

# Guião de Laboratório de Arquitectura de Computadores

## Simulação 2.11 – Contadores

### 1 – Objectivos

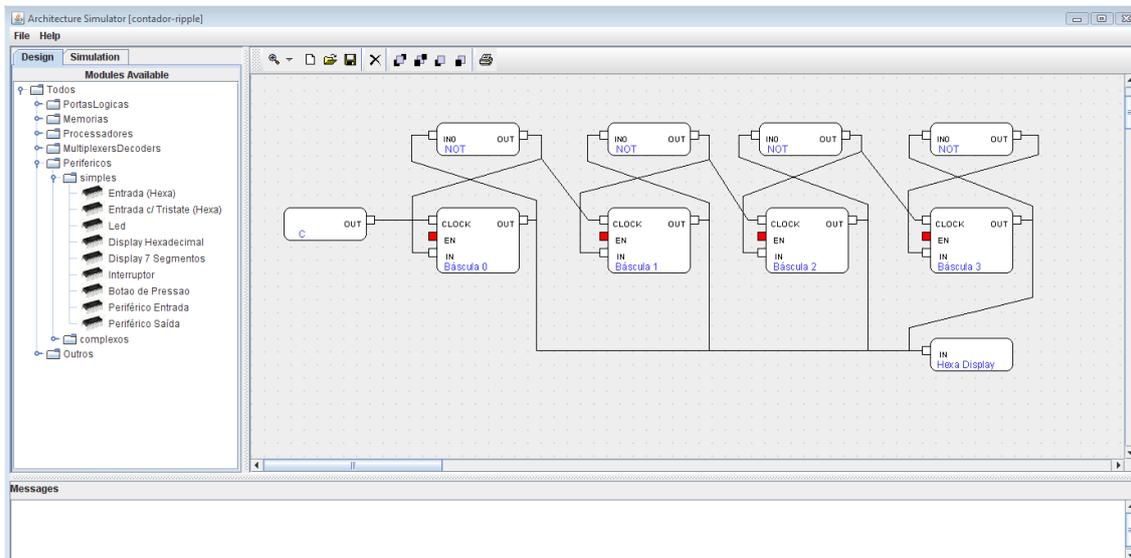
Esta simulação ilustra o funcionamento dos contadores com bsculas D, com base nas figuras e exemplos da secção 2.6.5. Os aspectos cobertos incluem os seguintes:

- Verificação do comportamento do contador em seqüência (*ripple*) e da soma dos tempos de atraso (Fig. 2.30);
- Verificação do comportamento do contador da Fig. 2.32 e do seu tempo de atraso face ao relógio;
- Contagem crescente com carregamento paralelo (Fig. 2.34);
- Contagem decrescente com carregamento paralelo (Fig. 2.35) e programabilidade do período de contagem.

NOTA – Na descrição da simulação 2.11, no livro, onde está “Fig. 2.33” devia estar “Fig. 2.34” e onde está “Fig. 2.34” devia estar “Fig. 2.35”. Este erro está corrigido aqui e registado na errata do livro.

### 2 – Contador com bscula D em seqüência (*ripple*)

Carregue o circuito seguinte do ficheiro “contador-ripple.cmod” e que implementa o circuito da Fig. 2.30. Note que as bsculas estão pela ordem inversa, apenas para facilitar a ligação entre bsculas no circuito, uma vez que os módulos têm sempre as entradas do lado esquerdo e as saídas do lado direito. Mas o circuito é o mesmo.



Passa para Simulação, faça Start e abra as janelas do botão e do Mostrador de 7 segmentos. Verifique que de cada vez que o botão vai de 0 para 1 o número no mostrador é incrementado.

