Proposed Protocols for Data Transmission and Control Functions for the ATLAS Silicon Micro-strip Detector

The following people have contributed to this proposal or supplied useful review of it, but the presence of their names in this list should not be construed as their approval or endorsement of its entirety:
Francis Anghinolfi, Kurt Borer, Philippe Farthouat, Alex Grillo,
Carl Haber, John Harton, Richard Nickerson, Tony Weidberg

17-November-1994

Overview:

This document proposes a protocol to be used by the ATLAS Silicon Micro-strip Detector for transmitting data from the detector to the DAQ/Trigger and monitor systems located off detector. Another protocol is also proposed for sending trigger and control function commands to the on-detector electronics. This work was started in order to provide guidelines for on-detector ICs which must be designed and prototyped in the near future to meet the expected construction schedule of the silicon micro-strip sub-system. Thought was given to the particular requirements of that sub-system where the on-detector part of the data transmission system must be inside the active volume of the detector requiring low mass, low power and radiation-hard components. However, the protocols at this level of definition are quite general so that they can accommodate all three of the readout architectures still being developed for the micro-strip detector. In a sense they provide a shell to surround data packets of potentially different content and form. Because of this generality, the protocols could be adaptable to other subsystems of the ATLAS Inner Detector.

Several assumptions were made in designing these protocols. They are:

1) The data will be transmitted from and to the detector via serial links, most probably optical. And these links will be highly segmented around the detector.

2) The bit error rate of these links is expected to be very low. Certainly < 10^{-11}, but more likely < 10^{-14}. The links going to the detector are expected to have considerably lower error rates since the transmitters can be operated at higher light outputs from outside the detector where power is not a great concern.

3) Because of this low error rate, no attempt is made in the data transmission protocol to detect or correct errors. Some attempt is made, however, to minimize the confusion of data packet boundaries due to single bit errors. A single bit error should not cause the loss of more than one data packet or the erroneous merging of two.

4) Likewise, with the control protocol, no attempt is made to include error detection or correction but some attempt is made to minimize problems due to single bit errors by choice of bit patterns.

5) One possible implementation of optical links will use LED transmitters. To remain compatible with that technology and maximize LED lifetime, a data format is used to allow idle states which do not drive light or make level transitions.

6) It is assumed that the clocks at both ends of the data transmission line have a common source and are, therefore, synchronized. Synchronization of received data with the local clock only requires establishing a constant phase shift which can be done via an initialization process. The data format does not have to provide a clock recovery scheme and is chosen not to since most such schemes require more transitions per bit transmitted and larger bandwidth.

7) The frequency of issuing Control Function Commands to the on-detector electronics will be low enough and the length of the command sequence short enough that disabling Level 1 Triggers during the transmission of such commands (not the execution of the commands necessarily but only the transmission of the command to the on-detector electronics) will not introduce an unacceptable dead time.

8) The trigger logic will force a gap of at least 2 beam crossing between any two Level 1 Triggers. The combination of (5) and (6) allows the Control Functions to be transmitted on the same link as the Level 1 Triggers.

Data Transmission Protocol

Data from physics events as well as monitoring and status information will be sent from the on-board electronics in serial form using NRZ-L format. This means that a "1" is represented as a high level (light on for optical) and a "0" is represented as a low level (light off for optical) during the bit time interval. No return to baseline (no return to zero) will occur between consecutive bit levels. Each serial link will transmit complete data from some subset of a detector subsystem (i.e. data is not logically multiplexed across fibers). The actual number of channels or readouts serviced by one link will be determined by the specific readout architecture. The protocol is defined in three levels. The first level defines a Data Packet which is in common with all architectures and all types of data. The second is the Data Header which is also in common with all architectures but defines the type of data present in the packet. The third level is the actual Data Format which is specific to the readout architecture. One Data Packet will be sent for each accepted Level 1 Trigger or each request for information (via control command).

Data Packet:

Each Data Packet will start with a Preamble consisting of the following bit pattern (data is shifted out to left with leftmost bit out first):

Preamble 11101 (5 bits total)
←

Each Data Packet will end with a Trailer consisting of the bit pattern:

Trailer 100000000 (9 bits total)
←

Between the Preamble and Trailer any data length field is allowed (including zero).

Transmitting Algorithm for Data Packet:

1. For each data packet sent the transmitter will initiate the transmission with a Preamble. Note that it is assumed that a Trailer has ended a preceding Data Packet and that idle cycles between Data Packets are effectively in "0" state. If an initial Data Packet is sent immediately upon power-up, transmitter must assure that a minimum number of "0" cycles, equal to the length of a Trailer, have appeared on the link before starting the Preamble.

