BNL-229597-2026-TECH

CA/AP/737

A multigigabit link layer protocol for single picosecond latency determinism using AMD ultrascale+ GTH and GTY transreceivers

P. Bachek

March 2026

Collider Accelerator Department

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC), Nuclear Physics (NP)

DISCLAIMER

A MULTIGIGABIT LINK LAYER PROTOCOL FOR SINGLE PICOSECOND LATENCY DETERMINISM USING AMD ULTRASCALE+ GTH AND GTY TRANSCEIVERS*

P. Bachek†
C-AD, Brookhaven National Laboratory, Upton, NY, USA

Abstract

Precision timing distribution systems require deterministic and repeatable high speed serial data link latency. The EIC Timing Data Link will employ a specialized link layer protocol for deterministic multigigabit communication using AMD Ultrascale+ GTH and GTY transceivers. Link latency must also be measurable to accurately compensate in real time for latency variations of the physical medium. The point-to-point link is full duplex and fully synchronous with a latency control algorithm to align the internal clocks of each system with picosecond resolution. Deterministic clock domain crossing between systems is ensured using static timing analysis of the elastic buffer control signals.

INTRODUCTION

Exchanging data between two systems with deterministic latency is critical for precision timing distribution. Ideally each system within the timing distribution network would operate using a synchronous phase aligned system clock. Data should be sent and received with a precisely known latency between each physically separated system.

High speed serial encoded data transmission is used to transfer data between link partners on the network. Modern multigigabit transceivers present several inherent challenges to achieving deterministic latency. Specifically, latency that is repeatable across system resets requires careful consideration of each clock domain within and between systems.

It is also desirable to compensate for latency variations arising from environmental fluctuations of the physical transmission medium. The components necessary to achieve precise timing synchronization are enumerated herein using AMD Ultrascale+ GTH and GTY transceivers.

EIC Timing Data Link

The EIC timing system requires a 100 MHz accelerator master clock to be distributed to each timing endpoint. This is implemented using EIC Common Platform hardware based on AMD Ultrascale+ FPGAs equipped with GTH and GTY multigigabit transceivers (MGT). The MGTs operate full duplex at 8 Gbps to transfer 8b10b encoded 64-bit data words synchronous with the 100 MHz system clock [1]. The upstream system will transmit the serialized data words framed by the 100 MHz system clock. The downstream system will use the MGT Clock and Data Recovery (CDR) unit to recover the line rate clock and extract the serial data frame clock, with proper phase alignment, for subsequent use as the 100 MHz system clock.

PHASE ALIGNMENT

Transmitter Frame Clock Alignment

On the upstream system the 100 MHz accelerator master clock is routed to the dedicated MGTREFCLK FPGA input pins. This allows the clock to be used as the QPLL reference clock as well as being forwarded to the FPGA clock fabric, see UG576 [2] Figure 3-29 for GTH and UG578 [3] Figure 3-30 for GTY. The QPLL output clock is 8 GHz which is input to the TX Phase Interpolator (TXPI) in the MGT channel which outputs PISO Serial Clock. PISO Serial Clock is divided down by a factor of 80 to generate PISO Parallel Clock at 100 MHz. The phase of PISO Parallel Clock controls the framing of serial data being transmitted. With no phase shift from the TXPI, PISO Parallel Clock assumes one of 80 possible discrete phases with respect to MGTREFCLK, each separated by 125 ps due to the clock divider. The phase of the clock divider generating PISO Parallel Clock changes after every reset which introduces nondeterminism to the phase of serialized data frames with respect to MGTREFCLK. A simplified diagram of the MGT transmitter clocks relevant to this application is shown in Figure 1.

To achieve deterministic latency, PISO Parallel Clock is manually phase aligned with MGTREFCLK so that the framing of the serial data being transmitted is always in phase with MGTREFCLK. The TXPI can be manually controlled using the DRP port to affect an arbitrary phase shift of PISO Seral Clock and subsequently PISO Parallel Clock with a resolution of $\frac{1}{128}$ Unit Interval (UI) [4]. Shifting by 128 steps causes a precise phase shift of one 125 ps UI which can be repeated as necessary until phase alignment is achieved with MGTREFCLK. Using this mechanism will allow for manual phase control, but the phase between PISO Parallel Clock and MGTREFCLK must still be measured with enough precision to verify that they are aligned to within one 125 ps UI.

Measurement of the clock phase can be accomplished by exploiting the Buffer Bypass Auto Alignment algorithm which includes a phase alignment circuit inside the MGT channel see UG576 Figure 3-17 for GTH and UG578 Figure 3-17 for GTY. The feature is used for bypassing the MGT elastic buffer by measuring the phase between PISO Parallel Clock and TX XCLK then delaying TXOUTCLK

* Work supported by Brookhaven Science Associates, LLC under Contract No. DE-SC0012704 with the U.S. Department of Energy
† pbachek@bnl.gov

