

# **SpixTools**

## **User's Manual**

v1.0.1

Copyright 2006 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303, U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copy-



Commands

sdas ( 1sh )

NAME  
sdas

## Commands

**NAME**  
spixstats -



**NAME**

spix\_sparc\_dis, spix\_sparc\_dis32 – Disassemble a SPARC instruction

**SYNOPSIS**

```
#include <spix_sparc.h>
```

```
size_t spix_sparc_dis(char *buf, size_t szbuf, spix_sparc_iop_t iop, const void *pinst,  
                    spix_addr64_t addr);
```

```
size_t spix_sparc_dis32(char *buf, size_t szbuf, spix_sparc_iop_t iop, const void *pinst,  
                      spix_addr32_t addr);
```

**DESCRIPTION**

These functions disassemble a single SPARC instruction into the given buffer. The buffer *buf* must have size *szbuf*. The instruction must be represented by the opcode value *iop*, which is the value returned by a previous call to **spix\_sparc\_iop(3sh)**. The pointer *pinst* must point to the beginning of the instruction, and *addr* must be the instruction's virtual address.

Both functions write the disassembly to *buf*. If there is not enough room in *buf*, only the first *szbuf*



**NAME**

spix\_sparc\_iop – Return a SPARC instruction's opcode value

**SYNOPSIS**

```
#include <spix_sparc.h>
```

```
spix_sparc_iop_t spix_sparc_iop(spix_sparc_ver_t ver, const void *pinst);
```

**DESCRIPTION**

The **spix\_sparc\_iop()** function calculates the an "opcode" value for a SPARC instruction. This opcode

Spix Library

spix\_sparc\_iop\_istype(3sh)

**NAME**

spix\_sparc\_iop\_istype

This type selects all instructions that are defined for the SPARC V9 architecture. (It does not include any of the UltraSPARC extended instructions.)

**SPIX\_SPARC\_ITYPE\_VIS**

This type selects all the UltraSPARC extended "vis" instructions.

**SPIX\_SPARC\_ITYPE\_PRIV**

This type selects all of the privileged instructions. It does not select instructions that are privileged only for certain operands (such as RDASR) or that are only privileged depending on the processor's configuration (such as RDTICK).

**SPIX\_SPARC\_ITYPE\_BAA**

This type selects all branch instructions that are contingent on the value of a register.

**SPIX\_SPARC\_ITYPE\_FPOP1**

This type selects all FPOP1 instructions as defined by the SPARC Architecture Manual.

**SPIX\_SPARC\_ITYPE\_ALU**

This type selects all instructions that perform an integer arithmetic or logical operation. This does not include load, store, branch, or floating point instructions.

**SPIX\_SPARC\_ITYPE\_ILOAD**

This type selects all instructions that load a value from memory into an integer register. This includes all atomic load/store instructions.

**SPIX\_SPARC\_ITYPE\_ISTORE**

This type selects all instructions that conditionally or unconditionally store a value from an integer register into memory. This includes all atomic load/store instructions.

Note, the Shade instruction class **SHADE\_ICLASS\_IWSTART** does not have a corresponding instruc-

**SEE ALSO**

spix\_sparc\_iop(3sh), shade\_iset(3sh).

Spix Library

spix\_sparc\_iop\_memsize ( 3sh )

NAME	iop	sparc_iop_3ση )
------	-----	-----------------

**NAME**

spix\_sparc\_iop\_name, spix\_sparc\_iop\_Lname – Return the name for a SPARC instruction opcode

**SYNOPSIS**

```
#include <spix_sparc.h>
```

```
const char *spix_sparc_iop_name(spix_sparc_iop_t iop);
```

```
extern const size_t spix_sparc_iop_Lname;
```

**DESCRIPTION**

The `spix_sparc_iop_name()`

**NAME**

spix\_sparc\_iop\_regpos – Return register usage au4dCion in SPARC instrucCion )







**SPIX\_SPARC\_RUPOS\_RD**

Register is in the RD position.

**SPIX\_SPARC\_RUPOS\_IMP**

The instruction does not explicitly reference the register, but its use is implied.

The third parameter to the *pfun* function is the register number. The function must interpret this register as either an integer, floating point, or special register number according to the value of the first parameter. If this is special register, the third parameter has one of the following values.

**SPIX\_SPARC\_SREG\_Y**

The %y register.

**SPIX\_SPARC\_SREG\_ASI**

The %asi register.

**SPIX\_SPARC\_SREG\_CCR**

The %ccr register.

**SPIX\_SPARC\_SREG\_FPRS**

The %fprs register.

**SPIX\_SPARC\_SREG\_FCC**

One of the condition code fields of the %fsr register.

**SPIX\_SPARC\_SREG\_RM**

The rounding mode field of the %fsr.

**SPIX\_SPARC\_SREG\_FSR**

Another other field of the %fsr.

**SPIX\_SPARC\_SREG\_TICK**

The %tick register.

**SPIX\_SPARC\_SREG\_GSR**

The UltraSPARC extended %gsr register.

The final parameter to the *pfun* function is the *pdata* value passed to **spix\_sparc\_reguse()**.

**sparc**  
\_

**NAME**

spixcounts – spixcounts file format

**SYNOPSIS**

```
#include <spixcounts.h>
```

**DESCRIPTION**

The **spixcounts** file format is generated by the **spixcounts(1sh)** Shade analyzer, and is consumed by several of the SpixTools. This file format contains three parts: a header, basic block lengths, and basic

**SEE ALSO**

spixcounts(1sh), sadd(1sh), sdas(1sh), spixstats(1sh), sprint(1sh).