SCSIInstr structure

```
#include <SCSI.h>

typedef struct SCSIInstr {         Size      Offset  Description
   unsigned short  scOpcode;}2       0         operation code
   unsigned long   scParam1;       4          2      first parameter
   unsigned long   scParam2;       4          6      second parameter
} SCSIInstr;                       10
```

A **transfer instructions block** tells the **SCSI Manager** what to do with the data bytes transferred during the data phase. A transfer instruction block contains a pseudo-program consisting of a variable number of instructions; it's similar to a subroutine except that the instructions are provided and interpreted by the **SCSI Manager** itself. The instructions are of a fixed size and are of type **SCSIInstr**.

Eight instructions are available; their operation codes are specified with the following predefined constants:

scInc	SCINC instruction
scNoInc	SCNOINC instruction
scAdd	SCADD instruction
scMove	SCMOVE instruction
scLoop	SCLOOP instruction
scNOp	SCNOP instruction
scStop	SCSTOP instruction

A description of the instructions is given below:

scOpcode = scInc scParam1 = buffer scParam2 = count
The SCINC instruction moves count bytes to or from buffer, incrementing buffer by count when done.

scOpcode = scNoInc scParam1 = buffer scParam2 = count
The SCNOINC instruction moves count data bytes to or from buffer, leaving buffer unmodified.

scOpcode = scAdd scParam1 = addrscParam2 = value
The SCADD instruction adds the given value to the address in addr. (The addition is performed as an MC68000 operation.)

scOpcode = scMove scParam1 = addr1 scParam2 = addr2
The SCMOVE instruction moves the value pointed to by addr1 to the location pointed to by addr2. (The move is an MC68000 long operation)

scOpcode = scLoop param1 = relAddr param2 = count
The SCLOOP instruction decrements count by 1. If the result is greater than 0, pseudo-program execution resumes at the current address+relAddr. If the result is 0, pseudo-program execution resumes at the next instruction. RelAddr should be a signed multiple of the instruction size (10 bytes). For example, to loop to the immediately preceding instruction, the relAddr field would contain -10. To loop forward by three instructions, it would contain 30.

scOpcode = scNOp scParam1 = NIL scParam2 = NIL
The SCNOP instruction does nothing.

scOpcode = scStop **scParam1 = NIL** **scParam2 = NIL**
The SCSTOP instruction terminates the pseudo-program execution, returning to the calling **SCSI Manager** routine.

scOpcode = scComp **scParam1 = addr** **scParam2 = count**
The SCCOMP instruciton is used only with a read command. Beginning at addr, it compares incoming data bytes with memory, incoming data bytes with memory, incrementing addr by count when done. If the bytes do not compare equally, an error is returned to the read command.

Example

This example give a transfer instruction block for a transfer of six 512-byte blocks of data from or to address 0x67B50

scOpcode	**scParam1**	**scParam2**
scInc	0x67B50	512
scLoop	-1O	6
scStop		