



**Universidade de Brasília**

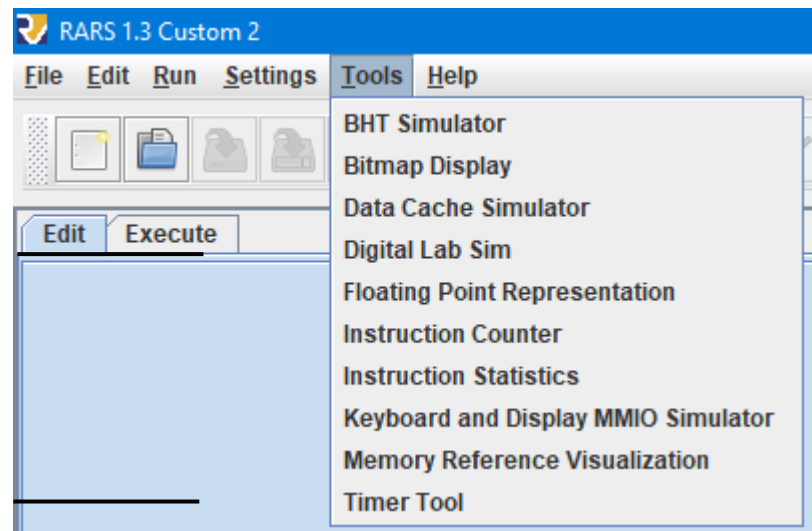
Departamento de Ciência da Computação

**Rars**

**RISC-V Assembler and Runtime  
Simulator**

# RARS 1.6 Custom 1

- O Rars é uma ferramenta que permite:
  - Escrita de um programa em Assembly RISC-V (Editor de Texto)
  - Montagem de Assembly para código de máquina (Montador)
  - Execução de um programa na ISA RV32IMFD ou RV64IMFD (Simulador)
  - Serviços básicos de sistema (Microssistema 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
- ☐ Simula IO por Polling ou por Interrupção

## ■ Bitmap Display

- ☐ Saída gráfica em display VGA
- ☐ Resolução selecionável (mas usem sempre 320×240)
- ☐ Acesso direto à Memória de Vídeo (sem GPU)

## ■ Sintetizador de Áudio

- ☐ Saída de áudio por sintetizador MIDI (*Musical Instrument Digital Interface*)
- ☐ Definição do instrumento, nota, duração e volume

