Упражнение № 3

Проектиране и симулация на цифрови схеми

Изображения за протокола:
- електрическа схема на полусуматора;
- електрическа схема на 1-битовия пълен суматор;
- електрическа схема на 3-битовия пълен суматор;
- файл с входни въздействия - в протокола се качва като текст, а не като изображение!
- резултати от симулацията.

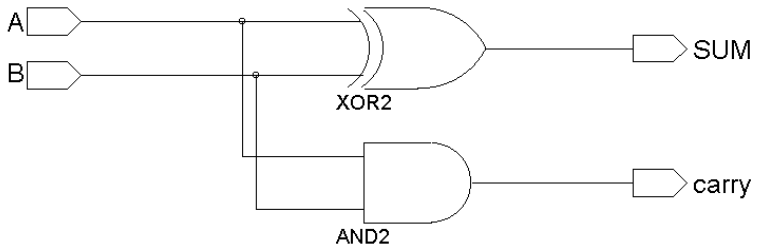
I. Изчертаване на схема.

Целта на упражнението е да се проектира 3-битов суматор с входен и изходен пренос. 3-битовият суматор трябва да се състави от отделни 1-битови суматори, всеки от които от своя страна е изграден от полусуматори.

ВНИМАНИЕ!!! Не използвайте тире в имената на клетките. Можете да използвате само букви, цифри и долна черта.

Използвани схеми

Показаните схеми на полусуматор (фиг. 1) и суматор (фиг. 2) са примерни, като не е задължително да се изградят по същия начин.

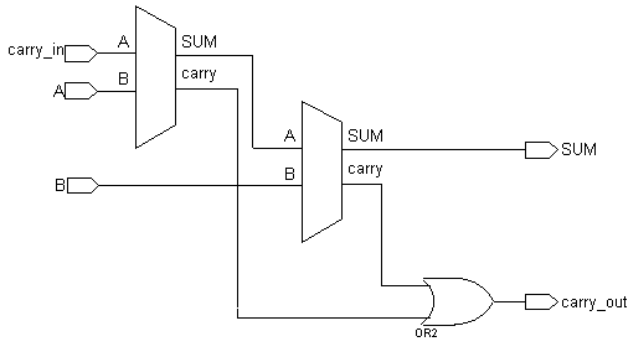


Фиг. 1. Схема на полусуматор.

За създаването на схемата на полусуматора да се използват логически елементи от библиотеката **CORELIBD - XOR2X1** (2-входов елемент изключващо ИЛИ)**, NAND2X1** (2-входов елемент И) и **INVX1** (инвертор), след което да се направи нейно представянето във вид на символ (**symbol**). За пълния 1-битов суматор може да се използва **OR2X1** (2-входов елемент ИЛИ).

Тъй като проектът има йерархична структура, трябва пълния едно-битов суматор също да се изгради като електрическа схема, съставена от полусуматори и символ. Всеки от входовете и изходите на схемите трябва да се свърже към входен/изходен порт (**pin**). Поставянето им се

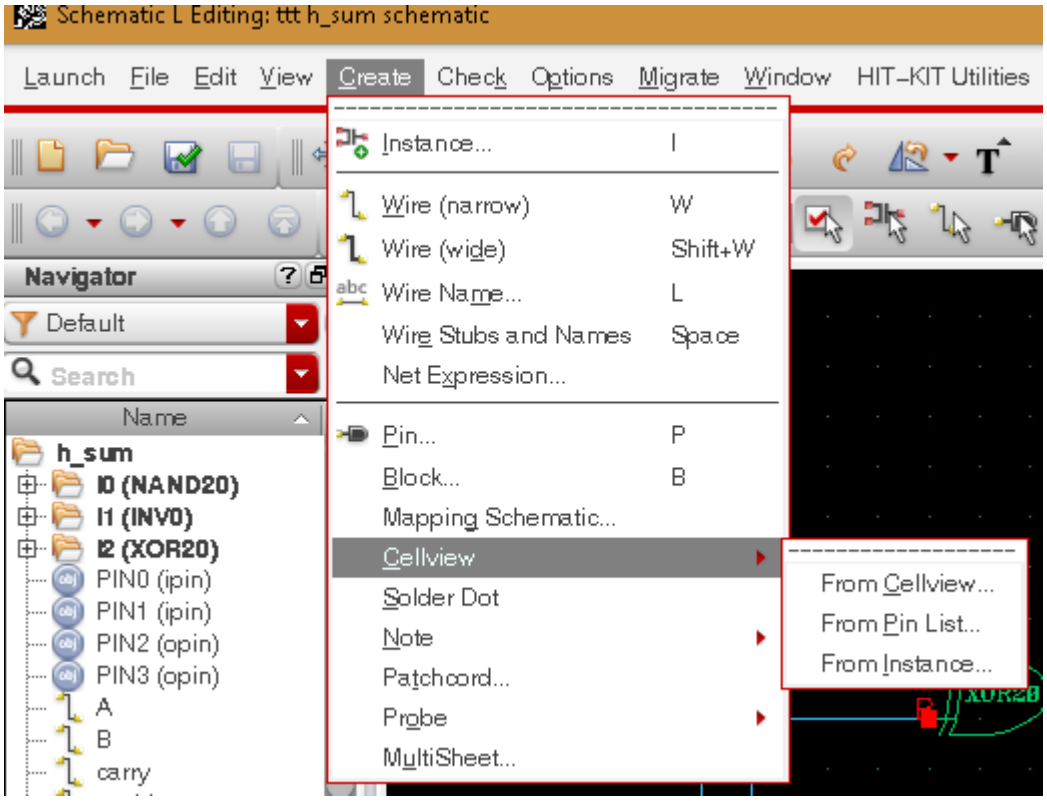
извършва с **Create => Pin**, като се внимава кои са входни (**input**) и кои - изходни (**output**).



Фиг. 2. Схема на пълен 1-битов суматор.

