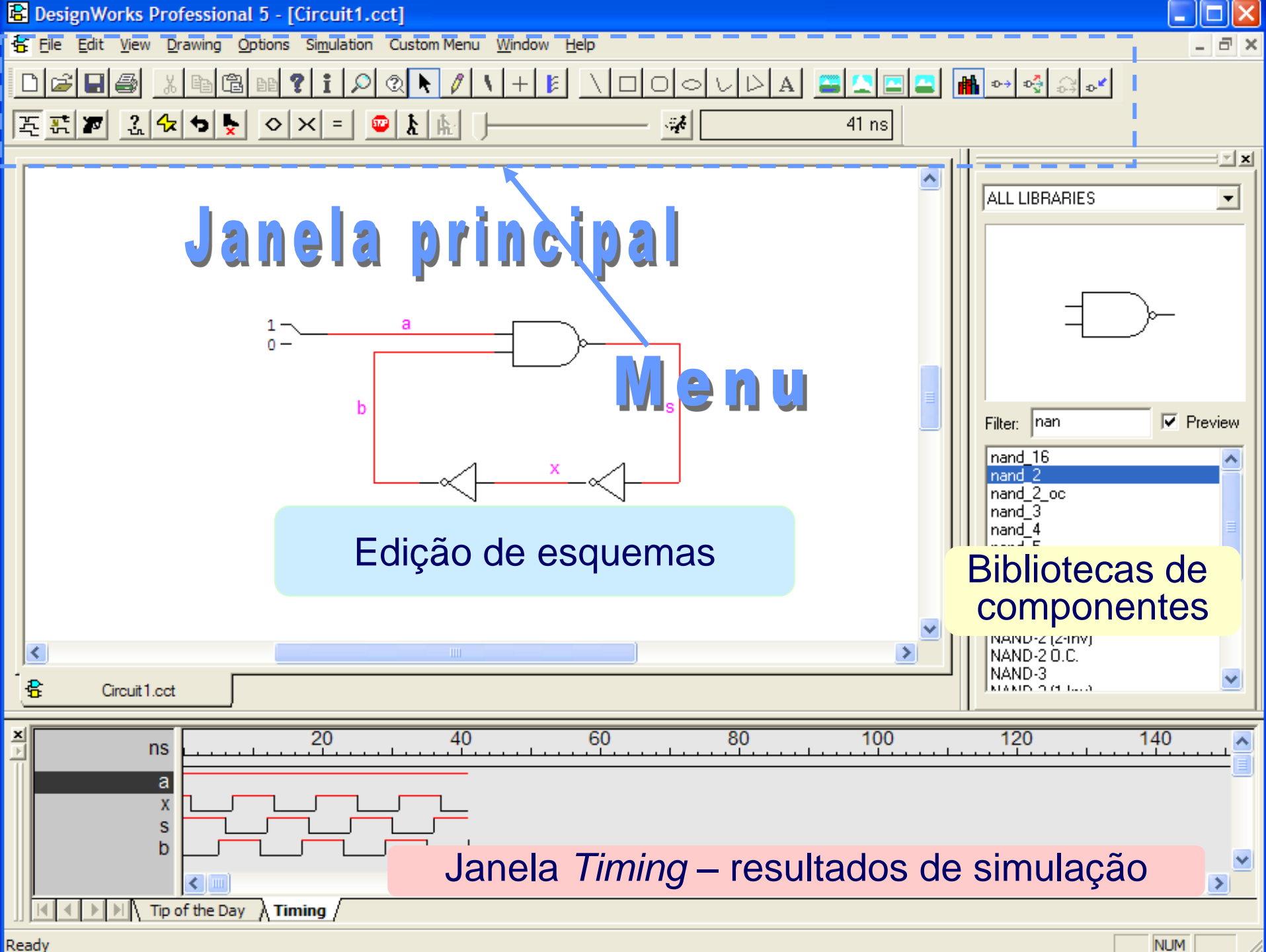


Tutorial

Exemplos de desenvolvimento
com a ferramenta DesignWorks 5

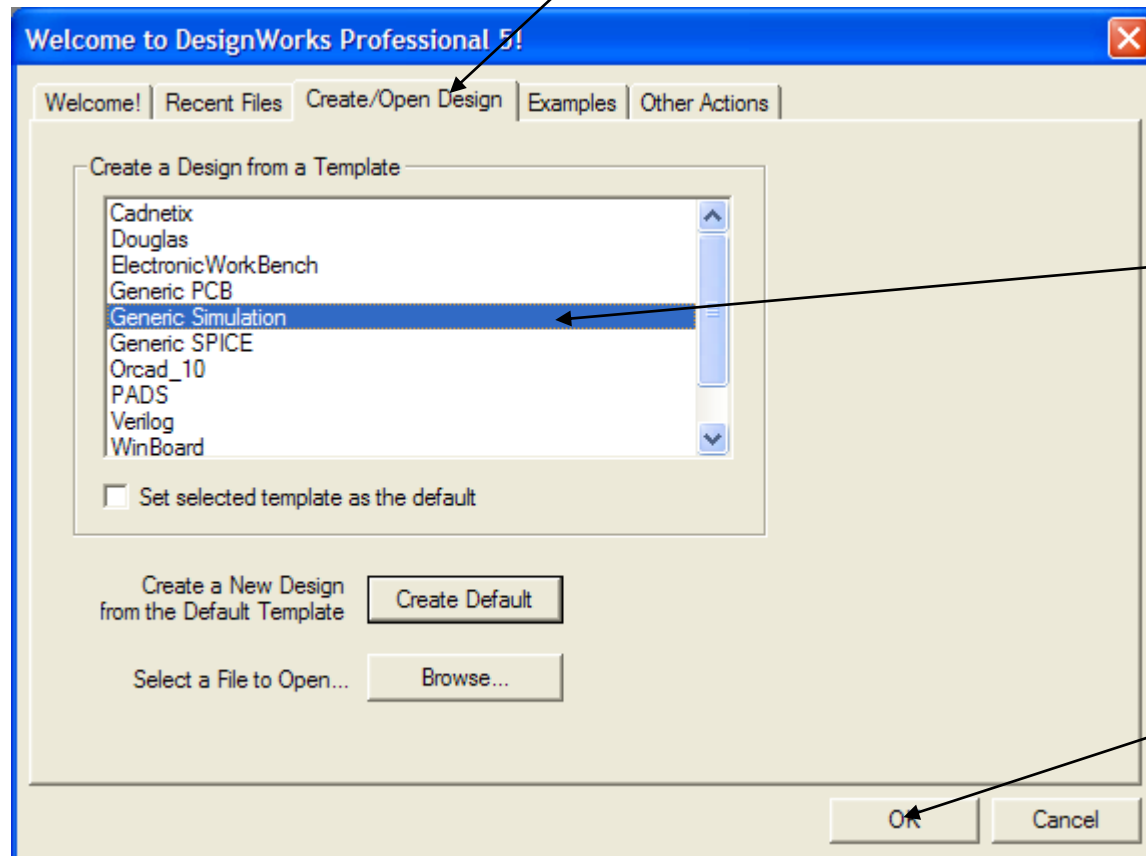
Iouliia Skliarova



Exemplo 1. Construção de um circuito combinatório simples

O exemplo ilustra todos os passos necessários para construir e simular um oscilador.

