IN-62
186022
14P

NASA Technical Memorandum 109001

A Hardware Implementation of a Provably Correct Design of a Fault-Tolerant Clock Synchronization Circuit

Wilfredo Torres-Pomales

July 1993

NASA

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23681-0001

Contents

1. Introduction

Many of the critical systems in aircraft are being implemented using electronic instruments. These systems must be able to operate in many different types of environments. Present reliability requirements for critical systems produce failure rate specifications that exceed those of commercially available digital devices. This forces the utilization of fault-tolerant architectures. This paper presents a hardware design of a fault-tolerant clock synchronization circuit and the results of the tests performed on it.

This work is part of NASA's effort towards the development of a practical validation and verification methodology for aircraft digital control systems. The circuit is intended to be part of a verified hardware base for the formally verified Reliable Computing Platform (RCP) for real-time digital flight control [1], currently under development at NASA Langley Research Center. Also, the circuit serves as an experimental test-bed for the ongoing FLY-BY-LIGHT/POWER-BY-WIRE project.

The system presented here was designed following the description given by Paul S. Miner in [2]. It is a four-clock system capable of achieving initial synchronization, and it tolerates a single transient or permanent fault. The system can tolerate only a single fault because no Byzantine Exchange is used in the communication among the clocks. The convergence function used is the fault-tolerant midpoint function. No further analysis of the theory behind the implementation will be presented here. This can be found in [2, 3].

2. System Description

The synchronization algorithm requires that the clock system operate in frames (i.e., cycles) and adjustments to the clocks be computed once every frame. The clocks have to exchange local times, and then compute adjustments to their own times using the information received. Each clock will then make its frame longer or shorter than the nominal frame length (R = 8192 ticks; 1 tick = 1 clock cycle) by an amount corresponding to the computed adjustment. Because the algorithm requires the time from at least three nodes to compute an adjustment, proper behavior must be defined when a clock circuit does not receive enough information. The design provides two options:

Assumed-Perfection - assume all clocks are observed to be in perfect synchrony, or

Assumed-End-of-Frame - assume that unobserved clocks are seen at the end of the frame (Local Clock = R), and then compute an adjustment.

Both options are available in the implementation. Prior to a test, the desired option can be selected by setting a DIP switch.

The dominant parameter in the formal theory is the inherent error in reading a remote clock. An efficient communications network is required to achieve a tight synchronization. This system uses point-to-point optical communications. The optical transmitter-receiver interface is composed of off-the-shelf components and provides a transmission rate of 125 Mbits/sec. A maximum read error of 0.5 ticks was estimated from an engineering analysis of the communication network.

Figure 1 shows a block diagram of one of the four identical clock circuits. Each circuit performs several functions including: keeping a local time, performing the transmissions and receptions of the frame counter values, voting the frame values, computing and applying an adjustment to local time, determining if there are at least three clocks synchronized in the system, and determining if its local time is synchronized with the rest of the system. Based on the current frame information and state, a clock is capable of determining what its next state should be: initialization, maintenance or recovery of synchronization. The clock circuits also provide external controls used during the experiments

The local time has two components: the Local Clock and the Frame Number. If i is the current Frame Number and LC is the current value of the Local Clock, then the time elapsed since the system achieved synchronization is iR + LC. Processors connected to this system must perform this operation to determine the local time from the information available at the Processor Inteface.

The Local Clock is a 16-bit counter driven by a 10MHz oscillator. It goes through the Monotonic Clock Logic before being sent to the Processor Interface. The purpose of this logic is to ensure that the local time is a monotonically non-decreasing function of real time. This logic limits the output of the local clock to R, so that backward jumps in local time are inhibited.

The Frame Counter keeps track of the number of frames since the system first synchronized. It is an 8-bit counter and its value is determined by the state of the clock. When a clock is synchronized with the system in normal operation, its Frame Counter is incremented by one at the beginning of each new frame. During initialization, the value is set to zero until synchronization is achieved. If a clock determines that it has lost synchronization, the Frame Counter will not be incremented; the Majority Voter will recover the correct value, which will be loaded in the Counter at the end of the frame.