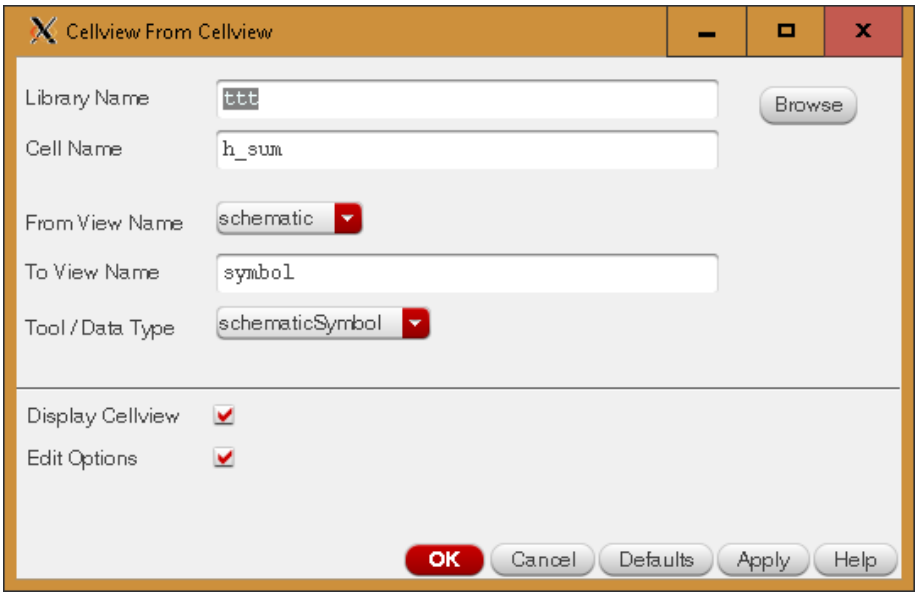
1. Създаване на символ.

След като е направена съответната схема, сложени са съответните входно/изходни портове и е направена проверка за грешки, може да се пристъпи към създаването на символ. За целта се извиква командата: **Create ⇒ Cellview ⇒ From Cellview** (фиг. 3).



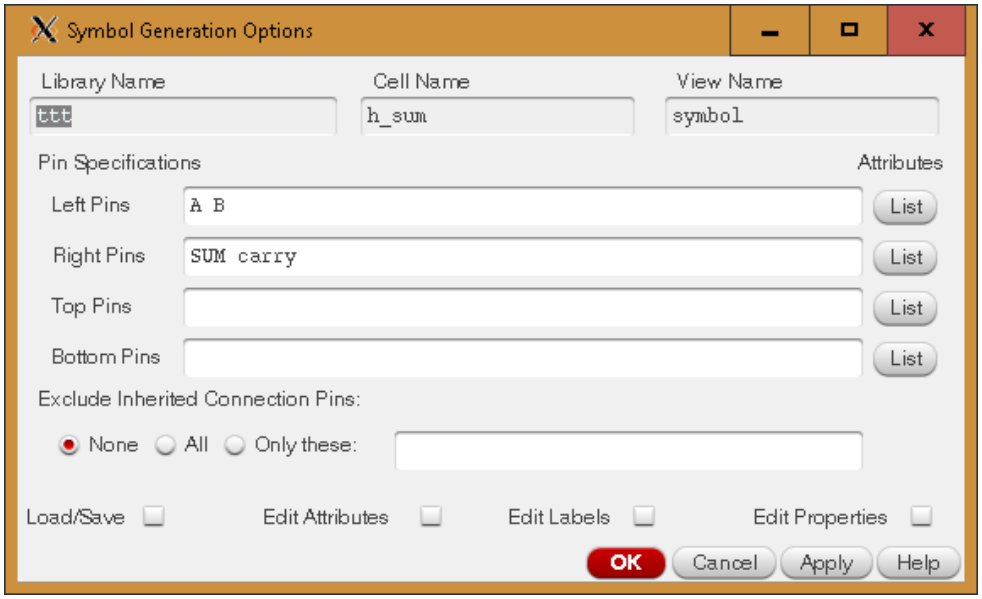
Фиг. 3. Създаване на символ.

Появява се прозорец, в който се указва новото представяне, което ще се генерира. В случая то е **symbol** (фиг. 4). Имената на клетката и библиотеката се запазват. Натиска се бутонът **OK.**



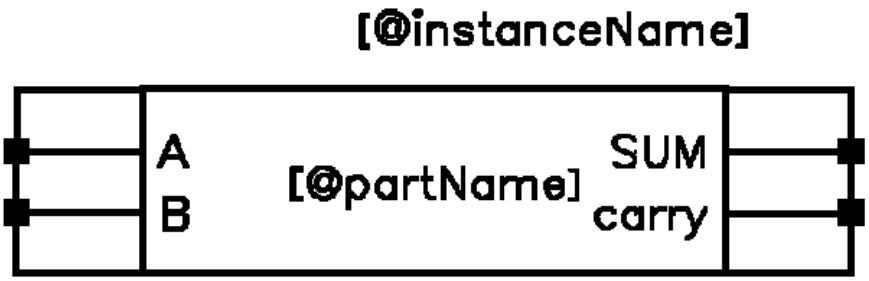
Фиг. 4. Указване на име за новата клетка и библиотеката, към която тя ще принадлежи.

Появява се прозорецът от фиг. 5, в който може да се укажат, имената, разположението на входно-изходните пинове и др. Формата е попълнена с необходимата информация и за това само се натиска **OK**.



