schuifregister met één data-ingang en vier parallelle uitgangen. In eerste aanleg zou men kunnen denken aan een representatie volgens figuur 74.a.



Figuur 74. Symbolen voor een 4-bit schuifregister.

De eerste (bovenste) sectie wordt geladen met de data van de serie-ingang b. De overige secties laden de uitgang van hun voorganger in de keten. Om externe doorverbindingen te vermijden is gebruik gemaakt van *Z-afhankelijkheid* (interconnect dependency), welke in A.13 is geïntroduceerd. Om het gebruik van *Z-afhankelijkheid* te vermijden kan men ook het symbool voor een *schuifingang* (shifting input) toepassen (figuur 75). De werking van een schuifingang wordt als volgt omschreven:

Telkens als de interne logische waarde van een met $\rightarrow m$ aangeduide ingang van 0 naar 1 verandert schuift de inhoud van het register éénmaal m posities van links naar rechts of van boven naar beneden. Verandert zijn interne logische waarde niet, of gaat deze van 1 naar 0, dan heeft deze ingang geen effect op de inhoud van het register (tenzij dit een of meer uitgestelde uitgangen heeft).



Figuur 75. Schuifingang.

De omschrijving van de werking van een ingang met de aanduiding volgens figuur 75.b wijkt hiervan slechts in die zin af dat de uitdrukking 'van links naar rechts of van boven naar beneden' vervangen moet worden door 'van rechts naar links of van beneden naar boven'.

De letter m moet altijd vervangen worden door het werkelijke aantal stappen, genoteerd in het tientalig stelsel. Is $m = 1$, hetgeen meestal het geval zal zijn, dan mag deze 1 worden weggelaten.

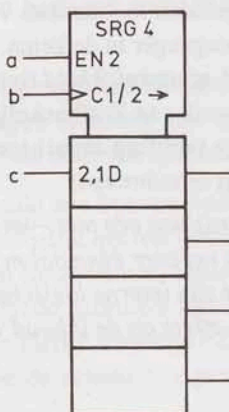
Wordt een schuifingang gebruikt in een register met flip-flops met uitgestelde uitgangen,

dan heeft de werking van de uitgestelde uitgang ook betrekking op de schuifingang, volgens de gespecificeerde timing. Aangezien de interne logische waarden van de uitgangen van een element overeenkomen met die van de inwendige toestand van het element, wordt bij toepassing van uitgestelde uitgangen ook het schuiven van de inhoud van het register uitgesteld.

Past men de schuifingang toe op de configuratie van figuur 74.a, dan ontstaat het in figuur 74.b weergegeven symbool. Allereerst dient hierin, om aan te geven dat het een schuifregister betreft, het *functiesymbool van een schuifregister* te worden vermeld. Dit bestaat uit de lettercombinatie SRG gevolgd door een decimale aanduiding van het aantal secties.

Voorts is ingang a weergegeven als een ingang met twee functies. De reden hiervan is deze. De schuifingang is geen beïnvloedende ingang, die via een of meer ervan afhankelijke data-ingangen zijn invloed doet gelden. De schuifingang heeft slechts betrekking op het *interne* datatransport in een register. Om de D-ingang van de bovenste sectie, waarop de externe serie-ingang b is aangesloten, te activeren is daarom een beïnvloedende C-ingang nodig.

Het komt vaak voor dat een schuifregister voorzien is van een enable-ingang. Passen we hiervoor EN-afhankelijkheid toe (A.16), dan ontstaat het symbool van figuur 76. We zijn er daarbij van uitgegaan dat de enable-ingang zijn invloed uitoefent door schakelen in de data en niet door schakelen in de klok (A.23/A.24).



Figuur 76. 4-Bit edge-triggered schuifregister met enable-ingang.

De schuifingang is afhankelijk gemaakt van de beïnvloedende EN-ingang, terwijl de afhankelijkheid van de serie-ingang niet via de beïnvloedende C-ingang maar bij de D-ingang zelf is aangegeven.

Voor schuifregisters met meer dan twee functies maken we bij voorkeur gebruik van MODE-afhankelijkheid.

A.26. M-afhankelijkheid

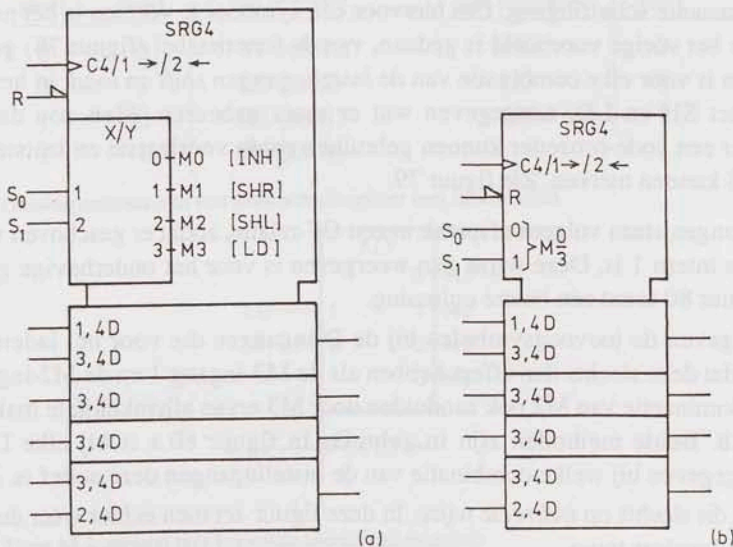
Mode-afhankelijkheid (M-afhankelijkheid/Mode dependency) wordt aangeduid met de letter M gevolgd door een identificatienummer. Een beïnvloedende M-ingang heeft op de ervan afhankelijke ingangen hetzelfde effect als een beïnvloedende EN-ingang. Als een

M-ingang of M-uitgang de interne logische waarde 1 heeft, dan hebben alle ingangen die van deze M-ingang of M-uitgang afhankelijk zijn hun normaal gedefinieerde invloed op de functie of toestand van het element. Uitgangen die van een M-ingang of M-uitgang afhankelijk zijn, bezitten hun normale gedefinieerde logische waarde, die als regel correspondeert met de toestand van het element.