The algorithm requires the clocks to exchange local time information in every frame. The clocks in this implementation do not transmit both their Local Clock and Frame Counter values. Instead, some time before the local clock reaches time Q (the transmission time, Q = R/2), the Timing Logic will signal the Transmitters to send the frame value to the other clocks. This transmission provides sufficient information about the transmitting clock's local time. The receiving clocks compute the difference between the actual times of arrival of the received transmissions and the expected arrival time, and then use those differences to compute an adjustment to their local clocks. The received frame values and the local Frame Counter value go through the majority voter which is used to maintain agreement among the clocks on this component of the local times. If the Majority Voter cannot find a majority among the frame values, it will assert the NO_MAJORITY signal at the Processor Interface and the local Frame Counter will keep its current value into the next frame. This way of exchanging local times satisfies the efficiency requirements to achieve a tight synchronization and requires minimum use of the communications network, which could be shared with other components in a fault-tolerant computing system.

The Timing Logic also controls other functions. It uses the local clock value and the computed adjustment to identify when the clock is close to the end of the frame and disables the Receivers to allow the electronics to process the received information before starting a new frame. The effects of this no-reception window on the performance of the system are discussed in section 3. The Timing Logic also provides the signals needed by the Adjustment Computation logic to compute an adjustment when not enough information has been received. At the end of the frame, the Timing Logic clears all the information processing circuitry to prepare the system for the next frame and signals the Local Clock to start a new frame and reset its value to zero.

The State Determination logic uses the received information and the current state to determine the next state. It determines whether or not there is a group of at least three clocks within the maximum allowed skew D (= 11 ticks or 1.1μs at 10MHz), and also compares the absolute value of the computed adjustment to D. The results are then used to determine if the system is synchronized and whether or not the local time is in sync with the rest of the system. The State Determination logic decides when the Frame Counter has to be incremented, signals the Restart Operation Logic when the system is not synchronized, and also provides the OUT_OF_SYNC signal at the Processor Interface to indicate when the local time is not synchronized.

The Restart Operation Logic is responsible for resetting the clock circuit. This reset will cause a clock to enter the initialization state in which the Frame Counter is kept at zero and the OUT_OF_SYNC signal asserted until synchronization is achieved. The circuit goes to this state immediately after power-up and also when the State Determination logic signals that the system has lost synchronization after already being synchronized.

Because of the experimental nature of this implementation, external controls were included. These allow the introduction of faults into the system and enable the experimenter to create situations of special interest when studying its performance. The DAS Decoder is used to decode the external control signals in real-time (DAS stands for Digital Acquisition System, which was the system chosen to control and study the implementation.). Some implementation parameters can be set using DIP switches. Some of these parameters include: the frame length R, the time for transmission Q, the maximum allowed skew D, and the length of the no-reception window at the end the frame. Also, the desired behavior when a circuit receives insufficient information to compute an adjustment can be selected using the DIP switches.

3. Experimental Results

There are several scenarios which were of special interest in this implementation. These are: achieving initial synchronization, maintaining synchronization, recovery from a single transient fault, recovery from a massive transient fault, and maintaining synchronization in the presence of a single permanent fault. There was also interest in determining the effect of the no-reception window at the end of the frames and comparing the behavior of the system for both initialization algorithms: Assumed-Perfection, and Assumed-End-of-Frame.

According to the theory, a system which correctly implements the synchronization algorithm will be able to achieve and maintain synchronization if there are enough good clocks present (i.e., at least three). Experimental observations did not show any significant difference in the normal behavior of the clocks when one or the other initialization algorithm was used. Because of this, the results will be presented for only one case: Assumed-End-of-Frame.

Figure 2 is a plot of a typical response of the system during initialization after a power-up. Only the local clocks are included because they are the ones directly affected by the synchronization algorithm. The reference time was provided by a 32-bit counter at 10MHz (1 tick = 100ns). The figure shows that after power-up the clocks start with random values. The system is not considered operational until after the initial reset of all the circuitry. This occurs from approximately time 16,000 on clock 1 until time 30,000 on clock 4. The net effect of the clocks resetting at different times is that they wake up at different times. As can be seen, the system achieves full synchronization by time 51,000 and then maintains the synchronization. Taking the time at which the last clock woke up as a reference (i.e., clock 4), synchronization was achieved in 21,000 ticks or 2.1ms. Once the system achieved initial synchronization, it stayed in that state for as long as the observations lasted. The longest continuous observation was of approximately 15 minutes. The synchronization was always within two ticks. This behavior was much better than the estimated 11 ticks because the characteristics of the parts used in the implementation exceeded the ones used in the analysis.