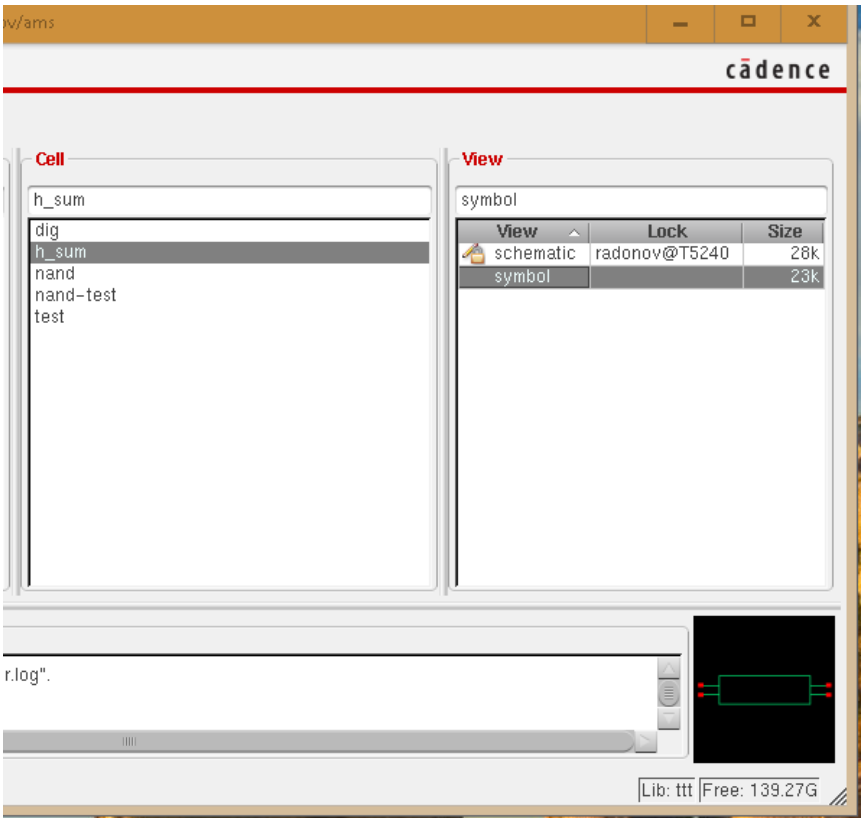
Фиг. 5. Задаване на параметрите на новата клетка.

Появява се нов прозорец, в който е показан видът на символа и неговите входно-изходни пинове (фиг. 6).



Фиг. 6. Новосъзданеният символ.

В горната част на прозореца има поле с бутони, чрез които могат да се извикат разнообразни функции за редактиране, така че символът може да се оформи по желания начин. В случая не е задължително той да има формата на трапец (вж. фиг. 2). След като редактирането на символа е приключило, запазват се направените промени и се затваря прозорецът. Новият символ се появява в библиотеката (фиг. 7).

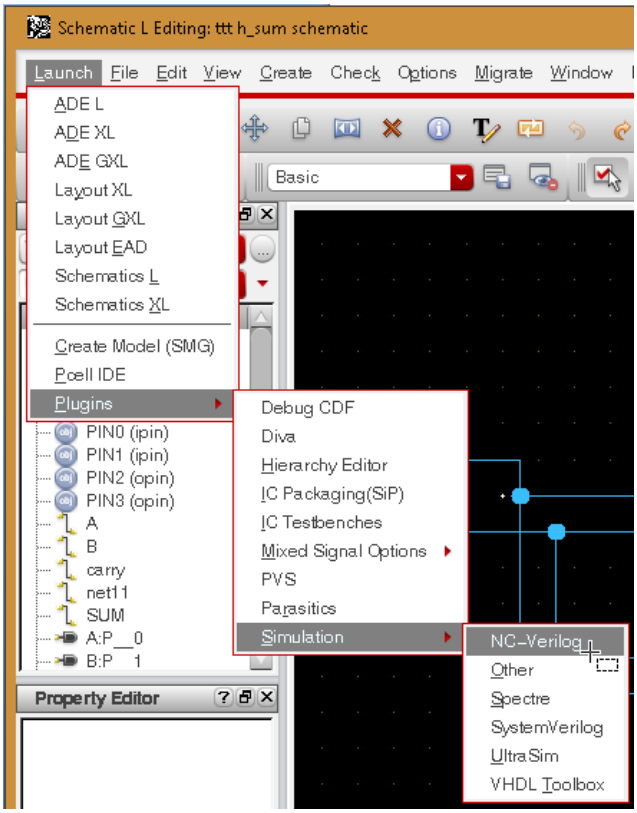


Фиг. 7. Библиотеката с новия символ

По аналогичен начин се прави символно представяне на схемата на 1-битовия суматор, което после се използва в схемата за суматор на две 3-битови числа. Последната схема трябва да я съставите Вие!

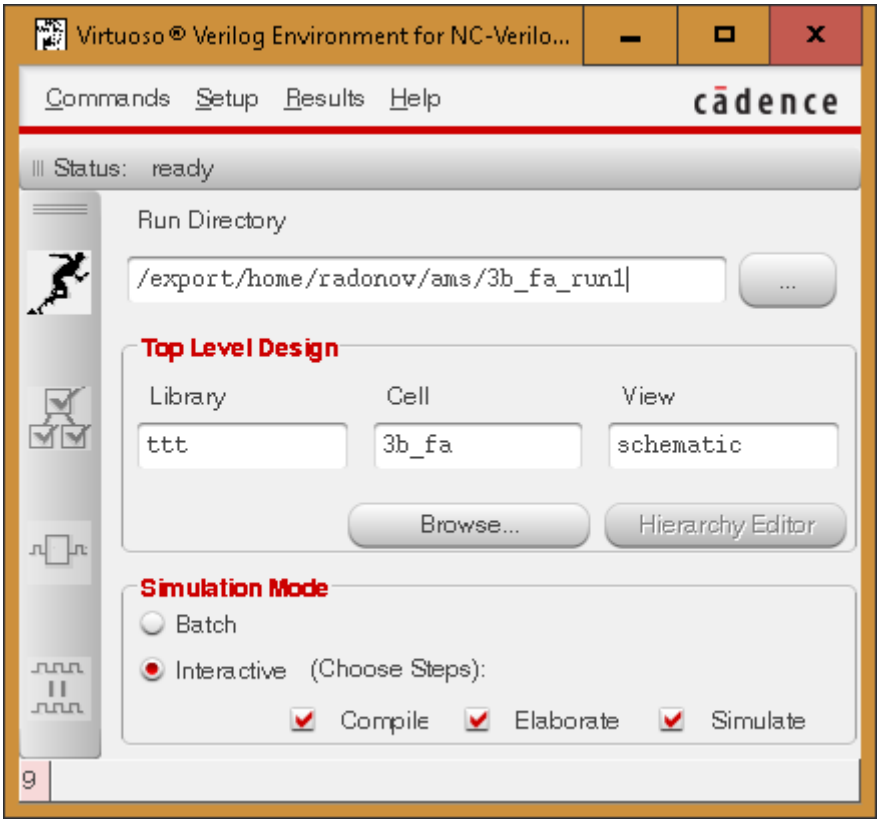
II. Симулация на схемата.

