

# **Guião de Laboratório de Arquitectura de Computadores**

## **Simulação 2.2 – Circuitos combinatórios**

### **1 – Objectivos**

Esta simulação ilustra o funcionamento dos circuitos combinatórios, usando o circuito da Fig. 2.8 como base. Os aspectos cobertos incluem os seguintes:

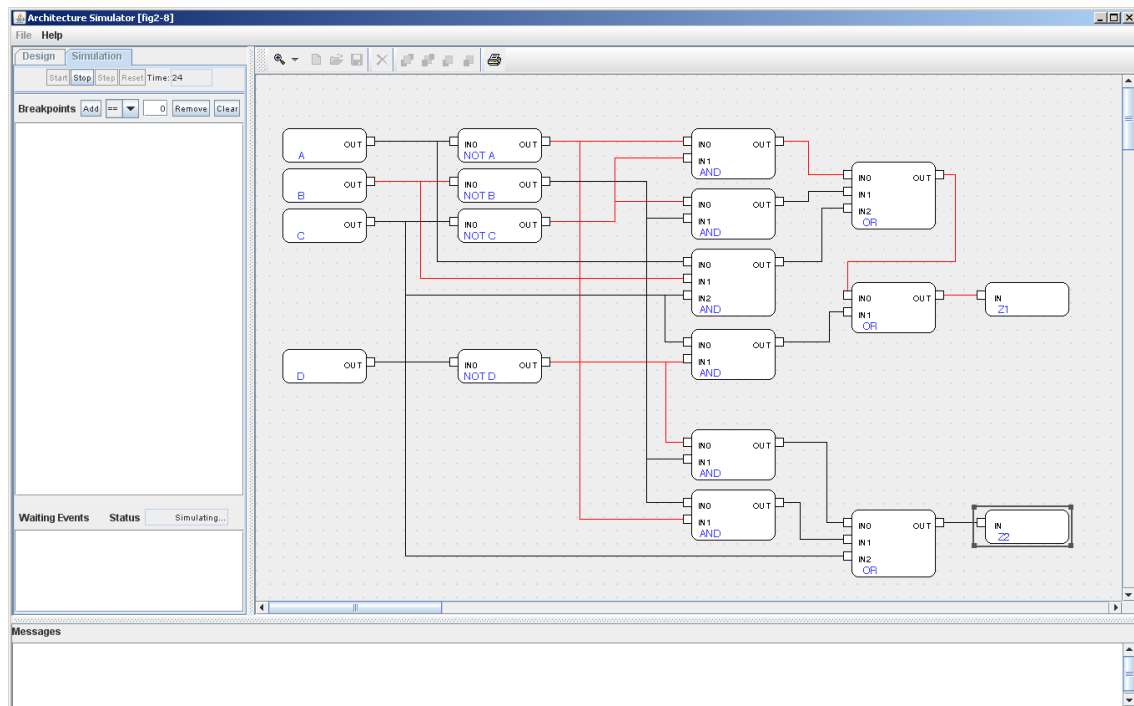
- Verificação do cumprimento da tabela de verdade da Tabela 2.4;
- Inspeção de valores das ligações intermédias;
- Observação dos picos causados pelo comportamento transitório dos valores das ligações enquanto decorrem as interações das mudanças de sinais;
- Verificação do tempo de atraso total do circuito.

### **2 – O circuito**

A figura seguinte mostra o circuito da Fig. 2.8 já no simulador (e já em Simulação). O OR de quatro entradas Z1 foi substituído por um de três entradas e outro de duas entradas, por não haver um de quatro entradas na paleta de objectos do simulador (embora este os suporte, mas é preciso configurar um ficheiro interno).

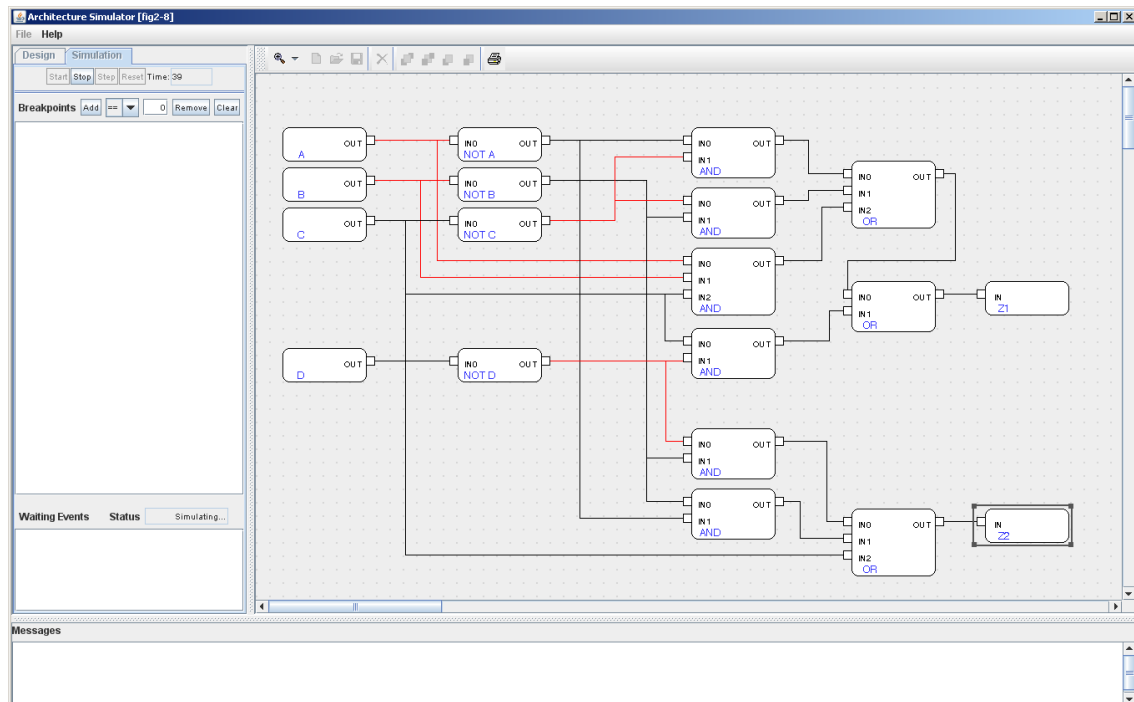
As entradas são feitas com interruptores e as saídas com leds. Este circuito está já disponível no ficheiro “fig2-8.cmod”.





