

Notas sobre o simulador de circuitos digitais “*Digital Works*”

1. Introdução

O *Digital Works* é uma ferramenta gráfica de simulação que permite construir e analisar o comportamento de circuitos digitais. Os circuitos a simular podem ser compostos por portas lógicas, flip-flops, unidades de memória, etc.

Uma das suas capacidades mais importantes é a criação de macros, através das quais é possível construir um novo componente a partir de um circuito desenhado pelo utilizador. Esse novo componente pode ser guardado para ser usado mais tarde na construção de outros circuitos mais complexos. Deste modo é possível ir acrescentando novos componentes à lista de componentes do programa. Por exemplo, podem criar-se componentes como contadores, comparadores, registos ou mesmo criar circuitos integrados como o 74HCT08.

A versão a usar nas aulas práticas de Sistemas Lógicos é a 2.0, que é freeware, embora haja outras mais recentes e com capacidades adicionais mas que são pagas. Operar com o programa é muito simples, existindo um *help* integrado que explica as diversas funcionalidades. Nesta introdução são apenas indicados os procedimentos elementares para começar a trabalhar com o simulador.

2. Ficheiros

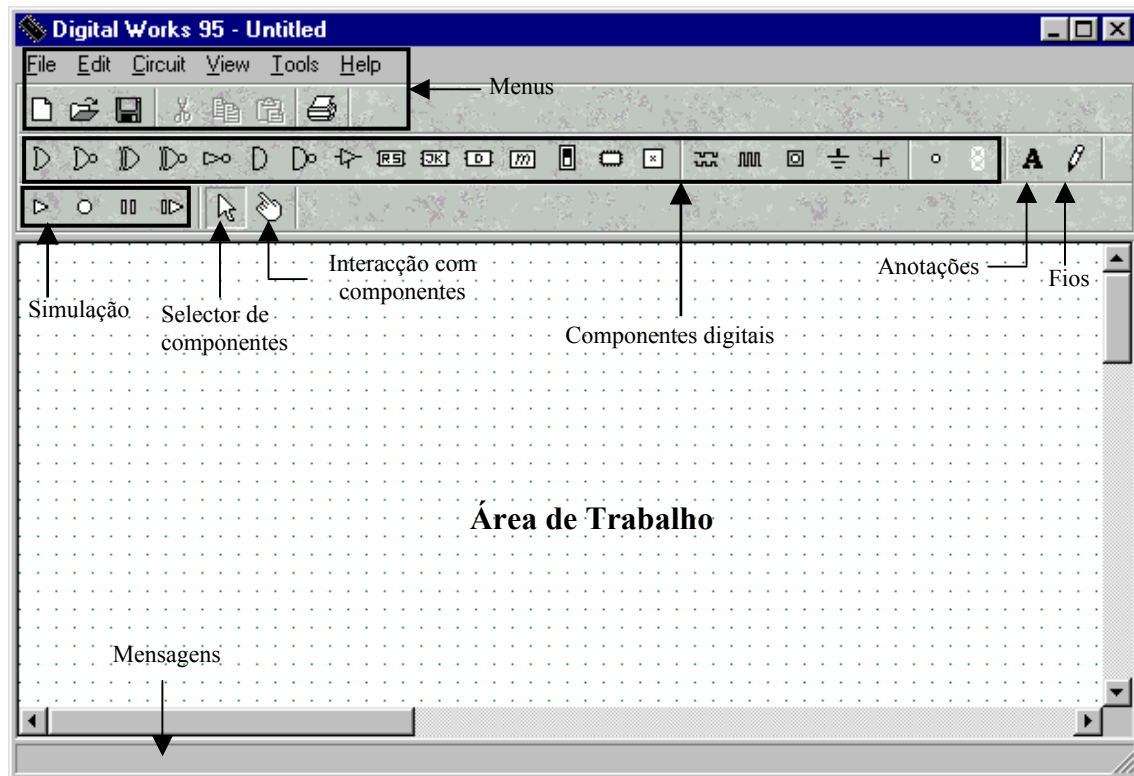
O simulador consta de um único ficheiro “Dw20_95.exe” a partir do qual é feita a instalação para qualquer versão do MS-Windows (esta versão foi desenvolvida para o Win95, mas corre em todas as versões até à XP). O ficheiro instala o executável e o desinstalador. É criada por defeito a directoria “\DigitalWorks” e mais algumas directorias nas quais existem ficheiros dos tipos “*.dwm” (digital works macros), “*.dwt” (digital works templates) e ainda “*.map” (memory mapping). As directorias são:

- \Integrated Circuits – alguns componentes pré-definidos (ex. portas lógicas)
- \Templates – formatos que os circuitos do utilizador podem assumir
- \Macros – circuitos criados pelo utilizador
- \Sample Circuits – exemplos de circuitos

É possível guardar os ficheiros criados pelo utilizador em quaisquer outras directorias.

3. Utilização

A figura seguinte apresenta o ecrã inicial do simulador.



Área de trabalho – espaço aonde são colocados os componentes do circuito a simular.

Mensagens – mensagens de acções que devem ser executadas, avisos de erros, etc.

Menus – menus de abertura/salvaguada de ficheiros, impressão, configuração, ferramentas e *help*.

Componentes digitais – lista de componentes que podem ser utilizados na construção dos circuitos.

Selector de componentes – permite seleccionar os componentes a colocar na área de trabalho.

Interação com os componentes – em modo de simulação, permite interagir com os componentes alterando o seu estado de funcionamento.

