



UNIVERSIDADE
FEDERAL DE
SERGIPE



DEPARTAMENTO
DE COMPUTAÇÃO

Interrupção e exceção

Arquitetura de Computadores

Bruno Prado

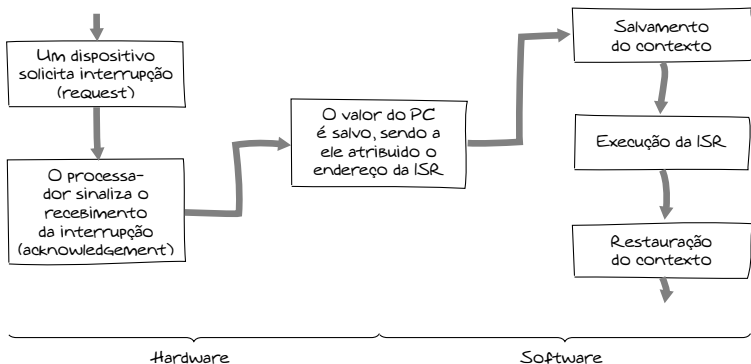
Departamento de Computação / UFS

Introdução

- ▶ O que é uma interrupção?
 - ▶ É um evento criado por um dispositivo de hardware (interrupção) ou gerado pela execução do software (exceção) que requisita a utilização de rotinas de tratamento de interrupção (ISR)

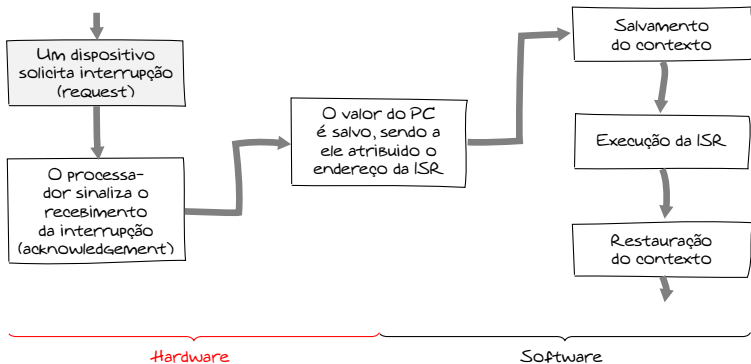
Introdução

- ▶ O que é uma interrupção?
 - ▶ É um evento criado por um dispositivo de hardware (interrupção) ou gerado pela execução do software (exceção) que requisita a utilização de rotinas de tratamento de interrupção (ISR)



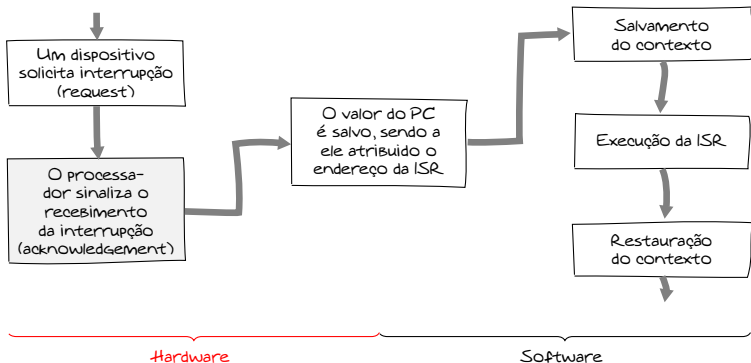
Introdução

- ▶ O que é uma interrupção?
 - ▶ É um evento criado por um dispositivo de hardware (interrupção) ou gerado pela execução do software (exceção) que requisita a utilização de rotinas de tratamento de interrupção (ISR)



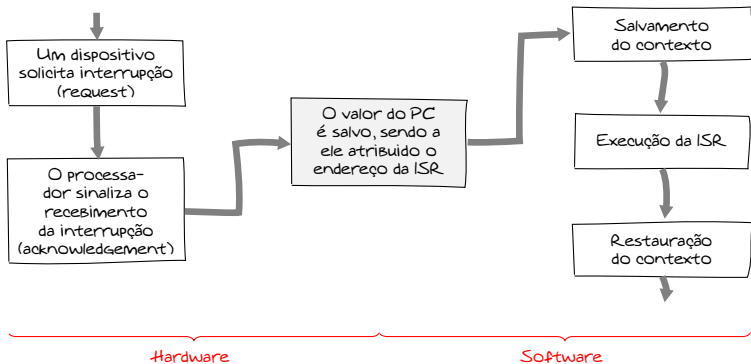
Introdução

- ▶ O que é uma interrupção?
 - ▶ É um evento criado por um dispositivo de hardware (interrupção) ou gerado pela execução do software (exceção) que requisita a utilização de rotinas de tratamento de interrupção (ISR)



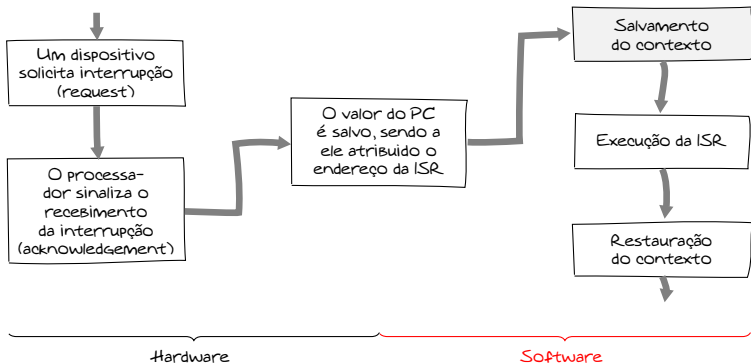
Introdução

- ▶ O que é uma interrupção?
 - ▶ É um evento criado por um dispositivo de hardware (interrupção) ou gerado pela execução do software (exceção) que requisita a utilização de rotinas de tratamento de interrupção (ISR)



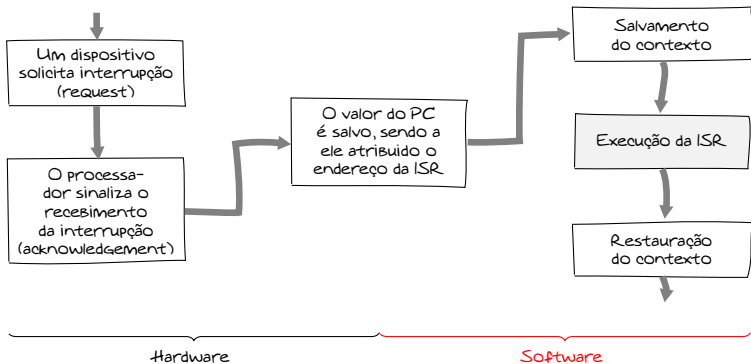
Introdução

- ▶ O que é uma interrupção?
 - ▶ É um evento criado por um dispositivo de hardware (interrupção) ou gerado pela execução do software (exceção) que requisita a utilização de rotinas de tratamento de interrupção (ISR)



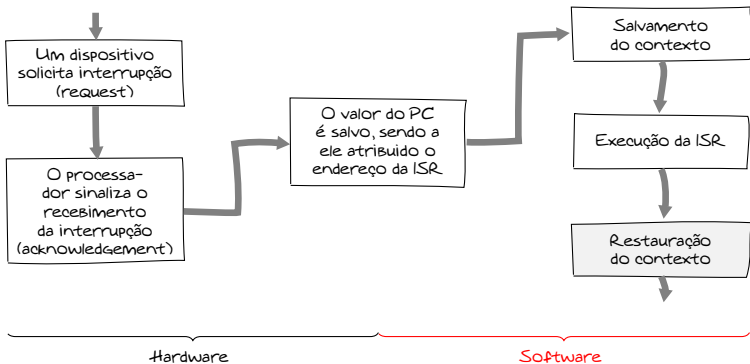
Introdução

- ▶ O que é uma interrupção?
 - ▶ É um evento criado por um dispositivo de hardware (interrupção) ou gerado pela execução do software (exceção) que requisita a utilização de rotinas de tratamento de interrupção (ISR)



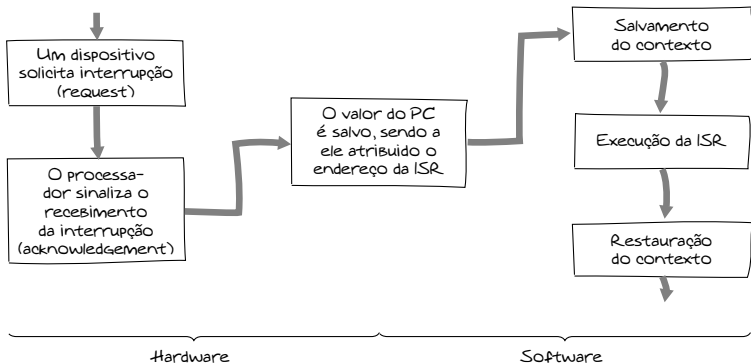
Introdução

- ▶ O que é uma interrupção?
 - ▶ É um evento criado por um dispositivo de hardware (interrupção) ou gerado pela execução do software (exceção) que requisita a utilização de rotinas de tratamento de interrupção (ISR)



Introdução

- ▶ O que é uma interrupção?
 - ▶ É um evento criado por um dispositivo de hardware (interrupção) ou gerado pela execução do software (exceção) que requisita a utilização de rotinas de tratamento de interrupção (ISR)



Introdução

- ▶ Por que utilizar interrupção é necessário?