Heeft een M-ingang of M-uitgang de interne logische waarde 0, dan hebben de ervan afhankelijke ingangen geen invloed op de functie van het element. Zij moeten dan als niet-bestaand worden beschouwd.

Gezien de definitie maakt het meestal geen verschil of men EN- dan wel M-afhankelijkheid toepast. Nu wordt EN-afhankelijkheid meestal gebruikt om van één bepaalde functie aan te geven of deze al dan niet verricht moet worden. M-afhankelijkheid wordt echter meestal gebruikt om in gevallen waar een keuze mogelijk is uit een aantal verschillende functies, aan te geven welke van deze functies uitgevoerd moet worden.

We stellen als voorbeeld symbolen op voor het register SN74LS194A (figuur 12.4, tekst). Dit register heeft, naast een directe reset, vier mogelijke functies in zich, die met twee instelsignalen s_0 en s_1 worden ingesteld. De vier mogelijke functies worden in gecodeerde vorm door s_0 en s_1 ingesteld. Deze gecodeerde informatie moet nu intern eerst gedecodeerd worden naar vier mode-signalen M0 t/m M3, één voor elk der in te stellen functies (INH/SHL/SHR/LD). In figuur 77.a is hiervoor een (interne) code-omzetter toegepast. Een code-omzetter heeft het functiesymbool X/Y, waarvan X de ingangscodes en Y de uitgangscodes symboliseert. Aldus wordt de 2-bit ingangscodes naar een 1-uit-4 uitgangscodes omgezet. Elke uitgang van de code-omzetter correspondeert nu met één mode-ingang op het gemeenschappelijk besturingsblok. Van de vier M-ingangen heeft er steeds één de interne logische waarde 1. De overige zijn intern 0 en de ervan afhankelijke ingangen worden dan als niet-bestaand beschouwd. Wordt de directe reset, die natuurlijk dominant is, even buiten beschouwing gelaten, dan blijkt in de mode 0 alleen de C-ingang zijn werk te kunnen doen. Omdat de schuif- en D-ingangen echter alle disabled zijn, heeft dit geen enkel effect.



Figuur 77. Symbolen voor het register SN74LS194A.

In mode 1 (M1 [SHR]) is de *eerste schuifingang* ($1 \rightarrow$) geactiveerd alsmede de bovenste D-ingang, de *serie-ingang* SHR, zodat naar beneden wordt geschoven. In mode 3 tenslotte zijn alleen de middelste vier D-ingangen geactiveerd. Er vindt dan parallel laden plaats.

Omdat een M-ingang hetzelfde effect op ervan afhankelijke ingangen heeft als een EN-ingang (niet bestaand of de normaal gedefinieerde werking), is zijn timing ook dezelfde. Hieruit volgt op dezelfde wijze als bij de EN-ingang in figuur 76 dat de mode-ingangen en dus ook de instelingen s_0 en s_1 een edge-triggered timing bezitten.

Omdat ingang M0 in figuur 77.a geen ervan afhankelijke ingangen of uitgangen bezit, kan deze in feite worden weggelaten, tezamen met het bijbehorende korte streepje en de 0 in de code-omzetter.

Figuur 77.b toont een enigszins vereenvoudigd symbool waarin de code-omzetter is vermeden door het bitgroeperingsymbool (A.17) toe te passen. Het nadeel is dat hierdoor de mogelijkheid om aanvullende informatie te vermelden vervalst.

Opmerking

De *identificatienummers van de M-ingangen* behoeven niet gelijk te zijn aan de getallen bij de uitgang van de code-omzetter. Ook bij verkorte notatie met het bitgroeperingssymbool hoeft dit niet.

Dominante instelingen

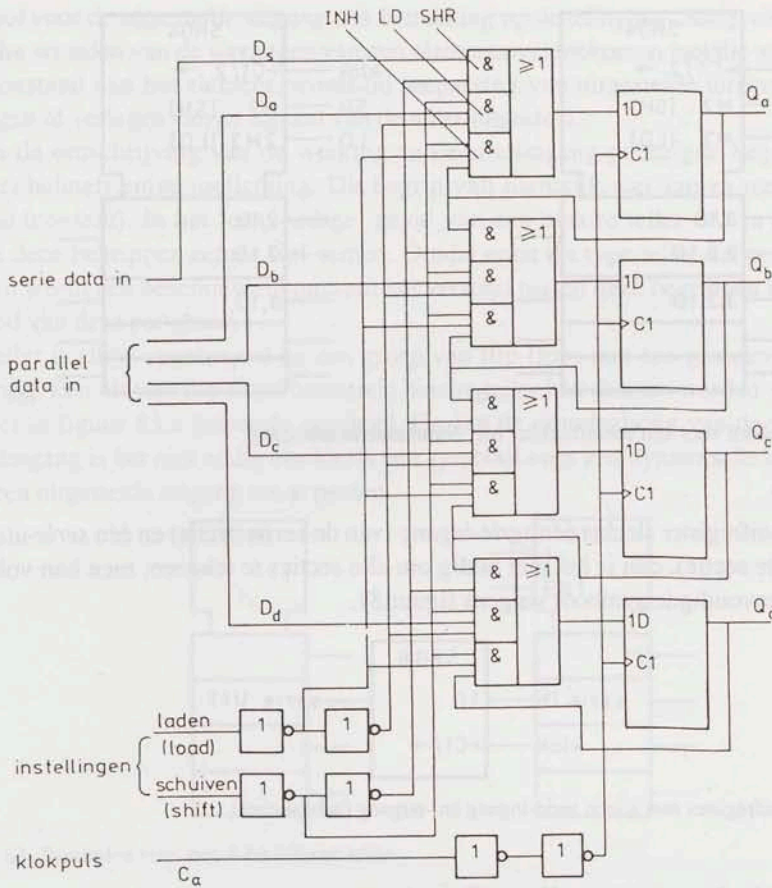
Bij een aantal types registers, tellers e.d., komen instelingen voor die dominant zijn over andere. Indien we dit zonder meer met een code-omzetter/mode afhankelijkheid zouden weergeven dan wordt het symbool vaak erg groot. Hiervoor bestaan betere oplossingen.