Figure 3 shows the response of the system to a single transient fault. The fault was introduced in clock 2 at approximately time 23,000 when it started a new frame at LC = 1360 instead of zero. The other clocks were able to maintain synchronization because there were enough good clocks present. Also clock 2 was able to get back in sync within one frame because the transmissions from the good clocks showed that the were synchronized. Recovery of the frame value also took only one frame for this type of fault.

An example of a massive transient fault is presented in figure 4. In this case clocks 2, 3, and 4 were affected by transients which made them reset to values different from zero. Clock 1 was not affected. The jumps in clocks 3 and 4 were to values of LC greater then the transmission time Q, and so they did not transmit in this frame. Because none of the clocks had sufficient information to compute adjustments they all assumed End of Frame arrival and extended their frames. However, because they were previously in sync, they decided that a massive transient occurred, reset their circuitry and started again. The system regained synchronization approximately 19,000 ticks or 1.9ms after the fault was introduced. Synchronization was later maintained.

The performance of the system under a single permanent fault was excellent. Figure 5 shows the response when a fault on clock 2 rendered its receivers inoperational after reference time 22,000. Because of this fault, clock 2 was unable to compute adjustments but kept transmitting in every frame. Since it did not have sufficient information to compute adjustments, it always assumed End of Frame arrival (Note the flat extension on its curve). Clocks 1, 3, and 4 never lost synchronization and the transmissions from clock 2 had no negative effects on their behavior.

The effect of the no-reception window at the end of every frame was also investigated. There was a concern with the possibility of enabling a 2-2 split by using this window. However, this condition has not been observed to date. Were it to occur, it would be an unstable state, since the drift and jitter in the oscillators would tend to move the clocks to a different state. Also, if a recovering clock receives all the transmissions during this window, then it assumes that the system has lost synchronization, resets its circuitry, and starts again. Synchronization takes only one frame following the reset because the clock circuit will receive three synchronized transmissions in the first frame.

The effect of the no-reception window could be a problem if an initializing clock happened to receive two or more of the transmissions in this window. In this situation the clock behaves as if there were no other clocks present in the system. If the clock is operating under the Assumed-Perfection algorithm, the drift in the oscillators will be the only factor which would get that clock out of this state. However, if the clock is in the initialization state and using the Assumed-End-of-Frame arrival option, then it will extend its frame and receive all the transmissions in the next frame. This will enable the clock to synchronize within one frame. This behavior of the system makes the Assumed-End-of-Frame option the preferred one to handle situations in which a clock does not have sufficient information to compute an adjustment.

4. Concluding Remarks

Based on the collected experimental data, the design seems to satisfy the requirements of the synchronization algorithm. As mentioned before, synchronization was better than expected: within 2 ticks for the experimental versus 11 ticks for the analytical. Also, the system never failed to achieve initial synchronization and always recovered from transient faults with expected behavior.

Work is currently under way to design and test a new version of the clock synchronization system. This design will use a different block model than the one presented in [2]. It will perform the same functions, without the experimental controls, and minimize the amount of logic needed. It is expected that the new system be will able to operate at frequencies in excess of 33 MHz.

5. References

[1] R.W. Butler and B.L. Di Vito, "Formal Design and verification of a reliable computing platform for real-time control: Phase 2 results", Technical Memorandum 104196, NASA, Langley Research Center, Hampton, VA, Jan. 1992

[2] P.S. Miner, "A verified design of a fault-tolerant clock synchronization circuit: Preliminary investigation", Technical Memorandum 107568, NASA, Langley Research Center, Hampton, VA, Mar. 1992

[3] P.S. Miner, P.A. Padilla, and W. Torres, "A provably correct design of a fault-tolerant clock synchronization circuit", Proceedings of the 11th DASC, Oct. 1992, pp.341-346