Introdução

- ▶ Por que utilizar interrupção é necessário?
 - ▶ Evitar a espera do processador e sem reduzir a eficiência de execução das operações

Introdução

- ▶ Por que utilizar interrupção é necessário?
 - ▶ Evitar a espera do processador e sem reduzir a eficiência de execução das operações
 - ▶ Permitir o funcionamento assíncrono que evita a utilização de *polling* em dispositivos de E/S lentos

Introdução

- ▶ Por que utilizar interrupção é necessário?
 - ▶ Evitar a espera do processador e sem reduzir a eficiência de execução das operações
 - ▶ Permitir o funcionamento assíncrono que evita a utilização de *polling* em dispositivos de E/S lentos

```
1 // Biblioteca de E/S padrão
2 #include <stdio.h>
3 // Função principal
4 int main() {
5     // Variável de nome
6     char nome[50] = { 0 };
7     // Mensagem de pergunta
8     printf("Qual é o seu nome?\n");
9     // Leitura do teclado
10    scanf("%s", nome);
11    // Mensagem de resposta
12    printf("Olá %s!\n", nome);
13    // Retorno sem erros
14    return 0;
15 }
```

Introdução

- ▶ Por que utilizar interrupção é necessário?
 - ▶ Evitar a espera do processador e sem reduzir a eficiência de execução das operações
 - ▶ Permitir o funcionamento assíncrono que evita a utilização de *polling* em dispositivos de E/S lentos

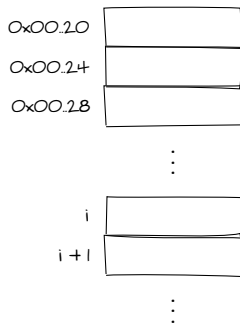
```
1 // Biblioteca de E/S padrão
2 #include <stdio.h>
3 // Função principal
4 int main() {
5     // Variável de nome
6     char nome[50] = { 0 };
7     // Mensagem de pergunta
8     printf("Qual é o seu nome?\n");
9     // Leitura do teclado
10    scanf("%s", nome);
11    // Mensagem de resposta
12    printf("Olá %s!\n", nome);
13    // Retorno sem erros
14    return 0;
15 }
```

► Comparativo entre tipos de interrupções

Hardware	Software
Assíncrona	Síncrona
Mascarável	Não mascarável
E/S em periféricos (envio ou recebimento de dados)	Execução de operações (instrução inválida ou divisão por zero)

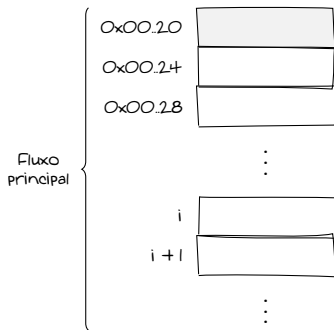
Interrupção

- ▶ Controle de fluxo para interrupção
 - ▶ Interrupção gerada durante execução da instrução i



Interrupção

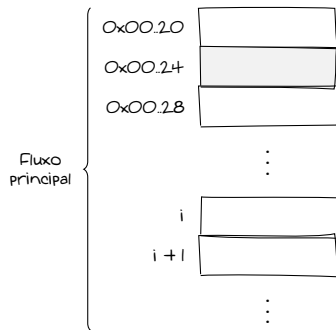
- ▶ Controle de fluxo para interrupção
 - ▶ Interrupção gerada durante execução da instrução i



Execução do fluxo principal da aplicação

Interrupção

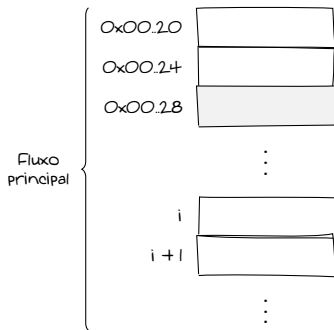
- ▶ Controle de fluxo para interrupção
 - ▶ Interrupção gerada durante execução da instrução i



Execução do fluxo principal da aplicação

Interrupção

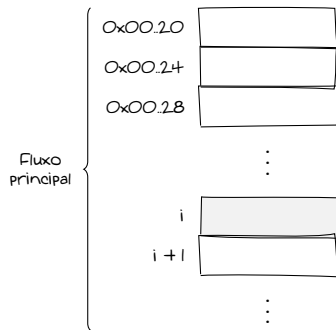
- ▶ Controle de fluxo para interrupção
 - ▶ Interrupção gerada durante execução da instrução i



Execução do fluxo principal da aplicação

Interrupção

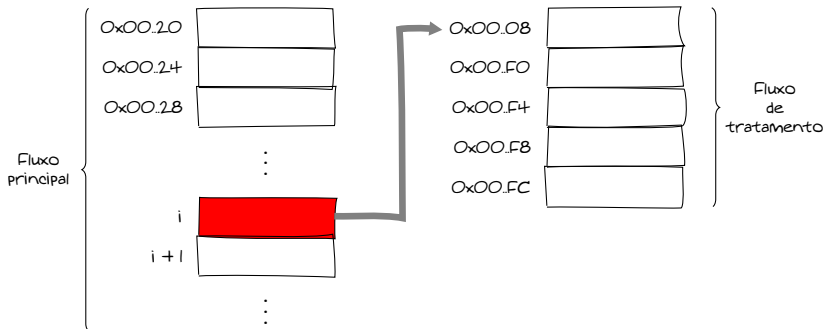
- ▶ Controle de fluxo para interrupção
 - ▶ Interrupção gerada durante execução da instrução i



Execução do fluxo principal da aplicação

Interrupção

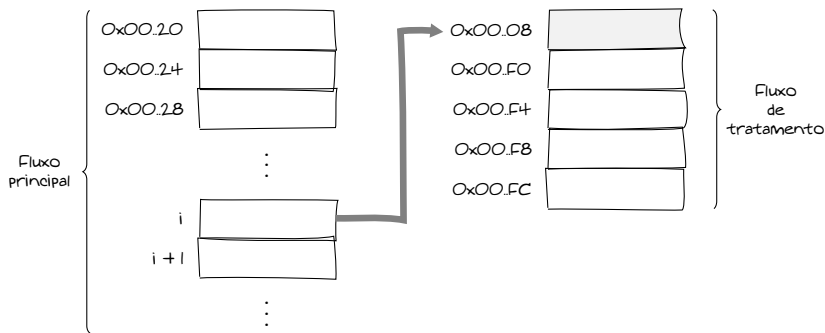
- ▶ Controle de fluxo para interrupção
 - ▶ Interrupção gerada durante execução da instrução i



Evento de interrupção gerado por hardware ou software causa um desvio para ISR

Interrupção

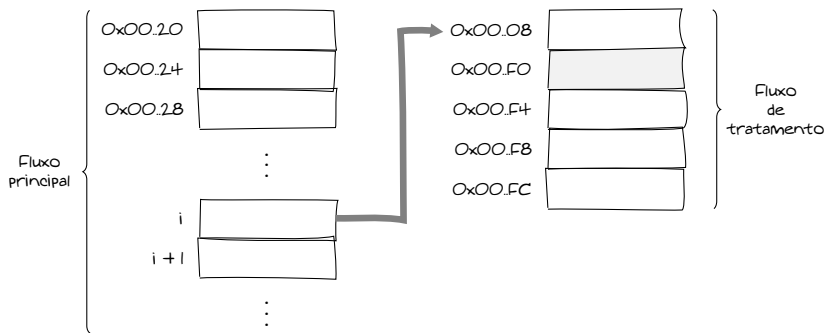
- ▶ Controle de fluxo para interrupção
 - ▶ Interrupção gerada durante execução da instrução i



Execução do fluxo de tratamento da interrupção

Interrupção

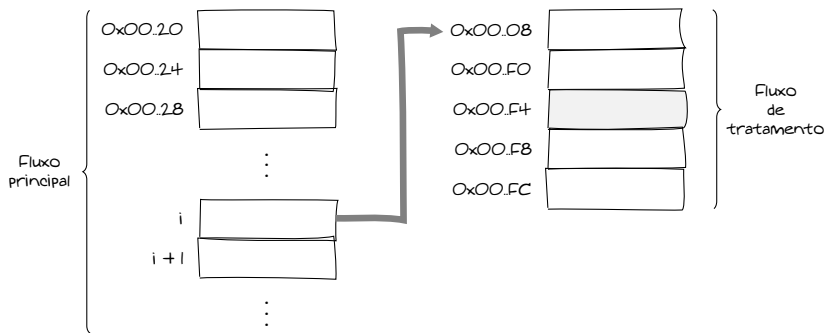
- ▶ Controle de fluxo para interrupção
 - ▶ Interrupção gerada durante execução da instrução i