2. As the data is serially transmitted, the transmitter will monitor outgoing data. After any sequence of 7 consecutive "0"'s (one fewer than the number of "0"'s in the Trailer) , the transmitter will insert a "1" into the outgoing data sequence. This of course does not apply when the transmitter is sending the Trailer.

Receiver Algorithm for Data Packet:

1. When Receiver is in idle state (i.e. not currently accepting a Data Packet), it is constantly looking for a match to the Preamble pattern. With each bit read, the Receiver will compare the most recently received 5 bits against each of the following patterns looking for a match (leftmost bit read first):

Uncorrupted Preamble	11101
Single bit flips	11100
	11111
	11001
	10101
	01101

Note: This action is immune to any single bit error within the Preamble or within the 5 bits preceding the Preamble.

When a match is found, the Receiver goes into Receiving State.

2. When Receiver is in Receiving State (i.e. actively processing a Data Packet), it must monitor all incoming data looking for a match to the Trailer pattern. When Trailer pattern is found, Receiver ends receipt of that Data Packet and goes to Idle State.

3. When Receiver is in Receiving State, if seven "0"'s (one less than in the Trailer) are found followed by a "1", the Receiver will remove the "1" from the data sequence.

Data Header:

The first 13 bits of data in a Data Packet after the Preamble will be a header with the following format:

DT (1 bit)	8 bits	4 bits	leftmost read first
0 => L1 data	Level 1 #	BC #	←
1 => Info data	IDPT	IDP #	←

DT (Data Type):

0: Level 1 Trigger Data

1: Information Data

Level 1 #: Current count of Level 1 Trigger modulo 256 since the last system reset. This field can be used for event building by the DAQ and also to monitor for lost data. The length of 256 was chosen to be safely longer than the Level 2 latency or buffer length.

BC #: Current count of Beam Crossing modulo 16 since the last system reset. This field is not really to tag Beam Crossing numbers since it wraps around so quickly. It is intended to monitor for clock pulses lost by the on-detector electronics.

IDPT: The Information Data Packet Type is an 8-bit field to designate what kind of information data is being sent. Type codes are not defined yet, but they could be the 8-bit Sub-command code used in the Control Function Protocol to request the information. (See description of Control Function Protocol in following section.)

IDPT #: Current count of Information Data Packets (IDPs) modulo 16 which have been sent since the last system reset. Note that there is only one rolling count of the number of IDPs on each link, not an individual count for each type of IDP. This is to minimize the number of counters in on-detector electronics. The purpose of this count is to facilitate detection of lost IDPs by the off-detector electronics as the Level 1 # is used to detect lost trigger data.

Level 1 Trigger Data:

If the DT value in the Data Header is "0", the data being sent is in response to a level 1 trigger with the on-detector readout system in normal data taking mode. The format of the rest of the Data Packet until the Trailer is dependent upon the readout architecture.

Information Data:

If the DT value in the Data Header is "1", the data being sent is in response to a request for information via the control system, an error status being reported, or data resulting from the on-detector readout system being in some type of diagnostic mode.

Data Format for Silicon Micro-strip Analog Readout Architecture:

The data format for this readout architecture has not been worked out in detail. Presumably, after the digital information contained in the Data Header, a sequence of analog pulses will follow. The first being a calibration pulse followed by a pulse for each strip readout by the serial link. Any further definition of data format for this architecture awaits input from the groups working on it.

Data Format for Silicon Micro-strip Digital Readout Architecture:

The data format will consist of a sequence of a strip address field followed by one or more pulse height fields. This pattern will be continued for each cluster of hits. A cluster is defined as any group of consecutive strips whose pulse height exceeds the threshold of the sparsification circuit. In the case that Neighbor Logic is active, the nearest neighbor strips of any exceeding threshold will be included in the cluster. The address of the cluster transmitted before the sequence of pulse heights will be the address of the first (or lowest address value) strip of the cluster.

Each address field will start with a "0" in the most significant (first read) bit position. Each pulse height field will start with a "1" in the most significant (first read) bit position. Since the number of bits in an address field and a pulse height field are fixed, the receiving circuitry has a well determined, if not, trivial task to separate the data.

Below is an example of a data sequence of two clusters, the first cluster having 3 strips reported, the second cluster has one:

0	IC/Ch Adrs	1	PH	1	PH	1	PH	0	IC/Ch Adrs	1	PH
	10 bits		7 bits		7 bits		7 bits		10 bits		7 bits

IC/Ch Adrs: 3 bits of IC address followed by 7 bits of channel address each read most significant bit first.

