



Universidade de Brasília

Departamento de Ciência da Computação

Rars IO



Exceções e Interrupções

- Exceções são mudanças no fluxo de execução devido a eventos inesperados gerados internamente ao processador.
 - ecall : chamadas às rotinas do sistema
 - instrução inválida
 - # overflow em operações aritméticas – Não detectado no RISC-V
 - singularidades matemáticas da FPU
 - ...

- Interrupções são mudanças no fluxo devido a eventos externos, tipicamente dispositivos de entrada e saída.
 - DMA (Direct Memory Access)
 - acesso ao barramento
 - solicitação de dispositivos
 - ...

Definições dependentes do fabricante: Intel usa o termo interrupção,
Ex.: int 21h : Dá acesso a 108 chamadas do sistema (similar ao a7 ecall)



Tratamento de Interrupções e Exceções

Para tratar a Interrupção/Exceção, Sistema Operacional necessita:

1) Conhecer o fato que gerou a Exceção/Interrupção

■ Uso de Registrador de Causa

Um registrador especial é utilizado para codificar o motivo da Exceção/Interrupção.

■ Uso de Interrupção Vetorizada

Um vetor (endereço na memória) é utilizado para indicar os endereços para as rotinas de cada interrupção.

2) Conhecer o Endereço da instrução onde ocorreu a exceção/interrupção

■ Registrador específico (EPC)

3) Executar uma rotina capaz de acessar recursos que podem não estar disponíveis ao usuário:

■ Modo Usuário (*user mode*) : Modo não privilegiado. Não pode acessar memória de programa, algumas instruções não são permitidas. (Usuário com Sistema Operacional)

■ Modo Supervisor (*supervisor mode*): Modo privilegiado. Pode acessar a memória de programa, instruções específicas. (Sistema Operacional)

■ Modo Máquina (*machine mode*): Sem nenhuma limitação de acesso (Usuário sem Sistema Operacional).



Tratamento de Interrupções e Exceções

Ex.: Registrador de Causa

Programa

```

...
LABEL1: add      →  overflow: Cause=12 / EPC=PC+4
          sub
LABEL2: add      →  interrupção 3: Cause=0 / EPC=PC+4
...
  
```

Rotina de tratamento de exceção

ExceptionHandler:

```

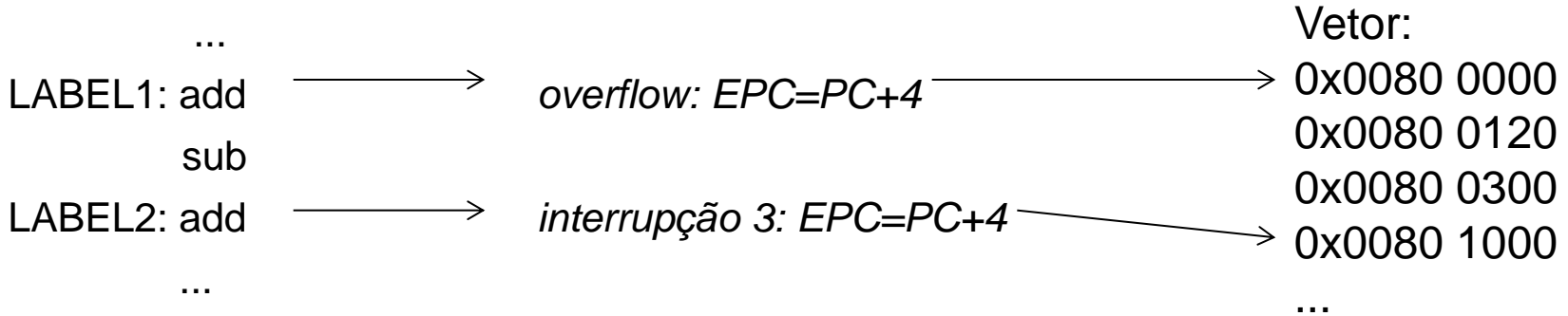
          se causa=0  então xxxx
          se causa=12 então yyyy
          ...
xxxx:....
          retorna
yyyy:....
          retorna
  
```



Tratamento de Interrupções e Exceções

Ex.: Interrupção Vetorizada

#Programa



Rotinas de tratamento 1

0x0080 0000:

retrona

...

#Rotina de tratamento 2

0x0080 1000:

retorna

...