Execução do fluxo de tratamento da interrupção

Interrupção

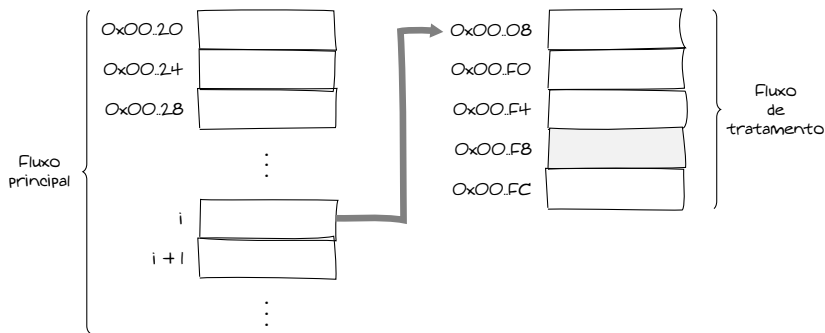
- ▶ Controle de fluxo para interrupção
 - ▶ Interrupção gerada durante execução da instrução i



Execução do fluxo de tratamento da interrupção

Interrupção

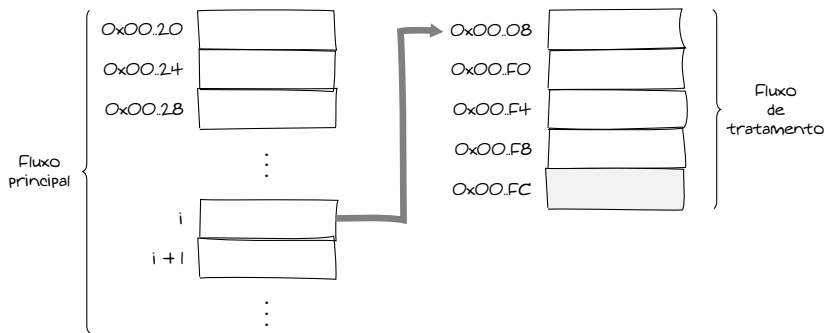
- ▶ Controle de fluxo para interrupção
 - ▶ Interrupção gerada durante execução da instrução i



Execução do fluxo de tratamento da interrupção

Interrupção

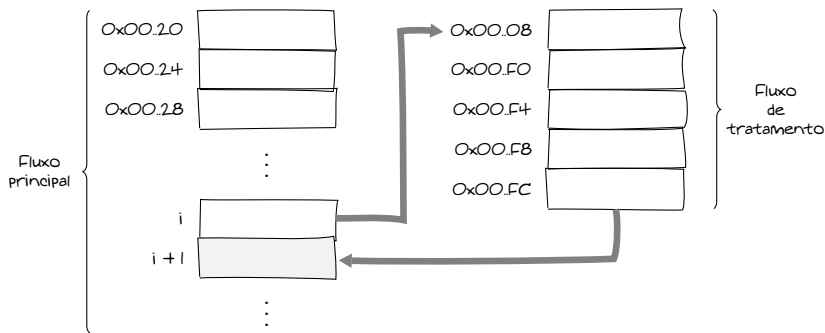
- ▶ Controle de fluxo para interrupção
 - ▶ Interrupção gerada durante execução da instrução i



Execução do fluxo de tratamento da interrupção

Interrupção

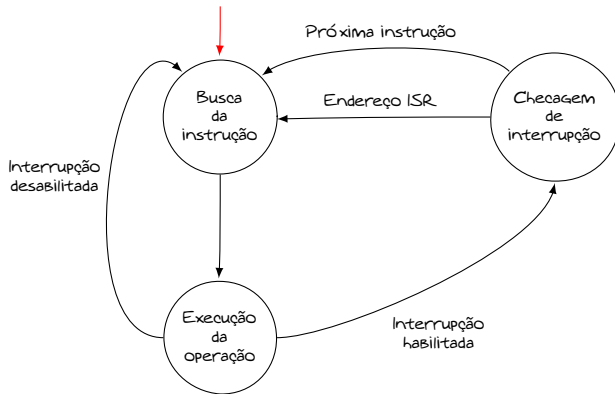
- ▶ Controle de fluxo para interrupção
 - ▶ Interrupção gerada durante execução da instrução i



Retorno ao fluxo anterior de execução

Interrupção

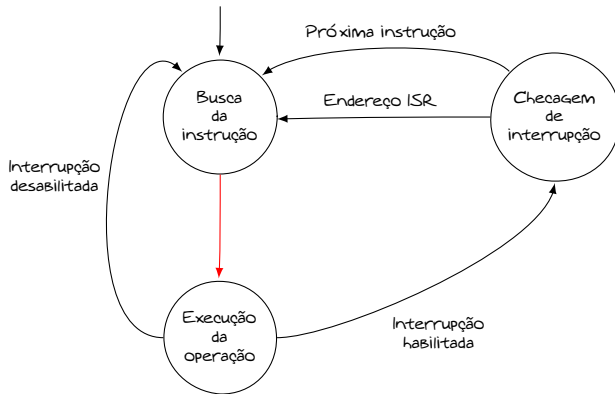
- ▶ Controle de fluxo para interrupção
 - ▶ Máquina de estados



Busca da próxima instrução

Interrupção

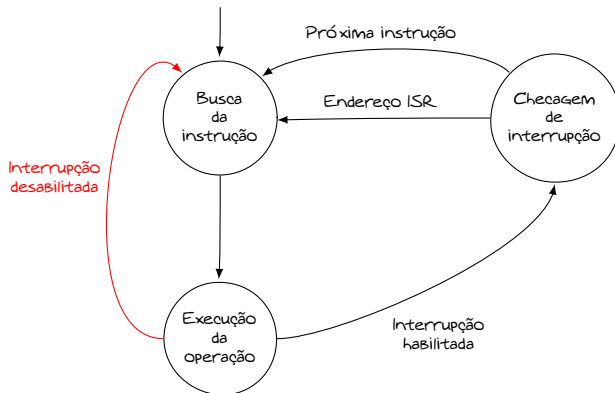
- ▶ Controle de fluxo para interrupção
 - ▶ Máquina de estados



Execução da operação

Interrupção

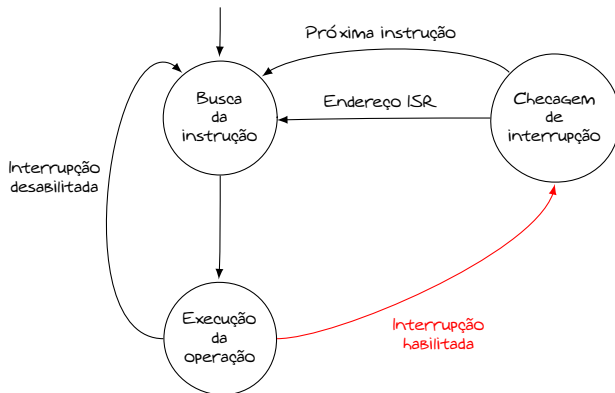
- ▶ Controle de fluxo para interrupção
 - ▶ Máquina de estados



Interrupções mascaráveis ficam pendentes

Interrupção

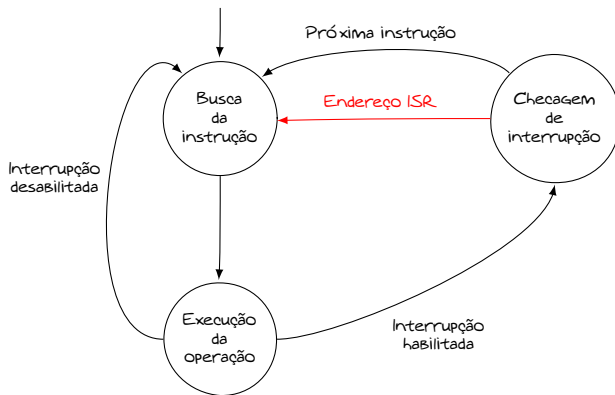
- ▶ Controle de fluxo para interrupção
 - ▶ Máquina de estados



É feita a checagem por interrupções pendentes

Interrupção

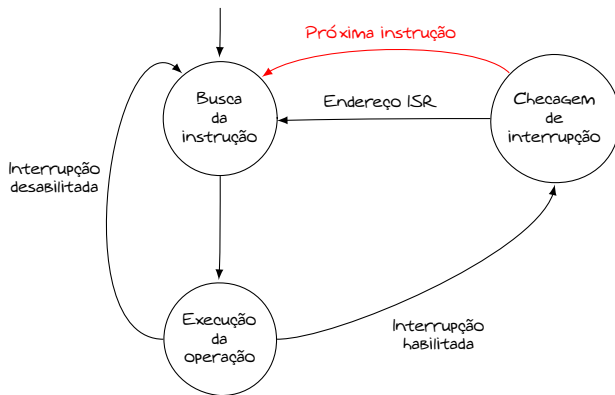
- ▶ Controle de fluxo para interrupção
- ▶ Máquina de estados