Note que o tempo de simulação evoluiu 12 unidades. Como cada dispositivo tem um tempo de atraso de 3 unidades, isto quer dizer que houve quatro dispositivos em série que mudaram o seu valor. Pode verificar-se que foi apenas Z2 que mudou, e que para a alteração do valor do interruptor B lá chegar o sinal tem de atravessar quatro dispositivos, contando com o próprio interruptor, que também introduz atraso. Os leds, sendo apenas para visualização sem interferir com os resultados, consideram-se sem atraso.

Mude agora o interruptor A para 1 também.



Agora é Z1 que muda, e o tempo evolui mais 15 unidades. Isto deve-se ao facto de, para alterar Z1 a partir do interruptor A, o sinal ter de atravessar cinco dispositivos, a 3

unidades de atraso cada um. Não é o caso da fig. 2.8 (representada atrás), que tem apenas quatro, mas no circuito do simulador introduziu-se mais um OR de duas entradas em série com o de três entradas (para não ter de usar o OR de quatro entradas).

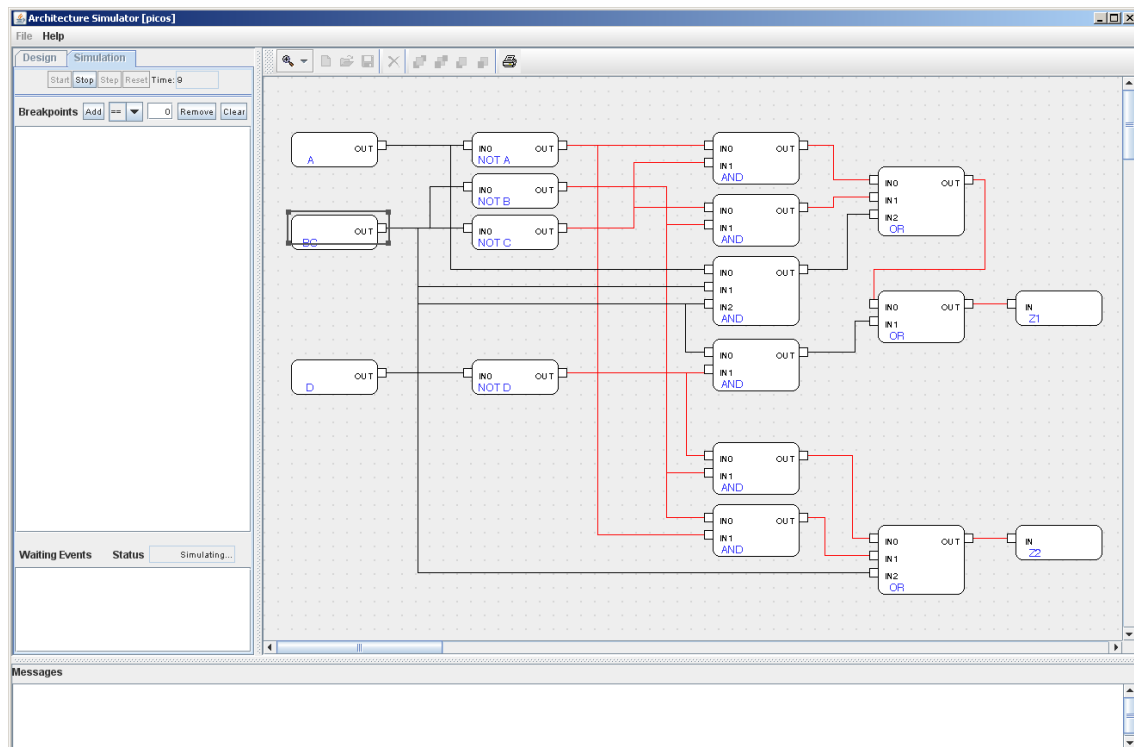
## 5 – Picos e transitórios na simulação

Este é um simulador baseado em eventos. Quando as entradas de um dado módulo (porta lógica, por exemplo) mudam, este é reavaliado para determinar os eventuais novos valores das saídas. Para simular os tempos de atraso, os novos valores são programados para um tempo de simulação futuro (no caso das portas lógicas, para 3 unidades de tempo depois).

No entanto, quando chegar o tempo desses novos valores aparecerem nas saídas, os módulos cujas entradas ligarem as essas saídas precisam também de ser reavaliados. O simulador entra assim em ciclo até não haver mais eventos, isto é, ter atingido uma situação estável.