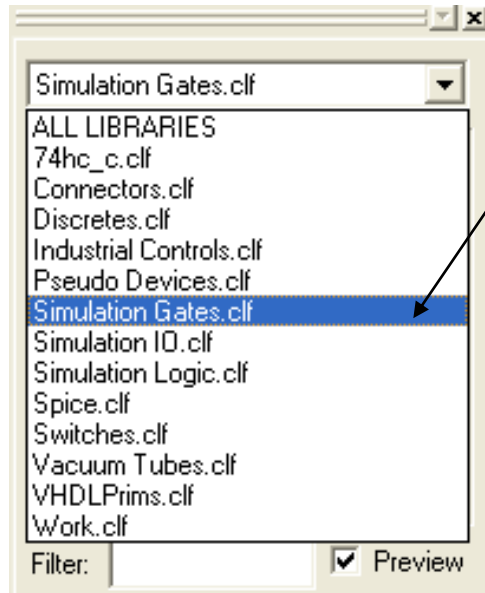
1 Selecione a *tab*
Create/Open Design



2 Selecione a opção
Generic Simulation

3 Clique no OK

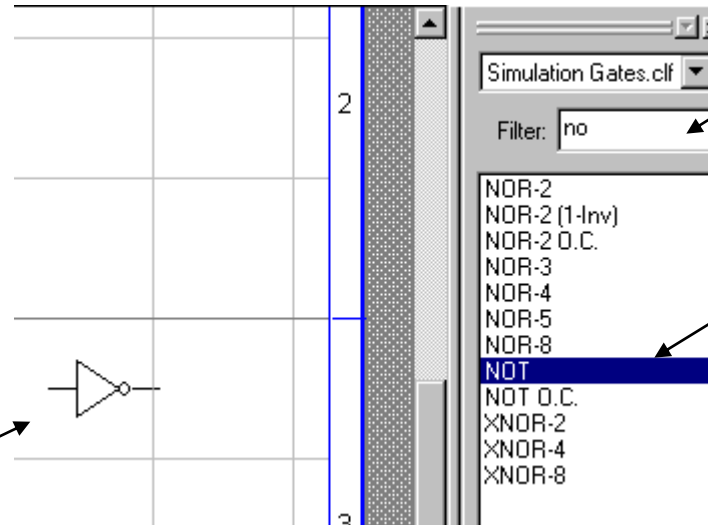
Selecione
uma biblioteca



4

Filtro para pré-seleção
de símbolos

5

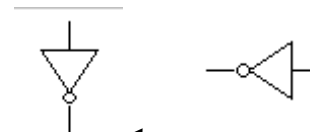


6

Escolha o
símbolo
desejado
clcando duas
vezes com o
botão
esquerdo do
rato

7

Agora é possível colocar o símbolo escolhido
no esquema clicando para tal com o botão
esquerdo do rato na zona pretendida

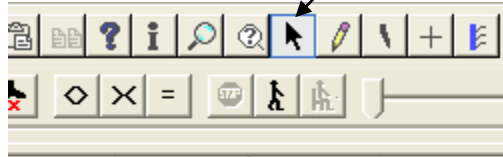


8

Pode-se mudar a orientação do símbolo (enquanto
não colocado) com a ajuda das teclas ↓ → ↑ ←

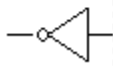
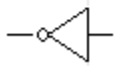
10

Para sair do modo de colocação de símbolos



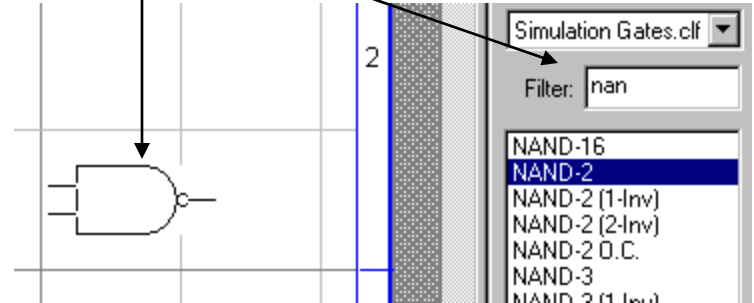
9

Coloque mais uma instância do símbolo *NOT*



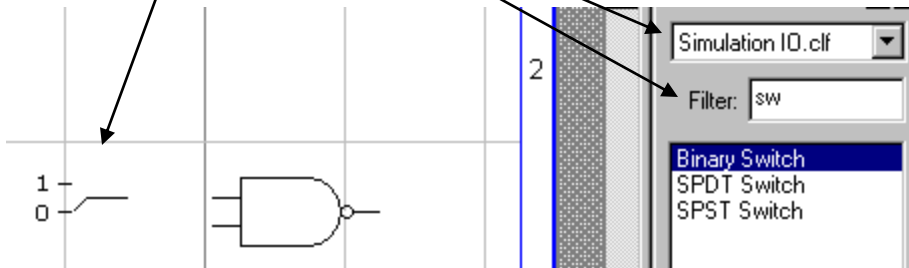
11

Selecione e coloque a porta lógica *NAND*



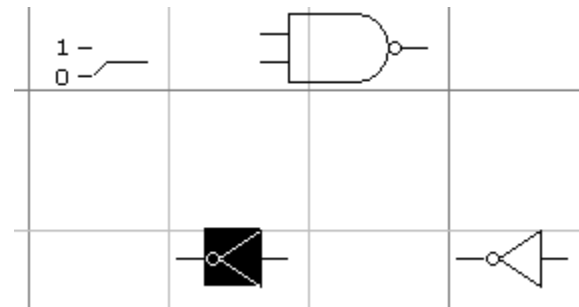
12

Mude para a biblioteca *SimulationIO* e selecione nela o elemento *Binary Switch*

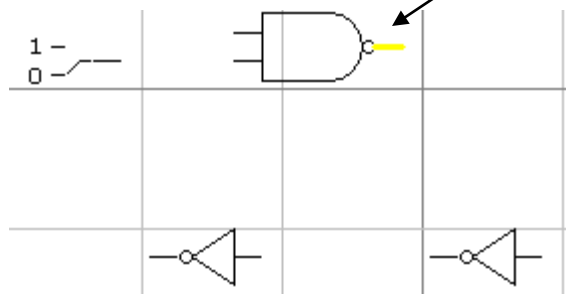


13

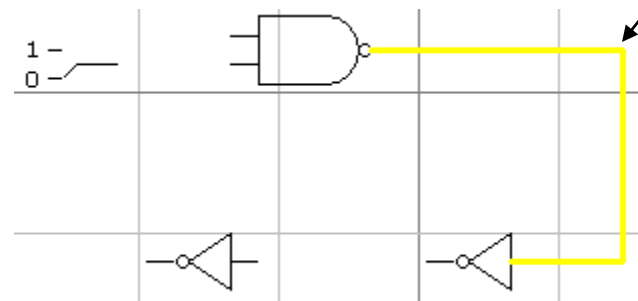
Para mudar a posição de um símbolo basta apontá-lo com o rato e arrastar para a posição desejada. Para selecionar o *Binary Switch* tem que pressionar simultaneamente a tecla *SHIFT*.



