NIST Technical Note 2138

Design and Capabilities of the Assistive Clock Fusion Testbed

D. M. Anand

This publication is available free of charge from:
https://doi.org/10.6028/NIST.TN.2138

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

NIST Technical Note 2138

Design and Capabilities of the Assistive Clock Fusion Testbed

D. M. Anand
Smart Grid Program
Engineering Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.TN.2138

February 2021

U.S. Department of Commerce
Wynn Coggins, Acting Secretary

National Institute of Standards and Technology
James K. Olthoff, Performing the Non-Exclusive Functions and Duties of the Under Secretary of Commerce for Standards and Technology & Director, National Institute of Standards and Technology

National Institute of Standards and Technology Technical Note 2138
Natl. Inst. Stand. Technol. Tech. Note 2138, 37 pages (February 2021)
CODEN: NTNOEF

This publication is available free of charge from:
https://doi.org/10.6028/NIST.TN.2138

Abstract

In response to the 2007 Energy Independence and Security Act (EISA), Public Law 110-140, NIST collaborates with the private sector and leads the coordination and acceleration of smart grid interoperability and security standards. This effort has resulted in the NIST Framework and Roadmap for Smart Grid Interoperability. This effort also involves NIST laboratories developing measurement science to advance research, standardization, testing and implementation of the NIST Framework. This measurement science and Framework ultimately help increase asset utilization and efficiency, improve grid reliability, and enable greater use of renewable energy sources in the grid.

Critical to smart grid operations is precision timing synchronization. It is needed to ensure grid functions occur at precise times and in correct sequences. To test and evaluate methods of time fusion, NIST contracted the design and building of a clock fusion testbed This device is intended to support software algorithms, as well as, some low-level frequency synthesis techniques. Called the "Assistive Clock Fusion Testbed", the device includes all basic components and is pre-loaded with a reference implementation of a simple fusion algorithm on a hybrid System-on-a-Chip (SoC). The testbed is able to meet the requirements of existing precision timing synchronization standards, ranging from one microsecond to hundreds of nanoseconds, and can support tests related to conformance and timing system, end-to-end performance.

This document outlines the design of the device. It also discusses a use case in which the device combines the frequencies generated by two oven-controlled crystal-oscillators to produce a synthesized reference frequency, compliant with wander limits specified in the International Telecommunication Union G.8262 (Timing Characteristics of Synchronous Equipment Clock) standard. The device features cover the most generic configuration of the system. Additionally, the document addresses the device's potential extensions and configurations, suited to specific experiments and performance tests, related to timing resilience.

Key words

Clock testing; Power Systems Metrology; Research apparatus; Smart Grid

Table of Contents

List of Tables

List of Figures

1. Introduction: The Need for Precise Time Synchronization on the Grid

The power distribution industry recognized the need for a distributed precise timing synchronized system after the 2003 Northeastern blackout. Conceivably, such a system could preempt cascading failures on the electrical grid through use of precisely synchronized measurement and control systems to support rapid detection and response to excursions in the grid. The blackout also heightened the need for more accurate data recorders on the grid to aid rapid fault diagnosis. Inaccurate timestamps caused significant delays in diagnosing the blackout's cause [1].

Realizing the importance of time synchronization, the power distribution industry adopted several standards, setting minimum requirements for time synchronization and time-stamping accuracy. However, the grid is just one of many nationwide interconnected systems, requiring distributed, time synchronized communication and control. Such networks with embedded devices – often called an "internet-of-things" or cyber-physical systems – are exemplified by our transportation, telecommunications (telecom), gas pipelines, and other networks.

The diversity of these systems has meant developing made to order solutions for timing and time transfer. Telecom operators, for example, use a combination of Global Navigation Satellite System (GNSS) and digital telephone signaling schemes at the cellular base stations.

More recently, two Ethernet compliant standards have been developed to meet the need for significant data backhaul capacity and address synchronization in the telecom environment:

- IEEE (Institute of Electrical and Electronics Engineers) 1588v2 PTP (Precision Time Protocol) [2] .
- ITU (International Telecommunication Union) G.8261 SyncE (Synchronous Ethernet) [3].

Some operators prefer SyncE because it conforms to the Layer-1 based approach to synchronization, used with Time Division Multiplexing. However, the challenge with SyncE is that it must be supported throughout the network, possibly requiring upgrades to a large number of network elements.

Other operators prefer IEEE 1588v2, since it can coexist with existing Ethernet backhaul equipment and is better suited for cases where smaller, more rapidly deployed cell sites are required [4]. Even with these Ethernet backhaul time transfer technologies, the loss of a T-Carrier, T1, Digital Signal, or an E-carrier, E1, phase synchronization signal, would likely cause a slow degradation of voice service, even if perceptible degradation of the service currently would likely take days or weeks.

[1] NERC Steering Group (2004). Technical analysis of the August 14, 2003, blackout: What happened, why, and what did we learn. Report to the NERC Board of Trustees.
[2] IEEE Standard 1588-2019 (Revision of IEEE Std 1588-2008)
[3] International Telecommunication Union "Timing and synchronization aspects in packet networks"; G.8261
[4] Lim, M. Deploying SynceE and IEEE 1588 in Wireless Backhaul. March 2012

The Long Term Evolution (LTE) requirements for telecom systems demand a more precise timing infrastructure. It will need a global phase, as well as, frequency and time-of-day (ToD) synchronization. This may only be required for network Multiple-Input Multiple Output (MIMO) and Multimedia Broadcast Multicast Service-Single Frequency Networks (MBMS-SFN) in the 3rd Generation Partnership Project (3GPP).

Currently, the only candidate technology, proven to support phase alignment, frequency recovery and time-of-day (ToD) synchronization, is GNSS/GPS. Phase synchronization may be possible with IEEE 1588v2 and ITU G.8261. However, challenges exist in implementing, as well as, standardizing for the current nominal performance of GNSS control and user segments. Also, cellular telecom providers are primarily reliant on GNSS, because of the need for phase synchronization, as well as, frequency and time, combined with the rate at which other standards-based packet synchronization solutions are scaling. Moreover, the resilience of the telecom system faces a serious threat, since most GNSS systems have been found to be vulnerable to jamming, spoofing and data attacks [5].

Operators seek an 'all-in' strategy, until complementary wide-area time dissemination standards and technologies reach the accuracy and flexibility of GNSS. This strategy calls for use of mulitple time transfer technologies, and would combine them at the cellular base station in order to simultaneously achieve required performance and resilience.

Large-scale cyber-physical systems share an interest in developing ways to combine several time transfer and synchronization methods. The power system stakeholders presented their case for a "diversely routed" timing system in a workshop at NIST, documented in NIST SP 1500-08 [6].

1.1 Need for a Testbed

Combining multiple time transfer technologies so as to maximize their relative advantages is still an open research question. Significant research is needed in this area and a set of common benchmarks must be developed. These will aid testing of integrated timing methods, and adoption across all domains that stand to benefit.

Algorithms and hardware that can combine multiple timing signals for a 'more reliable' source are prevalent in the timing world. Several exist in commercial systems. Such technologies are also identified in the literature as assisted partial timing support, clock fusion, diversely routed time, clock ensembling, automated timing failover, multi-source synchronization etc. These methods differ significantly to meet their unique domain application.

NIST identified the need for a testbed that could enable timing technologies to be combined and evaluated. Ultimately, such a testbed would accelerate cross-domain adoption and iterative improvements. Thus, NIST commissioned construction of the "Assistive Clock Fusion Testbed" (ACFT). The term 'Clock Fusion' is intended to be inclusive. It refers to all the methods, hardware and technologies that purport to improve the reliability of an expressed clock by fusing or combining multiple sources of phase, frequency and

[5] Buesnel, G. and Holbrow, M. GNSS Threats, Attacks and Simulations, June 2017.

[6] Li-Baboud, Y. et. al., Timing Challenges in the Smart Grid. NIST Special Publication (SP) 1500-08, 2017

ToD (or subsets thereof).

1.2 Design motivation for the Assistive Clock Fusion Testbed

In pursuing the ACFT, NIST sought a system that would:

- Test and evaluate a range of clock fusion technologies.
- Support a wide variety of signal processing algorithms, as well as, hardware components.
- Reflect implementation challenges, performance considerations and hardware limitations that might inhibit wide scale adoption of developed fusion technologies.
- Use commercial off-the-shelf components, where possible, so as to enable researchers to replicate the testbed and generate repeatable validation experiments.
- Provide researchers with:
  - Industry standard probing features including the ability to add software probes via common testing tools.
  - Hardware probes implemented in hardware logic on a Field Programmable Gate Array (FPGA) fabric.
  - A comprehensive set of analog probing points where synthesized frequencies, pulses and phase errors may be measured and compared with reference sources.

1.3 Assistive Clock Fusion Testbed – Equipment Overview

The fully assembled ACFT hardware is contained in a standard DTX (203mm $\times$ 244mm) chassis as shown in Figure 1. The device is portable, and has all the components needed to perform clock fusion experiments. Eight 50 Ω matched SubMiniature version A (SMA connectors) on the front panel provide easy access to frequency probe points on the device. These connectors can be configured as both inputs and outputs to two Frequency Synthesizer Boards and two Oven Controlled Crystal Oscillators (OCXOs) (see Figure 2). The ACFT's equipment is detailed as follows:

- Each frequency synthesizer board features a Silicon Labs Si5342 Jitter Attenuator with three independent Digital Signal Processor based Phase Locked Loops (DSPLLs).
- The Si5348 permits an external reference input to determine the device's overall frequency accuracy and stability.
- Each DSPLL may be individually configurable as a SyncE PLL, IEEE1588 digitally controlled oscillator (DCO) or a general-purpose PLL for an analog clock source.
- The DCO can be configured to provide clock steering or reference tracking to within 10^{-7}% error.

Fig. 1. Front view of ACFT showing the front panel connector panel.