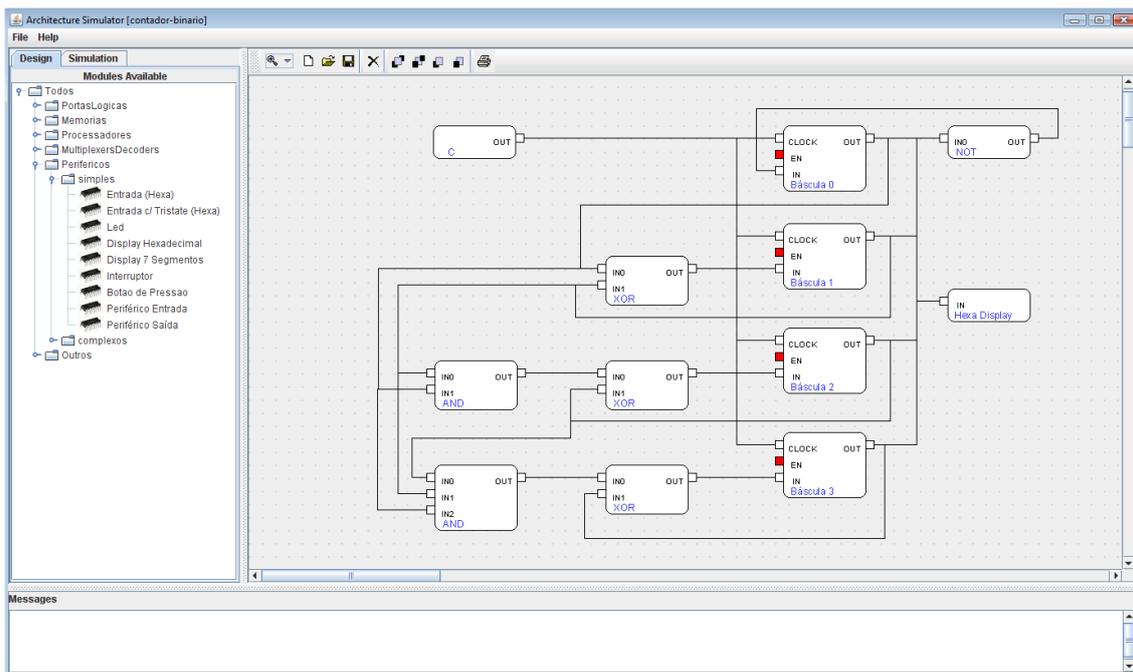
Note também que o número parece piscar às vezes (embora em computadores rápidos quase nem se note), e que o tempo de simulação avança bastante (umas vezes mais, outras menos) de cada vez que o botão muda de valor.

Tal deve-se à propagação dos vários sinais ao longo do circuito, que faz o mostrador passar por vários números intermédios (demasiado rápido para se ver, mas o suficiente para se notar o número a piscar, na transição) até atingir o valor de equilíbrio. Estes atrasos são ilustrados na sequência de sinais da Fig. 2.30.

NOTA – Para este circuito funcionar, a saída de cada báciaula tem de ligar num bit diferente da ligação ao mostrador de 7 segmentos, pois esta ligação é de 4 bits. Isto está programado no próprio circuito e afecta os sinais OUT de cada báciaula e os sinais IN0 de cada NOT.

### 3 – Contador binário

Carregue o circuito seguinte do ficheiro “contador-binario.cmod” e que implementa o circuito da Fig. 2.32.



Passa para Simulação, faça Start e abra as janelas do botão e do Mostrador de 7 segmentos. Verifique que de cada vez que o botão vai de 0 para 1 o número no mostrador é incrementado.