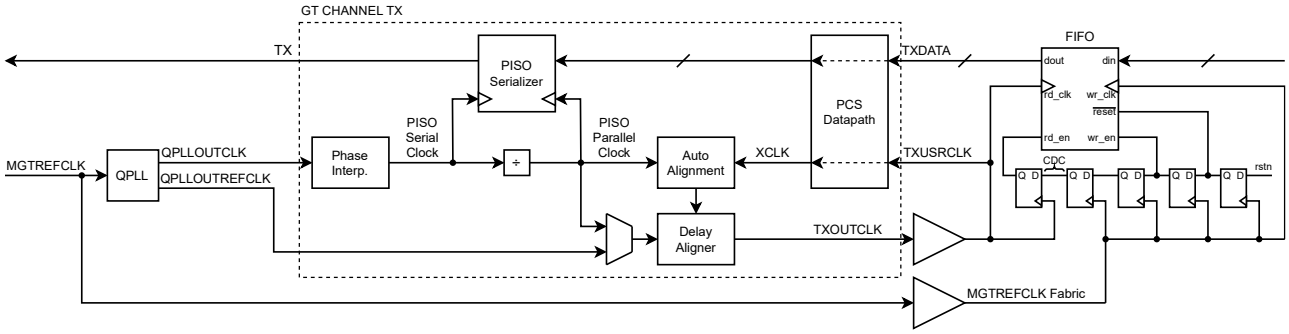

Figure 1: Simplified MGT channel transmitter clocking diagram.

using the Delay Aligner until they are aligned. The algorithm results in TX XCLK and PISO Parallel Clock being phase aligned on opposite edges; as evidenced by the half-cycle latency listed under "To Serializer" in the table of MGT latency values for GTH [5] and GTY [6]. The Delay Aligner is a 256-tap delay chain with a nominal full-scale range of 4 ns. Undocumented registers exist which allow the real-time Delay Aligner tap value to be read using the DMONITOR port (Table 1). The Delay Aligner tap value is output via the lower 8 bits of the DMONITOROUT signal. This allows for an indirect measurement of the phase between PISO Parallel Clock and MGTREFCLK.

Table 1: MGT configuration to read Delay Aligner.

Attribute	Value
TXPH_MONITOR_SEL	0x03
RXPH_MONITOR_SEL	0x03
DMONITOR_CFG1 (read TX tap)	0x05
DMONITOR_CFG1 (read RX tap)	0x03

With this capability the procedure for aligning the phase between MGTREFCLK and PISO Parallel Clock is as follows. First, TXOUTCLKSEL is changed to output TXOUTCLKPCS, which is a copy of PISO Parallel Clock, and the stable Delay Aligner tap value is read and stored. Then, TXOUTCLKSEL is changed to output TXPLLREFCLK_DIV1, which is a copy of MGTREFCLK, and the stable Delay Aligner tap value is read again. If the two measured values are equal within a sub-UI threshold, then the clocks are in phase. If the two measured values differ, then the clocks are not in phase, and PISO Parallel Clock is manually shifted by one UI. This is done by incrementing the 7-bit TXPI value 128 times starting from zero and rolling over at 127 back to zero. The procedure is then repeated until the clocks are measured to be in phase.

The sub-UI threshold is calculated by taking the value of each Dealy Aligner tap to nominally be $\frac{4\ ns}{256} = 15.625\ ps$ which yields eight taps for each 125 ps UI. Four different systems were each reset 1000 times at 30 °C and 60 °C with the average Delay Aligner tap differences recorded (Fig. 2). The experiment was performed using a GTH transceiver which has an added phase uncertainty due to the QPLL output clock being divided by two, with the VCO running at 16 GHz. The delay per tap was observed to vary with PVT. Peaks are visible showing the clock phases spaced by about eight taps or one UI. Small peaks at four tap spacing show half-UI steps caused by the QPLL output divider phase being unaligned with MGTREFCLK; in which case the QPLL is reset.

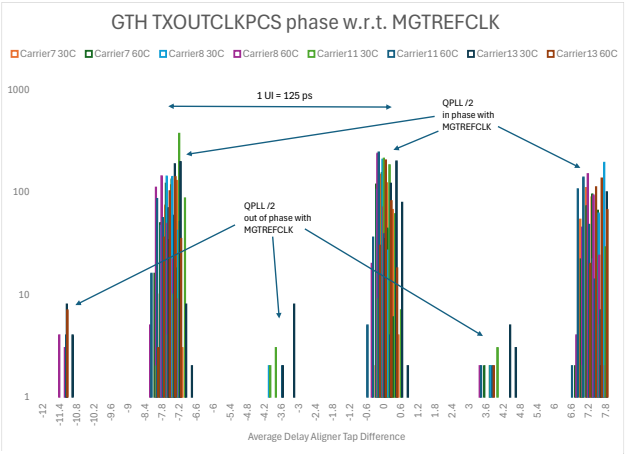

Figure 2: TXOUTCLKPCS w.r.t. MGTREFCLK.

After the manual phase alignment procedure, PISO Parallel Clock will have a deterministic phase relationship with MGTREFCLK and therefore serialized data frames will have a deterministic phase across resets. This procedure also allows for PISO Parallel Clock to be manually phase shifted with respect to MGTREFCLK by modulating the TXPI with a resolution of $\frac{125\ ps}{128} \cong 1\ ps$. This will ultimately be useful for precisely controlling the overall link latency.

