

Cadence

Oppstart og Skjemategning

Dag T. Wisland

17. januar 2005

1 Introduksjon

Cadence er en komplett pakke for konstruksjon av elektroniske kretser og inneholder en rekke forskjellige verktøy både for høynivåbeskrivelse (VHDL, Verilog), skjemategning, simulering og fysisk utlegg av ASIC's. I tillegg finnes også mulighet for å konstruere PCB-kort samt programmering av FPGA's. Denne guiden vil i første rekke dekke de deler av Cadence som er nødvendig for å konstruere en integrert krets helt fra de enkle biter som opprettelse av bibliotek og tegning av kretsskjema til de mer kompliserte deler som simulering og verifisering av ferdig utlegg. Motivasjonen for å skrive dette heftet er at vi har erfart at studenter som skal bruke Cadence bruker unødig mye tid på å sette seg inn i programmet istedenfor å konsentrere seg om selve oppgaven man skal løse. Programmet har en meget omfattende on-line dokumentasjon hvor det ofte er vanskelig å sile ut den nødvendige informasjon man trenger for å utføre en oppgave. Dette heftet vil derfor kun inneholde det man trenger for å komme igjennom en standard designprosess. Har man behov utover dette må man konsultere Cadence on-line dokumentasjon.

2 Oppstart av Cadence

Før man kan begynne å bruke Cadence må man definere et nytt alias. Dette gjøres ved å legge følgende linjer inn i `~/.cshrc` hvis man bruker `tcsh`-shell:

```
alias amscad35 'source /local/vlsi/cadsoft/cadence/cds5.00/tools/dfII/
local/localinit351;/local/vlsi/cadsoft/des_kits/ams351/artist/bin/
ams_cds -tech c35b4 -nologo -mode fb'
```

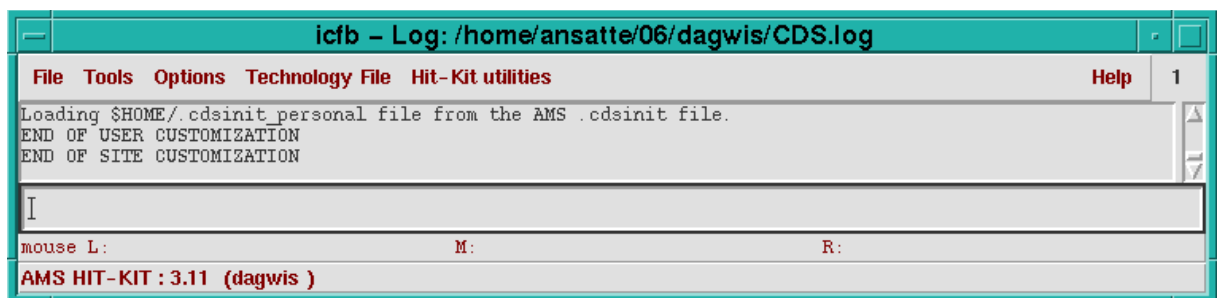
Hvis man bruker `bash`-shell må følgende linjer legges inn i `.bashrc`:

```
alias amscad35='source /local/vlsi/cadsoft/cadence/cds5.00/tools/dfII/
local/localinit351.bash;/local/vlsi/cadsoft/des_kits/ams351/artist/bin/
ams_cds -tech c35b4 -nologo -mode fb'
```

Pass på at det ikke er linjeskift i aliasdefinisjonen.
Pass på at du sitter ved en **Sun-maskin** og start programmet på følgende måte:

1. `mkdir ~/cadence`
2. `cd ~/cadence`
3. `amscad35 & (0.35um)`

Nå skulle programmet starte og to vinduer vil dukke opp på skjermen som vist i figur 1 og figur 2.



Figur 1: Hovedvinduet i Cadence.

3 Organisering av data

Selve hovedvinduet til Cadence er illustrert i figur 1. Dette vinduet brukes imidlertid stort sett bare når man skal avslutte programmet. Dette gjøres ved å klikke *File* → *Exit*. Det andre vinduet som kalles **Library Manager** har en meget sentral funksjon og holder orden på biblioteker og celler. Under kolonnen merket **Library** ligger alle tilgjengelige biblioteker, både predefinerte og de man oppretter selv. Generelt er bibliotekssystemet i Cadence organisert etter følgende struktur:

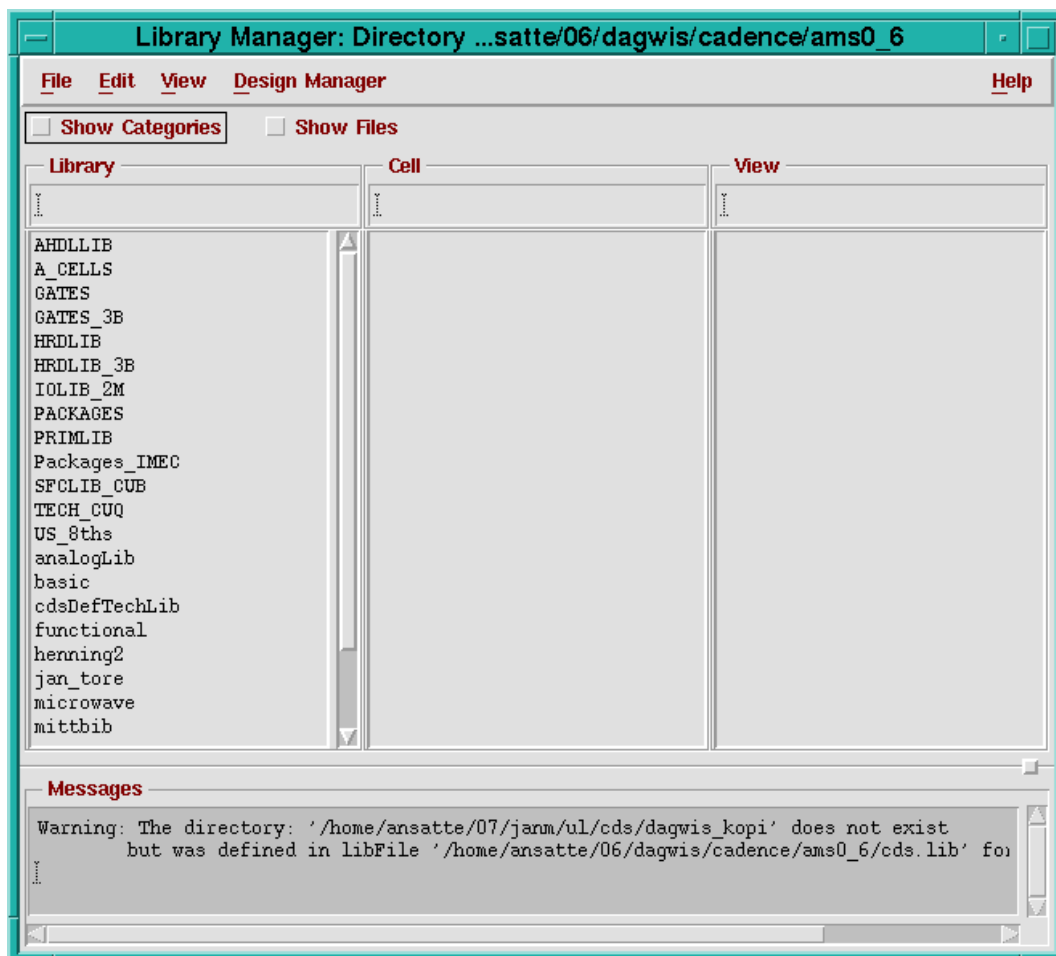
-Library: Biblioteker.

-Cell: Celler under hvert bibliotek.

-View: Hver celle kan ha forskjellige predefinerte view.

Cellene er typisk kretsblokker hvor hver celle igjen er inndelt etter hvordan den er beskrevet. I en typisk konstruksjonsprosess vil man operere med fire forskjellige view:

schematic: Inneholder skjematikk av cellen.



Figur 2: Library browser i Cadence.

symbol: Inneholder en symbolsk representasjon av skjematikken. Dette symbolet kan igjen inkluderes i andre skjemategninger.

layout: Fysisk utlegg av cellen.

extracted: Skjematikk generert fra fysisk utlegg.

Det finnes i tillegg en rekke andre view (verilog, ahdl +++) som vi ikke kommer til å vie oppmerksomhet her.