- A 12.8 MHz RAKON [7] low phase noise, oven controlled, compensated, crystal resonator serves as a default reference source.
  - This enables it to meet high-end, telecom accuracy requirements of frequency variation below 10^{-10} Hz measured over 10 seconds and drift less than 50 μs over a 24 hour hold over [8].
  - A more stable source can be substituted using appropriate front panel connector as described in Section 2.
- The underlying signal processing system uses a Xilinx ZYNQ-7000 System-on-Chip (SoC) [9]; this combines
  - An XC7Z020 Field Programmable Gate Array for signal processing.
  - Dual ARM Cortex-A9 Processors (for general purpose computing and logging functions.
- ARM Cortex-A9 provides sufficient computational bandwidth to run a full featured Linux Operating System (Fedora v16.2 supported, Xilinx Petalinux installed by default).
- A single board computer is built around the ARM processor with 1 GB DDR3 component memory to facilitate software development by connecting a keyboard and monitor, without the need to 'target' the processor or build software externally.
- For all internal development and testing, a compiler front-end and tooling infrastructure for the C family of languages called CLANG [10] is used.

[7] Commercial entities, equipment, or materials identified in this document in order to describe an experimental procedure or concept do not imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.
[8] Rakon, ROX5252T1 OCXO Datasheet Rev. 2
[9] See Footnote 7
[10] Open-source software released under the University of Illinois/NCSA License.

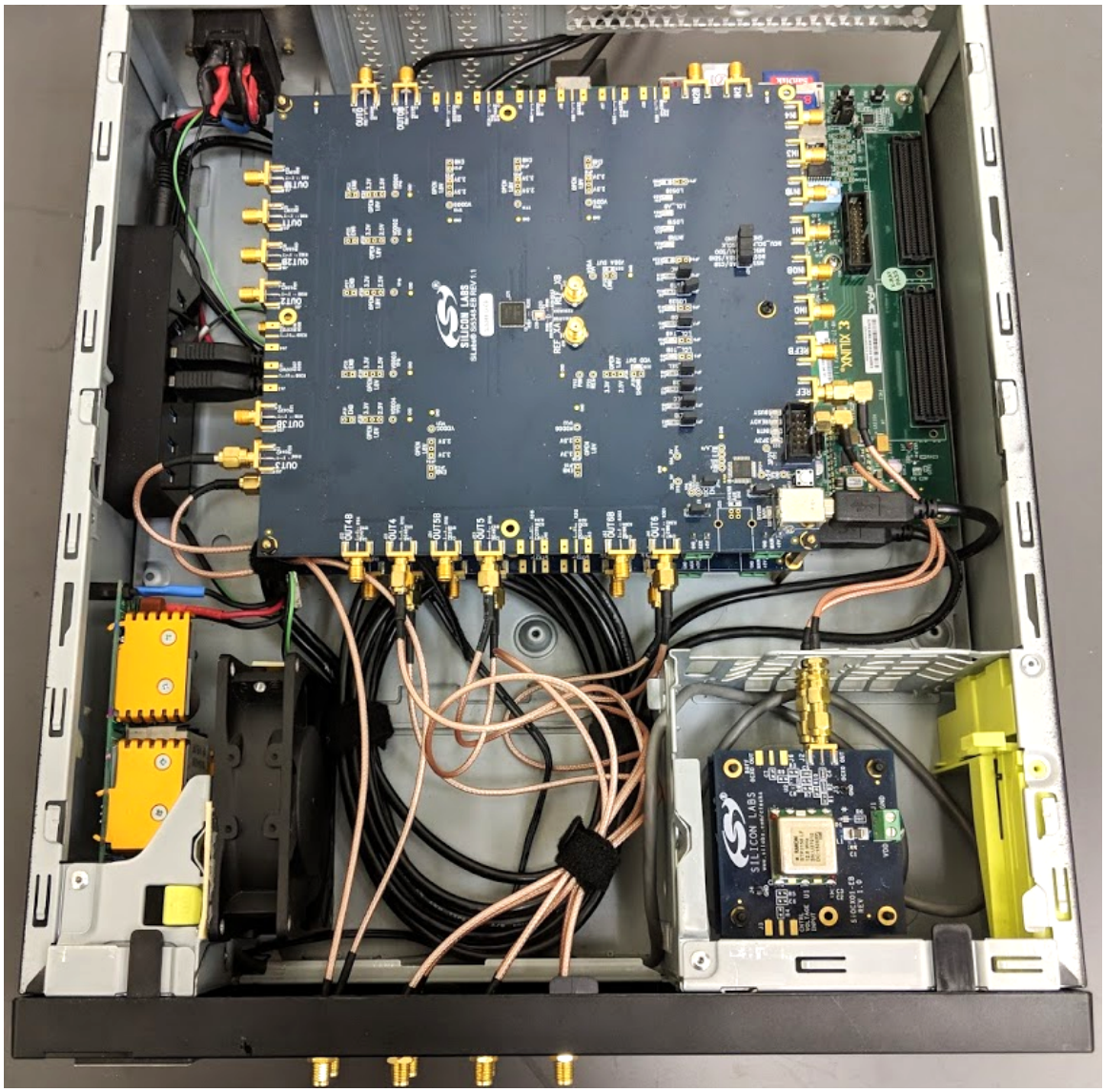


Fig. 2. Top view of ACFT (top cover removed) showing frequency synthesizer boards and installed reference frequency sources (OCXOs).

In developing the ACFT, NIST emphasized testing and implementing a wide variety of communication interfaces so as to accommodate as many test devices as possible. The version of the single board computer provides discrete pinouts and connectors for:

- Universal Serial Bus (USB)
- Universal Asynchronous Receiver/Transmitter (UART)
- Inter-Integrated Circuit (I2C)
- Controller Area Network (CAN) Bus
- 10/100/1000 Mbps Ethernet
- High-Definition Multimedia Interface (HDMI)

The hybrid Processor/FPGA SoC architecture enables Digital Signal Processing (DSP), monitoring and logging functions to be handled on the ARM processor, or on the 702 FPGA per experimenter preference. Industry-standard, FPGA Mezzanine Connectors (FMC) enable scaling and customization with daughter cards.

Additional daughter boards may be added via FMC. This allows the addition of hardware DSP processors, FMC compliant implementations of precision reference sources, such as a chip-scale atomic clock (CSAC), GPS receivers, or network based clock synchronization cards. Software drivers and FPGA interfaces for these daughter cards will be required for each card added.

To help diagnose and measure triggering functions, an FMC card with a digital-to-analog converter and low-voltage digital signaling interfaces have been installed by default. These cards enable physical connections to commonly used metrology equipment for precision clocks, such as Time Interval or Frequency counters and Triggered oscilloscopes.

The hybrid signal processing board has an analog-to-digital converter (ADC) that can be configured by an experimenter. The ADC can digitize two analog frequency outputs on the frequency synthesizer board at up to 106 samples per second. And interacts with a dedicated set of drivers and sampled data buffer called the “XADC Block” (This pipeline is described in more detail in Section 3).

These sampled data are discarded on a first-in, first-out basis from the buffer by default. However, an experimenter may copy the buffer to non-volatile, flash memory, if needed. Since this data pipeline (with drivers for the XADC block) is handled by the Linux operating system, the experimenter has the ability to add pre-processing steps that might filter, analyze or flag the sample data prior to committing to memory. A limitation of this pipeline is that the sampled data in the buffer are limited in rate and resolution by the Analog-to-Digital Converter on the hybrid signal processing board.

For experiments that do not require software-based preprocessing, experimenters may connect metrology equipment and external clocks, directly to analog frequency interfaces on the frequency synthesizer boards. This latter approach is the simpler choice to perform clock fusion experiments that only involve tuning the DSPLLs on the Si5342.

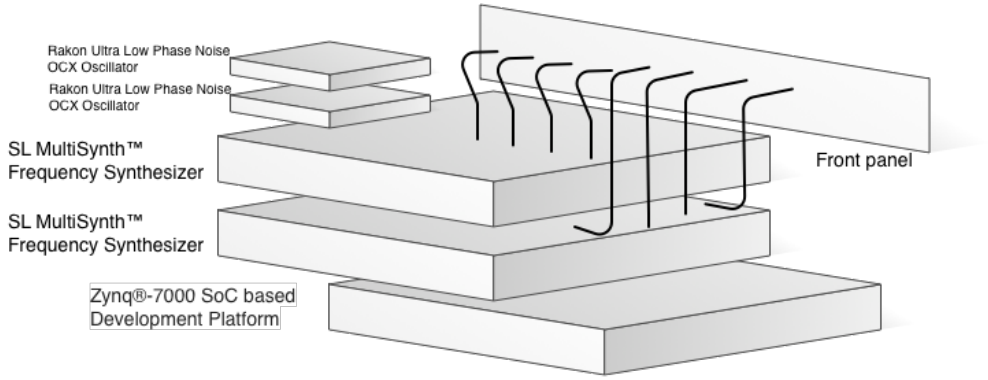


Fig. 3. Schematic layout of the functional hardware components in the ACFT. Front panel connections are reconfigurable. All connection options are discussed in Section 2.

The schematic in Figure 3 shows the layout of the frequency synthesizer boards, the ZYNQ SoC based hybrid signal processing/development platform, as well as, the two daughter cards with telecom grade, oven-controlled, crystal oscillators (OCXOs). Note the reconfigurable connections from the frequency synthesizer cards to the front panel via RF patch cables.

2. Hardware Components

The ACFT has three classes of hardware components: Crystal oscillators or OCXOs (two); frequency synthesizer boards (two); and a hybrid signal processor board. All the hardware components of the ACFT are enclosed in a ferrous metal enclosure. The top cover may be easily removed using the snap lock release latch. With the top cover off, all the RF leads are easily accessible for reconfiguration. The top cover is electrically connected to the rest of the chassis and, when attached and locked, aids in shielding the internal components from external electric and magnetic fields. Having the top cover closed during experiments may improve the quality of data.

All components are powered with a single 120 V AC power supply fused to 5 Amps. Internally, the input AC supply is filtered and connected to two independent DC power supplies. One power supply powers the hybrid signal processor board and the single board computer. The second DC power supply provides DC power at 5 V DC and 12 V DC with low ripple noise (Manufacturer specification: $<$10 mV peak-to-peak). This low noise source supplies all the clock elements on the frequency synthesizer boards, as well as, the crystal oscillators.

