PCS3432 - Laboratório de Processadores

Planejamento - E10

Bruno Mariz - 11261826

1)

O código não foi compilado por não ser possível realizar a instalação das ferramentas da disciplina no ubuntu 16.

2)

Código em irq.s:

Essa parte do código define o vetor de interrupções:

```
.global _start
.text
_start:
    b _Reset @posição 0x00 - Reset
    ldr pc, _undefined_instruction @posição 0x04 - Intrução não-definida
    ldr pc, _software_interrupt @posição 0x08 - Interrupção de Software
    ldr pc, _prefetch_abort @posição 0x0C - Prefetch Abort
    ldr pc, _data_abort @posição 0x10 - Data Abort
    ldr pc, _not_used @posição 0x14 - Não utilizado
    ldr pc, _irq @posição 0x18 - Interrupção (IRQ)
    ldr pc, _fiq @posição 0x1C - Interrupção(FIQ)

_undefined_instruction: .word undefined_instruction
_software_interrupt: .word software_interrupt
_prefetch_abort: .word prefetch_abort
_data_abort: .word data_abort
_not_used: .word not_used
_irq: .word irq
_fiq: .word fiq
```

Abaixo são definidas algumas constantes como o endereço de registradores de controle de interrupções:

```
INTPND: .word 0x10140000 @Interrupt status register
INTSEL: .word 0x1014000C @interrupt select register( 0 = irq, 1 = fiq)
INTEN: .word 0x10140010 @interrupt enable register
TIMER0L: .word 0x101E2000 @Timer 0 load register
TIMER0V: .word 0x101E2004 @Timer 0 value registers
```

```
TIMER0C: .word 0x101E2008 @timer 0 control register
```

Abaixo é definido o que é feito quando cada interrupção ocorrer:

```
_Reset:

    LDR sp, =supervisor_stack_top

    @ Inicializacao das stacks
    MRS r0, cpsr    @ salvando o modo corrente em R0
   MSR cpsr_ctl, #0b11010010 @ alterando o modo para irq - o SP eh
automaticamente chaveado ao chavear o modo
    LDR sp, =irq_stack_top @ a pilha de irq eh setada
   MSR cpsr, r0 @ volta para o modo anterior

    bl main
    b .
undefined_instruction:
    b .
software_interrupt:
    b do_software_interrupt @ vai para o handler de interrupções de
software
prefetch_abort:
    b .
data_abort:
    b .
not_used:
    b .
irq:
    b do_irq_interrupt @vai para o handler de interrupções IRQ
fiq:
    b .

do_software_interrupt: @Rotina de Interrupçãode software
    add r1, r2, r3 @r1 = r2 + r3
    mov pc, r14 @volta p/ o endereço armazenado em r14

do_irq_interrupt: @Rotina de interrupções IRQ
    SUB lr, lr, #4
    STMFD sp!, {r0 - r12, lr} @Empilha os registradores
    LDR r0, INTPND @Carrega o registrador de status de interrupção
    LDR r0, [r0]
    TST r0, #0x0010 @verifica se é uma interupção de timer
    BLNE handler_timer @vai para o rotina de tratamento da interupção de
timer
    vesaida:
    LDMFD sp!, {r0 - r12, pc}^ @retorna
```

Em seguida são definidas as instruções que ativarão as interrupções de timer e inicializarão o timer:

```
timer_init:
    @ Enable timer0 interrupt
    LDR r0, INTEN
    LDR r1,=0x10 @bit 4 for timer 0 interrupt enable
    STR r1,[r0]

    @ Set timer value
    LDR r0, TIMER0L
    LDR r1, =0xfffff @setting timer value
    STR r1,[r0]

    @ Enable timer0 counting
    LDR r0, TIMER0C
    MOV r1, #0xE0 @enable timer module
    STR r1, [r0]

    @ Enable processor interrupt in CPSR
    mrs r0, cpsr
    bic r0,r0,#0x80
    msr cpsr_c,r0 @enabling interrupts in the cpsr

    mov pc, lr
```

Por fim, temos o programa principal:

```
main:
    bl hello_world
    bl timer_init @ initialize interrupts and timer 0
loop:
    bl print_space
    b loop
```

3)

O timer é programado na subrotina timer_init, que habilita as interrupções e define o valor do timer.

4)

5)

Os registradores do timer são utilizados no programa principal, e servem para ativação de interrupções, definição de valores do timer e desativação de interrupções.

5)

Os registradores mapeados em memória da controladora de interrupção são utilizados tanto no interrupt handler, como em do_irq_interrupt, quanto no programa principal, em timer_init, e servem para verificar e habilitar/desabilitar interrupções.

6)

As flags precisam ser zeradas para que o programa não fique em loop gerando a mesma interrupção em sequência.

7)

Essa sequência de instruções serve para habilitar as interrupções de timer.