След като схемата е създадена е необходимо тя да се симулира, за да се провери дали функционира правилно. За целта се използва цифров симулатор. Той се извиква от менюто **Launch** ⇒ **Plugins** ⇒ **Simulations** ⇒ **NC-Verilog** (фиг. 8).



Фиг. 8. Стартиране на цифровия симулатор

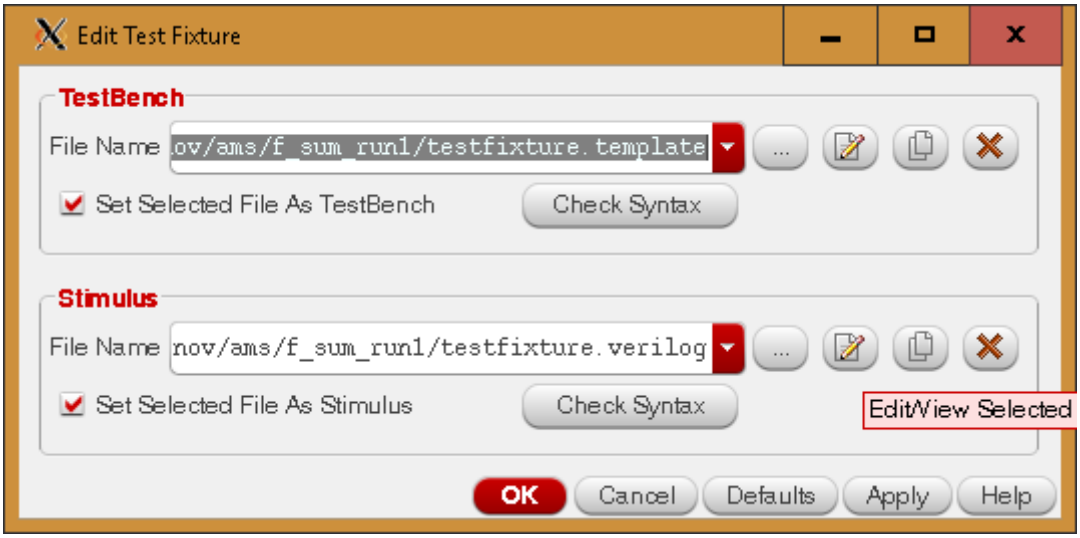
Появява се прозорецът, показан на фиг. 9, в който се уточнява името на директорията, в която ще се извършва симулацията и др.



Фиг. 9. Настройване на симулатора.

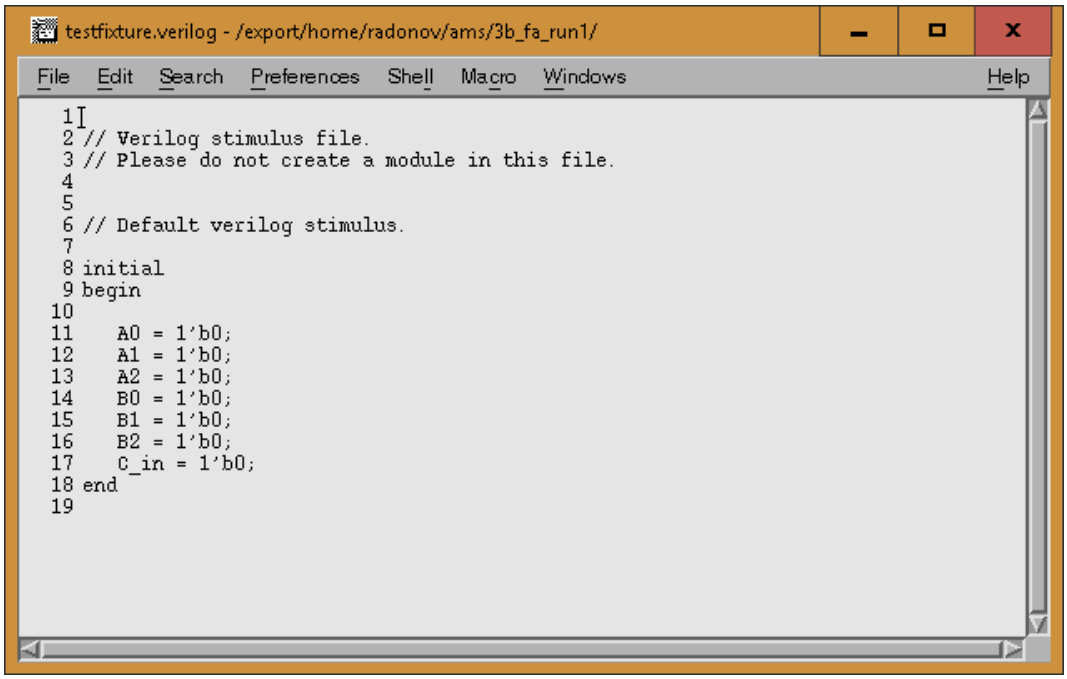
Началото на симулационния процес започва с инициализацията на проекта. За целта се избира **Commands ⇒ Initialize Design**. Преди това останалите команди от менюто са неактивни.

Следва генериране на нетлиста - **Commands ⇒ Generate Netlist** и въвеждане на входните въздействия - **Commands ⇒ Edit Test Fixture (**фиг. 10).



Фиг. 10. Задаване на входните въздействия**.**

За редактиране на файла `testfixture.verilog` се натиска бутона  в частта **Stimulus**. Появява се текстов редактор с частично попълнен код на езика **Verilog** - фиг. 11.