Receiver Frame Clock Recovery

On the downstream system the MGT receiver must recover the 100 MHz accelerator master clock from the link to be used as the system clock. The recovered clock from the MGT is output to the dedicated MGTREFCLK pins using the RXRECCLKOUT signal path, see UG576 Figure 4-17 for GTH and UG578 Figure 4-16 for GTY. This clock path does not route through FPGA fabric and therefore has superior phase noise characteristics compared to a clock output via fabric [7]. The recovered clock output is connected to a reference input on an external jitter cleaner Si5345 PLL. The PLL clock output is connected as an input to the other MGTREFCLK pins on the MGT quad. The PLL is configured to free run at 100 MHz when no recovered clock is present to allow for link initialization. After the recovered clock is stable, the PLL phase locks to it and

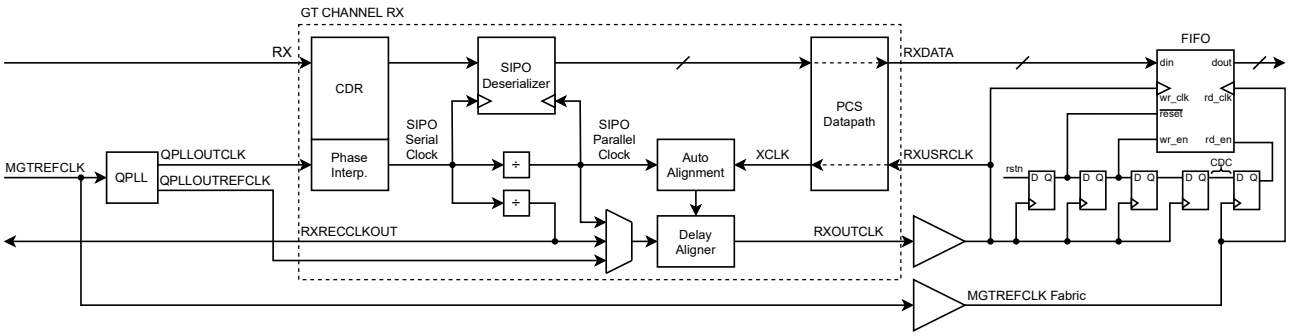

Figure 3: Simplified MGT channel receiver clocking diagram.

MGTREFCLK on the downstream system becomes synchronous with MGTREFCLK on the upstream system. A simplified diagram of the MGT receiver clocks relevant to this application is shown in Figure 3.

The CDR unit in the MGT receiver modulates the RX Phase Interpolator (RXPI) to ensure that SIPO Serial Clock is phase locked to the incoming data stream. SIPO Serial Clock at 8 GHz is divided down by 80 with independent clock dividers to generate SIPO Parallel Clock and RXRECCLKOUT both at 100 MHz. Both clock dividers generate nondeterministic clock phases across resets. The SIPO Parallel Clock phase determines the bit alignment of parallel data words presented to the user logic. The RXRECCLKOUT phase directly determines the phase of MGTREFCLK, the system clock, which must be deterministically aligned between systems.

To achieve deterministic clock phase recovery, RXRECCLKOUT must be in phase with the received serialized data frames. This requires that RXRECCLKOUT also be in phase with SIPO Parallel Clock. The solution involves two steps; first, SIPO Parallel Clock must be aligned with RXRECCLKOUT; second, SIPO Parallel Clock must be manually aligned to data frames in the serial data while preserving alignment with RXRECCLKOUT.

The algorithm for aligning SIPO Parallel Clock with RXRECCLKOUT is similar to the one previously presented for the transmitter. The Delay Aligner phase measurement method is used in the same fashion, see UG576 Figure 4-34 for GTH and UG 578 Figure 4-32 for GTY. First, RXOUTCLKSEL is changed to RXOUTCLKPCS, which is a copy of SIPO Parallel Clock, and the stable Delay Aligner tap value is read and stored. Then, RXOUTCLKSEL is changed to output RXPROGDIVCLK, which is a copy of RXRECCLKOUT, and the stable Delay Aligner tap value is read again. If the two measured values are equal within a sub-UI threshold, then the clocks are in phase. If the two measured values differ, then the clocks are not in phase, and SIPO Parallel Clock is manually shifted. To manually shift SIPO Parallel Clock the RXSLIDE PMA feature is used. Each RXSLIDE assertion causes a shift of SIPO Parallel Clock of one or more UI as determined by the value of the RX D clock divider. The procedure is then repeated until the clocks are measured to be in phase. Test results yielded a greater variability over PVT compared to the transmitter, but enough consistency to clearly distinguish the discrete UI phase jumps (Fig. 4).

Once the two clocks are in phase, SIPO Parallel Clock must still be aligned to a frame, or word boundary, of the serialized data. This is achieved by looking for a comma character in the lowest byte position of the received 64-bit parallel data word. If no comma is received after a certain timeout, a bitslip is issued by shifting both clocks by one UI. To preserve clock phase alignment this is done by momentarily overriding the RXPI value of the CDR. Care must be taken when overriding the CDR to not cause it to lose lock, which will trigger a reset and loss of clock phase alignment.

