

Atividades de Laboratório 9

Objetivos

O objetivo desta atividade é exercitar o tratamento de interrupções externas utilizando o conjunto de instruções da arquitetura RISC-V.

Descrição

Neste laboratório, você deve implementar uma biblioteca que será utilizada por um tocador de áudio [MIDI](#). Veja a descrição dos periféricos a serem utilizados no final deste documento. Sua biblioteca tem três funções principais:

1. Implementar uma função `_start`, que inicialize a pilha do programa (veja em dicas como configurar a pilha), configure as interrupções, e chame a função `main` (utilizando a instrução `jal`), fornecida em nossa aplicação.
 - Note que utilizaremos a instrução `jal`, pois não precisaremos mudar de modo. Nos próximos laboratórios, isso pode ser necessário (e o `mret` será utilizado no lugar do `jal`)
2. Programar e tratar interrupções externas de um dispositivo temporizador (GPT).
 - O tratador de interrupção do GPT deve simplesmente incrementar um contador global de tempo (`_system_time`), que deve ser iniciado em zero e que deve armazenar o tempo do sistema em milissegundos. (O tempo do sistema é o tempo que passou desde que o sistema foi iniciado)
 - Sugerimos programar o GPT para que gere interrupções a cada 100 ms;
3. Prover uma função `play_note` para acessar o periférico sintetizador de MIDI através de MMIO.
 - A assinatura da função é:

```
void play_note(int ch, int inst, int note, int vel, int dur);
```
 - Os parâmetros correspondem a:
 - i. `ch`: Canal;
 - ii. `inst`: ID do instrumento musical;
 - iii. `note`: Nota musical;
 - iv. `vel`: velocidade da nota;
 - v. `dur`: duração da nota.

Observações

- Para alocar as pilhas, você pode declarar dois vetores em seu programa. Quando for iniciar o apontador de pilha (SP), lembre-se que as pilhas do RISC-V são descendentes-cheia.

- Você deve implementar todas as funções em um único arquivo `lib.s`, em linguagem de montagem do RISC-V.
- Nós fornecemos a nossa aplicação no arquivo [lab9.c](#), que utiliza a sua biblioteca e que não deve ser alterada.
- Todas as funções devem seguir a ABI apresentada na disciplina.
- Todas as funções devem ser [thread-safe](#). Neste momento, você **não** precisa entender este conceito. Apenas garanta que seu código **não** utilize variáveis globais (as únicas exceções são as variáveis `_system_time` e as pilhas do programa e da ISR). Utilize apenas variáveis locais, alocadas na pilha ou em registradores.
- **Configurar o simulador:**
 - Para receber interrupções externas, configure o “Bus Controller Frequency Scaling” na aba “Hardware” em $1/2^7$.
 - Também na aba “Hardware”, adicione o GPT e, em seguida, o Sound Synthesizer (MIDI) (**nessa ordem**).
- Antes de começar a testar o Sintetizador MIDI, verifique o volume do seu navegador e computador.

Periféricos

- Você deve acessar os periféricos através de MMIO, assim como feito no laboratório 8.
- O `base_addr` de cada dispositivo pode ser visto na tabela “Memory Map” na aba “Hardware”.

GPT - General Purpose Timer

Endereço	Tamanho	Descrição
<code>base_addr</code>	word	Quando atribuído o valor 1, o GPT inicia a leitura do tempo atual do sistema. Quando a leitura é concluída, o GPT escreve o valor 0 no registrador associado a este endereço.
<code>base_addr + 4</code>	word	Quando finalizada a leitura do tempo atual do sistema, <i>i.e.</i> , quando o valor armazenado no registrador associado ao endereço <code>base_addr</code> for 0, este registrador armazena o tempo atual do sistema, em milissegundos. Este tempo é monotonicamente crescente, desde quando o simulador foi iniciado.
<code>base_addr + 8</code>	word	Quando atribuído um valor $v > 0$, programa o GPT para gerar uma interrupção em v milissegundos. Quando gerada, o valor 0 é atribuído a <code>base_addr + 8</code> .

MIDI Synthesizer

Endereço	Tamanho	Descrição
<code>base_addr</code>	<code>byte</code>	Quando atribuído um valor <i>ch</i> diferente de -1, inicia a reprodução da nota no canal <i>ch</i> .
<code>base_addr + 2</code>	<code>short</code>	Identificador do instrumento a ser tocado.
<code>base_addr + 4</code>	<code>byte</code>	Valor da nota (frequência) a ser tocada.
<code>base_addr + 5</code>	<code>byte</code>	Velocidade da nota.
<code>base_addr + 6</code>	<code>short</code>	Duração da nota.