# 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

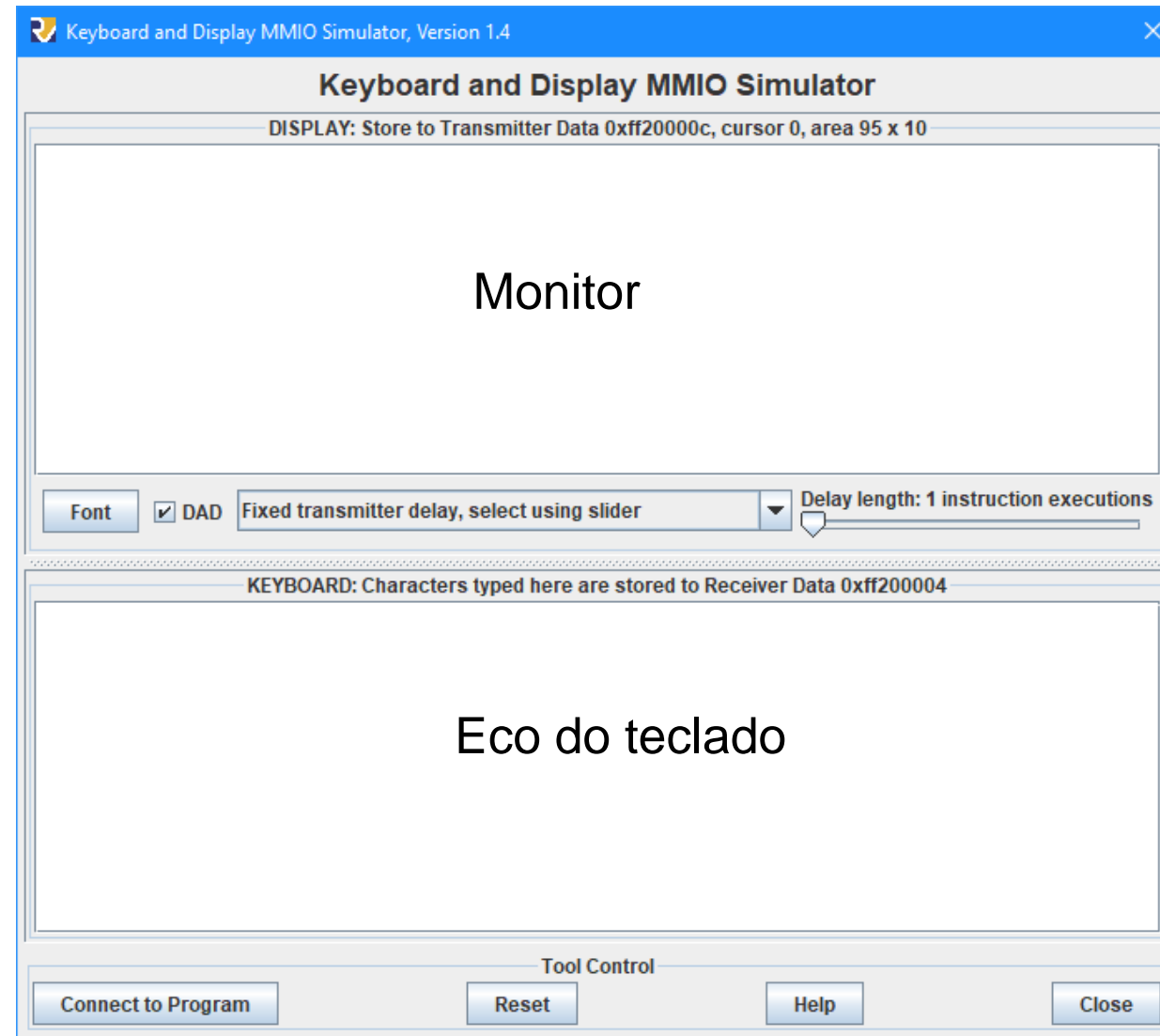
y: bits 19-8



# Keyboard and Display MMIO Simulator

## ■ Exemplos:

- ☐ `keypoll.s`
- ☐ `keyint.s`





# ecalls usando o Keyboard and Display

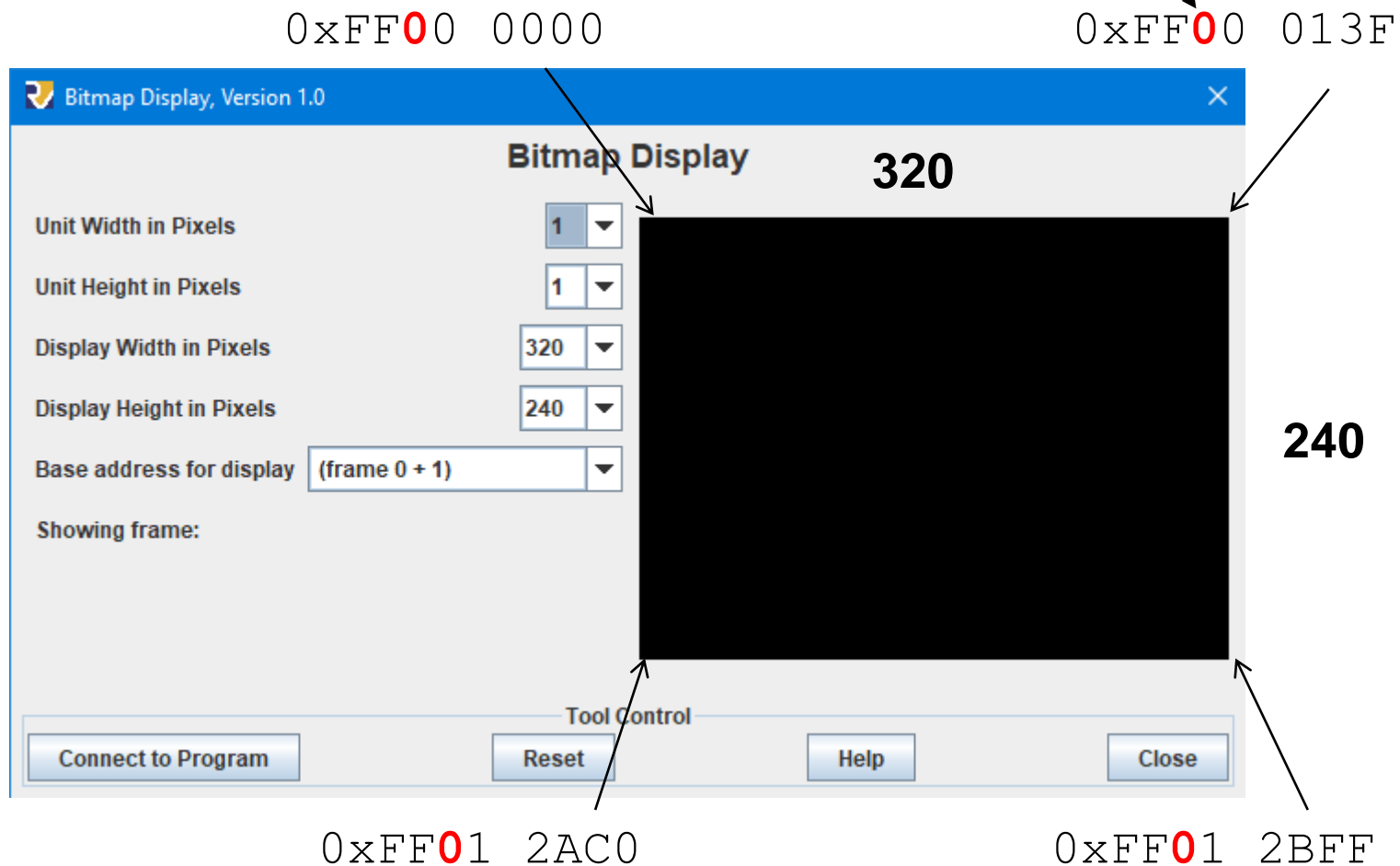
- Inclua no seu programa no Rars os arquivos MACROv21.s e SYSTEMv21.s