Additional precautions have been taken to minimize power line noise and interference on the DC supply to clock elements. A common ground plane is shared by all circuit boards and power supplies in ACFT. When connecting the device to external components using the USB interface or the front panel, it is critical to ensure that external components are grounded to a point electrically close to ACFT's ground plane. The shielded leads

connecting the frequency synthesizer board to the front panel are approximately 150 mm in length and may occasionally pick up induced electromagnetic noise. Common mode noise will likely be introduced into the signals presented at the output terminals on the front panel, if care is not taken prior to collecting data. An appropriate ferrite noise clamp may be used if this issue is observed.

Improper grounding may also overload the USB interface. As a safety precaution, the externally facing USB interface is connected to an active USB 3.0 splitter device with overcurrent and overvoltage protection. This device may be reset by disconnecting all power inputs to ACFT, waiting 20 seconds and then repowering. In case of serious overcurrent on the USB interface, the USB splitter device will trip off permanently and will have to be replaced.

2.1 Crystal Oscillator

The ACFT has two discrete Stratum 3E compliant OCXOs, operating at 12.8 MHz. They may be used as a primary frequency reference, or as one of the clock inputs to the frequency synthesizer boards. Both OCXOs are free-running. oven-controlled crystal resonators with statistical properties meeting the ITU-T G.8263/Y.1363 [11] standard for synchronized network equipment. By letting them remain free running during fusion experiments, both drift and stochasticity are unaltered by closed loop control.

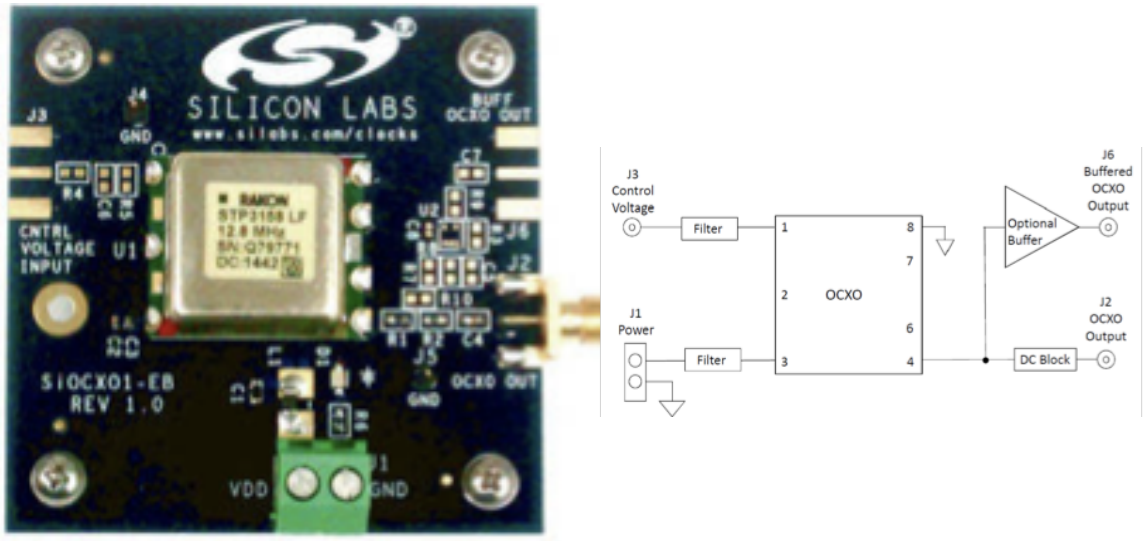
The OCXO used is a Rakon ROX2522S4 [12] has the following performance specifications:

- Variation over 10 seconds: $< 10\ \mu Hz/Hz$
- 24 hour hold over: $< 50\ \mu s$
- Stability over one year: $\pm 75\ nHz/Hz$
- Temperature drift: $0.16 \times 10^{-9}\ Hz/^{\circ}C$
- Oscillator output power: $3\ dBm$
- Harmonic power: $< -35\ dBc$

The OCXO is embedded in an SiOCXO1-EVB board which provides output terminations, as well as, optional control voltage terminations for the OCXO. Figure 4 shows a photograph and functional block diagram for the evaluation board. A resistor-capacitor termination is used at the output on Jumper-2 to match the impedance of the source to the frequency synthesizer board; this termination can be changed as needed. A buffered output is also available to connect directly to test and measurement equipment.

[11] International Telecommunication Union "Timing Characteristics of a Synchronous Ethernet Equipment Slave Clock (EEC)"; G.8263. "Timing characteristics of packet based equipment clocks (PEC) and packet based service clocks (PSC)"; Y.1363

[12] See Footnote 7



(a) Top view of the SiOCXO1-EVB. **(b)** Functional block diagram of the SiOCXO1-EVB.

Fig. 4. Both OCXOs used in the ACFT are embedded in a Silicon Labs SiOCXO1-EVB carrier board. Copyright Silicon Laboratories, Inc.; Used with permission.

2.2 Frequency synthesizer

For most simple fusion experiments, the frequency synthesizer is likely the only component that will need to be configured and controlled. An example of such an experiment would be the development of an algorithm that performs minimum variance, source selection, between multiple jitter filtered inputs. For such an experiment, AFCT users need a circuit with multiple input channels with individually configurable jitter attenuation loops. The algorithm must obtain a stream of phase detector measurements from an input stage DSPLL, and issue control commands to the programmable crosspoint switch. In addition, if the sources have different base frequencies, translation may be necessary using a Digitally Controlled Oscillator (DCO). The frequency synthesizer can perform all of these operations.

The two frequency synthesizer boards can cascade the inputs of one to the other, to expand the permutations when combining clock sources. The frequency synthesizer board is designed around a Silicon Labs Si5348 Jitter Attenuating Clock Generator [13] and contains three independent DSPLLs, each with a DCO. Each frequency synthesizer board has all the necessary magnetics and programming interfaces for the Si5348 to interface with other hardware components, control software and to the analog interfaces on the ACFT front panel. The Si5348 requires both a crystal and a reference input. By default, the 12.8 MHz OCXO on the crystal oscillator board is connected as a reference input, which determines frequency accuracy and stability. A 48 MHz crystal (XTAL) is installed on the frequency

[13] See Footnote 7

Fig. 5. Image of the Si5348-E-EVB development board with an embedded Si5348 Network Synchronizer Chip. This development board in conjunction with a reference time source, OCXO and interfaces to the hybrid signal processing platform comprises the frequency synthesizer. Copyright Silicon Laboratories, Inc.; Used with permission.

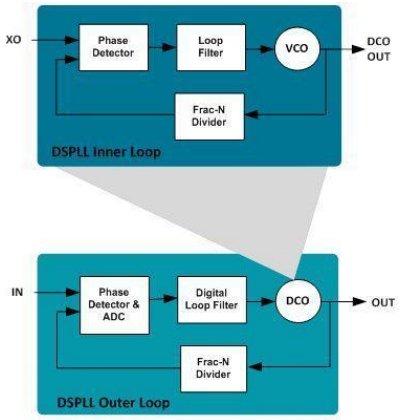


Fig. 6. Functional block diagram of the Silicon Labs DSPLL which integrates phase-locked-loop components, a digitally controlled oscillator and digital signal processing. Copyright Silicon Laboratories, Inc.; Used with permission.

synthesizer board to serve as the crystal input to the chip, this crystal determines the output jitter performance.

The Si5348 is programmable via a serial interface with in-circuit, non-volatile memory so that it always powers up with a known configuration. The frequency synthesizer board has an onboard USB microcontroller which communicates with the Si5348 device through a 4-wire SPI (Serial Peripheral Interface) link. The microcontroller is the SPI host and polls the Si5348 device. USB controllers on both frequency synthesizer boards are connected to a USB hub inside the chassis and can be configured using a single computer running the configuration software.

The frequency synthesizer board has eight SMA connectors (REF/REFB, IN0/IN0B – IN2/IN2B) for receiving external differential clock signals. The REF/REFB differential input clock is connected to the crystal oscillator, which determines the Si5348's wander performance. The four differential clock inputs (IN0/IN0B – IN2/IN2B) are AC-coupled and 50 Ω terminated, as are seven differential outputs (OUT0/OUT0B – OUT6/OUT6B).

The DSPLL's on the Si5348 are operated independently from each other and controlled through a common serial interface. Each DSPLL drives independent DCOs as illustrated in Figure 6. DSPLLs A, C and D (shown in Figure 7) have access to any of the three inputs (IN0 to IN2), as well as, the reference (REF) after having been divided down by the P dividers, which are either fractional or integer. DSPLL D has access to two additional CMOS inputs (IN3 and IN4). Clock selection can be either manual or automatic. Any of the output clocks (OUT0 to OUT6) can be configured to connect to any of the DSPLLs, using a flexible crosspoint connection. DSPLL B realizes the external reference for the system. Each DSPLL module performs input frequency translation, jitter attenuation and wander filtering. The DSPLL loop bandwidth determines the amount of input clock jitter attenuation. DSPLL loop bandwidth can be configured between 0.1 Hz up to 4 kHz. Since the loop bandwidth is controlled digitally, the DSPLLs will always remain stable with less

