GA–A25821

A SECOND GENERATION TIMING SYSTEM FOR DIII-D TIMING CONTROL

by
T.M. DETERLY, D.H. KELLMAN, and D.F. FINKENTHAL

JUNE 2007

GENERAL ATOMICS

DISCLAIMER

GA–A25821

A SECOND GENERATION TIMING SYSTEM FOR DIII-D TIMING CONTROL

by
T.M. DETERLY, D.H. KELLMAN, and D.F. FINKENTHAL*

This is a preprint of a paper presented at the 22nd IEEE/Symposium on Fusion Energy, Albuquerque, New Mexico on June 17-21, 2007 and to be published in the *Proceedings.*

*Palomar Community College, San Marcos, California.

Work supported by
the U.S. Department of Energy
under DE-FC02-04ER54698

GENERAL ATOMICS PROJECT 30200
JUNE 2007

GENERAL ATOMICS

A Second Generation Timing System for DIII-D Timing Control

T.M. Deterly[a], D.H. Kellman[a], and D.F. Finkenthal[b]
[a]*General Atomics, P.O. Box 85608, San Diego, California, USA*
[b]*Palomar Community College, San Marcos, California, USA*

***Abstract*—The DIII-D Tokamak relies on a facility wide timing network to synchronize machine operations. The first generation system was designed around cascaded CAMAC delay units feeding a custom timing network encoder. This system has become increasingly difficult to maintain and repair and the needs of DIII-D experiments are beginning to exceed its capabilities. To address these issues, a new second-generation system was designed with a modular architecture in a VME form factor that facilitates the future addition of features and output channels when required, while maintaining backwards compatibility with the original system. As part of the base design, modules for event triggers, multiple programmable sequences, first generation Bi-Phase serial outputs, fiber optic outputs, and event recording are provided. Each module is implemented with a form of programmable logic, either a CPLD or FPGA, which allows for future modification if needed. The system also has the capability of complete remote management, allowing for custom timing chains on a per-experiment basis. The feature set and design of this second-generation timing system is presented.**

Keywords: timing; network; data acquisition; clock

I. OVERVIEW

The DIII-D Tokamak, similar to other large experiments, requires facility wide timing synchronization [1]. This has been and is currently accomplished at DIII-D by means of a single site-wide timing network driven by a custom timing generator (TG). This network is available to all devices requiring timing synchronization. The information on the network provides the necessary information required by timing receiver modules to keep all devices in sync with machine operations.

A timing generator is responsible for driving the network. It is what generates the master clock signal along with events signals. The new generation TG is controlled by the central DIII-D control console. The console is capable of remote management of the TG providing capabilities including loading new timing chains and starting and stopping sequences. The complete timing system topology is shown in Fig. 1.

II. TIMING SYSTEM

Since the start of experiments at DIII-D there as been a timing system in place to keep the experiment synchronized. This new timing system is an upgrade from a previous generation system, and adds a host of new features and capabilities, while maintaining backwards compatibility. A brief description of the major timing system elements and their differences between the latest generations follows.

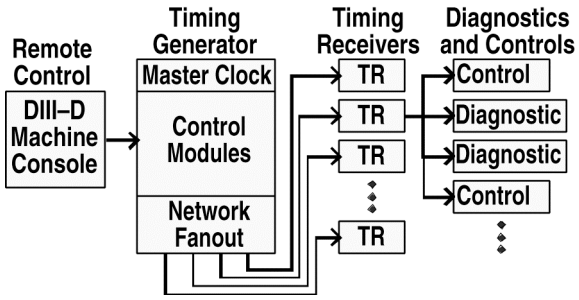

Figure 1. Timing system topology.

A. Timing Network

The timing system for both generations is based on a serial point-to-point network. The network is based on a Bi-Phase serial data encoded stream. This encoding method allows transmission of both data and clock signals on a single tri-axial cable. The data is encoded using a two-times clock, which is set at 2 MHz, providing a 1 MHz system clock. The encoding method is demonstrated in Fig. 2. In this figure the two-times clock is used to encode the data stream producing the signals A and B that, along with a return, form the network. This encoding allows for simple extraction of the clock from the data at the receivers, and also provides noise immunity due to its quasi-differential signaling structure.