Figure 1: Block Diagram of Clock Circuit

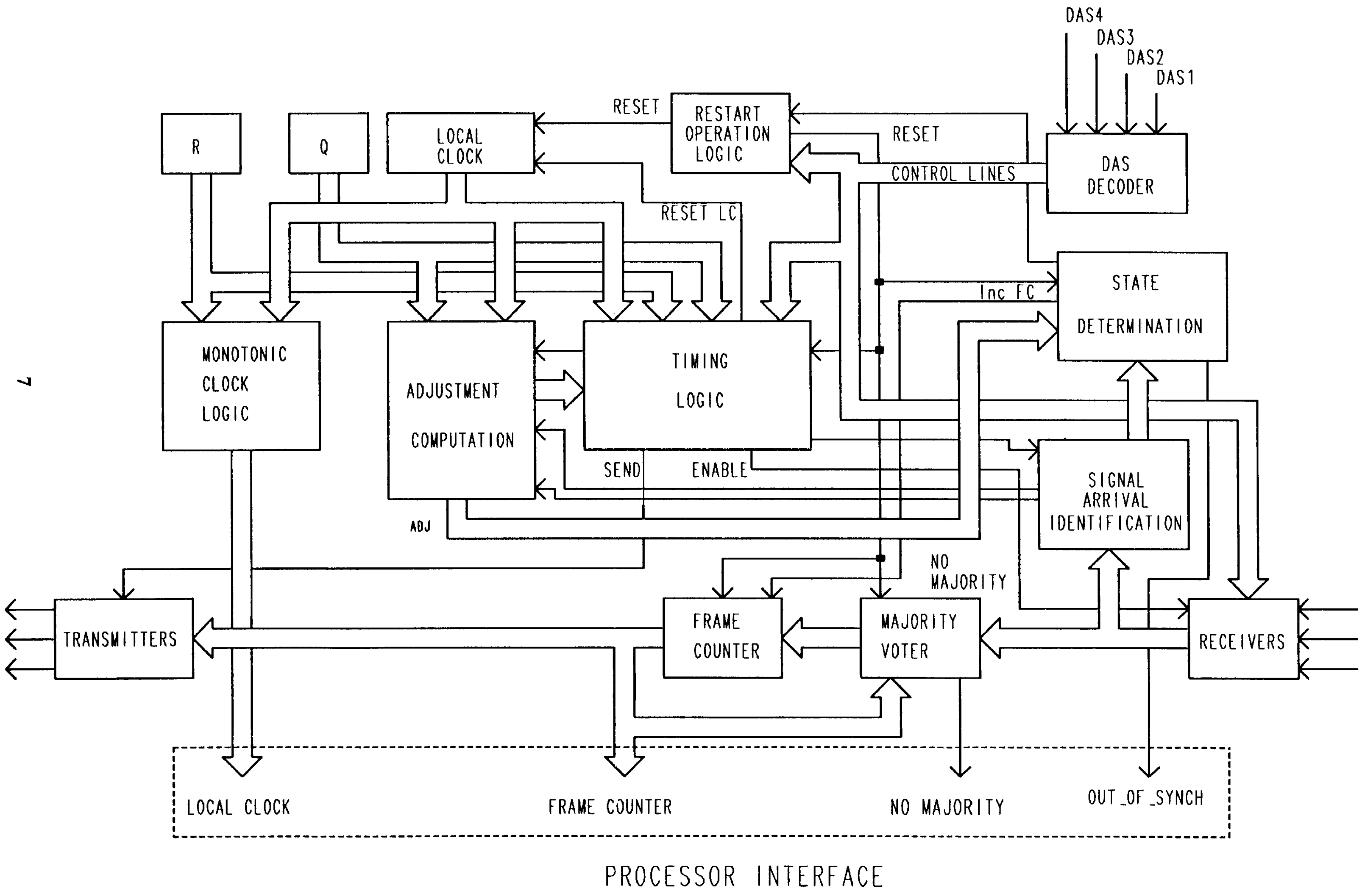


Figure 2: Clock System During Initialization

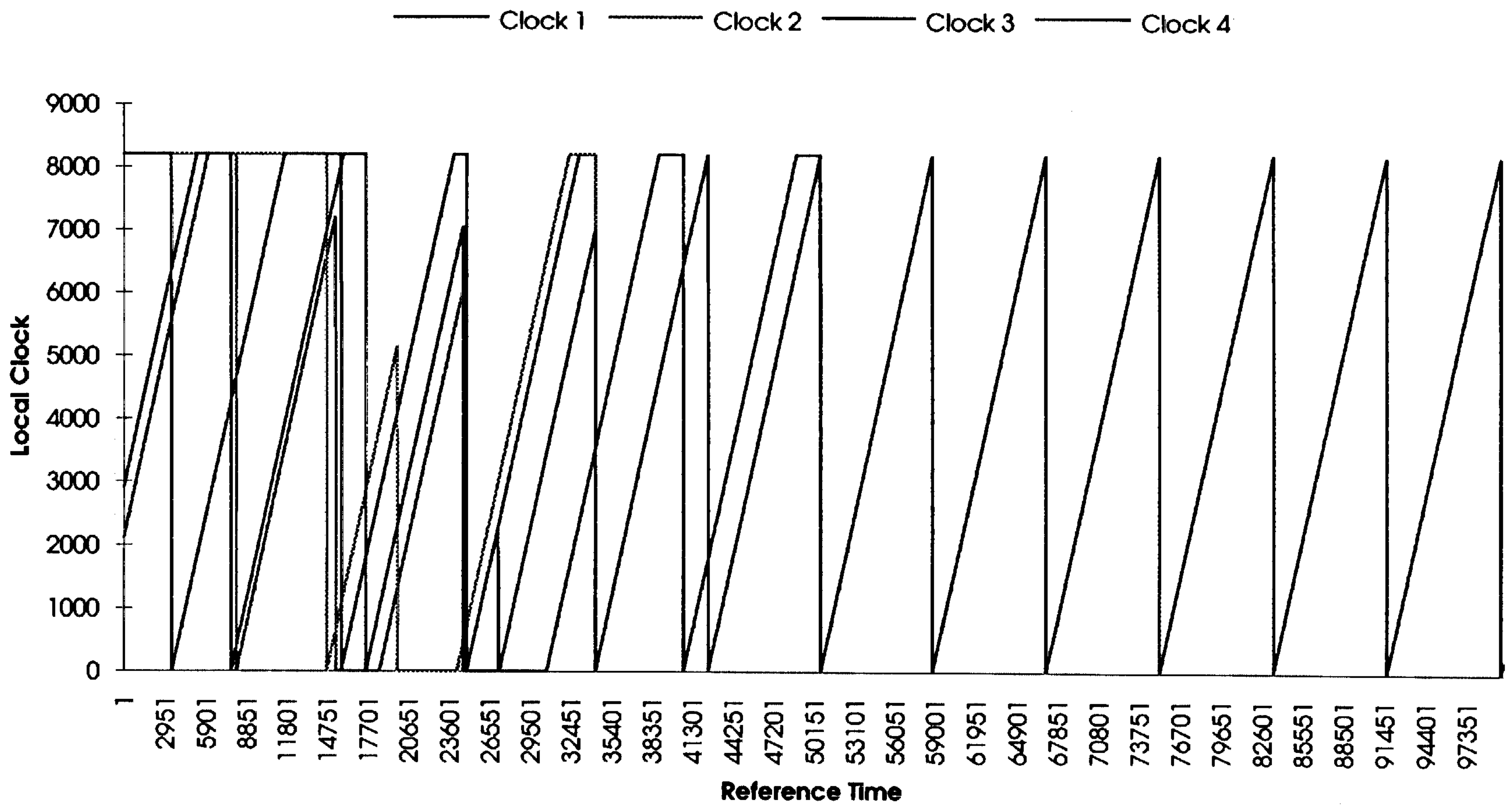


Figure 3: Response to a Single Transient: Clock 2 Resets to 1360

Algorithm: Assumed End-of-Frame Arrival

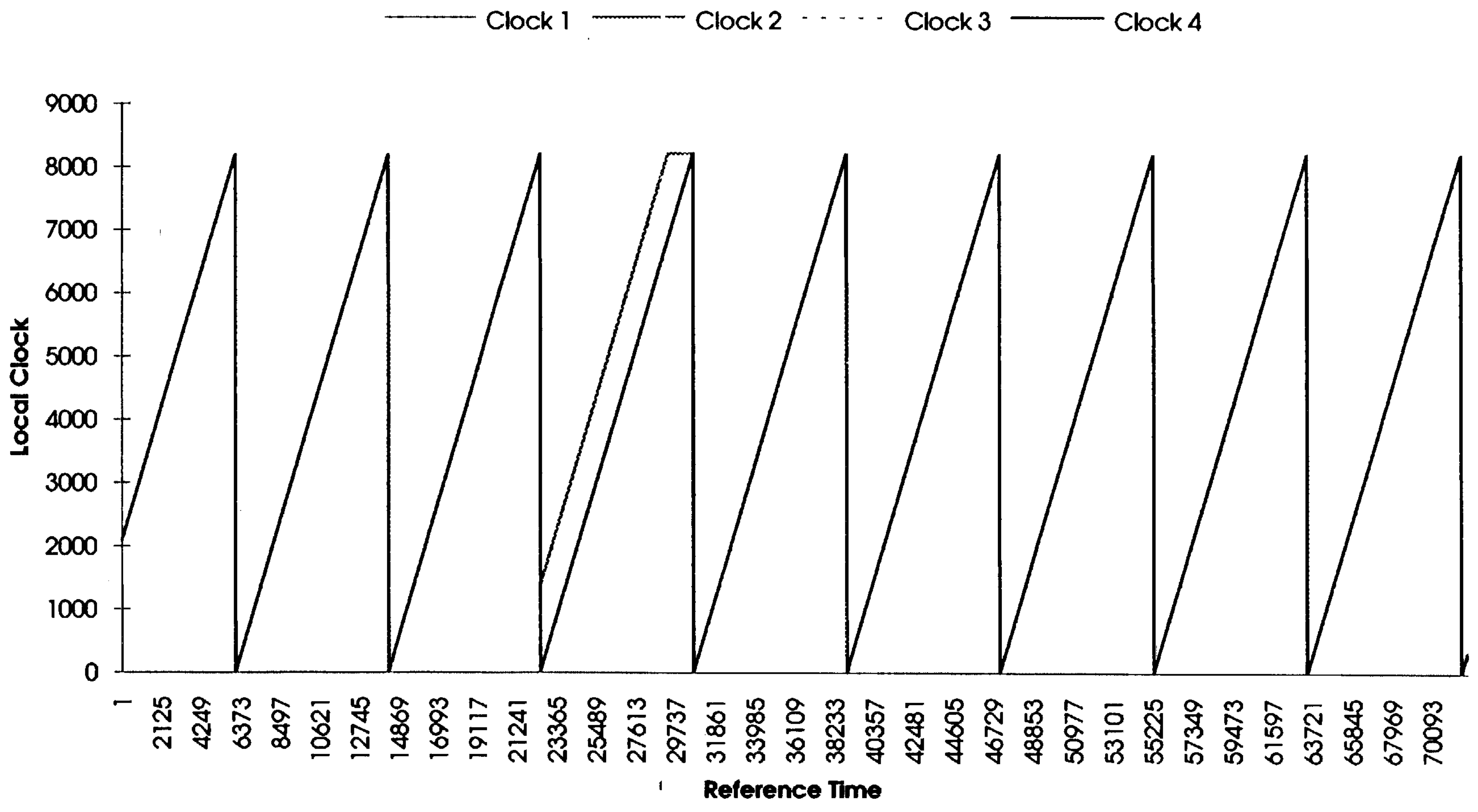