Voorbeeld

In figuur 78 is het ontwerpschema van een 4-bit schuifregister beschreven. Dit register heeft een dominante schuifingang. Om hiervoor een symbool te vormen is het nuttig om, evenals dit in het vorige voorbeeld is gedaan, van de functietabel (figuur 78) gebruik te maken. Hierin is voor elke combinatie van de instelingen *shift* en *load*, in het vervolg aangeduid met SH en LD, aangegeven wat er moet gebeuren. Men zou dan in het symbool weer een code-omzetter kunnen gebruiken en de voorlaatste en laatste uitgang beide met MS kunnen merken. Zie figuur 79.

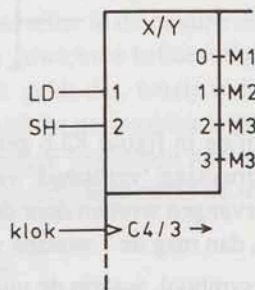
Deze M3-ingangen staan volgens afspraak in een OF-relatie, zodat er geschoven wordt als een van beide intern 1 is. Deze wijze van weergeven is voor het onderhavige geval vrij complex. Figuur 80 toont een betere oplossing.

In figuur 80 geven de toevoegsymbolen bij de D-ingangen die voor het laden worden gebruikt aan dat deze slechts dan effect hebben als de M3-ingang 1 en de M2-ingang 0 is. Men kan de dominantie van M2 ook aanduiden door M3 ervan afhankelijk te maken zoals in figuur 80.b. Beide methoden zijn in gebruik. In figuur 80.a is bij elke D-ingang expliciet aangegeven bij welke combinatie van de instelingen deze actief is. In figuur 80.b ziet men dit slechts op *indirecte wijze*. In deze figuur zet men echter weer duidelijker het dominante karakter terug.

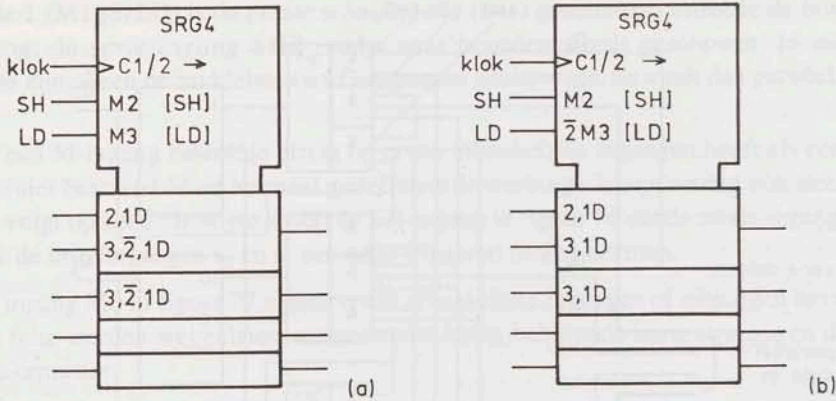


Instelling		Functie van het register	Functie van het register	Instelling	
shift	load			shift	load
0	0	INH (inhibit)	INH	0	0
0	1	LD (load)	LD	0	1
1	0	SH (shift)	SH	1	-
1	1	SH (shift)			

Figuur 78. Ontwerpschema van een 4-bit schuifregister met functietabel.



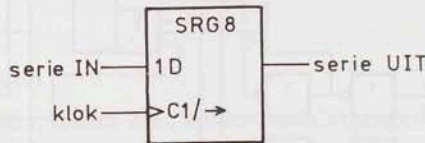
Figuur 79. Twee M-ingangen met hetzelfde identificatienummer.



Figuur 80. Symbool voor een schuifregister met dominante instellingang.

Opmerking

Heeft een schuifregister slechts één *serie-ingang* (van de eerste sectie) en één *serie-uitgang* (van de laatste sectie), dan is het niet nodig om alle secties te tekenen, men kan volstaan met het vereenvoudigde symbool volgens figuur 81.



Figuur 81. Schuifregister met alleen serie-ingang en -uitgang (acht secties).

A.27. Symbolen voor tellers I

In symbolen voor tellers maakt men gebruik van het symbool voor een *telingang* (*counting input*). De werking ervan (figuur 82) wordt als volgt omschreven:

Telkens als de interne logische waarde van een +m-ingang verandert van 0 naar 1 wordt de telstand van de teller met m verhoogd. Verandert zijn interne logische waarde niet, of gaat deze van 1 naar 0, dan heeft deze ingang geen effect op de telstand van de teller, tenzij deze laatste een of meer uitgestelde uitgangen heeft.



Figuur 82. Telingang.

