SNO-STR-95-008

Design of the SNO Timing System

R. Van Berg, Josh Klein and J. Wilkerson

February 7, 1995

1 Introduction

As an Observatory, SNO must be prepared to measure phenomena which depend upon natural timescales—from the potentially periodic variations in neutrino flux due to the solar cycle or the rotation and revolution of the Earth, to the aperiodic appearance of neutrino bursts due to Galactic supernovae or other interstellar sources. Each of these measurements requires very different levels of absolute timing accuracy. For variations with the solar cycle, we would be content to know the year or month in which the data was taken, while for a supernova neutrino burst measurement to submillisecond accuracy will allow us to compare the arrival time of the burst at SNO to that measured by other observatories, thereby providing information on the direction to the supernova or a comparison of the emission and travel times for different particle species. Perhaps most important will be the possibility of observing new time-dependent physics which we presently know nothing about.

As a particle physics experiment, the timing demands for SNO are very different. Coincidences between events on short time scales—β-n production, for example—require high resolution and precision for measurement of time intervals. For these measurements, we do not need a clock stable over periods significantly longer than the intervals themselves. Thus requiring such a clock to keep in step with an external time base is neither necessary nor practical.

The goal, therefore, for the SNO timing system is to provide both a highly accurate external time source for observation of time dependent physics, as well as a precise, high resolution clock for measurement of time intervals. In order to simplify analysis, we would like to tag every individual event with a time stamp without extensive off-line calibration. For a supernova burst or some other unusual event, rates in the detector may exceed 1 MHz, and we must ensure that there is no loss of either events or timing information in such extreme cases.

2 Time Scales, Standards and Transfer

A time scale is a reference system which one can use to assign a date or time to an event. There are many different time scales in use, each one appropriate for different needs. The three major classes of time scales are Universal Time, Atomic Time, and Coordinated time.

Universal Time (UT) is any time scale that is tied to the motion of the Earth, and hence is the most useful for astronomical observations. The simplest version of this is time based upon the solar day. Corrections can be added to this scale to account for the eccentricity of the Earth's orbit and the varying rate of the Earth's rotation due to polar motion (UT1) or random fluctuations (UT2).

Atomic Time—one version of which is called TAI for 'International Atomic Time'—is based instead upon a transition between two hyperfinely split ground state levels of the cesium 133 atom. The accuracy of a clock using cesium can be better than 1 in 10^{13}, and TAI is actually an average of the times kept by an array of atomic clocks spread across several countries. However, since the definition of the atomic second is unrelated to the motion of the Earth, Atomic Time drifts with respect to Univeral Time. While useful for global navigation or communication, it cannot provide the information needed by astronomers to calculate ephemerides.

Universal Coordinated Time (UTC) is an attempt to retain the best of both worlds. The definition of the second in UTC is the same as for Atomic Time, but UTC is never allowed to drift more than 900 milliseconds away from UT1. To keep UTC in step with UT1, leap seconds are added (or deleted) whenever necessary (usually either in June or December). Since the rate of the drift between UTC and UT1 is somewhat unpredictable (due to the random changes in the rotation rate of the Earth), there is often only a few months notice that a leap second will be added. For SNO, UTC is the time scale most useful: it allows calculation of ephemeris information through its tie to Universal Time (if more than 1 second accuracy is needed, one can later use the history of the Earth's rotation rate kept by the

International Earth Rotation Service to make corrections), and it is also in such widespread use that knowledge of UTC for an event will allow comparison to the time measured by other observatories.

In order to keep a measure of UTC, one needs some type of time (actually, frequency) standard. For example, setting a quartz watch to agree with UTC at some initial time will be a measure of UTC whose accuracy is determined by the drift of the quartz oscillator. A quartz oscillator is in fact a fairly good frequency standard for short time intervals—an accurate crystal will drift just a few milliseconds over a period of days. A rubidium atomic clock may do an order of magnitude better, and a cesium clock perhaps another two orders of magnitude. For very short time intervals, masers may be the most accurate, and for very long time intervals millisecond pulsars may be the best. However, all of these standards need to be periodically recalibrated to agree with UTC (or some other scale). In addition, atomic standards can be expensive.

A cheaper and better way to maintain accurate time locally is to continuously recalibrate a standard. To do this, time information must be transfered from the calibration source to the local clock. The Global Positioning System (GPS), created for navigational use, consists of an array of satellites, each with two atomic clocks on-board (one rubidium and one cesium). The satellites broadcast the times kept by these clocks as well as ephemeris data describing their orbits and information regarding the health of and current accuracy of the satellite. Observing four of these satellites at any one time allows one to reconstruct three position coordinates and one time coordinate. The GPS time scale is an atomic time scale, but it is synchronized by the United States Naval Observatory to be accurate to UTC, usually within 100 nanoseconds. By monitoring the times broadcast by the GPS satellites, one can adjust the frequency kept by even a mediocre oscillator to produce a local measurement which is in principle good to within 100 ns of UTC (better if more than one receiver is used). However, due to National Security Concerns, the times and ephemeris information that the satellites broadcast are often intentionally spoiled, producing accuracies somewhat worse.