4 Designprosessen

En typisk designprosess for en analog eller mixed-mode krets starter gjerne med at man tegner et transistor-skjema for den kretsen man ønsker å konstruere. Det er vanlig at hovedkretsen er bygget opp av mange subkretser som igjen kan bestå av subkretser. Dette kalles et hierarkisk design som gir en god og oversiktlig struktur. Når man er ferdig med å tegne skjema kan dette simuleres og eventuelt modifiseres inntil man er fornøyd med simuleringsresultatene. Neste steg i prosessen vil nå være å konstruere det fysiske utlegget med utgangspunkt i skjemaet. For å verifisere at utlegget stemmer med skjemaet har Cadence verktøy for såkalt LVS (Layout Versus Schematic). I tillegg må man verifisere at utlegget er tegnet i henhold til de gjeldende designregler for den prosessen man bruker. Dette gjøres ved å kjøre en såkalt DRC-sjekk (Design Rule Checker). Et fysisk utlegg vil alltid bestå av parasitiske komponenter som ikke finnes i opprinnelig skjema. Da disse komponentene kan ha betydelig innvirkning på hvordan kretsen fungerer, er det viktig også å simulere med utgangspunkt i fysisk utlegg. Dette gjøres ved at man kan få Cadence til å plassere parasittene fra utlegget inn i skjemaet. Deretter kan man simulere skjemaet med disse parasittene. Hvis man fremdeles er fornøyd med simuleringsresultatene er kretsen klar til produksjon. Siste steg er da å generere en såkalt gds-fil som er et standardisert format for overføring av kretsutlegg. Videre i dette heftet vil vi gå igjennom hver del av designprosessen i nødvendig detalj frem til ferdig krets.

4.1 Opprettelse av bibliotek

Før man kan begynne med design av kretsen må man opprette et bibliotek hvor man kan lagre de forskjellige deler av designet. Dette gjøres fra Library Manager (LM):

1. File → New → Library
2. Skriv inn et passende navn etter *Name* og klikk *OK*
3. Merk av for *Attach to an existing techfile* og klikk *OK*
4. Velg *TECH_XXX* hvor *XXX* er prosess og klikk *OK*

Nå skulle navnet på det nye biblioteket dukke opp under Library-kolonnen i Library Manager og man kan nå opprette nye celler under dette biblioteket.

5 Skjemategning med Composer

Composer er skjema-editoren i Cadence. Her har man mulighet for å tegne skjema og symboler samt å kjøre simuleringer. Før man kan begynne å tegne

må man opprette en ny celle med *schematic* view. Dette gjøres på følgende måte i Library Manager:

1. Venstreklikk på ønsket bibliotek.
2. File → New → Cell View...
3. Skriv inn et passende navn etter *Cell Name*
4. Velg Tool *Composer-Schematic* og klikk *OK*

Composer starter nå opp og man kan starte å tegne skjema.

5.1 Bruk av standard-komponenter

Det finnes flere biblioteker med standard-komponenter i Cadence, noen følger med selve Cadence mens andre er spesifikke for den prosessen man jobber med. Dersom man skal bruke AMS sin prosess vil man komme til å hente komponenter fra følgende bibliotek:

PRIMLIB: Inneholder transistorer, kondensatorer, motstander hvor simuleringsmodellene er i henhold til prosessparametre.

CORELIB: Inneholder ferdige digitale blokker som nand, nor, or, and +++.

A_CELLs: Inneholder ferdige analoge celler som forsterkere, oscillatorer o.l. I tillegg finnes også noen analoge pad'er her.

IOLIB_4M: Inneholder pad'er for kobling mellom pakke og krets.

IOLIB_ANA_4M: Inneholder analoge pad'er for kobling mellom pakke og krets.

analogLib Cadence standardbibliotek. Her brukes *vdd* og *gnd* symboler samt forskjellige spenningskilder. I tillegg finnes også ideelle kondensatorer og motstander her.

5.2 Hente inn komponent

For å hente inn en komponent i skjemaet gjør man følgende:

1. Klikk på button med krets på i Composer vinduet. Man kan også trykke på bokstaven *i*.
2. Klikk på *Browse* knappen i *Add Component* vinduet som kommer opp. Nytt vindu kommer opp.
3. Venstreklikk på ønsket *Library*.

4. Venstreklikk på ønsket *Cell*.
5. Venstreklikk på *symbol* under *View*.
6. Flytt musmarkøren inn i Composer-vinduet. Man vil nå få et symbol som følger markøren.
7. Man kan plassere et eller flere symboler ved venstreklikk.
8. Hvis symbolet har justerbare parametre settes disse i *Add Component*-vinduet før symbolet plasseres. Her kan man også speile og rotere symbolet.
9. Ved å endre tallet som står i *Coloumns* eller *Rows* rubrikkene, kan man plassere flere symboler i høyde eller bredde samtidig.

Husk å få med *vdd* og *gnd* i skjemaet. Hvis man senere ønsker å simulere skjemaet må man ha med de nødvendige spenningskildene:

- En DC-kilde (analogLib-vdc) må plasseres mellom *vdd* og *gnd*.
- Avhengig av krets må forskjellige kilder plasseres mellom *gnd* og de forskjellige innganger på kretsen.

