

# ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ УНИВЕРЗИТЕТ У БЕОГРАДУ

Друга лабораторијска вежба

Креирање и симулација модела на нивоу понашања



проф. др Милан Поњавић

доц. др Радивоје Ђурић

ас. мс Никола Петровић

## Увод

Циљ ове вежбе је генерисање модела компоненти на нивоу понашања. Моделовање и тестирање компоненти вршиће се у софтверу Cadence Virtuoso.

Целине које је потребно моделовати су:

1. Синусоидални напонски контролисани осцилатор (енг. Voltage Controlled oscillator VCO).
2. Памти прати коло (енг. Sample and Hold).
3. Осмобитни DA конвертор.
4. Осмобитни AD конвертор.

Потребно предзнање студената:

- Познавање рада напонско контролисаног осцилатора, памти прати кола, DA и AD конвертора.
- Основна предзнања моделовања компоненти у Verilogu A, декларација модула, дефинисање портова, инстанцирање компонента, покретање симулација у Virtuoso-у (предавања и вежбе из предмета Анализа и моделовање електронских кола употребом HDL-AMS језика).

## 1. Напонски контролирани осцилатор (VCO)

Напонско контролирани осцилатор на свом излазу има учестаност која је сразмерна улазном сигналу (који је углавном напон али може бити и струја):

$$f_{out} = K v_{IN},$$

Где је  $K$  појачање напонски контролираног осцилатора чије је јединица  $\frac{Hz}{V}$  (или  $\frac{Hz}{A}$ ) и често се означава као  $K_{VCO}$ . Најбољи начин да се моделује VCO је да се улазни сигнал интегрални у циљу добијања фазе излазног сигнала,

$$\phi(t) = 2\pi \int K v_{IN}(t) dt,$$

и онда је могуће добити излазни сигнал из фазе. Како је потребно генерисати синусоидални напонски контролирани осцилатор, напон на излазу је дат једначином

$$v_{out} = \sin \phi(t).$$

Учестаност је потребно ограничити на неку минималну  $F_{min}$  и максималну  $F_{max}$  вредност као што је потребно ограничити и амплитуду сигнала на излазу. Учестаност је онда једнака

$$f_{out} = \frac{(v_{IN} - V_{min})(F_{max} - F_{min})}{(V_{max} - V_{min}) + F_{min}}.$$

За израчунавање фазе могуће је користити специјалну функцију која се користи за моделовање VCO `idtmod(integrand, initial condition, modulus, offset, tolerance)`, где је само први аргумент обавезан док су сви остали опциони. Једначина ове функције је дата са

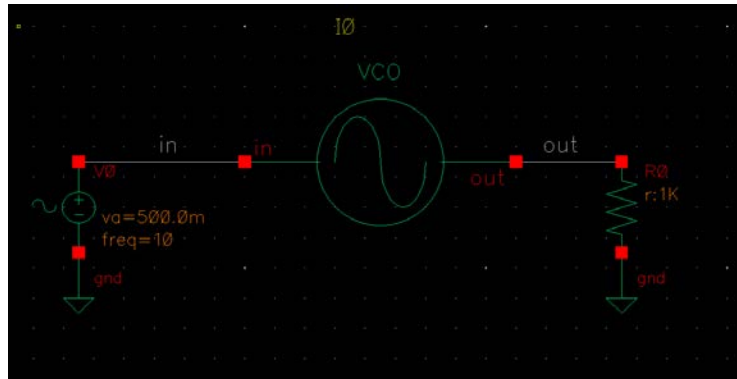
$$y(t) = \text{mod}(m) \left( \int_0^t x(\tau) d\tau + y_0 - b \right) + b$$

Где је  $x$  вредност која се интегрални,  $y_0$  је почетна вредност,  $m$  је модул и  $b$  је толеранција.

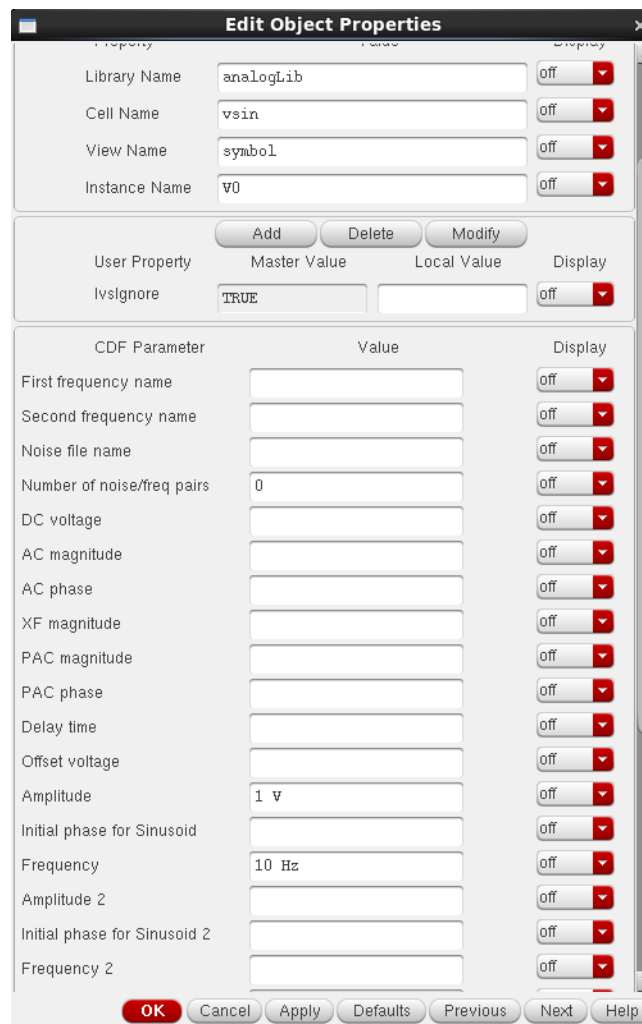
Помоћу функције `$bound_step()` могуће је ограничити максимални временски корак симулације.