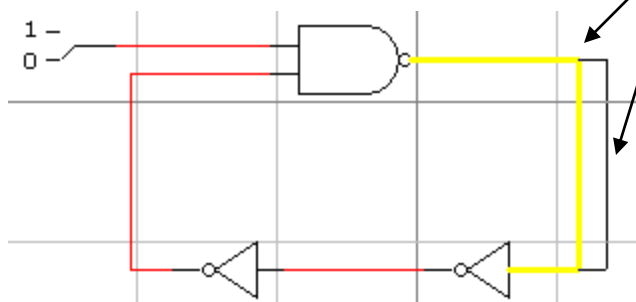
Para interligar as portas lógicas clique num pino



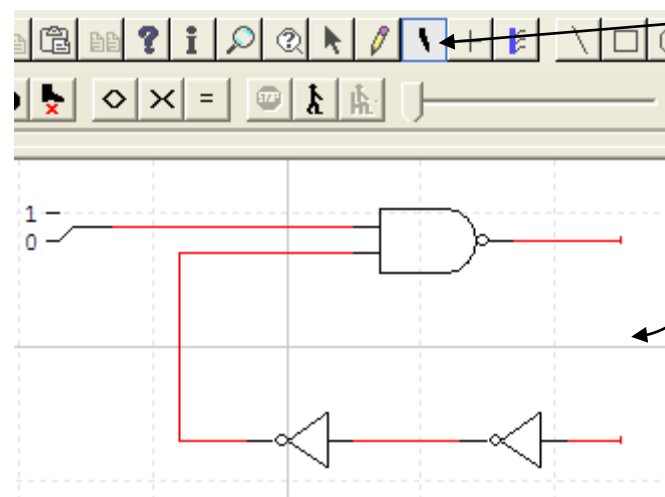
e arraste o rato até ao pino do símbolo com o qual se pretende estabelecer a ligação



Deste modo pode-se interligar todas as portas lógicas. A forma de qualquer ligação é modificável com a ajuda do rato.



Para apagar um fragmento de uma linha utiliza-se o botão Zap

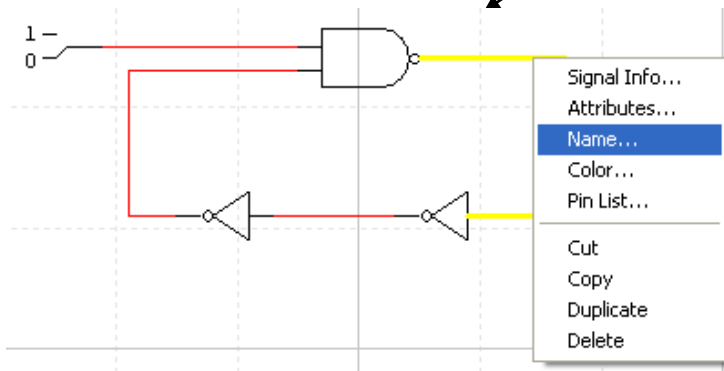


17

Para apagar um símbolo ou uma linha completa, selecione-o(a) e carregue na tecla *DEL*

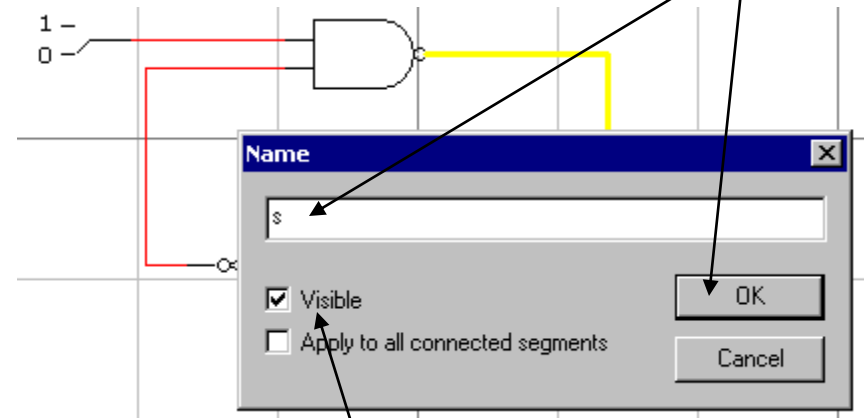
Se apontar para uma linha e
carregar no botão direito do rato
aparecerá um menu

19



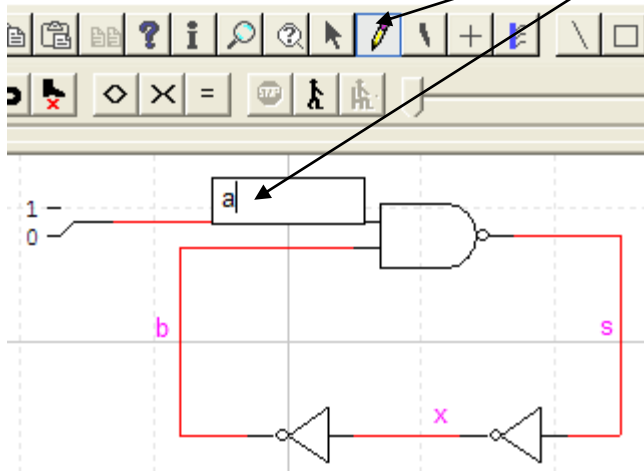
Este menu pode ser usado para
especificar o nome de um sinal

20



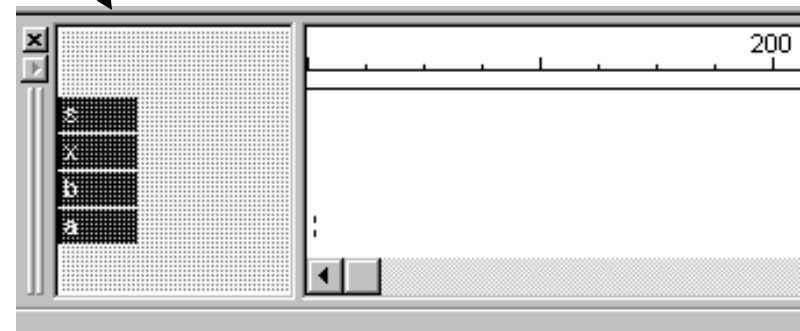
Como alternativa, pode-se
nomear um sinal com a ajuda
do botão *Naming*

21



Todos os sinais nomeados são
adicionados automaticamente à janela
Timing. Se isto não acontecer verifique
se a opção *Visible* foi selecionada e se
a opção do menu *Simulation -> Add
Automatically* é ativada.

22

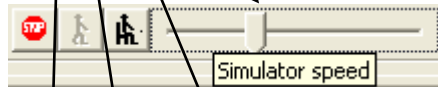


23

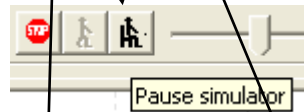
Opções do simulador:



Simulação passo a passo



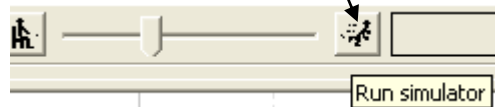
Velocidade de simulação



Pausar o simulador



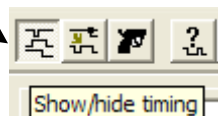
Parar o simulador



Simulação à velocidade máxima

26

Com a ajuda deste botão pode-se esconder a janela *Timing*



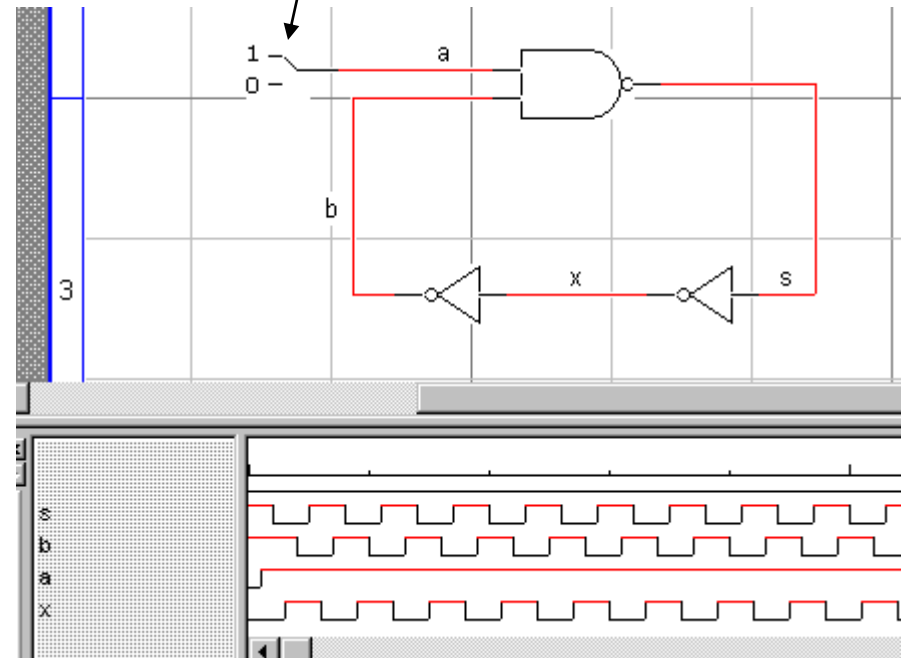
Para melhorar a visualização dos diagramas temporais pode-se utilizar as opções de *Zoom*

24



25

Pode mudar do estado do *Binary Switch* clicando nele com o botão esquerdo do rato durante a simulação



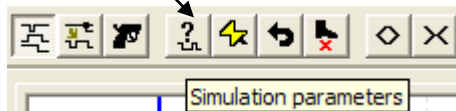
27

Para reinicializar o simulador carregue neste botão



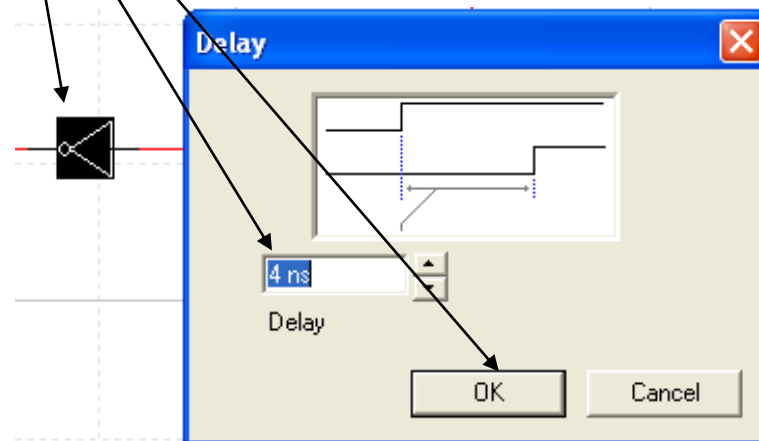
28

Para especificar parâmetros de simulação serve este botão



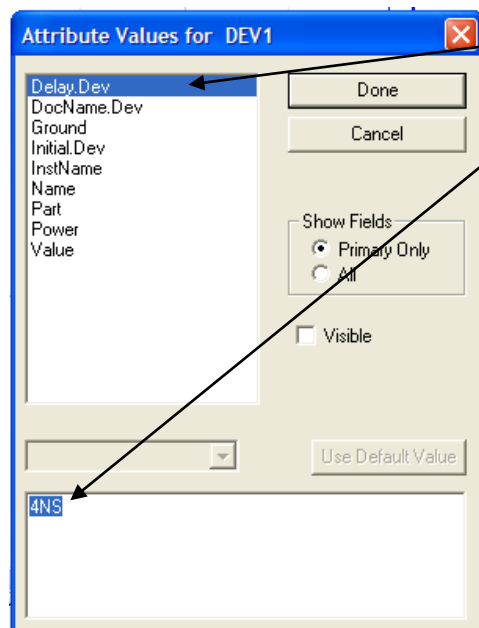
Por exemplo, para especificar o atraso de propagação de uma porta lógica deve selecioná-la e de seguida carregar no botão *Simulation parameters*. Como resultado aparecerá uma janela de diálogo na qual pode-se definir o atraso.

29

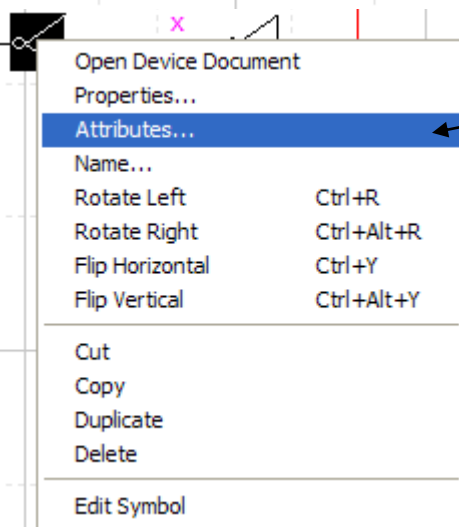


31

Por omissão, o atraso de um dispositivo primitivo é 1 ns. Aqui especificámos o atraso de 4 ns.



30



Alternativamente, pode carregar no símbolo com o botão direito do rato e usar a opção *Attributes...* do menu

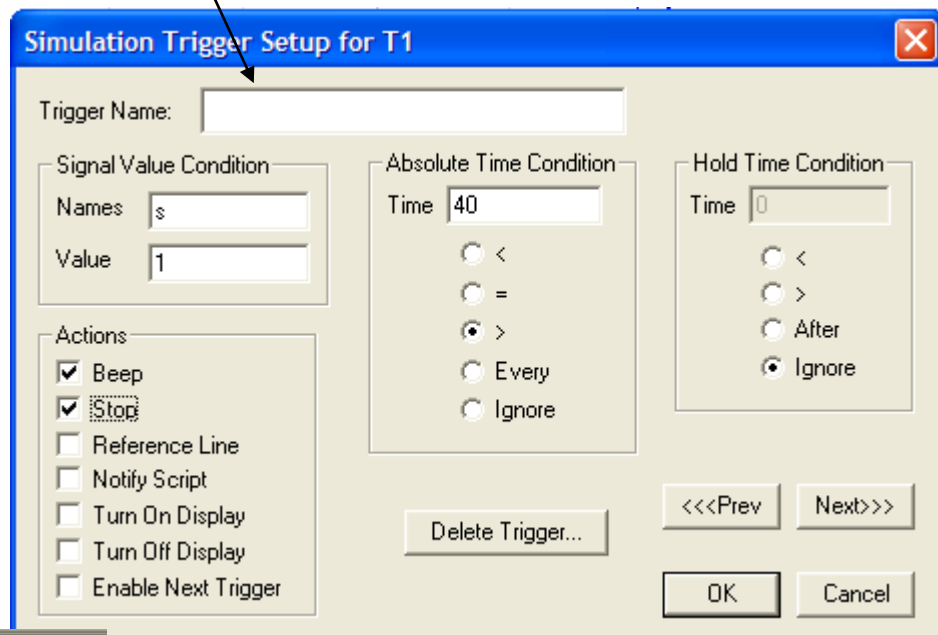
32

Este botão serve para detetar automaticamente uma situação específica durante a simulação