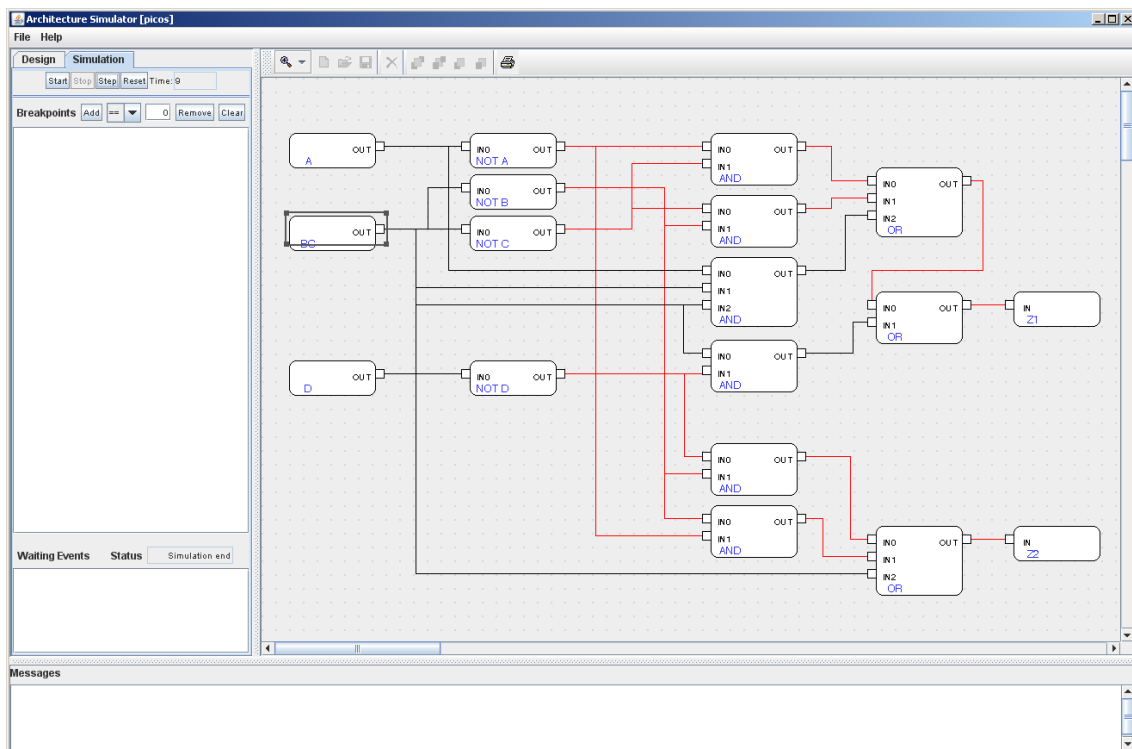
Na secção anterior, observou-se que o tempo evoluiu de 12 e de 15 unidades de tempo (o necessário para todas as alterações terem sido propagadas pelas diversas portas lógicas), mas não se viu o que sucedeu pelo meio. A experiência seguinte permite observar as mudanças de valores intermédias.

O circuito seguinte está contido no ficheiro “picos.cmod”. Os interruptores B e C foram fundidos num só (BC), pelo que as entradas B e C são agora a mesma. Carregue este ficheiro, passe para Simulação, carregue em Start e abra a janela de controlo do interruptor BC.





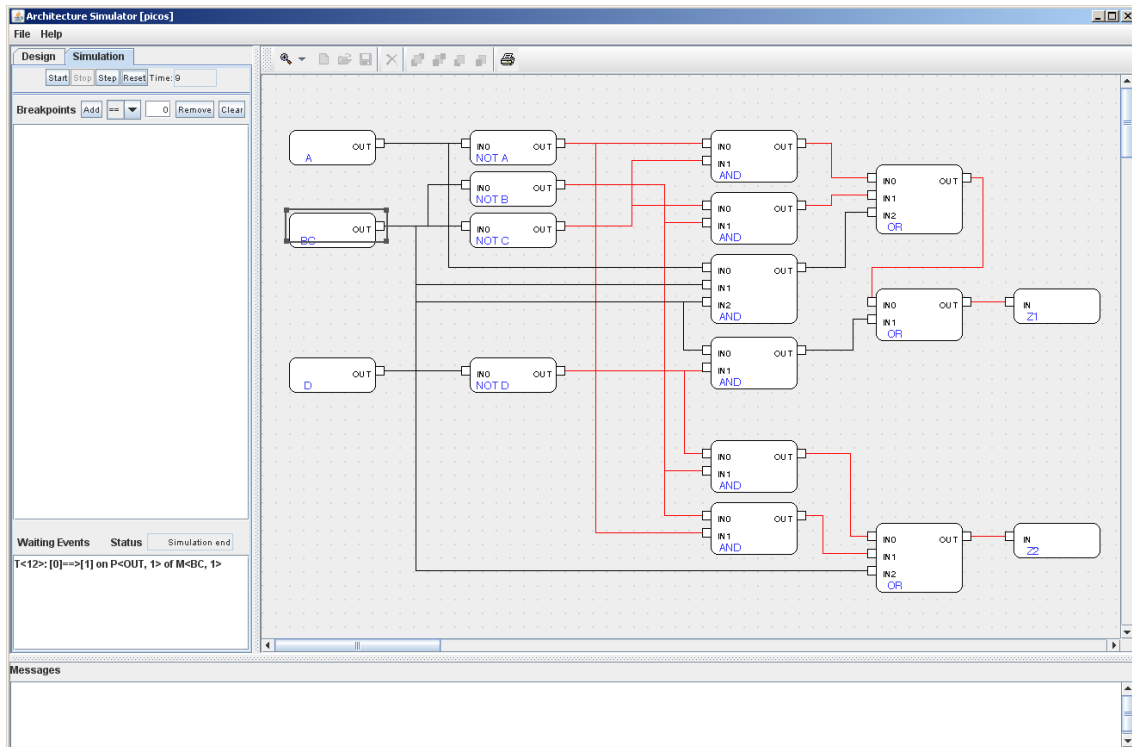
Carregue agora em Stop, o que faz parar a simulação.