The current RXPI value can be read from the DMONITOR port, and a new tap value can be written via the DRP port [8]. With the attributes from XAPP1252 applied, omitting RXCDR_CFG2, the CDR controlled RXPI value can be read from the lower 7 bits of the DMONITOROUT signal. The RXPI value can be overridden by writing a new value to the upper 7 bits of the RXCDR_CFG1 register via the DRP port. The new value is subsequently applied by pulsing the RXCDROVRDEN signal high.

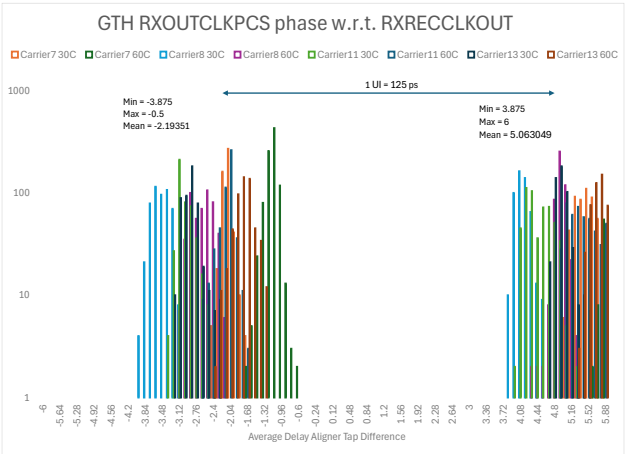

Figure 4: RXOUTCLKPCS w.r.t. RXRECCLKOUT.

The procedure to shift SIPO Serial Clock by one UI for a bitslip is as follows. First, the current RXPI value is read and stored as the initial value. Then, it is incremented and overridden until it rolls over and ends up back at the initial value but shifted by one UI. This must be done rapidly to avoid losing CDR lock, in the actual implementation it is more reliable to increment by four or eight taps per step, for 32 or 16 steps respectively.

With SPIO Parallel Clock simultaneously aligned with RXRECCLKOUT and received data frames,

MGTREFCLK now has a deterministic phase relationship between systems. This assumes that the jitter cleaner PLL output clock phase is deterministic which is achieved by placing the Si5345 in Zero Delay Mode. The downstream system can use the same algorithm previously described to align its transmitted data frames with MGTREFCLK.

Potential Complications

For a concise description of the phase alignment algorithms some complicating details were overlooked which must be addressed. In both transmitter and receiver clocking diagrams the D clock divider shown in the official documentation is omitted. The D clock divider divides the Phase Interpolator (PI) output clock down by a configurable value to generate the Serial Clock. It has a value of two in this application to generate a Serial Clock of 4 GHz. This is necessary because the PISO and SIPO registers operate in Double Data Rate (DDR) mode to transfer data on both rising and falling edges of the Serial Clock. A consequence is that the RXSLIDE PMA mode only allows SIPO Parallel Clock to be shifted in increments of the 4 GHz SIPO Serial Clock, two UI at a time. Therefore, the chance exists that it cannot be manually shifted into alignment with RXRECCLKOUT, in which case the RX Programmable Divider is reset and the alignment algorithm is restarted.

The Delay Aligner tap value fluctuates due to clock jitter and PVT fluctuations, so it must be averaged for several cycles to measure a stable value and achieve the necessary precision to ensure sub-UI clock phase alignment. Another complication arises when reading the Delay Aligner tap value; it must be considered that the maximum delay range is 4 ns which is less than the 10 ns clock period. The Delay Aligner may be unable to compensate for the full clock path latency in which case the tap value saturates high or low and the Auto Alignment procedure fails to complete. This can be handled by delaying TX/RXUSRCLK in fabric by simply inverting it or adding a static phase offset using a PLL or MMCM.

In the official recommended use case for the receiver, this is never an issue because RXOUTCLK is sourced from RXOUTCLKPMA which, contrary to the official documentation, has an independent phase from RXOUTCLKPCS. The receiver Auto Alignment algorithm uses a two-step process by first coarse shifting SIPO Parallel Clock independent from RXOUTCLKPMA before enabling the Delay Aligner for fine phase alignment. Changing RXOUTCLK to RXOUTCLKPCS renders the coarse shifting useless and Auto Alignment becomes fully reliant on the Delay Aligner to compensate for the entire phase offset.

Similarly for the transmitter, the official recommended use case has TXOUTCLK sourced from MGTREFCLK. The Auto Alignment algorithm coarse shifts PISO Parallel Clock using the TXPI before enabling the Delay Aligner. This coarse phase shift mechanism is once again rendered useless when TXOUTCLK is sourced from TXOUTCLKPCS and the Dealy Aligner must compensate for the entire phase offset.

The GTH transceiver presents yet another complication because it only has a 40-bit internal datapath which requires that TX/RXUSRCLK run at 200 MHz and TX/RXUSRCLK2 run at 100 MHz. This creates additional challenges for phase alignment which forces the use of a PLL or MMCM in fabric to generate the 200 MHz clock from the 100 MHz TX/RXOUTCLK.