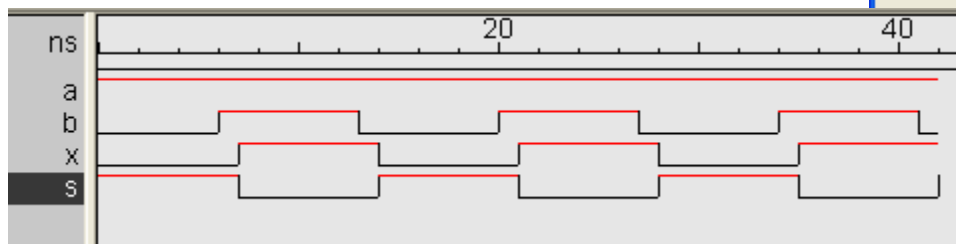
33

Por exemplo, podemos simular o comportamento do circuito até ao momento em que o sinal **s** receba o valor **1** mas não menos do que 40 ns



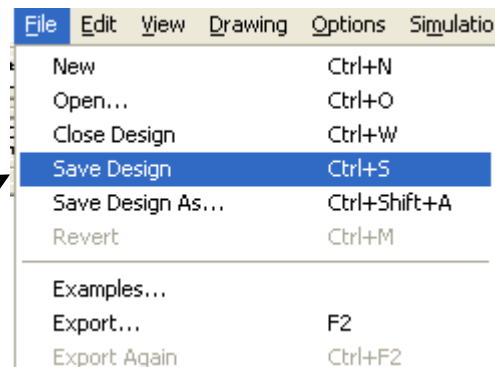
34

O simulador vai parar automaticamente e emitir um sinal sonoro quando atingir a situação especificada



Utilize a opção *Save Design* do menu *File* para gravar o circuito

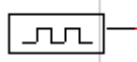
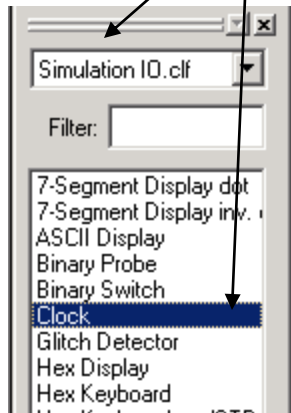
35



Exemplo 2. Experimentação com portas lógicas simples

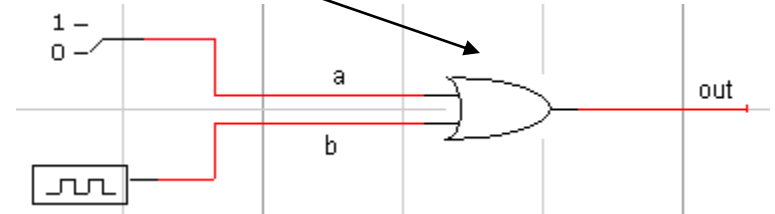
1

Selecione componente *Clock* na biblioteca *SimulationIO*



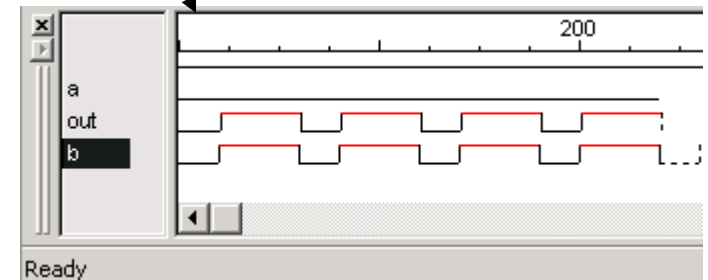
2

Construa o circuito seguinte. A porta lógica utilizada chama-se *OR-2*



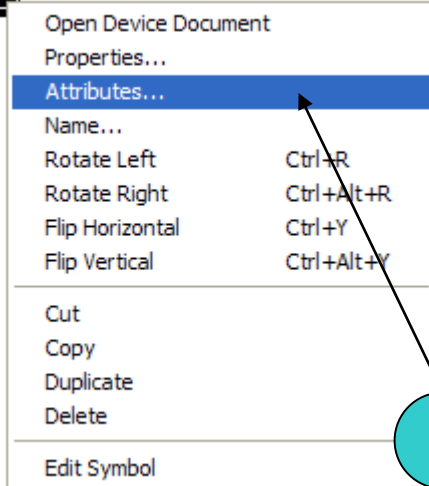
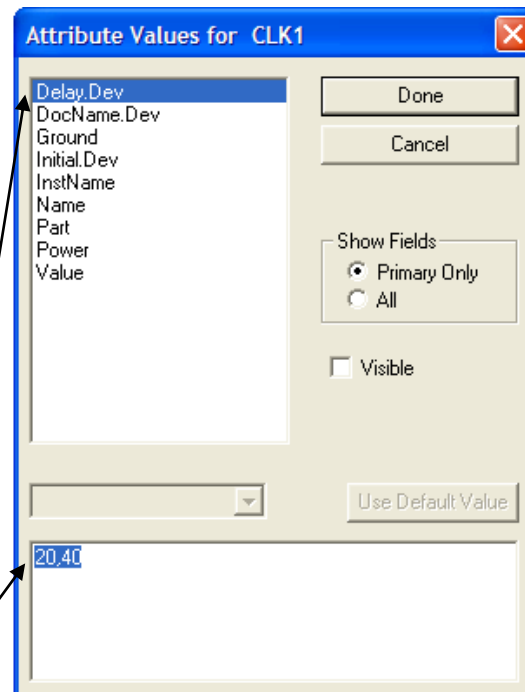
4

Verifique como os parâmetros definidos influenciam a simulação



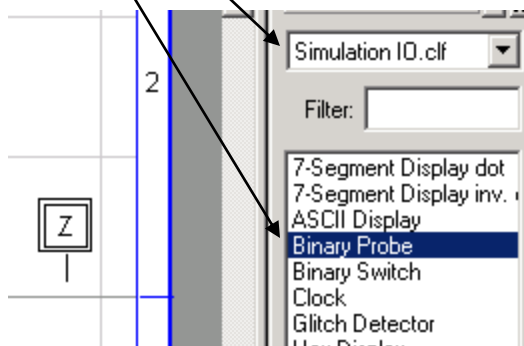
3

Nos atributos do componente *Clock* é possível especificar os parâmetros do relógio



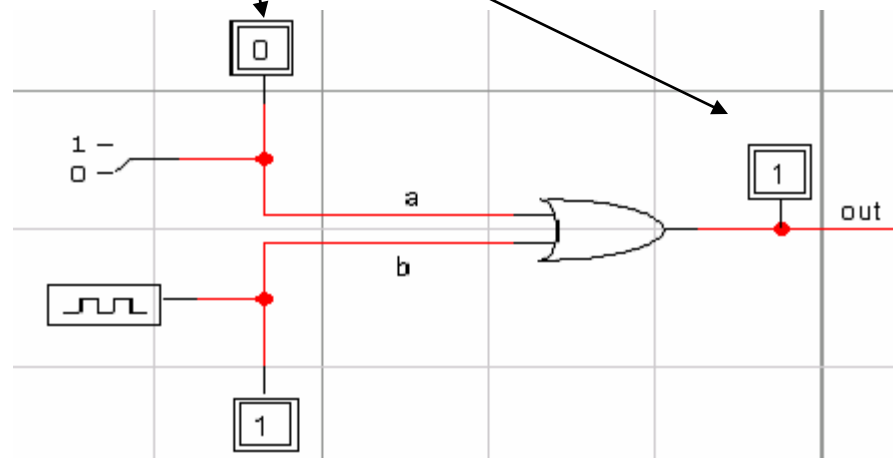
5

Para além dos diagramas temporais, pode-se usar o componente *Binary Probe* disponível na biblioteca *SimulationIO* para visualizar o valor corrente de um sinal