O endereço da ISR é atribuído ao PC, caso exista alguma interrupção pendente

Interrupção

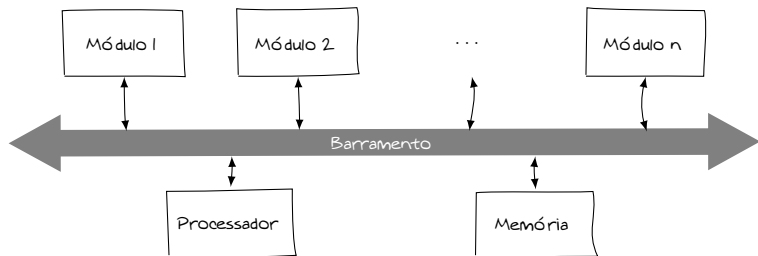
- ▶ Controle de fluxo para interrupção
 - ▶ Máquina de estados



Com nenhuma interrupção pendente,
a próxima instrução é buscada

Interrupção

- ▶ Eventos de interrupção de hardware
 - ▶ São requisições assíncronas de periféricos de E/S da plataforma que solicitam do processador a execução de rotinas para realizar a transferência de dados ou para realizar ações pré-definidas pelo programador



Interrupção

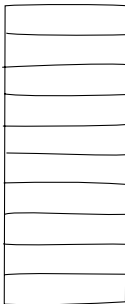
- ▶ Eventos de interrupção de software
 - ▶ Podem ser gerados explicitamente por instruções de interrupção ou implicitamente por exceções decorrentes de operações realizadas

```
1 // Função principal
2 main:
3     // Interrupção de software 7
4     int 7
5     // Divisão por zero (exceção)
6     divi r1, r2, 0
7     // Instrução inválida (exceção)
8     .4byte 0xF0F0F0F0
9     // Interrupção de software 0
10    int 0
```

Interrupção

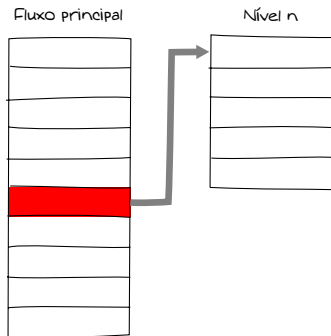
- ▶ Priorização das interrupções
 - ▶ Como as requisições são organizadas em uma fila de prioridade, é possível o aninhamento das ISRs

Fluxo principal



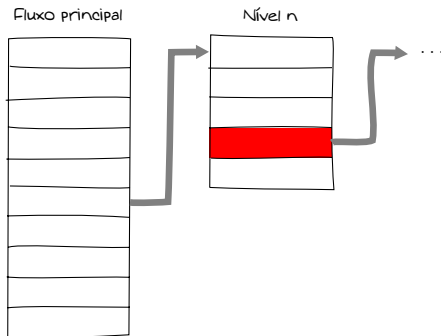
Interrupção

- ▶ Priorização das interrupções
 - ▶ Como as requisições são organizadas em uma fila de prioridade, é possível o aninhamento das ISRs



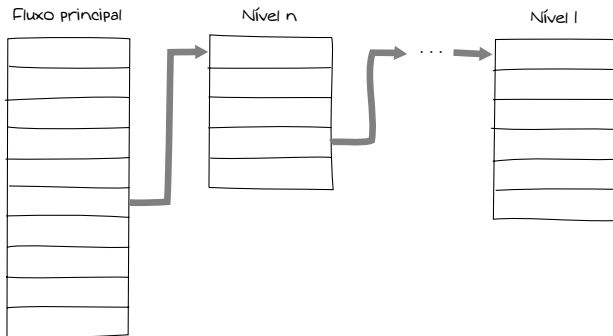
Interrupção

- ▶ Priorização das interrupções
 - ▶ Como as requisições são organizadas em uma fila de prioridade, é possível o aninhamento das ISRs



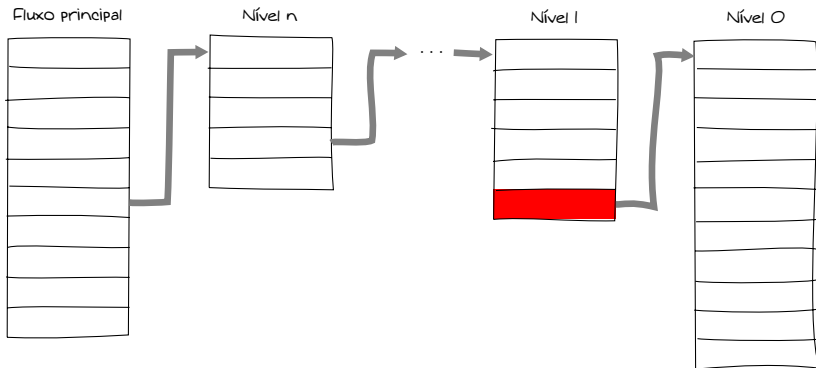
Interrupção

- ▶ Priorização das interrupções
 - ▶ Como as requisições são organizadas em uma fila de prioridade, é possível o aninhamento das ISRs



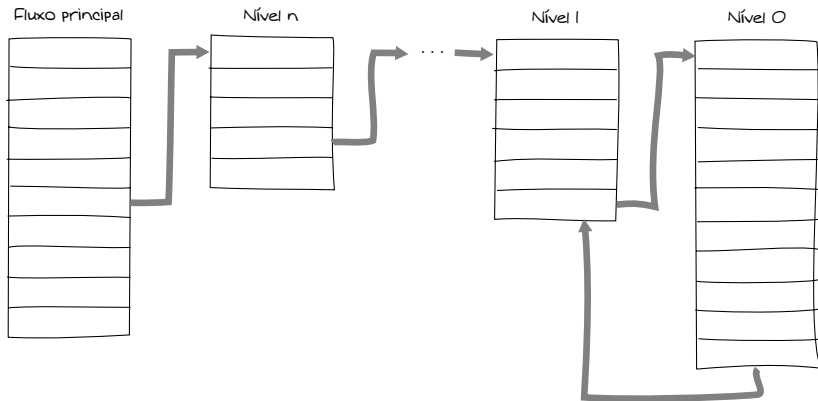
Interrupção

- ▶ Priorização das interrupções
 - ▶ Como as requisições são organizadas em uma fila de prioridade, é possível o aninhamento das ISRs



Interrupção

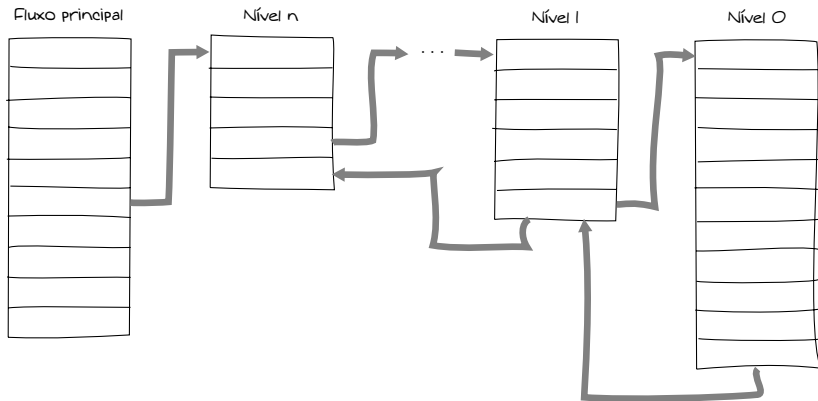
- ▶ Priorização das interrupções
 - ▶ Como as requisições são organizadas em uma fila de prioridade, é possível o aninhamento das ISRs



Geralmente os menores valores de nível possuem maior prioridade (0 - máxima e n - mínima)

Interrupção

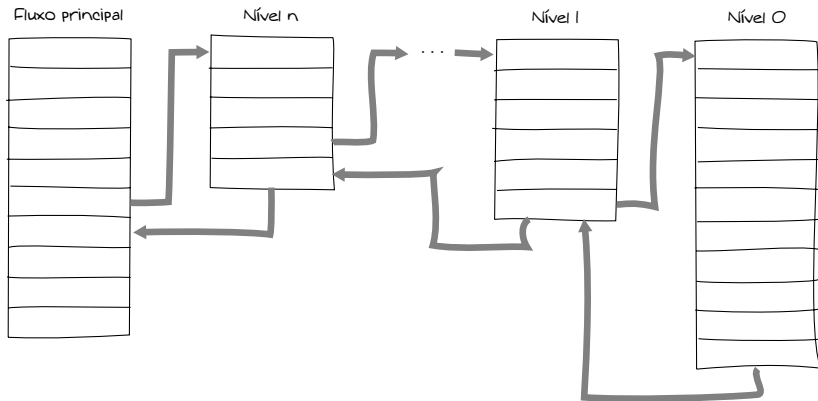
- ▶ Priorização das interrupções
 - ▶ Como as requisições são organizadas em uma fila de prioridade, é possível o aninhamento das ISRs



Geralmente os menores valores de nível possuem maior prioridade (0 - máxima e n - mínima)