REPEATABLE DETERMINISM

Deterministic Elastic Buffer Latency

Synchronous clock distribution and recovery with repeatable phase is only one part of ensuring deterministic link latency. It is also necessary for the coarse latency, in integer number of clock cycles, to be deterministic and repeatable. Uncertainly is introduced whenever data words are transferred between clock domains. No clock domain crossing is necessary if data is transmitted in one direction between only two systems, with the receiving device using the recovered clock as its system clock. But if data is to be received and retransmitted for link forwarding, fanout, or full duplex communication; the clock domain must be crossed between the receiver and subsequent transmitter, RXUSRCLK and TXUSRCLK respectively. This is critical for full duplex round-trip link latency measurement where a packet must be received and replied to with deterministic and repeatable latency.

The present solution is to break the clock domain crossing (CDC) into two separate ones which can be independently analysed. The MGTREFCLK clock domain will serve as the primary system clock to which transmit and receive data are both synchronous thus creating a CDC from MGTREFCLK to TXUSRCLK and another from RXUSRCLK to MGTREFCLK. This method also allows for deterministic link fanout or aggregation because multiple channels are all ultimately synchronized to the same MGTREFCLK domain.

With the transceiver in Buffer Bypass mode, a dual clock FIFO is instantiated in FPGA fabric to cross clock domains and serve as an elastic buffer (Fig. 1). If the FIFO has synchronous read and write clocks; never over or underflows; and both read and write enable signals remain asserted; then the latency from the write to the read port will be deterministic. Careful control of the timing path between read and write enable signals after a reset can be used to also ensure repeatable latency. This is only possible if the read and write clock domains are synchronous with a known phase relationship. Due to the clock recovery and phase alignment procedure previously outlined, all three clock domains are synchronous and have a deterministic phase relationship.

The FIFO reset procedure is as follows. First, align the read and write clocks to some nominal phase. Next, reset the FIFO to clear the read and write pointers with read and write enable deasserted. Next, release the FIFO from reset and assert write enable then after a fixed number of cycles assert read enable. The write enable signal is driven by a register on the write clock domain and the read enable signal is driven by a register on the read clock domain. Within

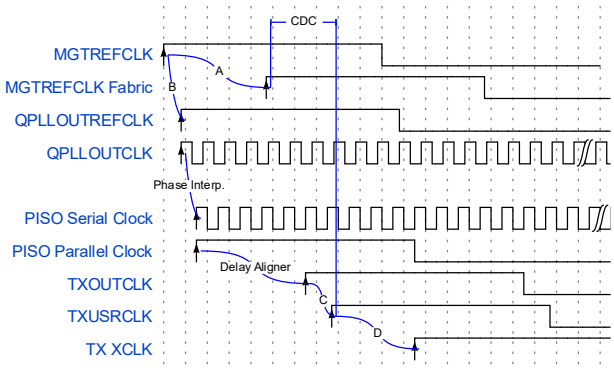

Figure 5: TX clock phase timing diagram.

this logic there is a critical timing path where the write enable signal is passed to the read clock domain. This path must be verified with static timing analysis to ensure a valid CDC when the two clock domains are aligned to their nominal phase.

Transmitter Elastic Buffer

In the case of the transmitter, the write enable signal must be passed from MGTREFCLK to TXUSRCLK requiring these two clocks to be timed together. The phase relationship between them after alignment can be estimated with sufficient precision to ensure deterministic timing. The timing diagram in Figure 5 illustrates the timing of the clocks in Figure 1. The CDC of interest is between MGTREFCLK Fabric and TXUSRCLK.

Delays A, B, and C are static and known to the timing analyser. The delay due to the TXPI, labelled “Phase Interp.,” is dynamic but manually controlled and set to zero for initialization. The delay due to the Delay Aligner is dynamically adjusted by the Auto Alignment algorithm to keep the rising edge of TX XCLK aligned with the falling edge of PISO Parallel Clock. Delay D, internal to the MGT channel, is static and undocumented.

To enable static timing analysis of the path from MGTREFCLK to TXUSRCLK a second clock object is added to the clock pin on the CDC capture flip flop strictly for timing analysis. The new clock object has an inverted waveform with respect to MGTREFCLK and a negative latency which represents how TX XCLK is aligned on opposite edges with PISO Parallel Clock and TXUSRCLK is delayed to TX XCLK. This fixes the timing relationship between both CDC clock domains and remains valid so long as the skew between the MGT channel TXUSRCLK pin and the capture flip flop clock pin is low. This skew and the small unknown delay “D” are accounted for by adding clock latency and uncertainty to the new clock object. With these timing constraints satisfied, the FIFO can be reset and initialized with deterministic latency if the initial CDC clock phase relationships are maintained during reset. After reset the TXPI can be modulated to shift TXUSRCLK phase with respect to MGTREFCLK and the FIFO level will rise and fall accordingly. Deterministic latency will be maintained so long as the FIFO never underflows or overflows.