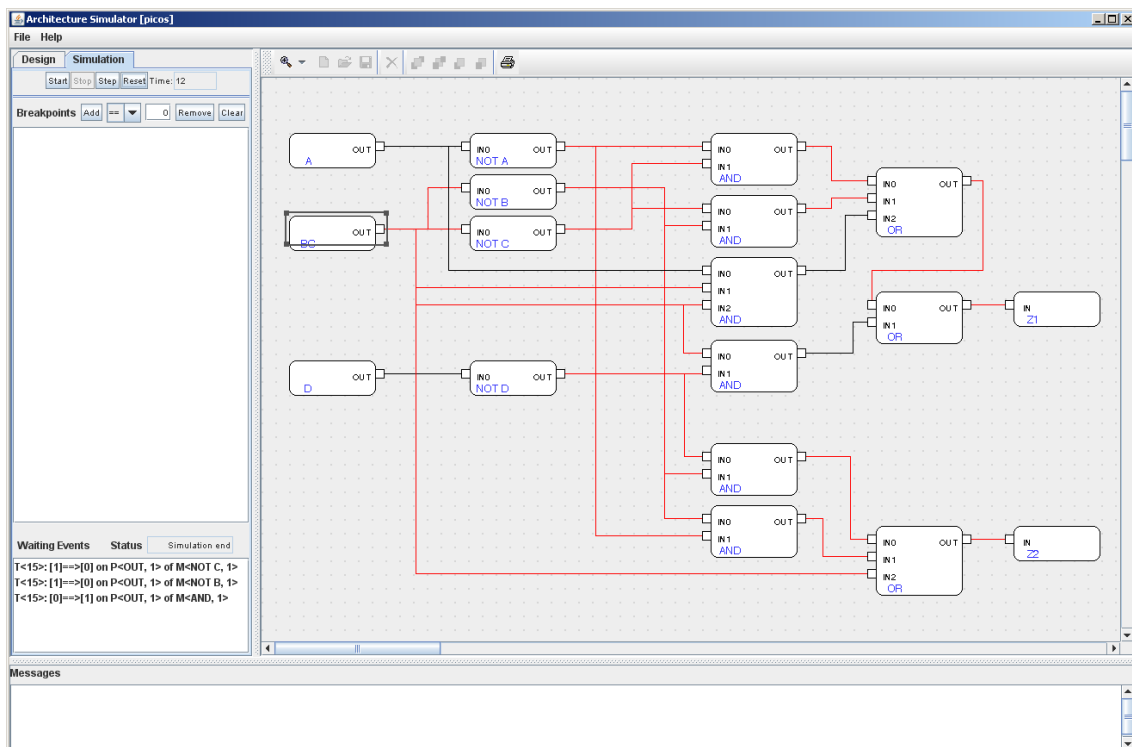
Carregue agora no interruptor BC, passando-o para 1.



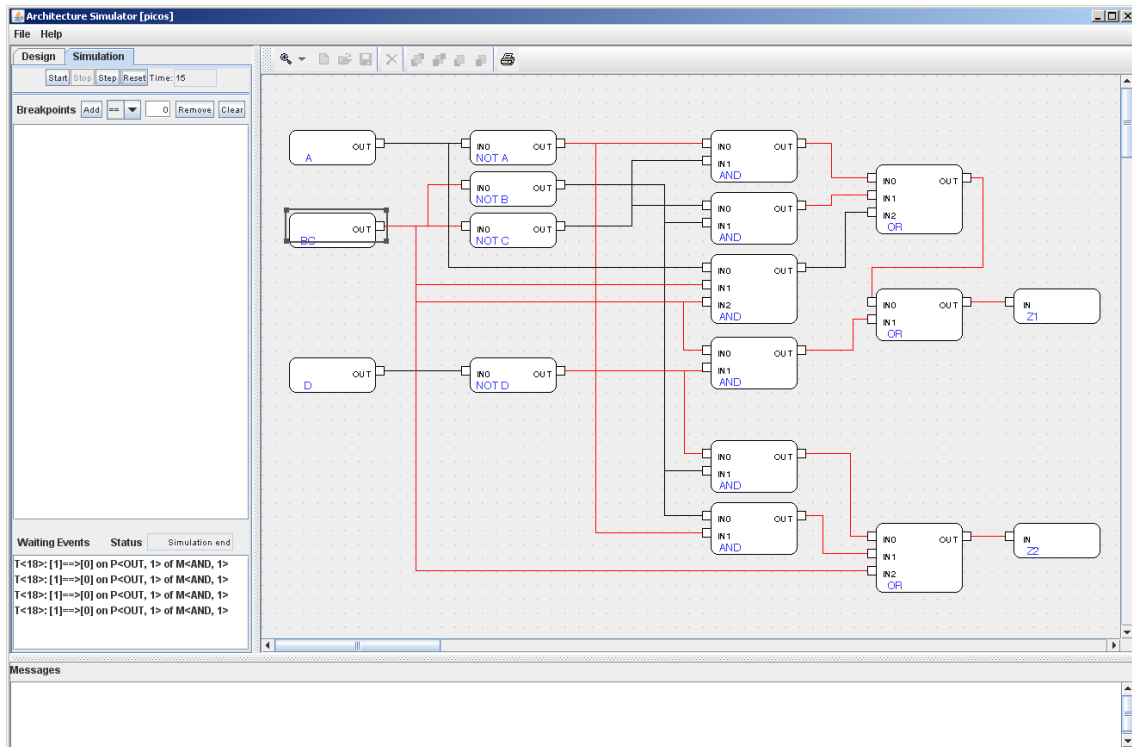
Como a simulação está parada, o circuito não reage. Carregue agora em Step. O simulador executa apenas uma iteração, isto é, programa uma alteração na saída do interruptor para daí a 3 unidades de tempo. Este evento futuro pode ser observado na janela “Waiting Events”.