5.3 Ruting av ledninger

Når de forskjellige komponentene er plassert er det klart til å rute opp ledningsnett:

1. Venstreklikk på *Wire (narrow)*-knapp for enkel ledning eller *Wire (wide)*-knapp for bus.
2. Venstreklikk på komponentterminal for startpunkt.
3. Trykk på F3. Man kan nå velge forskjellige måter å rute på.
4. Knekk på ledningen fæes ved venstreklikk.
5. Ledningen avsluttes ved venstreklikk på ny terminal eller dobbelt venstreklikk utenfor terminal.

5.4 Plassering av tilkoblingspinner

For at man senere skal kunne lage symbol som kan inkluderes i andre celler er det viktig å sette pinner på de inn- og utganger man trenger å aksessere. Hvis man kun bruker standard *vdd* og *gnd* symboler trenger man ikke pinner på disse. For andre inn- og utganger plasserer man pinner på følgende måte:

1. Venstreklikk på *Pin*-knapp. Man får nå opp *Add Pin*-vinduet.

2. Skriv inn samtlige pinnenavn adskilt med mellomrom etter *Pin Names*.
3. Før man plasserer en pinne er det viktig å passe på at retningen (Direction) er riktig.

input: Vanlige innganger (Spenning inn).

output: Vanlige utganger (Spenning ut).

input/output: Strøm inn/ut.

4. Når retning er valgt plasseres pinnen på ledning eller terminal ved å vesntreklikke.
5. Prosedyren gjentas til alle pinner er plassert.

5.5 Plassering av Sheet

For å holde orden på forskjellige celler og når de er tegnet, oppdatert o.l er det vanlig å plassere cellene i et såkalt *Sheet*. Her kan man legge til navn på designer, firmanavn, revisjon og andre notater. Dato for siste endring påføres automatisk. Generering av Sheet gjøres i Composer-Schematic på følgende måte:

1. Sheet → Edit Size ...
2. Velg størrelse som passer design. Klikk *OK*.
3. Sheet → Edit Title ...
4. Fyll inn de forskjellige rubrikkene. Klikk *OK*.

5.6 Lagring og sjekking

For å lagre en skjemategning klikker man på *Check and Save*-knappen. Cadence vil nå gå igjennom skjemaet for å sjekke at det ikke finnes tvilsomme koblinger og evt. kortslutninger. Hvis det finnes mulige feil vil disse begynne å blinke. For å få en forklaring på de forskjellige feilene gjør følgende i Composer-Schematic:

1. Check → Find Marker ...
2. Klikk på *Zoom To Markers*-knapp.
3. Klikk på *Apply*-knapp. Første feil vil nå bli zoomet inn i Composer-vinduet.
4. Klikk på *Next*-knapp for neste feil. Gjenta prosedyren til siste feil vises.

5.7 Editering på skjema

Composer har som en vanlig editor mulighet til å slette, flytte, kopiere og modifisere komponenter. Dette gjøres ved å klikke på de respektive knapper til venstre i Composer-vinduet og følge instruksjonene nederst til venstre. For å modifisere parametrene til komponenter gjør man følgende:

1. Plasser markøren i Composer-vinduet og trykk *Esc* på tastaturet. På denne måten avslutter man tidligere kommandoer slik at man f.eks. unngår å slette en komponent man bare skulle modifisere.
2. Venstreklikk på *Property*-knapp.
3. Venstreklikk på komponenten.
4. Modifiser verdier i Property-vinduet som kommer opp.
5. Venstreklikk på *Apply* eller *OK*.

5.8 Generering av symbol

Hvis man har planer om å lage et hierarkisk design hvor cellene man lager skal inngå som deler av andre celler lenger opp i hierarkiet, må man lage et såkalt *symbol-view*. Hvis man har passet på å sette pin'er på alle inn-oh utganger genereres et symbol på følgende måte fra Composer-Schematic vinduet:

1. Design → Create Cellview → From Cellview ...
2. Sjekk at data er riktig i vinduet som kommer opp. Videre skal det stå *From View Name: Schematic* og *To View Name: Symbol* Hvis dette er riktig klikk *OK*.
3. Vindu dukker opp hvor man kan spesifisere på hvilken side av et rektangulært symbol de forskjellige pinnene skal plasseres. Juster etter eget ønske og klikk *OK*.
4. Composer-Symbol starter opp og man kan nå modifisere utseendet på symbolet hvis man vil. Når man er fornøyd med symbolet klikk på *Save*-knappen.
5. Window → Close
6. Gå tilbake til Composer-Schematic vinduet og klikk på *Check and Save*
7. Composer vil nå sjekke at pinnene i skjema og symbol er like, hvis ikke dukker det opp feilmeldinger.

Nå skulle både skjema og symbol være ferdig, så neste steg i designprosessen vil være å simulere designet med *SpectreS* som er Cadence innebygde analog-simulator.