

Guião de Laboratório de Arquitectura de Computadores (1ª edição)

Simulação 7.7 – Programação cooperativa

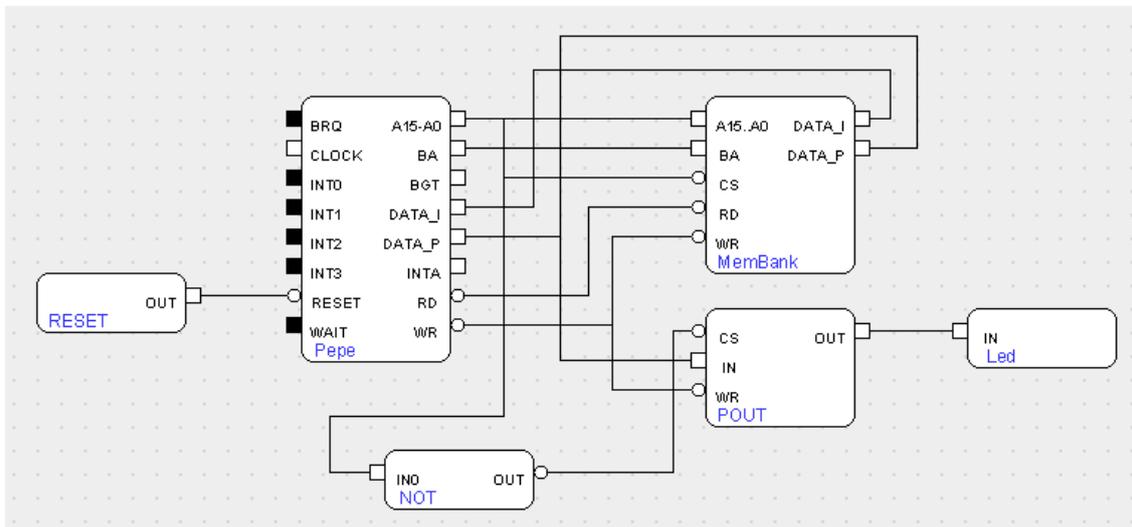
1 – Objectivos

Esta simulação toma como base o Programa 7.5 e exemplifica o funcionamento da programação cooperativa. São focados os seguintes aspectos, entre outros:

- Efeito da utilização de um ciclo de espera interno a um dos processos;
- Marcação das temporizações por interrupção (temporizador) em vez de ser por contagem de iterações (variante ao Programa 7.5).

2 – Circuito

O ficheiro “programacao-cooperativa.cmod” implementa o circuito da figura seguinte, que consiste apenas no PEPE, no banco de memória e um periférico ligado a oito leds.



3 – Simulação do programa 7.5

Carregue este circuito no simulador e passe para Simulação.

Abra o painel do PEPE e compile e carregue (📁) o ficheiro “programa7-5.asm”, que contém o respectivo programa em assembly. Este programa é semelhante ao programa 7.2, já usado na simulação 7.5, mas aqui usa-se programação cooperativa e não interrupções. Cada processo tem de cooperar, abrindo mão do processador (retornando da rotina) periodicamente, de modo a permitir que os outros processos se executem

também. A secção 7.7.4 explica o conceito e funcionamento da programação cooperativa.

Abra o painel dos leds e execute o programa em contínuo (botão  no painel do PEPE).

Verifique que o led 0 pisca (processo Pisca) e os leds 2 e 3 alternam entre si (processo Alterna). Se necessário, ajuste o valor da constante ESPERA (definida com EQU no início do programa) para que os led pisquem com um ritmo adequado. Este valor depende da rapidez do computador.

Note a sincronização entre o piscar do led 0 e a alternância dos leds 2 e 3. Esta é uma consequência do funcionamento cooperativo, pois ambos os processos executam uma iteração da espera em cada iteração do ciclo principal dos processos.