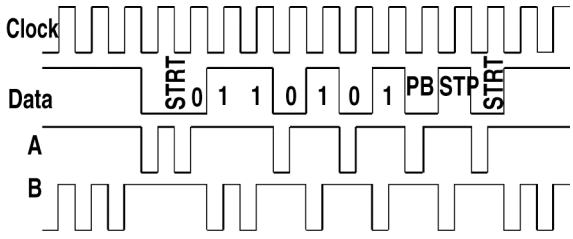

Figure 2. Bi-phase serial encoding. (2MHz Clock, Data: 1 Start Bit, 7 data bits (0x35), 1 Parity Bit, 1 Stop Bit).

The data sent on the network consists of single byte wide hexadecimal codes that are initiated by timing events. These events can either be predetermined (programmable triggers) or externally generated from real world events (event triggers). The previous system was limited to 32 different codes, being hexadecimal 0x60 through 0x7F (the prefix 0x denotes a hexadecimal number), due to hardware limitations. The new system is now capable of utilizing the full seven bits worth of

data (7 bits plus one parity bit) for a total of 128 possible triggers.

B. Previous Generation Timing Generator

The previous timing generator was based on CAMAC technology. The main unit consisted of 32 transformer coupled inputs, one for each possible event code. Most of the inputs were driven by a cascade of CAMAC delay generators that could be partially reconfigured by means of the CAMAC bus, but had a predetermined event chain. This cascade implemented the programmable sequence aspect of the timing chain. Besides these, a few inputs were used as asynchronous event triggers.

C. New Generation Timing Generator

As mentioned earlier the new timing generator uses the same serial network as the past generation, due in part to the large investment in network cabling and timing receivers already in place. Other than this, the system has been completely redesigned allowing for a system that is a great deal more flexible.

The new timing generator is based on a custom bus protocol, specially designed for deterministically timed data transfers while maintaining simplicity. While the protocol is custom, the physical bus is based on an off-the-shelf standard 6U VME backplane, which provides a standardized hardware solution.

The design of the timing generator is very modular. Each major function is implemented as a separate 6U VME card. Similar to other timing systems, a set of basic features are required for operation [2]. At the present, there are a number of different modules, including, a bus controller module, an event trigger module, a module for programmable sequences, a Bi-Phase serial output module, and an event-recording module. Of these cards, all of them can be swapped out for another card with different functionality, with the exception of the bus controller module. The bus controller module, as its name implies, is responsible for arbitrating the bus and hence is required for the operation of the generator.

D. Timing Receivers

The timing information generated by the TG after transmission on the network is decoded using timing receivers. The timing receivers are custom devices that accept the signals from the timing network and decode it into a useable from. They provide the system clock, along with variable length pulses that can be keyed to the various timing events. These pulses and the clock can then be used by devices for timing synchronization.

III. TIMING GENERATOR IMPLEMENTATION

As cited previously, the system is based on a 6U VME backplane and its functionality comes from a host of 6U plug-in cards. In addition to this card-based modularity, the system relies heavily on Complex Programmable Logic Devices (CPLDs) for each module's operations. This combination of programmable logic and modularity allow for a system that is vastly expandable and also simplifies system design and troubleshooting. The implementation of the bus and the modules will be discussed in the following subsections. Fig. 3 shows stylized front panels of the major modules.

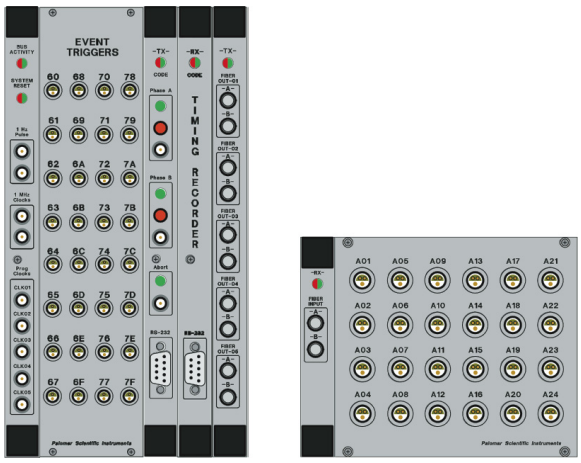

Figure 3. Timing modules (left to right): bus controller, event decoder, programmable sequence generator, timing recorder, bi-phase encoder, fan-out input, and fan-out outputs.