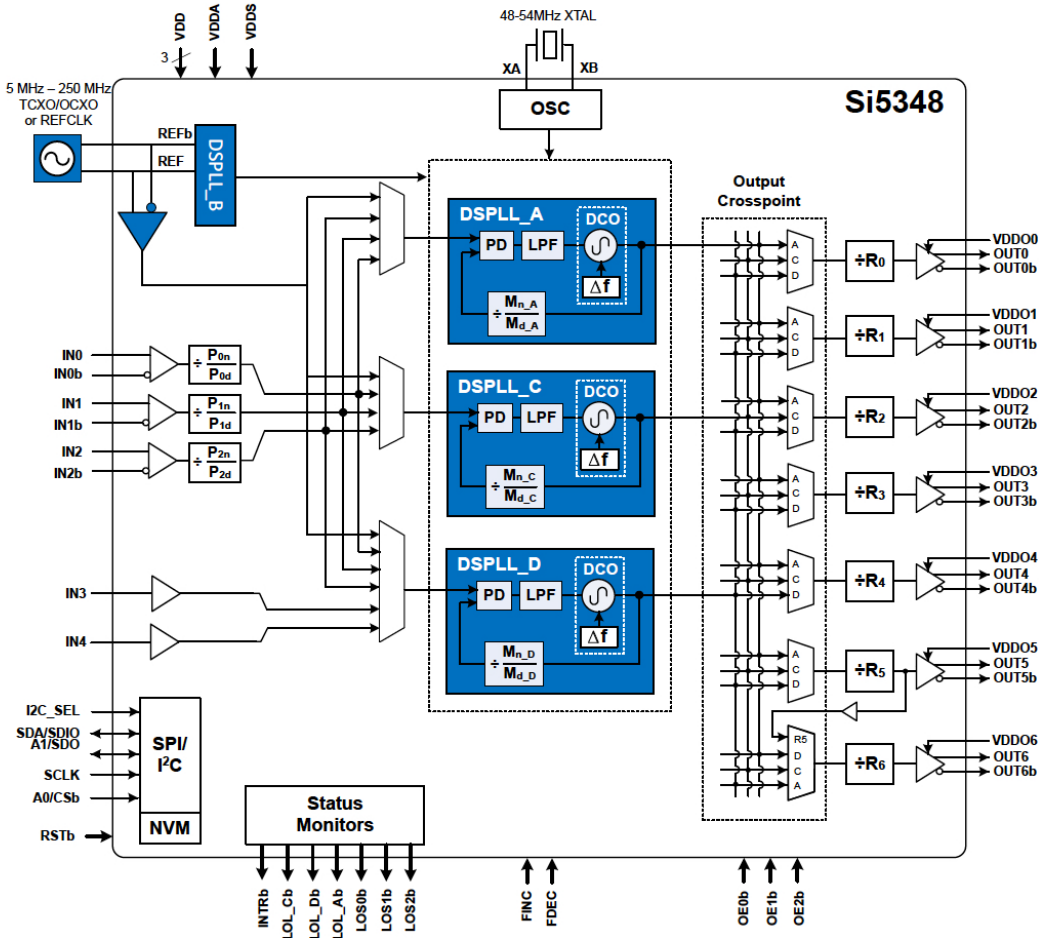


Fig. 7. Functional block diagram of the Silicon Labs Si5348 showing the three independent programmable DSPLLs, input multiplexers, multiplier stages and the output crosspoint. Copyright Silicon Laboratories, Inc.; Used with permission.

than 0.1 dB of peaking, regardless of the loop bandwidth selection.

Fractional input dividers allow the DSPLL to perform hitless switching between input clocks. The combination of fractional input dividers, fractional frequency multiplication and integer output division, allows each DSPLL to lock to any input frequency and generate virtually any output frequency. A crosspoint switch connects any of the DSPLLs to any of the outputs. An additional output stage integer divisor (Denoted $R_{0...6}$ in Figure 6) determines the final output frequency. The frequency configuration for each DSPLL is programmed through the serial interface and stored in non-volatile memory. In the default frequency plan deployed on the ACFT, it was assumed that the industry standard, 10 MHz reference frequency would be used and all dividers set accordingly. OUT6 has been configured to provide one pulse per second. If the input DSPLL generates 1 MHz, this output can be a dedicated test point for experiments that require testing against a reference pulse-per-second signal.

The benefit of this frequency synthesizer setup, besides the PLLs and dividers, is that outputs from one DSPLL can be wired to the input of another, allowing experimenters to cascade jitter and wander mitigation stages. Experimenters evaluating Best Reference Clock algorithms can manually trigger the hitless switching input stage to obtain wander and jitter statistics over a range of test intervals automatically. Since two frequency synthesizer boards are included in the ACFT, the output of one can be connected as the reference to the other, providing the ability to simultaneously compare the performance of several clock inputs against each other.

2.3 Si5348 Features

The following subsections highlight features in the Si5348 that may be of particular interest to the clock fusion experimenter. A comprehensive reference manual for the product may be found here.

2.3.1 Fault Monitoring

Input clocks (IN0, IN1, IN2, IN3, IN4) and the reference input REF/REFB are monitored for loss of signal (LOS) and input clocks (IN0, IN1, IN2) monitored for an Out-Of-Frequency (OOF). The REF/REFB input is used as the "reference monitor" to help determine an OOF on IN0, IN1, or IN2. The reference at the XA/XB pins is also monitored for LOS since it provides a critical reference clock for the DSPLLs. Each of the DSPLLs also has a Loss Of Lock (LOL) indicator, which is asserted when synchronization is lost with their selected input clock.

2.3.2 Hitless Clock Switching with Phase Build Out

Hitless clock switching is a feature that prevents a phase change from propagating to the output. This occurs when switching between two clock inputs that have exactly the same frequency and a fixed phase relationship (i.e. they are frequency locked, but with different

phases). When phase buildout is enabled, the DSPLL absorbs the phase difference between two input clocks during a clock switch. When phase buildout is disabled, the phase difference between the two inputs is propagated to the output, at a rate that is determined by the DSPLL loop bandwidth.

2.3.3 Ramped Input Switching

If switching between input clocks – that are not exactly the same frequency (i.e. are plesiochronous) and ramped clock switching is enabled – the servo controller executes a smooth transition between the two input frequencies. The controller makes this transition by briefly entering holdover, and then exits from holdover after slewing to the target frequency.

2.3.4 Synchronous/Asynchronous Output Selection

Outputs can be switched, either synchronously or asynchronously. In synchronous mode, the output will wait until a clock period has completed before the driver is disabled. This prevents unwanted runt pulses from occurring when disabling an output. A synchronous output selection may be required in cases where the two frequency synthesizer boards are cascaded (note that this is a common operating mode for the ACFT) in order to prevent spurious triggering of the downstream synthesizer. In asynchronous mode, the output clock will disable immediately without waiting for the period to complete.

2.3.5 Output drivers

The Si5348 supports seven differential output drivers. Each driver has a configurable voltage amplitude and common mode voltage, covering a range of differential signal formats including LVPECL, LVDS, HCSL, with CML-compatible amplitudes. In addition to supporting differential signals, any of the outputs can be configured as dual single-ended LVCMOS (3.3 V, 2.5 V, or 1.8 V), providing up to 14 single-ended outputs, or any combination of differential and single-ended outputs.

2.3.6 Digitally Controlled Oscillator (DCO) Mode

The DSPLLs support a DCO mode where the output frequencies are adjustable in predefined frequency step words (FSTEPW). The frequency adjustments are controlled through the serial interface, or by pin control, using frequency increments (FINC) or decrements (FDEC). A FINC will add the frequency step word to the DSPLL output frequency, while a FDEC will decrement it. The DCO mode is particularly useful in applications where clock corrections are inserted by a software-based servo loop running remotely. A common case for this mode of operation is when the clock corrections are computed based on timestamps, as is the case in IEEE 1588 (Precision Time Protocol). Some device tests for

synchrometrology also require generation of time and frequency steps. During development, an external counter/pulse generator was used to apply FINC and FDEC commands via pin control to generate a frequency drift and observe the fault/holdover operation in a test device.

2.4 Hybrid signal processing/ development platform

The frequency synthesizer provides a wide range of options for real time manipulation and control of clock sources, frequency multipliers, phase locking and jitter mitigation. However, many clock fusion experiments may also require the following features:

- A general-purpose ARM-based CPU with a Linux programming environment for data logging, storage, analysis and visualization.
- A general-purpose FPGA fabric for signal processing and deterministic data processing.
- A network interface, display drivers and USB peripheral support.

The ACFT, therefore, includes a signal processing /development platform, based on the Xilinx Zynq XC7Z020-1CLG484C System on a Chip (SoC). The SoC integrates a dual-core ARM Cortex-A9 based processor and Xilinx programmable logic in a single device. This platform also has many features common to modern embedded processing systems, including DDR3 component memory, an Ethernet interface, general purpose I/O, and two UART interfaces. Other features can be supported, using VITA-57 FPGA mezzanine cards (FMC). This document only provides an overview of capabilities derived from the reference manual for the Zynq XC7Z020-1CLG484C Hybrid SoC that can be found here [14]. For an experimenter, the Hybrid signal processing/development platform is a full function, embedded computer running the Linux Operating System (Xilinx PetaLinux). Experimenters only need to connect a monitor and keyboard or SSH over the Ethernet interface to program, visualize or run experiments. The signal processing /development platform operates in an independent clock domain, comprised of:

- A Fixed 200 MHz Low-voltage differential signaling oscillator
- An I2C programmable Low-voltage differential signaling oscillator (156.25 MHz by default)
- A Fixed 33.33 MHz single-ended system oscillator

If signals from the frequency synthesizer will be sampled by the signal processing platform, the signal of choice will be connected to one of two 12-bit, 1 MSPS analog to digital converters (XADC).

