



**Universidade de Brasília**

Departamento de Ciência da Computação

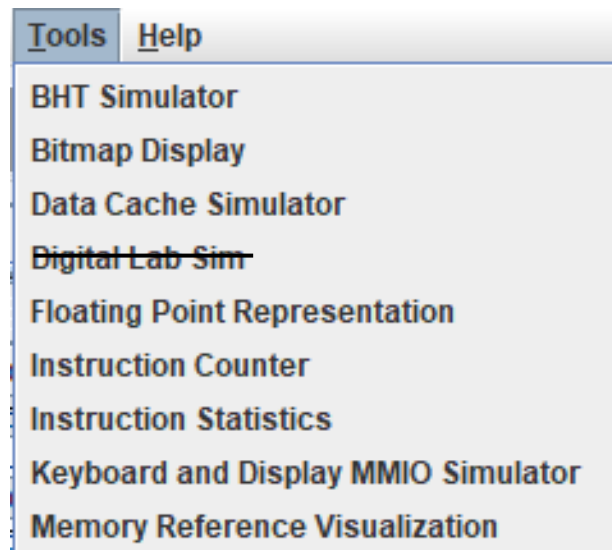
**Rars**

**RISC-V Assembler and Runtime  
Simulator**



# RARS

- O Rars é uma ferramenta que permite:
  - Escrita de um programa em Assembly RISC-V (Editor de Texto)
  - Montagem para código de máquina (Montador) e
  - Execução de um programa na ISA RV32IMF (Simulador).
  - Serviços básicos de sistema (Micro Sistema Operacional)
- Contém diversas ferramentas adicionais (Menu tools)





# Operações de Entrada e Saída

## ■ Por Polling: (software)

- ☐ O 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



# Operações de Entrada e Saída

## ■ Exemplo de Polling no RISC-V

Endereço BASE no registrador `s0`

No endereço `BASE+STATUS` temos a sinalização do dispositivo

No endereço `BASE+DATA` temos o dado enviado pelo dispositivo

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



# Ferramentas de IO hardware no Rars:

## ■ Keyboard and Display MMIO Simulator

- ☐ Entrada por leitura do teclado
- ☐ Saída em terminal de texto (não usada)
- ☐ Simula IO por Polling (Ok!) ou por Interrupção (não está Ok!)

## ■ Bitmap Display

- ☐ Saída gráfica em display VGA
- ☐ Resolução selecionável (mas usem sempre 320×240).
- ☐ 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

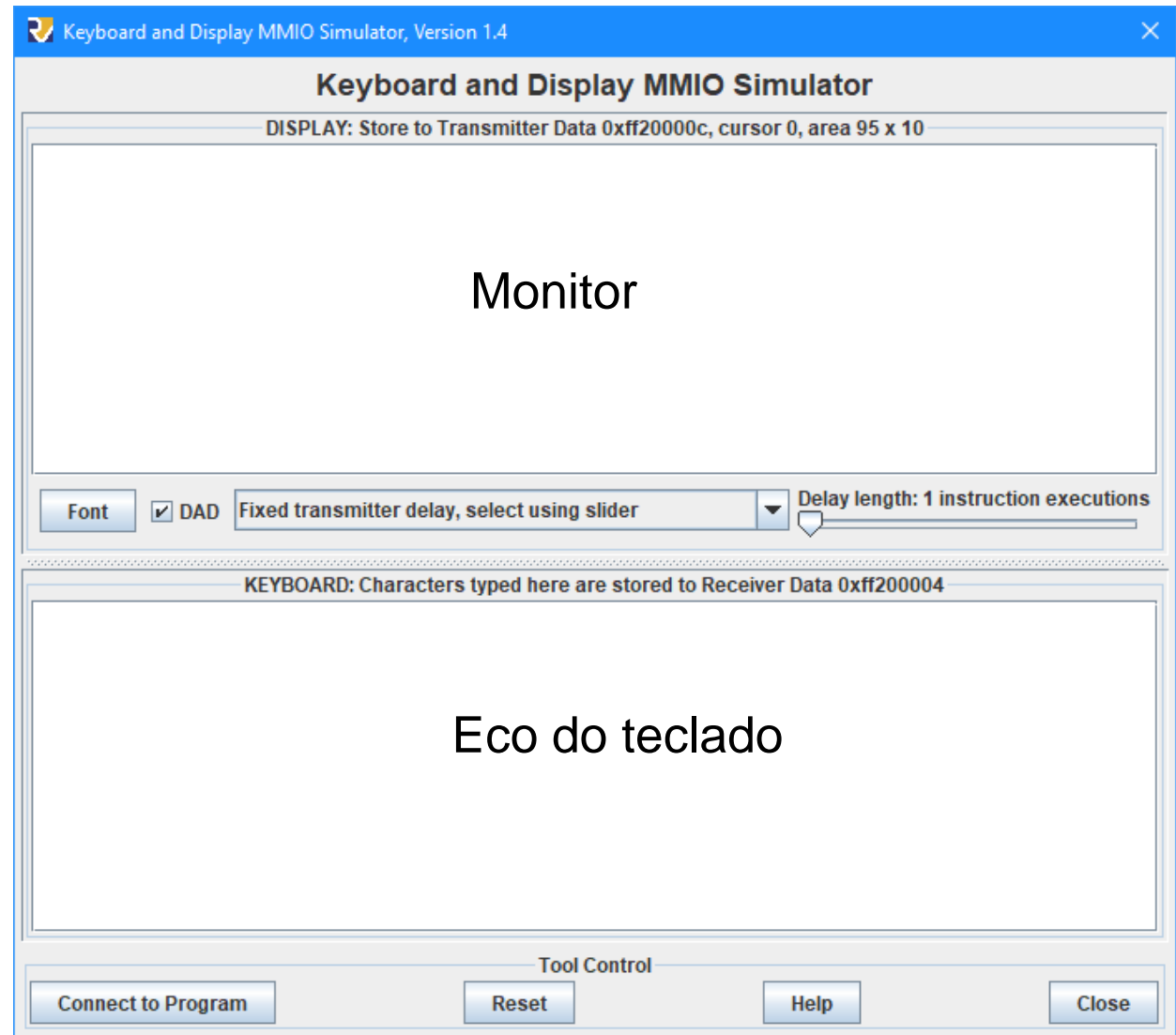
ASCII: 7 posiciona cursor em (x,y)