A. Bus Design

The heart of the bus operates with a classical bus model, having one bus controller along with a number of prioritized devices under its control. This implementation, while I/O intensive and somewhat rudimentary, is well adapted for a timing system of this nature. It allows for easy device implementation along with deterministic timing, both of which are required for an expandable timing system in a research environment. The complete bus consists of a number of different components, two parallel 8 bit buses, two serial buses, and a number of different control and clock lines. A diagram of the bus is shown in Fig. 4, and details of its implementation follow.

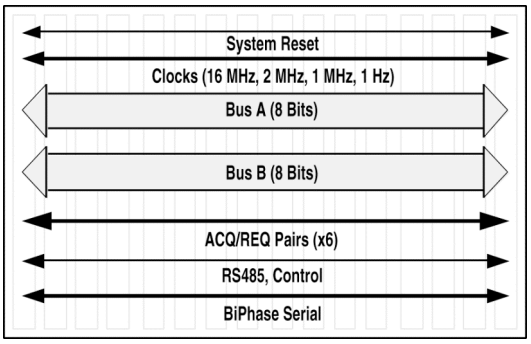

Figure 4. VME P1 backplane.

Controlling the bus is the bus arbitrator, or bus controller. This is the main module in the TG, and is responsible for the control of all traffic on the bus. The bus controller monitors all request lines and controls the acknowledgement lines. Through these control lines along with the two eight bit buses the bus controller transfers data on the bus from source devices to destination devices.

The bus controller module also contains the master clock source. The master clock is a low jitter precision crystal oscillator operating at 16 MHz. This base clock is then divided down by a CPLD to generate the networks time base of 1 MHz, along with the other required clock signals (2 MHz and 1 Hz).

In addition to the main bus controller there are two different classifications of devices operating on the bus, data generators and network interfaces. The generators are devices that by means of external stimulus or programmed sequences generate timing data. The network interfaces are devices that interface to the external timing network.

As mentioned previously the bus provides a number of different data transfer methods. At the root of the bus are two eight-bit data buses, along with pairs of request and acknowledge lines. One eight-bit bus (Bus A) is utilized by the data generator devices, while the other (Bus B) is used by the network interface devices. Each device that utilizes one of these buses receives its own pair of request and acknowledgement lines, which provides a simple bus arbitration protocol. The protocol operates as follows: A device on the bus that wishes to send data asserts its request line and waits for the bus controller to assert the accompanying acknowledgement line. After the device detects an acknowledgement it places its data on the appropriate eight-bit data bus, and lowers its request line. When the bus controller sees that the request line is de-asserted it processes the received data.

B. Event Decoder

A key module designed for the second generation timing generator is the event decoder. The event decoder provides 32 different optically isolated inputs representing hexadecimal codes 0x60 through 0x7F. These inputs when asserted send a request to the bus controller to broadcast the respective hexadecimal code on the timing network. The main purpose for the event decoder is to provide a means for sending event-driven timing information. In addition, it has allowed an incremental transition from the previous generation timing generator.

The functionality of the event decoder is straightforward. All 32 inputs are buffered through opto-couplers and are then sampled at a rate of 16 MHz by a Xilinx CPLD. The sampled signals are then latched on the first sample after a rising edge. A state-machine, running as a separate process, using a clock rate of 16 MHz looks at each latch and sends a transmission request to the bus controller for each asserted latch. If multiple latches are asserted the lower code values are transmitted first. Assuming the timing network is available, a latched code will be transmitted within one 1 MHz timing period.

C. Programmable Sequence Generator

The programmable sequence generator (PSG) provides much improvement over the previous timing generator. The PSG allows timing sequences to be remotely programmed, configured, and run all without hardware modifications. The PSG functions somewhat like an arbitrary function generator. It is designed to accept an ASCII file that lists times and associated trigger values (Fig. 5). It stores this file, and when either a software or a hardware trigger is asserted a timing sequence is started.

```
#PSG Example Script
#Time Code
0.0000  01  #Start Phase A
1.5000  62
2.0000  63
# ...
5.0000  04  #End Phase A
0.0000  02  #Start Phase B
# ...
2.0000  00  #End of Shot
2.5000  04  #End Phase B
5.0000  03  #Start Abort Phase
5.0000  00  #End of Shot
6.0000  04  #End Abort Phase
```

Figure 5. Timing script.