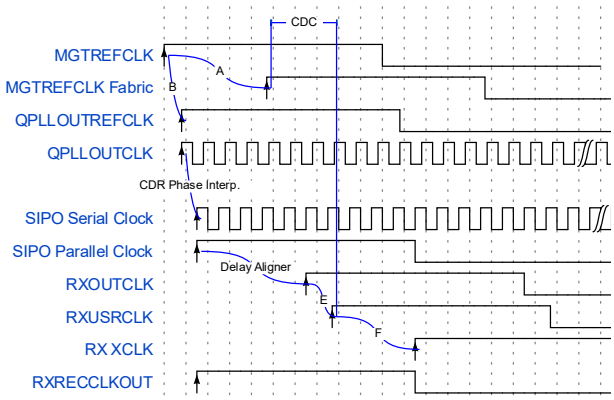

Figure 6: RX clock phase timing diagram.

Receiver Elastic Buffer

Similarly, in the case of the receiver, the write enable signal must be passed from RXUSRCLK to MGTREFCLK. The phase relationship between them after alignment can also be estimated with sufficient precision to ensure deterministic timing. The timing diagram in Figure 6 illustrates the timing of the clocks in Figure 3. The CDC of interest is between MGTREFCLK Fabric and RXUSRCLK.

Delays A and B refer to the same exact signals in both the receiver and transmitter diagrams. Delay E is static and known to the timing analyser. The delay due to the RXPI, labelled “CDR Phase Interp.,” is dynamic and automatically controlled by the CDR to phase lock to the received data. In the steady sate case with synchronous clock recovery, the RXPI will have a static value proportional to the phase offset between MGTREFCLK and RXRECCLKOUT. This phase offset will behave differently on the upstream and downstream systems. The delay from the Delay Aligner is dynamically adjusted by the Auto Alignment algorithm to keep the rising edge of RX XCLK aligned with the falling edge of SIPO Parallel Clock. Delay F, internal to the MGT channel, is static and undocumented.

On the downstream system, where the clock is recovered and routed through the external PLL, the phase between MGTREFCLK and RXRECCLKOUT is fixed by the onboard propagation delay between the FPGA pins and the PLL. On the upstream system, the MGTREFCLK phase is fixed by the external reference, and the phase of RXRECCLKOUT is determined by the full round trip link latency between systems. This results in an arbitrary phase on the upstream system which will be addressed, but for sake of explanation, assume that the two clocks possess some known phase relationship, as on the downstream system.

To enable static timing analysis of the path from RXUSRCLK to MGTREFCLK a second clock object is added to the clock pin on the CDC launch flip flop strictly for timing analysis. The new clock object has an inverted waveform with respect to MGTREFCLK and a negative latency which represents how RX XCLK is aligned on opposite edges with SIPO Parallel Clock and RXUSRCLK is delayed to RX XCLK. This fixes the timing relationship between both CDC clock domains and remains valid so long as the skew between the MGT channel RXUSRCLK

pin and the capture flip flop clock pin is low. This skew, the small unknown delay, F, and the clock latency through the external PLL are accounted for by adding clock latency and uncertainty to the new clock object. With these timing constraints satisfied, the FIFO can be reset and initialized with deterministic latency if the initial CDC clock phase relationships are maintained during reset.

On the downstream system this requirement is satisfied by resetting the RX FIFO after the link is up and the recovered clock is stable. On the upstream system, however, the phase of RXRECCLKOUT and MGTREFCLK must be manually aligned to achieve the same deterministic RX FIFO reset.

LATENCY CONTROL

Coarse Latency Control Using Delay Aligner

The phase of RXRECCLKOUT on the upstream system is determined by the total round trip link latency which will produce an arbitrary phase with respect to MGTREFCLK. The link latency naturally changes due to environmental factors such as temperature and length of the physical media. Alignment of these clocks is achieved by controlling the TXPI to adjust the overall link latency until RXRECCLKOUT is in phase with MGTREFCLK. To measure the phase between RXRECCLKOUT and MGTREFCLK the Delay Aligner phase measurement method is employed similar to how SIPO Parallel Clock was aligned with RXRECCLKOUT. Instead of measuring and shifting by discrete UI phase increments, the RX Delay Aligner is measured and the TXPI continuously modulated with an integrator feedback controller.

The control loop works as follows; RXOUTCLKSEL is changed to RXPLLREFCLK_DIV1, which is a copy of MGTREFCLK. The Delay Aligner tap value is filtered and subtracted from the stable Delay Aligner tap value which was stored earlier from the SIPO Parallel Clock alignment procedure, thus producing an error term. The error term is integrated and scaled to produce the TXPI offset as the control variable. When the error term is small and stable for some amount of time, the clock phases are deemed to be aligned to within a sub-UI threshold, and the control loop switches over to fine measurement mode. The alignment achieved in this step is sufficient to satisfy the timing requirements for a deterministic RX FIFO reset, so at this point during the link initialization, the RX FIFO is released from reset.

