

Handleiding SLS

30 september 2019

1 Inleiding

SLS is een afkorting van Switch Level Simulator en is een simulator welke gebruikt kan worden voor het simuleren van mos-circuits ter bepaling van het logische gedrag en timing eigenschappen. De simulator modelleert de mos-transistoren door geaarde capaciteiten en geschakelde weerstanden. Alle knooppunten in het netwerk krijgen na simulatie een logische toestand 0, 1 of X (ongedefinieerd) toegewezen.

Het SLS-pakket omvat 2 programma's: *csls* en *sls*. *csls* wordt gebruikt om een netwerk aan de circuit-database toe te voegen, *sls* is de eigenlijke simulator. Het gebruik van beide programma's wordt hieronder toegelicht.

2 Het gebruik van *csls* en de *sls*-taal

Met *csls* wordt een netwerk aan de circuit-database als volgt toegevoegd:

```
[op5u9/myproject]  csls <filenaam>
```

Het netwerk is in de database in een aantal binaire files opgeslagen. (Om van dit interne database-formaat weer een netwerk *sls*-beschrijving te krijgen moet men het programma *xsls* gebruiken.) De <filenaam> is een tekstfile waarin het toe te voegen netwerk in de *sls*-taal beschreven staat. De <filenaam> heeft bij voorkeur de naam van het netwerk met de extensie *.sls*. De beschrijving bestaat uit twee gedeelten. Eerst komt er een declaratie van externe netwerken. Dit zijn netwerken die reeds in de circuit-database aanwezig zijn en (eventueel) in het nieuwe netwerk als subcircuit aangeropen wordt. Het tweede deel bevat de eigenlijke netwerkbeschrijving. De externe declaratie is verplicht. Alle netwerken die in het nieuwe netwerk gebruikt worden moeten als extern netwerk gedeclareerd zijn.

Externe netwerkdeclaratie De externe netwerkdeclaratie begint met de sleutelwoorden **extern network**, gevolgd door de naam van het externe netwerk. Dan tussen ronde haakjes het sleutelwoord

terminal en alle externe aansluitingen gescheiden door komma's. De volgorde van deze aansluitingen hoeft niet gelijk te zijn aan de volgorde zoals deze was bij definitie van het netwerk. Wel legt het de volgorde vast van de aansluitingen bij de aanroep van het netwerk in het nieuw te beschrijven netwerk. Alle externe netwerkdeclaraties van de bibliotheekcellen worden bij het maken van een project met *mkopr* in een file *oplib.ext* in de project-subdirectory *sls_prototypes* gezet. Deze file kan als header (d.m.v. een `#include` statement) in een *sls*-file worden opgenomen.

Netwerkbeschrijving De netwerkbeschrijving begint met het sleutelwoord **network** gevolgd door de naam van het netwerk. Deze naam mag niet beginnen met een hoofdletter. Dit om verwarring te voorkomen met namen van netwerken die ontstaan uit extractie. Achter de naam van het netwerk komen tussen ronde haakjes het sleutelwoord **terminal** en alle externe aansluitingen gescheiden door komma's. De laatste twee aansluitingen moeten altijd *vss*, *vdd* zijn. Dan volgt een regel met accolade-open (`{`), gevolgd door een of meerdere regels met alle interne en externe subnetwerken waaruit het netwerk is opgebouwd, afgesloten met een punt-komma (`;`). De beschrijving eindigt met een accolade-sluiten (`}`).

De externe subnetwerken mogen vooraf gegaan worden met tussen accolades een specifieke naam voor dat subnetwerk. Dit kan handig zijn indien men onderscheid wil maken tussen subnetwerken die in de beschrijving meerdere keren voorkomen. Commentaar mag op elke plaats worden toegevoegd en moet staan tussen `/*` en `*/`.

De interne netwerken die gebruikt kunnen worden zijn:

| Naam | parameters | toelichting |
|------|---|---|
| cap | <i>waarde</i> (node1,node2) | Capaciteit ter grootte <i>waarde</i> in Farad |
| res | <i>waarde</i> (node1,node2) | Weerstand ter grootte <i>waarde</i> in Ω |
| nenh | <i>w=breedte l=lengte</i> (gate,drain,source) of (gate,source,drain) | enhancement nmos transistor De breedte en lengte parameters zijn in meters en optioneel |
| penh | <i>w=breedte l=lengte</i> (gate,drain,source) of (gate,source,drain) | enhancement pmos transistor De breedte en lengte parameters zijn in meters en optioneel |
| ndep | <i>w=breedte l=lengte</i> (gate,drain,source) of (gate,source,drain) | depletion nmos transistor De breedte en lengte parameters zijn in meters en optioneel |
| net | {node1,node2} | kortsluiting tussen node1 en node2 |

Naast deze ingebouwde interne netwerken kunnen een aantal logische functies als intern netwerk worden aangeroepen. Deze functies dienen vooraf gegaan te worden door het `@` karakter. De volgende logische functies zijn mogelijk:

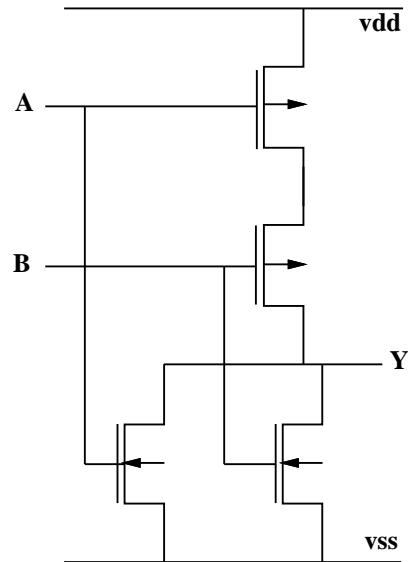
| Naam | parameters | toelichting |
|----------|------------------------------|------------------------------------|
| @ invert | (A, Y) | invertor met ingang A en uitgang Y |
| @ nand | ($A_1, A_2, \dots A_n, Y$) | n-input nand met uitgang Y |
| @ nor | ($A_1, A_2, \dots A_n, Y$) | n-input nor met uitgang Y |
| @ and | ($A_1, A_2, \dots A_n, Y$) | n-input and met uitgang Y |
| @ or | ($A_1, A_2, \dots A_n, Y$) | n-input or met uitgang Y |
| @ exor | ($A_1, A_2, \dots A_n, Y$) | n-input exclusive or met uitgang Y |

NB: De terminal-, netwerk-namen moeten voldoen aan de volgende voorwaarden:

- Een naam moet beginnen met een *letter* en mag verder alleen uit *letters*, *cijfers* en *underscores* (–) bestaan.
- De lengte van een naam mag niet groter zijn dan 14 karakters.

Voorbeelden Hieronder volgen twee voorbeelden van netwerkbeschrijvingen in de sls-taal.

Voorbeeld 1 :

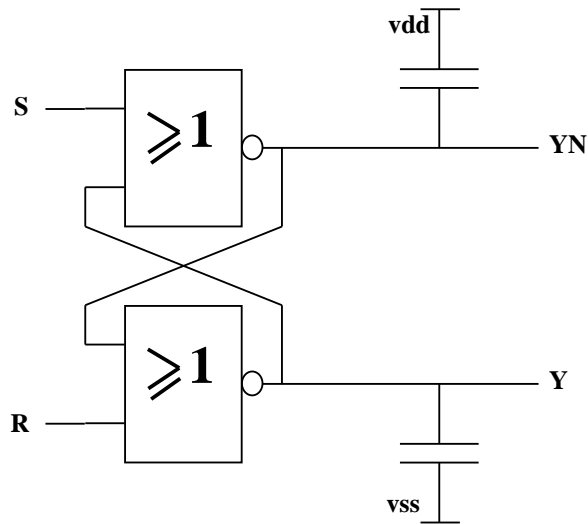


Figuur 1: Transistornetwerk van een 2-input nor

```
/* Netwerkbeschrijving van een 2-input nor nor2
   Dit voorbeeld heeft geen externe netwerk aanroep
   omdat alleen gebruik gemaakt wordt van interne netwerken */

network nor2 (terminal A, B, Y, vss, vdd)
{
    nenh l=1.6u w=23.2u (A, vss, Y);
    nenh l=1.6u w=23.2u (B, vss, Y);
    penh l=1.6u w=29.6u (B, 1, Y);
    penh l=1.6u w=29.6u (A, 1, vdd);
}
```

Voorbeeld 2 :



Figuur 2: SR-latch met 1 pF belasting aan beide uitgangen

```
/* Netwerkbeschrijving van een SR-latch met 1 pF belasting
   aan beide uitgangen */

extern network nor2 (terminal A, B, Y, vss, vdd)

network sr_latch (terminal S, R, Y, YN, vss, vdd)
{
    {nor1} nor2 (S, Y, YN, vss, vdd);
    {nor2} nor2 (R, YN, Y, vss, vdd);
    cap 1p (Y, vss);
    cap 1p (YN, vdd);
}
```

3 Gebruik van *sls* en de commandfile

De simulator *sls* kan zowel met *simeye* opgestart worden als vanaf de commandline. De aanroep van de commandline is als volgt:

```
[op5u9/myproject] sls <netwerdnaam> <commandfile>
```

<netwerdnaam> is de naam van het te simuleren netwerk, <commandfile> is de stimuli-file met bij voorkeur de extensie .cmd.

Voor het gebruik van *sls* in *simeye* wordt verwezen naar de handleiding van *simeye*.

In de commandfile worden de stimuli beschreven van de inputsignalen. Tevens kunnen een aantal opties meegegeven worden voor zowel *sls* als voor *spice*. Dit laatste kan belangrijk zijn indien het netwerk (b.v. vanuit *simeye*) met *spice* gesimuleerd wordt.

