STEREO IMPACT Intra-Instrument Serial Interface
Dave Curtis 2001-Mar-02

1. Scope

This document describes the serial interface used between the IMPACT IDPU and the PLASTIC, SEP, SWEA, and MAG instruments on the STEREO spacecraft. This interface is used to send command and control information from the IDPU to the instruments, and for the instruments to send data and status to the IDPU.

2. Requirements

2.1. *Environment*

The IDPU and SEP are processor-based systems. PLASTIC and SWEA, and MAG are not, containing only logic. SEP communicates mostly in CCSDS packets, while PLASTIC, SWEA, and MAG mostly generate blocks of raw data, and mostly ingest register setting commands.

2.2. *Philosophy*

A common interface design is used to minimize development effort and allow the use of common development tools. The number of wires has been minimized at the cost of more logic at the sending and receiving ends; wires add to weight and complexity, while the interface logic is easily implemented in a small part of an FPGA.

2.3. *EMC*

EMC requirements dictate a clock that is a multiple of 50kHz, slew-rate limiting of the signals, voltage rather than current mode interface, and double-shielding of the harness grounded to case/signal ground at both ends.

2.4. *Bitrate*

A high data transfer rate is required; PLASTIC needs close to 1Mbps to transfer its raw data to the IDPU.

2.5. *Sample Timing*

A method for synchronizing data sampling for the various instruments to a common time base may be required. This time base shall not be synchronized to the spacecraft clock.

2.6. *SEP Packet Time Stamps*

SEP requires knowing the spacecraft clock in order to time-stamp its packets. The spacecraft clock is in UT, with millisecond resolution. Some jitter (many milliseconds) is acceptable on the reconstruction of this clock.

2.7. *Interaction Timing*

Instruments shall be sufficiently self-contained that no high-speed interaction with significant timing requirements shall be required with the IDPU. The instruments shall sequence on their own based on the timing provided and control information transferred asynchronously over the command interface and the sample clock.

3. Overview

A three-wire serial digital interface will be used. A common continuous 1MHz clock signal, "CLK", is provided by the IDPU synchronizes the data transfer and is also the basis of the common sampling clock.

3.1. *Commands*

Commands from the IDPU to the instrument shall be formatted into 24 bit data words (8 bits of identification and 16 bits of information), and passed serially on the command signal "CMD". Start and stop bits shall be used to synchronize transmissions.

3.2. *Telemetry*

Data from the instrument shall be formatted into blocks of 16-bit words transferred serially over the data signal "TLM". Start gaps between messages and words synchronize transmissions.

3.3. *Handshaking*

No handshaking is planned. The IDPU shall be sized to ingest data as fast as the instruments can provide it. The instruments must buffer the commands as needed to keep up with several back-to-back commands, though the average command rate will be low. A failure of the synchronization scheme will result in a lost message that should be noted in the telemetry stream, but typically the message will not be repeated. For SEP, the command rate shall be limited to one command word every 10ms (TBR) to avoid overrunning the SEP command reciever

3.4. *Sample Clock*

Sampling timing is based on the 1MHz CLK signal, plus a sample command sent once a second to synchronize the sampling counters. This command shall be synchronized to provide a jitter-free sample clock. The command will contain a seconds counter to synchronize sampling that takes more than one second.

4. Serial Interface Circuit

Figure 1 shows the interface circuitry for the serial interface.