Фиг. 11. Текстов редактор с частично попълнен код.

Въвежда се показания по-долу код.

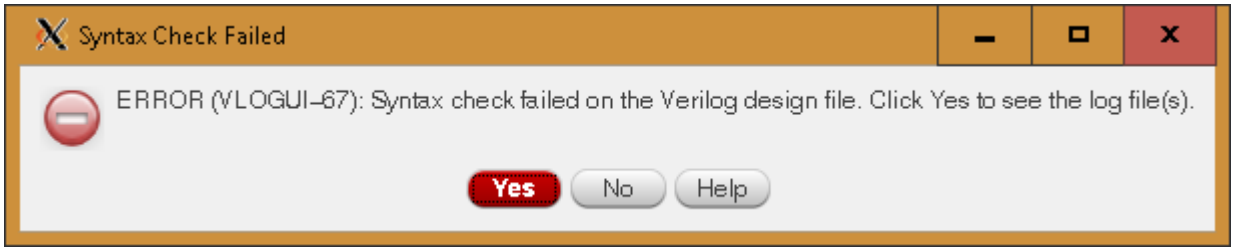
Примерен файл с входни въздействия

```
// Verilog stimulus file.
// Please do not create a module in this file.


// Default verilog stimulus.

initial
begin

   A0 = 1'b0;
   A1 = 1'b0;
   A2 = 1'b0;
   B0 = 1'b0;
   B1 = 1'b0;
   B2 = 1'b0;
   C_in = 1'b0;
end

initial #Y $finish;

 always
  begin
```

```
    #X C_in = 1'b1;
    #X C_in = 1'b0;
  end

  always
   begin
    #X A0 = 1'b1;
    #X A1 = 1'b1;
    #X A2 = 1'b1;
    #X A0 = 1'b0;
    #X A1 = 1'b0;
    #X A2 = 1'b0;
  end
```

Имената на входовете във файла (A0, ... B2, C_in) трябва да съответстват на вашите означения. На редовете, където се среща символа **X,** заменете **X** с числото, получено от последните 2 значещи цифри на факултетния Ви номер, умножено по 1000. Напр. за 101315**01** - 1000, за 101315**26** - 26000. Заменете **Y** с произведението на така полученото число и 10.

След описване на необходимите тестови вектори, се затваря текстовият редактор и се прави проверка на синтаксиса чрез натискане на бутона **Check Syntax**. При наличие на грешки излиза прозореца, показан на фиг. 12.1. Ако се натисне **Yes** се показва втори прозорец, в който е показана конкретната грешка - фиг. 13. При липса на грешка излиза съобщението, показано на фиг 12.2.

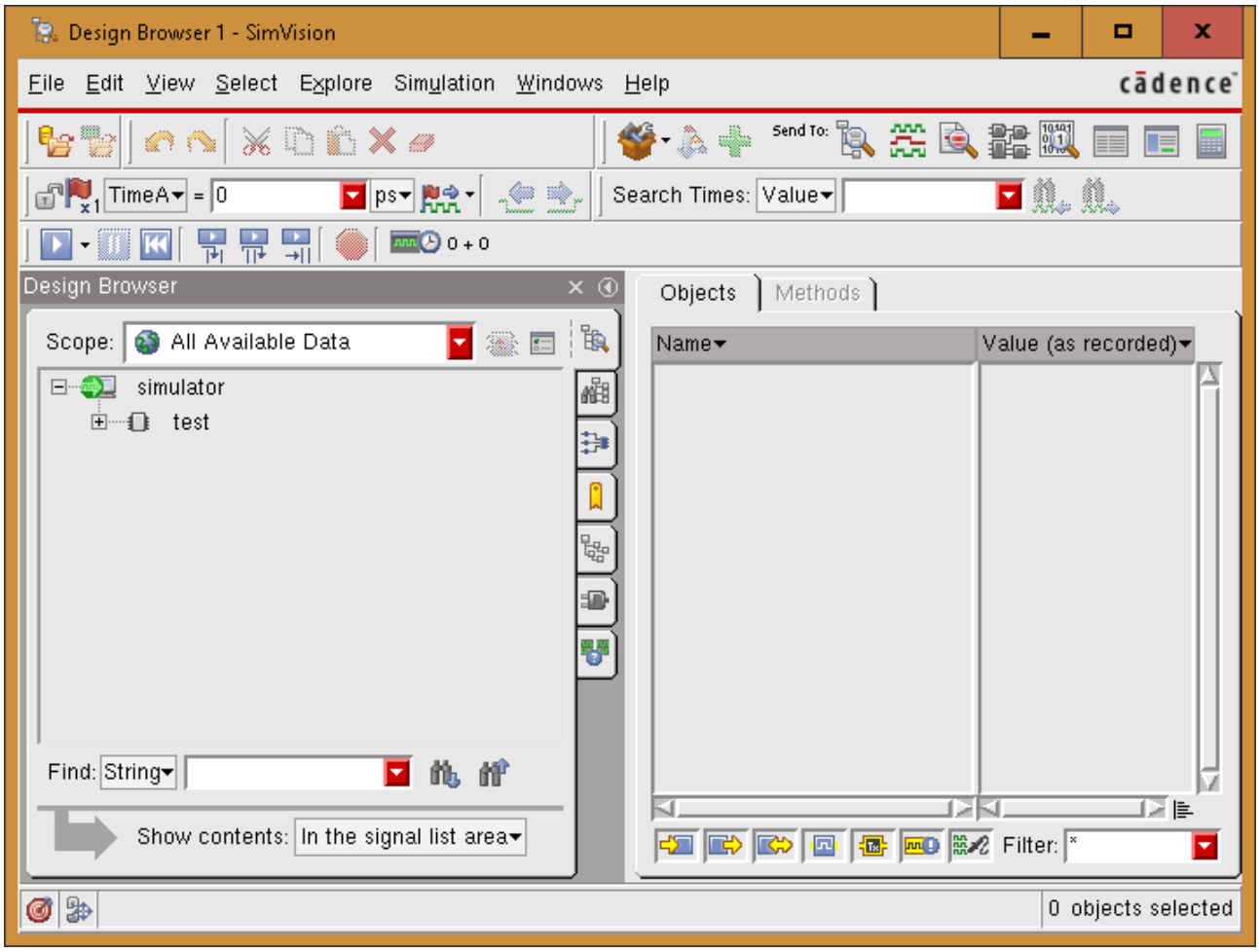


Фиг. 12.1. Съобщение за наличие грешка.