У циљу провере функционалности моделованог напонско-контролираног осцилатора потребно је направити шему као што је то приказано на слици 1.1. Подешавање напонског генератора дато је на слици 1.2. Након тога је потребно покренути ADE L и подесити *transient* анализу као што је то приказано на слици 1.3. Након

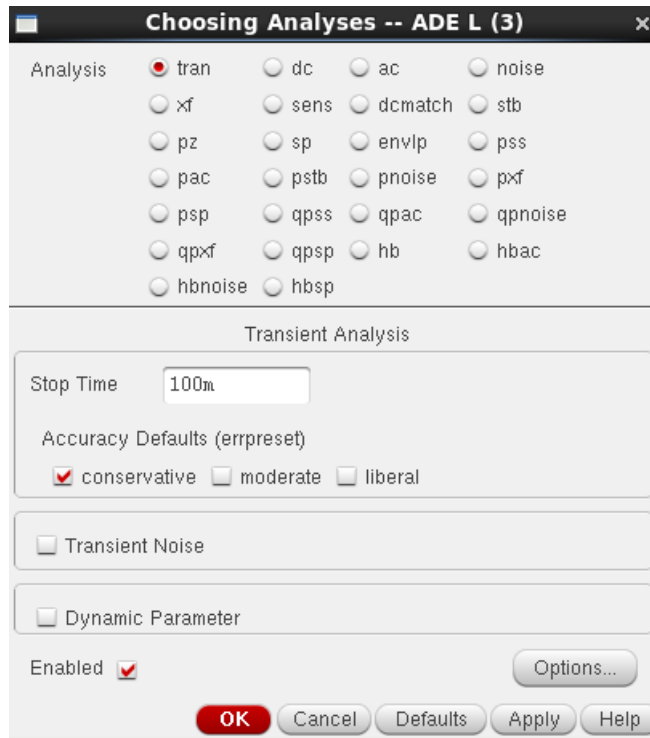
покретања симулације потребно је посматрати напоне на улазу и излазу кола. Изглед ADE L прозора је дат на слици 1.4 док су дијаграми напона на улазу и излазу дати на слици 1.5.



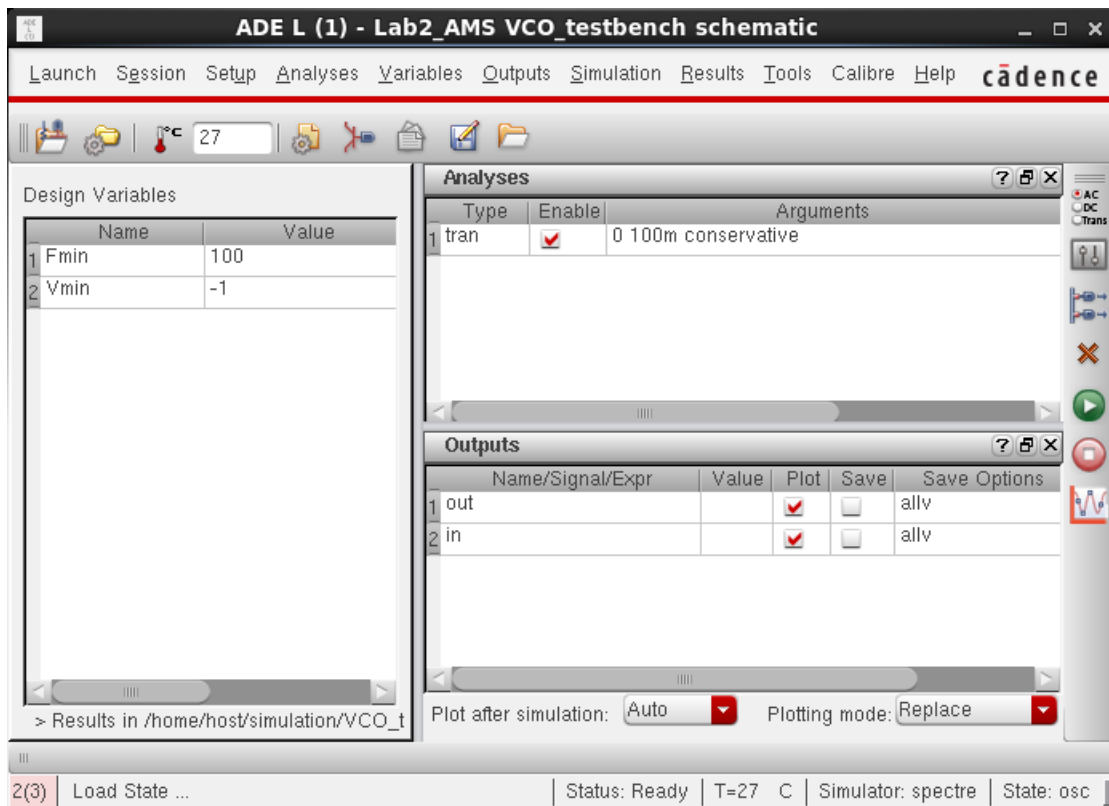
Слика 1.1. Шема за проверу функционалност моделованог напонско-контролисаног осцилатора.