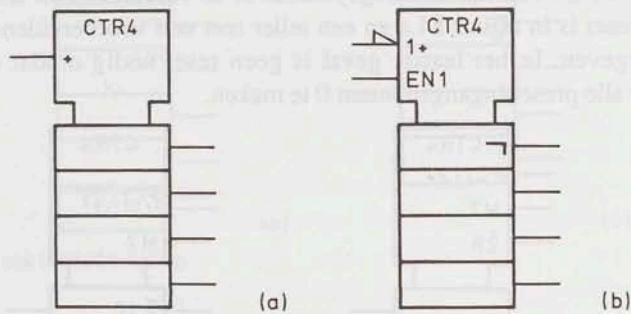
De omschrijving van de werking van de in figuur 82.b getekende telingang wijkt hier slechts in zoverre van af, dat de uitdrukking 'verhoogd' vervangen moet worden door 'verlaagd'. De letter m moet altijd vervangen worden door de 'werkelijke' waarde, genoteerd in het tientallig stelsel. Is $m = 1$, dan mag de 1 worden weggelaten.

Wordt een telingang gebruikt in een symbool, waarin de uitgangen gemerkt zijn met het toevoegsymbool voor een uitgestelde uitgang, dan heeft de werking van dit *toevoeg-*

symbool voor de *uitgestelde uitgang* ook betrekking op de *telingang*. Aangezien de interne logische waarden van de uitgangen van een element overeenkomen met die van de inwendige toestand van het element, wordt bij toepassing van uitgestelde uitgangen ook het verhogen of verlagen van de inhoud van de teller uitgesteld.

Het in de omschrijving van de werking van een *telingang* gebezigde begrip *telstand* (*count*) behoeft enige toelichting. Dit begrip valt namelijk niet samen met het begrip *inhoud* (*content*). In het 'eenvoudige' geval van een binaire teller of een decadeteller vallen deze begrippen echter wel samen. Omdat eerst dit type teller nader bekeken zal worden wordt een beschouwing omtrent het verschil tussen deze begrippen uitgesteld tot het eind van deze paragraaf.

Een teller is altijd opgebouwd uit een groep van flip-flops met een gemeenschappelijke besturing. Een uit vier flip-flops bestaande binaire teller kan daarom worden weergegeven met het in figuur 83.a getoonde symbool. Gezien de omschrijving van de werking van een *telingang* is het niet nodig om hierin een symbool voor een dynamische ingang of dat voor een uitgestelde uitgang toe te passen.



Figuur 83. Symbolen voor een 4-bit binaire teller.

Om aan te geven dat het symbool een teller representeert is in het gemeenschappelijk ingangsblok (c.q. besturingsblok) het *functiesymbool* CTR vermeld.

Wordt dit functiesymbool gevolgd door een getal (m) dan geeft dit niet alleen het aantal secties aan, maar het geeft tevens aan dat de cyclusbreedte van de teller 2^m bedraagt (in figuur 83.a dus $2^4 = 16$). Is de cyclusbreedte geen macht van 2, dan wordt achter het functiesymbool de aanduiding DIV_m toegevoegd, waarin m dan vervangen moet worden door het decimale getal dat de 'werkelijke' *cyclusbreedte* weergeeft. In het geval van een *decadeteller* is het functiesymbool dus CTRDIV10.

De inwendige toestanden van de in het symbool met rechthoeken aangeduide flip-flops representeren de inhoud van de teller in de *binaire code*. Zij hebben daarom verschillende gewichten (1,2,4,8,...). Deze gewichten behoeven niet te worden aangegeven, omdat in het IEC-systeem de afspraak geldt dat, tenzij anders is aangegeven, de sectie die het dichtst bij het gemeenschappelijk besturingsblok is getekend de *laagstwaardige sectie* is en de volgende secties opklimmende gewichten hebben. Het is daarom niet nodig om toevoegingsymbolen te plaatsen in de rechthoeken die de secties representeren.

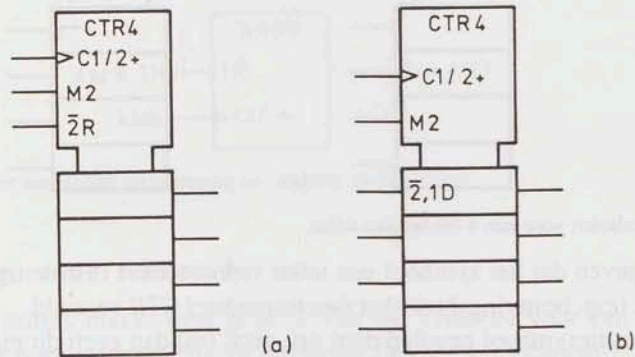
Tellers met enable-ingang

Het komt vaak voor dat een teller van een *enable-ingang* is voorzien. Schakelt deze in de data (in de data-ingangen van de secties), dan ontstaat met betrekking tot de timing van de

enable-ingang dezelfde problematiek als bij een schuifingang. De omschrijvingen van de schuif- en de telingang vertonen namelijk een grote mate van overeenkomst. Zij zijn allebei klok- en data-ingang tegelijk. De aanduiding van de timing van een enable-ingang met betrekking tot een telingang gebeurt daarom op dezelfde manier als bij de schuifingang. Zo representeert het in figuur 83.b getoonde symbool een binaire teller met cycluslengte 16 (standen 0 t/m 15) en met een enable-ingang met pulse-triggered timing.

Tellers met een reset en een preset

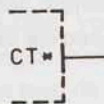
Veel tellers hebben de mogelijkheid om de inwendige toestanden van de secties 0 te maken (*reset*) of om deze een bepaalde voorinstelling te geven (*preset* of *parallel laden*). De reset kan zowel direct werkend als voorbereidend zijn, de preset is altijd voorbereidend. Is een reset of preset voorbereidend, dan is een (voorbereidende) mode-ingang nodig om het telmechanisme tijdens de reset of preset uit te schakelen. Een directe reset kan uiteraard worden aangeduid door deze ingang aan het gemeenschappelijke ingangsblok te tekenen en van het toevoegsymbool R te voorzien. Een teller met een voorbereidende reset is in figuur 84.a en een teller met een voorbereidende preset in figuur 84.b weergegeven. In het laatste geval is geen reset nodig omdat deze gegeven kan worden door alle preset-ingangen intern 0 te maken.



