



UNIVERSIDADE FEDERAL
DE SANTA CATARINA

Implementação de Interrupções em Arquitetura NEORV32

Disciplina de Microprocessadores

Me. Eng^o Luiz Felipe Baldo Marques

**Departamento de Engenharia
Elétrica e Eletrônica**

Interrupções - Conceitos Gerais

- Permitem a execução de tarefas predeterminadas (tratadores) a partir de requisições esporádicas.
- Interrompem a execução da rotina principal para executar outras tarefas sob demanda.
- **Polling:** processo no qual o processador testa sequencialmente todas as entradas de interrupção para saber quais estão solicitando a execução de tarefa.
- Em caso de mais uma interrupção solicitada, o processador precisa escolher qual interrupção tem maior prioridade.
- Interrupções de menor prioridade não serão executadas nesse momento e deverão esperar um novo ciclo de teste das entradas de interrupção.
- **Desvantagem:** Devido à necessidade do processador testar, frequentemente, se algum periférico necessita atenção, seu desempenho decai na execução de outras tarefas.

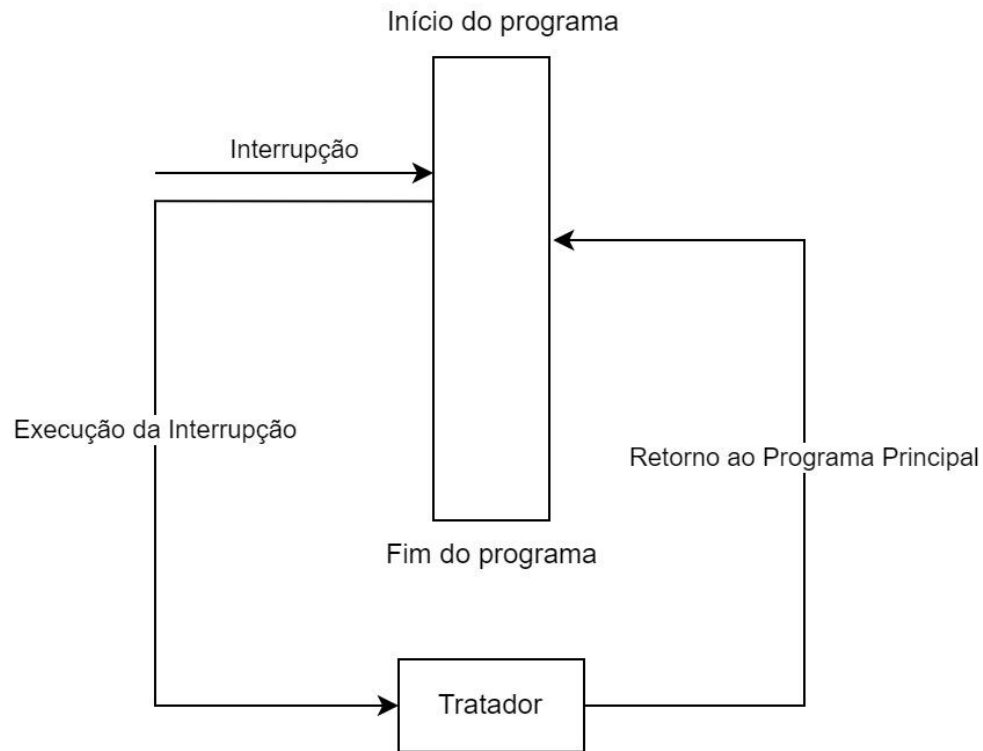
Interrupções Externas

- São implementadas através de portas físicas digitais do microprocessador, permitindo a interação do mesmo com dispositivos externos.
- Permitem a interação do microprocessador com o mundo externo, através de lógicas de reação a eventos.
- Exemplos:
 - Um sensor de presença que ativa uma câmera de monitoramento de tráfego.
 - Uma tela de banco que interrompe uma propaganda quando uma nova senha é chamada para atendimento no caixa.
 - Sistema de emergência de um carro em caso de detecção de colisão.

Interrupções Internas

- São implementadas através de conexões internas do microprocessador/microcontrolador com os periféricos.
- São utilizadas para interrupções de *timers*, barramento serial, e gerenciamento de erros internos.
- Muito úteis para o tratamento de erros, alarmes e exceções de funcionamento interno do microprocessador.
- Exemplos:
 - Detecção de divisão por zero.
 - Requisição de utilização de barramento serial.
 - Buffer de recepção serial cheio.
 - Monitoramento de *timeouts* de funções do programa.

Interrupções



Interrupções - NEORV32

- Possui 32 (0 a 31) canais de interrupções configuráveis.
- Estrutura de interrupções internas e externas é configurada via código VHDL.
- Prioridade: em ordem crescente de maior prioridade para menor prioridade.
- Canal 0: mais prioritário - Canal 31: menos prioritário.
- Interrupções não atendidas são mantidas como “pendentes” para serem executadas quando forem as mais prioritárias.

Interrupções - NEORV32 (C++)

- Bibliotecas: `#include "neorv32.h"; #include "neorv32_xirq.h"`
- `int neorv32_xirq_setup(void)`
 - Inicializa o controle de interrupções.
 - Remove todas as configurações e status existentes.
 - Retorna 0 se executa com sucesso, != 0 no caso de erro.
- `void neorv32_xirq_channel_enable(int channel)`
 - Habilita o canal "channel".
- `void neorv32_xirq_channel_disable(int channel)`
 - Desabilita o canal "channel".

Interrupções - NEORV32 (C++)

- `void neorv32_xirq_global_enable(void)`
 - Habilita todos os canais.
- `void neorv32_xirq_global_disable(void)`
 - Desabilita todos os canais.
- `int neorv32_xirq_install(int channel, void (*handler)(void))`
 - Define que a função “`void (*handler)(void)`” deve ser executada em caso de interrupção no canal “`channel`”.
 - Retorna 0 se executa com sucesso, 1 no caso de erro.
 - A função a ser executada, também chamada de “tratador de interrupção”, deve ser declarada fora de função “`install`”.

Interrupções - NEORV32 (C++)

- `int neorv32_xirq_uninstall(int channel)`
 - Remove o tratador de interrupção do canal “channel”.
 - Retorna 0 se executa com sucesso, 1 no caso de erro.
- `int neorv32_xirq_get_num(void)`
 - Retorna o número de canais habilitados.
- `void neorv32_xirq_clear_pending(int channel)`
 - Limpa o status de “pendente” do canal “channel”.
 - O evento de interrupção deverá ocorrer novamente para executar o respectivo tratador;
- `int neorv32_xirq_available(void)`
 - Retorna 1 se o módulo de interrupções está presente no processador, caso contrário retorna 0.