NAME

mgsimdev-gfx − Graphical framebuffer pseudo-device in MGSim

DESCRIPTION

The graphical display is a 2D pixel−oriented output device. It provides a linear framebuffer, without output to screen at a fixed set of SVGA resolutions.

An I/O device of this type can be specified in MGSim using the device type **GFX**.

CONFIGURATION

GfxFrameSize

Size of the framebuffer memory in bytes.

GfxEnableSDLOutput

Enable rendering the framebuffer to screen outside of the simulation. When this is not enabled, software running on the simulated platform can update the framebuffer but no pixels are displayed on screen.

SDLHorizScale, SDLVertScale

How many pixels on the real screen to use to display each pixel in the framebuffer. These parameters can be adjusted at run−time, see *On the host side of the simulation* below.

SDLRefreshDelay

Defines how many simulation cycles to wait between updates to the output display, when enabled. Can be adjusted at run−time.

PROTOCOL

Changing the display mode

When read from, words 1−3 indicate the current display mode.

The following process configures a new mode:

1. the desired mode is written into words 1−3 of the control device;
2. the command word 0 is written to, to commit the desired mode;
3. the words 1−3 are read back to check whether the mode was accepted.

Invalid mode configurations are ignored and leave the previous mode unchanged.

The following sections detail what configurations are accepted.

Pixel modes

The desired pixel mode is written at word offset 3 of the control device (see *INTERFACE* below), then committed by writing to word 0. The following modes are recognized:

Bits 0−15	Bits 16−31	Value	Resulting pixel mode
8	0	8	RGB 2−3−3
8	1	65544	8−bit indexed
16	0	16	RGB 5−6−5
24	0	24	RGB 8−8−8
32	0	32	RGB 8−8−8, upper 8 bits ignored

When in RGB mode, the color components of the output pixels are defined by the bits in the framebuffer. For example in pixel mode 24, 3 adjacent bytes in the framebuffer define one pixel on screen, with the first byte for red, 2nd byte for green, 3rd byte for blue. With pixel mode 8, one byte of the framebuffer is decomposed as 3 values, one value of 2 bits for red (bits 6−7), one value of 3 bits for green (bits 4−6), and one value of 3 bits for blue (bits 1−3).

When in indexed mode, the value in the framebuffer is used as an index in a palette which is defined separately from the framebuffer (in the control device). The palette then defines which R/G/B values to use.

The current pixel mode can be read from word 3 of the control device.

Color palette for indexed modes

The palette is defined at words 256 onwards in the control device. Word 256 corresponds to palette index 0, word 257 to palette index 1, and so on.

The palette can be both read from and written to, even without setting a new mode.

Output screen resolution

The desired output screen width and height in pixels are set at word offsets 1 and 2 of the control device, respectively (see *INTERFACE* below), then committed by writing to word 0.

The values in words 6 and 7 indicate the maximum supported resolution.

The desired resolution is rounded up to the nearest valid resolution, which must be one of the following: 10x10, 100x100, 160x100, 160x120, 320x200, 320x240, 640x400, 640x480, 800x600, 1024x768, 1280x1024.

The effect of setting an output resolution higher than the capacity of the framebuffer is undefined.

Changing mode vs clearing screen

As described above, writing to word 0 of the control device sets a new display mode.

If the value 0 is written to word 0, the new mode is set but the framebuffer is preserved.

If the value 1 is written, the new mode is set *and* the framebuffer is cleared.

Screen dump

When the current pixel mode is 32 (RGB 8−8−8), writing to control word 4 outputs the framebuffer content to an portable pixmap (PPM) image in the simulation's host environment.

The value written to control word 4 further configures the screen dump, as follows:

- bits 0−1 determine where the PPM data is output to. The value 0 causes the data to be output to a file. The value 1 causes the data to be printed to MGSim's standard output stream. The value 2 outputs to MGSim's standard error stream.
- bit 8 determines whether to embed a timestamp in the file name when bits 0−1 are set to 0.

INTERFACE

The pseudo−device presents itself to the I/O bus as two logical devices: the *control* interface and the *framebuffer* interface.

When multiple graphical outputs are connected to a bus, the device identifier of the framebuffers can be matched to their control devices via word 9 of the control device. (see below).

Control device

The gfx control device must be accessed using 32−bit I/O operations. Its device address space is as follows:

32−bit word	Mode	Description
0	R	Boolean: indicates whether the physical screen is connected

0	W	Command: commit the mode configured using words 1−3, non−zero clears screen
1	R	Current width in pixels
1	W	Desired width in pixels
2	R	Current height in pixels
2	W	Desired height in pixels
3	R	Current pixel mode (see below)
3	W	Desired pixel mode (see below)
4	W	Command: dump the framebuffer contents
5	R/W	Image index (key) for the next dump
6	R	Maximum supported width
7	R	Maximum supported height
8	R	Screen refresh interval in bus clock cycles
9	R	Device identifier of the corresponding framebuffer device on the I/O bus
256−511	R/W	Color palette (one 32−bit word per color index)

Framebuffer device

The framebuffer device can be accessed using any I/O data width, as long as no address past the framebuffer size is accessed.

The data in the framebuffer is organized as per the *Pixel modes* explained above, using row−major addressing (horizontally adjacent pixels have consecutive addresses in the device address space).

On the host side of the simulation

When the screen output is enabled, the following keystrokes are recognized:

Escape Closes the display.

Page down / Page up
Modify the scaling factor quickly (how many output pixels are used to display each logical pixel)

Home / End
Modify the scaling factor slowly.

Tab Restore the aspect ratio (set the horizontal scaling factor equal to the vertical factor).

Up / Down
Increase / decrease the refresh delay (refresh rate).

R Reset the delay and scaling factor to the base configuration.

Moreover, the display window can be interactively resized using the regular window size manipulation method (eg mouse) to adjust the scaling factor at a finer grain.

SEE ALSO

mgsim(1), mgsimdoc(7)

BUGS

Report bugs & suggest improvements to *microgrids@svp−home.org*.

AUTHOR

MGSim was created by Mike Lankamp. MGSim is now under stewardship of the Microgrid project. This manual page was written by Raphael 'kena' Poss.

COPYRIGHT