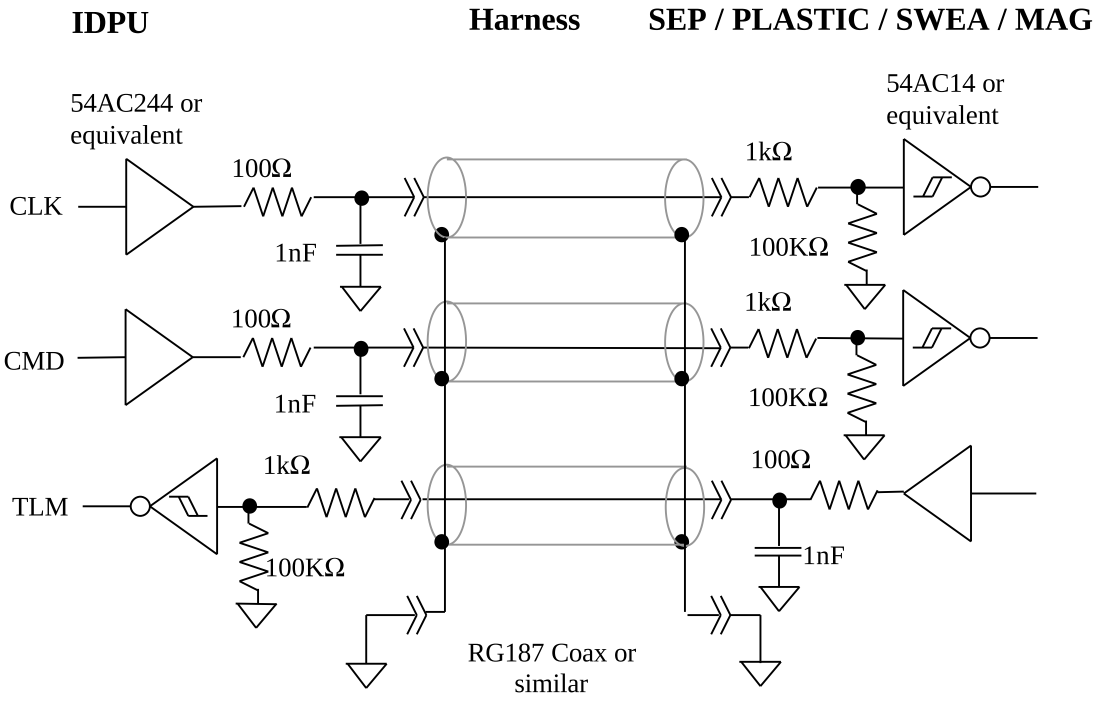

Figure 1 - Serial Interface Circuit

4.1. *Termination*

The RC circuit at the driving end provides series termination for the cable as well as 100ns RC time constant to remove the fast edges from the signals. The resistors at the receiving end protect the gate input and pull it to an inactive level when disconnected.

4.2. *Hysteresis*

The receive gates have hysteresis to provide noise immunity, nominally 54AC14. Note that this is an inverting gate; the signal levels described below are as measured on the harness. It is assumed that the interface logic following the 54AC14 makes up for the inversion.

4.3. *Power-off mode*

The IDPU shall output a low level on CLK and CMD when the instrument is powered off to avoid partially powering the instrument through the logic gate. Since the IDPU shall always be on whenever the instrument is on, no such logic is needed in the instrument end.

4.4. *Shields*

Only the first shield is shown. A common shield over this harness plus any other power or analog signals shall provide the second level of shielding. This shield shall also be grounded at both ends.

5. Command Interface

Figure 2 shows the command interface timing.

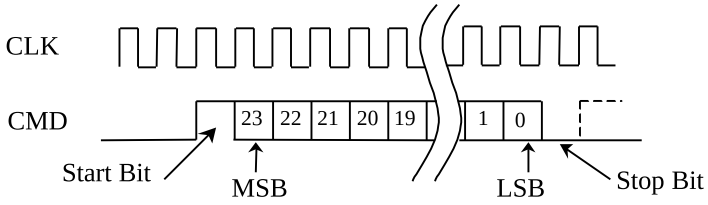

Figure 2 - Serial Command Timing

5.1. *Clocking edge*

The receiving circuit should clock in the data bits on the falling edge of CLK (to avoid a race between the CMD and CLK signals).

5.2. *Synchronization*

The system synchronizes by finding the first non-zero bit (the START bit), and verifies synchronization by the presence of a zero-value STOP bit. After a reset or loss of synchronization, the receiving system should look for 24 consecutive zero-level bits before starting to look for a start bit to avoid incorrect interpretation of a transfer in progress.

5.3. *Data Stream Format*

Commands are 24-bits long, preceded by a start bit, and followed by a stop bit. The 24 bits are sent MSB first. Commands can start on any rising edge of CLK, and any number of idle bit periods can occur between commands. The data is transferred Most Significant Bit (MSB) first. Messages consist of an 8-bit identifier (CMD_ID) in the 8 MSB, followed by a 16-bit data field in the LSB (CMD_DATA).

6. Telemetry Interface

Figure 3 shows the telemetry interface timing.

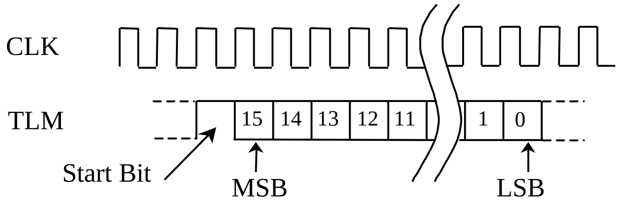

Figure 3 - Serial Telemetry Timing

6.1. *Clock Edge*

The instrument shall shift the next bit of the message out on the rising edge of CLK. The bit will be sampled by the IDPU on the next rising edge of CLK.