Interrupção

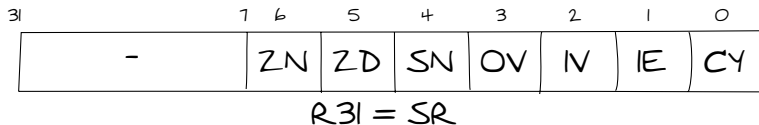
- ▶ Priorização das interrupções
 - ▶ Como as requisições são organizadas em uma fila de prioridade, é possível o aninhamento das ISRs



Geralmente os menores valores de nível possuem maior prioridade (0 - máxima e n - mínima)

Interrupção

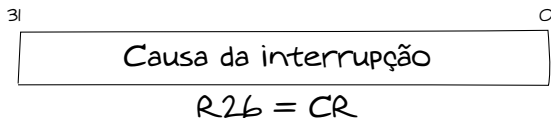
- ▶ Registrador de status (SR)
 - ▶ Índice 31



- ▶ Controle de interrupção (IE)
 - ▶ 0: Desabilitada
 - ▶ 1: Habilitado

Interrupção

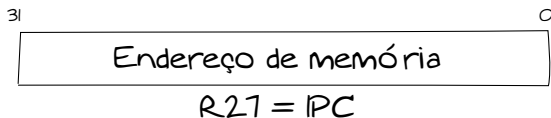
- ▶ Registrador de causa de interrupção (CR)
 - ▶ Índice 26



Armazena o código identificador das interrupções de hardware e de software

Interrupção

- ▶ Registrador de endereço de interrupção (IPC)
- ▶ Índice 27



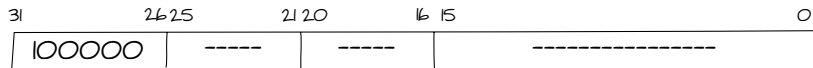
Armazena o endereço da instrução onde a interrupção foi gerada ou causada

Interrupção

- ▶ Preparação para execução da ISR
 - ▶ Este processo se inicia antes do processador realizar a preempção do fluxo de execução
 - ▶ Os valores dos registradores CR e IPC são salvos na pilha antes de receberem o código da causa e o endereço de retorno, respectivamente
 - ▶ $MEM[SP] = PC + 4, SP = SP - 4$
 - ▶ $MEM[SP] = CR, SP = SP - 4$
 - ▶ $MEM[SP] = IPC, SP = SP - 4$

Interrupção

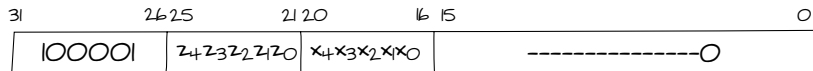
- ▶ Operação de retorno de interrupção (reti)
 - ▶ Tipo F
 - ▶ $SP = SP + 4, IPC = MEM[SP]$
 - ▶ $SP = SP + 4, CR = MEM[SP]$
 - ▶ $SP = SP + 4, PC = MEM[SP]$



Esta instrução deve ser utilizada
somente para retorno de ISR

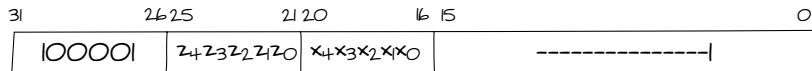
Interrupção

- ▶ Operação de limpeza de bit de registrador (cbr)
 - ▶ Tipo F
 - ▶ $R[z][x] = 0$



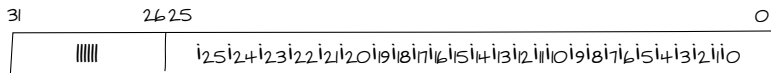
Interrupção

- ▶ Operação de ajuste de bit de registrador (sbr)
 - ▶ Tipo F
 - ▶ $R[z][x] = 1$



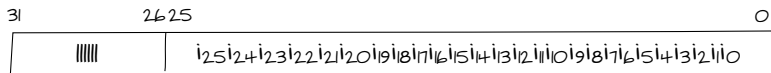
Interrupção

- ▶ Operação de interrupção de software (int)
 - ▶ Tipo S
 - ▶ $i = 0$, a simulação é finalizada
 - ▶ $i \neq 0 \rightarrow CR = i, IPC = PC, PC = 0x0000000C$



Interrupção

- ▶ Operação de interrupção de software (int)
 - ▶ Tipo S
 - ▶ $i = 0$, a simulação é finalizada
 - ▶ $i \neq 0 \rightarrow CR = i, IPC = PC, PC = 0x0000000C$



Não é mascarável

Interrupção

- ▶ Exceções de software
 - ▶ Instrução inválida
 - ▶ Não é mascarável
 - ▶ $IV = 1, CR = IR_{31:26}, IPC = PC, PC = 0x00000004$

Interrupção

- ▶ Exceções de software
 - ▶ Instrução inválida
 - ▶ Não é mascarável
 - ▶ $IV = 1, CR = IR_{31:26}, IPC = PC, PC = 0x00000004$
 - ▶ Divisão por zero
 - ▶ É mascarável e não fica pendente
 - ▶ $IE = 0 \rightarrow ZD = 1$
 - ▶ $IE = 1 \rightarrow ZD = 1, CR = 0, IPC = PC, PC = 0x00000008$

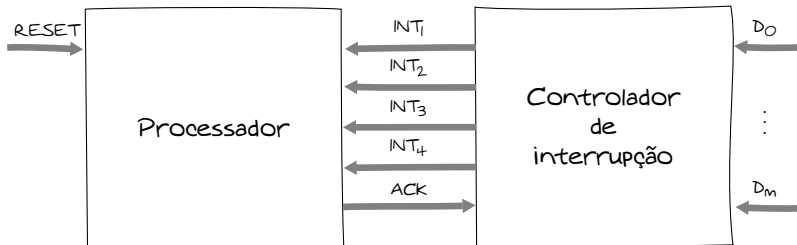
Interrupção

- ▶ Tabela de vetor de interrupção
 - ▶ Define o mapeamento dos endereços de ISR após a preempção do fluxo de execução

Tipo	Endereço	Mascarável	Prioridade
Inicialização	0x00000000	Não	0
Instrução inválida	0x00000004	Não	-
Divisão por zero	0x00000008	Sim	-
Software	0x0000000C	Não	-
Hardware 1	0x00000010	Sim	1
Hardware 2	0x00000014	Sim	2
Hardware 3	0x00000018	Sim	3
Hardware 4	0x0000001C	Sim	4

Interrupção

- ▶ Controlador de interrupção
 - ▶ Determina qual dispositivo deve ter sua requisição de interrupção atendida (fila de prioridade)



Interrupção

- ▶ Simulando interrupções de hardware e software
- ▶ Código de montagem

```
1 // Segmento de código
2 .text
3 // Tabela de vetor de interrupção
4 init:
5     bun main
6     bun isr
7     bun isr
8     bun isr
9     .align 5
10 // Rotina de tratamento de interrupção
11 isr:
12     // R1 = CR
13     mov r1, cr
14     // R2 = IPC
15     mov r2, ipc
16     // Retorno de ISR
17     reti
... ..
```

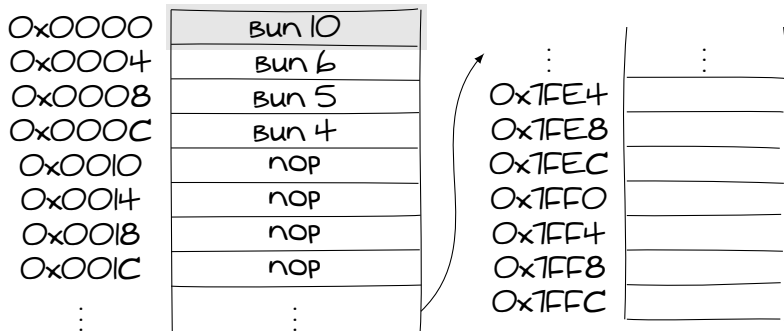
Interrupção

- ▶ Simulando interrupções de hardware e software
- ▶ Código de montagem

```
1 // Segmento de código
2 .text
...
18 // Função principal
19 main:
20     // SP = 32 KiB
21     mov sp, 0x7FFC
22     // Interrupção de software 5
23     int 5
24     // Habilitando interrupção (IE = 1)
25     sbr sr[1]
26     // Divisão por zero
27     div r1, r2, r0
28     // Instrução inválida
29     .4byte 0xF0F0F0F0
30     // Finalização de execução
31     int 0
```

Interrupção

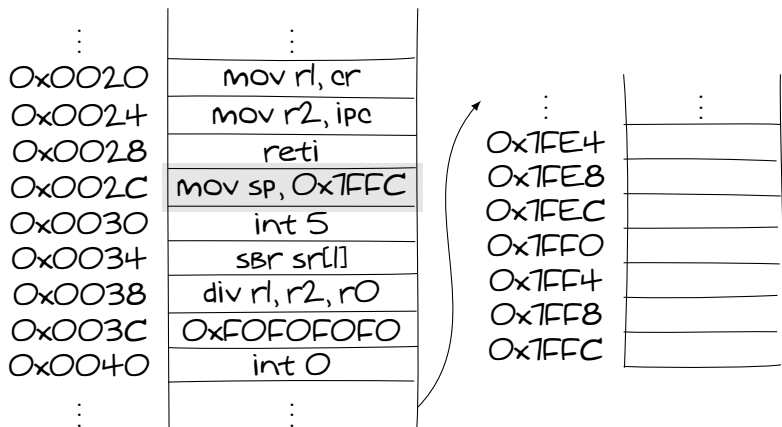
- ▶ Simulando interrupções de software
 - ▶ Tabela de vetor de interrupção