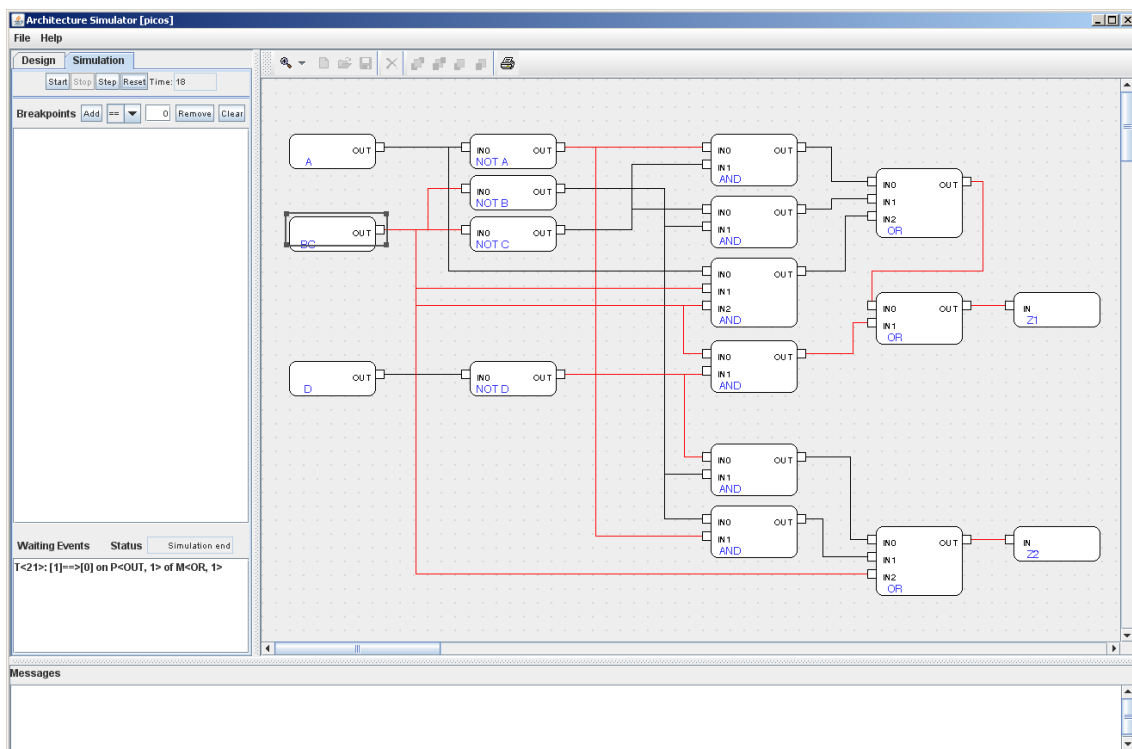
Carregue novamente em Step. O simulador avança 3 unidades de tempo, altera a saída do interruptor e reavalia as portas lógicas cujas entradas ligam ao interruptor BC (pois a saída deste mudou) e programa para um tempo futuro as alterações que houver nas saídas destas portas lógicas.



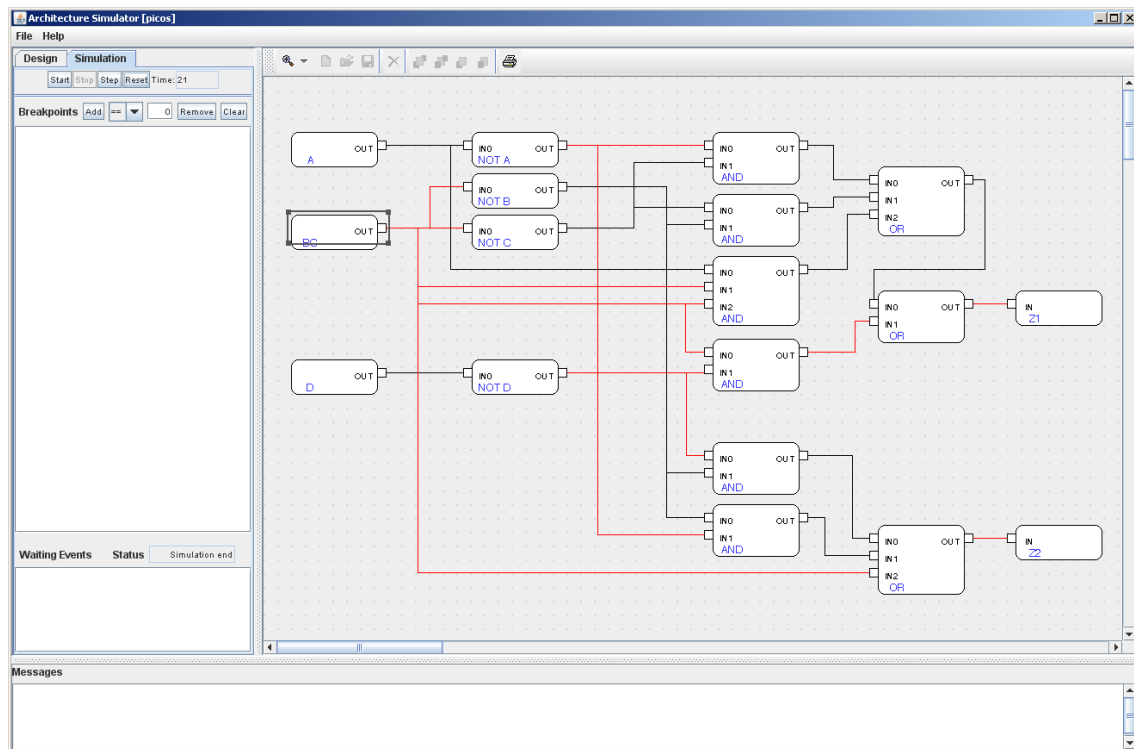
Carregue novamente em Step. As portas lógicas cujas entradas ligam às saídas das primeiras são reavaliadas e as alterações nas suas saídas são programadas.



Carregue novamente em Step, acontecendo nova iteração. O último evento programado é o do OR antes do led Z1.



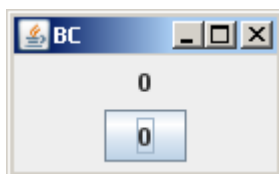
Carregue novamente em Step, o que executa o último evento programado mas não gera mais nenhum, pois atingiu-se uma situação estável.



Se carregar mais vezes em Step, nada sucede pois não há mais eventos para executar.