De structuur van de commandfile is als volgt:

```
inputstimuli en sls-opties
/*
*%
Transiënt-informatie voor spice
*%
Directe spice-commando's
*%
*/
```

Inputstimuli Alle inputsignalen worden gezet met het set-commando:

```
set A = (1*2 h*2)*2 1*~
```

Inputsignaal A is nu twee tijdstappen laag, 2 tijdstappen hoog, dan weer tweemaal laag en tweemaal hoog, en dan voor altijd laag.

Sls-opties Hieronder volgen een paar van de belangrijkste opties. Voor overige opties en instellingen wordt verwezen naar de "Switch Level Simulator User's Manual"

- option level = 1/2/3 (default is 1)
 - level = 1 : Puur logische simulatie zonder de circuitparameters te betrekken.
 - level = 2 : Logische simulatie gebaseerd op circuitparameters zoals de transistor-afmetingen, weerstandwaarden en capaciteitwaarden.
 - level = 3 : Logische simulatie en bepaling van vertragingstijden gebaseerd op circuitparameters zoals de transistor-afmetingen, weerstanden en capaciteiten.

- option `simperiod = n` (default `n` is oneindig)
Specificeert het aantal simulatietijdstappen.
- option `sigunit = getal` (default getal is 1)
Specificeert de duur van een tijdstap in seconden.
- option `outunit = macht_van_tien` (default `macht_van_tien` is macht van 10 het dichtst bij `sigunit`.
`Macht_van_tien` wordt aangeduid met de achterevoegsels `f,p,n,u,m,k,M,G`. Bijvoorbeeld: `10p`, `1u`.
Specificeert de tijdseenheid welke gebruikt wordt bij de uitvoer.
- option `outacc = macht_van_tien` (default `macht_van_tien` is `outunit`)
Specificeert de nauwkeurigheid van de simulatie-uitvoer (`outacc <= outunit`).
- option `sta_file is = on`
Deze regel specificeert of de file `<cel>.sta` moet worden ingelezen. Deze file wordt bij de automatische circuitgeneratie vanuit een toestandsdiagram door *kissis* aangemaakt en bevat de definitie van de toestandsvariabele `State`.
- option `initialize (full) random = on`
Soms is het nodig een schakeling te initialiseren in een bepaalde begintoestand, b.v. als flipflops zonder reset gebruikt worden. Dit voorkomt dat sommige knooppunten ongedefinieerd (X) blijven. Met 'full random' wordt de schakeling iedere keer anders geïnitieerd.
- define `XXXX` (`XXXX` is een naam voor een nieuw te definiëren variabele)
Met dit commando kunnen nieuwe variabelen gedefinieerd worden die een combinatie zijn van ingangs- of uitgangssignalen. Dit commando wordt o.a. gebruikt bij de definitie van de toestandsvariabele `State` bij de automatische circuitgeneratie vanuit een toestandsdiagram door *kissis*. `State` is dan een combinatie van alle flipflop-uitgangen die de toestand coderen.
- print `A,b,..` (`A,b,..` zijn inputsignalen en/of outputsignalen)
Deze regel specificeert welke inputsignalen en/of outputsignalen in de *sls*-uitvoer komen. Dus ook welke signalen bij *simeye* op het scherm getekend gaan worden.
- plot `A,b,..` (`A,b,..` zijn inputsignalen en/of outputsignalen)
Deze regel specificeert welke inputsignalen en/of outputsignalen in de *spice*-uitvoer komen. Dus ook welke signalen bij *simeye* op het scherm getekend gaan worden.

Transiënt-informatie voor spice In dit gedeelte kunnen een aantal transiëntparameters voor *spice* gezet worden:

| | | |
|---------------------|-------|---|
| <code>tstart</code> | getal | starttijd voor de output van de transiënt-analyse in seconden |
| <code>tstep</code> | getal | stapgrootte van de transiënt-analyse in seconden |
| <code>tfall</code> | getal | daaltijd van de inputsignalen in seconden |
| <code>trise</code> | getal | stijgtijd van de inputsignalen in seconden |
| <code>uic</code> | | use initial conditions" |
| <code>vhigh</code> | getal | spanningsniveau van het hoogsignaal in Volts |
| <code>vlow</code> | getal | spanningsniveau van het laagsignaal in Volts |

Directe spice-commando's In dit gedeelte kunnen commando's worden opgegeven die letterlijk aan de spice-invoerfile worden toegevoegd zoals:

```
.options cptime=100  
.ic v("3")=0 v("out")=5
```

Voor informatie over deze en andere mogelijke commando's wordt u verwezen naar de desbetreffende spice-handleidingen.

Voorbeeld commandfile sr_latch.cmd :

```
set S = l*2 h*2 l*6 h*2 l*8  
set R = l*6 h*2 l*6 h*2 l*2  
set vss = l*~  
set vdd = h*~  
print S ,R, Y, YN  
plot S, R, Y, YN  
option level = 3  
option sigunit = 1e-8  
option simperiod = 20  
option outacc = 10n  
/*  
*%  
tfall 0.5n  
trise 0.5n  
tstep 0.1n  
*%  
.options cptime = 500  
*%  
*/
```