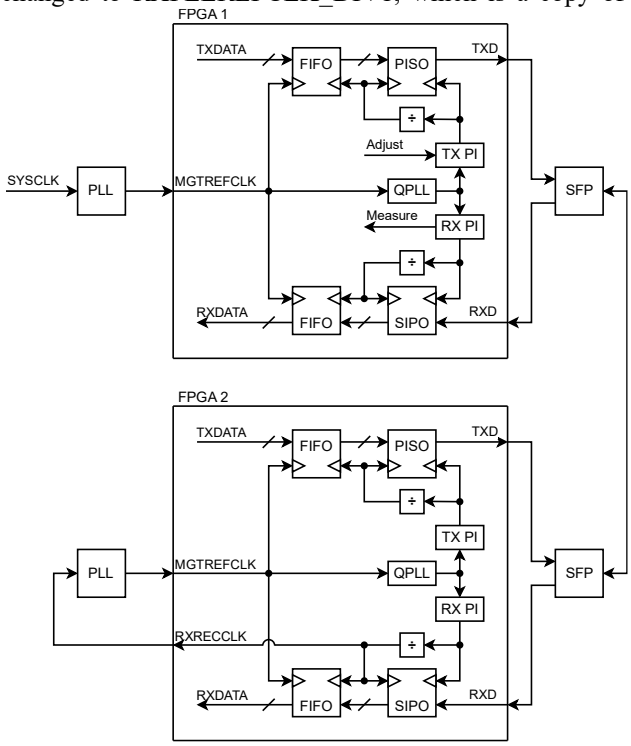

Figure 7: Full duplex latency control link diagram.

The TX FIFO must contain enough entries at initialization to accommodate the full possible swing of the physical media latency. As the media latency increases the TX FIFO level will fall to compensate; and as the media latency decreases the TX FIFO level will increase to compensate, maintaining the initial latency.

Fine Latency Control Using Phase Interpolator

The RXPI value is used as a precise measurement of the link latency. Its resolution is ~1 ps but its measurement range is limited to one UI. To allow for a large measurement range the 7-bit RXPI value is unwrapped and stored as a signed 24-bit number. Unwrapping the measured value is only feasible because it changes much slower than it can be sampled. The unwrapped value must be reset to zero when RXRECCLKOUT is initially coarse aligned with MGTREFCLK. This necessitates the previous step of aligning RXRECCLKOUT to MGTREFCLK within a sub-UI threshold.

An RXPI value of zero represents RXRECCLKOUT in phase with MGTREFCLK. Therefore, the error term to the TXPI latency control loop is switched over to be the RXPI value, unwrapped as a signed number. The gain of the loop is set very low to compensate for long term latency drift without introducing excessive jitter to the link. This enables ~1 ps resolution measurement and control of the overall link latency with deterministic CDC between systems across resets. A full diagram of the link between both systems with latency control is shown in Figure 7.

If the link latency is controlled to keep a constant phase at the receiver, the round-trip link latency will remain constant, but the one-way link latency will still fluctuate. To keep a constant one-way link latency the latency must be adjusted symmetrically on both the upstream and downstream links. This could be achieved by adjusting both upstream and downstream TXPIs to the same value. Instead, to avoid having to adjust the TXPI on the downstream system, the RXPI on the upstream system can be allowed to deviate from zero to compensate by the same amount. The upstream TXPI value will be controlled to be equal and opposite to the RXPI value, in other words, they will sum to zero. This changes the latency control loop error term and results in PISO Parallel Clock and RXRECCLKOUT that are offset from MGTREFCLK by the same magnitude but in different directions. This allows the latency control loop to be localized on the upstream system.

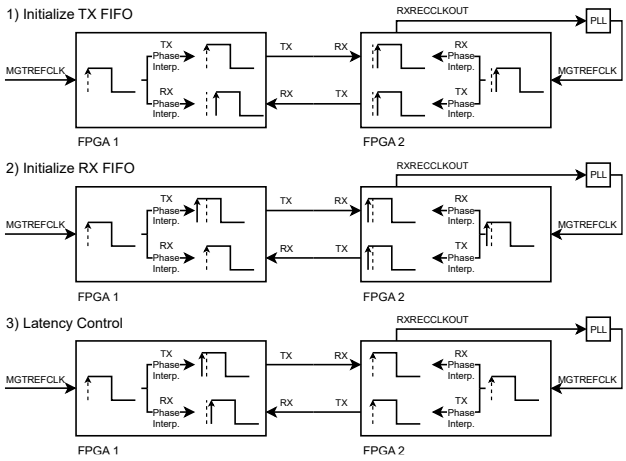

Figure 8: Steps for clock phase alignment.

Latency Measurement

The latency control algorithm used by the upstream system to maintain a deterministic and constant one-way link latency is summarized as follows. First, the transmitter clocks are aligned, and the TX FIFO is reset. Then, once the receiver link is up, the TXPI is modulated to align the receiver clocks, and the RX FIFO is reset. Finally, the TXPI value is controlled to be equal to the measured unwrapped RXPI value. This results in a constant one-way link latency, and a round-trip latency that is an integer number of clock periods. Figure 8 illustrates the relative timing between clocks as the phase is adjusted at each step.

Measuring the round-trip link latency is trivial once the links are up and the FIFOs are reset. A message is sent and echoed back from the upstream system and the number of clock cycles elapsed between transmission and reception is counted. The one-way link latency is simply calculated to be half of the number of clock cycles counted, which assumes perfectly symmetrical upstream and downstream latencies.