The PSG is set up to run three different timing sequences, or phases, triggered off of three different hardware and software triggers. The idea of the three phases was modeled after the previous generation timing generator, which had a pre-trigger CAMAC chain, a synchronous trigger chain, and an abort chain. In a normal experiment the pre-trigger chain is sent first, followed by a variable setup/delay time and finished with the synchronous trigger chain. In the ASCII file the phases are bounded by four special hexadecimal codes, 0x01, 0x02, 0x03, 0x04, with Phase A starting with code 0x01, Phase B code 0x02, and the Abort phase starting with 0x03. The last code of any phase is 0x04, which signifies the PSG to stop execution until retriggered. While the code 0x04 signifies the end of a phase, the code 0x00 signifies the end of the experiment.

Implementation-wise, the PSG utilizes a Xilinx CPLD, a parallel EEPROM, and a Microchip PIC microcontroller. The microcontroller is responsible for interfacing between the RS485 control network and the PSG. It coordinates downloading and verification of the ASCII files and handles software triggers between the PSG and the control computer. The CPLD controls all timing aspects of the PSG. Using a finite state machine, running at 16 MHz, it reads out of the EEPROM 32 bit-timing stamps comparing them to an internal 32 bit time base. Upon a match, the associated hexadecimal code is sent to the Bus Controller to be transmitted on the timing network.

D. Event Recording

The event recorder mirrors the programmable sequence generator in both functionality and implementation. The main difference being the recorder records incoming events and the PSG transmits events. The event recorder requires only power and the bi-phase serial stream from the timing generators backplane. All other required signals are generated onboard. This allows the event recorder to be an accurate measure of the workings of the generator.

Just as with the PSG, the event recorder utilizes a CPLD, PIC Micro, and an EEPROM for operations. The operation of each device is also the same but with data being piped from the

timing network to an ASCII file. The timing network is inspected by the CPLD and detected codes are decoded and timestamped by an internal counter. Both the timestamp and the code are stored in EEPROM. Upon receiving hexadecimal code 0x00, representing the end of the experiment, the CPLD releases control of the EEPROM to the PIC Micro. The micro then waits until the host computer requests the recorded data, at which time it sends the data over the RS485 bus to the host. On the host side the data is formatted and can be saved in a standard text file.

E. Bi-Phase Serial Converter

The bi-phase serial converter is the device that encodes and places timing codes on the timing network. The converter is basically a parallel to serial converter. It utilizes the 8-bit bus, bus B, on the backplane to receive codes from the bus controller and encodes them in the bi-phase protocol (Fig. 2). After encoding the data the converter drives five sets of fiber optic outputs with the encoded data. These five outputs represent the root level of the timing network. The outputs are designed to drive additional fan-out interfaces (Fig. 3), allowing for many network nodes all with the same driver biased propagation delay.

IV. CONCLUSIONS

There has always been a need for flexible facility wide timing synchronization at the DIII-D facility. This need over the years has been filled with multiple generations of timing networks, each generation improving upon the previous. The current generation, discussed here, allows for a great deal of flexibility in both functionality and expendability. The new system in its current configuration exceeds the past system in all measures, and adds a host of new features utilizing updated technology.

Future plans for this generation system include upgrades to existing features as well as new implementations. In the short term there are plans to upgrade the clock source to a higher accuracy more stable rubidium clock source. In addition there is the desire to add a higher resolution fiber optic based timing network in parallel with the current network. This network would support the newer higher speed data acquisition systems being installed.

ACKNOWLEDGMENT

This work was supported in part by the U.S. Department of Energy under DE-FC02-04ER54698.

REFERENCES

[1] C.A.F. Varandas, et al., "A VME timing system for the tokamak ISTTOK," Rev. Sci. Instrum. **66**, 3382, 1995.

[2] P. Sichta, J. Dong, R. Marsala, G. Oliaro, and J. Wertenbaker, "Developments to supplant CAMAC with industry standard technology at NSTX," Pinceton Plasma Physics Laboratory, unpublished.