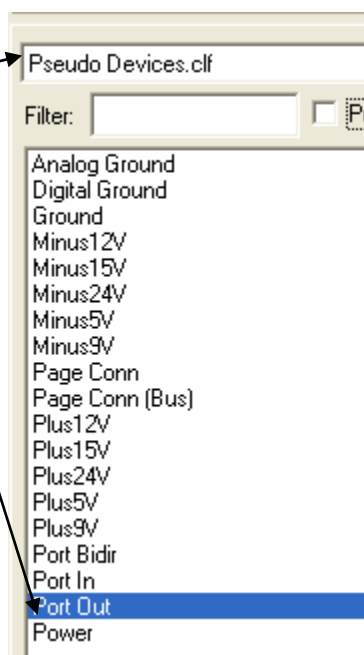
6

Adicione 3 *Binary Probes* conforme ilustrado na figura



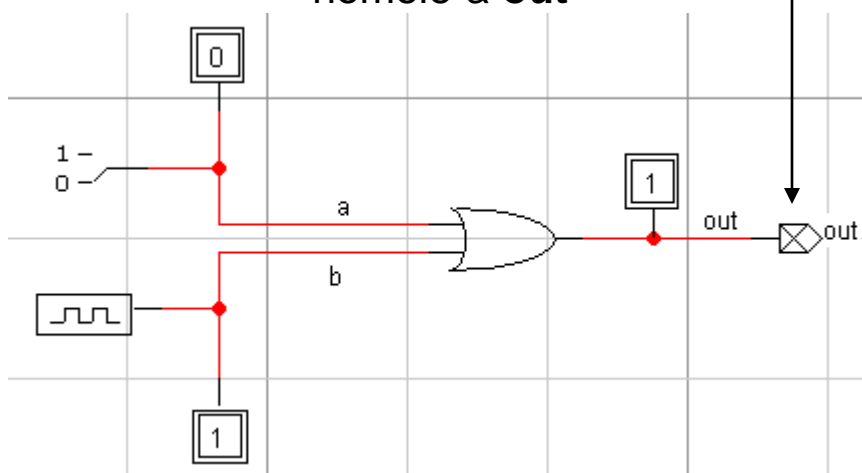
7

Se planeia utilizar o circuito construído como uma parte de um outro circuito maior, deve adicionar as portas de entrada e/ou saída localizadas na biblioteca *Pseudo Devices*



Crie uma porta de saída e nomeie-a **out**

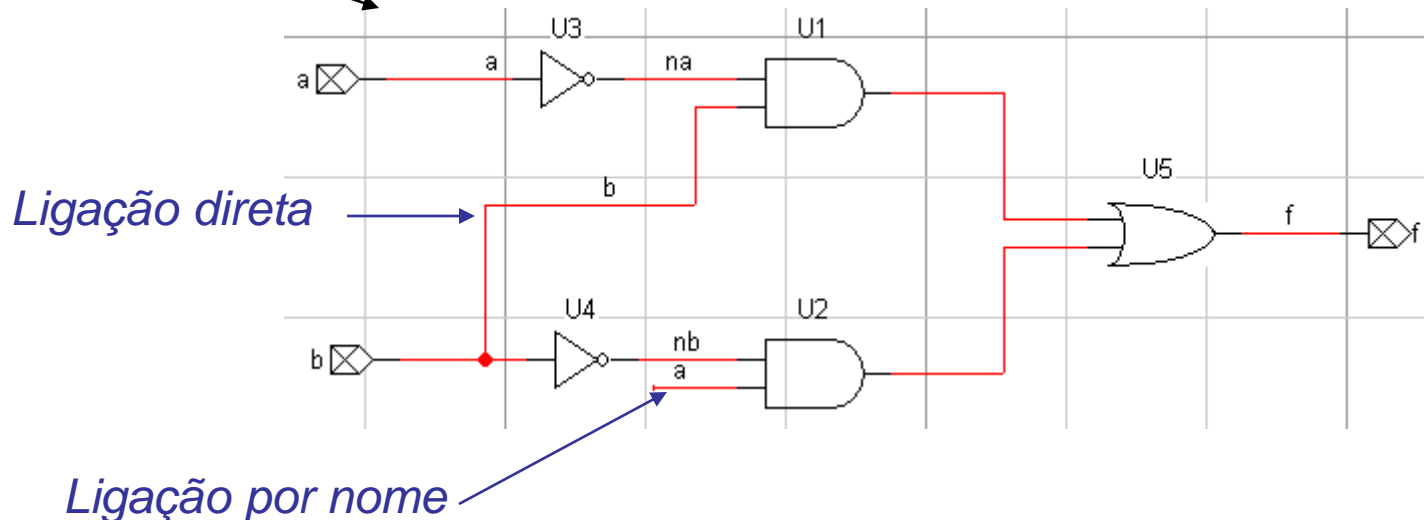
8



Exemplo 3. Construção de circuitos hierárquicos

O exemplo ilustra todos os passos necessários para construir e simular um circuito hierárquico.

- 1 Construa o circuito seguinte. **a**, **b** e **f** representam portas de entrada e de saída

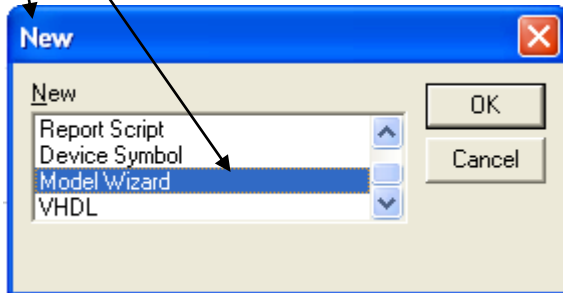


- 2 Repare que várias linhas podem ser interligadas *diretamente* (assim como a porta de entrada **b** com o inversor **U4** e a porta lógica **U1**) ou *por nome* (assim como a porta de entrada **a** com a porta lógica **U2**)
- 3 Grave o circuito com o nome *my_xor.cct*