Serviço	a7	Argumentos	Resultados
read integer	105		Lê um número inteiro do teclado para a0
read float	106		Lê um número float do teclado para fa0
read string	108	a0=endereço do buffer da string a1=número máximo de caracteres	Lê uma string de até a1 caracteres do teclado para o buffer em a0
read char	112		Lê um caractere ASCII do teclado para a0



# Bitmap Display

- Mapeamento da memória de vídeo VGA no Rars16\_Custom1 (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

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

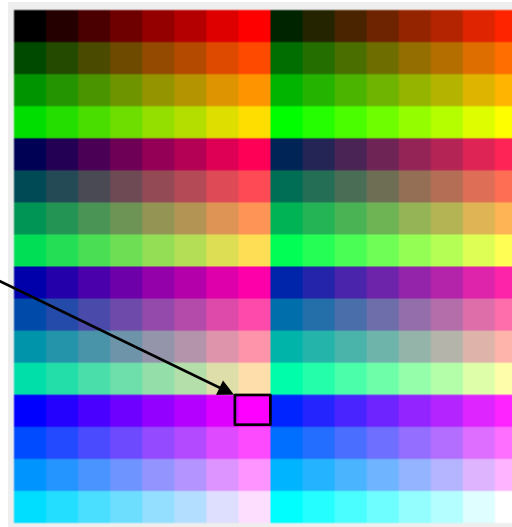
Obs.: a cor 0xC7 é transparente!

- Exemplos:

`bitmap.s`

`frames.s`

**Paleta de cores**





# ecalls usando o Bitmap Display

- Inclua no seu programa no Rars os arquivos MACROsv21.s e SYSTEMv21.s

Serviço	a7	Argumentos	Resultados
print integer	101	a0=inteiro a1=coluna a2=linha a3=cores a4=frame	Imprime o número inteiro complemento de 2 a0 na posição (a1,a2) da frame a4 com as cores a3={0...0BBGGRRRbbgggrrr} sendo BGR fundo e bgr frente
print float	102	fa0=float a1=coluna a2=linha a3=cores 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 a1=coluna a2=linha a3=cores 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) a1=coluna a2=linha a3=cores a4=frame	Imprime na frame a4 o caractere a0 (ASCII) na posição (a1,a2) com as cores a3
print int hex	134	a0=inteiro a1=coluna a2=linha a3=cores a4=frame	Imprime na frame a4 em hexadecimal o número em a0 na posição (a1,a2) com as cores a3

# ecalls usando o Bitmap Display

Serviço	A7	Argumentos	Resultados
print int unsigned	36 136	a0=inteiro a1=coluna a2=linha a3=cores a4=frame	Imprime na frame a4 o número sem sinal em a0 na posição (a1,a2) com as cores a3
clear screen	48 148	a0=cor a1=frame	Preenche a frame a1 com a cor a0
draw line	47 147	a0=x0 a1=y0 a2=x1 a3=y1 a4=cor a5=frame	Desenha uma reta na frame a5 do ponto (a0,a1) ao ponto (a2,a3) com a cor a4

Ex.: testeECALLv21.s



# Sintetizador de Áudio MIDI

- MIDI: *Musical Instrument Digital Interface*
- Protocolo de comunicação com instrumentos musicais
- Define 128 instrumentos, 128 notas, efeitos especiais, etc.
- Inclua no seu programa no Rars os arquivos MACROsv21.s e SYSTEMv21.s

Serviço	a7	Argumentos	Resultados
MidiOut	31	a0=nota a1=duração a2=instrumento a3=volume	Sintetiza a nota a0, de duração a1, com o instrumento a2 e volume a3, e retorna ao programa enquanto a nota é tocada. Na DE1-SoC o instrumento é fixo.
MidiOutSync	33	a0=nota a1=duração a2=instrumento a3=volume	Sintetiza a nota a0, de duração a1, com o instrumento a2 e volume a3, e pausa o programa até o final da execução da nota Na DE1-SoC o instrumento é fixo.
Sleep	32	a0=tempo	Pausa a execução do programa por a0 milissegundos

Ex.: `midi.s`