Figuur 84. Tellers met voorbereidende reset en preset.

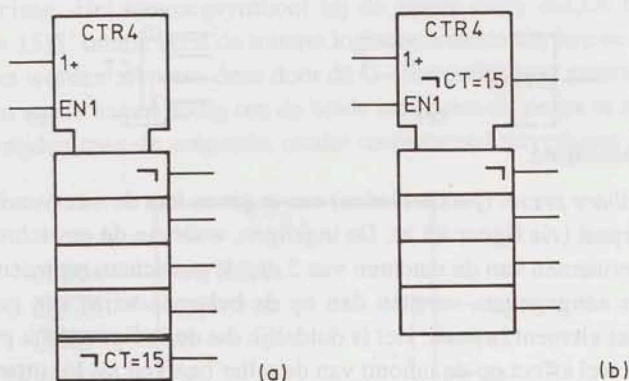
Tellers met een inhoudsuitgang

Veel tellers hebben een speciale uitgang die intern 1 is als de teller zijn *eindstand* bereikt heeft of als de inhoud aan een bepaalde voorwaarde voldoet. Dergelijke uitgangen kunnen worden aangeduid m.b.v. het symbool voor een *inhoudsuitgang* (*content output*), zie figuur 85. In dit symbool moet '*' vervangen worden door een 'geschikte en duidelijke' aanduiding van die inhouden van de teller waarbij de uitgang intern 1 is, bijvoorbeeld $CT = 15$ of $CT \geq 10$.



Figuur 85. Inhoudsuitgang.

Omdat een CT-uitgang in de meeste gevallen een functie van de uitgangen van alle secties is, wordt hij in het geval dat de teller is weergegeven als een groep van flip-flops met een gemeenschappelijk besturingsblok (zoals tot nu toe steeds gedaan is) als uitgang van een *gemeenschappelijk uitgangselement* getekend. Figuur 86.a laat dit zien. Het gemeenschappelijk uitgangselement behoeft in dit geval *geen functiesymbool* te bevatten omdat de werking ervan volledig bepaald is door het toevoegsymbool bij de uitgang. In A.12 is gesteld dat een gemeenschappelijk uitgangselement ook in een gemeenschappelijk ingangsblok getekend mag worden. Is de functie van de gemeenschappelijke uitgang volledig bepaald door diens toevoegsymbool, dan is het in dat geval niet nodig om de rechthoek, die het gemeenschappelijk uitgangselement representeert, te tekenen (zie figuur 86.b).



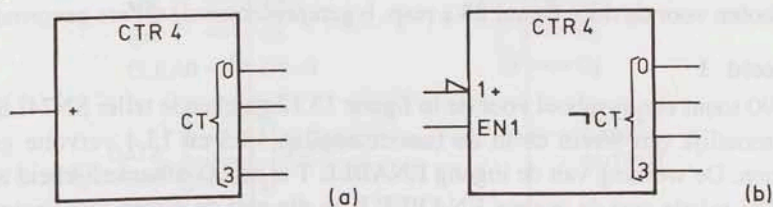
Figuur 86. Teller met een inhoudsuitgang.

Opmerking

In figuur 86 is de inhoudsuitgang voorzien van het toevoegsymbool voor een *uitgestelde uitgang*. Strikt genomen is dit niet noodzakelijk, omdat dit symbool al bij de uitgangen, waarvan het een functie is, is aangegeven. Om alle misverstand te voorkomen wordt dit toevoegsymbool echter meestal bij de gemeenschappelijke uitgang herhaald.

A.28. Symbolen voor tellers II

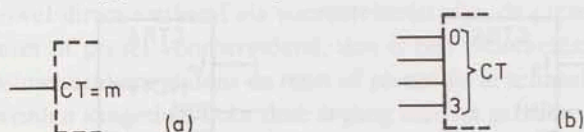
Tot nu toe zijn in de symbolen voor tellers alle secties door rechthoeken weergegeven. Dit is echter niet nodig als de *meervoudige inhoudsuitgang* wordt toegepast. Zo tonen de figuren 87.a en b alternatieven voor de symbolen van figuur 83.a resp. b. De aanduiding CT gevolgd door een accolade geeft aan dat de inhoud van de teller op de uitgangen wordt overgedragen. Deze hebben dan *gewichten* die machten van 2 zijn, waarvan de expo-



Figuur 87. Symbolen met een meervoudige inhoudsuitgang.

nenten achter de accoladen zijn aangegeven. Indien niet alle gewichten machten van 2 zijn moeten hun 'werkelijke' decimale waarden worden vermeld.

Heeft een teller die door een enkelvoudig symbool wordt gerepresenteerd een *reset-ingang*, dan is het (hoewel dit soms toch wordt gedaan) onjuist om deze met de letter R aan te duiden. Een R-ingang is namelijk een data-ingang van een enkelvoudig geheuelement (latch, flip-flop) dat één enkel bit onthoudt. Om een teller de inhoud 0 te geven kan men daarom beter gebruik maken van het symbool voor een *inhoudsingang* (*content input*), zie figuur 87.a. In dit symbool moet m vervangen worden door de 'werkelijke' waarde die de inhoud van het element aanneemt als deze ingang intern 1 is. Is hij intern 0, dan heeft hij geen invloed op de inhoud van het element.



Figuur 88. Inhoudsingang.