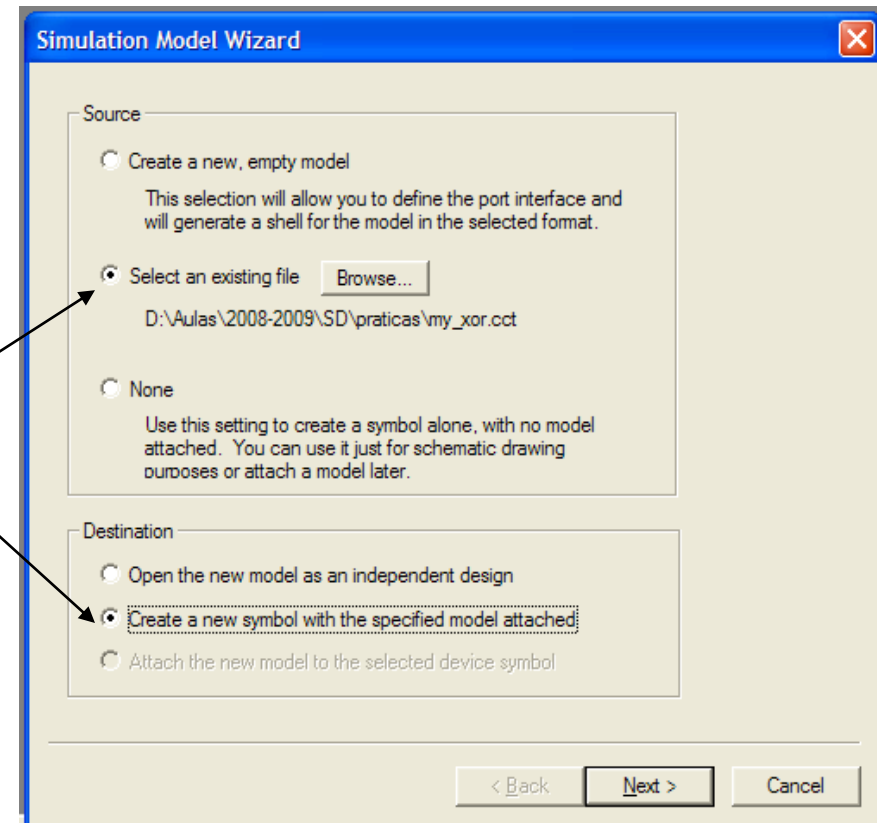
4 Suponhamos que precisamos de utilizar o circuito construído no nível superior de hierarquia. Para tal será necessário criar um símbolo para o nosso circuito para podermos posteriormente instanciá-lo.

5 A estratégia de projeto, na qual primeiro projeta-se um sub-circuito e só depois se cria um símbolo para ele, é conhecida sob o nome *bottom-up*

6 Para criar um símbolo novo (de maneira mais simples), feche todos os ficheiros abertos, selecione a opção do menu *File -> New* e na janela de diálogo que aparecerá escolha a opção *Model Wizard*

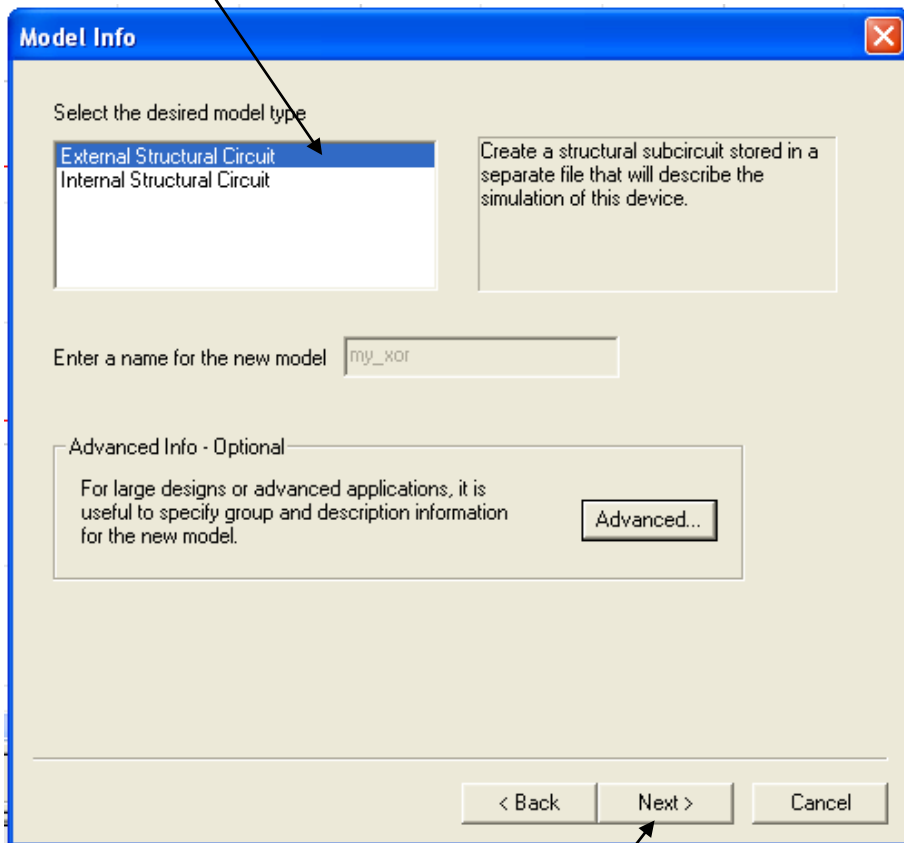


Na janela de diálogo que vai aparecer escolha as opções *Select an existing file* (aponte para o circuito my_xor.cct com a ajuda do botão *Browse...*) e *Create a new symbol with the specified model attached*



8

Na janela de diálogo seguinte
selecione a opção *External
Structural Circuit*

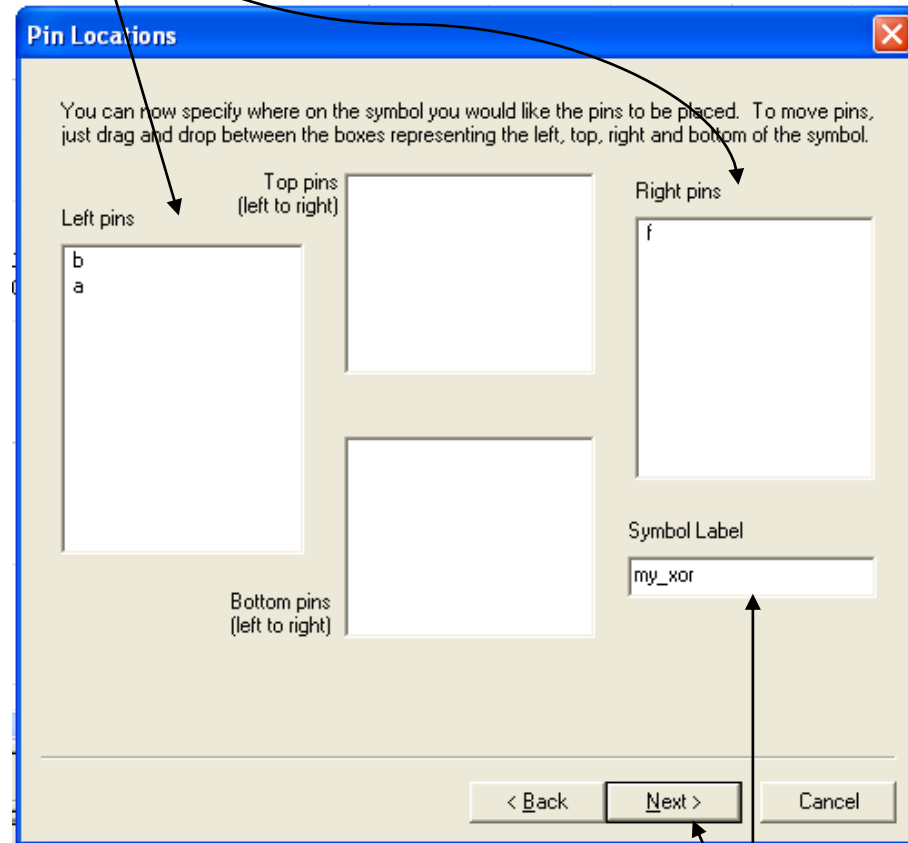


Carregue em *Next >*

9

10

Repare que os pinos de
entrada/saída que correspondem
às portas de entrada/saída do
nosso circuito apareceram na lista:



Pode alterar o nome do símbolo e
carregue no botão *Next >*

11

12

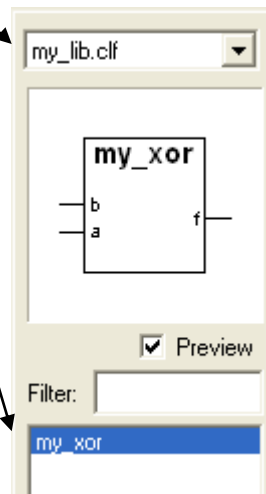
Para poder utilizar o símbolo criado deve guardá-lo numa biblioteca

13

Na janela de diálogo que aparecerá carregue no botão *New Lib...* e crie uma biblioteca chamada, por exemplo, *my_lib.clf*

14

Agora na janela de bibliotecas pode verificar a existência do símbolo *my_xor*



Save Symbol

☐ Place the block immediately without saving it in a library. WARNING: If you cancel the place operation, you will have to start over.