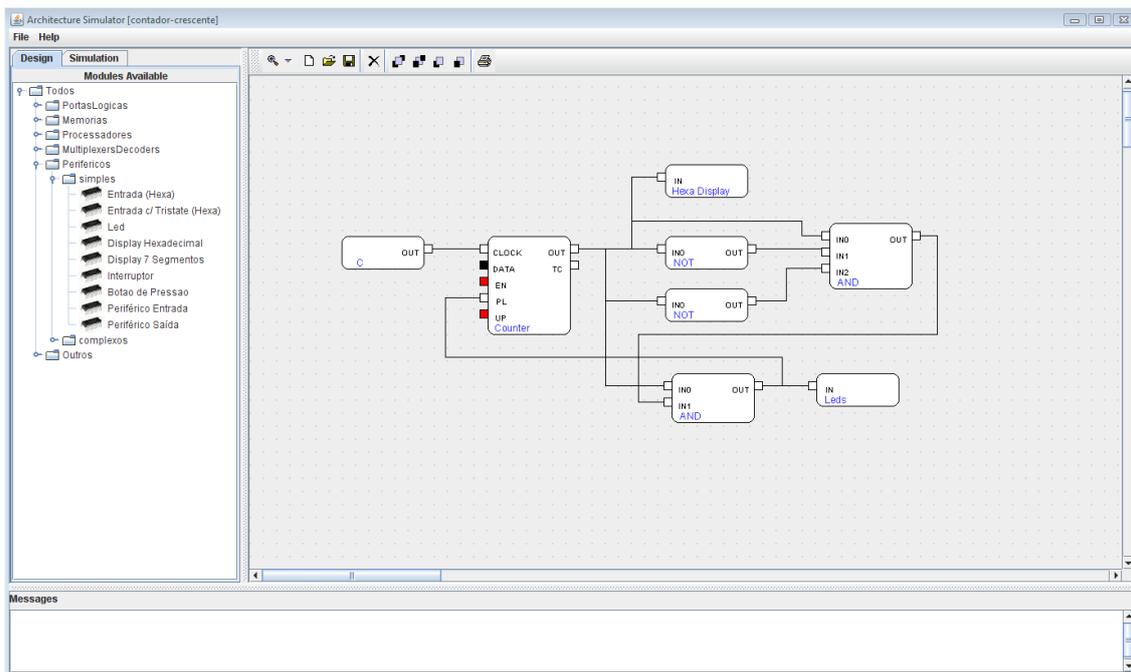
Note que, ao contrário do contador em sequência (*ripple*), o sinal de relógio (botão C) entra ao mesmo tempo em todas as bacias. Apesar de, em cada vez que o valor do botão muda de valor, o tempo de simulação também avançar algumas unidades (devido ao tempo de propagação dos sinais através das portas lógicas, tal sucede enquanto as bacias estão estáveis (só mudam na transição do sinal C de 0 para 1), e estas mudam todas simultaneamente. Os tempos de atraso das portas lógicas acontecem após as mudanças de estado das bacias e só têm influência na próxima transição do botão para

1. Desta forma, as transições do número no Mostrador de 7 segmentos evoluem de forma estável (sem nunca piscar) de cada vez que o botão passa de 0 para 1.

NOTA – Para este circuito funcionar, a saída de cada báscula tem de ligar num bit diferente da ligação ao mostrador de 7 segmentos, pois esta ligação é de 4 bits. Isto está programado no próprio circuito e afecta não só os sinais OUT de cada báscula como todos as entradas dos restantes módulos que a elas ligam.

#### 4 – Contador crescente com carregamento paralelo

Carregue o circuito seguinte do ficheiro “contador-crescente.cmod” e que implementa o circuito da Fig. 2.34, com a adição de um led no sinal de carregamento paralelo para melhor visualizar a ocorrência desta situação.



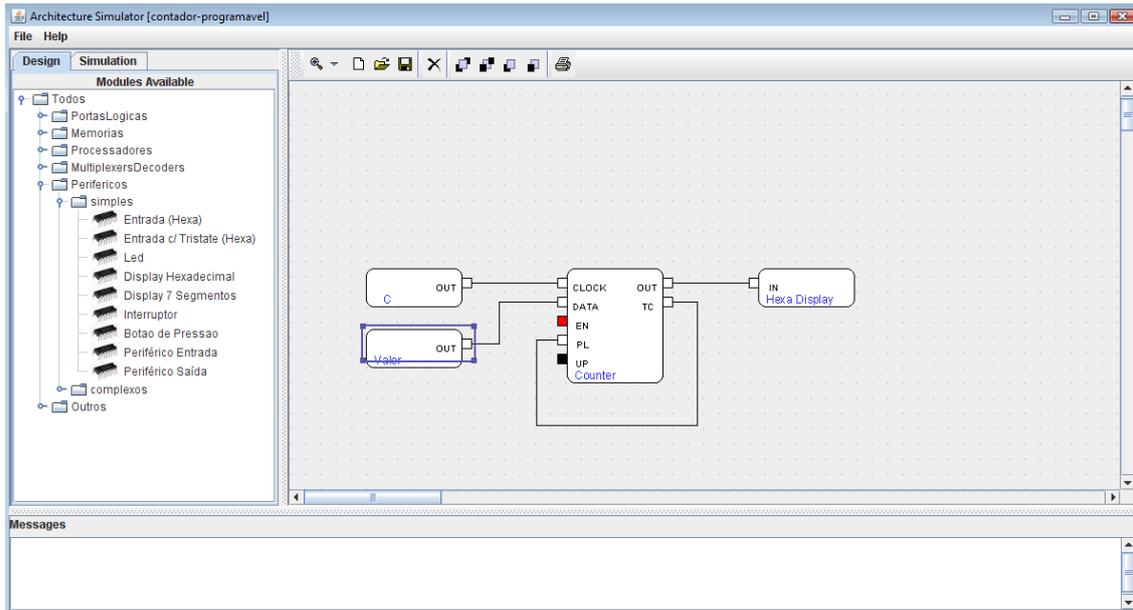
Passe para Simulação, faça Start e abra as janelas do botão, do mostrador e do led.