3 System Overview

The SNO timing system will consist of three major pieces: a GPS receiver and time code generator, a time card which will fit into the master acquisition crate along with the master trigger card and the single board computer (event builder), and the data acquisition software. Figure 1 schematically outlines the relation between the different components and the general flow of the timing.

The GPS receiver (which will be purchased from Datum, Inc.) will provide us with 3 different signals, a 10 MHz oscillator, a programmable synchronization pulse, and time codes that contain information about the time kept by the GPS satellites. The 10 MHz oscillator is disciplined to GPS time and expected to be accurate to 300 ns—better if the United States Naval Observatory (USNO) is not intentionally spoiling the signal. The 10 MHz clock will travel over a fiber to be counted by a counter on the timing card, and this count will be latched every time the detector is triggered.

In order to associate the 10 MHz count with an absolute GPS (or UTC) time, the GPS receiver will also provide us with synchronization (SYNC) pulses at pre-programmed times. The programming is done through an IEEE-488 interface from the main acquisition computer. Every hour or so, the acquisition software will request a SYNC pulse at a specified time. The computer will then inform the event builder of the next calibration time, and upon arrival of the SYNC pulse the 10 MHz count will be latched and that count can then be associated with the expected time. The size of the 10 MHz counter will be 52 bits, which is large enough to contain 10 MHz counts for up to 14 years. Every event will have a unique time associated with it, starting at a zero of time agreed upon before the first data tape is written.

In addition to the 10 MHz counter, the time card will also have an internal 50 MHz oscillator for use in measuring the time intervals between events. The 50 MHz clock will be counted by a separate counter and latched each time the detector produces a global trigger.

The SYNC pulse used to calibrate the 10 MHz GPS clock will also initiate a PING which will travel toward the surface along a separate fiber. The PING will be reflected back along the fiber at the surface and upon its return to the time card it will produce a trigger which will latch both the 10 MHz and 50 MHz clocks (the end of the fiber at the time card will feed a split signal—one for the outgoing ping and another for its return to the card) . Comparison then of the 50 MHz count or the 10 MHz count at the time of the SYNC pulse and the time of the return of the PING provides us with a measurement of the travel time along the fiber. The changes in this travel time from event to event as well as the more slowly varying daily and seasonal variations may ultimately limit the accuracy of our time measurement. If we find that the variation over an hour is greater than a few hundred nanoseconds, we would then need more frequent calibrations to maintain a high accuracy. However, if the jitter is small, reconstruction of absolute times off-line may only require a fixed calibration file containing the delay time as a function of time of day or year.

Under startup conditions, for example after temporary downtime due to crashes or maintenance, the acquisition system will read the time codes generated by the GPS receiver (translated into BCD by a separate module) in order to determine the current GPS time. The computer can then decide upon the earliest convenient time to request a SYNC pulse and inform the event builder when to expect the pulse. The event builder can then calculate the 10 MHz count appropriate to the requested SYNC time, and this count will be loaded in to the 10 MHz counter when the SYNC pulse arrives.

4 Time Card Details

As described in Section 3, the time card will contain two counter arrays, one for the 10 MHz clock provided by the GPS receiver, and one for the on-board 50 MHz oscillator. The counter arrays will be built from fast synchronous 8-bit counters, 7 for the 52 bits of 10 MHz counts and 6 for the 44 bits of 50 MHz. The 10 MHz counter array will also allow a parallel load of a preset count, in order to take care of the startup conditions described in Section 3. For the 50 MHz counter, we do not need an absolute offset so simply clearing that counter array on start-up will suffice.

A global trigger (which may arise from a real detector trigger or a calibration trigger like the SYNC pulse) will transfer the currently held counts in each counter array into a set of latches. In addition to the 96 bits of counter information, there will also be ~ 22 bits of trigger id, as well as flags corresponding to any ambient error conditions present on the board (see below), for a total of 128 bits for every event, all of which will be transferred to latches with every global trigger.