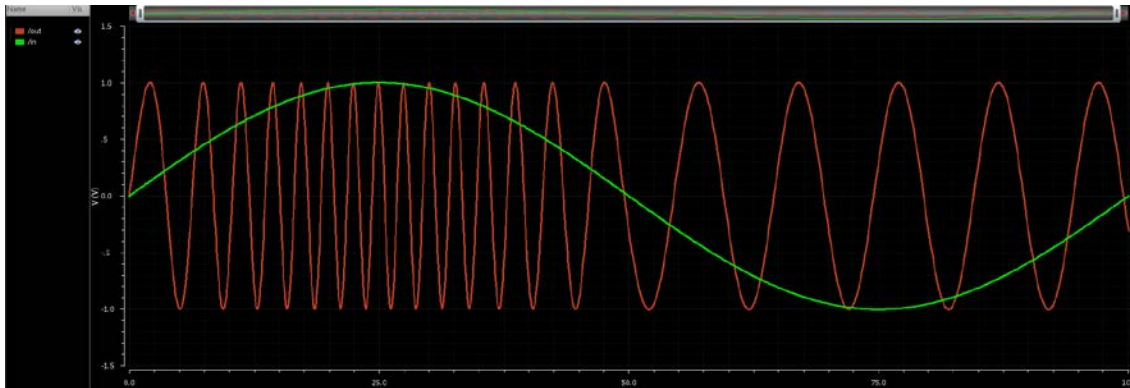
Слика 1.2. Подешавање vsin генератора.



Слика 1.3. Подешавање transient анализе.



Слика 1.4. Изглед ADE L прозора.



Слика 1.5. Временски дијаграм улазног и излазног сигнала осцилатора.

## 2. Памти прати коло

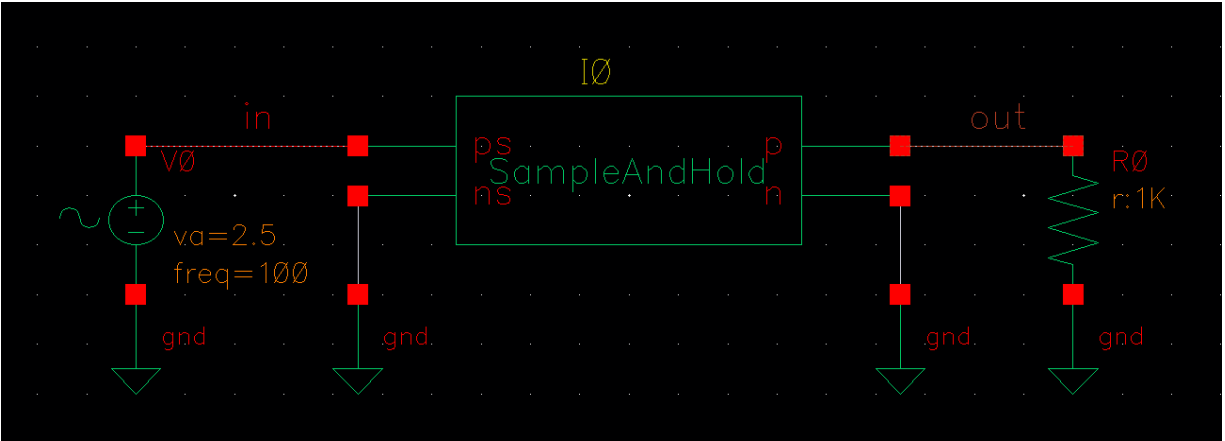
Излаз памти прати кола има вредност последњег одбирка улазног сигнала,

$$v_{OUT}(t) = v_{IN}(nT)$$

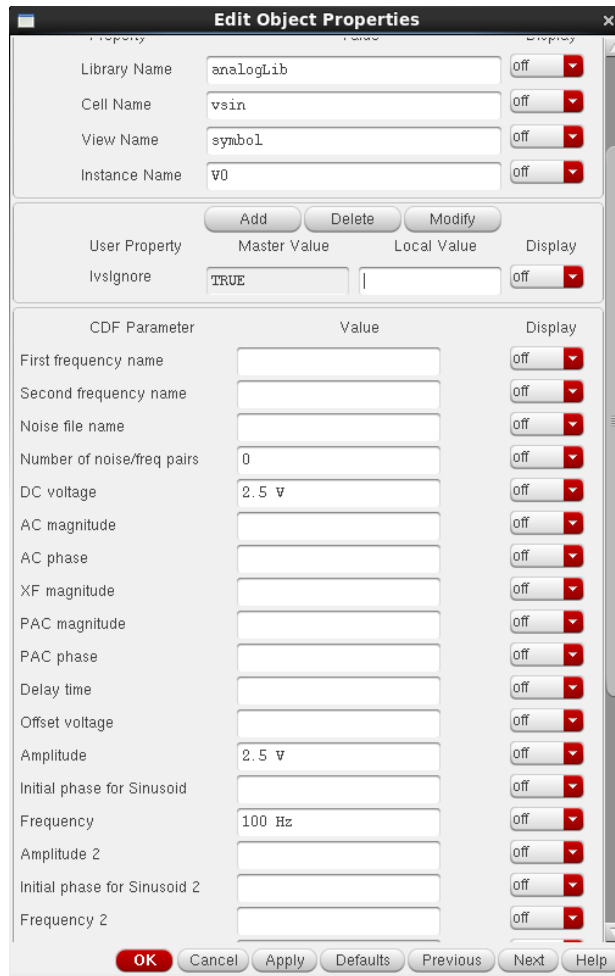
где је  $T$  период одабирања а  $n$  је највећи цео број мањи од  $t/T$ . Одабирање је пример догађаја и моделоваћемо га као тренутни догађај.