Anotações – permite introduzir textos tais como títulos ou notas sobre o funcionamento dos circuitos.

Fios – desenha os fios de ligação entre os vários componentes; o programa impede ligações impossíveis como, por exemplo, entre duas saídas. Vão sendo mostradas mensagens que indicam em cada momento o que pode ou deve ser feito.

Simulação – controla a simulação; em modo simulação pode interactivar-se com os componentes do circuito, actuando sobre as suas entradas e observando o efeito sobre as saídas.

4. Exemplo

Como exemplo de utilização do simulador, vai usar-se o circuito de controlo do motor do limpa-pára-brisas (LPB) apresentado nas aulas. A equação do circuito é: $LPB = I \cdot (C + R)$

em que: LPB = motor do limpa-pára-brisas

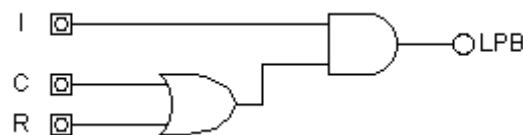
I = chave de ignição

C = comando manual do limpa-pára-brisas

R = detector de chuva

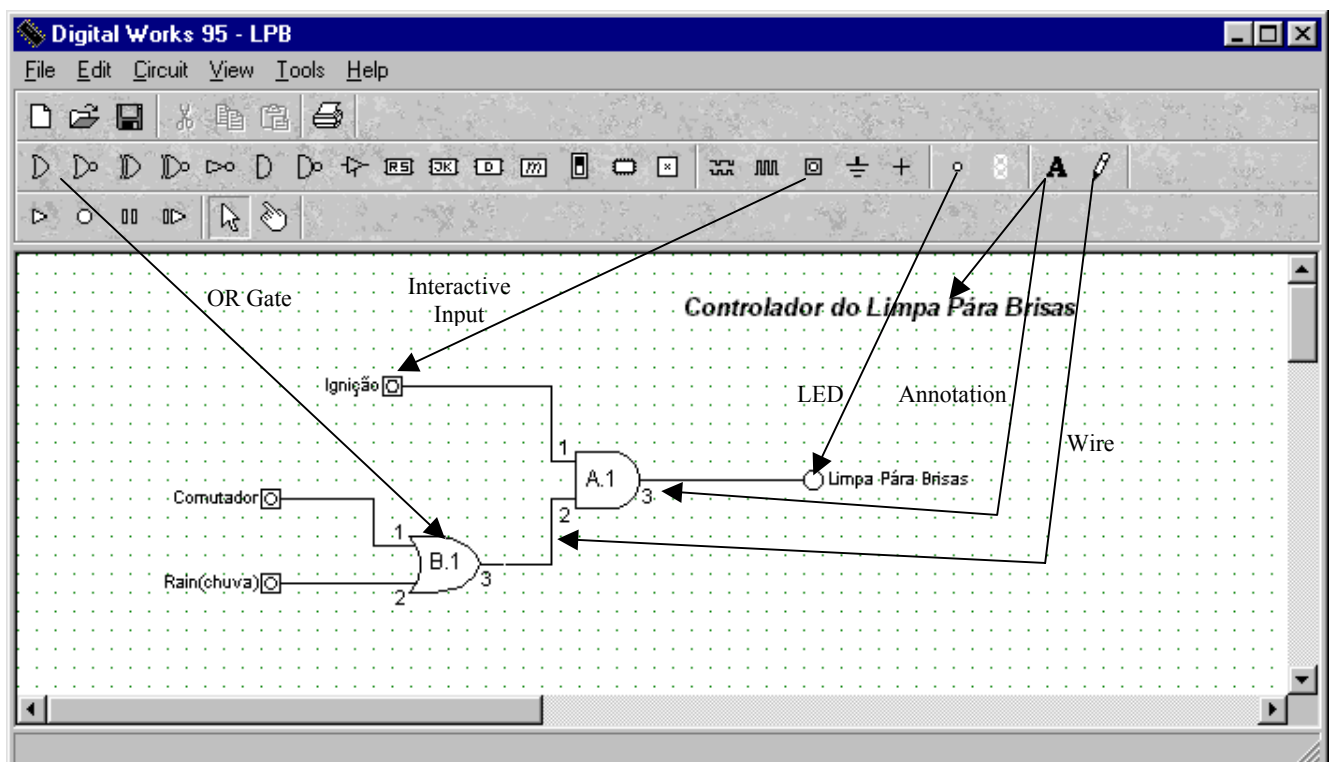
a que corresponde a seguinte tabela de verdade e o respectivo circuito lógico:

I	C	R	LPB
0	X	X	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



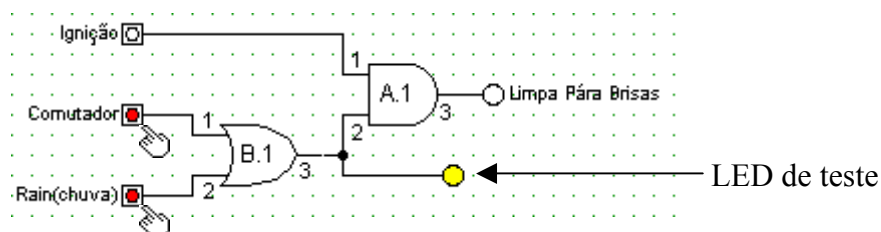
4.1 Desenho do circuito

A figura abaixo mostra o desenho do circuito no simulador. Os componentes (ex: porta OR) são inseridos clicando sobre eles e depois sobre a área de trabalho no local desejado. O posicionamento pode ser feito com o rato, ou, com mais precisão, usando as teclas do cursor. Enquanto se procede ao desenho, as saídas (LED do LPB) permanecem desligadas, não reagindo às entradas.

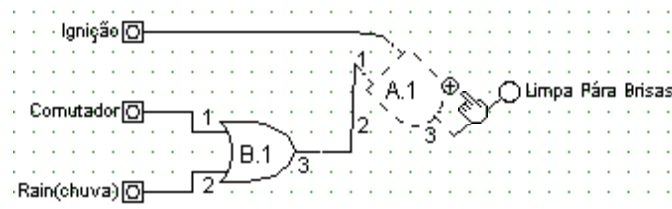


Notas:

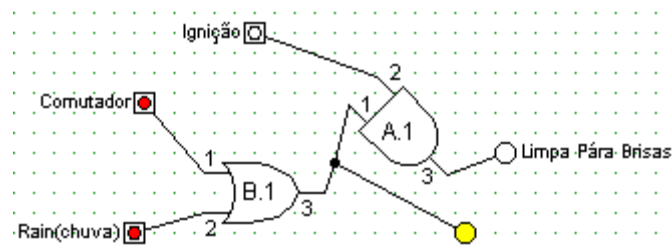
1. Quando um componente se encontra no nível lógico “0” aparece representado a branco, quando no nível “1” aparece a vermelho (neste caso é possível seleccionar outra cor).
2. Carregando em F1 quando um componente está seleccionado na área de trabalho, é mostrada a ajuda sobre esse componente.
3. Clicando com o botão direito do rato sobre um componente, aparece a lista de opções para esse componente (cores, textos, nº de entradas das portas, parâmetros, etc).
4. Se for necessário testar o estado de pontos intermédios do circuito, podem acrescentar-se LEDs adicionais que servem como pontos de teste; após o teste o LED deverá ser removido do circuito final. A figura seguinte exemplifica uma situação em que é usado um LED amarelo para testar o estado da saída da porta OR:



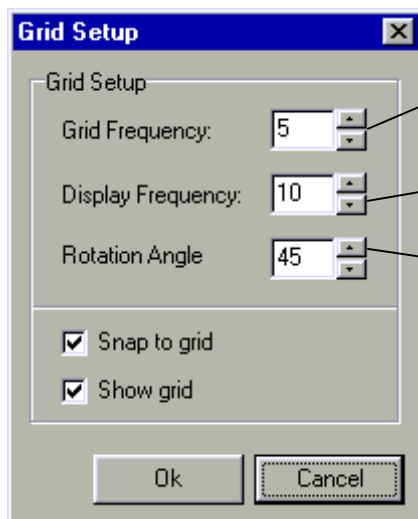
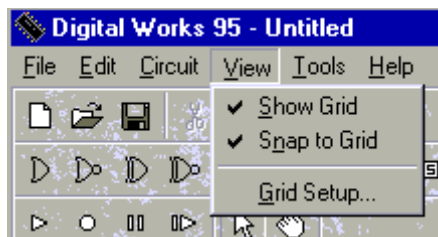
5. Quando um componente se encontra seleccionado na área de trabalho, aparece junto a ele um sinal (+) a partir do qual é possível rodar o componente usando o rato:



6. O modo organizado de dispor os componentes é um aspecto muito importante para manter a legibilidade dos circuitos, principalmente quando estes comportam muitos componentes. A figura seguinte ilustra um caso de má colocação dos componentes:



7. O posicionamento dos componentes na área de trabalho é feito de acordo com a grelha que estiver presente no ecrã. A alteração das características da grelha pode ser feita através do menu “View” ou clicando com o botão direito do rato sobre um local livre da área de trabalho.



Unidade de movimento dos componentes. Controla a amplitude do “salto” relativo ao posicionamento dos componentes. Um menor valor aproxima os pontos permitindo maior precisão no posicionamento dos componentes.

Espaçamento entre os pontos visíveis da grelha. Um menor valor aproxima os pontos permitindo maior precisão no posicionamento dos componentes. (Nota: Display Frequency \geq Grid Frequency)

Ângulo de rotação dos componentes

☒ Snap to grid os componentes movem-se por “saltos”

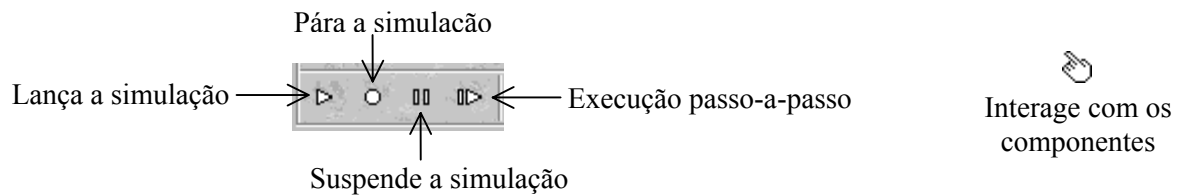
☐ Snap to grid os componentes movem-se livremente

☒ Show grid mostra a grelha

☐ Show grid esconde a grelha

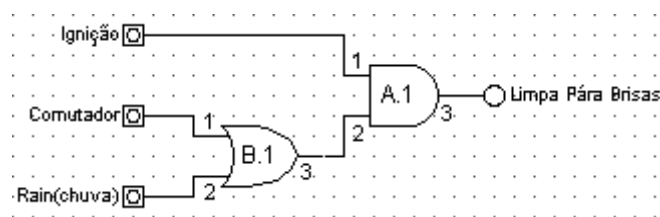
4.2 Execução da simulação

Para verificar se o circuito funciona correctamente deve executar-se a simulação. Os botões abaixo são utilizados para a controlar:

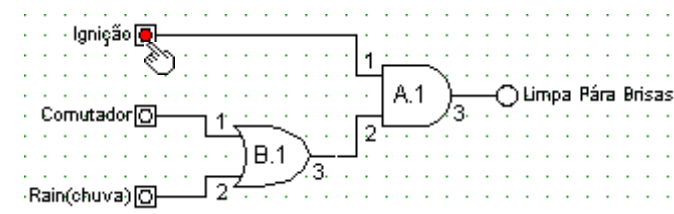


Após lançar a simulação pode usar-se o botão de interacção com os componentes para alterar o estado das entradas. As figuras a seguir mostram exemplos sucessivos dessa actuação.

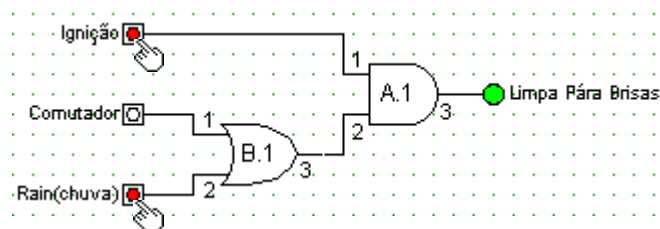
I	C	R	LPB
0	X	X	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



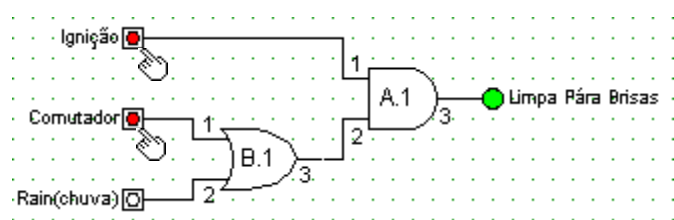
I	C	R	LPB
0	X	X	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



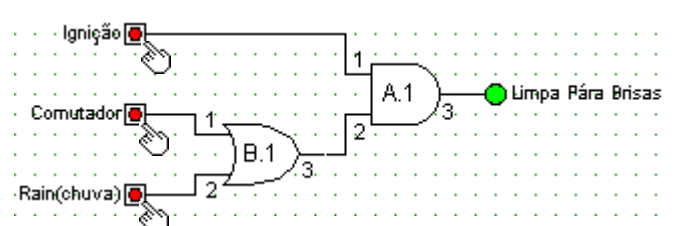
I	C	R	LPB
0	X	X	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



I	C	R	LPB
0	X	X	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



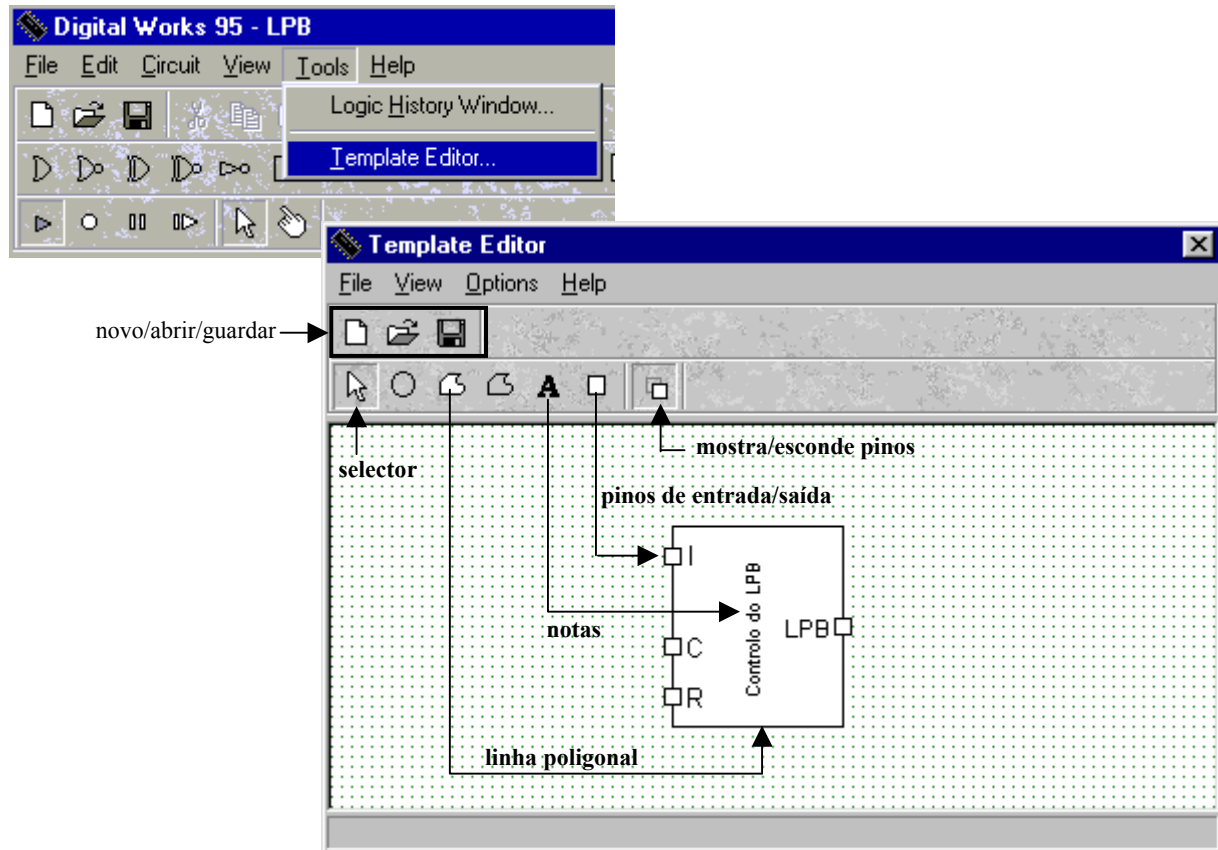
I	C	R	LPB
0	X	X	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



5. Templates

Um *template* é uma forma que um determinado componente ou circuito pode assumir, independentemente da sua funcionalidade. Por exemplo, os circuitos integrados digitais costumam ser representados por rectângulos nos esquemas electrónicos. O *Digital Works* permite criar *templates* que mais tarde servirão para encapsular os circuitos criados pelo utilizador (macros).

Uma das formas de activar o editor de *templates* é através do menu *Tools*:



O editor funciona como uma espécie de programa de desenho, possibilitando criar formas a partir de círculos, polígonos ou linhas. A janela acima apresenta um *template* possível para o circuito de controle do limpa-pára-brisas. Note-se que o *template* é apenas uma forma sem qualquer funcionalidade, a qual só será introduzida mais tarde, por associação com o circuito lógico do LPB. Uma vez que é só uma forma, o *template* pode ser utilizado para diferentes circuitos com diferentes funcionalidades. O *template* acabado de criar deve ser guardado junto de outros *templates*, por exemplo na directoria “\Templates”. Neste caso ele poderia chamar-se “LPB.dwt”.

6. Macros

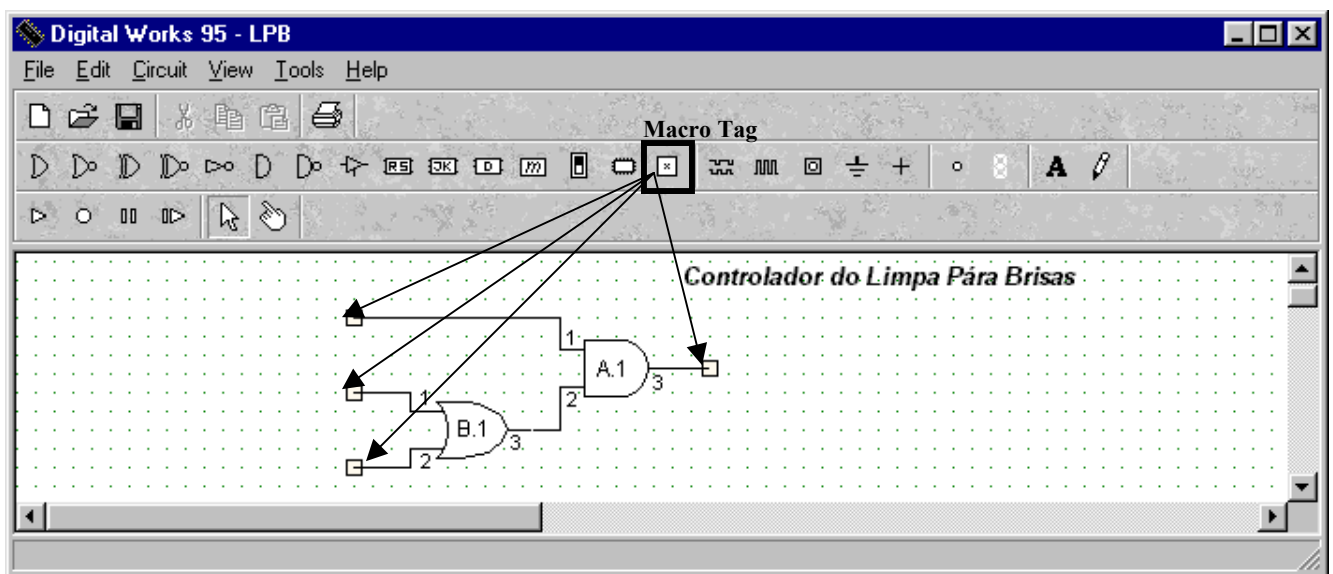
Uma macro corresponde a um circuito que apresenta uma determinada funcionalidade. (No *Digital Works* todos os circuitos são guardados sob a forma de macro, em ficheiros do tipo “*.dwm”)

Pode acontecer que seja necessário um circuito que vai ser usado como um bloco constituinte de projectos mais complexos. Nesse caso o bloco pode ser transformado numa macro a qual pode depois ser utilizada quantas vezes se quiser em projectos diferentes. Uma vez criada a macro ela “esconde” os seus pormenores internos, apresentando ao utilizador apenas as entradas e saídas, cujo comportamento depende da forma como foi definida¹. O mecanismo de criação de macros (uma das características mais poderosas do *Digital Works*) permite estender o conjunto de componentes de que o programa dispõe. Por exemplo, é possível definir novos circuitos integrados obtendo as suas funcionalidades a partir dos catálogos dos fabricantes e reproduzindo-as através de uma macro. A macro apresenta-se ao utilizador sob a forma dada pelo *template* que lhe está associado.

Criar uma macro é pois construir um circuito lógico e atribuir-lhe uma forma (*template*). De seguida será criada uma macro correspondente ao circuito do LPB visto atrás. A criação será feita através de um conjunto de passos executados em sequência.

6.1 – Criar o circuito lógico que será transformado em macro

O circuito lógico do LPB é o do parágrafo 4. Uma vez o circuito construído, deve verificar-se se o seu comportamento é correcto, nomeadamente se obedece à tabela de verdade que lhe deu origem. Uma vez verificada essa correcção, devem retirar-se todos os elementos das entradas (neste caso os três “Interactive Input” correspondentes a I, C e R) bem como todos os elementos das saídas (neste caso o LED correspondente ao LPB), substituindo-se cada um deles por uma “Macro Tag”.

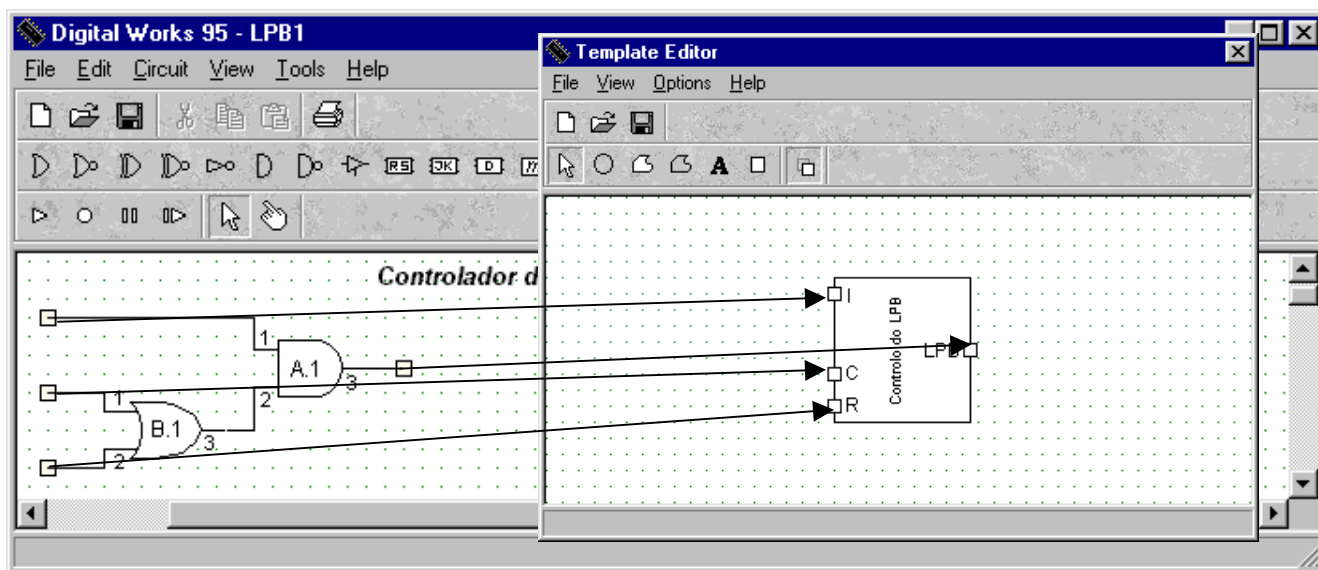


¹ É um conceito idêntico ao de função em programação; uma função apresenta um determinado comportamento cujos detalhes de implementação ficam escondidos ou encapsulados sob o nome da função.

6.2 – Associar o circuito ao template

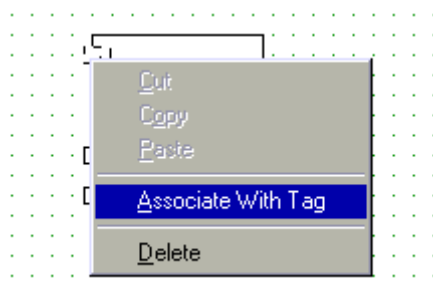
Neste passo vai associar-se cada uma das entradas e saídas do circuito com os pinos do *template* correspondente. Essa associação pode ser feita a um *template* já criado previamente (como o LPB.dwt) ou a um *template* novo, criado propositadamente para este circuito. Se a opção for por um *template* já criado, este deve ser aberto seleccionando-o da lista de *templates* disponíveis; se a opção for criar um *template* novo, este deve ser criado de acordo com o exposto na alínea 5. Em qualquer dos modos deve ser activado o editor de *templates*.

Neste exemplo a opção é por utilizar o *template* “LPB.dwt” criado anteriormente. A figura seguinte mostra as associações a estabelecer.



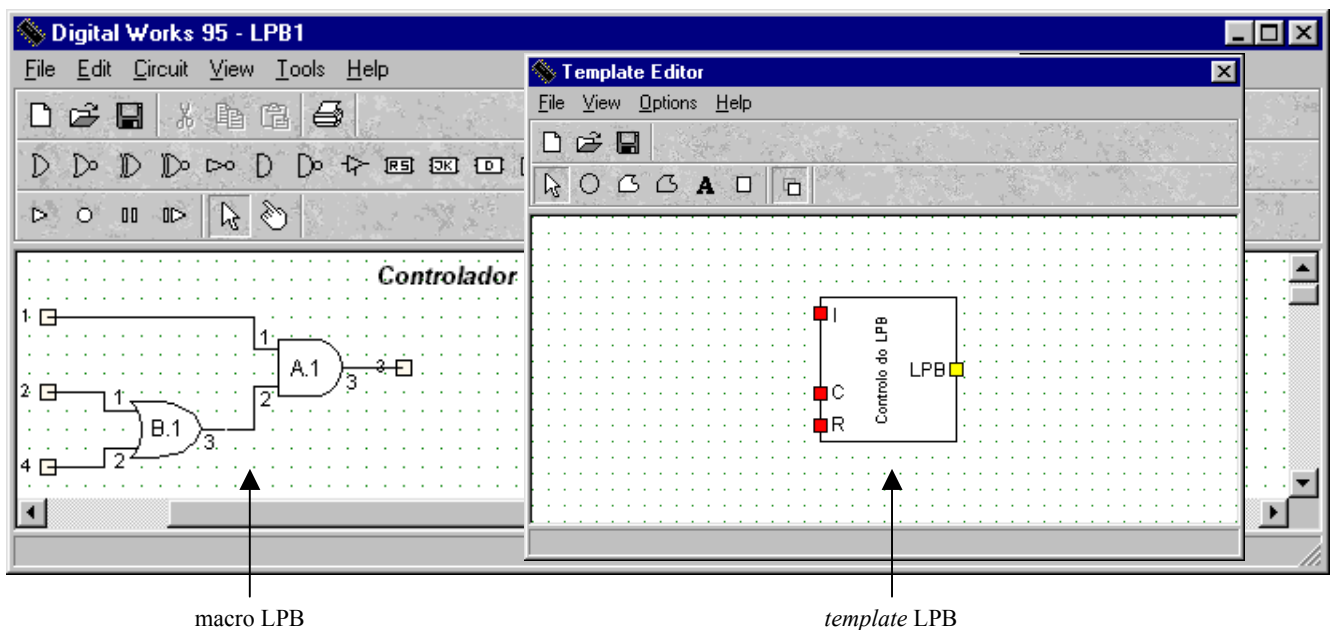
6.2.1 – Associar cada uma das entrada/saídas do circuito com um pino do template

Este processo deve executar-se com cuidado, percorrendo ordenadamente todas as entradas e saídas do circuito de modo a evitar enganos. Começando pela entrada I do circuito e clicando com o botão direito do rato sobre a correspondente “Macro Tag”, pode activar-se o “Template Editor”. Se ainda não tinha sido aberto um *template*, deve abrir-se agora o “LPB.dwt”. Como se começou pela entrada I deve agora clicar-se com o botão direito do rato sobre o correspondente pino (marcado com I) no editor de *templates*. Isso abre um menu *pull-down* com a opção “Associate With Tag” a qual deve ser activada para completar a associação entre a entrada I do circuito e o pino I do *template*.



Neste momento o pino I do *template* passa a amarelo e aparece o número 1 junto à entrada I do circuito lógico, indicando que foi feita a associação. A seguir deve fechar-se a janela do editor de *templates*.

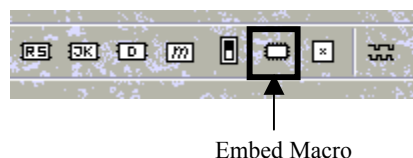
Repete-se o processo, clicando com o botão direito do rato sobre a “Macro Tag” C do circuito lógico de modo a activar de novo o editor de *templates* (quando este aparece o pino I já se encontra marcado a vermelho, indicando que já está associado e que não pode ser associado de novo). Clicando agora com o botão do rato sobre o pino correspondente do *template*, marcado com C, e depois sobre “Associate With Tag” do menu *pull-down*, o pino passa a amarelo e a associação fica completa. Fechar de novo o editor de *templates* e repetir o processo para a entrada R e a saída LPB (neste processo entradas e saídas são tratadas de igual modo). Após o processo completo obtém-se a seguinte situação:



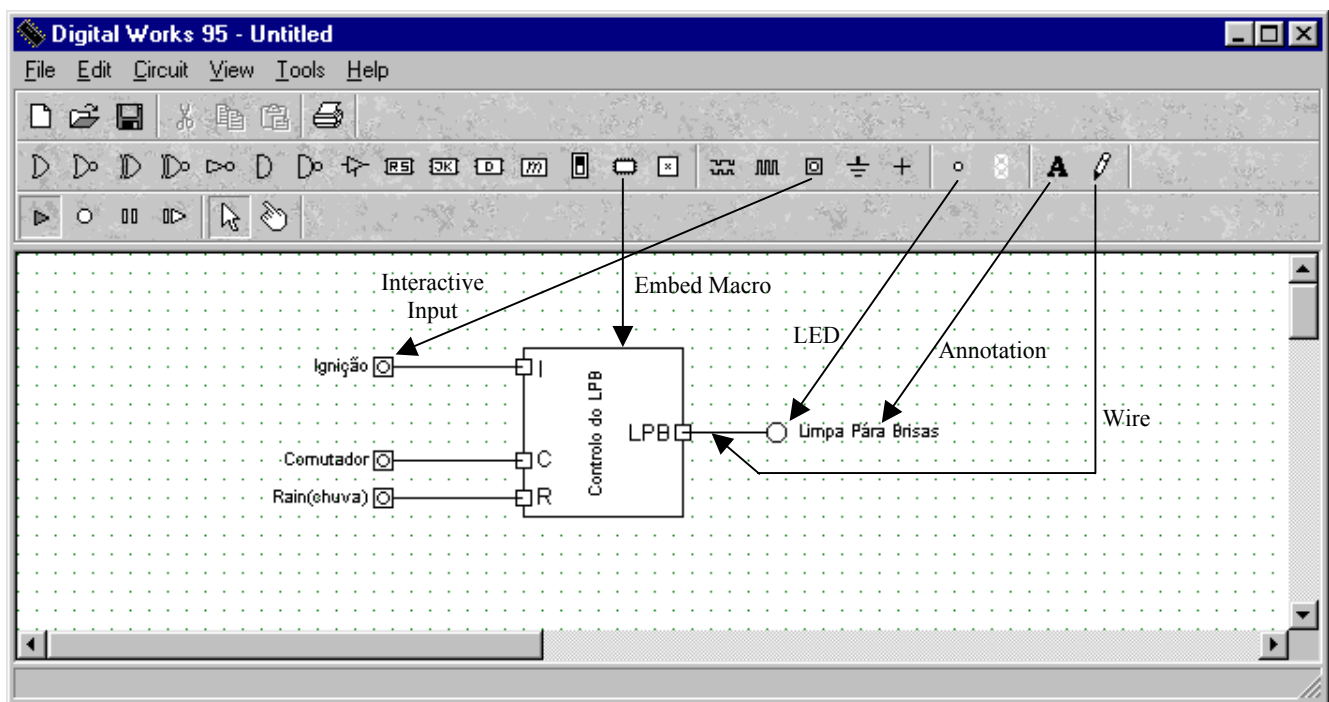
Uma vez fechado o editor de *templates*, o circuito lógico deve ser guardado com um nome apropriado (neste caso “LPB.dwm”). Está assim completa a criação da macro LPB.

6.4 –Utilização da macro

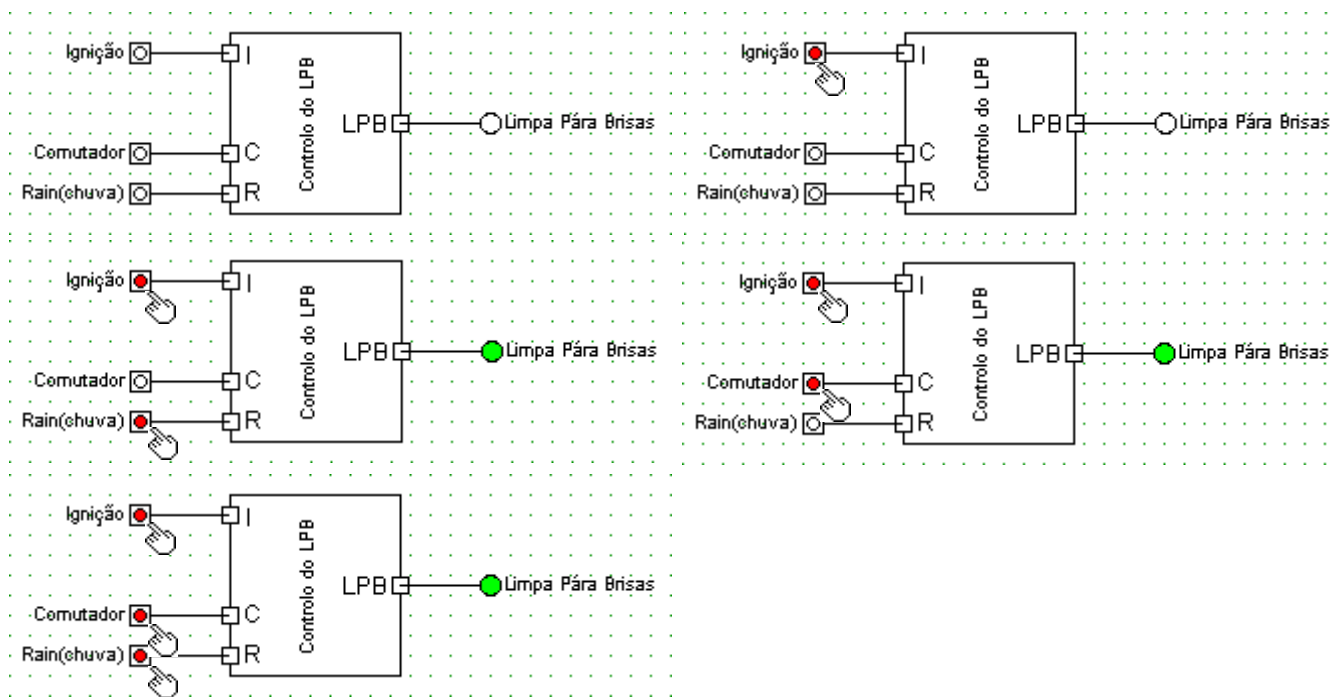
Para testar o funcionamento da macro vai executar-se a simulação do circuito do limpa-pára-brisas feita na alínea 4.2 mas usando a macro acabada de criar. A inclusão de macros nos circuitos é feita recorrendo ao botão “Embed Macro”:



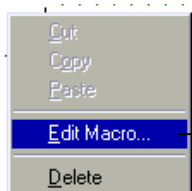
Uma vez seleccionado o botão abre-se uma janela que permite seleccionar a macro a incluir. Neste caso deve seleccionar-se “LPB.dwt”. Uma vez carregada a macro devem acrescentar-se os elementos de entrada e saída idênticos aos utilizados anteriormente. A figura seguinte mostra o aspecto final do circuito.



A partir daqui pode lançar-se a simulação e interagir com as entradas tal como anteriormente. As figuras seguintes mostram os diferentes casos possíveis verificando-se que o circuito se comporta de acordo com a tabela do limpa-pára-brisas. O bloco (macro) designado por “Controlo do LPB” pode agora ser usado na construção de circuitos mais complexos.



Clicando com o botão direito do rato sobre uma macro abre-se um menu *pull-down* com a opção “Edit Macro” que permite ver e editar o conteúdo da macro (modo Maximised View). Para fechar este modo e voltar à representação macro deve clicar no botão “Close Macro” (modo Minimised View).



Maximised View



Minimised View