x: bits 31-20

y: bits 19-8

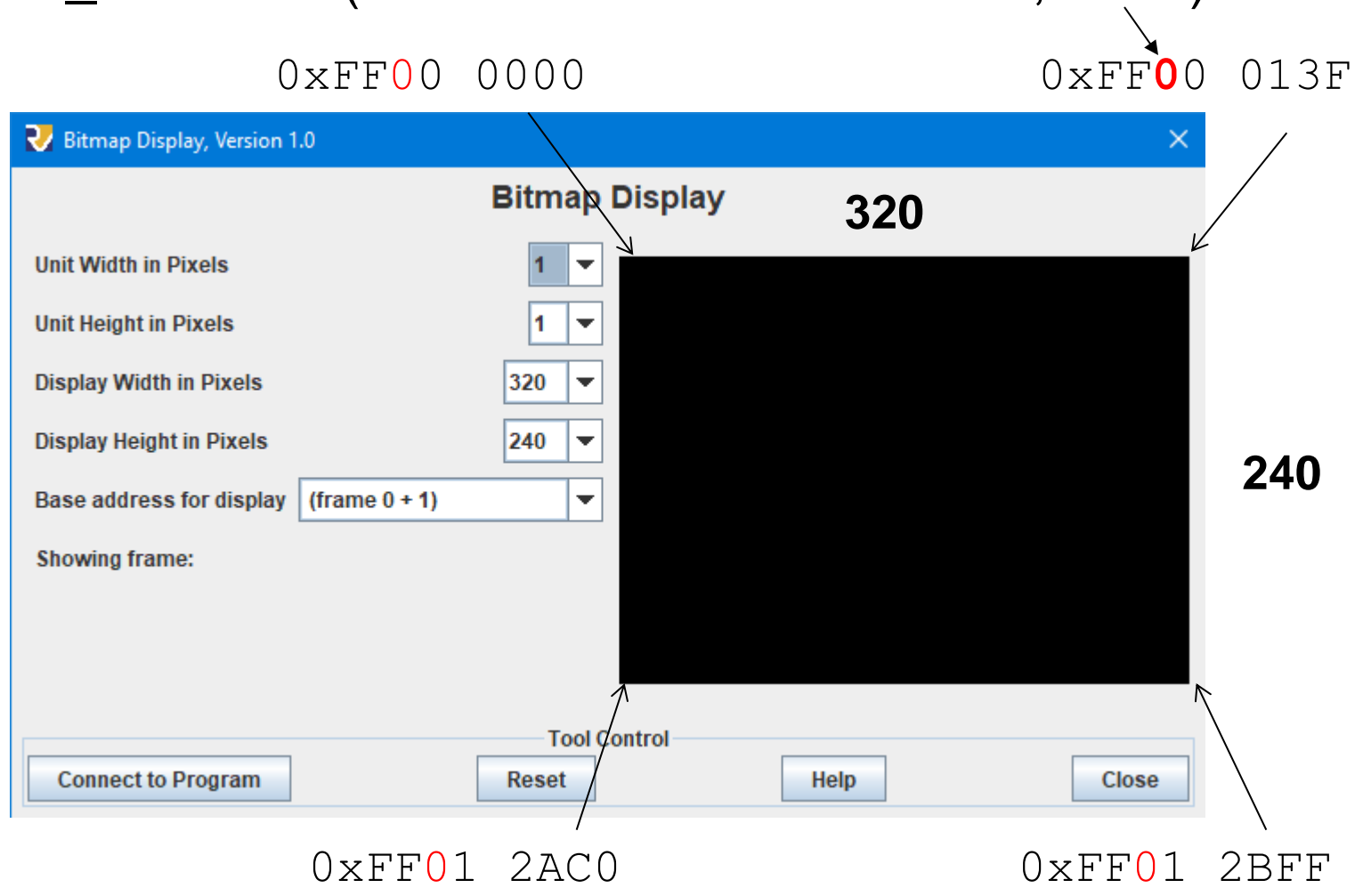
# Keyboard and Display MMIO Simulator

- Exemplo:
  - `keypoll.s`



# Bitmap Display

- Mapeamento da memória de vídeo VGA no Rars12\_Custom2 (com duas frames de vídeo, 0 e 1)





# Bitmap Display

- A frame que está sendo apresentada é selecionável escrevendo 0 ou 1 no endereço 0xFF200604
- $\text{Endereço}(X,Y) = \text{Endereço Base (0xFF00 0000)} + Y \times 320 + X$
- Com X de 0 a 319 e Y de 0 a 239.

- Codificação da Cor: 8 bits/pixel

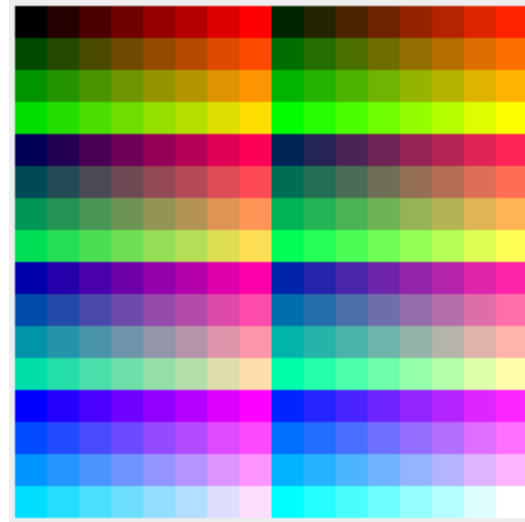
|            |              |              |
|------------|--------------|--------------|
| <b>7 6</b> | <b>5 4 3</b> | <b>2 1 0</b> |
| B B        | G G G        | R R R        |

- Exemplos.:

bitmap.s

frames.s

**Paleta de cores**



# ecalls usando o Bitmap Display

- Inclua no seu programa no Rars os arquivos macros2.s e SYSTEMv13.s

| Serviço       | a7  | Argumentos  | Resultados   |
|---------------|-----|---|--|
