fsxecute

This command is specific to...
PC + MRC500 framestore
PC + Synoptics Synergy framestore

keys:	**cmd**	*<number>*	specify a command number to execute
	arg, ar2....ar8	*<number>*	specify command argument(s)
	wait	*<number>*	specify number of seconds to wait for command to finish
	put	*<number>*	write the specified picture to framestore data-buffer
	get	*<number>*	read back the specified picture from framestore data-buffer
	gnumber	*<number>*	specify picture width to ***get***
	gfrom	*<number>*	start address of information in 2100 program memory to ***get***
	name	*'<text>'*	specify a filename, containing a command file for the ***load*** option
	port	*<number>*	specify an i/o port base address
options:	**reset**		reset the board
	halt		halt the board
	run		run the board
	load		download the framestore program memory from disc

The **fsxecute** command provides direct control of the MRC500 or *Synoptics* Synergy framestore from Semper itself. Note that this command can have far-reaching effects and is intended only for the experienced user.

Examples

```
fsxecute reset
```

This command resets the board.

```
fsxecute put 3 cmd 9
```

This command loads data from picture 3 into the overlay look-up table.

fsxecute

```
fsxecute 1 0,0,512,1,6 get 7 gnumber 512
```

This command reads the first 512 pixels of row 0 into picture 7.

```
fsxecute load name 'mine' run
```

This command loads in a command file called *mine* and sets the board into the *run* state.

```
rep: trap=40 fsxecute; type frc; jump rep
```

This command displays the status on an interrupt (for example, during slow scan), held in the variable *frc* (framestore return code).

Description

The syntax and functions of the **fsxecute** command can be divided into the following three areas:

- framestore command execution (**cmd**, **arg**, **ar2...ar8**, **wait**)
- reading and writing picture data (**put**, **get**, **gnumber**, **gfrom**)
- hardware control/ program loading (**reset**, **halt**, **run**, **load**, **name**, **port**)

Executing framestore commands

You can execute a framestore command by setting the key **cmd** to the number of the required command and the keys **arg**, **ar2**, **ar3...ar8** to the value of any required parameters. Refer to *Chapter 2, Command Set* of the following manual:

Synergy Image Processing Subsystem Software Manual

for a detailed description of each command and its corresponding number and arguments.

For example, to turn on a single pixel in overlay 0, type the command:

```
fsxecute cmd 15 args x,y,0,0
```

This executes the required command, waits for it to finish, and sets the value of the variable *frc* (framestore return code) to the value of the returned status. Because this is the most common use of the **fsxecute** command, the default keywords **$1**, **$2**, **$22...$28** are used so that you can type:

```
fsxecute 15 x,y,0,0
```

which has the same effect as the previous command example.

fsxecute

You can also use the **wait** key with the following effect:

wait=0	wait until a command finishes or a user interrupt (*frc* = status)
wait<0	do not wait (*frc*=–1)
wait>0	wait for specified number of seconds, then return *frc*=status if the command is finished or *frc*=–1 if it is not

The variable *frc* holds the *framestore return code*. The default value for **wait** is 0. Note that if a command returns a non-zero status, **fsxecute** assumes that this indicates an error and aborts using error code 40. You may need to avoid this for a few commands by trapping the error using **trap=40**.

Reading and writing picture data

You can write data to the framestore data-buffer by setting the key **put** to a picture number. If the picture's storage class is *fp* or *complex*, the real part is converted to 2100 fixed-point format. The rows are sent in order from the top row, and layers from the back layer. The picture class is ignored, and Semper only faults the picture if too large (that is, if its total pixels exceed 768).

You can read data back from the framestore by setting the key **get** to a picture number. By default, Semper reads back the whole data-buffer into an *integer* form *Image* class picture of size 768*1*1. You can read smaller or larger amounts of data by setting the key **gnumber** to the desired picture width. You can read data from other places in 2100 program memory (for example, the status buffer) by setting **gfrom** to the start address. You can change the class of the new picture (*integer* by default) using the general options **byte**, **fp** and **complex**. (For further detail of general keys and options, refer to *Appendix C: Semper Keys and Options*). For *Fp* and *Complex* class pictures the data is treated as 2100 fixed-point, and converted from the raw integers accordingly. The resulting picture is always 1-D, and has its origin set at the left.

The 2100 fixed-point format is described in the following manual:

Synergy Image Processing Subsystem Software Manual

and is a fractional representation where 0x8000= –1, 0x7FFF= (almost) +1.

Hardware control and program loading

You can **reset**, **halt** and **run** the board. If you specify the **load** option the framestore program memory is downloaded from disc. By default, the program memory is loaded from the file *semper.cmd*, unless you use the **name** key to specify a different filename (you do not need to specify the extension *.cmd* as Semper does this for you).

You can set the key **port** to specify an i/o port base address. This overwrites the previous value for the system and, with care, you can use this to switch Semper output between framestores.

fsxecute

Further Information

Semper performs the available keys and options in the following order:

port
halt
reset
load
run
put
(perform **cmd**)
wait
(set *frc*)
get

The **run** option is implicit if any command is to be executed, and **reset** is implicit if you specify the **load** option.

Notes

see also: **fscontrol**
variables set: *frc* (framestore return code)

Defaults and Ranges

keys/options	defaults	range
cmd	command –1	integer in range 0 to 41
arg, ar2...ar8	arg=0, ar2=0...ar8=0	valid command argument
wait	0	real number
put	*none*	valid picture number
get	*none*	valid picture number
gnumber	768	positive integer
gfrom	*none*	positive integer
name	*semper.cmd*	valid filename
port	*none*	positive integer