Одабирање се дешава или на период одабирања или на иницијални корак, ово је могуће реализовати помоћу `@(timer(toff, period) or initial_step)` где је `toff` офсет. Функција `timer` генерише прекиде на сваки умножак другог аргумента у односу на први аргумент функције. Параметри `toff` и `period` треба да узимају вредности од нуле до бесконачно. Такође, због прекида у излазном сигналу, потребно је додати `$discontinuity(0)` иако ово није обавезно урадити.

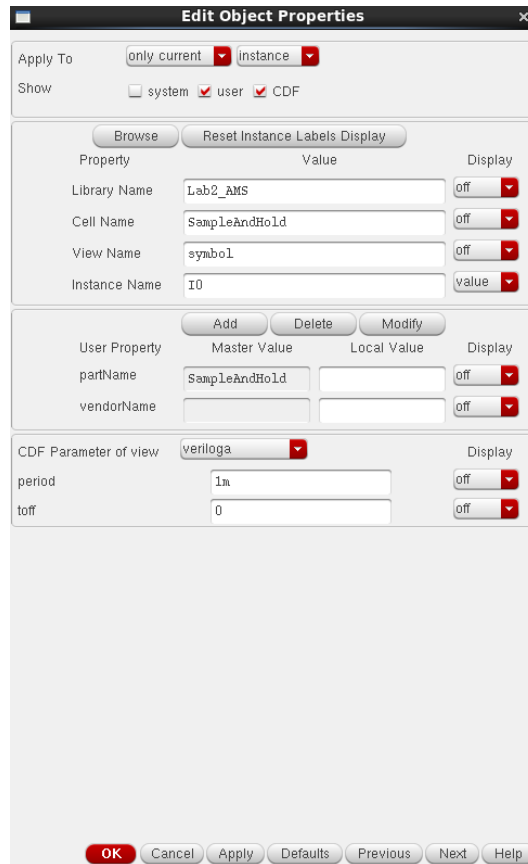
Коло за проверу функционалности памти прати кола дато је на слици 2.1. На слици 2.2. је дато подешавање генератора напона док је подешавање памти и прати кола је дато на слици 2.3. Потребно је покренути ADE L и подесити transient анализу као што је то приказано на слици 2.4. Након покретања симулације потребно је селектовати напон на улазу и излазу кола. Изглед ADE L прозора приказан је на слици 2.5. Дијаграми напона су приказани на слици 2.6.



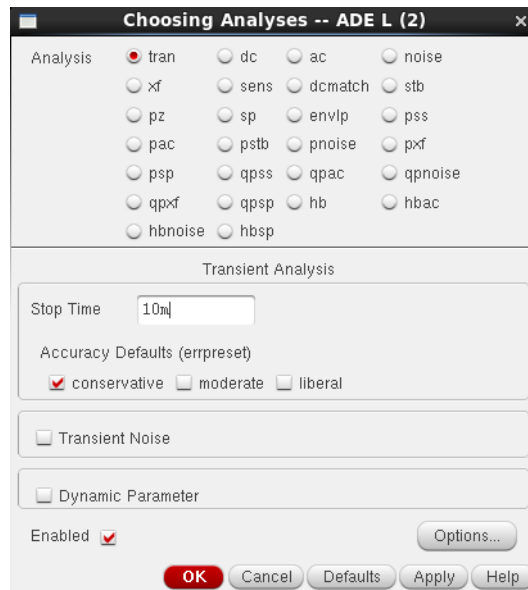
Слика 2.1. Шема за проверу функционалности памти прати кола.



Слика 2.2. Подешавање vsin генератора.