$R1 = 0, R2 = 0x0000, CR = 0, IPC = 0x0000,$
 $PC = 0x0000, SP = 0x0000, SR = 0x00$

Interrupção

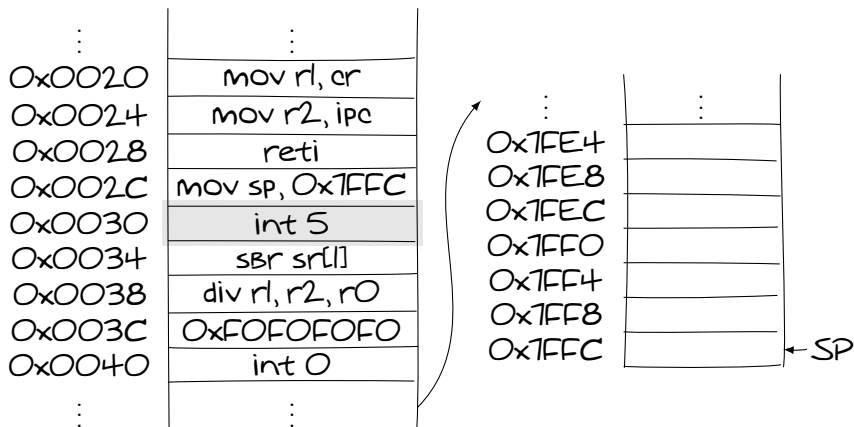
- ▶ Simulando interrupções de software
 - ▶ Funções ISR e principal



$R1 = 0, R2 = 0x0000, CR = 0, IPC = 0x0000,$
 $PC = 0x002C, SP = 0x0000, SR = 0x00$

Interrupção

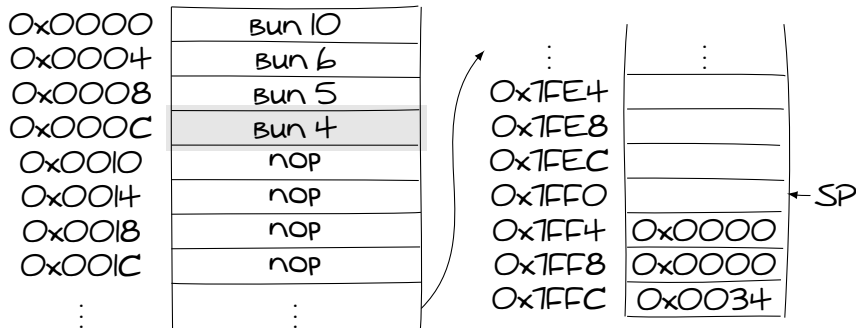
- ▶ Simulando interrupções de software
 - ▶ Funções ISR e principal



$R1 = 0, R2 = 0x0000, CR = 0, IPC = 0x0000,$
 $PC = 0x0030, SP = 0x7FFC, SR = 0x00$

Interrupção

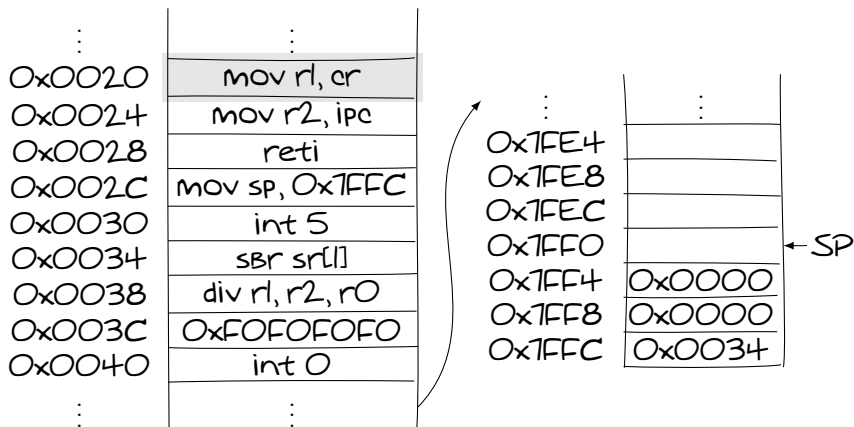
- ▶ Simulando interrupções de software
 - ▶ Interrupção de software 5



$R1 = 0, R2 = 0x0000, CR = 5, PC = 0x0030,$
 $PC = 0x000C, SP = 0x1FF0, SR = 0x00$

Interrupção

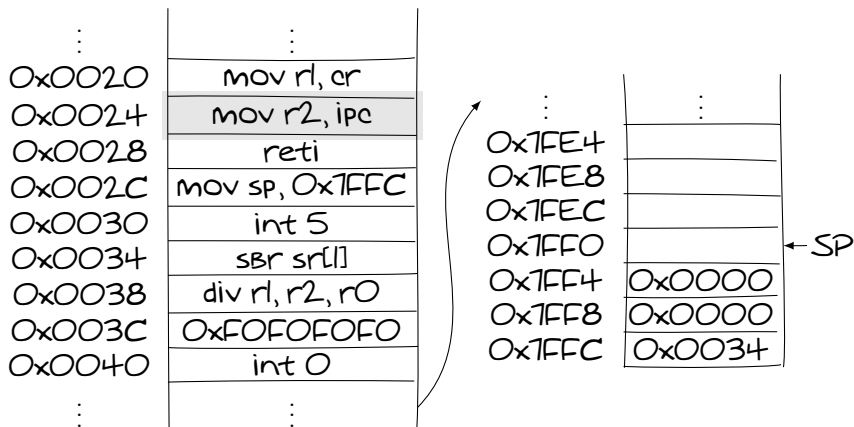
- ▶ Simulando interrupções de software
- ▶ Interrupção de software 5



$R1 = 0, R2 = 0x0000, CR = 5, IPC = 0x0030,$
 $PC = 0x0020, SP = 0x7FF0, SR = 0x00$

Interrupção

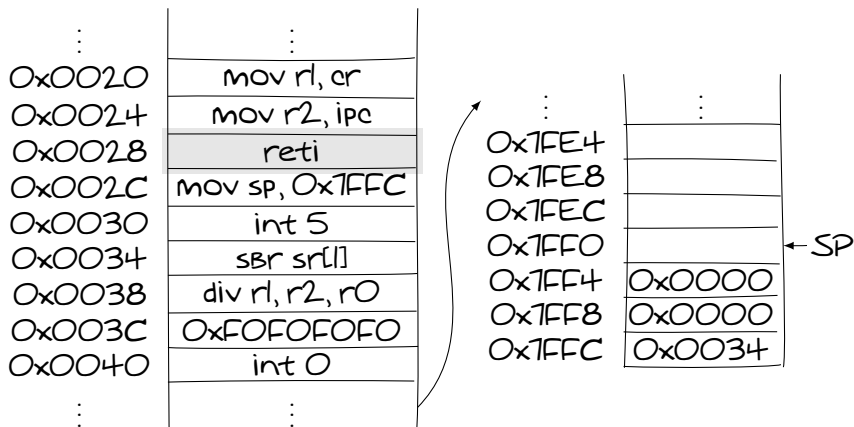
- ▶ Simulando interrupções de software
- ▶ Interrupção de software 5



$R1 = 5, R2 = 0x0000, CR = 5, IPC = 0x0030,$
 $PC = 0x0024, SP = 0x7FF0, SR = 0x00$

Interrupção

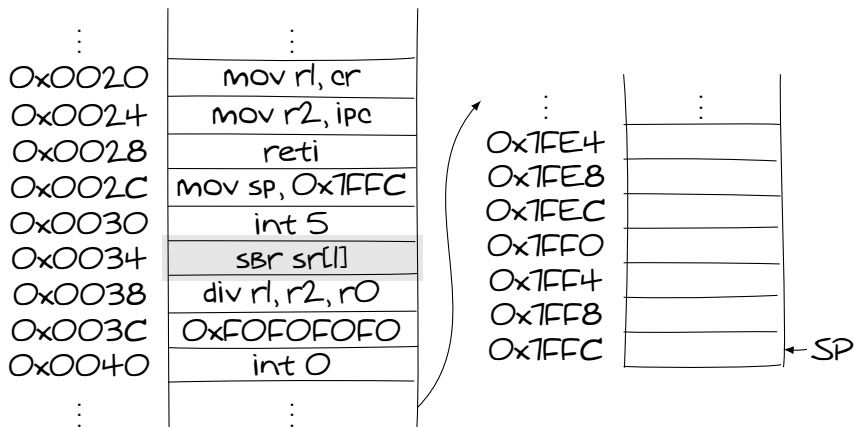
- ▶ Simulando interrupções de software
- ▶ Interrupção de software 5