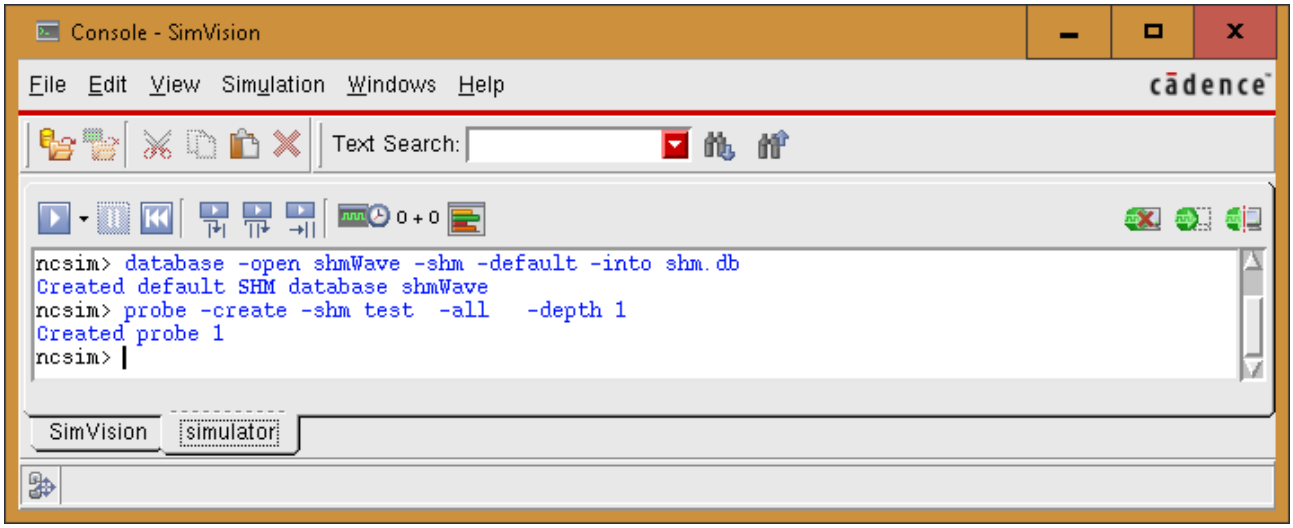
Фиг. 12.2. Съобщение липса за грешка.

Фиг. 13. Прозорец, показващ грешката във Verilog кода**.**

Стартира се симулацията от **Commands** ⇒ **Simulate**. След кратко време се показват два прозореца - **Design Browser 1 - SimVision** (фиг.14.1) и **Console - SimVision** (фиг.14.2).



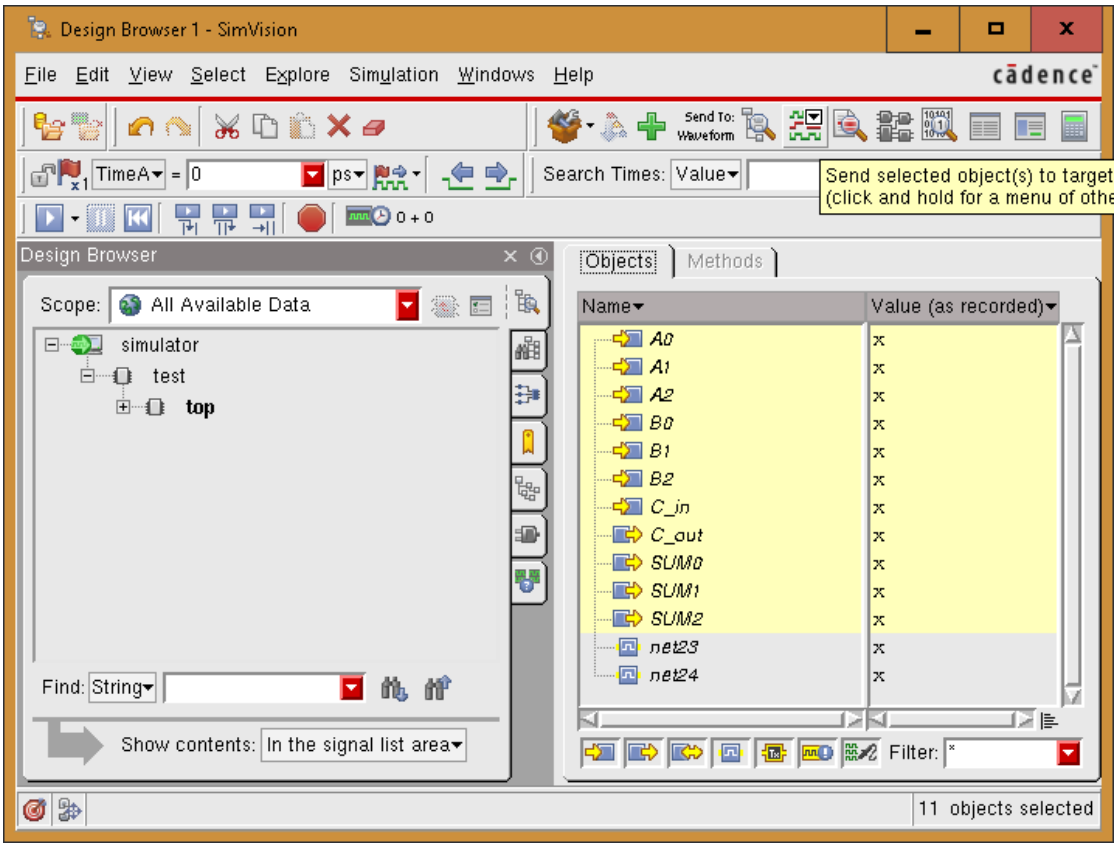
Фиг. 14.1. Design Browser 1 - SimVision.