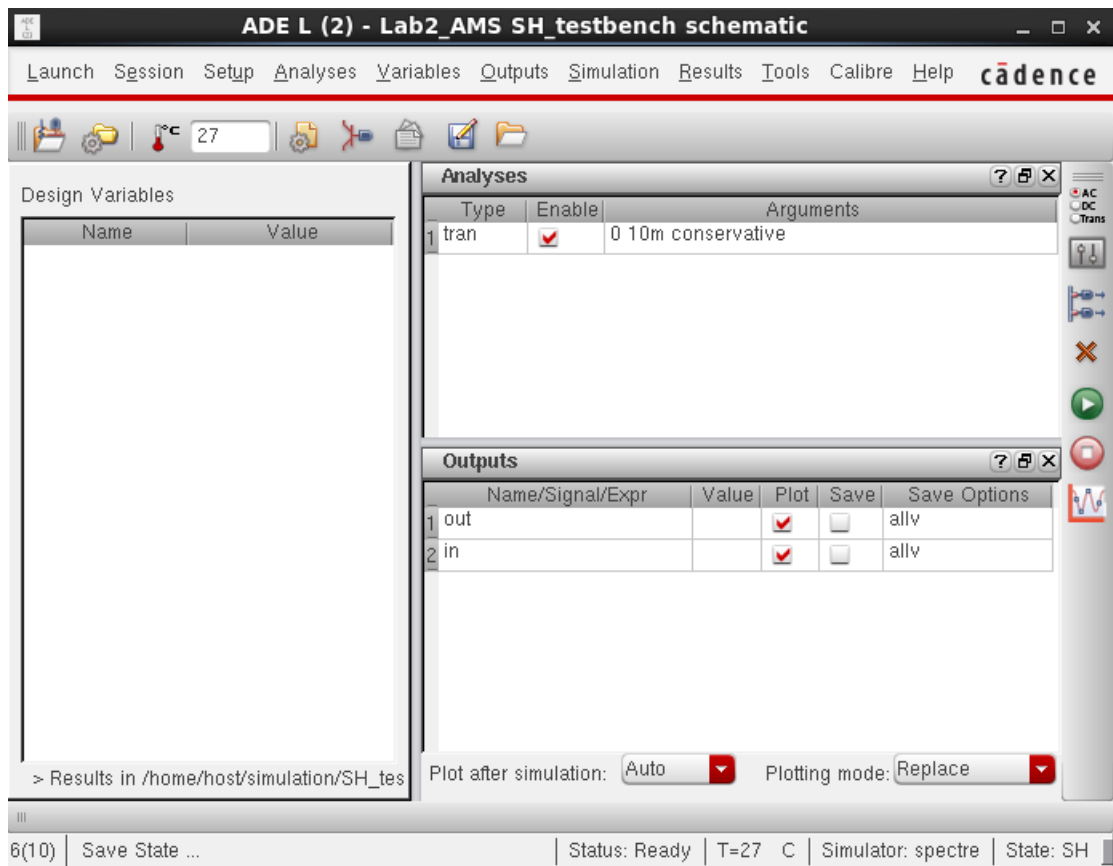


Слика 2.3. Подешавање памти прати кола.

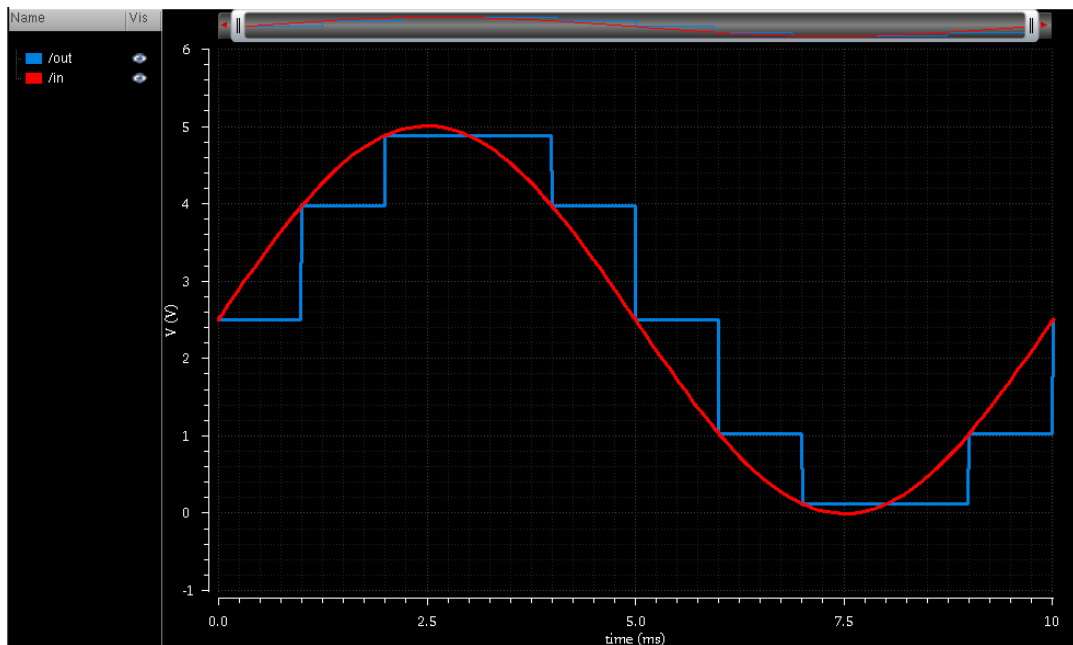


Слика 2.4. Подешавање transient анализе.





Слика 2.6. Изглед ADE L прозора након подешавања transient анализе.



Слика 2.6. Дијаграми напона на улазу и излазу прати памти кола.

### 3. Особитни DA и AD конвертори

У циљу реализације AD конвертора потребно је декларисати излаз као вектор помоћу

```
output [0:7] out;  
voltage [0:7] out;
```

где прва линија декларише порт а друга линија додељује дисциплину порту.