Verifique que de cada vez que a saída do botão passa para 1 o mostrador conta uma unidade e que quando este chega a 9 o led fica aceso. No próximo clique, o mostrador passa para 0, devido ao carregamento em paralelo do valor presente na entrada do contador (forçada a 0).

#### 5 – Contador decrescente com período programável

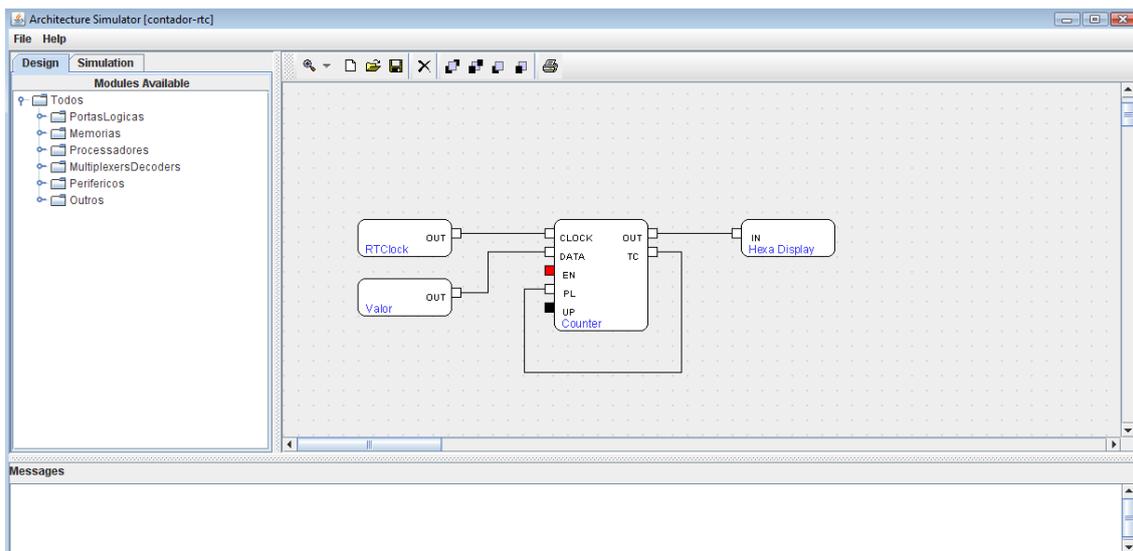
Carregue o circuito seguinte do ficheiro “contador-programavel.cmod” e que implementa o circuito da Fig. 2.35, mas em que o módulo Contador do simulador já produz um sinal (TC – Terminal Count) que já implementa o NOR, isto é, dá um impulso a 1 quando a contagem chega ao fim. O sinal UP está forçado a 0 para a contagem ser decrescente.

A janela de entrada (valor) permite especificar o período de contagem.



Passe para Simulação, faça Start e abra as janelas do botão C, da entrada Valor e do mostrador de 7 segmentos. Na janela de entrada, coloque 6 (como na Fig. 2.35, ou outro valor à sua escolha) e carregue em OK. Vá carregando no botão até o contador chegar a 0 (não se admire se o contador começar a contar num valor diferente daquele que especificou em Valor, pois este só é lido quando o contador TC estiver activo, e quando o contador arranca ainda não está inicializado. Mas após a primeira volta já fica certo). Verifique que no próximo ciclo de relógio o contador recomeça em 6 (ou no valor que consta na janela de entrada).

Use agora o ficheiro “contador-rtc.cmod”, que em vez de um botão utiliza um relógio de tempo real (RTC), com período de 1 segundo.



Faça como no caso anterior, mas em vez da janela do botão abra a janela do RTC e carregue em Start nesta janela.



O funcionamento do contador é agora automático, evoluindo em cada segundo. Experimente alterar o o valor na janela de entrada (fazendo OK) e verifique que este não tem efeito imediato no valor do contador. Só quando o contador chegar a 0 é que carrega o novo valor.