

Sistemas Embebidos II

Semestre de Verão de 2009/2010

Segunda actividade prática

1ª Parte

Mecanismo de interrupções do microcontrolador LPC2106

1º – Teste em modo não vectorizado

Criar um programa que afixe sucessivamente na consola, o valor de uma variável, que é incrementada na rotina de interrupção, gerada por um *timer*.

```
#include "lpc210x.h"
#include "io.h"
#include "console.h"

int counter = 5;

int main() {
    interrupt_init();
    timer_init(1000);
    console_init();
    interrupt_unmask(VIC_SOURCE_TIMER0);
    while (1) {
        console_printf("%d\r\n", counter);
    }
}
```

2º – Interrupções externas

Ligar um botão de pressão ao pino do sinal EINT0.

Modifique o programa anterior para aceitar também interrupções externas pelo sinal EINT0.

Modifique a rotina de atendimento para incrementar uma variável diferente para as interrupções externas.

3º – Teste em modo vectorizado

Ligar outro botão de pressão ao pino do sinal EINT1.

Modifique o programa anterior de modo a aceitar interrupções em modo vectorizado.

Neste modo existe uma rotina específica para cada entrada de interrupção. Estas rotinas são colocadas em execução directamente pelo hardware em função da entrada de interrupção activa.

Verifique que as rotinas são executadas ao se premirem os botões.

Se premir o botão 0 e depois o botão 1 a segunda rotina não é executada.

Se premir o botão 1 e depois o botão 0 ambas as rotinas são executadas.

A situação anterior deve-se à lógica de prioridades implementada pelo VIC: enquanto estiver a ser

atendida uma interrupção o VIC só gera nova interrupção (activa IRQ) se for activada uma fonte de maior prioridade.

O VIC precisa de ser sinalizado do fim das rotinas de interrupção para que possa voltar a gerar pedidos com origem em fontes de mais baixa prioridade. Essa sinalização é realizada por uma escrita (de um valor qualquer) no registo VicVectAddr. Cada escrita neste registo representa a conclusão da rotina de mais alta prioridade que estava a ser atendida.

```
io_write_u32(LPC210X_VIC + LPC2XXX_VICVectAddr, 0);
```

A partir deste ponto já pode verificar a execução da rotina de interrupção por cada clique no botão. Note que, por cada clique, o valor do contador não incrementa apenas uma unidade dá saltos de várias unidades. Porquê?

2ª Parte

Estruturação do atendimento de interrupções

Repetir o primeiro ensaio utilizando a API definida nos apontamentos “Interrupções no LPC2106”.

3ª Parte

Gestor de controlador Ethernet por interrupção

Na actividade prática anterior foi realizado um gestor de periférico (*device driver*), para rede Ethernet, sobre o controlador ENC28J60. Esse gestor não tem capacidade de armazenamento de tramas na transmissão, e na recepção está confinado à memória interna do controlador.

Mantendo a interface de utilização, modifique a programação do gestor de periférico, realizado na primeira actividade prática, para satisfazer os seguintes requisitos:

- A função **ethernet_send** é invocada para processar o envio de uma trama. Se não poder dar início à transmissão da trama que transporta, por estar a decorrer a transmissão de uma trama anterior, deve colocá-la numa fila de espera. O processamento relacionado com a terminação da transmissão deve ser executado via rotina de interrupção. Na sequência da terminação da transmissão de uma trama deve ser desencadeada a transmissão da próxima trama presente na fila.
- O gestor deve ter capacidade de armazenamento de tramas na recepção. Quando uma trama é recebida pelo controlador deve ser transferida para uma fila de espera na memória principal. Essa transferência deve ser operada via rotina de interrupção. A função **ethernet_recv** em vez de recolher as tramas recebidas da memória do controlador recolhe da fila de espera de recepção.

Referências:

ARM System-on-chip architecture – capítulo 5

UM10275 - LPC2104/2105/2106 User manual

AN10381 - Nesting of interrupts in the LPC2000

AN10414 - Handling of spurious interrupts in the LPC2000

Placa Ethernet ENC28J60-H <http://www.olimex.com/dev/enc28j60-h.html>

ENC28J60 Stand-Alone Ethernet Controller with SPI Interface
<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en022889>