Experimenters in most cases will use the XADC interface to measure the output of the frequency synthesizer, digitize the frequency and log the data. A programmable, low-jitter

[14] See Footnote 7

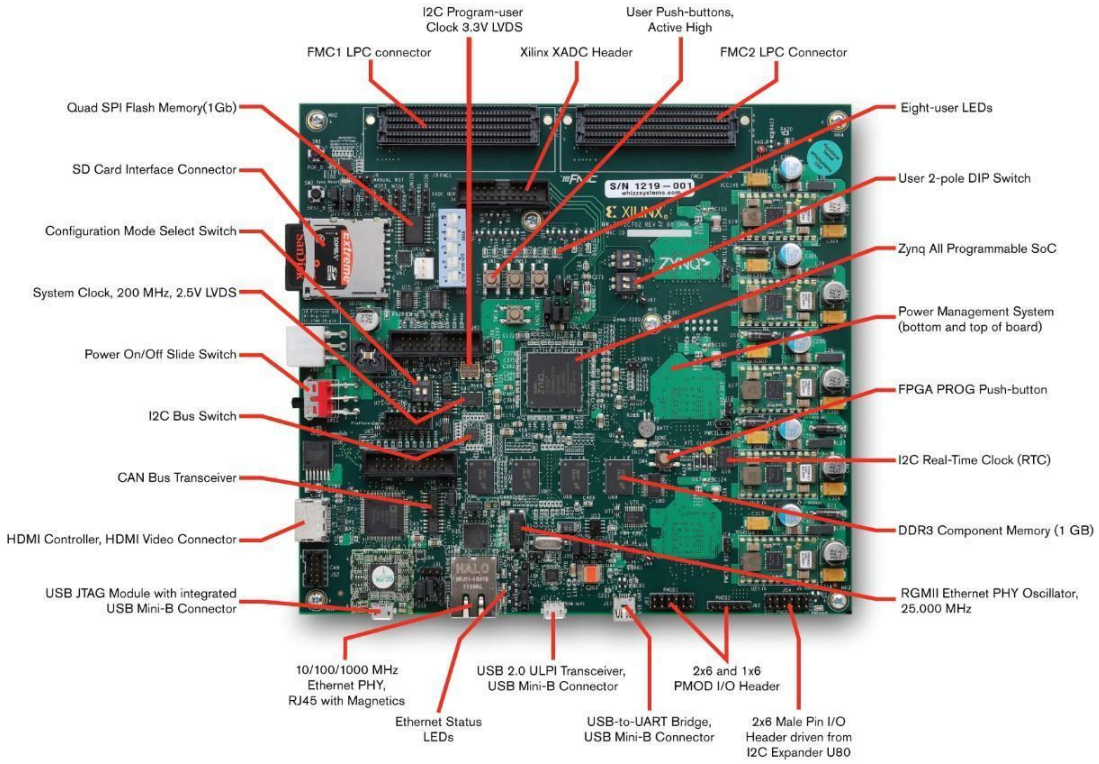


Fig. 8. Layout of the hybrid signal processing/ development platform. Material based on or adapted from figures and text owned by Xilinx, Inc., courtesy of Xilinx, Inc. © Copyright Xilinx 2020.

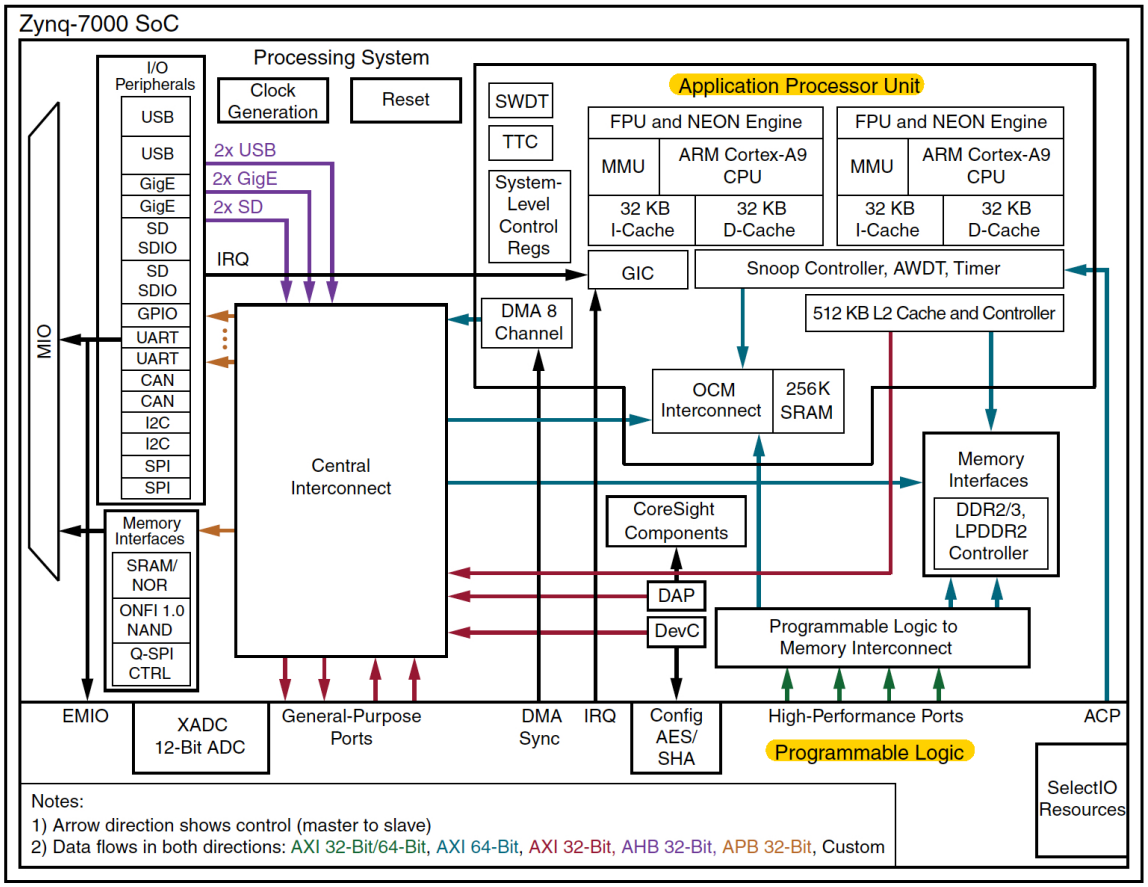


Fig. 9. Schematic diagram of the Zynq XC7Z020-1CLG484C Hybrid SoC. Material based on or adapted from figures and text owned by Xilinx, Inc., courtesy of Xilinx, Inc. © Copyright Xilinx 2020.

3.3 V differential oscillator drives the Analog-to-Digital Converter. On powering up, the clock defaults to an output frequency of 156.250 MHz. User applications can change the output frequency within the range of 10 MHz to 810 MHz, through an I2C interface. Silicon Labs Si570BAB0000544DG serves this purpose. Experimenters may set this frequency, as required, to efficiently sample the output of the frequency synthesizer board. Alternatively, a fixed 200 MHz oscillator may be selected as the FPGA clock. The CPU clock source is a 1.8 V single-ended, fixed 33.3 MHz oscillator. A 25.00 MHz crystal is the clock source for the Ethernet PHY.

2.4.1 Analog-to-Digital Converter

The XC7Z020 SoC provides an Analog Front End, Analog-to-Digital converter with associated firmware called the 'XADC block'. The XADC block includes a dual 12-bit, 1 MSPS, Analog-to-Digital Convertor (ADC) and on-chip sensors for voltage and tempera-

ture measurements and 17 external analog channels.

The CPU has a dedicated interface to the ADC, via the XADC device driver, which enables a reliable monitoring of the ADC buffer, directly from the Linux programming environment. The communication between the CPU and the XADC happens over a serial interface. The command from the CPU is stored in a command buffer and then serialized using a parallel-to-serial converter. The incoming serial data from the XADC is converted to parallel data, using a serial-to-parallel converter. The experimenter can then easily interact with the XADC through the L2 cache memory, without the need to implement code to read and write serial data. This architecture is purported to facilitate better determinism in code execution time if the experimenter requires tight control on CPU execution latency.

Advanced experiments where the experimenters may require the use of custom FPGA hardware, or mixed-signal processing, may be beyond the capabilities of the XADC to CPU pipeline. In this case, it may be necessary to use a purpose-built FPGA mezzanine card. Two VITA 57.1 FPGA Mezzanine Card (FMC) interfaces are provided, using Low Pin Count (LPC) connectors.

As noted earlier, the Zynq XC7Z020-1CLG484C System on a Chip (SoC) integrates a dual-core ARM Cortex-A9 based processor and Xilinx programmable logic in a single device. This hybrid architecture allows experimenters to develop, compile and run signal processing algorithms on the CPU, but may also use the FPGA fabric where appropriate. In the context of the ACFT, experiments may opt to use the FPGA for deterministic control of the frequency synthesizer board or to trigger switching or fault detection features.

In tests conducted during the development of the ACFT, FPGA designs were automatically generated and synthesized from code developed in MATLAB and Simulink, using the Mathworks' embedded coder and HDL coder tool suites [15]. This tool suite is able to automatically generate software components for both the CPU and the FPGA in the hybrid SoC. The tool suite is described in more detail in Section 3.

3. Software Components

The ACFT's interconnected hardware components have unique software needs and tools. Many software tools, needed to use the system, were introduced with their respective hardware components in Section 2. This section will briefly describe each software tool, mentioned, and provide relevant manufacturer references.

3.1 Clock Builder Pro

ClockBuilder Pro is a software application for clock tree design, configuration of the Si5348 clock generator and jitter attenuator. This application provides a graphical configuration tool and design dashboard for DSPLLs, multipliers and DCOs. The software

[15] Commercial entities, equipment, or materials identified in this document in order to describe an experimental procedure or concept do not imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

(See Footnote 15)	**Frequency synthesizer**	**Signal processing/ development platform**
Configure	ClockBuilder Pro/ Serial terminal	Vivado Design Suite
Program	ClockBuilder Pro/ JTAG Field programmer	CLANG C/C++ compiler, Mathworks Embedded & HDL Coder, Python
Operate	ClockBuilder Pro/ Serial interface	Xilinx PetaLinux
Monitor	ClockBuilder Pro/ Serial terminal	XAPP Web Server, Matplotlib

Table 1. Software components needed to configure, program, operate and monitor the two major hardware components in the ACFT.

tool also allows configuration of crosspoints and can be used to directly write or update the non-volatile configuration memory on the frequency synthesizer board. Once installed on a Windows PC, the software automatically connects and identifies the frequency synthesizer board, and all installed clock generation and jitter mitigation components.

A verified configuration for the Si5348 is called a frequency plan. A frequency plan can include one clock generation configuration or a configuration for multiple DSPLLs, multipliers and crosspoints. The plan can be edited and verified (against component specifications) prior to writing the plan to the Si5348.

The ACFT's default plan locks DSPLL A to the reference clock source. This plan allows experimenters to benchmark the nominal jitter performance of the system at the configured output. The default plan also sets the DCO to output a 10MHz nominal frequency. This facilitates easy measurement of drift and wander, between the device under test, and the OCXO included in the ACFT. The frequency plan also configures OUT6 as a pulse-per-second output to assist in measurements against UTC traceable pulse-per-second signals.