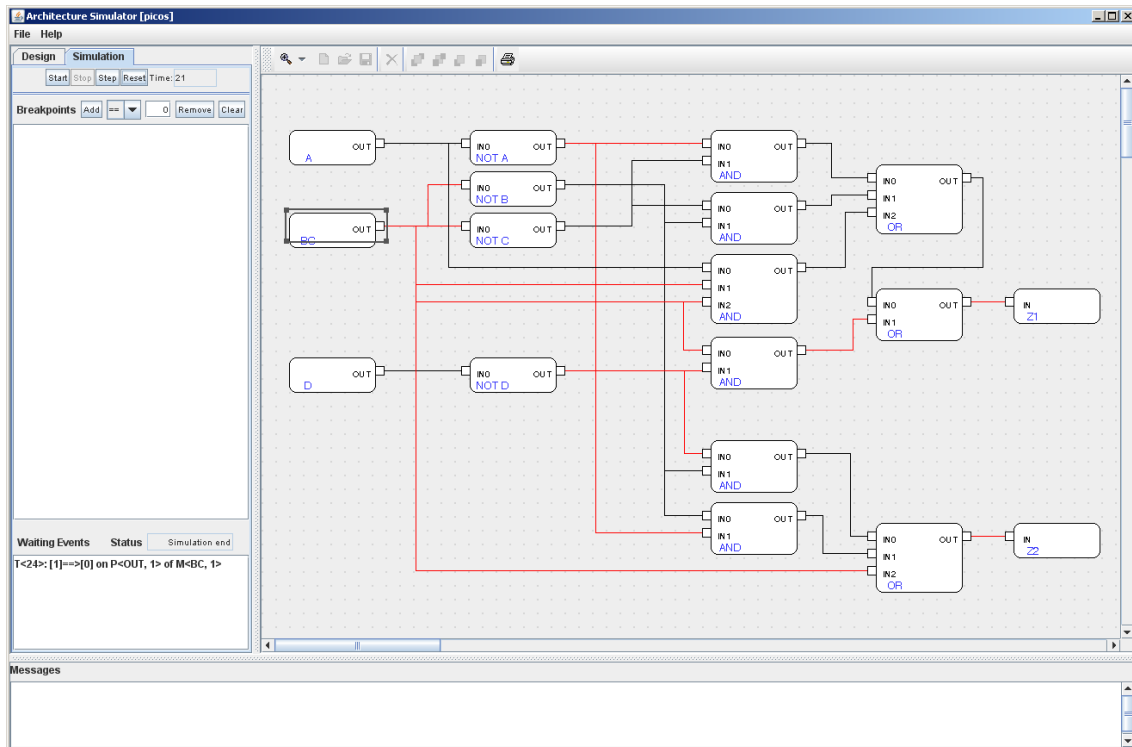
O tempo que tudo isto demora não depende do número de módulos em série, mas sim das alterações que é preciso propagar. Se um dado sinal não muda de valor, não é preciso reavaliar os módulos que a ele ligam.

Passa novamente o interruptor para 0.