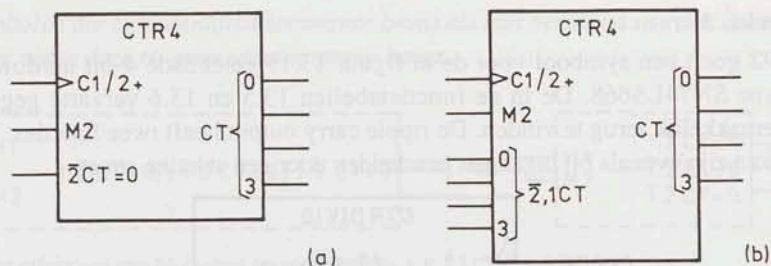
Om een *instelbare preset* (parallel laden) aan te geven kan de *meervoudige inhoudsingang* worden toegepast (zie figuur 88.b). De ingangen, waarvan de gewichten of, zoals in deze figuur, de exponenten van de machten van 2 die de gewichten representeren, links van de accolade zijn aangegeven, vormen dan op de bekende wijze een getal dat de nieuwe inhoud van het element bepaalt. Het is duidelijk dat de in figuur 88.a getoonde ingang als hij intern 1 is wel effect op de inhoud van de teller heeft en als hij intern 0 is niet. Hij kan daarom net als de S- en de R-ingang zowel direct werkend als voorbereidend zijn. De in figuur 88.b getoonde meervoudige ingang kan uitsluitend een voorbereidend karakter hebben. Hij heeft immers net als een D-ingang altijd een effect op de inhoud van het element (ook als alle samenstellende ingangen intern 0 zijn), tenzij hij disabled is. Uit dit alles volgt dat de inhoudsingang een data-ingang is. De werking van de enkelvoudige inhoudsingang is vergelijkbaar met die van een S- of R-ingang, en de werking van de meervoudige inhoudsingang met die van een D-ingang.

Het *identificatienummer* van een beïnvloedende ingang waarvan een inhoudsingang afhankelijk is wordt links van de lettercombinatie CT geplaatst; bij een meervoudige inhoudsingang dus tussen de accolade en CT. Zijn de individuele ingangen van een meervoudige inhoudsingang alle afhankelijk van dezelfde beïnvloedende ingang (bijvoorbeeld een G-ingang die hen alle intern 0 kan maken), dan mag diens identificatienummer eveneens tussen de accolade en CT geplaatst worden.

Met behulp van het voorafgaande kunnen nu de in figuur 89.a en b getoonde enkelvoudige symbolen voor de door figuur 84.a resp. b gerepresenteerde tellers gevormd worden.

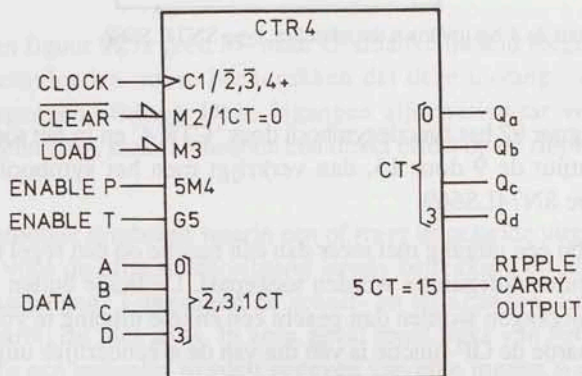
Voorbeeld 1

Figuur 90 toont een symbool voor de in figuur 13.12 getekende teller SN74LS163A. Het is niet moeilijk om hierin de in de functietabellen 13.3 en 13.4 vervatte gegevens te herkennen. De werking van de ingang ENABLE T is met G-afhankelijkheid aangegeven om de EN-relatie met de ingang ENABLE P en die met de ripple carry output te benadrukken. Bij de overige beïnvloedende ingangen is M-afhankelijkheid toegepast omdat de



Figuur 89. Enkelvoudige symbolen voor tellers met een voorbereidende reset en preset.

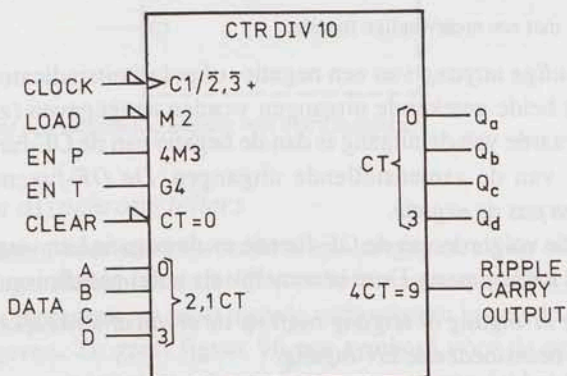
ervan afhankelijke telingang en de CT-ingang (denk aan diens overeenkomst met de D-ingang) dit vereisen. Het toevoegsymbool bij de ripple carry output zou eigenlijk moeten zijn '(CT = 15)5' omdat eerst de interne logische waarde als functie van de teller gespecificeerd moet worden alvorens deze door de G-afhankelijkheid gemodificeerd kan worden. Er zijn dan ronde haken nodig om de beide indicaties duidelijk te scheiden. Om deze haken te vermijden mag de volgorde, omdat misverstand uitgesloten lijkt, worden omgekeerd.



Figuur 90. Symbool voor de teller type SN74LS163A.

Voorbeeld 2

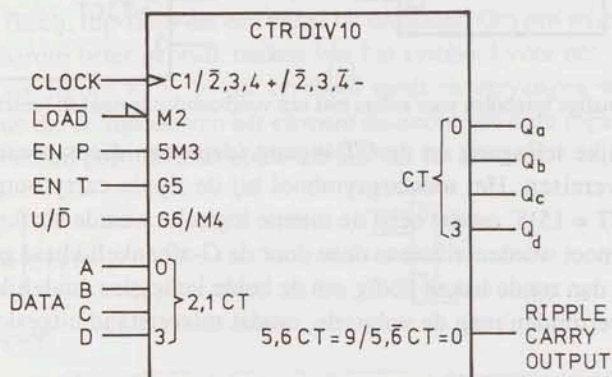
Figuur 91 toont een symbool voor de teller SN74LS160A. De werking is hierin gemakkelijk te herkennen.



Figuur 91. Symbool voor de decadeteller SN74LS160A.