У циљу добијања напона на излазу потребно је одабирати напон на улазу које се врши у иницијалном тренутку или када сигнал такта пређе неку задату вредност. Ово је могуће одрадiti помоћу

```
@(cross(V(clk)–thresh, +1) or initial_step)
```

Претварање аналогног сигнала у дигитални је могуће одрадiti помоћу **for** петље дате са

```
for (i = bits – 1; i >= 0; i = i – 1) begin  
    ...  
end
```

где је променљива *i* дефинисана помоћу **genvar** *i*. У циљу додељивања вредности потребном сигналу потребно је користити функцију **transition** на следећи начин:

```
for (i = 0; i < bits; i = i + 1) begin  
    V(out[i]) <+ transition(result[i], delay_from_clk_edge_to_out, transition_time);  
end
```

где result[i] садржи *i*-ти бит.

Пре реализацији DA конвертора потребно је улазни особитни дигитални сигнал претворити у аналогни сигнал. Стога је потребно дефинисати улазни сигнал као вектор

```
input [0:7] in;
```

Као и у случају AD конвертора, сигнал на улазу се одабира помоћу

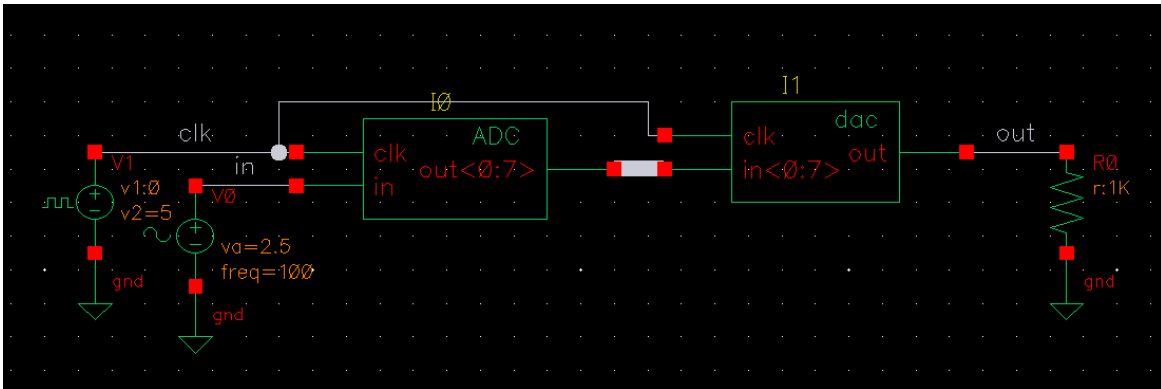
```
@(cross(V(clk)–thresh, dir) or initial_step) begin  
    ...  
end
```

где dir означава ивицу на коју се врши одабирање. За генерисање излазног сигнала, као и у претходном случају, потребно је користити

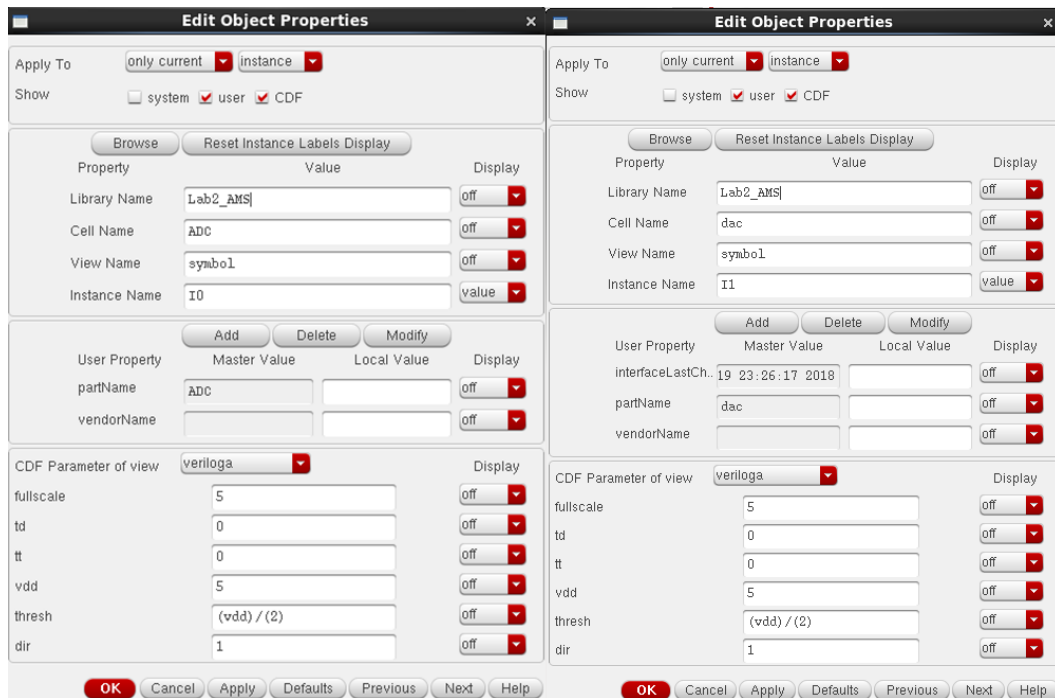
$V(\text{out}) <+ \text{transition}(\text{aout}, \text{delay\_from\_clk\_edge\_to\_out}, \text{transition\_time});$

где је аout аналогни сигнал који је потребно преписати у V(out).