3.2 Operating System and software development tools for Zynq SoC

As noted in Section 2, the signal processing/ development platform is based on a Xilinx Zynq XC7Z020 System on a Chip (SoC). This system uses a Linux based development workflow, allowing experimenters to develop applications, using high level programming tools, compatible with the embedded Xilinx PetaLinux distribution. The embedded operating system includes tools to build, develop, test and deploy embedded Linux applications; this includes support packages for peripheral hardware connected on the board and host development tools. The operating system image is stored on a removable SD card, which is mounted as an automatic, bootable drive, when the board is powered on. Boot options for the instance of PetaLinux allow selection of either a command line interface or a graphical

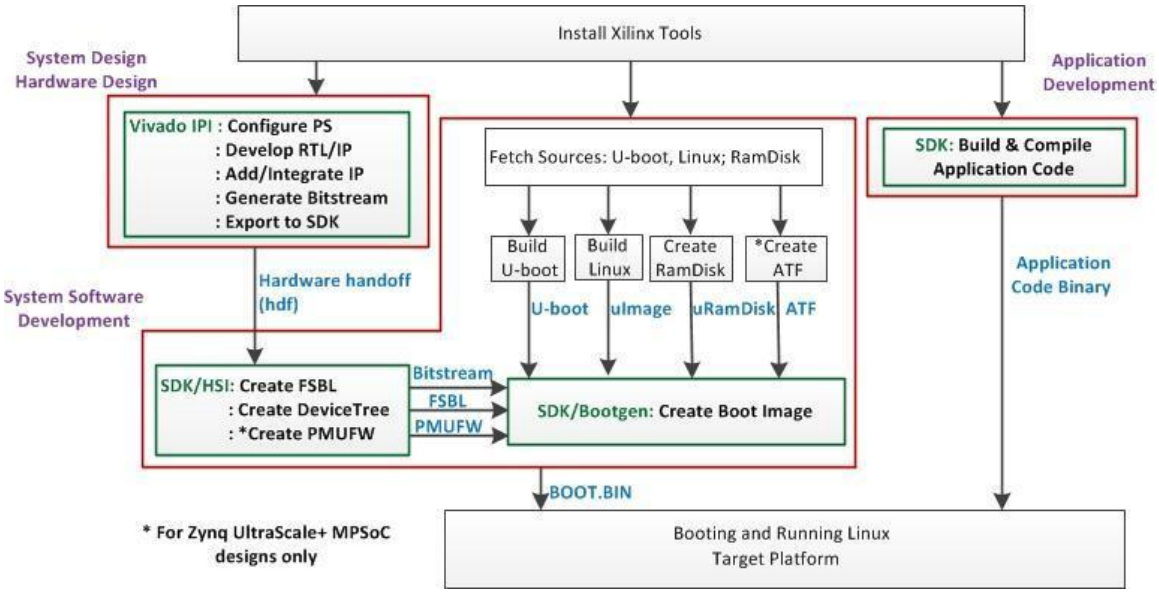


Fig. 10. The schematic shows three design workflows for designing embedded applications on the signal processing/ development platform. Material based on or adapted from figures and text owned by Xilinx, Inc., courtesy of Xilinx, Inc. © Copyright Xilinx 2020.

interface for the operating system.

Figure 10 illustrates three workflows for developing embedded applications for the hybrid SoC. In all cases, the experimenter first installs Xilinx tools directly on the SoC. The application development tool suite and support packages for peripheral hardware can be downloaded here.

The third workflow in the schematic has the fewest steps. It allows experimenters to develop Linux applications, using the Linux Industrial Input/Output (IIO) drivers, included in the Xilinx Software Development Kit (SDK). This workflow is ideally suited for rapid prototyping, since it does not involve FPGA hardware design or implementation of a custom Hardware Software Interface (HSI).

NIST used this workflow in four of its tests on the signal processing/ development platform, and to acquire and digitize signals from the Frequency Synthesizer. This capability (as described in Section 1) is an essential feature for the ACFT. It allows an experimenter to perform a clock fusion experiment on the Frequency Synthesizer board, and uses the signal processing platform to monitor, record, analyze and visualize signals of interest.

Figure 11 illustrates, via an interaction diagram, the design of an application that uses XADC drivers in the Xilinx SDK to acquire digitized signals from the analog-to-digital converter. These drivers use the IIO framework to provide a standard interface with experimenter's applications to configure and collect data from the XADC block.

Per this framework, the analog-to-digital converter is registered as an 'IIO device'. An experimenter's application can interact with this device, using two methods as illustrated by the two paths between "IIO Framework Subsystem" and the "Linux Virtual File system"

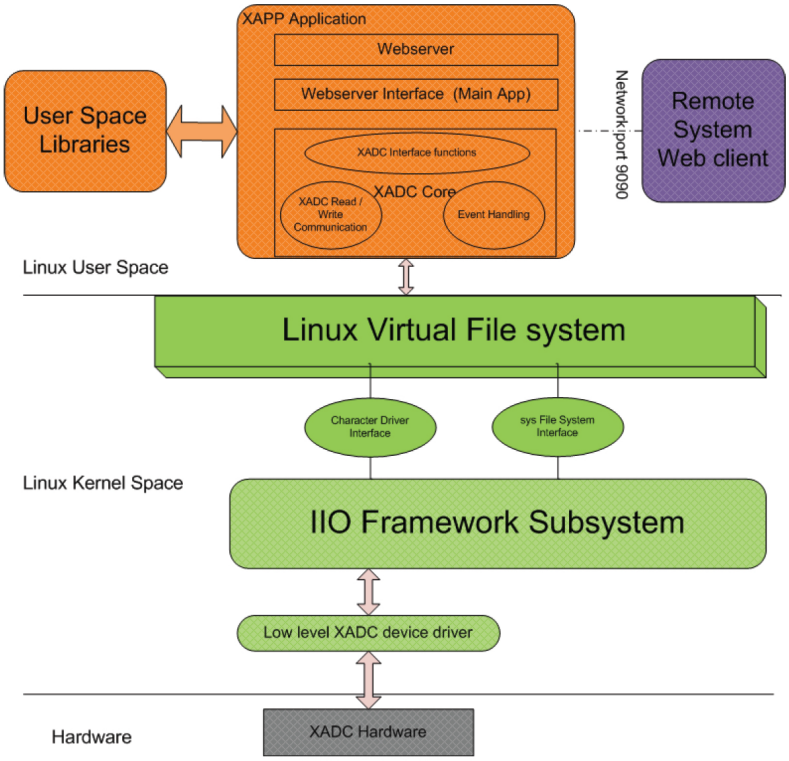


Fig. 11. An interaction diagram showing the software infrastructure on the ZYNQ SoC based signal processing/ development platform. The diagram isolates the software interactions required to implement the analog-to-digital converter as a web service. Material based on or adapted from figures and text owned by Xilinx, Inc., courtesy of Xilinx, Inc. © Copyright Xilinx 2020.

in the schematic.

- The first method is a file system interface (called the "sysfs" interface) that streams data channels from the analog-to-digital converter to a memory location in the file system.
- The second method is a character device interface, used for buffered data transfer, hardware instructions and for event information retrieval. The subsystem, called the "XADC core", comprises user APIs and libraries used to read/write data from the buffer and handle events.

In the most common use case expected for the development platform, the final output of the system is a visualization of data. To produce a visualization, the XADC core exposes an API for a web interface. The Xilinx SDK includes a webserver application that receives and passes values to the core. By default, the webserver can be accessed on http port 9090, allowing the experimenter to easily develop a dashboard or visualization, using HTML, Javascript and TCP-HTTP protocol handlers.

In manufacturer testing of the XADC core pipeline, the maximum external signal bandwidth has been measured and found to be 100 kHz. Note that this rate is 1/10th the specification of the ADC (1MSPS). Software latencies in the operating system and in data serialization are the primary contributors for this reduction in effective bandwidth. In testing, NIST found that this data rate was adequate for web-based visualization and to monitor the progression of a clock fusion experiment. However, in cases where precise metrology of frequency and phase were needed, we used an external frequency counter connected directly to the frequency synthesizer which in turn logged data to USB accessible memory.

A detailed description of the XADC core and an example of a web-based visualization program are available here.

3.3 The Vivado Design suite

The Vivado Design Suite is a software tool for designing and simulating the custom hardware designs for Xilinx FPGAs. As described in Section 2, the Zynq SoC integrates a dual-core ARM Cortex-A9 based processor and Xilinx FPGA programmable logic in a single device.

While most applications run effectively as a Linux application on the processor, there may be significant performance improvements to running some signal processing tools on the FPGA hardware. Hardware implementation may be a requirement for applications where deterministic execution time is critical. Lastly, it is common practice for commercial implementations of frequency synthesis and jitter mitigation algorithms to be implemented on an FPGA. In such cases, hardware designs, provided by third party vendors, may be directly applied to bare metal programmable logic on the SoC; this provides the capability for 'black-box' testing of hardware designs within the ACFT.

In testing, a cascaded, integrator-comb filter was implemented on the programmable logic to mitigate sampling artifacts from the ADC. Running the filter on the FPGA allowed

NIST to apply multiple comb and integrator stages, while still acquiring data at 1MSPS. NIST has not experimented with third party hardware designs, as of this report's publication.

NIST's implementation utilized the Mathworks HDL (Hardware Description Language) Coder package to generate synthesizable VHDL (Very High Speed Integrated Circuit Hardware Description Language) netlist directly from HDL-ready Simulink and MATLAB function blocks. Simulink function blocks were matched with the available port types on the SoC. Also, the 'Xilinx System Generator' subsystem was added and configured in the Simulink model. The Vivado Design Suite was then used to process the VHDL design and optimize placement and routing on the FPGA. The Design Suite also includes a graphical user interface to optimize a design from the perspective of timing, congestion, total wire length, utilization and power consumption.