Voorbeeld 3

Figuur 92 geeft een symbool voor de in figuur 13.15 getekende 4-bit up/down decade-teller, type SN74LS668. De in de functietabellen 13.5 en 13.6 vervatte gegevens zijn hierin gemakkelijk terug te vinden. De ripple carry output heeft twee functies. De indicaties hiervan zijn evenals bij ingangen gescheiden door een schuine streep.

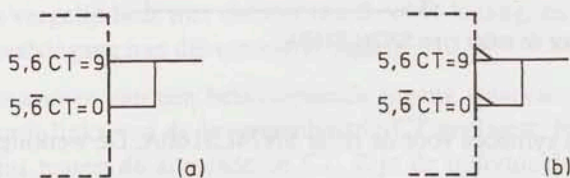


Figuur 92. Symbool voor de 4-bit up/down decadeteller, type SN74LS668.

Opmerking

Vervangt men in figuur 92 het functiesymbool door 'CTR 4' en in het toevoegsymbool bij de ripple carry output de 9 door 15, dan verkrijgt men het symbool voor de binaire up/down teller, type SN74LS669.

Om alle indicaties bij een uitgang met meer dan één functie op één regel te plaatsen kan de in figuur 93 getoonde configuratie worden toegepast. De beide buiten het symbool met elkaar verbonden uitgangen worden dan geacht één enkele uitgang te vormen waarvan de interne logische waarde de OF-functie is van die van de afzonderlijke uitgangen.



Figuur 93. Uitgangen met een meervoudige functie.

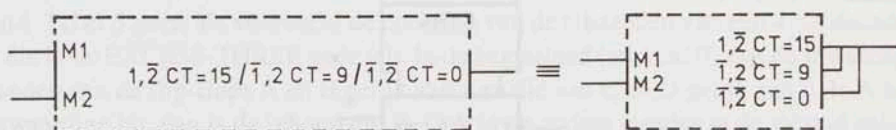
Moet een *meervoudige uitgang* van een negatie- of polariteitsindicator worden voorzien, dan moet deze bij beide getekende uitgangen worden aangegeven (zie figuur 93.b). De *externe* logische waarde van de uitgang is dan de negatie van de OF-functie van de *interne* logische waarden van de samenstellende uitgangen. *De OF-functie wordt dus eerst toegepast en daarna pas de negatie.*

Dit probleem van de volgorde van de OF-functie en de negatie kan vermeden worden door M-afhankelijkheid toe te passen. Deze is namelijk als volgt gedefinieerd.

Een beïnvloedende M-ingang of uitgang heeft op de ervan afhankelijke ingangen dezelfde uitwerking als een beïnvloedende EN-ingang.

Heeft hij de interne logische waarde 1, dan heeft hij geen effect op de ervan afhankelijke ingangen. Is hij daarentegen intern 0, dan moet bij een ervan afhankelijke uitgang elke rij

toevoegsymbolen die zijn identificatienummer bevat als niet bestaand worden beschouwd, althans voor zover deze rij geen schuine streep bevat.



Figuur 94. Het effect van een M-ingang op een uitgang.

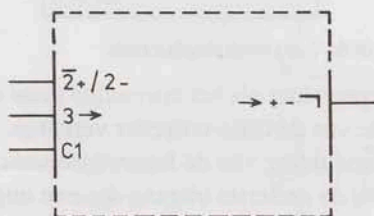
Figuur 94.a toont een voorbeeld. Is bijvoorbeeld ingang M1 intern 0 en ingang M2 intern 1, dan bepaalt alleen de middelste rij de functie van de uitgang. Deze is dus intern 1 als de inhoud van het element gelijk is aan 9. De overige rijen hebben dan geen enkel effect op de interne logische waarde van de uitgang. Figuur 94.b laat het alternatief zien met uitwendig doorverbonden uitgangen. Omdat er steeds maar één van de samenstellende uitgangen effect heeft en de overige als niet bestaand moeten worden beschouwd is toepassing van de OF-functie overbodig.

Opmerking

In het symbool van figuur 92 is geen M- maar G-afhankelijkheid toegepast om de ripple carry output te beïnvloeden om te benadrukken dat deze uitgang via poorten van de beïnvloedende ingangen afhangt. Deze ingangen zijn weliswaar voorbereidend met betrekking tot de telingang, maar zij hebben een direct effect op de ripple carry output.

Opmerking

In de tot nu toe getoonde symbolen waarin een of meer uitgestelde uitgangen voorkomen had het symbool voor de uitgestelde uitgang steeds betrekking op alle in het symbool voorkomende C-ingangen, T-ingangen en schuif- en telingangen. Er doen zich echter gevallen voor waarbij dit niet zo is. In zo'n geval moet links van het symbool voor de uitgestelde uitgang een indicatie worden gegeven van elke ingang waarop het symbool betrekking heeft. Figuur 95 geeft een voorbeeld.

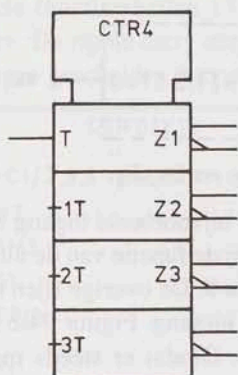


Figuur 95. Symbool voor een uitgestelde uitgang met extra indicaties.