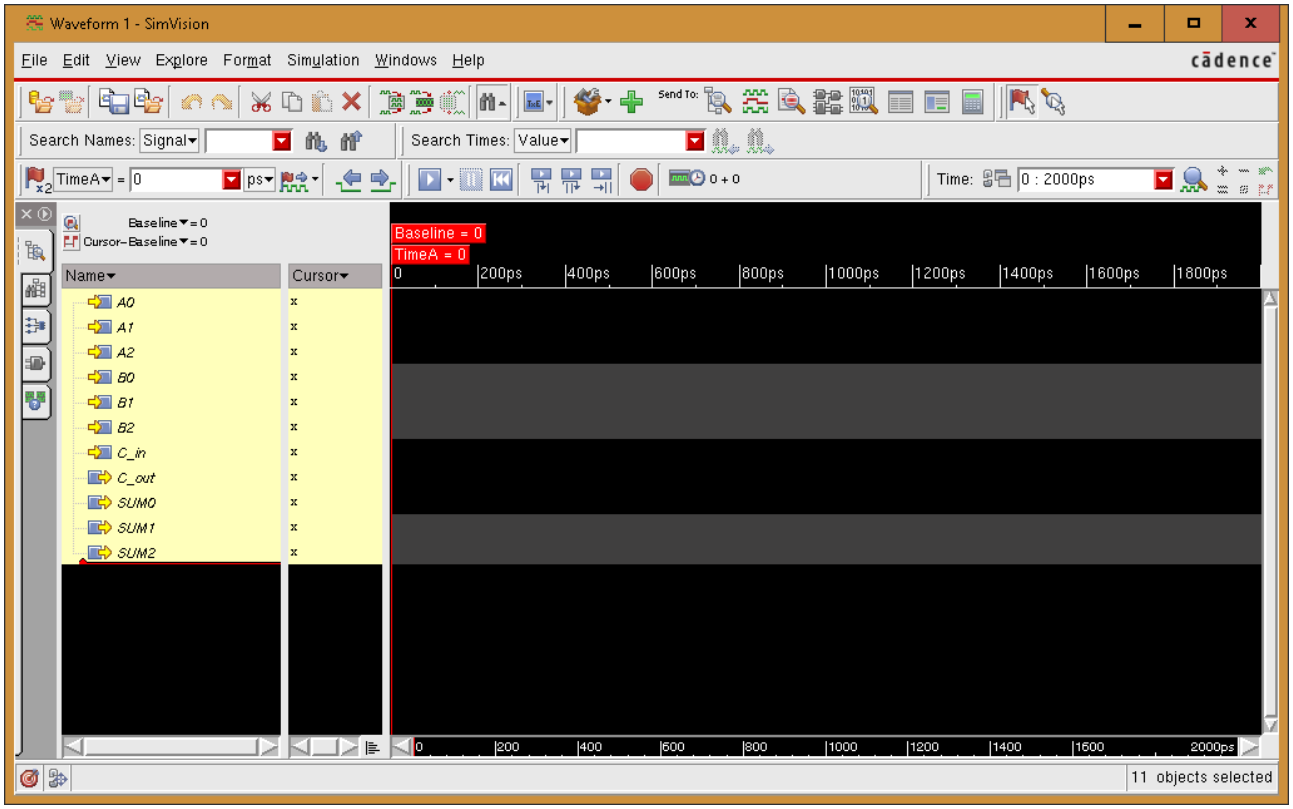
Фиг. 14.2. Console - SimVision.

Тъй като **Simulation Mode** е **Interactive** (фиг. 9), симулацията може да се управлява и от двата прозореца - графично от Design Browser и текстово, с команди от Console - SimVision.

Избират се сигналите, които искате да се визуализират след края на симулацията и се натиска бутона  - фиг. 15. Появява се визуализатора на сигнали **Waveform 1 - SimVision** - фиг. 16.

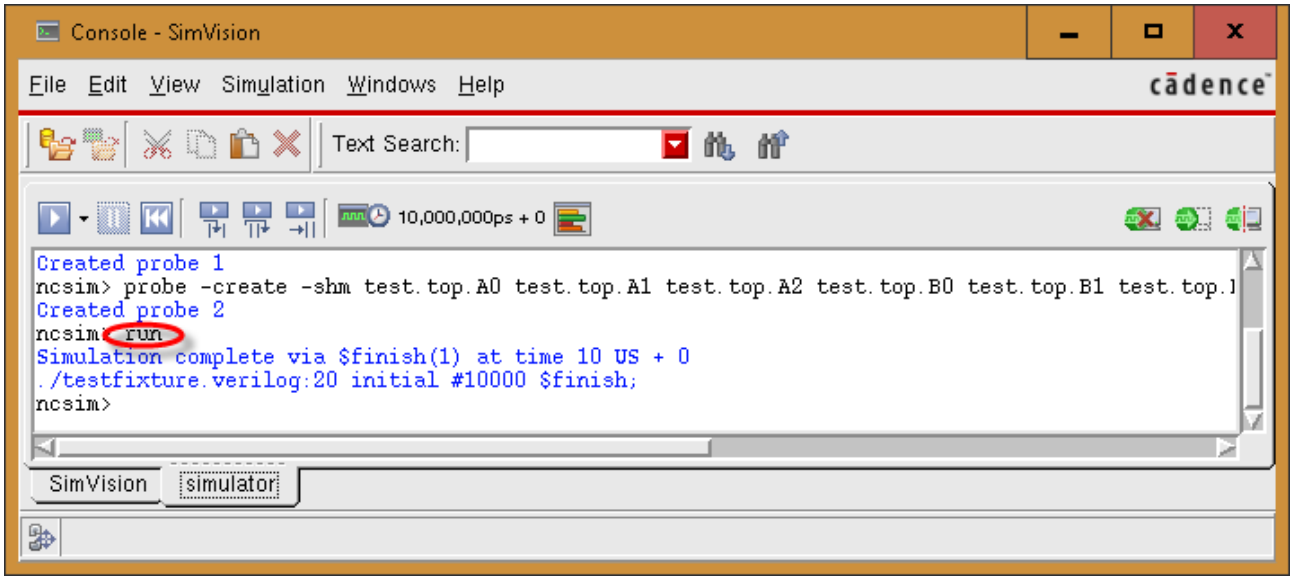


Фиг. 15. Избор на сигнали за визуализиране.



