VSI-S Usage Examples

Revision 1.0
13 February 2003

Table of Contents

1. Introduction

Presented in this document are several typical examples of VSI-S 'conversations' for a typical hypothetical VSI-S compatible system. DTS responses are indented for clarity.

2. Usage Examples

2.1 *Setup and Record*

Setup a DIM to record 8 bit streams at an effective sample rate of 16 Msamples/sec/bit-stream; set the DOT clock; start recording.

```
reset = system;                        reset system
    !reset = 0;                        reset successful
status?                                query system status
    !status? 0 : 0x0;                  OK
1PPS_source = alt1pps;                 specify 1-pps tick from alternate input
    !1PPS_source = 0;                  OK
CLOCK_frq = 32 : 16;                   Specify clock freq as 32 MHz; sample rate 16 MHz
    !CLOCK_frq = 0;                    OK
DOT_set = 2002y182d16h32m30s;          Enable DOT clock set on next ALT1PPS tick
    !DOT_set = 0;                      OK
(about a second later…..)
DOT?;
    !DOT? 0 : 1 : 2002y182d16h32m31.175s; DOT running; current DOT clock reading
BS_mask = 0xff;                        specify bit streams 0-7 as active
    !BS_mask = 0;                      OK
BS_mask?;                              query bit-stream mask
    !BS_mask? 0 : 0xff;                OK
receive = on;                          start recording
    !receive = 0;                      OK
status?;                               get status
    !status? 0 : 0x80;                 recording
(sometime later…end of media)
status?;                               get status
    !status? 0 : 0xa0;                 recording stopped (due to hitting end of media)
receive?;
    !receive? 0 : off;                 automatically set to 'off' at end-of-media
```

2.2 *Setup and playback*

Setup DOM to playback the data recorded in Example 2.1. Reproduce the 8 recorded bit streams to DOM output bit-streams 8-15, respectively, at 8 Msamples/sec/bit-stream with 32 MHz DPSCLOCK; set the ROT clock, start playback.

```
    status?                                   query system status
        !status? 0 : 0x0;                     all OK
    DPSCLOCK_source = dpsclock : 32;          32 MHz DPSCLOCK
        !DPSCLOCK = 0;                        OK
    RCLOCK_frq = 8;                           Reproduced data at 8 Mbps/bit-stream
        !RCLOCK_frq = 0;                      OK
    DPS1PPS_source = dps1pps;                 Set tick source
        !DPS1PPS_source = 0;                  OK
    ROT_set = 2002y182d16h32m35s;             Enable ROT clock set on next DPS1PPS tick
        !ROT_set = 0;                         OK
    (about a second later.....)
    ROT?;
        !ROT_set? 0 : 1 : 2002y182d16h32m36.875s; ROT running; current ROT clock reading
    crossbar = : : : : : : : :0:1:2:3:4:5:6:7;  re-map input bit-streams 0-7 to
                                              output bit-streams 8-15, respectively
        !crossbar = 0;                        OK
    transmit = on;                            start playback
        !transmit = 0;                        OK
    status?;                                  get status
        !status? 0 : 0x100;                   playback pending (i.e. sync'ing)
    (a few seconds later.....)
    status?;                                  get status
        !status? 0 : 0x200;                   playback active
    RCLOCK_frq?;                              get RCLOCK information
        !RCLOCK_frq? 0 : 8 : 8 : 16 : 0xff;   Retrieve current RCLOCK freq, plus original
                                              DIM BSIR (16) and original DIM bit-mask (0xff)
    (sometime later...end of media)
    status?;                                  get status
        !status? 0 : 0x300;                   playback stopped (due to hitting end of media)
    transmit?;
        !transmit? 0 : off;                   automatically set to 'off' at end-of-media
    status?;                                  query status
        !status? 0 : 0x300;                   status sticks until next transmit command
                                              (either 'on' or 'off')
    transmit = off;
        !transmit = 0;                        OK
    status?;
        !status? 0 : 0x0;                     idle
```

2.3 *Media copy*

Copy from a DOM to a DIM using PDATA/QDATA to automatically set the DOT clock in the DIM. Assume various DOM/DIM clocks and clock ratios are already properly set. DOM and DIM commands are shown separately since they may be separate units.

DOM:

QDATA_cntl = 0x2;	Causes QDATA to issue a 'DOT_set' command at every ROT1PPS tick, with the time adjusted forward by one second for proper setting of the DOT clock in the DIM.
!QDATA_cntl = 0;	OK
transmit = on;	Start DOM playback
!transmit = 0;	OK

DIM:

PDATA_cntl = 0x10;	Enable DIM to execute DOT_set commands arriving via PDATA
!PDATA_cntl = 0;	OK
receive = on;	Start DIM record
!receive = 0;	OK