Figure 4: Massive Transient: Clocks did not reset to 0

Algorithm: Assumed End-of-Frame Arrival

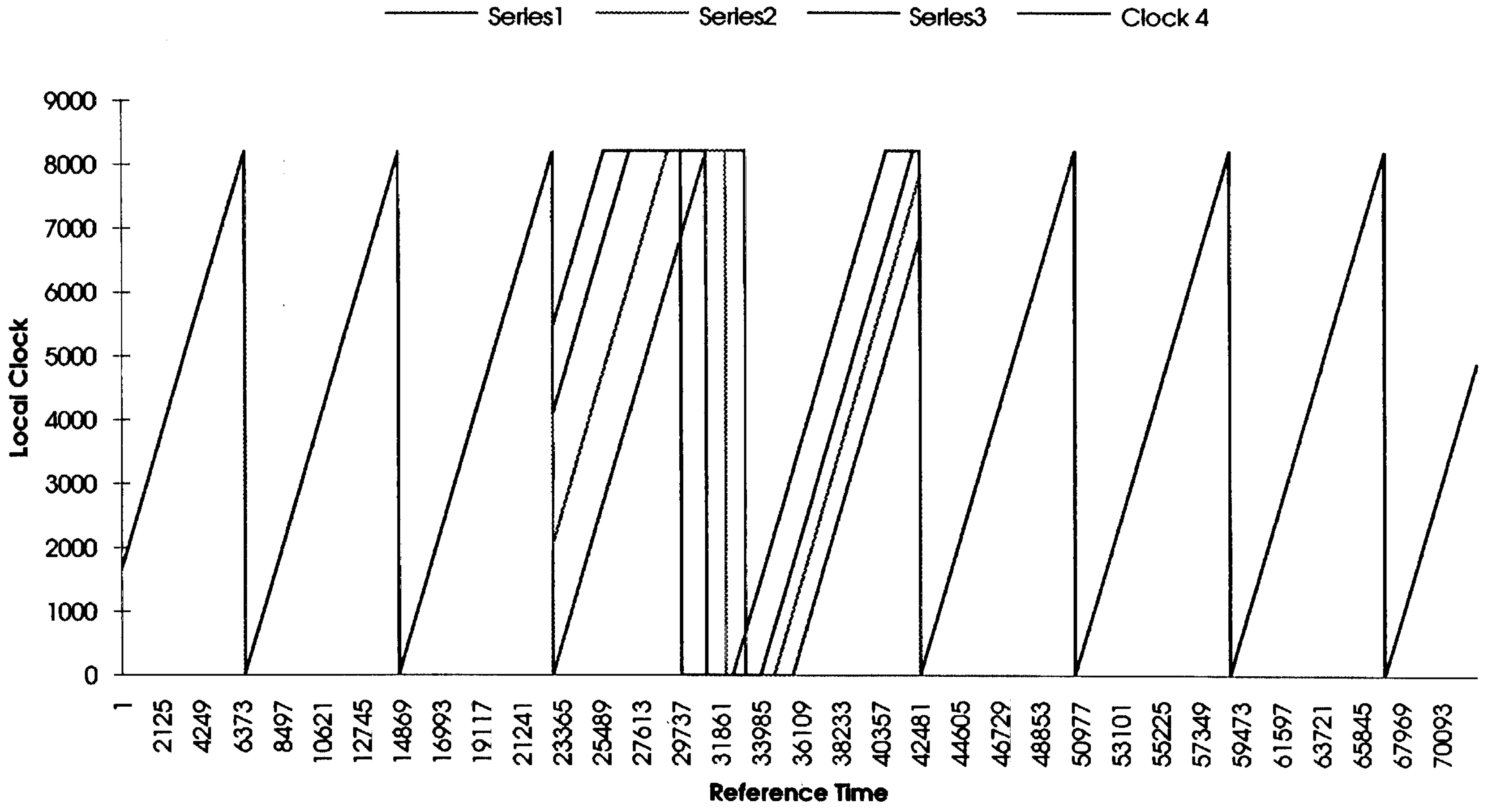
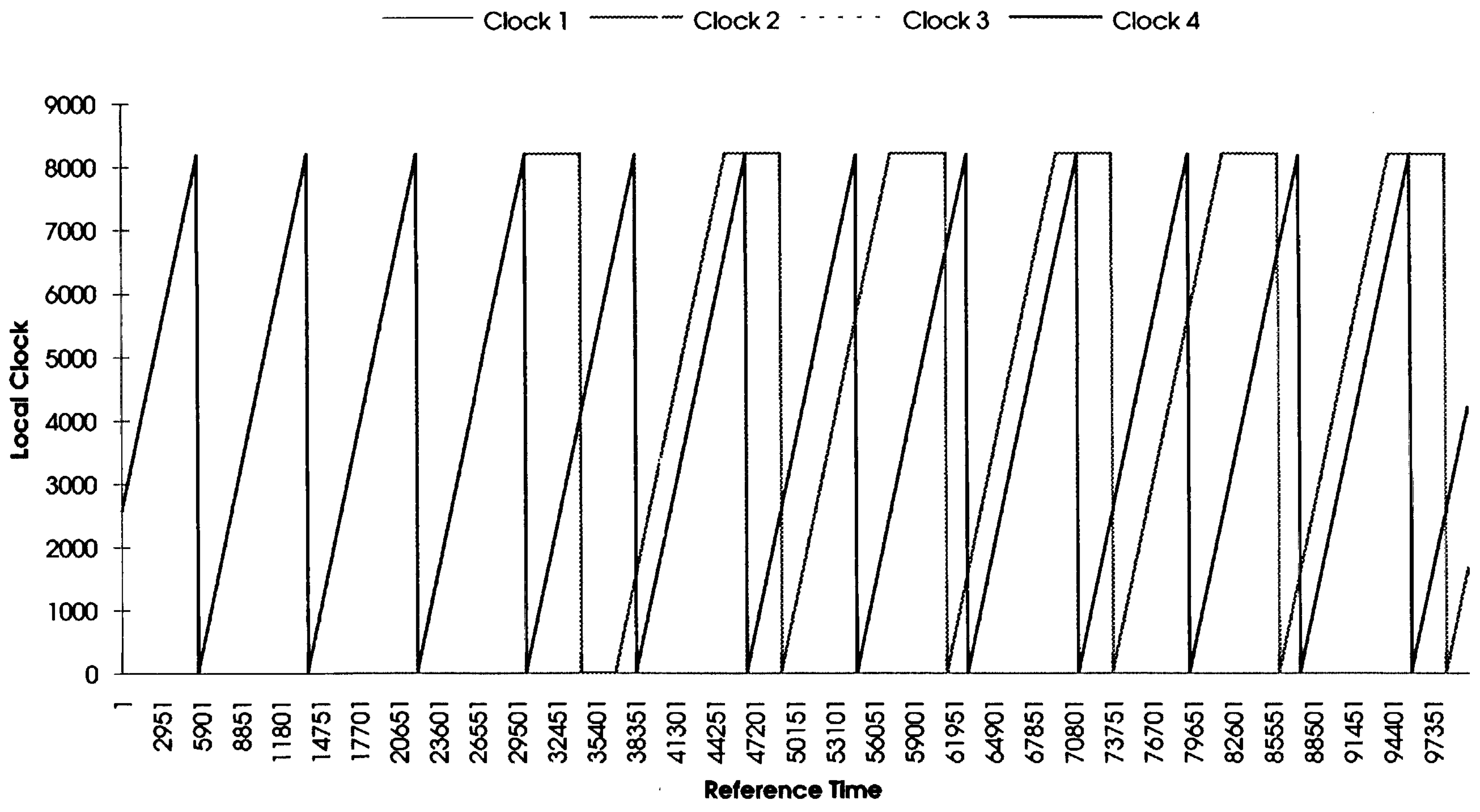


Figure 5: Single Permanent Fault: Receivers on Clock 2 Inoperational



REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE July 1993	3. REPORT TYPE AND DATES COVERED Technical Memorandum

4. TITLE AND SUBTITLE A Hardware Implementation of a Provably Correct Design of a Fault-Tolerant Clock Synchronization Circuit	5. FUNDING NUMBERS WU 505-64-10-10
6. AUTHOR(S) Wilfredo Torres-Pomales	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-0001	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001	10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-109001

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category 62	12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A fault-tolerant clock synchronization system was designed to a proven correct formal specification. Formal Methods were used in the development of this specification. This paper presents a description of the system and an analysis of the test performed. Plots of typical experimental results are included.

14. SUBJECT TERMS Clock Synchronizaiton, Formal Methods, Point-to-Point Communications	15. NUMBER OF PAGES 13
	16. PRICE CODE A03

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102