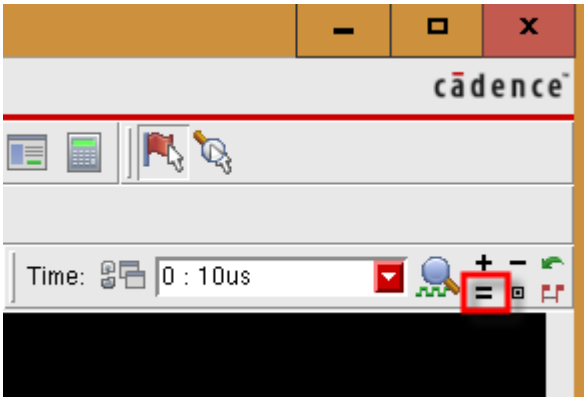
Фиг. 16. Визуализатор на сигнали.

Натиска се бутона  за начало на симулацията в прозореца **Design Browser**. Това е еквивалентно на изпълнение на командата **run** в прозореца **Console** - фиг. 17.



Фиг. 17. Прозорец Console с команда run.

За да се види целия диапазон на симулацията се натиска бутона със символ "=" (равно), ограден в червен правоъгълник на фиг. 18.



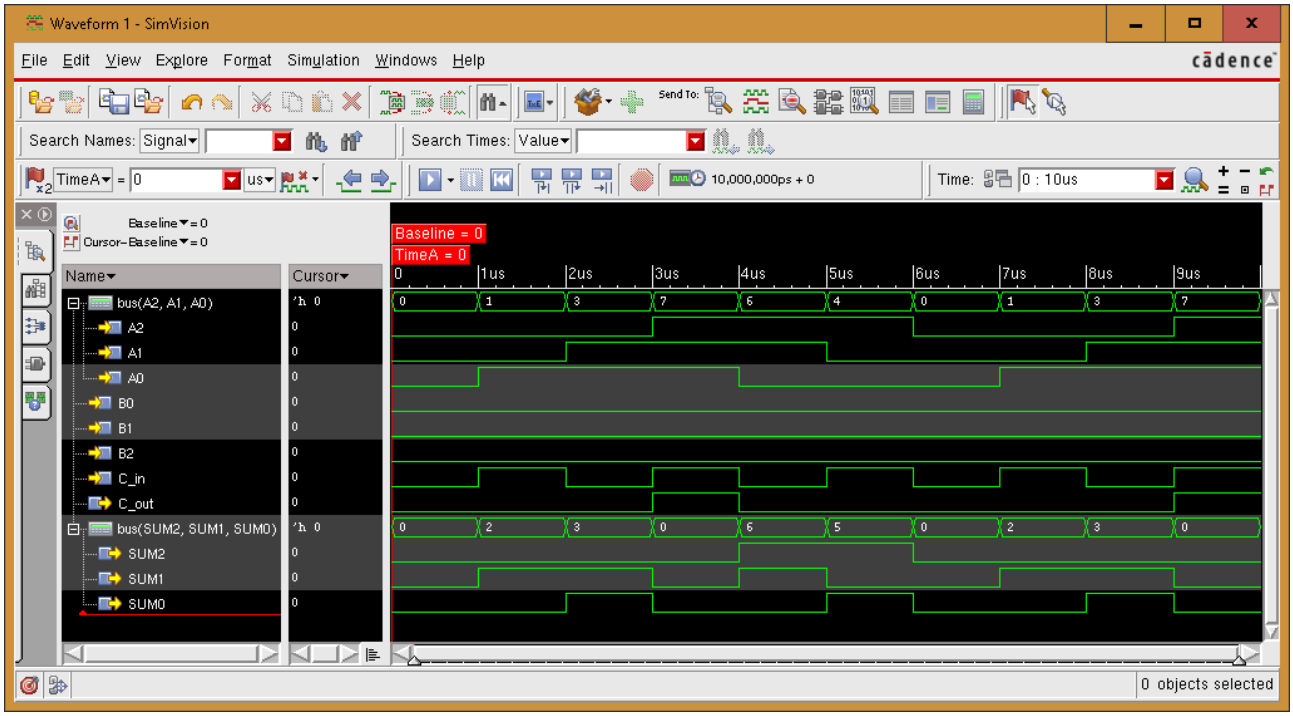
Фиг. 17. Визуализиране на пълния диапазон на резултатите от симулацията.

За да проверите резултатите, е удобно да групирате в шини (**bus**) сигналите от входовете *A0*, *A1* и *A2* и изходите *SUM0*, *SUM1* и *SUM2*. За да се получи правилната стойност е необходимо сигналите в шините да се подредят от най-старши към най-младши бит, напр. *SUM2, SUM1* и *SUM0*. За целта в прозореца **Waveform** се маркират, докато се държи натиснат клавиша Ctrl, първо *SUM2,* после *SUM1* и накрая *SUM0.* След това от менюто се избира **Edit** ⇒ **Create** ⇒ **Bus**.

При необходимост от корекция на файла с тестовите вектори, след неговото записване се извиква **Simulation** ⇒ **Reinvoke Simulator** от прозореца **Design Browser**, за да се отрязят въведените промени.

Ако сте работили с предложените тестови вектори, то на всеки **X**ns стойността на входовете *A* се променя, и оформена като 3-битово число представлява следната последователност: 0, 1, 3, 7, 6, 4 и след това се повтарят. За изхода стойностите са 0, 2, 3, 0, 6, 5 и след това се повтарят.

Един примерен резултат от симулацията е показан на фигура 13.



Фиг. 13. Резултати от симулацията.