4. Typical Use Cases for AFCT

The typical setup for ACFT is illustrated in Figure 12. As discussed in the design motivation for the system, ACFT offers researchers the flexibility to address the niche challenge of taking existing time and frequency sources, combining them programmatically, and then measuring the result on standard measurement equipment. The setup shown in Figure 12 shows three sources of frequency and phase information connected as inputs to ACFT (right side of Figure 12). Time of day information or other timing-related metadata, such as clock offset information or Stratum information about the sources are read by the ACFT, via the General Purpose Interface Bus (GPIB).

The ACFT can use one of the three sources for 'reference time', i.e., the phase reference used to compute offset and statistical variation in the clock sources being tested. Alternate sources of reference time include, a network time server, oven controlled crystal oscillator or a GPS disciplined clock installed as an FMC daughter card. The ACFT's digitally controlled oscillator, as well as, PLLs and jitter attenuator circuits denoted collectively by the 'programmable oscillator' symbol in the bottom of the figure.

This use case may be implemented using either of the two means of operation mentioned earlier. The first uses the hybrid signal processing board to read phase offsets between the input sources and the reference timing source. A filtered set of phase corrections are then transmitted to steer the programmable oscillator. The second operating mode uses the frequency synthesizer board directly. Here, the three inputs are applied to three DSPLLs. The experimenter can read phase measurements over a console interface and execute a clock fusion plan using the crossover feature.

Lastly, the icons on the left show standard measurement outputs. These outputs are exemplified by reports generated on the hybrid signal processing board. The reports are based on experimenter implemented, software-based, filter code or analog measurement outputs on the front panel, that can be connected to oscilloscopes or frequency counters. In order to reliably use the report generation function of the ACFT, calibration and validation of the internal metrology hardware must be performed prior to each fusion experiment.

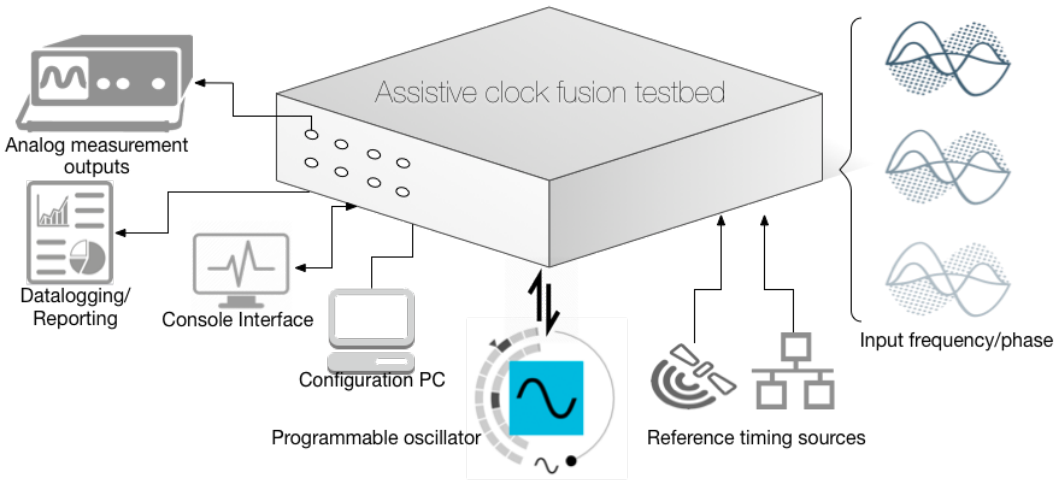


Fig. 12. The schematic diagram shows the typical use case for ACFT, serving as an evaluation and measurement platform between multiple clock sources, shown on the right, to standard measurement tools, on the left.

In most cases, custom report generation software will need to be developed for the experiment. With a dual core processor architecture, report generation and logging functions can be delegated to the core running non-real-time jobs to minimize the impact on real-time signal processing. Some general assessments of the internal clock hardware are included in this report in the following sections to provide the experimenter expectations of system uncertainty.

ACFT is a general platform for clock fusion research. As outlined in Sections 2 and 3, the capabilities offered by the hardware and software are extensive, and support a wide range of research activities in the field of clock synthesis and resilience. The following subsections present three illustrative use cases intended to help experimenters identify potential applications for the ACFT in their own research. These use cases are:

1. Reference clock switching
2. Phase and frequency recovery from Ethernet time transfer
3. Time scale generation

These cases are based on feedback received from experimenters, designing timing and synchronization solutions for the electric power industry.

The electrical grid is becoming ever more dependent on telecom networks to transmit data, monitor power flows and sensors, and control substation and grid edge equipment in real time, as outlined in the ITU report ITU-R SM.2351-2 [16]. These networks interact with physical components like power conditioners, stabilizers, and generators. It is critical to ensure reliable phase and frequency transfer between these components to ensure deterministic control outcomes.

The simultaneous expansion of network distribution of time and the increasing dependence on synchronization create some concerns for system operators. In no particular order, these concerns include:

- The need to seamlessly switch between reference clocks if one were compromised or lost.
- The need to recover time and frequency from different, diversely routed timing networks.
- The need to 'combine' multiple reference clocks to generate a robust time scale.

4.1 Use case – reference clock switching

A typical synchronized timing configuration may require synchronization clients to switch between reference clocks in a seamless manner (phase buildup, as opposed to a phase jump). The logic for when the synchronization client might need to switch reference clocks may be a topic of interest for experimental work.

[16] ITU-R SM.2351-2. Smart Grid Utility Management Systems, June 2017.

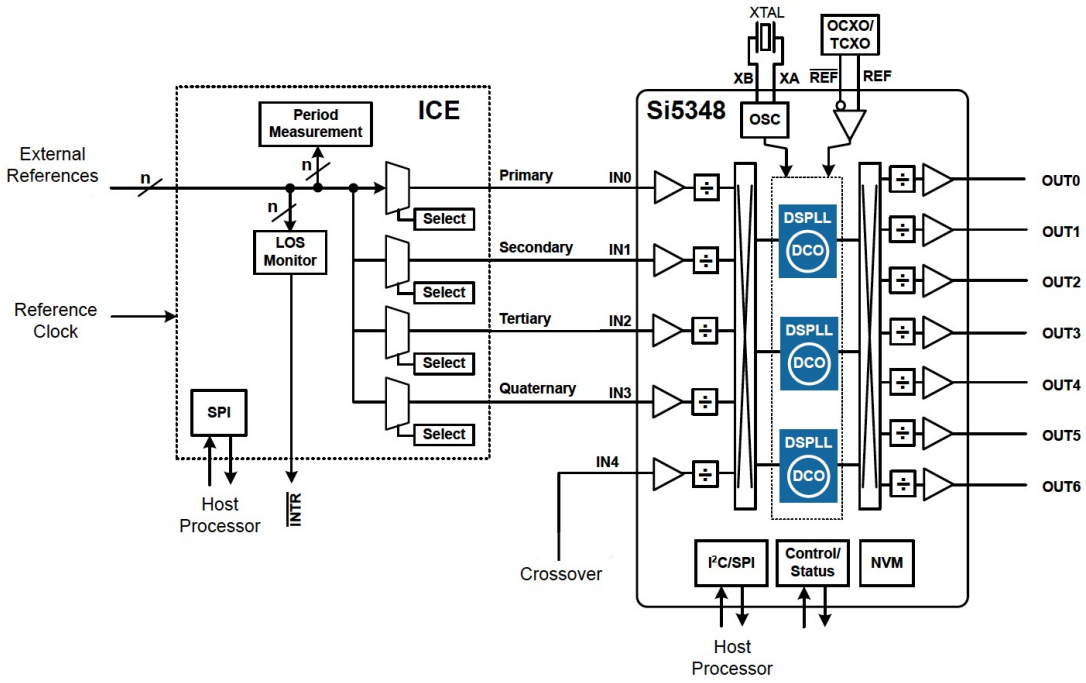


Fig. 13. A schematic showing a configuration where the host processor can switch between thirty-two (or more) input clock sources. Copyright Silicon Laboratories, Inc.; Used with permission.

In some domains, reference clock switching is referred to as Best Reference Clock Algorithm. Now, consider how an advanced version of this switching algorithm between 'n' reference clocks may be implemented, using the ACFT. Image 14 shows a configuration in which four inputs of the Si5348 were used to build up a phase, with switching between primary, secondary, tertiary and quaternary inputs. This mechanism is discussed in Section 2.

The added capability for a larger number of clock inputs is the addition of an FPGA block, called an Input Clock Expander (ICE), that provides 'glitchless' switching between multiple sources. The ICE can be configured to accept *n* clock inputs, and generates four clock outputs that the Si5348 uses to determine the final source of synchronization.

The Best Reference Clock selection is accomplished in two stages. The first stage, implemented in the ICE, allows a host processor to select any one of n inputs as primary, secondary, tertiary, and quaternary inputs to the Si5348. The ICE block uses both loss-of-signal (LOS) and period measurements to determine the best inputs to switch between the available external references. A glitchless clock switch includes a synchronizer circuit to avoid meta-stability, caused by asynchronous reference inputs. Note that apparently asynchronous clock sources observed within the switching interval are relatively common. This could be an artifact of the 'change in input,' triggered by the host processor, especially when the two sources have slightly different frequencies. It is of course also possible that two reference sources are totally unrelated to each other.

A synchronizer circuit is comprised of two stages of flip flops. The first stage helps stabilize data by latching it. This stage also later registers the select signal at the negative edge of the clock to ensure that no changes occur at the output while either of the clocks is at high level, thus protecting against chopping the output clock. An interlock between the selection of the first reference input and the second reference input forces the switch to wait for de-selection of one reference source, before starting the propagation of the selected reference source, thus avoiding glitches.

While there are several implementations of glitchless switching multiplexers, the FPGA hardware design, used in testing, can be downloaded here.

NIST was able to port this design to the Zynq XC7Z020-1CLG484C System on a Chip (SoC) in the hybrid signal processing/ development platform, and, thus, use the host processor to select reference inputs.

Input selection is done through the serial interface, between the ICE block and host processor. There is an input selection register for each of the inputs. Since the glitchless switch can take some time to execute, the status of the switch can be monitored by reading the associated busy bit. '1' indicates that the switch is in progress. '0' indicates the switch is complete.