If the round-trip latency is measured to be an odd number of cycles, then the one-way latency includes a half cycle. This is compensated for by adding an offset to the TXPI to decrease the link latency by a half clock cycle in the downstream direction. To achieve the offset, an offset equal to an entire clock cycle is summed into the TXPI and RXPI sum which generates the control loop error term. This offset keeps the round-trip latency as an odd number of clock cycles, but the downstream heading one-way latency is reduced by a half cycle, and the upstream heading one-way latency is increased by a half cycle. This way the measured round-trip latency can simply be right shifted and truncated to yield the downstream one-way link latency, while ensuring that it is a whole integer number of clock cycles.

Symmetrical Latency

The one-way latency is calculated to be exactly half of the measured round-trip latency. This method relies on the implicit assumption that the link is perfectly symmetrical with equal latencies in both directions. Any difference in latency will show up as a phase offset between the clocks of each system. Several factors contribute to the latency skew including the PCB trace geometry, SFP optical transceiver, and the fiber optics typically used as the physical transmission medium. In standard pair of OS2 single core fiber, the skew between strands can vary by several picoseconds per meter varying with factors such as temperature, strain, and manufacturing differences [9,10].

One solution is to use wavelength-division multiplexing (WDM) SFP modules which transmit and receive on a single fiber strand with different wavelengths for each direction [11]. The different wavelengths propagate at known different velocities through the medium, which can be compensated for. Alternatively, an optical circulator can be used to combine and split the individual TX and RX optical signals at the same wavelength onto a single fiber for long distance transmission. This allows for the same wavelength to be used for both directions of communication within the same fiber strand, improving latency symmetry. The optical wavelength generated by the SFP module at each end of the link may still vary within some tolerance and therefore change its propagation velocity in each direction.

CONCLUSION

A link layer protocol has been presented which enables single picosecond link latency measurement resolution and deterministic phase repeatability. The alignment scheme to achieve repeatable transceiver clock phases in AMD Ultrascale+ GTH and GTY transceivers requires careful measurement and control during the reset and link initialization procedures. With the clock phases aligned static timing analysis can be used to constrain the CDC between TX and RX clock domains. An elastic buffer implemented in fabric as a dual clock FIFO can be reset with deterministic and repeatable latency. A latency control feedback loop to maintain the one-way link latency is outlined. Single fiber WDM can be implemented to achieve more stable latency control by ensuring symmetric latency in each direction.

REFERENCES

[1] P. Bachek, T. Hayes, J. Mead, K. Mernick, G. Narayan, and F. Severino, “Development of a Timing and Data Link for EIC Common Hardware Platform”, in Proc. ICALEPCS'23, Cape Town, South Africa, Oct. 2023, pp. 228-232. doi:10.18429/JACoW-ICALEPCS2023-MO4AO05

[2] AMD, “UltraScale Architecture GTH Transceivers User Guide UG576 (v1.7.1),” Aug. 18, 2021. https://docs.amd.com/v/u/en-US/ug576-ultrascale-gth-transceivers

[3] AMD, “UltraScale Architecture GTY Transceivers User Guide UG578 (v1.4),” Dec. 19, 2025.

https://docs.amd.com/v/u/en-US/ug578-ultrascale-gty-transceivers

[4] AMD, “AR#73258 - UltraScale GTH PI management,” Sep. 23, 2021 https://adaptivesupport.amd.com/s/article/73258

[5] AMD, “AR#68177 - UltraScale+ GTH Transceiver: TX and RX Latency Values,” Sep. 7, 2022 https://adaptivesupport.amd.com/s/article/68177

[6] AMD, “AR#69011 - UltraScale+ GTY Transceiver: TX and RX Latency Values,” Sep. 7, 2022 https://adaptivesupport.amd.com/s/article/69011

[7] E. Mendes, S. Baron, C. Soos, J. Troska, and P. Novellini, “Achieving picosecond-level phase stability in timing distribution systems with Xilinx ultrascale transceivers,” IEEE Trans. Nucl. Sci., vol. 67, no. 3, pp. 473–481, Mar. 2020.

[8] Xilinx, “Burst-Mode Clock Data Recovery with GTH and GTY Transceivers XAPP 1252 (v1.3),” Apr. 12, 2019. https://docs.amd.com/v/u/en-US/xapp1252-burst-clk-data-recovery

[9] Corning Optical Communcation, “Transmission Skew in Optical Fiber Ribbons,” Apr. 28, 2017. https://www.corning.com/catalog/coc/documents/application-engineering-notes/AEN057.pdf

[10] N. Kashima, “Influence of fiber parameters on skew in single-mode fiber ribbons,” Journal of Lightwave Technology, vol. 15, no. 10, pp. 1858–1864, 1997, doi: https://doi.org/10.1109/50.633574.

[11] E. Brandao, S. Baron, and M. Taylor, “TCLink: A Timing Compensated High-Speed Optical Link for the HL-LHC experiments,” Topical Workshop on Electronics for Particle Physics, p. 057, Mar. 2020, doi: https://doi.org/10.22323/1.370.0057.