$R1 = 5, R2 = 0x0030, CR = 5, IPC = 0x0030,$
 $PC = 0x0028, SP = 0x7FF0, SR = 0x00$

Interrupção

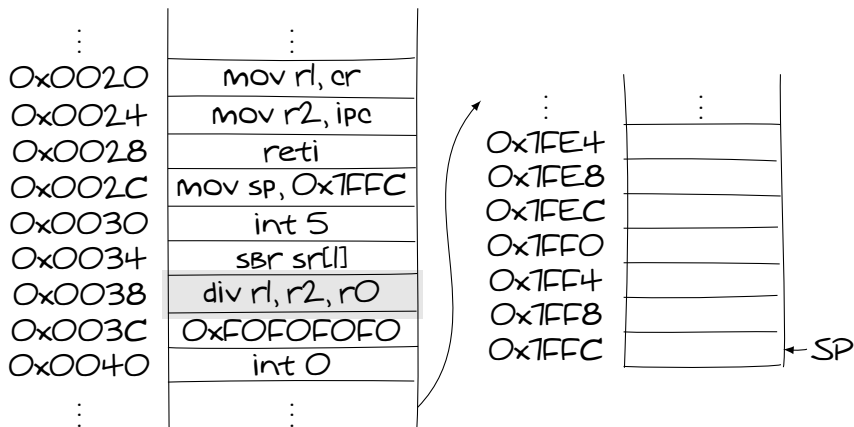
- ▶ Simulando interrupções de software
 - ▶ Interrupção de software 5



$R1 = 5, R2 = 0x0030, CR = 0, IPC = 0x0000,$
 $PC = 0x0034, SP = 0x7FFC, SR = 0x00$

Interrupção

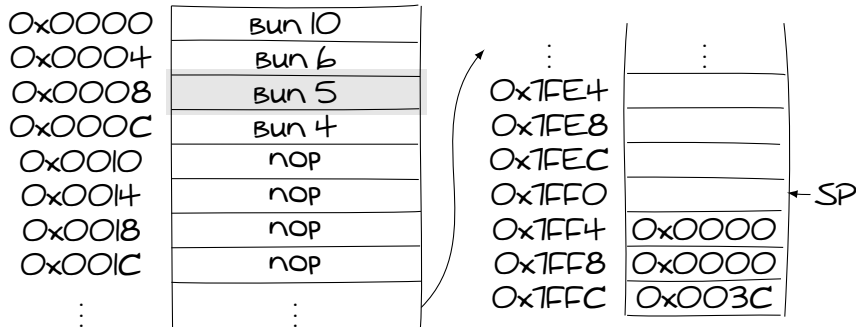
- ▶ Simulando interrupções de software
 - ▶ Exceção de divisão por zero



$R1 = 5, R2 = 0x0030, CR = 0, PC = 0x0000,$
 $PC = 0x0038, SP = 0x7FFC, SR = 0x02$

Interrupção

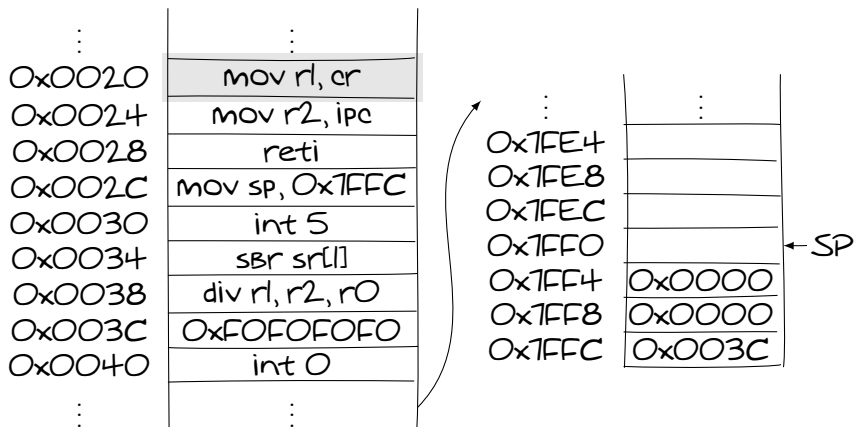
- ▶ Simulando interrupções de software
 - ▶ Exceção de divisão por zero



$R1 = 5, R2 = 0x0030, CR = 0, PC = 0x0038,$
 $PC = 0x0008, SP = 0x7FF0, SR = 0x22$

Interrupção

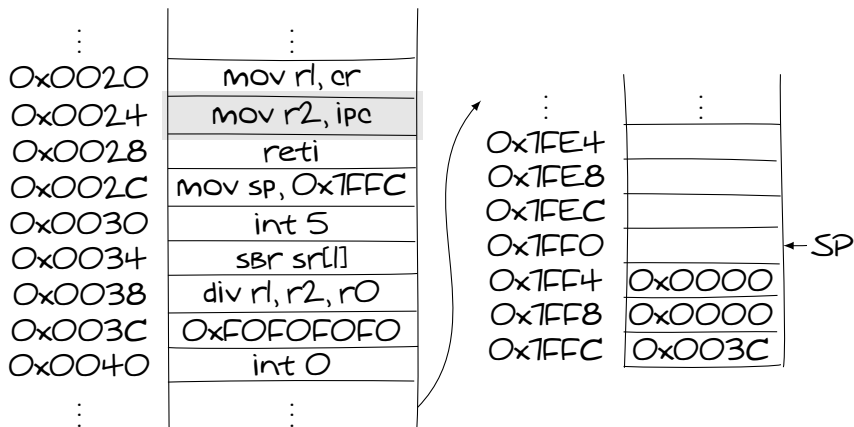
- ▶ Simulando interrupções de software
 - ▶ Exceção de divisão por zero



$R1 = 5, R2 = 0x0030, CR = 0, IPC = 0x0038,$
 $PC = 0x0020, SP = 0x7FF0, SR = 0x22$

Interrupção

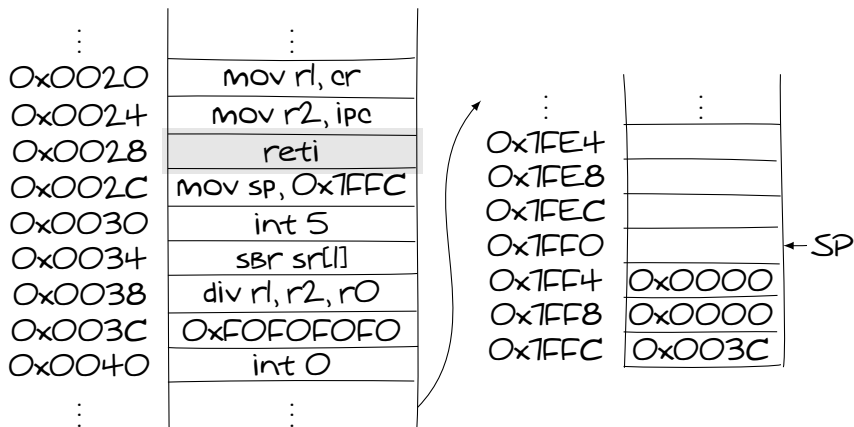
- ▶ Simulando interrupções de software
 - ▶ Exceção de divisão por zero



$R1 = 0, R2 = 0x0030, CR = 0, IPC = 0x0038,$
 $PC = 0x0024, SP = 0x7FF0, SR = 0x22$

Interrupção

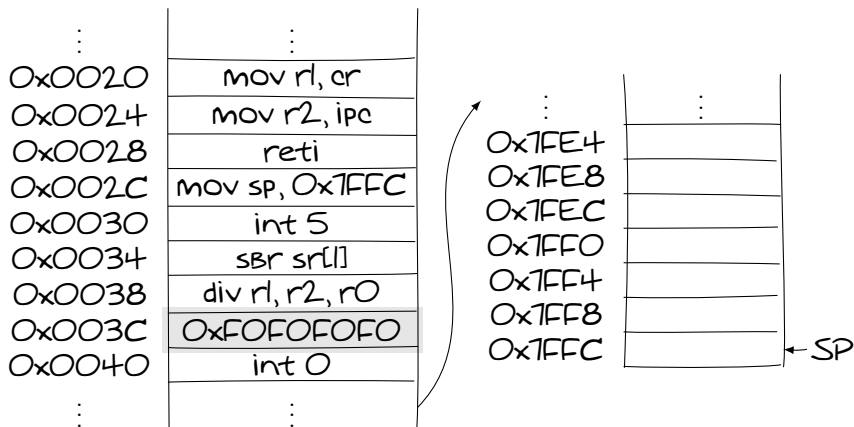
- ▶ Simulando interrupções de software
 - ▶ Exceção de divisão por zero



$R1 = 0, R2 = 0x0038, CR = 0, IPC = 0x0038,$
 $PC = 0x0028, SP = 0x7FF0, SR = 0x22$