Enter a name for the new part

Select the existing library to save the block to or create a new one:

my_lib.clf
Work.clf

Open Lib...

New Lib...

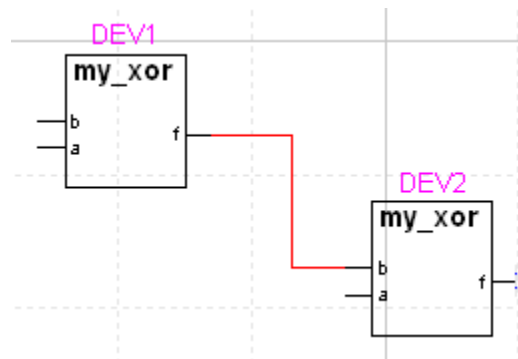
NOTE: Libraries marked as Read-Only are not shown in this list.

< Back

Finish

Cancel

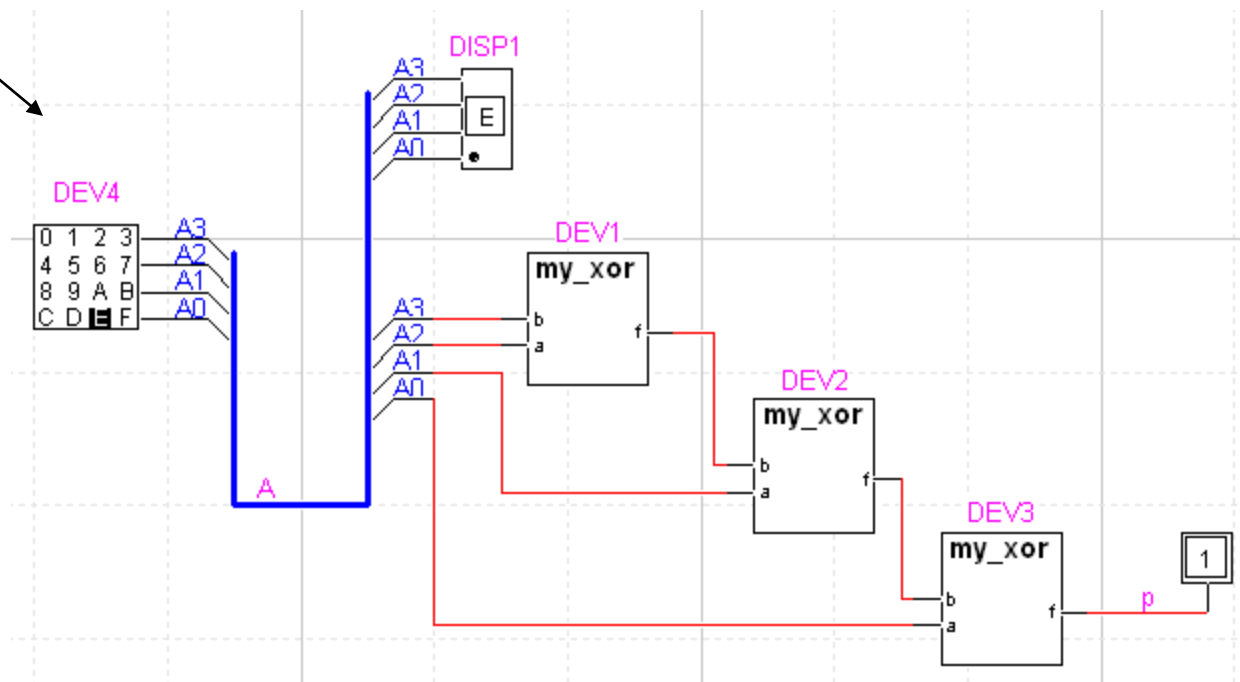
15



O componente *my_xor* pode ser utilizado no esquemático possibilitando deste modo a especificação hierárquica

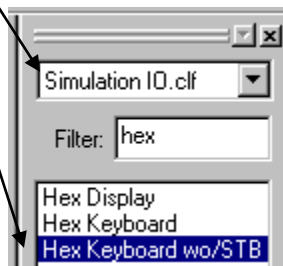
16

Vamos utilizar o componente *my_xor* para construir o circuito seguinte:



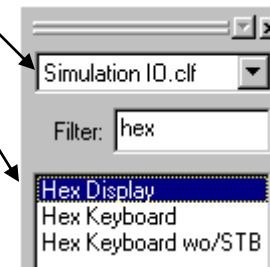
17

Componente *DEV4* é um teclado hexadecimal existente na biblioteca *Simulation IO*



18

Componente *DISP1* é um *display* hexadecimal existente na biblioteca *Simulation IO*

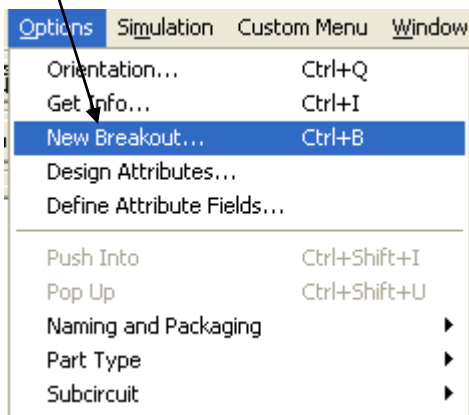


19

A é um barramento. Para ligar sinais individuais a um barramento utiliza-se um símbolo especial: *breakout*.

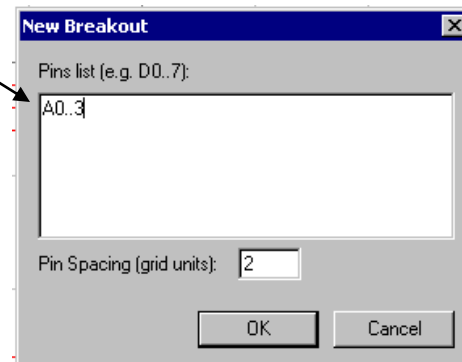
20

Para criar um *breakout* selecione a opção *New Breakout...* no menu *Options*



21

Na janela de diálogo que vai aparecer indique que sinais pretende ligar ao barramento (separando-os por espaços ou utilizando a notação ilustrada na figura)

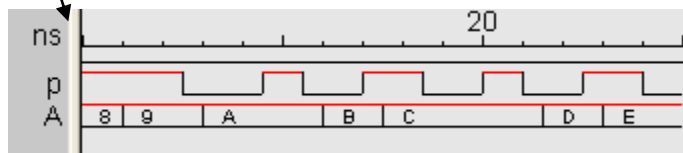


24

Como é possível otimizar o circuito construído?

22

Interligue todos os componentes e simule o circuito



23

Repare que o barramento **A** pode ser decomposto em sinais individuais:

