

Guião de Laboratório de Arquitectura de Computadores

Simulação 2.1 – Portas Lógicas

1 – Objectivos

Com este trabalho pretende-se que o leitor se familiarize com a interface do simulador, com um conjunto de circuitos exemplificativos.

A melhor forma de aprender o funcionamento de qualquer sistema é fazer experiências com ele. Existem implementações físicas das portas lógicas, em circuitos integrados, mas nem sempre existem as condições apropriadas para usar uma implementação física.

Felizmente, existem simuladores (em computador) que permitem reproduzir de forma muito realista o comportamento destes sistemas. Todo este livro está baseado nesta perspectiva. Em vez de se limitar a descrever as diversas técnicas, o livro está recheado de referências a guiões práticos (como esta Simulação 2.1) que permitem ao leitor experimentar, na prática, os sistemas e técnicas que acabou de descobrir.

O Apêndice C contém uma breve introdução às capacidades do simulador.

Esta simulação inclui os seguintes aspectos:

- Introdução ao simulador, com construção (no simulador) dos pequenos circuitos de teste das portas lógicas;
- Utilização de interruptores como dispositivos de entrada (produzindo 0 ou 1, bastando um clique com o rato para mudar o valor) e LEDs para visualizar o estado das saídas;
- Verificação da tabela de verdade das várias portas lógicas, com várias entradas;
- Verificação de algumas equivalências (por exemplo, uma porta NAND e o seu equivalente, uma porta AND ligada a uma porta NOT);
- Verificação do tempo de atraso das portas lógicas.

2 – O Simulador

O simulador permite testar o funcionamento de sistemas digitais e de computadores, permitindo fazer experiências sem o ónus da montagem física dos circuitos e em qualquer lugar.

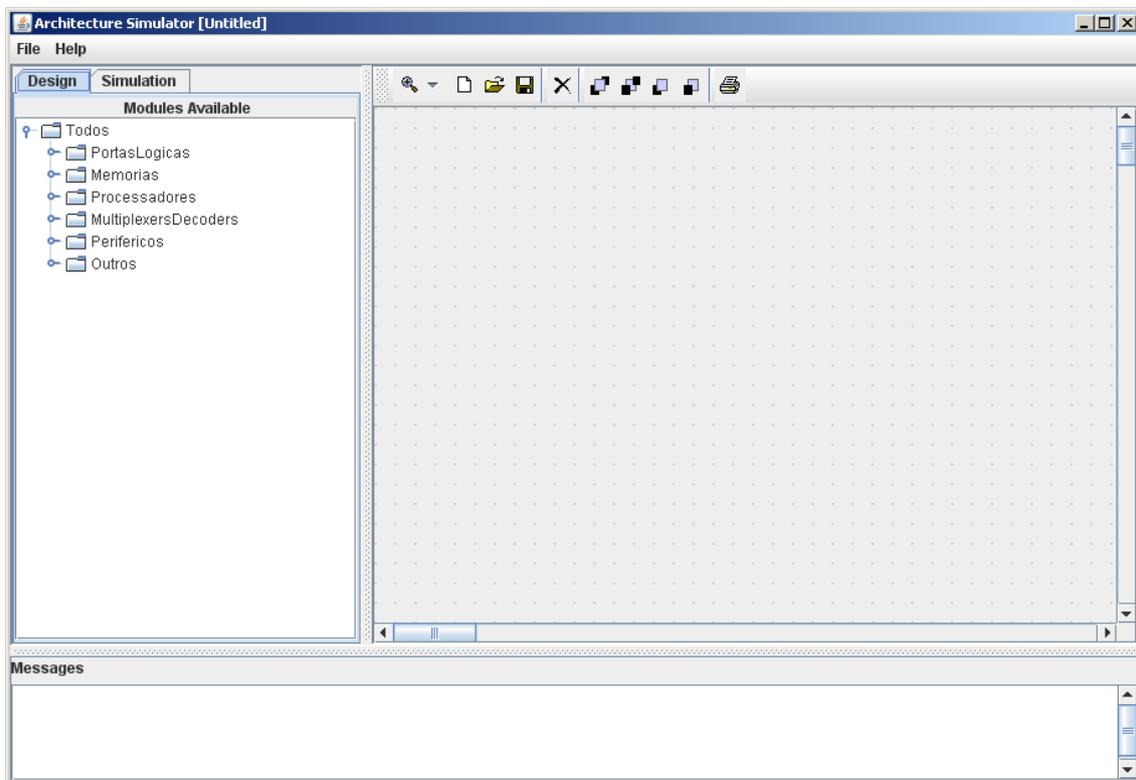
Esse simulador que pode ser executado em qualquer plataforma para a qual exista uma Java Virtual Machine (JVM). O *site* inclui todas as informações para poder instalar e executar o simulador.

O manual do simulador explica as operações necessárias para realizar as simulações, embora nesta simulação, dado que é a primeira, sejam exemplificados os passos fundamentais.

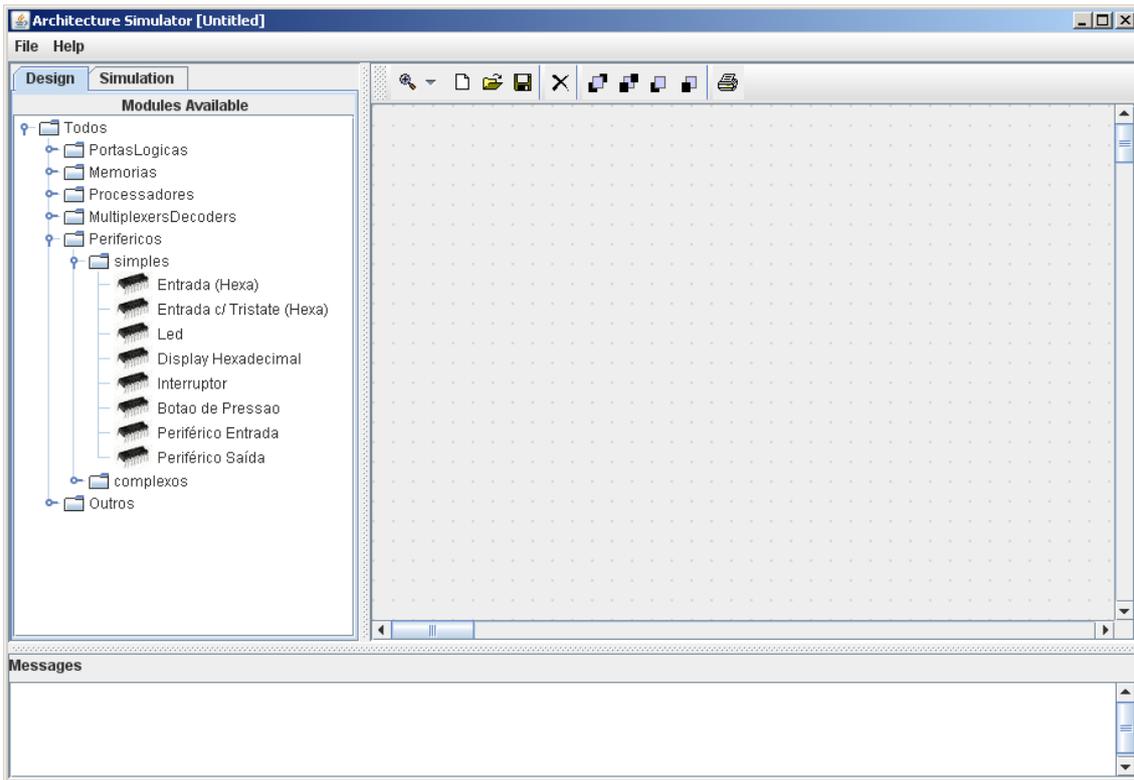
3 – Um circuito simples sem portas lógicas

3.1 – Criação do circuito

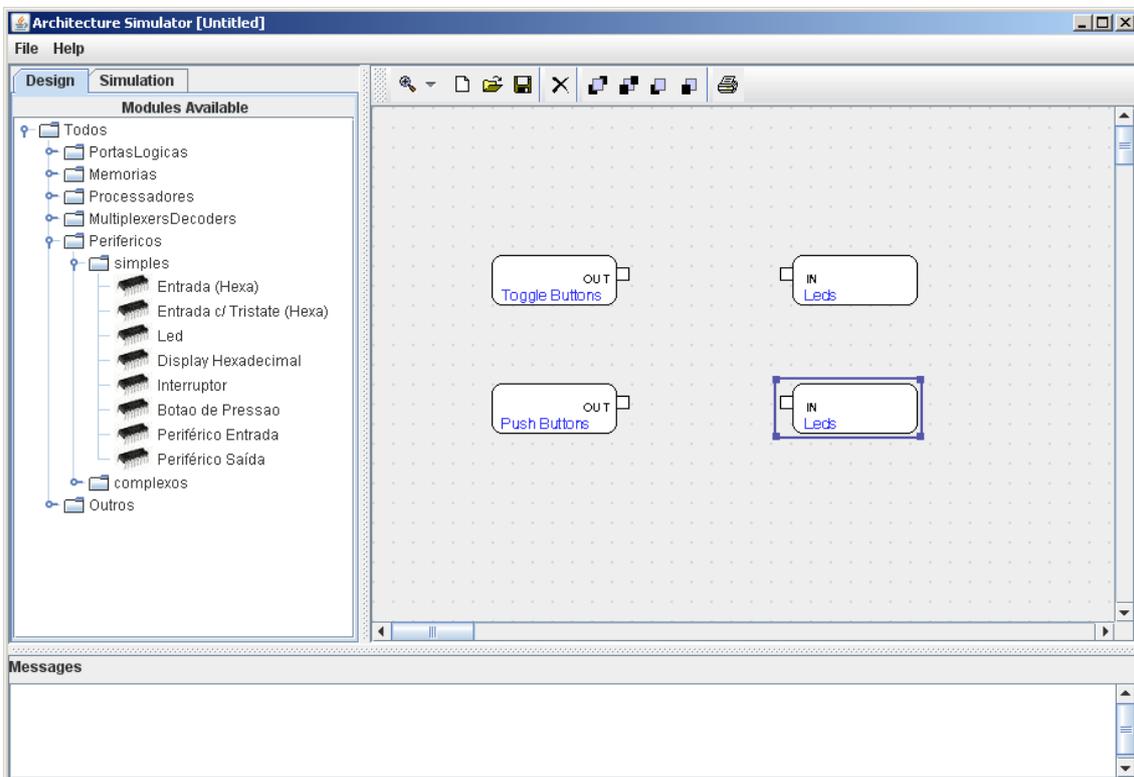
Execute o simulador, fazendo clique duplo sobre o ficheiro “simulador.jar”. Deverá aparecer, após o tempo necessário (alguns segundos) para iniciar a máquina Java, um écran como o seguinte. Se abrir com algum circuito que tenha aberto anteriormente, faça **New** no menu **File**.



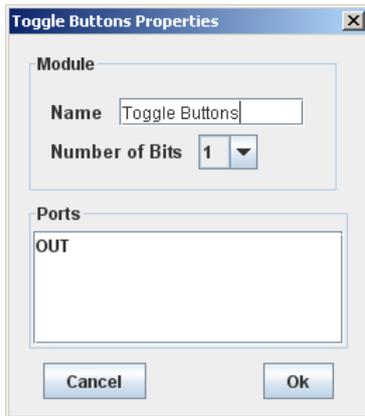
Faça clique em “Periféricos” e depois em “Simples”.



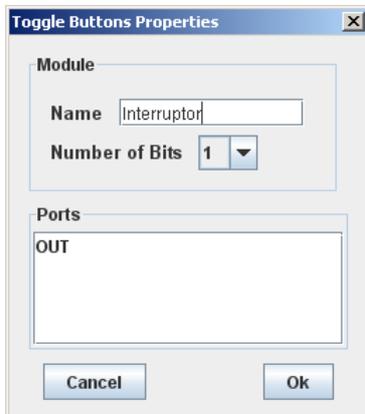
Clique em “Interruptor” e arraste para o espaço de trabalho, largando depois o botão do rato. Deve aparecer um objecto com o nome “Toggle Buttons”. Idem para “Botão de Pressão” e “Led” (este último, faça duas vezes). No final, terá algo como a janela seguinte (os objectos podem ser arrumados fazendo clique e arrastando).



Para mudar o nome do objecto “Toggle Buttons”, faça duplo clique nele, o que faz aparecer a sua janela de configuração:

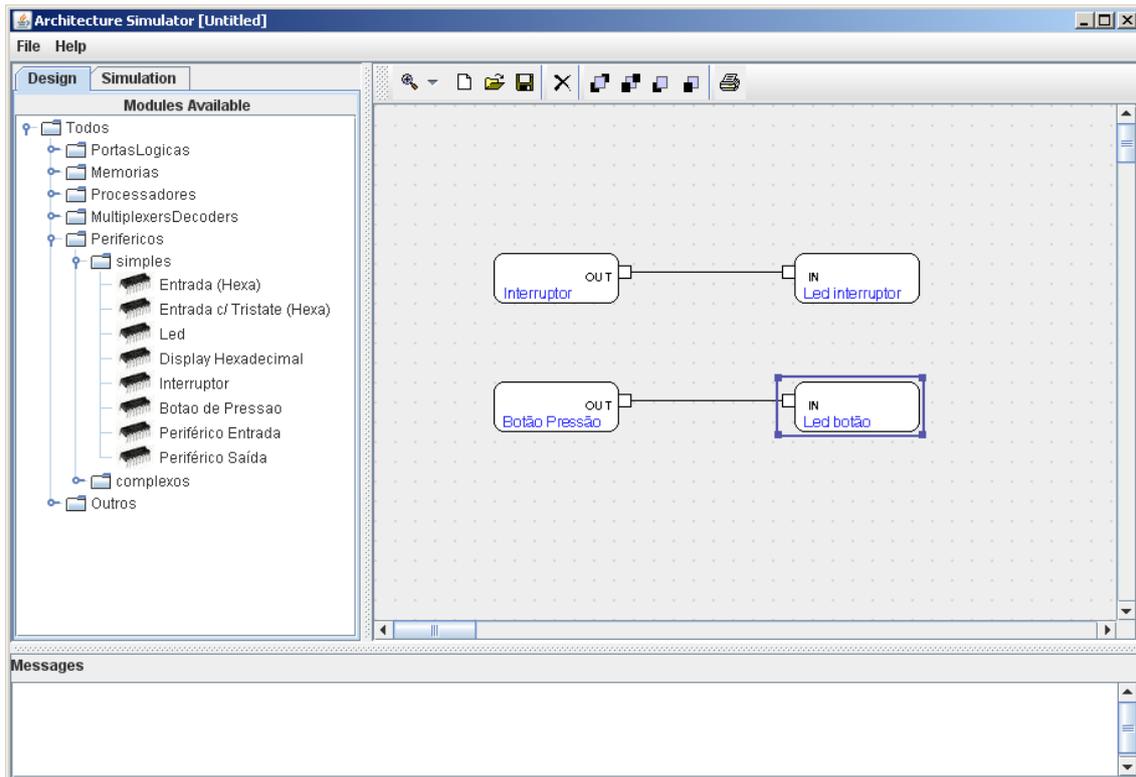


Mude-lhe o nome para “interruptor”.



Mude também o nome dos outros objectos, para serem mais facilmente identificáveis.

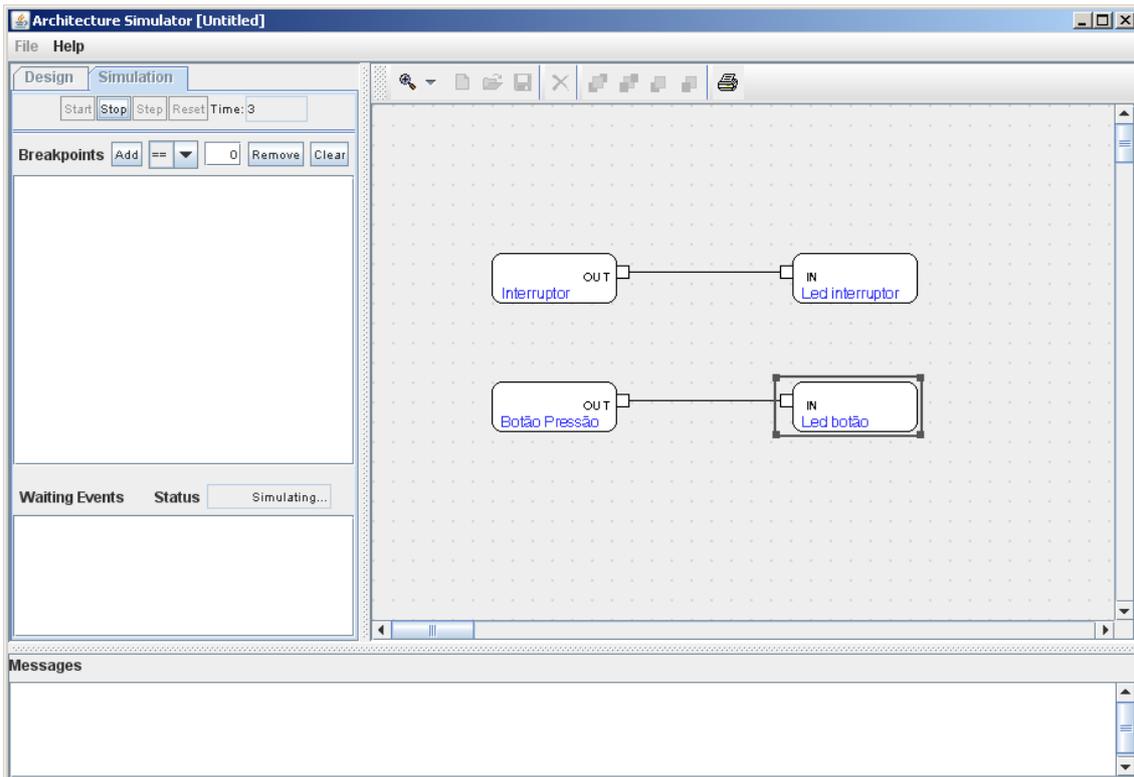
Para ligar o interruptor e o botão de pressão ao seu led respectivo, clique no pino OUT e arraste até ao pino IN do Led.



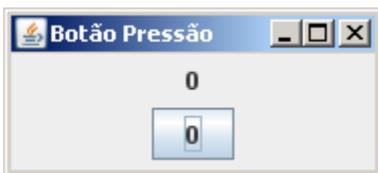
Se o desejar, pode guardar este circuito com **File** e **Save**. Pode mais tarde recuperá-lo com **File** e **Load**.

3.2 – Simulação do circuito

O circuito está construído. Passemos agora para “Simulação” (por baixo da barra de menu) e carregue em “Start”.



A seguir, precisamos de abrir as janelas de interface destes quatro objectos, para os podermos controlar/visualizar. Tal é feito com duplo clique sobre cada objecto. Alargue cada uma das janelas até o título estar visível. Pode deslocar as janelas para onde quiser no écran.



Agora faça clique no botão com “0” dentro da janela do interruptor. Esta muda para 1 (novo valor à saída do interruptor) e o Led fica aceso. O número “0” que se mantém indica apenas o número do bit (cada um destes objectos pode ter vários bits, e neste exemplo têm apenas um, correspondente ao bit 0).



Faça o mesmo para o botão de pressão. Note que neste caso, ao contrário do que sucede com o interruptor) o Led apaga-se quando se larga o botão. Há assim forma de gerar um

sinal que alterna entre 0 ou 1 de cada vez que se carrega no botão do rato (interruptor) ou um sinal que está a 1 apenas enquanto se está a carregar no botão do rato (botão de pressão).

Verifique que, em simulação, as linhas estão a preto quando têm o valor 0 e a vermelho quando têm o valor 1 (NOTA – isto só funciona em ligações de um bit, como é o caso).

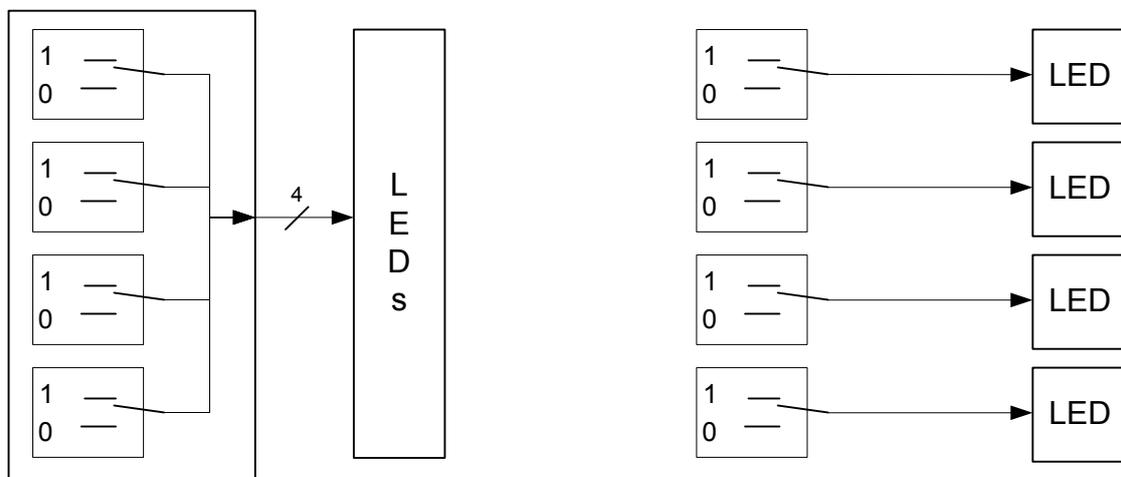
A figura seguinte esquematiza a ligação quer do interruptor quer do botão a um led, mas com esta diferença de comportamento.



3.3 – Interruptores e Leds de vários bits

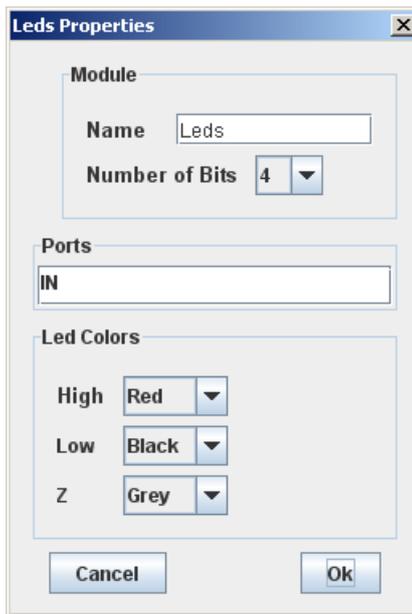
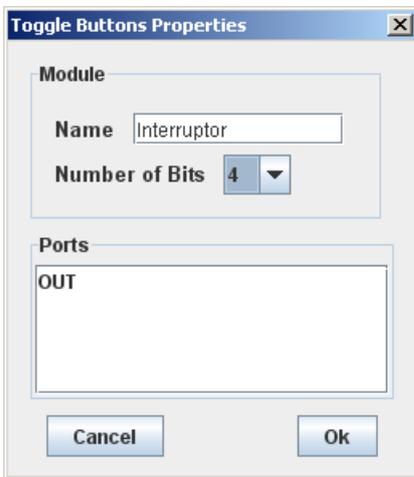
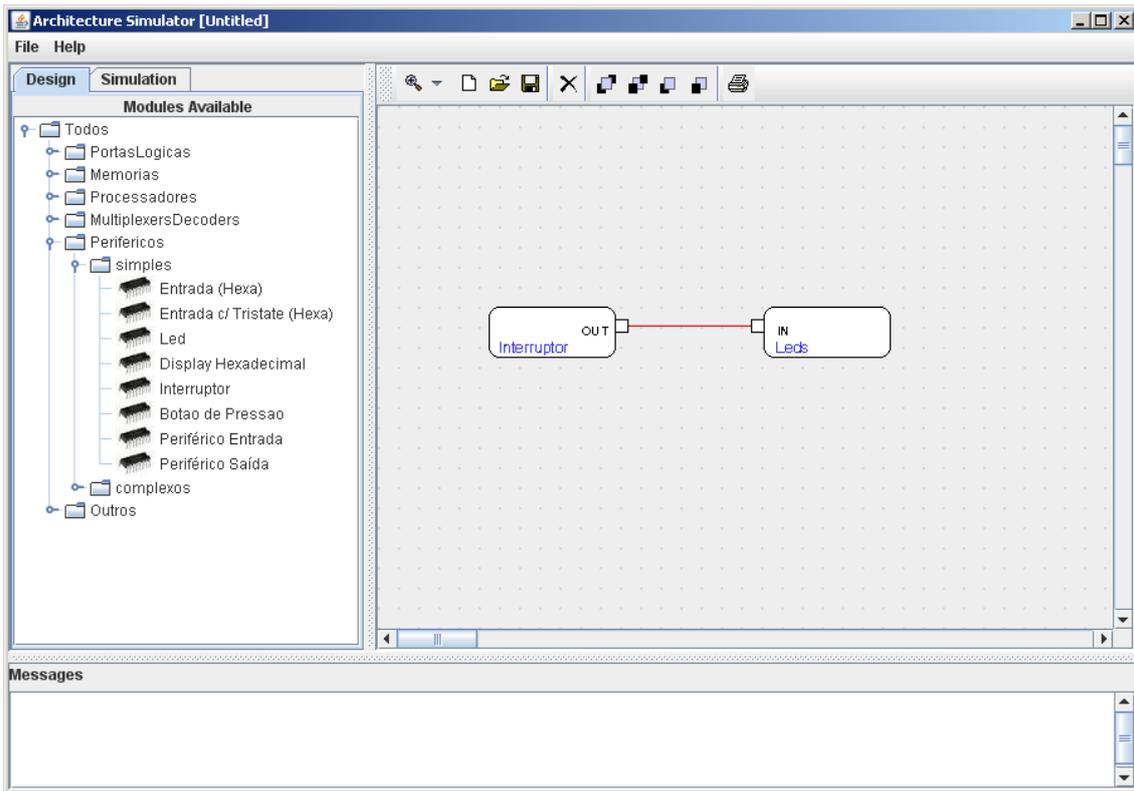
Tanto os interruptores como os Leds (e muitos outros módulos) podem ser configurados para ter vários bits. Por exemplo, a figura seguinte representa um interruptor de 4 bits ligado a um led também de 4 bits. A ligação entre os dois fica automaticamente com 4 bits. Do lado direito da figura aparece um circuito funcionalmente equivalente, com quatro interruptores e quatro leds individuais.

Qualquer dos circuitos se pode simular, mas o circuito do lado esquerdo tem apenas duas janelas de controlo (uma com quatro botões e outra com quatro leds), enquanto no circuito do lado direito se teriam de ter quatro janelas de interruptores e quatro de leds, todos independentes.

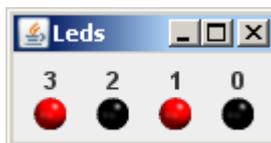


O circuito seguinte mostra um interruptor de 4 bits ligado a um Led de 4 bits. Em termos visuais não se distingue de um interruptor de 1 bit ligado a um led de 1 bit, mas fazendo duplo clique sobre o interruptor em modo Design faz abrir a janela de configuração do interruptor, permitindo configurá-lo para 4 bits. Idem para o Led (que permite ainda configurar as cores).

O ficheiro “int-led.cmod” já tem este circuito construído.