Operações de Entrada e Saída

■ Por Polling: (software)

- processador testa periodicamente se dispositivo está pronto para realizar a transferência de dados
- problema: toma muito tempo do processador

■ Por Interrupção: (hardware)

- o dispositivo avisa ao processador a sua disponibilidade
- Problema: hardware mais complexo, processador deve suportar interrupções
 - antes de iniciar a execução da próxima instrução, processador verifica se existe alguma solicitação de interrupção
 - caso haja, interrompe o processamento normal e executa uma rotina de tratamento de interrupções



Operações de Entrada e Saída

■ Exemplo de Polling no RISC-V

```
WAIT:  lw   t0, STATUS(s0)           # lê estado do dispositivo s0
        andi t1, t0, MASK           # Isola o bit status por MASK
        beq t1, zero, WAIT          # se não está pronto repete
        lw   s1, DATA(s0)         # senão lê o dado para s1
```



Ferramentas de IO no Rars:

- Keyboard and Display MMIO Simulator
 - ☐ Entrada por leitura do teclado
 - ☐ Saída em terminal de texto
 - ☐ Simula IO por Polling ou por interrupção (não está Ok!)
- Bitmap Display
 - ☐ Saída gráfica em display VGA
 - ☐ Resolução selecionável.
 - ☐ Acesso direto à Memória de Vídeo (sem GPU)
- Interface de Áudio
 - ☐ Saída de áudio por sintetizador MIDI (*Musical Instrument Digital Interface*)
 - ☐ ecall 31, 32 e 33



Keyboard and Display MMIO Simulator

■ Endereço

Função

0xFF20 0000

bit 0 → Status do teclado

bit 1 → Define Interrupção ou Polling

0xFF20 0004

bits 7-0 → Código ASCII da tecla

0xFF20 0008

bit 0 → Status do monitor

bit 1 → Define Interrupção ou Polling

0xFF20 000C

bits 7-0 → Define ASCII do caractere

ASCII: 12 clear screen

ASCII: 7 posiciona cursor em (x,y)