Representatie van asynchrone tellers

Enkelvoudige symbolen voor tellers worden alleen toegepast als de teller synchroon is, d.w.z. als alle geheugenelementen onder directe besturing van de klokingang staan. Om *asynchrone tellers* te representeren moet men de verbindingen tussen de secties (voorlopig nog) expliciet aangeven. Zo geeft figuur 96 een symbool voor de asynchrone binaire tweedelerteller. Indien men het niet nodig vindt om aan te geven dat het symbool een teller representeert mag het gemeenschappelijk ingangsblok worden weggelaten. De werking

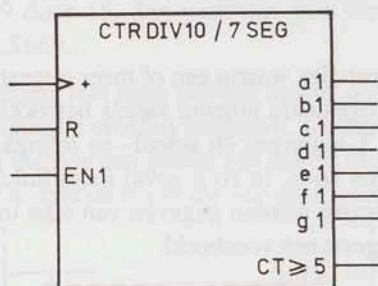
van de teller ligt dan nog volledig vast. Het functiesymbool CTR4 mag niet in de bovenste sectie geplaatst worden omdat dat zou betekenen dat de schakeling uit vier 4-bit tellers bestaat.



Figuur 96. Symbool voor een asynchrone binaire tweedelerteller.

Combinatie van het symbool voor een teller en dat voor een code-omzetter

Het kan voorkomen dat de inhoud van een teller niet in de binaire maar in een andere code op de uitgangen wordt overgedragen. Figuur 97 toont een voorbeeld van een decadeteller, die zijn inhoud in de 7-segment display code afgeeft.



Figuur 97. Teller met uitgangen in de 7-segment display code.

De *inhoud* van de teller fungeert hier als het *inwendige getal* van de code-omzetter, zodat de teller het ingangsgedeelte van de code-omzetter vervangt. In het uitgangsgedeelte van de code-omzetter is geen aanduiding van de lettercombinatie CT nodig. Deze is daarom ook niet vermeld behalve bij de onderste uitgang die een uitgang van de teller is en niet van de code-omzetter. Voor de enable-functie is geen 'gewone' niet-beïnvloedende EN-ingang toegepast omdat deze ook effect zou hebben op de met 'CT ≥ 5' gemerkte uitgang.

De telstand en de inhoud van een teller

Het verschil tussen de telstand en de inhoud van een teller komt tot uiting bij tellers die niet in de binaire code tellen.

Onder de *telstand* wordt verstaan het getal dat aangeeft hoeveel stappen de teller vanuit een bepaalde beginstand (de 0-stand) heeft gedaan.

De *inhoud* van de teller is het getal dat in de binaire code gerepresenteerd wordt door de

combinatie van de inwendige toestanden van de in de teller aanwezige geheugenelementen (meestal flip-flops).

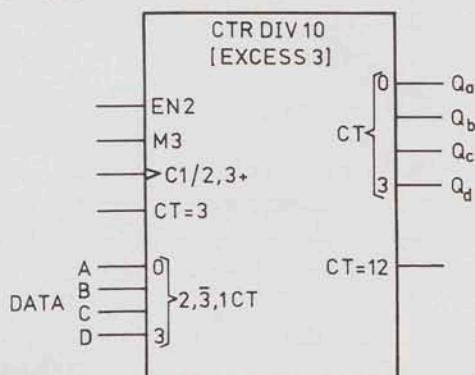
Telt de teller nu niet in de binaire code, dan is zijn inhoud niet altijd gelijk aan zijn telstand. Tabel 5 geeft als voorbeeld de codering van de telstanden van een 4-bit decade-teller die in de EXCESS-THREE code telt. In de beginstand (telstand 0) zijn de inwendige toestanden van de flip-flops A en B gelijk aan 1 en die van C en D gelijk aan 0. Is A het laagstwaardige bit, dan is de inhoud dus 3. Ook in de andere standen is de inhoud gelijk aan de telstand vermeerderd met 3. Dit verklaart ook de naam van de code.

telstand	codering				inhoud
	D	C	B	A	
0	0	0	1	1	3
1	0	1	0	0	4
2	0	1	0	1	5
3	0	1	1	0	6
4	0	1	1	1	7
5	1	0	0	0	8
6	1	0	0	1	9
7	1	0	1	0	10
8	1	0	1	1	11
9	1	1	0	0	12

Tabel 5. Codering van telstanden in de EXCESS-THREE code.

Figuur 98 toont een symbool voor de teller. Omdat de CT-ingangen en uitgangen *inhoudsin- en uitgangen* zijn hebben zij betrekking op de inhoud van de teller en niet op de telstand. De direct werkende reset die de teller in de 0-stand moet zetten is daarom aangegeven door 'CT = 3'.

De eindstand van de teller (telstand 9) wordt gedetecteerd door de uitgang met het toevoegsymbool 'CT = 12'.



Figuur 98. Symbool voor een 4-bit decadeteller die in de EXCESS-THREE code telt.

Wil men de teller via de data-ingangen in stand 5 zetten, dan moet men omdat het een EXCESS-THREE teller is de combinatie DCBA = 1000 aanbieden. Deze combinatie wordt wegens de binaire gewichten (eigenlijk exponenten) binnen de accolade omgezet in

CT = 8. De inhoud van de teller wordt dus 8, hetgeen overeenkomt met de bedoelde telstand 5. Aan de uitgangen verschijnt de combinatie $Q_D Q_C Q_B Q_A = 1000$, die daar weer in de EXCESS-THREE code als 5 geïnterpreteerd wordt.

Onder het functiesymbool is de telcode tussen rechte haken aangegeven. Is dit geen 'bekende' code dan kan net als in het symbool voor een code-omzetter met behulp van de indicatie '[T1]' (of '[T2]', enz.) naar een tabel worden verwezen waarin de telcode is aangegeven.

Slotopmerking

Bij eerste kennismaking lijkt het IEC-systeem ingewikkeld en niet gemakkelijk te hanteren. Dit ligt echter niet aan het systeem, maar aan de complexiteit van de structuren die ermee beschreven moeten kunnen worden. Het is een taal, waarin het mogelijk is om het uitwendig gedrag van een schakeling zeer compact en met een minimum aan tekenwerk weer te geven en wel op een zodanige wijze dat het de gebruiker de essentiële informatie verschaft die hij nodig heeft.

1618984

1618984