| print integer | 101 | a0=inteiro<br>a1=coluna<br>a2=linha<br>a3=cores<br>a4=frame         | Imprime o número inteiro complemento de 2 a0 na posição (a1,a2) da frame a4 com as cores a3={0...0BBGGGRRRbbgggrrr} sendo BGR fundo e bgr frente |
| print float   | 102 | fa0=float<br>a1=coluna<br>a2=linha<br>a3=cores<br>a4=frame          | Imprime na frame a4 o número float em fa0 na posição (a1,a2) com as cores a3   |
| print string  | 104 | a0=endereço string<br>a1=coluna<br>a2=linha<br>a3=cores<br>a4=frame | Imprime na frame a4 a string terminada em NULL presente no endereço a0 na posição (a1,a2) com as cores a3  |
| print char    | 111 | a0=char (ASCII)<br>a1=coluna<br>a2=linha<br>a3=cores<br>a4=frame    | Imprime na frame a4 o caractere a0 (ASCII) na posição (a1,a2) com as cores a3  |
| print int hex | 134 | a0=inteiro<br>a1=coluna<br>a2=linha<br>a3=cores<br>a4=frame         | Imprime na frame a4 em hexadecimal o número em a0 na posição (a1,a2) com as cores a3   |

Ex.: testeECALLv13.s



# 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`