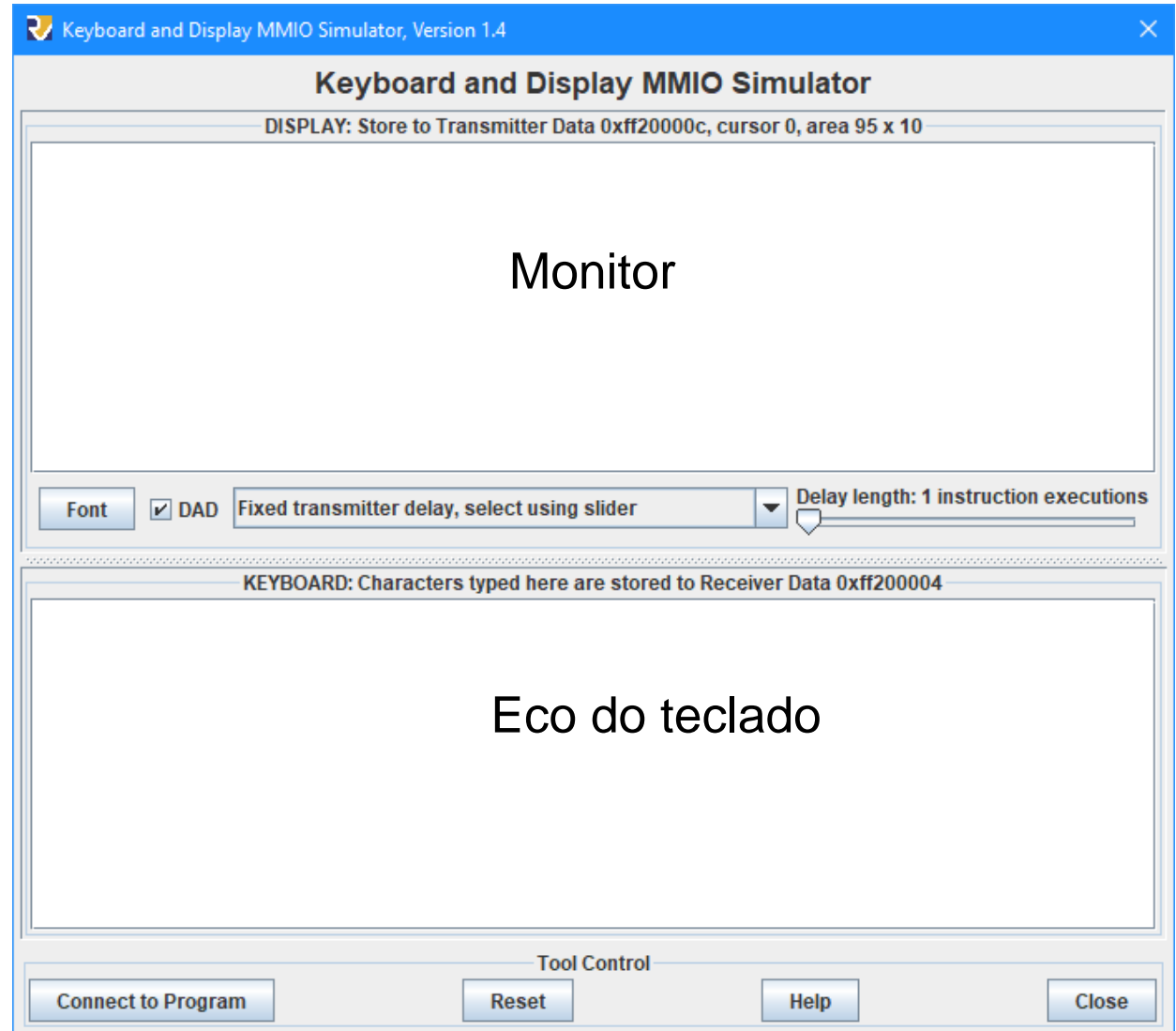
x: bits 31-20

y: bits 19-8

Keyboard and Display MMIO Simulator

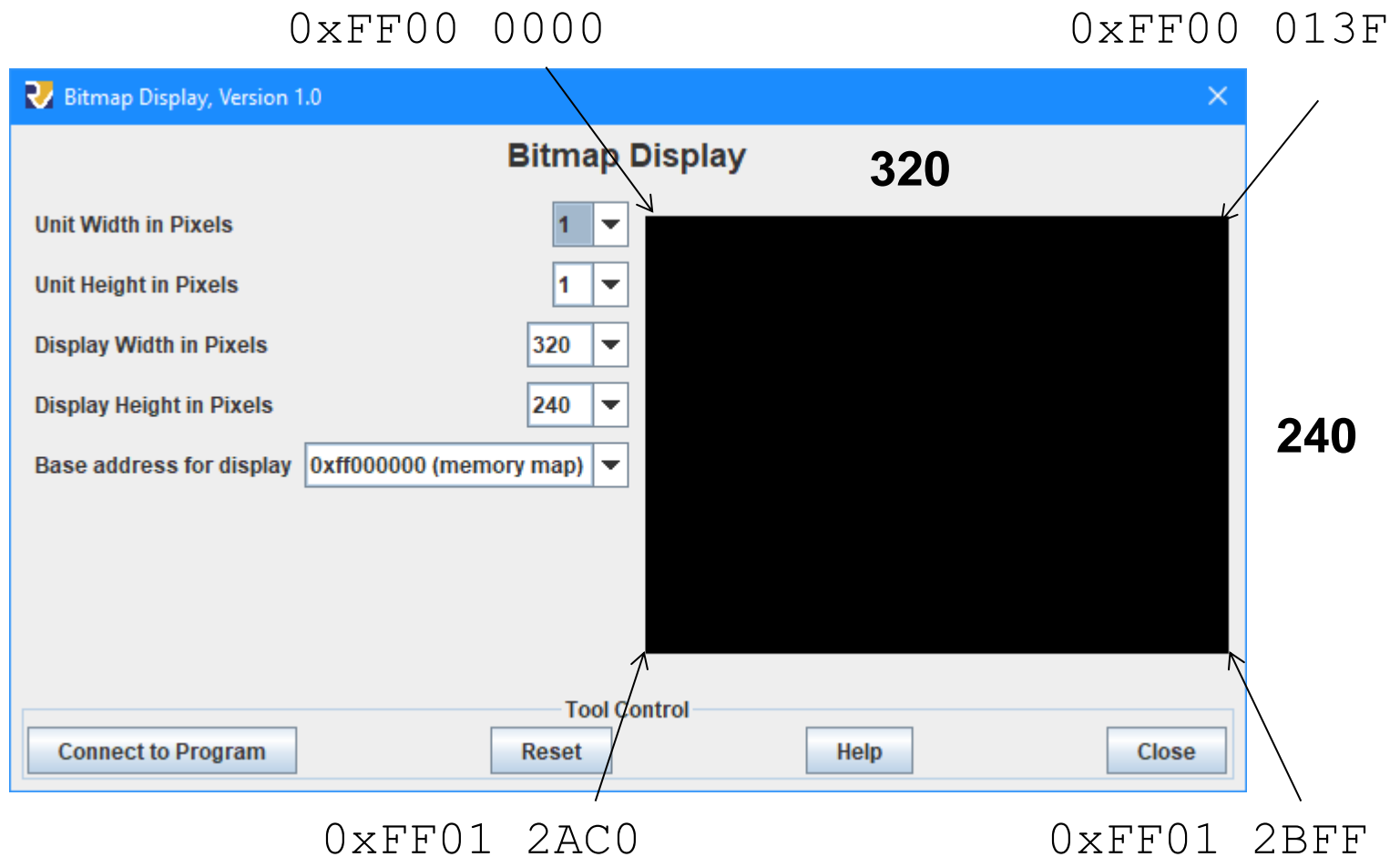
■ Exemplos:

- ☐ keypoll.s
- ☐ keyint.s



Bitmap Display

- Mapeamento da memória de vídeo VGA no RARS



Bitmap Display

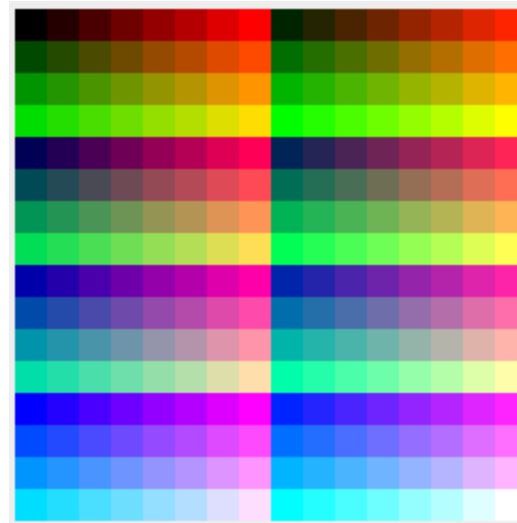
- Endereço(X,Y) = Endereço Base (0xFF00 0000) + Y * 320 + X
- Com X de 0 a 319 e Y de 0 a 239.

- Codificação da Cor: 8 bits/pixel

7 6	5 4 3	2 1 0
B B	G G G	R R R

- Ex.: bitmap.s

Paleta de cores



ecalls usando o Bitmap Display

- Inclua no seu programa no Rars o arquivo SYSTEMv1.s

Serviço	a7	Argumentos	Resultados
print integer	101	a0=inteiro a1=coluna a2=linha a3=cores	Imprime o número inteiro complemento de 2 a0 na posição (a1,a2) com as cores a3={0...0BBGGGRRRbbgggrrr} sendo BGR fundo e bgr frente
print float	102	fa0=float a1=coluna a2=linha a3=cores	Imprime o número float em fa0 na posição (a1,a2) com as cores a3
print string	104	a0=endereço string a1=coluna a2=linha a3=cores	Imprime a string terminada em NULL presente no endereço a0 na posição (a1,a2) com as cores a3
print char	111	a0=char (ASCII) a1=coluna a2=linha a3=cores	Imprime o caractere a0 (ASCII) na posição (a1,a2) com as cores a3
print int hex	134	a0=inteiro a1=coluna a2=linha a3=cores	Imprime em hexadecimal o número em a0 na posição (a1,a2) com as cores a3



Sintetizador de Áudio MIDI

- MIDI: *Musical Instrument Digital Interface*
- Protocolo de comunicação com instrumentos musicais
- Define 128 instrumentos, 128 notas, efeitos especiais, etc.
- No Rars:
 - ☐ ecall 33: Melodia (bloqueante)
 - ☐ ecall 32: Pausa (bloqueante)
 - ☐ ecall 31: Acorde (não-bloqueante)
- Ex.: midi.s