The fact that every event will produce 128 bits of information means that for the most extreme cases memory demands on the board will be very high. With a memory design like that on the Front End Card, the maximum rate at which we could write to memory is roughly 8 million longwords/second, or 2 million events/second. In order to increase the maximum bandwidth, each of the 4 longwords will be written in parallel out to the DRAM allowing us to run at up to 8 million events/second. In order to ensure that this rate can be maintained in the face of memory controller refresh cycles or VME reads, we will also add a set of FIFO buffers directly before the DRAM. The amount of DRAM we place on the board will be determined by the largest number of events we wish to store—16 Mbytes buys us 1 million events.

The potential for error conditions on the board will also be monitored by several flags corresponding to full FIFO's or the myriad states in which different FIFO's can get out of synchronization with one another (for example, some full and others not, etc.). These flags will be part of the 128 bit data word for each event.

In some rare cases, a true trigger may occur very close to an expected SYNC pulse. For this reason, a trigger hold-off will be present on the board which will prevent the calibration from latching any of the counters during an ongoing global trigger cycle. Although we will lose the calibration data for such a case, we do not expect every SYNC pulse to be crucial, and the frequency for such pile-up is in any case low.

The time card will also have a feature to allow test pulses to be sent to the Front End Cards to sample pedestals or test triggers. The timing of the test pulses will be programmed through the VME interface at the back of the card.

5 Acquisition Software

The two main aspects of the acquisition software will be the programming and monitoring of the SYNC pulses, and the handling of start-up conditions. As described in the overview of Section 3, the normal operations of the acquisition software will be to program the GPS receiver to provide a SYNC pulse every hour, and to inform the event builder of the time of the next SYNC pulse. The SYNC pulse is a special trigger: in addition to writing the information contained in the counters, the event builder wll also write out the expected time (in 100 ns ticks since time zero) for off-line comparison with the actual count. The event builder will also need to peform an on-line comparison of the expected count and the latched count. If they are found to be different, an error flag and alarm must be set indicating a mismatch between the expected time and the latched time.

Under startup conditions, the situation becomes somewhat more complicated. The acquisition computer must first determine (roughly) the time of year by reading the time codes from the GPS receiver. The time codes will be placed onto a VME bus by a separate time code translator (yet to be purchased). Once the time has been determined, the computer must select the next convenient time to request a SYNC pulse to begin the timing.

So, for example, if the computer finds that the time is 11:07:00 UTC, it then programs the receiver to provide a SYNC pulse at say 11:08, and then it informs the event builder that a SYNC trigger will occur at 11:08 UTC. The event builder then calculates the number of 100 ns ticks corresponding to 11:08 on that day and year, and puts that count on the lines that feed the parallel load into the 10 MHz counters. Once the SYNC pulse arrives a minute later, the correct count will be loaded into the counters which then will begin counting at 10 MHz. The acquisition computer can then request another SYNC pulse to check that the counters did get loaded with the correct count and are counting. For the 50 MHz counter array, which we do not intend to use as a measurement of absolute time, the counters can be cleared startup.

There is a small subtlety here. During the course of the experiment, leap seconds may be added several times to keep UTC in step with TAI (International Atomic Time). Therefore after a down time occuring several years into the experiment, the event builder must know about these leap seconds in order to calculate the correct time in 100 ns ticks. In addition, any missed counts or added counts that may occur during the experiment will also need to be kept track of so that the correct count can be loaded into the counters. There are two options for dealing with this problem—one that requires some level of maintenance and one that does not. The first would be to have a set of external calibration files read in by the acquisition system upon startup. These files would contain the dates and times of every leap second or missed (or extra) count that occurred in the system. The file would be downloaded to the event builder on startup and it could then correctly calculate the count to be loaded into the 10 MHz array. This file would have to be updated by hand each time such an event occurred.