Interrupção

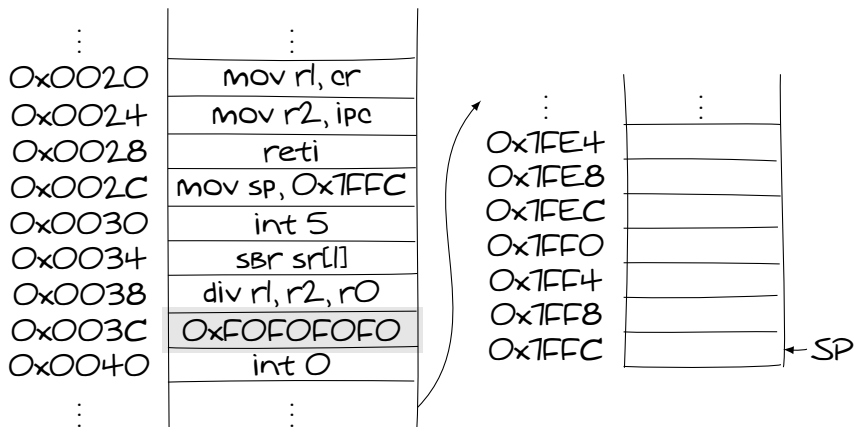
- ▶ Simulando interrupções de software
 - ▶ Exceção de divisão por zero



$R1 = 0, R2 = 0x0038, CR = 0, IPC = 0x0000,$
 $PC = 0x003C, SP = 0x7FFC, SR = 0x22$

Interrupção

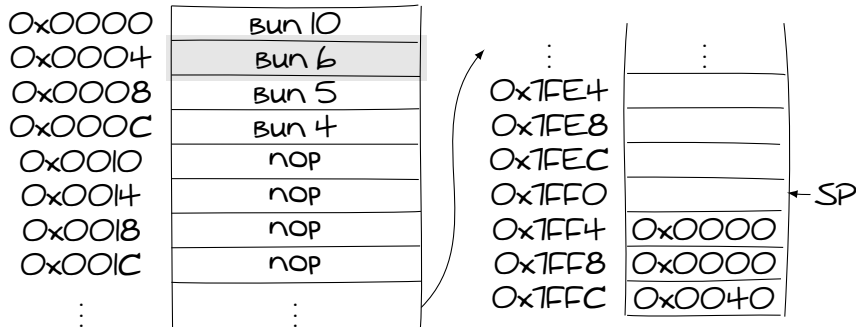
- ▶ Simulando interrupções de software
 - ▶ Exceção de instrução inválida



$R1 = 0x0000, R2 = 0x0038, CR = 0x003C,$
 $IPC = 0x3C, PC = 0x3C, SP = 0x7FFC, SR = 0x26$

Interrupção

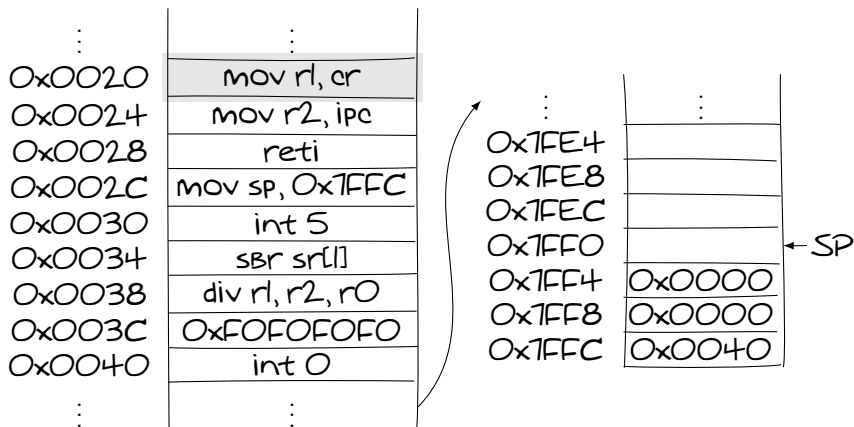
- ▶ Simulando interrupções de software
 - ▶ Exceção de instrução inválida



$RI = 0x0000, R2 = 0x0038, CR = 0x003C,$
 $IPC = 0x3C, PC = 0x04, SP = 0x7FF0, SR = 0x26$

Interrupção

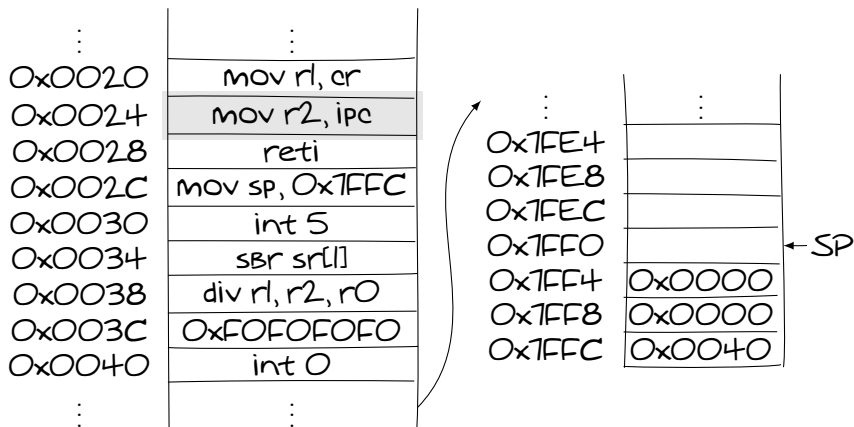
- ▶ Simulando interrupções de software
 - ▶ Exceção de divisão por zero



$R1 = 0x0000, R2 = 0x0038, CR = 0x003C,$
 $PC = 0x3C, PC = 0x04, SP = 0x7FF0, SR = 0x26$

Interrupção

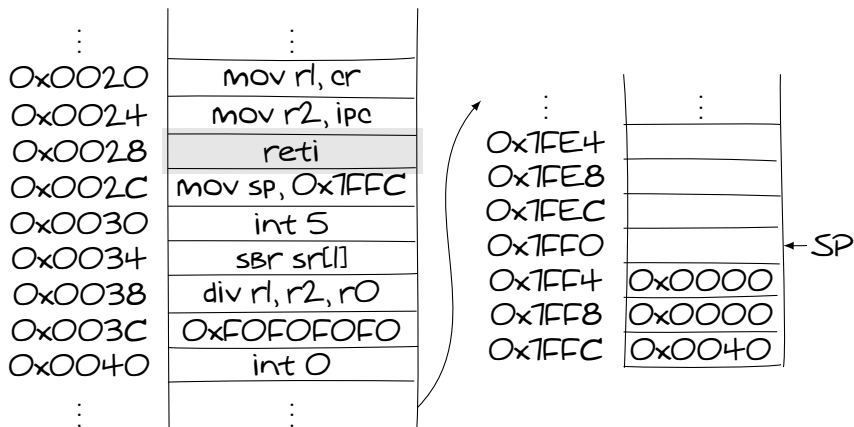
- ▶ Simulando interrupções de software
 - ▶ Exceção de divisão por zero



$R1 = 0x003C$, $R2 = 0x0038$, $CR = 0x003C$,
 $PC = 0x3C$, $PC = 0x04$, $SP = 0x7FF0$, $SR = 0x26$

Interrupção

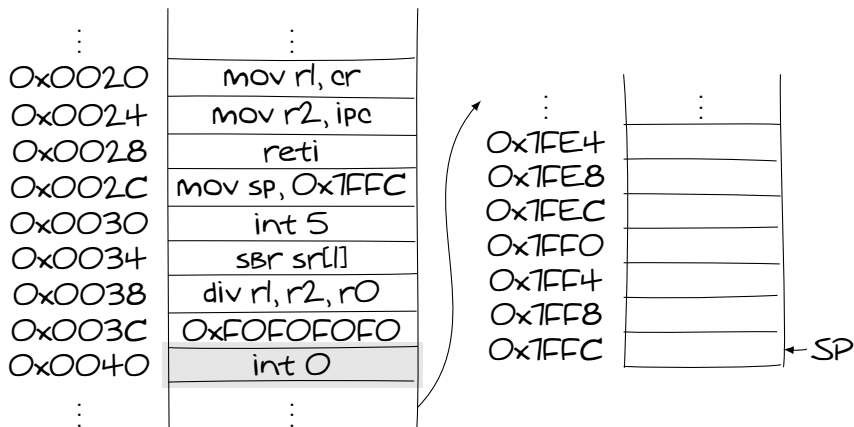
- ▶ Simulando interrupções de software
 - ▶ Exceção de divisão por zero



$R1 = 0x003C, R2 = 0x003C, CR = 0x003C,$
 $PC = 0x3C, PC = 0x04, SP = 0x7FF0, SR = 0x26$

Interrupção

- ▶ Simulando interrupções de software
 - ▶ Exceção de divisão por zero



$R1 = 0x003C, R2 = 0x003C, CR = 0x0000,$
 $IPC = 0x00, PC = 0x40, SP = 0x7FFC, SR = 0x26$