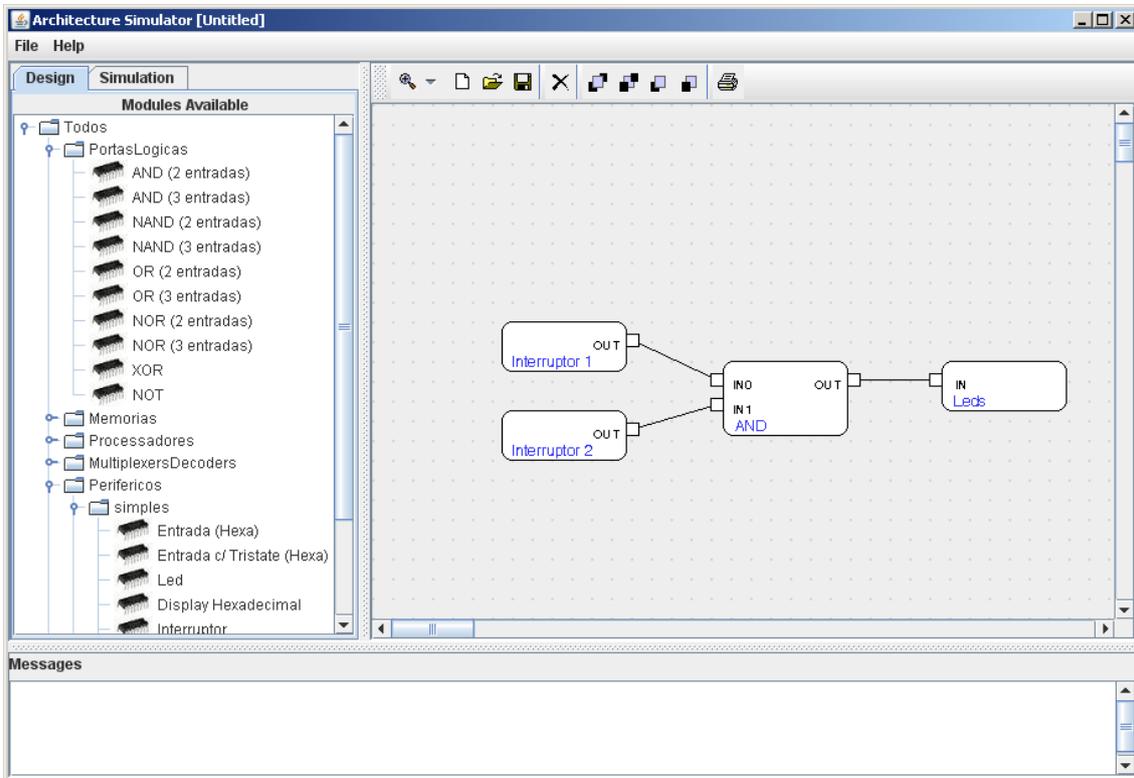


Passando depois para simulação, fazendo duplo clique no interruptor e no led faz abrir as janelas de controlo destes objectos, mostrando efectivamente que são de 4 bits. Carregando em cada um dos botões do interruptor, consegue-se controlar de forma independente dos restantes o led respectivo.



4 – Circuito com AND

Construa o circuito seguinte. O AND está disponível em “Portas lógicas”.



Passe para “Simulação”, carregue em “Start”, faça duplo clique nos interruptores e no Led e verifique que só quando ambos os interruptores estão a 1 é que o Led acende, tal como seria de esperar de um AND.

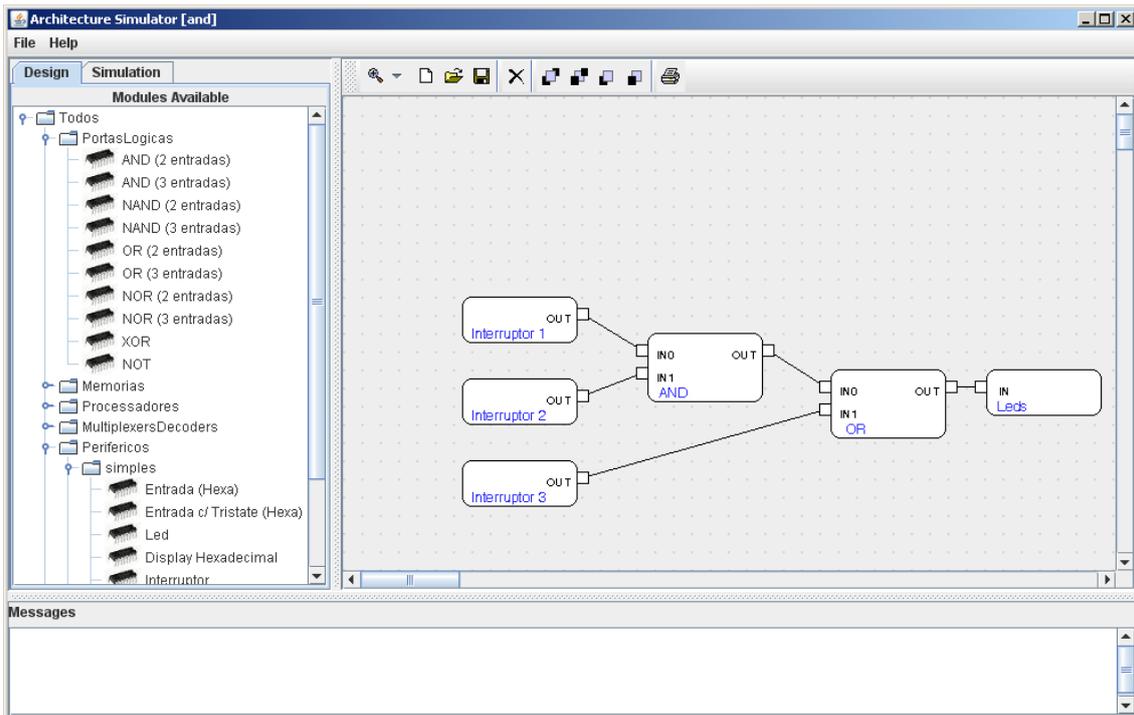
5 – Outros circuitos

Experimente também:

Construir outros circuitos usando outra das portas lógicas disponíveis (visíveis na figura anterior), incluindo portas de três entradas (terá de incluir outro interruptor).

A equivalência entre um NAND e um AND seguido de um NOT, verificando que ambos os circuitos se comportam de igual modo;

Incluir duas ou mais portas lógicas, como por exemplo o circuito seguinte. Para comodidade, este circuito está já disponível, através do ficheiro “and-or.cmod”.



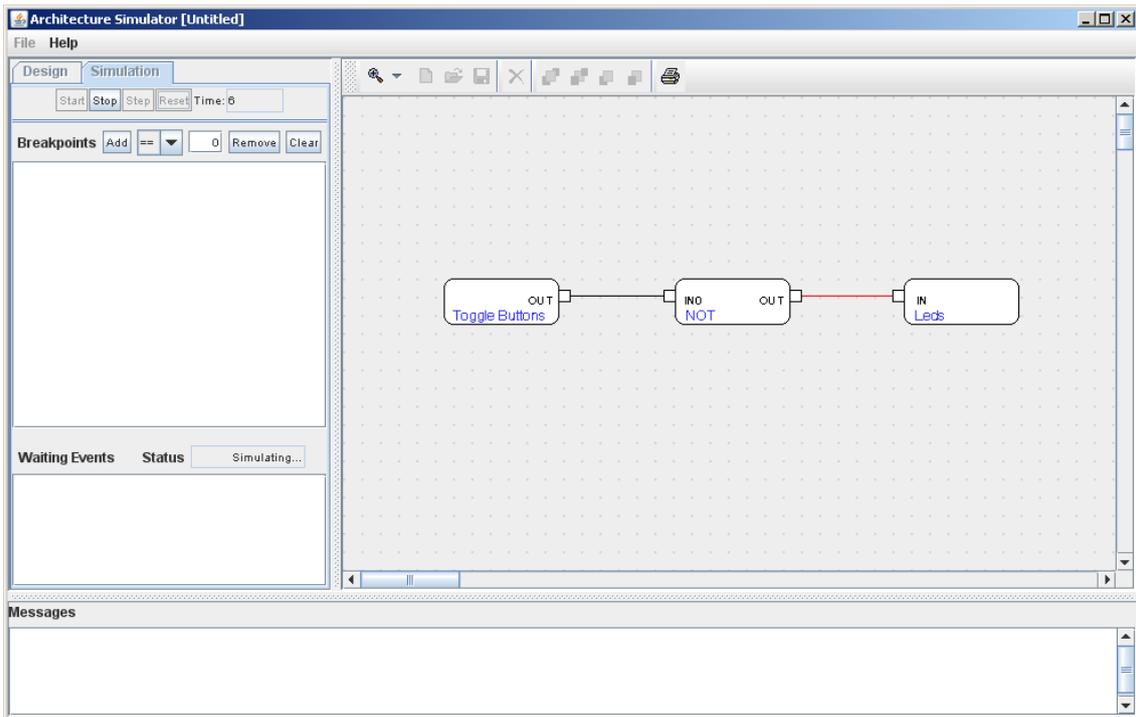
Verifique o funcionamento deste circuito (a sua tabela de verdade).

6 – Atraso das portas lógicas

O simulador permite simular o atraso inerente aos circuitos electrónicos (tempo que a saída demora a reagir às alterações nas entradas)

As portas lógicas têm um tempo de atraso pré-determinado de 3 unidades de tempo. Um unidade é o que se quiser, pois o tempo de simulação não é real. Embora o interruptor seja um dispositivo mecânico, terá um pequeno circuito electrónico associado (para produzir o 0 e o 1), pelo que por uniformização também tem um atraso de 3 unidades. O Led faz apenas visualização e assume-se que não tem atraso.

Construa o circuito seguinte, com um interruptor, um NOT e um Led. Quando passa para simulação, o tempo corrente é mostrado a seguir a “Time”.



De cada vez que se carrega e larga o interruptor, o tempo avança de 6 unidades. 3 unidades demora a saída do interruptor a reagir, e mais 3 demora o NOT a reagir.