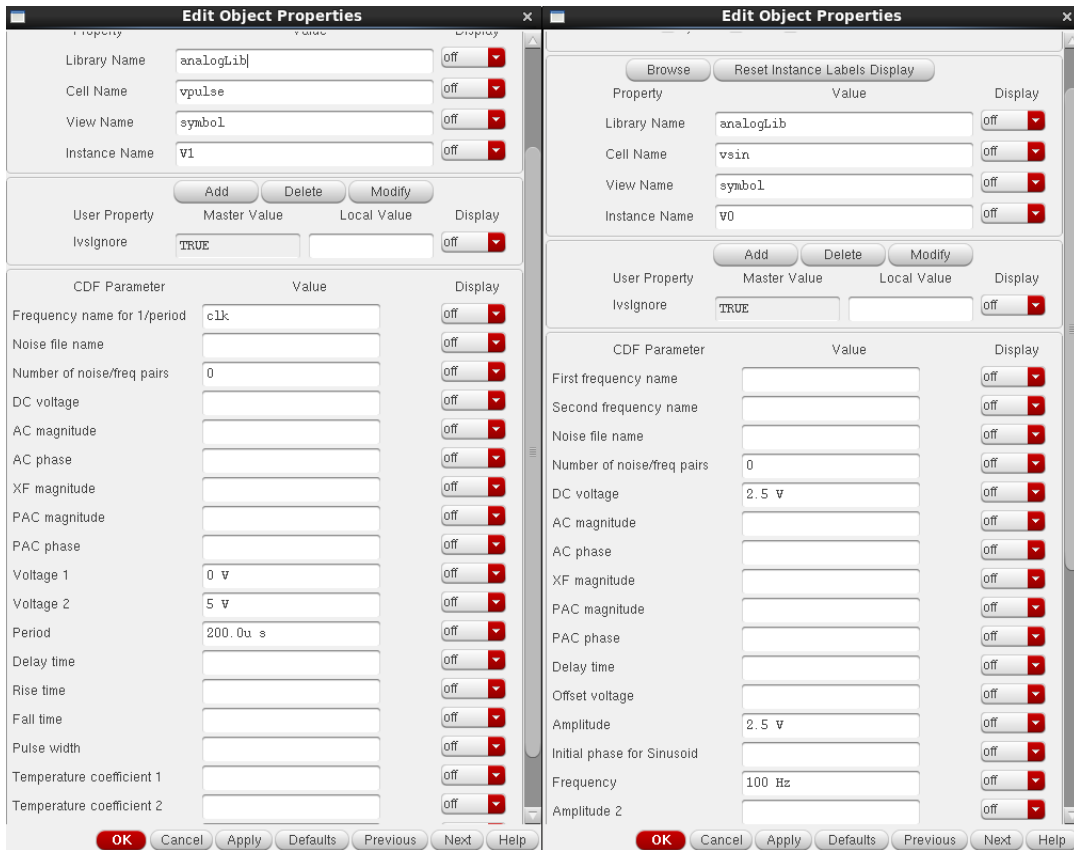
Шема за проверу функционалности кола AD и DA конвертора дата је на слици 3.1. Подешавање AD и DA конвертора дато на слици 3.2. Подешавање генератора четвртки и синусоидалног генератора је дато на слици 3.3. На слици 3.4. дато је подешавање transient анализе. На слици 3.5. је дат приказ ADE L прозора. На слици 3.6. дат је приказ напона на излазу AD конвертора док је на слици 3.7. дат приказ сигнала такта, улазног сигнала AD конвертора и излазног сигнала DA конвертора.



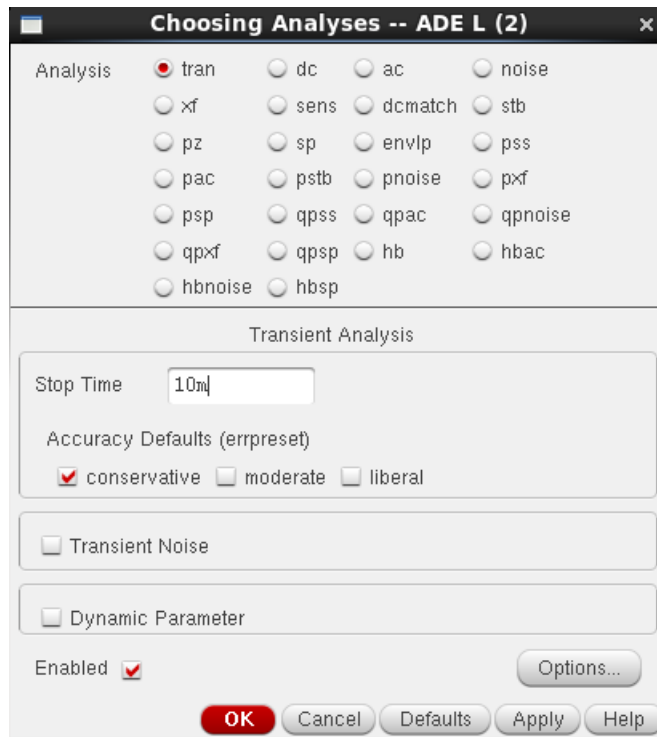
Слика 3.1. Шема за проверу функционалности AD и DA конвертора.