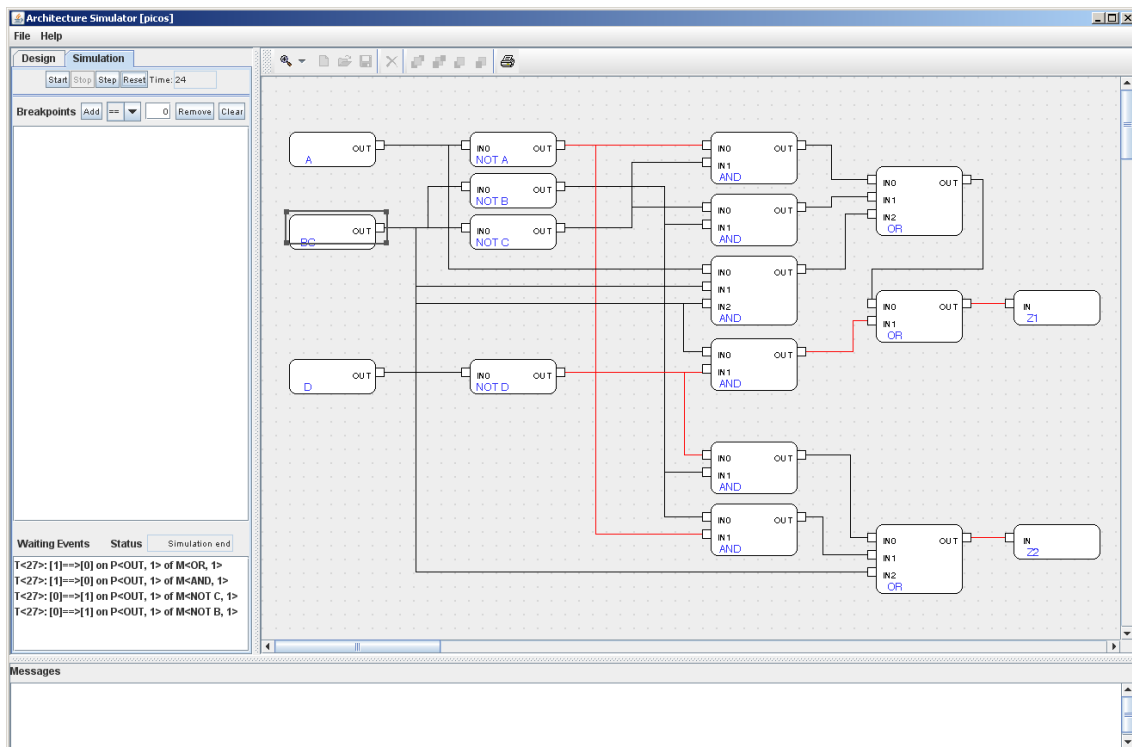


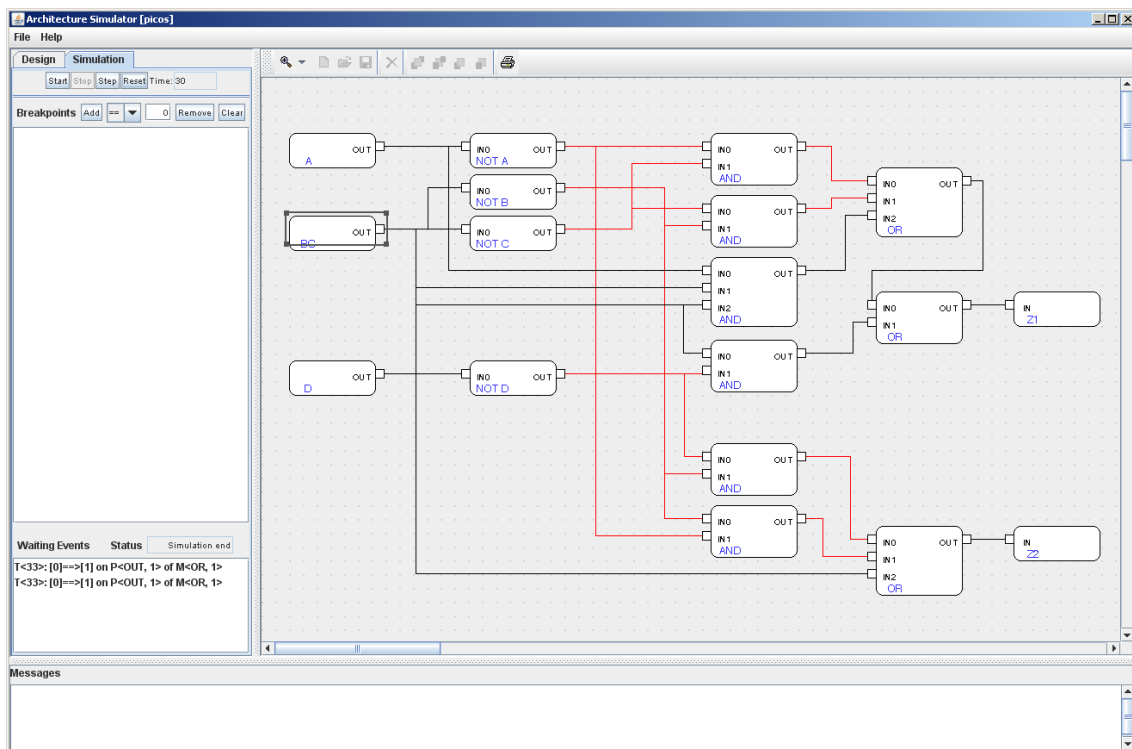
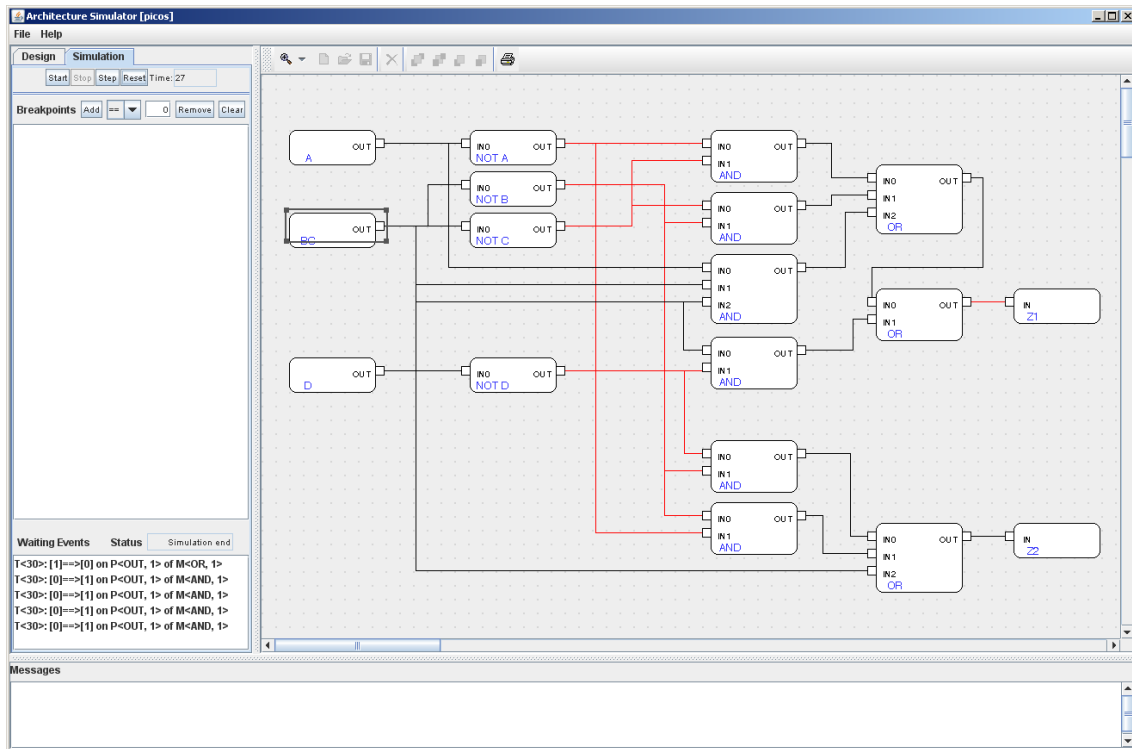
Carregue mais uma vez em Step. Novamente, surge o evento programado para alterar a saída do interruptor.