4 – Simulação de uma variante do programa 7.5 com tempo excessivo

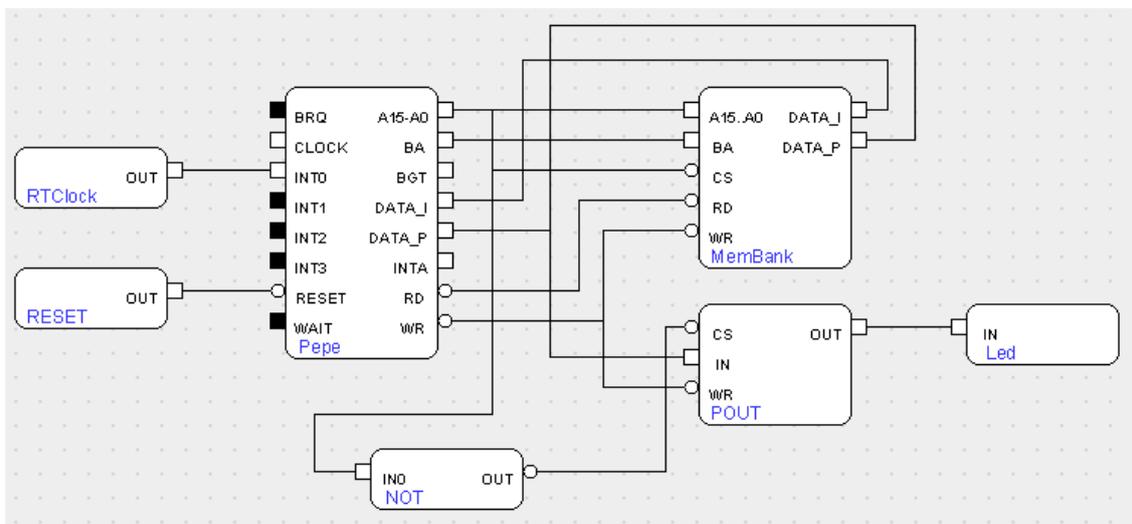
Faça reset ao simulador e compile e carregue o ficheiro “programa7-5-espera.asm”. Este programa é igual ao programa 7.5, com a adição de um ciclo no estado 2 do processo Alterna.

Execute o programa e verifique que, um dos estados dos leds que alternam demora muito mais que o outro, e que neste intervalo de tempo extra o led 0 também está parado.

Este exemplo ilustra que um dos processos pode bloquear os outros se não cooperar ou demorar demasiado tempo. Este problema não existiria se a mudança de processos fosse feita por interrupções.

5 – Simulação de uma variante do programa 7.5 com tempos por relógio

Faça reset ao simulador e carregue o circuito “prog-coop-interruptoes.cmod”, que possui agora um contador de tempo real ligado à interrupção 0. Isto vai permitir medir os tempos entre estados dos leds de forma rigorosa e não por ciclos de instruções para simular tempo de espera.



Abra o painel do PEPE e compile e carregue () o ficheiro “programa7-5.asm”, que Compile e carregue o ficheiro “programa7-5-relogio.asm”. Este programa é igual ao programa 7.5, mas com as temporizações marcadas por relógios e não por ciclos em software. Cada processo usa o seu próprio relógio, para ilustrar a interacção entre as interrupções e a programação cooperativa.

A interrupção 0 decrementa dois contadores, um para cada processo, para tornar a contagem de tempos independente de um processo para o outro. Cada processo vai vendo o valor do seu contador e, quando chegar a zero, repõe o valor inicial e muda o estado dos leds.

A espera continua a ser externa aos processos, no sentido de que cada processo testa o seu contador e, se ainda não tiver chegado a zero, sai e permite que o processo seguinte execute uma nova iteração. A diferença está na forma como o contador é decrementado, que nesta variante é feito ao ritmo de um relógio externo.

O período desse relógio (coloque 100, ou 100 milissegundos) é a base de tempos dos contadores, pelo que os tempos de espera dos processos serão um múltiplo deste valor. A constante ESPERA_PISCA, com o valor 10, indica que o tempo de espera entre mudanças do estado do led 0 é de 1 segundo (10 x 100), enquanto a constante ESPERA_ALT, com o valor 15, faz com que o tempo entre alternâncias de valor dos leds 2 e 3 seja de 1,5 segundos. Pode experimentar outros valores.

Abra o painel dos leds e execute o programa. Verifique o funcionamento e temporização dos leds, e que já não estão sincronizados (o tempo de espera dos processos já não é igual).