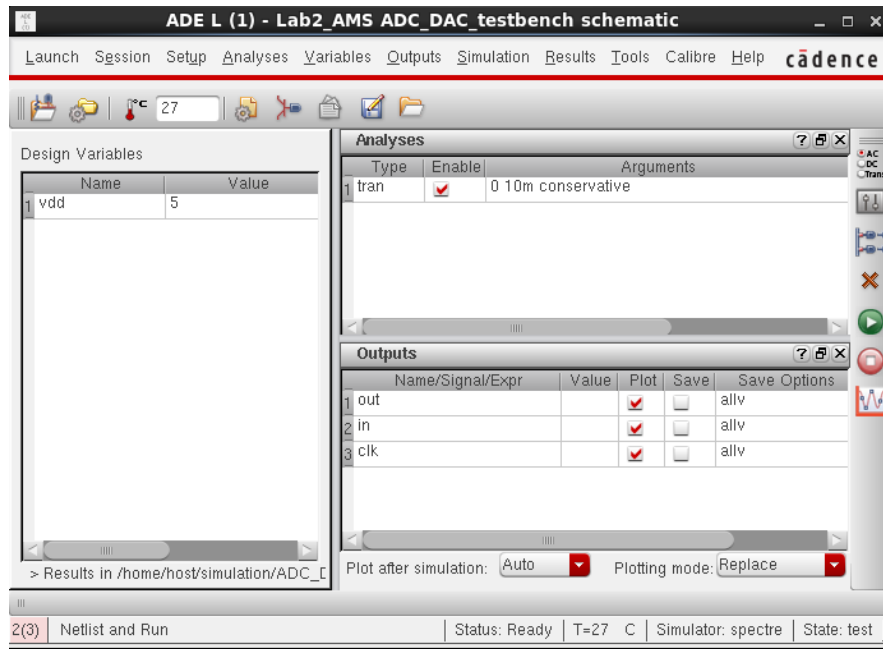
Слика 3.2. Подешавање AD и DA конвертора.



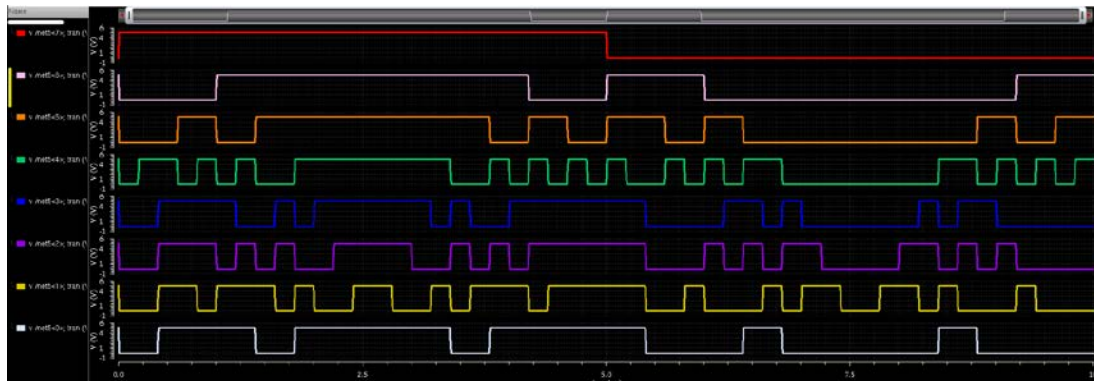
Слика 3.3. Подешавање генератора четвртки и генератора vsin.



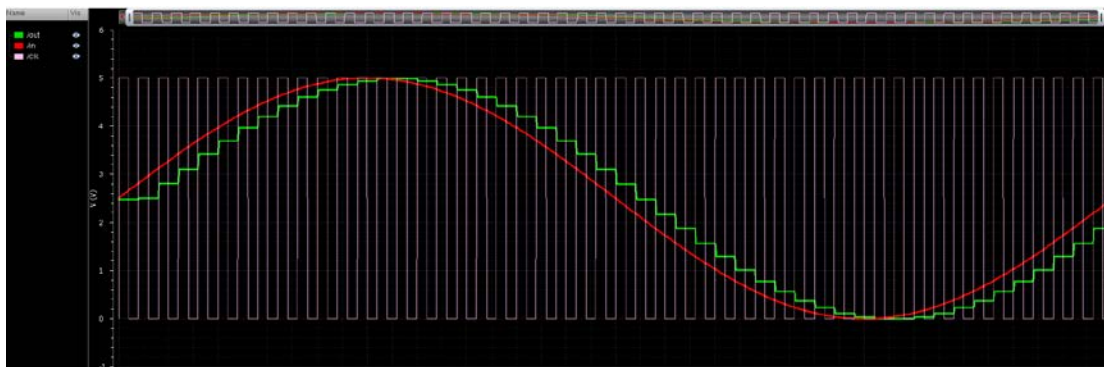
Слика 3.4. Подешавање transient анализе.



Слика 3.5. Изглед ADE L прозора након подешавања transient анализе.



Слика 3.6. Сигнал на излазу AD конвертора.



Слика 3.7. Сигнал на улазу у AD конвертор, сигнал на излазу DA и сигнал такта.