When complete, the host knows that the input selection register reflects the selected output clock. The time to complete a glitchless switch depends on the input clock mode. Switching between standard inputs, clocks will take $\approx 575\ \mu s$ for the busy bit to clear. Switching between 1 PPS input clocks will take $\approx 4.5\ s$ for the busy bit to clear.

It is important to note that a glitchless switch still induces a phase jump in the output signal. The term "glitchless" is common parlance for clock distribution schemes used for digital logic, but is different from the phase buildup, or slew that is required for jitter free transition from one reference clock to another.

The Si5348 provides the second stage of Best Reference Clock selection process. It does this by using its own LOS and Out-Of-Frequency (OOF) monitors and configurable priorities to automatically select the input used for synchronization. The Si5348 also provides phase matched switching between the four inputs to the second stage. As illustrated in Figure 13, the Si5348 has a bank of PLLs to lock in on any of the four input references from the ICE. All PLLs provide jitter mitigation capability. The Si5348 also has a DCO associated with each PLL in the frequency synthesis stage, and can be used to 'build up' phase at the desired rate.

4.2 Use case – phase and frequency recovery from Ethernet time transfer

Another use case for the ACFT lies in the evaluation and development of clock recovery from network-based. time protocols. Experimenters may evaluate frequency recovery from a Synchronous Ethernet (SyncE) carrier, while also recovering phase and time of day information from an IEEE1588v2 Precision Time Protocol (PTP) stack [17].

[17] IEEE Standard 1588-2019 (Revision of IEEE Std 1588-2008)

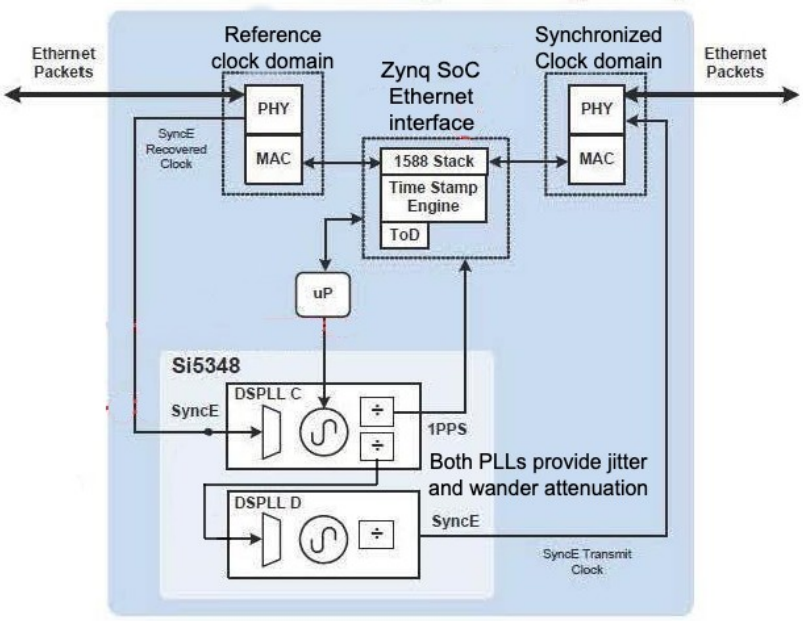


Fig. 14. Schematic diagram showing a clock recovery architecture for SyncE and IEEE 1588 based time transfer. Copyright Silicon Laboratories, Inc.; Used with permission.

This use case considers a scenario where the purpose of recovering frequency and phase information is to create a boundary clock source. This is a common implementation choice. Individual power substations may exist within their own clock boundaries, while using an Ethernet time transfer mechanism to share reference time to other substations. The intention being that components within the substation would continue to be coherent, even if the external link to a time reference is lost. In this scenario, experimenters may want to consider several tradeoffs in designing a boundary clock service that effectively integrates diversely routed Ethernet-based, timing signals.

The ACFT is uniquely suited to evaluate the performance tradeoffs and challenges inherent to this use case. Figure 14 illustrates a clock boundary, between a reference clock domain on the left, and a synchronized clock domain on the right.

First, frequency information contained in the SyncE carrier[18], on the left, must be recovered and transferred to the synchronized clock domain. In order to meet the tight jitter tolerances of ITU-T G.8262 standards, the DSPLL C on the Si5348 is used to lock to the SyncE carrier. Setting DSPLL C to a 100 mHz loop bandwidth and 100 Hz fast lock bandwidth output provide a jitter-free PLL output that can be divided as needed. In this case, one output is a pulse per second signal and the other is an unaltered 25 MHz frequency.

Second, phase and time of day information encoded in IEEE1588v2 Precision Time Protocol (PTP) must be recovered from the reference clock domain and retransmitted to the synchronized clock domain. Since PTP phase information is typically traceable to

[18]The SyncE standard permits carrier frequencies of 25MHz, 125MHz and 156.25MHz. A 25 MHz carrier was used in the described configuration.

UTC, the boundary clock may choose to align the phase of the downstream SyncE carrier to the phase recovered from PTP.

The SoC used in the ACFT has firmware for detecting PTP messages, pulse-per-second counters, nanoseconds-elapsed counters, and time of day registers to support hardware assisted, time stamping and phase recovery. An application note describing the PTP implementation can be found here.

The IEEE 1588 Stack outputs timestamps for the phase of the reference signal at source. In our experiment, DSPLL C was tuned with a loop bandwidth of 1 mHz and locked to the SyncE reference. The output was then divided down to 1Hz and used to generate IEEE 1588 timestamps. These timestamps were parsed by the host processor and used as a correction signal to advance, or retard the DSPLL C output phase, using its DCO feature.

The DCO has a resolution of 1 μHz/MHz. Since the control actuator operates on the derivative of phase, the servo loop bandwidth was set at 0.05 Hz – 0.1 Hz to ensure that the output of DSPLL C was a phase, corrected version of the reference SyncE carrier. The phase corrected SyncE carrier is then passed through DSPLL D, set to the jitter and wander mitigation parameters required to meet ITU-T G.8262. The output of DSPLL D is then used as the transmit carrier frequency for the synchronized clock domain.

4.3 Use case – Time scale generation

As a future research avenue, experimenters may intend to implement a time scale, using the ACFT to produce fractional frequency variation at the output that is lower than any of the input clocks. There are several strategies already proposed to generate a time scale from an ensemble of clocks. The AT1 algorithm, for example, has been used for many years at NIST to compute the weighted average of time differences acquired from an ensemble of cesium standards and hydrogen masers.

The data input to the algorithm is a series of time-difference measurements, between the reference clock and the other devices in the ensemble. An ideal time scale algorithm would generate time and frequency with more reliability, stability, and frequency accuracy than any one of the individual clocks in the ensemble. A time scale algorithm calculates the time offset of each clock, from ensemble time at a given reference time. Ensemble time – the time of the scale – is realized by applying the appropriate offset to the time of any one clock. The AT1 algorithm estimates the time and frequency offsets of each clock from ensemble time (and frequency).

An estimate of frequency drift for each clock can be entered and used for time prediction. The weight of each clock is determined by its prediction confidence, the normalized reciprocal of the squared prediction error. AT1 produces a time scale with a three-step calculation for each clock: time update, prediction confidence update, and the frequency update [19].

[19] Weiss, M. and Weissert, T., 1991. AT2, a new time scale algorithm: AT1 plus frequency variance. Metrologia.

Implementing the time scale algorithm on the ACFT requires the use of both the frequency synthesizer and the hybrid development/signal processing platform. The first step of the process is to install multiple clocks that participate in the ensemble. Each frequency synthesizer board can incorporate up to four clocks and perform hitless crossover between them. The ACFT can handle eight clocks with the two boards installed. If the ICE block shown in Figure 13 is used, 64 clocks can be connected at one time. The following sequence of updates must occur.

1. *Required time update:* This functionality was discussed in Section 4.2 where timestamps were used to update the phase of the DCO. Each DCO can be independently updated in phase with very fine resolution.

2. *Required prediction confidence update:* The TA1 time scale uses the TDEV metric to determine prediction confidence. Calculating TDEV requires accurate measurements of time interval and time difference between all ensemble clocks. In our trials, NIST was not able to implement this feature on the programmable logic of the SoC on the development platform. However, the clock specifications of the FPGA in the SoC seem adequate to serve as a time interval counter for all clocks in the ensemble, if an experimenter chooses. A standalone time interval counter was used during testing and transferred batches of time interval data from the time interval counter to the processor using, a Python device control library. TDEV was computed from the time interval measurements, using a numerical processing package for Python. We also measured the temperature on each of the clocks in the ensemble, using the ADC on the development platform, via the XADC block.

3. *Required frequency update:* This is based on a clock model for each participant in the ensemble. NIST implemented the AT1 algorithm, including a stateful clock model for each clock in Python. This was fairly straightforward, since the development platform runs Linux, and the processor was powerful enough to run an interactive session of Python. The frequency update commands were transmitted from the development platform, to the frequency synthesizer board over a serial interface. It was possible to mount the frequency synthesizer's USB interface as a "ttyUSBN" device in Linux to facilitate this. The update/ correction command was interpreted by the Si5348 on the frequency synthesizer, resulting in the appropriate servo stepping of the relevant DCOs.

Acknowledgments

Thomas Linn provided extensive technical writing and editorial support and was instrumental in getting this manuscript organized and edited.

The motivation for the ACFT was the direct outcome of extensive interaction with industry on the part of Ya-Shian Li-Baboud.

Early measurement data for the emerging field of network clock synchronization were obtained from test infrastructure developed by Julien Amelot.

The development of early prototypes of the ACFT was supported by the Software and Systems Division in the Information Technology Laboratory. In particular, Kevin Brady and John Messina were critical to continued development support.

Désiré Banse developed software, benchmarked the SoC and evaluated of the XADC pipeline and support several early tests of the compiler and synthesis tools.

Dr. Marc Weiss and Dr. Kishan Shenoi were instrumental in determining appropriate hardware specifications and they provided insightful design guidance throughout the design and test phase.