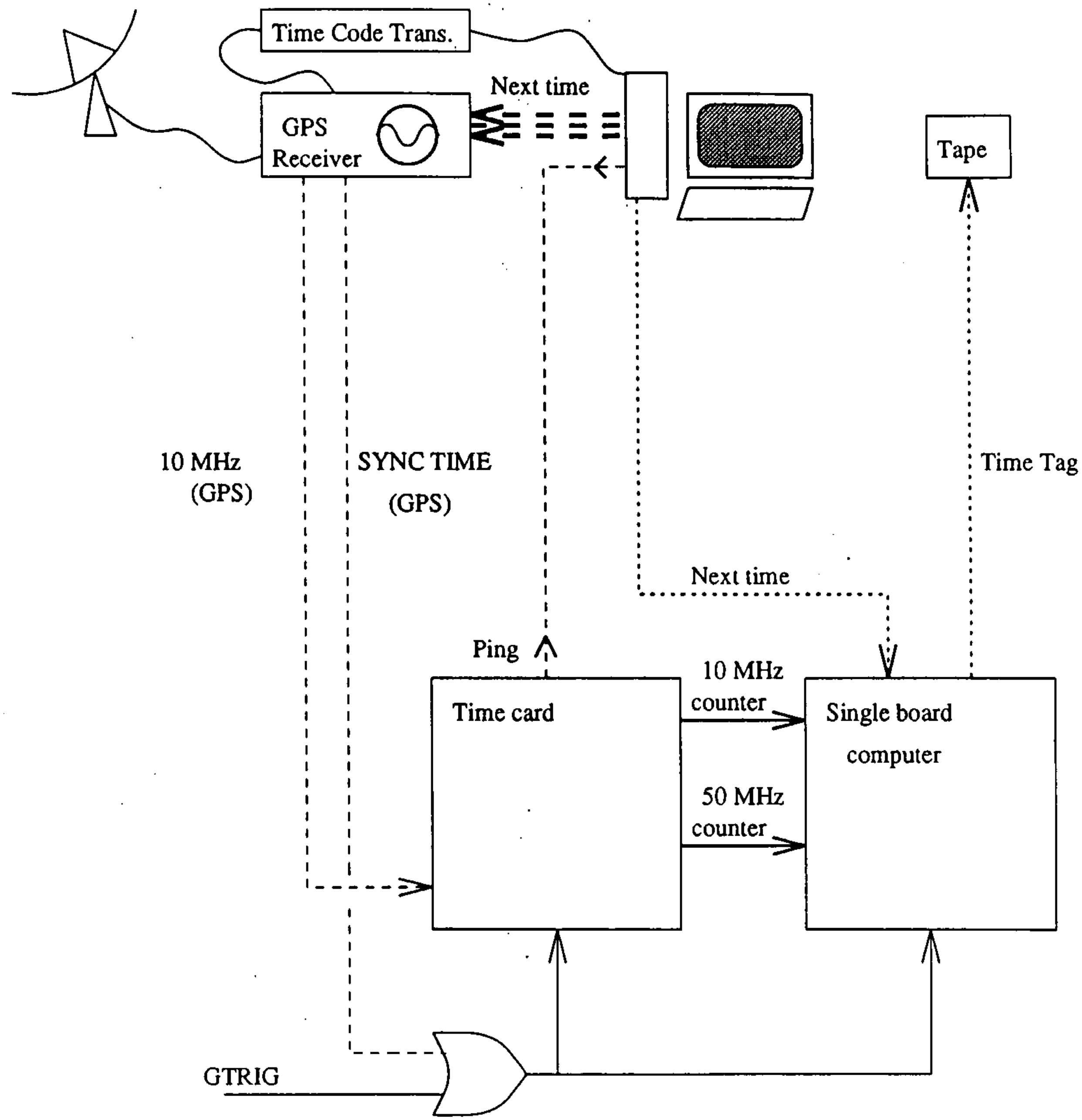
While we will certainly have to maintain such a file off-line anyway, ensuring that it was constantly correct at the detector might be difficult. A second option would be to keep the time and count of the last SYNC pulse stored (on disk perhaps, or a hardware buffer with a battery or even some of the battery powered memory in the main acquisition computer). Then after each downtime, the event builder could use this last time and count to calculate the new time and count, assuming no leap seconds have occured between that time and the start-up time (an unlikely scenario).

6 Conclusions

The timing system outlined above has several advantages. The first is its accuracy and precision: our absolute time may be more accurate than about 300 ns, depending upon the jitter in the transit time from surface to detector, and the precision in our relative time measurement from the 50 MHz clock will be much less than a few ns rms for a 20 ns resolution.

Second, the system requires little maintenance either on-line or off-line. On-line it will be completely automated and require no interference (except in the case of hardware failures). Off-line, we will have a data set in which every event has an absolute time stamp, the corrections to which will only be the varying (but measurable) transit time along the fibers from the surface.

The system also has some self-calibration built in. Since the 50 MHz oscillator will be very accurate on short time scales (hours to days), the SYNC pulse events will allow us to measure the stability of the GPS 10 MHz clock on those short scales. At the same time, the GPS clock will provide us with history of the 50 MHz clock over long time intervals. In the case of temporary failure of the GPS system, this history will allow us to better predict the 50 MHz clock behavior so it can function as a viable temporary back-up. If the failure mode is a severing of an optical fiber, we may also be able to keep track of UTC using the time codes generated by the GPS receiver. Time stamps added to the data can be used later to correlate the 50 MHz clock with UTC.



Time Code Trans.
Next time
GPS
Receiver
Tape
10 MHz
(GPS)
SYNC TIME
(GPS)
Time Tag
Next time
Ping
10 MHz
counter
Time card
Single board
computer
50 MHz
counter
GTRIG