PH: 7 bits of pulse height.

Note: The 10 bit of address assumes the current ATLAS baseline of 128 channels per IC and 8 ICs per double sided module. The assumed design will use 2 serial fibers per module initially with the possibility that one fiber can service the entire module if one fiber dies. Also, the 7 bit pulse height is tied to the Digitization Spec which calls for a 7-bit ADC.

Data Format for Silicon Micro-strip Binary Readout Architecture:

The data format will consist sequentially of data from each readout IC in the module. Data from each IC will start with a Chip Header of three bits. If there are no hits in the IC, that is all the data transmitted from that IC. If the Chip Header indicates at least one hit is present, it will be followed by a 16-bit Map. Each bit in the Map represents the OR of 8 channels in the IC. A "1" in the Map indicates at least one hit in those 8 channels. Following the Map will be a sequence of 8-bit Zoom Patterns one for each "1" in the Map. Each bit in the Zoom Pattern represents one channel. A "1" in that bit position indicates that channel was hit.

Data from an IC with not hit channels:

Chip Header
100

Data from an IC with a hits in channels 36, 97 and 102 (counting channel numbers 0-127 per IC):

Chip Header	Map	Zoom Pattern	Zoom Pattern
111	0000100000001000	00001000	01000010

Control Function Protocol

Level 1 Triggers and Control Function requests will be sent to the on-detector electronics via the same serial link. This reduces the cost, power consumption and material in the detector. Information will be transmitted on this serial link in Unipolar RZ format running at the 40 MHz clock rate of the machine and the on-detector electronics. A "1" will be represented by a high level (light on for optical) and a "0" will be represented by a low level (light off for optical) during the bit time interval which will be half a clock period. Between bit time intervals the signal will return to a low level (light off for optical). A valid Level 1 Trigger will be denoted by three bits "110" (the "0" received last) in time with three consecutive clock pulses. The selected beam crossing of such a trigger will be timed to the last of the three pulses minus a fixed level 1 latency. This can be repeated in succession. For example, a pattern of "110110" would denote two Level 1 Triggers on beam crossings 3 and 6 relative to the time of the first received pulse.

Other Control Functions will be initiated by a sequence of three consecutive pulses reading "101". In this case, these three bits must be followed by another 4-bit field which could define any one of 16 commands. Only 5 possible patterns of these 4 bits are chosen for use since other patterns are less immune to misinterpretation if transmission errors occur during a rapid sequence of Level 1 Triggers. Four of these 5 commands request immediate, fast response by the on-detector electronics (e.g. reset). These commands require no further fields to be transmitted. The last of these commands demands a following sub-command field of 8 bits to designate any one of 256 commands which may require parameters and may take considerable time to execute. Some of these "slow" commands may even put the electronics into a non-data taking mode. The number of fields and their length following each sub-command are determined by the specific command and a fixed (not variable) for each command. We have not defined any of these commands yet (some examples are given below). Some command codes should be in common across readout architectures but many will be specific to one type. Most of these command codes await a definition of the control architecture, but this protocol should be generic enough to service all that is planned.

One of the possible parameters required for a "slow" command might be a chip ID so that the command will only apply to a single front-end chip. For example, if the command is to load a mask register, only one chip is specified and only one mask is shipped as part of the parameter list. A restriction is placed upon any command which requests information to be returned by the on-detector electronics. Any such command must not specify a chip ID but rather must be global to all chips serviced by the link carrying the command. This simplifies the on-detector electronics in that Data Packet are formed in the standard way for all types of data or information.

The following table summarizes some possible Trigger/Control messages (specific Commands and Sub-commands are only examples):

Code (3 bits)	Command (4 bits)	Sub-command (8 bits)	Parameters (depends upon Subcommand)	Description
110	none	none	none	Level 1 Trigger
101	0100	none	none	Global Reset
101	0010	none	none	not yet defined
101	1110	none	none	not yet defined
101	0101	none	none	not yet defined
101	0111	0000 0000	none	Turn off Clock-thru-mode
101	0111	0000 0001	none	Turn on Clock-thru-mode
101	0111	0100 0001	0011, 128-bit mask	Set Mask Register for chip 3
101	0111	1000 0001	none	Read Mask Register for all chips
101	0111	0100 0100	0001 00 1000	Set Threshold DAC on chip 1 to 8
101	0111	1000 0100	none	Read Threshold DAC for all chips