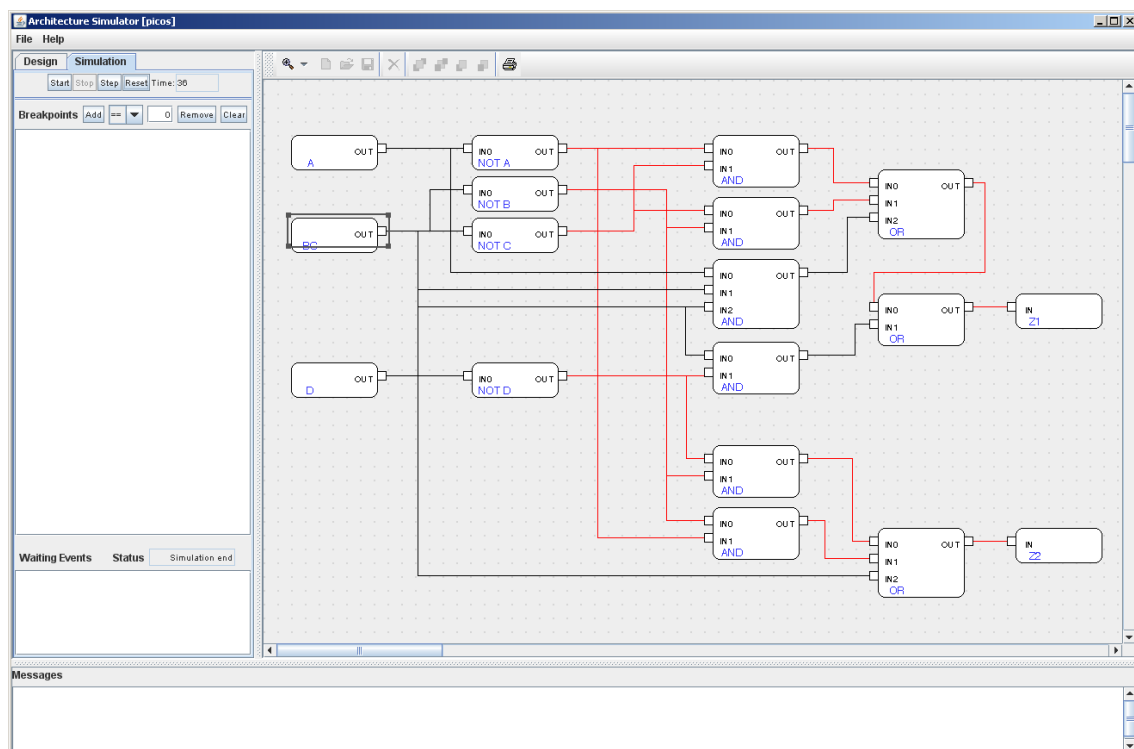
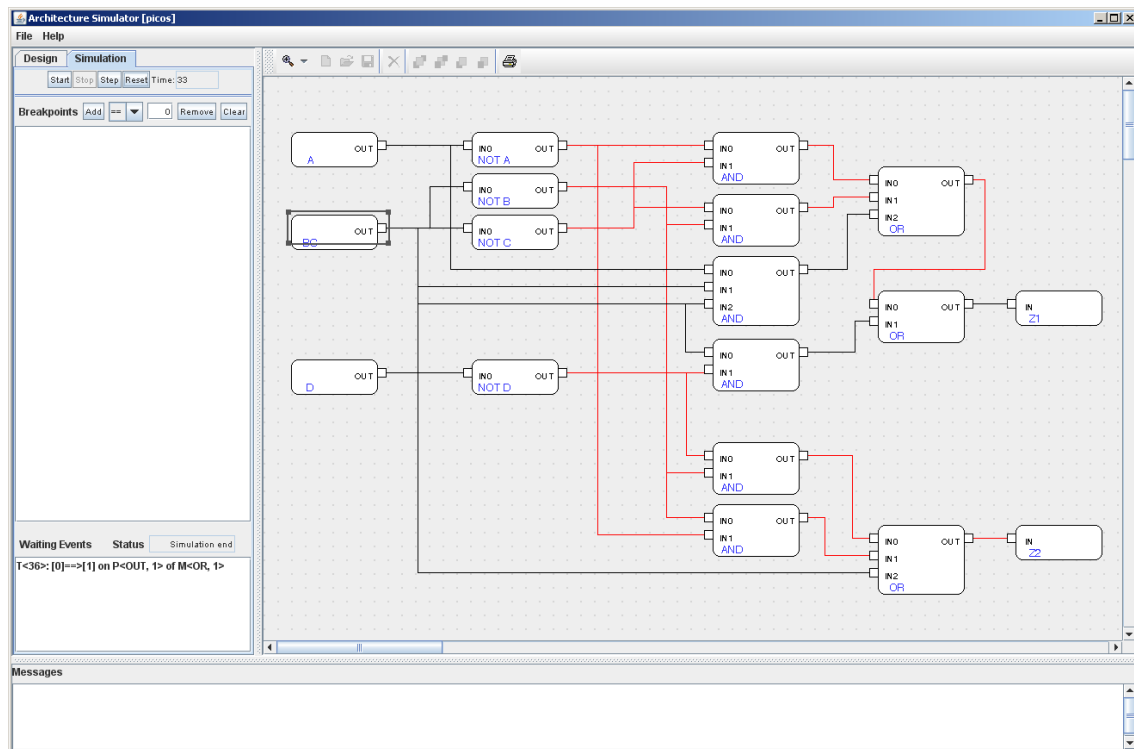




Vá carregando sucessivamente em Step e observe os valores das saídas. Note que elas começaram ambas em 1, passam ambas por 0 e terminam novamente ambas em 1.







Se esta sequência não fosse vista em Step a Step (passo a passo), nem se daria pelo facto de as saídas irem temporariamente a 0.

Ou seja: o facto de haver alterações nas entradas pode ocasionar picos transitórios nas saídas, enquanto o circuito não estabiliza.

No caso deste circuito, tal é provocado pelas negações dos sinais de entrada. O atraso introduzido por cada negação é suficiente para os sinais negados e não negados

atingirem as portas lógicas em instantes diferentes e provocar os picos até que tudo estabilize. A duração dos picos é da ordem dos atrasos das negações.

Em circuitos combinatórios, os picos não originam problemas. Mas se estas saídas fossem usadas como sinal de relógio de um circuito sequencial, seria introduzido um ciclo de relógio extemporâneo, não desejado e que poderia ocasionar um funcionamento incorrecto do circuito.