6.2. *Synchronization*

Messages are preceded by at least 17 bits of zero. The IDPU shall synchronize to the first non-zero bit as the START bit of the first word of the message. The end of message shall be indicated by a zero where the next start bit should be, followed by at least 16 more zeros. The number of zeros between messages shall be any number greater than or equal to 17. On reset, or in the event of a failure in the synchronization timing, the IDPU shall abort the message and ignore the data until at least 17 zeros in a row have been sent.

6.3. *Data Stream Format*

Messages shall consist of a block of one or more 16-bit words sent consecutively without gap, other than the START bit at the beginning of each word.

7. Command Message Coding

Command messages are somewhat different for the different instruments, as described below.

7.1. *Sample Clock Message*

This message is sent to all the instruments. The sample clock message is a special synchronous message used to synchronize the sample clocks of all the instruments. The CMD_ID of the message shall be hexadecimal code 0xF0, and the CMD_DATA shall be the sample clock. The message shall be sent once a second, every 1,000,000 cycles of CLK. The sample clock shall increment every second. The message is synchronized such that the rising edge of the clock between the last bit of the command (bit 0) and the STOP bit corresponds to the exact second tick.

NOTE: *We need to decide to what extent sampling needs to be synchronized between instruments and the IDPU, and if so, come up with commensurate sampling intervals. The only case that comes to mind is that SWEA and MAG should be synchronously sampled to simplify Pitch Angle computations. There may be advantages in ground processing to synchronize data sampling between instruments, but the disadvantage is that it forces different instruments to have commensurate sampling intervals which may be inconvenient.*

7.2. *Spacecraft Clock Message*

This message is only sent to SEP. SEP requires access to the spacecraft clock for time-stamping its telemetry packets. A message consisting of three commands will be sent once a second containing the TBD UT time code provided by the spacecraft to the IDPU. The time of transmission of this message will correspond to the UT indicated (perhaps with a fixed offset) to within a few milliseconds. SEP is responsible for interpolating this time code between messages using some kind of counter. The CMD_ID of these UT time commands shall be 0xF1, 0xF2, and 0xF3. The 16 MSB of the UT time code sent in the first word (CMD_ID = 0xF1), the middle 16 bits sent next (CMD_ID = 0xF2), and the 16 LSB of the UT time code sent last (CMD_ID = 0xF3).

7.3. *Reset Command*

This message may be sent to all instruments. It should be hardware-decoded by the SEP independent of the processor. It is used to reset the instrument (and SEP processor) back to its default configuration. It has CMD_ID = 0xFF, and CMD_DATA = 0xABCD.

7.4. *SEP Command Packet Message*

SEP will receive multi-word CCSDS command packets. These will be passed on as received by the IDPU from the spacecraft (if the packet ApID indicates the command packet is for SEP). The first word of the message shall have CMD_ID = 0x00. Subsequent words of the message shall have CMD_ID = 0x01. The command packet format has variable length, with the length included in the packet, so SEP should be able to determine when all of the command has been received. SEP should perform its own command verification, as needed, independent of the IDPU.

Other command messages shall be instrument-specific and are TBD. Most I expect will access mode/control registers. Some will load tables, using a sequence of commands, such as writing a memory address to one CMD_ID, followed by a series of data words to a second CMD_ID. Allocation of the rest of the command messages shall be up to the instruments, and will be documented here as they evolve. Other than the common messages described above, the CMD_ID codes for the different instruments are independent, and can be selected at the discretion of the instrumenter.

8. Telemetry Message Formats

Telemetry message formats are instrument specific. For all messages the first word, called the MESSAGE_ID shall indicate the type of message and its length (in words). The 10LSB of the MESSAGE_ID shall be the message length, and the 6MSB shall be a message type code. The message length shall be coded as the number of words in the message, including the MESSAGE_ID word, minus 2. The shortest possible message of two words shall have length code zero, while the longest possible message will consist of the MESSAGE_ID word plus 1024 data words, with a length code of 1023.

The message formats shall be defined by the instrument teams and described here. The types of messages include: SEP Telemetry packets, SEP Housekeeping blocks, PLASTIC raw data blocks of various kinds, PLASTIC memory dumps (which should include an embedded address), PLASTIC housekeeping blocks, SWEA counter readouts, STE counter readouts, SWEA/STE housekeeping blocks. Allocating the telemetry channel amongst the various data sources is the responsibility of the instrument (though the IDPU can help by sending appropriate commands if desired). Allocation of MESSAGE_ID type codes is up the instrumenter, and will be documented here.