

Computer Systems and Architecture

MIPS: Memory-Mapped I/O

Stephen Pauwels

Academic Year 2018-2019

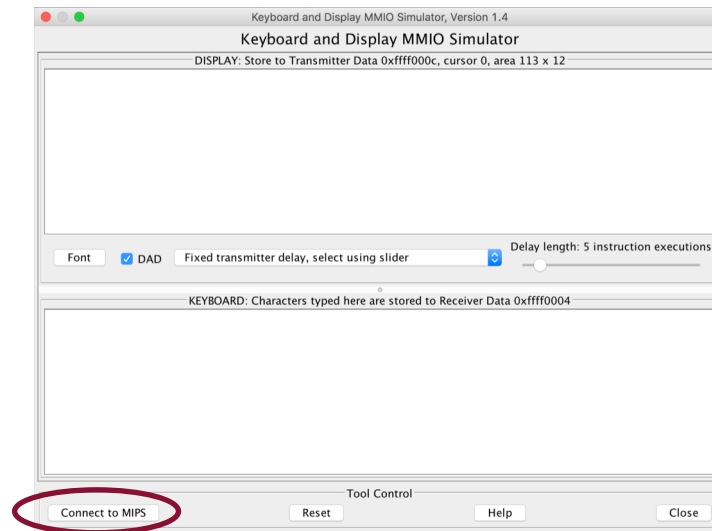
Outline

- Keyboard inputs
- Bitmap outputs
- File handling
- Sleep



Keyboard Inputs

- Menu > Tools > Keyboard and Display MMIO Simulator



- 0xffff0000: set to 1 when input available
- 0xffff0004: contains latest character in ASCII format
 - After loading of content 0xffff0000 resets to 0

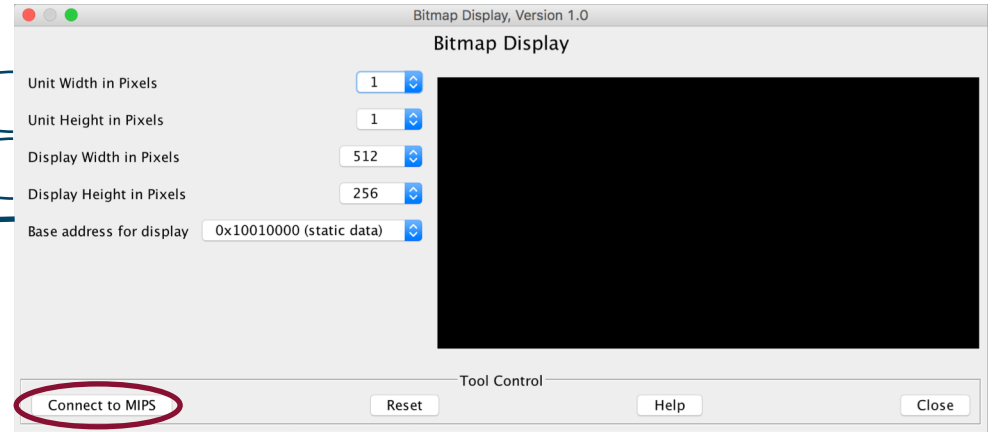
Bitmap outputs

- Menu > Tools > Bitmap Display

Set display-width for one pixel: set to 16 x 16

Set size of display in pixels: set to 512 x 512

Set base address to `$gp`



- Every memory location defines color for one pixel (aRGB in hexadecimal format)
- Row-major

Bitmap - Voorbeeld

Memory	+0	+4	+8	+c
0x10008000	0x00ff0000	0x00ff0000	0x0000ff00	0x0000ff00
0x10008010	0x000000ff	0x000000ff	0x00ffffff	0x00ffffff
0x10008020	0x00ff0000	0x00ff0000	0x0000ff00	0x0000ff00
0x10008030	0x000000ff	0x000000ff	0x00ffffff	0x00ffffff

Bitmap:



File Handling

- Read File Descriptor
 - Syscall with op-Code 13
 - `$a0`: address of null-terminated string (`.asciiz`) containing file location
 - `$a1`: flags (set to 0)
 - `$a2`: mode (set to 0)
- Load entire file into a buffer
 - Syscall with op-Code 14
 - `$a0`: file descriptor
 - `$a1`: address of the buffer
 - `$a2`: max number of characters to read



Sleep

- Used to create 'visual' time steps in our game
- Syscall with op-Code 32
- `$a0`: time to sleep in milliseconds



Example – File input

```
.data
fin:      .asciiz    "input.txt"          # filename for input
buffer:   .space    2048                 # space used as buffer

.text
#####
# Open (for reading) a file
li $v0, 13          # system call for open file
la $a0, fin         # output file name
li $a1, 0           # Open for writing (flags are 0: read, 1: write)
li $a2, 0           # mode is ignored
syscall            # open a file (file descriptor returned in $v0)

move $s6, $v0      # save the file descriptor

#####
# Read from file to buffer
li $v0, 14          # system call for read from file
move $a0, $s6       # file descriptor
la $a1, buffer      # address of buffer to which to load the contents
li $a2, 2048        # hardcoded max number of characters (equal to size of buffer)
syscall            # write to file, $v0 contains number of characters read

#####
# Close the file
li $v0, 16          # system call for close file
move $a0, $s6       # file descriptor to close
syscall            # close file
#####
```

